2017
6
ISSN 1433-2620 > B 43362 >> 21. Jahrgang >>> www.digitalproduction.com
Publiziert von DETAIL Business Information GmbH

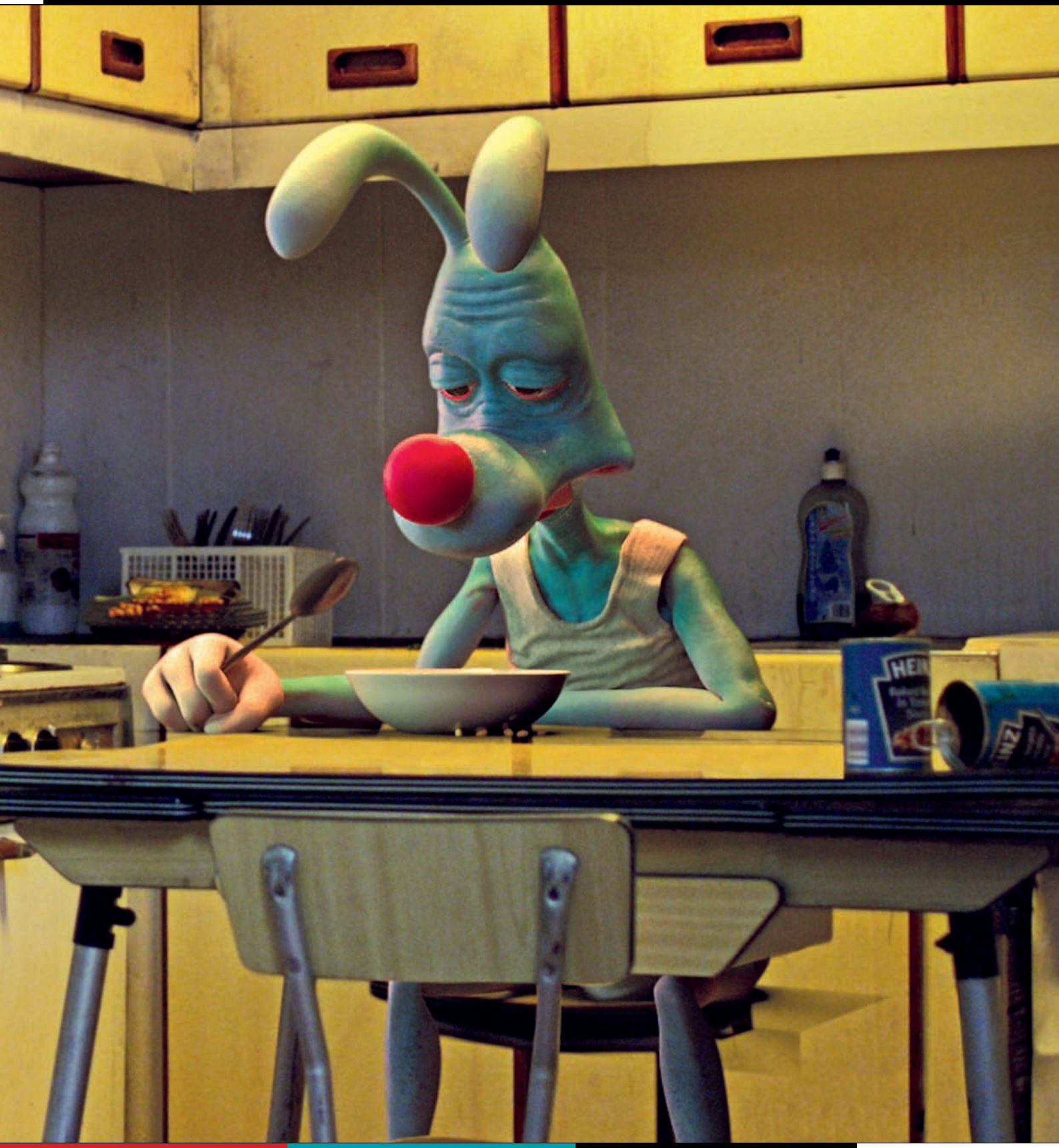
Deutschland	€	17,70
Österreich	€	19,–
Schweiz	sfr	23,–

DIGITAL PRODUCTION

DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPRODUKTION

NOVEMBER | DEZEMBER 06:2017



Heroes of 2017
Meet all the winners of the animago AWARD

Crowd Sim
Werde zum digitalen Herrscher der Heerscharen!

Toolbox!
Adobe AE 2017, Fusion 9, CityEngine und mehr

4 194336 217709 06 >

Der Golaem auf dem Prüfstand

Hinter einem Golem – das hebräische Wort für formlose Masse oder ungeschlachteter Mensch – versteht man im Allgemeinen ein stummes, menschenähnliches Wesen mit gewaltigen Ausmaßen und sehr viel Kraft. Das mystische Wesen soll in erster Linie Aufträge ausführen. Erschaffen wird ein Golem durch sagenumwobene Buchstabenmystik und Lehm, so erfährt man es zumindest in der jüdischen Literatur und Mystik ab dem frühen Mittelalter in Europa. Eine kleine Softwareschmiede hat sich den Namen dieser mystischen Figur zunutze gemacht und in leicht abgeänderter Form als Firmennamen genutzt. Das Produkt der Softwarefirme kennt man als das Maya Crowd Plug-in namens Golaem. Die neuste Version 6 wurde Ende Juni 2017 angekündigt, und was deren Tool dem Anwender bietet und es von Konkurrenten wie Miarmy unterscheidet, das zeigen wir hier.

von Rainer Duda

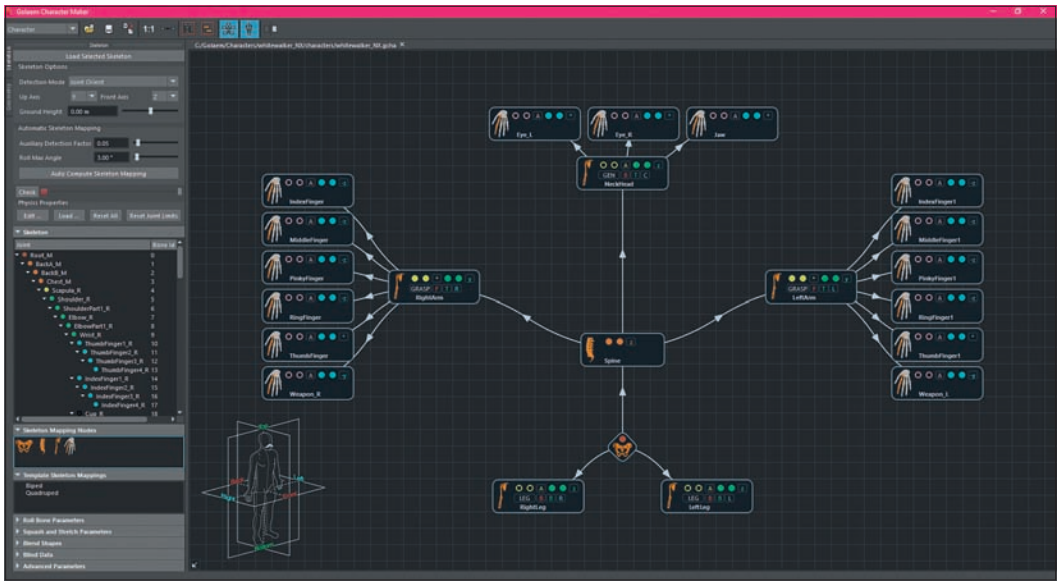
Große Massen an digitalen Charakteren bewegen und das am besten auf eine sehr einfache Weise und mit Möglichkeiten, die den künstlerischen Eingriff in die Simulation erlauben. Komplexität verschleiern, das Ganze verpackt in ein handliches UI mit einer gehörigen Portion Flexibilität. Keine leichte Aufgabe, der sich das Entwicklungsteam um Goleam 6 gewidmet hat, und doch ein Wunsch eines jeden Crowd TDs.

