MODUL 335

CsBe – Unity Projekt

Elias Kupeczki
10/02/2023

Inhaltsverzeichnis

Zeitplan

Table 1 - Zeitplan

	06-Feb	07-Feb	08-Feb	09-Feb	10-Feb
Planen	**0.5**	**0.5**			
Documentation Schreiben		**2**	**1**	**1**	**2**
Scene erstellen		**1**			
C# Classes erstellen		**2**	**2**	**2**	
Testing				**3**	**3**
GUI erstellen		**2**	**2**		
zusatz funktionen				**1**	**2**

Arbeitsjournal

Tag 1

Table 2 – Arbeitsjournal Tag 1

	06-Feb		07-Feb		08-Feb		09-Feb		10-Feb	
	SOLL	IST	SOLL	IST	SOLL	IST	SOLL	IST	SOLL	IST
Planen	**0.5**	**1**	**0.5**	**0.5**						
Docum-entation Schreiben			**2**	**3**						
Reflektion	Heute habe ich an der Planung für die Erstellung eines Pong-Spiels in Unity gearbeitet. Ich begann mit der Erstellung eines Plans für das Spiel, einschließlich der Regeln, Ziele und des Gesamtdesigns. Als Nächstes sammelte ich die Assets, die ich für die Erstellung des Spiels benötigte, z. B. Sprites, Hintergründe und Sounds. Danach habe ich an der Spiellogik gearbeitet, einschließlich des Punktesystems und der Spielmechanik. Schließlich testete ich das Spiel, um sicherzustellen, dass es richtig funktionierte. Es war eine Menge Arbeit, aber ich freue mich schon darauf, das Endergebnis zu sehen!									

Tag 2

Table 3 - Arbeitsjournal Tag 2

	06-Feb		07-Feb		08-Feb		09-Feb		10-Feb	
	SOLL	IST	SOLL	IST	SOLL	IST	SOLL	IST	SOLL	IST
Scene erstellen			**1**	**1**	**0**	**0.5**				
C# Classes erstellen			**2**	**2**	**2**	**2**				
GUI erstellen			**2**	**2**	**2**	**2**				
funktionen										
Reflektion	Die Entwicklung von Pong in Unity war für mich eine Herausforderung, da ich zum ersten Mal C# und Unity verwendet habe. Ich war mit der Syntax und der Struktur des Codes nicht vertraut, aber er hatte ähnliche Komponenten wie Java. Ich musste lernen, wie ich mich in der Unity-Benutzeroberfläche zurechtfinde, die neu war, aber der Unreal Engine 5 ähnelte (mit der ich schon einige Erfahrung habe). Ich musste lernen, wie man Objekte erstellt, wie man Spiellogik erstellt und wie man sie mit der visuellen Schnittstelle verbindet.Anfangs fand ich es schwierig zu visualisieren, wie mein Code mit dem Geschehen auf dem Bildschirm zusammenhängt, weil ich es gewohnt war, den Ablauf meines Codes in Unreal mit den "Blueprints" visuell darzustellen. Ich musste viel recherchieren und experimentieren, um den Code und die Benutzeroberfläche zum Funktionieren zu bringen.Schließlich habe ich die Schwierigkeiten bei der Entwicklung von Pong in Unity überwunden, indem ich Schritt für Schritt vorgegangen bin und aus meinen Fehlern gelernt habe. Ich musste auch lernen, wie man den Code an die visuelle Schnittstelle bindet und wie man Fehler oder Bugs behebt, auf die ich stieß.									

