

Hochschule Harz

Fachbereich Automatisierung und Informatik

Einführung virtuelle Welten

Prof. Dr.-Ing. Simon Adler

Pool Billard

Entwurfsarbeit

vorgelegt von:

Fabian Siemens
u37052@hs-harz.de

Studiengang:

Informatik

Jahrgang:

2021

Fachsemester:

6

Matrikelnummer:

m29620

Prüfer:

Prof. Dr.-Ing. Simon Adler

Ort, Datum:

Wernigerode, 15.08.2024

I. Inhaltsverzeichnis

I.	Inhaltsverzeichnis	I
II.	Abbildungs- /Tabellenverzeichnis	I
1	Einleitung	2
2	Angepasste Spielregeln	2
3	Projektarchitektur	3
3.1	Szene.....	3
3.2	Code.....	4
4	Funktionsumfang.....	5
5	Nutzungshinweise	6
6	Ausblick.....	6
III.	Literaturverzeichnis	7
IV.	Anhang.....	8

II. Abbildungs- /Tabellenverzeichnis

Abbildung 1:	Unity Szenegraph	3
Abbildung 2:	Vereinfachtes Klassendiagramm; Erstellt mit Visual Paradigm.	8
Abbildung 3:	Pool Tisch in Unity.....	8
Abbildung 4:	Szene in Unity.....	9
Abbildung 5:	Alle GUI-Elemente in Unity überlagert.	9
Abbildung 6:	Billardtisch Holz-Textur Graph; Erstellt in Blender.	10
Tabelle 1:	Zusammenfassung Funktionsumfang.....	5
Tabelle 2:	Zusammenfassung Spielsteuerung	6

1 Einleitung

Ziel dieses Projekts ist die Erstellung eines Pool Billard Spiels. Es soll eine Runde 8-Ball Pool Billard vollständig spielbar sein. Hierfür werden die wichtigsten Spielregeln von 8-Ball implementiert, der Spielstand erfasst und angezeigt. Das Spiel wird mit zwei Personen am gleichen Rechner spielbar sein, wenn diese die Mauskontrolle untereinander abwechseln. Die Kamera wird statisch von oben auf das Spielfeld hinabblicken, während das Spiel über Maus und Tastatur gesteuert wird. Dabei erfolgt das Stoßen der Kugeln durch intuitives Zielen und Ziehen der Maus. Das Projekt wird in Einzelarbeit umgesetzt, fremde Inhalte werden als solche benannt.

2 Angepasste Spielregeln

Das 8-Ball Pool Billard ist eine Billardvariante mit einer weißen und 15 bunten Kugeln. Für dieses Projekt werden einige Regeln weggelassen und nur die wichtigsten Regeln implementiert. Die weiße Kugel, wird im Folgenden Spielball oder Queueball genannt. Die 15 farbigen Bälle werden als Objektbälle bezeichnet. Hierbei gibt es 7 „halbe“ Kugeln, diese sind nur zur Hälfte eingefärbt, und 8 „volle“ Kugeln. Das Spiel ist für zwei Spieler ausgelegt. Ziel ist es, alle Objektbälle der einem zugeteilten Farbe (halb, oder voll) zu versenken, ohne dabei die schwarze Kugel (8) zu versenken. Nachdem alle Kugeln seiner Farbe versenkt sind, muss ein Spieler die 8 versenken, um das Spiel zu gewinnen. Wird die 8 versenkt, ohne dass vorher alle 7 anderen Kugeln versenkt wurden, gewinnt der Gegner das Spiel. Wird der Queueball versenkt, gilt dies als Foul und die Kugel wird auf den Startpunkt zurückgesetzt.