Bei Crowd-Systemen gibt es mehrere Knackpunkte, die ein Studio unter die Lupe nehmen muss, um zu wissen, ob die Neuanschaffung eines Tools gerechtfertigt ist oder eine eventuelle Anpassung der hauseigenen Tools wirtschaftlicher ist. Zunächst einmal geht es an die Prüfung, wie einfach selbsterstellte Charaktere ins System überführt werden können. Jedes Studio hat eine Charakter Pipeline und möchte dementsprechend die eigenen Charaktere in diesem Fall in Golaem 6 übertragen. Darauf aufbauend ist es wichtig, aus einem digitalen Charakter in aller Kürze eine große Masse an digitalen Charakteren erstellen zu können – quasi eine vollautomatisierte Lehmfabrik. Wichtig ist zudem der Boden unter den Füßen. Die symbolische Lehmarmee muss einer Oberfläche zugewiesen werden und sich dieser 1 zu 1 anpassen, bevor es daran geht, die Gesellen mit einem digitalen Gehirn auszustatten, damit sie einer bestimmten Logik folgen.

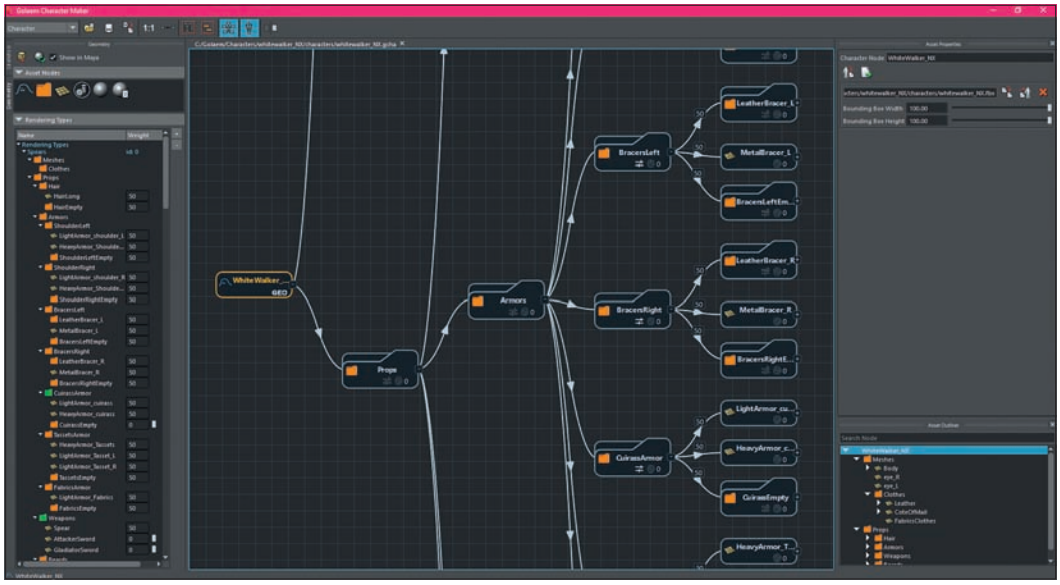
Dazu kommen noch gewollte und einfache Möglichkeiten, Variation in die Lehmarmee einzubringen im Sinne unterschiedlicher digitaler Charaktere inklusive Farb- beziehungsweise Materialvariationen. Die Lehmarmee wird mittels Accessoires oder Kleidung mit sogenannten Rich Details ausgestattet, damit dem Realismus nichts Großes mehr im Wege steht. Je einfacher die zuvor genannten Schritte von einer Software durchgeführt werden können, desto eher ist man als Hersteller einer solchen Software in der engeren Auswahl für den Produktionseinsatz beim Neukunden.

Eigene Charaktere einbringen

Wurde Golaem 6 erfolgreich installiert, dann findet sich ein Golaem Tab im Shelf oberhalb des Viewports. Das ist erstmal nichts Neues für ein Plug-in, das anwenderfreundlich aufgesetzt wurde. Besonderes Augenmerk sollte man auf die einzelnen Funktionen legen. Was man auf Anhieb wahrnimmt, ganz gleich welches Symbol man im Golaem Shelf betätigt, die Benutzeroberflächen der dahinterliegenden Editoren sind wohlwollend aufgebaut und in der Funktion fast selbsterklärend. Es wird zudem mit einer Vielzahl an visuellen Hilfsmitteln wie Icons, Building Blocks und weiteren visuellen Hilfen gearbeitet. Beim großen Konkurrenten Miarmy von Basefount schlägt man stattdessen lieber einen puristischen Weg seitens des UI ein mit deskriptiven Texten als Funktionen. Zurück zu Golaem 6. Eins der ersten Symbole im Shelf öffnet den Character Maker. Dahinter verbirgt sich ein Editor, der den



Golaem ermöglicht dem Anwender vollautomatisch ein Skeleton-Mapping vorzunehmen, was sehr gut funktioniert – zumindest mit Standard-Rigs wie Bipeds. Custom Rigs werden kaum erkannt, aber dafür darf man mit intuitiven Tools selbst ein Skeleton Mapping vornehmen und abspeichern.



Wenn ein Charakter mit den nötigen Accessoires ausgestattet ist und Sockets nebst notwendigem Aufbau aufbereitet wurden, dann lassen sich die Objekte samt Hierarchie dem Agenten zuweisen.

Import von digitalen Charakteren auf einfachem Weg in das Crowd Tool erlauben soll.

Möchte man einen eigenen Charakter in das System einbinden, dann genügt es zunächst, den Root Bone des Rigs auszuwählen und ihn dem Charakter Maker über einen Knopfdruck auf die dafür vorgesehen Schaltfläche mitzuteilen. Daraufhin erscheint ein neues Fenster, das dem Anwender ermöglicht selbst festzulegen, wohin der Charakter im Szenengraph schaut und welche Achse die Höhe angibt. Bei der Konkurrenz wird ein striktes Reglement festgelegt, das eingehalten werden soll. Immerhin lässt das System von Golaem etwas mehr Freiheit, wenn es darum geht, Custom Objekte wie Umgebungsobjekte, Fahrzeuge, Kreaturen oder Tiere in das Crowd-System einzupflegen.

