

JEDE AUSGABE MIT FIRMIENREGISTER

05/2014 € 6,90

OFFIZIELLER PARTNER VON
GAME
Bundesverband der Computerspielindustrie

making games

DESIGN | BUSINESS | ART | TECHNOLOGY



MAKING OF DEAD ISLAND 2

AMNESIA UND SOMA
SO VERSCHMILZT FRICTIONAL
ERZÄHLUNG UND GAMEPLAY

PROZEDURALES TERRAIN
WIE MATHEMATISCHE FORMELN
RIESIGE WELTEN ERSCHAFFEN

CRYENGINE & HAVOK
DIE MACHER DER ENGINES ÜBER
ZUKUNFTSPÄNE UND NEXT-GEN

animago 2014

AWARD & CONFERENCE



READERS:
USE PROMOTION CODE
2014-animago-making-games
AND GET A 20% DISCOUNT
ON ALL TICKET CATEGORIES!
www.animago.com/tickets

- 16/17 October 2014 +++ Metropolis Hall at Filmpark Potsdam-Babelsberg
- Visual Effects +++ Animation +++ Game Design +++ Interactive Media



VISIT www.animago.com FOR TICKETS AND ALL DETAILED INFORMATION.

Presented by:
**DIGITAL
PRODUCTION**

Funded by:
medienboard 10 JAHRE
Berlin-Brandenburg GmbH

www.animago.com



Images: animago AWARD & CONFERENCE 2013

Sponsored by:



DOSCH DESIGN

FROST
motion design

MAXON



F.N.Y.

TOPALSSON



DIE ZUKUNFT SPRICHT ENGLISCH

Viel logischer hätte sich für Making Games der Kreis nicht schließen können: Zum allerersten Mal veröffentlichen wir mit »The Making of Dead Island 2« (ab S. 12) eine komplett englischsprachige Titelstory. Dabei ist Dead Island 2 zumindest auf den ersten Blick eigentlich ein urdeutsches Projekt – eine Zusammenarbeit des Münchner Publishers Deep Silver mit dem Berliner Entwicklerstudio Yager. Aber eben nur auf den ersten Blick. Am Zombie-Actionspiel arbeiten Developer aus über 15 Nationen, die Büro- und Projektsprache ist logischerweise Englisch. Ob der ambitionierte NextGen-Titel sein Millionenbudget wieder einspielt, entscheidet sich nicht in Deutschland, sondern auf der ganzen Welt. Im Zeitalter des digitalen Self Publishings müssen selbst kleine Indie Studios wie Megagon (S. 58) international denken, agieren und kooperieren, wenn sie langfristig wirtschaften und überleben wollen.

Als wir Making Games vor genau 9 Jahren gründeten, war es ein deutsches Magazin für die deutsche Szene. Selbst Artikel von englischen Teams haben wir damals ins Deutsche übersetzt, was ehrlich gesagt schon 2005 ein ziemlicher Quatsch war. Aber je stärker sich die Branche internationalisiert hat, desto internationaler ist auch Making Games geworden.

Andere Sprache, gleicher Anspruch

Seit 2009 veröffentlichen wir mit Key Players ein englischsprachiges Branchenregister, dessen neueste kostenlose Ausgabe sich jetzt irgendwo in Nähe dieser Zeilen befinden sollte. Seit 2012 belassen wir englischsprachige Artikel im Original, seit August 2013 gibt's Making Games wahlweise auch als englische Tablet-Ausgabe. Seit September 2013 wird auf unserer Facebook-Seite ausschließlich auf Englisch gepostet und – neu – ab September 2014 wird auch unsere Website makinggames.biz komplett englischsprachig sein (S. 6). In diesem Heft finden Sie natürlich trotzdem noch jede Menge deutsche Artikel, zum Beispiel Deck13's Case Study über die Spezialeffekte von Lord of the Fallen (S. 28) oder Nurdogam Erdems interessanten Ansatz für »Tile-based Maps« (S. 49). Die englische Website gibt uns nun aber die Möglichkeit, diese großartigen Arbeiten in übersetzter Form auch einem internationalen Publikum zugänglich zu machen.

Genau wie Yager bei Dead Island 2 sind auch wir bei Making Games überzeugt, dass wir – so viel Eigenlob sei erlaubt – selbst im weltweiten Vergleich durchaus mithalten können. Wird Zeit, das auch zu beweisen!

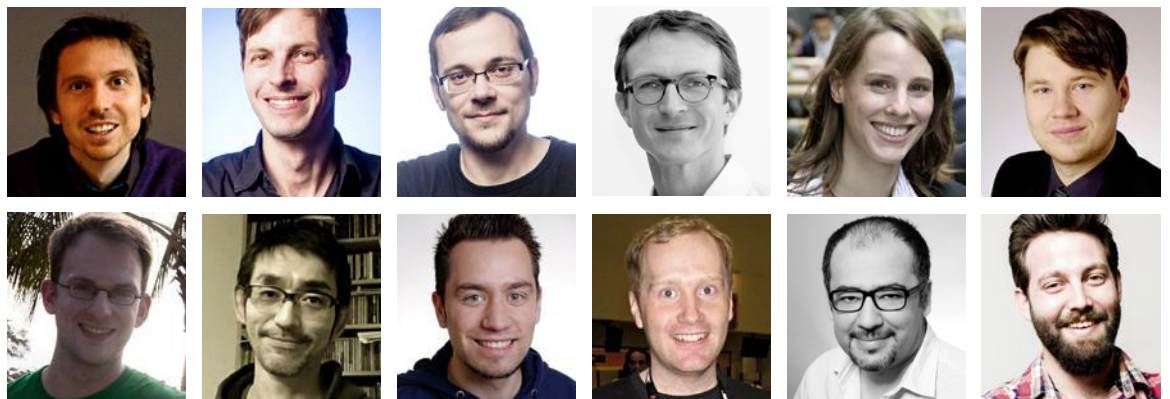
Viel Spaß mit der aktuellen Ausgabe!

das Team von Making Games

Heiko Klinge
ist Chefredakteur vom
Making Games Magazin.



»Ob Dead Island 2 sein Millionenbudget wieder einspielt, entscheidet sich nicht in Deutschland.«

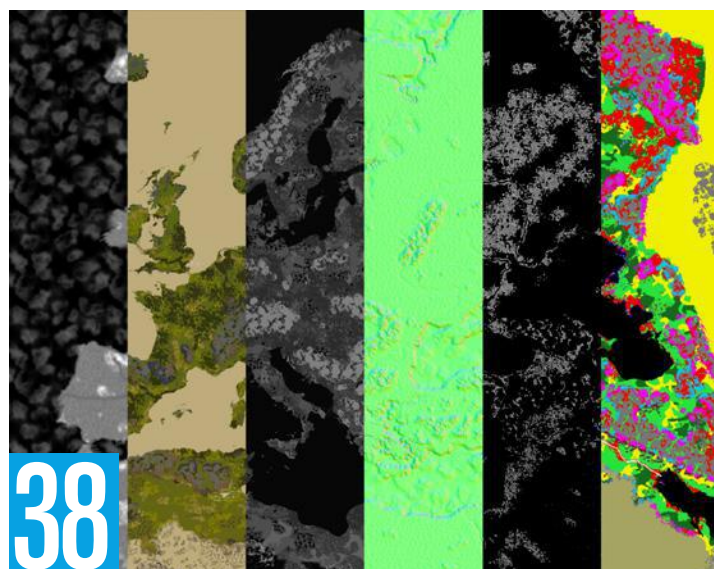
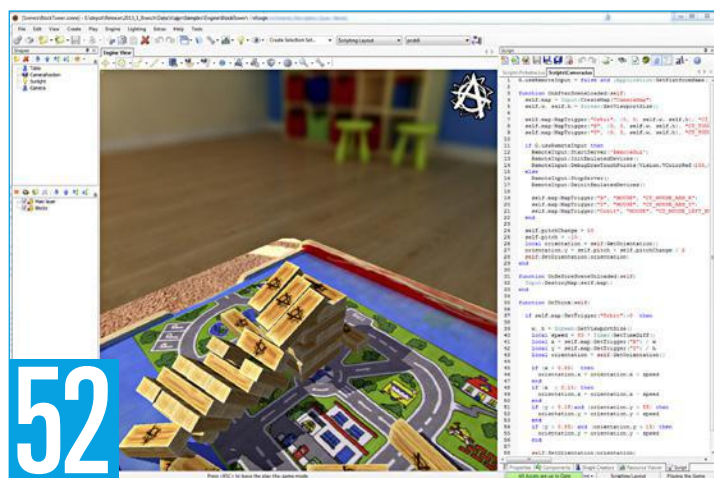
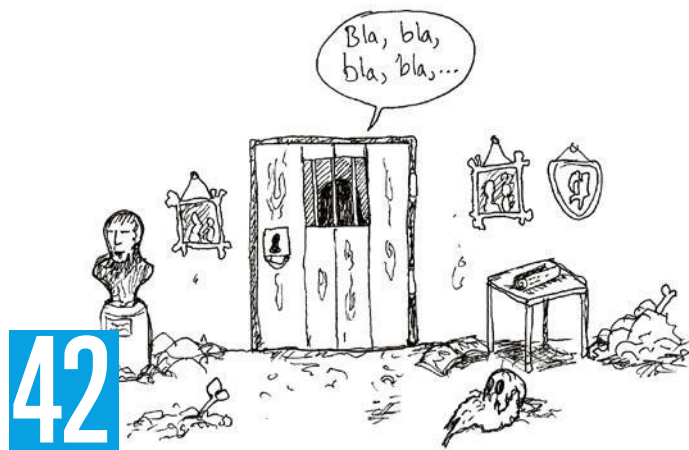


INHALT

05/2014

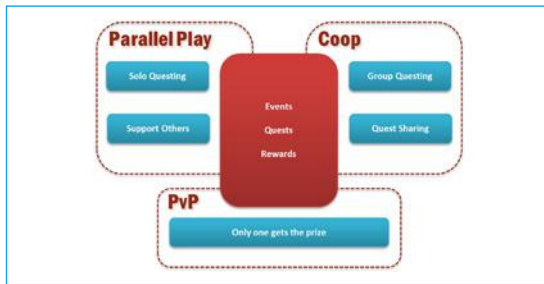
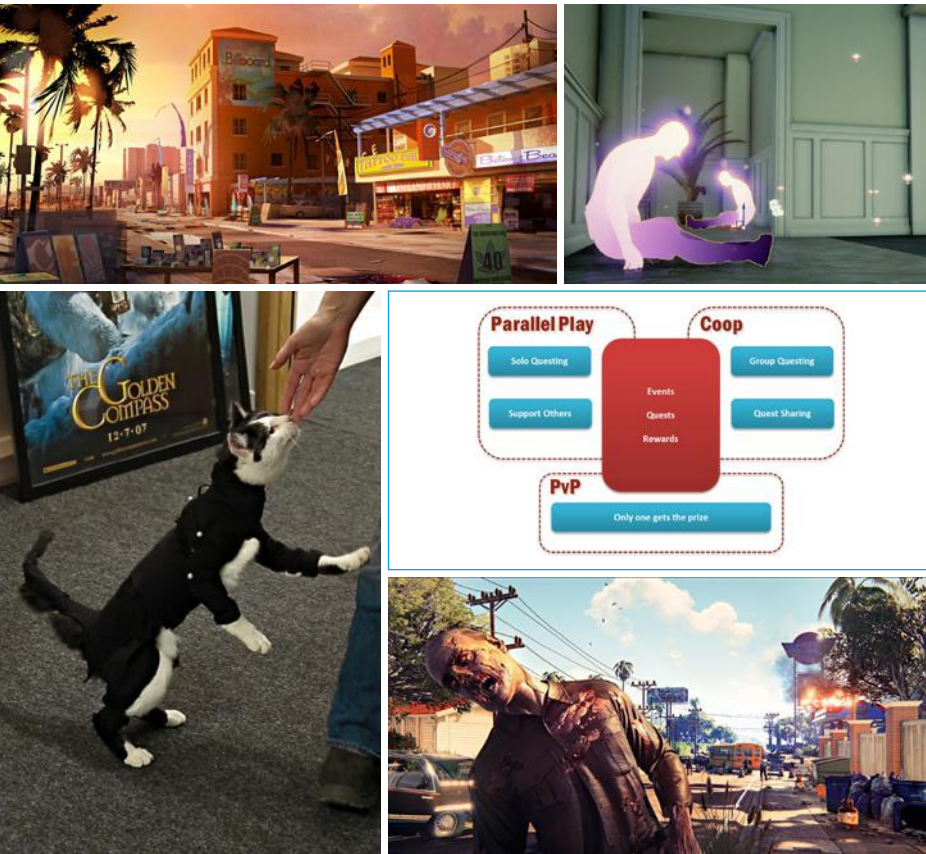
- 03 Editorial
- 80 Ein Tag bei...
Bohemia Interactive
- 82 Vorschau
- 82 Impressum

- 06 Branche / Köpfe
Bemerkenswertes aus der Branche
- 07 Eventkalender
Wichtige Branchenveranstaltungen
- 08 Passiert auf makinggames.de
Interessantes aus dem World Wide Web
- 10 Making Games Tools
Die Meinung der Profis zu aktueller
Soft- und Hardware
- 11 Fokus G.A.M.E.
von Thorsten Unger

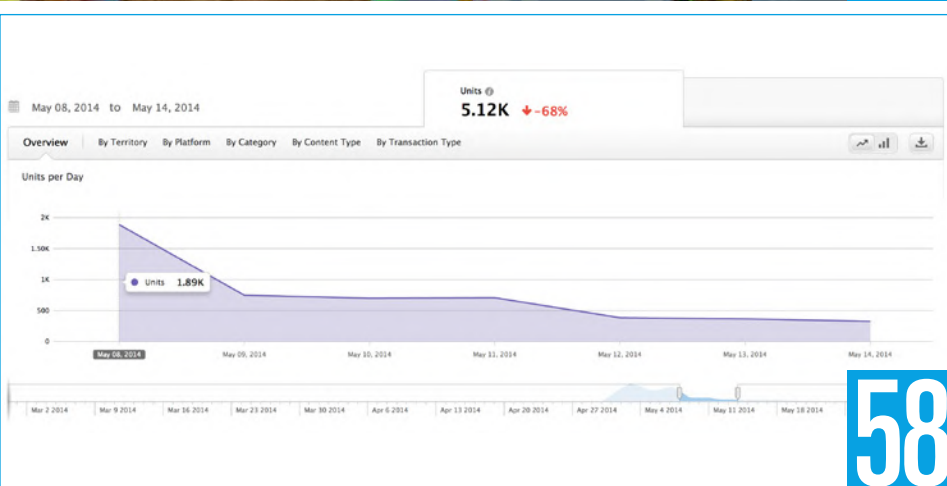


TITELSTORY

Making of Dead Island 2



- 12** **Producing Case Study**
Teaming Up for Expanding a AAA Franchise von Vivien Dollinger, Michael Kempson und Carsten Lindner
- 16** **Game Design Case Study**
Designing the World's Smallest MMO von Bernd Diemer und Jan-Eric Laube
- 20** **Graphics Case Study**
Building an Open World with Unreal Engine 4 von André Dittrich
- 24** **Interview mit Michael Kempson und Carsten Lindner**
The Challenges of Next-Gen Development von Sebastian Weber



- 28** **Case Study**
Partikeleffekte für die nächste Generation von Benjamin Glatzel, Dominik Lazarek und David Reinig
- 36** **Interview mit Yasuhiro Wada**
Der Konsolenmarkt als Nische von Patricia Geiger
- 38** **Case Study**
GPU-unterstützte Generierung des Terrains für das Spiel Grand Ages: Medieval von Mark Külker
- 42** **Best Practice**
Four Layers – A Narrative Design Approach von Thomas Grip
- 49** **Programmierung**
In die Karten geschaut von Nurdogan Erdem
- 52** **Interviews mit Carl Jones und Andrew Bowell**
Die neue Engine-Generation: Crytek & Havok von Patricia Geiger und Sebastian Weber
- 58** **Post Mortem**
... And Then It Rained – A Bittersweet Success Story von Daniel Helbig
- 62** **Case Study**
Vier Plattformen, ein identisches Spiel von Jan Reichert
- 66** **Sound Post Mortem**
Interaktives Sounddesign für The Last Tinker: City of Colors von Filippo Beck Peccoz
- 73** **Marktforschung DataFlow**
GameStar-/GamePro-Leserdaten von Patricia Geiger
- 74** **Firmenregister**
Die Branche im Überblick

58

AUS MAKINGGAMES.DE WIRD MAKINGGAMES.BIZ!

Making Games gibt's als schickes Print-Magazin mit einem modernen Layout, als interaktive zweisprachige Tablet-App ... und als acht Jahre alte Website. Eines davon passt nicht. Und das muss sich schleunigst ändern, sonst fällt unsere Online-Präsenz noch unter Denkmalschutz!

Rund 18 Monate schrauben wir nun schon am überfälligen Relaunch unserer Website, und so langsam biegen wir auf die Zielgerade ein. Irgendwann im Verlauf der nächsten zwei Monate werden Sie jedenfalls ein ungewohntes Bild erblicken, wenn Sie – wie hoffentlich jeden Tag – Ihre Lieblingsseite makinggames.de öffnen. Es wird schicker sein und deutlich näher am ebenso bewährten wie beliebten Magazin-Layout. Lange schwarze Texte auf weißem Grund

ohne Multimedia-Schnickschnack, Flash-Attacken und Clickbait-Schweinereien. Wie bei allem machen wir auch online genau das, was wir selbst am liebsten lesen und täglich besuchen würden.

Manches wird Ihnen dabei vielleicht nicht Spanisch, aber doch zumindest Englisch vorkommen. Denn ab dem Relaunch werden wir nur noch englischsprachige News und Features online stellen, deutsche Artikel werden entsprechend übersetzt. Konsequenterweise ändern wir auch unsere Web-Adresse von makinggames.de in makinggames.biz.

Für uns ist das der letzte fehlende Schritt auf dem Weg zu einer internationalen Game Developer Plattform. Schon jetzt stammen 70% unserer App Downloads aus dem Ausland, unsere englischsprachige Facebook-Seite ist eine der größten europäischen Game Developer Communities. Und genau diese Internationalität wollen wir auch mit unserer neuen Website erreichen. Wir möchten, dass unsere und Ihre Themen künftig auf der ganzen Welt gelesen und diskutiert werden.



News und Artikel sind natürlich nur der Anfang, auch der Eventkalender, unser Firmenregister (inklusive Key-Players-Integration) sowie – Tusch – eine nigelgelagelte Stellenbörse werden in den kommenden Wochen auf makinggames.biz ihr neues Zuhause finden. Das alles werden wir Ihnen in der nächsten Ausgabe selbstverständlich noch ausführlicher vorstellen. Drücken Sie uns die Daumen, dass beim Umzug alles glatt geht! Und schauen Sie doch ab und zu mal vorbei in unserem neuen Online-Zuhause, wir freuen uns auf Sie! **Heiko Klinge**

WWW.makinggames.de
aktuelle News und Gerüchte

COMPUTEC MEDIA ÜBERNIMMT ARUBA EVENTS



Mitte Juli gab das Medien-Unternehmen Computec Media aus Fürth bekannt, die Mehrheit an Kongressveranstalter Aruba Events zu übernehmen. Aruba Events wurde 2003 gegründet und hat sich bislang unter anderem mit Veranstaltungen wie der Entwicklerkonferenz Quo Vadis in Berlin, dem Deutschen Entwicklerpreis in Essen oder dem European Games Award einen Namen gemacht.

Stephan Reichart, bisheriger und künftiger geschäftsführender Gesellschafter von Aruba

Events, sieht in Computec Media den richtigen Partner, um sich künftig verstärkt auch auf internationale Veranstaltungen auszurichten. Gerade die redaktionelle Erfahrung sowie die guten Kontakte in die Games-Branche sollen die Weiterentwicklung der Events vorantreiben.

Für Computec Media stellt die Zusammenarbeit laut Geschäftsführer Hans Ippisch eine ideale Erweiterung der Geschäftsfelder dar. Ippisch nimmt fortan die Rolle des Geschäftsführers bei Aruba Events ein.

CRYTEK ZIEHT DIE NOTBREMSE

Nachdem Crytek wochenlang zu den Gerüchten rund um eine drohende Insolvenz geschwiegen hatte, meldeten sich die Frankfurter Ende Juli zu Wort und bestätigten die finanziellen Probleme. So befände sich Crytek laut offiziellem Statement in einem Umbruch, weg von Retail-Produkten hin zu Online-Services und der Etablierung als Online-Publisher. Jedoch solle der dafür nötige Kapitalbedarf inzwischen gedeckt sein.

Einige Tage später verkündete Crytek, dass Publisher Koch Media die Marke »Homefront« mit samt Cryteks Studio in England aufkaufen würde. Die vorherigen Crytek-Mitarbeiter, die angeblich monatelang keinen Lohn erhalten haben sollen, könnten unter der Flagge der neugegründeten Dambuster Studios weiter am Ego-Shooter arbeiten. Crytek wolle zudem die Mannschaft in seiner Filiale in Austin, Texas deutlich verkleinern und von dort aus nur noch Support für Engine-Lizenznehmer in den USA leisten. Mitarbeiter, die Kündigungen fürchten müssen, könnten sich für offene Stellen in Frankfurt bewerben. Die Zukunft der Crytek-Studios in Shanghai und Seoul sei noch ungewiss; in Frankfurt, Budapest, Istanbul, Kiev und Sofia würde der Betrieb dagegen wie gehabt weiterlaufen.



Eventkalender

AUGUST 2014

Unite 2014

Ort Seattle, Marion Oliver McCaw Hall
Termin 20. bis 22. August 2014
Thema Konferenz
Kosten 79 bis 264 Euro
Homepage unity3d.com/unite/unite2014

SEPTEMBER 2014

IFA

Ort Berlin, Berlin Expo Center City
Termin 05. bis 10. September 2014
Thema Messe
Kosten 8 bis 35 Euro
Homepage b2c.ifa-berlin.de

Reeperbahn Festival

Ort Hamburg, Spielbudenplatz
Termin 17. bis 20. September 2013
Thema Festival
Kosten 24 bis 89 Euro
Homepage www.reeperbahnfestival.com

Tokyo Game Show

Ort Makuhari Messe
Termin 18. bis 21. September 2013
Thema Messe
Kosten 8 bis 22 Euro
Homepage tgs.cesa.or.jp

Ludicious - Zürich Game Festival

Ort Zürich, Kasernenareal
Termin 18. bis 21. September 2014
Thema Festival
Kosten Kostenlos
Homepage www.ludicious.ch

EGX London

Ort London, Earls Court
Termin 25. bis 28. September 2014
Thema Messe
Kosten 13 bis 25 Euro
Homepage www.egxlondon.net

Avni Yerli
ist Co-Founder &
Managing Director
von Crytek.



Vier Fragen an ...

Making Games Haltet ihr nach wie vor am ursprünglich geplanten Wandel zum Entwickler und Publisher von Free2Play-Titeln fest, wie ihn Cevat vor Jahren angekündigt hat, oder war diese Entscheidung rückblickend der Grund für die finanzielle Schieflage?

Avni Yerli In »Warface« haben wir heute weltweit rund 30 Millionen registrierte User und seit der Veröffentlichung auf Steam vor wenigen Wochen ist noch einmal eine weitere Million dazu gekommen. Insofern sind wir mit der Performance von »Warface« zufrieden, aber sehen gleichzeitig – insbesondere in Europa und Nordamerika – noch viel Potenzial. Wir werden uns auch weiterhin auf Online-Game-Services konzentrieren, die wir entweder selbst oder über Partner betreiben. Das Konzept ist aufgegangen, denn »Warface« sorgt für konstante und planbare Einnahmen.

Making Games AAA-Spiele, wie etwa »Ryse 2« zu dem es Gerüchte gab, dass ihr mit Microsoft in Verhandlung wärt, sind also keine Option mehr?

Avni Yerli »Ryse 2« wurde weder von Microsoft noch von uns offiziell angekündigt. Von daher werden wir hierzu auch nicht weiter Stellung nehmen und hoffen auf Verständnis.

Making Games Mit welchen Projekten wollt ihr in diesem Geschäftsjahr euer Budget sichern bzw. stabilisieren?

Avni Yerli Wir fokussieren uns nach den bereits angekündigten strukturellen Änderungen auf unser Engine-Geschäft, den Betrieb von »Warface« und die Fertigstellung von »Arena of Fate« und »Hunt: Horrors of the Gilded Age«. Hiermit sehen wir uns gut aufgestellt für die Zukunft.

Making Games Welchen Einfluss werden die neuen Geldgeber künftig auf Crytek und eure Arbeit haben, die euch unter die Arme gegriffen haben, um den Kapitalbedarf zu decken?

Avni Yerli Die Annahme, wir hätten den Kapitalbedarf durch einen Investor gedeckt, ist falsch. Wir könnten unseren Kapitalbedarf durch Einnahmen decken die wir erwirtschaftet haben. Durch die Fokussierung unseres Kerngeschäfts – Entwicklung von Inhalten und Technologie – sowie durch die hohen Investitionen in neue IPs der vergangenen Jahre und die ergriffenen Strukturmaßnahmen sehen wir einer positiven Geschäftsentwicklung entgegen. Wir haben in 2012/2013 deutlichen Zuwachs bei Umsatz und Gewinn verzeichnet und das wird auch für 2013/2014 gelten.

OKTOBER 2014

Brasil Game Show

Ort São Paulo, Expo Centre Norte
Termin 08. bis 12. Oktober 2014
Thema Messe
Kosten 13 bis 100 Euro
Homepage www.brasilgameshow.com

animago AWARD & CONFERENCE

Ort Potsdam-Babelsberg, Metropolis Halle
Termin 16. bis 17. Oktober 2014
Thema Konferenz
Kosten 48 bis 98 Euro
Homepage www.animago.com

www.makinggames.de

■ Online-Eventkalender mit Suchfunktion und Veranstaltungsinfos

Passiert auf makinggames.de

Microsoft entlässt 18.000 Angestellte Microsoft

Microsoft richtet sich unter CEO Satya Nadella verstärkt auf die Bereiche Mobil und Cloud aus. Nadella hat in einem Schreiben an alle Angestellten nun bekanntgegeben, dass diese Umstrukturierung zu einer Streichung von rund 18.000 Stellen innerhalb eines Jahres führen wird. Die Zusammenführung von Nokia und Microsoft wird allein rund 12.500 Arbeitsplätze kosten.

Schon jetzt will Microsoft anfangen, insgesamt 13.000 Positionen zu streichen. Nadella verspricht, den Prozess so offen wie möglich zu gestalten, bei der Suche nach neuen Jobs zu helfen und Abfindungen zu bezahlen. Die jeweiligen Abteilungsleiter werden die Mitarbeiter über die geplanten Schritte informieren. Laut Nadella sollen durch die Streichungen die Managementebenen reduziert und so Entscheidungen beschleunigt werden. Auch die Teams bei Microsoft sollen besser und mit mehr Vertrauen zusammenarbeiten.

Im Rahmen der Umstrukturierung hat Microsoft auch die Schließung der Xbox Entertainment Studios bestätigt. Das Studio war hauptsächlich für die Produktion von Film- sowie TV-Inhalten für die Xbox-Marke beauftragt. Aktuell noch in der Produktion befindliche Projekte sollen nicht beeinflusst sein. Dazu zählt beispielsweise die Live-Action-Serie »Halo: Nightfall«. Bei den Xbox Entertainment Studios sind derzeit 175 Mitarbeiter beschäftigt. Die Schließung des Studios soll laut Microsoft innerhalb der nächsten Monate über die Bühne gehen. Ein konkreter Termin ist bisher noch nicht bekannt.

Volition plant 100 Neueinstellungen

Das unter anderem durch die »Saints Row«-Reihe bekannte Entwicklerstudio Volition plant eine massive Vergrößerung seines Standortes in der amerikanischen Universitätsstadt Champaign. Angeblich hat das Unternehmen die Stadtverwaltung um eine Investition in Höhe von 200.000 Dollar gebeten, um die eigenen Büros in der Innenstadt neu anordnen und erweitern zu können. Offenbar plant Volition die Einstellung von bis zu 100 neuen Mitarbeitern. Sollte die Stadt den Antrag bewilligen, würde das eine Vergrößerung des Entwicklerteams um satte 50 Prozent bedeuten. Derzeit arbeiten rund 200 Angestellte für das Studio.

Allerdings hat Volition noch weitere Forderungen gestellt. Unter anderem soll die Stadtverwaltung 1.000 Dollar Prämie für jeden neuen Angestellten bewilligen und zudem für eine günstige Erweiterung der Büro-Parkplätze sorgen. Sollte den Anträgen nicht stattgegeben werden, zieht Volition aus Kostengründen einen Umzug in Betracht. Volition wurde nach der Insolvenz des Publishers THQ im vergangenen Jahr an Deep Silver verkauft. Zuletzt veröffentlichte das Studio »Saints Row 4«, aktuell arbeitet das Team am 2015 erscheinenden »Mad Max«.



World of Warcraft weiterhin umsatzstärkstes Abo-Spiel

Die Marktforschungs-Plattform SuperData hat eine Liste mit den zehn umsatzstärksten Onlinespielen mit Abo-Modell für 2013 erstellt. Für die Zahlen beruft sich SuperData auf die Auswertung der digitalen Käufe von fast 40 Millionen Spielern sowie Angaben von Entwicklern, Publishern und Online-Bezahlssystemen. SuperData kommt zu dem Ergebnis, dass »World of Warcraft« auch letztes Jahr der Marktführer unter den Abo-MMOGs war. Der Titel habe über eine Milliarde Umsatz generiert und beanspruche damit einen Marktanteil von 36 Prozent. SuperData Liste bezieht sich auf die Umsätze weltweit. Den zweiten Platz belegt der Klassiker »Lineage«, der vor allem in China und Südkorea noch sehr populär ist. Auch Titel die sich kostenlos spielen lassen aber über optionale Abo-Angebote verfügen, werden in der Liste berücksichtigt. Laut SuperData ist der Gesamtumsatz im Abo-Markt seit 2011 rückläufig. Die Marktforschungs-Plattform führt allerdings an, dass sie eine »Stabilisierung des Abo-Marktes im Laufe der nächsten Jahre« erwarte, »während die Free2Play-Alternativen eine Sättigung erreichen«.



Die Top 10 der umsatzstärksten Abo-Spiele 2013

1. **World of Warcraft** (Activision/ Blizzard): _____ \$1.041 Millionen (36%)
2. **Lineage 1** (NCsoft): _____ \$253 Millionen (9%)
3. **TERA** (NHN): _____ \$236 Millionen (8%)
4. **Star Wars: The Old Republic** (EA): _____ \$165 Millionen (6%)
5. **Der Herr der Ringe Online** (Turbine): _____ \$104 Millionen (4%)
6. **EVE Online** (CCP): _____ \$93 Millionen (3%)
7. **Aion** (NCsoft): _____ \$88 Millionen (3%)
8. **Blade & Soul** (NCsoft): _____ \$65 Millionen (2%)
9. **Lineage 2** (NCsoft): _____ \$45 Millionen (2%)
10. **RIFT** (Trion): _____ \$36 Millionen (1%)

Quelle: SuperData

Großbritannien stoppt Strafverfolgung von Internet-Piraterie



Nachdem in Großbritannien kürzlich noch über ein härteres Vorgehen gegen Medien-Piraterie diskutiert wurde, entkriminalisiert der VCAP-Entwurf nun Filesharing. Ab 2015 werden lediglich vier Mahnungen an File-Sharer verschickt, in denen die Gesetzeswidrigkeit klargestellt wird, weitere Schritte gibt es nicht. Das Voluntary Copyright Alert Programme (VCAP) ist das Ergebnis langer Verhandlungen zwischen Internet-anbietern, Politikern und der Musik- und Videoindustrie. VCAP soll geplante Anti-Piraterie-Maßnahmen ersetzen, zum Beispiel die dauerhafte Trennung der Internetverbindung und eine File-Sharer-Datenbank im Stil einer Blacklist.

Laut einem Sprecher des britischen Musikindustrieverbandes BPI sollen so die Belehrbaren überzeugt werden, zum Beispiel Eltern, die nicht wüssten, was mit ihrer Internetleitung passiere. VCAP sei kein Mittel, um den Internetzugang einzuschränken, sondern die Einstellung und das Bewusstsein dafür zu schärfen, was das Richtige ist.



Ex-Infinity-Ward-Mitarbeiter gründen Reload Studios

Nach Respawn Entertainment folgt mit Reload Studios bereits die zweite Neugründung eines Studios von ehemaligen Entwicklern der »Call of Duty«-Reihe. Zu den fünf Gründungsmitgliedern gehört James Chung, der für Infinity Ward an der Marke mitarbeitete. Außerdem mit dabei sind Taehoon Oh, der an der Entwicklung von insgesamt sechs Serien-Ablegern beteiligt war, Pete Blumel, einem der für »Modern Warfare« und »Ghosts« verantwortlichen Produzenten und Hougan Chen, der nicht nur bei der Fertigstellung verschiedener CoD-Ableger mithalf, sondern auch für Chaos Studios an »Homefront« arbeitete. Komplettiert wird das Quintett durch Nik Ranieri, einem ehemaligen Disney-Trickzeichner, der unter anderem am Zeichentrickfilm »Arielle, die Meerjungfrau« beteiligt war. Das Studio arbeitet derzeit an einem First-Person-Shooter für PlayStation 4, Xbox One, mobile Endgeräte und Virtual-Reality-Headsets. Eine Veröffentlichung des Spiels ist für 2015 vorgesehen.

Daedalic gründet neues Studio in Düsseldorf



Daedalic Entertainment hat ein zweites Entwicklerstudio in Düsseldorf gegründet. Das neue Studio namens Daedalic Entertainment Studio West ist ein Tochterunternehmen der Daedalic Entertainment GmbH, die als Hauptgesellschafterin an dem Unternehmen beteiligt ist. Weitere Gesellschafter sind Andreas Suika, der als Creative Director die Entwicklungsarbeiten in Düsseldorf leiten wird, sowie Dirk Steenpaß, der als Technical Director die technische Seite der Produktionen verantwortet. »Mit der Gründung der Daedalic Entertainment Studio West GmbH setzen wir unsere Strategie fort, Spiele von hoher Qualität zu entwickeln. Es gibt viele inhaltliche Parallelen, doch am neuen Standort werden weniger Adventures, sondern vorwiegend Spiele in anderen Genres produziert – was genau, verraten wir bald«, sagt Geschäftsführer Carsten Fichtelmann zur Studiogründung.



Oculus VR stoppt Lieferung an Ebay-Wiederverkäufer

Oculus VR hatte schon die Lieferungen nach China wegen Wiederverkäufern eingestellt, nun geht das Unternehmen auch gegen Besteller vor, die das zweite Entwicklerkit der Oculus Rift auf Ebay verkaufen wollen. Ein Nutzer des Forums von Oculus VR hatte auf Ebay bemerkt, dass dort ein Verkäufer Vorbestellungen für das Development Kit 2 (DK2) von Oculus Rift annimmt und dafür 499 Britische Pfund verlangt – umgerechnet rund 630 Euro beziehungsweise über 850 US-Dollar. Im Forum von Oculus Rift meldeten sich sowohl der Community Manager unter dem Alias »cyboreality« als auch der für den Kundenservice zuständige »Twistgibber« und erklärten, dass man die Bestellungen dieser Ebay-Verkäufer gefunden und storniert habe. Außerdem wies der Community Manager darauf hin, dass Verkäufe der Entwicklerkits nicht erlaubt seien und das Unternehmen für Oculus Rift nur dann Garantieleistungen übernehme, wenn diese vom ursprünglichen Käufer eingefordert werden. »Wenn ihr also auf Ebay kauft, dann seid ihr allein«.



Chris Allcock räumt mit Irrtümern über die Arbeit in der Videospielebranche auf.
www.makinggames.de/10-misconceptions

Der 12-jährige Sam Smith hat mit etwas Hilfe seines Vaters ein eigenes Spiel entwickelt.
www.makinggames.de/spacepants

Im Video von PixelStormFilms wird deutlich, weshalb man seinen PC nicht unbeaufsichtigt lassen sollte.
www.makinggames.de/takeover

Ron Gilbert hat 25 Jahre alte Design-dokumente von »Maniac Mansion« auf seinem Blog veröffentlicht.
www.makinggames.de/maniac-mansion



BRANCHEN-GEZWITSCHER @MAKINGGAMES_DE/INTERESTING-PEOPLE



@GEORGE3DR
It's 2014. A Quakecon exclusive Doom 4 reveal is pure nonsense.

George Broussard, Erfinder von Duke Nukem



@ROCKET2GUNS
So an e-sport wants to be like »real« sports so bad, that we have to copy all the discrimination and testosterone madness as well? Good luck
Dean Hall, Erfinder von Day Z



@THA_RAMI
I am not sure what the potato salad Kick-starter means for crowdfunding, humanity, consumerism or opportunity, but it sure is amusing.
Rami Ismail, Mitgründer von Vlambeer



@THEREALCLIFFYB
Only one troll in my AMA earlier what is happening?!

Cliff Bleszinski, Lead Designer von Unreal Tournament



@DUVALMAGIC
Random fact: Borderlands: The Pre-Sequel has a 2 to 1 female to male player character gender ratio – plus one robot for the AI demographic!
Randy Pitchford, Präsident von Gearbox Software



@NOTCH
I woke up in the middle of the night and realized my life was still incomplete. How many years has it been? Is there still hope? λ³
Markus Persson, Minecraft-Erfinder



@BENJAMINCOUSINS
I think there's a potential market out there for free-to-play games that get harder when you spend money
Ben Cousins, Independent Videogames Consultant



@MIKEBITHELL
No amount of entrepreneurial buzz phrases can save any of us from the reality that we will not create hit after hit. No human ever has.
Mike Bithell, Schöpfer von Thomas Was Alone



@LARALYN
The years have trained me. Group lunch, dev conference. Woman talks about makeup. I think »Don't do that if you want to be taken seriously.«
Laralyn McWilliams, Creative Director bei The Workshop

JOIN THE CONVERSATION

TOOLS

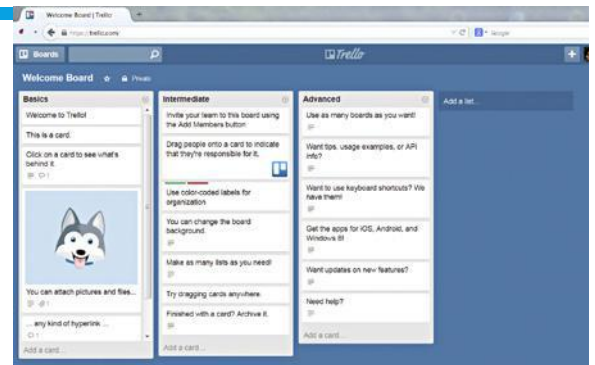
Keine guten Werkzeuge – keine guten Spiele! Experten aus der Branche sagen ihre Meinung zu aktuellen Programm-Updates sowie neuen Tools und verraten ihre persönlichen Software-Geheimtipps.

TRELLO  WWW.TRELLO.COM



Ralf C. Adam
ist Geschäftsführer
bei Virgin Lands.

Online-Programme, mit denen sich die aus dem Agile-Development bekannten Kanban-Boards erstellen, verwalten und teilen lassen gibt es inzwischen massig. Meine liebste Variante ist aber Trello von Fog Creek Software. Das webbasierte Programm ist schnell, übersichtlich, einfach zu bedienen und bietet alle Funktionen, die man für kleine bis mittelgroße Projekte benötigt. Wir haben damit sogar in Teilbereichen wie dem Modelling und Texturing die Erstellung aller Assets für einen kompletten Kinofilm gesteuert. In den Boards kann man beliebig Listen für einzelne Bereiche anlegen und diese mit Karten füllen, die sich per Drag&Drop verschieben lassen. Inzwischen gibt es auch eine Trello-App für iOS und Android, so dass man auch unterwegs jederzeit über den aktuellen Projektstand informiert ist.



Mit dem webbasierten Programm Trello lassen sich bequem Kanban-Boards erstellen. Darin kann der Nutzer beliebig viele Listen anlegen, die man bequem per Drag&Drop verschieben kann.

TUMULT HYPE  [HTTP://TUMULT.COM/HYPE](http://TUMULT.COM/HYPE)



Tumult Hype dient dazu, auf HTML5-Basis animierte und interaktive Inhalte ohne Flash zu realisieren.

Tumult-Hype wurde von zwei Apple Mitarbeitern entwickelt, um auf HTML5-Basis animierte und interaktive Webinhalte schnell, einfach und ohne Flash realisieren zu können. Die Bedienung erfordert keine Programmierkenntnisse und basiert auf Keyframe Animationen mit vielen nützlichen Einstellungsmöglichkeiten. Das Programm lässt sich intuitiv bedienen und ist zusätzlich auf der Homepage gut dokumentiert. Die Animation läuft auf Desktops, Smartphones und iPads und auch hier gilt die Regel »What you see is what you get«. Tumult Hype ist im App Store für 30\$ erhältlich. Eine nützliche Ergänzung ist die kostenlose App »Hype Reflect«, mit der man sein Ergebnis direkt auf dem iPad anschauen kann.



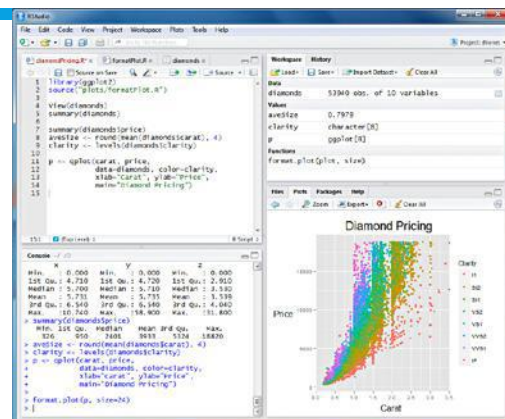
Marc Sodermanns
ist Interface- und Game-Designer
bei Funatics Software.

RSTUDIO  WWW.RSTUDIO.COM



Christoph Safferling
ist Head of Game Analytics
bei Blue Byte.

Wer schon mal mit der Statistik-Programmiersprache R gearbeitet hat, weiß: Die Lernkurve der Sprache ist eher eine Lernklippe, und eine ansprechende Benutzerumgebung fehlt. Da kommt RStudio daher: eine open-source IDE für R, übersichtlich und komfortabel. Die Liste der Vorzüge ist lang. Neben üblichen IDE Features (tab/autocomplete, highlighting, etc.) gibt noch weitere Vorteile: »workspace panel« stellt alle Datensätze und Variablen dar, mit Markdown werden Dokumente in HTML, pdf, oder LaTeX direkt aus R geschrieben. Außerdem werden Profile unterstützt, Plots und Graphen gesammelt und geordnet angezeigt, der Help-Viewer ist grandios, eine Git- und Svn-Integration besteht, Projekte (und -bäume) werden unterstützt und vieles mehr.



RStudio ist eine übersichtliche open-source IDE für die Statistik-Programmiersprache R.

FOKUS GAME

WIR BRAUCHEN EINEN »DEUTSCHEN« SPIELEPREIS

Der Deutsche Computerspielpreis (DCP) muss sich weiterentwickeln und seinen Fokus mehr auf deutsche Produktionen legen. GAME-Geschäftsführer Thorsten Unger sieht im Wechsel der Zuständigkeit in der Regierung eine Chance, fordert jedoch auch Unterstützung für die nun verantwortlichen Stellen.

Nach dem DCP ist vor dem DCP. So war es die letzten Jahre, und jedes Jahr sind wir angetreten, den bestmöglichen Preis auf die Beine zu stellen. Und jedes Mal gingen wir mit gemischten Gefühlen aus der Veranstaltung. Mal war es ein vermeintlich bestes Spiel (»Anno«), das sich überraschenderweise gegen einen US-amerikanischen Blockbuster-Titel durchsetzen konnte, mal war es ein Shooter, hinter dem zwar die Jury, wohl aber nicht der Ausrichter auf Seiten der Bundesregierung stand. Offenkundig wurde lediglich, dass politisch gewollt nicht immer auch politisch akzeptiert heißt. Das sind nur einige Beispiele, und wer sich mit dem Preis beschäftigt, wird problemlos weitere finden.

Der Wechsel als Chance

Als mit dem Regierungswechsel die Auflösung der Zwangsehe zwischen DCP und Bernd Neumann kam, war die Erwartung groß, welche die Games-Branche an die damals noch unbenannten neuen Bundesbeauftragten für Kultur und Medien hegte. Was dann kam, war überraschend: Aus Monika Grütters wurden Dorothee Bär und Alexander Dobrindt. Aus der Bundesbeauftragten wurden eine parlamentarische Staatssekretärin und ein Bundesminister. Aus Kultur und Medien wurde ein Ministerium für Verkehr und digitale Infrastruktur. Aus Kulturetat wurde »Digitale Agenda«. Doch genug der Geschichte. Jeder Wechsel ist eine Chance, und die nun Handelnden haben diese allemal verdient. Dorothee Bär als Mitinitiatorin des DCP muss man nicht erklären, wie viel Potenzial Spiele in den Medien sowie der Kultur- und Kreativwirtschaft haben. Also volle Unterstützung, bitte. Der DCP 2014 kann deshalb kein Maßstab sein – das sieht für 2015 ganz anders aus.

Fokus auf die deutsche Branche

Die Herausforderungen bleiben groß, und wir tun gut daran, die Dinge nun mit dem nötigen Maß an Realismus anzugehen, ohne dabei das Ziel von wettbewerbsfähigen Standortbedingungen aus den Augen zu verlieren.

Der DCP ist wie die Stiftung Digitale Spielekultur im Koalitionsvertrag verankert – und gehört damit zum Programm dieser Legislaturperiode.

Nicht nur deshalb: Der GAME hat sich daher gemeinsam mit vielen Experten ausgiebig mit der Weiterentwicklung des Preises beschäftigt. Ein zentraler Faktor sticht dabei ins Auge: Wir finden, dass die gesamte Aufmerksamkeit des Preises auf der deutschen Branche liegen muss. So verlockend »Super Mario« und »Assassin's Creed« zur Attraktivitätssteigerung des Mediums Computerspiel auch sind: Sie verklären den Blick auf das, was wir in Deutschland leisten können. Hier müssen wir ansetzen. Unsere Protagonisten heißen nicht Sam Fisher, Alan Wake, Lara Croft oder Sonic. Sie heißen Edna (die ausbricht) oder Jack Keane. Sie heißen vielleicht auch »Angry Birds« – weil der neueste Teil aus Deutschland kommt. Und ein großartiges Spiel aus Deutschland heißt »Crysis«. Das gehört in den Deutschen Computerspielpreis – und zwar nicht nur ein bisschen, sondern voll und ganz.

Der Award ist die Bühne der deutschen Branche. Er ist der Preis der Branche und der Bundesregierung zur Steigerung der Leistungsfähigkeit und Visibilität der deutschen Games-Branche. Damit ist schon mal ein Fokus klar definiert – diskutieren wir weiter.

Thorsten Unger



Thorsten Unger
ist Geschäftsführer des GAME Bundesverband der deutschen Games-Branche e.V.

Thorsten Unger ist heute Geschäftsführer des Verbands. Der GAME ist die mitgliederstärkste deutsche Interessensvertretung für die Anbieter und Produzenten von digitalen Spielen in Deutschland. Unger ist Gründer des auf Serious Games spezialisierten Unternehmens Zone 2 Connect und Gründer des Expertennetzwerks Wegesrand.

Der GAME Bundesverband der deutschen Games-Branche fordert die Weiterentwicklung des "Deutschen Computerspielpreises" analog zu seinem Auftrag gemäß des ursprünglichen Beschlussantrages des Bundestages (<http://dip21.bundestag.de/dip21/btd/16/080/1608033.pdf>).

Demnach handelt sich um eine Auszeichnung für die Entwickler, Kreativen und Kulturschaffenden der Bundesrepublik Deutschland - sie stehen im Mittelpunkt der Veranstaltung und der Preis ist Bühne wie Förderinstrument zugleich. Die Visibilität der Veranstaltung und die mediale Aufmerksamkeit stellen die deutsche Branche in den Mittelpunkt und inszenieren sich vergleichbar zum Deutschen Film durch den ebenfalls durch die Bundesregierung vergebenen „Deutschen Filmpreis“. Die Kategorien spiegeln die Schaffenskraft und Varianz des digitaler Spiele in Deutschland wieder: Spiele zur Unterhaltung für verschiedene Altersgruppen, Spiele als Kunst, Spiele im täglichen Leben (Gamification), Spiele zum Lernen (Game-based-Learning /Serious Games), Spiele als Technologie und Innovation.

In einem Positionierungspapier fordert der GAME die Weiterentwicklung des Deutschen Computerspielpreises. Der Verband veröffentlichte das Papier auf seiner Website unter www.game-bundesverband.de.



Making Games 05/2014

TEAMING UP FOR EXPANDING A AAA FRANCHISE

Evolving a successful franchise needs careful planning and execution. The teams from Deep Silver and YAGER work closely together to ensure that Dead Island 2 even exceeds all expectations.



Vivien Dollinger
is Development Director
at Deep Silver.

Vivien's enthusiasm for games was first triggered by »Giana Sisters«, in which she invested a ridiculous amount of time some 25 years ago. While school wasn't a priority, her interest in games drove her to do a doctoral thesis later on. After some years in consulting, she joined Travian Games and thereafter Deep Silver, where she currently is the Development Director, leading the producing unit with all her heart.



Carsten Lindner
is Senior Producer
at Deep Silver.

Carsten has been working in the gaming industry for over 17 years on both the publishing and development sides. He is experienced on most platforms for games ranging from big console and PC developments to mobile and social games, and has already been involved in creating more than 100 titles.



Michael Kempson
is Producer
at YAGER.

Michael has been in the industry for nine years, started life out within EA, then a stint with Ninja Theory before returning to EA and Criterion. He then met a wonderful German lady, discovered YAGER and started life again in Berlin.

Once upon a time, we released a lone beautiful and sunny zombie game called »Dead Island« to conquer the hearts of gamers. And so it did, extremely well. Although this was an amazing experience, we faced the challenges of an overwhelmingly successful first title within a (at that time) new IP. Just a straight sequel would not do; we needed to expand on the core and take it to the next level: embed it in the right context and build a lasting, rich universe to meet and exceed our fans' expectations. So we concisely shaped everything from IP vision, and values to a full background story for the Dead Island universe and designed a roadmap of products to build up a cohesive experience for our players. This gloriously leads us to the beginning of Dead Island 2, the most important building block of this venture.

Sunny Side of the Apocalypse

Dead Island 2 reflects the sunny side of the zombie apocalypse. As with all Dead Island titles, we focus on the bright and beautiful aspects that are all too often neglected when a zombie outbreak occurs. Our mission and pas-

sion is to encourage people to celebrate post-apocalyptic life, to enjoy a leisurely lifestyle beneath the warm Californian sun and to have great, zombie-slashing fun together – and best of all: no-one around telling anyone what they can or cannot do. In the end, it is really about taking second chances and living life to the max ... ok, and it's also about loads of awesome, sophisticated ways to kill zombies!

