



Belegarbeit “Last Hero”

Zhavarankau Illia

August 2022

| | |
|------------|---------------------------------|
| Erstprüfer | Prof. Dr. rer. nat. Marc Ritter |
| Co-Prüfer | M. Sc. Manuel Heinzig |
| Fach | Game Programming, Belegarbeit |
| Hochschule | Hochschule Mittweida |
| Semester | Sommersemester 2022 |

Inhaltsverzeichnis

| | |
|--|----|
| Inhaltsverzeichnis | 2 |
| 1. Spielkonzept | 3 |
| 2. Herausforderungen und deren Bewältigung | 3 |
| 3. Software-Architektur | 4 |
| 4. Beschreibung der wesentlichen Implementierungsdetails | 5 |
| 5. Spielanleitung | 7 |
| 6. Fazit | 8 |
| 7. Entwicklungstools | 8 |
| 8. Literaturverzeichnis | 8 |
| 9. Eigenständigkeitserklärung | 10 |

1. Spielkonzept

Battle-Royale Spiele sind in letzter Zeit sehr beliebt geworden. Inspiriert von diesem Genre beschloss ich, ein Minispiel wie Battle City (8 Bit) aus meiner Kindheit in diesem Genre zu schreiben. Dazu benötige ich ein Spielfeld (im Folgenden auch das Brett), das aus Zellen besteht, die man nicht verlassen kann. Das Brett wird in verschiedenen Farben (Texturen) gestrichen. Das Spielfeld muss sich in jedem Zeitintervall (im Folgenden "Runde") verkleinern. Das Brett hat 3 Zustände: grün (braune Erde), gelb (rissige Erde mit Lava) und rot (Lava). Der Spieler und seine Gegner (Objekte in Form von Panzern) müssen sich bewegen und schießen können. Wenn Panzer auf einer roten Zelle steht, wird Gesundheit weggenommen werden. Das Spiel endet, wenn der Spieler alle Lebenspunkte verloren hat oder der letzte Überlebende ist. Während des Spiels erhält ein Spieler Punkte, die in die Rangliste eingetragen werden. Wenn er die 10 trifft, wird sein Punktestand gespeichert.

Da die Zeit für die Entwicklung begrenzt ist, habe ich mich auf eine Einzelspielervariante beschränkt. Um das Interesse zu steigern, gibt es ein Menü mit Optionen, mit denen man Feinde aufrüsten kann, was zu Punktmultiplikatoren führt.

2. Herausforderungen und deren Bewältigung

Das Verhalten von Gegnern zu gestalten, hat sich als eine nicht ganz einfache Aufgabe erwiesen. Alle im Folgenden beschriebenen Funktionen sind im Skript `Enemy.cs` implementiert. Zunächst muss der Gegner das nächstgelegene Ziel finden und sich in dessen Richtung drehen. Dann schießt er 2 Mal pro Sekunde (standardmäßig) auf den anderen Panzer. Es wurden mehrere Funktionen geschaffen, um den Gegner in Bewegung zu halten. Befindet sich der Gegner auf einem gelben oder roten Feld, will er davon weglaufen und sucht sich ein freies grünes Feld, falls es noch welche gibt. Wenn er auf einem sicheren grünen Feld steht, bewegt er sich chaotisch. Wenn der Feind in der Nähe ist, versucht er, sich in die entgegengesetzte Richtung zu bewegen. Dieses Verhalten ist nicht das klügste, aber es ist eine angemessene Überlebensstrategie.

Die Kollision erwies sich als der schwierigste Punkt, was die Genauigkeit der Programmierung anbelangt. Da der Panzerkörper quadratisch ist, befindet sich der Projektil-Startplatz an der Seite und nicht an der Ecke. Wenn Sie sich kurzzeitig drehen, können Sie von Ihrem eigenen Projektil Schaden nehmen. Da jeder gegen jeden spielt, reicht es nicht aus, den Spieler und den Gegner zu erkennen. Es soll in der Lage sein, alle Gegenstände auf der Bühne zu erkennen. Deshalb haben alle gegnerischen Geschosse und der Spieler eindeutige Namen, so dass man erkennen kann, wer auf wen geschossen hat. Daher mussten wir allen Objekten Namen hinzufügen und bei der Erkennung mit der Substring -Funktion des Objektnamens arbeiten.

