

Maturitätsarbeit an der Kantonsschule Oerlikon



Engineprogrammierung und Gamedesign von Grund auf

Erstellt und Konzipiert von Nicolas Hafner, 6S
Betreut von Clemens Holenstein

Zürich, Dezember 2011

Inhaltsverzeichnis

1	Einführung	2
1.1	Idee	2
1.2	Übersicht	3
1.3	Konzepte	4
2	Enginedesign & Programmierung	5
2.1	Einführung	5
2.2	Techniken & Tools	5
2.3	Ziele & Einsatzmöglichkeiten	5
2.4	Codedesign	5
2.5	Probleme im Detail	5
2.5.1	Kollisionserkennung	5
2.5.2	AI	5
2.5.3	Events & GUI	5
3	Grafikdesign	5
3.1	Einführung	5
3.2	Ideen & Konzepte	5
3.3	Einschränkungen	5
4	Gamedesign	5
4.1	Einführung	5
4.2	Charakterdesign	5
4.3	Spielmechaniken	5
4.4	Story	5
4.5	Leveldesign	5
5	Schlussfolgerung & Aussicht	5
5.1	Schlussfolgerungen	5
5.2	Erweiterungsmöglichkeiten	5
6	Danksagung, Referenzen & Weiteres	6
6.1	Persönlicher Dank	6
6.2	Tools, Webseiten & Bücher	7

Liste der Abbildungen

1	Darstellung einiger Bereiche des Game Designs	3
---	---	---

1 Einführung

1.1 Idee

Ich hatte schon früh in meiner Jugend Interesse daran, Spiele selbst zu erstellen. So hielt ich oft Ausschau nach Spielen, die einen Level Editor beinhalteten, damit ich mit diesem meine eigenen Welten erstellen konnte. Es dauerte auch nicht lange bis ich auf die Idee kam eigene Spiele von Grund auf zu bauen, und bin so auf GameMaker gestossen. GameMaker ist ein vereinfachtes Tool zur Erstellung von Videospielen. Allerdings gibt es auch bei diesem Programm relativ grosse Einschränkungen. Es brachte mir aber dennoch einen Einstieg in die Welt des Gamedesigns und ich habe einige Spiele erstellt, von welchen aber keines je fertiggestellt wurde oder auch irgendwie Nennenswert wäre.

Auf meiner Suche nach mehr Möglichkeiten habe ich mich dann irgendwann entschieden eine richtige Programmiersprache zu lernen und bin so auf Java gestossen, welches ich nun seit vielen Jahren extensiv gebrauche und schon etliche Programme damit erstellt habe. Es war mir deshalb schon Jahre vorher klar, dass ich für die Maturarbeit ein Programm schreiben würde. Was es genau sein würde, wusste ich aber lange noch nicht. Ich hatte ursprünglich viele Ideen, wie z.B. einen Raytracer¹, einen Fraktal Generator/Animator², oder einen Simulator für Neuronale Netze³.

Allerdings habe ich vor etwas mehr als Zweieinviertel Jahren angefangen zu Zeichnen, und das Programmieren ist für einige Zeit in den Hintergrund gerutscht. Heute habe ich eine Balance von beidem erreicht. Durch das neu gefundene Interesse im eher künstlerischen Bereich haben sich aber auch meine Ziele und Absichten verändert. Als sich nun die Frage der Maturarbeit offenbarte war mir klar, dass ich etwas finden müsste, das beide Aspekte, sowohl das Zeichnerische als auch das Programmiertechnische, beinhaltet.

Ich hatte auch bis dann noch nicht die Sehnsucht, ein eigenes Spiel zu erstellen, verloren. Nur war die Zeit oft knapp und ich hatte bereits viele andere Projekte die meine Freizeit in Anspruch nahmen. Die Erstellung eines Spiels nimmt enorm viel Zeit in Anspruch. Zeit, die ich sonst nicht hatte.

