2017
3
ISSN 1433-2620 > B 43362 >> 21. Jahrgang >>> www.digitalproduction.com
Published by ATEC

Deutschland € 17,70
Österreich € 19,–
Schweiz sfr 23,–

DIGITAL PRODUCTION

DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPRODUKTION

MAI | JUNI 03:2017

Plug-ins
Schneller und leichter durch die Pipeline gleiten

Bildungswege
Neue Chancen & Horizonte – ohne Uni-Alltag!

Workshops
Houdini 16, Baselight 5, C4D, Modo und mehr

4 194336 217709 03 >

Der Wald wurde mit finalen Visuals zum Schauplatz der Game-Demo.

Die Fabelwesen-Welt der Ameliè

Im großen visuellen Maßstab: Ein Studententeam des Studiengangs „Zeitbasierte Medien" an der Hochschule Mainz hat als Bachelorabschlussprojekt die Realisation eines Thirdperson-Adventure-Games in Angriff genommen. Hier erklärt das Team, wie der Prozess für „Sane", das nach wie vor in der Mache ist, bislang ablief.

von Mirja Fürst

Schauplatz von „Sane" ist Frankreich im Jahr 1919. Der Spieler schlüpft in die Rolle von Ameliè, einem dreizehnjährigen Mädchen, das die Schrecken des ersten Weltkriegs und den Tod ihrer Eltern durch die Flucht in ein fantasiertes Wunderland verarbeitet. Die Geschichte erlebt der Spieler in einer Semi-Open-World – das bedeutet, dass Umgebungen ähnlich wie in „Dark Souls" frei erkundbar und logisch verbunden sind, man jedoch nicht wie zum Beispiel in „Skyrim" jeden Berg erklimmen kann. Die Erkundungstour ist außerdem durch Ameliès Fähigkeiten eingeschränkt. Diese entwickeln sich erst im Laufe des Spiels durch die Unterstützung der sogenannten „Companions" – jenen Fantasiewesen, welche sich dem Mädchen anschließen und durch ihre jeweiligen Charaktereigenschaften neue Areale öffnen. Vergleichbar ist das Prinzip mit dem von „Zelda", wo neue Waffen und Gegenstände Zugang zu anderen Dungeons gewähren.

Zur Umsetzung des ambitionierten Projekts verwendet das Team die Unreal Engine 4 zusammen mit Maya, ZBrush, Substance Painter, Marvelous Designer, Photoshop und Megascans. Zwei der Teammitglieder, Christian Müller und Lukas Kölz, haben mit „Sane" ihren Bachelorabschluss gemacht.

Hierfür mussten die Studenten sich auf ein spezielles Fachgebiet konzentrieren. Christian entschied sich für Character Locomotion und Animation in UE4, Lukas für Environment Art. Da es an der Hochschule nicht gestattet ist, mehr als drei Monate an der Bachelorarbeit zu werkeln, wurden alle Arbeiten in diesen Bereichen im letzten Semester erledigt. Zu diesem praktischen Teil gehörte auch noch eine theoretische Abschlussarbeit.

Dauer & Entwicklungsstand

Bislang hat das Team insgesamt vier Semester an „Sane" gearbeitet, wobei sich die Teambesetzung im Laufe der Zeit verändert hat. Am Anfang war geplant, das Projekt in drei Semestern fertigzustellen: ein Semester für Konzeption und Planung, eines zur Erstellung eines spielbaren Prototypen und eines zur Kreation einer Demoversion mit finalen Bildelementen. In der Praxis hat sich das Ganze etwas anders entwickelt: Lukas Kölz, Projekt-Lead und Artist für Charaktere und Environments, legte für ein Semester eine Pause vom Studium ein, um ein Praktikum bei Guerrilla Games in Amsterdam zu machen und am Spiel „Horizon: Zero Dawn" als Environment Artist zu arbeiten - eine Chance, die er sich natürlich nicht entgehen lassen konnte. Ein Teil des Teams musste in diesem Semester den Abschluss machen und stand nicht mehr zur Verfügung, als das Projekt im darauffolgenden Semester fortgesetzt wurde. Das neue Team brauchte etwas Zeit, um sich einzuspielen, und einige konzeptionelle Elemente wurden überarbeitet, wodurch sich die Entwicklungszeit um ein Semester verlängerte. Jetzt liegen vor: die vollständige Konzeption für das Spiel inklusive aller Elemente (Story, Gameplay, Charaktere, etc.), ein Prototyp, der das komplette erste Kapitel des Spiels umfasst (circa 4 Stunden Gameplay) und eine kurze Demo, die den finalen Look des Spiels zeigt. Während des Projektverlaufs haben sich die ursprünglichen Pläne für „Sane" verändert: Um einen Publisher zu finden oder eine Kickstarter-Kampagne zur Finanzierung des Games zu starten, hätten zunächst verschiedene Schlüsselpositionen wie beispielsweise ein Programmierer besetzt werden müssen. Als Lukas jedoch ein Jobangebot von Playground Games in England erhielt, ging das Projekt „Sane" für unbestimmte Zeit in den Hiatus. Was nicht das Ende für „Sane" bedeutet – zu einem späteren Zeitpunkt würde das Team die Konzeption sehr gerne vollständig realisieren, aber zunächst wird es etwas ruhiger werden.

