



Computerspiele programmieren - mit kostenlosen Tools

Christoph Kaindel

Nach den Verkaufszahlen der wichtigsten Spieleserien der letzten Jahre, wie "Call of Duty", "Assassin's Creed", "Uncharted" etc., zu schließen, können Computer- und Konsolenspiele heute als kommerziell erfolgreichstes Unterhaltungsmedium gelten. Auch auf Mobiltelefonen und Tablets machen Spiele den Großteil der beliebtesten Anwendungen aus. Sie werden heute nicht mehr vorwiegend von Kindern und Jugendlichen gespielt, im Durchschnitt sind Spieler und Spielerinnen Mitte dreißig, etwa die Hälfte sind weiblich. Damit sind Spiele ein bedeutender Wirtschaftsfaktor geworden, Entwicklerstudios werden immer größer und ausgebildete Spieldesigner und -designerinnen sind gefragte Fachkräfte. Seit Jahren schon gilt, dass die Leistungsfähigkeit moderner Computer – neben Spezialanwendungen wie Filmschnitt und Animation – von den meisten Konsumenten und Konsumentinnen nur durch Spiele wirklich ausgereizt werden kann. Auch die Wissenschaft nimmt sich mittlerweile in vielfältiger Weise der Computerspiele an, jenseits der „klassischen“ Medienwirkungsforschung, die leider immer noch die öffentliche

Diskussion dominiert. Studiengänge an Universitäten und Fachhochschulen vermitteln theoretische und praktische Kenntnisse zu dem nicht mehr ganz neuen Medium, internationale Konferenzen beschäftigen sich mit verschiedenen Aspekten der *game studies*. Von einigen Autoren werden Computerspiele bereits als das Leitmedium des 21. Jahrhunderts bezeichnet.

Dennoch findet eine komplexe inhaltliche Auseinandersetzung mit Computerspielen in der sogenannten breiten Öffentlichkeit kaum statt. In Zeitungen und Zeitschriften gibt es regelmäßig Besprechungen aktueller Film- und Theaterproduktionen wie auch literarischer Neuerscheinungen, doch Computer- und Konsolenspiele kommen nur sporadisch vor. Aber auch in Spiele-Magazinen und auf Spielekritik spezialisierten Webseiten ist die inhaltliche Kritik mangelhaft; da geht es vor allem um technische Brillanz, um zeitgemäße Grafik und ein packendes Spielerlebnis, in einigen Genres auch um eine komplexe Hintergrundgeschichte. Fragen nach ästhetischen, historischen oder medialen Einflüssen, einer Erzähltradition oder gar einer gesellschaftspolitischen Aussage, wie sie in der Literatur- oder Filmkritik selbstverständlich sind, kommen in der Spielekritik so gut wie gar nicht vor. Das mag daran liegen, dass Computer- und Konsolenspiele trotz ihrer kulturellen Bedeutung noch immer weitgehend als Schund, als kindisch, im besten Fall als populäre Unterhaltung, jedenfalls als einer ernsthaften Kritik nicht würdig angesehen werden.

Umso wichtiger ist es, Computer- und Videospiele in der Arbeit mit Kindern und Jugendlichen zu thematisieren. Im folgenden möchte ich als einen Weg zur Annäherung an das Medium die praktische kreative Medienarbeit mit Computerspielen vorschlagen. Die Produktion von Machinima, also in Spielen gedrehten Filmen, mit Spielen wie *Die Sims* hat in den letzten Jahren weite Verbreitung gefunden. Jugendprojekte, in denen es um das Design und das Programmieren von Spielen geht, werden dagegen, so ist mein Eindruck, noch sehr selten durchgeführt. Das mag daran liegen, dass bis vor einigen Jahren kaum günstige und einfach zu bedienende Spielentwicklungsprogramme verfügbar waren.

Doch mittlerweile gibt es zahlreiche kostenlose Werkzeuge, die die Programmierung einfacher Spiele auf durchschnittlichen Computern ermöglichen; einige dieser Werkzeuge und ihre grundlegenden Funktionen möchte ich hier vorstellen.