3 Projektarchitektur

3.1 Szene

Der Szenegraph (siehe Abbildung 1) enthält neben einigen beiläufigen Komponenten, vor allem die folgenden Objekte: Die Kamera, die Beleuchtung, der Queue-Controller, der Billardtisch, und die graphische Nutzeroberfläche (GUI). Der Billardtisch wurde eigens in Blender 4.0 erstellt. Hierfür wurden die offiziellen Maße der Deutschen Billardunion (vgl. [1]) verwendet. Die Textur des Billardtischs wurde in Blender prozedural generiert. Der entsprechende Graph kann im Anhang (Abbildung 6) gefunden werden. Um die Texturen in Unity zu übernehmen, mussten diese vorher in Blender gebacken (baking) werden. Die Kamera (Top-Down-Camera) schaut von oben auf den

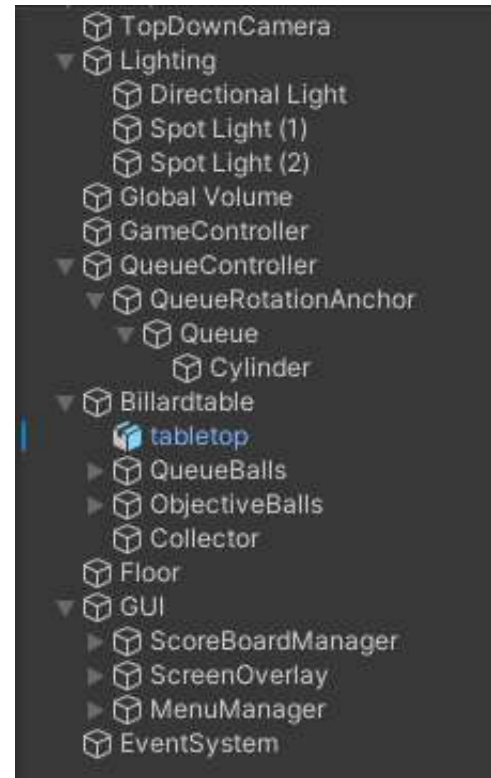


Abbildung 1: Unity Szenegraph

Billardtisch herab und ist statisch. Um die Szenerie zu beleuchten und reale Billard-Beleuchtung nachzuahmen, werden ein direktionales Licht, sowie zwei Scheinwerfer verwendet. Um Billard spielen zu können, werden zudem ein Queue und Bälle gebraucht. Der Queue ist im Queue-Rotation-Ancor enthalten, welcher notwendig ist, damit der Queue sich um den Spielball rotieren kann. Die Bälle sind aufgeteilt in zwei Eltern-Objekte, den Queue-Balls und Objective-Balls. Ersteres enthält i.d.R. nur einen Ball, den Spielball. Letzteres hingegen enthält die 15 Objektbälle. Damit die Bälle unterscheidbar sind, haben sie die offiziellen Farben und Nummerierungen. Die dafür notwendigen Texturen wurden online heruntergeladen (vgl. [2]). Zudem umfasst der Billardtisch einen Kollektor, welcher die Bälle, die versenkt oder über die Spielfläche hinausgeschossen werden, erkennt und zerstört. Das GUI enthält drei Komponenten, zur Darstellung von Informationen und zur Nutzerinteraktion. Die erste Komponente ist der Scoreboard-Manager, welcher das Scoreboard enthält. Die zweite Komponente ist das Screen-Overlay, hier werden u.a. der aktuelle Spieler, sowie Benachrichtigungen angezeigt. Die letzte Komponente ist der Menu-Manager, welcher das Hauptmenü und die darin verwendeten Knöpfe enthält. Für die Darstellung werden Canvase,

und Text-Mesh-Pro-Komponenten, wie Buttons und Text verwendet. Abbildungen der Szene bzw. aus dem Spiel sind im Anhang zu finden.