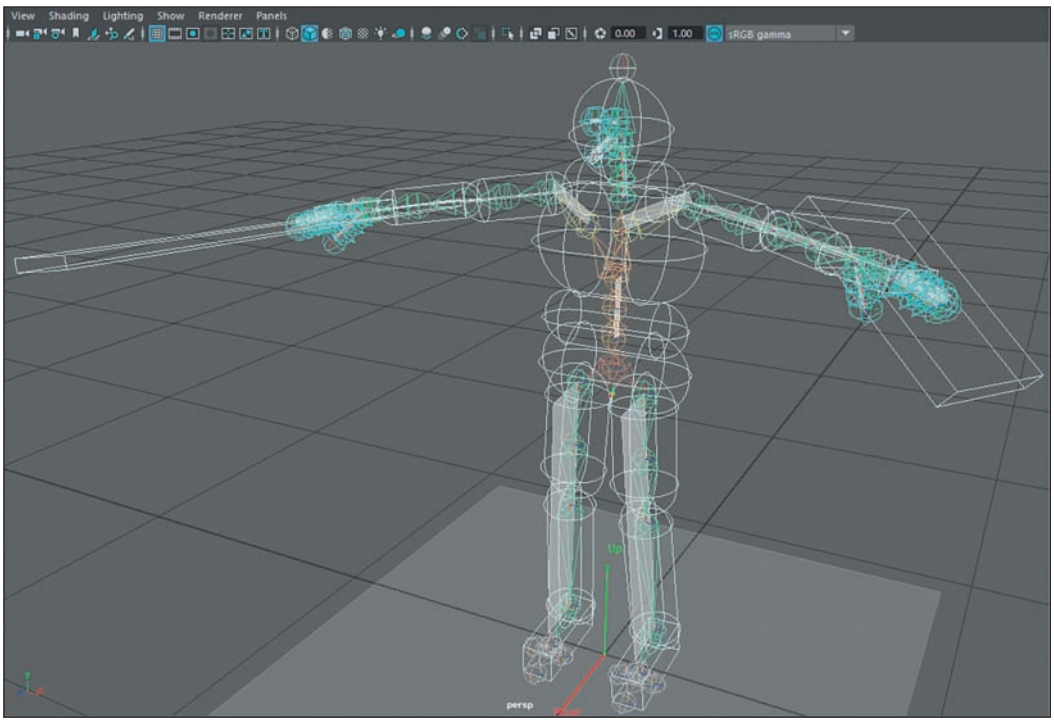
Die besagten visuellen Bausteine helfen dem Anwender zudem bei einem wichtigen Schritt, nämlich der Erstellung des Skeleton Mappings basierend auf dem Root Bone nebst der Child Joints. Entweder darf man selbst das Mapping mit selbsterklärenden Building Blocks aufsetzen, oder man greift auf einen automatischen Prozess zurück, der das Mapping vollautomatisch durchführt und im Graphen aufzeigt.

Nachdem das Sekeleton Mapping ordnungsgemäß durchgeführt wurde, geht es daran, den digitalen Charakter für die Simulation aufzubereiten. Bei der Konkurrenz Miarmy wird jeder Bone des digitalen Charakters mit einer Art Kollisions-Proxy versehen und als Proxy Box dargestellt. Für den Anwender heißt das, dass jede dieser Proxy Boxen in der Größe angepasst werden muss.

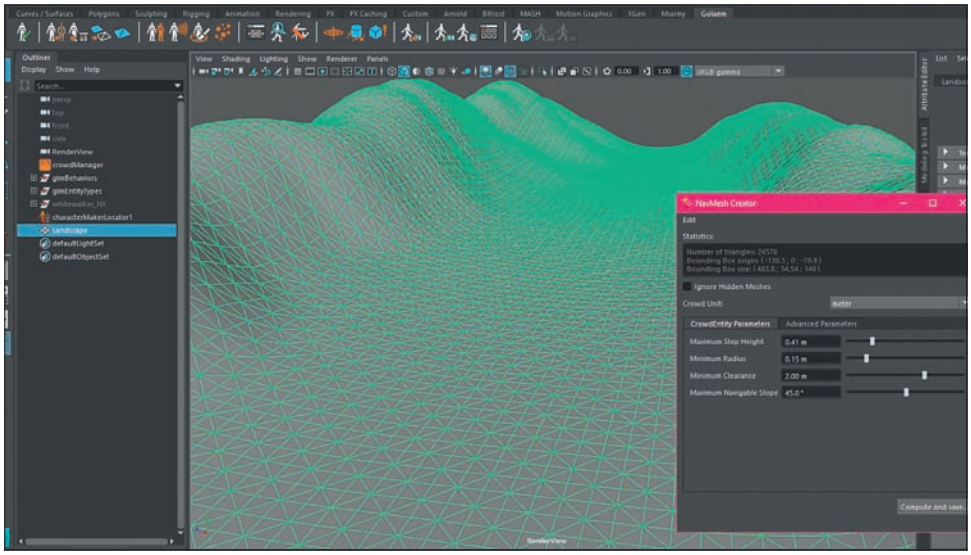
Bei Golaem 6 geht der Prozess ähnlich vonstatten, nur etwas schlanker möchte man meinen. Durch die Funktion „Edit Physics Properties" werden nur den wichtigsten Bones innerhalb des Skeleton Mappings Proxy-Objekte zugewiesen. Die Proportionen der Proxy-Objekte müssen von Hand angepasst werden. Interessant ist an dieser Stelle, dass man die Erscheinungsform der Proxy-Objekte selbst bestimmten kann. Ob es nun Boxen sind oder Kapseln, das darf man selbst entscheiden. Wobei es sich hier um mehr als nur einen kosmetischen Punkt handelt, vielmehr verändert sich das Kollisionsverhalten beziehungsweise die Kollisionserkennung und wirkt bei Kapseln etwas harmonischer. Hat man die Form der Wahl für sich gefunden, dann darf man in den Tab „Geometry" wechseln.

An dieser Stelle stellt man erneut fest, dass das Team von Golaem die Bereiche User Experience und Usability sehr ernst nimmt. Unter dem Tab „Geometry" wird dem Rig vereinfacht gesagt Geometrie zugewiesen.

Hierfür wird von Golaem 6 ein spezieller Aufbau benötigt. Man muss quasi wie beim Konkurrenten Miarmy einen im Miarmy-Kontext Original Agent – einen Mustercharakter – erstellen, eine Blaupause. Das allerdings nicht durch Selektion, Verschiebung und Betätigung von Funktionen, sondern bei Golaem wird mit einem übersichtlichen



Man darf als Anwender die Kollisionsgeometrie ebenfalls von Hand anpassen, sofern man mit dem Simulationsverhalten nicht zufrieden ist. Planes geben zusätzlich an, wo sich welche Joints biegen.



Wenn man eine Crowd in Bewegung setzen möchte, dann muss man dieser ein Stück Land unter die Beine geben. Hierfür kann vorhandene Terrain-Geometrie in ein Navigation Mesh konvertiert werden.