Tag 3

Table 4 - Arbeitsjournal Tag 3

	06-Feb		07-Feb		08-Feb		09-Feb		10-Feb	
	SOLL	IST	SOLL	IST	SOLL	IST	SOLL	IST	SOLL	IST
Testing							**3**		**2**	
GUI erstellen							**0**	**1**		
Zusatz Funktionen							**1**		**2**	
Reflektion	Heute war ein schwieriger Tag, da ich die GUI einrichten und testen musste, was ich bisher gemacht habe, sowie die zusätzlichen Funktionen. Das war oft ein Problem, da ich testete, ob etwas funktionierte, versuchte, es zu reparieren oder zu verbessern und dann etwas anderes kaputt machte.									
was ich gelernt habe	Ich habe daraus gelernt, dass ich mich im Voraus besser vorbereiten sollte, denn wenn ich es nur einmal mache und später keine neuen Dinge hinzufüge, muss ich nicht alles wiederholt testen, falls die neue Funktion das Spiel kaputt macht.									

Tag 4

Table 5 - Arbeitsjournal Tag 4

	06-Feb		07-Feb		08-Feb		09-Feb		10-Feb	
	SOLL	IST	SOLL	IST	SOLL	IST	SOLL	IST	SOLL	IST
Documentation Schreiben							**1**	**1**		
C# Classes erstellen							**2**	**2**		
Testing							**3**	**3**		
zusatz funktionen							**1**	**1**		
Reflektion	Heute ging es hauptsächlich darum, die neuen Zusatzfunktionen wie Rumble und Persistenz so hinzuzufügen, dass sie in ihrem Bereich funktionieren und gut funktionieren, ohne das Spiel zu zerstören. Eines der ersten Dinge, die bei der Persistenz auftraten, war, dass sie nur den Spielstand speicherte, nicht aber den Highscore. Da Rumble nur auf meinem Telefon funktioniert, konnte ich es auch nicht mit der Remote-Unity-App auf meinem Telefon testen, was eine Weile gedauert hat, um herauszufinden, und musste sicherstellen, dass es auf dem Telefon funktioniert, also gab es eine Menge APKs zu installieren.									

Tag 5

Table 6 - Arbeitsjournal Tag 5

	06-Feb		07-Feb		08-Feb		09-Feb		10-Feb	
	SOLL	IST	SOLL	IST	SOLL	IST	SOLL	IST	SOLL	IST
Documentation Schreiben									**2**	**2**
Testing									**3**	**3**
zusatz funktionen									**2**	**2**
Reflektion	Heute wurde daran gearbeitet, die App insgesamt zu verbessern und die letzten Teile der App fertigzustellen. Dazu gehörten auch die zusätzlichen Funktionen und das Testen aller Dinge, die ich bisher gemacht habe, um sicherzustellen, dass alles gut funktioniert, und so gab es eine Menge Iterationen und Änderungen bei der Installation auf meinem Telefon, um sicherzustellen, dass alles gut funktioniert, und Wiederholungen									
was ich gelernt habe	Ich hätte die Tests während der Arbeit an dem Projekt durchführen sollen, so dass ich, sobald ich einen Abschnitt berührt habe, diesen fertigstellen könnte und nie wieder darauf zurückkommen müsste, was die Arbeit wesentlich einfacher machen würde, da ich, sobald das Spiel fertig ist, keine weitere Arbeit mehr erledigen müsste.									

Testkonzept

Das Ziel des Testkonzepts ist es, bestimmte Fehler im Unity Game zu finden und zu dokumentieren zu dokumentieren und nach einem standardisierten und strukturierten Verfahren zu beheben. Der Test dieser Anwendung besteht aus funktionalen und nicht-funktionalen Testfällen.

Testfälle

Dieses Testkonzept ist für das zu testende System, ein Pong-Spiel, das in Unity unter Verwendung von C# entwickelt wurde. Die relevanten Testfälle basieren auf der Kernmechanik des Spiels, wie der Bewegung des Balls, der Bewegung des Paddels und der Kollisionserkennung. Die erwarteten Ergebnisse für diese Testfälle sind, dass sich das Spiel wie erwartet verhält, dass sich der Ball und das Paddel bewegen und auf Benutzereingaben reagieren, und dass das Kollisionserkennungssystem genau registriert, wenn der Ball entweder das Paddel oder die Wand trifft.