Therefore, choosing the studio for the sequel of Dead Island that would truly bring our next-gen vision for Dead Island to life in AAA quality was at the core of our strategy. We were aiming for the highest quality in terms of creativity, art direction, technical capability to create an ambitious open world and importantly, a team with the spark that demonstrates true enthusiasm for Dead Island. With this strict criteria in mind, we sent out requests for proposals.

Dead Island 2: A true German endeavor

Coming to the end of »Spec Ops: The Line« (SOTL) and immediately being afforded the opportunity to pitch and work on a title like Dead Island? A game which we watched come out and then played with as much enthusiasm



Deep Silver felt that a straight sequel to the first »Dead Island« game wouldn't do and instead decided to expand the core franchise.

as the rest of its fans? It wasn't an opportunity the team at YAGER wanted to squander. It was also something that came with an awful lot of responsibility. A great many people had played and enjoyed the game (us included), Deep Silver as a company clearly had a huge vested interest and great love for the brand. A lot to play for then. As YAGER had matured as a team on SOTL, the one critical element we wanted to bring to Dead Island 2 was »doing the right thing for this game«. Meaning: simply repeating on what was done well on SOTL was not going to work and wouldn't have felt right in this new universe.

While we at Deep Silver received several cool and fun pitches from great studios and did not rule out overseas studios beforehand, YAGER was an instant match on every level. Already from the first meetings we felt that the YAGER team consisted of many true Dead Island fans and provided a valuable foundation for starting off. With the additional benefit of cultural and geographic closeness facilitating an open and close way of collaboration, we were happy to take on Dead Island 2 as a true joint German endeavor.

Looking back now, we shared many memorable moments and good laughs on the way to creating the fun and lighthearted atmosphere that permeates through Dead Island 2 now:

At one time, when you wanted to rush at a zombie and perform a combo or kick then swing to trigger an uppercut with the sledgehammer, this resulted in the zombie flying 60 feet in the air. Always a »laugh out loud«-moment for both those playing and watching.

Then we had a scene implemented, when you walked in to a zombie-filled wedding party room, where you could turn on the DJ turntable to play »Love is All Around« by Wet Wet Wet to awaken the oblivious zombies and smash away. Again, giggles and smiles for everyone around.

Finally, have you ever seen a cat walk in a motion capture suit? The casting was astonishing. Most cats forgot they were cats and walked like very fat bears, zombies, or not at all. Maybe at some point we will be able to release outtakes, so you can see for yourselves.

However, as in most cooperation, it wasn't always a great fairy-tale sugary cake-with-pink-icing picnic, but had its challenges, puzzles, and struggles as every good adventure should have.

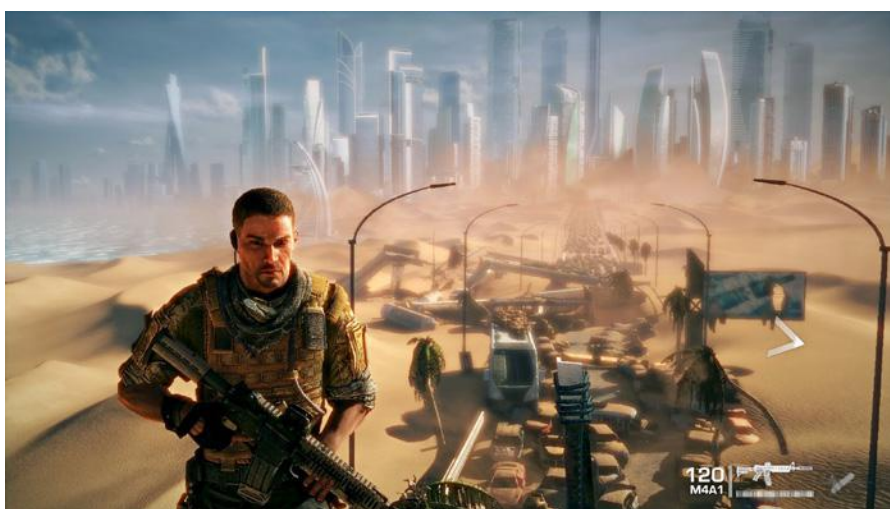
Joining forces: The super-evil publisher and the faithful developer

Never satisfied, always criticizing, always wanting more – more, more, more ... we all know that is what publishers want, don't they? And worse: on top of that they usually want timelines. And worst: later on they want to stick to them. And so on.

So, how to best tame the beast? At Deep Silver (the evil publisher) we think the key lies in building a partnership based on aligned goals – long before the start of pre-production. This way conflicts in the process can be minimized and a



Dead Island 2 presents a bright and beautiful world instead of zombie-induced apocalyptic chaos.



The YAGER-team knew from the start that repeating what worked for »Spec Ops: The Line« wouldn't work with the Dead Island franchise.

transparent, close working collaboration based on trust and mutual respect may be established.

With this prerequisite, every title Deep Silver produces is coordinated by an interdisciplinary team of a creative producer, a producer, and the art director, deeply embedded in the development teams.

With this team being always very close to day-to-day production, there is mutual understanding of the realities of the development process, and this makes things smoother for everyone on the project. So while the Deep Silver producing team is always bringing the bad news (to all stakeholders involved, not only the developer) it isn't to create conflict but rather to work constructively towards the best possible solution on all levels. The goal here is to minimize dissatisfaction and friction on all sides to increase efficiency, and invest everyone's time, enthusiasm and money where it creates true value.

In short, the Deep Silver producing team serves also as the diplomat of the game: when not with the developer, they stay in close communication with all involved departments and show the progress being made.





Playing »Love is All Around« from Wet Wet Wet wakes up the zombies at a wedding reception.

Hands-on, Deep Silver and YAGER achieved this close level of mutual partnership by applying the same level of transparency and inclusivity that we do internally. In this case, that meant Deep Silver producing being embedded within the YAGER team on a biweekly, if not weekly basis and with complete access to all live documents staff of both parties at all levels are constantly in contact via email or Skype while also regularly attending meetings and reviews together, allowing for quick turnover cycles.

With YAGER's development structure, it was also of vital importance ensuring that the DS producing team was present with the development team during planning weeks. Having everyone actively participating in our team environments helped us consistently manage mutual expectations.

Tracking production progress is a key part of both production teams' jobs. We compile all of the data (digital, analogue, wiki and Hansoft) and make it relatable to the milestone plan which is routinely shared between both parties, both on and during milestones.

This does require effort, and we often incorporate more automation into the process. In most cases this started out as team-built and team-led workflows, which were then automated later on. For example most tracking and task management is currently running through Hansoft. Another example being that our Wiki information often becomes directly hyperlinked to our milestone planning in Hansoft itself.

Because the »evil« publisher needs to look at the overall plan from different angles from the YAGER team; after much iteration and experimenting with handmade sheets, we developed a system where the project status could be exported from Hansoft and then formatted and edited to meet the needs of the publishing group.

This way, we now feel – across all levels of hierarchy – that we can deliver the best possible game within the given constraints while still enjoying the upsides of working in the video games industry.

Bridging business needs and agile development

One of the big needs when a publisher works with external developers is bridging business needs and long-term development milestone schedules with agile development processes in modern developer structures.

Deep Silver, similar to YAGER, is a very flat organization with little hierarchical structure and builds on self-responsibility of each and every team member. Accordingly, as any good parent, every Deep Silver team member is therefore not only very proud of its baby, but sometimes also a little protective in order to ensure the baby grows to be the best and most beautiful title it is destined to be.

So, why do we need milestones, then? Firstly, no one wants to sign a development contract worth XX million €, without knowing what they will be getting for their investment at the end. Usually, game development is not an easy challenge. There is a lot of creativity and unknown factors at work and it seems to be almost impossible to put that into a milestone plan at the start of the development. But if you want to sell something, you have to find a way to make it tangible.

However, it's not only a necessity for publishers working with external developers, it also gives you huge production advantages to have this kind of exoskeleton around your production. It allows you to always see where you are from an objective point of view and to keep track of the macro layer, even when your daily business is very deep in the micro layer. And it helps you to keep focus, as losing focus on a project is one of the main reasons for escalating costs in production. Changing visions, losing the vision, spinning up cool ideas. Adding a feature here, adding content there ... This can easily get out of control. Establishing some boundaries helps you to focus and streamline to the right things. And everyone who is in the daily business of a project gets blind to the big picture – in this case it's necessary to have something that reminds you where you should be and what to expect next.

But of course this is totally contradictory to the idea of agile development, which always demands maximum freedom and it's not possible to know everything in detail upfront. However, just the action of planning in as much detail as possible helps to streamline your production in the beginning, warns you of impending traps and helps to create projects you can handle.

So how do you start? First, plan the project in phases. At the very least, the Concept Phase, Pre-Production and Production phases should always be accounted for and added to the plan at the very beginning. This is already a standard in games development, so this shouldn't need any further explanation. And of course, use the phases to eliminate the risks and inform the next phase as much as possible. Get all proof of concepts very early and focus



on great core gameplay loops very early on. Usually things in reality don't turn out to be as initially designed. Make sure you know what you're doing during pre-production. Stability in the meta-design helps you to keep your schedules. Conversely, for a game like Dead Island 2 it is important to leave space for iterations in the micro design. What's fun in combat? How does it feel? How is the world brought to life?

The importance of milestones

And then as mentioned before, you need milestones to detail the phase, which should be defined before the start of each phase. Take your time to evaluate them with the whole team. But finding the right detail level for your milestone plan is the key. Keep flexibility in the details, but also nail down the scope and make every milestone concrete and measurable. Remember, your goal with the milestone plan is to keep track of your progress and to tell you early enough when you have a discrepancy between planning and reality. Use milestone approvals to ground yourself to the earth and to communicate transparently where you are. The earlier you find discrepancies, the earlier you can react to them and shift deliverables, redesign features etc. Also the milestone plan is a living document and needs an official amendment every few milestones. Don't amend every milestone – have some stability. You will never keep it as it was intentionally planned in the beginning. But every time you have to adjust the plan, take a look at the big picture and try to implement changes early on. Don't wait until the end to exchange, downscale or reschedule but keep the macro layer, the big picture and the vision for the game in mind all the way through.

So as the framework helping you to focus your path, agile processes help you to work efficiently in day to day operation and in the sprints. Game development needs flexibility and reiterations on a daily basis. Plan sprints, use Scrum walls and react on every point with flexibility. But it is still important to keep the deliverables of the next milestone in your mind and work to those targets, verifying the status of the project when you reach there. This will cause issues and friction during your development, but you need to be aware of these issues to be able to react to them. There needs to be a lot of communication and commitment between both producers on the developer and publisher sides to solve all of the major issues in development but fast, efficient progress. Great game quality will be your reward for these efforts.

YAGER & agile team structure

YAGER is a very fluid place to work creatively and technically. It operates with a relatively flat structure compared to a traditional business or studio. This allows greater flexibility within a project and encourages the team members to work to a level of ownership and responsibility we couldn't achieve otherwise.

So in our lack of structure, we have as a small Production team with Michael Kempson as Producer and Moritz Finck as Associate Producer.

We also have a small team of discipline Directors: Design Director Jörg Friedrich, Technical Director André Dittrich, Art Director Sebastian Schild and lastly Game Director Bernd Diemer. The team itself is without discipline leads, but is a mix of seniors, regulars and juniors to whom the production team and Directors mentor and guide.

The goal of this structure is not to simply execute an agile development, but is in place to achieve the goal at YAGER of giving as much responsibility as we can to our team to educate and develop our team to the best senior team we could ever possibly develop.

So in Production's case, our job is not to form and create workflows and processes top-down to the team, as we think is fit. Our goal is for the team to develop their processes themselves in a way that makes sense to them. These workflows take more time to emerge during pre-production, but have far more longevity and are also more open to change and remain flexible. We guide the teams through this and develop the overview of all the processes to smooth through unification of them between teams.

One of the borders of this is that no matter the process you work through, we always ensure that the information is broadcasted on a transparent platform, usually Hansoft or Wiki. This then provides production with RAW data to transfer to an overview for the team and external partners.

Our Directors operate in much the same fashion; creative decisions are brought down to all levels of the team. This means that the idea we develop has just as much of a chance of coming from the junior as it does coming from the director. The directors' responsibility in the equation is to always ensure the creative focus, to act as an experienced sounding board and most importantly: ensure that the correct questions and challenges are put to the team along with proposing the correct approach to ensure the result is on target.

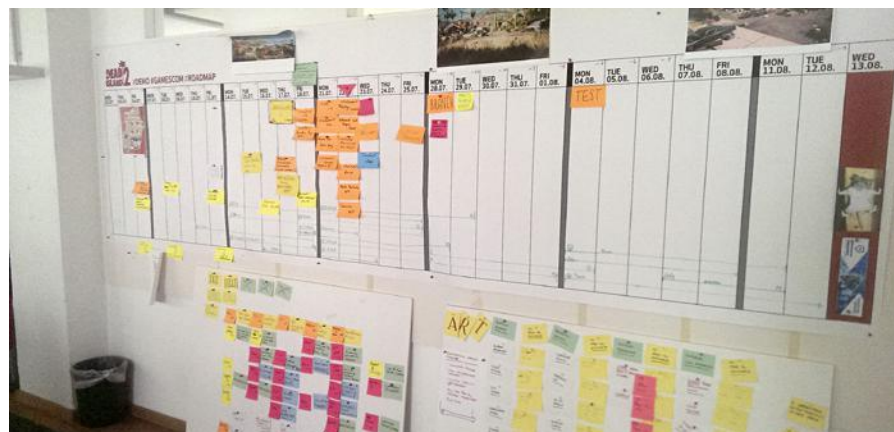
Vivien Dollinger,
Carsten Lindner, Michael Kempson



The motion capture suit made many cats forget about their feline nature and they started walking like fat bears or even zombies.



The teams from Deep Silver and YAGER are constantly in contact via email and Skype and also attend meetings and reviews together.



Scrum walls help the team to keep track of milestones and urgent topics.



Making Games 05/2014



DESIGNING THE WORLD'S SMALLEST MMO

Beaches, palm trees ... and zombies. Jan-Eric Laube and Bernd Diemer explain why California is the perfect setting for Dead Island 2 and how they plan to further improve the multiplayer aspect of the game, taking notes from a game you wouldn't expect: Journey.



Jan-Eric Laube is Creative Producer at Deep Silver.

Jan-Eric has gathered tons of experience as a Game Designer at Acony where he explored every detail of first person shooters and free2play-models. About a year ago he came to the dark side. With his new acquired powers he is supervising the entire Dead Island IP.



Bernd Diemer is Game Director at YAGER.

Bernd worked on Crysis and Ryse at Crytek and teaches at the Games Academy in Berlin.

When defining a setting for a sequel, it's important to choose one that fits the expectations of the franchise and provides familiarity while offering something new at the same time. In order for us to choose the new home of Dead Island 2 we had to define what makes a Dead Island setting:

- **Paradise Gone to Hell:** One of the core pillars of the whole Dead Island IP involves creating the contrast of beautiful locations and the chaos of a zombie apocalypse.
- **Bright and Beautiful:** The most memorable areas of the original Dead Island were the sunny and beautiful ones such as the Hotel Resort in the beginning of the game.
- **Places Where People Want to Go:** Beaches and palm trees are the obvious choices, but in the end the perfect location needs to be a place that people want to visit and explore in their real life when going on vacation.

After evaluating many different locations from around the world, we found the perfect location for what we wanted to achieve: **California**. Not only did California fulfill all the requirements with its beautiful and famous scenery, but it also gave us the possibility to introduce new themes to the franchise thanks to its wide variety of different locales, from the San Francisco Harbor, Hollywood Hills and Venice Beach.

Seamless multiplayer: Building the world's smallest MMO

One of the biggest success factors of the original Dead Island was its simple and fun multiplayer experience. Even today people are still playing together and sharing epic moments. The biggest accomplishment regarding coop was how easy it was for players to actually play together. During normal gameplay, a small pop-up informed the player that another player with similar progress was nearby and – by pressing a single button – the two players were playing together. But the



experience wasn't flawless: players needed to have a similar progression status, and it was only fully seamless for one of the players.

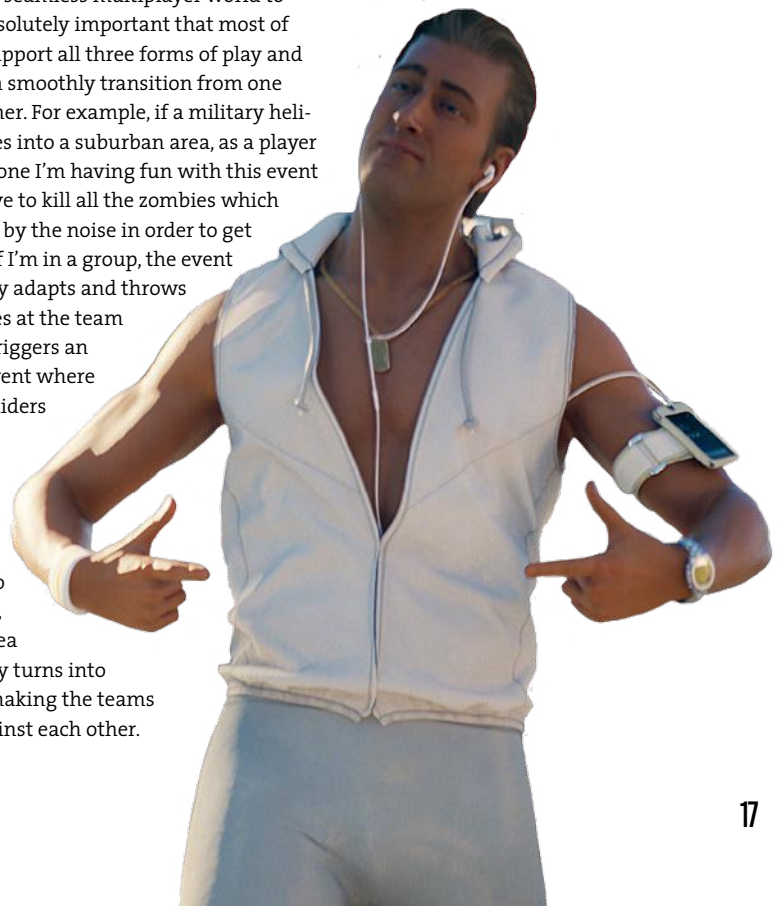
For the sequel, it was important for us to remove these remaining issues and provide an even more immersive and seamless experience. The game which inspired us the most in regards to this was *Journey*. In *Journey* you experienced this vast and immersive world alone when suddenly there was this other player, and it was up to you to decide how you want to approach this encounter. You could just ignore the player, have some random fun with him and then go separate ways after a few minutes, or play the whole game together. Combining this new way of experiencing multiplayer with the over-the-top coop action of the original game was our goal for *Dead Island 2*. In order to achieve such an ambitious goal, we looked at our game and created three core aspects of play:

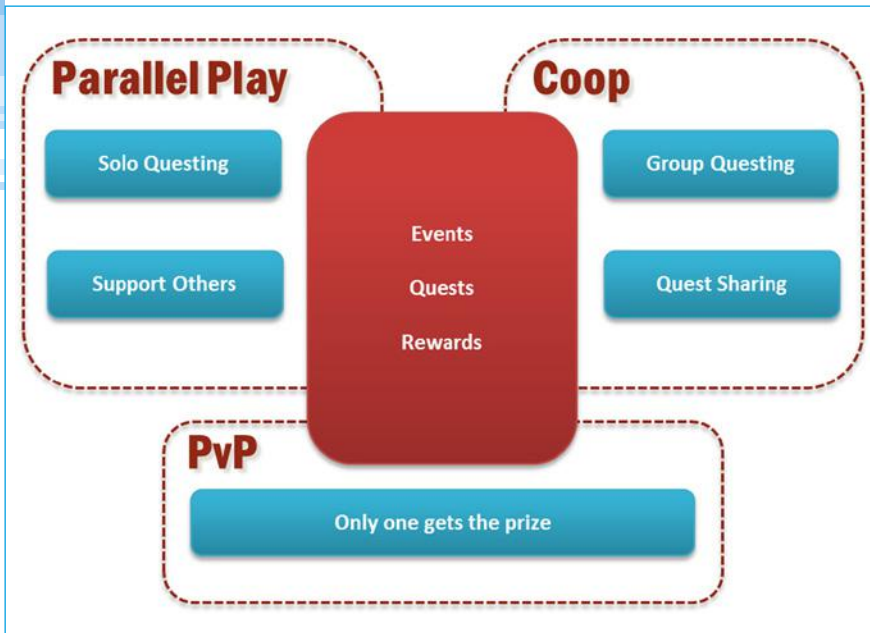
- **Parallel Play:** Replacing single player, parallel play is the default setup when entering a game. Like in an MMO, players share the same world but still can follow their own agenda. On occasion players will meet and potentially support each other loosely.
- **Coop Play:** During parallel play, by participating in more challenging events players will start to fight alongside each other, sharing their quests and rewards.

- **Competitive Play:** Competitive play begins when players go for the same special event but there's only one reward, so the groups can either negotiate or fight over it.

In order for a seamless multiplayer world to work, it is absolutely important that most of the events support all three forms of play and that they can smoothly transition from one type to another. For example, if a military helicopter crashes into a suburban area, as a player who plays alone I'm having fun with this event because I have to kill all the zombies which got attracted by the noise in order to get the reward. If I'm in a group, the event automatically adapts and throws more enemies at the team and maybe triggers an additional event where some NPC Raiders attack the players who now have to defend their loot. And if two teams go for the event, the whole area automatically turns into a PvP zone, making the teams compete against each other.

California fulfilled all requirements to be turned into the setting of *Dead Island 2*, which can be summed up as »Paradise Gone to Hell«.

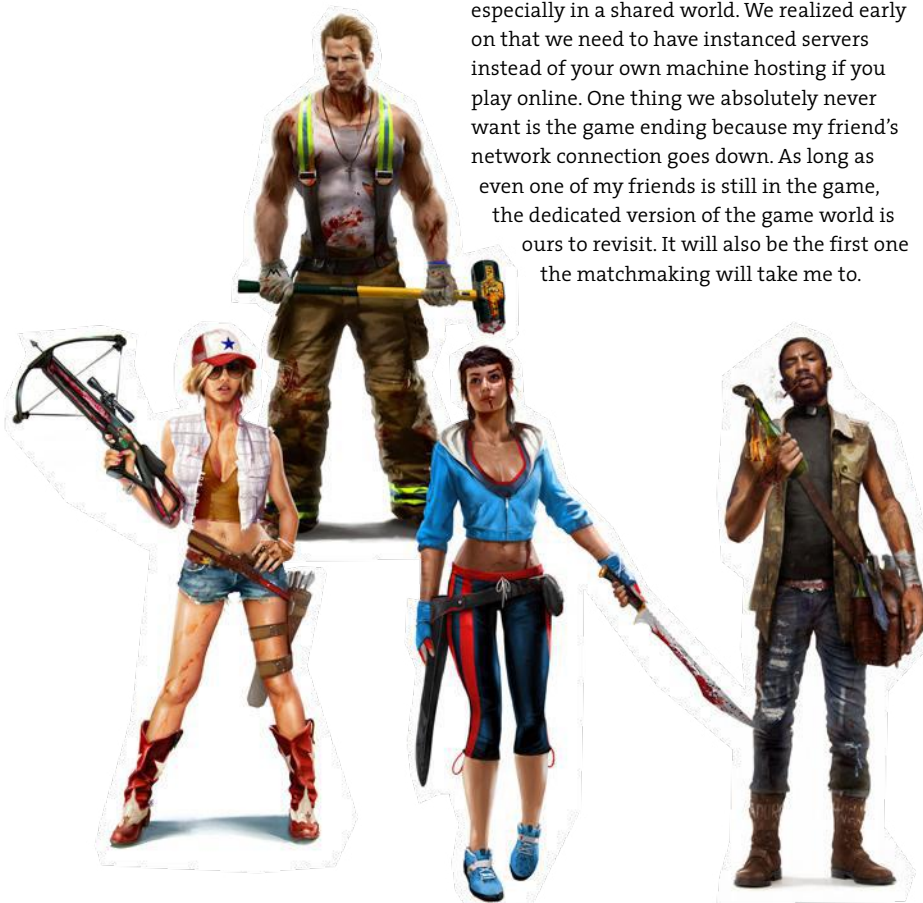




To make sure that the seamless multiplayer world works, the events and quests have to support the three core types of play.

By having this MMO-inspired approach where players are always connected we are offering a more open sandbox with non-linear objectives. A world that allows players to enter and leave at any point without missing out on a quest, having to finish a certain level first, or failing to receive a reward because they were a bit too late when the loot dropped. In essence: a true multiplayer world.

Persistency is essential for immersion, especially in a shared world. We realized early on that we need to have instanced servers instead of your own machine hosting if you play online. One thing we absolutely never want is the game ending because my friend's network connection goes down. As long as even one of my friends is still in the game, the dedicated version of the game world is ours to revisit. It will also be the first one the matchmaking will take me to.



Variety where it counts

Our areas are defined by unique locations that fit the theme and mood. The Hollywood Hills feature a Movie Bar, which has been built up into a defendable fortress by some Survivors. This is the place to get quests, where you learn what is going on around you and where players naturally converge. One quest may require you to help the bar patrons with repairing their electric fence generators. For this they need electronics and gas to fill up the tank. Normally this would be a typical fetch quest. However, creating a specific quest item to acquire does not work for us. Instead we fill the world with multiple locations where you can collect the required resources.

This allows players to choose their favorite play style, rewarding you for exploring the world and playing together. The obvious place to look for gas would be the next gas station. But it also is the favorite haunt of one of our special enemies: the Suicider. Better get someone to come with you! Or if you are more of a cautious player you could loot the whole residential area. There is gas to be found, but it will take you a long time to gather enough. So you can choose between a high risk / high reward and a low risk / low reward challenge. You can play alone, or get your friends to help you out.

There is always something left to do

Don't give players a sense of »running out of space«. To let people play together it is essential that the world never ends. It needs to maintain enough persistency to give me an individual sense of progress and reward me appropriately for doing awesome things. Our world locations have states that have enough variety to stay fresh, yet don't give players a sense of »running out«. A house can be full of zombies, it can be barricaded with a survivor on the roof asking for help, or a special zombie is guarding it. There is always loot, but the obstacle varies in form and difficulty.

Give people their stories instead of a story

Dead Island 2's world is shared by many players and the different areas might be in combination of states when a player enters it. To still provide a feeling of personal progression and story we brought in Max – our commentator, cheerleader, quest giver and exposition machine. His motivation drives your individual story forward while revealing more and more of the world narrative.

He is the narrative instance in the game. He comments on the state of the world as you spread chaos, kill zombies, finish quests and explore the area. He also helpfully points out new things to do. We track what you do, where you do it, who you do it to, and who is with you. This data is used as a basis for his dialogue so players feel that they create their own personal story.

Jan-Eric Laube, Bernd Diemer



Jetzt auch für iPad und Android Tablets



Jetzt
kostenlos
testen!

✓ **Hochwertig**

Edles Layout,
optimiert für das mobile Lesen

✓ **Interaktiv**

Bildergalerien, Videos,
weiterführende Quellen

✓ **Preiswert**

Für Print-Abonnenten
nur 14,40 Euro im Jahr

**Alle Infos für Neubesteller
und Umsteiger:**

www.makinggames.de/app



www.makinggames.de/appstore



www.makinggames.de/googleplay



BUILDING AN OPEN WORLD WITH UNREAL ENGINE 4

When the development of Dead Island 2 began, UE4 had just become available. André Dittrich explains how that influenced the entire process, how different disciplines worked together and how that helped to strengthen the team.



André Dittrich
is Technical Director
at YAGER.

André joined the YAGER team in 2003 as an intern in the final months of the production of their first title »Yager«. During the development of »Spec Ops: The Line« André was leading the engineering team. Right now he is working as Technical Director on Dead Island 2.

Setting out to do Dead Island 2 for next-gen presented us with a number of new challenges: an open world game, which is best to be played in coop on a hardware that was initially unfinished using an engine that was also not quite done. This article will give you an idea of some of these difficulties and how we dealt and still are dealing with them.

Building an open world game with 80 people's manpower

Having done »SpecOps: The Line« (SOTL) before, we knew pretty well how to build and tune a corridor game. In SOTL literally every meter of the journey was hand-tailored and tuned to fit the technical limitations whilst still fully supporting the vision we had for the game. So if we found out that a certain setup of content was not performing well enough technically, or was not good enough to support the game, all that hard work had to be touched again by hand.

We had workflows in place that tried to make sure that levels were built iteratively to avoid as much of that rework as possible, but it still happened often enough to be a real time sink. Building a level that literally spans multiple

square kilometers in the same way was not an option as it would require a way bigger team than the one that had done SOTL.

Right from the start we were clear that even if we could, we would not want to grow the team a lot bigger than that size. We simply thought that our way of working would become a lot harder if not impossible if numbers would grow too far. So we needed to find technical and work flow solutions to the problem.

To fill an open world like ours with content, you need to enable artists and level designers to do that in broad strokes. They should not need to place every bush, piece of rock, tree or construct every house and piece of gameplay by hand. We also needed to find a way to update content that was already placed in the level in a fast and efficient way that required as little manual work as possible.

The landscape

A big portion of our game plays out in the open, so being able to create huge areas of hills and natural environment is crucial for us. Most modern day engines support to create landscape by interactively sculpting it and interactively paint textures onto it. As soon as you try to achieve a more realistic look you need to start applying



A test level is painted with foliage collections.



weathering techniques to make your landscape look like wind and rain had actually sculpted it.

Doing this by hand is an extremely tedious process. So we looked at doing that procedurally. Sadly, Unreal Engine 4 does not have that kind of functionality yet. But using an external tool called »Worldmachine« and some clever ways of importing data back into UE4 allowed us to achieve highly convincing results in a very short time.

The next thing we need is to add life to our landscape with grass, trees, bushes, rocks and so forth. UE4 includes a pretty powerful foliage painting system already. It allows us to paint different pieces of foliage onto the landscape with a high level of control of how this piece is randomly placed. If you try to paint convincing plant life that still is a pretty tedious problem as you have to paint the grass first, then add the flowers, then add the bushes, then add the rocks and so forth.

Looking at how that was done in the editor we saw that actually huge areas of the map were sharing the same combination of individual pieces painted onto the landscape. So we implemented a collection system to combine pieces of foliage and there painting settings into one group that you could paint in one stroke – the foliage collections. These collections of settings can be stored as individual files. Their content can be changed at a later point and these changes can be applied back into the levels where they had been used before. That

system actually allowed us to go even further as our foliage artists now could not only create meshes and materials but also could prepare full collections that could then be used by others and themselves to paint life into our levels.

Adding gameplay

A good way to describe the complexity of adding gameplay into our beautiful landscapes is by describing a short sequence of gameplay:

Nina is playing a Berserker. She stands in front of a small suburban house that she is going to loot now. The first thing she does is kick in the front door. She could have just opened it or if it was locked she might have had to pick the lock. Just as she enters the house she is attacked by a group of zombies. A few swings of the sledge hammer make short work of them. She walks into the living room and starts to look around for things to loot. A glass cupboard in the corner seems to contain something valuable and she smashes it open to get at the loot inside. A few packed bags in the corner had been left by the previous inhabitants.

She is looking through those as well, when she hears growling noises from the garage. Zombies, they probably had been alarmed by the noise she had made so far. Opening the door to the garage reveals another larger



Figure 1 Doors that can be opened, kicked in or lock picked.



Figure 2 A cupboard that can be opened, smashed in or lock picked to get at the loot inside.



Figure 3 Zombies that are spawned to attack the player.



Figure 4 A gas canister that can be thrown and explodes on impact.

group of zombies. Upon entering the garage she notices a gas canister on the left. She picks it up and throws it at the horde which is obliterated in the explosion.

This example includes a number of interactions and gameplay elements and **figure 1 to figure 4** show how they are represented within the level editor.

All of these elements can be simply dragged and dropped into a level and come with complete functionality. In addition to that are elements that are not that obvious to the player like for example audio volumes, post FX settings and of course the house itself. The actual experience for the player is forged by the combination of the individual elements.

»AssetGroups«

Two systems are heavily involved here: the blueprint system of Unreal Engine 4 and a system we build: the so-called »AssetGroup«. Both support the general idea of being able to simply drag and drop something into the level that works and is playable right away.

We call this Functional Content. Blueprints are used to create entities like the cupboard the player can open, lock pick, smash open and then loot. They also allow us to combine art content like meshes, animation, audio, particle FX and so forth with functionality. Art content can be arranged and parameterized and functionality

can be added by using a visual scripting language. This combination can be stored as an asset for the engine and can be dropped into the level.

Most importantly new functional content can be created or existing can be updated by technical artists and designers.

A house like the one described above uses art content like meshes, particle FX, animations and functional content like doors, loot containers, AI spawners, windows, breakable pieces, audio volumes and many more. The combination of these pieces is not something we could easily create with the Blueprint system. So we implemented the AssetGroup functionality. Any combination of objects that can be placed in a level can be combined into an AssetGroup, which then can be drag-and-dropped into a level. AssetGroups can be updated and the changes can be distributed automatically through all our levels. They are also mainly built and maintained by artists and designers. So we wanted to give them a tool they were already familiar with – the level editor.

AssetGroups are created in a level and can be play tested and tweaked in there. As soon as the creator is happy, the individual pieces that are to be turned into an asset group are selected and saved as an asset. Changing an existing asset group is as simple as placing it in a level, do the changes and update the existing asset group asset. So the whole house the player in the exam-



ple has explored becomes an asset like a mesh or particle system that can be placed in the level.

Now creating a whole neighborhood of houses becomes a matter of selecting the right fully functional houses and drag them into a level. Level designers now can focus on adding quests and filling up the area with unique little pieces that turn the area into an interesting place. Artists can either improve the Asset-Groups directly when polishing the houses or adding the little touches that give character to an area directly in the level. So they can focus on turning something that »just« works into something that is special.

Dealing with the uncertainties of unfinished hardware and software

When we started working on Dead Island 2 neither Xbox One nor PS4 was available in a finished state. Also Unreal Engine 4 had just become available and was still highly in flux. So when questions were raised like »How big can our textures be, how much memory do we have for that, how much physics can we have, how many Zombies can we render in what quality?« – we could not really answer them.

But we also knew that we would not be a launch title so we could rely on first experiences from the launch titles being shared on conferences and in articles and we would actually have final hardware to test and analyze our game on for a significant amount of time.

So key for us became to create the content for our game in a way that would allow us to adapt it very late in the project in an efficient way. So the update-ability of our systems described above served a double purpose.

To give one specific example: In order to be able to build a huge number of houses efficiently, the art team came up with a system of building blocks that can easily be arranged to create a multitude of different houses.

When we build the first few for the game we did not yet know how the rendering system would handle them. Should we leave it in smaller pieces to be able to cull more efficiently or would it be better to combine all the pieces into one or a small number of meshes to reduce draw call count? How would we create the LODs for the houses?

Initially there were no LODs but as we had expected they became necessary very soon. So we extended the houses to support LODs and distributed that change to all our houses in the levels automatically. Once we know exactly what balance between number of meshes and their size works best for the culling system of the renderer we also will be able to update all the already placed content in the game.

The closer our levels get to final quality the better we are able to analyze the performance and memory behavior and adapt our content accordingly – again and again relying on the ability to update the content of our game in the easiest way possible.

Game development has become a much more technical process

The technical solutions described above were implemented by a mixture of engineers, level designers, technical artists, artists and so forth. For a project like Dead Island 2 we involved content creators much more into the technical side than before, because allowing them to create the game in an efficient and fun way is as important as getting it done in a way that will give us good results.

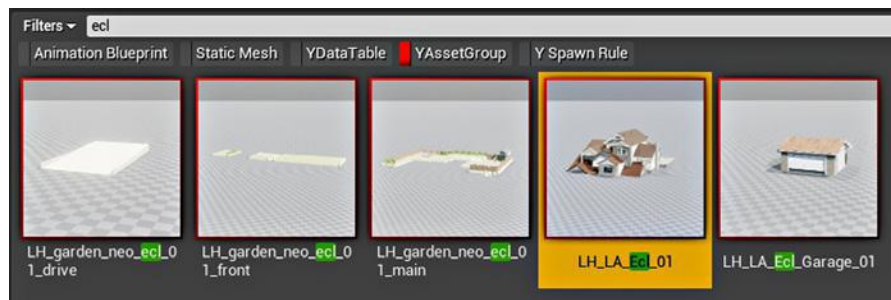
New rendering techniques like physical based shading require content creators to have a much deeper understanding of the technical details of it to create content in the right way. A system like the »mix and match« system that allows us to create hundreds of different looking zombies was mainly invented by the character art team.

The foliage collection system was developed by a mixed team of engineers and artists. Functional content like doors and containers are completely implemented by technical designers and technical artists. The game designers are actual part of the implementation process of their features.

So one of the biggest changes we see when looking at the way we build Dead Island 2 is how much closer people from different departments collaborate when working on the game and how much more everybody needs to understand of the way his peers do their work. This shortens iteration times dramatically.

It creates a new level of understanding between the different disciplines and by that strengthens the team.

André Dittrich



The upper image shows the house assets in the editor; on the lower image, you can see the entire house placed in the level editor.



THE CHALLENGES OF NEXT-GEN DEVELOPMENT

Deep Silver and Yager decided to go all in and release *Dead Island 2* for Xbox One and PlayStation 4 only. We talked with Carsten Lindner and Michael Kempson about advantages and disadvantages of next-gen and how workflows and costs are influenced by new technology.



Carsten Lindner
is Senior Producer
at Deep Silver.

Carsten has been working in the gaming industry for over 17 years on both the publishing and development sides. He is experienced on most platforms for games ranging from big console and PC Developments to mobile and social games, and has already involved in more than 100 titles.



Michael Kempson
is Producer
at YAGER.

Michael has been in the industry for nine years, started life out within EA, then a stint with Ninja Theory before returning to EA and Criterion. He then met a wonderful German lady, discovered YAGER and started life again in Berlin.

Making Games Before Yager worked on *Dead Island 2*, Techland invented the series. Why did Deep Silver choose to change the team behind the game?

Carsten Lindner We have big plans with the *Dead Island* series, that's why we did some of the spin-offs. For the lead title *Dead Island 2* we really wanted to bring the game onto the next level, that's why we thought about who might be the best partner to do that and the pitch from Yager was really promising. That was the reason for this change.

Making Games How many teams did a pitch and what was the reason that Yager won this competition in the end?

Carsten Lindner Four or five studios did a pitch for *Dead Island 2*. When we start a pitch like this we always provide the studios with a high level vision and our target expectations as well as some open questions that should be solved, just to guide them in a certain direction. After that we support the studios during the pitch development and Yager's pitch was just a perfect fit. Other than that Yager convinced us that they had totally adapted the feel of the game and the vision of *Dead Island* in their pitch. Yager also proved that they can handle very big and high quality projects with *Spec Ops* before.

Making Games Before you presented your final pitch to Deep Silver, how many concepts and prototypes did you work on?

Michael Kempson That was an interesting process. Over the course of nearly a year we iterated various prototypes mainly on themes, tone and so on. I think we had five main prototypes. The final prototype that we had at the end of preproduction was the one that we showed behind closed doors at E3 and that one we had shown to Deep Silver. But we had worked on that together to find the right tone and to fit the IP in one solid prototype that

covers everything like the visuals, how the zombies work and how the feeling of melee combat should be.

Making Games If you compare *Dead Island 2* as it is now to your initial pitch, how much did the game change?

Michael Kempson One thing that really changed is how far we can take multiplayer, as co-op was always really important for *Dead Island*. From day one we knew that we have to iterate on that and try to improve what was really very well done before. It got more and more focused as we went through preproduction, simply by the virtue that we started to listen more closely to what people had written about *Dead Island's* co-op. Then we started to talk to first party about our vision of a next-gen online-environment. That gave us the direction which we could work on with *Dead Island* and how to take full advantage of everything that first party can offer. That is one of the key aspects that formed the game most, I think.

Making Games How much influence does Deep Silver take on the game and on Yager's work?

Carsten Lindner We prefer to work very close with the developers. We have some people from Deep Silver in the team, we have of course all the creative force from Yager together and we really prefer to work together as partners. Our guys are with Yager almost every week or every two weeks, are brainstorming together, writing designs together, play testing together – so it's a very close cooperation.

Michael Kempson It is really important that this is not a distant relationship, only having contact every few months. There is no barrier in the relation between Deep Silver's guys and all the designers and people here at Yager and it is very important that all those guys can just talk freely and iterate on the project as much as possible. We really like that we don't have some kind of

artificial or location based barrier between us that would prevent us from having this kind of cooperation. That is the reason why we decided to work together as much as possible, to have complete transparency and 100 percent access. Deep Silver's producer for example is always in contact with every single designer and knows what is going on from day to day – and that just works out nicely that way.

Making Games Dead Island 2 is announced for PS4 and Xbox One only. Why did you decide to go for next-gen only and not for PS3 and Xbox 360?

Carsten Lindner This decision was made very early in preproduction when we were shaping out our target visuals and our vision of the game as well as the seamless multiplayer which is a big step in development. All those things would not be possible on »old-gen« consoles.

Michael Kempson It is always an interesting question, why you would not have as many people play your game as possible. From a development point of view we all have been in productions during »cross gen«-times when everything gets released to every platform. Back then when PS3 and Xbox 360 came out, all the games were released for the new gen but also for PS2 for example. From a development side you always have to make compromises when you decide to do that as you always have to make sure that you achieve the best quality even on the less capable platform. Going for next-gen only seems much more fitting and much more appropriate as we think that Dead Island 2 is going to be a really strong title. And you have to think about possible compromises very early on in development and it would have affected the complete production of the game. Therefore it was a difficult but necessary decision to make the very best game we possibly could. Especially when thinking about seamless multiplayer and a big seamless open world you need that kind of strong starting position.

Making Games So you did go all in on the next-gen consoles and did not want to wait how consumers would adapt to the new consoles?

Carsten Lindner We expected at least an ok-ish start of the next-gen consoles as it was really time for a change in hardware. But yeah, we had to go all in there because there was no time to wait as we wanted to see all of our visions and expectations for Dead Island 2 come true. So we had to go for next-gen, there was no other possibility. I think in the end it was a good decision as the new consoles were adapted even better than we expected. Beside of that Dead Island 2 could not be the same game on »old-gen« that just would not be possible. A lot of games that are released right now on all platforms are only some kind of upgraded games for next-gen. But we decided to fully go for next-gen and are using Unreal Engine 4 and therefore it is pretty much nailed to those machines.

Making Games What are the biggest advantages in your eyes which the next-gen consoles bring?

Michael Kempson Well a big advantage of course is that they have a lot more power, with more RAM, a great GPU, improved designs of the controller and all those things consumers do like about the consoles. I really like to state the advantages for multiplayer, even if people say that of course multiplayer was possible before. But fact is that without Xbox One and PS4 all the features like cloud based servers simply wouldn't have come along. On »old gen« Microsoft and Sony just iterated on their services and I think they needed new platforms to justify those big steps in their services. For a developer this is the best thing to come along and sometimes you need this extra push in hardware and all those extra services you would not necessarily subscribe to. You could set up a cloud server and a server farm without a new piece of hardware but you need a reason to make that work, you need a reason to justify it and you need players to be excited about it and a new platform be it Xbox One or PS4 generates that for you. Therefore I think a lot of those things we do with online would not have been possible without the new platforms – as a developer I think the new consoles gave online this extra push which is a big advantage for us.

»We expected an ok-ish start of the next-gen consoles.«



The visuals of Dead Island 2 »would not be possible on old-gen consoles.«



Yager and Deep Silver collaborated on »how the zombies work and how the feeling of melee combat should be.«

»Unreal Engine 4 is bad-ass. That's pretty much it.«

Making Games As you are talking about online services: You are calling Dead Island 2 the world's smallest MMO. Why did you limit the amount of players to eight then? Is that out of technical reasons or just for balancing issues?

Michael Kempson I love that questions, because when people talk about online games one of the first words always is limited. To be honest, during development nobody ever thinks in this terms. The reason is: Our design teams are looking at how the game is played, how big the world is, how you come across other players, how a certain amount of players does interact with a certain amount of zombies and in the end what is the right balance and mix. That is where you start from. All the programmers and technical guys come in later and start talking about streaming, frame rate and so on ... but that always comes much, much later. In our case our design teams figured that eight players that inhabit the game world and interact with all the object is the amount that feels the most correct. And luckily our programmers and engineering teams are quite happy with that, too.

Making Games What are the main differences between PS4 and Xbox One in your eyes and how do you adapt to that?

Michael Kempson There is a lot of parity and only very few fundamental differences between those two systems. PS3 and Xbox 360 had crazily different architecture in terms of how you develop for them and it took a long time to figure out the best ways to deal with them. With Xbox One and PS4 there is a lot more parity and a lot less restrictions. But they still depend a lot on the developer. So if you sit down with an experienced engineer you still have to decide on how should the budget be spent to get the best out of this hardware. So this question is still there and is not different for the new generation, it simply has a different answer. Fortunately if you get an answer,

the answers for the two new platforms do not differ too much from one another. In terms of online I think Xbox did always have a very good service all the way through and it is still incredibly good and robust particularly with the new generation. Sony also did improve PSN very well and I think PSN really took off to the end of PS3 – so far both are a joy to work with.

Making Games Did you have to adapt your workflow in any way when switching to next-gen?

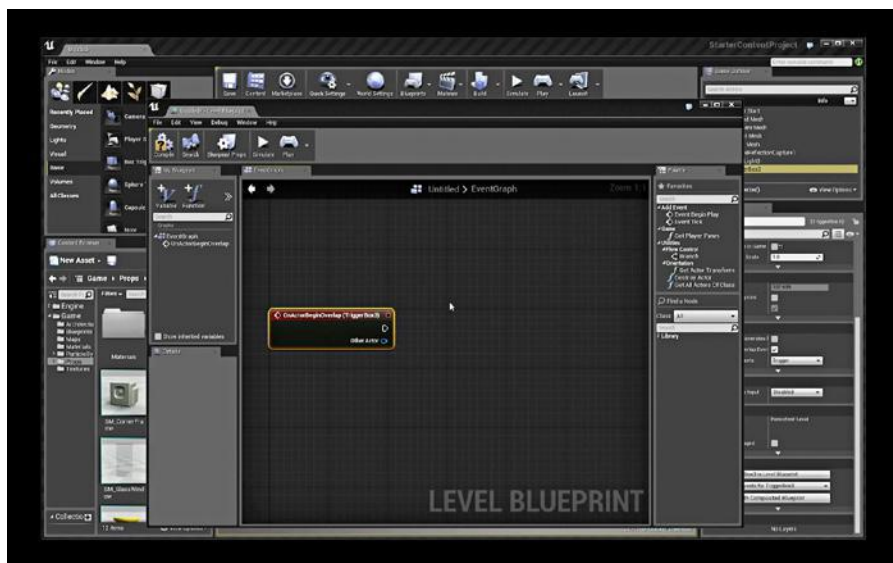
Michael Kempson When we started developing Dead Island 2 a lot of work was done for PC, I guess 90 to 95 percent – that's the reality of development and has been for a long time. We had to make some changes to the way we work especially the way we build levels. We worked with Unreal Engine 3 for Spec Ops for a long time. Unreal Engine 4 presents some more challenges as you have to rethink how the engine works and how it handles content, what is possible and how to get it into the engine, which has not necessarily to do with the next-gen platforms. A lot of that stuff we had to rethink and redevelop even early on during preproduction. For the later on prototypes during preproduction we switched to UE4 and started nailing down contents for that.

Making Games Was there a specific reason to choose Unreal Engine 4?

Michael Kempson Unreal Engine 4 is bad-ass. That's pretty much it (laughs). We have a pretty good relation to Epic, because we worked with Unreal Engine 3 for so long. When we were looking for what is coming up next in Unreal Engine 4 and when we at Yager and Deep Silver looked at the demos our eyes and mouths went gushing and foaming like everybody else's, so we could not wait to work with it. Of course you can always think about different engines like Unity or CryEngine and all those very capable engines but we worked with Unreal engine for so long and it is very much part of how Yager does its business and how we build things so that it made complete sense to choose it. They just ticked all of our boxes in terms of what we wanted.

Making Games You can quite often read comments which state that developing for next-gen is way more expensive than developing for old-gen. What is your opinion on that?

Carsten Lindner In general I think game development costs are increasing every year because of better visuals and higher expectations of gamers, so I think this is given as a general fact. And of course that is the same when developing for next-gen consoles as for better visuals you have to spend more money. But you also have to look for better engines and better processes to cut down costs like with Unreal Engine 4 and its blueprint-system. Things like this help you to work more efficiently and save time and money.



Unreal Engine 4 offers systems like blueprints with which content creation is much easier: »Why have four guys if one can do it.«

Michael Kempson Carsten is absolutely right. If money is no object – for some franchises this is obviously the case – then you can always spend more. I think neither next-gen nor Unreal Engine 4 or any of those are contributing to costs going up as costs have always gone up all the way to where we are now, even with Xbox 360 and PS3. Again, why we chose Unreal Engine 4 is because of how fast the engine lets you create content and how quickly you can iterate on things. A lot of production costs can be cut when you ask yourself how much content you are going to do and how long it will take you to get it in the engine and of course once it is in the engine, how fast you can play it and iterate on it. A big thing for developers is, how fast can I play the game in its entirety and make sure that it is a good game. If your pipelines for getting content in are not very robust and take a long time then of course the development takes a long time and if your development takes a long time then your costs go up. Unreal Engine 4 is just a very robust and streamlined tool for that. It's a very iterative engine that lets you get a running game very quickly and that lets you get your content into the game very quickly as well.

As Carsten said, the blueprint-system is very helpful there, too. Imagine a functional object in the world like a box that you can open. Back in the old days you would need a programmer, an animator and all those guys ... so you would

need four people to create a box that opens. In Unreal Engine 4 you only need one guy. One guy who is capable of making that box and to feed it into the blueprint-system, he can have control over moving it and player interaction – he can basically achieve what needed four people to do. That is the larger philosophy behind Unreal Engine 4. Things like this free up a lot of time. That does not mean that you need less people, but it means that you suddenly have got more people on your production with whom you can do more cool stuff and get it into the game faster. By that you are achieving increasing quality.

Carsten Lindner Yeah, you don't need to do special voodoo for the next-gen consoles, it's just that increased visual fidelity and more advanced gameplay features increase the costs. But of course it is a fact that the overall budgets in games development increase every year. We always want better visuals, more gameplay features ... we always want to go ahead with things. That is what increases the costs and that is something that we have to face of course. But it is definitely not like, »oh there is a new generation of consoles and that is doubling the costs«.

Michael Kempson I think the perception does not come from suddenly things cost twice as much but more likely from developers wanting to put twice as much stuff into their game. That is probably the better way of thinking about it.