Ursprünglich hatte das Team für März 2017 das Release einer Pilot-Episode geplant, die den Umfang des Prototypen haben sollte, also rund 4 Stunden Spielzeit. Durch die Verzögerung um ein Semester wird dieses Vorhaben jedoch noch mindestens drei Monate Zeit benötigen. In Kürze soll ein Playthrough der Demoversion online gehen. Eine spielbare Version der Demo wird es wohl leider nicht geben, da hier zunächst Tests vonnöten wären, um zu gewährleisten, dass das Spiel auch auf allen Rechnern zuverlässig funktioniert.

Gaming-Plattformen

Das Spiel ist konzipiert als PC- und Current-Gen-Konsolen-Titel, der grafisch einiges hermachen soll. Bisher haben die Studenten das Game nur auf PC getestet. „Den aktuellen Build auf PS4 und XBox One zum Laufen zu bringen, sollte jedoch wenig Schwierigkeiten bereiten, da Unreal für die Multiplattformentwicklung ausgelegt ist. Der Großteil der Arbeit würde sich hier auf Optimierung und Kompatibilitätsprobleme mit ein paar der von uns verwendeten Plug-ins wie etwa NeoFur beschränken.

Der Spieler nimmt in „Sane" Ameliès Rolle ein. Hinsichtlich Alter und Look nahm das Team an dem Character im Laufe des Prozesses diverse Anpassungen vor.

Reynard ist einer von Ameliès imaginierten Freunden. Er ist eine Hybrid-Creature zwischen Junge und Fuchs.

Die Steuerung war von Anfang an auf die Nutzung eines Gamepads ausgerichtet, da sich dieses für ein entspanntes Third-Person-Adventure besser eignet als Maus und Tastatur", erklären die Studenten.

Vorbilder

Bei der Beschreibung der Handlung fällt natürlich sofort eine Referenz ein: „Alice im Wunderland". Neben der Geschichte von Lewis Carroll war aber das etwas düsterer angelegte „Pans Labyrinth" von Guillermo del Toro noch mehr Vorbild für „Sane". Die Frage, ob die Geschehnisse real sind oder nur im Kopf der Hauptperson zur Überwindung von Traumata stattfinden, ist hierbei zentraler Punkt. Viele der Filme von Miyazaki wie „Mein Nachbar Totoro" behandeln ähnliche Themen, ohne jedoch auf den psychischen Zustand der Kinder oder die Frage nach den Fabelwesen einzugehen.

In punkto Gameplay sollte die ruhige Stimmung eines klassischen Adventure-Games wie die von „Monkey Island" vermittelt werden, ohne dass dabei auf Neuerungen wie freie Bewegung, 3D, Thirdperson usw. verzichtet wird. Inspiration für das Level Design boten „Zelda" und „Dark Souls" – beides Spiele, in welchen erst spät im Spielverlauf erlangte Items und Fähigkeiten das Voranschreiten blockieren und das erneute Erkunden von alten Umgebungen spannend machen. Darüber hinaus war den Entwicklern eine sehr persönliche Spielerfahrung wichtig, bei der sich das Game an die Spielweise anpasst und nicht nur durch Dialogentscheidungen definiert wird. Referenz hierfür war ebenfalls „Dark Souls" – meist nur genannt für seinen hohen Schwierigkeitsgrad, bietet das Game mehrere Enden, Charakter Arcs, die nur voranschreiten, wenn man im richtigen Moment an der richtigen Stelle ist, und eine Vielzahl an alternativen Routen. Alle diese Elemente wollten die Studenten in ihrem Spiel wiederfinden.

Die Wahl der Engine

Bei dem vorherigen Projekt „Bedtime Story" nutzten die Studenten noch die Cryengine. „Damals trafen wir diese Entscheidung, da es uns um die realistische Darstellung von natürlichen und

Für das finale Spiel konzipierten die Studenten neun Environments, wovon sich sechs im spielbaren Prototypen als Greybox befinden.

freiläufigen Außenarealen ging und wenig eigentliches Gameplay stattfand. Bei ‚Sane' brauchten wir dagegen eine Engine, die neben der Darstellung von Wäldern auch sehr detaillierte und aufwendig gerenderte Innenareale ermöglichte. Was die Cryengine natürlich auch leisten kann. Allerdings war uns zusätzlich ein performantes Ergebnis wichtig, weshalb wir ein Lichtsystem mit gebackenen Lightmaps verwendeten, welche die Cryengine nicht unterstützt", erläutert das Team seine Engine-Wahl.

Neben der Entscheidung im visuellen Bereich war auch der Workflow wichtig: Unreal bot hier eine simple Lösung mit dem direkten Import von .fbx-Dateien sowie Source Control, durch die mehrere Personen zeitgleich am gleichen Projekt arbeiten konnten. Bei „Sane" musste auch Gameplay programmiert und Levelstreaming-Systeme aufgesetzt werden, um ein vollständiges Spiel und nicht nur hübsche Landschaften zu erhalten. Da es im Team keinen Programmierer gab, war praktisch, dass Unreal die Arbeit mit Blueprints auch ohne Programmierkenntnisse erlaubte, was den ganzen Prozess massiv vereinfachte. Unity war eine andere Engine, über welche die Studenten zu Beginn des Projekts nachdachten – nach ein paar kurzen Tests entschied sich das Team aber für Unreal, da den Studenten die Art-Pipeline und der Editor, speziell für die Materialerstellung, mehr zusagten.

