



Team oder solo?

Description

Seit geraumer Zeit fällt mir besonders auf Twitter auf, dass sich viele Spieleentwickler als „solo Developer“ beschreiben. Worin liegen die Vor- und Nachteile, Spiele ganz alleine zu machen?

Persönlicher Hintergrund

Ich fing 1993 an, mein erstes Spiel zu entwickeln. Ein [Textadventure in QBasic](#). Es war total billig gemacht. Statt eines Parsers musste man an jeder Stelle exakt das Richtige eingeben. Anfang 1994 war es dann endlich fertig. Ohne Bücher, Videos und Internet war es viel harte Arbeit. Und ich hatte Hilfe. Mit Charly als zwei Mann-Team war es für mich, mit vierzehn Jahren, gerade so zu stemmen.

Später war ich ein Teil verschiedener Gruppen, die Spiele, [Intros](#) und Demos [in der Demoszene](#) veröffentlichten. Kaum jemand arbeitete alleine, was unterschiedliche Gründe hatte. Fast niemand war in sämtlichen Disziplinen sehr gut. Außerdem dauerte es Ewigkeiten, wollte man alles solo tun.

In den folgenden Jahren arbeitete ich in kleineren, manchmal etwas größeren Teams an diversen Projekten, doch die Solo-Projekte nahmen immer mehr Raum ein. Diese Spiele waren i. d. R. deutlich bescheidener, hatten jedoch durchaus ihren Reiz. Fast 30 Jahre nach meinen ersten Zeilen in QBasic kann ich aber beide Seiten ganz gut verstehen.

Teams sind ein Privileg

Früher war es für mich völlig normal, in einem Team zu arbeiten. Diese entstanden im Freundeskreis, über das Internet oder – in den 1990er Jahren – durch Kontaktanzeigen einer Zeitung. In der Jugend hat man eher das Glück, mit Freunden zusammen zu sein, die das gleiche Hobby teilen. Bei mir fing es mit Schulfreunden an und irgendwie landete ich mehr oder weniger automatisch in „diesen Kreisen“. Zu Beginn der 2000er konnte man in Foren sehr oft Gesuche lesen, etwa auf Developia. Auf einschlägigen Entwicklertreffen in Deutschland hatte man die Möglichkeit, neue Leute kennenzulernen. Zwar hatten die meisten auch eigene Projekte, aber man kam zusammen. Irgendein Programmierer hatte seine selbstgeschriebene Spieleengine und es fanden sich Menschen, die an der Vision, ein

eigenes Spiel zu entwickeln, teilhaben wollten.



Symbolbild für Team Spirit. Aus der Theorie besser bekannt als aus der Praxis

Doch das ist nicht selbstverständlich – und heute noch weniger als damals. Teams, die ihre Freizeit für ein gemeinsames Projekt opfern, bergen Schwierigkeiten. Unterschiedliche Charaktere in verschiedenen Altersstufen mit individuellen Vorstellungen müssen gemanagt werden. Interessanterweise ist das in jungen Jahren einfacher. Die Fähigkeiten sind zwar geringer, die intrinsische Motivation aber deutlich höher. Der eigene Antrieb, das Vorhaben zu beenden, übersteht sehr viele Konflikte. Wenn man jahrelang solche Projekte abwickelt, sinkt der Ansporn etwas, dafür steigen die Ansprüche an Teammitglieder und deren Fertigkeiten, da man es darunter ja auch selbst machen könnte.

Teams haben jedoch auch große Vorteile. Man muss nicht jede Entscheidung selbst treffen, sich nicht um alles kümmern und im besten Fall kann man sich auf sein eigenes Fachgebiet konzentrieren. Zumindest in der Theorie.

Was muss ein Spieleentwickler können?

Die Fachgebiete für ein komplettes Spiel sind seit den 1990er Jahren ziemlich identisch. Ein Jahrzehnt davor konnte man noch auf Sound verzichten. Wer ein Game ganz alleine erstellen will, sollte folgende Dinge beherrschen:

- Programmierung
- Grafik und Animation
- Soundeffekte und Musik
- Game-Design
- ggf. Leveldesign

Dies sind die Hauptzutaten. Dass die Spiele auch an Gamer gebracht werden wollen, kommt oben drauf. Selbst bei einem kostenlosen Spiel hätte man gerne etwas mehr Spieler als den eigenen Freundeskreis, sofern vorhanden.

Das ist sehr viel. Wer nicht gerade „Wunderkind“ mit zweitem Vornamen heißt, braucht ziemlich lange, um alles zu beherrschen. Meiner Erfahrung nach werden dabei vorwiegend die Game-Design-Elemente massiv unterbewertet. Darin stecken zahlreiche Unterbereiche und es ist viel mehr, als lediglich ein paar wichtige Entscheidungen zu treffen. Diese hängen jedoch vor allem mit Genre und Umfang des Games zusammen. Bspw. muss nicht jedes Spiel eine epische Story haben, aber es hilft dennoch ungemein, wenn man gut schreiben und Geschichten erzählen kann.