Interview: Sebastian Weber

 »Next-gen is not contributing to costs going up.«

co | wana
MARKETING AGENCY

Für einen führenden Videospielepublisher mit Sitz in den USA und Niederlassungen in Europa, Asien und Ozeanien suchen wir für den Standort Frankfurt am Main mit Marketing- und Vertriebsverantwortung für Deutschland, Österreich und der Schweiz einen engagierten

Junior PR Consultant (m/w)

Wir bieten:

- eine hochspannende Tätigkeit in einem internationalen Unternehmen mit langer Tradition
- eine abwechslungsreiche Aufgabe mit Gestaltungsmöglichkeiten und Eigenverantwortung
- einen fantastischen AAA-Produktkatalog und weltbekannte Spielmarken
- flache Hierarchien, internationales Arbeiten und ein hochmotiviertes Team

Ihre Hauptaufgaben:

- Unterstützung des Marketing- und PR-Teams bei der Presse- und Öffentlichkeitsarbeit
- Proaktive Pflege und Ausbau des bestehenden Kontaktenetzwerks
- Verfassen und Übersetzen von Produkt- und Presstexten
- Distribution von Produktinformationen und Mustern
- interne und externe Produktpräsentationen und -schulungen
- Planung von Inhalten für die deutschsprachigen Social-Media-Kanäle
- Teilnahme an und Organisation von Presse- und Publikums-Events im In- und Ausland
- Dokumentation der Berichterstattung
- Koordination mit Europazentrale und Global HQ

Für diese Position qualifiziert Sie:

- eine abgeschlossene Berufsausbildung oder Studium
- Berufserfahrung oder Praktika im Medienbereich
- umfangreiche Social-Media- und Videopielkenntnisse
- ausgezeichnete MS-Office-Kenntnisse
- exzellente Deutsch- und sehr gute Englischkenntnisse in Wort und Schrift
- hervorragende Teamfähigkeit, Flexibilität und Belastbarkeit

Wir freuen uns auf Sie! Senden Sie uns Ihre aussagekräftigen Unterlagen mit Angabe der Position, Ihres frühestmöglichen Eintrittstermins und Ihrer Gehaltsvorstellung direkt an micha@cowana.de.

EINE EIGENE ENGINE BAUEN – TEIL 4

PARTIKELEFFEKTE FÜR DIE NÄCHSTE GENERATION

Deck13 hat für das Projekt Lords of the Fallen eine eigene Multiplattform-Technologie namens Fledge entwickelt. Im vierten und letzten Teil der Artikelserie dreht sich alles um die realistische Physik-Simulation – von Millionen an Partikeln bis hin zu lebensechtem Haar und Fell.

Die Serie im Überblick

- **Teil 1**
Auf dem Weg in die nächste Generation
- **Teil 2**
Into the Guts of the Rendering-Engine
- **Teil 3**
Visuelle Effekte für die nächste Generation
- **Teil 4**
Partikeleffekte für die nächste Generation

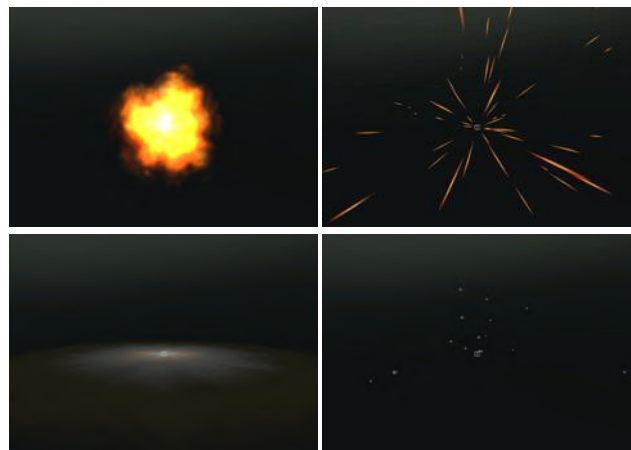
Dieser Artikel unserer Serie geht weiterhin näher auf ausgewählte grafische Effekte ein, die wir für Lords of the Fallen entwickelt haben. In der vergangenen Ausgabe haben wir das volumetrische Beleuchtungssystem von Fledge vorgestellt sowie einen Überblick über unsere Implementierungen für »Deferred Decals«, »Bokeh Depth of Field« und das in Lords of the Fallen verwendete Lens-Flare-System gegeben. Diesmal wollen wir einen Einblick in das Partikelsystem von Fledge sowie eine Beschreibung der APEX-Integration in die Engine ermöglichen. Außerdem besprechen wir einige Themen, die wir bisher noch nicht produktiv verwenden und die daher vermutlich erst in zukünftigen Projekten zum Einsatz kommen.

Erweitertes Partikelsystem

In der aktuellen Version von Fledge haben wir das Partikelsystem erweitert und zusätzlich zur bereits vorhandenen CPU-basierten Partikelsimulation ein neues System für GPU-Partikel entwickelt. Diese Hybrid-Lösung erlaubt es uns, sowohl Partikeleffekte mit Millionen winziger

Partikel als auch Effekte mit wenigen, über komplexere Materialien definierte Partikel zu erzeugen. Das GPU-Partikelsystem integriert sich hierbei in unsere vorhandene Tools, womit es möglich ist, weitgehend transparent zwischen GPU- und CPU-basierten Partikeln zu wechseln.

Das System verwendet Module, um Effekte zu beschreiben. Es unterscheidet dabei zwischen Effekt, Emmitter und Emmitter-Modul. Ein einzelner





GPU-Partikeln können mit ihrer Umgebung kollidieren, die dafür notwendige Simulation ist für tausende Partikel möglich.

Effekt kann aus einer beliebigen Menge von Emittlern bestehen. Die Aufgabe von Emittlern ist es, neue Partikel zu erzeugen und vorhandene Partikel zu simulieren, also durch die Welt zu bewegen. Wie die Engine einen Partikel erzeugt oder simuliert, lässt sich über verschiedene Module beschreiben: Es werden beispielsweise Module angeboten, welche die Startposition von Partikeln festlegen. Diese können etwa in einem Punkt auf einer Kugel oder sogar auf einem bestimmten Mesh entstehen (»spawnen«). Andere Module regeln die Farbe, Größe oder Geschwindigkeit von Partikeln.

Für GPU-Partikel wurden zusätzliche Module entwickelt, welche die CPU-Partikel nicht unterstützen; beispielsweise können GPU-Partikel mit der Umgebung kollidieren. Dafür verwenden wir einen Screenspace-Ansatz, der Informationen aus dem G-Buffer analysiert und somit äußerst effizient Kollisionstests für Tausende Partikel durchführen kann.

Die Interaktion mit Vektorfeldern ist ein weiteres Feature, das nur für GPU-Partikel verfügbar ist. Vektorfelder können entweder zufallsgeneriert sein oder aus externen Tools wie »Houdini« stammen und exportierte Simulationsergebnisse sein. Repräsentiert werden Vektorfelder mithilfe volumetrischer Texturen, wobei die Voxel auf ein regelmäßiges Gitter im Simulationsraum abgebildet werden. Während der Simulation auf der GPU liest das System Vektoren innerhalb des Volumens und beeinflusst die Bewegung des simulierten Partikels entsprechend. Hierfür ergibt sich ein weiterer Vorteil aus der Verwendung volumetrischer Texturen, weil von der hardwarebeschleunigten trilinearen Interpolation zwischen den Voxeln profitiert werden kann.

Die Engine bietet außerdem viele Optionen, um Partikelmaterialien zu konfigurieren. Ein Material beschreibt das Aussehen eines Partikels. In frühen Partikelsystemen war das zumeist auf eine einfache Textur beschränkt. Mittlerweile gibt es eine Vielzahl von Techniken, die Partikelmaterialien interessant machen. Dies variiert von Techniken wie Softpartikeln zum Verstecken von Clipping-Artefakten bis hin zu fortgeschrittenen Techniken wie Flow Maps oder ähnlichen, die speziell auf bestimmte visuelle Effekte abzielen.

Andere Material-Features, die Fledge anbietet, sind zum Beispiel ein steuerbarer Glow-Faktor per Emitter oder ein Verzerren der Partikel-Textur anhand von Distortion Maps. Kombiniert lassen sich mit diesen Einstellungen äußerst interessante Effekte erzeugen, wie in **Abbildung 1** (Seite 30) zu sehen ist.

Ein weiteres wichtiges Thema für das Projekt war die Interaktion von Partikeln mit der Szenenbeleuchtung. Diese erlaubt es, Effekte konsistenter in die Umgebung zu integrieren, und befreit uns zudem davon, Effekte je nach Umgebung und Lichtsituation speziell einstellen zu müssen.

Während der Projektlaufzeit haben wir dafür diverse Ideen entwickelt und implementiert. Einer der ersten Ansätze war die Verwendung von »Forward«-Beleuchtung. Um diese möglichst effizient einzusetzen, berücksichtigten wir nur die räumlich nächsten N-Lichtquellen für einen Effekt und das System übergab die Eigenschaften der Lichter beim Rendern der



David Reing
ist Engine/Graphics
Programmer
bei Deck13 Interactive.

David arbeitet seit Juli 2011 bei Deck13. Aktuell ist er an der Portierung und Optimierung der Xbox-One-Version von Fledge beteiligt. Neben der Pflege und Optimierung vorhandener Systeme der Engine war David maßgeblich für die Portierung der Engine zu DirectX 11 verantwortlich und steuerte einige Subsysteme zur Engine bei, wie zum Beispiel das Partikelsystem und diverse grafische Features.

[@D13_Dreinig](#)



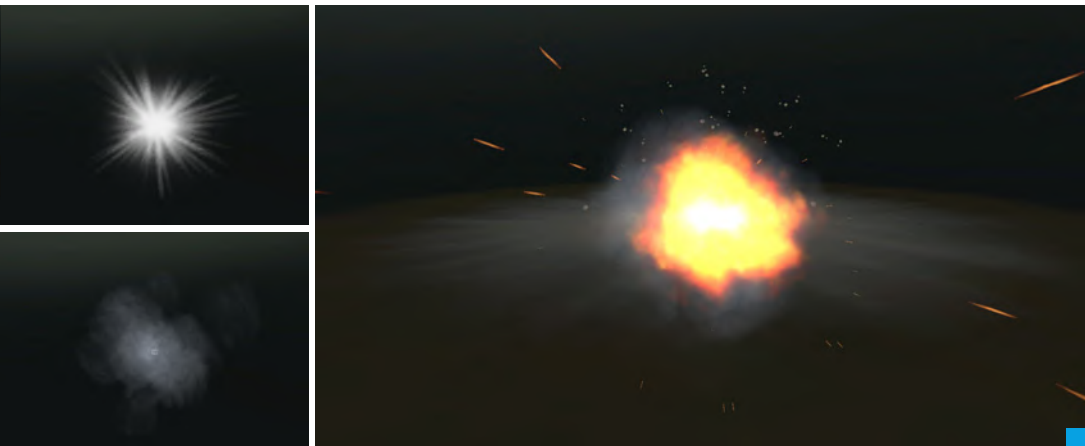
Benjamin Glatzel
ist Engine/Graphics
Programmer
bei Deck13 Interactive.

Benjamin arbeitet seit Anfang 2012 bei Deck13 und ist aktuell für die Portierung und Optimierung der PS4/GNM-Version von Fledge sowie für sämtliche physikalischen Effekte verantwortlich. Neben vielen Low- bis High-Level-Optimierungen hat er zudem zentrale Subsysteme der Engine beigesteuert, wie beispielsweise das auf Software-Rasterisierung basierende Occlusion Culling System, das volumetrische Beleuchtungssystem sowie das Character-IK-System.



Dominik Lazarek
ist Engine/Graphics
Programmer
bei Deck13 Interactive.

Dominik arbeitet seit September 2013 bei Deck13. Aktuell ist er mit der Portierung von Fledge-Subsystemen und Programmibliotheken für Xbox One und PS4 beschäftigt. Zuvor war er für die Entwicklung eines Simulations- und Rendering Frameworks für Haare, Fell und Gras im Rahmen seiner Masterarbeit zuständig.



Die Fledge-Engine baut Effekte aus verschiedenen Emittlern auf, die neue Partikel erzeugen (siehe Bilder oben). Alle Einzelteile fügt das System dann zum finalen Effekt zusammen - hier zu einer Explosion.



Beispiel für die GPU-Partikel-Interaktion mit einem animierten Vektorfeld.



Abbildung 1 Mithilfe von Flow Maps lassen sich zum Beispiel solche Feuereffekte erstellen.

Partikel als Shader-Parameter. Das Ergebnis war zufriedenstellend. Allerdings haben wir einen Ansatz gesucht, der möglichst effizient und nicht auf eine kleine Menge potentieller Lichtquellen eingeschränkt war.

Die Lösung, die in Lords of the Fallen zum Einsatz kommt, entkoppelt – ähnlich wie Deferred Rendering – die Beleuchtung vom Rendern von Partikeln. Dies gestattet eine weitaus größere Menge an Lichtquellen zum Beleuchten der Partikel.

Hierfür schreibt das System die Positionen der Partikel-Vertice in einen linearen Puffer. Um diese Referenzpunkte zu beleuchten, werden die jeweiligen Positionen in einem weiteren Schritt rekonstruiert und die Beleuchtung für den entsprechenden Punkt ermittelt. Wenn die Partikel im letzten Schritt gerendert werden, dann kann das System die finale Beleuchtung für die jeweilige Position anhand des Vertex-Index aus dem zuvor erzeugten Puffer lesen. Diese Methode funktioniert unabhängig für jeden Lichttyp und schränkt die Menge der verwendeten Lichter quasi nicht ein.

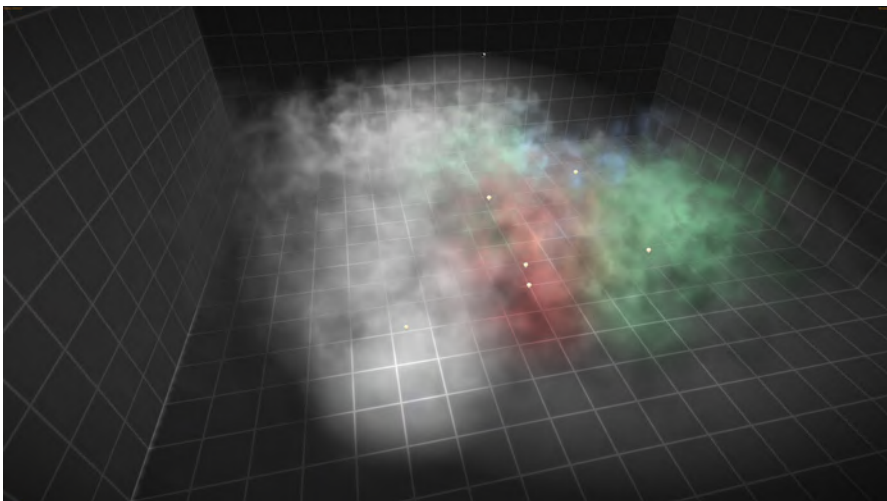
Die beschriebene Methode haben wir sowohl für DirectX-11- als auch für DirectX-

9-Hardware implementiert. Die Indizierung der Vertices realisieren wir für DirectX 9 direkt über generierte Vertex-Attribute. Für DirectX 11 können wir auf die automatisch erzeugten Systemattribute zurückgreifen, in diesem Fall speziell SV_VertexID.

Eine weitere Idee, die bisher noch nicht umgesetzt wurde, ist die Beleuchtung von Partikeln mit indirektem Licht. Das für Lords of the Fallen verwendete GI-System liefert bereits indirekte Umgebungsbeleuchtung als Volumen bestehend aus vielen Lightprobes die einfallendes Licht in Form von Kugelflächenfunktionen (»Spherical Harmonics«) speichern. Damit sind theoretisch alle Daten vorhanden, um Partikel auf indirekte Beleuchtung reagieren zu lassen.

Im weiteren Projektverlauf stellten wir schnell fest, dass wir viele Teile des vorhandenen Systems grundlegend überarbeiten mussten, um die große Menge an Effekten effizient verarbeiten zu können. Wichtig hierfür war die Parallelisierung der Simulation sowie das Erzeugen von Render-Daten. Das aktuelle System verteilt die Simulation einzelner Emitters auf verschiedene Prozessorkerne und vermindert die benötigte Ausführzeit damit effektiv. Außerdem konnten wir die notwendige Prozesslaufzeit teilweise verstecken, indem die Engine dieses System parallel zu anderen Subsystemen ausführt. Neben der Parallelisierung des Partikelsystems haben wir darauf geachtet, das System weitgehend zu vereinfachen und viele Altlasten zu entfernen, die nicht mehr notwendig beziehungsweise auf andere Art und Weise realisierbar waren. Weiterhin stellten wir das System auf einen stark datengetriebenen, Cache-effizienten Ansatz zur Simulation einzelner Effekte um, der zusätzlich von speziell SIMD-optimierten Code-Pfaden profitiert.

Ein weiterer Teil des Systems, der stark optimiert werden musste, war das Erzeugen der Render-Daten für einzelne Partikel. In diesem Zusammenhang haben wir den dafür verwendeten Vertex-Cache des Systems im Hinblick auf Vermeidung von CPU-GPU-Sync-



Die Fledge-Engine erlaubt es, Partikel dynamisch zu beleuchten, und schränkt dabei die Menge an Lichtquellen kaum ein.

Points umgeschrieben. Das aktuelle System unterstützt außerdem Instancing, um den Datenverkehr zwischen CPU und GPU zu reduzieren. Ein dadurch entstandener Vorteil war es, viel Arbeitslast auf die Grafikkarte auslagern zu können (Erzeugen und Ausrichten der Partikel-Sprites etc.), was den notwendigen CPU-Overhead stark verminderte.

Erweiterte physikalische Effekte mithilfe von PhysX und APEX

Interaktivität ist einer der zentralen Punkte, welche die Immersion der Spieler in die Spielwelt ermöglichen. Aus diesem Grund haben wir uns bereits in der frühen Planungsphase des Projekts dafür entschieden, die Integration der Physik-Middleware PhysX zu überarbeiten und auf die gehobenen Anforderungen des Projekts vorzubereiten. Ein früher Bestandteil der Planung war unter anderem die Entwicklung einer Ragdoll-Simulation, die problemlos mit einer Vielzahl von Charakteren funktioniert und sich bei der Konfiguration möglichst automatisiert an die Skelette dieser anpasst. In diesem Kontext haben wir uns ebenso dazu entschieden, die auf PhysX aufbauende Middleware APEX der Firma Nvidia zu integrieren.

Erweiterte physikalische Effekte auf Basis von PhysX

Die Ragdoll-Simulation in Lords of the Fallen wird für eine Vielzahl unterschiedlicher Situationen eingesetzt. Einerseits benutzen wir die Simulation natürlich, um Gegner nach ihrem Tod physikalisch simuliert in die Umgebung einzugliedern. Mithilfe von Kraftfeldern, die an die Waffen gebunden sind, und durch bloße physische Berührung kann der Spieler – sowie sämtliche anderen Charaktere in der Welt – die Ragdolls in Bewegung versetzen. Andererseits werden die Ragdolls für die aktive Simulation verschiedenster Objekte verwendet. Unsere Animatoren können Charaktere partiell simulieren und somit zum Beispiel Ketten oder Waffen, die Charaktere mit sich herumtragen, passend zur Animation des gesamten Skeletts mitbewegen. Dieses System haben wir beispielsweise auch schon verwendet, um statische Objekte interaktiver zu gestalten – etwa eine Lampe, die an einer Kette befestigt ist. Die Kette und die Lampe werden dabei als »Rigid Bodies« simuliert und mithilfe des zugrunde liegenden Animationsskeletts animiert. Teile der bestehenden skelettalen Animation können durch eine partielle oder aktive Ragdoll-Simulation auf die Simulation übertragen werden. So befinden sich einige Teile der soeben erwähnten Lampen immer in Bewegung, während diese ihre »animierte Bewegung« auf die simulierten Teile der Ragdoll übertragen. Dadurch erreichen wir eine Simulation, die glaubwürdig und nahezu animiert wirkt, gleichzeitig jedoch mit den Charakteren im Level interagieren kann. Die aktive Ragdoll-Simulation verwenden wir zudem dafür, dass

beispielsweise Gegner den Helden greifen und durch die Luft schleudern können. Dabei haben wir die Möglichkeit, den gesamten Flug physikalisch zu simulieren und zugleich mit bereits existierenden Animationen lebendiger zu gestalten. Im Anschluss können wir die Ragdoll gemächlich über einen kurzen Zeitabschnitt in eine vorgefertigte Pose versteifen und somit auf eine vorgefertigte Aufsteh-Animation vorbereiten. Diese wird dann mit gewöhnlichem Animation Blending eingeblendet und im Anschluss abgespielt.

Massive Simulation von Kleidung mithilfe von APEX Cloth

Zu Beginn des Projekts haben wir die auf Partikeln basierende Simulation von Kleidung, welche ein Teilmodul der Middleware APEX ist, nur prototypisch integriert und für die physikalische Animation eines einfachen Umhangs des Hauptcharakters verwendet. Weil uns die Ergebnisse überzeugt haben, erweiterten wir das System über die Projektlaufzeit hinweg stark und integrierten es vollständig in die Eingeweide von Fledge. Die APEX-Simulation kann dabei entweder vollständig CPU-basiert oder auf Nvidia-Grafikkarten unter Zuhilfenahme zusätzlicher CUDA-Kernel beschleunigt ablaufen.

Mittlerweile gibt es in Lords of the Fallen fast keinen Charakter mehr, der nicht mindestens ein Cloth Asset irgendwo an seinem Körper versteckt hat. Es ist vielmehr zur Regel geworden, dass wir die Spielfiguren aus bis zu acht verschiedenen Kleidungsstücken zusammensetzen und selbst die Waffen hin und wieder mit kleinen simulierten Fähnchen oder Kleidungssetzen dekorieren. Die Simulation hat sich jedoch nicht nur für Stoff als geeignet erwiesen. Wir konnten diese beispielsweise auch für die physikbasierte Animation von Stahlketten und Haaren erfolgreich einsetzen.

Abbildung 2 zeigt beispielsweise ein Rüstungsset, das wir aus vier verschiedenen Cloth Assets zusammengesetzt haben: Oben ist eine Ruheanimation zu sehen, mittig eine Kampfanimation. Die untere Abbildung zeigt den Charakter zum Vergleich ohne Kleidung. Gut zu erkennen ist, dass das APEX-Cloth-Modul nicht nur eine kleine Beigabe, sondern vielmehr einen zentralen Part des Spielerlebnisses und der visuellen Qualität von Lords of the Fallen darstellt.

Unsere Animatoren können mit den von APEX gelieferten Plug-ins und Tools sämtliche Kleidungsstücke detailliert konfigurieren. Weil die Simulation auf die Verwendung in Kombination mit skelettalen Animationen spezialisiert ist, können diese beispielsweise vereinfachte Kollisionsgeometrien um die Bones der Charaktere stülpen. Sie sind für die Interaktion mit der Kleidung verantwortlich und werden dynamisch an die Animation angepasst, die wir über das Animationssystem von Fledge abspielen. Zudem stehen unzählige Einstellungsmöglichkeiten für die Konfigura-



Abbildung 2 Dieses Beispiel zeigt den Hauptcharakter des Spiels einmal in der Ruheposition (oben) und in einer Kampfhaltung (Mitte), wobei sich die Kleidung realistisch der Pose der Figur anpasst. Die untere Abbildung zeigt den Charakter in der Ruheposition ohne APEX-Cloth-Elemente.

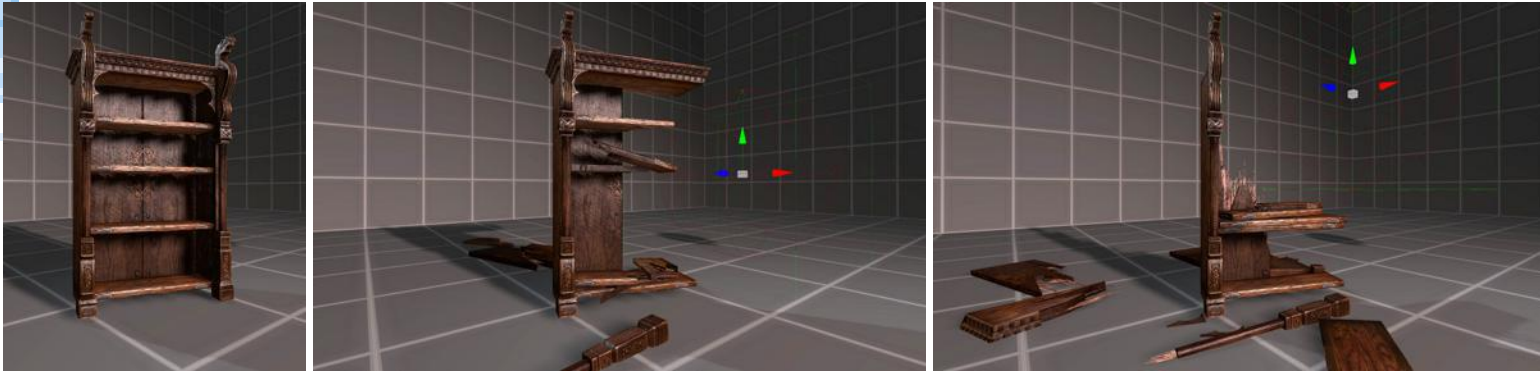


Abbildung 3 Mithilfe eines Kraftfelds lässt sich der Schrank Schritt für Schritt in seine Einzelteile zerlegen (von links nach rechts). Durch die exakte Form der Kollisionsgeometrien können einzelne Bretterstücke auf den noch nicht gelösten Teilen des Objekts zur Ruhe kommen. Durch die Verwendung von Kraftfeldern reagieren Objekte dynamisch und zur Position und Stärke der wirkenden Kraft passend.

tion des Verhaltens des Kleidungsstücks zur Verfügung. Dazu gehören etwa die Steifheit, die Trägheit, die Reibung und verschiedenste Dämpfungsfaktoren. Zudem können wir genau festlegen, wie weit die simulierte Kleidung von dem zugrunde liegenden animierten Skelett im Hintergrund abweichen darf. Dieses als Backstopping bezeichnete Subsystem lässt sich über ein Vertex-Brush-Tool für sämtliche Vertices genauestens konfigurieren.

Da die Interaktion zwischen der Cloth-Simulation mit unserer komplexen Level-Kollisionsgeometrie viel zu kostspielig wäre, verwenden wir für die Interaktion zwischen Kleidung und Umgebung einen einfachen Trick. Dazu spawnen wir direkt unter den Charakteren eine einfache Ebenegeometrie, die wir auf dem nächsten Punkt auf der Oberfläche der detaillierten Kollisionsgeometrie unter dem Charakter platzieren. Das Endresultat ist natürlich nicht hundertprozentig korrekt, war aus unserer Sicht jedoch visuell vollkommen zufriedenstellend. Ein ähnliches System benutzen wir auch für die Interaktion zwischen Charakteren und großflächigen simulierten Fahnen, die wir zum Beispiel zum Verstecken von Geheimnissen einsetzen. Hierzu erzeugen wir für alle Charaktere, die sich in der Nähe eines für die Spielerinteraktion vorgesehenen Assets befinden, einfache Kapselgeometrien und lassen diese sich mit der Positionsänderung des Spielers mitbewegen. Falls der Spieler den definierten Minimalradius verlässt, wird die Physikgeometrie gelöscht und es findet keinerlei Interaktion mehr statt.

Dynamische Zerstörung von Objekten mit APEX Destruction

In vorherigen Produktionen haben wir ein einfaches, bei uns entwickeltes Destructible-System verwendet. Hierfür entwarfen unsere Designer zwei Versionen des zerstörbaren Objekts. Die erste Version entsprach dem intakten Zustand, die zweite der Menge der zu spawnenden Bruchteile. Wenn unsere Designer ein solches Objekt in der Welt platziert hatten und der Spieler es beispielsweise mit dem Schwert traf, dann tauschte das System

das Basisobjekt gegen die aus Einzelteilen bestehende Version aus. Die Einzelteile wiederum umhüllt das System mit einer einfachen Boxkollisionsgeometrie auf Basis ihrer sogenannten »Bounding Box« und simuliert sie als Rigid Bodies in der Physikszenen.

Für Lords of the Fallen haben wir uns dafür entschieden, von der Verwendung der herkömmlichen Destructibles abzusehen und sämtliche zerstörbaren Objekte innerhalb der Spielwelt mithilfe des Destruction-Moduls von APEX umzusetzen. Der Workflow ist dabei ähnlich geblieben und unsere Designer modellieren die Einzelteile immer noch von Hand. Das Gesamtobjekt jedoch generiert APEX automatisch aus der Menge der Einzelteile. Die Toolchain des Frameworks bietet zudem die Möglichkeit, Meshes automatisiert in Einzelteile zu zerlegen. Weil wir jedoch die vollständige Kontrolle über »Look and Feel« der Objekte behalten wollten, haben wir uns gegen die Verwendung dieses Features entschieden. Über die von Nvidia bereitgestellten Tools können wir die APEX Destructibles zudem detailliert konfigurieren. So werden dort die Einzelteile zunächst geladen und von uns häufig so konfiguriert, dass sie dynamische Verbindungen zwischen den benachbarten Teilstücken des Objektes und der Level-Geometrie herstellen. In der Engine selbst können wir die Objekte dann mit einer erweiterten Version unserer Physikkraftfelder beeinflussen und schließlich zerstören.

Das Besondere bei der Simulation der Destructibles mithilfe von APEX ist, dass das System die Einzelteile mit Rigid Bodies simuliert, die der Form der grafischen Repräsentation sehr nahe kommen. Dazu verwendet APEX ein Verfahren, das es ermöglicht, eine oder mehrere konvexe Hüllen für die Einzelteile zu generieren, die zu der finalen Kollisionsgeometrie des Einzelstücks zusammengesetzt werden. Zudem werden die Objekte durch Verwendung von Skinning dargestellt, weshalb auch das Rendern von vielen verschiedenen Bruchstücken kein Problem ist und in einem einzigen Draw Call ausgeführt werden kann. Durch die vollständig automatisierte Verbindung der Einzelstücke können wir die zerstörbaren Objekte

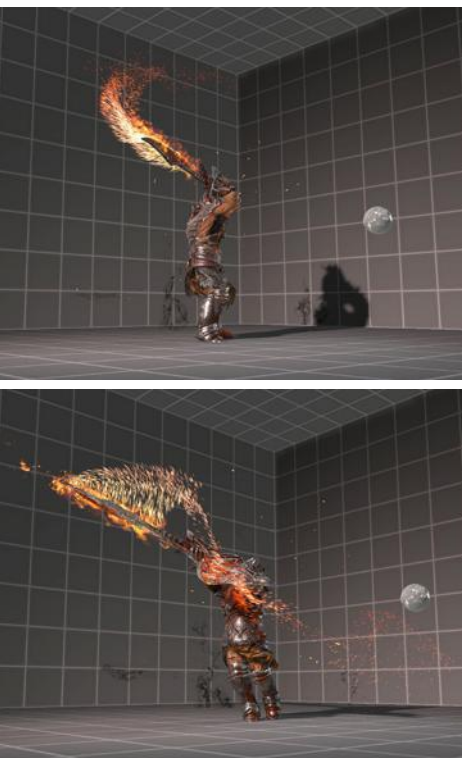


Abbildung 4 Einer unserer Bossgegner führt einen Schwerttanz auf, von dem wir mehrere aufeinanderfolgende Zeitabschnitte festgehalten haben. Die emittierten Partikel werden dabei von einer Fluid-Simulation in Echtzeit in Bewegung versetzt.

Fragment für Fragment eliminieren.

Abbildung 3 zeigt einen Schrank, den die Engine als APEX Destructible geladen hat. Mithilfe eines kleinen Kraftfelds kann das Objekt vollkommen dynamisch und passend zu der Position und Stärke der gewirkten Kraft zerstört werden. Wir verwenden Kraftfelder für sämtliche Interaktionen zwischen dem Spieler und der Umgebung. So befestigen wir Kraftfelder, die wir kurzzeitig in der Welt platzieren und aktivieren, sowohl an den Waffen als auch animationsgetrieben beim Rollen oder Springen der Charaktere.

GPU-gestützte Simulation von Partikeln mithilfe von APEX Particles und APEX Turbulence

Auch das APEX-Particles-Modul haben wir anfangs nur prototypisch zu Evaluationszwecken integriert. Im Lauf der Entwicklung haben wir das Modul für mehr und mehr Effekte in Lords of the Fallen verwendet, die mit der zu dieser Zeit verwendeten Version unseres hauseigenen Partikelsystems nicht möglich waren. Aktuell verwenden wir das Modul primär für die CPU- und GPU-basierte Simulation von Sprite- und Mesh-Partikeln, die realistisch mit der Umgebung interagieren können. Im Gegensatz zu unserem eigenen GPU-Partikelsystem basiert die Simulation auf CUDA und kann nur auf Nvidia-Grafikkarten beschleunigt ausgeführt werden. Dafür stehen uns auf unterstützten Systemen viele erweiterte Techniken zur Verfügung, die mit unserer hauseigenen Lösung nicht möglich sind.

So können die Partikel auch mit der Umgebung interagieren, wenn diese in der aktuellen Kameraperspektive nicht sichtbar ist. Hierfür zieht das System die Kollisionsgeometrie zurate und führt tatsächliche Schnitttests mit der umliegenden Umgebung aus. Zudem können wir die Partikel mit dem sogenannten »Turbulence Module« und mithilfe von Vektorfeldern in Bewegung versetzen, die wir durch Fluidsimulation generieren. Die Simulation findet dabei vollständig in Echtzeit statt.

Abbildung 4 zeigt einen Partikeleffekt, den wir für einen Angriff einer der Bossgegner

entworfen haben. Dabei befindet sich der Boss in einem dynamischen Fluidfeld, das mit dessen Position mit transformiert wird. Durch einen an der Waffe befestigten Rigid Body versetzt die Engine das Feld in Bewegung und verwirbelt die gleichzeitig emittierten Partikel. Dieser Effekt lässt sich ausschließlich auf CUDA-fähigen Nvidia-Grafikkarten simulieren. Deswegen setzen wir beispielsweise auf den Next-Gen-Konsolen eine vereinfachte Variante auf Basis unseres GPU-Partikelsystems und der dort zur Verfügung stehenden, statischen Vektorfelder ein.

Ein Blick in die Zukunft – Haar-, Fell- und Grassimulation

Viele Spiele stellen Haare und Fell durch simple, texturierte Polygone dar. Eine Textur beinhaltet dabei meist eine Vielzahl von einzelnen Haarsträhnen. Auf Bewegungen der Charaktere reagieren derartige Haare oft gar nicht oder lediglich durch vorher erstellte Animationen. Eine andere Möglichkeit, um vor allem Fell detaillierter umzusetzen, sind vorberechnete, volumetrische Methoden wie etwa »Shells and Fins«.

Mit moderneren Grafikkarten und DirectX 11 oder OpenGL 4.3 steht allerdings genug Rechenleistung zur Verfügung, um diese althergebrachten Techniken zu ersetzen. Durch Compute Shader und sogenannte UAV-Ressourcen (Texturen und Buffer, auf die Shader sowohl Lese- als auch Schreibzugriff haben) ist es möglich, mehrere tausend Haarsträhnen individuell in Echtzeit zu simulieren und zu rendern.

Eine Strähne ist dabei aus einer Reihe von Partikeln zusammengesetzt, die auf der Grafikkarte physikalisch korrekt simuliert werden.

Abbildung 5 zeigt beispielhaft den Aufbau einer einzelnen Strähne in der Simulation, mitsamt dem gerenderten Mesh.

Die Simulation berechnet die Kopfbewegungen des Charakters, aber auch die Erdbeschleunigung und Windkräfte. Weil es sich bei den simulierten Partikeln nicht um frei bewegliche Punkte handelt, sondern sie einzelne Haarsträhnen beschreiben, muss die

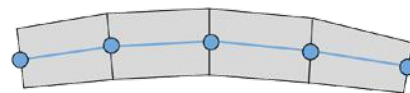


Abbildung 5 Der Aufbau einer Haarsträhne zeigt die simulierten Partikel in Blau und das resultierende Mesh in Grau.

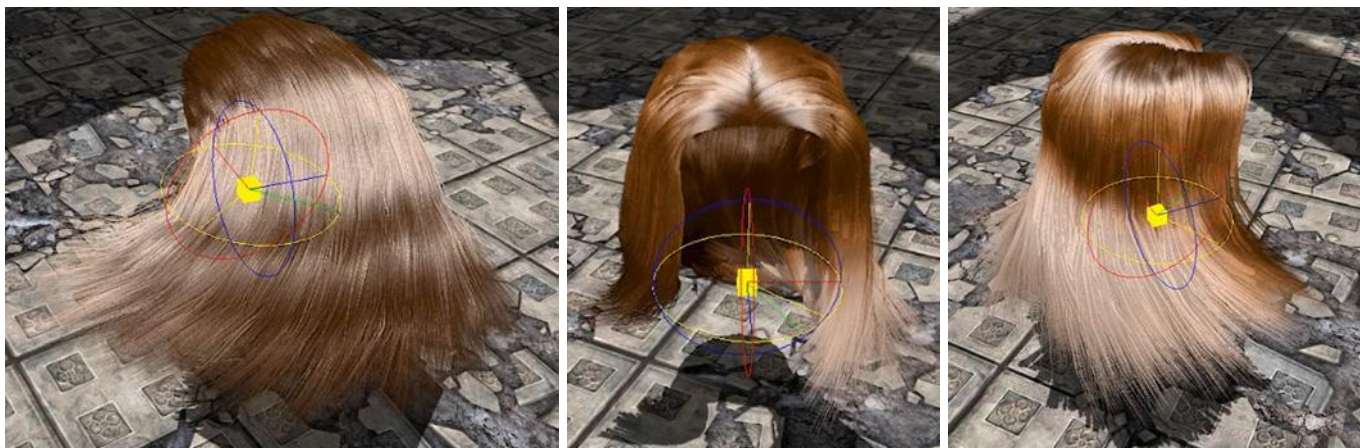


Abbildung 6 Ähnlich wie die von Crystal Dynamics verwendete Technik »TressFX« soll auch die Fledge-Engine Haare und Fell realistisch auf Bewegungen - hier im Beispiel eine Kopfdrehung - reagieren lassen.

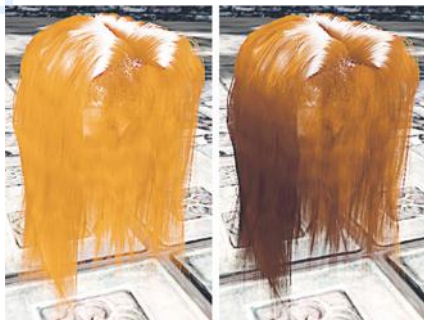


Abbildung 7 Die Schatten-Berechnung für einzelne Haarsträhnen fällt aufwendig aus, deshalb verwendet die Fledge-Engine Deep Opacity Maps dafür. Das Beispiel zeigt einmal Haare ohne Schatten (links) und einmal mit (rechts).

Engine sie zusätzlich durch eine Reihe von Constraints behandeln. Die Constraints werden durch weitere Compute Shader umgesetzt, die zum Beispiel sicherstellen, dass die Strähnen nicht übermäßig gestreckt werden. Je nach gewünschten Features können wir noch weitere Constraints berechnen lassen. Im Fall von langen, menschlichen Haaren ist es etwa vorteilhaft, wenn sich die Strähnen nach schnellen Bewegungen nicht in unerwünschten Formen verwickeln, sondern wieder in eine von den Artists definierte Frisur zurückkehren.

Neben aufwendigen Shape-Matching-Methoden ermöglicht vor allen ein neuartiges Verfahren von AMD dieses Feature, das die Frisur sowohl global als auch lokal über die Länge einer einzelnen Strähne wiederherstellt. Gerade die lokale Eigenschaft erlaubt es, auch gewellte und lockige Haare zu simulieren, ohne dass die Strähnen unnatürlich steif wirken. Diese Art der Simulation hat Crystal Dynamics unter dem Namen »TressFX« auch in »Tomb Raider« verwendet.

Bei Deck13 arbeiten wir zurzeit intensiv an einer ähnlichen Technik. Eine Frisur, die wir auf diese Weise in Fledge simulieren, ist in **Abbildung 6** (Seite 33) zu sehen. Bei der Implementierung ist es uns wichtig, dass das gleiche Framework neben Haaren auch Fell unterstützt. Im Unterschied zu Haaren simuliert die Fledge-Engine Fell mit einer geringeren Partikelzahl pro Strähne. Eine von Artists erstellte Frisur gibt es hier nicht. Stattdessen verwenden wir eine Reihe von Texturen, die verschiedene Eigenschaften beschreiben – etwa die Dichte, Länge und Richtung des Fells. Anhand dieser Texturen erzeugen wir die einzelnen Strähnen dann prozedural auf der Oberfläche eines beliebigen Modells.

Neben der Simulation ist auch das Rendern von Haar und Fell eine Herausforderung. Durch die Herangehensweise, jede Strähne individuell zu berechnen, ist es auch notwendig, jede einzelne Strähne zu rendern. Um die benötigte Polygonanzahl so gering wie möglich zu halten, haben wir uns dafür entschieden, eine Strähne nicht als dreidimensionales Mesh darzustellen. Stattdessen kommt ein zweidimensionaler Polygonzug zum Einsatz, den die Engine entsprechend der Kameraposition und der Richtung der jeweiligen Strähne ausrichtet.

Alle Strähnen einer kompletten Frisur oder Fellstruktur werden durch Instanced Rendering mit nur einem DirectX-Aufruf gerendert. Um einen möglichst weichen Eindruck zu erhalten, lassen wir die Haare als transparente Objekte darstellen und per Alpha Blending übereinanderblenden. Speziell angepasste Beleuchtungsberechnungen sorgen für anisotrope Lichtreflexionen, die typisch für Haare sind.

Leider ist es schwierig, die bestehende Deferred Shading Pipeline so zu erweitern, dass sie komplexe Transparenz-Effekte und spezielle Beleuchtungsberechnungen unterstützt. Aus diesem Grund werden Haar- und

Fellstrukturen erst nach allen anderen Objekten durch klassisches Forward Rendering berechnet und dargestellt.

Um die Farben aller transparenten Strähnen korrekt miteinander zu vermischen, müssen sie anhand ihrer Distanz zur Kamera sortiert gerendert werden. Weil die Simulation die Form und Position der Strähnen dynamisch verändert, ist es schwierig, diese Sortierung zu berechnen. In der Computergrafik ist dieses Problem zwar altbekannt, aber auch heute noch ein aktives Forschungsfeld.

Bei Deck13 experimentieren wir zurzeit mit verschiedenen Herangehensweisen. Eine Möglichkeit besteht zum Beispiel in der Verwendung von sogenannten »Per-Pixel Linked Lists« (PPLL). Dieses Verfahren sammelt pro Pixel die Farben von allen hintereinanderliegenden transparenten Strähnen, um sie dann später zu sortieren und entsprechend zu mischen. PPLL erzeugen Ergebnisse von hoher Qualität, benötigen aber viel Speicherplatz auf der Grafikkarte, um die Pixel-Listen zu konstruieren und vorzuhalten.

Eine Alternative stellen Methoden aus dem Bereich der »Blended Order-Independent Transparency« dar. Diese Verfahren setzen keinerlei Sortierung voraus und versuchen stattdessen, den Einfluss jeder transparenten Farbe auf den endgültigen Pixel mathematisch abzuschätzen und entsprechend zu gewichten. Eine derartige Berechnung kann unter anderem über die Distanz der jeweiligen Strähne zur Kamera erfolgen. Diese Technik arbeitet wesentlich schneller als PPLL und hat weniger Speicherbedarf, die Ergebnisse sind allerdings auch weniger genau.

Da die einzelnen Haarsträhnen transparent sind, fallen auch Schatten-Berechnungen aufwendiger aus als bei normalen Objekten. Die äußersten Haarschichten blockieren das Licht nicht komplett, sondern nur teilweise, und so wandert das Licht abgeschwächt weiter durch die Haarstruktur. Dieses Lichtverhalten kann eine einfache Shadow Map nicht ausreichend beschreiben. Stattdessen kommen mehrere Shadow Maps, sogenannte »Deep Opacity Maps« zum Einsatz. **Abbildung 7** zeigt diese Art der Schattierung in der Fledge-Engine.

Das Simulations-Framework für Haare und Fell lässt sich in leicht veränderter Form auch auf Gras übertragen. Jeden Grashalm simuliert das System ebenfalls mit einer zusammenhängenden Abfolge von Partikeln, auf die verschiedene Constraints wirken. Im Unterschied zu Haaren rendert jedoch eine gewöhnliche Deferred Shading Pipeline die Gras-Objekte, weil sie weder transparent noch durch spezielle Beleuchtungsberechnungen dargestellt werden müssen.

Um beliebige Gras-Modelle zu simulieren, haben Artists die Möglichkeit, in 3D-Programmen wie Autodesk Maya oder 3ds Max zu bestimmen, welche Vertices von welchen simulierten Partikeln behandelt werden. Das geschieht durch gewöhnliche Skinning-Zuweisungen an

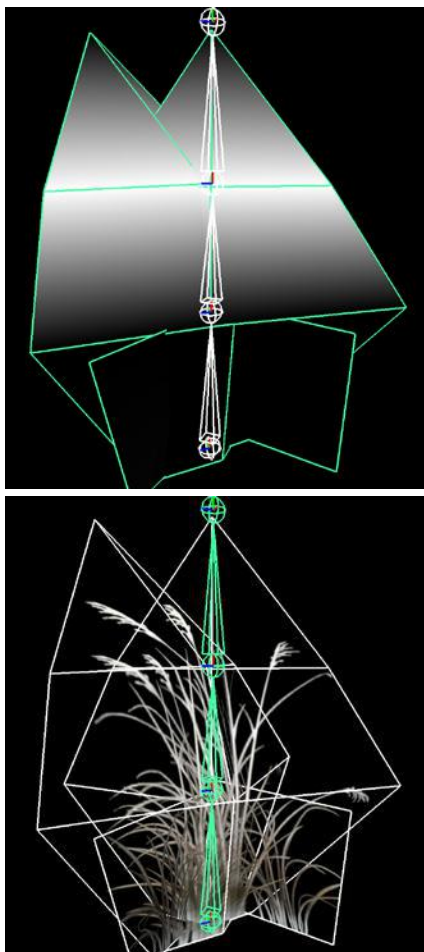


Abbildung 8 Vorbereitung von Gras-Meshes in Autodesk Maya, wo sich festlegen lässt, wo simulierte Partikel wirken sollen.



Abbildung 9 Die Engine simuliert jeden Grashalm einzeln, sodass diese sowohl auf komplexe Wind-Effekte als auch auf Kollisionen reagieren.

eine Menge von Bones, welche die simulierten Partikel zeigen. **Abbildung 8** veranschaulicht eines dieser vorbereiteten Gras-Meshes.

Weil die Engine jeden einzelnen Grashalm individuell simuliert, ist es möglich, komplexe Wind-Effekte durch dreidimensionale Vektorfelder darzustellen, welche die Simulations-Shader in Form von Texturen berücksichtigen. **Abbildung 9** zeigt einige simulierte Grashalme, auf die ein wirbelsturmartiger Wind wirkt. Neben Wind-Effekten unterstützt die Simulation auch Kollisionen mit dynamischen Objekten im Spiel, sodass sich zum Beispiel der Spieler-Charakter durch das Gras bewegen kann und dadurch die Grashalme zur Seite biegt.

Um auch große Grasfelder mit Hunderttausenden Halmen zu unterstützen, organisieren wir alle Halme in einem »Quadtree«. Dabei handelt es sich um eine hierarchische Datenstruktur, die es ermöglicht, schnell den sichtbaren Teil des Grasfeldes zu ermitteln und dann nur diesen Teil zu simulieren und zu rendern.

Screenspace Shadows

Ähnlich zu den Screenspace Reflections haben wir sogenannte »Screenspace Shadows« in die Engine integriert, um die Schatten als auch die Selbstschattierung für die Gesichter der Charaktere in Nahaufnahmen detaillierter zu gestalten. Während in Fledge die Anzahl der gleichzeitig schattenwerfenden Lichtquellen auf drei bis fünf beschränkt ist, funktioniert dieses Verfahren mit einer beliebigen Anzahl von Lichtern. Die Screenspace Shadows haben wir direkt in den Lighting Pass integriert, und grob betrachtet funktioniert das Verfahren ähnlich zu einem Schattenfühler, den Raytracing-Verfahren einsetzen. Hierzu wird bei einem erfolgreichen Schnittpunkt mit einer darzustellenden Geometrie ein weiterer Strahl in Richtung der Lichtquellen abgesendet, um festzustellen, ob der betrachtete Schnittpunkt im Schatten des gleichen oder eines weiteren Objektes der Szene liegt. Mit dem großen Unterschied, dass wir

natürlich keine komplexen Schnittpunkte gegen die Geometrie ausführen und lediglich mit den Informationen arbeiten, die uns im Screenspace im Kontext des aktuell betrachteten Frames zur Verfügung stehen. Ausgehend von der aktuellen Fragmentposition im Viewspace tasten wir einige Positionen auf einem Strahl in Richtung der Lichtquelle ab. Dabei vergleichen wir die aktuellen Tiefenwerte der Position auf dem Strahl mit den Daten des Tiefenpuffers. Falls der Strahl die Geometrie durchdringt, beginnen wir, das aktuell betrachtete Fragment um einen gewissen Faktor abzudunkeln. Entsprechend gilt: Je häufiger der Strahl die Geometrie durchdringt, desto intensiver die Abdunkelung. Durch diese Vorgehensweise können wir sogar Schatten mit der Anmutung einer sanften Penumbra erzeugen. Unsere Designer können sowohl die zu testende Distanz als auch den abzuziehenden Faktor pro Kollision justieren.

Weil das Verfahren aktuell noch mit über 64 Samples auf einem Vollbild-Render-Target operiert, verwenden wir es bis jetzt noch nicht aktiv in Cutscenes. Wir planen, an dieser Stelle Low-Resolution Rendering, Dithering und Temporale Reprojektion als Optimierung einzusetzen – ähnlich zu dem Weg, den wir bei der Implementierung des volumetrischen Beleuchtungssystems gewählt haben.

Abschließende Worte

Wir hoffen, dass wir in diesem Artikel einen Einblick in die Techniken gewähren konnten, mit denen wir Lords of the Fallen von einem visuellen Betrachtungswinkel her auf die nächste Generation vorbereitet haben. Wir sind alle sehr gespannt, wie sich Fledge in den kommenden Monaten und Jahren entwickelt und welche neuen Techniken ihren Weg in die Engine finden – besonders weil wir bereits durch das Wachstum und das Engagement unserer Teams einen kleinen, visuellen Quantensprung miterleben durften.

David Reinig, Benjamin Glatzel, Dominik Lazarek

Bibliografie

Vlachos, Alex

»Water Flow in Portal 2«

http://www.valvesoftware.com/publications/2010/siggraph2010_vlachos_waterflow.pdf

Lengyel, Jerome, Praun, Emil, Finkelstein, Adam & Hoppe, Hugues

»Real-Time Fur over Arbitrary Surfaces«

<http://research.microsoft.com/en-us/um/people/hoppe/fur.pdf>

Müller, Matthias, Heidelberger, Bruno, Teschner, Matthias & Gross, Markus

»Meshless Deformations Based on Shape Matching«

http://www.beosil.com/download/MeshlessDeformations_SIG05.pdf

Dongsoo, Han

»Real-time Hair Simulation with Efficient Hair Style Preservation«

<http://www.potentialmotion.com/home/hairsimulation>

McKee, Jay

»Real-time concurrent linked list construction on the GPU«

http://developer.amd.com/wordpress/media/2013/06/2041_final.pdf

McGuire, Morgan & Bavoil, Louis

»Weighted Blended Order-Independent Transparency«

<http://jcgf.org/published/0002/02/09/paper.pdf>

Yuksel, Cem & Keyser, John

»Deep Opacity Maps«

http://www.cemyuksel.com/research/deepopacity/deepopacitymaps_techreport.pdf



Der Vergleich zeigt die Wirkung von Screenspace Shadows: Während oben der Effekt deaktiviert ist, hebt er unten selbst kleine Nuancen des Charakters hervor – erkennbar zum Beispiel an den Hautfalten an der Stirn.