Blueprints & Mastermaterial

Mithilfe der Blueprints wurde das Platzieren von Geröll, Zweigen und Vegetation automatisiert und über Rule Sets entschieden, wo was gesetzt werden soll, was den Prozess deutlich beschleunigte. Für die Photogrammmetrie-Assets – hauptsächlich Steine und Vegetation – kreierte Lukas ein Mastermaterial, welches gleich mehrere Automatisierungsfunktionen bot: Zum einen erledigte es einen großen Teil der Photoshop-Arbeit wie das Reduzieren von Lichtinformation in der Albedo Map, zugleich extrahierte es Informationen für Specular und Roughness Maps aus den gebackenen Texturen.

In den Instanzen des Materials ist es damit möglich, die Assets mit Moos zu überdecken, welches sich automatisch an die richtigen Stellen platziert und die Seams zwischen den einzelnen Meshes verschwinden lässt. Das vereinfachte das Kitbashing deutlich. Weitere Funktionen im Material: Faden zu Detailtexturen, wenn sich der Spieler nähert, sowie Vertex-Farbe für Farbvariationen und Kontrolle über Effekte wie das Moos und einem gedithertem Blend zwischen dem Asset und dem Terrain, um auch hier Seams zu vermeiden. Die Bewegung der Vegetation passiert ebenfalls vollständig über den Shader. Hierfür wurden gleich zwei Systeme erstellt: Eines für simple Windanimation und ein komplexes System, welches aus der Vertexfarbe der Assets Informationen über den Pivot der einzelnen Elemente erhält und so komplexe Bewegung sowie Interaktion ermöglicht.

Texturen & Materialien

Bei der Textur-Gewinnung wählte das Team stets den schnellsten Weg. Insbesondere bei Tiling Textures, von denen mehrere Variationen benötigt wurden (überwuchert, abgenutzt, brandneu) sowie bei solchen, die für mehr künstlerische Freiheit von Hand erstellt werden mussten, wie etwa die Texturen für

Für die Ausleuchtung der umfangreichen Vegetation kam ein Hybrid-Modell zum Einsatz, in welchem statische Objekte in Lightmaps gebacken und die Vegetation mit dynamischen Schatten versehen wurde.

die Charaktere. Von Quixels Megascans ließen sich eine Vielzahl von Stein-, Boden- und Vegetations-Textur-Atlassen beziehen. „Da wir als Studenten finanziell limitiert sind, konnten wir nicht einfach wild drauflosskaufen, weshalb wir eines der vielen Asset Packs erwarben. Hier gibt es eine gute Variation an Assets für verschiedene Bereiche zur Auswahl. Besonders zu Wald und Prärie ist das Angebot sehr gut, und man bekommt wirklich viel für sein Geld, da jedes Asset mit Low-Poly, LODs, Texturen, Hi-Poly-Sculpt und sogar daraus gewonnen Brushes für ZBrush kommt", so das Team.

Vegetation findet man bei Megascans nur als Texturatlas, weshalb hier die Meshes von Hand erstellt werden mussten. Auch bei den anderen Assets achtete das Team darauf, sie auf den Look und Workflow anzupassen. Viele der Assets von Megascans sind nach unten offen, weshalb sie in ZBrush nachbearbeitet wurden, um ein geschlossenes, modular einsetzbares Mesh zu gewinnen. Die bereits erwähnten Materialien in Unreal sorgten im Anschluss dafür, dass die Assets zusammenpassen und sich nahtlos integrieren.

