Arcade Racing Game

IPT-2.1

Unity

Inhaltsverzeichnis

1. Idee und Grobziele

Meine Idee ist, ein kleines Rennspiel mit Unity zu programmieren. Es werden vielleicht Gegner hinzugefügt, die durch einen Algorithmus versuchen den Spieler zu behindern.

Ziele:

1. Keine Fehler/möglichst wenige oder keine Bugs
2. Neues Lernen
3. Den Code verstehen
4. (Fast) Nichts aus dem Internet kopieren
5. Probleme versuchen selbst zu lösen, wenig im Internet nachschauen
6. Durch Algorithmen die Spielwelt verschönern

2. Planung

1. In Blender werde ich simple Modelle designen.
2. Ich werde eine einfache Steuerung für ein Auto programmieren.
3. *Die Drift-Erkennung des Autos programmieren. (optional)*
4. Als Algorithmus werde ich eine Weltgeneration machen. Es werden zufällig Bäume platziert.
5. Objekte platzieren (Kisten, Mäuse usw.)
6. Steuerung, Grafik usw. optimieren

3. Benutzerhandbuch

<table>
<tr>
<td>Spiel Starten</td>
<td>
1. Spiel Herunterladen (Siehe Kapitel 7)
2. «IPT.exe» ausführen
3. Grafikeinstellungen auswählen
4. Modus auswählen


</td>
</tr>
<tr>
<td>Steuerung</td>
<td>
<table>
<tr><td>W</td><td>Gas</td></tr>
<tr><td>S</td><td>Bremse/Rückwärts</td></tr>
<tr><td>A</td><td>Nach links lenken</td></tr>
<tr><td>D</td><td>Nach rechts lenken</td></tr>
<tr><td>R</td><td>Neustarten</td></tr>
<tr><td>Escape</td><td>Ins Menü</td></tr>
</table>

Die Steuerung ist auch einem Controller möglich;
(PS4 Controller mit Kabel verbinden)