Ich wollte die Ergebnisszene ähnlich wie die Herzen und Smileys auf Streaming-Plattformen dekorieren. Ein Partikelsystem, das Emoticons verschießt, war dafür geeignet. Es wurde in der Ergebnisszene implementiert.

Um Optionen und die Bewertungstabelle zu speichern und zu laden, ist es notwendig, mit Dateien zu arbeiten. Zu diesem Zweck mussten wir die Objektklassen Config.cs und ResultData.cs schreiben, die im Skript SaveResult.cs verwendet werden. Um es dem Benutzer zu erschweren, Einstellungen oder gespeicherte Ergebnisse zu manipulieren, werden sie in einer Binärdatei gespeichert, weshalb Config.cs und ResultData.cs serializable sind.

In den Optionen wird ein Float-Parameter verwendet, um die Geschwindigkeit des Gegners anzupassen, die einen Punkt hat, aber beim Parsen der Zeichenkette ein Komma erhält. Daher muss ich Replace (",", ".") verwenden, damit es richtig funktioniert.

3. Software-Architektur

Das Spiel besteht aus 6 Szenen: Menü, Spiel, Optionen, Anweisungen, Bewertung, Ergebnisse. Der Übergang zwischen ihnen ist in der Abbildung dargestellt. Jede Szene hat ein oder mehrere Skripts.

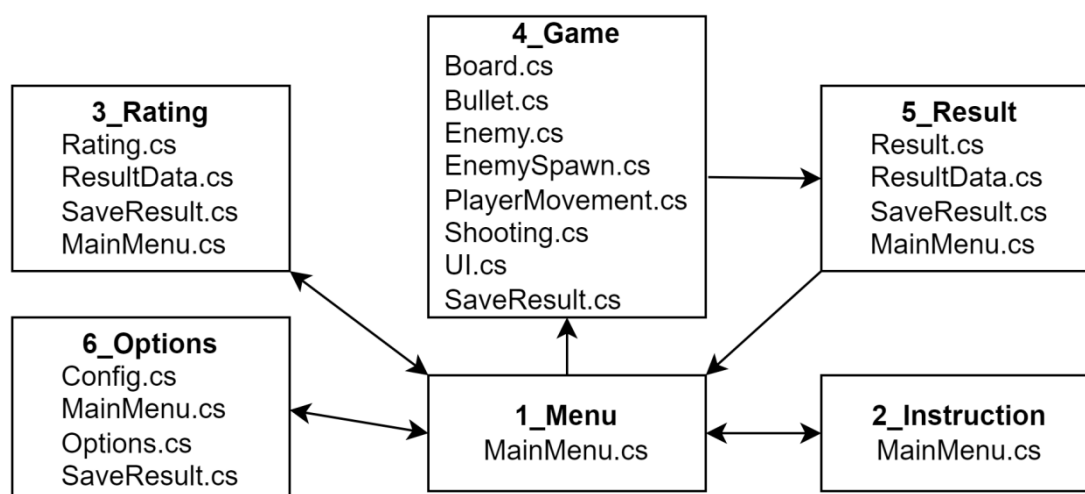


Abbildung 1. Übergänge zwischen Szenen mit Skripten im Einsatz

Es gibt 7 Hauptspielobjekte, aus denen das Spiel aufgebaut ist. Rand der Zellentafel, drei Arten von Zellen (von sicher bis gefährlich). Der Panzer des Spielers ist grün, der Panzer des Gegners lila und das Geschoss sieht bei Spieler und Gegner gleich aus.

