

Anforderungsumsetzungsliste

... des Spiels Super Rolling Ball von Dominik Heckner und Pascal Friedrichsen

Wir haben alles auf Exzellenz gemacht, wo man Exzellenz machen kann. Fremdcode und Fremd-Assets haben wir versucht so gering wie möglich zu halten.

Arbeitsblatt 1

1.1:

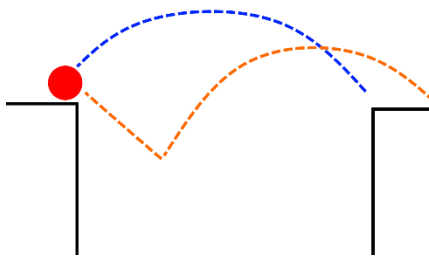
- Minimal: Anforderungen erfüllt

1.2:

- Objekte mit ProBuilder gebaut und bewegliche Plattformen verschiedener Art (Level 1001 hat bewegliche Plattformen, sowie auch Level 1003)

1.3:

- Minimal: Kugel kann sich bewegen und springen
- Standard: Keine Sprünge an Wänden
- Exzellenz: Es existieren Walljumps, man kann den Sprung aufladen und die Sprungaumladung wird als UI Element angezeigt. Springen erzeugt partikel.



Wir haben unser Springen so umgesetzt: Beim Berühren von Böden wird der Bool `_jumpAllowed` auf `true` gesetzt. Springen lässt sich nur aufladen, wenn `jumpAllowed` ist, also die Kugel noch ihren Sprung besitzt. Wir weichen damit ein bisschen der Anforderungen ab, da man so, falls man von einer Klippe springt, noch in der Luft springen kann. Dies ist aber von uns gewollt, da man so eine vielfältigere Steuerung hat, die das Spielerlebnis weniger einschränken.

Um das zu unterstützen wird beim Springen auch die auf der Kugel existierende Gravitationskraft wieder auf 0 gesetzt, damit die Kugel auch beim Springen von einer Klippe nach oben springt.

1.4:

- Alle Anforderungen sind umgesetzt. Die Kamera lässt sich orbital um die Kugel rotieren (dazu mehr bei Arbeitsblatt 4). Kamerafahrten zu Levelstarts sind implementiert, genau so wie das Scrollen. Die POV verändert sich mit der Kugelgeschwindigkeit.

Arbeitsblatt 2

2.1:

- Minimal: Start und Endszene existieren
- Standard: Halbwegs ansprechend gemacht durch Font
- Exzellenz:
 - Levelauswahlmenü und Speicherdaten Löschen Menü, sowie die Anzeige der gesammelten Münzen.
 -

2.2:

- Minimal: HUD erstellt
- Standard: HUD mit Lebenspunkt- Trefferpunkt- , Münz- und Sprunganzeige, sowie Anzeige zum aktuellen Kamerasteuerungsmodus
- Exzellenz: Pause-Funktion existiert

2.3:

- GameManager der alles handhabt ist implementiert. Die meiste Kommunikation mit dem GameManager ist mit ActionListenern implementiert. Freigeschaltene Level, Münzanzahl und Upgrades bleiben selbst nach Spielneustart gespeichert.

Arbeitsblatt 3

3.1:

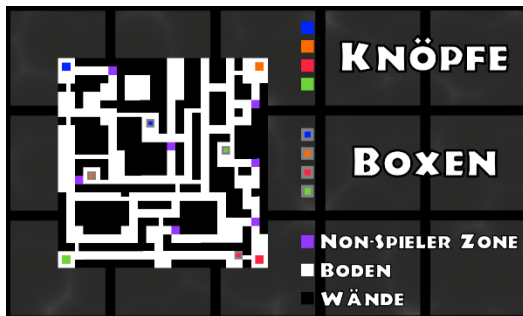
- **Statisch:** Stacheln auf den Boden, die auf und ab fahren
- **Beweglich:** Stachelball, der einen per NavMesh verfolgt
- **Schütze:** Geschützturm, der einen im bestimmten Radius anvisiert und dann schießt.

