Spieleentwicklung und die zweiten 90% – Feinschliff und Details

Description

Eine alte Regel der Spieleentwicklung besagt: „Nach den ersten 90% kommen die zweiten 90%!“ Das gilt nicht nur für Spieleentwickler, sondern bei nahezu allen kreativen Arbeiten. Aber was ist damit eigentlich gemeint? Dieser Artikel zeigt, worauf zu achten ist.

Der Prozess der Spieleentwicklung

Ein Spiel zu entwickeln ist nicht nur ein sehr kreativer Prozess, sondern auch verdammt viel Arbeit. Eine Grundregel lautet: Wenn man ein richtig gutes Spiel machen möchte, ist jedes Spiel – und sei es nur ein **Tetris** – mit extrem viel Arbeit verbunden.

Was man unter einem „richtig guten Spiel“ versteht, hängt u. a. vom Genre und der Zeit ab, zu der es entwickelt wurde. Will man – um beim Beispiel zu bleiben – ein Tetris erstellen, wie es in den 1980er Jahren üblich war, hält sich der Aufwand tatsächlich in überschaubaren Grenzen. In den 2020ern hingegen gibt es viel mehr Featurewünsche und vermeintliche Kleinigkeiten, die man entwickeln muss, damit es bei den Spielern gut ankommt und nicht so riecht wie Opas alte Socken.

Workflow der Spieleentwicklung

Am Anfang steht eine Idee. Manchmal ist sie sehr konkret, häufig aber nur sehr grob und wird im Laufe der Entwicklung verfeinert. So kann die Grundmotivation daraus bestehen, einen Ego-Shooter entwickeln zu wollen. Oder etwas konkreter: „Ein Spiel wie **Doom**“. Natürlich will man nicht alles 1:1 kopieren und so denkt man sich neue Features und Eigenschaften aus, die es möglichst in dieser Form noch nicht gab.

Blake Stone – Aliens of Gold (1993) basiert auf der Wolfenstein 3D-Engine, besteht aber aus unzähligen kleinen Details

Anschließend überlegt man sich die technische Umsetzung. Nimmt man eine bestehende Engine oder programmiert man alles selbst? Wie sollen die Grafiken aussehen? Gepixelt? Gerendert? Fotos oder als 3D-Modelle? Wie soll die Steuerung sein? Wie soll die Musik klingen? Welche Einstellungsmöglichkeiten soll es geben? Die Liste der Fragen, welche vom Game Designer beantwortet werden müssen, ist schier endlos, aber nur der Anfang.

Gleiches gilt für die Frage, ob man alles alleine machen möchte, im Team arbeitet oder man auf fertige Assets aufbaut, die man ggf. kaufen muss.

“Bämm! Headshot – Voll ins Knie!” – Spiele wie Doom II machen heute noch Spaß

Anschließend geht es an die Umsetzung. Man programmiert die Engine oder – falls vorhanden – die Spiellogik. I. d. R. beginnt man mit den wesentlichsten Dingen und arbeitet sich zu den unwesentlicheren vor. Um beim Doom-Beispiel zu bleiben: Man erstellt dreidimensionale Räume, Steuerung, Kollision, Schießen, Gegner, Items zum Aufsammeln, Aufzüge usw. und landet dann irgendwann bei Laden und Speichern, Statistiken und Menüs.

Die ersten 90%!

Die ersten 90% sind normalerweise abgeschlossen, wenn alle wesentlichen spielerischen Merkmale eingebaut sind. Häufig hat man bereits viele kleinere Features implementiert, das Spiel läuft relativ rund und möglicherweise sind bereits Musik, Soundeffekte, Partikel und zusätzliche Animationen eingebaut. Viele Dinge, etwa das Leveldesign, wurden bereits ordentlich getestet und es lassen sich kaum noch größere Fehler finden.

Zu diesem Zeitpunkt hat man normalerweise bereits einige Screenshots und Videos auf Social Media veröffentlicht und wenn man Glück hat, haben sich bereits ein paar potentielle Spieler gemeldet, die es kaum noch erwarten können.