3.2 Code

Im Folgenden wird kurz auf die Struktur des Codes, sowie die grundlegenden Funktionen der wichtigsten Klassen eingegangen. Die Abbildung 2 im Anhang zeigt das Klassendiagramm dieser Anwendung. Im Zentrum der Anwendung stehen die Klassen GameController, QueueController und BallManager. Der GameController umfasst die wichtigste Spiellogik, wie z.B. Runden, Rundenwechsel, Spielstandserfassung und Siegerfassung. Der QueueController behandelt die Steuerung des Queues. Zudem erlaubt er das temporäre Unterdrücken des Queues, damit bspw., während das Menü geöffnet ist, oder die Bälle rollen, kein weiterer Stoß durchgeführt werden kann. Der BallManager kümmert sich um die Verwaltung der Bälle. Er führt die erstmalige Platzierung der Bälle durch und stellt öffentliche Methoden zur Neuplatzierung der Bälle bereit. Neben diesen drei zentralen Klassen gibt es für den Spielbetrieb noch eine weitere wichtige Klasse. Der BallCollector reagiert auf Interaktionen von anderen Collidern mit seinem Collider, prüft, ob der fremde Collider ein Ball ist und ruft, sollte dies der Fall sein, die OnBallCollected-Methode des GameControllers auf. Damit die Bälle im Code unterschieden werden können, existiert die Klasse BallMarker. Diese Klasse besitzt drei Boolean-Flaggen, welche im Unity Editor gesetzt werden können. Die Flagge „isFull“ beschreibt, ob die Farbe des Balls voll oder halb ist. „isQueue“ ist wahr, wenn der Ball der Spielball (Queueball) ist und „isEight“ ist true, wenn der Ball die 8 ist. Zur Steuerung der GUI-Komponenten, werden die Klassen MenuManager, ScoreBoardManager, TurnDisplay und OverlayMessageManager genutzt. Der MenuManager reagiert auf das Drücken der Escape-Taste, um die Sichtbarkeit des Hauptmenüs umzuschalten. Der ScoreBoardManager funktioniert analog zum MenuManager, jedoch erwartet dieser, dass die Tab-Taste gehalten wird, um das Scoreboard anzuzeigen. Das TurnDisplay prüft in regelmäßigen Abständen, welcher Spieler an der Reihe ist, und zeigt dies oben links auf dem Bildschirm mit einem kurzen Schriftzug an. Der OverlayMessageManager stellt eine Funktion zur Verfügung, um eine Nachricht für eine bestimmte Zeit in der Mitte des Bildschirms anzuzeigen.

4 Funktionsumfang

Nachdem die Anwendung gestartet wurde, kann bereits mit der Runde 8-Ball Pool Billard begonnen werden. Das Spiel ist auf die Teilnahme von 2 Personen ausgelegt, welche sich Maus und Tastatur teilen. Spieler 1 darf zuerst den Queueball spielen, wie oben links angezeigt wird. Locht Spieler 1 mit seinem Stoß eine farbliche Kugel ein, wird ihm automatisch die Farbe (halb oder voll) der Kugel und Spieler 2 die übrige Farbe zugeteilt. Diese Zuteilung wird mittels einer kurzen Nachricht auf dem Bildschirm, sowie einem Vermerk im Scoreboard kommuniziert. Jeder Spieler kann, nachdem er eine Kugel versenkt hat, ein zweites Mal den Queueball spielen. Versenkt der Spieler die weiße Kugel, oder schafft es nicht eine andere Kugel zu versenken, wird an den nächsten Spieler, in diesem Fall Spieler 2 abgegeben. Der Rundenwechsel wird sowohl oben links mit dem Zugindikator als auch über eine kurze Nachricht auf dem Bildschirm angezeigt. Sollte durch das Versenken des Queueballs ein Foul begangen worden sein, wird dieser automatisch neu platziert. Werden farbliche Kugeln versenkt, wird dem Spieler, dem diese Farbe zugeteilt ist, ein Punkt zugeschrieben. Der aktuelle Punktestand kann jederzeit im Scoreboard (Tab) nachgeschaut werden. Sollte die 8 versenkt worden sein, ist die Partie beendet. Hat der Spieler vorher alle 7 anderen Kugeln seiner Farbe versenkt, gewinnt er, sonst der Gegner. Der Sieger wird in einer Nachricht auf dem Bildschirm mitgeteilt. Daraufhin öffnet sich das Hauptmenü, welches jederzeit über Escape geöffnet werden kann, um einen Neustart oder Verlassen des Spiels zu ermöglichen.