Was kann man durch das Programmieren von Spielen lernen? Nun, zunächst ist ein Computerspiel nicht anderes als ein Programm; das Entwickeln von Computerspielen ist also eine lustvolle Methode, die Grundlagen des Programmierens zu lernen – einige der unten angeführten Programme sind zu diesem Zweck entwickelt worden. Heute, da Computer zu *black boxes* geworden sind, unter deren Oberfläche nur mehr wenige Fachleute schauen können, ist es gut zumindest ein wenig zu verstehen, was in den Geräten vorgeht. So lernt man dann auch, was beim Spielen an Konsolen oder auf Handys „unter der Haube“ abläuft. An den grundlegenden Spielabläufen hat sich in den etwa vierzig Jahren der Computerspielgeschichte nicht viel geändert. Damals wie heute mussten ähnliche Probleme gelöst werden, wenn auch heute die Rechengvorgänge ungleich komplexer sind. Ganz am Beginn des Designprozesses stehen damals wie heute grundsätzliche Fragestellungen wie: Worum geht es überhaupt? Was ist das Thema, was das Spielziel? An welche Zielgruppe richtet sich das Spiel? Dann sind zahlreiche weitere Entscheidungen zu treffen:

- Spieloberfläche – Wie sieht der Spielbildschirm aus, wie erhalten die Spielenden Informationen über den Spielverlauf?
- Steuerung – Wie wird das Spiel gesteuert, welche Interaktionsmöglichkeiten stehen zur Verfügung?
- Kollisionsabfrage – wann berührt ein Objekt der Spielwelt ein anderes, und was passiert dann?
- Pathfinding – Wie finden Spielfiguren oder Gegner und Gegnerinnen den Weg von A nach B, wie umgehen sie Hindernisse?

Diese und zahlreiche weitere Fragen müssen im Verlauf der Spielentwicklung beantwortet werden, egal ob es sich um Pac-Man oder einen modernen Militärshooter handelt. Daher kann man sich mit dem Medium auch anhand der einfachsten Spiele auseinandersetzen.

Spielentwicklung beinhaltet zahlreiche Aufgaben und eignet sich gut für die Gruppenarbeit. Am Beginn steht die grundsätzliche Konzeption; dabei

muss auch immer mitbedacht werden, welche technischen und zeitlichen Möglichkeiten zur Verfügung stehen. Im Zuge dieser Phase sollte ein sogenanntes „Design Document“ verfasst werden, in dem Thema, Gameplay, Spielfiguren etc. möglichst detailliert beschrieben werden. Bei einfachen Spielen wird dieser Text nicht mehr als ein bis zwei Seiten umfassen. Dann folgt die Entwicklung eines Prototypen mit den Basisfunktionen und ganz einfacher Grafik. Schließlich kommen wir zur eigentlichen Spielentwicklung. Hier fallen zahlreiche zu verteilende Aufgaben an – von der Erstellung der Grafiken, gezeichnet oder auf Fotos basierend, der Aufnahme von Sounds und Musik, der Programmierung an sich bis hin zum Testen des Spieles. Das klingt sehr aufwändig, tatsächlich können einfache Spiele bereits in wenigen Stunden programmiert werden. In den meisten der unten angeführten Programme sind schon fertige Spielfiguren, Sounds etc. verfügbar. In "Kodu Game Lab" etwa sind alle verwendbaren Spielelemente bereits vorgegeben und können sehr einfach zusammengestellt werden. So stellt sich schnell ein Erfolgserlebnis ein.

Im Folgenden werde ich die drei Programme "Kodu Game Lab", "GameMaker 8" und "Stencylworks" etwas genauer beschreiben. Im Anschluss folgt eine Liste weiterer Gratistools.

Kodu Game Lab

Kodu basiert auf dem ebenfalls kostenlosen XNA Game Studio von Microsoft. Es ist ein sehr einfach strukturiertes Programm, mit dem auch Kinder in die Spieleprogrammierung – bzw. allgemein in den Aufbau von Computerprogrammen – hineinschnuppern können. Kodu ist ursprünglich für die Xbox360 entwickelt worden, mittlerweile gibt es eine kostenlose PC-Version, natürlich nur für Windows. Zahlreiche Beispiel-„Welten“ sind inkludiert, im Spiel gibt es eine Reihe von Tutorials. Welten lassen sich einfach modifizieren, alle Spielelemente können per drag-and-drop in die Welt hineingezogen werden. Es können keinerlei

eigene Elemente importiert werden, allerdings ist genug Material vorhanden, um einige Zeit für kreativen Spielspaß zu sorgen.

Alle Objekte können nach dem Prinzip „WHEN – DO“ programmiert werden. So kann etwa die Spielfigur Kodu das Programm

- *WHEN-sehen-Apfel – DO-hinbewegen*
- *WHEN-stoßen an-Apfel – DO-essen-Apfel*

erhalten und wird dann selbständig alle Äpfel in der Spielwelt suchen und essen. Alle Programmelemente sind durch leicht erkennbare Symbole gekennzeichnet. Das ist für Kinder auch darum günstig, weil das Programm nur unvollständig ins Deutsche übersetzt ist.