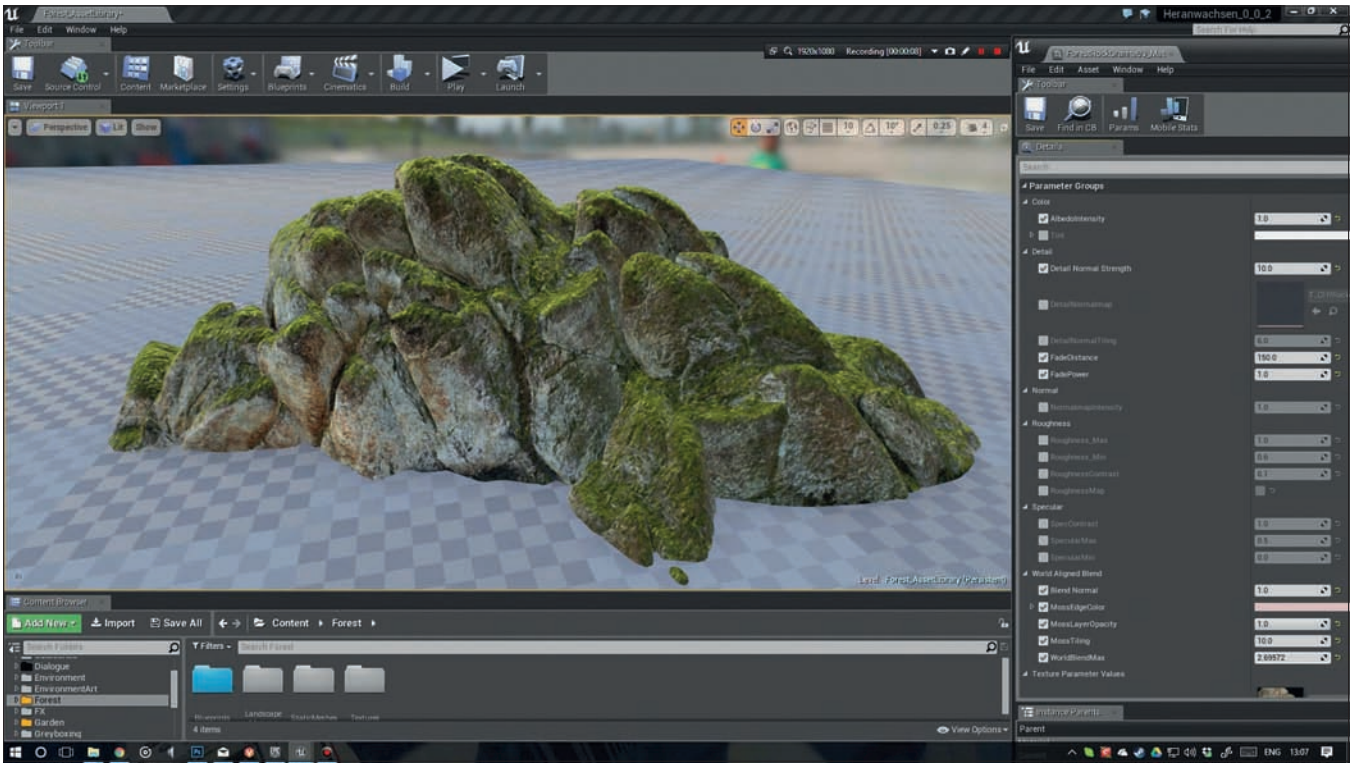
Modeling, Maya & MoCap

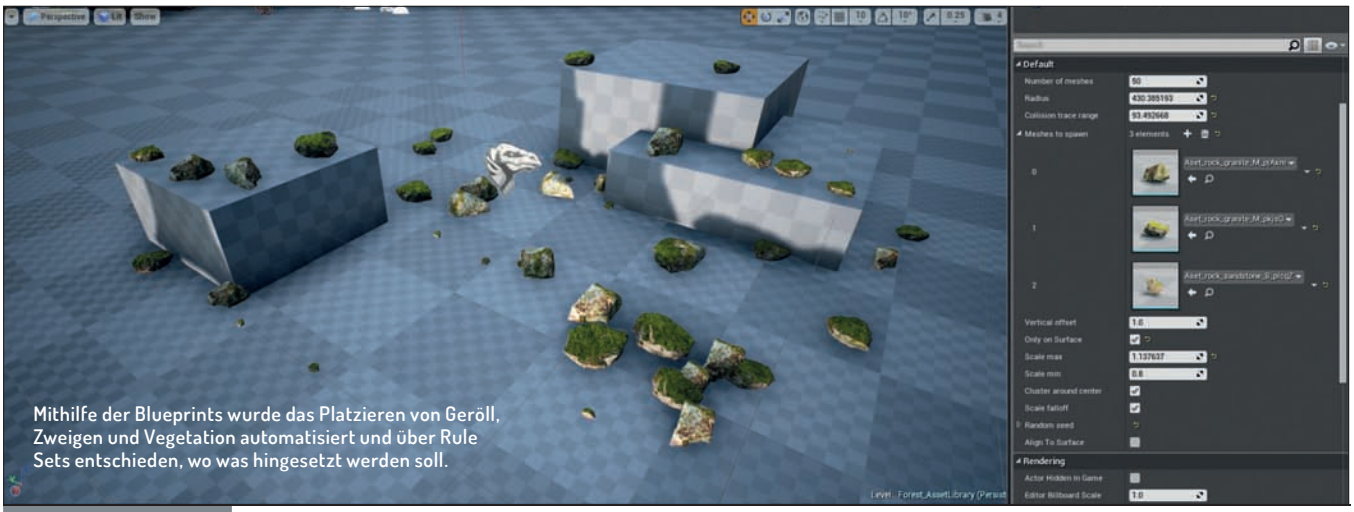
Für das Modeling benutzte Lukas Maya, in erster Linie aus Gewohnheit. Der Student besitzt dafür bereits eine stattliche Script-Bibliothek. Abseits des Projekts hat er seit einiger Zeit begonnen, auch mit Modo zu modellieren. Für „Sane" stand diese Pipeline aber noch nicht und der Erfahrungsschatz war zu gering, um sich darauf verlassen zu können. Für die Animation verwendete Christian neben Maya für die Keyframe-Animation das dem Perception Neuron beiliegende Programm „Axis", mit dem er MoCap-Daten generierte.

Das Mastermaterial ermöglichte, in den Instanzen des Materials die Assets mit Moos zu überdecken, welches sich automatisch an die richtigen Stellen platzierte und die Mesh-Seams verschwinden ließ.