Organisation

Wer ein Spiel alleine macht, muss erstaunlich wenig organisieren. Meistens schreibt man ein paar Ideen in eine Textdatei und schaut gelegentlich hinein. Der Rest ergibt sich – mehr oder weniger – von selbst. Im besten Fall lagert man die Daten in einer Cloud aus, setzt sich noch Meilensteine und viel mehr ist nicht zu tun.

In einem Team sieht es anders aus. Ist man nur zu zweit oder zu dritt, kann man das Designdokument relativ klein halten, aber auch nicht immer. Viele Dinge bei der Spieleentwicklung bestehen aus Improvisation. Man stellt fest, dass irgendwelche Features nicht funktionieren oder man noch weitere Funktionen braucht – und dann gehen die Diskussionen los. „Das steht aber nicht im Designdokument“ oder „das war nicht so vorgesehen“ sind die üblichen Parolen, gefolgt von „das bringt uns auch keine 10.000 Spieler!“

Wenn man ein bestehendes Spielkonzept 1:1 kopiert, etwa Tetris, ergeben sich fast alle Features von selbst und es bedarf nicht viel Planung und Kommunikation. Sobald man eigenständige Ideen verfolgt und diese mit einem Team abarbeitet, beginnt der Eiertanz. Am besten führt man alles in einem großen Designdokument aus und verweist an entsprechenden Stellen darauf, dass gewisse Dinge noch nicht planbar sind. Allerdings bringt es weitere Nachteile mit sich. Je genauer man etwas erklärt, umso angreifbarer wird man.

„Du hast 10 Waffen definiert, jetzt willst Du aber 12 Waffen.“

Außerdem werden lange Designdokumente meiner Erfahrung nach nur selten gründlich gelesen und verstanden. Das führt regelmäßig zu Katastrophen.

Hinzu kommt die Betreuung. Mindestens einer im Team muss sich um alle kümmern. Ein Punkt, der total unterschätzt wird. Man sollte bspw. meinen, dass sich mehrere Grafiker untereinander austauschen und auf Grafikstil, Formate etc. einigen. In der Praxis macht aber jeder sein Ding und am Ende passt nichts zusammen. Auch der Informationsaustausch zwischen Programmierung und Grafik ist extrem wichtig, vor allem bei einer eigenen Engine, die von gängigen Standards abweichen kann. Wenn man hier z. B. nicht Hand in Hand arbeitet, um Workflows für Animationen zu erarbeiten, bekommt man am Ende nur einen großen Haufen Datenmüll fabriziert.

Organisation ist Kommunikation. Das erfordert viel Zeit, Planung und Geduld. Darüber sollte nicht nur einer im Team verfügen, sondern im Idealfall alle. Einen kommunikativen Krüppel an der falschen Stelle bringt das ganze Projekt zum Einsturz.

Sorgenfreie Entwicklung

Aus Sicht der Hobbyspieleentwicklung leben wir in einer wunderbaren Zeit. Niemand muss mehr eine eigene Engine schreiben, da es schon alles gibt. Egal ob 2D, 3D, [rein textbasiertes Spiel](#), es ist bereits vorhanden. In den meisten Fällen sollte man dennoch programmieren können, aber das Niveau für Gamecode ist fast immer deutlich niedriger als Enginecode. Da die Mehrheit der Engines für den Hobbybereich kostenlos sind oder nur wenig Geld kosten (früher konnte das in die Millionen gehen), gibt es kaum ein Argument für eine eigene. Außer, man will das tun oder hat einen Sonderfall, etwa eine ganz neue oder sehr alte Technologie. Möglicherweise noch ein Limit für die Dateigröße.

Ansonsten greift man zu einer der vielen Engines, besorgt sich ein Buch und/oder Online-Tutorials und schon geht es los.

Assets machen das Leben zusätzlich leichter. Musik und Grafik gibt es wie Sand am Meer. Oft kostenlos, manchmal sind sie für wenige Euro zu haben. Und wer es noch einfacher haben will, holt sich zur Engine eine Spielvorlage, etwa für ein RTS. Diese enthalten meist schon die GUI, Spiellogik und in Ansätzen eine kleine KI, mitunter für Wegfindung. Daraufhin passt man alles seinen Wünschen an und erweitert es so lange, bis man zufrieden ist.

2022 ist niemand mehr auf ein Team angewiesen. Ja, als Soloentwickler muss man sich um alles kümmern, aber wenn das, was man haben will, im Asset-Store mit drei Klicks zu haben ist, so ist das wesentlich nervenschonender als mit einem Team zu arbeiten. Und eine eigene Engine kann mit den Features von Unity, Unreal Engine, Godot oder GameMaker ohnehin nicht mithalten. Dabei geht es oft nicht einmal um die grafischen Besonderheiten, sondern die vielen kleinen Optionen und Hilfen, die einem die tägliche Arbeit, etwa Levels erstellen, erleichtern.

