



Game Design Workshops mit Schulklassen

Christoph Kaindel

In der spielebox gibt es seit einigen Jahren Workshops und Vorträge zu Computer- und Videospiele. In diesem neuen, von mir konzipierten dreistündigen Workshop-Angebot konnten Schulklassen über Games reflektieren und lernen, wie man mit einfachen Mitteln sein eigenes Spiel entwickeln kann. Die Workshoptermine waren nach wenigen Tagen ausgebucht. Eine Fortsetzung 2016 ist geplant.

1. Das Konzept

Unser Ziel war zu zeigen, welche vielfältigen kreativen Aufgaben in der Entwicklung eines Computerspieles zu bewältigen sind. Texte, Bilder, Animationen, Ton und Musik müssen produziert werden, bei der Programmierung werden mathematische Kenntnisse benötigt. Fast immer ist die Entwicklung eines Spieles Teamarbeit, alle sollen ihre jeweiligen Stärken und Interessen einbringen können. Viele kostenlose Werkzeuge sind mittlerweile verfügbar, die Spieleproduktion mit geringen technischen Mitteln – oft direkt im Internetbrowser – ermöglichen. So

eignet sich Spielentwicklung gut für praktische medienpädagogische Projekte.

2. Der Einstieg

Am Beginn stand eine kurze Diskussion über Spielvorlieben und -gewohnheiten. Nahezu alle Kinder hatten Spielerfahrung und die meisten spielten regelmäßig auf verschiedenen Geräten. Viele gaben an, das Smartphone eher zum Spielen als zum Telefonieren zu nutzen. Altereinstufungen von Spielen scheinen wenig wirksam zu sein, vielen Jüngeren waren Spiele wie Grand Theft Auto, Call of Duty oder Left 4 Dead nicht nur dem Namen nach bekannt.

Um in das Thema einzusteigen, haben wir das bekannte „Roboter-Spiel“ gespielt: Paarweise bewegen sich die SpielerInnen durch den Raum, wobei jeweils ein „Roboter“ – mit geschlossenen Augen – von einem Lenker/einer Lenkerin gesteuert wird und unfallfrei auf die gegenüberliegende Seite des Raumes gelangen soll. Dieses Spiel eignet sich ausgezeichnet, um im Anschluss über die Grundbestandteile eines Computerspieles zu diskutieren. Die meisten konnten dank ihrer digitalen Spielerfahrung die Gemeinsamkeiten identifizieren und auf Games übertragen: der Roboter als **Spielfigur**, die vom **Spieler/der Spielerin** gesteuert wird; der Seminarraum als **Spielwelt/Level**; die anderen Roboter und LenkerInnen als **Hindernisse**; die Beschreibung des Spieles als **Spielziel** bzw. **Spielregeln**. Diese Grundelemente wurden auf einer Flipchart festgehalten.

3. Praxis 1: Spielkonzepte entwickeln

Im ersten Praxisteil des Workshops wurden die SchülerInnen in Gruppen von 4–6 Personen eingeteilt. Deren Aufgabe war es, ein Spielkonzept zu entwickeln und den anderen SchülerInnen anhand eines Plakats zu präsentieren. Als Anhaltspunkt wurde ein Gesamtthema vereinbart, auf das sich alle Spiele beziehen sollten, etwa „Steinzeit“ oder „Entdeckungen“. Ansonsten war die einzige Einschränkung bei der

Ideenfindung, dass kein bereits existierendes Spiel neu erfunden werden sollte – denn sehr oft kam der Wunsch, einfach „FIFA 17“ zu beschreiben, oder „GTA 6“. Ein „GTA in der Steinzeit“ wäre allerdings zulässig und sicher spannend gewesen. Zur Unterstützung bei der Ausarbeitung gab es einen Katalog mit Fragen wie:

- Worum geht es in dem Spiel, was ist das Thema?
- Welchem Genre ist das Spiel zuzuordnen?
- Wie wird es gesteuert?
- Für welche Plattform (PC, Konsole, Smartphone) ist es gedacht?
- Welche Spielfiguren, welche Gegner gibt es?
- Welche Eigenschaften und Fähigkeiten haben sie?

Nach etwa 50 Minuten der Spieleentwicklung mithilfe von Schere, Kleber, Buntstiften und Flipchart, unterstützt von uns und den Lehrpersonen, wurden die Entwürfe präsentiert. Es sind eine Menge origineller Ideen entstanden, Abenteuer- und Rollenspiele, Jump and Run Games aus verschiedenen Genres, Sportspiele etc. Es zeigte sich die große Bandbreite der Spielerfahrungen der Kinder und Jugendlichen, und ihre Fähigkeit diese Erfahrungen kreativ in eigene Projektideen einzubringen. Anhand vieler kluger Nachfragen wurden die Konzepte von den EntwicklerInnen erklärt.

