

Berk Binay

Game development tools

2011

<https://doi.org/10.25969/mediarep/844>

Veröffentlichungsversion / published version
Zeitschriftenartikel / journal article

Empfohlene Zitierung / Suggested Citation:

Binay, Berk: Game development tools. In: *Navigationen - Zeitschrift für Medien- und Kulturwissenschaften*, Jg. 11 (2011), Nr. 2, S. 81–84. DOI: <https://doi.org/10.25969/mediarep/844>.

Erstmalig hier erschienen / Initial publication here:

<https://nbn-resolving.org/urn:nbn:de:hbz:467-8205>

Nutzungsbedingungen:

Dieser Text wird unter einer Deposit-Lizenz (Keine Weiterverbreitung - keine Bearbeitung) zur Verfügung gestellt. Gewährt wird ein nicht exklusives, nicht übertragbares, persönliches und beschränktes Recht auf Nutzung dieses Dokuments. Dieses Dokument ist ausschließlich für den persönlichen, nicht-kommerziellen Gebrauch bestimmt. Auf sämtlichen Kopien dieses Dokuments müssen alle Urheberrechtshinweise und sonstigen Hinweise auf gesetzlichen Schutz beibehalten werden. Sie dürfen dieses Dokument nicht in irgendeiner Weise abändern, noch dürfen Sie dieses Dokument für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, aufführen, vertreiben oder anderweitig nutzen.

Mit der Verwendung dieses Dokuments erkennen Sie die Nutzungsbedingungen an.

Terms of use:

This document is made available under a Deposit License (No Redistribution - no modifications). We grant a non-exclusive, non-transferable, individual, and limited right for using this document. This document is solely intended for your personal, non-commercial use. All copies of this documents must retain all copyright information and other information regarding legal protection. You are not allowed to alter this document in any way, to copy it for public or commercial purposes, to exhibit the document in public, to perform, distribute, or otherwise use the document in public.

By using this particular document, you accept the conditions of use stated above.

GAME DEVELOPMENT TOOLS

VON BERK BINAY

Geht es nach dem schottischen Essayisten und Historiker Thomas Carlyle, so hat der Mensch seine Errungenschaften den Werkzeugen zu verdanken, die ihm zu Grunde liegen: »Man is a Tool-using animal [...]. Nowhere do you find him without Tools; without Tools he is nothing, with Tools he is all.«¹

Ein integraler Bestandteil der Entwicklung eines guten Spiels ist die Erstellung eindrucksvoller Inhalte. Damit diese entstehen können und später wettbewerbsfähig sind, benötigt das Design-Team passend entworfene und stabile Game Development Tools. Dementsprechend führt der Weg zur Konzeption und Entwicklung eines guten Spiels über die Auswahl (bzw. Entwicklung) relevanter Tools.²

Man könnte annehmen, dass der heutige freie Software-Markt genügend gängige Lösungen bzw. *Off-The-Shelf-Solutions*³ bieten sollte; kommerzielle Standardsoftware, Tools und Middleware, die von den Studios lizenziert und eingesetzt werden können. Beispielsweise sind Autodesk's *Maya* und *3ds Max* mächtige und praktisch unverzichtbare Werkzeuge für die Generierung dreidimensionaler Modelle und Animationen, bieten allerdings für viele Designer nicht genügend Flexibilität, wenn Levels oder Maps erstellt und bearbeitet werden sollen:

The sad fact is, you can find scads of good software for designing your own patio or managing your recipes, but there are only a handful of packages dedicated to our most unique and difficult art task. [...] *Max* and *Maya* are very powerful and provide exact control. They'll empower you to fiddle with every vertex in your world [...]. Unfortunately, that level of control may be more curse than blessing if you're working on a game in which careful calibration of the play spaces is a key to success.⁴

Wenn also der Software-Markt nicht in der Lage ist, die Tools zu liefern, die man braucht, muss man sich als Spiele-Entwickler fragen, ob man seine eigenen Tools entwickeln kann. Üblicherweise haben die meisten großen Game Studios ihr eigenes Tools Development Team, das im Rahmen der Spiele-Entwicklung sogenannte *Custom-* bzw. *In-House-Tools* entwickelt. Einige Praxisbeispiele für maßgeschneiderte, intern entwickelte und genutzte Tools sind:

-
- 1 Carlyle: »Sartor Resartus«, o.S.
 - 2 Vgl. Rouse III: »Designing Design Tools«.
 - 3 Vgl. Goodman: »Build or Buy?«.
 - 4 Theodore: »Custom Tools«, o.S.

- Plugins für 3D-Software wie *Maya* und *3ds Max*, die das Importieren von animierten Modellen und Texturen in die Game Engine ermöglichen
- Datenbank-Manager, die Spiele-Statistiken im Tabellenformat in ein Dateiformat umwandeln können, welches für die Game Engine lesbar ist
- Automatisierung von manuellen Prozessen im Arbeitsablauf durch Skripte
- Tools für die Lokalisierung (Übersetzung)
- Level-/Map-Editoren

Im Gegensatz zur Grafikprogrammierung oder der Entwicklung der Künstlichen Intelligenz wird die Programmierung von Tools oft als weniger glamourös empfunden;⁵ die Beispiele verdeutlichen, dass der Tools Developer keine Inhalte generiert und eher indirekt an der Entwicklung des eigentlichen Spiels beteiligt ist. Workflow-Analysen sind ein wesentlicher Teil seiner Arbeit, weshalb er im engen Kontakt mit den anderen Abteilungen steht und den Leuten bei der Arbeit sozusagen über die Schulter schaut. Schwerpunkt dieser Analysen ist vor allem das Erkennen von Möglichkeiten, zeitsparende, produktive, effiziente und automatisierende Features in den Arbeitsablauf des Game Designers zu integrieren. Eine Software-Lösung in Form eines Tools, Plugins oder Skripts bietet sich also immer dann an, wenn in einem bestehenden Arbeitsablauf sich wiederholende, komplexe oder langweilige Prozesse vorkommen.

