

Driver Game

Fachhochschule Erfurt

BA5 WS-2020-2021

Alejandro Restrepo Klinge

Benito Grauel

Ahmad Abo Louha

Spielkonzept	3
Installationshinweise & Systemanforderungen	4
Bedienungsanleitung	4
Spielmechanik & Game loop	5
Vorbereitung	5
Spielverlauf	5
Spielende	5
Levels	6
Design & Look	6
Assets	7
Assets Struktur	7
Car Controller	7
Mission Controller	8
Process Controller	8
Externe Assets	9
Projektmanagement	10
Tester Review	10
Noch To-Do & Ideen für Weiterentwicklung	11
Sonstige	12

Spielkonzept

DriverGame ist ein Open-World und Rennspiel, in dem der Spieler ein Auto steuert. Der Spieler kann als Taxifahrer betrachtet werden. Das Ziel des Spieles ist eine Reihe von vordefinierten Missionen zu erfüllen, wobei jede Mission ein anderes Ziel hat.

In jeder Mission soll der Spieler von einem Ort zu einem anderen Ort in der Stadt fahren. Manche Missionen erfordern aber auch, dass der Spieler zwischen dem Startpunkt und Endpunkt auch andere Punkte erreichen soll.

Die Passagiere müssen in einer bestimmten Zeit zum Endpunkt transportiert werden. Der Spieler muss aber darauf aufpassen, dass in manche Missionen Fremde und Hindernisse geben können, die verhindern, dass der Spieler zum Endpunkt rechtzeitig ankommt.

Das Auto hat aber auch eine "Healthbar", welche nach jeder Kollision reduziert wird. Wird die "Healthbar" leer, so verliert der Spieler das Spiel.

Das Spielgefühl nähert sich dem Motto, "Easy to learn, hard to master". In dem Spiel sind auch Geheimnisse versteckt, welche der Spieler erkunden soll. Da die Missionen auch eine Wertung haben, kann der Spieler immer wieder die gleichen Missionen spielen, um bessere Ergebnisse zu bekommen.

Eine sehr große Inspirationsquelle ist das Spiel "Crazy-Taxi". Andere Spielideen sind in Open-World-Spielen sowie Grand-Theft-Auto inspiriert.

Es eignet sich für alle Spielertypen, die nach Spaß und Adventure beim Spielen suchen.

Auch Conqueror kommen auf ihre Kosten, wenn es darum geht die beste Zeit aus einer Mission herauszuholen oder in jeder Mission 3 Sterne zu verdienen.

Installationshinweise & Systemanforderungen

Das Spiel muss nicht installiert werden, es reicht den DriverGame zip Datei zu entpacken und die "*DriverGame.exe*" Datei auszuführen

Die Spielgröße beträgt 150 MB aber andere mind. Anforderungen konnten wir nicht bestimmen, da es keine Möglichkeit gibt, das Spiel auf verschiedenen Systemen und Plattformen zu testen.

Das Spiel ist auf folgenden Rechnern ausgeführt ohne Probleme beim Laden oder Laufen

	Rechner 1	Rechner 2	Rechner 3
Betriebssystem	Windows 10 Education	Windows 10 Home	Windows 10 Home
Prozessor	Intel Core i5-3230M 2.60GHz	Intel Core i5-8250U 1.8GHz	Intel Core i3-6100U 2.30GHz
RAM	8 GB	16 GB	8 GB
Systemtyp	64-Bit	64-Bit	64-Bit
Grafikkarte	AMD Radeon HD 4GB	NVIDIA GeForce MX150	NVIDIA GeForce 940 MX

Bedienungsanleitung

Taste	Funktion
Up-Arrow, 'W'-Taste	Beschleunigen
Down-Arrow, 'S'-Taste	Bremsen
Left und Right Arrows bzw. 'A' und 'D-Tasten'	Lenken
'E'-Taste	Hupe
'Q'-Taste	Lustige Hupe
Leertaste	Handbremse
"ESC"-Taste	Pause

Spielmechanik & Game loop

Vorbereitung

DriverGame ist ein Einzelspieler, wo der Spieler ein Auto kontrolliert und in einer Open-World Stadt fährt, die Stadt hat Hoch- und Einfamilienhäuser dazu kommen Straßen, die verschiedene Größen haben. (enge und breite Straßen sowie eine Autobahn mit einer Brücke).

