

# MAZERUN

Gesamtprojekt

## THEMA

PLATZHALTER für (Bild, Grafik)

Aufbau, Schaltplan...  
oder Logo

**Teammitglieder:**

BURCAK

SÖNMEZ

RATH

HERCEG

Rankweil, am 08.04.2022

**Betreuer:**

Dipl.-Ing. Rusch Helmut Elmar

Abgabevermerk:

DA original, am 08.04.2022

.....

Dipl.-Ing. Rusch Helmut Elmar

DA digital, am 08.04.2022

.....

Dipl.-Ing. Rusch Helmut Elmar



## Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Rankweil, am 08.04.2022

.....  
Semih Can

.....  
Semih

.....  
Manuel

.....  
Mateo

## Abkürzungen & Hinweise

DA	Diplomarbeit
WPF	Windows Presentation Foundation
CAD	Computer-aided design
Regex	Regular expression
GPL	General Public License
SI	Internationales Einheitensystem
COM	Component Object Model
GUI	Graphical User Interface

# Inhaltsverzeichnis

<b>DIPLOMARBEIT DOKUMENTATION</b>	<b>5</b>
<b>DIPLOMA THESIS DOCUMENTATION</b>	<b>7</b>
<b>ZUSAMMENFASSUNG</b>	<b>9</b>
<b>Abstract (English)</b>	<b>10</b>
<b>1 Lastenheft / Pflichtenheft</b>	<b>11</b>
1.1 Kurzbeschreibung	11
1.2 Aufgaben einzelner Projektmitglieder	11
1.3 Projektziele nach SMART	11
1.3.1 Spezifisch	11
1.3.2 Messbar	11
1.3.3 Realistisch	12
1.3.4 Terminiert	12
1.4 Produkteinsatz	12
1.5 Funktionale Anforderungen	12
1.6 Projektphasen und Meilensteine	13
1.6.1 Spiel	13
1.6.2 Kontroller	13
1.7 Arbeitspakete (aufgelistet)	13
1.7.1 Semih Can Burcak	13
1.7.2 Semih Sönmez	14
1.7.3 Manuel Rath	14
1.7.4 Mateo Herceg	15
<b>2 Kapitel 2</b>	<b>16</b>
2.1 BlaBla 1	16
2.1.1 BlaBla 2	16
2.2 UI Beschreibung MainForm	16
2.3 Code Behind MainForm	19
<b>3 Abbildungsverzeichnis</b>	<b>24</b>
<b>4 Tabellenverzeichnis</b>	<b>25</b>
<b>5 Codeverzeichnis</b>	<b>26</b>
<b>6 Wochenberichte</b>	<b>27</b>
6.1 Semih Can BURCAK	27
6.1.1 KW38: 20.09.2021 bis 26.09.2021	27
6.1.2 KW 39: 27.09.2021 bis 03.10.2021	28
6.1.3 KW40: 04.10.2021 bis 10.10.2021	29
6.1.4 KW41: 11.10.2021 bis 17.10.2021	30
6.1.5 KW42: 18.10.2021 bis 24.10.2021	31

6.1.6	KW43: 25.10.2021 bis 31.10.2021 . . . . .	32
6.1.7	KW44: 01.11.2021 bis 07.11.2021 . . . . .	33
6.1.8	KW45: 08.11.2021 bis 14.11.2021 . . . . .	34
6.1.9	KW46: 15.11.2021 bis 21.11.2021 . . . . .	35
6.1.10	KW47: 22.11.2021 bis 28.11.2021 . . . . .	36
6.1.11	KW48: 29.11.2021 bis 05.12.2021 . . . . .	37

## **7 Wochenberichte 38**

7.1	Semih SÖNMEZ . . . . .	38
7.1.1	KW38: 20.09.2021 bis 26.09.2021 . . . . .	38
7.1.2	KW39: 27.09.2021 bis 03.10.2021 . . . . .	39
7.1.3	KW40: 04.10.2021 bis 10.10.2021 . . . . .	40
7.1.4	KW41: 11.10.2021 bis 17.10.2021 . . . . .	42
7.1.5	KW42 + KW43: . . . . .	43
7.1.6	KW44: 01.11.2021 bis 07.11.2021 . . . . .	46
7.1.7	KW45: 08.11.2021 bis 14.11.2021 . . . . .	47
7.1.8	KW46: 15.11.2021 bis 21.11.2021 . . . . .	49
7.1.9	KW47: 22.11.2021 bis 28.11.2021 . . . . .	50
7.1.10	KW48: 29.11.2021 bis 05.12.2021 . . . . .	51
7.1.11	KW49: 06.12.2021 bis 12.12.2021 . . . . .	52
7.1.12	ab KW50 . . . . .	54

## DIPLOMARBEIT DOKUMENTATION

Name der Verfasser   innen	Semih Can, Semih, Manuel, Mateo
Jahrgang   Schuljahr	5Klasse   20SJ   20SJ
THEMA der Diplomarbeit	Spiel- und Kontrollerentwicklung
Aufgabenstellung	Es ist ein Videospiel welches mit einem Kontroller einfach zu bedienen ist. Dabei soll auch eine kleine Geschichte mit dem Spiel erzählt werden. Die Diplomarbeit soll auch ein How-To für Erstellung eines Unity-Spiels gesteuert durch Megacard und ESP Kontrollers sein.

## Individuelle Themenstellung im Rahmen des Gesamtprojektes

Semih Can	Herr Burcak macht das Projektmanagement des Projektes, die Auslesung der Controller Signale und das Audio, Story, Benutzeroberfläche, Webserver und den Multiplayer-Teil des Spiels.
Semih	Die grundlegenden Bewegungen vom Charakter und Gegner, sowie die Physik des Spiels werden von Semih Sönmez programmiert. Auch die Umgebung mit Missionen und Levels wird vom Herrn Sönmez entwickelt.
Manuel	Manuel Rath ist für die Steuerung des Spiels zuständig. Der Controller wird aus der Megacard von der Schule erstellt. Die Kommunikation zwischen Controller und Spiel erfolgt über USB oder Bluetooth.
Mateo	Herr Herceg entwickelt den 2. Controller für die Steuerung. Die Daten werden mittels ESP Moduls per WLAN oder Bluetooth an den PC gesendet. Zudem werden für beide Controller ein Gehäuse und Stromversorgung angelegt.
Realisierung	Bei Realisierung angeben, sonst freilassen.
Ergebnisse	Das Spiel ist in Unity-3D zu realisieren. Die Highscores werden auf einem zentralen Server gespeichert. Die Eingaben vom Controller sollen von der Megacard/ESP ausgelesen und mittels USB/WLAN/Bluetooth an das Spiel weitergeleitet werden und die Spielfigur soll sich entsprechend bewegen. Der aus einem eigenen Akku und Gehäuse bestehender Controller soll auch als Feedback beim Spielen vibrieren.
Einsichtnahmen**)	Archiv der HTL Rankweil, <a href="http://www.diplomarbeiten.berufsbildendeschulen.at">www.diplomarbeiten.berufsbildendeschulen.at</a>



# DIPLOMA THESIS DOCUMENTATION

Author(s)	Semih Can, Semih, Manuel, Mateo
Form   Academic year	5Klasse   21SJ   22SJ
Diploma Thesis Topic	Game and Controllerdevelopment
Assignment of Tasks	At the end of this diploma thesis a multiplayer game and two controllers made with the in-house Megacard and ESP to control the game will be developed.

## Individual Tasks within the overall project

Semih Can	Mr. Burcak will do the management of this project, reading and evaluating signals from the controllers, the audio, the story, UI, Webserver and the multiplayer of the game.
Semih	The character movements and the physics of the game will be developed by Mr. Sönmez. The environment, missions and levels will also be developed by Mr. Sönmez.
Manuel	Mr. Rath is responsible for the controls of the game. The Megacard-Controller will be developed by him and it will communicate via USB and Bluetooth.
Mateo	Mr. Herceg is also responsible for the controls of the game. He will develop the second controller, it will be developed with ESP and it will communicate via WIFI and Bluetooth.
Realisation	Bei Realisierung angeben, sonst freilassen.
Result	The game is to be developed in Unity-3D. The highschores will be saved on a central server. The inputs from the controllers are to be sent via USB/WIFI/Bluetooth to the game and character should move according to the input. The controllers will vibrate as feedback and have their own power source.
Publication**) )	Archiv der HTL Rankweil, <a href="http://www.diplomarbeiten.berufsbildendeschulen.at">www.diplomarbeiten.berufsbildendeschulen.at</a>

## ZUSAMMENFASSUNG

Darlegung des Themas, der Fragestellung, der Problemformulierung sowie der Ergebnisse. Kurze, prägnante Information über den Inhalt der Arbeit; keine Wertungen oder Meinungen (allerdings in ganzen Sätzen!).

