

# **Entwicklung eines Spiels mit Unity**

Niels Niederwieser (4g)  
betreut von Lukas Fässler

Maturitätsarbeit  
MNG Rämibühl Zürich  
31.12.2019

# Inhaltsverzeichnis

Abstract.....	2
Vorwort.....	2
Einleitung.....	3
Begriffs-Klärung.....	3
Spiel-Engine.....	3
Unity.....	3
3D-Models:.....	3
UV-sphere:.....	4
.....	4
Icosphere:.....	4
Vorgehen und Planung.....	5
Programmieren und Entwicklung des Spieles.....	5
Ein simples Beispiel um Unity zu benutzen.....	5
3D-Modelle.....	5
Schuss.....	5
Planeten.....	5
Raumschiff.....	5
Asteroid.....	6
Skripte für Bewegungen und Verhalten.....	6
KeyManager.....	6
movement.....	6
variables.....	6
MainKamera.....	7
Asteroidenfeld.....	7
Arbeitsbericht.....	7
Dokumentation.....	7
Ergebnisse.....	9
Reflexion.....	9
Literaturverzeichnis.....	10
Internetquellen.....	10
Anhang.....	11
Eigenständigkeitserklärung.....	12

## Abstract

*Fragestellung, wichtigste Methoden, Ergebnisse und Schlussfolgerungen in Kürze (100-200 Wörter)*  
*Abstract: Ziele, Methoden, Ergebnis und kritischer Rückblick in wenigen Sätzen. (100 – 200 Wörter,*

## Vorwort

Als ich mir die Frage stellte, was ich für ein Thema für meine Maturarbeit wählen sollte, war mir sofort klar dass ich eine Informatik Arbeit machen will. Programmieren hat mir seit langer

Zeit gefallen, wie zum Beispiel mit Scratch (eine visuell basierte Programmiersprache) oder im Informatik Unterricht mit Java. Ich hatte mehrere Maturarbeitsvorträgen zugehört, welche Unity benutzten, ebenso hatte ich mir kürzlich YouTube Videos über Unity angesehen. Ausserdem gehören Computerspiele zu meinen Hobbys unter anderem auch das Verstehen deren Mechanik, weshalb ich mich schlussendlich entschieden habe, ein Spiel mit Unity zu programmieren.

Dankesworte gehen an meinen Vater Marc Niederwieser und an meinen Betreuer Lukas Fässler. Zusätzlich inspirierten mich viele YouTuber insbesondere Brackeys und Blackthornprod.

*Dinge, die nicht in die Arbeit selbst gehören, z.B. Beweggründe, die zur Wahl des Themas geführt haben oder Dankesworte an jemanden. Die unterstützenden Personen und Institutionen werden genannt.*

## Einleitung

Das Ziel meiner Arbeit war ein interessantes Spiel mit Unity zu programmieren. Ich hoffte, dass ich mit einem Arbeitsaufwand von ca. 4h pro Woche für die Spielentwicklung, ein einigermaßen ansprechendes Spiel erstellen könnte. Als Umgebung habe ich den Weltraum gewählt, weil der Weltraum relativ leer ist und nicht viele Objekte dargestellt werden müssen. Dabei habe ich mich für ein Singleplayer (Einzelspieler) Spiel entschieden.

## Begriffs-Klärung

### Spiel-Engine

Ein Spiel kann man von Grund auf programmieren oder man benutzt eine Spiel-Engine. Eine Spiel-Engine ist ein Tool (Programmierzuszeug), welches das Programmieren erleichtert.

Um dies zu verstehen, muss man als erstes wissen, wie Computerspiele aufgebaut sind.

Meistens bestehen Spiele aus mehreren Komponenten wie zum Beispiel einen Teil, der alles Visuelle zeichnet, einen, der Physikberechnungen wie Kollisionen macht und einen weiteren, durch den spezifische Spielmechaniken definiert werden. Eine Spiel-Engine kümmert sich vor allem um den visuellen Teil und im Fall von Unity auch um Physikberechnungen.

### Unity

Unity ist eine populäre Spiel-Engine, welche sehr benutzerfreundlich ist und für nicht kommerzielle Zwecke auch kostenlos ist. Ausserdem kann man es für 2D und auch für 3D Spiele benutzen. Für meine Arbeit habe ich jeweils die aktuellste gratis Version benutzt.

### 3D-Models:

Ein 3D-Model ist ein Objekt, welches eine 3D Figur definiert. Die am häufigsten genutzte Variante ist durch Vertices (Ecken), Edges (Kanten) und Faces (Flächen) definiert.