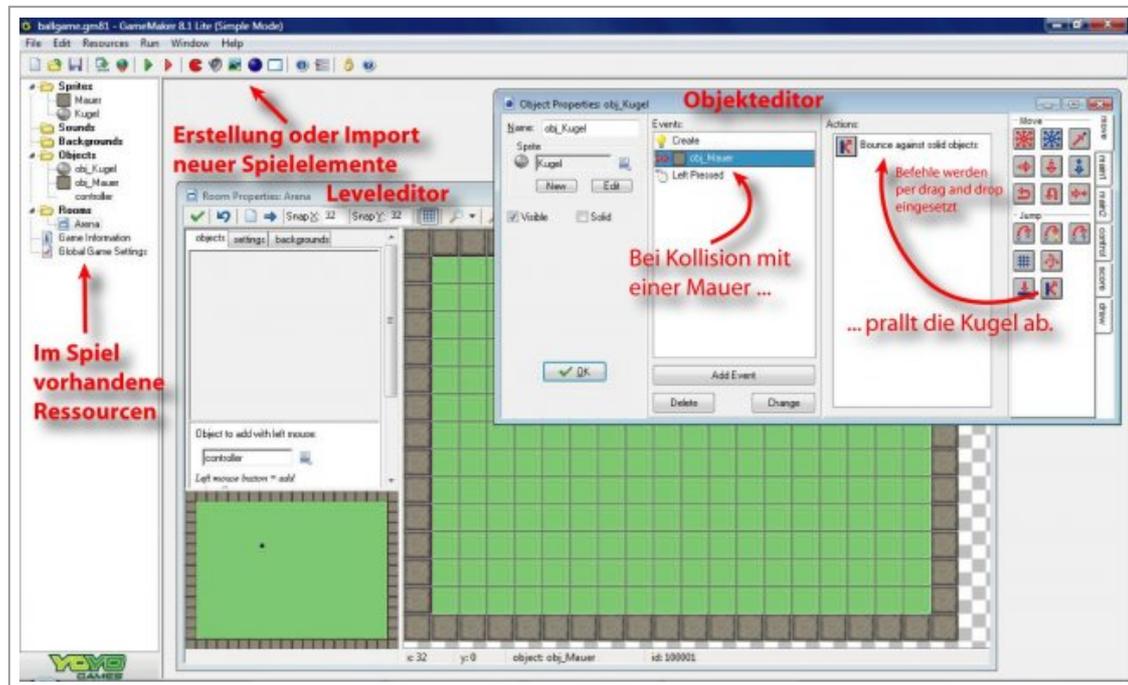
Vorteile:

- Hübsches, cartoonartiges 3D-Design – für Jugendliche aber etwas zu kindlich
- Sehr einfache Programmierung verschiedener Spiele – Jump-and-Run, Rennspiel, Adventure etc
- Viele Tutorials und auch Zusatzmaterialien für PädagogInnen verfügbar

Nachteile:

- Keine eigenen Elemente importierbar
- Spiele sind nur in Kodu Game Lab lauffähig und können nicht als eigenständige Programme exportiert werden
- Ressourcenhungrig und auf älteren Computern kaum lauffähig

GameMaker



Screenshot GameMaker

Der GameMaker von Marc Overmars ist mittlerweile in der Version 8 (für Windows, Version 7 für Mac) verfügbar und wird von Yoyogames vertrieben. Für die meisten Projekte wird man mit der kostenlosen Basisversion auskommen, die Pro-Version kostet derzeit 40\$. GameMaker ist für klassische sprite-basierte 2D-Spiele gedacht, die Pro-Version verfügt über begrenzte 3D-Funktionen. Das Programm funktioniert auch auf älteren Computern. GameMaker-Spiele können über den grafischen Objekt-Editor sehr einfach per drag-and-drop programmiert werden, simple Geschicklichkeitsspiele entstehen so in wenigen Minuten; grundsätzlich ist GameMaker für Spiele jedes Genres geeignet. Zusätzlich gibt es die Scriptsprache GML, die komplexere Programmierung ermöglicht.

In GameMaker können selbst erstellte Bilder und eigene Sounds und Musik verwendet werden. Das Programm enthält auch einen Grafikeditor, mit dem etwa einfache Animationen gezeichnet werden können.

Das Programm ist in englischer Sprache, auf der deutschen Community-Seite gibt es eine ausführliche deutschsprachige Hilfedatei und viele Tutorials auf Deutsch. Spiele können als eigenständige Programme exportiert oder auf die Webseite von Yoyogames hochgeladen werden, wo man auch online spielen kann.

Vorteile:

- Von einfacher bis komplexer Programmierung, alles ist möglich
- Große Community
- Eigene Elemente verwendbar
- Spiele können als eigenständige Programme exportiert werden
- Auch auf älteren Computern funktionsfähig

Nachteile:

- Programm in englischer Sprache
- Optisch nicht mehr auf dem neuesten Stand
- Nur zweidimensionale Spiele
- Gratis-Version hat Einschränkungen

Stencyl Works

StencylWorks ist ein kostenloses Spielentwicklungsprogramm für Mac und Windows. Spiele können im Flash-Format exportiert und daher sehr einfach in die eigene Homepage eingebunden oder auf die Spieleplattform Kongregate verlinkt werden. Es gibt auch eine Exportmöglichkeit für iOS.