DER KONSOLENMARKT ALS NICHE

Die japanische Spieleentwicklung entfernt sich immer weiter von den klassischen Konsolen. »Harvest Moon«-Schöpfer Yasuhiro Wada über sich verändernde Märkte, die Arbeit an vollkommen unterschiedlichen Projekten sowie die Unterschiede zwischen der westlichen und der japanischen Entwicklerszene.



Yasuhiro Wada
ist CEO von Toybox.

Yasuhiro Wada begann seine Karriere in der Videospieleindustrie 1991 als Producer bei Pack-in-Video. 1996 wurde Pack-in-Video von der Spieleabteilung von Victor Entertainment übernommen. Unter dem Studienamen Victor Interactive Software schuf Wada das erste Spiel der »Harvest Moon«-Serie, die inzwischen über 20 Spiele umfasst. Nach einem kurzen Zwischenstopp 2009 bei Grasshopper Manufacture hat er 2011 sein eigenes Studio namens Toybox gegründet.

Making Games In den Spieleserien »Harvest Moon« und »Rune Factory« dreht sich alles darum, etwas aufzubauen: Pflanzen zu züchten und ähnliche Aufgaben zu erfüllen. In »Hometown Story« geht es nun darum, einen Laden wieder aufzubauen. Was macht für dich die Faszination dieses Genres aus?

Yasuhiro Wada Meiner Meinung nach macht die Faszination dieser Spiele aus, dass du deine Pläne selbst aufstellen kannst – und sich alles genau so aufbauen lässt, wie du es möchtest. Jeder Spieler macht so eine einzigartige Erfahrung, weil jeder seinen eigenen Spielstil hat. Manchmal kreieren sie sogar einen ganz neuen Stil, den wir Entwickler so gar nicht auf dem Zettel hatten. Und das wiederum fasziniert mich an dieser Art von Spielen.

Making Games Du lässt Erfahrungen aus deinem Leben in deine Spiele einfließen. Wie zeigt sich das in »Hometown Story«?

Yasuhiro Wada In »Hometown Story« ging es mir vor allem darum, Charaktere und Geschichten zu erschaffen. Du triffst viele Charaktere, musst dich aber von genauso vielen wieder verabschieden – und jeder von ihnen hat seine ganz eigene Geschichte. Diese Geschichten sind auf jeden Fall von all den Leuten inspiriert, denen ich im Lauf meines Lebens begegnet bin. Und natürlich von all den Geschichten, die ich gelesen habe, und all den Inspirationsquellen, die so viele großartige Künstler in der Vergangenheit erschaffen haben.

Making Games Du hast auch an »No More Heroes« und »Deadly Premonition« gearbeitet. Diese Spiele entsprechen nicht gerade deiner restlichen Arbeit. Welche neuen Erfahrungen hast du bei diesen Projekten gemacht?

Yasuhiro Wada Als Entwickler habe ich viele herzerwärmende Spiele gemacht. Als Spieler interessieren mich aber auch First-Person-Shooter, Action-Adventures und zahlreiche andere

Genres. Was »No More Heroes« und »Deadly Premonition« angeht – ich habe mit den Schöpfern (Suda 51 & Swery 65, Anm. der Red.) dieser Spiele zusammengearbeitet, die sehr starke Charaktere sind und dadurch das Team immer wieder angespornt haben.

»Spielerentwickler müssen bei allem Enthusiasmus auch einen objektiven Blick auf ihre Arbeit haben.«

Making Games Wie haben diese Projekte deine Arbeit im Allgemeinen beeinflusst?

Yasuhiro Wada Diese Projekte haben nicht nur meine Arbeit beeinflusst. Es war für mich eine großartige Erfahrung, die Schöpfer dieser Spiele zu unterstützen und damit fantastische Spiele auf den westlichen Markt zu bringen, die einen bleibenden Eindruck bei den

Spielern hinterlassen haben. Außerdem freut es mich natürlich sehr, dass mein Name dadurch auch Spielern ein Begriff geworden ist, die meine eigenen Titel nicht gespielt haben.

Making Games Hast du ein spezielles Publikum im Hinterkopf, wenn du Spiele entwickelst?

Oder folgt dein Entwicklungsprozess eher dem Ansatz »Wenn der Titel mir Spaß macht, dann werden auch andere Spieler ihn mögen«?

Yasuhiro Wada Das spielt eine wichtige Rolle – die Spiele müssen mir selbst auf jeden Fall Spaß machen. Daher habe ich auch immer eine Sicht auf meine Spiele, die vom Blickwinkel des Schöpfers abweicht. Wir Spieleentwickler müssen bei allem Enthusiasmus trotzdem auch einen objektiven Blick auf unsere Arbeit haben. Ich muss mich, während ich an einem Spiel arbeite, ständig fragen, ob es wirklich etwas Neues ist, woran ich arbeite, oder ob es nicht doch Ähnlichkeiten mit einem Spiel hat, das es schon gibt. Manchmal ist das gar nichts, was die Spieler unbedingt brauchen. Aber meiner Meinung nach ist dieses Überraschungselement genau das, was den Spielern am Ende neue Erfahrungen und vor allem Spaß bringt. Daher halte ich bei meinen Spielen immer an diesem Punkt fest.



Die Charaktere in »Hometown Story« haben alle ihre eigene Geschichte: »Diese Geschichten sind von all den Leuten inspiriert, denen ich im Lauf meines Lebens begegnet bin.«



Die Mitarbeit an »Deadly Premonition« (links) und »No More Heroes« (rechts) hat Wada geprägt: »Es war für mich eine großartige Erfahrung, die Schöpfer dieser Spiele zu unterstützen.«

Making Games Du arbeitest schon lange in der Spielebranche. Wie hat sich die japanische Entwicklerszene im Lauf der Zeit verändert? Früher lag ihr Fokus auf Konsolen. Wie macht es sich bemerkbar, dass mobile Endgeräte (Smartphones, Tablets) immer wichtiger werden?

Yasuhiro Wada Bis zum Beginn der Playstation-2-Ära waren die Möglichkeiten der japanischen Entwickler im Hinblick auf Konsolenspiele auf eine allumfassende Weise sehr effizient. Danach wurde der Umfang der Spieleentwicklung jedoch immer größer, und dadurch sind auch die Budgets stetig gewachsen. Als Resultat dieser Vorgänge wurde es zwangsläufig notwendig, den ganzen Entwicklungsprozess straffer zu organisieren. Einige der großen Publisher und Entwickler sind diesem Weg gefolgt – und auch Unternehmen mittlerer Größe haben es versucht. Allerdings sind sie daran gescheitert und suchten andere Wege finden, um ihre Risiken bei der Entwicklung zu minimieren. Der Markt wurde stetig breiter gefächert und die Publisher haben erkannt, dass sie die hohen Entwicklungskosten nicht wieder einspielen können, wenn sie nur einen Sektor bedienen. Die großen Publisher konnten auf dem weltweiten Markt zwar noch genug abschöpfen, um die Kosten zu decken, aber alle anderen müssen sich auf charakterzentrische Spiele oder Otaku-Games verlagern, wenn sie ohne Verlustrisiko entwickeln wollen. Allmählich ist der Konsolenmarkt auf die Größe eines Nischenmarkts geschrumpft. Währenddessen wurde die Performance von mobilen Endgeräten immer besser, und heute kann jeder spielen, ohne dafür eine Konsole einschalten zu müssen. Die Nutzer haben nach und nach zu diesem neuen Markt gewechselt.

Making Games Wo siehst du die Unterschiede zwischen der westlichen und der asiatischen Entwicklerszene?

Yasuhiro Wada In der japanischen Entwicklerszene ist es üblich, dass ein einzelner Game Director das gesamte Projekt leitet. Ich habe den Eindruck, dass in der westlichen Spieleentwicklung mehr in unterschiedlichen Abteilungen gearbeitet wird. Der Game Director ist zwar

wichtig, aber meiner Meinung nach ist die Entwicklung insgesamt organisierter.

Making Games Wie haben sich die unterschiedlichen Spielkulturen in den vergangenen Jahren gegenseitig beeinflusst?

Yasuhiro Wada Wirklich beeinflusst haben sie sich meiner Meinung nach nicht. Bei den Konsolenspielen habe ich den Eindruck, dass die japanischen Spiele mehr für ein Nischenpublikum gemacht werden, während die westlichen Spiele wesentlich mehr Nutzer ansprechen sollen. Man könnte sagen, dass es auf der einen Seite das Otaku-Geschäft gibt und auf der anderen das Hollywood-Geschäft.

Making Games Was können westliche und asiatische Entwickler voneinander lernen?

Yasuhiro Wada Vor über zehn Jahren hatte ich den Eindruck, dass westliche Spiele zwar technisch stark waren, als tatsächliche Spiele allerdings zu schwerfällig. Japanische Titel waren im Gegensatz dazu nutzerfreundlicher. Die westlichen Spiele haben diesen Punkt inzwischen für sich übernommen und angepasst. Meiner Meinung nach werden die japanischen Entwickler im Gegenzug die organisatorischen Strukturen und die Technik der westlichen Spieleindustrie immer stärker adaptieren.

Making Games Welche westlichen Spiele haben bei dir einen bleibenden Eindruck hinterlassen?

Yasuhiro Wada Lass mich ein paar Jahre zurückdenken ... da fallen mir »Heavy Rain« und die »Uncharted«-Spiele ein. Und natürlich »Oblivion«, die »GTA«-Reihe, »Red Dead Redemption« und noch einige andere.

Making Games Welche drei asiatischen Spiele sollte sich jeder westliche Entwickler zumindest mal angesehen haben?

Yasuhiro Wada Das Fußballspiel »Calcibit« für den 3DS, das klassische JRPG »Bravely Default« (ebenfalls für den 3DS) und das 2D-Fantasy-Action-RPG »Dragon's Crown« für PlayStation 3 und PS Vita sollte man auf jeden Fall mal gespielt haben.

Interview: Patricia Geiger

»Ich habe den Eindruck, dass die japanischen Spiele mehr für ein Nischenpublikum gemacht werden.«



GPU-UNTERSTÜTZTE GENERIERUNG DES TERRAINS FÜR DAS SPIEL GRAND AGES: MEDIEVAL

Gesamteuropa als Schauplatz eines Strategiespiels nachzubauen, wäre ein kleinteiliges und zeitintensives Unterfangen. Mark Külker zeigt auf, wie die Kalypso-eigenen Gaming Minds Studios mithilfe einer GPU-unterstützten Terrain-Engine die Erstellung der Spielwelt komplett automatisiert haben.



Mark Külker
ist Technical Artist bei
Gaming Minds Studios.

Mark arbeitet seit 1997 in der Games-Branche. Er begann zunächst als Renderer sowie 3D-Modeller und wechselte einige Jahre später in die technische Grafik. Hier gehören die Erstellung der Asset-Pipelines, Scripting und Tool-Programmierung zu seinem Tätigkeitsfeld. Sein Schwerpunkt und damit verbunden seine unschätzbaren Erfahrungen liegen jedoch im Bereich der Shader-Programmierung. Seit ihrer Gründung im Jahre 2009 ist Mark technischer Grafiker bei den Gaming Minds Studios.

Für unseren demnächst erscheinenden Titel *Grand Ages: Medieval* war schon frühzeitig klar, dass wir uns mit der Erstellung und dem Balancing eines riesigen Areals auseinandersetzen mussten. Geplant war ein Ausschnitt der realen Weltkarte,

der sich über den gesamten europäischen Raum und angrenzende Bereiche erstreckt. Über die grafische Darstellung hinaus mussten Regionen festgelegt werden, aus deren Definition sich eine Verteilung von etwa 20 Ressourcen wie Holz, Lehm oder Eisen ableiten ließ.

Äußere Anforderungen

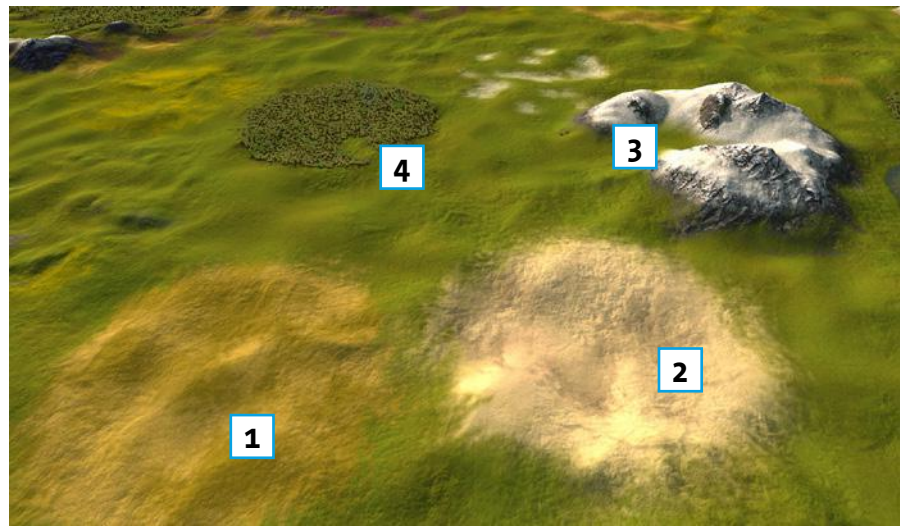
Es sollte über die gesamte Dauer des Projekts möglich sein, die Platzierung und Ausdehnung der verschiedenen Regionen zu verändern. Dies war insbesondere deswegen wichtig, weil ein direkter Zusammenhang zwischen verfügbaren Ressourcen (Gameplay-Balancing) und der grafischen Darstellung gewünscht war.

Bereits zu Beginn der Entwicklungszeit sollte deshalb eine brauchbare Repräsentation des Terrains als Grundlage für die Spiellogik und die weitere Konzeption vorhanden sein. Das Terrain musste sich somit früh in Funktion und Darstellung verwenden lassen.

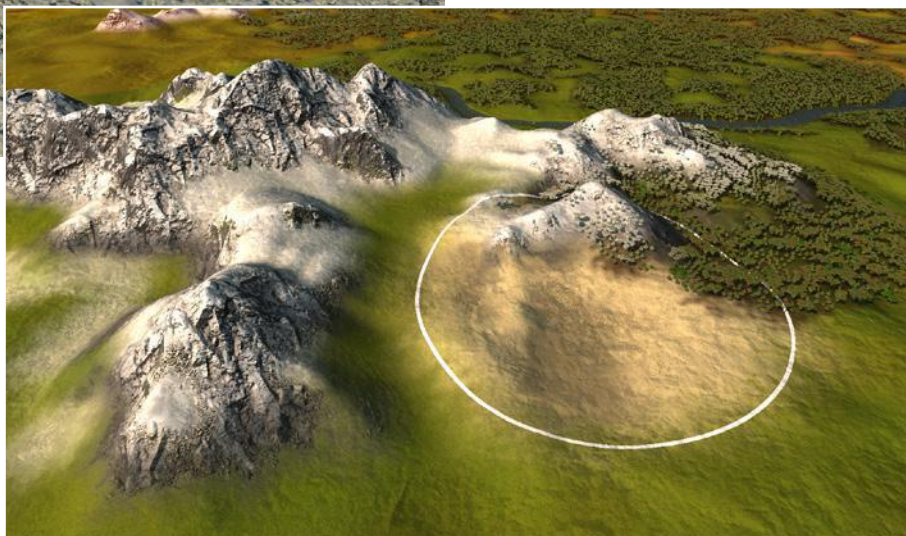
Zu guter Letzt wollten wir es dem Spieler ermöglichen, dass er bis zur Darstellung einzelner Häuser heranzoomen kann und dennoch so weit herauszoomen darf, bis nahezu die gesamte Spielwelt sichtbar ist.

Technische Anforderungen

Für die Topografie und die Küstenlinien der Spielwelt zogen wir reale Höhenkarten heran, jedoch war diese zu hochfrequent. Daher übernahmen wir aus den realen Daten zwar die genauen Küstenverläufe, befolgten die Höhenzüge für die Gebirge jedoch nur grob. Alle anderen Terrainarten bestimmten beziehungsweise definierten wir selbst. Für die Logik war es wichtig, jeden Ort auf der Karte eindeutig



Dieser Screenshot aus dem Editor zeigt ein flaches Wiesenland **1**, mit dem einzelne Ressourcen-Maps verrechnet wurden. Vorn rechts ist Wüste **2**, dahinter Gebirge **3** und hinten links Waldgebiet **4**.



Der Level-Designer kann die einzelnen Ressourcen direkt im Editor platzieren. Die Auswirkungen werden sofort sichtbar, ebenso die Übergänge.

einer Ressource wie Holz, Lehm oder Eisen zuordnen zu können. Wir mussten also eine Lösung entwickeln, die es uns ermöglicht, das Terrain grafisch ansprechend zu editieren und dennoch immer eine klare Repräsentation der zugehörigen Ressource zu erhalten.

Dazu kam, dass unterschiedlichste Elemente, die teilweise zusammenspielen müssen, eine Region repräsentieren sollen. Wälder beispielsweise symbolisieren Regionen, in denen der Spieler Holz abbauen kann. Ziel war es aber, dass der Baumbestand für die jeweilige Region Europas typisch ist. Demzufolge mussten Höhenlage beziehungsweise das Klima den Baumbestand und -typ beeinflussen. Wir hatten geplant, dass etwa Palmen im südlichen Flachland den Regionstypus Wald darstellen, während im nördlichen Hochland Kiefern und Tannen wachsen.

Mittels theoretischer Überlegungen zum Gameplay legten wir schließlich die Größe der Ressourcenkarte fest. Diese verfügt insgesamt über mehr als 600.000 Felder, die praktisch in jeder möglichen Konstellation vorliegen können (etwa Wald neben Wiese neben Gebirge).

Deshalb suchten wir nach einer Möglichkeit, die es dem Level-Designer erlaubt, Änderungen in Echtzeit nachzuverfolgen, sodass er direkt erkennt, wie sich Veränderungen an einem Waldgebiet auf eine Region auswirken.

Aufgrund der Vielzahl von Gebietskonstellationen sollte die Engine automatisch weiche Übergänge errechnen, damit etwa zwischen Bergen, Wiesen, Tälern oder Strandbereichen keine unrealistischen Ränder entstehen. Liegt beispielsweise eine Bergregion neben Flachland mit Wald, so sind drei Übergänge notwendig:

- Die Bergtopografie muss in die Flachlandtopografie übergehen.
- Der Wald muss weich auslaufen, das heißt, die Baumgrenze darf nicht scharf sein.
- Die Bodentexturen müssen ineinander übergehen.

Workflow

Damit alle Bereiche (Level-Design, Grafik und Terrain-Generierung) gleichzeitig und flüssig arbeiten konnten, entschieden wir uns für einen semiprozeduralen und nicht destruktiven Ansatz. Zur Erstellung des fertigen Terrains ver-

Das Spiel

Der Kalypso-eigene Entwickler Gaming Minds verlässt das Genre der Handelssimulationen und wendet sich mit »Grand Ages: Medieval« der Globalstrategie zu. Dazu bauen die Gütersloher eine über 30 Millionen Quadratkilometer große mittelalterliche Spielwelt. In dieser können auf der Strategen an jedem gewünschten Ort eine Siedlung errichten, große Armeen befehligen, mit 20 Waren europaweit Handel treiben und 50 technologische Errungenschaften erforschen. Das authentische hochmittelalterliche Setting schafft die passende Atmosphäre für eine groß angelegte Kampagne, diverse Sandbox-Modi und 8-Spieler Multiplayer-Modus.





Abseits der automatischen Generierung der Spielwelt erlaubt die Engine auch das stufenlose Zoomen. So kann der Spieler Städte und Dörfer aus der Nähe bis hin zu einzelnen Gebäuden und deren Details oder die Karte in ihrer Gesamtheit betrachten.



Die Platzierung von Ressourcen wie Holz richtet sich nach der Beschaffenheit der Region. Nur wenn das Wachstum der Pflanzen realistisch möglich ist, platziert die Engine die Bäume entsprechend. Außerdem bestimmt das Klima ob etwa Palmen oder ein Nadelwald sprießen.



Bei der Gestaltung der Welt legen die Entwickler ihren Fokus eher darauf, dass die Übergänge hübsch gelingen, als dass die Gegebenheiten mit der Realität übereinstimmen. Deshalb sehen die Alpen nicht zwingend so aus, wie das echte Gebirge.

rechnet das Programm nacheinander folgende Daten und Maps:

- Höhenkarte mit Küstenlinien und Mulden
- Flachland mit Hügeln
- Gebirgskarte zur Kennzeichnung der Bergregionen
- eine eigene Karte für jede einzelne Ressource (Holz, Lehm, Eisen und so weiter)
- Flüsse mittels Polygonzügen

Um alle notwendigen Darstellungen und ihre Übergänge umsetzen zu können, erstellte das Level-Design 16 Regionskarten und pflegte diese fortlaufend während des Projektes. Diese Karten wurden binär erstellt und in einem Folgeschritt in »Signed Distance Fields« umgewandelt. Dieser Schritt war notwendig, um je nach Region angepasste Übergangsbereiche definieren und diese grafisch darstellen zu können.

Ist ein Areal beispielsweise als Gebirge definiert, so berechnet das Programm das genaue Aussehen der Berge, indem es ein wiederholendes Gebirge-Pattern zu Hilfe nimmt und außerdem die Eigenschaften der umliegenden Gebiete berücksichtigt. Dies bedeutet auch, dass uns der schöne Übergang von zwei unterschiedlichen Regionen wichtiger war, als dass unsere Alpen genauso aussehen wie die echten Alpen.

Die finale Karte wird wie folgt definiert:

- die Höhenstruktur unter Zuhilfenahme der Höhenkarte und der prozeduralen Modifikationen durch das Gebirge-Pattern
- Selektion von Oberflächenstrukturen und Details
- Einfärbungen, abhängig von der Region in der Welt
- Bepflanzung durch Bäume und Bodenpflanzen
- Erstellung der Ressourcenkarte für die Programmlogik

Das Programm wendet alle Regionstypen in einer bestimmten Reihenfolge auf das finale Terrain an. Dafür kann jedes Areal auf die generierten Daten der vorangegangenen zugreifen und entsprechend darauf reagieren. Beispiele: Lehmgruben auf der einen Seite erscheinen nur dann, wenn in der Lehm-Ressourcenkarte Lehm aktiviert ist und an der Position auch eine Senke vorliegt.

Bäume auf der anderen Seite erscheinen nur, wenn in der entsprechenden Ressourcenkarte Wald definiert ist und das Programm einen Untergrund errechnet hat, der das Wachstum von großen Pflanzen erlaubt (kein steiniger Boden, nicht zu steil, keine Wasserfläche).

Metalle als drittes Beispiel können nur abgebaut werden, wenn an der Metallposition auch ein gültiges Gebirge besteht. Texturfärbungen machen die Abbaumöglichkeit von Metall sichtbar.

Ein Großteil der Berechnung der für die Realtime-Darstellung nötigen Daten findet auf der GPU statt. Jede Art von Modifikation lässt

sich durch einfaches Malen auf dem Terrain bewerkstelligen. Den Bereich des Terrains, den die Veränderung beeinflusst, generiert die GPU im selben Frame und wendet ihn direkt auf das Terrain an.

Texturerstellung

Die Erstellung der für das Terrain notwendigen Texturen erfolgt in mehreren Schritten:

- Erstellung des »Signed Distance Field« für alle Regionen
- Erstellung der Höhendaten des gesamten Terrains
- Berechnung der Farb- und Strukturindex-Map des gesamten Terrains
- Erstellung der Normal-Map für das Terrain
- Berechnung der Map für die Vegetation des gesamten Terrains
- Berechnung der Ressourcen-Map des gesamten Terrains
- Filterung der Farb-Map, um eventuell auftretenden Übergangartefakten entgegenzuwirken

Der Shader

Zur Generierung des fertigen Terrains schrieben wir einen umfangreichen Shader. Er verrechnet nacheinander die notwendigen Daten und Maps und wird für die verschiedenen Texturerzeugungen mit unterschiedlichen Parametern aufgerufen. Die Reihenfolge der Verrechnung der Basisdaten ist dabei fest vorgegeben und von Bedeutung.

Hierzu noch einmal das Beispiel mit der Lehmgrube: Zunächst berechnet der Shader Gebirge, Hügel, Flachland und Mulden. Anschließend wird die Lehm-Ressourcenkarte angewendet. Sie legt fest, wo sich später Lehm abbauen lässt und wo nicht.

Dieser Vorgang benötigt auf der GPU für das gesamte Terrain etwa ein bis zwei Sekunden und findet im Normalfall nur beim ersten Start des Editors statt. Weil fast alle Änderungen lediglich kleine Bereiche betreffen, also nur ein Bruchteil neu erzeugt werden muss, ist das Editieren auf einem Mittelklasse-PC quasi in Echtzeit (30-60 fps) möglich.

Durch das entkoppelte Erstellen der Ressourcen-Maps und deren Anwendung können Änderungen an jeder Stelle vorgenommen werden, ohne dass eine spezielle Nacharbeit notwendig ist.

Beispiele: Sollen etwa die Höhendaten der Berge angepasst werden, muss nur die Gebirge-Map bearbeitet werden.

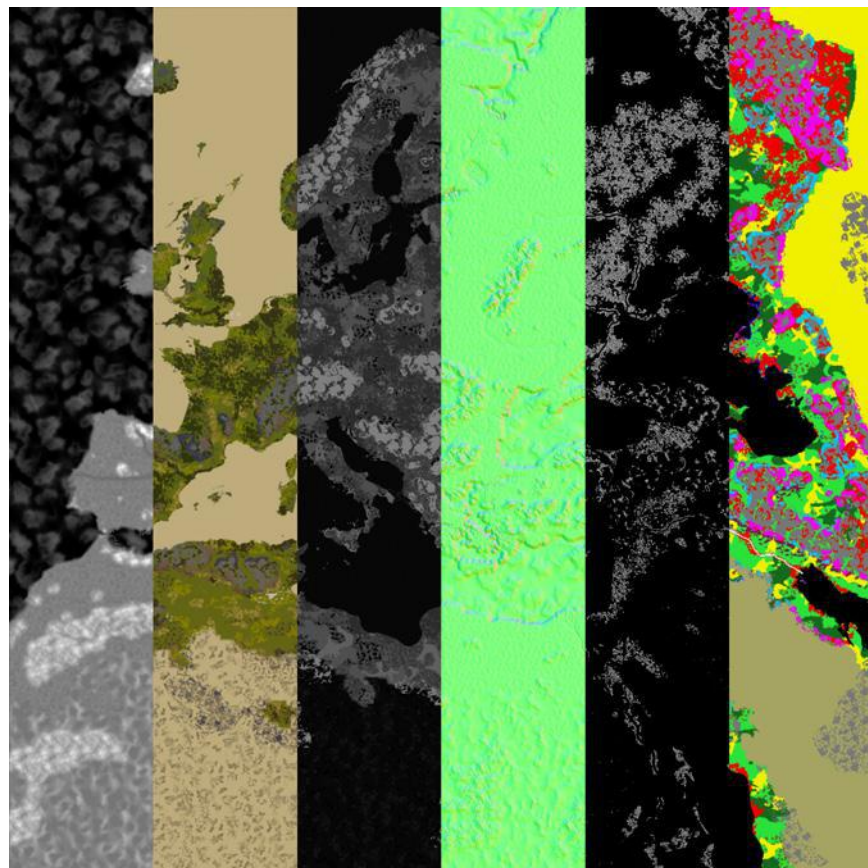
Sollen dagegen aus Gründen des Balancings weniger Lehmgruben verfügbar sein, dann passen wir den entsprechenden Shader-Code oder dessen Parameter an. Lehmgruben fallen in der betroffenen Region später kleiner aus oder erscheinen weniger häufig.

Sobald alle Veränderungen vorgenommen wurden, können die finalen Daten abgespeichert werden und stehen im Spiel bereit. Zusätzlich speichert das Programm immer

So entsteht Europa in Grand Ages: Medieval

Dies ist die verwendete Europakarte mit den sechs Maps, deren Basisdaten die Engine zur Generierung des Terrains miteinander verrechnet.

Die sechs Maps sind von links nach rechts: die Höhenkarte, die Farb-Map, die Strukturindex-Map, die Normal-Map, die Vegetation-Map und schließlich die Ressourcenkarte. Letztergenannte sagt der Programmlogik, wie welcher Rohstoff abgebaut werden kann.



die aktuellen Basisdaten ab, damit wir beim nächsten Start des Editors an gleicher Stelle weiterarbeiten können und eine erneute Erzeugung möglich ist.

Darstellung in Spiel und Editor

Die Darstellung im Spiel unterscheidet sich nicht von der im Editor. In beiden Fällen entsteht das Terrain durch die permanente Anwendung der generierten Höhenkarte in Echtzeit. Der Index der Struktur-Map dient zur Wahl der benötigten Texturen aus einem Array. In den Übergangsbereichen werden die Texturen automatisch überblendet.

Wir fügen in den Bereichen, die von Fels bedeckt sind, bei naher Kameraansicht auf das Gebirge zusätzliche Höhendetails hinzu. Das soll eine authentische Komplexität erzeugen. Diese Details haben eine größere Auflösung als das Terrain selbst, sind aber eben nur in der Nahansicht zu sehen.

Mark Külker



FOUR LAYERS

A NARRATIVE DESIGN APPROACH

Turning simple puzzles into narrative experiences is not easy to accomplish. Thomas Grip explains what you have to keep in mind when you want to interweave gameplay and story elements and gives a step-by-step introduction to the »4 Layers Approach«.



Thomas Grip is Creative Director and Co-Founder at Frictional Games.

Thomas Grip has been part of making the Penumbra and Amnesia series and has a disturbing fascination for anything horror. He is currently hard at work on SOMA, a sci-fi horror game about the deep sea and the mystery of consciousness.

This article will be about a new way to approach narrative design in games – the »4 Layers Approach«, as I call it. The article is based on a GDC talk I gave in March this year. The approach is primarily meant to suggest a workflow that focuses on the story and makes sure the narrative and gameplay are connected. The end goal is to create games that provide a better interactive narrative.

Narrative Basics

First off, »narrative« will need to be defined. At its most fundamental level, the narrative is what happens as you play the game over a longer period. It is basically the totality of the experience, something that happens when all elements are taken together: gameplay, dialog, notes, setting, graphics etc. Basically, it's the player's subjective journey through the game. I know this clashes with other definitions that refer to narrative as a separate aspect of the game, but I think this is the one that's most helpful when discussing game design. It also fits with job titles such as »narrative designer«, who is a person that doesn't just deal with writing or cut-scenes, but who works at a much higher level.

Let's compare this to the other basic elements of a game. Looking at a game second-by-second, you see the core mechanics. Moving up to look at it using a time-frame of minutes, you see tactics and problem-solving (which also includes things like puzzles). Higher still, often on the scale of hours, you see the narrative. Almost all game design is focused on the two lower levels, mechanics and tactics, and narrative mostly comes out as a sort of byproduct. Designing the narrative becomes a sort of patchwork process, where you try and create a coherent sense of storytelling from the small gaps left behind by the layers below. For instance, in games based on combat mechanics the narrative usually just acts as a form of set-up for encounters and is heavily constrained by how the fights work and so forth.

So a crucial step towards better storytelling in games is to pay at least as much attention to the narrative layer as to the other two layers, mechanics and tactics. It is important to not devote all the focus to the story though; having a symbiosis between all of layers is a core element of what makes video games special. If we want a proper interactive story, we need to preserve this.

Simply saying that we want to put more focus on the narrative level is still pretty vague; it doesn't tell us anything useful. So I'll make it a bit more concrete by listing five required cornerstones of an interactive story. This is where we get into highly subjective territory, but that can't be helped – there's a wide span of opinions on how narrative and gameplay should work together (some would even object to having focus on the narrative layer at all!). But in order to move on we need to have something concrete; if we just continue to talk in vague terms of »improving storytelling«, any suggestion can be shot down on the basis of some personal preference. Doing it like that will just get us stuck in boring discussions and it becomes much harder to set a proper goal.

Core Elements of Storytelling

The following elements shouldn't prove too controversial and I think most people will agree with them. But it still feels important to acknowledge that this is an opinion and not something I regard as an eternal truth. That said, here are my core requirements for a game with focus on narrative.

1) The focus is on storytelling.

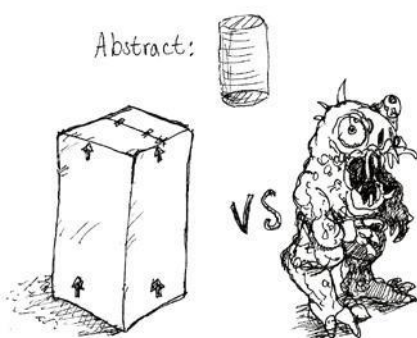
This is a trivial requirement, but still way too uncommon. Basically, the main goal of the game should be for the player to experience a specific story.

2) The bulk of the gameplay time is spent playing.

We want interactive storytelling, so players should play, not read notes, watch cutscenes, etc. These things are by no means forbidden, but they should not make up the bulk of the experience.



Narrative is basically the player's subjective journey through the game, brought to life by story elements.





Neither »Heavy Rain« (left) nor »BioShock« (right) fulfill the core requirements of narrative.

3) The interactions make narrative sense.

This means actions that:

- Move the story forward.
- Help the player understand their role.
- Are coherent with the narrative.
- Are not just there as padding.

4) There's no repetition.

Repetition leads to us noticing patterns, and noticing patterns in a game system is not far away from wanting to optimize them. And once you start thinking of the game in terms of »choices that give me the best systemic outcome«, it takes a lot of focus away from the game's narrative aspects.

5) There are no major progression blocks.

There is no inherent problem with challenge, but if the goal here is to tell a story, then the player should not spend days pondering a puzzle or trying to overcome a skill-based challenge. Just as with repetition this takes the focus away from the narrative.

Good Examples to Strive For

Now for the crucial follow up question: what games satisfy these requirements? Does »Heavy Rain« manage this? Nope, there's too little gameplay (requirement #2). »BioShock«, with all the environmental storytelling? Nope, too much shooting (requirement #4). These two games symbolize the basic issues almost all video game storytelling have: either you do not play enough, or most of what the gameplay does is not related to the narrative.

There are a few good examples, though. »Thirty Flights of Loving« is a game that I think lives up to the requirements. But the problem here is that the storyline is extremely fuzzy and disjointed. The game is a series of vaguely connected scenes, and is lacking a certain pure storytelling quality.

We come much closer to finding something that lives up to the requirements by looking at specific sections in games. Two good ones are the giraffe scene in »The Last of Us« and the end sequence in »Brothers: A Tale of Two Sons«. Both of these sections have this strong sense of being inside a narrative and fulfill my requirements. You are definitely playing a story here. But these

are just small scenes in a much larger game, and that larger game breaks most of the core elements that I have gone over. So what we really want is to have a full game filled with these sorts of sections. That would be perfect!

However, that isn't possible. These scenes depend on tons of previous game content and are extremely hard to set up. You cannot just simply strive to fill the game with stuff like this, it's just not doable. In order to get a game that consistently evokes this feeling, we have to approach it from a different direction.

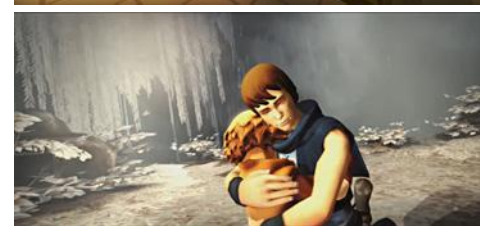
This leads us to the main bulk of this post, where I'll talk about a way to achieve this. This is an approach named »4 Layers« and the basic idea is to not attack the problem directly, but reduce it into steps and thereby be able to get proper interactive storytelling into just about any section of the game.

The 4 Layers Approach

The framework is something that's been developed between myself and Adrian Chmielarz, the man responsible for »Painkiller«, »Bulletstorm«, etc. At Frictional Games we are using this a cornerstone for our new game »SOMA«, and Adrian's new company, The Astronauts, is using it for their upcoming »The Vanishing of Ethan Carter«.

The way this approach works is that you divide the design process into four big steps. You start with the gameplay and then work your way through adding more and more layers of storytelling. The additional layers are called »Narrative Goal«, »Narrative Background« and finally »Mental Modeling«.

Before I get more in-depth it is important to note that in order to use this approach correctly, the game must be broken down into scenes. Each scene could be a puzzle, an enemy encounter, and so on. Normally, gameplay refers to how the game plays out as a whole, but for this framework, we must split it up into sections. This is connected with the above requirement of not having repetition, and usually means that there needs to be a lot of logic and gameplay coded into the world. I think that this presents a crucial piece of the puzzle for



»Thirty Flights of Loving« (upper image), the end sequence of »Brothers: A Tale of Two Sons« (middle) and the giraffe scene in »The Last of Us« (lower image) are closer to meeting the requirements, but the games lack in other aspects.

having better storytelling: to drop the need for an overarching play loop and instead make the gameplay fit each specific scene of the game.

So instead of having the gameplay describe the player's overall experience of the game, the narrative will provide this structure. Exactly how this is done will become more apparent as we go through the different layers.

Layer 1: Gameplay

First we need to start with the basic gameplay and it's crucial that the narrative aspects are kept in mind from the get-go. If the gameplay doesn't fit with the story, then problems will start to accrue and it'll make the later layers much harder to achieve and reduce the final quality. As a first step for ensuring this, there are four basic rules that must be followed.

Coherency

The gameplay must fit with the game's world, mood and characters. There should be no need for double-thinking when performing an action; it should fit with what has been laid out by the narrative. The player should be able to think about the actions made to get a deeper understanding of the game's story. What the player does must also make some sort of sense and not just be a sequence of random or nonsensical interactions. The infamous »mustache and cat«-puzzle from »Gabriel Knight 3« is a shining example of what not to do.

Streamlining

It is important that the gameplay is not too convoluted and doesn't have too many steps. This is partly to minimize the chance of the player getting stuck. When the player is stuck for longer periods they focus on the mechanics or tactics for gameplay. Also, we want to have situations where the player can plan ahead and feel like they understand the world. If the steps required for any moment are too complicated, it's very easy to lose immersion and to lose track of the goal. This happens very often in classic adventure games, where the solution to something straightforward requires a massive number of steps to accomplish.

A Sense of Accomplishment

This sort of thing is normally built into the core gameplay, but might not be as straightforward in a narrative-focused game. It is really easy to fall in the trap of doing »press the button to progress« type of gameplay when the main

goal is to tell a story. But in order to make the player feel agency, there must be some sense of achievement. The challenge needed to evoke this sense of accomplishment does not have to be skill or puzzle-based, though. Here are a few other things that could be used instead: memory tasks, out-of-the-box thinking, grind, endurance tests, difficult story choices, sequence breaks, understanding of the plot, exploration, navigation, maze escape, overcoming fear and probably tons more.

Action Confirmation

When the player does something in the game, they must understand what it is that they are doing and why they are doing it. For basic mechanics this comes naturally, »I jumped over the hole to avoid falling down«, »I shot the guy so he would not shoot me« and so forth. But when taken to the level of a scene it is not always as straightforward. For instance, the player might accidentally activate some machinery without being aware that this was going to happen beforehand and afterwards not knowing what it accomplished. If this occurs too frequently, the player starts optimizing their thinking and stops reasoning about their actions. This then leads to an experience where the player feels as if they are just pulled along.

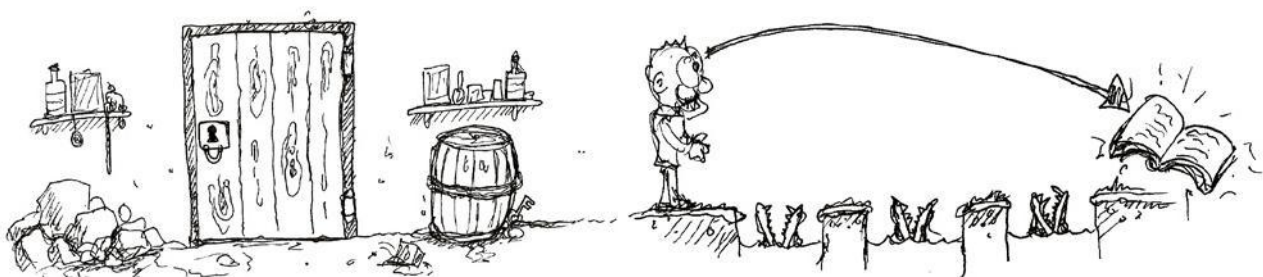
Getting all of these four rules into a gameplay scene and also making sure it is engaging is no small feat. Most games that want to focus on storytelling stop here. But in the 4 Layer Approach this is just the first step.

Before moving on to the next layer of the framework, I will give a simple gameplay example. Say the player encounters a locked door blocking their path. Some earlier information has hinted that there a key is hidden nearby, and now they need to search the room to find it. Once they find the key they can unlock the door and progress. Very simple, and not very exciting, but it fulfills rules set up above.

- 1) A locked door and hidden key should not conflict with the story.
 - 2) Given that the search space for the key is small, it is not likely the player will get stuck.
 - 3) It requires enough from the player to give a sense of accomplishment.
 - 4) Set up correctly, it should be very obvious to the player that the door needs to be opened and the key is the item used to accomplish this.
- I will come back later and expand upon this



The 4 Layers Approach will be put into use in Frictional Games' »SOMA« (upper image) and »The Vanishing of Ethan Carter« (lower image) by The Astronauts.



The puzzle to open a locked door should not conflict with the story but still requires enough from the player to give a sense of accomplishment.

Short-term narrative goals keep the player invested in the story instead of rather focusing on the gameplay action.

with the other layers to give you a better feel for how the approach works.

Layer 2: Narrative Goal

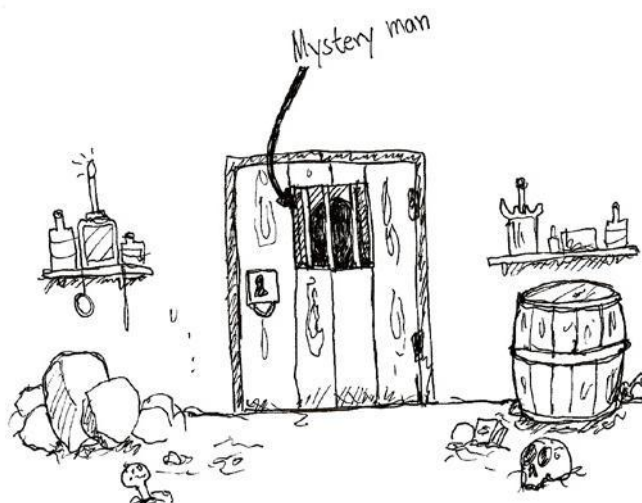
So, next step: the narrative goal. Normally the reason for the player to get through some gameplay segment is just pure progress. There is often some overarching story goal like »kill the evil wizard«, but that is quite far into the future, so when the player encounters an obstacle they try to overcome it because that is what the game demands of them. It is often very clear that they are in »gamer mode« at this point and until the obstacle is cleared. This is useful in order for the player to know what to do, but it is very problematic for the narrative – it removes the experience of being inside a story. The player stops seeing their actions as part of a story and instead sees them as steps towards an abstract gameplay goal. What can often happen is that the player starts thinking stuff like »Now I just need to get this section out of the way so I can get on with the story«, a forced mental division between narrative and gameplay, which is diametrically opposed to the fusion we're striving for.

The way to fix this is to give the player some sort of short-term narrative goal, one that is directly connected to the current gameplay. The aim is to keep the player in narrative mode so they do not brush the story aside for some puzzling or shooting action. When the player is engaged in the gameplay at hand we want them focused on and motivated by this narrative goal. This makes it harder for the player to separate the two components, as the narrative goal is always in sight. It is no longer about »doing stuff to get the story going«, instead it is about »doing stuff because of the story«. The distinction might not seem that big, but it makes all the difference. Keep in mind this is at a local level, for a scene that might just last a few minutes or less; the narrative goal is constantly visible to the player and a steady reminder of why they are going through with the actions.

A nice side-effect of this is that since the goal is narrative in nature, it becomes a reward for completing the gameplay section. The player is motivated to go through actions because of the story and is then promptly rewarded with a fresh piece of it. In all, this binds the gameplay much more tightly to the storytelling. An additional side-effect is that it can keep the player on the right track. The player might not be sure what to do next, but if the narrative goal is connected with the solution to the obstacle, then the player will progress simply by being interested in the story. Here are three different types of narrative goals that could be used.

Mystery

The most obvious and simple is mystery; that there is something unknown you want find out about. It's pretty easy to have environmental assets that constantly reminds the player of



Putting a person behind the locked door gives the player a short-term goal for wanting the door open.

this – this sort of goal is also pretty easy to fit into a gameplay scene.

Uncomfortable Environment

Another way is to give the scene a narrative reason for the player not wanting to stick around. The most trivial example of this would be a dark and scary environment; the player is scared and wants to leave. It could also be that the situation is awkward or emotional in a way that the player can't cope with and wants to escape. For example, it could be a depressing scene, like a funeral reception, that makes the player sad. It's important, though, not to get caught up in game mechanics; it must be a story reason that makes the player uncomfortable, not some mechanic (spikes popping up here and there, etc.). We want the focus to be on the narrative, not the underlying systems.

Character Conflict

Character-based conflict can also be used as a narrative goal. The »Walking Dead« adventure from Telltale is full of this; what are really just fairly simplistic activities become engaging because of story reasons. A great example is the food distribution »puzzle« in the first season where the player is instructed to determine how the remaining stash of food is divided. What makes it interesting is that the player cannot come up with a division that doesn't upset at least one of the characters. Any gameplay that results in the player changing the social dynamics can act as powerful narrative goal.

These are just three examples of what could be done and there are bound to be a ton more. I think you could use basic writing techniques to come up with more.

Now let's update the example with the locked door and add a narrative goal. To keep it simple let's go with some mystery. Say there's a man on the other side of the door trying to get in. He wants to retrieve something that's in the room that the player is currently in, and is asking them to open the door. Now all of a sudden there's a short-term goal for wanting the door open, and it's no longer just due to wanting to progress. »Who is this man?«, »What object is it that he's after?« You want to get these questions answered and that adds narrative motivation.

Important note: The 4-Layers framework is not a linear method, you'll have to constantly skip back and forth between the layers. In this case, you need to check the first layer, gameplay, and see if there's anything that could be updated in order to improve the narrative goal. You might need to change where the key is hidden, or even exchange the key for something else.

Narrative Background

With the addition of a narrative goal, the scene is now framed in a much more story-like manner. But there is still an issue: the actions the player does are quite gameplay-focused. In the locked-door-example, the player searches the environment simply in order to find a certain item; there is no proper sense of story-telling going on as the player goes through these actions. That is what this layer is all about fixing.

The basic idea is that the actions the player is supposed to be doing are immersed in story substance. So when the player is interacting, it is not just pure gameplay, they are constantly being fed story at the same time. When the narrative goal was added, the player's thinking was changed from »doing stuff to get the story going« to »doing stuff because of the story«. With narrative background in place we change it to »doing stuff in order to make the story appear«. Narrative-wise, the player's actions are no longer just a means to an end, they are what causes the story to emerge as you play. Or at least that's how we want it to appear to the player. By having the gameplay actions and the narrative beats coincide, we make it hard for the player to distinguish between the two. The goal is for this to lead to a sense of always being inside a story. Here are a few examples of the kind of background that can be used.

Story Fragments

This means having narrative clues scattered through the environment which are stumbled upon while playing. An important note is that shouldn't just be the standard audio logs and diary entries. While it can consist of those sort of elements, it's important that they never

mean a large interruption in the gameplay, and that they're found as the player goes through with the actions needed to overcome the obstacle. The act of collecting clues should not feel like a separate activity, but come as a part of the scene's main gameplay.

Complementary Dialog

There can also be dialog going on at the same time, giving context to the player's actions.

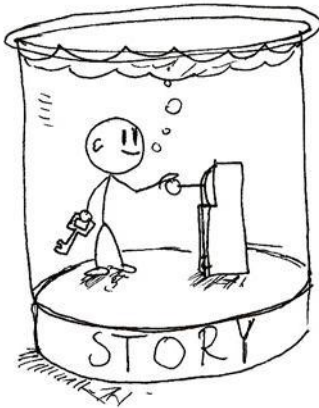
»Bastion« uses this to great effect. All of the standard gameplay elements like enemies, power-ups and breakable crates are given a place in the world and a sense of importance. It also gives a great sense of variation to similar activities, as their narrative significance can be quite diverse. Dear Esther is another good example of this at work. Here the simple act of walking is given the sense of being vital to the story.

Emotionally Significant Assets

If the items involved in the gameplay have some sort of emotional value or a strong connection to the story, the player is much less likely to see them as abstract tools. Inside of picking up the »item needed to progress«, the player finds something that can be a story revelation in itself. There is a huge difference in finding »standard knife A« and »the murder weapon from a hideous crime«.

These three are of course not the only methods at your disposal to create narrative background. Just like with the previous layer, there are bound to be tons of other things too.

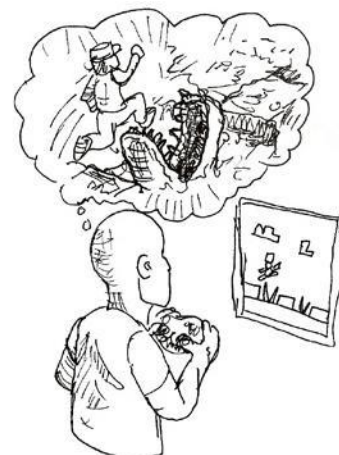
To make things a bit more concrete, let's go back to the example scene and add some narrative background. First off, let's add story fragments in the form of clues. These can give hints to the player about who the man behind the door is. Pictures, painting, documents and so on. So while the player is searching for the key they'll also be fed hints about the story. Secondly, let's have the man comment on the player's actions and give hints, making him reveal his character a bit. Third, we could say that it was the man who hid the key and that he did so for some very important reason. That way



The player's actions are not just a means to an end, they are what causes the story to emerge as you play.



Adding clues to a scene and letting the man behind the locked door comment on the player's actions gives the key narrative significance.



The player's brain build up a mental model of the game, a sort of virtual representation based upon what they see, hear and do.

the key has some narrative significance and is not just an abstract tool. Getting all of these things in might require us to change the puzzle a bit, but as said before, this not a linear design approach. What you learn from the later layers must be fed back into the previous ones.

Layer 4: Mental Modeling

Now comes the 4th, and final, layer – Mental Modeling. The goal with this layer is to change the way the player perceives and thinks about the game. We want to tap into how the player evaluates their experience.