Der “Boss-Key” ist eines dieser unvergesslichen Details

Als Entwickler neigt man dazu, hier die Bremse zu ziehen und langsam zur Veröffentlichung über zu gehen, da sich alles fertig anfühlt und die Einträge im Designdokument abgearbeitet wurden. Doch nun ist es eigentlich an der Zeit, sich an die zweiten 90% zu machen, nämlich dem Feinschliff!

Was ein großartiges von einem guten Spiel unterscheidet!

Es gibt sehr viele gute Computerspiele, die Anzahl der großartigen Spiele ist aber deutlich geringer. Eines der wesentlichen Unterschiede sind die „1000 Kleinigkeiten“, die es eigentlich nicht gebraucht hätte, aber dazu gehören, wenn man Spieler dauerhaft begeistern will. Scheinbar unnötige Elemente, die zum Teil nicht bewusst von Spielern wahrgenommen werden. Sie werden aber unbewusst wahrgenommen und würde man sie wieder streichen, dann würde tatsächlich viel fehlen.

Und ja: Dies klingt bisher bewusst sehr allgemein, weil es sich sehr stark vom Genre und der Zielgruppe unterscheidet. Schauen wir uns hierfür einige mögliche Optimierungen an.

Effekte

Ein gegnerisches Raumschiff kann einfach nur in Flammen aufgehen oder explodieren. Aber hier kann man häufig sehr viel optimieren. Partikel, die „spritzen“, Lichtblitze, Trümmerteile und vieles mehr kann man einbauen. Verschiedene Arten von Mündungsfeuer, Rauch, Blitze, Lichter, Verzerrungen. Das Thema der Effekte ist schier endlos und es lassen sich viele Dinge erweitern und verbessern. Der Spieler bemerkt natürlich nicht jede dieser Änderungen bewusst, aber wenn man sich als Entwickler bemüht, macht das Gesamtbild sehr viel mehr her – auch auf Screenshots.

Gerade mit Partikeln und Lichtern kann man sehr viel erreichen. Da solche Effekte i. d. R. im Spiel häufig vorkommen, lohnt es sich, viel Arbeitszeit zu investieren. Das gilt nicht nur für Actionspiele. Als Beispiel seien Match-3-Spiele erwähnt. Selbst so ein simples Spiel kann durch entsprechende Effekte enorm aufgewertet werden.

Animationen

Grundsätzlich sollte man zwischen Echtzeit und vorberechneten Animationen unterscheiden. Letzteres sind normalerweise gepixelte oder gerenderte Bildfolgen. Hier wiederum unterscheidet man gerne zwischen Animationen, die für das Spiel zwingend benötigt werden und Animationen, die das Spiel zusätzlich aufwerten.

Dazu ein konkretes Beispiel: Angenommen, du machst ein Point-and-Click-Adventure. Pflicht sind alle Animationen, welche deine Figur durch den Raum bewegen. Zusätzliche Animationen sind diejenigen, die nur selten oder gar einmalig im Spiel gezeigt werden. Etwa das Greifen nach Objekten (selten) oder das einmalige Krabbeln in ein Erdloch. Gerade die LucasArts-Abenteuer zeichneten sich durch zahlreiche einmalige Animationen aus, die nicht nur für die reine Optik, sondern auch Immersion verantwortlich waren. Häufig sorgten sie auch für heitere Momente – etwa Slapstick-Einlagen.

> Im Menü von Earthworm Jim 1 tanzen die Buchstaben beim ausgewählten Menüpunkt rauf und runter

Echtzeitanimationen sind zwar mit Programmierarbeit verbunden, sorgen aber in einer eher statischen Welt für Bewegung. Das kann ein simpler Sternenscroller im Menü sein, der auf jeden Fall eine bessere Wahl ist als ein statisches Bild. Es kann aber auch das Menü selbst sein. Etwa der Rahmen/Hintergrund, der sich schnell öffnet oder schließt. Oder wackelnde Buchstaben beim ausgewählten Menüpunkt, wie etwa im Menü von **Earthworm Jim** 1.

Die zwei Fragen, die man sich dabei stellen muss:

1. Was ist theoretisch möglich?
2. Was davon passt gut zum Spiel?

Easter Eggs

Spieler lieben Easter Eggs! Für Entwickler ist das oft ein ziemlich großer Aufwand und mit dem Risiko verbunden, dass diese virtuellen Ostereier nicht gefunden werden. Aber sie gehören zu den besonderen Momenten, über die gerne Videos gemacht und berichtet wird.