Graphen nebst selbsterklärender Building Blocks gearbeitet.

In der Mitte des Graphen findet sich ein vordefinierter Character Node, der mit sogenannten Asset Groups ausgestattet werden muss. Im einfachsten Fall besitzt man bereits einen Basis-Charakter, der mit unterschiedlichen Gruppen ausgestattet ist wie zum Beispiel Oberkörper, Kopfbedeckung, Beinkleid oder Schuhen. In diesen Gruppen befinden sich unterschiedliche Variationen der jeweiligen Objekte. Diese Ordner muss man in den Graphen übertragen und die Zuweisung zum Character Node vollziehen.

Von dieser Node-Blaupause lässt sich auf Knopfdruck ein Agenten-Typ definieren. So ist man als Anwender in der Lage, auf schnellem Wege Fussballmannschaften zu trennen nebst Schiri und Fangemeinde. Die Auswahl der jeweiligen Objekte beziehungsweise Accessories für die jeweiligen digitalen Charaktere findet über einen Schieberegler statt. Voraussetzung für die Funktionalität ist, dass die unterschiedlichen Accessoires mit einem Weighting versehen sind.

Wenn man sich bereits mit Crowd-Systemen auseinandergesetzt hat, dann weiß man, wie schnell sich doch kleinste Veränderungen in große Wirkung transformieren lassen. Gut, wenn gewollt – ist man in einer laufenden Produktion, wird es kniffelig. Zudem kommen noch künstlerische Kontrollstrukturen zur Verschleierung der zugrundeliegenden Komplexität. Das beste Beispiel hierfür ist der Character Maker, der mit relativ einfachen Schritten und einem verständlichen System eigene Charaktere einbindet.

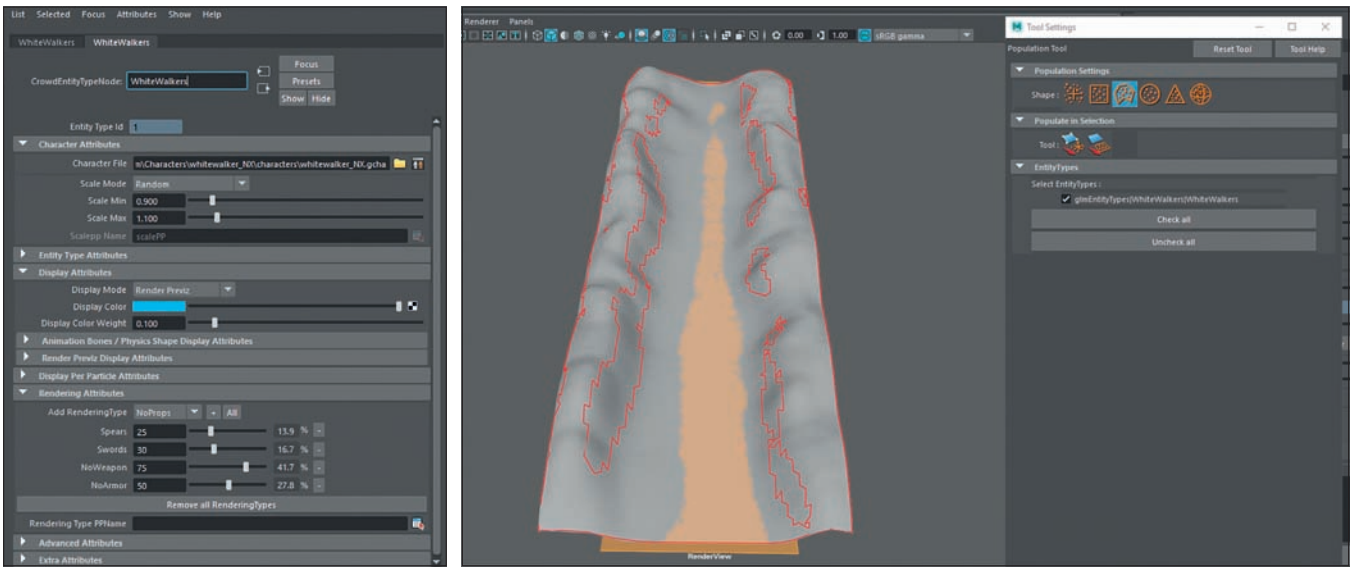
Bei der Konkurrenz Miarmy möchte man beim Anblick und der Arbeit darin meinen, dass rein die technischen Köpfe das Sagen beim Aufbau des Tools hatten beziehungsweise sie in der Überhand sind. Was mitunter auch gut sein kann und produktiv macht, wenn man mit puristischem Design arbeiten kann.

Betrachtet man erneut die Punkte Simulation, Komplexität, Accessoires und Flexibilität des Systems, dann gibt es einen wichtigen Punkt, den man wissen sollte: Die Geometrie wird nicht fest mit den Agenten in der Simulation verschweißt in Form eines Baking-Prozesses. Hat man eine Simulation erstellt und im Nachhinein hätte man lieber mehr digitale Charaktere mit einer Mütze, dann kann man ein paar Schritte zurückgehen und die jeweiligen Schieberegler beim Weighting innerhalb des Geometry Tabs im Character Maker anpassen, ohne die Simulation erneut zu berechnen – was ein erheblicher Pluspunkt ist. Gerade wenn man digitale Massen im komplexen Terrain bewegt.