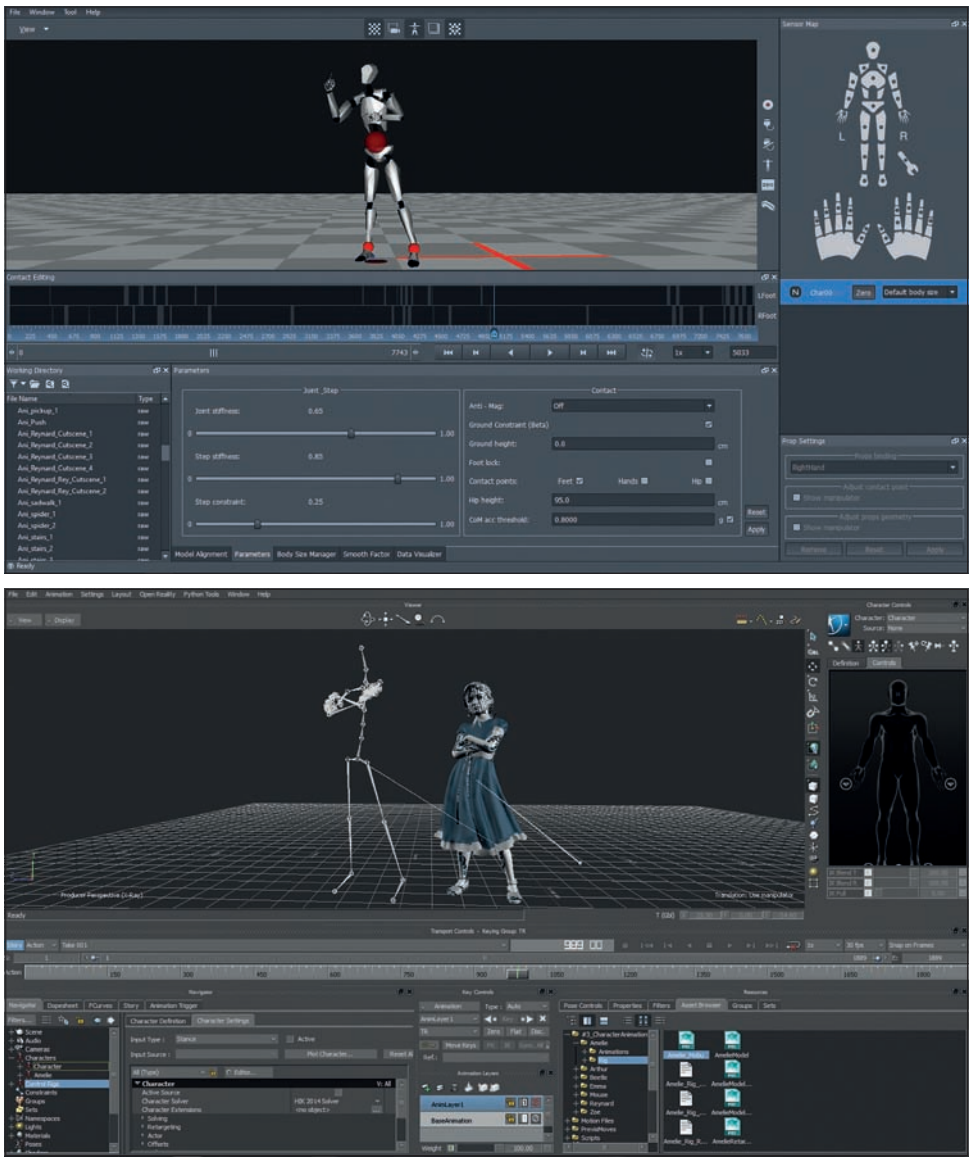
Da sich die Animationen noch in einem unfertigen Zustand befinden, handelt es sich zum derzeitigen Stand mehr um Platzhalter, welche dem finalen Look und insbesondere der Geschwindigkeit schon sehr nahe kommen. „Das Schöne an Neuron war definitiv, wie schnell wir erste Bewegungen auf unser Rig erhielten. Alles war zügig eingerichtet und die Software gut zu bedienen." Alles klappte bei diesem Workflow jedoch nicht: „Die Aufnahmen selbst sahen auf den ersten Blick ebenfalls sehr gut aus, bei genaueren Untersuchungen stellten wir aber einige Probleme fest. Wir verwendeten ein volles Set, bestehend aus 32 Neurons, sodass auch Fingerbewegungen erkannt wurden.

An einzelnen Fingern kam es dabei des Öfteren zu Problemen, die wir beim Aufnehmen nicht gleich entdeckten und die dann im Motion Builder korrigiert werden mussten. Auch die Kalibrierung ergab gelegentlich Fehler wie gebeugte Oberkörper oder fehlplatzierte Füße." Abhilfe schaffte hier nur





Mithilfe der Blueprints wurde das Platzieren von Geröll, Zweigen und Vegetation automatisiert und über Rule Sets entschieden, wo was hingesetzt werden soll.



Neben Keyframe-Animation in Maya nutzte das Team Motion Capturing. Zum Erfassen von schnelleren Animationsdaten kam die Axis-Software vom Perception-Neuron-System zum Einsatz, mit Motion Builder wurden Bewegungen gecleant.

das Entgegenwirken mit einer modifizierten Positionierung des Schauspielers während der Kalibrierung. Über eine längere Nutzungsdauer kam es vor, dass Gelenke sich verdrehten und alles neu kalibriert werden musste.

Die erhobenen Animationsdaten befanden sich nach der Aufnahme zunächst in der mitgelieferten Axis-Software auf einem eigenen Rig, von dort ließen sie sich in gängige MoCap-Formate wie .fbx oder .bvh exportieren und auf ein anderes Rig retargeten. Ebenfalls ist es möglich, Aufnahmen in der Software zu ändern, beispielsweise die Kontaktpunkte des Rigs.

Leider konnten die Rigs der menschlichen Charaktere im Zeitrahmen der Bachelorarbeit nicht fertiggestellt werden, wodurch das Skinning unvollständig ist und die Gesichter keine Animation besitzen. Der Plan war, bei letzterem mit Blendshapes zu arbeiten, was in frühen Tests mit Faceware in Kombination mit MoCap-Daten zu vielversprechenden Ergebnissen geführt hatte.