Da ich aber so oder so einige Zeit in die Maturarbeit investieren müsste, war es nicht weit hergeholt, das Thema meiner Maturarbeit um die Spielprogrammierung und das allgemeine Gamedesign zu setzen. So hätte ich endlich genug Zeit eingeräumt, im etwas anständiges zu produzieren. Das Ziel war es damals, ein möglichst weit entwickeltes Spiel zu produzieren, damit ich auch endlich mal etwas richtiges zu zeigen hätte und auch etwas gemacht hätte, worauf ich stolz sein könnte.

¹ Ein Raytracer erzeugt ein zweidimensionales Bild einer dreidimensionalen, digitalen Szene durch Anwendung von physikalischen Eigenschaften, Berechnungen und Verfahren. Raytracing ist ein sehr breites und komplexes Feld der Informatik und ist stark mit Mathematik und Physik verbunden.

² Ein Fraktal Generator errechnet und färbt einen Ausschnitt eines Fraktals. Dabei kann der Benutzer den Ausschnitt frei wählen und sich im Fraktal bewegen.

³ Simulatoren für Neuronale Netze sind der Versuch, ein Gehirn, oder zumindest die Art wie das Gehirn lernt mit Software zu repräsentieren und so eine künstliche, lernende Intelligenz zu erstellen.

1.2 Übersicht

Normalerweise werden Spiele in Teams erstellt, wo meistens eine grosse Menge and Spezialisten am Werk sind und miteinander kooperieren, Ideen bringen und ihr Wissen mit einflüssen lassen. Dies ist vor allem so, da Game Design und Spieleerstellung ein enormes Gebiet ist, welches viele verschiedene Bereiche anspricht. Um alle diese Bereiche zu decken ist ein enormes