Folgende Liste fasst den Funktionsumfang des Spiels zusammen:

Tabelle 1: Zusammenfassung Funktionsumfang

Stoßen des Queueballs durch Ziehen und Zielen
Lokaler Mehrspieler
Automatischer Rundenwechsel
Automatische Zuweisung der Objektkugelfarben
Automatische Neuplatzierung des Queueballs bei einem Foul
Punkteerfassung und Darstellung im Scoreboard
Siegerfassung
Hauptmenü
Neustart

5 Nutzungshinweise

Die Handhabung des Spiels ist weitestgehend intuitiv, wird im Folgenden aber kurz erläutert. Der Queue zeigt immer in Richtung Mauszeiger und kann durch Klicken der linken Maustaste aufgeladen werden. Hierfür muss bei gehaltenen Linksklick die Maus gezogen werden, um den Queue, ähnlich wie einen Bogen aufzuladen bzw. zu spannen. Sobald die linke Maustaste gelöst wird, wird der Queueball in die Richtung der Maus gestoßen. Die Distanz zwischen Startpunkt und Endpunkt des Linksklicks, bestimmt die Stärke des Stoßes. Der aktuelle Spielstand, sowie die Zuweisung der Kugelfarben kann im Scoreboard nachgeschaut werden. Zum Öffnen des Scoreboards muss die Tabulator-Taste gedrückt gehalten werden. Wenn das Spiel neu gestartet oder beendet werden soll, muss der Spieler mit Escape das Hauptmenü öffnen und kann dann mit der Maus die gewünschte Aktion auswählen. Folgende Tabelle fasst die Steuerungsmöglichkeiten zusammen.

Tabelle 2: Zusammenfassung Spielsteuerung

Taste	Aktion
Linke Maustaste (Halten und Ziehen)	Aufladen des Stoßes
Linke Maustaste (Lösen)	Durchführung des Stoßes
Tabulator (Halten)	Anzeigen des Scoreboards
Escape	Umschalten des Hauptmenüs

6 Ausblick

Das in diesem Projekt umgesetzte Pool Billard Spiel kann in vielerlei Hinsicht erweitert und verbessert werden. Zunächst wäre ein Ausbau der Spiellogik notwendig, um alle offiziellen Regeln des 8-Ball Billards zu verwirklichen. Zudem könnte die Logik erweitert werden, um auch andere Billard-Varianten spielen zu können. Ein online Multiplayer würde sich ebenfalls anbieten und erfordert über ein Peer-To-Peer System nicht einmal eine Server-Infrastruktur. Um das Spiel immersiver zu machen, sollten Soundeffekte eingefügt und die Grafik verbessert werden. Das Wichtigste ist jedoch die Implementierung einer eigenen Physik-Engine, um das physikalische Verhalten und die Interaktion zwischen den Kugeln realistischer zu gestalten.

III. Literaturverzeichnis

- [1] Deutsche Billard-Union e.V., Materialnormen Pool, <http://files.billard-union.de/handbuch/16-Materialnormen-Pool-12-2007.pdf>: Deutsche Billard-Union e.V..
- [2] FacadeGaiken, "Billard Balls | OpenGameArt.org," OpenGameArt.org, 25 Januar 2014. [Online]. Available: <https://opengameart.org/content/billiard-balls>. [Accessed 14 August 2024].
- [3] zaikman, "InspectorButton - Add a custom button to your script's Inspector with one line of code," Reddit, 5 Dezember 2013. [Online]. Available: https://www.reddit.com/r/Unity3D/comments/1s6czv/inspectorbutton_add_a_custom_button_to_your/. [Accessed 15 August 2024].