Das Testkonzept prüft absichtlich nicht die Gesamtleistung des Spiels, wie z. B. die Bildrate und die Latenzzeit. Dies ist darauf zurückzuführen, dass das Testen dieser Elemente den Zugang zu spezieller Hardware erfordern würde, was nur im Rahmen von White-Box-Tests möglich sein wird.

Zu den relevanten Testmitteln für dieses Testkonzept gehören ein Computer, auf dem Unity und das Pong-Spiel laufen, sowie eine Maus, eine Tastatur und ein Touch für Benutzereingaben. Die Testmethode wird eine Kombination aus manuellen Tests sein, bei denen der Tester das Spiel spielt.

Durchführung und Auswertung der Tests

White Box

Ich werde das Spiel mehrmals spielen, um sicherzustellen, dass die Spielmechanik und die Regeln korrekt umgesetzt werden. Ich werde das Spiel mit verschiedenen Schwierigkeitsgraden und Eingaben testen, um sicherzustellen, dass das Spiel richtig reagiert. Ich werde das Spiel auch mit verschiedenen Kombinationen von Eingaben testen, um sicherzustellen, dass der Ball richtig von den Schlägern, Wänden und anderen Objekten abprallt. Ich überprüfe den Spielstand, um sicherzustellen, dass er korrekt ist und nach jeder Runde aktualisiert wird. Ich stelle sicher, dass alle Spielregeln und Ziele korrekt befolgt werden. Ich überprüfe auch, ob alle Spielelemente korrekt und in der richtigen Reihenfolge, mit den richtigen Farben und Texturen gerendert werden. Schließlich werde ich das Spiel testen, um sicherzustellen, dass alle Menüs und Bildschirme richtig funktionieren.

Black Box

Dazu wird das Spiel manuell gespielt, um sicherzustellen, dass es korrekt funktioniert, die visuellen Komponenten des Spiels getestet, um zu gewährleisten, dass alle Grafiken korrekt angezeigt werden, die Benutzeroberfläche getestet, um sicherzustellen, dass alle Schaltflächen und Menüs korrekt funktionieren, die Leistung des Spiels auf verschiedenen Geräten getestet und die Funktionen zum Speichern und Laden des Spiels getestet, um sicherzustellen, dass sie korrekt funktionieren.

Table 7 - Testfall 01

Testfall: 01	**Bezug: Überprüfen der Ballbewegung**
Ablauf	Starten Sie das Spiel und beobachten Sie die Bewegung des Balls.
Erwartetes Ergebnis	Der Ball sollte sich im Spiel bewegen.

Table 8 - Testfall 02

Testfall: 02	**Bezug: Überprüfen der Punkteerhöhung**
Ablauf	Starten Sie das Spiel, beobachten Sie den Punktezähler
Erwartetes Ergebnis	Die Punkte sollten sich erhöhen, wenn der Ball die Paddles trifft.

Table 9 - Testfall 03

Testfall: 03	**Bezug: Überprüfen der Wandkollision**
Ablauf	Starten Sie das Spiel, beobachten Sie die Ballkollision
Erwartetes Ergebnis	Der Ball sollte von den Wänden abprallen.

Table 10 - Testfall 04

Testfall: 04	**Bezug: Überprüfen der Ballgeschwindigkeit**
Ablauf	Starten Sie das Spiel und beobachten Sie die Geschwindigkeit des Balls.
Erwartetes Ergebnis	Die Geschwindigkeit des Balls sollte sich erhöhen, wenn er auf die Paddles trifft.