### A Aufgabenstellung

- Von welchem Wissens- oder Entwicklungsstand im Umfeld der Aufgabenstellung wird ausgegangen bzw. welche Ergebnisse und Erkenntnisse gibt es bereits zum Thema?
- Welches Ziel soll erreicht werden?
- Warum und für wen ist das definierte Ziel von Interesse?

### B Umsetzung

- Auf welche fachtheoretischen/-praktischen Grundlagen wurde zurückgegriffen?
- Welche Lösungsansätze/Methoden wurden gewählt?
- Warum gerade diese und keine anderen?
- Welche Alternativen gäbe es noch?
- Es könnte ev. auf den Bearbeitungsprozess im Team eingegangen werden (Reflexion).

### C Ergebnisse

- Worin besteht der Beitrag zur Lösung der Aufgabenstellung? (Website-Erstellung, Marketingkonzept, ...)
- Welches Produkt soll erstellt werden?
- Es könnte ev. darauf eingegangen werden, ob die Diplomarbeit bei Wettbewerben eingereicht wurde oder ob es Prämierungen gab.

Das sind Fragestellungen, die in der ZUSAMMENFASSUNG und auch in der Übersetzung in englischer Sprache im ABSTRACT vorkommen können (als Vorschlag – je nach Projekt ergeben sich auch andere Fragen). Die Struktur mit A Aufgabenstellung, B Umsetzung und C Ergebnisse muss aber jedenfalls eingehalten werden!

## Abstract (English)

s. Zusammenfassung

## Diplomarbeit "MazeRun" 5AHEL 2021-22

# 1 Lastenheft / Pflichtenheft

## 1.1 Kurzbeschreibung

Das Spiel ist in der 3D-Umgebung in Unity zu realisieren. Die Highscores werden auf einem zentralen Server gespeichert. Die Eingaben vom Kontroller sollen von der Megacard/ESP ausgelesen und mittels USB/WLAN evtl. Bluetooth an das Spiel weitergeitet werden. Anschließend sollte sich die Spielfigur entsprechend bewegen. Der aus einem eigenen Akku und Gehäuse bestehender Kontroller soll auch als Feedback beim Spielen vibrieren. Die Diplomarbeit soll auch ein How-To für Erstellung eines Unity-Spiels/Megacard & ESP Kontrollers sein.

## 1.2 Aufgaben einzelner Projektmitglieder

Name	Individuelle Themenstellung
Semih Can Burcak (Hauptverantwortlich)	Projektmanagement / Spielsoftwareentwicklung / Spieldesign / Onlinespielentwicklung
Semih Sönmez	Spielsoftwareentwicklung / Spieldesign
Manuel Rath	Schaltungsdesign / Analoglayout / Microcontroller Programmierung
Mateo Herceg	Schaltungsdesign / Analoglayout / Microcontroller Programmierung

## 1.3 Projektziele nach SMART

### 1.3.1 Spezifisch

Das Spiel soll professionell aussehen und ohne Probleme online von jeden gespielt werden können. Die Spielstände der Spieler sollen einwandfrei auf einem Server gespeichert werden. Die selbstgemachten Kontroller sollen ebenfalls einwandfrei funktionieren. Die Verbindung zwischen dem Spiel (C#) und Kontroller (C) soll automatisch sein bzw. die Kontroller sollen Plug & Play Kontroller sein.

### 1.3.2 Messbar

Jede neue Funktion des Spiels soll ausführlich getestet werden. Die Verbindung des Kontrollers zum Spiel wird im Spiel getestet, indem wir schauen ob die Tasten und die Joysticks das machen, was wir wollen.

Nachdem das Spiel und die Controller fertig entwickelt wurden, wird alles nochmals getestet um evtl. Bugs zu finden und zu schauen ob alles wie geplant funktioniert.

### 1.3.3 Realistisch

Die Programmiersprachen C# und C wurden im Unterricht schon durchgenommen. Die Kommunikation zwischen PC und  $\mu$ C wurde noch nicht gemacht, aber mit der richtigen Hilfe bzw. Professor sollte es realisierbar sein. Das Projekt wird Schritt für Schritt entwickelt und ist daher realistisch.

### 1.3.4 Terminiert

<i>Datum</i>	<i>Meilenstein</i>
<b>Anfang Oktober (04.10.2021)</b>	Auswahl und Layout der Hardware, Anfangen das Spiel zu programmieren
<b>Anfang November (01.11.2021)</b>	Die Umgebung/Charaktere des Spiels sollen fertig sein,
<b>Mitte bis Ende November (30.11.2021)</b>	Entwicklungsphase des Spiels fertig,
<b>Dezember und Jänner</b>	Online-Funktion des Spiels programmieren
<b>Ende 1. Halbjahr (14.02.2022)</b>	Fertigstellung des Projekts

**Tabelle 1.2:** *Meilensteine*

Das Projekt soll in diesem Halbjahr (Wintersemester SJ 2021-22) fertiggestellt werden.

## 1.4 Produkteinsatz

Das Spiel sollte mit einem Computer gestartet werden, der Zugang auf ein Internet verfügt. Die Controller sollten entweder über USB, Bluetooth oder WLAN mit dem PC verbunden sein. Bei kabellosem Controller soll sicher gestellt werden, dass die Batterie noch Energie hat.

## 1.5 Funktionale Anforderungen

Durch den Controller sollten folgende Bewegungen durchgeführt werden können:

- Grundlegende Bewegungen (vorne, hinten, rechts, links,)
- Springen des Spielers
- Bewegung der Kamera

## 1.6 Projektphasen und Meilensteine

### 1.6.1 Spiel

Zuerst sollen die Grundlegenden Bewegungen vom Charakter programmiert werden. Anschließend kann das Multiplayer Spielmodus programmiert werden. Das Know-How soll bis das Spiel bereit ist (Bewegungen, Umgebung etc.), um es online zu spielen, erschafft werden.

### 1.6.2 Kontroller

Zuerst werden Layout und Schaltplan erstellt und daraus wird die Leiterplatte entstehen. In der zwischen Zeit werden die einzelnen Bauteile bestellt und Angesteuert. Nach der Fertigung der Hardware starten wir mit einzelnen Testprogrammen, die wir später für die fertige Software brauchen. Zu guter Letzt wird ein passendes Gehäuse mit einem Akku konstruiert.

## 1.7 Arbeitspakete (aufgelistet)

### 1.7.1 Semih Can Burcak

- Benutzeroberfläche des Spiels
  - Username Eingabe - must have
  - Wahlmenü (Einzelspieler oder Mehrspieler, Level usw.) - must have
  - Es soll aus HTL Rankweil Farben bestehen (Blau, Grau, Schwarz)
  - Benutzerfreundlich gestalten - must have
- Audio
  - Copyrightfreie Audios verwenden - must have
  - Die Musik und die Sounds sollen dem Spiel mehr energetisch machen - must have
  - Lautstärke gut einstellen (nicht zu laut oder leise)
- Multiplayer - must have
  - Webserver einrichten
  - Datenbank erstellen
  - Datenbank und Webserver mit Unity verbinden
  - Schauen, dass jeder Spieler ein Charakter zugeteilt kriegt
- Story
  - Es soll nicht langweilig sein und Interesse wecken - must have

- Zuerst auf dem Papier das Konzept bzw. Story festigen und erst programmieren
- Projektmanagement - must have
  - Das Team soll immer organisiert bleiben
  - Jeder soll seine Aufgaben rechtzeitig erledigen, damit kein Zeitdruck entsteht
  - Ordentlich Dokumentieren

### 1.7.2 Semih Sönmez

- Entwicklung des Hauptcharakters
  - Grundlegende Bewegungen (vorne, hinten, rechts, links) -must have
  - Springfunktion um über Hindernissen zu springen -must have
  - Schussfunktion um Gegner zu eliminieren
- Entwicklung der Gegner
  - Die Gegner sollen zum Hauptcharakter rennen -must have
  - Bei einem Kontakt, soll das Leben vom Hauptcharakter weniger werden -must have
- Umgebung
  - Die Charaktere sollen die Umgebung nicht verlassen dürfen -must have
  - Die Umgebung soll gut beleuchtet sein -must have
- Missionen & Levels
  - 5 verschiedene Levels -must have
  - Missionen sollen interessant und bewältigbar sein -must have
- Kamera
  - Bei Einzelspielermodus soll die Kamera den Hauptcharakter folgen -must have
  - Bei Multiplayermodus soll die Kamera für beide Spieler angepasst werden -must have