Der Spieler kontrolliert das Auto durch die vier Pfeiltasten sowie WASD-Tasten. Der Spieler kann auch eine Handbremse dadurch aktivieren, dass die Leertaste gedrückt wird. Außerdem kann er mit den Tasten "e" und "q" hupen.

Wenn das Auto mit einem Haus, einem Zaun oder einem anderen Auto zusammenstößt, wird das "Healthbar" kleiner, bis es leer wird. Ist das "Healthbar" leer, wird das Spiel beendet und der Spieler verliert. Der Spieler kann aber auch während des Spiels durch einer Tankstelle fahren und das "Healthbar" wieder auf 100% setzen.

Spielverlauf

Das Spiel fängt so an, dass der Spieler beliebig durch der Stadt fahren kann, ohne eine bestimmte Mission gestartet zu haben. Wenn er zu einem Mission-Start ankommt, kann er mit Drücken der Taste "e" die Mission starten.

Nach erfolgreiche Beendigung einer Mission bekommt der Spieler eine Bewertung des Missions und kann wieder frei in der Stadt fahren. Nachdem kann der Spieler entscheiden, ob er zum nächsten Mission-Start fährt oder einfach die Stadt erkunden will.

In bestimmten Zeitpunkte wird das Spiel gespeichert. Einmal wenn das Spiel anfängt und nach Beendigung eines Missions.

Während des Spiels kann das Spiel mit "esc" pausiert werden. Von dem Pause-Menü kann man die letzte gespeicherte Zustand laden oder zurück zur Hauptmenü gehen.

Während das Spiel kann der Spieler Münzen sammeln, um ein Geheimnis zu erkunden.

Ein Spieler verliert, wenn entweder eine Mission nicht in der vorgegebene Zeit abgeschlossen wurde oder das Healthbar leer ist. In diesen Fälle kann der Spieler auch die letzte gespeicherte Zustand laden.

Spielende

Das Spiel kommt zur Ende wenn alle Missionen durchgespielt worden sind.

Levels

DriverGame hat bis jetzt drei verschiedene Missionen:

#	name	Feinde	Beschreibung	Story
1	Easy Mission	nein	Spieler fährt vom Startpunkt zum Endpunkt	Der Fahrer soll einen Passagier zum Krankenhaus fahren.
2	Medium Mission	Ja	Spieler fährt durch verschiedene Checkpoints in bestimmter Reihenfolge	Der Fahrer holt den Passagier ab und soll ihn nach Hause durch die Autobahn fahren. In der Autobahn sind Feinde zu finden, die verhindern, dass der Fahrer seine Mission abschließt.
3	Hard Mission	Ja	Spieler hat verschiedene Checkpoints, eine bestimmte Reihenfolge gibt es aber nicht	Der Fahrer soll eine bestimmte Anzahl an Pakete in der Stadt verteilen. Dabei gibt es Feinde.
(4)	Secret Mission	Ja	Ähnlich zu Mission 2. Mit der Unterschied an Musik und Design	Ein Virus hat die Stadt angegriffen. Alle Bewohner sind Zombies geworden, nun muss der Fahrer von der Stadt fliehen.

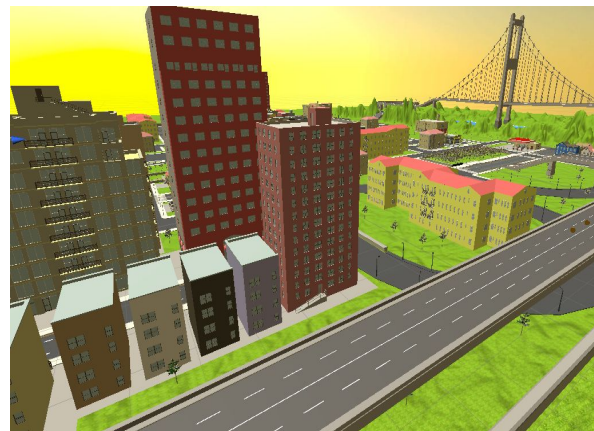
Design & Look

Für das Design der Stadt wurde ein City Pack von Unity Store genutzt, die Gebäude, Straßen, Zäune, Lampen und Straßenschilder hat.