Der Ameliè-Charakter

Das Grunddesign für Ameliè stand relativ früh. Der Mädchen-Charakter sollte ein eher schlichtes, für ein Mädchen aus gutem Hause im Jahr 1919 passendes Design bekommen. Inspirationen boten hierfür Fotografien und Zeichnungen aus der Zeit. Ursprünglich war Ameliè deutlich jünger konzipiert und ihr Kleid bestand aus mehreren Schichten. Diese Designvariation schaffte es sogar bis zu einem finalen Modell in die Engine. Nach dem einsemestrigen Hiatus wurden jedoch einige Story-Elemente umgestellt und Ameliès Alter von 11 auf 13 Jahre geändert. Zudem wurde der Detailgrad der menschlichen Charaktere deutlich erhöht, woraufhin Ameliè sowie einige andere Charaktere einen kompletten Redesign-Prozess durchliefen.

Am Ende existierten von Ameliè fünf verschiedene Konzeptzeichnungen und zwei finale Ingame-Modelle sowie einige Revisionen des Hi-Poly-Sculpts, bis der finale Look gefunden war. Besonderheiten wie Dynamics für Haar und Kleid sorgten zudem für Anpassungen: Bei der Apex-Simulation des Kleides entschied das Team beispielsweise, die mehrfachen Schichten des ursprünglichen Designs gegen ein einlagiges Kleid zu tauschen. Die simulierte Kleidung in Marvelous diente zur schnellen Gewinnung eines realistischen Hi-Polys. Für das Ingame-Modell wurde hier nochmals eine Retopo angefertigt, die den Polycount reduzierte und den Edgeflow anpasste. Während Marvelous mittlerweile über gute Tools zum Reduzieren des Polycounts und Umwandeln der Tris in Quads verfügt, empfand das Team die Ergebnisse mit Meshlab oder von Hand nach wie vor als besser.

Ameliè durfte als Kind neben den anderen non-player Charakteren nicht proportional falsch wirken, deshalb achtete das Team schon früh auf einen realistischen Maßstab. Sie ist in der finalen Version 1,56 Meter groß, im Vergleich dazu hat der männliche Charakter Arthur eine Größe von 1,80 Meter. Das entspricht zwar nicht der Durchschnittsgröße der Leute in Frankreich um 1919, ergibt für das heutige Auge jedoch ein stimmiges Gesamtbild.

Die zeitliche Begrenzung von drei Semestern zwang die Artists dazu, den Workflow für die Charaktere zu optimieren: Die Pipeline bestand aus Hi-Poly-Sculpts in ZBrush und Marvelous Designer, dabei verzichteten die Studenten auf kleine Details wie Nähte oder Poren, da diese Arbeit später vom Unreal-Shader übernommen wurde. Auf den Sculpt folgte die Retopo, um das Ingame-Modell zu erhalten, sowie das anschließende Texture Painting im Substance Painter. Das Team setzte verschiedene Techniken ein, um den Retopologie-Prozess zu beschleunigen: Die Kleidung aus Marvelous erhielt perfekt flache UVs – so ließen sich komplexe Meshes aus einfachen Planes reprojizieren, welche anschließend zusammengenäht werden konnten.

Die bereits angesprochenen Shader in Unreal erledigten viel der Detailarbeit, außerdem waren sie universell auf alle Charaktere anwendbar. So besitzt der Cloth-Shader eine Funktion, die den zweiten UV-Channel des Meshes nutzt, um zu entdecken, an welchen Stellen Nähte sein sollen. Dies hat den Vorteil, dass man sie nicht für jeden Charakter in den Sculpt einarbeiten muss und sie zudem unabhängig von der Texturauflösung sind. Eine Technik, die auch für die Hauttextur und die Stoffmuster zum Einsatz kam. Mithilfe von gebackenen oder gemalten Masken fügten die Studenten den

Materialien zuletzt noch Wear & Tear sowie Misscoloration hinzu.

Fantasie-Wesen

Für die Demo plante das Team insgesamt fünf menschliche Charaktere und drei Fabelwesen, wobei es nur zwei Menschen und zwei Fabelwesen in die finale Version geschafft haben. Der Rest der Charaktere ist zwar visuell auf dem gleichen Stand wie diejenigen in der Demo, wurde aber aus Gameplay-Gründen gekürzt. „Da es sich um eine überschaubare Anzahl an Charakteren handelt, die sich auch visuell deutlich voneinander unterscheiden, haben wir uns nicht weiter um eine Automatisierung gekümmert, sondern extra Hi-Poly-Sculpts für jede der Figuren angefertigt. Für die Retopo verwendeten wir aber das Mesh des ersten fertigen Charakters wieder, indem wir es auf die anderen Hi-Polys projizierten", so die Studenten.

State Machines in der Unreal Engine zur Kontrolle der verschiedenen Animationsarten

Die Creatures aus Ameliès Fantasie entsprechen alle einer ihrer eigenen, unterdrückten Wesensarten. So tritt die Maus beispielsweise als listiger Dieb auf, was ihre rebellische Seite zeigt. Das Charakter-Design ergab sich aus der Funktion, welche die Figur erfüllen soll. Im Fall der Maus war dies das Erreichen kleiner, versteckter Orte. Die Studenten dazu: „Wie bei allen visuellen Elementen in unserem Game wurden auch hierfür Moodboards und Stylesheets erstellt, dazu schauten wir Referenzen von echten Tieren, Illustrationen von Fabelwesen sowie ein paar eigene Concepts an. Die meisten dürften für die Schnecke angefallen sein, da wir bei ihr den Charakterzug mehrmals änderten. Ani Kümmel, unsere Companion Designerin, zeichnete etwa fünf verschiedene Entwürfe der Schnecke, bis das finale Design stand."