### 1.7.3 Manuel Rath

- Leiterplatte - must have
  - Schaltplan mit geeigneten Bauteilen & Stromversorgungsschaltung erstellen
  - Layout möglichst kompakt
- Gehäuse - must have
  - handliches und nicht zu großes Gehäuse
  - mit 3D Drucker erstellen
- Verbindung von Kontroller zum PC
  - Daten zuerst über UART an den PC senden - must have
  - Daten mit Bluetooth Modul an den PC senden
- Komponenten Ansteuern mit der Megacard - must have



- Joystick ansteuern und daten auslesen
- Vibrationsmotor ansteuern
- Tasten auslesen
- Programme in einem Spiel testen

#### 1.7.4 Mateo Herceg

- Leiterplatte - must have
  - Layout und Schaltplan mit diversen Bauteilen (+ESP Modul) entwickeln & Stromversorgungsschaltung hinzu entwickeln
  - Strukturiertes Layout + übersichtlicher Schaltplan
- Verbindung Kontroller/Computer
  - Daten mittels ESP Modul über WLAN an den PC senden - must have
  - Daten mittels ESP Modul über Bluetooth an den PC senden
- Ansteuerung - must have
  - Knöpfe/Tasten ansteuern und Signale/Daten auslesen mittels ESP Modul
  - Joystick ansteuern und Signale/Daten auslesen mittels ESP Modul
  - Vibrationsmotor ansteuern mittels ESP Modul
  - Alle angesteuerten Komponenten an PC senden und Testen
- Gehäuse - must have
  - Gehäuse konstruieren
  - Gehäuse mittels 3D Drucker erstellen
  - einfaches handliches Gehäuse

## 2 Kapitel 2

### 2.1 BlaBla 1

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. <sup>1</sup>

#### 2.1.1 BlaBla 2

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. <sup>1,2</sup>

### 2.2 UI Beschreibung MainForm

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine

---

<sup>1</sup> Bringhurst1993

<sup>2</sup> wiki:latex.

falsche Anmutung vermitteln.

```

18 namespace Rainfall_StatisticApp
19 {
20     /// <summary>
21     /// Interaction logic for MainWindow.xaml öüß
22     /// </summary>
23     public partial class MainWindow : Window
24     {
25         RainValue_List rainValueItems = new RainValue_List();
26         public MainWindow()
27         {
28             InitializeComponent();
29         }
30
31         private void ButtonAdd_Click(object sender, RoutedEventArgs e)
32         {
33             string location = textboxLocation.Text;
34             double amount;
35
36             try
37             {
38                 amount = double.Parse(textboxAmount.Text);
39             }
40             catch (Exception exp)
41             {
42                 textboxAmount.Background = Brushes.Red;
43                 Console.Beep();
44                 MessageBox.Show(exp.Message);
45                 return;
46             }
47
48             try
49             {
50                 RainValue r1 = new RainValue(location, amount,
51                     checkboxIsCapital.IsChecked.Value,
52                     ↳ datepickerTimeStamp.SelectedDate.Value);
53                 listboxRainValue.Items.Add(r1);
54                 rainValueItems.Add(r1);
55                 labelaverage.Content = $"Durchschnitt:
56                 ↳ {rainValueItems.Average:0.0}";

```

```

56         }
        catch (Exception exp)
58     {
        MessageBox.Show(exp.Message);
60         return;
        }
62     }

64     private void TextboxAmount_SelectionChanged(object sender,
        ↪ RoutedEventArgs e)
        {
66         textboxAmount.Background = Brushes.White;
        }
68     }
}

```

**Listing 2.1:** Code behind MainWindow.xaml.cs

Computerarchitektur siehe **Tanenbaum2006** von **Tanenbaum2006**<sup>3</sup>

- DictLeoOrg
- USAB2011
- DictChemnitz

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele

<sup>3</sup> **Tanenbaum2006.**

verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Ebenso erwähnenswert ist **KeithThesis**.

Ein lesenswertes Buch ist **einstein**<sup>4</sup>

Der Code vom Code Schnipsel 2.1, s. S 18, hat die Aufgabe. Wie man hier sieht, kann gut auf Code mit einem label referenziert werden.

## 2.3 Code Behind MainForm

Und hier folgt der Code für das UI.

```

14 <DockPanel Background="LightBlue" Grid.Column="0">
    <WrapPanel Margin="5" DockPanel.Dock="Top">
16         <Label Width="50"> Ort:</Label>
        <TextBox Name="textboxLocation" Width="100"> </TextBox>
18     </WrapPanel>
    <WrapPanel Margin="5" DockPanel.Dock="Top">
20         <Label Width="50"> Menge:</Label>
        <TextBox Name="textboxAmount" Width="100" SelectionChanged="
            ↪ TextboxAmount_SelectionChanged"></TextBox>
22     </WrapPanel>
    <WrapPanel Margin="5" DockPanel.Dock="Top">
24         <CheckBox Name="checkboxIsCapital">Hauptstadt</CheckBox>
    </WrapPanel>
26     <WrapPanel Margin="5" DockPanel.Dock="Top">
        <Label Width="50"> Datum</Label>
28         <DatePicker Name ="datepickerTimeStamp"/>
        <Label Name="labelaverage"/>
30     </WrapPanel>
    <Button DockPanel.Dock="Left" Name="buttonAdd" Margin="5"
        ↪ Width="100" HorizontalAlignment="Left" Padding="5"
        ↪ Click="ButtonAdd_Click">Hinzufügen</Button>
32 <ListBox ScrollViewer.VerticalScrollBarVisibility="Visible"
        ↪ Name="listboxRainValue"/>
</DockPanel>

```

<sup>4</sup> einstein.

### Listing 2.2: *MainWindow.xaml*

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

fas	fsa
fas	fas

**Tabelle 2.1:** *Beschriftung der Tabelle*

Time (s)	Triple chosen	Other feasible triples
0	(1, 11, 13725)	(1, 12, 10980), (1, 13, 8235), (2, 2, 0), (3, 1, 0)
2745	(1, 12, 10980)	(1, 13, 8235), (2, 2, 0), (2, 3, 0), (3, 1, 0)
5490	(1, 12, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
8235	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
10980	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
13725	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
16470	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
19215	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
21960	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
24705	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
27450	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
30195	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
32940	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
35685	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
38430	(1, 13, 10980)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
Fortsetzung nächste Seite		

**Tabelle 2.2 – Fortsetzung**

Time (s)	Triple chosen	Other feasible triples
41175	(1, 12, 13725)	(1, 13, 10980), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
43920	(1, 13, 10980)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
46665	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
49410	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
52155	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
54900	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
57645	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
60390	(1, 12, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
63135	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
65880	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
68625	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
71370	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
74115	(1, 12, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
76860	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
79605	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
82350	(1, 12, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
85095	(1, 12, 13725)	(1, 13, 10980), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
87840	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
90585	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
93330	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
96075	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
98820	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
101565	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
104310	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
107055	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
109800	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
112545	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
115290	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
Fortsetzung nächste Seite		

**Tabelle 2.2 – Fortsetzung**

Time (s)	Triple chosen	Other feasible triples
118035	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
120780	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
123525	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
126270	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
129015	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
131760	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
134505	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
137250	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
139995	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
142740	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
145485	(1, 12, 16470)	(1, 13, 13725), (2, 2, 2745), (2, 3, 0), (3, 1, 0)
148230	(2, 2, 2745)	(2, 3, 0), (3, 1, 0)
150975	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
153720	(1, 12, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
156465	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
159210	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
161955	(1, 13, 16470)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)
164700	(1, 13, 13725)	(2, 2, 2745), (2, 3, 0), (3, 1, 0)

**Tabelle 2.2:** Eine lange Tabelle



**Abbildung 2.1:** Beschriftung unter Abbildung<sup>5</sup>



Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

---

<sup>5</sup> (htl'Logo)

### 3 Abbildungsverzeichnis

2.1	Beschreibung für Verzeichnis . . . . .	22
7.1	. . . . .	43
7.2	. . . . .	44
7.3	. . . . .	47
7.4	. . . . .	48
7.5	. . . . .	49
7.6	. . . . .	51
7.7	Charakter . . . . .	52
7.8	. . . . .	55
7.9	. . . . .	55
7.10	. . . . .	56
7.11	. . . . .	56

## 4 Tabellenverzeichnis

1.2	Meilensteine . . . . .	12
2.1	Beschriftung der Tabelle - für Verzeichnis . . . . .	20
2.2	Eine lange Tabelle . . . . .	22

## 5 Codeverzeichnis

2.1	Code behind MainWindow.xaml.cs . . . . .	18
2.2	MainWindow.xaml . . . . .	20

## 6 Wochenberichte

### 6.1 Semih Can BURCAK

#### 6.1.1 KW38: 20.09.2021 bis 26.09.2021

##### **Arbeit in der Schule**

Nach der Besprechung mit Professor Rusch, haben wir über das Projekt diskutiert. Wir haben über den Antrag, Pflichtenheft, Verlauf des Projektes (wer was macht, wie man einzelne Meilensteine realisieren wird, welche Schnittstellen werden verwendet, welche Assets werden verwendet, was soll die Geschichte des Spiels sein etc.)