Table 11 - Testfall 05

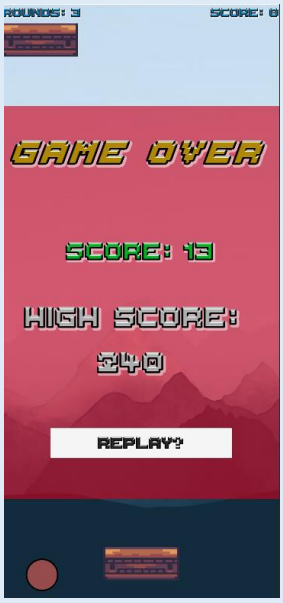
Testfall: 05	**Bezug: Überprüfen Sie das Zurücksetzen des Spielstands**
Ablauf	Starten Sie das Spiel, starten Sie das Spiel neu und beobachten Sie den Spielstand.
Erwartetes Ergebnis	Der Spielstand sollte zurückgesetzt werden, wenn das Spiel neu gestartet wird.  *Figure 1 - Test 05*

Table 12 - Testfall 06

Testfall: 06	**Bezug: Überprüfen der Spielsteuerung Beschreibung**
Ablauf	Starten Sie das Spiel, drücken Sie die Spielsteuerung und beobachten Sie die Ergebnisse
Erwartetes Ergebnis	Die Spielsteuerung sollte korrekt funktionieren.

Table 13 - Testfall 07

Testfall: 07	**Bezug: Prüfen die Anzahl der Runden verlieren**
Ablauf	Starten Sie das Spiel und spielen Sie, bis das Runden verloren wenn die ball geht unten oder oben
Erwartetes Ergebnis	Rund zahl sollte ein wenige

Table 14 - Testfall 08

Testfall: 08	**Bezug: Überprüfen der Rumpelfunktionalität**
Ablauf	Schlagen Sie den Ball mit dem Paddel
Erwartetes Ergebnis	das Telefon wird rumpeln

Table 15 - Testfall 09

Testfall: 9	**Bezug: Testen, ob das Startmenü funktioniert**
Ablauf	Öffnen Sie das Spiel und klicken Sie auf die Schaltfläche "Start".
Erwartetes Ergebnis	sollte die Szene wechseln und das Spiel beginnen