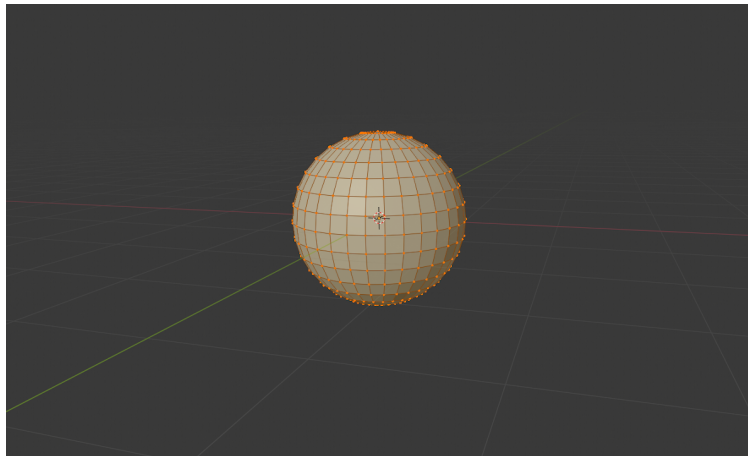
Dabei ist ein Vertex ein Punkt in einem 3 Dimensionalen Koordinatensystem, eine Edge eine

Verbindung von 2 Vertices und eine Face mehrere Edges (meist drei). Ein solches 3D-Model ist leicht mit dem gratis Programm Blender zu machen.

### UV-sphere:

Eine UV-sphere ist eine Art eines 3D-Models einer Kugel. Diese gleicht dem Koordinatensystem der Erde. Dabei wird eine Kugel aus horizontalen und vertikalen Kreisen erstellt (siehe Bild).

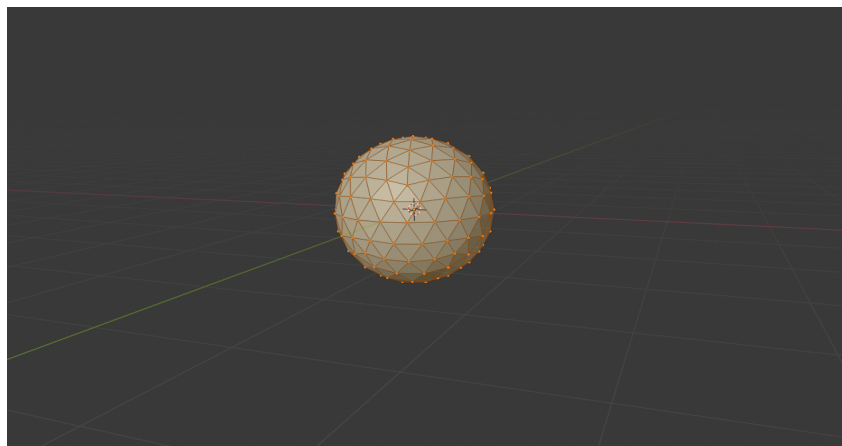
Eine UV-sphere ist sehr praktisch um einer Kugel eine Textur zu geben, da man die Geometrie leicht zu einem Rechteck auffalten kann. Eine rechteckige Weltkarte ist somit die Textur der Erde.



### Icosphere:

Eine Icosphere ist eine andere Art eines 3D-Models einer Kugel. Dabei besteht das Model aus gleichgrossen Dreiecken (siehe Bild).

Der Vorteil gegenüber der UV-sphere ist, dass die Dichte der Geometrie an jeder Stelle gleich ist, aber sie wirkt weniger rund.



*Bedeutung, Zielsetzungen, Fragestellungen und Hypothesen darlegen: Warum und wozu macht man diese Untersuchung? Worum geht es, was möchte man herausfinden? An welche Arbeiten mit gleichem oder ähnlichem Thema lehnt man sich an, wie ist der aktuelle Wissensstand?*

*Einleitung:*

*Hier wird die Projektidee beschrieben und die Problemstellung erläutert. Wichtig ist auch die Eingrenzung des Projekts.*

## **Vorgehen und Planung**

Ich wusste nicht viel über Unity, weshalb ich als erstes eines der eingebauten Tutorials durcharbeitete. Leider waren damals diese Tutorials ziemlich oberflächlich, darum habe ich auf YouTube nach einem Einsteiger Tutorial gesucht. Schlussendlich bin ich auf ein Tutorial von YouTuber Brackeys gestossen<sup>1</sup>. Es ist sehr anfangergefreundlich und erklärt gleichzeitig viele nützliche Dinge. Es dauert insgesamt 100 Minuten und um es zu absolvieren und zu verstehen habe ich ungefähr vier Stunden gebraucht. Danach habe ich mit der Programmierung meines Spieles gestartet. Ich wusste bereits, dass es im Weltraum handeln soll mit einem Raumschiff aber viel mehr war mir noch nicht klar.