##### **Arbeit außerhalb der Schule**

- DA-Antrag, Pflichtenheft - ca 2h
- LaTeX und Virtual-Machine einrichten - 1h
- Unity Installation und Beginner Tutorials - 2h

Die Installation von LaTeX machte mir am Anfang Probleme aber ich habe es gelöst, indem ich den ganzen Prozess nochmals durchgemacht habe. DA-Antrag musste ich verbessern und manche Teile neu schreiben, hier habe noch zusätzlich die Sachen zum Spiel geschrieben und die anderen Teile des Projekt wurden von den entsprechenden Mitglieder geschrieben. Am Ende habe ich alles in einer Datei zusammengefasst. Bei dem Lastenheft/Pflichtenheft habe ich mein Teil bezüglich des Spiels und Projektmanagement geschrieben und anschließend es meinen Teamkollegen geschickt. Ich habe das Lastenheft am Sonntag geschickt, was meiner Meinung nach spät ist aber ich werde versuchen mehr diszipliniert zu arbeiten, damit ich meinen Teamkollegen kein Zeitdruck erzeuge.

##### **Was ist geplant für die Nächste Woche**

Diese Woche werde ich mich auf Unity konzentrieren und viele Tutorials durchmachen. Da ich für ein nicht-existierendes Spiel keine Online-Funktion programmieren kann, werde ich schauen, dass wir das Spiel schnell wie möglich für Online bereitmachen. Ich werde aber natürlich in dieser Zeit auch das Know-How für Onlinespielentwicklung erschaffen.

## 6.1.2 KW 39: 27.09.2021 bis 03.10.2021

### Arbeit in der Schule

Wir haben in der Schule Verbesserungsvorschläge bezüglich unseres Antrags vom Herrn Professor Rusch bekommen. Wir haben eine Stunde Zeit bekommen um den Antrag zu verbessern. Dies haben wir auch getan und anschließend im System des Bildungsministeriums abgegeben. Danach hat jeder Teammitglied seine eigenen Meilensteine definiert und es im Pflichtenheft geschrieben.

### Arbeit außerhalb der Schule

- DA-Antrag - 0.5h
- Unity Menu (UI) Tutorials - 2.5h<sup>6</sup>

Diese Woche habe ich angefangen Unity-UI zu üben. Dazu habe ich auf YouTube ein gutes Starter-Tutorial gefunden. Jedoch lief bei mir nicht alles wie geplant. Da ich Unity nicht so gut kenne, musste ich immer wieder mit Fehler kämpfen, z.B. die Buttons sind auf einmal nicht gegangen oder Unity erkannte mein Code bzw. die Funktionen im Code nicht mehr usw.. Ich konnte bisher diese Fehler nicht vollständig verbessern, denn sie erscheinen immer wieder und jedes Mal ist etwas anderes Falsch, also ich kann dieselbe Lösung, wie vorher, nicht anwenden. Herr Professor Rusch hat uns noch paar kleine Verbesserungsvorschläge gemacht bezüglich des Antrags, ich habe den Antrag verbessert und anschließend abgegeben.

### Was ist geplant für die Nächste Woche

Ich werde nächste Woche schauen, dass ich ein Start-Menü erstelle und es meinen Teamkollegen zeige, damit wir uns auf ein Design einigen. Wenn ich mit dem Menü rasch fertig werde, werde ich über den Multiplayer-Modus (+Webserver) in Unity recherchieren. Ich werde auch schauen, dass wir jetzt von Typora auf LaTeX wechseln.

---

<sup>6</sup> MenuTutorial.

### 6.1.3 KW40: 04.10.2021 bis 10.10.2021

#### Arbeit in der Schule

Ich habe in der Schule weiter am Start-Menü geschaffen, ich war auf der Fehlersuche, wieso die Buttons nicht funktionieren. Ich habe dann festgestellt, dass ein EventHandler gefehlt hat, nachdem haben die Buttons funktioniert. Dann wurde die Funktion im Code von Unity nicht erkannt, ich habe die Lösung noch nicht gefunden. Wir haben auch über den Megacard-Kontroller gesprochen, welche Einzelteile wir brauchen, wie man den Akku wieder aufladen kann etc., bei Fragen haben wir uns an den Herrn Professor Zudrell-Koch gewendet. Nach der großen Pause (15:20) haben wir uns mit dem Herrn Professor Rusch in der Klasse getroffen und er hat uns gezeigt, wie man LaTeX und Gitlab verwendet, was sehr hilfreich war.

#### Arbeit außerhalb der Schule

Ich habe meine Wochenberichte und Pflichtenheft auf LaTeX übertragen, am Anfang war es ein bisschen kompliziert aber man kommt schnell im Rhythmus. Ich habe versucht Quellen einzufügen, aber es ist leider nicht gegangen, ich habe dann später entdeckt, dass ich Syntaxfehler hatte aber auch nach der Verbesserung wurde meine eingefügte Quelle im Quellenverzeichnis nicht gezeigt.

Am Montag (11.10.2021) hat mir Herr Professor Rusch gezeigt wie man Quellen einfügt und somit habe ich das Problem lösen können.

Ich habe das Start-Menü in Unity fertig gemacht. Ich musste mit vielen Fehlern kämpfen, die Buttons haben auf einmal nicht funktioniert, ich musste ein Event-System einfügen um das zu lösen. Dann hat Unity mein Code nicht erkannt, die Lösung dafür war: ich musste das GameObject selbst verwenden anstatt der C#-Datei.

Mehr konnte ich leider nicht machen, weil ich über das Wochenende krank geworden bin.

- Wochenberichte und Pflichtenheft in LaTeX einfügen - 2h
- Start-Menü erstellen - 1.5h

#### Was ist geplant für die Nächste Woche

Ich werde schauen, dass ich die Menüs gut designe und über Multiplayerspielentwicklung in Unity (+Webserver) recherchiere.

#### 6.1.4 KW41: 11.10.2021 bis 17.10.2021

##### **Arbeit in der Schule**

Herr Professor Rusch hat uns gezeigt wie man Quellen einfügt in LaTeX. Danach habe ich gefunden, was mein Fehler war in Unity. Ich konnte die Buttons nicht drücken, ich habe dieses Problem gelöst, indem ich ein EventSystem eingefügt habe in meinem Projekt.

##### **Arbeit außerhalb der Schule**

Ich war leider krank und konnte für die Diplomarbeit nichts machen.

##### **Was ist geplant für die Nächste Woche**

siehe letzte Woche (KW40)



### 6.1.5 KW42: 18.10.2021 bis 24.10.2021

#### Arbeit in der Schule

In der Schule habe ich über Online-Multiplayer-Spielentwicklung (OMSE) in Unity recherchiert. Es gibt eine große Auswahl von Services bzw. Bibliotheken, die man wählen kann für OMSE, aber ich habe Photon gewählt, weil Sie gratis einen cloudbasierten Webserver anbieten und das Service ist auf C# basiert.

Für Mehreres hatte ich leider keine Zeit mehr, da wir nach der zweiten Stunde für Maturafotos fort müssten.

#### Arbeit außerhalb der Schule

Ich habe ein Tutorial-Spiel<sup>7</sup> heruntergeladen und habe angefangen mithilfe Photon, und mit einem entsprechenden Tutorial<sup>8</sup> dazu, das Spiel in ein Online-Multiplayer-Spiel zu umprogrammieren. Da es mein erstes Mal war mit Photon, musste ich zuerst mal die Lernkurve überwinden. Ich bin noch nicht fertig geworden mit dem Umprogrammieren. Ich habe geschafft das Spiel mit dem Server zu verbinden und Räume zu erstellen, ich konnte mit zwei Builds in den gleichen Raum beitreten.