RandomCellof wird verwendet, um diese Funktion aufzurufen. In Zeile 3 prüft die Bedingung die Runde. Mit dieser Bedingung können wir festlegen, dass in den ersten Runden die bemalten Zellen nur vom Rand des Spielbretts aus zufällig verteilt werden. Zu diesem Zweck wird in den Zeilen 5 und 6 ein Zufallswert verwendet. In der Schleife suchen wir nach einem freien Feld vom Rand des Spielbretts, bis wir eines finden. Als nächstes wird in den Zeilen 24-26 eine rekursive Funktion aufgerufen, um die schattierten Zellen weiterzugeben. In Zeile 28 Bedingungen, wenn wir nicht bereits die ersten Runden, dann können wir eine zufällige Zelle nehmen und von ihm wird Lava Ausbreitung gehen.

```

18         }
19     }
20     randomX = Random.Range(0, widthBoard);
21     randomY = Random.Range(0, widthBoard);
22 }
24 replaceCell(randomX,randomY,cell_yellow);
25 selectNeighboringCells(widthBoard/2,randomX,randomY);
26 round++;
27 }
28 else{
29     randomX = Random.Range(0, widthBoard);
30     randomY = Random.Range(0, widthBoard);
31     selectNeighboringCells(widthBoard/2,randomX,randomY);
32 }
33 }

```

5. Spielanleitung

Das Spiel ist für Windows kompiliert, also führen Sie einfach LastHero.exe aus.

Spielregeln: Der Spieler und die gegnerischen Panzer spielen jeder für sich. Der Panzer des Spielers ist grün, die gegnerischen Panzer sind blau. Das Ziel ist es, so lange wie möglich durchzuhalten. In jeder Runde (Basiswert 15 Sek.) wird die Größe des Spielbretts um die halbierte Länge reduziert, abgerundet auf eine ganze Zahl. Es gibt eine Begrenzung des Spielfelds, die nicht überschritten werden darf - beim Überschreiten werden keine Lebenspunkte abgezogen. Es gibt 3 Arten von Käfigen. Grüne (braune Erde) ist sicher, gelbe (rissige Erde mit Lava) ist auch sicher, aber wenn die Zeitspanne abgelaufen ist, verwandelt sie sich in rote (Lavaquadrate). Auf roten Feldern erleidet jeder Tank jede halbe Sekunde Schaden in Höhe von 2. Wenn ein Projektil auf eine Granate trifft, wird niemandem ein Schaden zugefügt.

Das Ziel des Spiels ist, so viele Punkte wie möglich zu erzielen. Punkte gibt es für das Überleben in jeder Runde +10 Punkte, das Verursachen von Schaden an Gegnern +15 Punkte pro Schuss und +3 Punkte für einen Rammbock, wenn man einen der Gegner zerstört, egal ob der Spieler oder ein anderer Gegner +50 Punkte. Um den Punktemultiplikator zu erhöhen, können Sie den Schwierigkeitsgrad der Gegner in den Einstellungen erhöhen. Je größer das Spielfeld ist, desto mehr Runden und Gegner gibt es, was sich indirekt auf die maximal mögliche Punktzahl auswirkt.

Steuerung: Um den Panzer zu bewegen, benutze: nach oben – W oder Pfeil nach oben, nach unten – S oder Pfeil nach unten, nach links – A oder Pfeil nach links, nach rechts – D oder Pfeil nach rechts. Die Maus wird für die Schussrichtung verwendet, in die die Geschosse fliegen. Wenn Sie die linke Maustaste drücken, wird der Panzer abgefeuert.

Wenn Sie die ESC-Taste drücken, wird das Spiel angehalten (Pause), drücken Sie erneut, um fortzufahren.

6. Fazit

Es wurde ein Battle-Royale-Spiel entwickelt, das unter Windows läuft. Die größten Herausforderungen waren das Verhalten des Gegners, die Ausarbeitung der Kollisionsarten zwischen den Objekten und das dynamische Spielfeld. Alles wurde auf der Basis der Unity-Engine erstellt. Das Spiel besteht aus 6 Szenen und 7 Grundobjekten. Ziel des Spiels ist es, so viele Punkte wie möglich zu erzielen, indem man einen Panzer mit der klassischen Steuerung steuert, auf Feinde schießt und länger durchhält als andere.

Das Spiel ist erweiterungsfähig, zum Beispiel durch neue Mechanismen (Waffen, Gesundheit, Gegenstände) und Multiplayer.