Disziplin!

Ein mittelgroßes Spiel selbst zu entwickeln erfordert viel Disziplin. Solche Projekte können mehrere

Jahre andauern. Eine Zeit, in der einiges passieren kann. Das ist bei Teams problematisch, aber wenn man gute Leute hat, motivieren sie sich gegenseitig. Normalerweise verspürt man Motivation, sobald die Bemühungen der Anderen sichtbar sind. Ausfallzeiten über einige Monate lassen sich ebenfalls kompensieren, sofern es sich nicht um Dinge handelt, die das ganze Projekt blockieren.

Entwickelt man alleine, betrifft jeder Ausfall 100 % des Spiel. Und je länger man nicht daran arbeitet, umso schwerer fällt einem die Motivation. Das merke ich an mir selbst. Ideen sind schnell aufgeschrieben, erste Konzepte oder ein Prototyp zügig fertiggestellt. Dann beginnt die eigentliche Arbeit, die einen nicht mehr richtig motivieren will. Vor allem, wenn man bereits rund 20 Spiele entwickelt hat, wird es immer schwieriger, sich für etwas Großes aufzuraffen.

Doch auch ein Team braucht Disziplin. Seien es inhaltliche Vorgaben, stilistische oder technische Dinge. Wenn ein Grafiker laufend Objekte mit viel zu vielen Polygonen erstellt, ist das ziemlich großer Mist. Und es ist nervig, wenn die Dateien in den falschen Ordnern landen und auch die Dateinamen nicht vereinbarten Konventionen entsprechen. Ebenso anstrengend sind Typen, die sich nicht um ihre eigenen Tasks bemühen und nicht in der Lage sind, Bescheid zu geben, wenn sie etwas fertiggestellt haben. Teamwork erfordert viel Disziplin, nur weiß man vorher leider nie, wer dieser Herausforderung gewachsen ist.

Die Kunst der Demotivation

Für Teams wird immer wieder das Argument der Motivation angeführt. In der Theorie und auch manchmal in der Praxis trifft dies zu: Innerhalb eines guten Teams können sich die Mitglieder motivieren und zu Höchstleistungen antreiben. Doch in vielen Fällen ist genau das Gegenteil der Fall.

Wenn andere nichts tun, wird man selbst demotiviert. Das wirkt sich sogar auf Bereiche außerhalb der eigentlichen Spieleentwicklung aus. Man sollte bspw. meinen, dass eine Gruppe ein Spiel besser vermarkten kann als ein Einzelner. Jeder hat sein Netzwerk auf bekannten Plattformen und wenn alle die Inhalte teilen, erreicht man mehr Leute. In der Praxis hingegen stellt sich gerne eine Mischung aus „die Anderen machen das schon“ und „ich teile nur meine Werke“ ein. Das ist noch ein günstiger Fall.

Es geht eine Stufe extremer. Die Teammitglieder weigern sich, Marketing zu betreiben. Das beginnt mit „ich kenne niemanden“ und endet irgendwo bei „ich hätte schon etwas bei Reddit gepostet, aber da gibt es keine Foren.“

Am Ende bleiben Marketing, Organisation, Kommunikation und viele andere Aufgaben an einer Person haften, die irgendwann völlig entnervt aufgibt, woraufhin die Verwunderung des sog. „Teams“ grenzenlos ist.

Ein Team funktioniert nur, wenn sich jeder für das Ziel voll einbringt und es keine Ego-Show wird, in der individuelle Befindlichkeiten Vorrang haben. Da man heutzutage vergleichsweise einfach alleine hochwertige Projekte auf die Beine stellen kann, scheinen immer weniger Personen für gute Teamarbeit bereit zu sein.

Teilweise Spieleentwickler

Die Frage ist, ob man sich überhaupt die Mühe machen will, ganze Spiele zu kreieren. Manche wollen sich nur auf einen Teilaspekt, also Grafik, Sound oder Code konzentrieren. Und so spezialisieren sich immer mehr Entwickler darauf, Assets zu erstellen. Egal ob Objekte, Sprites, Animationen, Texturen, Soundeffekte, Musik oder irgendwas programmiertes: Die Erzeugung von Miniprojekten kann sehr motivierend sein und mit etwas Glück verdient man damit noch ein bisschen Taschengeld. Solche Projekte haben für den Einzelnen viele Vorteile. Sie sind vergleichsweise schnell abgeschlossen, man konzentriert sich auf einen kleinen Aspekt der Spieleentwicklung und wenn es richtig gut ist, landen die eigenen Werke nicht nur in einem Spiel, sondern in Dutzenden!

Letztlich mache ich kaum etwas Anderes. Statt laufend eigene Projekte zu starten, teile ich meine Erfahrungen in Textform und mache dabei das, was mir am meisten Spaß macht: schreiben. Man muss auch nicht immer dem nächsten großen Ding hinterherrennen. ?

Date Created

29. April 2022

Author

sven