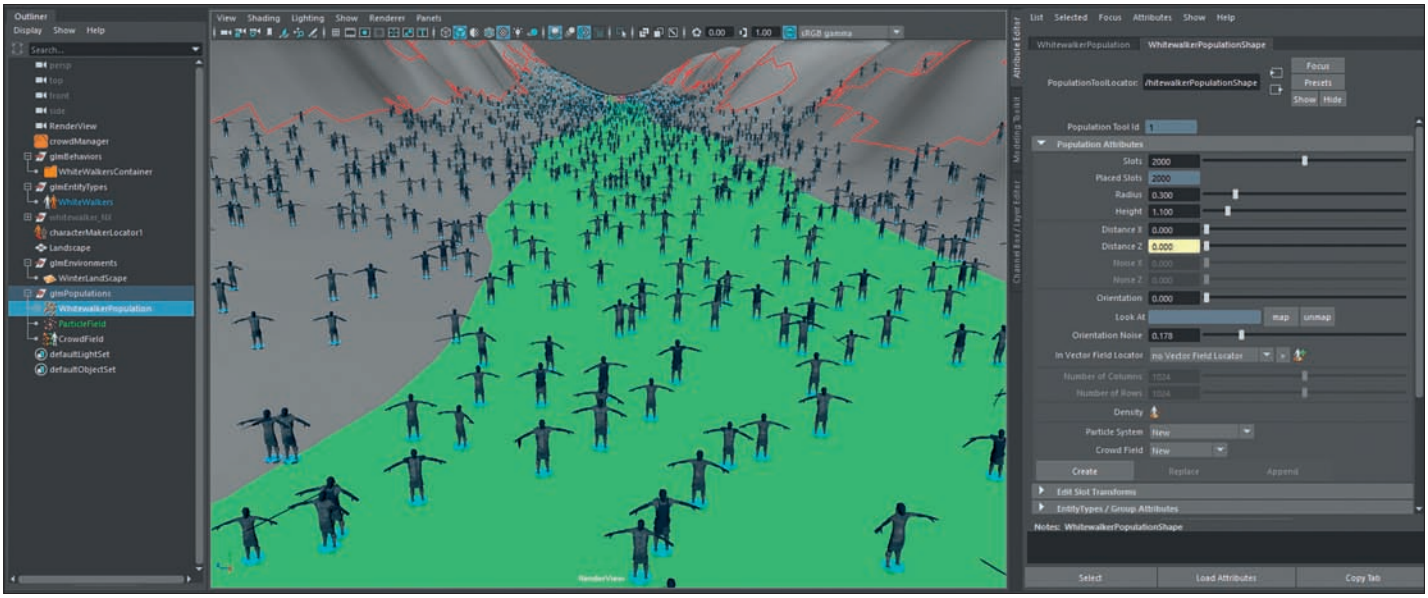
Auf die Plätze, fertig, los

Bevor man die digitale Armee in Bewegung setzen darf, müssen noch ein paar Vorarbeiten vonstatten gehen. Dazu zählt die Deklaration der Navigation Meshes. Man muss der Simulation mitteilen, was denn genau die Geometrie, das Terrain ist nebst Objekten, die als Hindernis dienen sollen.

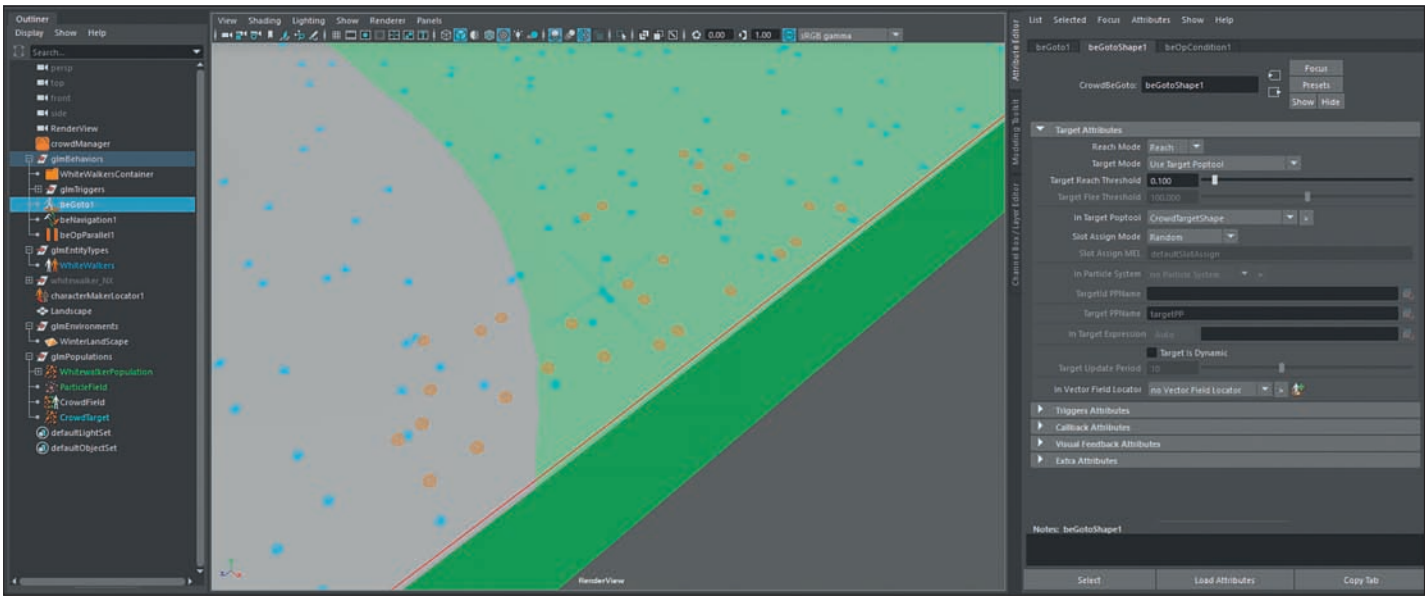
Dafür gibt es eine Funktion im Golaem Shelf namens „Nav Mesh Creator". Ein Editor, der selektierte Geometrie in die besagten Navigation Meshes konvertiert und zudem als neue Datei abspeichert und im Outliner ein Terrain-Objekt einfügt.



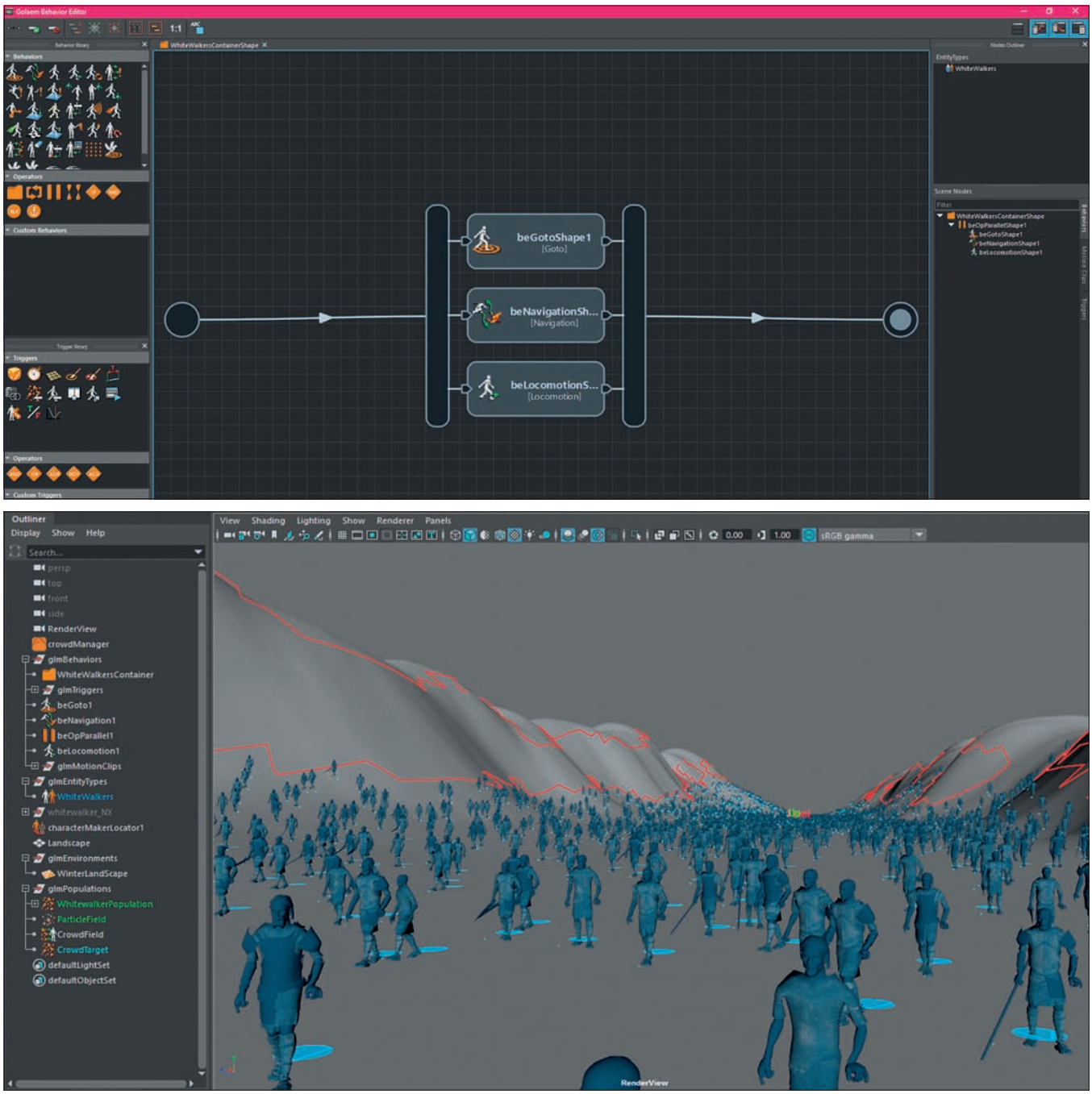
Es gibt unterschiedliche Möglichkeiten, wie man Agenten platzieren darf: automatisch, halb-automatisch per Pinsel oder mit selbstgezeichneten Grundformen.