7. Entwicklungstools

Als Begleitmaterial wurden die Vorlesungen "Game Programming" und dazugehörige Zusatzmaterialien verwendet. Für die Entwicklung wurden Unity 2020.3.19f1 (64-bit) und Visual Studio Code verwendet. Assets [1,2] wurden für die Visualisierung genutzt. Informationen über Objektbehandlung, Kollisionen und mehr in der Unity-Dokumentation [3,4]. Für eine visuelle Präsentation wurden YouTube-Videos als zusätzliche Ressourcen verwendet [5-15].

8. Literaturverzeichnis

1. Unity Asset Store, Tank Constructor. [Online]. Verfügbar unter: <https://assetstore.unity.com/packages/2d/textures-materials/tank-constructor-131316>. Zugriff am: 25. Juli 2021.
2. Unity Asset Store, Lowpoly Textures Pack. [Online]. Verfügbar unter: <https://assetstore.unity.com/packages/2d/textures-materials/lowpoly-textures-pack-140717#publisher>. Zugriff am: 25. Juli 2021.
3. Unity Documentation, Importing from the Asset Store. [Online]. Verfügbar unter: <https://docs.unity.cn/ru/2017.4/Manual/AssetStore.html>. Zugriff am: 25. Juli 2021.
4. Unity Documentation, Unity User Manual 2021.3 (LTS). [Online]. Verfügbar unter: <https://docs.unity3d.com/Manual/>. Zugriff am: 27. Juli 2022.
5. Youtube, START MENU in Unity. [Online]. Verfügbar unter: https://www.youtube.com/watch?v=zc8ac_qUXQY. Zugriff am: 27. Juli 2022.

6. Youtube, Программирование для детей | Нажатие кнопки | Вывод в текст Unity | Unity для детей [Programmieren für Kinder | Tasten drücken | Unity Textausgabe | Unity für Kinder]. [Online]. Verfügbar unter: <https://www.youtube.com/watch?v=lyrb8tUcWeo>. Zugriff am: 28. Juli 2022.
7. Game Dev Beginner, How to use random values in Unity (with examples). [Online]. Verfügbar unter: <https://gamedevbeginner.com/how-to-use-random-values-in-unity-with-examples/>. Zugriff am: 29. Juli 2022.
8. Youtube, TOP DOWN SHOOTING in Unity. [Online]. Verfügbar unter: <https://www.youtube.com/watch?v=LNLVOjbrQj4>. Zugriff am: 29. Juli 2022.
9. Youtube, 2D CAMERA FOLLOW PLAYER IN UNITY | Cinemachine 2D Camera Movement In Unity | Unity Tutorial. [Online]. Verfügbar unter: <https://www.youtube.com/watch?v=B6zwEv28pLQ>. Zugriff am: 31. Juli 2022.
10. Youtube, How to make a Video Game in Unity - COLLISION (E05) [Online]. Verfügbar unter: https://www.youtube.com/watch?v=gAB64vfbrhl&ab_channel=Brackeys. Zugriff am: 1. August 2022.
11. Youtube, Six Ways You Can Move a 2D Object in Unity [Online]. Verfügbar unter: https://www.youtube.com/watch?v=pwRywYmajsY&ab_channel=PixelbugStudio. Zugriff am: 3. August 2022.
12. Youtube, PAUSE MENU in Unity [Online]. Verfügbar unter: <https://www.youtube.com/watch?v=JivuXdrIHK0>. Zugriff am: 8. August 2022.
13. Youtube, Unity Particle System 2D - Eigene Particle erstellen und System verstehen - Unity Deutsch/German [Online]. Verfügbar unter: https://www.youtube.com/watch?v=W3GxW9A4PLs&ab_channel=nobreakpoints. Zugriff am: 8. August 2022.
14. Youtube, SAVE & LOAD SYSTEM in Unity [Online]. Verfügbar unter: https://www.youtube.com/watch?v=XOjd_qU2Ido&ab_channel=Brackeys. Zugriff am: 9. August 2022.
15. Youtube, High Score Table with Saving and Loading (Unity Tutorial for Beginners) [Online]. Verfügbar unter: https://www.youtube.com/watch?v=iAbaqGYdnyl&ab_channel=CodeMonkey. Zugriff am: 9. August 2022.

9 Eigenständigkeitserklärung

Ich erkläre, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.



Waldheim, den 15. August 2022

Illia Zhavarankau