3.1 Exzellenz: Beweglich und Schütze sind besiegbare, in dem man auf deren Kopf springt. Todespartikel werden dabei erzeugt.

3.2:

- **Oberfläche:** Wasser, animiert und texturiert mit Shadergraph, welches die Kugel verlangsamt sobald sie im Wasser ist. Nach dem Verlassen des Wassers hat die Kugel für kurze Zeit den „Wet“ Debuff, welcher die Kugel verlangsamt.
- **Schalter:** Zwei verschiedene Schalter mit eigenen Animationen: Ein Schalter, den die Kugel benutzt um ein Rätsel zu lösen und ein Schalter auf den eine entsprechende Box muss, damit er aktiviert wird.
- **Physikalisch:** Eine Kette die mit Fixed Joints zusammenhält, welche die Kugel durch gegenspringen zerstören muss.

3.2 Exzellenz:



Karte des größeren Rätsels in Level 1003

- Verschiedene Boxen mit dazugehörigem Schalter, die auf die Schalter geschoben werden müssen um ein Rätsel zu lösen.
- Beim Lösen des Rätsels wird eine Animation abgespielt, in der sich der Weiterweg bildet.
- Die Kugel kann mit E Boxen in einem kleinen Radius an sich ziehen.
- Boxen respawnen mit Partikeleffekt, falls sie runterfallen

3.3:

- Minimal: Münzen
- Standard: Permanente Upgrades, die z.B. das Leben oder Trefferpunkte erhöhen.
- Exzellenz:
 - Sprunghöhe Pickup, welcher für kurze Zeit einen Buff gibt, der die Sprunghöhe erhöht.
 - Münzen und PermaUpgrades bleiben beim Spielneustart erhalten. PermaUpgrades verschwinden permanent, sobald sie 1 Mal aufgesammelt wurden.

Arbeitsblatt 4

4.1: Gegner und Objekte von Aufgabe 3 sind implementiert.

4.2 Minimal:

- Anstelle von Shift+Links und Shift+Rechts haben wir ShiftLinks und ShiftRechts, die Pfeiltasten Links und Rechts, sowohl auch die Mausbewegung nach links und rechts.
- C zum Kameramodus umstellen implementiert.
- Kugel rollt in Kamerarichtung bei W

4.2 Standard: Umgesetzt mit den Pfeiltasten oben und unten, sowie der Mausbewegung nach oben und unten.

4.2 Exzellenz: Eine Verfolgerkamera macht keinen Sinn, da somit die Steuerung in Sichtrichtung sehr nervig wäre, da sich die Sichtrichtung schnell umstellt. Dafür haben wir eine komplexere Kamerakonfiguration hinzugefügt:






Steuerung:

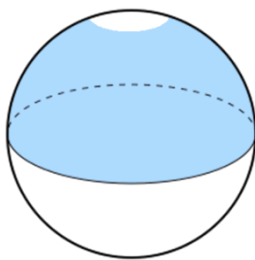
C – Wechselt den festen Kamera Modus

V – Wechselt ob X-Achsen Kamerasteuerung aktiviert ist

B – Wechselt ob Y-Achsen-Kamerasteuerung aktiviert ist

Das HUD/UI Element:

Feste Kamera	Nur X-Achse	Nur Y-Achse	Volle Bewegung	Keine Achse aber fest eigene Kameraposition
				



Dazu haben wir eingebaut, dass sich die Kamera nicht unter den Spieler positionieren kann und auch nicht komplett nach oben, da es sonst Probleme geben könnte beim ermitteln wo „vorne“ ist. In Blau sieht man jede mögliche Kameraposition, in Rot die „geblockten“.

4.3:

- Minimum: Drei Level existieren. Fehler sind bei mehrfachen Playtests nicht vorgekommen.
- Standard: Eigene Texturen und ProBuilder Modelle
- Exzellenz: Jedes Level hat ein eigenes Theme. Viele Partikeleffekte. Shadergraph und Post-Processing wurden auch benutzt.