Das erste Easter Egg wurde bereits im Spiel **Adventure** von 1980 eingebaut. In diesem Textadventure versteckte Warren Robinett seinen Namen im Spielcode. Durch Eingabe von „XYZZY“ konnte der Spieler einen geheimen Raum betreten, in dem Robinett sich mit einer Nachricht verewigte. Diese Buchstabenkombination wiederum wurde als Hommage bei späteren Spielen gerne als Cheat benutzt. Das **Minesweeper**-Spiel unter älteren Versionen von Microsoft Windows verfügte über einen Cheat-Modus, der durch die Eingabe des Befehls `xyzzy`, das anschließende Drücken der Tastenfolge `Umschalt` und dann `Eingabe` ausgelöst wurde und ein einzelnes Pixel in der oberen linken Ecke des gesamten Bildschirms in ein kleines schwarzes oder kleines Pixel verwandelte weißer Punkt, je nachdem, ob sich der Mauszeiger über einer Mine befindet oder nicht.

https://www.bytegame.de/wp-content/uploads/2024/04/Minesweeper-cheating.mp4

Minesweeper ist mit dem Cheat ein Kinderspiel

In **Diablo II** gelang es Spielern durch Kombinieren bestimmter Gegenstände, ein Portal zu einem geheimen Level zu öffnen, das voller Kühe und anderer seltsamer Kreaturen war. Das „Kuh-Level“ wurde zum Kultphänomen.

In **Torchlight 2** huldigten die Entwickler dem Schöpfer von **Minecraft**, Notch, mit einer geheimen Mine namens „Notch’s Mine“. In der Mine konnten Spieler einzigartige Gegenstände und Ressourcen finden, die eine Anspielung auf Minecraft waren.

Auch in **Quake** gibt es zahlreiche Easter Eggs. Nur ein Beispiel: In der Crypt of Decay (E2M3) befindet sich ein einzigartiges verstecktes Gebiet, in dem der mythische Dopefisch wohnt. Der Name des Ortes sowie die dort verwendeten Texturen sind eine Anspielung auf id Softwares früheste Spieleserie – **Commander Keen**. Der Brunnen der Wünsche ist ein spezieller Unterwasser-Level in **Commander Keen 4: Secret of the Oracle**.

Easter Egg in Quake I

Wie man sehen kann, sind manche Easter Eggs sehr aufwändig, aber wenn das Spiel erfolgreich ist, sorgen diese Momente für große Aufmerksamkeit.

Umgebungsdetails

Darunter versteht man kleine visuelle Elemente wie fliegende Blätter, glitzernde Partikel oder sich bewegende Wolken, die zur Atmosphäre beitragen, aber keinen direkten Einfluss auf das Gameplay haben. Gerade Spiele, die ansonsten eher statisch wirken, wie etwa Point-and-Click-Adventures, können davon stark profitieren. Da kann bereits ein tropfender Wasserhahn viel ausmachen.

Hintergrundgeschichten

Zusätzliche Geschichten, die in der Spielwelt existieren und die Hintergrundgeschichte vertiefen, aber nicht unbedingt für das Verständnis oder den Fortschritt des Spiels erforderlich sind. Damit sind aber nicht zusätzliche Quests gemeint, sondern eine weitere, tiefere Ebene der Story und der Welt. Etwa Tagebücher von NPCs, Audioaufnahmen von vorherigen Abenteurern usw.

Dekorative Objekte und Details

Zusätzliche Objekte oder Elemente in der Spielwelt, die dazu dienen, die Umgebung zu verschönern oder interessanter zu gestalten, ohne dass sie für das Spielgeschehen von Bedeutung sind. Das können weitere Pflanzen in einem RTS sein, nicht benutzbare Objekte in einem Ego-Shooter oder zusätzliche Statisten in einem Adventure, die lediglich die Aufgabe haben, eine Szene lebendiger wirken zu lassen.

Weitere Optimierungen

Viele der bereits genannten Punkte fallen in den Bereich *Polishing*. Doch nicht nur grafisch und akustisch kann optimiert werden. Wichtig ist u. a. die Performance-Optimierung, welche ebenfalls in die zweiten 90% fällt.