The first and crucial fact you must be aware of is that what is actually on the screen when the player is playing is not what ends up in their head. Nor does the player rely directly on any abstract system to make choices. The player's brain build up a mental model of the game, a sort of virtual representation based upon what they see, hear and do. It's this model that's used when you choose what to do next.

This might seem a bit bizarre and counter intuitive but it really isn't. Just consider how a player doesn't rely on direct feedback from the underlying systems in order to traverse a space. They don't bump into every wall in order to check where they can go. Instead they use their knowledge of the real world, intuition of the systems, and visual and auditory clues to plan a path. And once that plan is finished (which for simple tasks like walking takes a fraction of a second), the plan is executed. Stated like this it sounds really trivial, but if you think about it a bit more, it's actually quite profound.

The underlying gameplay systems only really become evident for the player if they do something wrong or when they directly contradict their mental model. Otherwise they play and plan largely part based on an imaginary game. Obviously the underlying system is what keeps it all working, and the feedback between the systems and the player's input is crucial for anything to happen. But the systems are never directly queried to lay out the boundaries and options available to the player. In fact, keeping the player's sense of immersion is often directly related to keeping the systems hidden. The player is not a computer and doesn't make decisions based on tables of abstract data. Built-in brain functions handle all that, and the smoothest sense of play comes about when the player is relying on gut feeling and intuition. Constantly having to probe a system to figure out its exact make-up is almost never a pleasing experience. Unless that is what the game is all about, as is the case with some puzzle games.

Side note: I need to note that the player's intuition is updated the more that a system is revealed to them. If the player first assumes some enemies can jump but later finds out that they can't, their mental model is updated accordingly. This can have devastating effect on a narrative-focused game, making life-like characters turn into dumb automatons and

so on. For more information on how all that works, check this out.

Brian Upton has a great example of mental modelling in action based on his work with the original 1998 »Rainbow Six«. In »Rainbow Six« the player dies from a single shot and has to be very careful how they progress. Since they are constantly on the lookout for hostiles, even a very simplistic world can have a lot of gameplay, and that's without the player doing much. For instance, if they are about to enter a new room they stop and try to figure out the best approach. They need to consider if someone might be hiding out of sight and so forth. Based on their mental model of the game they will simulate many different approaches in their mind, trying to figure out which will work best. Even an empty hallway can conjure up these sorts of thought processes. The game is filled with possibilities that the player needs to be aware of, and the only way to do this is to use their intuition on how the game's virtual world and its inhabitants work. These constant mental gymnastics are a crucial piece of the experience.

The important point here is that most of what exists in the player's mind has no systemic counterpart. The player might imagine a guard hiding behind a corner, thinking of how he might be looking around. But in reality there is no guard behind the corner. Thus, a great deal of the playing time is spent just imagining stuff. This might seem like a cop-out, and not like proper gameplay, but that's not the case at all. It's sort of like chess, where most of the gameplay comes from you thinking about a situation, and the actual interaction only makes up minor portion of the playing time. Making mental models is very much a valid form of play.

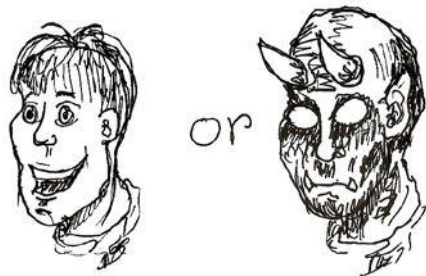
Player interpretation ties to the narrative

The takeaway here is that there is a lot of gameplay which doesn't translate into an input-output loop within the game's systems. And more importantly, this sort of mental model-based gameplay comes from the player's high level interpretation of the game's systems, graphics, sound and so forth. This means that it basically ties directly into narrative. The mental model and the narrative lie on the same level, they are the accumulation of all the lower level stuff. And if we can get them to work together, then what we have is the purest form of playable story where all your gameplay choices are made inside the narrative space. This is clearly something worth striving for.

What's also interesting is that these sort of thought processes share the imaginary nature of books and film. The player doesn't have to be 100% correct with all assumptions, just like you don't have to have a perfect mental recreation of the locale a novel takes place in. If the player imagines a non-existent guard being around the corner then that is OK. He might approach slowly trying to get signs of the guard's



In »Rainbow Six«, the player can die from a single shot and has to be very aware of how the game's virtual world and its inhabitants work.



Adding the possibility of danger to the man behind the locked door turns a simple puzzle into a narrative experience.

whereabouts and not finding a guard behind the corner doesn't need to mean the fantasy is broken. The player can now imagine that the guard soundlessly snuck away, or something similar. When interacting directly with systems, like shooting bullets at a clearly visible enemy, the player's assumptions can't stray very far from reality. If the player imagines the bullets hitting when they in fact don't, that fantasy will quickly be broken.

Quick note: In case you haven't already noticed, this layer isn't just confined to a single scene. It's something that overlaps a lot if you consider the entirety of the game. While you could potentially have mental models that only last for a short durations, it is more effective when it spans a greater part of the game.

Many narrative games already have some degree of mental modeling, but in the worst way possible: collectables. Say you have this story about a creepy forest and a protagonist trying to figure out what is real. And then picture the mental model constantly saying: »find all the thermoses, you know there are some around«. This will obviously make the game lose a lot of its potential. Be wary of this kind of issue.

Instead you want to have a mental model that fits with the rest of the narrative. What follows are a few suggestions.

Danger

There is something lurking about that constitutes a threat for the player. It's important that this threat is not some common occurrence that relies on twitch reflexes or similar, as it's just a normal gameplay element then. Instead it must be something hiding, only making brief appearances. The idea is for the player to constantly scan the environment for clues that the danger is near and present.

Goal-focused Mystery

This can mean that the player has the objective of solving a crime or similar. What we are after is that the player should see the game world as a place where important clues are to be discovered. So whenever the player finds a new location they should instantly start thinking about what new things it can teach them about the mystery.

Social Pressures

The player is amongst other people that they have to try and figure out. Now whenever the player finds something new or watches NPCs interact it updates their mental model of what makes the characters tick and what their motivations are.

The above should give an idea of what is possible, but as before, there are probably tons more to explore.

Now it's time to go back to the example scene and update it with the 4th and final layer. Let's add some sort of danger. Say the player is hunted by shape-shifting demons throughout the game and that these are also a big part of the story. This means the player won't be sure if the man behind the door is a

friend or foe. We can tie this into the layer 3 stuff as well; as the player uncovers the narrative background they receive hints about the true nature of the man behind the door as well.

We've now gone from just having a really simplistic puzzle about opening a door to an entire story experience. The player is now under threat that there might be some kind of demon on the other side and is desperately trying to find clues on what the secret man's true identity is. At the same time, the man is also the key to a mystery, a mystery the player is very curious to figure out. The player is scavenging for the key, digging up more information as he goes along and when he finally finds it he needs to decide whether to use it or not. The basic gameplay hasn't changed much, but we've changed the wrapping and it totally transforms the experience.

Endnotes

What I think is extremely interesting about this approach is that it always forces you to think about story. Normally it's so easy to just be satisfied with a well-thought-out gameplay segment and to leave it at that. But when you follow 4-Layers you need to make sure that there's some story significance to the activity the player is currently doing. Story becomes an essential part of the game design.

It can also act as a filter. You can evaluate every gameplay scene and make sure it fulfills the criteria in each of the layers. This way you can easily tell if a segment is just filler, or lacks in some other way. This is a great way to keep the design on track and make sure there is a strong narrative focus.

This method is not without its very own problems though.

First problem is that it requires a lot of planning. You need to design a lot of this up front and it's not very practical to build a scene from experimentation and iteration alone. Design documents are crucial, as there are just too many aspects to keep track of.

Second is that its core strength is also the biggest weakness. The gameplay and narrative are intertwined and if you change one the other needs to be updated too. This means that you need to throw out and remake a lot more than usual during development. But I don't see this as a failure, I see this as evidence that the approach really is bringing gameplay and narrative close together.

In a way this approach doesn't really change the core ingredients of a game. It just adds a bit of trickery on top. This is exactly what I like about it though. It doesn't rely on anything that we don't have at our disposal. And, as with all good storytelling, it relies on the audience's imagination doing the bulk of the work. I am really excited to see how this approach will turn out in the finished games. So far it's been of great use to us, and hopefully someone else will be inspired to give it a go. **Thomas Grip**

Acknowledgements

Adrian Chmielarz
for all the great e-mail discussions that led to all this and feedback on the talk.

Ian Thomas
for copy-editing this whole thing.

Brian Upton
for letting me read an early copy of his book and providing the basis for the Mental Model section.

Matthew Weise
for providing valuable feedback to the lecture.



IN DIE KARTEN GESCHAUT

»Tile-based Maps« etablierten sich zu einer Zeit, als es an Arbeitsspeicher noch mangelte. Doch bis heute ist die Technik bei der Generierung von Karten gängige Praxis. Nurdogan Erdem erklärt, wie das Ganze im Detail funktioniert und wo sich die Technik anwenden lässt.

Eine Karte aus sich wiederholenden Kacheln zu generieren, ist eine nach wie vor elementar eingesetzte Vorgehensweise in der Entwicklung von Computerspielen. Die auch als »Tile-based Map« bekannte Technik etablierte sich zu einer Zeit, als der Arbeitsspeicher rar war, und ist auch heute noch populär. Ob nun Retrospiele wie »Super Mario« oder Anwendungen wie »Google Maps«, die Funktionsweise bleibt gleich. Bei einer Tile-based Map werden aus einzelnen grafischen Kacheln größere Karten gebaut, die in Spielen beispielsweise die Level-Grafiken definieren, auf denen Super Mario herumtänzeln kann.

Der einfachste Kartentyp ist die orthogonale Karte, in der man die Szene von oben oder von der Seite betrachtet. Etwas aufwendiger sind isometrische und hexagonale Karten. Dabei gilt grundsätzlich, dass sich Karten, die nur einen Bildschirm groß sind, einfacher realisieren lassen als Karten, die größer sind und daher bewegt werden müssen. Denn hier kommt das sogenannte »Scrolling« ins Spiel – das Bewegen der Karte in eine oder mehrere Richtungen.

Was man falsch machen kann

Um eine möglichst stabile Bildrate zu gewährleisten, sollten spezielle Routinen zum Zeichnen und Bewegen der Karte vorhanden sein. Denn was mit kleinen Karten noch gut funktioniert, muss es mit größeren nicht auch tun. In vielen Programmiersprachen wie Java

oder Actionscript werden Container zum Gruppieren und Manipulieren von Grafiken verwendet. Angenommen, wir möchten eine sehr große Karte mit insgesamt 100 x 100 Kacheln bauen. Jede Kachel soll eine Breite und Höhe von 128 Pixeln betragen. Wir nehmen nun besagten Container und fügen alle Kacheln als Kind-Objekte hinzu. Anschließend könnten wir einfach die x- und y-Koordinaten des Containers verwenden, um die Karte in unserem Bildschirmausschnitt zu bewegen. Fertig wäre unsere Tile-based Map. So weit die Theorie.

Bei einer kleinen Karte wäre dieser Ansatz, unter gewissen Voraussetzungen, noch in Ordnung. Bei einer Karte mit beispielsweise 100 x 100 Kacheln wäre das jedoch keine sehr effiziente Lösung. Das Problem wäre hier, dass bei jeder Änderung der Koordinaten des Containers dieser all seine Kind-Objekte durchlaufen müsste. Je kleiner die Kacheln und je größer die Karte, desto mehr Arbeit für die CPU – und desto schlechter die Performanz. Eine alternative Möglichkeit wäre es, die Kacheln zu »baken«, also in eine Bitmap zu zeichnen, und dann nur noch diese zu bewegen. Die Kacheln wären somit keine einzelnen Kind-Objekte mehr, die durchlaufen werden müssten. Aber auch diese Methode hätte einen Haken: Die Dimensionen für eine Bitmap sind begrenzt. Oft betragen sie 8192 x 8192 Pixel, womit auch diese Herangehensweise an den Voraussetzungen einer größeren Karte scheitert. Allerdings kann der Ansatz mit der Bitmap ausgebaut werden, wie ich etwas später zeigen werde.



Nurdogan Erdem
ist Senior Software Developer
bei InnoGames.

Nurdogan lebt in Hamburg und arbeitet als Senior Software Developer bei InnoGames. Vorher war er bei Bigpoint als Tech Lead tätig. In seiner Freizeit arbeitet er an einem eigenen RPG im Retrostil.

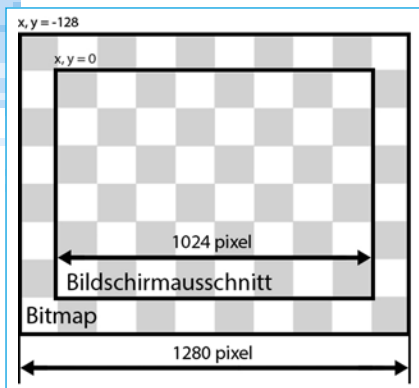


Abbildung 1 Bei einer Kachelgröße von 128 Pixeln und einem Bildschirmausschnitt von 1024 x 768 Pixeln müssen die x- und y-Koordinaten auf -128 gesetzt werden, um die Bitmap zu zentrieren.

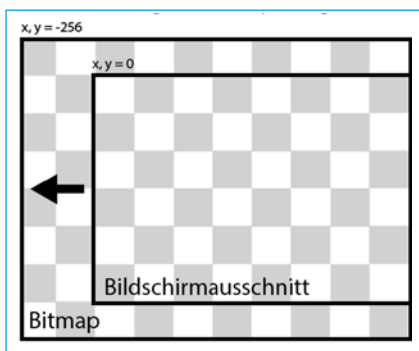


Abbildung 2 Soll die Karte nach links bewegt werden, wird die x-Koordinate der Bitmap dekrementiert, bis ihr Wert -256 beträgt und somit um eine Kachel verschoben wurde.

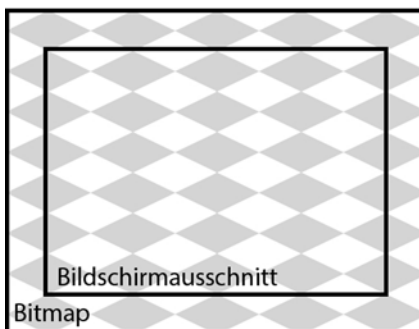


Abbildung 3 Bei einer isometrischen Karte wie beispielsweise in »Forge of Empires« (unten) sind die Kacheln nicht rechteckig, sondern diamantförmig (oben), weil die Karte um 45 Grad gekippt ist.

Wie es die Großen machen

Spiele wie »Grand Theft Auto« oder »Assassin's Creed« bieten mittlerweile ein großartiges Open-World-Erlebnis. Macht man sich jedoch bewusst, dass die Welt, die man sieht, eigentlich nur unmittelbar berechnet und gezeichnet wird, dann wirkt GTA auf einmal nicht mehr so groß und faszinierend. Zu spät eingeblendete Grafiken wie Bäume oder Häuserfassaden zeugen von einer noch nicht ganz ausgereiften Technik, die jedoch ihren zweckmäßigen Dienst verrichtet. Würde das Spiel jederzeit alle Figuren und den gesamten Verkehr berechnen, wäre es nahe an der Unspielbarkeit. Aber nicht nur Rechner sind auf optimierte Routinen zur Reduzierung der Datenflut angewiesen, auch unser Gehirn ist ein wahrer Meister im Filtern von unnützen Informationen. Oder hören Sie in diesem Moment, wie Sie durch die Nase atmen? Jetzt wahrscheinlich schon.

Bewusst zeichnen

Wir werden uns nun eine Methode ansehen, welche die Ressourcen des Rechners schont und es gleichzeitig erlaubt, auch mit sehr großen Karten zu arbeiten. Vereinfacht ausgedrückt: Wir werden die Kacheln, beziehungsweise einen bestimmten Bereich der Karte, in eine Bitmap zeichnen und diese anschließend bewegen. Weil die Bitmap nur unwesentlich größer sein wird als unser Bildschirmausschnitt, wird die Bewegung immer nur maximal eine Kachelgröße betragen. Zum Schluss werden wir die Bitmap in ihre Ausgangsposition zurücksetzen und den Vorgang wiederholen. Durch fortlaufende Wiederholungen werden wir so den Eindruck eines nahtlosen Scrollings bekommen.

Wir geben unserer Bitmap also zunächst die Größe des Bildschirmausschnittes und addieren zusätzliche 128 Pixel an allen Rändern. Diese zusätzlichen Pixel werden benötigt, um das Bewegen der Bitmap um eine Kachelgröße zu gewährleisten. Bei einem Bildschirmausschnitt von 1024 x 768 mit einer Kachelgröße

von 128 Pixeln hat unsere Bitmap daher nun die Größe von 1280 x 1024 Pixeln. Weil wir die Bitmap in alle Richtungen bewegen möchten, müssen wir sie zentrieren. Dazu setzen wir die Koordinaten x und y der Bitmap auf -128 (Abbildung 1).

Nun muss unsere Routine die Kacheln in die Bitmap kopieren. Natürlich nicht alle 100 x 100 Kacheln der Karte, bei einer Größe von 1280 x 1024 Pixeln für die Bitmap müssen wir nur die Kacheln im Bereich 10 x 8 der Karte berücksichtigen. In der Regel würde man hier einen zweidimensionalen Array nehmen, dessen Elemente die Kachel-ID vorhalten. Das Kopieren der Kacheln sollte mit schnellen Funktionen der jeweiligen Programmiersprache erfolgen. In Actionscript heißt diese Funktion beispielsweise »copyPixels«.

Ist das Kopieren beendet, können wir uns Schritt zwei zuwenden: dem Scrollen der Karte. Weil die Karte links und oben bereits aufhört, werden wir sie zunächst einmal nach links bewegen. Dazu dekrementieren wir die x-Koordinate der Bitmap, bis sich diese um eine Kachelgröße nach links bewegt hat. Die x-Koordinate hat nun den Wert -256 (Abbildung 2).

Weiter nach links scrollen können wir nicht, weil auf der rechten Seite die Bitmap zu Ende ist. Was machen wir? Wir setzen die Bitmap nun auf ihre Ausgangsposition zurück, also auf x = -128. Anschließend wiederholen wir den ersten Vorgang und zeichnen die Bitmap komplett neu. Auch diesmal durchlaufen wir nur einen Bereich von 10 x 8 Elementen in unserem Array. Allerdings erhöhen wir den x-Index, mit dem wir auf die Elemente des Arrays zugreifen, um den Wert 1.

Dadurch wird die Karte um eine Kachelgröße nach links versetzt in die Bitmap gezeichnet. So entsteht, wie bereits eingangs erwähnt, der Eindruck einer sich bewegenden Karte. In Wahrheit jedoch ist es lediglich die Bitmap, die sich immer nur um maximal eine Kachelgröße bewegt und anschließend wieder zurückgesetzt wird.

Weitere Optimierungsmöglichkeiten

Sicherlich kann man diese Routine noch weiter optimieren. So könnte man beispielsweise bei jedem Zeichenvorgang nur die Kacheln am Rand neu zeichnen. Oder den bereits vorhandenen Bereich in der Bitmap durch spezielle Funktionen der verwendeten Programmiersprache verschieben. Dadurch würde die Anzahl der Kopiervorgänge schrumpfen – das könnte sich ganz besonders bei kleinen Kachelgrößen auszahlen.

Hardwarebeschleunigung

Hat man zusätzlich Zugriff auf die Hardware, kann man die Performanz noch weiter in die Höhe treiben, indem man die Kachel-Grafiken in einem sogenannten »Texture Atlas« zusammenfasst. Dabei handelt es sich um eine einzelne große Grafik, die mehrere kleinere



Zahlreiche Spiele verwenden unterschiedliche Kartentypen, hier im Beispiel »Forge of Empires«: Eine orthogonale Karte wird zur Darstellung der Kontinente verwendet (oben), während das eigentliche Spielgeschehen in einer isometrischen Karte stattfindet (unten).

Grafiken zusammenfasst, die hier Texturen genannt werden. Weil auf der Grafikkarte immer nur eine Textur aktiv sein kann und ein Wechsel auf eine andere Textur seine Zeit braucht, bringt ein Texture Atlas hier enorme Vorteile in Sachen Performanz. Allerdings muss man in diesem Fall keine Kacheln mehr in eine Bitmap kopieren, sondern in der Regel die Texturen in eine sogenannte »Batch« hinzufügen. In Actionscript beziehungsweise Starling findet man eine solche Klasse unter dem Namen »Quad Batch«, in anderen Sprachen auch unter »Sprite Batch«. Die Funktionsweise unserer Routine jedoch würde weitgehend erhalten bleiben, außer dass man nun keine Kacheln mehr kopiert, sondern diese einfach nur noch der Batch hinzufügt.



Isometrie

Wir haben bisher nur eine Karte behandelt, auf die man von oben herab schaut. Wie sieht es jedoch mit isometrischen Karten aus, beispielsweise im Browsergame »Forge of Empires« oder im Klassiker »UFO: Enemy Unknown«? Funktioniert die Routine zum Zeichnen der Karte hier noch? Weil unsere Routine von einer rechteckigen Landschaft ausgeht, die isometrische Karte jedoch in einem Winkel von 45 Grad gekippt ist, müssen wir uns eine Erweiterung ausdenken. Zunächst müssen wir berücksichtigen, dass unsere Kacheln nicht mehr rechteckig sind, sondern diamantförmig (**Abbildung 3**).

Zunächst nehmen wir uns den Array vor, der die Kachel-IDs vorhält. Wir kopieren die Elemente unseres Arrays um 45 Grad gedreht in einen neuen und entsprechend größeren Array. Nun können wir durch diesen neuen, projizierten Array iterieren und uns die notwendigen Kachel-IDs holen. Weil wir es mit diamantförmigen Kacheln zu tun haben, zeichnen wir die Kacheln einfach um eine halbe Kachelgröße versetzt. Und zwar sowohl in horizontaler als auch vertikaler Richtung. Dadurch sparen wir uns aufwendige mathematische Berechnungen zum Positionieren der Kacheln. Dies ist zwar eine recht triviale, aber auch effektive Lösung, die nur mit diamantförmigen Kacheln funktioniert.

Hexagone

Hex-Feld-Schlachten sind spätestens seit »Battle Worlds: Kronos« wieder in Mode. Können wir die bisherige Logik zum Zeichnen der Kacheln auch auf Hex-Feldern anwenden? Prinzipiell ja. Man kann auch hier auf die üblichen mathematischen Berechnungen für das Positionieren von Hex-Feldern verzichten. Der Trick ist, einfach zwei Bitmaps zu benutzen, in die man die Felder abwechselnd zeichnet. Alle geraden Spalten kommen in die erste und alle ungeraden Spalten in die zweite Bitmap. Die Ausgangsposition der zweiten Bitmap wird jedoch leicht verschoben, sodass beide Bitmaps übereinandergelegt wie Puzzlestücke ineinandergreifen. Die Implementierung muss im Vergleich zur normalen Karte nur unwesentlich erweitert werden. Anschließend hat man aber auch hier eine effiziente Tile-based Map.

Ein paar Tipps zum Schluss

Falls beim Scrollen der Karte immer noch Ruckler auftreten, könnte eine niedrige Bildwiederholrate dafür verantwortlich sein. Scrollt sich die Karte sehr schnell, sollte die Bildwiederholrate am besten 50 oder 60 Bilder pro Sekunde betragen. Bei langsamem Scrollen sind auch 30 Bilder pro Sekunde noch in Ordnung. Wer sich das mühsame Editieren der Maps per Hand ersparen möchte, kann außerdem zu einem Tile-Editor greifen. **Nurdoğan Erdem**

DIE NEUE ENGINE-GENERATION TEIL 2

Eine neue Konsolengeneration und immer leistungsfähigere PCs stellen stetig steigende Anforderungen an Spiele – und damit auch an die Engines. Carl Jones von Crytek und Andrew Bowell von Havok erläutern die Zukunftspläne von CryEngine, Vision und Project Anarchy.



Carl Jones
ist Director of Global Business Development bei Crytek.

Carl blickt auf über 15 Jahre Erfahrung in der Spieleindustrie zurück und arbeitet seit 2008 für Crytek, wobei sein Fokus zunächst auf der Lizenzierung mit der CryEngine 3 lag. Später übernahm er die Abteilung Business Development komplett. Seither überwacht Carl alle Bereiche, in denen Crytek tätig ist, vom sozialen Netzwerk GFACE über interne Spieleprojekte bis hin zu den Lizenznehmern der CryEngine. Carl begann seine Karriere beim britischen Militär, wo er an Simulationen für die Einsatzkräfte arbeitete. Danach wechselte er in die Spieleindustrie, wo er unter anderem für die Studios Wide Games (wo er zum Gründungsteam gehörte), Tragnarion Studios, Kujira Entertainment sowie Sega und The Creative Assembly arbeitete. In seinem Portfolio finden sich deshalb Marken wie die Total-War-Reihe, Total Warrior, House of the Dead, Sonic the Hedgehog, Prisoner of War und The Matrix.

Making Games Ihr bietet zwei verschiedene Abo-Modelle an. Eines für rund 10 Dollar pro Monat und eine sogenannte »Full License«. Wie genau zieht ihr die Grenze zwischen diesen beiden Gruppen?

Carl Jones Das ist etwas, das nicht wir festlegen. Die »Engine als Service«, wie wir das monatliche Modell nennen, ist zunächst einmal für jedermann verfügbar. Von Einzelpersonen, Hobbyentwicklern, Indie-Entwicklern über kleinere Studios bis hin zu den ganz Großen – wirklich jeder kann unsere Engine so nutzen.

Die Einschränkungen bei der Abo-Variante liegen zum Beispiel auf einigen der Low-Level-Rendering-Systeme, zu denen wir keinen Source Code rausgeben. Zudem können wir die Kunden mit monatlichem Abo natürlich nicht so umfassend unterstützen wie die Full-License-Kunden.

Die Full-License-Kunden sind diejenigen, die wirklich auf jede Zeile Programmcode Zugriff

haben möchten, die in Kontakt mit unseren Entwicklern stehen und die alle Funktionen und Tools der CryEngine nutzen wollen.

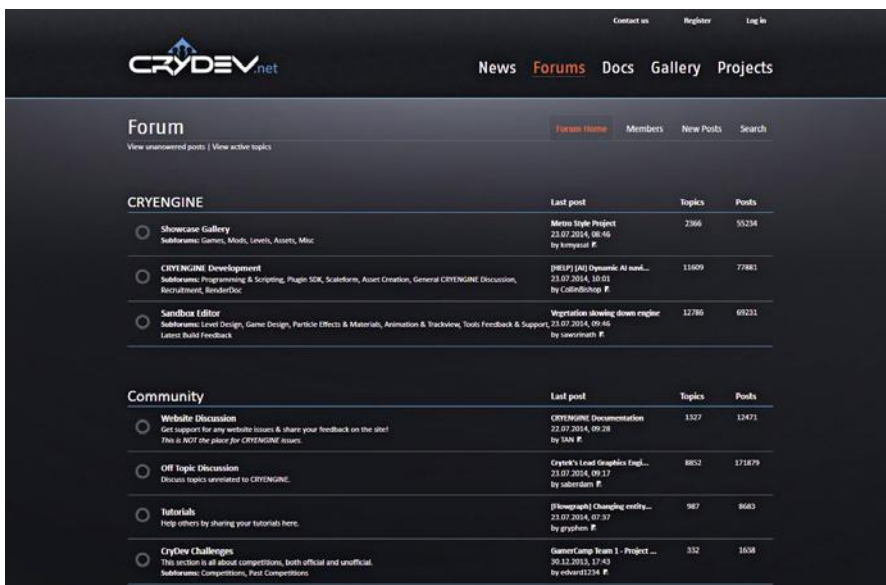
Making Games Welche Art von Support bietet ihr euren Kunden, gerade wenn jemand noch nie mit der CryEngine gearbeitet hat?

Carl Jones Hilfestellungen bieten wir allen an, also egal ob nur Monats-Abo oder Full License. Wir sind allerdings noch nicht mit dem Umfang zufrieden. Doch wir arbeiten daran, hier nachzubessern. Wir haben ausführliche Anleitungen zu den verschiedenen Tools der Engine, aber auch zu Themen wie Asset-Erstellung. Den Bereich Quellcode haben wir zwar ausreichend dokumentiert, allerdings können wir gerade hier meiner Meinung nach noch besser werden.

Vorlagen haben wir zu allen möglichen Arten von Spielen oder Spielsituationen im Angebot. Wir versuchen, auch dieses Angebot so schnell wie möglich auszubauen. Abseits dessen haben wir jedoch eine starke Community an Entwicklern – inklusive unserer eigenen –, die wir immer ermutigen, sich aktiv zu beteiligen. Den monatlichen Abonnenten können wir aber keine direkten Ansprechpartner garantieren, denn das würde eine riesige Zahl von Mitarbeitern voraussetzen.

Making Games Wie schnell seid ihr in der Lage, auf Anforderungen zu reagieren, die ein Entwickler haben könnte?

Carl Jones In solchen Fällen müssen wir den Kompromiss zwischen Geschwindigkeit und Qualität eingehen. Wenn wir die Engine zum Beispiel komplett öffnen würden und keiner würde die Qualität von Quellcode-Änderungen überprüfen, dann bestünde die Gefahr, dass irgendwas in der Engine völlig schief läuft. Und gerade bei so vielen Leuten, die mit unserer Engine arbeiten, wäre das ein enormes Risiko. Im schlimmsten Fall wird irgendetwas in der Engine verändert, was ein bestehendes Projekt zerstört.



Crytek setzt auf eine starke Entwickler-Community – inklusive der eigenen –, die sich aktiv beim Support beteiligen und in den Foren Hilfestellung leisten soll. Nur Full-License-Nehmer können auf direkte Ansprechpartner hoffen, sonst wären zu viele Mitarbeiter nötig.



Crytek konnte in den vergangenen Monaten viele Kunden gewinnen – aus ihrer Sicht gibt es keine Engine, die so viele hochkarätige Lizenznehmer hat.

Deshalb sammeln wir alle Anforderungen, die wir von unseren Lizenznehmern, aber auch Abonnenten erhalten. Danach schauen wir, welche Dinge am häufigsten gefordert werden. Wenn also 90 Prozent der Leute sagen, sie möchten diese oder jene Funktion implementiert haben oder dies oder jenes sollte so und so funktionieren, dann wandern solche Dinge mit auf die Liste, die wir abarbeiten.

Making Games Wie handhabt ihr das mit den Gebühren und Umsatzbeteiligungen? Epic zum Beispiel verlangt ja rund 20 Dollar und 5 Prozent Umsatzbeteiligung.

Carl Jones Wir setzen beide auf ein Abo-Modell. Epic verlangt 19 Dollar und 5 Prozent Umsatzbeteiligung. Wir dagegen stellen 9,90 Dollar in Rechnung, ohne Umsatzbeteiligung.

Wenn wir aber über Full Licenses reden, da hat auch Epic nicht bekannt gegeben, wie sie das Finanzielle regeln – soweit ich mich erinnere. Wir verraten das natürlich auch nicht. Außerdem ist das immer eine Frage, wie wir

dies mit einzelnen Entwicklern oder Publishern aushandeln. Die Preise hängen davon ab, welche Anforderungen der einzelne Kunde benötigt hat, und dann versuchen wir einen Vertrag aufzusetzen, der für alle Parteien akzeptabel ist.

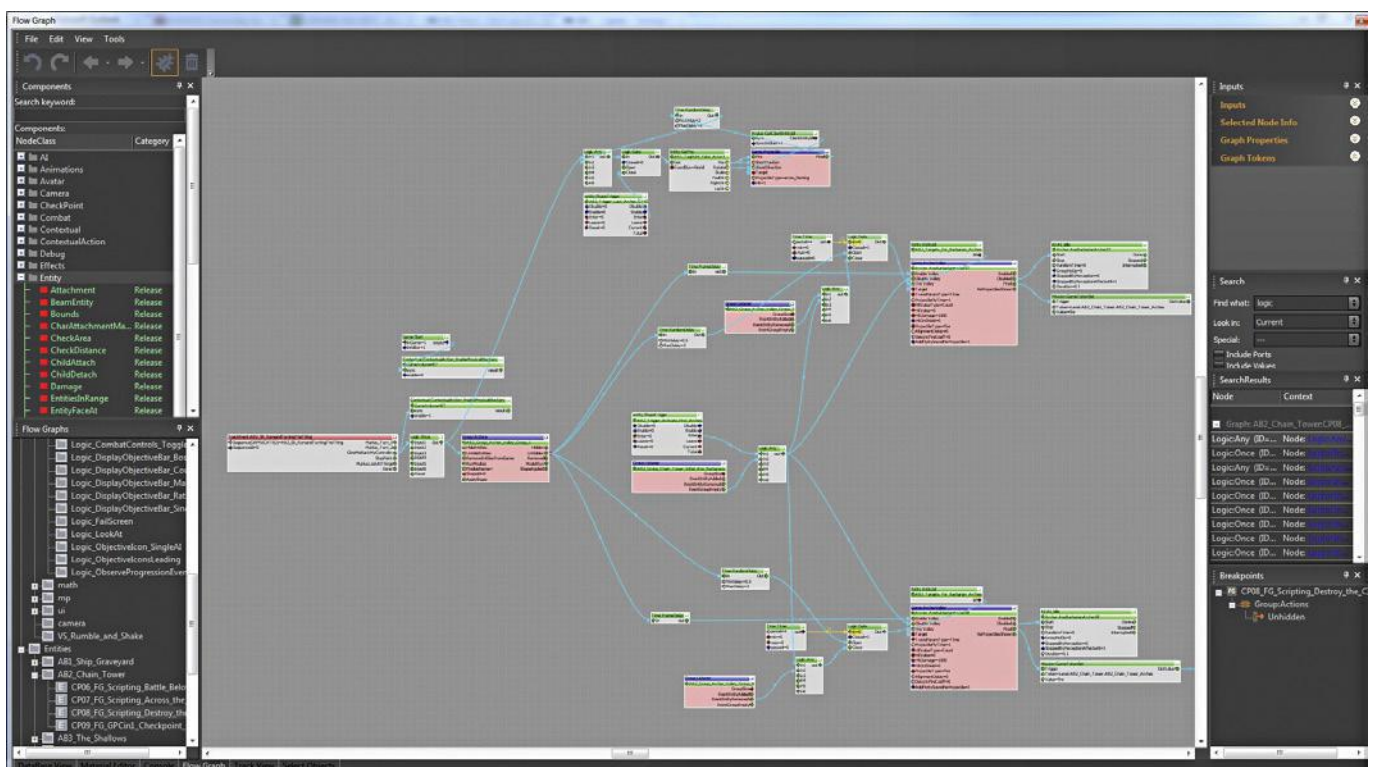
Making Games Mit der Umstellung eures Geschäftsmodells ging sicherlich die Hoffnung einher, dass sich mehr Entwickler dafür entscheiden, die CryEngine zu nutzen. Inwieweit geht dieser Plan bisher auf?

Carl Jones Natürlich verfolgen wir hochgesteckte Ziele, so wie es die anderen Engine-Entwickler vermutlich auch tun. Wir denken schon länger über Wege nach, ein größeres Publikum zu erreichen, eigentlich schon, seit Unity damit angefangen hat, eine offenere Version ihrer Engine zu veröffentlichen. Es ist ganz klar unser Ziel, die Verbreitung unserer Engine zu erhöhen, aber gleichzeitig auch, eine Engine zu liefern, mit der wirklich jeder arbeiten und eigene Spiele umsetzen kann. So ein Ziel lässt sich jedoch nicht erreichen, ohne mit

Die Serie im Überblick

- Teil 1
Unreal Engine 4 und Unity 5
- Teil 2
CryEngine und Vision

»Die ›Engine als Service‹ ist für jedermann verfügbar.«



Crytek will seinen Kunden leistungsstarke Werkzeuge an die Hand geben – Flow Graph ist ein Beispiel. Ohne Kenntnisse im Bereich Scripting oder Programmierung lassen sich hier komplexe Scripting-Logiken erstellen.

»Auf den
Next-Gen-Konsolen
sind wir in der
besseren Position.«

der bestehenden Community zusammenarbeiten. Das ist genau das, was wir mit unserem Service-Gedanken zu erreichen versuchen.

Making Games Was könnten Gründe dafür sein, dass die Unreal Engine 3 so viel beliebter war als die CryEngine?

Carl Jones Ich denke, hier spielt der Zeitpunkt der Veröffentlichung die größte Rolle. Bis wir unsere Engine auf die letzte Konsolengeneration angepasst hatten, waren uns die anderen Entwickler deutlich voraus. Offensichtlich hielten es viele deshalb für ein Risiko, mitten in einem Konsolenzyklus auf eine neue Engine umzusteigen, vor allem auf eine Engine, die ihre Qualitäten noch nicht bewiesen hatte. Weil während der letzten Konsolengeneration einige Launch-Titel bereits die Unreal Engine nutzten, war die Entscheidung der Entwickler, Epic als Partner zu wählen, natürlich einfacher. Bei den Next-Gen-Konsolen dagegen denke ich, dass sich das Blatt gewendet hat und wir in der besseren Position sind, weil wir mit Ryse einen der am besten aussehenden Launch-Titel für die Xbox One veröffentlicht haben. Wir mussten unsere Engine für die Next-Gen-Konsolen nicht großartig modifizieren. Als wir die CryEngine entwickelt haben, hatten wir schon vor rund fünf Jahren die Next-Gen-Konsolen im Blick und haben die Engine entsprechend aufgebaut, sodass zum Start der Konsolen keine massiven Änderungen mehr vorgenommen werden mussten.

Making Games Weil du gerade von Next-Gen-Konsolen gesprochen hast – welche Vorteile bringen die Konsolen eurer Meinung nach?

Carl Jones Ganz einfach ausgedrückt, sind sie wesentlich leistungsfähiger. Es steht eine große Menge an Arbeitsspeicher zur Verfügung, die noch nicht mal in jedem PC als Standard angesehen werden kann. Auch die Prozesso-

ren sind leistungsstark und übertrumpfen so manchen PC. Natürlich kann man leicht PCs zusammenbauen, die mehr Power haben. Aber ich glaube, im Moment sind wir tatsächlich an dem Punkt angelangt, an dem Konsolen mit PCs gleichziehen. Dieser Gleichstand wird natürlich nicht von langer Dauer sein, einfach weil sich PC-Hardware schnell weiterentwickelt.

Making Games Abseits des Fakts, dass die neuen Konsolen über mehr Leistung verfügen, ermöglichen sie denn bestimmte Funktionen, die zuvor nicht möglich waren?

Carl Jones Vor allem ermöglichen sie uns, noch näher an »Close to Metal«-Programmierung heranzukommen und nach Lösungen zu suchen, die etwa GPGPU nutzen. Sie erlauben es uns, viel mehr Dinge synchron zu simulieren, die mit der vorherigen System-Architektur nicht möglich gewesen wären. Letztendlich geht es uns dabei um Tools, sodass wir den Entwicklern nicht sagen müssen, wir haben jetzt Normal Mapping, Tessellation oder so. Vielmehr wollen wir ihnen leistungsstarke Werkzeuge an die Hand geben, mit denen sie herumspielen, etwas Eigenes erschaffen und Ideen verwirklichen können.

Making Games Ihr habt auf der GDC in San Francisco die Zusammenarbeit mit AMD angekündigt. Welche Vorteile erhofft ihr euch von deren Technologie?

Carl Jones Mantle dreht sich komplett darum, »Close to Metal«-Programmierung zu erreichen. Metal von Apple versucht das ebenso, und auch DirectX 12 geht in diese Richtung. Das ist großartig, denn je transparenter und flexibler die Hersteller unseren Entwicklern Zugriff bis in die tiefsten Ebenen der Hardware erlauben, desto mehr können wir aus ihr herausholen. Generell ermöglicht uns das, die Performance erheblich zu verbessern und neue Grafiklösungen zu entwickeln – es lässt sich am besten damit vergleichen, von einem Kinderspielzeug auf echtes Profiwerkzeug umzusteigen.

So gesehen ist es natürlich klar, dass jeder, der Mantle benutzt, momentan einen Vorteil gegenüber DirectX-Nutzern hat. Das ist schon per Definition so, weil Mantle Funktionen mitbringt, die DirectX zurzeit noch nicht bietet. Doch auch DirectX 12 steht schon vor der Tür und gibt uns hoffentlich genauso viel Freiheiten und Möglichkeiten der Kontrolle, und deshalb bin ich überzeugt, dass gerade PC-Spieler, aber hoffentlich auch andere Plattformen in den nächsten Jahren immer bessere Grafikqualität genießen werden können.

Making Games Abschließend, gibt es irgendwelche Spiele oder Genres, die du gern mit der CryEngine umgesetzt sehen würdest?

Carl Jones Puh, das ist eine schwere Frage. Um mich aus der Affäre zu ziehen – ich denke, alle Spiele, die jemals veröffentlicht wurden, würden mit der CryEngine besser aussehen. (lacht)



Steam als Vertriebsplattform für die CryEngine soll als Anreiz für Entwickler dienen, weil ein funktionierendes Ökosystem im Hintergrund arbeitet.

Interview: Sebastian Weber

Making Games Wer Vision nutzen möchte, muss eine Lizenz kaufen. Auf eurer Website gibt es allerdings keine Informationen zu den Kosten der Lizenz. Habt ihr unterschiedliche Modelle für AAA-Studios und Indie-Entwickler?

Andrew Bowell Ja, wir haben unterschiedliche Modelle. Wir haben das Programm »Havok Strike«, das ausschließlich für Indie-Entwickler gedacht ist. Unsere Middleware-Pakete sind so aufgebaut, dass sich die Gebühren an deinem Entwicklungsbudget orientieren. Das ist für Indies super, weil sie mit einem kleinen Budget arbeiten. Für sie sind die Gebühren also entsprechend niedrig. Zu den konkreten Kosten und Prozentsätzen kann ich leider nichts sagen.

AAA-Studios haben offensichtlich teilweise sehr große Budgets. Wir lassen die Gebühren für unsere Lizenz aber nicht bis ins Unendliche mitwachsen; für diese Studios haben wir also einen Festbetrag. In dieser Hinsicht sind wir auf jeden Fall konkurrenzfähig.

Making Games Für kleine Studios gibt es also keine Vorgaben wie Umsatzbeteiligung, wie zum Beispiel bei der Unreal Engine?

Andrew Bowell Das gibt es bei uns nicht. Wer nur für Mobile entwickelt, kann auf Project Anarchy ausweichen. Das ist komplett kostenlos, es gibt keine Nutzungsgebühren oder Vorabkosten.

Die Nutzer bevorzugen normalerweise Modelle, bei denen sie sich nicht auf laufende

Gebühren festlegen müssen. Das Modell von Epic ist für viele sicher auch eine gute Variante, aber wir haben die Erfahrung gemacht, dass die Leute im Zweifelsfall lieber vorab bezahlen und sich das Thema damit erledigt hat. Bei uns ist es momentan so, dass du erst die normale Lizenz bezahlen musst, wenn du zum Beispiel ein MMO für den PC entwickeln willst.

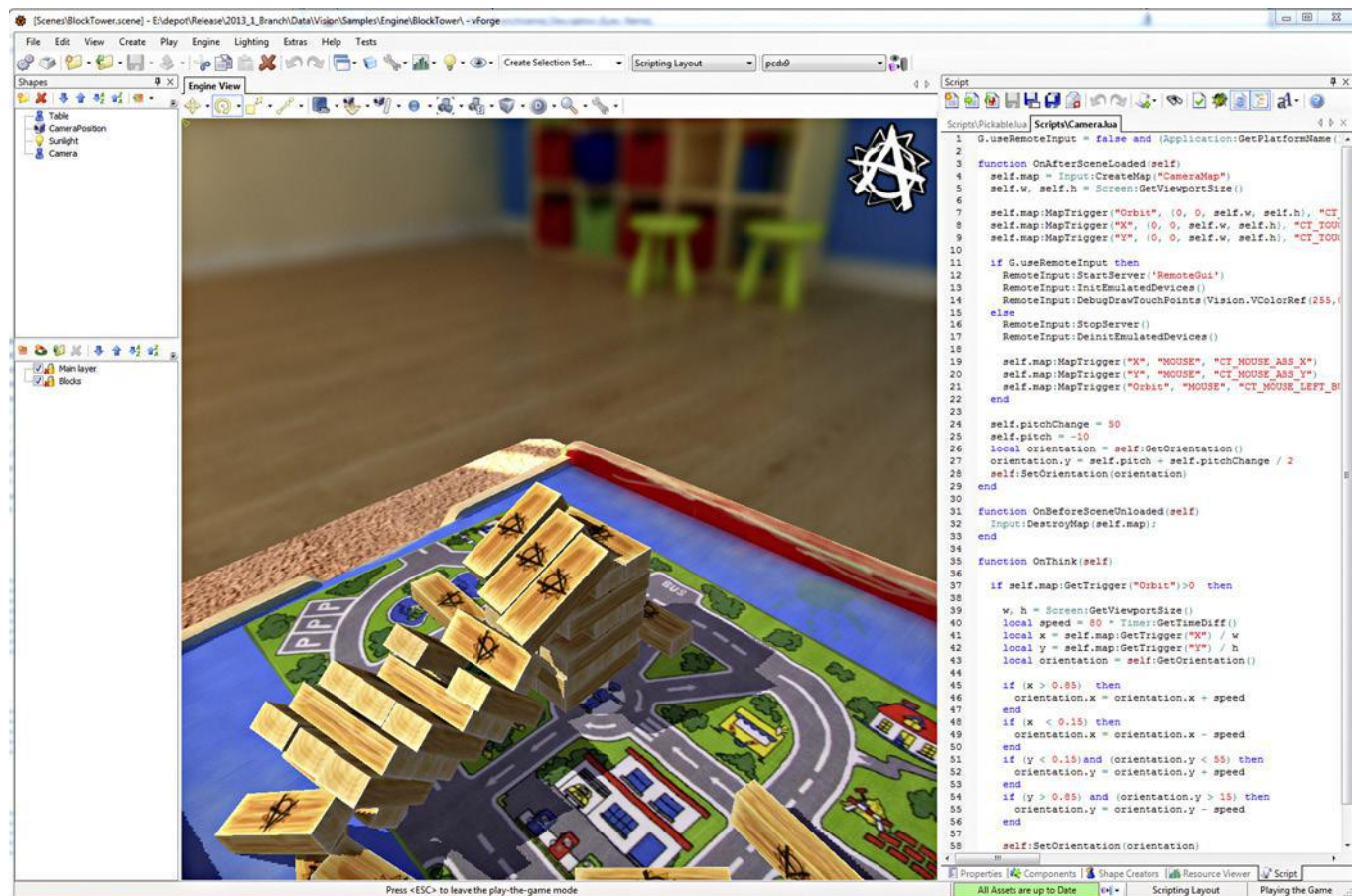
Making Games Wie machen sich die unterschiedlichen Modelle beim Support bemerkbar?

Andrew Bowell Havok ist ja für seinen Support bekannt. Das ist also natürlich ein Punkt, den wir bei den Geschäftsmodellen für kleinere Studios keinesfalls gefährden wollten. Wir haben somit lediglich die Anzahl der parallel laufenden Support-Anfragen nach unten korrigiert. Selbstverständlich wissen wir, dass es in Crunch-Zeiten zu vermehrten Anfragen kommen kann, daher räumen wir den Studios eine bestimmte Anzahl von Zeitfenstern ein, in denen sie den vollen Support von uns bekommen; aber beispielsweise in der Mitte des Produktionszyklus schrauben wir die Anfragen-Zahl herunter. Wer voll für die Engine bezahlt hat, bekommt natürlich auch den vollen Support. Wir schicken sogar unsere Mitarbeiter raus in die Studios. Wenn jemand deine Engine kauft, dann hängen bis zu 70 Prozent des Spiels von deinem Produkt ab. Daher tun wir alles, was wir können, um den Entwicklern zu helfen.



Andrew Bowell
ist Head of Product Management bei Havok.

Andrew hat über 14 Jahre Erfahrung in der Entwicklung neuer Technologien. Bei Havok ist er als Head of Product Management für alle Produktlinien verantwortlich - von der Planung bis zur Ausführung. Andrew arbeitet eng mit Engineering, Sales, Marketing und Support zusammen, um sowohl Umsatz als auch Kundenzufriedenheit sicherzustellen. Seit 2000 hatte er bei Havok schon unterschiedliche Positionen inne, unter anderem Regional Sales Manager für Europa, Pre-Sales Engineer und Product Engineer. Vorher war Andrew IT Consultant und vor allem in London und New York tätig. Er hat einen Abschluss in Informatik am Trinity College in Dublin gemacht.



Project Anarchy ist für die Mobile-Entwicklung ausgelegt: »Diese Version unserer Engine ist komplett kostenlos.«



Das Spacetrading-Spiel »Cosmonautica« des deutschen Studios Chasing Carrots hat bei der Project Anarchy Challenge von Havok das Preisgeld von 100.000 Dollar gewonnen.

»Wenn jemand deine Engine kauft, hängen bis zu 70 Prozent des Spiels von deinem Produkt ab.«

Making Games Habt ihr für eure Produkte ein Abo-Modell in Betracht gezogen?

Andrew Powell Anarchy ist kostenlos, daher kommt ein Abo-Modell dafür nicht in Frage. Wir haben eine Möglichkeit für Entwickler, die ihr Spiel für den PC portieren und auf Steam veröffentlichen möchten. Dieses Upgrade für Anarchy werden wir noch 2014 veröffentlichen. Zurzeit sieht es so aus, als würden wir dafür eine Pauschalgebühr verlangen, doch wir denken auch über andere Optionen nach. Du kannst unsere Engine kostenlos nutzen, wenn du ein Mobile-Spiel entwickelst – und wenn du es auch auf dem PC veröffentlichen willst, dann holst du dir das Lizenz-Upgrade. Hier experimentieren wir noch ein wenig. Unserer Erfahrung nach ist es den Kunden lieber, Sicherheit zu haben. Das soll heißen: Sie bezahlen lieber und können die Technologie dann voll nutzen, anstatt im Hinterkopf behalten zu müssen, dass sie noch weitere Abgaben leisten müssen.

Making Games Wie unterstützt ihr Entwickler, die von einer anderen Engine auf Vision umsteigen möchten?

Andrew Powell Es gibt einen gewissen Unterschied in der Technologie, aber ich denke, dass jemand Havok in Betracht zieht, weil er mit einer C++ Engine arbeiten möchte. Natürlich kann man Assets aus den Modelling-Datensätzen wiederverwenden. Eigentlich alle Daten, die man für Unity oder die Unreal Engine erstellt hat. Da gibt es keinen größeren Unterschied als beim Umstieg von Unity auf UE oder umgekehrt. Allerdings nutzen wir Lua als Skriptsprache, während Unity Java Script nutzt und Epic keine festgelegte Skriptsprache hat. Es gibt also unterschiedliche Konzepte, aber wir unterstützen natürlich jedes Team, das zu unserer Engine wechseln möchte. Wir haben intern auch einige Mitarbeiter, die sich um Business2Business-Angelegenheiten für Vision kümmern.

Making Games Aber es gibt keine Templates, Demolevel oder Ähnliches?

Andrew Powell Nicht wirklich, weil es von Fall zu Fall unterschiedlich ist. Wenn dein Spiel viel Scripting hat, ist das eine andere Voraussetzung als bei einem Spiel mit vielen Standards.