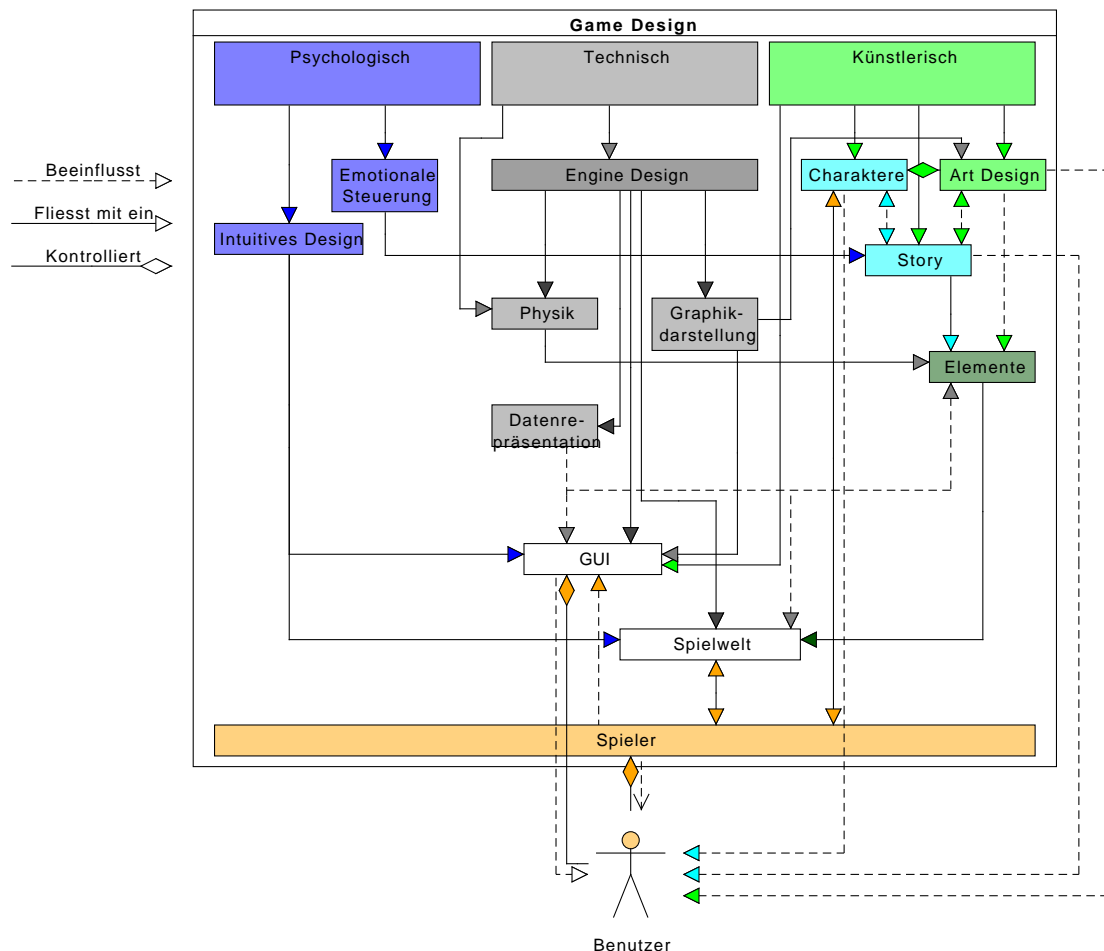


Abb 1: Darstellung einiger Bereiche des Game Designs

Wissen notwendig. Viele der Bereiche können auch nur durch ständiges Ausprobieren und Testen perfektioniert werden, was viel Zeit beansprucht.

In meiner Arbeit habe ich mich vor allem auf das Technische konzentriert, versuchte aber dennoch auf die anderen Bereiche einzugehen und habe mich ausgiebig über sie informiert. Auch wenn im praktischen Produkt nicht auf alles eingehen werden konnte, da es aus zeitlichen Gründen nicht reichte, wird dennoch auf all diese Aspekte in diesem Dokument eingegangen. Dabei handelt es sich um Psychologische, Anthropologische, Mathematische, Physikalische, Künstlerische und Musische Überlegungen und Konzepte, die alle beim Game Design eine Rolle spielen.

1.3 Konzepte

Als ich mit dem Projekt begann habe ich mir vor allem Dinge überlegt, die erst sehr spät im Prozess zum Einsatz kämen. So hatte ich Ideen für Spielkonzepte, eine Story und einige Grundcharaktere. Das Wichtigste dabei war mir, dass das Spiel in 2D war und einen Jump & Run Ansatz wie z.B. Super Mario oder Sonic the Hedgehog verfolgte. Allerdings ist dies ja nicht eine so originelle Idee, weshalb ich mir einige Spezialitäten überlegen musste.

Auf der Suche nach neuen Ansätzen dachte ich zurück an Spiele, die mir selbst sehr gut gefielen. Am stärksten trat dabei Okami¹ hervor. Daraus habe ich auch das Konzept der Fähigkeiten übernommen. Damit meine ich, dass der Spieler über den Verlauf des Spiels neue Fähigkeiten erlangt, die er dann benutzen kann um neue Regionen in der Welt zu erreichen, die er vorher noch nicht erreichen konnte. Dies ist im starken Kontrast zu herkömmlichen J&Rs, wo einem alle Fähigkeiten (bis auf Power-ups) von Beginn an verfügbar sind. Des weiteren erlaubt dieses Konzept einen weitaus grösseren Grad an Komplexität und Wiederspielbarkeit. So steht dem Spieler die ganze Welt von Anfang an offen, was ihm erlaubt sie ausgiebig auszukundschaften und sich schon im voraus Gedanken zu machen, was später einmal möglich sein wird oder interessant wäre.

¹ www.okami-game.com/

2 Enginedesign & Programmierung

2.1 Einführung

2.2 Techniken & Tools

2.3 Ziele & Einsatzmöglichkeiten

2.4 Codedesign

2.5 Probleme im Detail

2.5.1 Kollisionserkennung

2.5.2 AI

2.5.3 Events & GUI

3 Grafikdesign

3.1 Einführung

3.2 Ideen & Konzepte

3.3 Einschränkungen

4 Gamedesign

4.1 Einführung

4.2 Charakterdesign

4.3 Spielmechaniken

4.4 Story

4.5 Leveldesign

5 Schlussfolgerung & Aussicht

5.1 Schlussfolgerungen

5.2 Erweiterungsmöglichkeiten

6 Danksagung, Referenzen & Weiteres

6.1 Persönlicher Dank

Über die Zeit, die ich an dieser Arbeit verbracht habe, habe ich mit einigen Personen immer wieder Ideen und Konzepte diskutiert, anfällige Probleme bearbeitet und nach neuen Ansätzen gesucht (und zum Teil, wenn es aussichtslos schien, holte ich mir auch emotionale Unterstützung bei ihnen). Ich möchte all diesen Personen meinen grössten Dank verrichten und bin mir sicher, dass ich ohne sie diese Arbeit nie so weit gebracht hätte oder überhaupt erst auf diese Idee gekommen wäre.