Environments

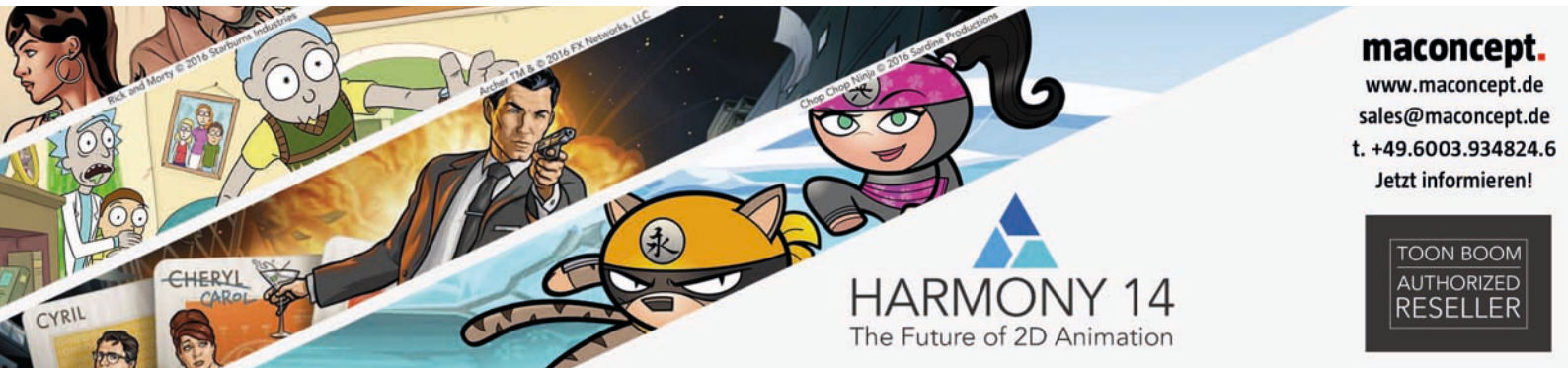
Bei der Erstellung der Landschaften gingen die Studenten ähnlich vor wie bei den Charakteren: Zunächst wurden die Gameplay-Spezifikationen abgesteckt und in Blaupausen festgehalten. Anschließend wurde eine Greybox-Version des Levels mit simplen Boxen in der Engine gebaut, um zu sehen, ob sich das Areal spielen lässt wie geplant oder ob Änderungen nötig sind. Auch ließen sich hier bereits Größenverhältnisse und Sichtlinien überprüfen. Sobald diese Grundvoraussetzungen gegeben waren, ging es an die Referenzmaterialsuche und das Erstellen von Moodboards und Stylesheets, was durch die Kreation eigener Konzeptzeichnungen abgeschlossen wurde.

Um die recht großen Spielareale in einem realistischen Zeitraum füllen zu können, legten die Studenten für die einzelnen Gebiete Bibliotheken mit modularen Assets an, die in möglichst wiederverwendbarer Form gestaltet sind. Für die Waldszene der Demo liegen etwa 40 individuelle Assets vor. Beim Set Dressing ging das Team in mehreren Passes vor: „Wir starteten mit der Levelbegrenzung und tauschten Stück für Stück die Greyboxes gegen finale Assets aus, wobei mit jedem Pass der Detailreichtum gesteigert wurde. Wichtig war hierbei, auch negativen Raum zu belassen. Unserer Erfahrung nach bietet es sich an, den vom Spieler zu beschreitenden Weg etwas simpler zu halten und die Details an den Wegrand zu platzieren. So wird der Spieler auf natürliche Weise durch das Level geführt."

Weitere Passes befassen sich mit dem Licht- und Post-Processing der Szene, welchem die Studenten viel Aufmerksamkeit schenkten. Gutes Licht kann eine tolle Szene optimieren, schlechtes kann sie ruinieren – zudem eignet es sich perfekt, um Points of Interest zu betonen, die den Spieler weiterleiten. Für die viele Vegetation wählten die Artists ein Hybrid-Modell zur Ausleuchtung der Environments, in welchem statische Objekte wie Felsen und Terrain in Lightmaps gebacken und die Vegetation mit dynamischen Schatten versehen wurde. Für kleine Bodenvegetation wurden performancesparende Contact Shadows eingesetzt, die nicht viel mehr als ein Post-Effekt sind, aber gute Ergebnisse liefern. So warfen selbst Gras, Klee und andere kleine Elemente Schatten.

Die Assets für das Environment gestalteten die Studenten hauptsächlich mit Maya, Substance Designer und ZBrush, zum Formen

Anzeige



des Terrains und Set Dressing kam die Unreal selbst zum Einsatz. World Machine fand für die großen Felsformationen im Hintergrund Verwendung. Für das finale Spiel konzipierten die Studenten neun Environments, wovon sich sechs im spielbaren Prototypen als Greybox befinden. Der Wald wurde mit finalen Visuals zum Schauplatz der Demo.

Die Fantasiewesen in der Welt von „Sane" repräsentieren jeweils einen von Ameliès Wesenszügen – so zeigt sich die Maus als geschickter Dieb, der überall hinkommt.