Eine Reihe an selbsterklärenden Optionen ermöglicht es dem Anwender, mehr Variation in Bewegung, Positionierung und visuelles Auftreten zu bringen.



Die Platzierung der Agenten ist relativ einfach. Über ein Navigation Mesh, das einem Terrain zugewiesen wird, werden automatisch die Charaktere platziert.



Sobald eine Logik aufbereitet wurde, geht es ans Testen – mit einem Klick auf das Plus-Symbol. Die Charaktere sind farblich mit der zuvor festgelegten Farbe abgesetzt.

Zusätzlich wird dem Anwender ein Fenster mit Optionen für die physikalischen Eigenschaften beziehungsweise Laufeigenschaften bereitgestellt. Die Navigation-Mesh-Funktion gibt jedoch nur eine Indikation, welche Geometrie als statisch angesehen wird. Um nun aus bestimmten Objekten Hindernisse zu formen, muss eine weitere Funktion mit dem Namen „External Entity Locator" genutzt werden. Die Fläche des Terrains wird rot umrandet, während die Hindernisse exkludiert sind.

Nun wird es spannend: Man möchte meinen, die größte Arbeit sei getan und man könne per Knopfdruck eine ganze Armee durch das Terrain jagen. Ganz so einfach ist es nicht. Zunächst muss ein sogenanntes Entity-Type-Objekt erstellt werden, in das der zuvor exportierte Charakter vom Charakter Maker geladen werden muss.

Man könnte sagen, dass man den digitalen Pinsel des Golaem Population Tools in eine Entity taucht und daraufhin die Agenten auf das Terrain pinselt. Bei Miarmy wird eine Art Placement Node in die Szene eingefügt.

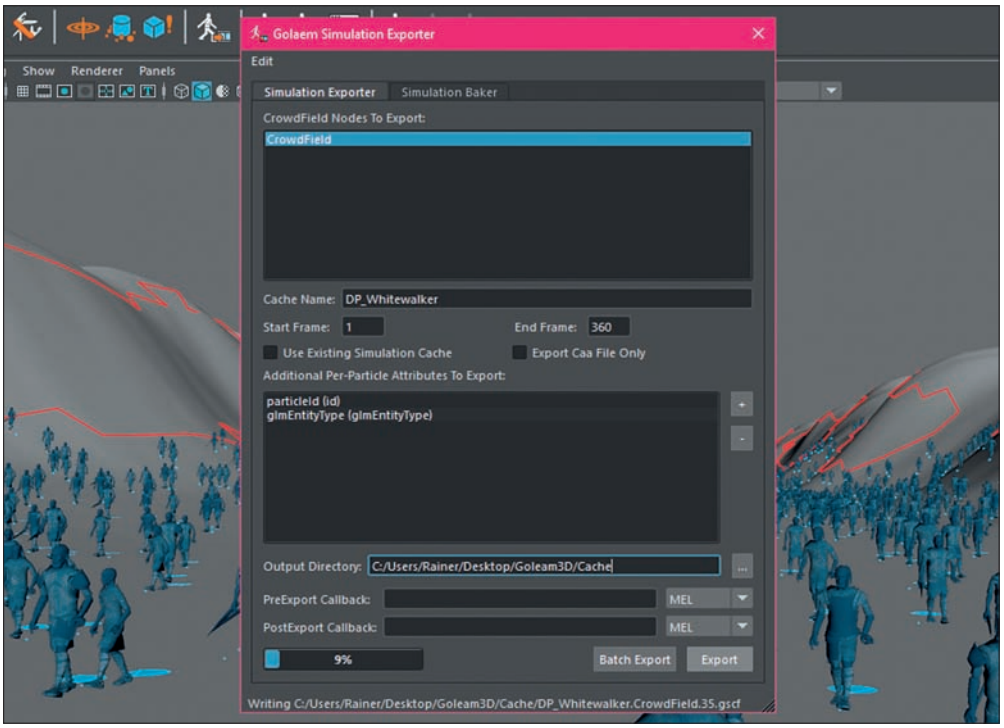
Der Placement Node lässt sich durch Attribute so beeinflussen, dass unterschiedliche Charaktere mit unterschiedlichen Variationen in unterschiedliche Richtungen laufen. Bei Golaem ist das Prinzip ähnlich, jedoch mit fast spielerischen sowie künstlerischen Hilfsmitteln ausgestattet. Anstatt ein Placement Node einzufügen, wird bei Golaem zum Beispiel ein Polygon auf das Terrain gezeichnet. Innerhalb des Polygons werden dann wie bei Miarmy unterschiedliche Attribute genutzt, um der digitalen Armee Raum zu schaffen.

Ganz gleich wie man das Placement-Polygon bewegt, es passt sich dem Terrain an und lässt nur an den vermerkten Stellen Agenten zu. Die zuvor genannten Entity Types für die Simulation, also die Agenten basierend auf den Blaupausen des Character Makers, können ebenfalls per Schieberegler miteinander vermischt werden. Stichwort: Weighting.

Ist man mit der Platzierung der Entity Types zufrieden, dann darf man den Create Button betätigen. Daraufhin wird ein Partikelobjekt nebst Crowd-Field im Outliner erstellt. Letzteres wandelt bildlich gesprochen die Partikel in Agenten um. Der Einfachheit halber darf man die Funktionen über das Shelf aufrufen, oder man geht den Weg über das Menü im Population Tool Channel beziehungsweise Attribute Editor. Betätigt man nun die Zeitleiste, dann sollte nach ein paar Frames die Simulation samt aller Agenten sichtbar sein.