Als erstes möchte ich Herrn Clemens Holenstein für eine gute Betreuung und nahtloses Zusammenarbeiten bei der Maturarbeit bedanken. Ich bin über alles froh, dass ich dieses Thema wählen durfte und er mir die Freiheit gegeben hat, die ich benötigte um diese Arbeit anzugehen.

Des Weiteren danke ich Frau Tanja Dorigo für die Zustimmung die Rolle der Zweit-Bezugsperson zu übernehmen. Ausserdem bedanke ich mich für eine sehr informative und fördernde Diskussionsstunde, die wir zusammen geführt haben.

Obwohl Herr Michael Stadelmann nicht mehr an unserer Schule ist, so danke ich ihm für das ursprüngliche Anspornen und die Überzeugungskraft, dass ich dieses Projekt auch in angriff nehmen könne, die er mir gegeben hat.

Selbstverständlich bin ich sehr froh über die Unterstützung, die mir meine Familie und meine Klassenkameraden gegeben haben, und das Interesse, das sie an meiner Arbeit gezeigt haben. Auch dafür möchte ich meinen Dank aussprechen, und hoffe dass sie auch weiterhin Interesse an der Weiterführung meiner Arbeit zeigen werden.

Speziell möchte ich noch Yassin Sdiri erwähnen, der mir als lang bekannter Freund auch oft zur Seite gestanden ist und mir auch seine Hilfe angeboten hat.

Weiterhin danke ich Malakin, welcher mir bei einigen Problemen im technischen Bereich mit seinem Know-How zur Hilfe kam und mir auch oft geholfen hat grobe Fehler in den Konzepten auszumerzen.

Ich möchte auch all meinen anderen Freunden, die ich nur über das Internet erreichen kann, danken und hoffe auf eine lange Bekanntschaft und Zusammenarbeit.

Schlussendlich will ich an dieser Stelle all jenen Danken, die an den Projekten und Büchern gearbeitet haben, die ich für diese Arbeit benutzt habe. Es wäre ohne ihre Vorarbeit und Bemühung nie für mich möglich gewesen auch nur mit der Arbeit zu beginnen. Sie haben den Grundstein und die Grundwerkzeuge bereitgestellt, die nötig waren um mir eine zielgerichtete und produktive Arbeitsumgebung zu ermöglichen.

6.2 Tools, Webseiten & Bücher

Programme, Libraries & Tools

- Sun Java 6 JDK <http://jdk6.java.net/>
- LightWeight Java Game Library (OpenGL & OpenAL) <http://www.lwjgll.org/>
- Slick-Util (Teil von Slick2D) <http://slick.cokeandcode.com/>
- TymoonNET/NexT library <http://tymoon.eu>
- NetBeans IDE 6.9 <http://netbeans.org/>
- GIT Version Control <http://git-scm.com/>
- SmartGIT GIT Client <http://www.syntevo.com/smartgit/>
- Paint Tool SAI <http://www.systemax.jp/en/sai/>
- Adobe Photoshop CS3 <http://www.adobe.com/products/photoshop.html>
- Renoise <http://www.renoise.com>
- Audacity <http://audacity.sourceforge.net/>

Webseiten

- Stack Overflow <http://www.stackoverflow.com>
- Stack Exchange/Gamedev <http://gamedev.stackexchange.com/>
- Java 6 javadoc <http://download.oracle.com/javase/6/docs/api/>
- LWJGL Forum <http://www.lwjgll.org/forum/>

Bücher

- The Art of Game Design (Englisch) <http://artofgamedesign.com/>
- OpenGL The Red Book (Englisch) http://www.opengl.org/documentation/red_book/