Orga ist alles

Zum Managen der Workfiles und des Projekts nutzten die Studenten Perforce auf einem eigenen Server. Auch hierbei bewährte sich Unreal in den Augen der Studenten: „Durch die in Unreal integrierte Perforce-Unterstützung zur Source Control war es uns möglich, mit mehreren Personen gleichzeitig am selben Projekt zu arbeiten. Das Schöne daran war auch, dass man immer zu alten Revisionen zurückkehren konnte und sich nicht um Backups kümmern musste." Neben den Projektdaten galt es aber auch das Arbeitspensum zu managen – was sich als deutlich schwieriger als gedacht erwies. Bei der Verwaltung der Aufgaben half Trello, zusätzlich fanden wöchentliche Scrums statt, bei denen das Team den aktuellen Stand besprach. Gegen Ende des Projekts gab es fast täglich Skype-Konferenzen, damit das Team sich auf dringliche Elemente fokussierte. Da die Studenten bereits von Anfang an eine klare Aufgabenteilung eingerichtet hatte, wusste jedes Teammitglied, woran es arbeiten musste und an wen man sich bei Fragen wenden konnte. Zusätzlich wurde ein umfangreiches Game Design Document (GDD) erstellt, in welchem man bei Unklarheiten bezüglich der Geschichte oder Gameplay-Elementen nachschlagen konnte.

Weg in die Praxis

Zwei der „Sane"-Teammitglieder, Lukas Kölz und Christian Müller, haben in diesem Semester mithilfe von „Sane" ihr BA-Studium abgeschlossen. Christian befindet sich derzeit noch auf der Suche nach einem Job im Bereich „Animation und Games", Lukas hat vor kurzem ein Jobangebot von Playground Games in England erhalten, wo er in Kürze anfangen wird. Sound Designer Philipp Grzemba und Komponist Tristan Blaskowitz machen derzeit noch ihr Masterstudium, welches sie erst in circa einem Jahr beenden. Ani Kümmel und Jonathan Waack befinden sich noch ganz am Anfang ihres BA-Studiums; Ani hat bereits entschieden, Charakter Design auch beruflich anzustreben.

Hinsichtlich einer Karriere in der Gamesbranche nach dem Studium ist sich das Team einig: „Der Einstieg ist schwer. Die deutsche Gamesbranche ist nach wie vor klein, vor allem auch, weil es an Geld fehlt. Staatliche Förderung ist in diesem Bereich kaum vorhanden – wer ein eigenes Studio aufbauen möchte, muss dies also aus eigener Tasche zahlen oder per Crowdfunding finanzieren. Einen Job in einer der großen Spieleschmieden zu landen, ist ebenfalls nicht leicht, da man dort natürlich vorwiegend Leute mit jahrelanger Erfahrung sucht und Junior-Positionen rar gesät sind. Unser Tipp ist, sich dennoch unentwegt bei den Spielefirmen zu bewerben (auch auf Praktikumsplätze), da eine einzige, kurze Anstellung oft reicht, um den Fuß in die Tür zu bekommen und Anschlussjobs zu finden. Während der Jobsuche und natürlich auch davor ist entscheidend, ein starkes Portfolio aufzubauen. Auch in Junior-Positionen kommt in dieser Industrie niemand rein, der nicht bereits ein Grundverständnis für den Job besitzt. Woher die Leute kommen und was für einen Schulabschluss sie haben, ist den meisten Studios unserer Erfahrung nach egal – und das ist auch gut so." **› mf**

Team „Sane"

Lukas Kölz – Projektlead & Environment Art
▷ lukaskoelz.artstation.com

Christian Müller – Animation & Locomotion

Ani Kümmel – Concept & Character Art
▷ minibabylizard.artstation.com

Jonathan Waack – UI Design
▷ jonwaack.artstation.com

Tristan Blaskowitz – Musik
▷ www.tristanblaskowitz.com

Philipp Grzemba – Sound Design
▷ soundcloud.com/sheyvan

Im Zuge des Projekts führte das Team einen Dev-Blog im Unreal-Forum und richtete aufgrund der hohen Nachfrage auch einen Youtube-Channel mit Videotutorials ein.

Devblog
▷ bit.ly/2mFI3LF

Youtube-Kanal
▷ bit.ly/2mlj6DI

Webseite „Sane"
▷ sanethegame.net

Die meisten Concepts fielen für die Schnecke an, da ihre Funktion für das Gameplay mehrmals verändert wurde.

Hochschule Mainz

Das Lehrangebot der Hochschule Mainz umfasst Bachelor- und Master-Studiengänge in Vollzeit und Teilzeit. Die Studienfächer sind breit gefächert und gliedern sich in die drei Fachbereiche: Technik, Gestaltung und Wirtschaft. Im Fach „Zeitbasierte Medien" kann an der Hochschule ein Bachelor- sowie Masterabschluss gemacht werden. Darüber hinaus gibt es zahlreiche Qualifizierungsmöglichkeiten: Studierende können in verschiedenen Bereichen wie IT oder Sprachen Weiterbildungskurse mit Inhalten außerhalb des regulären Curriculums besuchen.
▷ www.hs-mainz.de

DIGITAL PRODUCTION

DIGITAL IN DER APP LESEN

IM APP STORE UND BEI GOOGLE PLAY

- ALS EINZELAUSGABE ODER IM ABO
- REINTEXTMODUS ODER PDF-ANSICHT
- EINFACH DOWNLOADEN UND MOBIL LESEN

ANDROID APP ON Google play

Available on the App Store

Auch als digitales Upgrade für Printabonnenten:
www.digitalproduction.com/app