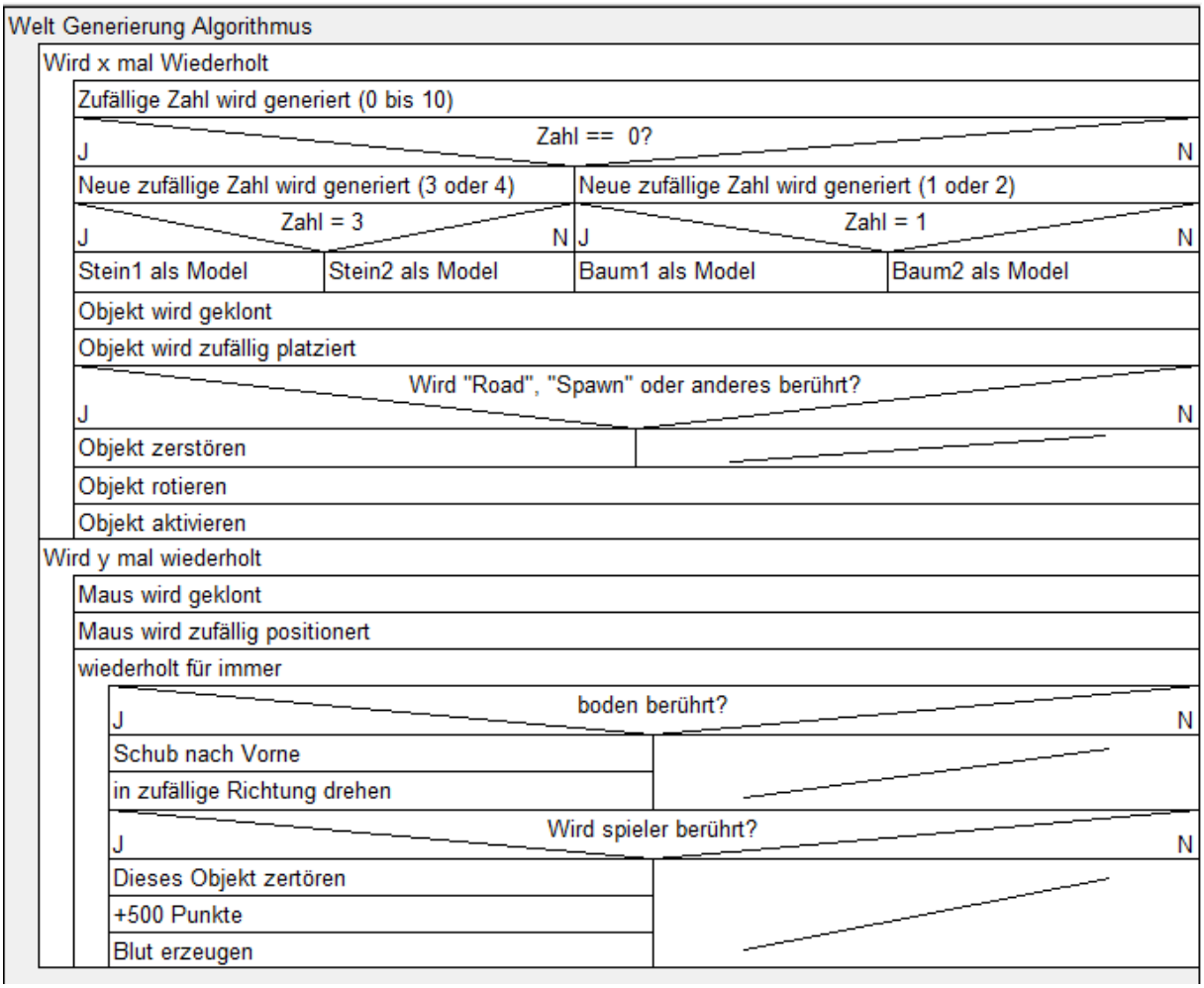

(sollte auch für Xbox Controller funktionieren)
</td>
</tr>
</table>

Punktesystem	Man erhält Driftpunkte, wenn die Differenz zwischen der Rotation des Autos und die Fahrrichtung über 50° ist. Je schneller man fährt, desto mehr Punkte bekommt man beim Driften. Man kann auch Punkte erhalten, wenn man Mäuse überfährt (100 P.). Wenn man das Auto resetten muss, verliert man 500 Punkte.
GUI	Oben links zeigt es bei «Speed» die Geschwindigkeit an. Unter der Geschwindigkeit zeigt es die Punkte und die Mäuse an. Oben rechts hat es eine Minimap.
Ziel	Das Ziel ist, so viele Punkte zu sammeln wie möglich. Es hat kein Zeit-Limit.
Programme/ Webseiten	Folgende Programme/Webseiten habe ich für das Spiel benutzt: Programme: • Blender 3.1.2 • Unity 2020.3.33f1 • Visual Studio 2019 • Paint.Net Webseiten: • stackoverflow • Unity Forum • Unity Documentation

4. Algorithmus

Ich habe einen Algorithmus für die Spielwelt erstellt. Es platziert Bäume und Steine (Zufällige Position und Rotation). Falls es z.B eine Strasse berührt, wird es zerstört.

Es werden auch Mäuse zufällig platziert. Sie bewegen sich zufällig und werden sterben, falls es Kontakt mit dem Auto gibt.



5. Lernjournal

Datum	Beschreibung
Vor Ferien	Ich habe schon eine Idee, mit Unity ein Spiel zu programmieren. Ich habe angefangen, Modelle für das Spiel in Blender zu designen.
In Ferien	In den Ferien habe ich schon mit Unity angefangen. Am Anfang wollte ich einen Shooter programmieren, weil aber der Aufwand viel zu gross wäre, habe ich mich für ein Rennspiel entschieden. Eine einfache Steuerung habe ich in den Ferien programmiert. Ich hatte wenig bis keine Probleme beim Programmieren.
02.05.2022	Ich habe das Skript optimiert. Weil der Boden nur eine Farbe war, konnte man sich nicht orientieren, deshalb habe ich eine Bodentextur hinzugefügt.
09.05.2022	Ich habe das Skript weiterhin optimiert, wie zum Beispiel die Drifterkennung, dass es erkennt, wenn man driftet. Ich habe auch Bäume und eine Rampe designt und hinzugefügt. Die Vorlage für die Dokumentation habe ich erstellt.
16.05.2022	In Unity habe zwei Bäume designt. Ich habe auch einen Algorithmus für die Generation der Bäume erstellt. Es wählt einen Baum-Model aus und platziert es überall, ausser auf den Strassen und gewählten Objekten.
23.05.2022	Ich habe Mäuse designt in Blender. Auf Map können x viele Mäuse erstellt werden. Die Mäuse haben eine Zufällige Bewegung und werden beim Kontakt mit dem Auto zerstört.
30.05.2022	Ich habe neben Bäume auch Steine hinzugefügt. Es hat eine Chance von 1 bis 10, dass es statt einem Baum gewählt wird. Am Schluss habe ich das Spiel habe mit «Post-Processing» verschönert. Ich habe ein Flussdiagramm von der Baum/Welt Generierung gemacht.
7.06.2022	Keine Arbeit -> Frei
14.06.2022	Ich musste ein paar Sachen optimieren, weil es beim Ausführen anders war als in Unity. Ich habe das Spiel auf GitHub hochgeladen und abgegeben.