Making Games Vision wurde für die PS3, Xbox 360, Wii U, PC etc. optimiert. Wie passt ihr die Engine an die neue Konsolengeneration an?

Andrew Powell Wir konzentrieren uns mit Vision im Moment mehr auf Mobile als auf PlayStation 4 und Xbox One. Deshalb haben wir auch noch kein Datum für unsere Next-Gen-Version der Engine bekannt gegeben. Aber die aktuelle Vision-Fassung ist voll kompatibel mit DirectX 11 – es ist also kein Problem, damit für die Xbox One zu entwickeln. Die PlayStation 4 ist spezifischer, es kommt deshalb ganz auf die Nachfrage an. Und aktuell ist die Nachfrage im Mobile-Bereich viel größer. Wir behalten die Entwicklung aber natürlich genau im Auge.

Making Games Wie unterstützt ihr die Entwicklung für Virtual Reality aktuell und wie wollt ihr sie in Zukunft unterstützen? Ist das für euch überhaupt ein interessantes Thema?

Andrew Powell Das Thema ist für uns sehr interessant. Momentan bieten wir vollen Support für Oculus Rift an, wir arbeiten eng mit Oculus zusammen und haben auch Zugang zu einigen ihrer Programme. Morpheus spielt für uns noch keine Rolle, weil die PlayStation 4 noch nicht zu unseren Zielplattformen gehört. Aber an und für sich geht es bei VR immer darum, die Latenzzeit so gut es geht zu verringern und eine möglichst hohe Framerate zu erreichen. Havok war immer dafür bekannt, dynamische und detailreiche Welten zu erschaffen. VR gibt dir ganz offensichtlich eine neue Perspektive. Wenn du das Headset aufsetzt, bist du plötzlich in der Spielwelt und deine Figur sowie alles um dich herum müssen sich realistisch verhalten. Wir sind auf jeden Fall sehr begeistert von den vielen Möglichkeiten, die sich uns durch VR bieten.

Making Games Ihr liefert Vision komplett mit dem Source Code. Wie unterstützt ihr Entwickler, die damit arbeiten möchten? Bei der Unreal Engine wird beispielsweise GitHub verwendet, damit User Ideen austauschen oder ihre Arbeit anderen zur Verfügung stellen können.

Andrew Powell Wir haben ein ähnliches Modell. Im Grunde haben wir die Release-Version von Anarchy auf Git gestellt und ermutigen die User, damit zu experimentieren und Plugins, die sie erstellt haben, mit der Community zu teilen. Wir haben ein Team, das nur an Features arbeitet, die über Git veröffentlicht werden. Kürzlich haben wir ein 2D-Toolkit herausgegeben. Jetzt können die User mit Anarchy 2D-Spiele bauen.

Es ist wichtig, den Usern früh Zugang zu neuen Features zu geben. Bisher haben wir das nur für Anarchy gemacht. Ein Nutzer, der ein PC-MMO entwickeln möchte, würde also einfach nur den kompletten Quellcode bekommen. Die Philosophie der Vision Engine ist ihre Flexibilität und dass die User sie ganz an ihre individuellen Bedürfnisse anpassen können. Von uns bekommen sie zwei bis drei Updates pro Jahr, die sie dann in ihre lokale Version einfügen können. Das funktioniert momentan sehr gut.

Making Games Im Gegensatz zu Vision bietet ihr Project Anarchy kostenlos an. Wieso habt ihr euch für diesen Schritt entschieden?

Andrew Powell Unser Mutterkonzern Intel hat großes Interesse an Anarchy. Sie möchten ein starkes Netzwerk von Android-Entwicklern aufbauen. Die Unterstützung, die wir von Intel bekommen, ermöglichte uns den kostenlosen Release der Engine. Es war eine Voraussetzung des ganzen Projekts, dass wir eine gebührenfreie Version zur Verfügung stellen können. Das hilft natürlich dabei, die Entwickler-Community aufzubauen. Und vor allem die Community derer, die für Intel Android x86 entwickeln. Davon werden wir bald viel auf dem Markt sehen.

Making Games Ihr bittet die User, die Project Anarchy nutzen, um die Möglichkeit des Co-Marketings. Wie funktioniert das?

Andrew Bowell Das deutsche Studio Chasing Carrots hat unseren Wettbewerb gewonnen und für sein Spiel »Cosmonautica« 100.000 Dollar erhalten. Wir werden eng mit dem Team zusammenarbeiten, gemeinsame Pressemitteilungen veröffentlichen, sie auf unserer Website vorstellen und Case Studies erstellen. Co-Marketing heißt, dass das Spiel eine Menge Publicity bekommt – und damit auch Project Anarchy. Meiner Meinung nach sind viele Teams froh, wenn sie bei der Vermarktung ihres Spiels Unterstützung bekommen. Wie weitreichend die Zusammenarbeit ist, hängt auch vom jeweiligen Titel ab. Wir würden die Spiele gerne auf unserer Website nennen und darauf hinweisen, dass es ein Project-Anarchy-Titel ist.

Making Games Habt ihr basierend auf Feedback Änderungen an Anarchy vorgenommen?

Andrew Bowell Das war eine der größten Überraschungen, als wir Anarchy veröffentlicht haben. Wir waren überwältigt von der Menge an Feedback, das wir von der Community erhalten haben. Viele der Änderungen, die wir in den ersten sechs Monaten vorgenommen haben, resultierten aus den »Erstkontakt-Erfahrungen«, wie wir sie nennen. Wir haben sogar zum Downloadprozess der Engine Feedback bekommen, zum Beispiel, dass man sich durch zu viele Seiten klicken muss oder das Ganze zu lange dauert. Wir haben unser Bestes gegeben, um

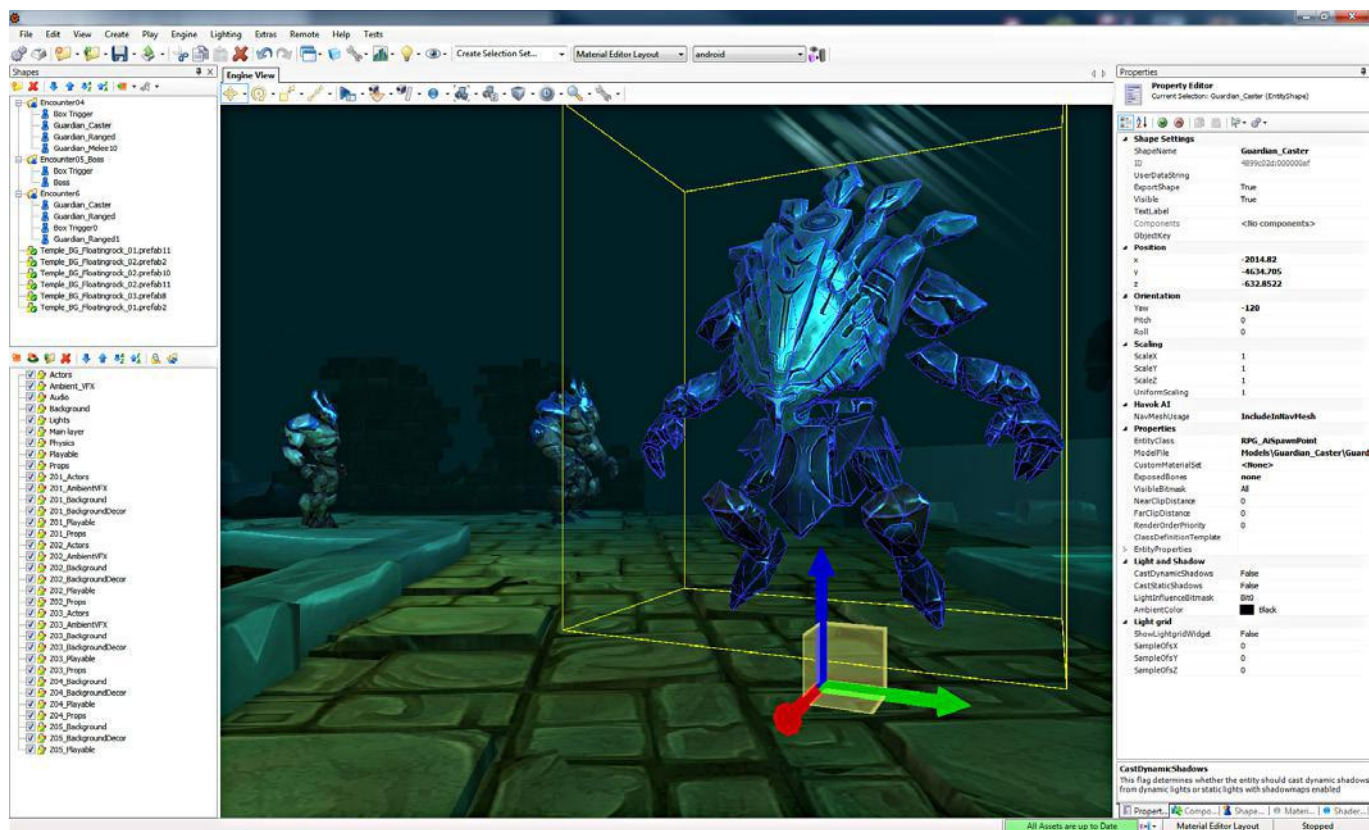
das zu verbessern. Oder eben auch die ersten paar Klicks, die man nach dem Download von Project Anarchy im Editor macht, um sich einen Eindruck zu verschaffen. Wir haben seit der Beta im vergangenen Jahr zwei große Updates veröffentlicht, und viele der Änderungen sind durch das Feedback der User zustande gekommen.

Wir stehen aktuell vor dem Release vom Support von Leaderboards für Google Play und iTunes. Damit erleichtern wir den Entwicklern die Verwaltung ihrer User, nachdem sie uns in diesem Punkt um Hilfe gebeten haben. Wir haben ein gutes Verhältnis zu unserer Community. Auf unserer Website haben wir einen speziellen Bereich, in dem die Nutzer öffentlich ihre Ideen und Vorschläge posten und mit anderen Entwicklern diskutieren können.

Making Games Welche Spiele würdest du gern mit Vision oder Anarchy entwickelt sehen?

Andrew Bowell Viele Spiele (lacht). Das Schöne an Vision ist, dass die Engine so flexibel ist. Wir haben von einem Spacetrading-Mobile-Spiel (Cosmonautica) über »Orcs Must Die« bis hin zu J-RPGs schon alles gesehen. Anarchy enthält auch unsere Animation-Studio-Produkte, mit denen die Entwickler wirklich detaillierte Charaktere erschaffen können. Ich glaube, dass wir in Zukunft Spiele mit coolen Charakteranimationen und cleverer KI sehen werden. Außerdem wird es eine Menge 2D-Spiele für Mobile im Angry-Birds-Stil geben, aber auch einige interessante 3D-Titel. Wir werden auf jeden Fall eine Menge Spaß haben. **Interview: Patricia Geiger**

»Wir waren überwältigt von der Menge an Community-Feedback.«



Seit dem Release von Project Anarchy gab es zwei große Updates: »Viele der Änderungen resultierten aus den »Erstkontakt-Erfahrungen« der Entwickler.«

... AND THEN IT RAINED

A BITTER-SWEET SUCCESS STORY

When developing your first game without any insights how the business works, a lot of backlashes might hit you. But for indie developer Megagon Industries the very successful release of their game had negative side-effects, too.



Daniel Helbig is Freelance Game Designer and Cofounder of Megagon Industries.

Daniel Helbig has been working as a Game Designer for six years now. After his guitar teacher attested his complete lack of musical talent to him at an early age, Daniel put all his creative energy into developing interactive experiences, hoping that he would someday be able to feel like a rock star. He started working as a freelance Game Designer in Berlin in 2010 and co-founded the creative and game thinking agency »Die Hobrechts« as well as the indie studio »Megagon Industries«.

[@DannyHellfish](#)

On May 1st, 2014 we released our first game »... and then it rained« for iOS. Knowing that most indie game productions fail from a commercial point of view – even very good ones – and that for every success there are probably a few hundred games which never make it to the top, we were already prepared to share that fate. Lucky us, we made our peace with that long before we actually started to work on the game.

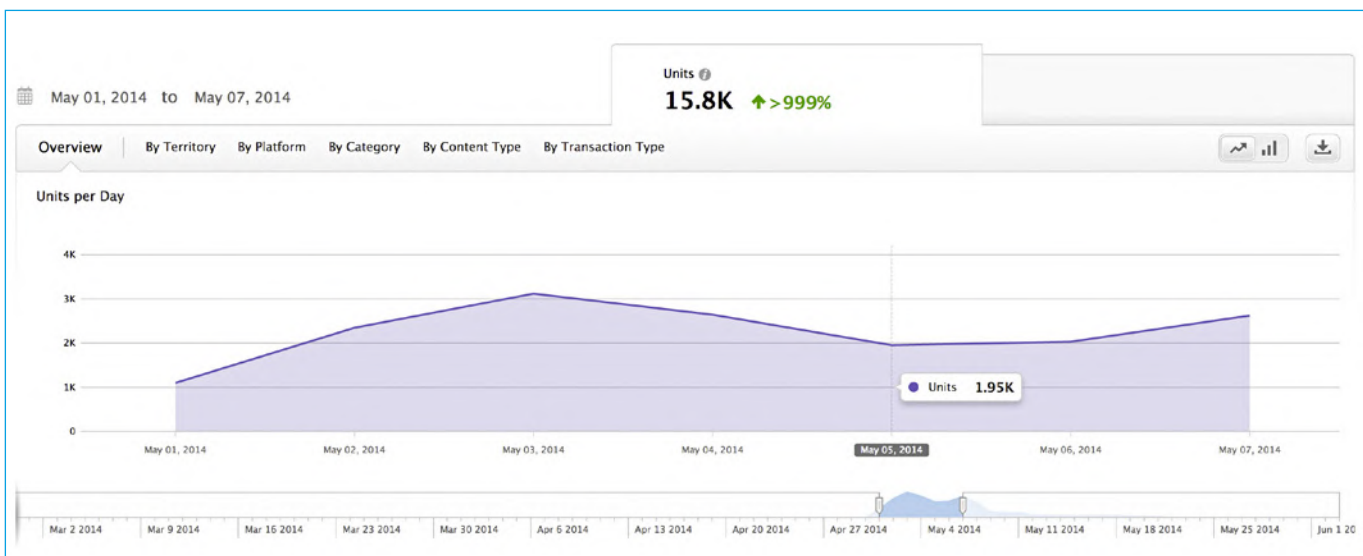
Megagon Industries consists of a bunch of game industry freelancers, who all have their own business and customers, and – being burned by a few previously released games in different constellations – we never actually planned to make a living from the games we make. This independence allows us to make very unique games for a very small audience (mainly ourselves) and also take as much time as we need to polish them till it hurts.

Unexpected success

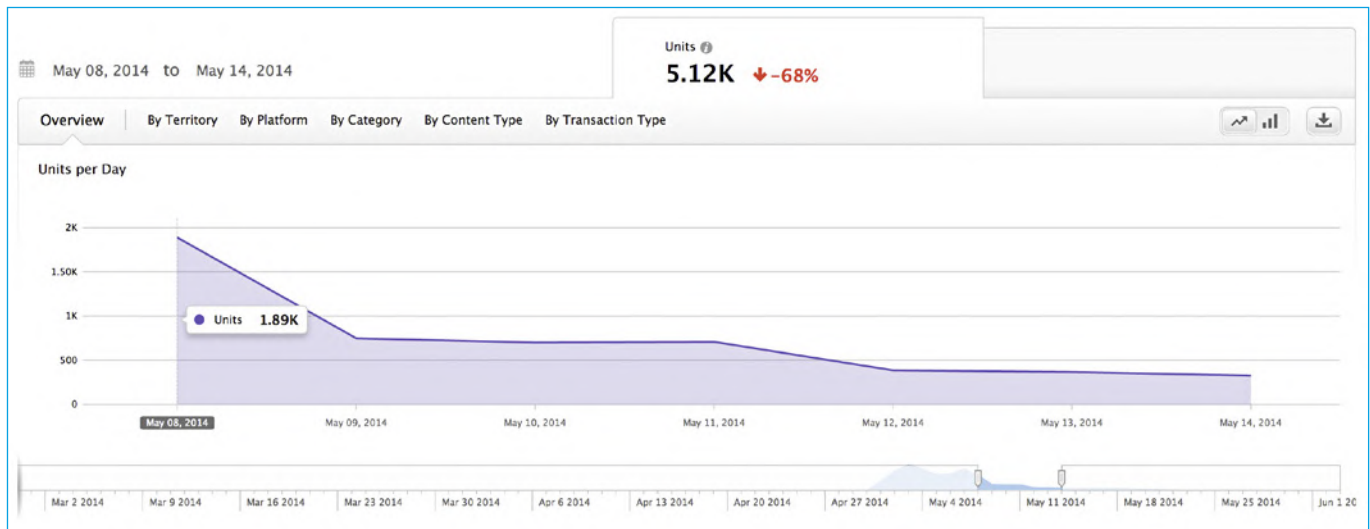
»... and then it rained« was therefore never aimed to be a commercial success. It is the first game I have ever programmed and in the beginning it was primarily meant to be a learning experience. But as the positive feedback for the game became more and more enthusiastic, it somehow evolved into an eight month long serious production.

Being in the luxury situation to not rely on App Store revenues also allowed us to stay completely ignorant to the current app market business situation and so we decided to make a premium app (tier 2) with no In-App purchases or ads. Just pay and play.

We managed to keep our expectations and stress levels fairly low during the whole production until a few weeks before release, when we got the chance to show the game directly to Apple and got very positive feedback. An App-Store-feature was now suddenly within the range of possibility, but being grumpy Germans



Before releasing »... and then it rained« no one ever had heard of Megagon Industries. But yet, the game sold nearly 16.000 copies the first week after release - a welcome surprise at first.



Sales dropped rapidly. While »... and then it rained« sold more than 1,000 copies a day during the first two weeks, only an average of 30 copies per day was downloaded two weeks later.

(the glass is always half empty ... and dirty ... and probably filled with poison ...) we were still very skeptical about a possible commercial success.

One of the main reasons for our skepticism was that we absolutely lacked press contacts. We knew not a single person from the hundred people on our press list, who received a promo code the day before release and we knew that neither of them had ever heard of Megagon Industries before.

So when May 1st finally arrived, we were prepared for the worst. But we sold nearly 16.000 copies the first week!

Reach your audience fast or never?

We got a small Apple feature in »New & Noteworthy« and a small banner. Not only did the game sell well, we also got amazing reviews from a lot of big blogs and magazines. People who had obviously deleted our release mail, suddenly asked us for promo codes. The German indie community was also an amazing support and the first week we were all over Twitter and Facebook. It really felt like we hit the indie game jackpot.

We knew that most mobile games make 90 per cent of their sales during the first week, but when Apple added us to »Our Favorite Games« in the beginning of the second week and sales were still strong at around 1.000 units per day, we really thought that we might be the exception from the rule. No matter how experienced and hardened you are, the dream of making a living through your own games is always lurking in the dark shadows of every indie game developer's mind and we slowly started to believe in that dream again. We began to hope that the game might continue to sell well enough for a longer time to provide some kind of financial baseline for us.

We couldn't be more wrong! Four weeks after release, IndieCade announced us as part of their E3 Showcases. That day we sold 27 units ... And this was only the latest disappointment

in a series of »this is absolutely amazing, but doesn't make you sell one more unit«-events. Here is a short list of all the fantastic things happening to us over the last month:

- A lot of awesome reviews (Cult of Mac, PC Mag, AppAdvice, 148 Apps, Indie Game Mag, TUAW)
- App-store-features like »Favorite Game« and »Exclusive on the App Store«
- We got recommended by Apple through their official Twitter channel
- Guy Cocker recommended the game on Sky News
- We made some visual postmortems, which were received very positively on Facebook
- IndieCade E3 Showcase



We shared our success story with our fans on Facebook by posting info charts like those, which were received very positively.

Hard lessons to be learned

Each one of these events was absolutely fantastic in itself and really boosted our developers pride, but to our astonishment they didn't affect our sales at all.

This leads us to some disappointing insights:

A hit is not necessarily a hit!

Although we made a seemingly very good and unique game with good reviews and all the support one can wish for, we only managed to sell 24.000 units. On the one hand this is totally awesome for us because we never calculated with any money at all from the game, on the other hand this means that if we would want to make a living from our games, it would not work.

But why feel sad about this? I think we already knew that in a way and it even proves us right not to have founded a classical studio, but to stay freelancers. I guess somehow we still had the hope that with every game we make, there is always the slight chance to have a hit game and to be suddenly financially independent enough that we can devote all our time (and not only a small part which mainly consists of evenings and weekends) to make our own games. Now we have learned that even a small surprise hit won't provide that kind of independence.

Press only works the first week.

We can't tell how press articles transformed into sales during the first week as so many things happened together, but all the press we got the following weeks didn't result in any visible sales spikes at all. Of course one could argue that we might have dropped even more without it, but my personal assumption is that as an indie developer you will find the largest part of your core audience during the first week.

Maybe not doing marketing is a stupid idea.

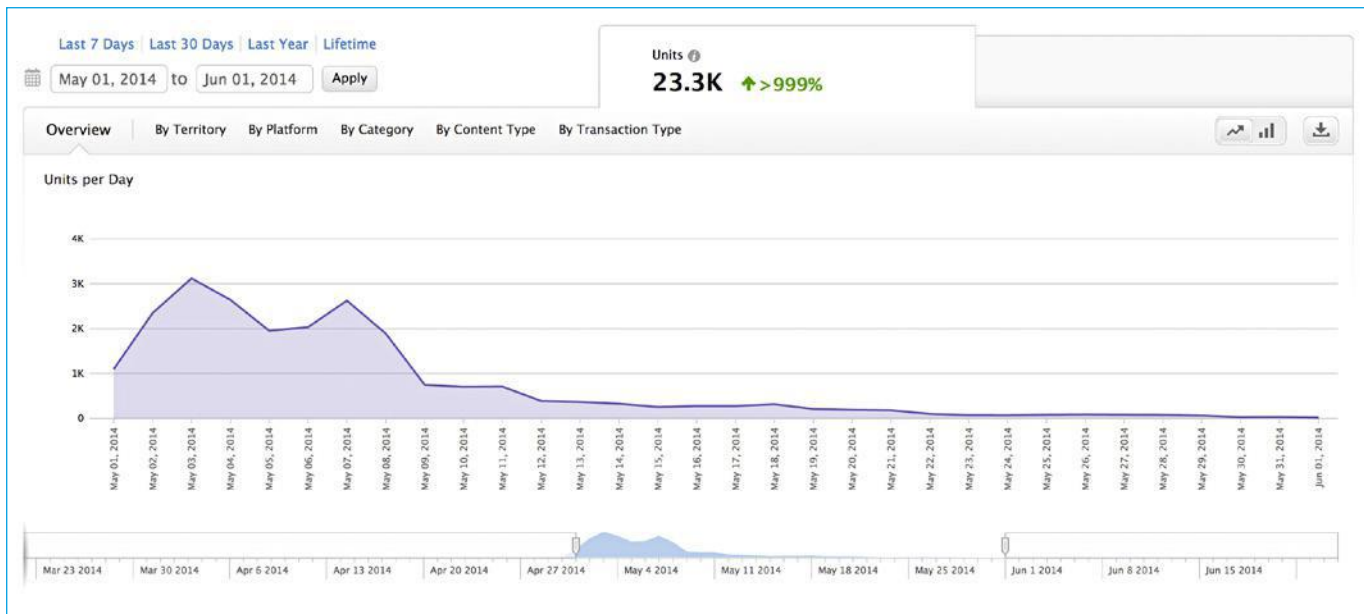
I always thought that good press articles would do much better – as for sales – than any advertising ever could and I also believed that good press is everything when it comes to successful indie games. Now I wonder if this might be naive. Maybe, when we entered the top 25, we should have taken the opportunity and reinvested the incoming money into ads to push us even further up the mountain. Maybe we might have reached all those people then who never actually read the Apple-, mobile- or indie-game-magazines and -blogs. The real problem is: We don't actually know if it would have helped or not ...

It's your fellow indie developers who give you your start push.

It was an amazing experience to see how the German indie community rallied their troops and supported us during the first release week. How they pushed us into the German top charts, even before any big press reviews came out, demonstrated to us how strongly we neglected the international community during the last year. Being part of a strong community is probably even more important than having good press reviews.

You can't rely on one platform.

We really like to develop for iOS. The technical fragmentation is manageable and the games we want to make go very well with the overall feeling of the platform. It's also probably the only place where there is at least a slight chance to sell some units with a premium app. And finally the people at Apple gave us the best, deepest and most critical feedback yet. But the last four weeks also demonstrated the limits of the paid app market on iOS and of being featured by Apple. If we would seriously try to make a living as an indie developer, we would need to look into other platforms and



»... and then it rained« did sell around 24.000 units. As the chart shows most were downloaded during the first week after release due to not enough long-time marketing and press-coverage.

think about bringing our games to PC/Mac or even a stationary console.

Good is just not good enough.

We knew that »... and then it rained« is not the perfect game. You need to be into minimalist arty hipster apps to really appreciate its style and atmosphere and it is probably too hard for a lot of people. It also crashed way too often before the latest update. But because it can be picked up and played within seconds and people all over the globe seemed to really have a strong positive resonance with the sound and atmosphere of the game, we started to think that it might appeal to a much broader audience than we initially thought.

In the end this doesn't seem to be the case. Maybe the game is just not special enough. Although it's unique, it's neither provocative nor does it have any memorable moments forcing you to desperately talk about it with your colleagues over lunch. It might just not be sticky enough in times where hundreds of games get released every week.

Success changes you.

If the game had sold 20 units per day from the very beginning, it would have been completely okay for us. But although we never did the game for commercial reasons, at some point during the last four weeks the measure for success shifted away from quality and good reviews to the number of units sold per day (not the actual profit by the way). Selling only 20 units per day feels disappointing, which is utterly stupid.

Conclusion

We focused this postmortem mainly on the negative sides of success, because we feel that these are the areas where we learned the most (especially about ourselves) and can share the most valuable insights with other developers. But that doesn't mean that the game isn't a big personal success for us. Sometimes it even felt like experiencing a personal version of »Indie Game: The Movie«.

We got so many supporting mails by gamers and especially by fellow developers that it really overwhelmed us. Because of the game and thanks to IndieCade, we now also have the special opportunity to exhibit the game at E3, which is something we would never thought possible a few weeks ago.

The success during the last months also strongly motivates us to keep working on the game and on ourselves: We began to look into porting the game to other platforms and this article and the upcoming E3 are our humble start to reach out to the international community. We will also keep updating the game to make it even better over the next year. We started to have a closer look into the whole App store optimization thing as we never really cared about that before. We are also working on the next game based on an idea, which might be stickier than »... and then it rained«.

Megagon Industries



Megagon Industries publishes games developed by Jan Bubenik, Severin Brettmeister, Daniel Helbig and friends. Located in Berlin

and Canada we make games exactly as we want them to be. They are currently working on their second game »Plixel«.

But now we are also aware of the limits of what we can achieve as a studio at this very moment and that we will have to continue to make games the way we did before. This means working half our time as freelancers and only using the remaining free time to develop our own games.

And as much as we want to, we also can't shake off this lingering bitter taste, because for a moment it seemed like this whole indie thing could work. But then this dream crawled back into the shadows.

Daniel Helbig



»... and then it rained« got a lot of attention in the beginning - even Apple did a feature via their official Twitter-account.



4 PLATTFORMEN, EIN IDENTISCHES SPIEL DIE ENTSTEHUNG EINES BASKETBALL GAMES MIT ADOBE AIR

Wer ein Spiel für mehrere Plattformen identisch entwickeln will, steht bei der Wahl der Technik vor einer schwierigen Entscheidung. Jan Reichert erklärt, wieso sich das Studio für eine Entwicklung mit Adobe Air entschied und welche Hürden dabei überwunden werden mussten.



Jan Reichert
ist CEO von kr3m. media.

Jan ist Gründer und Geschäftsführer von kr3m. media. Er ist in erster Linie für Konzept, Marketing und Business Development zuständig. Von 1999 bis 2008 leitete er die Entwicklung und programmierte in den Anfangstagen der Firma die meisten Anwendungen selbst, etwa die erste Version der Kindercommunity »Ravensburger Kinderwelt« sowie zahlreiche Casual Games. Aktuell hält er regelmäßig Vorträge und Workshops über Gamification, Marketing Games und Social Media Infotainment. In seiner Freizeit unterrichtet Jan den Vollkontakt-Kampfsport Enshin Karate. Er trägt den vierten Dan und kämpfte dreimal bei der Weltmeisterschaft. Trotz Firma und intensiver Vereinstätigkeit ist er gerade zehn Jahre verheiratet und hat eine Tochter (8) und einen Sohn (4).

Ein identisches Spiel auf allen verfügbaren Plattformen (Desktop, Facebook, Smartphones und Tablets), das Spieler aus ganz Deutschland zu einem überregionalen Wettbewerb zusammenbringt, grafisch ansprechend in echtem 3D gestaltet ist, die Möglichkeiten der sozialen Netzwerke einbindet und ein authentisches Basketball-Gefühl bei gleichen Chancen für alle Spieler vermittelt: Es waren nicht gerade kleine Ziele, die wir uns bei der Entwicklung unseres Basketball Games gesteckt hatten. Doch dank der jahrelangen Erfahrung als Entwickler von Marketing Games konnten wir diese Vorgaben innerhalb von weniger als zwölf Wochen erfüllen und ein wirklich innovatives Produkt fertigen.

Das Ziel: Alle Plattformen erreichen

Dreh- und Angelpunkt sämtlicher Entwicklungsüberlegungen und der zugrunde liegenden Konzeptionierung war die Prämisse, alle Plattformen mit dem Spiel zu erreichen – und das eben in vollem 3D! Egal ob im Desktop Browser, als Facebook-Spiel oder als mobile App für iOS- und Android-Smartphones sowie

-Tablets – die App sollte überall die gleichen Features aufweisen, die gleiche ansprechende grafische Gestaltung und die gleiche Usability. Nur so können alle relevanten Zielgruppen plattformübergreifend erreicht und verbunden werden sowie ein authentisches Wettkampfgefühl innerhalb der sozialen Einbindung entstehen. Drei technische Lösungsansätze nahmen wir unter die Lupe: eine Umsetzung in HTML5 mit WebGL, in Unity 3D oder mit Flash/AS3 mit Stage3D und der Away3D-Library.

HTML5 mit WebGL und Unity3D schieden schnell aus beinahe den gleichen Gründen aus: Beide haben auf jeweils unterschiedlich relevanten Plattformen eine zu geringe Marktdurchdringung. WebGL-fähig sind nach aktuellen Zahlen gerade mal 5% der Browser auf mobilen Geräten und circa 40% auf Desktop. Die Verbreitung des Unity3D-Plugins im Desktop Browser für eine breite Zielgruppe ist ebenfalls nach wie vor viel zu gering.

Die Entscheidung für AS3 und Adobe Air

Wir entschieden uns für AS3 mit FlashDevelop als Entwicklungsumgebung. Das Flash-Browser-Plugin und der Export als native App



Das Gameplay auf iPhone und Android aus zwei der vier unterschiedlichen Blickwinkeln.

über Adobe Air garantieren noch immer eine maximale Marktreichweite, wie sie gerade bei einem crossmedialen Marketing Game gewünscht ist. Wir entwickelten die komplette Grundanwendung in AS3. Aus dem Quellcode kompilierten wir mit leichten Anpassungen native mobile Apps für Android und iOS mithilfe von Adobe Air und Flash-SWFs für das Desktop-Browser-Plugin mit Flex SDK. Zusätzlich möchte ich anmerken, dass die Zukunft von WebGL auf mobilen Endgeräten aktuell alles andere als gesichert ist. So wurde eine WebGL-Unterstützung zwar für Android 4.4 Kitkat angekündigt, seitens Google aufgrund von Sicherheitsbedenken aber in letzter Minute wieder entfernt.

Verringerung der Rechenlast

Zu Beginn der grafischen Entwicklung modellierten wir zuerst das virtuelle Stadion als komplettes und detailliertes 3D-Modell in Blender nach einer realen Vorlage – bis hin zur originalgetreuen Bandenwerbung und den individuellen Besuchern auf den Rängen. Aus diesem Modell wurden dann 20 verschiedene Ansichten gerendert und perspektivisch korrekt verzerrt auf den Skydome als Texturen aufgetragen, der das virtuelle Spielfeld umspannt. So konnten wir die Illusion erzeugen, im realen Stadion Freikörbe zu werfen, und mussten dennoch nicht mehr als 20 Polygone für die Umgebung verwenden. Als »reale« 3D-Objekte müssen daher im Spiel selbst nur Ball, Korb und Spielfigur berechnet werden. Dieser Ansatz verringerte auch nochmals deutlich die Rechenlast für mobile Endgeräte,

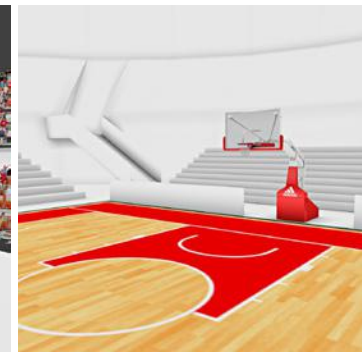
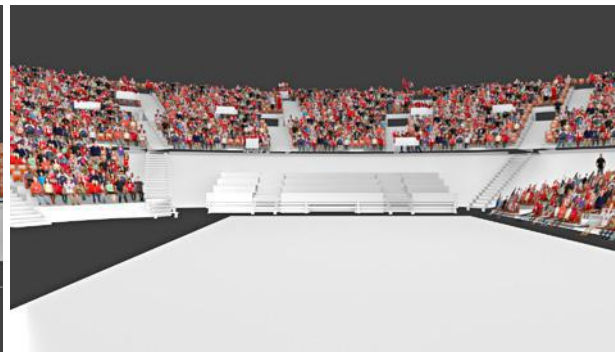
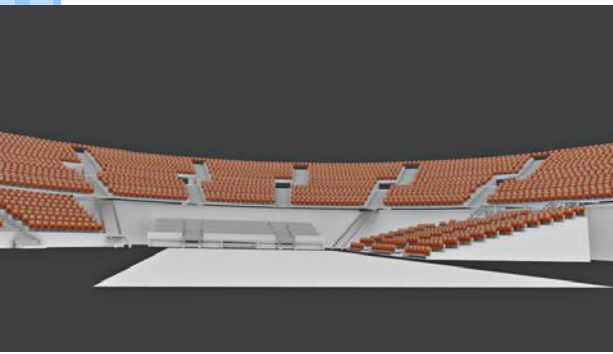
ohne an der Grafikqualität Abstriche machen zu müssen. Mit dieser Lösung wäre sogar eine völlig freie Kameraausrichtung durch den Spieler möglich gewesen, aber für das finale Spiel genühten vier verschiedene Blickwinkel, die jederzeit frei gewechselt werden können. Eine völlig freie Kameraplatzierung hätte Einbußen an der Usability bedeutet, das Benutzerinterface unnötig aufgebläht und viele Casual Gamer erst mal überfordert.

Das Unternehmen

Bereits seit 1999 steht die Karlsruher Agentur kr3m. media für Kommunikation und Reichweite durch Games, Apps und Social Media. Ihre Vorreiterrolle zeigte sich nicht zuletzt an der Auszeichnung als Facebook Preferred Marketing Developer für Apps. Mit ihren beiden Abteilungen kr3m. social media marketing und kr3m. games & apps bietet sie eine umfassende Betreuung, kreative Konzepte sowie durchdachte und zielgerichtete Lösungen, die Marketing mit spielerischem und emotionalem Mehrwert verstärken. Alle Projekte werden mit 35 Mitarbeitern komplett inhouse erledigt. Zu den Kunden der Agentur zählen unter anderem Disney, Deutsche Telekom, Daimler oder Deutsche Bahn.

www.kr3m.com





Das linke Bild zeigt das 3D-Modell des Audi Domes, in der Mitte der 3D-Dome mit Zuschauern auf den Rängen und Tribünen. Auf dem rechten Bild ist das Spielfeld mit Korb und leeren Zuschauerrängen zu sehen.



Das Spielfeld, zusammengesetzt aus den vorher erstellten und berechneten Elementen: Das 3D-Modell des Korbes vor der perspektivisch richtig verzerrten 2D-Ansicht, die aus dem 3D-Modell des Domes gerendert wurde.

Fairness für alle durch flexibles UI

Neben der grafischen Qualität war – wie für ein Sportspiel typisch – auch das richtige Spielgefühl für den Erfolg entscheidend: Für eine glaubwürdige Flugkurve des Basketballs sorgt die Open-Source-Physik-Engine »Bullet-Physik«, die bereits in Spielen etwa der »GTA«-Reihe oder in Pixar-Filmen zum Einsatz kommt.

Genauso viel Sorgfalt wie dem Gameplay widmeten wir der Steuerung. Das User Interface wurde auf die wichtigsten Elemente reduziert, um auf allen Plattformen intuitiv und zugleich beherrschbar zu sein. Mit nur einem Klick legt der Spieler die Flugkurve des Balles fest, der Impuls des Wurfes variiert dabei konstant und wird über ein Interface am linken Bildschirmrand ausgegeben. Die Kombination aus der optimalen Wucht und der richtigen Kurve entscheidet dann darüber, ob der Ball durchs Netz geht, abprallt oder gar im Publikum landet. Um allen Spielern zudem die gleichen fairen Spielbedingungen zu bieten, ist das Interface jederzeit zwischen einer für Rechts- und Linkshänder angepassten Variante umschaltbar – eine Funktion, die viele Mobile Games ihren Spielern immer noch schuldig bleiben. Ebenfalls aus Fairness-Gründen entschieden wir uns gegen eine Swipe-Steuerung. So intuitiv diese bei Geräten mit Touchdisplay auch sein mag – sie würde Desktop User mit Maus und Tastatur benachteiligen, weil eine Mausgeste nicht die gleiche Genauigkeit und Haptik mit sich bringt. Weil aber gerade Fairness und Vergleichbarkeit des Wettbewerbs auf allen Plattformen ein absolutes Key Feature waren und kein Spieler bevorzugt oder benachteiligt werden durfte, schied das damit also aus.

Um das User Interface zudem möglichst reaktionsschnell zu gestalten, setzten unsere Entwickler bei seiner Umsetzung auf das Starling Framework. Diese Open Source Actionscript Library erlaubt es, 2D-Darstellungen direkt über die GPU rendern zu lassen und ihnen damit einen massiven Performance-Schub zu verpassen.

Unspielbarkeit durch Berechnung von 3D-Körpern auf der GPU

Bei der Entwicklung mit Stage3D hatten wir ein Problem, das uns sehr viel Zeit kostete: Flash 3D setzt zur Berechnung von dreidimensionalen

Körpern auf die GPU und damit die Hardwarebeschleunigung des jeweiligen Systems – aber bestimmte Bedingungen können zu einem massiven Performance-Abfall führen. Dann fällt die Grafik-Berechnung plötzlich auf die CPU zurück, was vor allem bei älteren Smartphones ein K.-o.-Kriterium darstellt. Gerade diese älteren Modelle haben oftmals zwar schon eine sehr leistungsstarke GPU, allerdings eine im Vergleich nicht wirklich performante CPU.

Dieser Performance-Absacker trat auch während der Entwicklung unseres Spiels auf. So erwies sich eine erste Version der Anwendung in allen aktuellen Browsern als sehr gut spielbar – aber auf einigen Smartphones erinnerte die Framerate eher an ein schlechtes Daumenkino bei unter 10 fps. Schnellere Smartphones waren von diesem Problem nicht betroffen, weil sie die nötige Rechenleistung mitbrachten, doch gerade auf den in Deutschland immer noch weit verbreiteten Modellen iPhone 4 und Samsung S3 wurde unsere App auf einmal unspielbar – der absolute GAU.

Die Lösung: Neugestaltung des Spielermodells

Zuerst gingen wir ganz klassisch an dieses Problem heran. Wenn eine Anwendung plötzlich in der Performance so massiv einbricht, liegt das meist daran, dass man sie mit Features überfrachtet hat. Also prüften wir in einer



Eine Neugestaltung des Spielermodells sorgte dafür, dass das Spiel plattformübergreifend eine Performance auf Desktop-Niveau erreicht.

Das Entwicklerteam

Projektmanagement

Christian Kaufmann

Konzeption & Spieldesign

Christian Kaufmann

Entwicklung & Programmierung

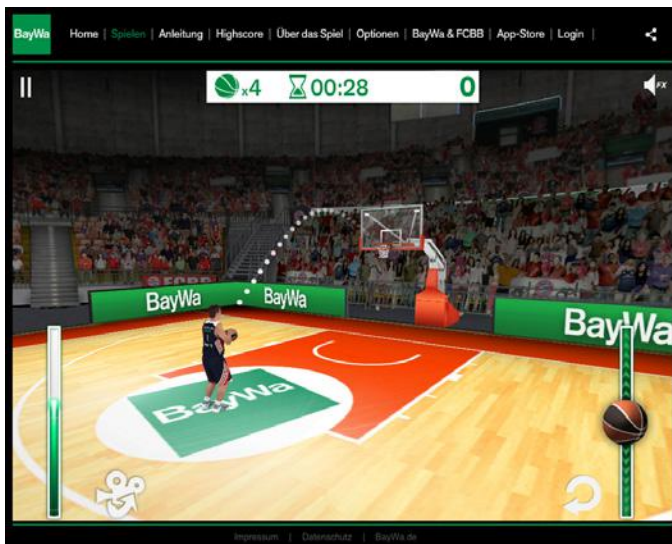
Robert Krasky

Hendrik Nagel

Charly Brischle

2D/3D-Artist

Florian Frey



Der grüne Balken links unten zeigt die Wurfstärke an, der Regler auf der rechten Seite bestimmt die Ballkurve. Per Klick löst er den Wurf aus.

gemeinsamen Taskforce aus Entwicklern und Grafikern jeden Programmbestandteil einzeln durch, von der Polygonzahl bis hin zur Animation des Spielers und des Basketballs. Aber des Rätsels Lösung war dann doch eine ganz andere: die Anzahl der Bones pro Vertex des verwendeten 3D-Modells. Bei mehr als vier Bones verlagert Stage3D die Berechnung von der GPU auf die CPU; sie erfolgt dann nicht mehr hardwarebeschleunigt, sondern muss rechenzeitintensiv vom Prozessor erledigt werden. Ein gründlicher Review des Spielermodells und eine damit verbundene Neugestaltung konnten diesen Stolperstein entfernen und die Performance auch auf mobilen Endgeräten wieder auf Desktop-Niveau bringen. Hier zeigte sich auch, dass gerade für eine plattformübergreifende Entwicklung ein kontinuierliches Testing durch die QA ab der Alphaversion eines Produktes von entscheidender Wichtigkeit ist, insbesondere auf langsamen Endgeräten – nerdige Entwickler haben meist performante, aktuelle Geräte.

Einteilung der Spieler in Teams

Ebenfalls nicht ganz unwichtig waren die Vernetzung aller Spieler, die Einteilung in konkurrierende Teams sowie die Ermittlung des Highscores pro Mannschaft und Region. Um den Wettkampfcharakter des Marketing Games zu forcieren, teilten wir Deutschland entlang der Regierungsbezirke auf. Jeder Spieler schloss sich nach Registrierung – entweder per Facebook-Login oder mit E-Mail-Adresse – anhand seiner Postleitzahl einer Region und damit einem Team an und spielte für dieses. Zusätzlich zum reinen Korbwerfen gab es nach jeder Runde mit fünf Versuchen eine Quizfrage zum Thema Basketball zu beantworten. Bei richtiger Beantwortung warf diese nochmals Bonuspunkte ab. Um die einzelnen Regionen direkt miteinander vergleichen zu können, wurde ein Mittelwert aus den erreichten Punkten der 15 besten Spieler pro Team gebildet.

Wichtig war uns ebenfalls die Möglichkeiten der sozialen Netzwerke und – geschuldet seiner vorherrschenden Marktstellung – natürlich primär von Facebook einzubinden. So entstand das Feature, Freunde direkt einzuladen und über eine interaktive Deutschlandkarte angezeigt zu bekommen. So konnte der jeweilige Spieler gleich auf einen Blick sehen, welcher seiner Freunde für welches Team gegen ihn oder gemeinsam mit ihm antrat. Dadurch wurde ein direktes Wettkampfgefühl im sozialen Miteinander der Spieler erzeugt. Umsetzung fand unsere Entwicklung in Kooperation mit Aquarius Consulting für BayWa AG als BasketBattle – ein Spiel, das Basketballfans in ganz Deutschland begeisterte.



Jan Reichert

Das Spiel

BayWa BasketBattle ist nach wie vor unter www.basketbattle.de in der Browservariante verfügbar sowie kostenlos im Apple App Store und auf Google Play erhältlich.



INTERAKTIVES SOUNDDESIGN FÜR THE LAST TINKER: CITY OF COLORS



2.700 Soundeffekte und rund 70 Minuten Musik - das Spiel »The Last Tinker: City of Colors« hat einen umfangreichen und vielseitigen Soundtrack. Filippo Beck Peccoz erklärt, wieso das Orchester für ein Minispiel absichtlich schiefe Töne anschlagen musste und wie selbst ein Hund zu den Aufnahmen beigetragen hat.



Filippo Beck Peccoz ist Audio Design Freelancer.

Von Filippo gibt es Aufnahmen auf Kassette, in denen er »Super Mario« und »Mega Man« nachträllert - im Alter von 5 Jahren. Die Leidenschaft für Videospelmusik konnte er schon damals nicht verleugnen. Nach einem erfolgreichen Studium in Filmmusikkomposition in den Vereinigten Staaten (Berklee College of Music in Boston, MA) machte er sich wieder auf nach München, um dort als selbstständiger Audio Designer Sounds und Musik für Games zu entwickeln. Seit 2010 arbeitet er mit verschiedensten Entwicklerstudios zusammen. In Bayern hat Filippo an allen Projekten von Mimimi Productions, Bit Barons und Klonk Games gearbeitet sowie für Ravensburger Digital, Travian Games und Koch Media Audio produziert. Nächstes Jahr wird sich Filippo mit seiner Frau Ilenia an die Entwicklung ihres ersten Spiels »Josephine« machen, dessen Konzept 2011 vom FFF Bayern gefördert wurde.

Als frischgebackener Uni-Absolvent bin ich nach München gezogen, auf der Suche nach jungen Teams mit tollen Ideen, die viel Audio brauchen. Beim Global Game Jam lernte ich bald den Großteil des Mimimi-Teams kennen - und seitdem arbeiten wir zusammen.

Unser erstes kommerzielles Projekt, das Zen-artige Mobile Game »daWindci«, hat uns endgültig gezeigt, dass wir sowohl aus ästhetischer wie aus organisatorischer Sicht bestens zusammen funktionieren.

Die Musik in The Last Tinker

2011 begann das Team von Mimimi Productions, einen Prototyp für einen 3D-Platformer zu entwickeln. Ich hatte das Privileg, schon die allerersten Skizzen sehen zu dürfen, was von großem Vorteil für die Musik war: Der Soundtrack konnte über einen langen Zeitraum hinweg mit dem Spiel zusammenwachsen.

Der allererste Track

City of Colors, eine Stadt, die nur aus Farbe, Papier und Kleber besteht, war lange Zeit ein Ort der Harmonie und Toleranz. Doch die Bewohner verfielen dem Streit, welche Farbe die schönste sei - und schon bald begannen sie, sich zu isolieren und gegenseitig zu hassen.

Das Grundkonzept von The Last Tinker ist die Benutzung von drei Farben, um die Emotionen von Charakteren und Feinden zu verändern. Der Held des Spiels, ein Junge namens Koru, ist in dieser zersplitterten Welt das letzte Wesen, das diese Kraft besitzt.

Diese Story war mir 2011, bis auf den Einsatz der unterschiedlichen Farben, noch vollkommen fremd. Der erste Track entstand mit folgenden Adjektiven im Kopf: märchenhaft, fröhlich, farbenfroh, orchestral. In den Soundbeispielen ist dieser erste Track zu hören.

Als die Entwicklung voranschritt, wurden die Story und die Charaktere klarer definiert. Koru selbst veränderte sich drastisch, und es



Der niedliche Tap begleitet den Titelhelden Koru durch die Welt von The Last Tinker. Eine handgemachte Version von Mimimi-Artist Bianca Dörr leistet uns im Studio bis heute Gesellschaft.

kamen immer mehr quirliche Figuren dazu, die dem Spiel einen ganz besonderen Touch gaben. Wir wussten, dass nun die ursprüngliche Musik plötzlich viel zu kindlich und brav klang. Besonders die Holzbläser im Original klangen einfach nicht mehr »richtig«. Also begann ich, nach dem »wahren« Tinker-Sound zu suchen.

Ich habe mich von der collageartigen Pappmachee-Grafik stark inspirieren lassen. Die Idee war, nicht mehr ganz in diese mittlerweile zu märchenhafte Richtung zu gehen, sondern viele Elemente wild zu mischen – Gitarren mit Elektronik, organisch klingende Perkussionen mit Drum-Maschinen, Orchester mit Synthesizer. Eben ein musikalisches Potpourri, so farbenfroh wie die Welt von Last Tinker selbst.

Den ersten Versuch in diese Richtung kann man im Audiobeispiel »nicetry« hören. Er war allerdings noch gar nicht zufriedenstellend ... ich war weiterhin in diesem märchenhaften, »politically correct« Musik-Modus, wollte aber viel mehr Farbe reinbringen. Mir gefiel zwar die Idee mit den Gitarren und der Marimba als leitende Instrumente, doch es fehlte einfach etwas Grundlegendes. Der Song blieb somit in der Luft hängen.

Mittlerweile war auch klar, dass das Spiel in den Slums beginnen würde, also dem Teil der City of Colors, der noch alle Farben beherbergt. Ich wollte sofort sämtliche wichtigen Elemente in einen Track zusammenbringen. Nach ein paar Tagen der wilden Experimente im Studio hatte ich dann endlich den Slum Song. Er hat akustische Gitarren und Streicher, klingt jedoch nicht mehr nach Märchen; er hat aber auch sehr viel Perkussion, allen voran den Klang der Hang Drums, den ich persönlich sehr liebe und der zu einem der wichtigsten Elemente im Soundtrack geworden ist. Synthesizer spielen über Streicher, der Bass ist ebenfalls synthetisch. Die Intensität variiert, aber als Loop funktioniert der Track auch. Eine kunterbunte Mischung also – wir entschieden uns, in diese Richtung weiterzumachen.

Musikalische Charakterisierung der Districts

Mittlerweile hatte sich auch die Grundstruktur des Spiels herauskristallisiert: Die Welt wurde in drei verschiedene »Districts« aufgeteilt, welche die drei Farben Rot, Grün und Blau repräsentieren. Aber nicht nur der grafische Stil ist in diesen Vierteln unterschiedlich: Die Farbe spiegelt auch die Emotionslage der Bewohner wider.

Somit sind die Echsen aus dem Red District permanent wütend und aggressiv, die kleinen Hasen im Green District schrecklich schüchtern und verängstigt und die großen blauen Bären im Blue District hoffnungslos deprimiert.