IV. Anhang

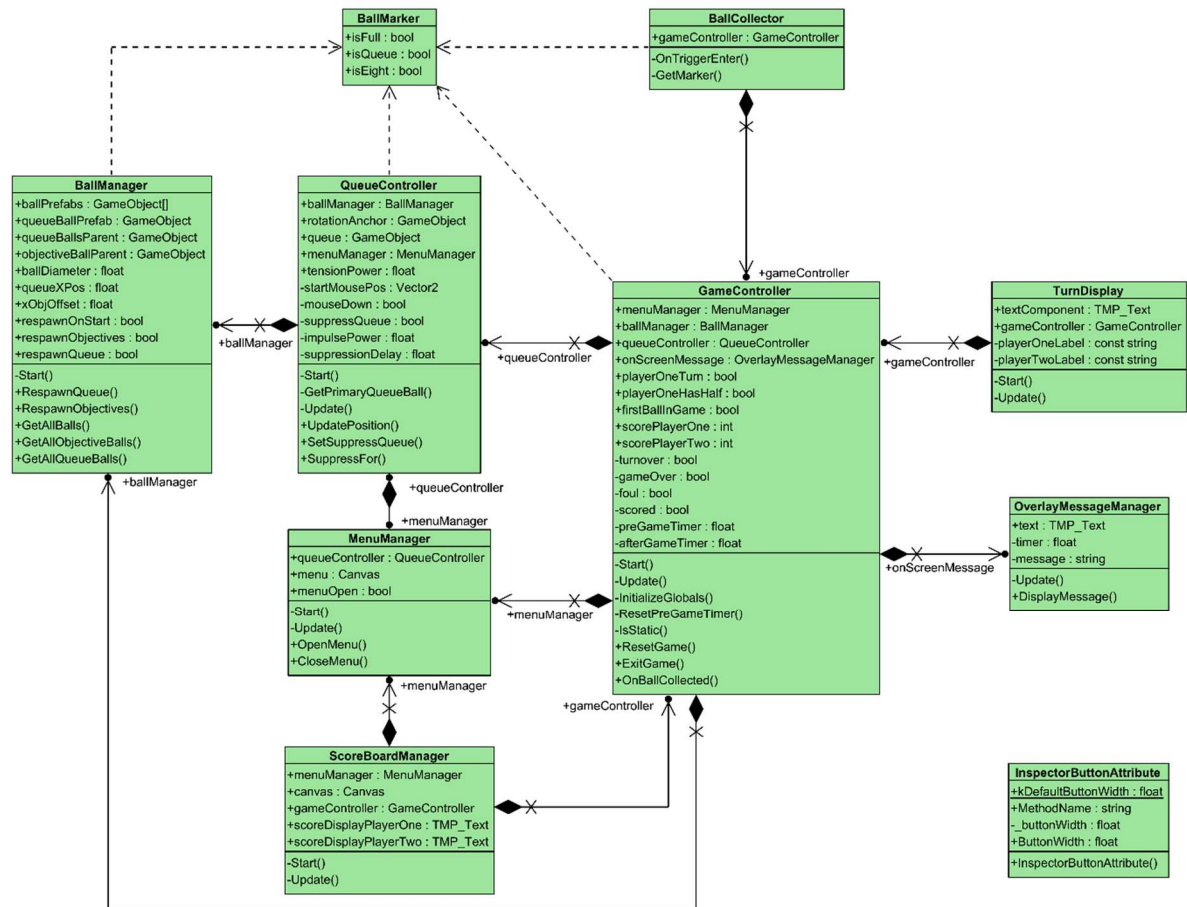


Abbildung 2: Vereinfachtes Klassendiagramm; Erstellt mit Visual Paradigm.