Die Identifizierung und Behebung von Leistungsproblemen, um sicherzustellen, dass das Spiel auf verschiedenen Plattformen und Geräten flüssig läuft und keine unnötige Ressourcennutzung aufweist, ist bei einem Spiel extrem wichtig. Selbst wenn das Spiel auf dem eigenen PC flüssig läuft, sollte man die Framerate im gesamten Spiel messen und schauen, an welchen Stellen sie einbricht. Diese Stellen sind Hinweise darauf, dass es auf schwächeren Computern zu Rucklern oder anderen Problemen, etwa dem Timing, kommen könnte.

Die Idee für die drei Lösungswege in “Indiana Jones and the Fate of Atlantis” kam ziemlich spät auf und kostete weitere sechs Monate Arbeit. In dieser Szene mit Sophia gibt es gleich mehrere einmalige Animationen. Der Aufwand hat sich gelohnt: das Spiel wurde ein Hit

Behebung von Fehlern und Problemen, die während der Entwicklung oder in späteren Testphasen aufgetreten sind, um sicherzustellen, dass das Spiel stabil und fehlerfrei läuft, sind ebenfalls sehr wichtig. Da es sich, je nach Komplexität des Spiels, um sehr viele Fehler handeln kann, sollte man mit einem Bugtracker arbeiten.

Das Balancing ist vor allem, aber nicht nur, beim Multiplayerspielen ein langwieriger Prozess. Anpassung von Spielmechaniken, Schwierigkeitsgraden und anderen Variablen, um ein ausgewogenes Spielerlebnis zu gewährleisten, das sowohl anspruchsvoll als auch fair ist.

Beispiel Braid

Das Indie-Spiel Braid wurde von Jonathan Blow entwickelt und 2008 veröffentlicht. Der Prototyp mit allen Funktionen war bereits nach rund einer Woche fertig. Dies kann als die ersten 90% angesehen werden. Für den Rest brauchte Jonathan Blow ein ganzes Jahr.

Je nach Spielmechanik und Umfang kann es also einen gewaltigen Unterschied zwischen den ersten und den zweiten 90% geben. Es ist nicht unüblich, dass diese Optimierungen wesentlich länger dauern, als die Entwicklung des Kernspiels.

Die dritten 90%?

Zugegeben, das ist kein offizieller Ausdruck. Es soll hier lediglich kennzeichnen, dass es nach der Veröffentlichung weitergehen kann, nicht nur bei AA oder AAA Titeln. Vor allem kleinere und/oder kommerziell erfolglose Titel zeichnen sich dadurch aus, dass man rund um Version 1.0 nichts mehr von den Entwicklern hört. Manche Indie-Entwickler warten dann, dass das Spiel *von selbst* erfolgreich wird, und sind erst ab einer bestimmten Zahl von Spielern oder Downloads bereit, weiter zu arbeiten. Dabei wird eher umgekehrt ein Schuh daraus: Gute Pflege eines Spiels kann für mehr Spieler sorgen.

So sind bspw. auf Steam bestimmte Werbeaktionen nur in Verbindung mit neuen Updates verbunden. Dabei kann es sich um Bugfixes, kosmetische Verbesserungen aber auch mehr Content handeln. Idealerweise geht man dabei auf die Wünsche der Spieler ein – sofern vorhanden. Wenn nicht, wird man selbst zum schärfsten Kritiker seines Spiels und überlegt, wie man die Details verbessern, Fehler beheben und den Content ausbauen kann. Das garantiert nicht zwangsläufig den Erfolg, aber wenn man als Entwickler selbst nichts tut, kann man nicht erwarten, dass sich neue Spieler finden lassen, zumal jedes Update ein guter Grund ist, um als Entwickler darüber zu berichten und auf sein Spiel aufmerksam zu machen.

Weiterführende Links

Die 22 PIXAR-STORYTELLING-REGELN – und was man als Gamedesigner daraus lernen kann
Cheating in Spielen – und was man dagegen tun kann
Die Kunst der Geschwindigkeit: Speedruns in Computerspielen
Lernkurven in Spielen
Die hohe Kunst des Rätseldesigns

Externe Links

Bugtracker in der Wikipedia

Date Created
26. April 2024
Author
sven