Das ist natürlich eine perfekte Voraussetzung für einen extrem abwechslungsreichen Soundtrack. Die Charakterisierung der Districts sollte nicht subtil, sondern ganz klar erkennbar sein: Von den Instrumenten zu den Rhythmen und Harmonien sollen sie 1:1 die respektive Emotion spiegeln. Dies hat auch den Vorteil,

dass der Spieler nicht nur durch die Grafik in eine komplett andere Welt gebracht wird, sondern sich auch akustisch alles wandelt.

Der Red District zum Beispiel sollte trockenstaubig und metallisch klingen, was dazu führte, dass ich einiges an Töpfen und Geschirr als Perkussion missbraucht habe. Der Green District dagegen musste die Paranoia der Bewohner einfangen, aber auf ironische Weise: Pizzikato-Streicher und die Ukulele als Hauptinstrument haben sich als gute Mischung erwiesen.

Zu guter Letzt war der Blue District an der Reihe. Schon länger hatten wir dort als sogenannten Temp Track (also ein provisorisches Musikstück, das als Richtlinie funktionieren soll) einen Tango. Uns gefiel diese Idee. Wir griffen zum Bandoneon und zum Klavier, um das Phlegma der dicken Bewohner effektiv rüberzubringen, und mischten diese Instrumente mit den mittlerweile für den Soundtrack unerlässlichen Hang Drums.

Thematische Elemente und besondere Musikstücke

In allen drei Districts gibt es sogenannte Domes (nicht ungleich den Mini-Dungeons in »Zelda«), die der Spieler aufsucht, um dort den jeweiligen Spirit zu finden und die Macht eines solchen zu erlangen. Bei den Spirits handelt es sich um magische Wesen, die Koru im Laufe des Spiels zu seinen Freunden zählen wird. Ich wollte auf jeden Fall ein einziges Thema für die Domes haben, um es dann mit der jeweiligen Instrumentierung und dem jeweiligen Rhythmus an die Districts zu binden.

Die Variationen des Themas sind ebenfalls in den Audiobeispielen enthalten: Zuerst hört man das Thema des Red Dome. Viel verzerrtes Gitarrenfeedback und eine Bouzouki spielen hierbei die Hauptmelodie. Danach folgt das Thema in der Variante des Green Dome – es ist im Vergleich viel fragiler und hektischer. Und als Letztes kommt der Blue Dome, der den Tango-Touch des Districts mit einem Bandoneon unterstreicht. Die Melodie ist im Kern jedoch immer die gleiche.

Es gab viele Momente, die eine besondere musikalische Gestaltung verdient haben. Im Spiel lassen sich Traumsequenzen, ein Interludium à la Film noir und noch mehr finden. All diese speziellen Momente haben ihren eigenen Klang, zu finden in den Audiobeispielen als »noir«, »good_dream« und »market«.

Der letzte Schliff

Mike Skalandunas, der 2011 mein Assistent war und mittlerweile in Seattle wohnt, half mir mit der Produktion von ein paar Songs in The Last Tinker: Er schrieb die Musik für die Bosskämpfe im Spiel, während ich noch an den District Tracks arbeitete.

Gegen Ende der Produktion ließen wir die gesamte Musik von Guido Hieronymus in Berlin mischen. Guido hat sehr viel Erfahrung im Filmbereich (er hat viele Werner-Herzog-



Die akustische Gitarre ist eins der wichtigsten Instrumente des Soundtracks zu The Last Tinker.



Ein Teil der bösen Bleak Spirit Stimme kommt von Link, unserem Golden Retriever, der für jeden Soundeinsatz zu haben ist.

Probehören

Eine Auswahl des Soundtracks steht unter www.makinggames.de/last-tinker-sounds zum Probehören zur Verfügung.

Produktionen gemischt) und hat für The Last Tinker wirklich tolle Arbeit geleistet, indem er die individuellen Eigenschaften der District Tracks durch sein magisches Mixing noch stärker zum Vorschein brachte.

Für mich war die Komposition dieses Soundtracks wie ein riesiger Spielplatz, auf dem ich mich austoben konnte. Vom Film-noir-Song über den Surf-Rock-Track bis hin zu Orches-

tersound ist auch alles dabei: Ein so breites musikalisches Spektrum ist nur selten in einem einzigen Spiel vertreten.

Sound Design, Ambience und Voiceover

The Last Tinker wurde mit Unity entwickelt. Ich arbeite sehr oft mit dieser Engine und habe daher gelernt, nicht nur Assets zu liefern, sondern sie auch direkt in ein Projekt einzubinden. Zusätzlich hat das Team von Mimimi Tools entwickelt, mit denen wir dann die Implementierung der Soundeffekte selbst übernehmen konnten.

Ambience

Ein gutes Beispiel für diese Implementierung sind die Ambience-Klänge, die zum größten Teil meine Frau Ilenia aufgenommen, editiert und eingebaut hat. Ilenia hat Installationskunst und Audio studiert und unterstützt mich im Soundeffektbereich seit »daWindci«.

Es war uns wichtig, die Welt nicht nur durch Musik lebendig zu machen. Besonders am Anfang eines Districts wird das Gameplay absichtlich musikalisch nicht untermalt. In diesen Fällen war eine abwechslungsreiche und positionsbasierte Ambience sehr wichtig. Um dies zu erreichen, haben wir im Unity Editor grüne Boxen positioniert, die jeweils einen Ambience Loop beinhalten. Läuft der Spieler in die für ihn unsichtbare Box hinein, wird der Soundeffekt gestartet. Ein Lautstärke-Fade übernimmt das graduelle Eintreten der Ambience.

Der große Vorteil dieses Systems ist natürlich, dass es für uns dadurch recht einfach war, die Levels in verschiedene Zonen einzuteilen – so konnten wir zum Beispiel das Vogelgezwitscher oder den Wind in höheren Gebieten des Levels viel stärker betonen.

Die Umgebungsgeräusche selbst kamen hauptsächlich von unserer hauseigenen Library, bis auf einige Regenwald-Geräusche (irgendwann machen wir uns mal auf in den Dschungel, mit Recorder und Mikro!) und spezifische Sounds. Glücklicherweise haben wir schon einiges an eigenen Sounds in unserem Studio, die wir dann editieren und zu einem nahtlosen Loop machen können.

Besonders in den Slums war es nicht immer leicht, diese Boxen auf Anhieb sinnvoll zu legen. In diesem District kann Koru sehr schnell von einer Küste zu einem hohen Berg hinaufklettern. Die Ambience musste sich vor allem dort rasch anpassen, sich aber auch nicht andauernd ändern. Das würde dem Spieler negativ auffallen und die Immersion stören. Nur häufiges Testen direkt im Unity Editor hat geholfen, den besten Kompromiss zu finden.

Soundeffekte

Die Stimme von Koru musste besonders bei Kämpfen dynamisch wirken und sich mit den verschiedenen Hit-Sounds gut kombinieren lassen. In diesem Fall werden die Sprachsam-



Die drei »Districts« der Welt in The Last Tinker wurden mit blau, grün und rot nicht nur farblich voneinander abgegrenzt, sondern haben auch zahlreiche akustische Alleinstellungsmerkmale, die sich beispielsweise an den Eigenschaften der District-Bewohner orientieren.

ples (die übrigens alle Moritz Wagner, Game Designer bei Mimimi, eingebrüllt hat) per Zufallsprinzip abgespielt. Sehr viele Soundeffekte besitzen Variationen, die ebenfalls zufällig oder anhand einer Gewichtung abgefeuert werden.

Der Rest der Soundeffekte wurde entweder an Animationen gekoppelt und dann abgespielt (in Unity konnten wir die Animationen abspielen und Soundtrigger legen) oder in spezielle Audiocontainer gelegt, die auch diverse Parameter wie Lautstärke, Gewichtung und Ähnliches beinhalten.

Eine recht große Herausforderung war das Mischen und Einstellen der Sounds im 3D-Raum. Sicherzustellen, dass wichtige Sounds lauter und von weiter her zu hören sind oder dass die Positionierung der Sounds von jeder Kameraperspektive korrekt ist, hat länger gedauert als erwartet, teilweise auch, weil wir einfach viel mehr Sounds gemacht haben, als wir zunächst geplant hatten.

Die Stimmen in The Last Tinker

Alle Texte einzusprechen, war aus Zeit- und Budgetgründen einfach ausgeschlossen. The Last Tinker hat rund 2.000 Zeilen Text (und da wäre ja dann noch die Lokalisierung!) – da musste eine andere Lösung her.

Voiceover Sessions hausgemacht

An dieser Stelle möchte ich ein besonderes Lob an folgende Voiceover-Champions aussprechen: Lucas, Martin, Cem, Moritz von Mimimi und Ilenia, die ihre Wochenenden teilweise in der Sprachkabine verbracht haben, um ihren Charakteren Leben einzuhauchen.

Ein Blick auf eine unserer größten Inspirationen für das Game, dem N64-Rare-Klassiker »Banjo Kazooie«, half uns, einen Lösungsansatz für unser Sprachendilemma zu finden. Wir entschlossen uns, ein Sprachsystem aus kurzen Lauten aufzustellen, sodass bei jeder Konversation eine Art »Flow« entsteht – die Illusion, dass

die Charaktere in ihrer Sprache reden. Sie sollen daher konstant sprechen: Ein einziger Laut zu Beginn der Konversation, wie zum Beispiel bei »Zelda«-Titeln üblich, war uns für The Last Tinker etwas zu asketisch. Die lustigen Tierchen sollten schon einen Hang zum Brabbeln haben.

Dies funktionierte nicht auf Anhieb. Wir versuchten, längere »Sätze« einzusprechen, das wurde aber im Game sehr schnell zu repetitiv. Extrem kurze Samples waren auch nicht gut, weil sich die Stimmen zu zerhackt und mechanisch anfühlten.

Etwas ratlos schrieb ich eines Morgens Grant Kirkhope, dem ehemaligen Sound Designer von »Banjo Kazooie«, eine E-Mail. Der musste ja wissen, wie so was geht! Grant war so freundlich, uns zu antworten.

Sein Ratschlag lautete: einfach weiterprobieren, bis es gut klingt, und nicht aufgeben! Bei »Banjo Kazooie« war er nämlich in einer ähnlichen Situation (»There wasn't really any magic formula, it was just a lot of messing around until it sounded okay.«).

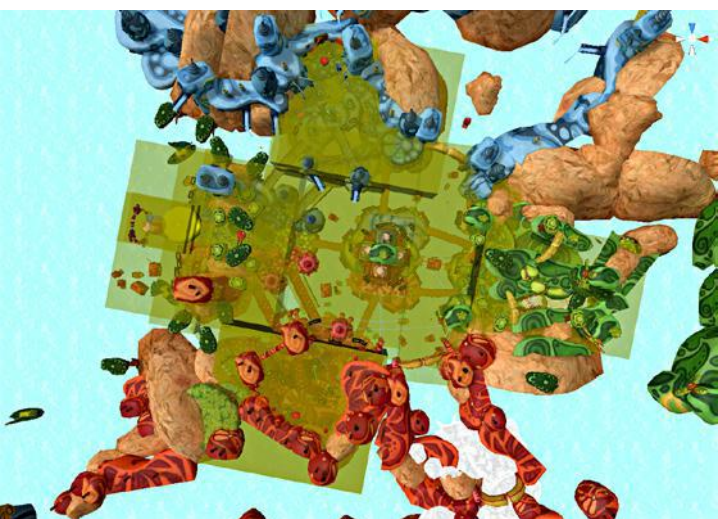
Der Durchbruch kam, genau wie Grant es beschrieben hatte, letztendlich durch einfaches Herumprobieren. Als die Länge der einzelnen Sounds und somit der Workflow entschieden war, machten wir uns auf, allen generischen Charakteren eine Stimme zu verleihen sowie die wichtigen und »uniquen« Figuren extra zu vertonen.

Wir wussten, dass wir für jeden der NPCs mindestens die für das Gameplay wichtigen Emotionen Wut (Rot), Furcht (Grün) und Trauer (Blau) sowie eine »normale« Stimme als VO brauchen. Bald kam aber noch ein »happy« Sound hinzu, weil dieser auch als Animation für jede Figur eingebaut wurde. Alles in allem hatten wir also fünf Emotionen pro NPC und circa zehn einzelne Sounds pro Emotion.

Es war natürlich recht viel Arbeit, alles einzusprechen, zu editieren und in das Spiel einzubinden, aber der Spaß, den wir bei



Für den Tinker-Sound kamen jede Menge Perkussion, Spielzeug und zwei Kokosnusshälften zum Einsatz.



Die grünen Boxen (linkes Bild) triggern jeweils einen Ambience-Soundeffekt, wenn sich der Spieler dorthin bewegt. Hier sehen wir beispielsweise den Market District von oben, mit verschiedenen Ambience-»Zonen«. Die Slums (rechtes Bild) stellten eine besondere Herausforderung für die Positionierung der Ambience-Sounds dar.

diesen VO-Sessions hatten, und das Resultat wiegen den Aufwand definitiv auf.

Charakterisierung durch Vokale

Besonders bei den verschiedenen Spirits, den wahrscheinlich wichtigsten Figuren des Spiels, haben wir versucht, sie einerseits sehr unterschiedlich zu gestalten, aber andererseits trotzdem einen gewissen Leitfaden beizubehalten.

Insgesamt haben wir fünf Spirits: Red, Green, Blue, Purple und Bleak. Es traf sich daher gut, jedem der Figuren einen Selbstlaut zuzuweisen – sozusagen einen Grundton, der den Charakter des Spirits ausmacht.

Die Stimme des Red Spirits, des überaus cholerischen Geists, ist daher hauptsächlich vom Vokal »A« geprägt. Der ängstliche Green Spirit bekam dagegen »I« als Grundton, was seine Natur sehr gut widerspiegelt. Blue Spirit, der ewig traurige, wird durch »U« sehr gut vermittelt. Im Falle des Purple Spirits und des Oberbösewichts, des Bleak Spirits, haben wir zusätzliche Sounds hinzugefügt: So ist der Purple Spirit durch ein mehrstimmiges, leicht verstörendes »O« auszumachen, während der Bleak Spirit ein heiseres, zischendes »E« spricht und zusammen mit tiefen Schmatzgeräuschen (für die unser Hund Link verantwortlich ist) ein bedrohendes Gesamtbild liefern soll.

Diese grobe Einteilung in Vokale war für uns eine sehr willkommene Stütze, da wir sofort eine Referenz hatten. Nach dem ersten Einspielen des Red Spirits wussten wir, dass dieses System funktionieren kann.

Diese Idee wird teilweise auch bei den generischen Charakteren weitergeführt. So haben zum Beispiel die kleinen Bewohner des Green Districts eine sehr piepsige Stimme, die immer »Mimimi« zu sagen scheint, also wird hier wieder das »I« des Green Spirits aufgegriffen. Und natürlich rein zufällig sagen sie auch den Namen eines gewissen bayerischen Entwicklerstudios ... das ist selbstverständlich vollkommen ungewollt!

Alles in allem sind wir glücklich mit dem Resultat. Die Abwesenheit von »echter« VO ist bestimmt nicht jedermanns Sache, andererseits haben wir versucht, aus unseren limitierten Ressourcen das Beste zu machen und den Charakteren auch klanglich Tiefe zu geben.

Intro und Outro

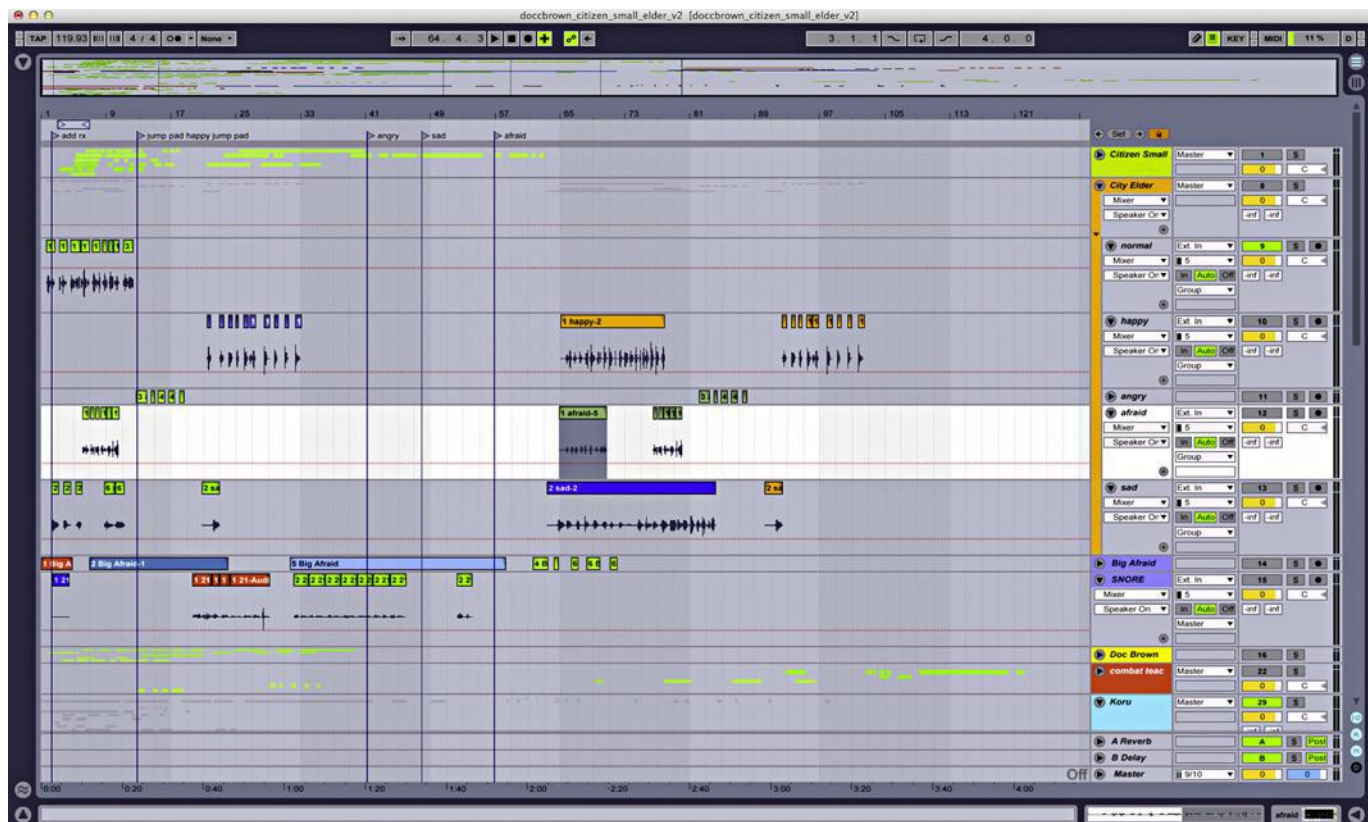
Zum Einsprechen der Erzählung im Intro und Outro haben wir uns für einen professionellen Sprecher (Peter Baker) entschieden. Peter traf den Ton wunderbar: Er brachte diese »Opa erzählt eine Geschichte«-Stimme inklusive ironischem Touch sehr gut rüber. Wir wollten besonders im Intro betonen, dass sich The Last Tinker mit seinen quirligen Charakteren selbst nicht ganz ernst nimmt. Peter hat das sehr gut umgesetzt.

Interaktive Musik und musikalische Minispiele

Beginnt beispielsweise ein Kampf, dann wird die Musik nicht unterbrochen. Uns gefiel die Idee des »Combat Loops« (eines speziell für



Im Studio ertönten ein paar Wochen lang lustige Klänge. Hier nahmen wir gerade Martin Hamburgers Stimme für den Admiral im »Blue District« auf.



Ein VO-Session-Projekt in Ableton Live. Links sieht man die fertig geschnittenen, kurzen Samples, rechts die Originalaufnahme.

den Kampf geschriebenen Tracks) nicht, daher suchten wir nach einem anderen Ansatz.

Combat Tracks

Dieser Ansatz sieht wie folgt aus: Bei Beginn des Kampfes wird das Musikstück mit einem Highpass-Filter in Echtzeit modifiziert, damit die tiefen Frequenzen verschwinden. Zugleich wird ein Perkussions-Loop dazugemischt. Dadurch wird die Musik sehr stark animiert, bleibt aber immer im Takt. Diese Tracks geben den Anschein, die Musik doppelt so schnell zu machen, wodurch wir die Kampfsequenzen betonen wollten, ohne den eigentlichen Flow der Lieder zu unterbrechen. Die Perkussions-Loops sind nicht in der gleichen Länge der Songs, müssen es aber auch nicht sein. Wichtig ist nur, dass sie von der Geschwindigkeit her richtig passen und eine Länge haben, die mit dem Loop des Songs zusammenarbeitet.

Interaktive Musik in der Windmühle

Es gibt einige Momente in *The Last Tinker: City of Colors*, die interaktive Musik verwenden und nicht nur auf die Thematisierung der einzelnen Districts bauen. Eines dieser Beispiele ist die Windmühle im Green District. Der Spieler muss sich zunächst zu den Füßen dieser alten, riesigen Mühle durchkämpfen, ehe er ins Innere des Gebäudes gelangt.

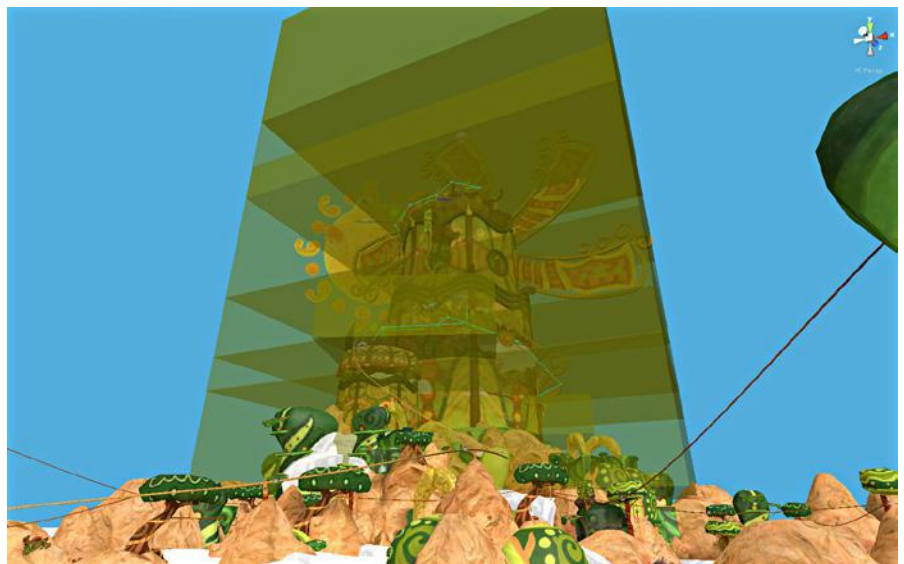
Hier war die Gelegenheit einfach zu perfekt, um den Level »nur« linear zu vertonen. Wir wollten eine gewisse bedrohliche Leere in der Musik beibehalten, aber deren Intensität schrittweise steigern, während der Spieler die Mühle hinaufklettert.

Also produzierte ich den Windmill Song schon mit unterschiedlichen Intensitätsstufen im Kopf. Es würden stufenweise Instrumente und Harmonien dazukommen. Dadurch hatte ich die Möglichkeit, zuerst nur einen mysteriösen Klangteppich aus einem scheinbar rhythmisch losen Hang Drum Groove zu kreieren, der in Wirklichkeit später als Bass in einem triumphalen Song seine eigentliche Rolle übernimmt.

Eine leicht bearbeitete Version des Ambience-Box-Tools lieferte das benötigte Werkzeug: Ich konnte somit genau entscheiden, wann welche musikalische Intensitäts-Ebene aktiviert wird.

Als erstes Element kamen wieder einmal die Hang Drums (die diesmal sehr tief gespielt wurden) ins Spiel. Der zweite Layer sind nur rhythmische Elemente im mittlerweile fest beschlossenen Tinker-Stil. In der Mitte der Windmühle kommen tiefe Streicher hinzu und die Situation wird bedrohlich – noch kein Anzeichen von Triumph oder Hoffnung. Nur die letzten beiden Klangebenen wälzen das Gesamtbild drastisch um. Das Hauptthema erklingt, der Spieler hat die Windmühle erklommen, die Gefahr ist gebannt.

Das Aufbrechen eines Musikstücks in diverse Einzelteile macht mir als Komponisten



Die Ambience-Boxen waren auch als Trigger für interaktive Musikelemente hilfreich.



Hier gilt es, die musikalischen Pilze in der richtigen Reihenfolge »abzuspielen«.

sehr viel Spaß und hat in dieser spezifischen Situation gut funktioniert.

Die Höhle und das Orchester-Minispiel

Gegen Ende des Spiels finden sich einige ganz besondere Levels, allen voran die musikalische Höhle, die das Gehör des Spielers herausfordert – und wie wir sehen werden, muss Koru auch dem heimischen Pilzorchester wieder auf die Sprünge helfen. Dieses gibt nämlich im Voraus schon mal eine Kostprobe seiner Fähigkeiten, und das Resultat ist ... sagen wir es so: Es hat sehr viel Spaß gemacht, all die falschen Töne einzuspielen! Es ist also dem Spieler überlassen, dem Orchester den alten Glanz der glorreichen Tage wiederzubringen. Davor muss er aber Notenblätter finden, die in der Höhle verteilt wurden.

Als kleinere Musikpuzzles haben wir Pilze in der Höhle platziert, die jeweils ein Fragment einer einfachen Melodie spielen. Der Spieler

muss die Notenfolge erkennen und die Pilze in der richtigen Reihenfolge treffen, damit die Melodie richtig gespielt wird.

Der zweite Teil der Höhle basiert auf sich im Rhythmus der Musik bewegenden Plattformen und kleinen Schildkröten, die der Spieler nutzen muss, um voranzukommen. Wir einigten uns auf eine bestimmte Schnelligkeit für die Musik und definierten Zonen, in denen man verschiedene Instrumente hören würde. Auch hier bauen sich kurze Grooves zu einem Gesamtbild auf. Alle Soundsamples werden gleichzeitig gestartet und präzise mit den Animationen und den Bewegungen der Plattformen abgespielt.

Als der Dirigent des Pilzorchesters, das in der Höhle wohnt, verzweifelt unseren Helden um Hilfe bittet, muss Koru selbst den Dirigentenstab in die Hand nehmen und versuchen, das nicht gerade sehr talentierte Ensemble wieder richtig gut spielen zu lassen.



Die Schildkröten in der Höhle bewegen sich zum Beat der Musik.



Das überaus talentierte Orchester der Pilzhöhle ...

Der Spieler wird also mit einem rhythmischen Minispiel im Stil von »Guitar Hero« konfrontiert. Der größte Unterschied ist jedoch, dass das Orchester standardmäßig schlecht spielt. Koru muss folglich so gut es geht den lausigen Pilzen helfen, durch den Song zu kommen.

Das Pilzorchester ist in drei Sektionen aufgeteilt, die auch als separate Audiolayer abgespielt werden. Diese drei Sektionen verlangen in bestimmten Zeitintervallen, dass sie der Spieler mit der korrekten Farbe (wieder einmal Rot, Grün oder Blau) bewirft. Schafft er das, so spielt die Sektion gut – ansonsten klingt es einfach nur schrecklich.

Technisch gesehen wurde der Orchestersong in sechs verschiedene Teile aufgebrochen: drei gut eingespielte Versionen für Streicher, Perkussion und Bläser, und drei sehr schlecht eingespielte. Die Menge an krummen Tönen und falschen Rhythmen ist einfach herzerreißend, wenn alle drei Sektionen falsch spielen. Damit haben wir auch den Vorteil, dass es verschiedene Kombinationen und Stufen von gutem bzw. schlechtem Spielen gibt, weil die drei Sektionen auf separaten Spuren laufen.

Die Höhle war einer der Höhepunkte in Sachen interaktiver Musik. Besonders das Orchester-Minispiel wurde mehrfach angepasst, damit es sich gut anfühlte. Knackpunkt war in diesem Fall wirklich, das Orchester standardmäßig schlecht spielen zu lassen, was auch die Komik der ganzen Situation unterstreicht.

Fazit

The Last Tinker wurde erst vor kurzem veröffentlicht, aber schon in den ersten Reviews wird der Sound in den allermeisten Fällen nicht nur angesprochen, sondern auch sehr gelobt – besonders die Musik, worüber ich natürlich sehr glücklich bin.

Dieses Projekt war für uns alle eine Herzensangelegenheit, in die wir all unser Können, unsere Zeit und unseren guten Willen investiert haben. Viele hielten es für unmöglich, ein Spiel eines solchen Umfangs mit einem Team von nur elf Leuten zu stemmen.

Für mich persönlich war Last Tinker das bisher umfangreichste Projekt. Ein Blick in das Audio-Verzeichnis zeigt über 70 Minuten Musik und knapp 2.700 Soundeffekte – das war ein ziemlicher Marathon.

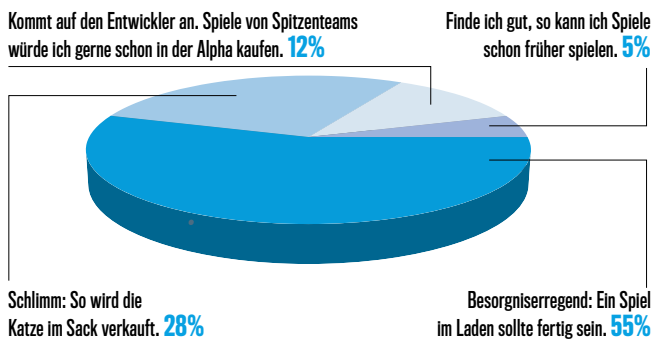
Ich kann nicht in Worte fassen, wie viel ich durch dieses Projekt gelernt habe. Ideen für mehr Musik, Verbesserungen an vielen Soundeffekten, noch mehr musikalische Minispiele und vieles mehr sind natürlich wie in jeder Produktion in die »Nächstes Mal«-Schublade gewandert, aber wer weiß ... das könnte ja alles in Last Tinker 2 umgesetzt werden.

Filippo Beck Peccoz

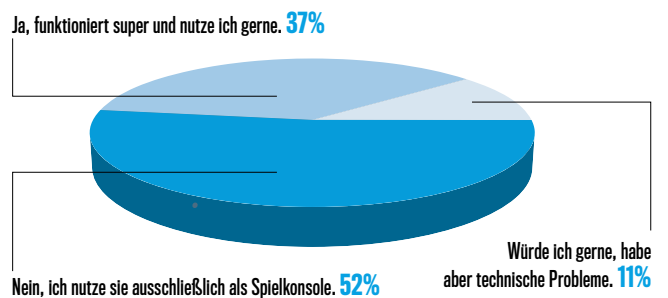
DATAFLOW

GameStar und GamePro befragen regelmäßig Heftkäufer und Website-Besucher zu aktuellen Games-Themen. Eine kleine Auswahl finden Sie auf dieser Seite.

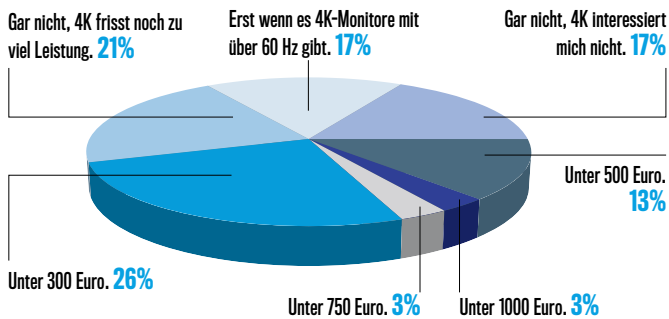
Early-Access-Spiele zum Vollpreis - was halten Sie von dieser Entwicklung im Markt?



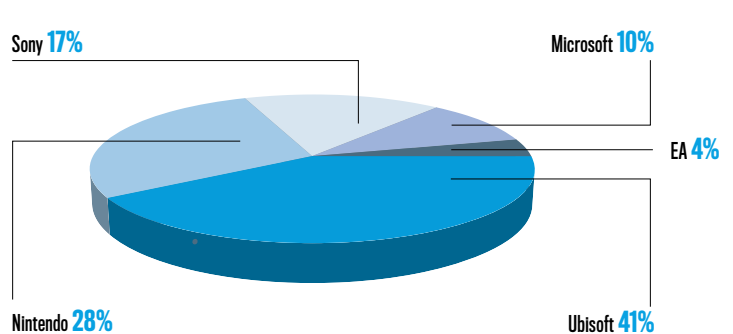
Nutzen Sie die Xbox One als Multimedia-Center im Wohnzimmer?



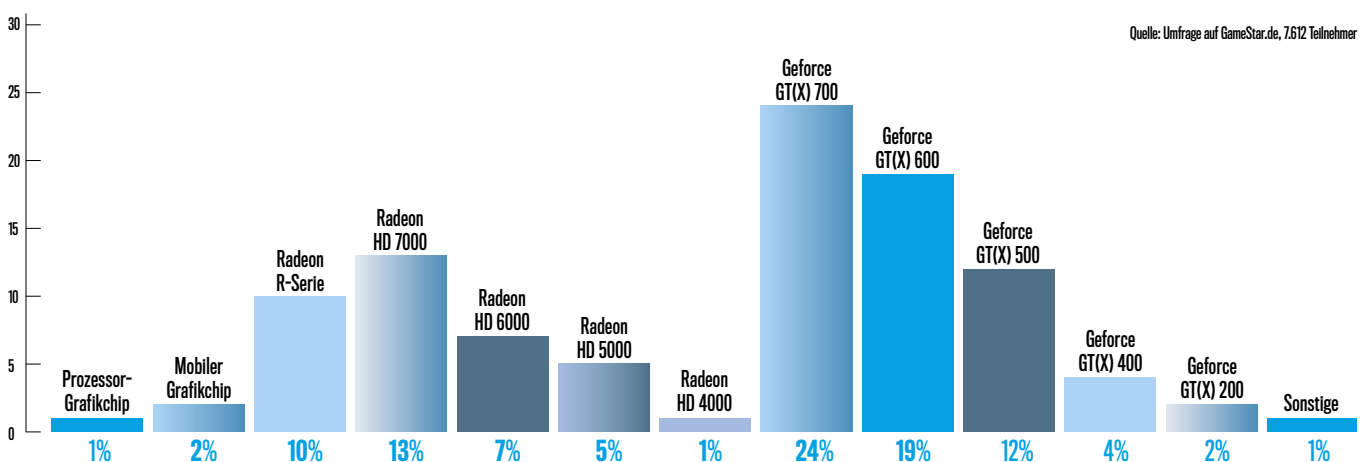
4K-Monitore werden immer günstiger. Ab welchem Preispunkt schlagen Sie zu?



Wer hatte die beste E3-Pressekonferenz?



Welche Grafikkarte besitzen Sie?



WWW.makinggames.de

Online-Register mit Suchfunktion. Zu jedem Key Player finden Sie zusätzlich ein detailliertes Firmenportrait.

Helfen Sie uns!

Sie haben einen Fehler entdeckt oder vermissen Ihre Firma? Dann schreiben Sie uns:
info@makinggames.de, Betreff: Firmenregister

DIE BRANCHE IM ÜBERBLICK FIRMENREGISTER

Vergessen Sie die Auskunft! Hier finden Sie sämtliche uns bekannten Publisher, Entwickler, Dienstleister, Ausbilder und Verbände der deutschsprachigen Spiele-Industrie - von 3d-io bis Z-Software.

DEVELOPMENT					
Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail	
3d-io	Rheingaustraße 94	65203 Wiesbaden	0611 / 5829602	info@3d-io.com	
5d lab	Georges-Köhler-Allee 79	79110 Freiburg	0761 / 2038043	info@5dlab.com	
Adrastea	Bielefelder Str. 5	90425 Nürnberg	0700 / 42300000	info@adrastea.com	
ANIMA Entertainment	Konsul-Smidt-Straße 84	28217 Bremen	0421 / 5784220	info@anima-entertainment.de	
APP33	Fuggerstraße 12	84419 Schwindegg	032 / 221476589	info@app33.de	
Avaloop IT Solutions	Hütteldorferstraße 253	A-1140 Wien		office@avaloop.com	
B-Alive	Biberacher Straße 38	88444 Ummendorf	07351 / 441307	admin@b-alive.de	
BigBlackBlock Gamestudio	Hohenbudberger Straße 133	47229 Duisburg	02065 / 9609747	info@big-black-block.com	
BIGITEC	Theaterstraße 22	53111 Bonn	0228 / 42976585	info@bigitec.com	
BIGPOINT	Drehbahn 47-48	20354 Hamburg	040 / 8814130	info@bigpoint.net	
binary madness	Bundesallee 55	10715 Berlin	030 / 25093746	info@binarymadness.de	
b-interactive	Lohbachstraße 12	58239 Schwerte	02304 / 3382391	info@binteractive.com	
Bit Barons c/o Wailua Games	Schellingstraße 35	80799 München	089 / 41617744	info@bitbarons.com	
Bitfield	Rigaer Straße 56	10247 Berlin	030 / 40045181	kontakt@bitfield.de	
Black Forest Games	Innovationszentrum, Hauptstraße 1a	77652 Offenburg	0781 / 31057910	info@bfgames.biz	
Blue Byte	Adlerstraße 74	40211 Düsseldorf	0211 / 338000	info@bluebyte.de	
Blue Byte Mainz	Römerpassage 1	55116 Mainz	06131 / 554470	info@related-designs.de	
BlurredVision	Glashütter Weg 80	22844 Norderstedt	040 / 53532683	info@blurredvision.de	
Bongfish	Schönaugasse 8a/2	A-8010 Graz	+43 / 676 / 7822203	info@bongfish.com	
Brightside Games	Marie-Elisabeth-Lüders-Straße 1	10625 Berlin	0177 / 3354447		
Broken Rules Interactive Media	Johann Strauss Gasse 20/8	A-1040 Wien		hello@brokenrul.es	
Bronx Studios Entertainment	Olof-Palme-Straße 13	60439 Frankfurt	069 / 870050660	info@bronxstudios.com	
BSS web consulting	Savignystraße 74	45147 Essen	0201 / 5200044	info@bss-consulting.de	
Buntware Games	Colmarerstrasse 9	79106 Freiburg	0761 / 3891598	contact@buntware.com	
BURNS Games	Danziger Straße 118	10405 Berlin	030 / 40301763	info@burns-games.com	
Bright Future	Im Zollhafen 15-17	50678 Köln	0221 / 78982171	info@brightfuture.de	
Bumblebee.	Gerrshheimer Straße 7	51674 Wiehl	02261 / 4795001	tassilo@bumblebee-games.com	
ByteRockers	Schivelbeiner Straße 1	10439 Berlin	030 / 232573390	info@byterockers.de	
Bytro Labs	Rödingsmarkt 29	20459 Hamburg		info@bytro.com	
Caipirinha Games	Kiefernweg 1a	24161 Altenholz	0431 / 324540	info@caipirinhagames.de	
Candygun Games	Hamburger Strasse 132	22083 Hamburg	040 / 20947878		
cerasus.media	Rosenthaler Straße 51	10178 Berlin	030 / 802089990	info@cerasus.de	
Chasing Carrots	Hornbergstraße 94	70188 Stuttgart	0711 / 64519353	office@chasing-carrots.com	
Chimera Entertainment	Sendinger-Tor-Platz 6	80336 München	089 / 210205790	info@chimera-entertainment.com	
CipSoft	Gabelsbergerstraße 11	93047 Regensburg	0941 / 6308280	kontakt@cipsoft.com	
Claw Design	Ludwigstraße 8	72474 Winterlingen	07434 / 3931	info@clawdesign.com	
Cliffhanger Productions	Seidengasse 32/NB/7	A-1070 Wien		office@cliffhangerproductions.com	
ClockStone STUDIO	Claudiastraße 14	6020 Innsbruck	+43 / 512 / 890377	info@clockstone.com	
Cooe	Europaallee 7-9	67657 Kaiserslautern	0900 / 1002273		
Crafty Studios	Lichtenwörthergasse 112a	A-7033 Pötsching		info@craftystudios.com	
createle	Mörkestraße 28/2	71636 Ludwigsburg	07141 / 2399563	contact@createle.de	
CreTeam Software	Sünderup 6	24943 Flensburg	0461 / 64808		
Crenetic	Kreuzstraße 3	45468 Mülheim an der Ruhr	0208 / 4442142	info@crenetic.de	
Crytek	Grüneburgweg 16-18	60322 Frankfurt am Main	069 / 21977660	info@crytek.de	
Cyber Arena Entertainment Group	Sebastian Kneippgasse 30	2380 Perchtoldsdorf		info@c-arena.com	
Dachshund Games	Turmstraße 48	10551 Berlin		team@dachshund-games.com	
Daedalic Entertainment	Papenreye 51	22453 Hamburg	040 / 43261270	info@daedalic.de	
Decane	Hallerstraße 5e	20146 Hamburg	040 / 42910947	info@decane.net	
Deck13 Interactive	Gutleutstraße 82	60329 Frankfurt am Main	069 / 71671660	info@deck13.com	
Deep Silver Fishlabs	Großer Burstah 50-52	20457 Hamburg	040 / 8888000	business@fishlabs.net	
Destrax Entertainment	Merkurstraße 10	33739 Bielefeld		info@destrax.de	
Digitalmindsoft	Wichernstraße 10	89073 Ulm		info@digitalmindsoft.eu	
Doublesmith	Ranzonigasse 3/3	A-3100 St. Pölten	+43 / 699 / 19447880	mail@doublesmith.com	
dreiNET	Bergedorfer Straße 96	12623 Berlin		philip@dreinet.org	
DynamicHead	Hauptstraße 6	55283 Nierstein Schwabsburg	061 / 33572494	timo.weisse@dynamichead.com	
elusion	Paul-Lincke-Ufer 39	10999 Berlin		contact@elusion.com	
Egosoft	Heidestraße 4	52146 Würselen	02405 / 4239970	egoinfo@egosoft.com	
Enter-Brain-Ment	Littenstraße 106/107	10179 Berlin		interesse@enter-brain-ment.com	
Envision Entertainment	Binger Straße 38	55218 Ingelheim		info@envision-entertainment.de	
eos interactive	Carl-Zeller Straße 16	75015 Berlin	07252 / 957757	info@eosinteractive.de	
exDream	Hildesheimer Straße 5	30169 Hannover	0511 / 37077675	info@exdream.com	
exozet	Platz der Luftbrücke 4-6	12101 Berlin	030 / 24656000	games@exozet.com	
extra toxic	Hanauer Landstraße 196a	60314 Frankfurt am Main	069 / 48440040	info@extratoxic.com	
eyefactive	Feldstraße 128	22880 Wedel	04103 / 903800	info@eyefactive.com	
FAKT Software	Dufourstraße 28	04107 Leipzig	0341 / 14990564	info@fakt-software.de	
Fameside Entertainment	Storkower Straße 158	10407 Berlin	030 / 40577053	info@fameside.com	

DEVELOPMENT					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Fancy Bytes		Niederstraße 5	26931 Huntorf	02151 / 3259912	biz@fancy-bytes.de
FDG Entertainment		Agnes-Pockels-Bogen 1	80992 München	089 / 18941370	info@fdgertainment.com
flaregames		Kaiserstraße 146-148	76133 Karlsruhe	0721 / 1831548	info@flaregames.com
Freudenreich & Tollknaepper		Paracelsusstraße 57	42283 Wuppertal	0202 / 2721048	mail@german-railroads.com
Funatics Software	key player	Essener Straße 66	46047 Oberhausen	0208 / 3066670	info@funatics.de
funworld		Photo Play Straße 1	A-4860 Lenzing	+43 / 7662 / 67850	info@photoplay.com
Fusionsphere Systems		Rosenau 8	86551 Aichach	0700 / 38746600	info@fusionsphere.com
GameArt Studio		Bismarckstraße 107	10625 Berlin		kontakt@GameArtStudio.de
GameDuell		Taubenstraße 24-25	10117 Berlin	030 / 288768210	impressum@gameduell.de
Gameforge		Albert-Nestler-Straße 8	76131 Karlsruhe	0721 / 3548080	info@gameforge.de
gameus		Kufsteinerstraße 40	83064 Raubling	08035 / 6689	info@gameus.de
Gaming Minds Studios		Georgstraße 6	33330 Gütersloh		contact@gamingmindsstudios.com
Gbanga		Braverstraße 31	8004 Zürich	+41 / 43 / 5366701	matthias@gbanga.com
GIANTS Software GmbH		Rütistrasse 18	8952 Zürich-Schlieren	+41 / 44 / 5082176	mail_us@giants-software.com
Goal Games		Feurigstraße 54	10827 Berlin		info@goal-games.de
goodbeans		Winsstraße 62	10405 Berlin	030 / 802087400	jobs@goodbeans.com
Goodgame Studios	key player	Theodorstraße 42-90, Haus 9	22761 Hamburg	040 / 2198800	pr@goodgamestudios.com
Greentube		Mariahilfer Straße 47/1/102	A-1060 Wien	+43 / 1 / 4945056	office@greentube.com
HandyGames		i Park Klingholz 13	97232 Giebelstadt	09334 / 97570	info@handy-games.com
iLogos Europe		Pezolddamm 98a	22175 Hamburg	0160 / 7002927	alexander@ilogos.biz
Independent Arts Software		Münsterstraße 5 / HAMTEC Haus 2b	59065 Hamm	02381 / 488818	info@independent-arts-software.de
InnoGames	key player	Friesenstraße 13	20097 Hamburg	040 / 78893350	info@innogames.de
Instance Four		Friederikastraße 65	44789 Bochum	0234 / 89038971	info@instancefour.com
Island Games		Vahrenwalder Straße 7	30165 Hannover	0511 / 35319898	kontakt@islandgames.de
Jo-Mei		Monumentenstraße 35, Aufg. C	10829 Berlin	030 / 81494080	info@jo-mei.com
K1010 Media		Erkelenzdamm 59-61	10999 Berlin	030 / 57709936	info@k1010.de
keen games		Dreieichstraße 59	60594 Frankfurt am Main	069 / 66058838	info@keengames.com
King Art	key player	Martinstraße 47-49	28195 Bremen	0421 / 322760	contact@kingart.de
KORION Simulation		Mömpelgardstraße 16	71640 Ludwigsburg	07141 / 3898231	info@korion.de
kr3m.media	key player	Kaiserstraße 158	76133 Karlsruhe	0721 / 18395918	
Kritzelkratz 3000		Dieselstraße 2-6	97082 Würzburg	0931 / 3536091	info@kritzelkratz.de
kunst-stoff		Kottbusser Damm 73	10967 Berlin	030 / 2345178	info@kunst-stoff.de
Limbic Entertainment		Heinrich-Hertz-Straße 21	63225 Langen	06103 / 5019780	info@limbic-entertainment.de
MADE		Alter Schlachthof 39	76131 Karlsruhe	0721 / 78809460	
MaDe Games GbR Neetix - Wycislik		Hochstraße 34	41334 Nettetal		info@madegames.com
marune		Wallbergstraße 16	82008 Unterhaching	089 / 66500749	info@marune.de
McPeppergames		Nikolaus-Molitor-Straße 11a	97702 Münnerstadt	09733 / 781006	info@mcpeppergames.com
Media Seasons		Liviastraße 6	4105 Leipzig	0341 / 2156752	contact@mediaseasons.com
Mex IT		Mallinckrodtstraße 320	44147 Dortmund	0231 / 58680722	mex@mex-solutions.com
Mimimi Productions		Berg-am-Laim-Straße 64	81673 München	089 / 809115070	info@mimimi-productions.de
Mind Elevator Games		Prenzlauer Promenade 190	13189 Berlin	030 / 91744394	info@mindelevatorgames.com
mobivention		Gottfried-Hagen-Straße 24	51105 Köln	0221 / 6778110	info@mobivention.com
Moonbyte Games		Wormser Landstraße 17	67346 Speyer	06232 / 8774482	info@moonbytegames.com
NeoBird		Südwestpark 37-41	90449 Nürnberg	0911 / 3092097	info@neobird.de
netmin games		Philipp-Reis-Straße 6	55129 Mainz	06131 / 507896	schreiber@netmin.de
Neutron Games		Gustav-Meyer-Allee 25	13355 Berlin	030 / 46307222	kontakt@neutron-games.de
northworks		Stahltwiete 23	22761 Hamburg	040 / 78107075	info@northworks.de
Noumena Studios		Lützowstraße 33	10785 Berlin	030 / 53148825	info@noumenastudios.com
NovaTrix / Animedo		Schlüsselwiesen 19	70186 Stuttgart	0711 / 3102770	info@animedo.de
Nurogames GmbH		Schaafenstraße 25	50676 Köln	0221 / 3988040	info@nurogames.com
OG-Soft Productions		Zirbelstraße 48	86154 Augsburg		info@og-soft.de
online4ever Limited		Manitiusstraße 6	1067 Dresden	0351 / 2063875	info@online4ever.biz
OFM - OnlineFussballManager	key player	Eupener Straße 60	50933 Köln	0221 / 88892920	office@onlinefussballmanager.de
Outline Development		Hauptstraße 122	57074 Siegen	0271 / 3131331	webmaster@outline-development.de
Pappuga		Fürther Straße 122	90429 Nürnberg	0911 / 32378310	info@pappuga.com
Phobator		Wiesentalstraße 5	90419 Nürnberg	0911 / 9389292	wendt@phobator.de
Piece of Pie Studios		Sandlinger-Tor-Platz 6	80336 München	089 / 210205780	info@pieceofpiestudios.com
Piranha Bytes		Ruhrallee 63	45138 Essen	0201 / 806720	info.nospam@piranha-bytes.com
pixelamer.net		Clausewitzstraße 6	10629 Berlin	030 / 34347690	welcome@pixelamer.net
Pixon		Jaffestraße 10	21109 Hamburg	040 / 33399430	welcome@pixon.de
Playnik		Bielefelder Str. 5	90425 Nürnberg	0700 / 42300000	info@adrastea.com
playzo		Heinrich-Herz-Straße 6	64295 Darmstadt	06151 / 6673700	info@playzo.biz
plazz entertainment		Augustmauer 1	99084 Erfurt	0361 / 2169460	info@plazz-entertainment.com
Pop Rocket Games		Gasstraße 10	22761 Hamburg	040 / 822449040	
Procontis		Am Bauhof 18	64807 Dieburg	06071 / 430111	sales@procontis.de
Promotion Software		Karlstraße 3	72072 Tübingen	07071 / 91670	mail@promotion-software.de
qforge		Südwestpark 37 - 41	90449 Nürnberg	0911 / 3092097	office@qforge.de
Quadriga Games		August-Bebel-Straße 27	14482 Potsdam	0331 / 23189660	info@quadrigagames.com
Rabenaue		Ebelingstraße 17	10249 Berlin	030 / 21092345	the_contact@rabenaue.com
Rat King Entertainment		Robert-Franz-Ring 3	Halle	0345 / 6141741	mail@ratking.de
Ravensburger Digital		Kaflerstraße 8	81241 München	089 / 12228180	tellme@ravensburger-digital.com
Reality Twist		Berg-Am-Laim-Straße 64	81673 München	089 / 380129500	mail@reality-twist.com
Realmforge Studios		Hofer Straße 15	81737 München	089 / 55069134	info@realmforgestudios.com
RedMoon Studios		Dieselstrasse 16	86368 Gersthofen	0821 / 45598000	info@redmoonstudios.de
REDOX Game Labs		Jakob-Haringer Straße 5a	A-5020 Salzburg	+43 / 662 / 454555444	info@redox-labs.com
remote control productions	key player	Sandlinger-Tor-Platz 6	80336 München	089 / 21020570	info@r-control.de
RockABYTE		Schaafenstraße 25	50676 Köln	0221 / 80147950	info@rockabyte.com
SAFKAS Development		Schulze-Delitzsch-Straße 39	04315 Leipzig		mail@sarkas.de
Sandbox	key player	Bornholmer Strasse 71	10439 Berlin	030 / 40045181	info@sandbox-interactive.com
Scorpius Forge		Nelly-Sachs-Straße 119	89134 Blaustein	0731 / 14410099	info@scorpius-forge.de
Serious Games Solutions		August-Bebel-Straße 27	14482 Potsdam	0331 / 23189680	mail@serious-games-solutions.com
Sharkbombs Studios		Durlacher Allee 53	76131 Karlsruhe	0160 / 92502633	contact@sharkbombs.de
SilentFuture		Triebelsheide 37b	42111 Wuppertal	0202 / 3935366	info@silentfuture.de
Silver Seed Games		Sandtorstraße 23	39106 Magdeburg	0391 / 5448619372	info@silverseedgames.com
Simlity		Moritz-von-Rohr-Straße 1a	07745 Jena	03641/3279684	info@simlity.com
Skunk Brothers		Dieselstraße 12	61191 Rosbach		hello@skunkbrothers.de
SlipShift		Friedrichstraße 122	10117 Berlin	030 / 27581581	info@slipshift.net
Snowstep Development		Heinrich-Holtzschneider-Weg 76	40489 Düsseldorf	0201 / 17848176	business@snowstep.com