Ich ging ohne genauer Planung an diese Arbeit heran, sondern fing ganz begeistert an zu programmieren und versuchte meine Ideen kreativ zu verwirklichen. Manche Ideen konnte ich jedoch nicht verwirklichen, weil sie zu schwierig waren, oder weil ich nicht allen Ideen nachgehen konnte.

## **Programmieren und Entwicklung des Spieles**

### **Ein simples Beispiel um Unity zu benutzen**

#### **3D-Modelle**

#### **Schuss**

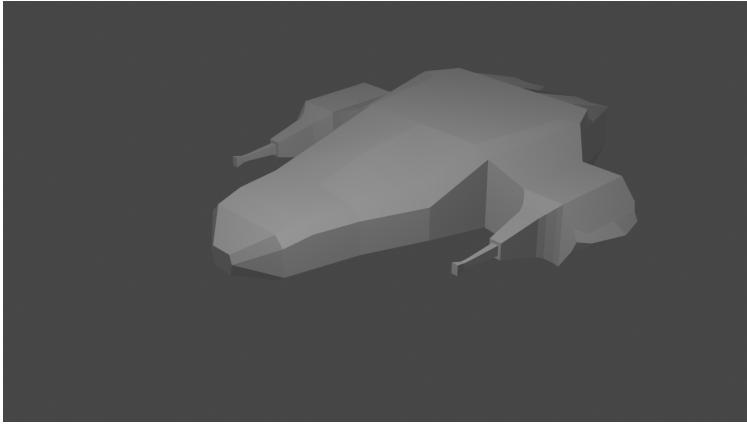
#### **Planeten**

#### **Raumschiff**

In meinem Spiel wollte ich mit einem Raumschiff herumfliegen können. Dafür brauchte ich ein 3D-Model von einem Raumschiff.

---

<sup>1</sup> (Brackeys, 2017): *How to make a Video Game in Unity*



## **Asteroid**

Für mein Asteroiden-3D-Model habe ich als Basis eine Icosphere genommen. Um einen länglichen Asteroiden zu machen, habe ich die Icosphere an einer Achse skaliert. Durch verschiedene Werkzeuge im Sculptmodus (Ein Modus in Blender) habe ich die Finale Form modelliert. Ich wollte meinen Asteroid aber auch während des Spiels zerstören können, weshalb ich ihn mit einem Addon (Cell Fracture) für Blender in kleine Teile geteilt habe. Um das ganze und das zerstörte Asteroiden-Modell nun zu Unity zu exportieren, musste ich darauf achten, dass ich die zerstörte Version mit allen Teilen exportiere.

## **Skripte für Bewegungen und Verhalten**

### **KeyManager**

in diesem Skript sind alle Tastatur und Maus Eingaben gesteuert. Somit kann ich, wenn ich eine Taste ändern will dies sofort in diesem Skript machen

### **movement**

Dieses Skript ist verantwortlich für die Bewegung des Raumschiffes

### **variables**

Mit diesem Skript werden verschiedene Attribute des Raumschiffes festgelegt.

## MainKamera

Dieses Skript kontrolliert die Kamera, welche dem Spieler mit einer schwachen Verzögerung verfolgt.

## Asteroidenfeld

Ich wollte ein Asteroidenfeld machen und musste mir zuerst überlegen wie. Meine Idee war es eine Normalverteilung auf allen drei Dimensionen zu machen und damit entscheiden an welchem Ort ein Asteroid erzeugt werden soll. Durch langes schuften habe ich eine Formel der Normalverteilung programmieren können, aber ich bemerkte erst spät dass mir die Formel nicht viel bringt. Ich konnte zwar die Wahrscheinlichkeit an einem Punkt berechnen, aber habe keine Zufallszahl welche Normalverteilt ist. Irgendwann bin ich auf eine nützliche Website<sup>2</sup> gestossen die mir die Polar-Methode von George Marsaglia erklärte. Nachdem ich diese Methode implementiert hatte, machte ich die Rotation zufällig und die Skalierung auch mit einer Normalverteilung.