4. Praxis 2: Spiel programmieren

Im zweiten Praxisteil, ebenfalls etwa eine Stunde lang, haben wir anhand der visuellen Programmieroberfläche Scratch die Grundlagen der Spieleprogrammierung (bzw. des Programmierens an sich) vermittelt. Zur Einführung nutzten wir die Übung „Flappy Code“ in Code Studio. Innerhalb von 10 Minuten kann hier ein Flappy Bird Spiel nachgebaut werden. Da allen Kindern das Spielprinzip bekannt war und Bedienung und Bildschirmlayout von Code Studio dem Aufbau von Scratch nachempfunden ist, lässt sich diese Seite gut nutzen, um erste spielerische Erfahrungen mit dem Programmieren bzw. "Coding" zu machen. Code Studio erlaubt das Anlegen von kostenlosen LehrerInnen- und SchülerInnen-Profilen und bietet mehrstufige Programmierkurse mit

Übungen ohne Computer an. So eignet sich die Seite auch unabhängig von Scratch zur niederschweligen Einführung in die Informatik.

Wir aber sind nach Abschluss der Übung zu Scratch übergegangen. Hier war die Aufgabe, ein vorgegebenes Spiel Schritt für Schritt nachzuprogrammieren. In dem Spiel „Mäusejagd“ steuert man eine Maus, die Käsestücke einsammelt und dabei von einer Katze gejagt wird. Für jeden Käse erhält die Maus Punkte. Am Beginn stand das Programmieren der Steuerung der Maus mithilfe der Cursortasten. Dann wurde der Käse gezeichnet und programmiert. Am Ende wurde als zusätzliche Bedrohung die Katze eingefügt und mit einfachem Code versehen. Das Spiel konnte nicht in allen Workshops fertig programmiert werden, die Steuerung der Maus – ohnehin das Herzstück des Spiels – ging sich in jeder Gruppe aus. Einige Kinder haben bereits zusätzliche Funktionen eingebaut, etwa zwei Mäuse gegeneinander spielen lassen, Hintergründe verändert etc. Anhand dieses einfachen Spieles können grundlegende Bestandteile von Programmabläufen demonstriert werden, wie Befehle, Variablen, Bedingungen, Schleifen oder Boole'sche Operatoren. Die SchülerInnen haben die Oberfläche von Scratch kennengelernt und die ersten Schritte im Bereich von Bildbearbeitung und Animation gemacht.

Zur Auflockerung haben wir am Ende ein Flappy Bird ähnliches Spiel gespielt, das durch Mikrofoneingabe gesteuert wird. Indem sie laut oder leise waren, konnten die Kinder eine Fledermaus durch eine Höhle lenken. Eine solche Spielsteuerung ist in Scratch leicht umzusetzen, da es als Flash-Anwendung auf Mikrofon und Kamera zugreifen kann. Die komplexere Scratch-Modifikation Snap! kann das leider nicht mehr ...

Dieses Spiel kam sehr gut an und musste auf Wunsch der Kinder mehrmals gespielt werden. Spontan gab es auch Wettbewerbe zwischen Mädchen und Burschen – unfair allerdings, weil die Spielwelt zufällig generiert wird und jedesmal unterschiedlich schwierig ist. An der Erstellung fixer Levels arbeite ich noch. Alle LeserInnen sind herzlich eingeladen, das Spiel im Klassenraum auszuprobieren, es dauert wenige Minuten. Es ist, wie alle Scratch-Spiele, zur Modifikation freigegeben. Auch als Bewegungseinheit lässt es sich nutzen, indem etwa nur durch

Klatschen oder Stampfen Geräusche erzeugt werden dürfen ... Ich freue mich auf eure Ideen.

5. Fazit

Ich war selbst gespannt, wie das Konzept, das wir erstmals umgesetzt haben, funktioniert; abgesehen von kleineren Anpassungen, die wir spontan vorgenommen haben, ist es gut aufgegangen. Bei den Programmierübungen wurde der Computer meist paarweise genutzt. Da aber nur 10 Computerplätze zur Verfügung standen, mussten wir die Gruppen ab 20 Kindern teilen und Entwurfs- und Programmierpraxis parallel führen. Dadurch waren wir weniger flexibel in den zeitlichen Abläufen und mussten gerade die Programmierübungen abbrechen, bevor das Spiel fertig programmiert war. Die Motivation der meisten Kinder war jedenfalls groß, und meine Hoffnung ist, dass einige Geschmack am kreativen Coding gefunden haben und sich selbständig weiter damit beschäftigen. Mit Scratch lässt sich Programmieren ja selbst als ein Spiel begreifen, ein selbstgestelltes Rätsel, das es mit den gegebenen Möglichkeiten zu lösen gilt. Wie die zigtausenden mit Scratch erstellten Spiele, Präsentationen, Simulationen, Kunstprojekte usw. zeigen, lassen sich mit den auf den ersten Blick recht begrenzt erscheinenden Elementen sehr, sehr viele kreative Ideen umsetzen.