Die Gebäude haben verschiedene Arten wie hoch- oder Einfamilienhäuser, die Straßen haben keine Kurven, deswegen wurde für ein schönes Fahrerlebnis eine andere Assets Pack benutzt, die Kurven, Autobahnen sowie Brücken hat.

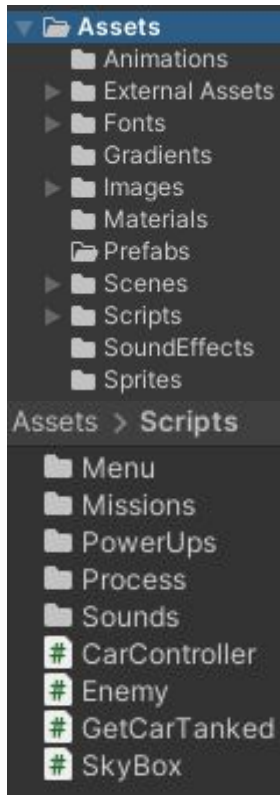
Außerdem wurde für die Autos ein "Stylized Vehicles Pack" importiert, die verschiedene Autos wie Jeep, Bus und Sport Auto hat.

Für jede Mission gibt eine andere Skybox mit andere Lichteinstellungen so, dass eine Wechselung von Tag bis mitternacht mit Nebel in manche Missionen gibt.



Assets

Assets Struktur



Es wurde versucht, eine eindeutige Struktur so halten und eine Namensgebung so zu wählen, dass es immer Eindeutig wird, welcher Element wohin gehört.

Im folgenden Absatz werden die "External Assets" erläutert.

Das war besonders wichtig für das Scripts-Unterordner.

Die drei Wichtige Skripte in unserem Projekt sind:

- Car Controller
- Mission Controller
- Process Controller

Car Controller

Zuständig für alle Elemente vom Auto. Die Bewegung des Auto ist mit einem Kugel verbunden:



```
//Move our object to the position of the sphere
transform.position = sphereRigidBody.transform.position;
```

Mission Controller

Kümmert sich um den Ablauf des Spieles. Besitzt eine Liste von "SingleMissions", welche zu verschiedene Kategorien gehören können. Jede Art von Mission hat einen bestimmten Ablauf, welcher in der "MissionType" festgestellt wurde.

```
//Enum for the definition of the mission type
6 references
public enum MissionType
{
    simpleMission, //From point A to point B
    missionArrayOfObjectivesOrdered, //From point A to point B through a list of points
    missionArrayOfObjectivesDissordered
}
```

An der Seite der Mission Controller wird das ganze mit einem Switch-Case gesteuert.

```
//Define start procedure for each mission type
switch (currentMission.missionType)
{
    //Mission is to go from point A to point B
    case SingleMission.MissionType.simpleMission:
        //Go through a list of objects in a given order
    case SingleMission.MissionType.missionArrayOfObjectivesOrdered:
        //Go through a list of objects in any order (Hit enemies for example)
    case SingleMission.MissionType.missionArrayOfObjectivesDissordered:
```

Process Controller

Zuständig für das speichern von Zustände und das Laden von gespeicherte Zustände und das Aktivierung und steuern der noch zu sammelne Münzen.

```
//Function to save the current progress
1 reference
public void SaveGame(Vector3 position, Quaternion rotation,

//Function to load the last saved game
1 reference
public void LoadGame()...

2 references
public void ActiveNotCollectedCoins()...
```


Externe Assets

Für manche Funktionen wurden externe Bibliotheken benutzt.

Package	Beschreibung
POLYGON City Pack	Assets für die City (Häuser, Straßen...).
Stylized Vehicles Pack Free	Verschiedene Autos einschließlich das Hauptauto.
ToonyTinyPeople	Menschliche Figuren (Enemies)
BrokenVector	Straßen Assets (Brücken, Kurven und Autobahn)
Low Poly Water	See
EffectTexturesAndPrefabs	Particle Effects
Day-Night Skyboxes	Materials für die Skybox
Ground textures pack	Materialien für die Boden (Gras)
GasPumpMiniMap	Tankstelle Minimap-Symbole
Clock Obj.	PowerUp Zeiterhöhen
NavMesh Components	Feinde Mesh
Universal Accessories	Power Ups und Auto Teile
Cron_audio_8_bit_loops_free	Hintergrund Musik
Electric Sfx	Sound effects
TextMesh Pro	Fonts und UI design
Low Poly Survival Pack	Gegenstände für Camp
Tents	Zelte für Camp