## Arbeitsbericht

### Dokumentation

Datum	Zeitaufwand [Minuten]	Tätigkeit
2019-03-16	240	Brackeys Tutorial
2019-04-01	180	3D-Model Raumschiff, Bewegungs-Controller vom Raumschiff, Schuss Skript[während 3 Wochen]
2019-04-10	180	Verzögerte Kamera folgen des Spielers
2019-04-26	30	Schuss Abgabe Klonen (Instantiate)
2019-04-26	60	Einführung und Speicherung des Projektes auf GitHub mit Vater
2019-04-29	60	Schuss an zwei Orten des Raumschiffes erzeugen
2019-05-02	30	3D Model Space Station
2019-05-02	15	Schuss nur nach einer Zeit abfeuerbar
2019-05-03	60	kurz anhaltender Boost
2019-05-10	60	Problem 1 versucht durch Google zu lösen
2019-05-12	60	Problem 1 vereinfacht probiert zu lösen
2019-05-12	30	mit Slomo Kamera versucht zu verstehen

---

<sup>2</sup>Zucconi, Alan (2015) How to generate Gaussian distributed numbers

2019-05-27	30	Problem 1 bei Unity Forum beschrieben
2019-06-14	30	Problem 1 gelöst und Forum Eintrag aktualisiert
2019-07-04	60	Spielplanung auf Blattpapier
2019-07-16	30	Lebenszeit einer Kugel
2019-07-24	30	Variablen des Spielers
2019-07-26	60	Menü angefangen
2019-07-27	60	Ostions Menü angefangen
2019-07-28	15	Raumschiffvariablen Energie/Max
2019-07-28	15	variablen public -> [SerializeField] private
2019-07-31	45	Regeneration von Energie + Energie zu Schild
2019-08-04	15	Scrollwheel Input
2019-08-04	60	Scrollwheel kontrollierbare Geschwindigkeit des Schiffes
2019-08-05	30	Tasteneinput neu geordnet
2019-08-06	15	Elementen Materials gegeben (=> Farbe)
2019-08-07	60	Kontrollierbare Geschwindigkeit optimiert
2019-08-08	60	Abzug von Schiffsenergie bei Beschleunigung/ Schildenergie Erhöhung/ Schuss
2019-08-10	60	Lebens- usw. Leiste angefangen
2019-08-11	60	Leiste fertig gemacht
2019-08-12	90	Asteroid 3D Model gemacht
2019-08-12	45	Asteroid zerstörte teile mit Blender Addon Cell Fracture
2019-08-13	45	Asteroid + zerstörte Version zu Unity importieren
2019-8-28	30	Asteroid leben
2019-9-01	30	Asteroid zerstören
2019-09-26	30	Gausssche Formel gesucht
2019-09-26	30	Gaussche Formel Bedeutungen wie „mean“ versucht zu lernen
2019-09-26	30	Gaussche Formel implementiert
2019-09-27	15	bemerkten dass Gauss Formel nicht normalverteilte Zufallszahl gibt
2019-09-27	60	Gaussche Zufallszahl recherchiert
2019-09-27	30	normal verteilte zahlen berechnen mit der Polar-Methode von George Marsaglia implementiert
2019-09-27	30	asteroidenspawner programmiert (noch verbesserungswürdig)
2019-10-08	60	Asteroidengröße zufällig gemacht.
2019-??-??	420	Versuchung Spaceship controller zu verbessern
2019-11-24	60	Mond in Blender gemacht
2019-11-30	60	probiert den Mond in Unity zu implementieren
2019-11-30	60	Masterarbeit schreiben teil Struktur



2019-12-01	60	probiert ein cube mit Texturen von Blender zu Unity exportieren
2019-12-01	30	Vorwort Maturarbeit
2019-12-02	30	probiert ein Material von Blender zu Unity exportieren
2019-12-02	30	Einleitung 3D-Models + Asteroidmodel Maturarbeit
2019-12-04	60	Geschützturm der den spiele treffen soll angefangen
2019-12-04	30	Einleitung UV-sphere + Icosphere Maturarbeit
2019-12-05	30	Geschützturm algorithmus um Spieler zu treffen

#### *Material und Methoden*

*Hier geht es darum, dass die Methoden so dargestellt werden, dass sie von anderen problemlos nachgeprüft werden können (Reproduzierbarkeit): Vorgehen und Anordnung der einzelnen Versuche genau beschreiben, verwendete Instrumente, Mittel und Arbeitsmethoden genau angeben, Orts- und Zeitangaben nicht vergessen. Auch die Art und Weise der Auswertung bzw. der Darstellung der Ergebnisse muss man genau beschreiben*

## **Ergebnisse**

Das Spiel Silent Space kann über folgenden Link ausprobiert werden:

Einige Personen, z.B. meine Brüder und meine Eltern konnten das Spiel schon ausprobieren und es hat ihnen gefallen.