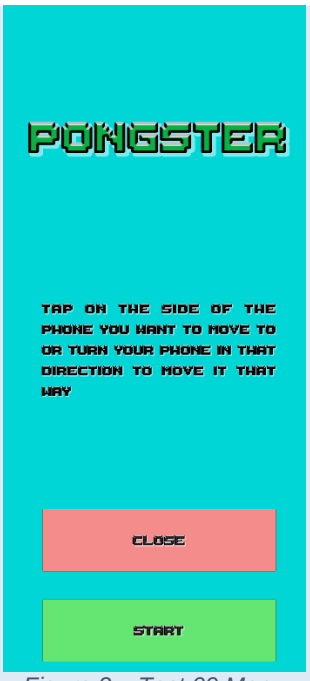


Figure 2 – Test 09 Menu

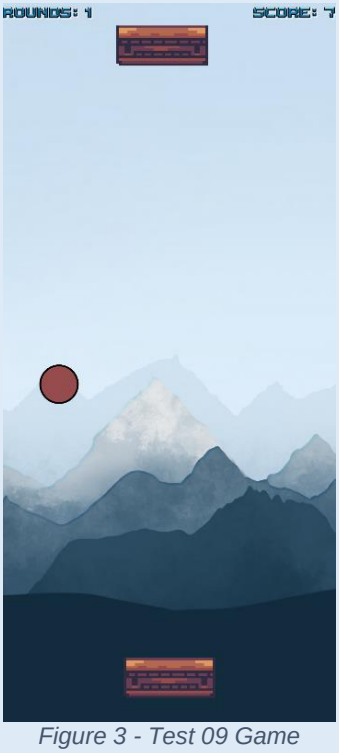


Figure 3 - Test 09 Game

Deployment-Ziele

Das Ziel dieses Projekts ist es, das Spiel auf Android-Geräten nur in vertikaler Ausrichtung zugänglich zu machen. Wir haben nicht vor, das Spiel in den Play Store zu stellen, aber wir hoffen, es in Zukunft verfügbar zu machen. Unser Hauptaugenmerk liegt darauf, sicherzustellen, dass das Spiel auf Android-Geräten reibungslos läuft und dass die Nutzer ein angenehmes Spielerlebnis haben. Wir suchen auch nach Möglichkeiten, das Spiel für eine bessere Benutzererfahrung zu optimieren, z. B. um sicherzustellen, dass die Steuerung einfach zu bedienen ist und die Grafik dem Auge schmeichelt. Wir planen auch, zusätzliche Funktionen wie Gyro- und Touch-Steuerung hinzuzufügen, die das Spiel noch angenehmer und einfacher machen. Mit diesen Zielen im Hinterkopf hoffen wir, ein unterhaltsames und vielseitiges Spiel zu entwickeln, das allen Android-Nutzern Spaß macht und eine Herausforderung darstellt.

Design

Das Pong-Spiel wird nur für Mobiltelefone entwickelt, damit ich mich nicht mit Tablets und deren unterschiedlichen Größen beschäftigen muss.

Die Mobile App wird nur vertikal sein, da dies die Möglichkeit von Problemen durch Rotation einschränkt. Außerdem würde Pong von oben nach unten auf einem Querformat-Bildschirm nicht so gut funktionieren, da der Abstand zwischen den Spielern im Vergleich zur Höhe eines vertikalen Bildschirms begrenzt ist.

Grundlegendes Design und wenn ich nicht wirklich grundlegendes Material verwende, werde ich es vom Unity Asset Market beziehen. Der Grund dafür ist, dass dort qualitativ hochwertige Assets kostenlos zur Verfügung stehen. Unity Market-Assets für die Wände und den Spieler, weil ich im Design nicht sehr geschickt bin.

die Assets für das Spiel werden aus dem Unity-Asset-Store bezogen. Ich werde das Sunny Land Character Pack und den Night Mountain Lake Background verwenden.

Die Versionskontrolle wird über Git abgewickelt und auf Github gespeichert, so dass ich auf alte Versionen zugreifen und im Falle größerer Fehler zurückkehren kann. (https://github.com/eliasrk/Project-335)