Da die Qualität seiner Arbeit sich direkt auf die Produktivität und Effizienz der Designer und Entwickler auswirkt, ist der Tools Developer ein bedeutender Akteur bei der Spiele-Entwicklung:

In-house tools are one of the most important aspects of producing high performance console video game software. Efficient tools have always been important, but as the data content in video games has grown exponentially over the last few years, in-house tools have become increasingly more important to the overall development process. In the not too distant future, the focus on tool programming techniques may even exceed the focus on run-time programming issues. It is not unreasonable that the most impressive video games in the future may end up being the ones that have the best support tools.⁶

Dieser Umstand ist den Studios bewusst. So waren Ende 2004 etwa 30% aller BioWare-Programmierer an der Entwicklung von Tools beteiligt.⁷ Rund 30 % des

5 Vgl. Moar: »Growing a Dedicated Tools Programming Team«.

6 Gavin/White: »Teaching an Old Dog New Bits«, o.S.

7 Vgl. Moar: »Growing a Dedicated Tools Programming Team«.

gesamten Quellcodes (exklusive Middleware) von Double Fine's *BRÜTAL LEGEND* (2009) gehören den Tools.⁸

Die Vorteile effektiver *In-House*-Tools sind naheliegend: Höhere Produktivität und Effizienz bedeuten, dass der Produktionszyklus des Spiels schneller und häufiger iteriert, was mehr QA (*quality assurance*) bzw. mehr Zeit zum Testen bedeutet, was wiederum weniger Fehler im Endprodukt zur Folge hat oder haben sollte.⁹ Eine Besonderheit bieten sogenannte *World-Creation-Tools* bzw. Level- und Map-Editoren, deren Vorteile offensichtlich sind:

WYSIWYG = What you see is what you get.

Editoren wie der Sandbox-Editor von Crytek ermöglichen eine Art Vorschau-Funktion auf das Endprodukt bereits beim Erstellen und Bearbeiten von virtuellen Welten. Während man früher unzählige Zeilen Quellcode programmieren musste, um ein erstes visuelles Feedback zu erhalten, genügen heutzutage mit Hilfe solcher Werkzeuge nur wenige Maus-Klicks:

The advantages of tightly integrating your tools and your game engine should be obvious. For starters, you can build and texture your world with the textures and materials your players will see, so that artists get immediate feedback. Having game controls and game physics running can also make it easier to see your level from the player's point of view, complete with animation and interactive behaviors. Not only does this cut down iteration time, it's also a great corrective to the level artists' perennial temptation: worrying about how the level looks in the overhead view that only the author ever sees. To round it all out, your game engine probably has better interactive performance than a standard app – after all, it's optimized for the kind of content you're creating. At least, you'd better hope it is!¹⁰

Die Mächtigkeit solcher Werkzeuge liegt vor allem darin begründet, dass der Tools Developer die Mathematik, die hinter der Game Engine steht, vom Design-Prozess abstrahiert und die Game Engine durch eine benutzerfreundliche Oberfläche dem Designer zugänglich macht. Die Engine wird somit selbst zum Editor.

8 Vgl. Esmurdoc: »Postmortem: Double Fine's Brutal Legend«.

9 Vgl. Moar: »Growing a Dedicated Tools Programming Team«.

10 Theodore: »Custom Tools«, o.S.

BERK BINAY

LITERATURVERZEICHNIS

- Carlyle, Thomas: »Sartor Resartus«, http://en.wikisource.org/wiki/Sartor_Resartus/Book_1/Chapter_5, 15.09.2011.
- Esmurdoc, Caroline: »Postmortem: Double Fine's Brutal Legend«, http://www.gamasutra.com/view/feature/4308/postmortem_double_fines_brutal_.php, 15.09.2011.
- Gavin, Andy/White, Stephen: »Teaching an Old Dog New Bits: How Console Developers are Able to Improve Performance When the Hardware Hasn't Changed«, http://www.gamasutra.com/view/feature/3398/teaching_an_old_dog_new_bits_how_.php, 15.09.2011.
- Goodman, Dan: »Build or Buy? Finding the Right Tools Solution for Your Game Company«, http://www.roboticarmsoftware.com/Goodman-Build_or_Buy.pdf, 15.09.2011.
- Moar, Don: »Growing a Dedicated Tools Programming Team: From Baldur's Gate to Star Wars: Knights of the Old Republic«, http://www.gamasutra.com/view/feature/2055/growing_a_dedicated_tools_.php, 15.09.2011.
- Rouse III, Richard: »Designing Design Tools«, http://www.gamasutra.com/view/feature/3443/designing_design_tools.php, 15.09.2011.
- Theodore, Steve: »Custom Tools: Environment Artists and Game Editors«, http://www.gamasutra.com/view/feature/3989/custom_tools_environment_artists_.php, 15.09.2011.