*Möglichst klare und übersichtliche Darstellung der Ergebnisse, unterstützt durch Grafiken und Tabellen. Dazu gehören auch Erläuterungen zu den wichtigsten Ergebnissen.*

#### *Diskussion*

*Hier geht es um die Interpretation der Resultate, und zwar sowohl der eigenen als auch im Vergleich zu anderen Ergebnissen aus anderen Arbeiten. Mögliche Fehlerquellen werden beurteilt, Schlussfolgerungen werden gezogen. Dazu kommt eine Reflexion über die eigene Arbeit.*

## **Reflexion**

Eigentlich wollte ich noch andere Ideen verfolgen, wie z.B. eine Raumstation, oder das Movement des Raumschiffes zu verbessern, doch klappte nicht alles. Wenn ich jetzt zurück denke würde ich natürlich vieles anders machen, aber das liegt vor allem daran, dass ich viel mehr Erfahrung habe.

*Dieser Teil der Arbeit enthält die reflektierende Auseinandersetzung mit der eigenen Arbeit. Er wird in einzelne Kapitel und Unterkapitel eingeteilt. Hier kann auch die Dokumentation des Arbeitsprozesses erfolgen. In welcher Form die Dokumentation konkret erfolgen wird (separates Skizzenbuch, Arbeitsmappe etc.), ist zu Beginn des Prozesses mit der Betreuungsperson zu vereinbaren.*

# Literaturverzeichnis

- Landolt, Robin (2019): «Unity» als Spiel-Engine für ein zug- basiertes Role Playing Game. Maturitätsarbeit MNG Rämibühl, Zürich.

## Internetquellen

- Brackeys (2017): How to make a Video Game in Unity: <https://www.youtube.com/playlist?list=PLZiMaaLtfkR7zz9qebmvGMNHAbZkUKYpR>
- Brackeys (2017): SHATTER / DESTRUCTION in Unity: <https://www.youtube.com/watch?v=EgNV0PWVaS8&>
- BurgZerg Arcade (2016): Smooth FollowCam: <https://www.youtube.com/watch?v=pNPuMMR5cSk>
- Code Monkey (2018): How to make a Health Bar: <https://www.youtube.com/watch?v=Gtw7VyuMdDc>
- CGMatter (2019): Create A Realistic Moon: [https://www.youtube.com/watch?v=UAOy42TR\\_Rk](https://www.youtube.com/watch?v=UAOy42TR_Rk)
- Zucconi, Alan (2015): How to generate Gaussian distributed numbers: <https://www.alanzucconi.com/2015/09/16/how-to-sample-from-a-gaussian-distribution/>

*Die in der Arbeit erwähnten Quellen (Texte und Abbildungen) werden alphabetisch nach Autoren aufgelistet. Für Internetquellen wird eine gesonderte Liste erstellt.*

# Anhang

Auf dem beigelegten USB-Stick sind alle wichtigen Daten aufgeführt.

Inhalt des USB-Sticks:

-README_(Anleitung).txt	[Anleitung zu den Files]
-Silent_Space.exe	[Spiel]
-Skripts.odt	[Alle Skripts]
-SilentSpace	[Projektordner (kann man mit Unity öffnen)]

{noch kein Stick}

*Hier erscheinen – wenn sinnvoll  
–Rohdaten und Beilagen.*

# Eigenständigkeitserklärung

Der Unterzeichnende bestätigt mit Unterschrift, dass die Arbeit selbständig verfasst und in schriftliche Form gebracht worden ist. Die Mitwirkung anderer Personen beschränkt sich auf Beratung und Korrekturlesen. Alle verwendeten Unterlagen und Gewährpersonen sind aufgeführt.

Pfaffhausen, [insert date]

---

*[Der Umfang einer Maturitätsarbeit beträgt ca. 15–25 Seiten, exkl. Titelblatt, Anhang. Für umfangreichere Arbeiten muss das Einverständnis der betreuenden Lehrperson eingeholt werden.]*

*[Der Umfang des Theorie- bzw. Reflexionsteils muss mit der betreuenden Lehrperson abgesprochen werden.]*