Ich habe mich diese Woche mit den Menü-Designs nicht beschäftigt, obwohl es geplant war. Weil, ich denke, dass es momentan nicht sehr nötig ist bzw. eine hohe Priorität hat.

- Recherche über Photon - 1h
- Tutorial-Spiel auf Online-Spiel umprogrammiert - 3h

#### Was ist geplant für die Nächste Woche

Das Tutorial-Spiel fertig umprogrammieren.

---

<sup>7</sup> Karting Microgame Unity3D.

<sup>8</sup> Unity Photon Tutorial.

## 6.1.6 KW43: 25.10.2021 bis 31.10.2021

### Arbeit in der Schule

**\*\*HERBSTFERIEN\*\***

### Arbeit außerhalb der Schule

Ich habe das Tutorial-Spiel<sup>9</sup> entsprechend zu dem Tutorial<sup>10</sup> fertig umprogrammiert. Ich musste diese Woche nur noch die separate Bewegung und Synchronisation der Spieler programmieren. Aber da der Code vom Spiel nicht von mir stammt und auch sehr kompliziert war, musste ich viel mit Fehlern kämpfen. Doch ein Fehler konnte ich nicht lösen. Der Bewegung-Code hatte noch die Photon-Bibliothek nicht und es erkannte sie auch nicht. Ich musste zuerst die Photon-Bibliothek Referenzen einfügen, was leider nicht nur mit einem Rechtsklick auf Referenzen zu machen war, obwohl es so machbar sein sollte, es hat mich sehr viel Zeit gekostet aber zumindest hat Visual Studio die Photon-Bibliothek erkennen können. Doch dann kam der nächste Fehler, der Code wurde nicht kompiliert von Unity. Ich habe den Fehler mit paar Tricks, wie z.B. sehr viele Leerzeichen einfügen, lösen können aber auch nachdem Unity mein Code kompiliert hat, bekam ich immer noch den Fehler, dass die Photon-Bibliothek nicht gebunden war und deshalb manche Sachen im Code nicht erkannt werden können.

- Spiel fertig umprogrammiert - 1h
- Fehlersuche - 5h

### Was ist geplant für die Nächste Woche

Eine Datenbank erstellen mit MySQL und es mit Unity verbinden.

---

<sup>9</sup> Karting Microgame Unity3D.

<sup>10</sup> Unity Photon Tutorial.

### 6.1.7 KW44: 01.11.2021 bis 07.11.2021

#### **Arbeit in der Schule**

**\*\*HERBSTFERIEN\*\***

#### **Arbeit außerhalb der Schule**

Ich habe eine Datenbank erstellt mittels MySQL. Ich habe ein Login-Menü erstellt und einen einfachen Klick-Counter programmiert um Scores zu speichern können. Ich habe auch was sehr wichtiges gelernt, ich habe den Namen der Szene geändert bevor ich es gespeichert habe und alles hat sich gelöscht. Man muss die Szene immer speichern bevor man den Namen ändert.

Ich habe auch Kapiteln "DIPLOMARBEIT DOKUMENTATION" und "DIPLOMA THESIS DOCUMENTATION" ergänzt.

- Fehlende Teile der Doku ergänzt - 1h
- Datenbank-Setup, Recherche über Unity und MySQL, Programmierung und Erstellung des Login-Menüs und Klick-Counter. -2.5h

#### **Was ist geplant für die Nächste Woche**

Die Datenbank mit Unity verbinden.

### 6.1.8 KW45: 08.11.2021 bis 14.11.2021

#### **Arbeit in der Schule**

Wochenberichte verglichen und über die Doku diskutiert. Überschriften und Sektionen der Doku und Wochenberichte diskutiert.

Wochenberichte von Manuel Rath exportiert auf LaTeX.

#### **Arbeit außerhalb der Schule**

Ich war leider mit der Schule beschäftigt und konnte für die DA nicht vieles machen.

Menü Designs verschönert - 1h Musik eingefügt (Menüs) - 30min

#### **Was ist geplant für die Nächste Woche**

Musik Überlappung Fehler beheben. Evtl. die Datenbank mit Unity verbinden.

#### **Welche Schwierigkeiten hat es gegeben**

Die Musik überlappt sich bei jeder Rückkehr zum Start-Menü.

### 6.1.9 KW46: 15.11.2021 bis 21.11.2021

#### **Arbeit in der Schule**

Fehlersuche Musik. Ich habe den Fehler nicht direkt lösen können und habe ein extra Menü erstellt das man nur am Anfang sieht und nicht wieder zurück kann und ich starte die Musik dort.

#### **Arbeit außerhalb der Schule**

Ich habe leider wegen Tests für die DA nichts machen können.

#### **Was ist geplant für die Nächste Woche**

Die Datenbank mit Unity verbinden.

#### **Welche Schwierigkeiten hat es gegeben**

-

**6.1.10 KW47: 22.11.2021 bis 28.11.2021****Arbeit in der Schule**

Wir haben unsere Unity-Projekte mit Herrn Sönmez zusammengeführt. Ich habe die aktuelle Game-Szene mit der von Herr Sönmez gewechselt.

**Arbeit außerhalb der Schule**

Ich habe beim Tutorial weitergemacht und die Unity bzw. C#-Teil und den PHP-Teil für die Datenbankverbindung programmiert. - 2h

Fehlersuche Datenbankverbindung - 2h

**Was ist geplant für die Nächste Woche**

Fehlerbehebung Datenbank

**Welche Schwierigkeiten hat es gegeben**

Ich habe Syntaxfehler in meinen Codes gefunden und das Passwort für die Server-Connection (PHP) hatte ich falsch. URL für das PHP-Programm in C# war auch nicht korrekt. Aber ich kann leider immer noch nicht auf das PHP-Programm zugreifen mit Unity.

**6.1.11 KW48: 29.11.2021 bis 05.12.2021****Arbeit in der Schule**

Antrag für die Unityumgebung geschrieben und an den Herrn Rusch geschickt.  
Fehlersuche PHP und C#, ich kann jetzt mit Unity auf das PHP-Programm zugreifen. Unity Package gekauft.

**Arbeit außerhalb der Schule**

Fehlersuche Datenbank - 2h Recherche über andere Datenbanken - 1h

**Was ist geplant für die Nächste Woche**

Datenbank Fehler beheben.

**Welche Schwierigkeiten hat es gegeben**

Leider funktioniert die Verbindung noch nicht vollständig. Am Anfang konnte ich mit der Datenbank nicht verbinden, ich habe dann im Code WWW mit UnityWebRequest (weil WWW nicht mehr von Unity unterstützt wird) gewechselt und es hat funktioniert. Aber ich kann leider die Spieler nicht in die Datenbank einfügen.

## 7 Wochenberichte

### 7.1 Semih SÖNMEZ

#### 7.1.1 KW38: 20.09.2021 bis 26.09.2021

##### **Arbeit in der Schule**

Besprechung Diplomarbeit mit Herrn Rusch: 1h

Es wurde besprochen wie wir am Diplomarbeit arbeiten sollen. Jede Woche soll ein Wochenbericht geschrieben werden, welches am Schluss in die Dokumentation eingefügt wird. Das Wochenbericht entsteht aus den folgenden Punkten:

- Was wurde gemacht
- Was ist geplant für die Nächste Woche
- Welche Schwierigkeiten hat es gegeben
- Schwierigkeiten oder Fragen die Aufgekommen sind
- Jeder Schüler muss 150 Stunden außerhalb der Schule am Diplomarbeit arbeiten.

Planung des Spiels. Wie soll das Spiel programmiert werden?

Um das Spiel zu planen, soll ein Lastenheft und ein Pflichtenheft erstellt werden. Damit können wir unser Projektplanen und in kleine Arbeitspakete gliedern.

##### **Arbeit außerhalb der Schule**

Virtuelle Machine: Latex heruntergeladen und installiert 1h

Antrag auf die Diplomarbeit fertig geschrieben 1h

Lastenheft erstellt 1h

Unity-tutorial abgeschlossen 2h

##### **Was ist geplant für die Nächste Woche?**

Es soll mit der Programmierung des Spiels angefangen werden.

Der Charakter des Spiels wird zuerst entwickelt

##### **Welche Schwierigkeiten hat es gegeben**

Es war schwer, das Spiel aufzuteilen. Wer macht was ?