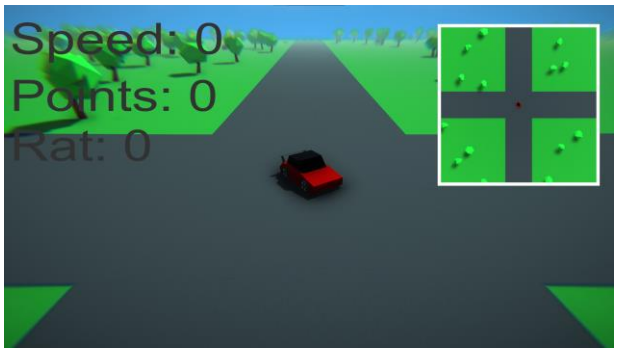
6. Fazit

Das Programmieren war keine Herausforderung für mich, nur das Arbeiten mit Unity auf der VM war manchmal mühsam. Manchmal kam es vor, dass zum Beispiel ein Objekt einen anderen Namen hat als im Code, und das ganze Spiel am Schluss nicht funktionierte. Solche und andere Probleme konnte ich aber beheben.

Anders als am Anfang geplant, kamen die Mäuse in Spiel dazu. Die Mäuse entstanden, als ich in Blender etwas testen wollte.

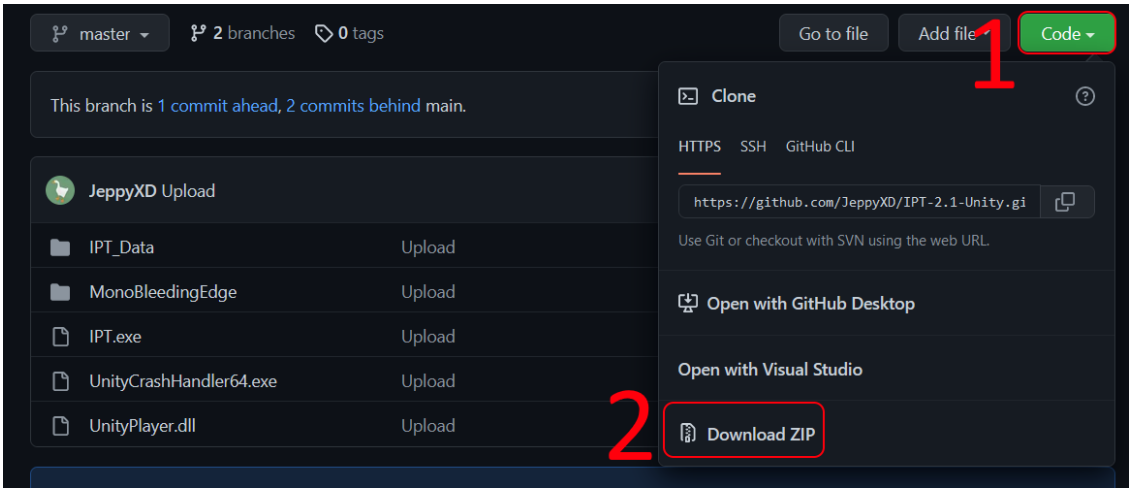
Mit dem Endprodukt bin ich sehr zufrieden. Ich bin stolz darauf, dass ich fast alles im Code selbst programmiert habe, verstehe und auch beschreiben kann, was es tut.

Ich bin mir sicher, dass ich ein bisschen zu viel Zeit in das Projekt investiert habe, aber es hat mir zumindest Spass gemacht.



7. Spiel Herunterladen

https://github.com/JeppyXD/IPT-2.1-Unity/tree/master



Die Dateien als .ZIP herunterladen, entpacken und die «IPT.exe» Datei ausführen.