

Proposal

Rolling Balls

Fortgeschrittene Spiele-Entwicklung

von

Andreas Buck

und

Stefan Döbbert

5. Mai 2014

1 Einführung

Im Fach Fortgeschrittene Spieleentwicklung soll von den Projektgruppen ein Spiel in C++ und OpenGL erstellt werden. Hierbei soll der Fokus vor allem auf den grafischen Effekten liegen.

Unsere Projektgruppe hat sich für einen Ableger des bekannten Spiels Tower Defense entschieden und das Projekt Rolling Balls genannt.

1.1 Spielprinzip

Rolling Balls besteht aus einer einfachen Fläche als Spielumgebung. Ziel des Spiels ist es, rollende Kugeln zu zerstören, bevor diese das Ende der Spielfläche erreichen. Die Spielfläche wird in ein Grid unterteilt und der Spieler kann auf jedes Grid einen Turm bauen, der die heranrollenden Kugeln zerstören kann. In der ersten Implementierung werden die Kugeln jeden Turm, mit dem sie kollidieren beschädigen und zerstören, sofern sie nicht schnell von den Türmen bekämpft werden.

1.1.1 Schwierigkeitsgrad

Der Spieler kann den Schwierigkeitsgrad selber wählen, hierbei stehen verschiedene Möglichkeiten zur Schwierigkeitssteuerung zur Verfügung:

- Rollgeschwindigkeit der Kugeln (je schneller, desto schwerer)
- Breite des Spielfelds (je breiter, desto schwerer)

- Länge des Spielfelds (je länger, desto leichter)

1.1.2 Economy

Der Spieler braucht eine Menge X an Spielwährung, die er benötigt, um sich einen Turm zu kaufen. Für jede abgeschossene Kugel erhält der Spieler eine Summe an Geldwährung. Sobald der Spieler genügend Kugeln zerstört hat, kann er sich einen Turm kaufen.

1.1.3 Spielende

Es gibt kein fixes Spielende, jede Runde werden mehr Kugeln erzeugt, die aufgehalten werden müssen. Der Spieler hat eine bestimmte Anzahl an Leben und für jede Kugel, die das Ende des Spielfelds erreicht, wird dem Spieler ein Leben abgezogen. Sobald der Spieler keine Leben mehr hat, wird das Spiel beendet. Der Spieler erhält pro zerstörter Kugel Spielpunkte, welche am Ende als Highscore angezeigt werden. Um Exploits während des Spiels zu minimieren, bekommt jede Kugel einen Countdown-Timer. Sobald der Countdown-Timer abgelaufen ist, explodiert die Kugel und zerstört im Umkreis von 2 Feldern in alle Richtungen die Türme. Der Countdown-Timer wird für jede Zeile in Y-Richtung zurückgesetzt, d.h. man hat pro Y-Zeile eine gewisse Zeit, um die Kugel durchzuschleusen oder zu zerstören. Der Timer wird eingeführt, damit eine Kugel sich nicht eine endlos lange Zeit in ein und derselben Y-Zeile befindet.

1.1.4 Pathfinding

Eine Kugel überprüft, welche der nächsten Felder nicht mit einem Turm bebaut sind. Priorität hat immer die positive Y-Richtung, danach wird in positiver X-Richtung, danach in negativer X-Richtung und zuletzt in negativer Y-Richtung geprüft. Hat eine Kugel sich für eines der nächsten Felder entschieden, so wird dieses Feld für den Turmbau gesperrt, bis die Kugel das nächste Feld verlassen hat. Der Spieler muss dann die Turmbau-Aktion wiederholen, sofern das Feld gesperrt war.

2 Prototyp

Der Prototyp wird erstellt, um die generelle Spielidee vorzuführen. Hierbei wird auf THREE.js und WebGL zurückgegriffen. Im Prototyp werden folgende Funktionalitäten verfügbar sein:

- Darstellung des Spielfelds
- Einteilung des Spielfelds in Grids
- Erscheinen der Spielkugeln als THREE.Sphere
- Türme werden als THREE.Cube dargestellt
- Interaktion mit dem Spielfeld zum Bauen der Türme

chapterInteface Startbildschirm mit Optionen Startbildschirm enthält die Namen der Autoren, einen Start Button und eine Möglichkeit die Spieleinstellungen zu verändern

was brauchen wir für den Prototypen wer macht was welche abhängigkeiten (was muss vor etwas anderem erledigt sein) Projektplan

3 Ideensammlung

- Baum errechnen um schnellsten Weg der Kugel zum Ende des Spielfelds zu berechnen (rekursives Prüfen der Position)
- IntroScreen? Einstellung des Spielfelds
- Spielstart darf den Spieler nicht überraschen
- Pause-Funktion (optional)
- Pfadfindung mittels A-Star-Algorithmus