

SPIELER

VON CHRISTIAN TRAPP

Die in diesem Band vorgestellten Aktanten des Game Design-Prozesses haben gemeinsam, dass sie (im Wesentlichen) vor der Veröffentlichung eines Spiels ihre Handlungsmacht bzw. Agency entfalten. Der Spieler hingegen konsumiert ein Spiel i.d.R. erst nach dem Release.¹ Auf den eigentlichen Entwicklungsprozess hat er also anscheinend keinen Einfluss. Allerdings zeigt das Phänomen des Modding zahlreiche Beispiele für eine – nachträgliche – Veränderung eines Spiels durch den Spieler. So gibt es viele Möglichkeiten, durch Anpassungen von Level- und Charakterdesign sowie teilweise auch von Regelwerken ein Spiel völlig umzugestalten. Ob ein Spieler damit zum Aktanten des Game Design-Prozesses wird, soll im Folgenden diskutiert werden.

Modding bezeichnet eine teilweise oder vollständige Veränderung von Spielinhalten. Modifikationen der Levelgestaltung, der Figuren oder des Regelwerks sind nur einige Anpassungsmöglichkeiten, die ein Spieler mit den nötigen Fähigkeiten und Werkzeugen an einem Spiel vornehmen kann. Von einer Partial Conversation spricht man in diesem Zusammenhang, wenn das Basisspiel weitgehend erhalten bleibt und nur Teile, bspw. das Figurendesign durch sogenannte Skinpacks oder die grafische Gestaltung der Spielwelt durch modifizierte Texturen, verändert werden. Mit dem Begriff der Total Conversation wird eine vollständige Umgestaltung der Spielwelt und ggf. auch des Regelwerks bezeichnet. Das Basisspiel ist hier nach der Bearbeitung i.d.R. nicht mehr zu erkennen. Teilweise wird nur die gleiche Game-Engine wie im Originalspiel als Basis für die Total Conversation genutzt.²

Ein Spieler benötigt für solche umfangreichen Änderungen des Spiels, je nach Grad der Modifikation, Wissen in Bild- und Grafikbearbeitung sowie evtl. Programmierkenntnisse. Während sich in den Anfängen des Modding kleinere Änderungen mit einem Hex-Editor realisieren ließen³, sind für umfangreichere Modifikationen Software-Bibliotheken und spezielle Editoren nötig. Für einige Spiele stellen die Hersteller den Modding-Communities diese Editoren, häufig Level-Editoren oder SDKs (Software Development Kits) zur Verfügung. Das DCK (*DOOM Construction Kit*) ist ein frühes Beispiel für einen Level-Editor.⁴ Valve bietet für *HALF-LIFE 2*-Besitzer (2004) ein kostenlos erhältliches SDK zum Download

1 Ausnahmen bilden öffentliche Betatests, die jedoch nicht für alle Spiele angeboten und auch nicht von allen Spielern genutzt werden können. Vgl. hierzu den Aktanten **Beta-test** in diesem Band.

2 Vgl. Kücklich: »Modding, Cheating und Skinning«, S. 4.

3 Vgl. ebd.

4 Vgl. Au: »Triumph of the Mod«, o.S.

an.⁵ Neben Bibliotheken und Quellcode enthält das SDK auch den *Hammer-Editor* zum Erstellen neuer Level für *HALF-LIFE 2*. Die grafische Benutzeroberfläche (Abb. 1) bietet Spielern eine vergleichsweise zugängliche Möglichkeit zur Kreation neuer Level. Diese Arbeit ist dank des Editors einfacher, als ein Level ausschließlich auf Basis von abstraktem Programmcode zu erstellen. Trotzdem ist oft noch eine langwierige Einarbeitungszeit nötig, wenn der Nutzer kein Vorwissen in 3D-Konstruktionsprogrammen hat, um ein neues, spielbares, Level zu erstellen.