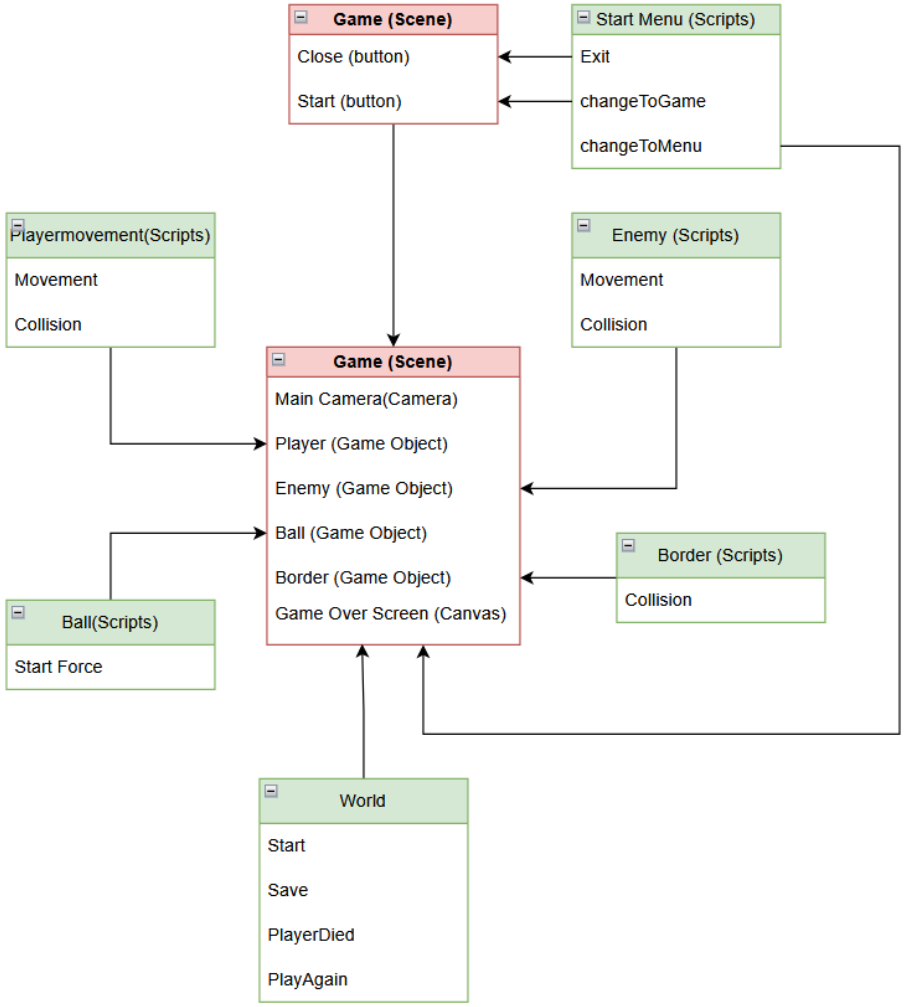


Figure 4 - Design UML

Spieler

```
if (Input.GetKey(KeyCode.A)){

transform.Translate(Vector3.left
* Time.deltaTime * 10);

        }
```

Figure 5 - Code nach links verschieben

The Der Benutzer hat die Möglichkeit, sich nach links und rechts zu bewegen. Dies schließt den Touchscreen und die Drehung des Bildschirms ein. Dies geschieht mit (Beispiel rechts), um den Spieler nach links gehen zu lassen, und wird dann geändert, um nach rechts zu gehen, indem "links" mit "rechts" und (KeyCode.A) zu (KeyCode.D) vertauscht wird. Dies wird alles in der Update()-Funktion getan, so dass jedes Bild (60 Mal pro Sekunde) geprüft wird, ob der Benutzer diese Taste drückt. jedes Mal, wenn der Spieler und der Ball zusammenstoßen, wird das Telefon vibrieren

Ball

```
if (collision.gameObject.tag ==
"ball")

        {

rigidbody.AddForce(new Vector2(0,
1000));
        }
```

Figure 6 - Kollision mit dem Ball

Der Ball wird initionally fallen aus der Mitte der Karte, um den Spieler in einem zufälligen Winkel. Die Kraft wird nicht durch den Ball, sondern durch die Kollision mit Objekten hinzugefügt, entweder durch rigidbody.addForce oder durch die Bounce-Komponente von Unity. Auf der rechten Seite ist ein Beispiel dafür,

was passieren würde, wenn der Spieler

den Ball trifft. Auch der Ball wird Einschränkungen haben, wie weit er in der Welt gehen kann. Eine davon ist, dass wenn er irgendwie aus dem Spielbereich herausfällt, er die Szene neu startet.

Gegner

Der Gegner hat nur die Funktion, dem Ball langsamer zu folgen, und aktualisiert nur alle 0,05 Sekunden, wo sich der Ball befindet, und bewegt sich dann nur mit einer Geschwindigkeit von 1,9. er hat auch die gleiche Funktion wie oben, indem er rigidbody.AddForce() verwendet.

Spiel Szene

Die Szene enthält alle zuvor erwähnten Assets und den Game-Over-Bildschirm mit den Runden, der Punktzahl, dem Highscore und zwei Schaltflächen für Wiederholung und Beenden. Hier wird auch der von Sollberger bereitgestellte Code verwendet, um den sicheren Bereich zu sperren.

Start Menu Szene

Das Startmenü enthält den Titel, Anweisungen zum Spielen und zwei Schaltflächen, eine für den Szenenwechsel zum Spiel und die andere zum Schließen der App.