Der Prozess, um die zuvor erstellten digitalen Charaktere auf ein Terrain zu platzieren und in Bewegung zu setzen, wirkt im Gegensatz zu Miarmy langatmiger, da die einzelnen Schritte und Zuweisungen halbautomatisch vonstatten gehen und der Anwender darauf achten muss. Interessant ist bei Golaem, dass die Erscheinung der einzelnen Agenten in der Simulation per Menü geändert werden kann. Dafür muss man lediglich das jeweilige Entity Type selektieren und im Attribute Editor unter dem Feld „Display Attributes" den Display Type auf „Physics Shape" ändern. Die Veränderung der Ansicht ist essentiell in einer Simulation, um das Verhalten und die Bewegungen der Agenten im Detail begutachten zu können, damit in einer professionellen Medienproduktion so wenig wie nötig nachgearbeitet werden muss. Wer dem Ganzen noch mehr Detail abverlangt, der darf in den Einstellungen die zuvor im eigenen digitalen Charakter hinterlegte Geometrie anstatt des Physics Bodys einblenden. Somit sieht man im Viewport die eigentlichen Charaktere anstatt Dummies. Bei der Konkurrenz wird die Geometrie der Charaktere erst zum Renderzeitpunkt sichtbar, wobei das natürlich ebenfalls eine Frage des Workflows ist. Für einen Crowd TD steht in erster Linie eine korrekt verlaufende Simulation im Vordergrund, bevor Geometrie und vor allem hochaufgelöste Geometrie ins Spiel kommt.

Der Vorteil bei Golaem ist, dass man direkt prüfen kann, ob Blendshapes funktionieren, ob die Einstellungen seitens der Variation stimmig sind und das Gesamtbild wie gewünscht dargestellt wird. Andernfalls lassen sich vor dem Rendering und dem Erstellen des Geo-Caches noch Änderungen durchführen. Bei einem Punkt darf man sich jedoch nicht verunsichern lassen: Was man in der Render-Previs-Einstellung nicht sieht, sind die Variationen im Bereich des Shadings.



Bevor es ans Rendering geht, muss erstmal ein Simulation Cache erstellt werden. Das macht zudem die Betrachtung der Bewegungen im Viewport angenehmer.

Logik ins Spiel bringen

Um der digitalen Armee nun mitzuteilen, wie sie sich verhalten soll, kommt ein neuer Editor namens „Behavior Editor" ins Spiel – zu erkennen am Graphen-Symbol im Golaem Shelf. Erfreulich ist, dass der Behavior Editor ebenfalls selbsterklärend aufgebaut ist. Im Graph befindet sich ein Start- sowie Endpunkt, beide miteinander verbunden. Sie wirken richtungsweisend von links nach rechts und drum herum sowie dazwischen lassen sich sämtliche links angelegte Logikbausteine einfügen. Man merkt an dieser Stelle, dass einige Bausteine heller wirken als andere. Das liegt daran, dass man die Nodes für den jeweiligen Entity Type aktivieren muss. Dafür müssen die Logikbausteine zwischen Start- und Endpunkt gerückt werden.

Zu berücksichtigen gilt, dass zuvor unterschiedliche Entity Types erstellt wurden. Rechts oben im Behavior Editor sind die Entity Types abgelegt, und jeder hat seinen eigenen Logik-Graphen. Der Aufbau eines Graphen ist an sich eine spannende Sache. Im Grunde äußerst komplex, aber mithilfe der selbsterklärenden Bausteine, die in unterschiedliche Bereiche unterteilt sind, wirkt der Aufbau einer Logik fast spielerisch. Der erste Bereich gilt dem Verhalten von einfachem Navigieren. Über Kollisionserkennung bis hin zu Ragdoll-Animationen sind insgesamt 24 Bausteine hinterlegt. Die Verhaltensbausteine lassen sich mit logischen Operatoren verbinden, die im zweiten Bereich hinterlegt sind. Der dritte Bereich nennt sich „Custom Behaviors" und dient als Speicherort für die eigene Logik der jeweiligen Entity Types, sobald man eine Logikschaltung speichert. Im vierten Feld befinden sich sogenannte Trigger-Objekte mit denen unterschiedliche Verhalten ausgelöst werden, falls zum Beispiel ein bestimmtes Objekt gesichtet wird oder anderweitige Kontrollobjekte mit einem Trigger versehen sind, um die digitale Armee zu steuern.

Was Golaem gut gelöst hat, ist der Aufbau der Logik der unterschiedlichen Entity Types. Bei Miarmy wird mit deskriptiven Texten als Menüzeile sowie puristischen Menüs gearbeitet. Das ist sicher ein Punkt, der den einen oder anderen kreativen Kopf abschreckt und die Fehlersuche in der Logik zeitlich in die Länge zieht. Allen voran, weil die Informationen nicht alle sofort ersichtlich sind wie in einem selbsterklärenden Graphen mit Blöcken samt deskriptiven Icons. Aber man muss auch sagen, dass die Logik für alle Entity Types erstellt werden muss, falls mehrere vorhanden sind. Die unterschiedlichen Typen sehen sich quasi nicht direkt und gehen nicht davon aus, dass sie miteinander interagieren. Zumindest muss man einen Logikbaustein einfügen, der eine Kollision darstellt beziehungsweise die Kollisionserkennung aktiviert.

Wie sieht es jedoch aus mit unterschiedlichen Animationen? Eine Frage die sicher jeden Crowd TD und Animator beschäftigt. Natürlich müssen die unterschiedlichen

Animationen innerhalb des Behavior Editors in einen Locomotion-Baustein eingeladen werden, damit sie ansprechbar sind. Das heißt, dass davor Motion Clips erstellt werden müssen, ganz wie bei Miarmy.

Blickt man nach den Motion Clips in eine etwas gezieltere Logik, zum Beispiel um die digitale Armee in Bewegung mit einer Art Ziel zu setzen, dann sollte man sich die Bausteine etwas genauer ansehen und die Target-Funktion unter die Lupe nehmen. Anstatt ein Null-Objekt als Ziel anzugeben, darf man auf eine virtuelle Zielscheibe zurückgreifen, die zudem erlaubt, Offset-Punkte zu definieren, damit nicht alle Charaktere auf denselben Punkt zuläuft. Wahlweise darf es sich zudem um ein dynamisches Target handeln, das mit Keyframes versehen ist. Für grundlegende und komplexe Simulationen sind somit alle Bausteine vordefiniert. So auch bei Miarmy, nur eben in Textform innerhalb eines puristischen Editors. Man könnte quasi sagen, beide Tools unterliegen einfach unterschiedlichen Geschmäckern.