### 7.1.2 KW39: 27.09.2021 bis 03.10.2021

#### **Arbeit in der Schule**

Besprechung Diplomarbeit mit Herrn Rusch **1h**

Die Arbeit wurde gerecht aufgeteilt, so dass alle Mitglieder gleich viel Arbeit haben. Wer macht was? **1h**

Planung des Spiels. Wie soll das Spiel programmiert werden? **1h**

Virtuelle Machine: Latex heruntergeladen und installiert **1h**

#### **Arbeit außerhalb der Schule**

Antrag auf die Diplomarbeit fertig geschrieben **1h**

Lastenheft erstellt **1h**

Unity-tutorial abgeschlossen **2h**

#### **Was ist geplant für die Nächste Woche?**

Es soll mit der Programmierung des Spiels angefangen werden

Der Charakter des Spiels wird als zuerst entwickelt

#### **Welche Schwierigkeiten hat es gegeben?**

Es war schwer, das Spiel aufzuteilen.

### 7.1.3 KW40: 04.10.2021 bis 10.10.2021

#### Arbeit in der Schule

Besprechung mit Herrn Rusch:

Wir haben gemeinsam mit Herrn Rusch das Programm Latex auf unserer Virtual Machine installiert. Anschließend haben wir unser Repository, die wir im Gitlab erstellt haben, in unserem PC geklont.

Dies wurde mit dem folgenden Befehl realisiert: `git clone [url-vom Projekt]`.

**Daten upload:** Um Daten auf GitLab hochzuladen, werden folgende Befehle ausgeführt:

Zuerst muss im Terminal auf unser Repository-Ordner navigiert werden:

`ls -l` : Inhalt des Verzeichnis

`cd Verzeichnisname` : in den Ordner wechseln

`git status` : Abfrage ob im lokalen Repository etwas verändert wurde

`git add` : eine Änderung aus dem Arbeitsverzeichnis wird zur Staging-Umgebung hinzugefügt

`git commit -am "name"` : Der Befehl committet den Snapshot im Staging-Bereich in die Projekt-Historie

`git push` : Änderung werden hochgeladen

**Daten download:** hier muss wieder auf unseren Repository-Ordner navigiert werden.

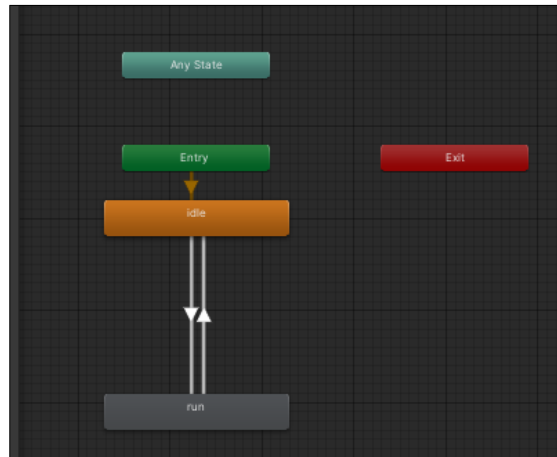
`git pull` : Inhalte vom Repository herunterladen um das lokale Repository zu aktualisieren

Danach hat uns Herr Rusch eine Einführung zum Programm LaTeX gegeben.

**Arbeit außerhalb der Schule** Ich habe am Programm, welches für die Bewegung vom Charakter dient, weitergearbeitet.

Es wurde ein Animator Controller erstellt, bei der 2 Zustände im State Machine definiert worden sind. Nachdem das Spiel gestartet wird(Entry), wird sofort der Zustand idle übernommen. Mit der Animation idle steht der Charakter ohne Bewegung. Doch wenn die Boolvariable `IsWalking` wahr wird, geht der Zustand von idle zu run, welches den Charakter in Bewegung setzt. Wenn die Boolvariable

IsWalking wieder unwahr wird, kährt sie zum default state, also zum idle wieder zurück. Die Animationen wurden vom Character Pack: Free Sample entnommen.



Anschließend wurde der Animator Controller zum Prefab Model zugewiesen.

### **Was ist geplant für die Nächste Woche**

Es soll mit dem Character Controller Programm weitergearbeitet werden.

#### 7.1.4 KW41: 11.10.2021 bis 17.10.2021

##### **Arbeit in der Schule**

Besprechung mit Herrn Rusch:

Wir haben Git auf unserem Windows Rechner installiert. Anschließend wurde eine neue SSH generiert und ins Gitlab eingefügt. Danach wurde das Projekt geklont und die Unity-Projekte hochgeladen. Somit haben wir jetzt sowohl im Windows-Rechner als auch in der Virtual Machine Zugriff auf das Git, wo die Arbeit immer gespeichert werden kann.

##### **Arbeit außerhalb der Schule**

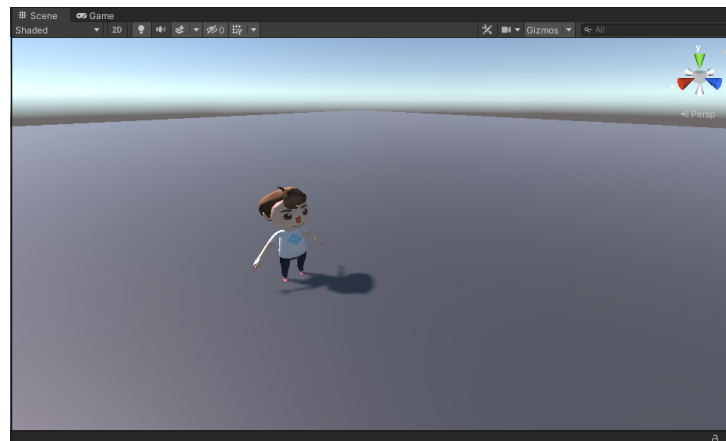
Da ich im Git ständig zur Diplomarbeit Ordner navigieren musste und dies sehr Zeit in Anspruch nahm, habe ich eine Lösung gefunden. Im Git-Ordner C:\Program Files\Git\etc gibt es ein File namens bash.bashrc, welches beim Starten des Gits aufgerufen wird. In dieser Datei habe ich folgendes eingefügt:

```
#set Diplomarbeit directory as default directory cd C:/HTL_2021-22/Diplomarbeit/maz
```

Anschließend habe am Programm, welches für die Bewegung vom Charakter dient, weitergearbeitet und versucht die Bugs zu lösen.

##### **Was ist geplant für die Nächste Woche**

Das Programm für die CharacterController soll fertig programmiert werden.



**Abbildung 7.1**

### **7.1.5 KW42 + KW43:**

#### **Arbeit in der Schule**

Im Unity Asset Store wurde der Character Pack: Free Sample heruntergeladen und in das Projekt eingefügt. Anschließend wurde der Prefab Model in die Scene eingefügt. In der Scene wurde auch mit einem 3D-Cube der Boden erstellt. Danach wurde der Animator Controller und das Character Movement Programm, dem Charakter zugewiesen.

Ergebnis: Der Charakter kann grundlegende Bewegungen wie Rennen, Drehen und Springen ohne Probleme ausführen.

Anschließend habe ich ein bisschen recherchiert wie ich die Gegner programmieren soll, da dies meine nächste Aufgabe ist.

#### **Arbeit außerhalb der Schule**

Meine nächste Aufgabe ist es die Gegner zu programmieren. Dafür habe ich auf der Seite [www.mixamo.com](http://www.mixamo.com) ein Lehrer-Charakter und eine Lauf-Animation heruntergeladen und sie im Projekt eingefügt.