Benutzerfreundlichkeit

Ich habe das klassische Menü- und Spieldesign verwendet, so dass der Benutzer mit meinem Spiel vertraut ist, wenn er schon einmal ein Spiel gespielt hat. Zum Beispiel im Menü "Spiel spielen", gefolgt von "Beenden". Innerhalb des Spiels wird der Spielstand in der Ecke mit einer Beschriftung und einem aktuellen Spielstand des Benutzers angezeigt. Wenn sie es nicht getan haben, wird es immer noch klar zu verstehen sein.

Storyboard

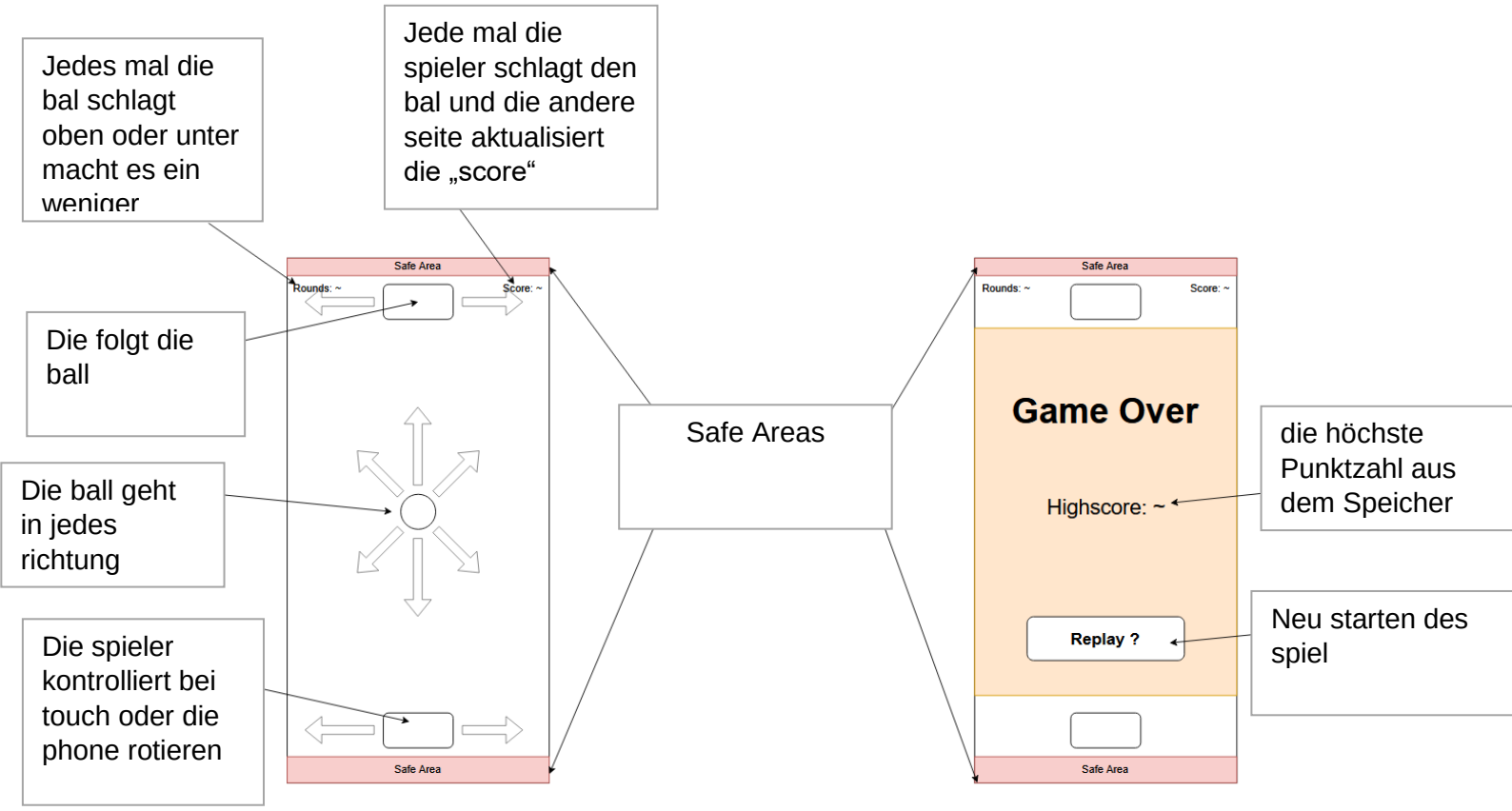


Figure 7 - Storyboard Spiel

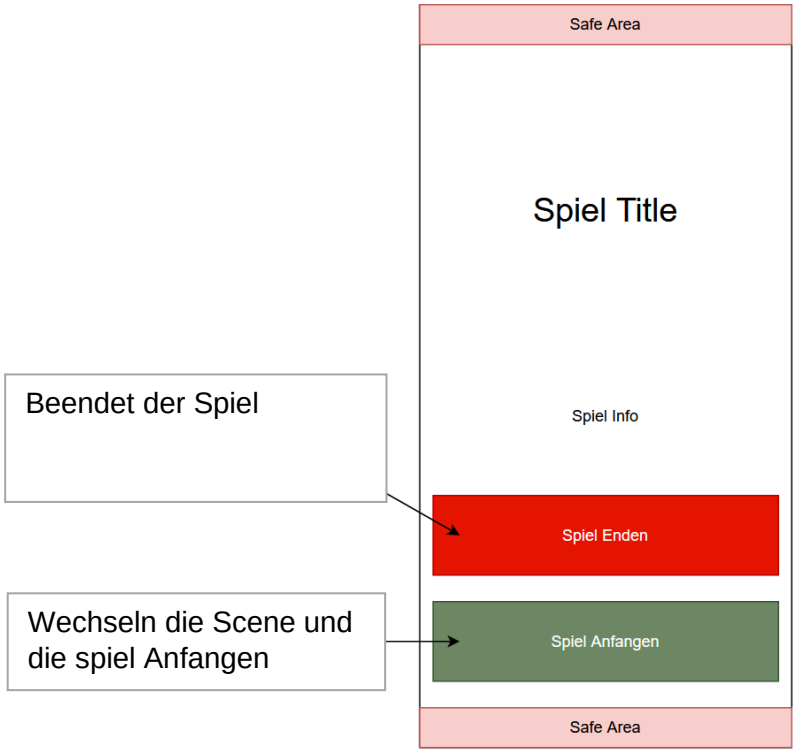


Figure 8 - Storyboard Menu

Abbildungsverzeichnis

Tabellenverzeichnis

Quelleverzeichnis

Ansimuz. “Sunny Land | 2D Characters | Unity Asset Store.” *Assetstore.unity.com*, 10 Nov. 2022, assetstore.unity.com/packages/2d/characters/sunny-land-103349. Accessed 8 Feb. 2023.

Manuel Sollberger, ”Canvas Helper” *CanvasHelper.cs, Project 335\Assets\Scripts\UI\CanvasHelper.cs, 10 Feb. 2023*

Manuel Sollberger, ”Safe Area”, SafeArea*.cs, Project 335\Assets\Scripts\UI\ SafeArea.cs,10 Feb. 2023*

Pathway. “Night Mountain Lake [SEAMLESS] | 2D Environments | Unity Asset Store.” *Assetstore.unity.com*, 3 2019, assetstore.unity.com/packages/2d/environments/night-mountain-lake-seamless-127703. Accessed 8 Feb. 2023.

Worlds, Tiny. “Free Pixel Font - Thaleah | 2D Fonts | Unity Asset Store.” *Assetstore.unity.com*, 5 Apr. 2019, assetstore.unity.com/packages/2d/fonts/free-pixel-font-thaleah-140059. Accessed 8 Feb. 2023.