Wichtig bei Crowd-Systemen ist außerdem das Feedback der Agenten, sozusagen eine Art Erkennung und Aufzeigen der Verhaltensmuster. Bei Miarmy selektiert man Agenten und erhält direkt ein kleines Menü mit wichtigen Informationen. Bei Golaem gibt es einen ganz eigenen Editor für diesen Prozess. Die Rede ist vom Crowd Visual Feedback Editor, der es ermöglicht, eine Vielzahl von Agenten- und Simulationsinformationen an einer Stelle preiszugeben. Nun kommt ein weiterer wichtiger Aspekt ins Spiel. Ist man mit der Simulation zufrieden, möchte man das Ergebnis auch rendern.

Dafür muss zunächst die zugrundeliegende Simulation exportiert werden. Würde man an dieser Stelle rendern, ohne zu exportieren, dann würde man nur eine Layout-Geometrie sehen. Dafür gibt es ebenfalls eine automatische Funktion namens „Golaem Crowd Simulation Exporter" – zu finden im Shelf als Symbol eines weißen Charakter der nach rechts läuft nebst rotem Pfeil. Wurde das gewünschte Crowd Field ausgewählt und Frames, Output Pfad und Format eingestellt, dann darf die Simulation exportiert werden. Beim Format kommt es darauf an, ob man in Maya oder außerhalb in einer hauseigenen Software rendern möchte. Wenn in Maya, dann bietet es sich an, das native Cache-Format zu nutzen. Wenn außerhalb, dann sollte man in ein gängiges Format exportieren mit dem Hintergedanken, genügend Speicher vorrätig zu haben.

Wie bei Miarmy kann man bei Golaem den Simulation Cache in die Szene einladen und die Simulation in Echtzeit im Viewport begutachten. Zum Einladen des Caches gibt es zwei Symbole weiter vom Export-Fenster die Möglichkeit, den Cache einzuladen. Sieht man den Cache im Viewport und ist die Zeitleiste grün hinterlegt, dann spielt man den Cache ab, ohne Veränderungen an der Simulation vornehmen zu dürfen. Um in den Simulationsmodus zurückzukehren, gibt es das ganz rechte Symbol im Shelf. Es gibt je nach Lizenz noch die Möglichkeit, den Simulation Cache mit einem Deformer zu belegen und den Cache zu layouten. Wirklich gut gelöst ist, dass sich die Agenten dem Terrain anpassen, selbst wenn es sich nur um einen Cache handelt. Der Cache bleibt zudem unberührt, es handelt sich nur um quasi-lokale Overrides, die reversibel sind. Gleiches gilt für die unterschiedlichen Accessoires, die von Hand ausgetauscht werden dürfen.

Der letzte Schritt vor dem Rendering ist die Erstellung eines Render Proxies – AI Stand-in, ganz wie bei Miarmy. Daraufhin darf man den Render View öffnen und die digitale Armee in voller Pracht bewundern.

Nun fragt man sich sicher, was so viel Flexibilität und Produktivität in einem Tool für den Anwender kostet.

Kostenfaktor

Es gibt sehr erfreuliche Informationen im Bezug auf die Kosten. Zunächst gibt es eine kostenfreie PLE-Version mit allen Features und ohne Zeitlimit. Für akademische Einrichtungen gibt es ebenfalls kostenfreie Lizenzen inklusive Floating Lizenzen, unbegrenzte Anzahl von Render Nodes und sämtlichen Upgrades gefolgt von Support und direkt zu benutzenden Charakteren und deren Animationen. Grundsätzlich ein sehr großer Pluspunkt, schließlich reicht es nicht, eine hervorragende Softwarelösung zu besitzen, man benötigt auch Anwender dafür. Und wenn den Anwendern für die Lernphase Unterstützung und Lizenzen kostenfrei geboten werden, dann ist das ein großes Stück. Passend, dass Produkte von Autodesk und gerade Maya ebenfalls für den akademischen Bereich kostenfrei verfügbar ist. Klar, man muss sind Wasserzeichen leben, aber für die Lernphase ist das zu verschmerzen. Bleibt die Frage offen, wie es für Freiberufler und Studios aussieht. Im Fokus liegt hier die gesamte Suite sowie Populations- und Layout-Tools. Hier wird dementsprechend zur Kasse gebeten. Eine permanente Lizenz kostet 7.000 Euro. Für Support und Upgrades werden nochmals 1.750 Euro pro Jahr fällig. Klar, die Weiterentwicklung der Software muss ja finanziert werden und die vorliegenden Projekte sollten das Tool plus Maintenance amortisieren. Man muss selbst eruieren, wie oft man Crowd Tools benötigt. Um das Bild zu komplettieren, eine Site-Lizenz kostet 43.000 Euro für die erste Site und 21.500 Euro für folgende Sites mit 10.750 Euro pro Jahr Maintenance für die erste Site und 5.375 Euro für die weiteren. Ein stolzer Preis für ein stolzes Tool. Immerhin bietet Golaem Anwendern und Studios ebenso die Möglichkeit, das Tool zu mieten. Entweder für 650 Euro pro Tag oder 1.500 Euro pro Quartal für kleine Teams und Freiberufler.

Nun stellt sich zudem die Frage, welches der Tools für den Einsatz im Bereich Crowd-Animationen besser geeignet ist: Miarmy oder das hier aufgezeigte Tool Golaem, beides Plug-in-Lösungen für Maya. Die Frage nach besser oder schlechter stellt sich bei beiden Tools nicht, vielmehr sollte man an dem Punkt der Auswahl auf die eigene Pipeline achten und wie flexibel sich die verhält. Wenn man mit einem Team arbeitet, das technisch überaus versiert ist und mit Grafiken innerhalb von groß aufgesetzten Graphen nichts so wirklich anfangen kann, dann sollte man sicherlich den Weg von Miarmy wählen. Wer einen hochgradig künstlerischen Einfluss innerhalb einer Crowd-Pipeline als wünschenswert erachtet, der kommt an Golaem 6 nicht vorbei. Was noch aufgefallen ist, gerade bei Golaem, ist der Fakt, dass immer wieder von einem Partikelsystem geredet wird und auch ein Partikelobjekt im Outliner angelegt wird. Wer sich mit intelligenten Scattering-Systemen auseinandergesetzt hat, der weiß, dass sich das wunderbar zweckentfremden lässt – ganz unabhängig von der Anwendung digitaler Charaktere – um große Szenen mit unterschiedlichen Populationen von diversen Umgebungsobjekten zu versehen.

Tauscht man die digitalen Charaktere also aus, dann hat man zugleich ein intelligentes Scattering Tool. Das Team um Golaem 6 macht keinen Hehl daraus, dass man das System ebenfalls zur Population von Objekten nutzen darf. Bei Miarmy ist etwas Handarbeit angesagt. Man muss das Original-Agent-System mit der zugrundeliegenden Methodik etwas grob zweckentfremden. Beispielhafter gesprochen müssen Umgebungsobjekte vorab mit wenigstens einem Bone versehen werden, um als Original Agent weiterverarbeitet zu werden.

Ab dann darf man sich der Layout Tools bedienen. Das Fazit ist kurz und knapp. Wer das Prinzip eines Crowd-Systems verstanden hat, ist blitzschnell in der Lage, Golaem und Miarmy für die eigenen Zwecke einzusetzen. **›ei**