Links:

Youtube-tutorial für Charakter:<sup>11</sup>

Walking Animation:<sup>12</sup>

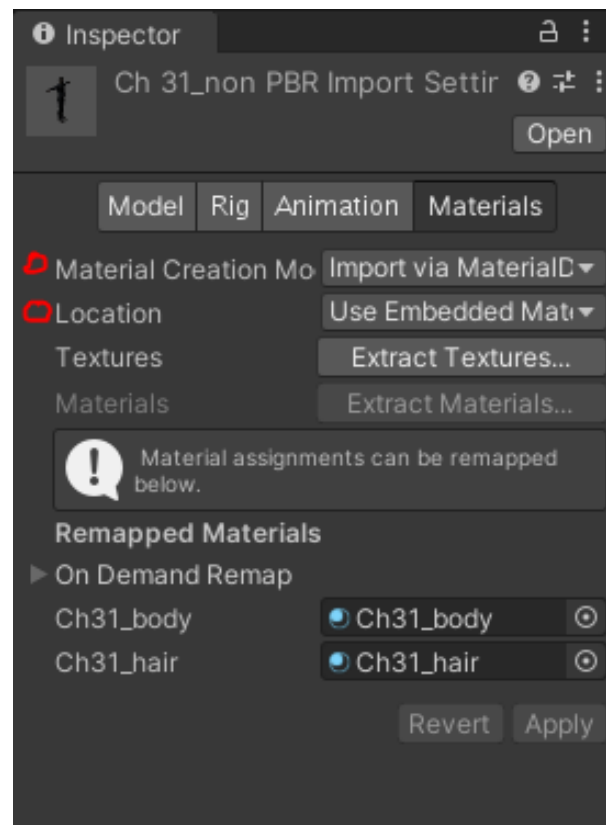
Charakter Leonard:<sup>13</sup>

---

<sup>11</sup> **CharakterTutorial.**

<sup>12</sup> **WalkingAnimation.**

<sup>13</sup> **LehrerCharakter.**



**Abbildung 7.2**

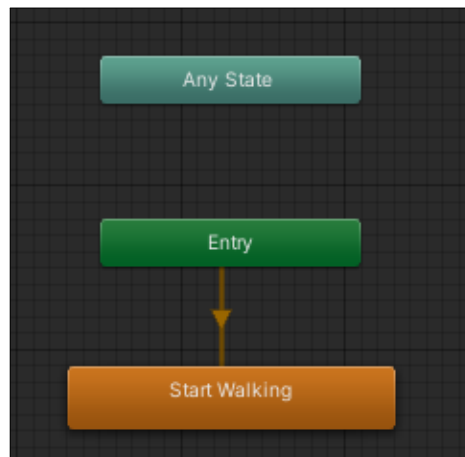
Der eingefügte Charakter war in der Scene grau zu sehen. Also habe ich sein Material vom Charakter im Inspector window extrahiert. Anschließend war der Charakter wieder bunt zu sehen.

Jetzt muss dem Charakter die Walkinganimation zugewiesen werden.

Inspector window: Rig:

- Animation Type auf "Humanoid und Avatar Definiton auf "Copy from other Avatar" umgestellt
- Beim Source den teacher Avatar ausgewählt

Danach wurde ein Animator Controller erstellt, in der wir unsere Animation eingefügt haben. Wie im Bild zusehen ist, wird beim Starten des Spiels, der Start Walking Animation durchgeführt.



### 7.1.6 KW44: 01.11.2021 bis 07.11.2021

#### **Arbeit in der Schule**

Die Funktion vom Lehrer Charakter: Wenn unser Hauptcharakter(Schüler) sich im Blickwinkel vom Lehrer befindet soll das Spiel beendet(Game Over) werden.

Ich habe schon diese Funktion im Projekt John Lemon beim Charakter Gargoyle verwendet. Also wurde aus dem Projekt John Lemon, der Charakter Gargoyle herauskopiert und in unser Projekt eingefügt. Anschließend wurde der Gargoyle Charakter getestet. Leider hat es nicht funktioniert.

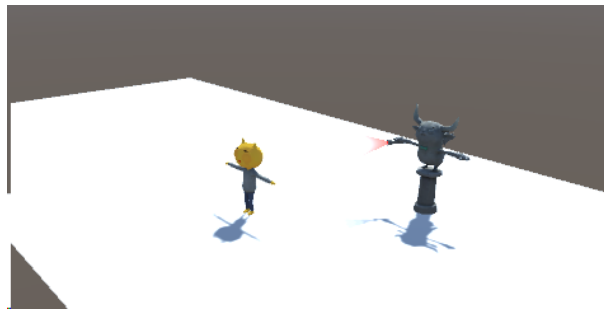
#### **Arbeit außerhalb der Schule**

Ich habe versucht das Problem zu beheben.

#### **Welche Schwierigkeiten hat es gegeben**

Der Charakter Gargoyle hat nicht funktioniert, obwohl es im anderen Projekt funktioniert hat.





**Abbildung 7.3**

### **7.1.7 KW45: 08.11.2021 bis 14.11.2021**

#### **Arbeit in der Schule**

Da der Gargoyle Charakter nicht funktioniert hat habe ich die Schritte nochmals wiederholt. - import vom JL-Game folgendes: JohnLemon, Gargoyle, FaderCanvas

- export in das eigene Spiel
- Prefab Modelle von JohnLemon und Gargoyle in die Scene eingefügt
- FaderCanvas prefab eingefügt welches folgende Bilder beinhaltet: -ExitImageBackground
- CaughtImageBackground

Der CaughtImage wird angezeigt wenn der Gargoyle Charakter den Haupcharakter in seiner PointOfView hat.

- Das PrefabModell GameEnding wurde auch vom JL-Game importiert. Funktion von GameEnding: wenn der Character den Ziel(GameEnding Cube) erreicht, wird ExitImageBackground angezeigt.
- Ergebnis: Wenn J.L im Blickwinkel vom Gargoyle ist wird CaughtImage angezeigt und wenn J.L das Ziel erreicht wird ExitImage angezeigt.
- Das GameEnding Programm wurde im J.L Kurs in den Sommerferien geschrieben.

Abbildung 7.3

#### **Arbeit außerhalb der Schule**

- Das Programm soll jetzt auch für unsere Charaktere funktionieren.

Unsere Charaktere: Hauptcharakter Schüler: Abbildung7.1

Lehrer Charakter: Abbildung 7.4

- Hauptcharakter Schüler wurde in die Scene eingefügt und die Parameter von Gargoyle für den Hauptcharakter angepasst.



**Abbildung 7.4**

Ergebnis: Gargoyle erkennt den Hauptcharakter.

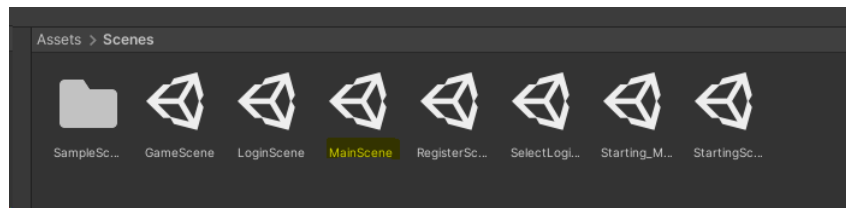
- Charakter Lehrer wurde in die Scene eingefügt.

### **Was ist geplant für die Nächste Woche**

Die Funktion vom Charakter Gargoyle soll dem Charakter Lehrer implementiert werden. Ein dynamischer Gegner soll entwickelt werden, wie z.B. der Charakter Ghost.

### **Welche Schwierigkeiten hat es gegeben**

Die Funktionen von Charakteren richtig Laufen zu bringen.



**Abbildung 7.5**

### **7.1.8 KW46: 15.11.2021 bis 21.11.2021**

#### **Arbeit in der Schule**

Der Ghost Charakter wurde eingefügt, welche ein dynamischer Gegner ist. Der Ghost bewegt sich in einer bestimmten Umgebung und schaut ob er den Charakter in seiner Point of View hat.

#### **Arbeit außerhalb der Schule**

Ich konnte in GIT meine Arbeit nicht speichern, da ein Fehler auftauchte. Ich habe im Internet recherchiert und in einem Beitrag gelesen, dass man auf Git nicht mehr als 2GB Daten hochladen soll, da sonst Probleme auftauchen könnten. Doch unsere Repository war ca. 4Gb groß. Nachdem wir die unwichtigen Dateien gelöscht haben und ich die Repository nochmals geclont habe, hat alles wieder funktioniert.

### 7.1.9 KW47: 22.11.2021 bis 28.11.2021

#### **Arbeit in der Schule**

Die Szenen von meinem Partner habe ich in den Ordner, wo meine Szenen sind hineinkopiert, um sie nacher zu verbinden. Abbildung 7.5. Anschließend haben wir folgende Funktionen programmiert: Wenn im Menü auf spielen gedrückt wird, wird der Main-Szene, also das Spiel gestartet. Wenn das Spiel Game-Over wird, wird das Spiel von neu gestartet

#### **Arbeit außerhalb der Schule**

Meine nächste Aufgabe ist es, die Kamera zu programmieren. Mit der Funktion Cinemachine von Unity kann sehr leicht eine dynamische Kamera programmiert werden. Das Cinemachine Brain verwaltet alle virtuellen Kameras und entscheidet, welcher virtuelle Kamera die eigentliche Kamera folgen soll. In unserem Spiel wird die Kamera unseren Hauptcharakter folgen.

