

## Dokumentation Galaxy Invaders

### 1 Ziele

Prototyp für das Minispiel Galaxy Invaders  
Fertigstellung des Minispiels

### 2 Umsetzung

#### 2.1 Herangehensweise

In den letzten drei Wochen haben wir uns mit der Erstellung des Spiels Galaxy Invaders auseinander gesetzt. Zuerst haben wir uns zusammengesetzt um alle Kriterien des Spiels abzusprechen. Wir wollten ein Spiel programmieren, um auf unseren Programmieranteil zu kommen. Anschließend haben wir den geschätzten Aufwand gerecht aufgeteilt.

#### 2.2 Lösungskonzepte

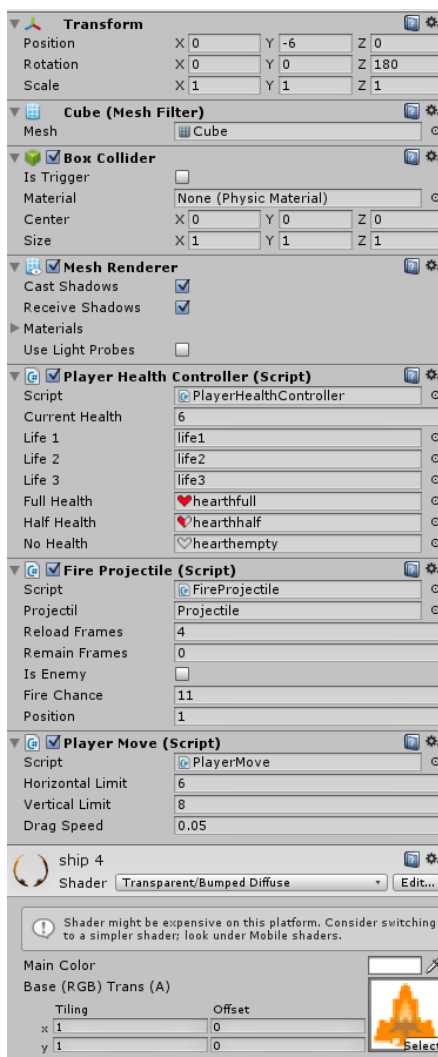


Abbildung 1 Inspector Player

Wir haben versucht alle Elemente in nicht spezialisierten Skripten zu handhaben. Dadurch konnte das selbe Skript für mehrere Bereiche verwendet werden, z.B.

FireProjectile. Dies wurde durch die Nutzung von möglichst vielen globalen Variablen realisiert. Der Aufbau in Unity3d erlaubt einen sehr modularen Aufbau. So sind alle Objekte mit ihren angepassten Eigenschaften verknüpft.

Im Bild ist der Inspector des Spielers zu sehen. Die Verwendung der globalen Variablen macht ein optimales Anpassen des Spielgefühls möglich.

Zu den eigenen Skripten kommen die Unity3d eignen Skripte, z.B. der Box Collider.

## 2.3 Beschreibung Ist-Zustand

- Der Spieler kann die Spielfigur steuern.
- Der Spieler kann Projektile abschießen.
- Gegner erscheinen.
- Gegner schießen zufällig Projektile ab.
- Es gibt 3 verschiedene Gegner.
- Die Spielfigur hat ein gewisses Leben.
- Die Gegner haben ein gewisses Leben.
- Die Spielfigur kann sterben.
- Die Gegner können sterben.
- Wenn die Gegner eine gewisse Höhe erreicht hat, verliert der Spieler.
- Es gibt einen Szenenwechsel wenn der Spieler kein Leben mehr hat.
- Es gibt einen Highscore.
- Wenn Gegner oder Projektile abgeschossen worden, wird der Highscore aufaddiert.



Abbildung 2 Spiel Ist-Zustand

## 2.3 Probleme/Tests

### Probleme

P1

Problem: Der Spieler konnte keinen Schaden nehmen. Zwei Projectile konnten sich nicht gegenseitig zerstören.

Ursache: Dies lag wahrscheinlich an einem Unity3d internen Bug innerhalb des Rigidbody und der Trigger Funktion.

Lösung: Die „EnemyProjectiles“ wurden mit einem Rigidbody ausgestattet.

P2

Problem: Der Spieler blieb auf einem Leben hängen.

Ursache: Der Szenenwechsel war noch nicht eingebaut.

Lösung: Ein Szenenwechsel wurde eingebaut.

### Tests

T1 Build&Run

Das Spiel wurde per Build&Run getestet. So konnte ein Großteil der Feinheiten angepasst werden. Des Weiteren wurde die Mehrheit der aufgetretenen Fehler auf diese Art entdeckt.

T2 Testen auf dem Endgerät

Die App wurde auf dem Endgerät getestet. Dort wurde sichtbar, dass Die komplette Skalierung nicht vollständig funktioniert. Dieser Fehler lässt sich mit dem aktuellen Unity3d beheben, welches uns bis dato nicht zur Verfügung stand.

## 3Fazit

### 3.1 Ziele erreicht?

Wir haben unser Minispiel fast vollständig fertigstellen können. Die Muss-Ziele sind erfüllt. Uns fehlt leider noch eine Unterscheidung wenn das Spiel über „Alle Spiele“ oder die Map gestartet wird. Des Weiteren fehlt eine vollständige Skalierung der App an die Bildschirmgröße, dies war im Rahmen der Zeit nicht möglich.

### 3.2 Zeitangabe zu den einzelnen Anforderungen

Programmieren: ~ 28 Stunden

Überarbeitung von Testkonzept/Designrichtlinien: ~ 2 ½ Stunden (Meise)

Dokumentation: ~ 3 Stunden

### 3.3 Was ist gut/schlecht gelaufen?

Die Arbeit innerhalb des Teams war gelungen. Die Aufgabenteilung hat sehr gut geklappt, wir haben jeweils unsere Aufgaben sauber erledigen können. Die Kommunikation außerhalb des Unterrichts ist noch verbesserungswürdig. Dennoch sind wir insgesamt mit unserer Leistung zufrieden, da wir innerhalb der kurzen Zeit fast alles erfüllen konnten.