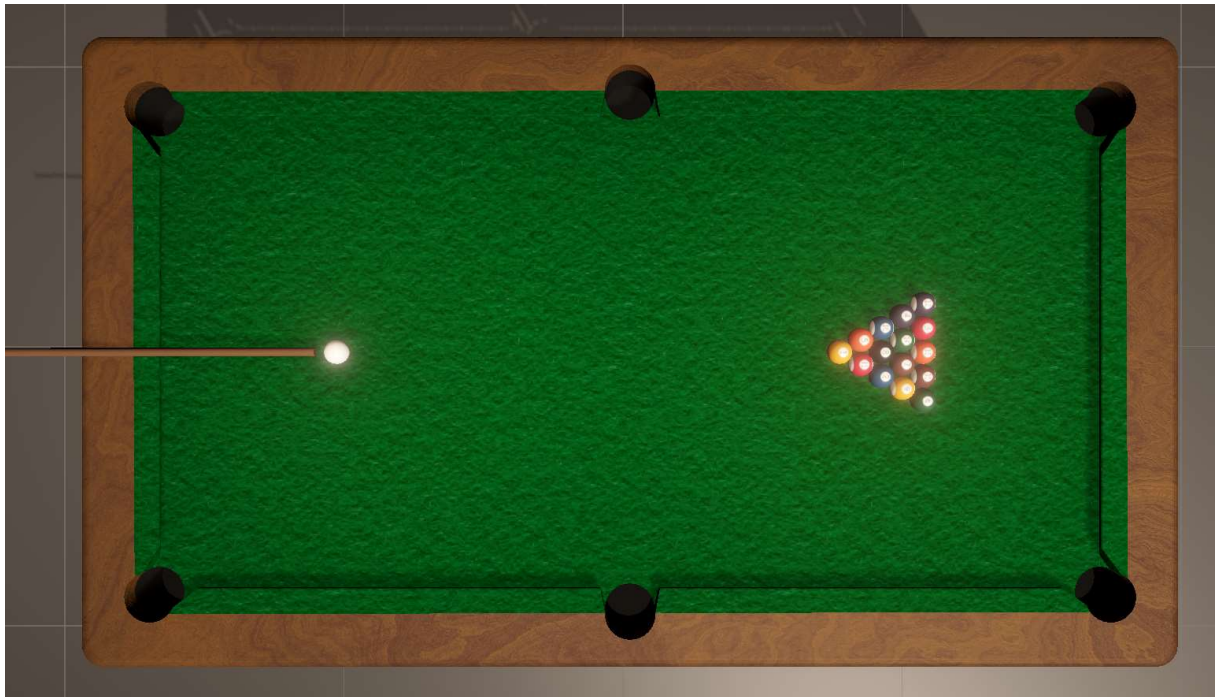


Abbildung 3: Pool Tisch in Unity.