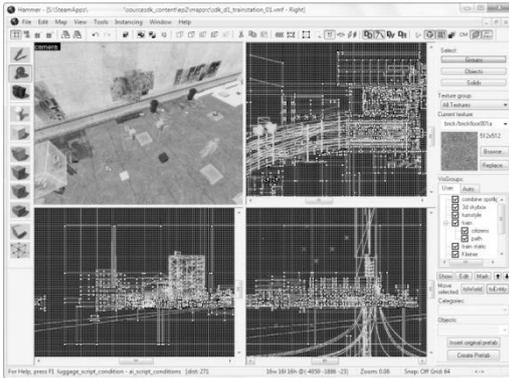


Abb. 1: *Hammer-Editor*.⁶



Abb. 2: *LITTLEBIGPLANET-Editor*.⁷

Eine neuere Entwicklung der Spiele-Industrie ist der Level-Editor als ›Feature‹. Das Beispiel *LITTLE BIG PLANET* (Media Molecule, 2008) hebt seinen Level-Editor (Abb. 2) als zentrales Element des Spiels hervor.⁸ Im Vergleich zum *Hammer-Editor* ist der Editor in *LITTLE BIG PLANET* vollständig integriert. Die Bedienung ist intuitiv und auch für Laien sehr schnell erlernbar. Damit verbunden sind die eingeschränkten Möglichkeiten, die der *LITTLE BIG PLANET*-Editor bietet. Fehlkonstruktionen in der Levelgestaltung und daraus resultierende Spielprobleme sind durch die eingeschränkten Designoptionen praktisch ausgeschlossen. Das Spiel beschränkt sich auf die Designmöglichkeiten, die im Anschluss spielbar sind. Es stellt sich demnach die Frage, ob *LITTLE BIG PLANET* noch Modding im engeren Sinne bietet. Zwar erstellt der Spieler auch hier neue Level, aber er erstellt sie im Rahmen der Spieloberfläche. Bei einem so einfach gehaltenen, stark geführten Editor-Prinzip bleibt kaum mehr Freiheit, etwas außerhalb der vom Hersteller vorgesehenen Level- und Regelwerke zu konstruieren. Der Editor ›gaukelt‹ dem Spieler potentielle Kreativität vor, verhindert aber ein potentielles Nicht-Funktionieren der Level durch eine interne Logikprüfung. Es können sich keine Bugs einschleichen, das Spiel kann nicht einfrieren, die Spielfigur nirgendwo feststecken usw.

5 Vgl. http://developer.valvesoftware.com/wiki/Create_a_Mod, 15.09.2011.

6 Quelle: <http://developer.valvesoftware.com/wiki/Category:Hammer>, 15.09.2011.

7 Quelle: <http://www.gamepro.de/misc/galleries/detail.cfm?pk=54839&fk=1214112>, 15.09.2011.

8 Vgl. Beil: »Vom CASTLE SMURFENSTEIN zum LITTLE BIG PLANET«, S. 204-214.

Ein Vergleich des *Hammer*- und des LITTLE BIG PLANET-Editors zeigt zwei grundlegend verschiedene ›Philosophien‹ des Modding. Dabei mag auf den ersten Blick der LITTLE BIG PLANET-Editor durch seine direkte Integration ins Spiel einen stärkeren ›Mod-Charakter‹ aufweisen. Auf den zweiten Blick zeigt sich jedoch, dass der Level-Editor als ›Feature‹ nur ein sehr begrenztes bzw. kontrolliertes Maß an Änderungen erlaubt. Der Spieler ›spielt‹ im Grunde nur das Spiel in den Grenzen der vorgegebenen Entfaltungsmöglichkeiten, indem er Level gestaltet.

Im Fall des *Hammer-Editors* ist zwar eine ungleich aufwendigere Einarbeitung und Ausdauer bei der Bedienung der Software nötig. Doch sofern Spieler die nötigen Werkzeuge bedienen können und die entsprechenden Fähigkeiten besitzen, können Modifikationen wie Level- und Skinpacks bis hin zu Total Conversations entwickelt werden. Dabei sind die Arbeitsabläufe der Modding-Communities in einigen Fällen durchaus mit denen professioneller Entwicklerteams vergleichbar.

Nimmt man den Aspekt hinzu, dass einige Spiele gezielt im Hinblick auf Modding-Möglichkeiten von den Herstellern konzipiert werden, ist der Spieler als ›Beteiligter‹ in einem nachgelagerten Game Design-Prozess keine Ausnahme, sondern fest eingeplant: »As with QUAKE [id, 1996], Carmack and his id team made sure QUAKE II [id, 1997] was coded from the very start to support fan mods.«⁹

Zudem stellen Mods für Spieler und Entwickler einen Zusatznutzen dar: für den Spieler in Form von neuen Leveln, Designänderungen des Charakters etc. und für den Hersteller durch verlängerte Aktualität bzw. die längere Nutzung des Spiels sowie damit verbunden die Wahrscheinlichkeit höherer Absatzzahlen. Im Falle der wohl berühmtesten Mod – COUNTER-STRIKE – lässt sich gar ein direkter Einfluss auf die (professionelle) Spiele-Entwicklung beobachten. Der Erfolg dieser 1999 von Hobbyprogrammierern entwickelten Modifikation veranlasste den HALF-LIFE-Entwickler Valve dazu, das COUNTER-STRIKE-Mod-Team unter Vertrag zu nehmen. Seit dem Jahr 2000 wird COUNTER-STRIKE als offizielle Erweiterung vertrieben. So startete COUNTER-STRIKE zwar als typisches Mod-Projekt, entwickelte sich aber letztlich zu einem kommerziellen Produkt, das darüber hinaus durch seinen enormen Erfolg die Gaming- wie die Modding-Szene nachhaltig geprägt hat.

9 Au: »Triumph of the Mod«, o.S.

CHRISTIAN TRAPP

LITERATURVERZEICHNIS

Au, Wagner James: »Triumph of the Mod«, 2002, <http://dir.salon.com/story/tech/feature/2002/04/16/modding/>, 15.09.2011.

Beil, Benjamin: »Vom CASTLE SMURFENSTEIN zum LITTLEBIGPLANET. Modding, Leveleditoren und Prosumenten-Kulturen«, in: Abresch, Sebastian/ders./Griesbach, Anja: Prosumenten-Kulturen, Siegen 2009, S. 191-214.

Kücklich, Julian: »Modding, Cheating und Skinning. Konfigurative Praktiken in Computer- und Videospiele«, 2004, <http://www.dichtung-digital.org/2004/2-Kuecklich-b.htm>, 15.09.2011.