- Das Paket Cinemachine vom PackageManager wurde heruntergeladen und installiert.
- Neue Virtuelle Kamera wurde erstellt
- folgende Parameter wurden im Inspector-Window von der virtuellen Kamera geändert:
  - Composer: Do Nothing
  - Follow: Hauptcharakter
  - Transposer: Framing Transposer
  - Rotation: x-axis to 45
  - Camera distance: 8
- Post Processing Layer erstellt um der Kamera einen Filter zu geben
- Post Process vom Packet Manager heruntergeladen und installiert
- Im Main Camera Inspector Window ein Post Processing Layer erstellt
- Im Post Process Layer component den Mode von No Anti-aliasing zu Fast Approximate Anti-aliasing (FXAA) geändert
- Mit diesem Filter haben wir jetzt eine bessere Qualität.

```
private void CameraRotation()
{
    // if there is an input and camera position is not fixed
    if (_input.look.sqrMagnitude >= _threshold && !LockCameraPosition)
    {
        _cinemachineTargetYaw += _input.look.x * Time.deltaTime * LookSensitivity;
        _cinemachineTargetPitch += _input.look.y * Time.deltaTime * LookSensitivity;
    }
}
```

**Abbildung 7.6**

### 7.1.10 KW48: 29.11.2021 bis 05.12.2021

#### **Arbeit in der Schule**

Antrag für Unity-Spielumgebung an Herrn Rusch geschrieben.

#### **Arbeit außerhalb der Schule**

- Die Umgebung gekauft und heruntergeladen
  - Im packet-manager, die Umgebung in das Projekt importiert
  - SchoolScene geöffnet
  - Meine Charaktere in die Umgebung eingefügt, die jedoch nicht funktioniert haben
  - Ein funktionierendes Charakter gefunden und eingefügt
  - Charakter mit Cinemachine verbunden: Kamera folgt dem Charakter
  - Die Kamera, die man mit der Maus bewegen kann ist zu schnell, deshalb wurde im Code folgende Änderungen vorgenommen um die Sensivität einzustellen.
- Abbildung 7.6 Eine neue Variable LookSensitivity wurde definiert und mit der x y Werte von der Kamera multipliziert. Da die Variable public definiert wurde kann sie jeder Zeit im Inspektor Menü verändert werden.

-

#### **Welche Schwierigkeiten hat es gegeben**



**Abbildung 7.7**

#### **7.1.11 KW49: 06.12.2021 bis 12.12.2021**

##### **Arbeit in der Schule**

- Game-Ending Objekt eingefügt: Wenn Hauptcharakter Game-Ending Objekt erreicht wird das Spiel beendet
- Gegner Gargoyle eingefügt: Wenn Hauptcharakter im PointOfView vom Gargoyle ist, wird Spiel beendet.

##### **Arbeit außerhalb der Schule**

- Da unserer Charakter nicht besonders gut aussieht und in unsere Geschichte nicht hineinpasst, wird stattdessen ein passender Schülercharakter ins Spiel integriert.
- Um dies zu verwirklichen wurde zuerst das fbx Model vom Charakter heruntergeladen.
- Model wurde beim rig auf Humanoid eingestellt und ein neues Avatar wurde erstellt
- funktionierendes Charakter Prefab: Unpack completely

- in Player Armature: Geometry: unser Prefab Model eingefügt
- Sketlon gelöscht
- Avatar von unserem Charakter zugewiesen

**Welche Schwierigkeiten hat es gegeben**

### 7.1.12 ab KW50

#### Arbeit in der Schule

In meiner Git Repository konnte ich den Main Branch nicht finden. Anschließend habe ich folgende Befehle verwendet um das Problem zu lösen Abbildung: 7.8.

Charakter Direktorin erstellt Charakterskin und Animationen von Mixamo.com  
geholt Animation: einfaches idee zum stehen

Meine nächste Aufgabe: Dialog mit Lehrer programmieren: Wenn Schüler in in der Nähe vom Lehrer oder Direktorin ist, soll ein Dialogfenster öffnen, wo das Gespräch läuft. Beim Dialog stellt der Lehrer nach einem kurzen Gespräch fragen an den Schüler. Schüler soll mit Buttons auf die Fragen antworten können. Wenn auf Fragen falsch beantwortet wird, soll sein leben weniger werden.

Funktion: Wenn Schüler im Blickfeld(Triggerzone: Abbildung 7.9) und die Taste F gedrückt wird soll das Dialogsystem das Dialogfenster öffnen.

#### Arbeit außerhalb der Schule

Ich habe Wochenlang versucht, die Aufgabenstellung die ich beschrieben habe zu realisieren. Ich habe viele Programme gefunden, die sehr lange und komplex waren. Zum Schluss habe ich mich entschieden das Programm selber zu schreiben, welches sehr Zeit in Anspruch nahm.

Übersicht Programm:

1. Empty DialogueSystem: hier werden unsere Dialog GUIs sein, die mit dem Script DialogueSystem verwaltet werden. Diese entscheidet welches GUI geöffnet werden soll, indem Sie schaut ob das Gespräch eine Frage beinhaltet
2. GUI DialogueUI: Hier werden die Gespräche im Textbox angezeigt
3. DialogueUI Question Buttons: Dieses GUI wird angezeigt, wenn der Lehrer, dem Schüler eine Frage stellt. Somit kann also der Schüler mittels Buttons auf die Frage beantworten. Im Script DialogSystem werden die Antworten vom Spieler mit den richtigen Antworten verglichen. Wenn die Antwort richtig ist wird der Button grün. Doch wenn die Antwort falsch wird der Button rot und das Leben vom Spieler wird weniger.

Erledigte Aufgaben:

4. Die schon davor programmierte, statische Gegner ins Spiel platziert
5. Dynamischer Gegner programmiert : Abbildung 7.10

Funktion:



```
semih@LAPTOP-45KC37VA MINGW64 /c/HTL_2021-22/Diplomarbeit/DA---MazeRun/.git/refs/heads (GIT_DIR!)
$ rm main

semih@LAPTOP-45KC37VA MINGW64 /c/HTL_2021-22/Diplomarbeit/DA---MazeRun/.git/refs/heads (GIT_DIR!)
$ ls
Future_Probe

semih@LAPTOP-45KC37VA MINGW64 /c/HTL_2021-22/Diplomarbeit/DA---MazeRun/.git/refs/heads (GIT_DIR!)
$ cd ..

semih@LAPTOP-45KC37VA MINGW64 /c/HTL_2021-22/Diplomarbeit/DA---MazeRun/.git/refs (GIT_DIR!)
$ cd ..

semih@LAPTOP-45KC37VA MINGW64 /c/HTL_2021-22/Diplomarbeit/DA---MazeRun/.git (GIT_DIR!)
$ cd ..

semih@LAPTOP-45KC37VA MINGW64 /c/HTL_2021-22/Diplomarbeit/DA---MazeRun (main)
$ git reset

semih@LAPTOP-45KC37VA MINGW64 /c/HTL_2021-22/Diplomarbeit/DA---MazeRun (main)
$
```

**Abbildung 7.8**



**Abbildung 7.9**

Wenn Hauptcharakter im Blickfeld: Leben wird reduziert

Figur bewegt sich zwischen den angegebenen Koordinaten

Programm Übersicht:

NavMesh Agent Component: Figur bewegen

Parameter(Zielkoordinaten als Array, Figur)

Programmstart: Figur geht zur Zielkoordinate[0]

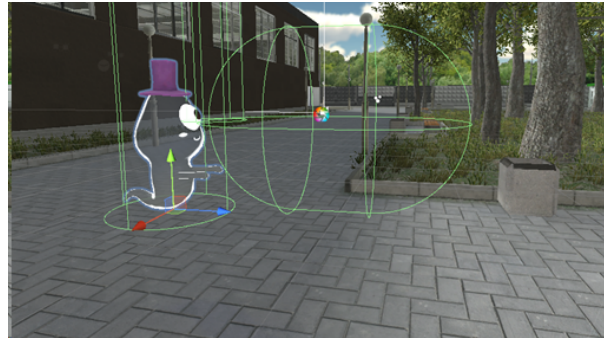
Wenn die Figur ihr Ziel erreicht hat: nächstes Ziel

6. Es wurde ein Healthbar(Lebensskala) programmiert. Wenn der Hauptcharakter in Sicht eines Gegners kommt wird sein Leben weniger.

Veränderungen an statischen und dynamischen Gegnern vorgenommen: Healthbar Funktion eingefügt und Werte angepasst

7. Andere Charakter Lehrpersonen aus der Website Maximo.com geholt und sie entsprechend angepasst. Abbildung: 7.11

8. Semih Can beim Multiplayerprogramm weitergeholfen



**Abbildung 7.10**



**Abbildung 7.11**