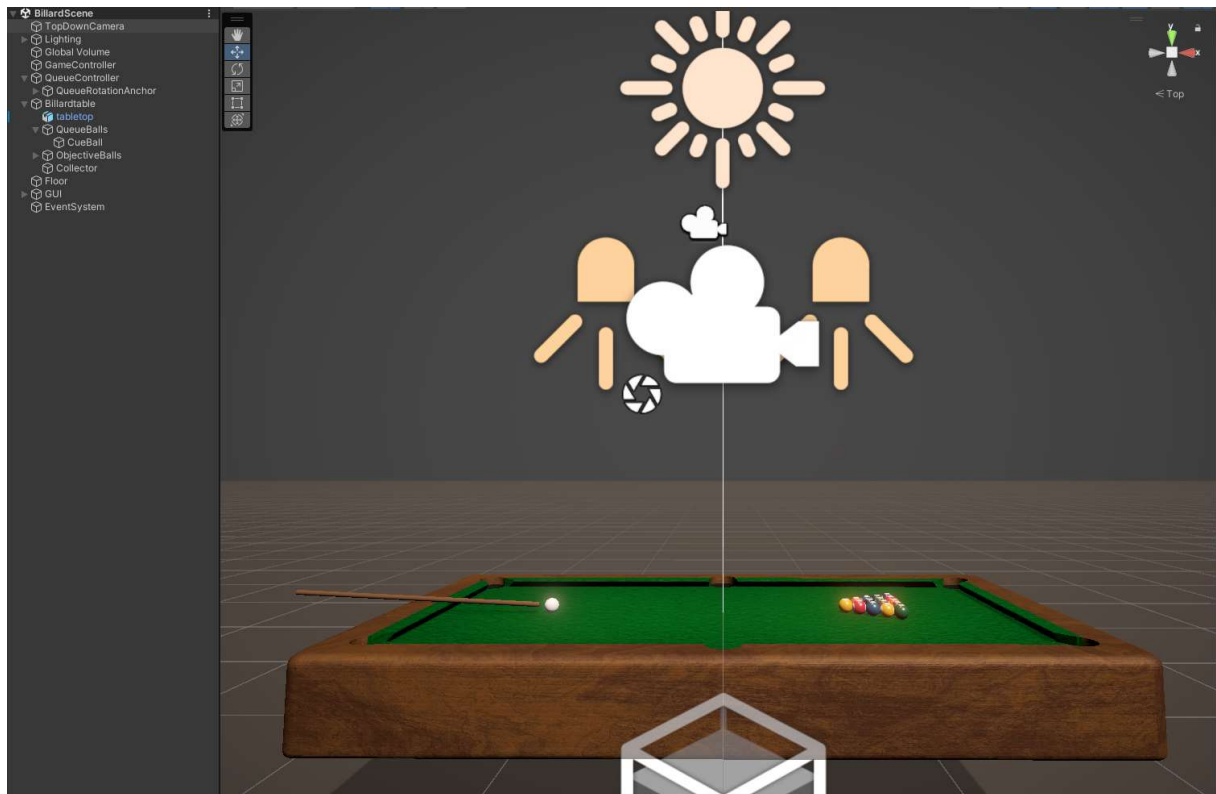


Abbildung 4: Szene in Unity.

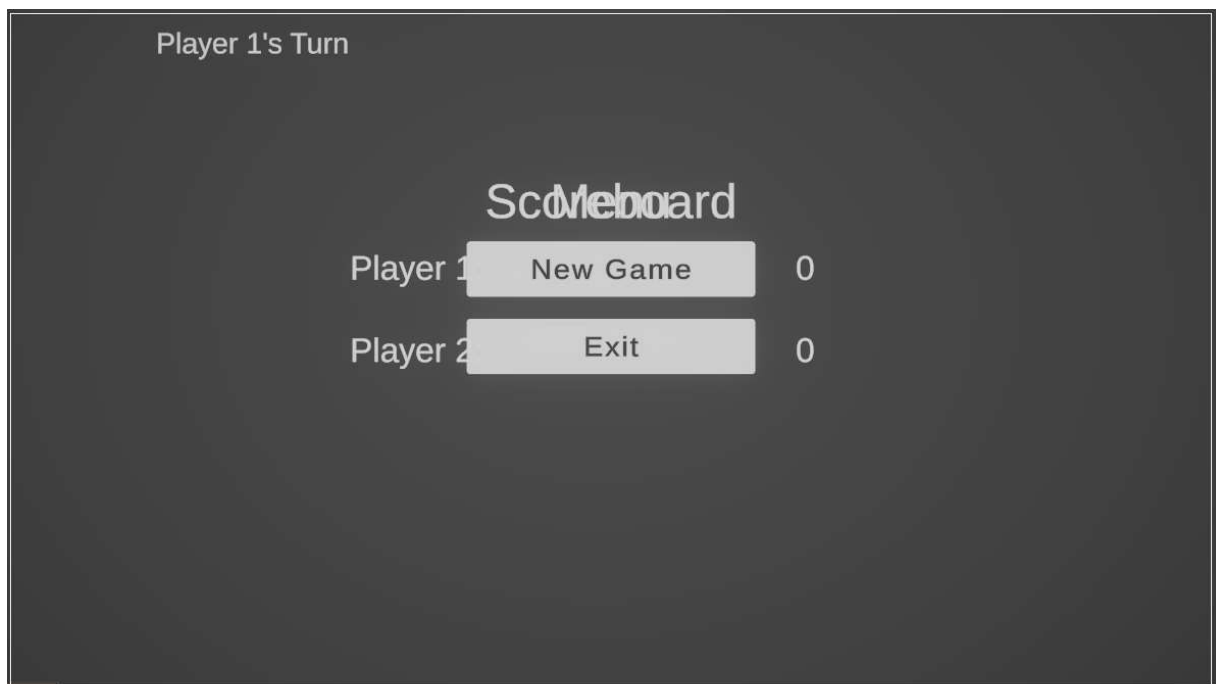


Abbildung 5: Alle GUI-Elemente in Unity überlagert.

Abbildung 6: Billardtisch Holz-Textur Graph; Erstellt in Blender.