DEVELOPMENT				
Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Socialspiel	Schmalzhofgasse 26 / Top 5-6	A-1060 Wien		office@socialspiel.com
Solimedia Productions	Bahnofstr. 4a	99084 Erfurt	0361 / 658550	hatmut.niemer@solimedia.de
Sparrow Games	Wegenerstraße 1	13088 Berlin	030 / 50915356	mail@sparrowgames.de
SpinBottle Games	Weidkamp 3	45355 Essen	0201 / 1789614	info@spinbottlegames.com
Spirit	Friedrichstraße 15	70174 Stuttgart	0711 / 96685510	weyrich@go4spirit.com
Spitscreen Studios	Neuer Kamp 30	20357 Hamburg		info@spitscreenstudios.com
Sportradar Virtual Gaming	Konsul-Smidt-Straße 8f	28217 Bremen	0421 / 6195640	info@attainment.de
Springwald Software	Alter Eistreff 36	44789 Bochum	0700 / 777464925	info@springwald.de
Sproing Interactive Media	Fernkorngasse 10	A-1100 Wien	+43 / 1 / 60430280	office@sproing.com
Studio Fizin	Hofstraße 20	71636 Ludwigsburg	07141 / 9923866	hello@studio-fizin.de
Sunlight Games	Im Klapperhof 7-23	50670 Köln	0221 / 16823466	kontakt@sunlight-games.com
symblCrowd	In der Schleh 27	5224 Eschweiler	0152 / 55878044	apps@symblcrowd.de
Synetic	Auf der Haar 47	33332 Gütersloh	05241 / 4009900	synetic@synetic.de
Team Vienna Games	Kaiserstrasse 84/1/6	A-1070 Wien		office@teamviennagames.com
Travian Games	Wilhelm-Wagenfeld-Straße 22	80807 München	089 / 3249150	mail@traviangames.com
unikat media	Eckdrift 10	19061 Schwerin	0385 / 64105688	info@unikatmedia.de
upjers	Hafenstraße 13	96052 Bamberg	0951 / 510908100	mail@upjers.com
VisionaryX	Friedrichstraße 12	71011 Schönaich	07031 / 2850 / 414	contact@visionaryx.com
V-Play	Kolonitzgasse 9/11-14	A-1030 Wien		team@v-play.net
Vulcando Games	Dasinger Straße 2	86165 Augsburg	0821 / 4550834	info@vulcando-games.de
Wolpertinger Games	Sendlinger-Tor-Platz 6	80336 München	089 / 210205700	contact@wolpertingergames.com
wooga	Saarbrücker Straße 38	10405 Berlin	030 / 32505294	info@wooga.com
Xendex	Mariahilfer Strasse 176/6	A-1150 Wien	+43 / 1 / 89080090	office@xendex.com
XYRALITY	Friedensallee 290	22763 Hamburg	0 / 4035730010	info@xyrality.com
YAGER Development	Pfuelstraße 5	10997 Berlin	030 / 69597660	contact@yager.de
zeitland media & games	Königsallee 43	71638 Ludwigsburg	07141 / 5050565	contact@zeitland.com
Zeroscale	Schwedter Straße 9a	10119 Berlin	030 / 405057929	info@zeroscale.com
Zone 2 Connect	Fürstenwall 70	40219 Düsseldorf	0211 / 5422400	info@zone2.de
Z-Software	Lindemannstrasse 81	44137 Dortmund	0231 / 33015031	contact@z-software.net

FREELANCER DEVELOPMENT				
Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Caspar Siebel	Theresienhöhe 1	80339 München	017661615291	mail@casparsiebel.com
Eric Jannot (Game Design, Game Writing)	Sonnenallee 70	12045 Berlin	0178 / 5223697	
Filippo Beck Peccoz	Franz-Joseph-Straße 19	80801 München	089 / 80033204	mail@fbpsound.com
Jochen Hamma (Producing)	Wehinger Weg 9	78583 Boettingen	07429 / 910793	jochen.hammer@fantastic-realms.com
Jörg Beilschmidt (Game Design)	Berliner Ring 20	21406 Melbeck	0172 / 4311243	j.beilschmidt@creatown.de
Kim Lange (Full Service Agency)	Landwehr 192	46045 Oberhausen	0178 / 2375418	lange@eins2design.de
Marco Sowa (Producing, Development)	Postfach 420566	50899 Köln	0221 / 16916844	mail@marcosowa.de
Mariusz Ginel (Mediengestaltung, SED)	Stettiner Straße 31	53840 Troisdorf	02241/9321556	mginel@mg-mediengestaltung.de
Martin Linnartz (3D Artist, Game Developer)	Bleichstraße 12	41460 Neuss	02131 / 3133306	
Martin Nerurkar (Game Design)	Durlacher Allee 53	76131 Karlsruhe		mnerurkar@sharkbombs.com
Nico Nowarra (Storytelling)	Friedbergstraße 9	14057 Berlin	030 / 31801899	nno@menintext.com
Niels Bauer Games (Game Design)	Stickingenstraße 37	79117 Freiburg	0761 / 66947	contact@nielsbauergames.com
Thomas Obermaier (Programmierung)	Kapellenstraße 1	86869 Guttenberg	0170 / 3815144	contact@polyforming.com
Wolfgang Walk (Full Scale Producing)	Karl-Leopold-Straße 6	76229 Karlsruhe	0721 / 48399963	wolfgang.walk@t-online.de

PUBLISHING/DISTRIBUTION				
Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
505 Games	Brunnfeld 2-6	93133 Burglengenfeld	09471 / 3088792	
Activision Blizzard Deutschland	Fraunhoferstraße 7	85737 Ismaning	089 / 9998170	info@activision.de
ad2games	Rosenstraße 17	10178 Berlin	030 / 240888221	info@ad2games.com
Aeria Games Europe	Schlesische Straße 27, Aufgang C	10997 Berlin		info@aeriagames.com
Aerosoft	Lindberghring 12	33142 Büren	02955 / 760310	info@aerosoft.de
Application Systems Heidelberg	Pleikartsförsterhof 4/1	69124 Heidelberg	06221 / 300002	kontakt@application-systems.de
AppLift	Rosentraße 17	10178 Berlin	030 / 240888208	info@applift.com
astragon Software	Limitenstrasse 64-78	41236 Mönchengladbach	02166 / 146450	info@astragon.de
BHV	Novesiastraße 60	41564 Kaarst	02131 / 76501	info@bhv.de
Bigpoint	Drehbahn 47-48	20354 Hamburg	040 / 8814130	info@bigpoint.net
bitComposer Entertainment	Mergenthalerallee 79-81	65760 Eschborn	06196 / 7793810	info@bit-composer.com
Capcom - CEG Interactive Entertainment	Barmbeker Straße 4b	22303 Hamburg	040 / 6965620	info@capcomeuro.de
Covus Crobo	Schwedter Straße 263	10119 Berlin		info@crobo.com
Crimson Cow	Tarpen 40, Haus 5	22419 Hamburg	040 / 59355237	info@crimsoncow.de
Crytek	Grüneburgweg 16-18	60322 Frankfurt am Main	069 / 21977660	gregory.wintgens@crytek.com
Daedalic Entertainment	Papenreye 53-Workport Unit 2	22453 Hamburg	040 / 43261270	info@daedalic.de
Deutsche Telekom	T-Online-Allee 1	64295 Darmstadt	06151 / 6800	presse@gamesload.de
Disney Interactive Studios	Kronstadter Straße 9	81677 München	089 / 993400	info@disney.com
dreamfab	Domplatz 3	93047 Regensburg	0941 / 569591520	
dtp entertainment	Goldbekplatz 3-5	22303 Hamburg	040 / 6699100	info@dtp-entertainment.com
dtp young entertainment	Goldbekplatz 3-5	22303 Hamburg	040 / 6699100	info@dtp-young.com
Electronic Arts	Im Zollhafen 15-17	50678 Köln	0221 / 975820	
European Games Group	Wiener Platz 7	81667 München		contact@gamesgroup.eu
EuroVideo Bildprogramm	Oskar-Messer-Straße 15	85737 Ismaning	089 / 9624440	info@eurovideo.de
flashpoint	Valvo Park - Haus 5c / Tarpen 40	22419 Hamburg	040 / 710060	info@flashpoint.de
GameDuell	Taubenstraße 24-25	10117 Berlin	030 / 288768211	impressum@gameduell.de
Gameforge	Albert-Nestler-Straße 8	76131 Karlsruhe	0721 / 3548080	info@gameforge.de
GameGenetics	Alte Jakobstraße 85-86	10179 Berlin	030 / 922512769	info@gamegenetics.de
Gameloft	Karl-Liebknecht-Straße 5	10118 Berlin	030 / 20188429	gregory.wintgens@gameloft.com
GameTwist	Mariahilfer Straße 47/1/102	1060 Wien	+43 / 1494 / 5056	3rdparty@gametwist.com
gamigo	Behringstraße 16b	22765 Hamburg	040 / 4118850	info@gamigo.com
Halycon Media	An der Autobahn 24	23858 Reinfeld	04533 / 61090	info@mail@halycon.de
Headup Games	Nordstraße 102	52353 Düren	02421 / 4868700	info@headupgames.com
HitFox Group	Rosenstraße 17	10178 Berlin	030 / 240888200	info@hitfoxgroup.com
Infernum Productions	Unlandstraße 175	10719 Berlin	030 / 91201090	contact@infernum.com
InnoGames	Friesenstraße 13	20097 Hamburg	040 / 78893350	info@innogames.de
Intenium	Neuer Pferdemarkt 1	20359 Hamburg	040 / 25494296	info@intenium.de
Kalypso Media Group	Wilhelm-Leuschner-Straße 11-13	67547 Worms	06241 / 5061900	info@kalypsomedia.com

PUBLISHING/DISTRIBUTION					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Koch Media		Lochamer Straße 9	82152 Planegg/München	089 / 242450	office@kochmedia.de
Konami		Berner Straße 103-105	60437 Frankfurt am Main	069 / 9855730	support@konami.de
McGame.com		Fürstenwall 69	40217 Düsseldorf	0211 / 63550100	support@mcgame.com
Microsoft		Konrad-Zuse-Straße 1	85716 Unterschleißheim	089 / 31760	kunden@microsoft.com
Millform		Ringstraße 110	CH-4106 Therwil	+41 / 61 / 7220712	werner.sala@millform.ch
Morphicon Limited		Lindwurmstraße 88	80337 München	089 / 45235460	info@morphicon.de
MMM MultiMediaManufaktur		Ludwigstraße 33-37	60327 Frankfurt am Main	069 / 974600	info@multimediamanufaktur.com
Namco Bandai Partners		Carl-Benz-Straße 21	60386 Frankfurt am Main	069 / 24449100	de.info@namcobandaipartners.com
NBG		Brunnfeld 2-6	93133 Burglengenfeld	09471 / 70170	info@nbg-online.de
Nintendo		Nintendo Center	63760 Grossostheim	06026 / 9500	info@nintendo.de
Novitas		Frankfurter Straße 2	38122 Braunschweig	02574 / 9270	kontakt@greenpepper.de
Phenomenia		Friedrich-Ebert-Straße 14	44866 Bochum	02327 / 9970	info@phenomenia.com
Playa Games		Alstertor 9	20095 Hamburg	040 / 76793220	info@playa-games.com
Players Rock Entertainment		Goseriede 4	30159 Hannover	0511 / 3539940	contact@playersrock.com
ProStebenSat.1 Games		Medienallee 6	85774 Unterföhring	089 / 950710	info@sevengames.de
qforge		Südwestpark 37 - 41	90449 Nürnberg	0911 / 3092097	info@qforge.de
Ravensburger Digital		Kaflerstraße 8	81241 München	089 / 12228180	tellme@ravensburger-digital.com
Reality Twist		Berg-Am-Laim-Straße 64	81673 München	089 / 380129500	mail@reality-twist.com
redspotgames		Postfach 310464	80104 München	089 / 4036370	info@redspotgames.com
Rondomedia		Limitenstraße 64-78	41236 Mönchengladbach	02166 / 618660	info@rondomedia.de
RTL Games		Achenerstraße 1036	50858 Köln	0221 / 4560	entwickler@rtl-interactive.de
Rumble Media		Kriegsstraße 5	76137 Karlsruhe	0721 / 2018020	info@fettspielen.de
S.A.D.		Rötelbachstraße 91	89079 Ulm	07305 / 96290	kontakt@s-a-d.de
Sega		Rablstraße 24	81669 München	089 / 22848000	info@sega.de
SERU		Industriestraße 8	24589 Nortorf	04392 / 84900	info@seru.de
Sony Computer Entertainment		Frankfurter Straße 233	63263 Neu-Isenburg	06102 / 4330	scee_presscentre@scee.net
Square Enix		Domstraße 17	20095 Hamburg	040 / 30633400	
Take 2		Agnesstraße 14	80798 München	089 / 278220	info@take2.de
Thumbr		Pfistelstraße 5	10997 Berlin	030 / 300137100	info@thumbr.com
Tipp24 Entertainment		Straßenbahnring 11	20251 Hamburg		info@t24.de
Tivola Publishing		Oeverseestraße 10-12	22769 Hamburg	030 / 5363580	mail@tivola.de
Traffic Captain		Wendenstraße 21b	20097 Hamburg	040 / 23706800	info@trafficcaptain.com
Travian Games		Wilhelm-Wagenfeld-Straße 22	80807 München	089 / 3249150	mail@traviangames.com
Ubisoft		Adlerstraße 74	40211 Düsseldorf	0211 / 338000	info@ubisoft.de
United Soft Media Verlag		Thomas-Wimmer-Ring 11	80539 München	089 / 29088175	info@usm.de
upjers		Hafenstraße 13	96052 Bamberg	0951 / 510908100	mail@upjers.com
ZeniMax Germany		Schillerstraße 15/17	60313 Frankfurt am Main		kontakt.de@zenimax.com
Zuxxez		Rittnerstraße 36	76227 Karlsruhe	0721 / 464720	info@zuxxez.com

PUBLIC SECTOR					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
BIU		Rungestraße 18	10179 Berlin	030 / 24087790	info@biu-online.de
FilmFernsehFonds Bayern		Sonnenstraße 21	80331 München	089 / 5446020	filmfoerderung@fff-bayern.de
Film- und Medienstiftung NRW		Kaistrasse 14	40221 Düsseldorf	0211 / 9305000	info@filmstiftung.de
G.A.M.E.		Windscheidstraße 18	10627 Berlin	030 / 346550980	info@game-bundesverband.de
gamearea FRM		Holzhausenstraße 22	60322 Frankfurt am Main	069 / 94419441	info@gamearea-frm.de
gamecity:Hamburg		Habichtstraße 41	22305 Hamburg	040 / 2270190	info@gamecity-hamburg.de
GAMEplaces		Hanauer Landstraße 126-128	60314 Frankfurt am Main	069 / 12236214	manuela.schiffner@gameplaces.de
games.net berlinbrandenburg		Wahlertstraße 12 - 13	10115 Berlin	030 / 24628570	games.net@medianet-bb.de
GDI.Ruhr		Kreuzstraße 1	45468 Mülheim an der Ruhr	0208 / 69801520	info@games-ruhr.com
Hessen-IT		Abraham-Lincoln-Straße 38 - 42	65189 Wiesbaden	0611 / 7748481	info@hessen-it.de
medienboard berlin-brandenburg		August-Bebel-Straße 26-53	14482 Potsdam-Babelsberg	0331 / 743780	info@medienboard.de
Mitteldeutsche Medienförderung		Hainstraße 17-19	4109 Leipzig	0341 / 269870	info@mdm-online.de
nordmedia		Expo-Plaza 1	30539 Hannover	0511 / 1234560	info@nordmedia.de
Northstar Developers		Bödekerstr 84	30161 Hannover	0160 / 97880272	d.brueinig@collision-studios.com
OVUS		Aichhornaße 14	A-1120 Wien	+43 / 1810 / 328912	office@ovus.at
Stiftung Digitale Spielkultur		Torstraße 6	10119 Berlin	030 / 29049290	kontakt@stiftung-digitale-spielkultur.de
USK		Torstraße 6	10119 Berlin	030 / 24088660	kontakt@usk.de
WERK1 München		Grafinger Straße 6	81671 München	089 / 9952990	info@werk1muenchen.de

EDUCATION					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Acagamics e.V.		Universitätsplatz 2, G29	39106 Magdeburg		info@acagamics.de
Akademie für Medien		Balinger Straße 39A	70567 Stuttgart	0711 / 4209444	h.ehlers@amk-net.de
Cologne Game Lab / Fachhochschule Köln		Ubierring 40	50678 Köln	0221 / 82753095	info@colognegamelab.de
Designhochschule Schwerin		Bergstraße 38	19055 Schwerin	0385 / 5559775	info@designschule.de
EC Europa Campus		Lyonerstraße 34	60528 Frankfurt am Main	0621 / 8425660	
FH OÖ Studienbetriebs GmbH		Softwarepark 11	A-4232 Hagenberg	+43 (0) 5080420	info@fh-hagenberg.at
FH Trier		Schneidershof	54293 Trier	0651 / 8103345	sek@informatik.fh-trier.de
FH Technikum Wien		Mariahilfer Straße 37-39	1060 Wien	+43 / 1 / 588390	info@technikum-wien.at
FH Salzburg		Urstein Süd 1	A-5412 Puch/Hallein	+43 / 50 / 22110	press@fh-salzburg.ac.at
Games Academy Berlin		Rungestraße 20	10179 Berlin	030 / 29779120	info@games-academy.de
Games Academy Frankfurt		Hanauer Landstraße 146	60314 Frankfurt	069 / 42696460	info@games-academy.de
Gesellschaft für Personalentwicklung und Bildung		Beuthstraße 7-8	10117 Berlin	030 / 9339480	gpb-berlin@gpb-berlin.de
Hochschule Mittweida (FH)		Technikumplatz 17	09648 Mittweida		kontakt@hs-mittweida.de
HTW Berlin		Wilhelminenhofstraße 75a	12459 Berlin	030 / 50190	
Macromedia Akademie		Eisenheimerstraße 63	80687 München	089 / 96160800	ausbildung@macromedia.de
Macromedia Akademie		Naststraße 11	70376 Stuttgart	0711 / 2807380	info.stgt@macromedia.de
Macromedia Akademie		Brüderstraße 17	50667 Köln	0221 / 3108223	akademie.koeln@macromedia.de
Macromedia Hochschule		Gollierstraße 4	80339 München	089 / 5441510	info.muc@macromedia.de
media Akademie - Hochschule Stuttgart		Tübinger Straße 12-16	70178 Stuttgart	0711 / 925430	
Mediadesign Hochschule		Lindenstraße 20-25	10969 Berlin	030 / 3992660	info-ber@mediadesign-fh.de
Mediadesign Hochschule		Werdener Straße 4	40227 Düsseldorf	0211 / 1793930	info-dus@mediadesign-fh.de
Mediadesign Hochschule		Claudius-Keller-Straße 7	81669 München	089 / 4506050	info-muc@mediadesign-fh.de
S4G School for Games		Gubener Straße 47	10243 Berlin	030 / 96595244	info@school4games.net
SAE Institute Berlin		Soltauer Straße 18-22	13509 Berlin	030 / 43094470	berlin@sae.edu

EDUCATION					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
SAE Institute Bochum		Metzstraße 23	44793 Bochum	0234 / 93451310	bochum@sae.edu
SAE Institute Frankfurt		Homburger Landstraße 182	60435 Frankfurt am Main	069 / 543262	frankfurt@sae.edu
SAE Institute Hamburg		Feldstraße 66	20359 Hamburg	040 / 23688080	hamburg@sae.edu
SAE Institute Köln		Medienzentr. Ost, Carlswehrstr. 11c	51063 Köln	0221 / 9541220	koeln@sae.edu
SAE Institute Leipzig		Dittrichring 10	40109 Leipzig	0341 / 3085160	leipzig@sae.edu
SAE Institute München		Bayerwaldstraße 43	81737 München	089 / 89068771	muenchen.sae.edu
SAE Institute Stuttgart		Stuttgarter Straße 23	70469 Stuttgart	0711 / 81473690	stuttgart@sae.edu
SRH Fachschulen		Bonhoefferstraße 1	69123 Heidelberg	06221 / 884225	it.heidelberg@fachschulen.srh.de
SRH Hochschule Heidelberg		Ludwig-Guttman-Straße 6	69123 Heidelberg	06221 / 882790	Daniel.Goerlich@Hochschule-Heidelberg.de
Zürcher Universität der Künste		Ausstellungsstraße 60	CH-8005 Zürich	+41 / 43 / 4464646	hs.admin@zhdk.ch


BUSINESS SERVICE					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
4-REAL INTERMEDIA (Lokalisation)		Ludwigstraße 32	63067 Offenbach	069 / 8090880	info@4-real.com
adjust		Saarbrücker Strasse 36	10405 Berlin	030 / 91460083	
Anakan (Lokalisation)		Pfuelstraße 5	10997 Berlin	030 / 531420450	ana@anakan.de
Aruba Events		Kreuzstraße 1-3	Mülheim an der Ruhr	0208 / 6982608	infos@aruba-events.de
arvato games task force		Carl-Bertelsmann-Straße 161F	33332 Gütersloh	05241 / 803074	kontakt@gamestaskforce.de
ATLAS Interactive Deutschland		Christoph Probst Weg 3	20251 Hamburg	040 / 413300185	sales@atlasinteractive.de
Brehm & v. Moers		Anna-Louisa-Karsch-Straße 2	10178 Berlin	030 / 2603050	berlin@bvm-law.de
Die Hohrechts (Game Thinking)		Hobrechtstraße 65	12047 Berlin	030 / 62901232	info@diehohrechts.de
Dr. Behrmann & Härtel Rechtsanwälte		Kantstraße 98	10627 Berlin	030 / 319984910	info@behrmannhaertel.de
Buschbaum Media & PR		Emanuel-Leutze Straße 21	40547 Düsseldorf	0211 / 5988140	presse@buschbaum-media.com
Clans.de (Network)		Pinnsberg 47	20359 Hamburg	040 / 78800990	info@clans.de
CULTURETRANSLATE		Lyoner Straße 34	60528 Frankfurt am Main	069 / 65998000	info@culturetranslate.de
deal united		Landshuter Allee 10	80637 München	089 / 416157014	info@dealunited.com
DELASOCIAL		Christoph-Probst-Weg 31	20251 Hamburg	040 / 87979890	contact@delasocial.com
Digital River		Vogelsanger Straße 78	50823 Köln	0221 / 310 88614	sales@digitalriver.com
Effective Media		Josef-Haumann-Straße 10	44866 Bochum	02327 / 22310	info@effective-media.de
elblabs		Bernstorffstraße 120	22767 Hamburg	040 / 76797218	info@elblabs.com
Frankfurter Buchmesse		Reineckstraße 3	60313 Frankfurt am Main	069 / 21020	info@book-fair.com
Freaks 4U Gaming		An der Spreeschanze 10	13599 Berlin	030 / 41719113	info@freaks4u.de
Friedmann Kommunikation		Schanzenstraße 36	51063 Köln	0221 / 5348710	info@fr-k.de
gamescom		Messeplatz 1	50679 Köln	0221 / 8213894	info@koelnmesse.de
Games Foundation		Neuer Kamp 30	20357 Hamburg		info@gamesfoundation.com
Games Quality		Zum Wasserwerk 15	15537 Erkner	03362 / 885688	info@games-quality.com
GAN Game Ad Net		Stresemannstraße 342	22761 Hamburg	040 / 67586750	info@gan-online.com
Ganz & Stock (Personnel Consulting)		Westhafentower, Westhafenplatz	60327 Frankfurt am Main	069 / 710456380	a.stock@ganz-stock.de
Game Developers Conference Europe		Kaiser-Wilhelm-Straße 30	12247 Berlin	030 / 34622644	carsten.kohlenbeck@ubm.com
GAMINSIDE		Kaiserstraße 2-4	1070 Wien	+43 / 0676 / 5379776	info@gaminside.com
Gärtner PR		Deistouchesstraße 68	80796 München	089 / 30766854	info@gaertner-pr.de
GRAEF Rechtsanwälte		Jungfrauenthal 8	20149 Hamburg	040 / 80600090	www.graef.eu
HoneyTracks (Game Analytics)		Schelling Straße 35	80799 München	089 / 28723995	info@honeytracks.com
IDG Entertainment Media		Lyonel-Feininger-Straße 26	80807 München	089 / 360860	dbhulapatna@idg.de
IEM Consulting (Business Development, Finanzierung)		Hanauer Landstraße 192	60314 Frankfurt am Main	069 / 15346479	info@iem-consulting.com
IHDE & Partner (Rechtsberatung)		Schoenhauser Allee 10-11	10119 Berlin	030 / 44318660	berlin@onlinelaw.de
IMC (Full-Service Dienstleister)		Hauptstraße 172	51143 Köln	02203 / 949092	info@imc-online.net
INPROMO		Gasstraße 6a	22761 Hamburg	040 / 23881771	kontakt@inpromo.de
iVentureCapital		Wendenstraße 21 B	20099 Hamburg	040 / 8081250	info@iventurecapital.com
LaterPay (Zahlungsdienstleister)		Auenstraße 100	80469 München	089 / 416137319	info@laterpay.net
Lightning PR		Gundelandsstraße 13	60435 Frankfurt		koetting@lightning-pr.de
Marchsreiter Communications		Guldenstraße 41a	80339 München	089 / 51919942	dm@marchsreiter.com
Medienachse (Full-Service, Businessplanung)		Trappentreustraße 10	80339 München		info@medienachse.de
M.I.T. - Media Info Transfer		Albert-Einstein-Ring 8	22761 Hamburg	040 / 30066890	info@mediainfotransfer.de
MOTION AREA		Sommerstraße 3	65197 Wiesbaden	0611 / 56597577	info@motionarea.de
Osborne Clarke (Anwaltskanzlei)		Innere Kanalstraße 15	50823 Köln	0221 / 51084000	info@osborneclarke.de
Partnertrans (Lokalisation)		Leipziger Straße 28	40822 Mettmann	02104 / 172660	info@partnertrans.com
paysafecard.com Wertmarken		Am Europlatz 2	1120 Wien	+43 / 1 / 72083800	sales@paysafecard.com
PIXABLE STUDIOS		Buchenstraße 16b	01097 Dresden	0351 / 56341360	info@pixable.de
Pixelworkshop / IG Computergrafik		Hirschengasse 1/26	1060 Wien	+43 / 1 / 5880118698	info@pixelworkshop.at
Play + Smile Marketing (PR, Marketing)		Am Wasserturm 6	40668 Meerbusch	02150 / 705660	info@play-and-smile.de
ProCom International LTD		Kisdorfer Weg 3b	24568 Kalkenkirchen	040 / 71669763	info@procom-international.com
Quinke Networks (PR, Marketing)		Bei den Mühnen 70	20457 Hamburg	040 / 43093949	mail@quinke.com
remote control productions (Producing)		Sendlinger-Tor-Platz 6	80336 München	089 / 210205700	info@r-control.de
Rode + Mathé (Rechtsberatung)		Henriettenweg 4	20259 Hamburg	040 / 43270432	info@rodemathé.de
ROESSLER PR (Marketing, PR, Consulting)		Walter-Leiske-Straße 2	60320 Frankfurt am Main	069 / 514461	communicate@roesslerpr.de
S&H Entertainment Localization (Lokalisation)		Weidenstraße 10a	82110 Germering	089 / 80076290	stefan@shentloc.com
SCC (Crossmedia)		Orleansstraße 5a	81669 München	089 / 33094660	info@scc-feld.de
Schanz International (Consulting)		Bgm.-Oberhettinger-Straße 2a	67146 Deidesheim	06326 / 6010	info@schanzgames.com
Schulte Riesenkampf (Rechtsberatung)		An der Hauptwache 7	60313 Frankfurt am Main	069 / 900266	schulte@schulte-lawyers.de
Selected Minds (Personalberatung)		Walther-Rathenau-Straße 16	64521 Groß-Gerau	06152 / 1871830	info@selected-minds.de
Serious Games Conference		Expo Plaza 1	30539 Hannover	0511 / 1234560	sgc@nordmedia.de
Sputnic Consulting		Ehrenbergerstr. 11	98693 Ilmenau	03677 7996990	media@sputnic-consulting.com
Station54 Community Management		Pankstraße 8-10	13127 Berlin	030 / 44716415	info@station54.de
swordfish PR (PR, Marketing)		Habsburgerplatz 2	80801 München	089 / 96160840	info@swordfish-pr.de
Translation-Taxi		Beusselstraße 28	10553 Berlin	030 / 28371428	info@translation-taxi.com
Turtle Entertainment (eSport)		Siegburger Straße 189	50679 Köln	0221 / 8804490	info@turtle-entertainment.de
Two Pi Team		Gerresheimer Straße 9	40721 Hilden	02103 / 9411914	info@two-pi-team.de
Visibility Communications (PR, Text)		Wichertstraße 16/17	10439 Berlin	030 / 58859341	hi@visi.bi
wazal		Sonnenallee 70	12045 Berlin	0178 / 5223697	jannot@wazaservices.de

FREELANCER BUSINESS SERVICE					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Ahmet Isciturk		Kernstraße 37	90429 Nürnberg	0911 / 27472885	texte@me.com
Anja Weillgmann		Kramers Kamp 1a	48291 Telgte	02504 / 7399601	kontakt@spielworte.de
Anne-Petra Lellwitz (Marktforschung)		Fichtestrasse 65	63303 Dreieich	0160 / 2014364	info@apl-mafo.de


FREELANCER BUSINESS SERVICE					
Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail	
Arwed-Ralf Grenzbach (Consulting)	Friedrich-Ebert Anlage 18	60325 Frankfurt am Main	0171 / 6486720	arg@conflutainment.com	
Christiane Gehrke	Friedrich-Wilhelm-Straße 15	12103 Berlin	030 / 72010080	cg@tom-putzki-consulting.com	
Elisabetta Corapi			0176 / 99763437	info@corapi-translations.com	
Falko Löffler (Autor)	Lindenstraße 8a	36355 Grebenhain-Ilbeshausen	06643 / 918577	mail@falkoeffler.de	
Kerstin Fricke	Wissmannstraße 9	12049 Berlin	030 / 20059246	info@kf-uebersetzungen.de	
Marc Huppke (Projektmanagement)	Frankfurter Straße 5	64521 Gross-Gerau	06152 / 9614432	marc@make-projects.com	
Marc Oberhäuser (Consulting)	Berzbuier Strasse 98	52355 Düren	02421 / 51547	marc.oberhauser@freelancer-games.com	
Michael Bhatti (Entwicklung, Consulting)	Bismarckstraße 24	49076 Osnabrück	0541 / 1810660	michael@bhatty.de	
Nicole Klinge (PR, Marketing, Events)	Schopenhauerstraße 101	80809 München	0173 / 7872013	nk@gconsultnet.de	
Stefan Köhler (Lektor / Autor)	Am Salgenteich 34	38259 Salzgitter	0176 / 22512273	stefanclenemkoehler@web.de	
TECHNICAL SERVICE					
Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail	
Akamai Technologies	Betastraße 10B	85774 Unterföhring	089 / 94006308	contact-germany@akamai.com	
Augenpulver (Grafik)	Dotzheimerstraße 164, 4.0G	65197 Wiesbaden	0611 / 94574340	info@augenpulver-design.de	
Augmented Minds	Fasangartenstraße 134	81549 München	089 / 69386904	info@augmented-minds.com	
BiteTheBytes (Middleware)	St. Gallener Straße 9	36039 Fulda	0661 / 2006899	info@bitethebytes.com	
B.TON Medien (Audio, Musik)	Luise-Ullrich-Straße 4	82031 Grünwald	089 / 189425490	mix@bton.de	
Centroid (Motion Capturing)	Schlesische Straße 27	10997 Berlin	030 / 69807474	sven.bergmann@centroidanimation.com	
Chromatrix (Mobile Gaming)	Kalkofenstraße 2	72411 Bodelshausen	07471 / 740828	info@chromatrix.com	
Codecult (3D-Engine)	Kemnastraße 21a	44866 Bochum	02327 / 35474	info@codecult.com	
Contentainer (Leveldesign)	Eichelkopfstraße 17	63584 Gründau	06058 / 9178150	info@contentainer.de	
Creature Factory	Graefestraße 33	10967 Berlin	030 / 26301330	info@creature-factory.com	
Crytek (3D-Engine)	Grüneburgweg 16-18	60322 Frankfurt am Main	069 / 21977660	cryengine@crytek.com	
definitiv:ton (Audio)	Mozartstraße 35-37	50674 Köln	0221 / 8234220	mhoechsmann@definitiv-ton.de	
Die Kolonie (Art, Design)	Adelheidstraße 82	65185 Wiesbaden	0611 / 5058785	contact@diekolonie.com	
Doublesmith (Art/Animation)	Ranzonigasse 3/3	A-3100 St. Pölten	+43 / 699 / 19447880	mail@doublesmith.com	
Dutyfarm	Oranienstraße 6	10997 Berlin	030 / 30368430	info@dutyfarm.com	
Dynamedion (Sounddesign)	Barbarossaring 8	55118 Mainz	06131 / 5847895	info@dynamedion.com	
DynamicHead (Unity Services)	Hauptstraße 6	55283 Nierstein Schwabsburg	061 / 33572494	timo.weisse@dynamichead.com	
eurosimtec	Merowingerplatz 1	40225 Dusseldorf	0211 / 3018560	info@eurosimtec.de	
Exit Games (Network Engine)	Hongkongstraße 7	20457 Hamburg	040 / 4135960	business@exitgames.com	
Eye Rock Media (Design, Animation)	Essenerstraße 3	46047 Oberhausen	0208 / 82858923	contact@eyerock.de	
fadeout (Audio Design)	Austraße 6	CH-4153 Reinach	+41 / 7 / 92793244	info@fadeout.ch	
fatfoogoo	Mariahilferstraße 50	A-1070 Wien	+43 / 1 / 23622970	office@fatfoogoo.com	
Flow Studios	Gustav-Meyer-Allee 25	13355 Berlin	030 / 80929203	info@flow-studios.eu	
Games Quality (Quality Assurance)	Zum Wasserwerk 15	15537 Erkner	03362 / 885688	info@games-quality.com	
Global Producer (Production)	Hauptstraße 172	51143 Köln	02203 / 9772400	getintouch@global-producer.com	
Glare Studios (Grafik)	Mansteinstraße 18	20253 Hamburg	02307 / 4388404	contact@glarestudios.de	
Goal Games	Feurigstraße 54	10827 Berlin	030 / 78957102	info@goal-games.de	
GreenMamba-Studios (Animation)	Aloysstraße 7	48249 Dülmen	02594 / 7928217	info@greenmamba-studios.de	
Hans HiScore (Audio)	Friedrich-Ebert-Anlage 11-13	60327 Frankfurt	0151 / 11616133	info@hans-hiscore.de	
Havok	Arbachtalstraße 6	72800 Eningen	0712 / 1986993	info@havok.com	
Intulo (3D, Animation)	Lindholz 89	31139 Hildesheim	05121 / 2944022	info@intulo.de	
Johan Weigel (Musik/Ton/Sound)	Christburger Straße 28	10405 Berlin	0176 / 4146494	johan@sonic-gallery.com	
Keuthen	Robert-Koch-Straße 41	55129 Mainz	06131 / 880890	info@keuthen.net	
Konsole Labs	Gritznerstraße 42	12163 Berlin	030 / 23634838	info@konsole-labs.com	
Lingooona (Lokalisations-Middleware)	Hildastraße 38	68723 Plankstadt	06202 / 4095030	info@lingooona.com	
Living Liquid Software	Langer Weg 15	A-6020 Innsbruck	+43 / 0 / 512263535	info@livingliquid.com	
metricminds (Grafik)	Rüsselsheimer Straße 22	60326 Frankfurt am Main	069 / 7593380	info@metricminds.com	
MobileBits (Mobile Engine)	Lerchenstraße 28	22767 Hamburg	040 / 33429566	info@mobilebits.de	
morro images (Grafik)	August-Bebel-Straße 27	14482 Potsdam-Babelsberg	0331 / 97996610	info@morrimages.com	
Neviso (Tools & Middleware)	Lyrenstraße 13	44866 Bochum	02327 / 8369840	info@neviso.com	
Periscope Studio (Audio)	Am Diebsteich 55	22761 Hamburg	040 / 31811767	info@periscopestudio.de	
Phenomatics (XNA-Entwicklung, Consulting)	Hafenstrasse 47-51	A-4020 Linz/Donau	+43 / 732 / 90155230	office@phenomatics.com	
Polyce (Grafik)	Littenstraße 106/107	10179 Berlin	0178 / 3592560	info@polyce.de	
Polygonfabrik (Grafik)	Friesenstraße 21	28203 Bremen		kontakt@polygonfabrik.de	
Rabcat Computer Graphics (Grafik)	Fernkornegasse 10	A-1100 Wien	+43 / 1 / 5237425	mail@rabcat.com	
Razer (Hardware)	Winterhuder Weg 82	22085 Hamburg	040 / 41929300		
Rocketbox Studios (Grafik)	Leonhardtstraße 10	30175 Hannover	0511 / 8984384	info@rocketbox.de	
Spinor (3D-Engine)	Agnes-Pockels-Bogen 1	80992 München	089 / 54043980	info@spinor.com	
The Light Works (Grafik)	Otto-Hahn-Straße 7	50997 Köln	02236 / 967322	info@thelightworks.com	
ToBringAlive (Animation, Design)	Ostmarkstraße 49	48145 Münster	0251 / 392312	angerbauer@tobringalive.com	
Tektroop (Support)	Dechenstraße 8	40699 Erkrath		info@tektroop.ch	
TON & SPOT audiodesign (Sound)	Immanuelkirchstraße 15	10405 Berlin	030 / 92129727	post@ton-und-spot.de	
Toygardens Media (Grafik, Video)	Walderseestraße 54	30163 Hannover	0511 / 33659800	akunze@toygardens.com	
VCC Perfect Pictures (Grafik, Video)	Doormannsweg 43	20259 Hamburg	040 / 431690	hamburg@vcc.de	
Versant (Data Management)	Wiesenkamp 22b	22359 Hamburg	040 / 6093090	info@versant.com	
Virgin Lands (Grafik, Cinematics)	Gneisenaustraße 10/11	97074 Würzburg	0931 / 8049000	info@virgin-lands.com	
xaitment (KI-Middleware)	Fischbacherstraße 92	66267 Quierschied	06897 / 600800	info@xaitment.com	
Z-Ground Illustration (Grafik)	Oeltzenstraße 17	30169 Hannover	0511 / 1696959	jz@z-ground.com	
zuuka! (Audio, Lokalisation)	Christian-Pless-Straße 11-13	63069 Offenbach am Main	069 / 2475700	mail@zuuka.de	
FREELANCER TECHNICAL SERVICE					
Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail	
Alexander N. Ostermann (Animator)	Breite Strasse 45	14199 Berlin	0176 / 10365759	nico@nico-ostermann.com	
Andreas Adler (Audio)	Wernigeroder Weg 18	30419 Hannover	0511 / 3889163	andreas@adler-audiopictures.de	
Calined (Audio)	Lange Straße 22	58089 Hagen	01577 / 9508456	business@calined.com	
Daniel Pharos (Audio)	Golddistelanger 15	80937 München	0179 / 5210100	daniel@knightsoundsoundtrack.com	
Grafikbüro Werner (Grafik)	Beethovenstraße 40a	48317 Drensteinfurt	02508 / 985801	info@grafikbuero-werner.de	
Johan Weigel (Musik / Ton / Sound)	Christburger Straße 28	10405 Berlin	0176 / 4146494	johan@sonic-gallery.com	
Sound42 / Lukas Hasitschka	Kulmgasse 38/8	1170 Wien	+43 / 6641059157	lukas@sound42.com	
Maurice T. Anderson (Sounddesign)	Klosterstraße 3	63486 Bruchköbel	0176 / 26228136	contact@maurice-t-anderson.com	
Max Schulz				info@xoco1.com	
Olaf Bartsch (Sounddesign)	Sültstraße 62	10409 Berlin	0170 / 7743432	info@olafbartsch.com	
Sabrina Heuer (Audio Production)	Bachemer Straße 191	50935 Köln	0163 / 8604004	sabrina@froschtatze.de	

Ein Tag bei ...


BOHEMIA INTERACTIVE




Willkommen im Büro von Bohemia Interactive in Mníšek pod Brdy, in der Nähe von Prag. Im Moment sind unsere Entwickler-Teams auf fünf Standorte verteilt: Mníšek, Prag, Brünn und Bratislava.




In Mníšek hat die Geschichte von Bohemia Interactive ihren Lauf genommen und das Büro beherbergt einen Teil des Arma-3-Teams.




Unsere Empfangsdame Jitka sorgt für den reibungslosen Ablauf des Tagesgeschäfts. Außerdem entsorgt sie die Körperteile, die der Entwicklung zum Opfer fallen – siehe die abgetrennte Hand rechts außen.




Unsere Programmierer zeigen sich überraschend diskussionsfreudig.




Tester Ondrej Kuzel rüstet sich mit allem erdenklichen Equipment, um sich den Herausforderungen des Design-Teams zu stellen.



Vojtech Hladik, Lead-Programmer von Arma 3, trägt sein Baby stolz auf der Brust vor sich her – so muss das sein.



Vertreter des Internationalen Komitees des Roten Kreuzes (IKRK) besuchen uns in Mníšek. Es steht eine Diskussion über das Humanitäre Völkerrecht in Arma 3 auf dem Tagesplan.



Für einige Szenen im Einzelspielermodus von Arma 3 nehmen wir Schauspieler auf, um deren Aktionen ins Spiel zu übersetzen.

Das Ferienhausidyll mag auf den ersten Blick täuschen, doch ein Besuch bei den Militär-Experten von Bohemia Interactive enthüllt Hightech-Ausstattung und konzentrierte Arbeit unweit von Prag.



Diese Jungs hauchen den Spielinhalten von Arma 3 Leben ein. Achtung: Kein Hüpfball wurde während der Produktion des Spiels verletzt.



In täglichen Scrum-Meetings stehen offene Aufgaben und Deadlines zur Diskussion.



Für die Bearbeitung der dabei entstandenen Aufnahmen muss es aber ... dunkel sein.



Heute haben die Audio-Spezialisten von Arma 3 Glück und machen einen Ausflug auf einen Flugplatz des Tschechischen Militärs.



Die DayZ-Fans in unserem Team holen sich ihre Getränke in unserem Garten. Aber wir haben auch traditionelle Durstlöcher vorrätig.



Die Umgebung dient oft als Inspiration für Spielinhalte, aber auch um den Kopf frei zu bekommen.



Sobald unsere Programmierer an ihren Platz zurückkehren, erinnern wir sie daran, wo ihre Prioritäten liegen sollen.



In der Zwischenzeit sorgt eine Armee von Testern in unserem Büro in Prag dafür, dass die Kollegen in Mníšek am nächsten Tag genug zu tun haben. Im gleichen Gebäude residieren übrigens die Teams von DayZ und Take on Mars.



Erfolg freigeschaltet: Sie haben einen Tag bei Bohemia Interactive überlebt.

MAKING GAMES VORSCHAU

06|2014
ERSCHEINT AM 24.10.2014
MAKING GAMES

Making of Dragon Age Inquisition

Nanu, die aktuelle Ausgabe hatte doch schon ein Making of als Titelgeschichte? Stimmt, aber die Entwicklung von BioWares nächstem großen Rollenspiel ist so spannend, dass wir gern mit der »ein Making of pro Jahr«-Tradition brechen. Das Entwicklerteam erklärt uns die Mechanik der Drachen und wie Geschichte und Erkundung ineinander greifen. Außerdem erfahren wir in einer Design Case Study, wie sich die Charakterfähigkeiten zwischen Taktik- und Echtzeitkampf unterscheiden.

Weitere Themen

Interview

Google-Earth-Mitbegründer John Hanke erläutert die Hintergründe seines location-based Augmented-Reality-Spiels »Ingress«.

LEGO Minifigures Online

Das Funcom-Team gibt einen Einblick in die Crossplattform-Entwicklung seines kindertauglichen MMOs.

Eine IP neu erfinden

Die Entwickler von »Angry Birds Epic« erklären den Weg von der Vogelschleuder zu rundenbasierten Kämpfen und was dazu alles notwendig war.

IMPRESSUM

Chefredakteur

Heiko Klinge
(verantwortlich,
Anschrift der Red.)

Redaktion

Yassin Chakhchoukh
Patricia Geiger (Trainee)
Markus Schwerdtel
Sebastian Weber

Layout & Design

Sigrun Rüb (Itd.)
Manfred Aumaier
Alexander Wagner

Eva Zechmeister
Anita Blockinger
(Freelancer)

Lektorat (Freelancer)

Marion Schneider

Freie Mitarbeiter in dieser Ausgabe (Redaktion)

Ralf Adam, Filippo Beck Peccoz, Bernd Diemer, André Dittrich,
Vivien Dollinger, Nurdogan Erdem, Benjamin Glatzel,
Thomas Grip, Daniel Helbig, Michael Kempson, Mark Kölker,
Jan-Eric Laube, Dominik Lazarek, Carsten Lindner, Jan Reichert,
David Reinig, Marc Sodermanns, Thorsten Unger

Wir danken den Interview-Partnern:

Andrew Bowell, Carl Jones, Michael Kempson, Carsten Lindner,
Yasuhiro Wada

Assistenz: Isa Stamp, Anita Thiel, Annie Weissenberger

KONTAKT

Anzeigenanfragen:

Nicole Klinge, 089 / 360 86-680
nklinge@idg-consultant.de

Artikel- & Themenvorschläge:

editor@makinggames.de

Recruiting- & Projektanfragen:

projects@makinggames.de

Adressänderungen Abo:

service@makinggames.de

Änderungen für Firmenregister:

info@makinggames.de

Betreff: Firmenregister

Digitale Anzeigenannahme Print:

Manfred Aumaier, 089 / 360 86-602
anzeigendisoprint@gamestar.de

Digitale Anzeigenannahme Online:

banner@idg.de

Anzeigenpreise:

www.idg-eas.de

KUNDENSERVICE

Abonnement und Einzelbestellung:

Making Games Kundenservice
ZENIT Pressevertrieb GmbH
Postfach 81 05 80, 70522 Stuttgart
Tel.: 0711 / 7252-275

Fax: 0711 / 7252-377

Österreich: Tel.: 01 / 219 55 60

Schweiz: Tel.: 071 / 314 06-15

E-Mail: service@makinggames.de

Web: www.makinggames.de/shop

Bezugspreise:

Einzelpreis: 6,90 Euro

Jahresabo: 35,40 Euro

(beinhaltet 6 Ausgaben)

Auslandspreise u. Mengenrabatt auf Anfrage

Zahlungsmöglichkeit:

Postbank Stuttgart, BLZ 600 100 70

Konto-Nr.: 311 704

So erreichen Sie die Redaktion:

IDG Entertainment Media GmbH

Redaktion Making Games

Lyonel-Feininger-Straße 26

80807 München

Tel.: 089 / 360 86-660, Fax: 089 / 360 86-652

VERTRIEB HANDELSAUFLAGE

MZV GmbH & Co. KG

Ohmstraße 1, 85716 Unterschleißheim

Telefon: 089 / 319 06-0, Fax: -113

E-Mail: info@mzv.de

Web: www.mzv.de

VERLAG

IDG Entertainment Media GmbH

Lyonel-Feininger-Straße 26

80807 München

Tel.: 089 / 360 86-0, Fax: 089 / 360 86-501

www.idgmedia.de

Gründer: Patrick J. McGovern (1937 - 2014)

Geschäftsführer: York von Heimburg

Verlagsleitung: Frank Maier

Director Sales: Ralf Sattelberger (-730)

Leiter CRM/Marketing: Matthias Weber (-154)

Veröffentlichung gemäß §8, Absatz 3 des Gesetzes

über die Presse vom 08.10.1949: Alleiniger Gesell-

schafter der IDG Entertainment Media GmbH ist die

IDG Communications Media AG, München, die 100%

Tochter der International Data Group Inc., Boston,

USA, ist.

Vorstand:

York von Heimburg, Keith Arnot, David Hill

Produktionsleitung: Druck und Beilagen:

SDV Direct World GmbH

01159 Dresden

HAFTUNG & CO.

© Copyright IDG Entertainment Media GmbH

Haftung: Für die Richtigkeit von Veröffentlichungen können Redaktion und Verlag trotz Prüfung nicht haften. Die Veröffentlichungen in Making Games erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Auch werden Warennamen ohne Gewährleistung einer freien Anwendung benutzt.

GDC NEXT FEATURING ADC RETURNS
NOVEMBER 3-4, 2014

**making
games**

READERS,

USE CODE **GDCN14MG**

TO SAVE AN ADDITIONAL 10%
ON SELECTED PASSES!

REGISTER BY SEPTEMBER 18 AND SAVE UP TO \$500!

GDC NEXT

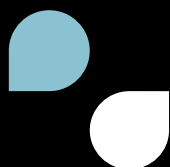
FEATURING **ADC**
APP DEVELOPERS CONFERENCE

GAME DEVELOPERS CONFERENCE® NEXT

FEATURING APP DEVELOPERS CONFERENCE

LOS ANGELES CONVENTION CENTER · LOS ANGELES, CA

NOVEMBER 3-4, 2014



GDCNEXT.COM



YAGER

15

1999
2014

YEARS



YAGER

YAGER

**SPEC OPS
THE LINE**

**DEAD
ISLAND 2**

DREDDNOUGHT

WWW.YAGER.DE/CAREER

©All trademarks and registered trademarks are the property of their respective owners.