Im Aufbau ähnelt StencylWorks dem GameMaker, es ist allerdings in der Bedienung etwas komplexer. Das Programm ermöglicht ebenfalls die Erstellung zweidimensionaler Spiele, die drag-and-drop Programmierung erfolgt aber nicht über Bildsymbole, sondern eine grafische, farbkodierte Programmiersprache, die an das Projekt Scratch angelehnt ist. Mit Hilfe dieser Programmiersprache können „Behaviours“ für die einzelnen Spielobjekte programmiert werden. Das ist zwar einfacher als etwa bei der Flash-Programmierung, aber deutlich schwieriger als in GameMaker. StencylWorks greift allerdings auf die Online-Ressourcensammlung StencylForge zu, wo zahlreiche fertige „Behaviours“ verfügbar sind, die man als Bausteine in eigenen Spielen verwenden kann. Auch ganze frei

modifizierbare Spiele gibt es auf StencylForge, zusätzlich zu den Beispielen, die in Stencylworks bereits verfügbar sind. Auch eigene Spielelemente wie Grafiken oder Musik können in StencylWorks verwendet werden.

Die Online-Hilfe ist sehr ausführlich, es gibt einige nützliche Tutorials, allerdings nur auf Englisch. Das Programm selbst ist nur teilweise ins Deutsche übersetzt.

Vorteile:

- Export für Flash oder iOS
- Online-Ressourcenpool
- Gute Einführung in Programmiersprachen

Nachteile:

- Für AnfängerInnen zu komplex
- Nur in englischer Sprache verfügbar
- Muss in Windows7 im Administratormodus ausgeführt werden

Weitere kostenlose Spiele-Programmierwerkzeuge

Einstiegsprogramme

Scratch ist wie gesagt eine farbkodierte Programmier-Umgebung, für die es jede Menge Tutorials und andere Ressourcen gibt. Spiele können auf der Projektseite gespielt werden. Alice ist eine 3D-Programmierumgebung für die Erstellung von Spielen, animierten Filmen etc. Platinum Arts Sandbox ist ein kostenloser 3D-Gamemaker, angeblich sehr einfach und auch kindgerecht. Der RPGToolkit ist ein open-source-Werkzeug, das vor allem für Rollenspiele und Jump-and-Runs geeignet ist. Der GameMaker Engine001 ähnelt dem GameMaker 8, ist aber für nicht-kommerzielle Zwecke völlig kostenlos. 3D-GameMaker ist ein 3D-GameMaker. E-Adventure ist ein in Java geschriebenes open-source Programm zur Erstellung von point-and-click-Adventures, das aus einem Projekt über den Einsatz von Adventures im Unterricht hervorgegangen ist; es ist auch auf Deutsch verfügbar.

Programme für Fortgeschrittene

Das 3D-Gamestudio ist in der eingeschränkten Basisversion gratis, die Vollversionen kosten bis zu € 800,-. Das gilt auch für die beliebte Spiel-Engine Unity, die Pro-Version ist hier noch etwas teurer (eine Engine ist sozusagen der Motor, der ein Spiel antreibt). Das kostenlose XNA Game Studio ist die „große Schwester“ von Kodu Game Lab; damit können Spiele für Windows PCs und Xbox360 programmiert werden. Sowohl die Unreal Engine 3 als auch die CryEngine 3 sind aktuelle high-end Game-Engines; beide sind für nicht-kommerzielle Projekte frei nutzbar und verfügen über umfangreiche Dokumentationen. Objekte und Spielfiguren müssen allerdings in anderen Programmen erstellt werden, etwa in Blender, einem Open Source 3D-Grafikprogramm, das auch eine eigene 3D-Engine beinhaltet.

Am Ende möchte ich noch ein paar Worte sagen zum Mission Maker, einer Software, die offenbar für Jugend- und Schulprojekte immer wieder eingesetzt wird: Das mag ja ein gutes Programm sein. Es mag ja auch recht einfach sein, damit schnell zu guten Ergebnissen zu kommen. Trotzdem: Es ist vollkommen unverständlich, wenn angesichts eines mittlerweile so großen Angebots von kostenloser, guter Software ein veraltetes Programm so teuer verkauft wird – eine Schul-Lizenz kostet etwa € 1800,- (plus kostenpflichtiger Unterrichtsmaterialien und Support), eine Einzel-Lizenz immerhin ca. € 95,-. Gratis-Versionen gibt es nicht. Finger weg und lieber eines der oben angeführten Programme verwenden!