Projektmanagement

Die Entwicklung war nah an einer agilen vorgehensweise. Das ganze Team hat (fast) jeder Woche sich gesammelt. In diesen Meetings wurden Themen über den Stand des Spiels diskutiert, sowie Aufgaben verteilt, die bis nächstes Meeting erledigt sein sollten.

Die Arbeitsaufteilung war so, dass jedes Mitglied an alle Themen mitgemacht hat.

Wir hatten einen Plan mit Meilensteinen, den wir zwar manchmal anpassen mussten, welcher aber an sich gut geklappt hat. Wir haben uns dabei eher aufwendige Teile auf verschiedene Wochen gelegt und nebenbei nicht so schwere Sachen gemacht. Zum Beispiel gab es eine Woche indem wir uns eher mit dem MissionController und den Missionen an sich beschäftigt haben und eine Woche danach haben wir uns die Enemies vorgenommen (beide Teile haben einen hohen Aufwand). Nebenbei haben wir uns dann mit dem Design und z.B. der Gestaltung der PowerUps oder der Gestaltung der Partikel auseinandergesetzt.

Tester Review

Tester #1:

Ich hätte mich auch gewünscht, dass die Schwierigkeit des Spiels ausgewählt werden kann. Im allgemein finde ich, dass die Idee und Grafik des Spiels sehr schön ist. Die Anleitung für das Spiel ist gut, was für als Spielanfänger sehr sinnvoll ist. Ich würde gerne die Möglichkeit haben, das Auto so konfigurieren, wie ich es gerne haben würde (Geschwindigkeit, Wie das Auto Lenkt, die Farbe des Autos usw.)

Tester #2:

Das Spiel fühlt sich gut an und sieht auch cool aus. Was man ändern könnte ist der Wegpunkt der zur nächsten Mission zeigt. Wenn er außerhalb des Sichtbereiches ist, weiß man nicht wo man lang fahren muss. Man könnte ihn vielleicht an den Bildschirmrand immer machen, wenn er außerhalb wäre.

Tester #3:

Sehr gute Idee mit dem Spiel! Die Level sind cool und machen Spaß. Ich würde mir nur ein bisschen mehr zum Konfigurieren wünschen, wie z.B. ein neues Auto mit anderer Geschwindigkeit und vielleicht auch eine größere Stadt.

Noch To-Do & Ideen für Weiterentwicklung

Autos Raucheffect dynamisch mit der Geschwindigkeit so zu ändern, dass mehr Rauch kommt wenn das Auto schneller fährt oder aufsteigt.

Es ist auch gewünscht zusätzliche Missionen mit verschiedene Komplexitäten zu implementieren sowie die Menschliche Figuren ins Spiel zu bringen und zu animieren.

Außerdem wäre es besser, wenn manche Elemente des Spiels künstliche Intelligenz hätten. Zum Beispiel:

- Enemies mit mehr Freiheit haben in der Stadt sich zu bewegen
- Leben in die Stadt zu bringen (Stadtbewohner und Autos, die auf die Straßen fahren usw.)

Es war auch geplant, dass wenn der Spieler nicht in eine Mission sich befindet, andere Beschäftigungen in der Stadt zur Verfügung stehen:

- Achievements sammeln: So schnell fahren wie möglich durch ein checkpoint, eine bestimmte Zeit sich in der Luft zu befinden usw.
- Optionale Missions

Man könnte die Stadt noch erweitern und mehr Gebiete sowie Stadtteile und Parks hinzufügen. Der Map sind keine Grenzen gesetzt, wenn es um die Größe geht.

Das CarController könnte mehr entwickelt werden.

Sonstige

Die Inspiration für das Projekt war das “Crazy Taxi” Spiel, welches wir früher gespielt haben und uns viel Spaß beim Spielen gebracht hat.



Wir haben uns überlegt, wie wir ähnliche Spiel implementieren können und die erste Vorstellung des Spiels war:

