|  |  |
| --- | --- |
| Midnight Runner  Das Spiel der Spiele | Unser Spiel  Das Spiel der Spiele. Hier stellen wir Euch vor, wie wir als 4er-Gruppe ein eigenes Spiel programmieren. Der «MID-NIGHT RUNNER»! Wieso dieser Name? Was kann dieses Spiel? Liest es selbst.  Bogdan Krbanjevic, Shahin Afshar, Luca Duttwiler, Gregory Bobst |

**Metainformationen**

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Version** | **Autor** | **Bemerkung** |
| 2020-11-11 | 1.0 | Gregory Bobst | Erste Version |
| 2020-11-18 | 1.1 | Gregory Bobst | Bearbeitete Version |
| 2020-12-02 | 1.2 | Gregory Bobst | Fertige Version |

Inhalt

[Informieren 4](#_Toc57810376)

[Projektauftrag 4](#_Toc57810377)

[Idee 5](#_Toc57810378)

[Beschaffung der Informationen 5](#_Toc57810379)

[Mailverkehr 5](#_Toc57810380)

[Zeitplan 6](#_Toc57810381)

[Planen 8](#_Toc57810382)

[Tätigkeitsliste 8](#_Toc57810383)

[Pflichtenheft 10](#_Toc57810384)

[Anforderungsanalyse 11](#_Toc57810385)

[Testfallspezifikationen 12](#_Toc57810386)

[Entscheiden 15](#_Toc57810387)

[Ausgangslage 15](#_Toc57810388)

[Entscheidungsmatrix 15](#_Toc57810389)

[Fazit der Entscheidung 15](#_Toc57810390)

[Realisieren 16](#_Toc57810391)

[Grafiken in Photoshop erstellen 16](#_Toc57810392)

[Erstellung Parallax Background 18](#_Toc57810393)

[1. Layer 18](#_Toc57810394)

[2. Layer 18](#_Toc57810395)

[3. Layer 19](#_Toc57810396)

[4. Layer 19](#_Toc57810397)

[5. Layer 19](#_Toc57810398)

[Erstellung Player Character 20](#_Toc57810399)

[Animation: Rennen 20](#_Toc57810400)

[Animation: Springen 20](#_Toc57810401)

[Animation: Rutschen 21](#_Toc57810402)

[Extra Animation: Sterben 21](#_Toc57810403)

[Erstellung der Objekte/Gegner unseres Charakters 21](#_Toc57810404)

[Animation: Polizist mit Laser-Schwert 21](#_Toc57810405)

[Animation: Fliegendes Laser-Schwert 22](#_Toc57810406)

[Animation: Sterben des Charakters 22](#_Toc57810407)

[Soundtrack 23](#_Toc57810408)

[Code: Player Controller 24](#_Toc57810409)

[Code: Obstacle 25](#_Toc57810410)

[Code: Background Looper 25](#_Toc57810411)

[Code: Obstacle Spawner 26](#_Toc57810412)

[Kontrolliren 27](#_Toc57810413)

[Testprotokoll 27](#_Toc57810414)

[Auswerten 28](#_Toc57810415)

[Fazit zum Projekt 28](#_Toc57810416)

[Reflexion Luca Duttwiler 28](#_Toc57810417)

[Reflexion Shahin Afshar 29](#_Toc57810418)

[Reflexion Bogdan Krbanjevic 29](#_Toc57810419)

[Reflexion Gregory Bobst 29](#_Toc57810420)

# Informieren

## Projektauftrag

|  |
| --- |
| Projektname: |
| Mid-Night Runner |

|  |
| --- |
| Beschreibung: |
| Wir wollen nun ein Handyspiel in Stil eines Endlessrunner wie Subway Surfer und Temple run etc. programmieren, jedoch mit einem Twist, dem Coronavirus. |

|  |
| --- |
| Module |
| - |

|  |
| --- |
| Zusätzlich benötigtes Material: |
| Unity Game-Engine |

|  |  |  |
| --- | --- | --- |
| Auftraggeber: | Weitere Kontaktpersonen: | Begleitperson BBB: |
| - | - | Sascha Fiechter |

|  |  |
| --- | --- |
| Projektleiter: | Teammitglieder: |
| Bogdan Krbanjevic | Gregory Bobst, Schahin Afshar, Luca Duttwiler |

|  |  |
| --- | --- |
| Termin Projektstart: | Termin Projektende: |
| 21.10.2020 | 25.11.2020  Verlängert auf den 02.12.2020 |

## Idee

Als eine 4er-Gruppe haben wir das Ziel ein Projekt zu erarbeiten, bei welchem wir etwas lernen können. Jeder von uns spielt täglich irgendein Spiel auf dem Smartphone, und nun wollen wir unser eigenes Spiel spielen. Zusammen sind wir auf den Gedanken gekommen ein endless-Run Spiel zu programmieren. Was das genau ist erfährt Ihr in der Dokumentation.

## Beschaffung der Informationen

Die Informationen stammen von verschiedenen Quellen/Medien. Wir werden hauptsächlich mit Informationen aus dem Netz arbeiten, da dort vieles über das reine Programmieren von Spielen, viel zu finden ist. In YouTube werden natürlich auch nach passenden Videos recherchieren.

## Mailverkehr

Wir hatten ein Mail an Herr Fiechter gesendet für den Projektantrag.

Mail:



## Zeitplan

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Arbeitsschritte** | | |  | **21. Oktober** | | | | **28. Oktober** | | | | **4. November** | | | | **11. November** | | | | **18. November** | | | | **25. November** | | | | **2. Dezember** | | | |
|  | V | | N | | V | | N | | V | | N | | V | | N | | V | | N | | V | | N | | V | | N | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Inform.** | 1.0 | Anforderungsanalyse |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.1 | Idee |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.2 | Zeitlimit |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Planen** | 2.0 | Zeitplan erstellen |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.1 | Pflichtenheft |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.2 | Tätigkeitsliste |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.3 | Testspezifikationen |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Entschd.** | 3.0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.1 | Aufgabenverteilung |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Realisieren** | 4.0 | Mit Unitiy vertraut machen |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4.1 | Assets aussuchen |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4.2 | Quellenverzeichnis erstellen |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4.3 | Hintergrund erstellen |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4.4 | Scripts schreiben |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4.5 | Gegner designen |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4.6 | Theme alles aufeinander abstimmen |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4.7 | Musik komponieren |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Kontrollieren** | 5.0 | Testfälle definieren |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5.1 | Features löschen/hinzufügen |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5.2 | Nach Bugs/ungewollten Features suchen |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5.3 | Grundmechaniken testen (touch enemy=death?) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5.4 | Testprotokoll ausfülllen |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Ausw.** | 6.0 | Fazit erstellen |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6.1 | Reflexion durchführen |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# Planen

## Tätigkeitsliste

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Buchstabe | Maßnahme | Verantwortlich | Priorität | Status | Startdatum | Enddatum | Abhängigkeit |
|  | Ziel Nr. 1: Informieren |  |  |  |  |  |  |
| a | Idee | Team | mittel | Offen | 21.10.2020 | 21.10.2020 |  |
| b | Zeitlimit | Team | niedrig | Offen | 21.10.2020 | 21.10.2020 | a |
|  | Ziel Nr. 2: Planen |  |  |  |  |  |  |
| c | Zeitplan erstellen | Duttwiler | mittel | Offen | 21.10.2020 | 21.10.2020 | b |
| d | Pflichtenheft | Afshar | mittel | Offen | 21.10.2020 | 21.10.2020 | a |
| e | Tätigkeitsliste | Afshar | mittel | Offen | 21.10.2020 | 21.10.2020 | a |
| f | Testspezifikationen | Krbanjevic | hoch | Offen | 21.10.2020 | 21.10.2020 | b |
| g | Ziel Nr. 3: Endscheiden |  |  |  |  |  |  |
| h | Aufgabenverteilung | Krbanjevic | mittel | Offen | 21.10.2020 | 21.10.2020 | a,e |
|  | Ziel Nr. 4: Realisieren |  |  |  |  |  |  |
| i | Mit Unitiy vertraut machen | Team | hoch | Offen | 21.10.2020 | 21.10.2020 | a |
| j | Assets aussuchen | Team | mittel | Offen | 28.10.2020 | 28.10.2020 | a |
| k | Quellenverzeichnis erstellen | Afshar | niedrig | Offen | 28.10.2020 | 28.10.2020 | a,i,j |
| l | Hintergrund erstellen | Duttwiler | mittel | Offen | 28.10.2020 | 04.11.2020 | a |
| m | Scripts schreiben |  | hoch | Offen | 04.11.2020 | 11.11.2020 | a,i |
| n | Gegner designen | Duttwiler | mittel | Offen | 28.10.2020 | 11.11.2020 | a |
| o | Theme alles aufeinander abstimmen |  | hoch | Offen | 04.11.2020 | 11.11.2020 | i,m |
| p | Musik komponieren | Krbanjevic | mittel | Offen | 28.10.2020 | 18.11.2020 | a |
|  | Ziel Nr. 5: Kontrollieren |  |  |  |  |  |  |
| q | Testfälle definieren | Krbanjevic | hoch | Offen | 04.11.2020 | 25.11.2020 | m |
| r | Features löschen/hinzufügen |  | hoch | Offen | 04.11.2020 | 25.11.2020 | q,m |
| s | Nach Bugs/ungewollten Features suchen | Duttwiler | hoch | Offen | 04.11.2020 | 25.11.2020 | q |
| t | Grundmechaniken testen (touch enemy=death?) |  | hoch | Offen | 04.11.2020 | 25.11.2020 | q,s |
| u | Testprotokoll ausfülllen | Krbanjevic | hoch | Offen | 11.11.2020 | 02.12.2020 | q,r,s,t |
|  | Ziel Nr. 6: Auswerten |  |  |  |  |  |  |
| v | Fazit erstellen | Afshar | mittel | Offen | 11.11.2020 | 02.12.2020 |  |
| w | Reflexion durchführen | Team | mittel | Offen | 11.11.2020 | 02.12.2020 |  |

## Pflichtenheft

**Ausgangslage**Unsere Vierergruppe, bestehend aus 4 lehrenden aus dem Basislehrjahr, haben als Projekt die Erstellung eines Handyspiels gewählt. In diesem Projekt wollen wir als Team ein Handyspiel erstellen, welches vollfunktionsfähig funktioniert.

**Ist Zustand**Schließlich haben wir bis auf einen Mitschüler wenig Ahnung von Programmieren, geschweigen von der Erstellung eines Handyspieles.

**Soll Zustand**Nach diesem Projekt soll jeder von uns wissen, welche Tätigkeiten unternommen werden müssen, mit welcher Programmiersprache es geschrieben wird und schließlich, wie man programmiert, um ein Spiel selbst zu entwickeln.

**Ziele**Das Ziel ist selbst ein funktionsfähiges Spiel zu entwickeln. Dazu soll am Schluss jeder von uns wissen wie man ein solches Spiel programmiert und unser Wissen dementsprechend erweitern.

**Anforderungen**Die Anforderungen des Projektes sind, dass wir ein selber ein Handyspiel erstellen, welches funktioniert, und erkennen was hinter diesem Steckt.

**Kosten**Die Kosten werden in diesem Projekt nicht vorhanden sein.

**Personen**Die beteiligten Personen an diesem Projekt sind: Gregory Bobst, Bogdan Krbanjevic, Luca Duttwiler, Schahin Afshar

## Anforderungsanalyse

Bevor wir eine arbeitet getätigt haben, haben wir uns überlegt was das Spiel können sollte. Folgend seht Ihr unsere Anforderungsanalyse mit den 10 wichtigsten Punkten was das Spiel können sollte, wenn wir fertig sind.

|  |  |
| --- | --- |
| Anforderungsnummer | Anforderung |
| 1 | Die Animationen werden richtig dargestellt |
| 2 | Der Charakter fällt nicht durch den Boden |
| 3 | Wenn man im Startmenu auf Spielen drückt. Kann man spielen |
| 4 | Der Soundtrack startet, sobald man das Spiel startet |
| 5 | Die Soundeffekte spielen im richtigen Moment |
| 6 | Die Steuerung funktioniert per Touch |
| 7 | Das App Icon wird auf dem Smartphone korrekt dargestellt |
| 8 | Funktionieren Kollisionen mit Hindernissen |
| 9 | Startet das Spiel, wenn man die App anklickt |
| 10 | Die Karte wird per Zufall generiert |

## Testfallspezifikationen

1. Nummer: 1
2. Getestete Anforderung: 1
3. Voraussetzung:
   1. Alle Pixelgrafiken wurden erstellt
   2. Der Code für die Abläufe der Bilder wurde erstellt
4. Eingabe
   1. In eine beliebige Richtung rennen, springen oder sliden
5. Ausgabe
   1. Der Spielcharakter führt die Animation für die jeweilige Bewegung aus.
6. Nummer: 2
7. Getestete Anforderung: 2
8. Voraussetzung:
   1. Der Boden wurde in die Scene platziert
   2. Der Boden besitzt eine Rigidbody Funktion
9. Eingabe
   1. Das Spiel starten
10. Ausgabe
    1. Der Spielcharakter fällt nicht durch den Boden und kann normal Laufen
11. Nummer: 3
12. Getestete Anforderung: 3
13. Voraussetzung:
    1. Das Spiel ist fertig programmiert und auf dem Handy geladen
14. Eingabe
    1. Sobald das Spiel startet soll der Startbildschirm als erstes kommen und dann soll man „Spielen“ auswählen
15. Ausgabe
    1. Das Spiel startet
16. Nummer: 4
17. Getestete Anforderung: 4
18. Voraussetzung:
    1. Der Soundtrack wurde fertiggestellt
    2. Der Soundrack wurde exportiert und in das Spiel eingefügt
19. Eingabe
    1. Spiel starten
20. Ausgabe
    1. Der Soundrack startet automatisch
21. Nummer: 5
22. Getestete Anforderung: 5
23. Voraussetzung:
    1. Die Soundeffekte wurden aufgenommen
24. Eingabe
    1. Spiel spielen
25. Ausgabe
    1. Beim rennen hört man Schritte
26. Nummer: 6
27. Getestete Anforderung: 6
28. Voraussetzung:
    1. Der Code für die Steuerung wurde geschrieben
29. Eingabe
    1. In eine beliebige Richtung rennen, springen oder sliden
30. Ausgabe
    1. Der Spielcharakter führt die Bewegung für die jeweilige Eingabe aus.
31. Nummer: 7
32. Getestete Anforderung: 7
33. Voraussetzung:
    1. Das Spiel wurde auf das Smartphone geladen
34. Eingabe
    1. Das Handy entsperren und auf den Homescreen gehen
35. Ausgabe
    1. Das Spiel wird auf dem Homescreen mit dem richtigen Icon dargestellt
36. Nummer: 8
37. Getestete Anforderung: 8
38. Voraussetzung:
    1. Hindernisse wurden designed
    2. Hindernisse wurden mit einem Rigidbody ausgestattet
    3. Der Spieler stirbt bei Aufprall mit Hindernissen
39. Eingabe
    1. Mit dem Charakter in ein Hindernis rennen
40. Ausgabe
    1. Der Charakter stirbt und man landet wieder im Main-menu
41. Nummer: 9
42. Getestete Anforderung: 9
43. Voraussetzung:
    1. Das Spiel wurde mit dem Icon verknüpft
44. Eingabe
    1. Das Icon anklicken
45. Ausgabe
    1. Das Spiel startet
46. Nummer: 10
47. Getestete Anforderung: 10
48. Voraussetzung:
    1. Es wurden mehrere Layouts für das Spielfeld erstellt
    2. Der Code wurde so geschrieben, dass das Spiel automatisch die nächsten zwei Spielfelder automatisch aus dem Spielfelderpool nimmt
49. Eingabe
    1. Das Spiel lange genug spielen
50. Ausgabe
    1. Das Spiel endet erst wenn man stirbt. Man hat nie zwei Mal hintereinander das gleiche Karten Layout

# Entscheiden

## Ausgangslage

Zu dem Programmieren von einem Handyspiel, wird auch ein Programmierprogramm benötigt. So haben wir eine Entscheidungsmatrix erstellt, um ein geeignetes Programm zu finden. Wir haben uns zwischen 2 Programmen entschieden, welche Visual Studio und Notepad++ heißen.

**Notepad++** ist ein Texteditorprogramm des Entwicklers Don Ho. In Notepad++ kann man Codes schreiben und diese dann in den verschiedenen Programmiersprachen wie auch in verschiedenen Programmen anwenden.

**Visual Studio** ist ein Programm direkt von Microsoft. Auf Visual Studio kann man mit begrenzter Anzahl an Programmiersprachen programmieren. Jedoch hilft das Programm direkt mit und zeigt einem Fehler und auch andere Möglichkeiten an.

## Entscheidungsmatrix



## Fazit der Entscheidung

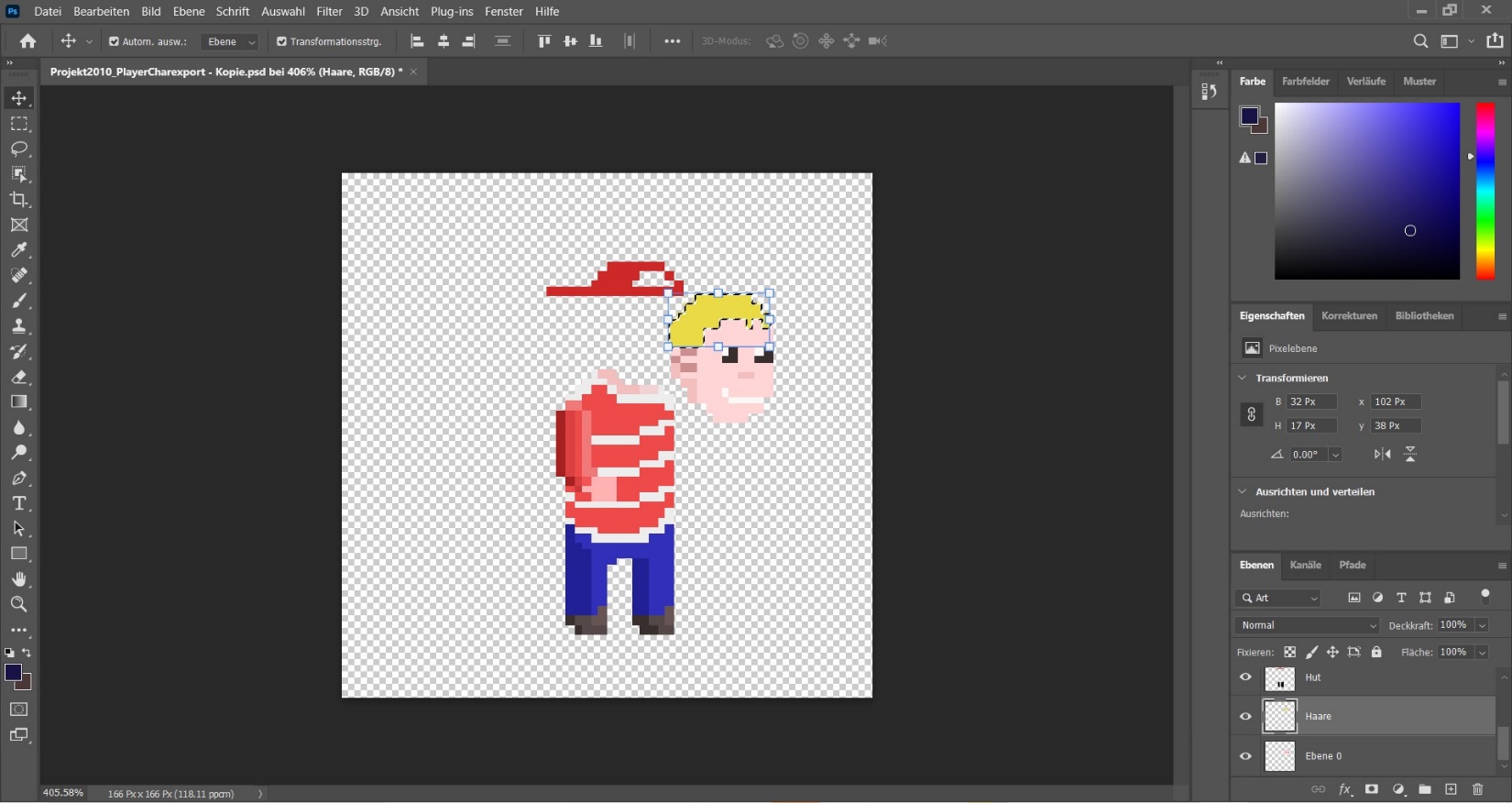
Wir haben die Entscheidungsmatrix nach Auf folgende Kriterien aufgebaut: Schwierigkeit, Verständnis, Aufwendigkeit und Vorteile. Das Visual Studio ist unserer Meinung nach besser für uns geeignet, da wird gewisse Grundkenntnisse besitzen. Dazu bietet Visual Studio eine Korrekturhilfe, was das Programmieren einfacher macht.

# Realisieren

## Grafiken in Photoshop erstellen

Wir haben alle Grafiken selbst in Photo Shop erstellt. Damit ist jeder einzelne Frame , ein Bild einer Charakters, wie auch jeder einzelne Layer, die Hintergrundelemente, gemeint.

Während der Bearbeitung sieht das ganez wie folgt aus:



Auf der linken Seite Rot markiert befinden sich alle Werkzeuge.  
Wir brauchten hauptsächlich wir das Pfadauswahl-Werkzeug um ganze teile zu markieren (So wie die Haare im Bild), das Pinsel-Werkzeug um die gewünschten Teile einzufärben, das Verschiebungs-Werkzeug um ganze Teile zu verschieben und natürlich auch den Radiergummi um die Fehler zu beheben.

Auf der rechten Seite Oben Blau markiert befindet sich die Farbpalette.  
Dort kann man nach dem RGB-Farbmodell alles Farben auswählen.

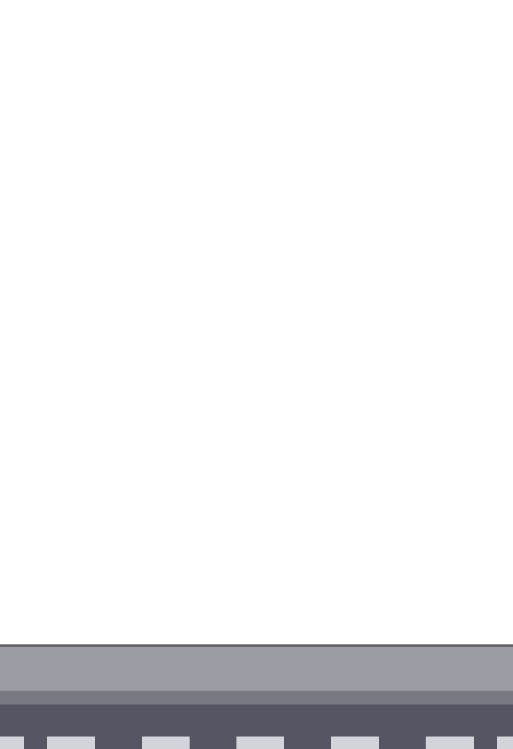
Auf der rechten Seite in der Mitte Grün mariert ist die Transformation.  
Dort kann man die Position wie auch die Winkel der einzelnen Teile transformieren/bestimmen.

Auf der rechten Seite unten Lila markiert sind die einzelnen Teile zu finden. Nach dem man mit dem Pfadauswahl-Werkzeug ein Teil bestummen hat kann man dieses bennenen und dann hier auswählen und sie dann nach belieben einstellen mit den oben genannten Teile.

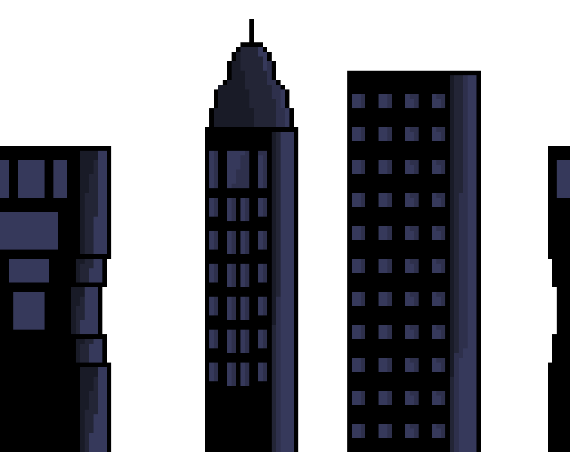
## Erstellung Parallax Background

Für unser Spiel mussten wir einen Parallax Background erstellen. Dies ist nichts weiter als ein Hintergrund, der sich fortlaufend bewegt. Damit wir diesen Hintergrund erhalten haben, mussten wir den ganzen Hintergrund in 5 Teile unterteilen. Jeder dieser Teile nennt man «Layer». Also besteht unser Hintergrund aus den folgenden «Layers»:

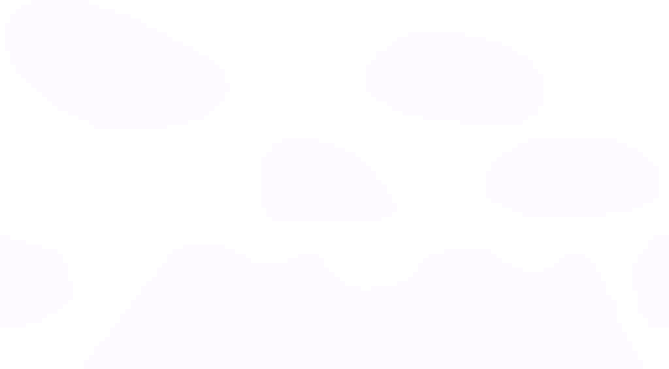
### 1. Layer

Als erstes kommt die Strasse. Wir haben einen Untergrund kreieren wollen auf welchem der Player rennen kann. Dieser Layer bewegt sich am schnellsten von allen.

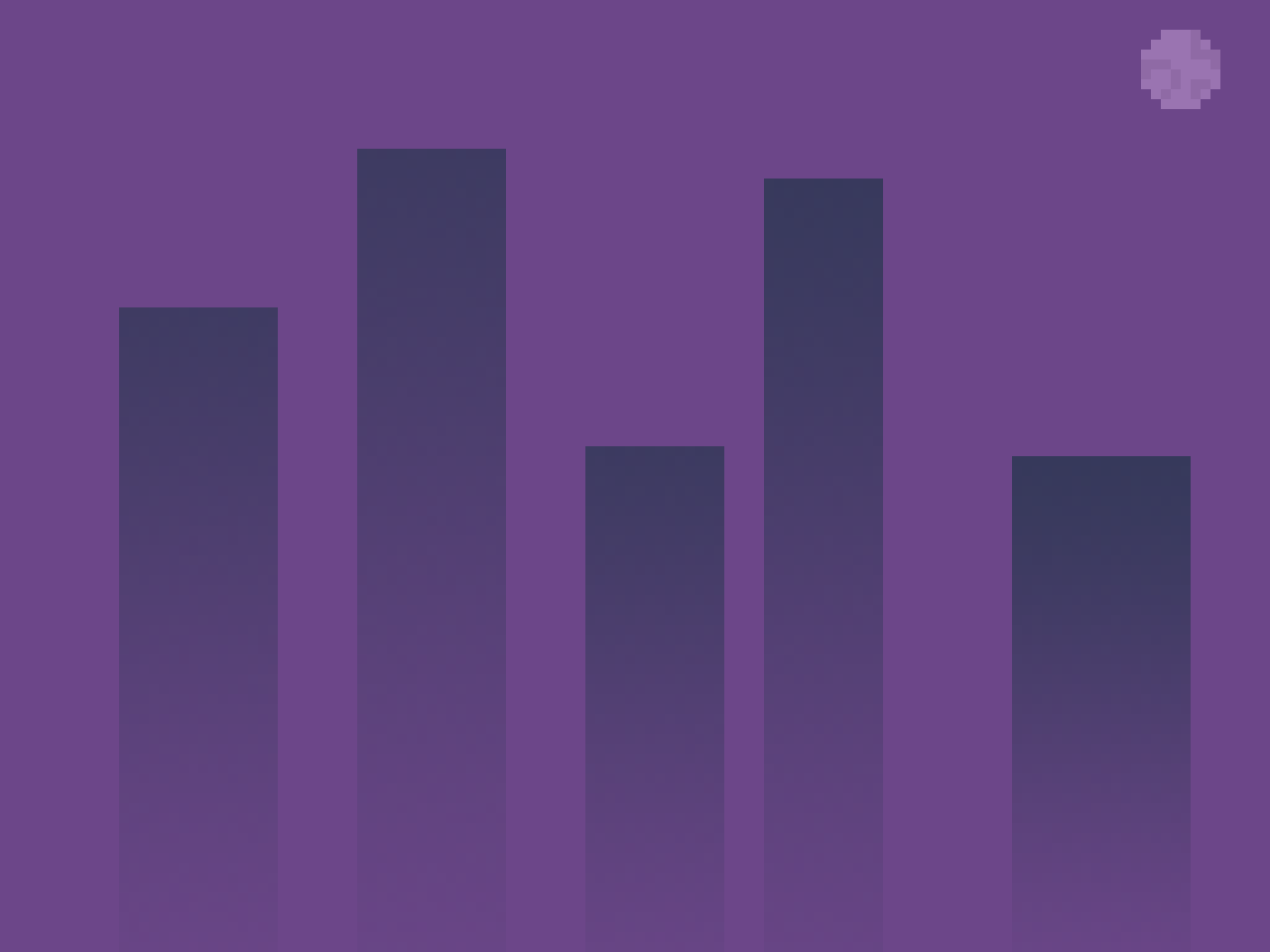
### 2. Layer

Danach haben wir die Skyline erstellt. Dieser Layer soll hinter der Strasse sein, damit es so scheint als würde der Player an der Skyline entlanglaufen. Bei Diesem Bild rechts sieht man schön, wie sich das Bild am Ende beginnt zu wiederholen, damit wir immer wieder das gleiche Bild nacheinander einsetzten können und die Häuser nie abgeschnitten werden, sondern immer weiterlaufen. Dieses Bild bewegt sich etwas langsamer als die Strasse, damit der Effekt gegeben wird, dass die Häuser etwas weg von der Strasse sind.

### 3. Layer

Dieser Layer ist hinter der Skyline zu finden. Er bewegt sich auch noch, jedoch am langsamsten von den 3 ersten Layers, da diese Wolken auch am weitesten Weg von der Strasse sein sollen.

### 4. Layer



Das ist der erste Layer der sich nicht bewegen darf, da sonst die Logik nicht aufgehen würde. Es soll so wirken, dass dieses Bild am weitesten weg von der Strasse sein soll. Wenn man genau hinsieht, merkt man, dass oben rechts der Mond zu finden ist. Der darf sich auf keinen Fall bewegen damit es realistisch aussehen soll.

### 5. Layer

Der 5. und letzte Layer ist nur für den Kontrast zuständig. Dieser darf sich auch nicht bewegen, da sonst das Bild verzerrt wird und da sonst an gewissen orten kein Kontrast mehr vorhanden wäre.

## Erstellung Player Character

Für unseren Charakter haben wir zuerst das Grundgerüst gewählt. Das meint wir haben den Mann gestaltet, wie er aussehen soll. Unser Charakter soll 3 Animationen beherrschen. Er soll rennen, rutschen und auch springen können. Für jede dieser Animation braucht man 3 Frames. Ein Frame ist nicht anderes als ein Bild, in dem der Charakter eine gewisse Haltung hat. Wenn man die drei Frames nacheinander animiert, sieht es aus als würde er rennen, springen oder rutschen. Wiederum sind auch alle diese Frames mit Photoshop erstellt.

### Animation: Rennen

Die drei folgenden Frames lassen wir nacheinander laufen, damit es aussieht als würde der Charakter einen rennen machen.

«1. Bein nach Vorn» «Beine kreuzen» «2. Bein nach Vorn»

### Animation: Springen

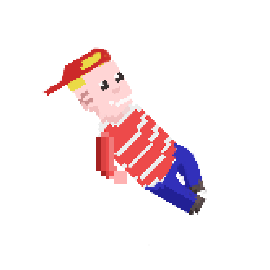
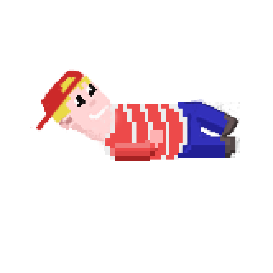
Die drei folgenden Frames lassen wir nacheinander laufen, damit es aussieht als würde der Charakter springen.

«abspringen» «fliegen» «landen»

### Animation: Rutschen

Die drei folgenden Frames lassen wir nacheinander laufen, damit es aussieht als würde der Charakter rutschen.

«runterfallen» «rutschen» «aufstehen»

### Extra Animation: Sterben

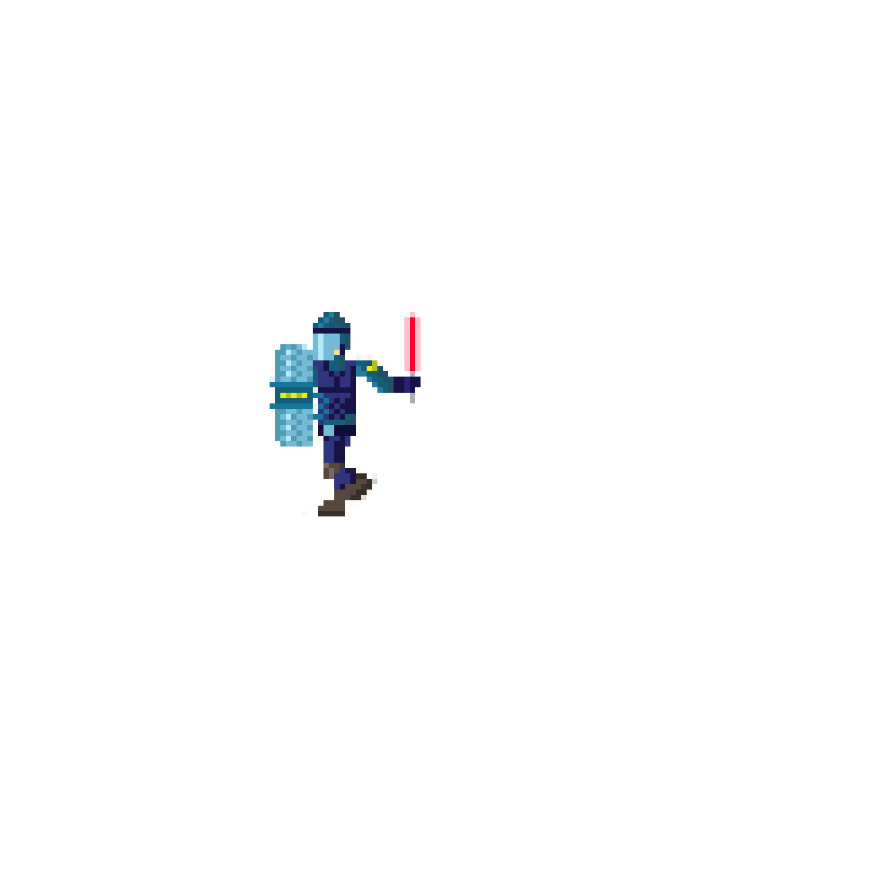
Die drei folgenden Frames lassen wir nacheinander laufen, damit es aussieht als würde der auf lustige weise seinen Kopf verlieren.

## Erstellung der Objekte/Gegner unseres Charakters

Was wir auch noch erstellen mussten waren Objekte, welchen der Charakter ausweichen muss. Diese Objekte kommen dann im Spiel dem Charakter entgegen. Als Objekt, über welches er springen soll, haben wir einen Polizisten mit einem Laser-Schwert erstellt. Dazu haben wir noch ein Objekt, welches dem Charakter entgegenfliegt, damit man unten durchrutschen muss. Dies ist nur das Laser-Schwert.

### Animation: Polizist mit Laser-Schwert

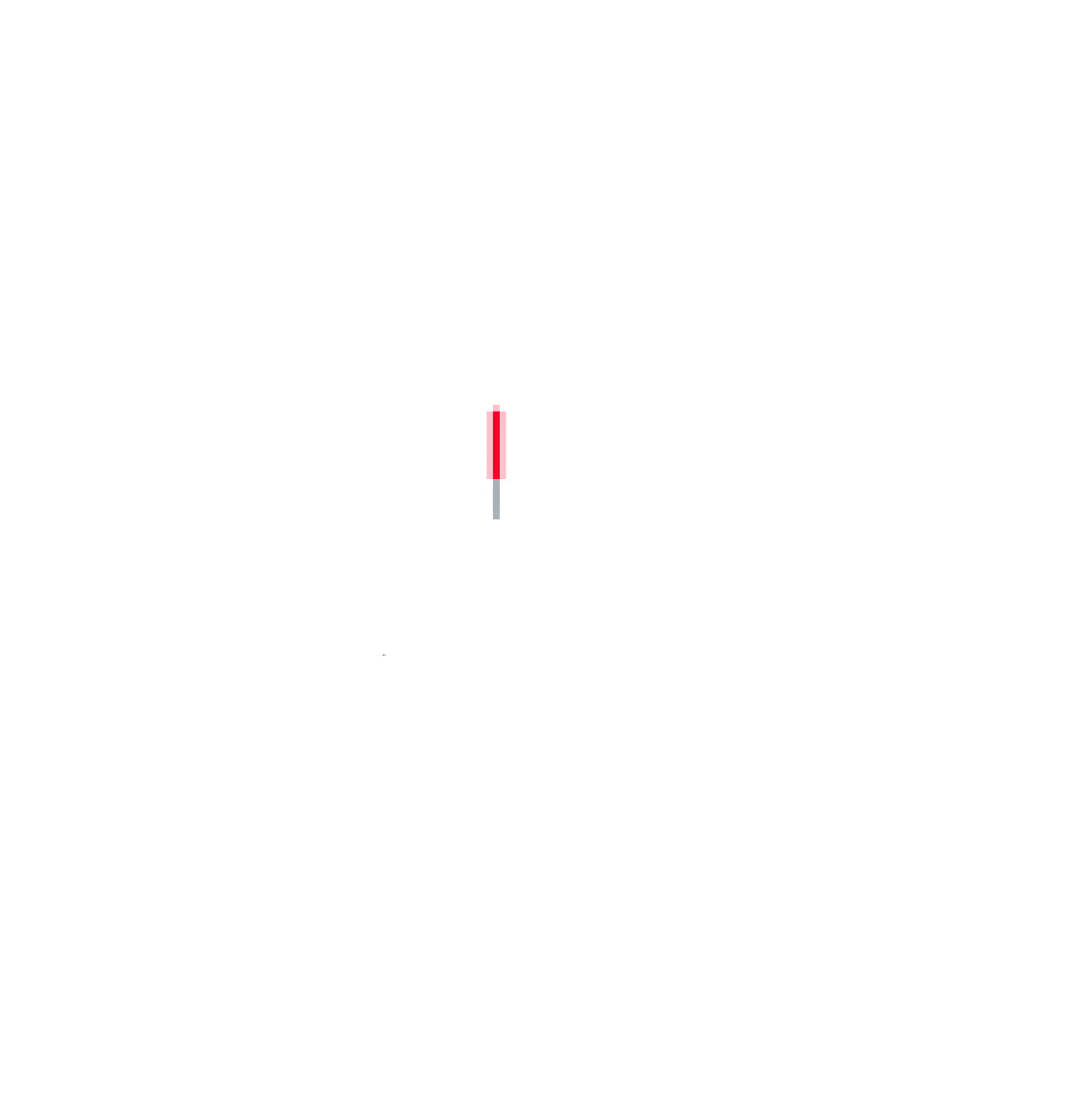
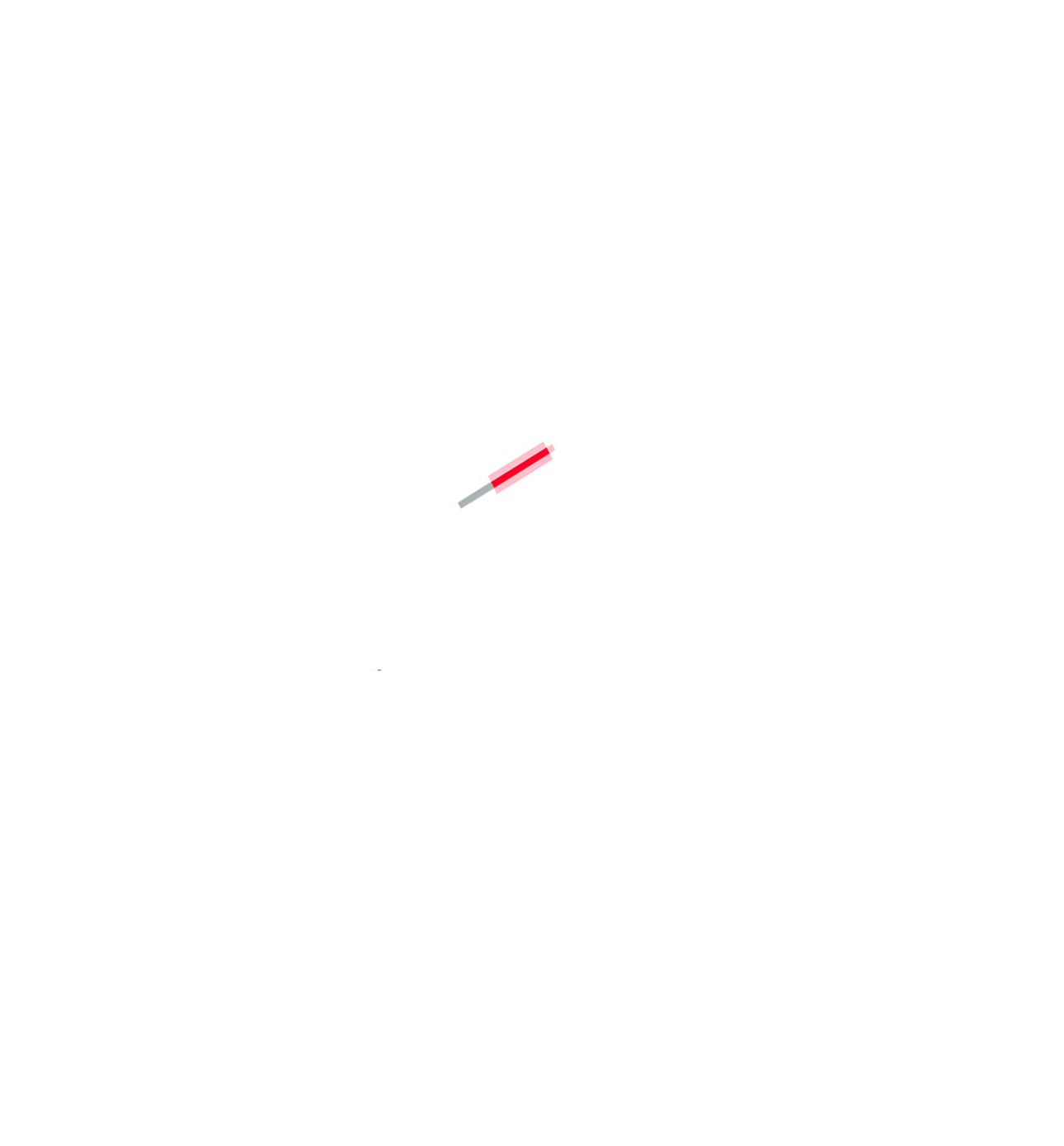
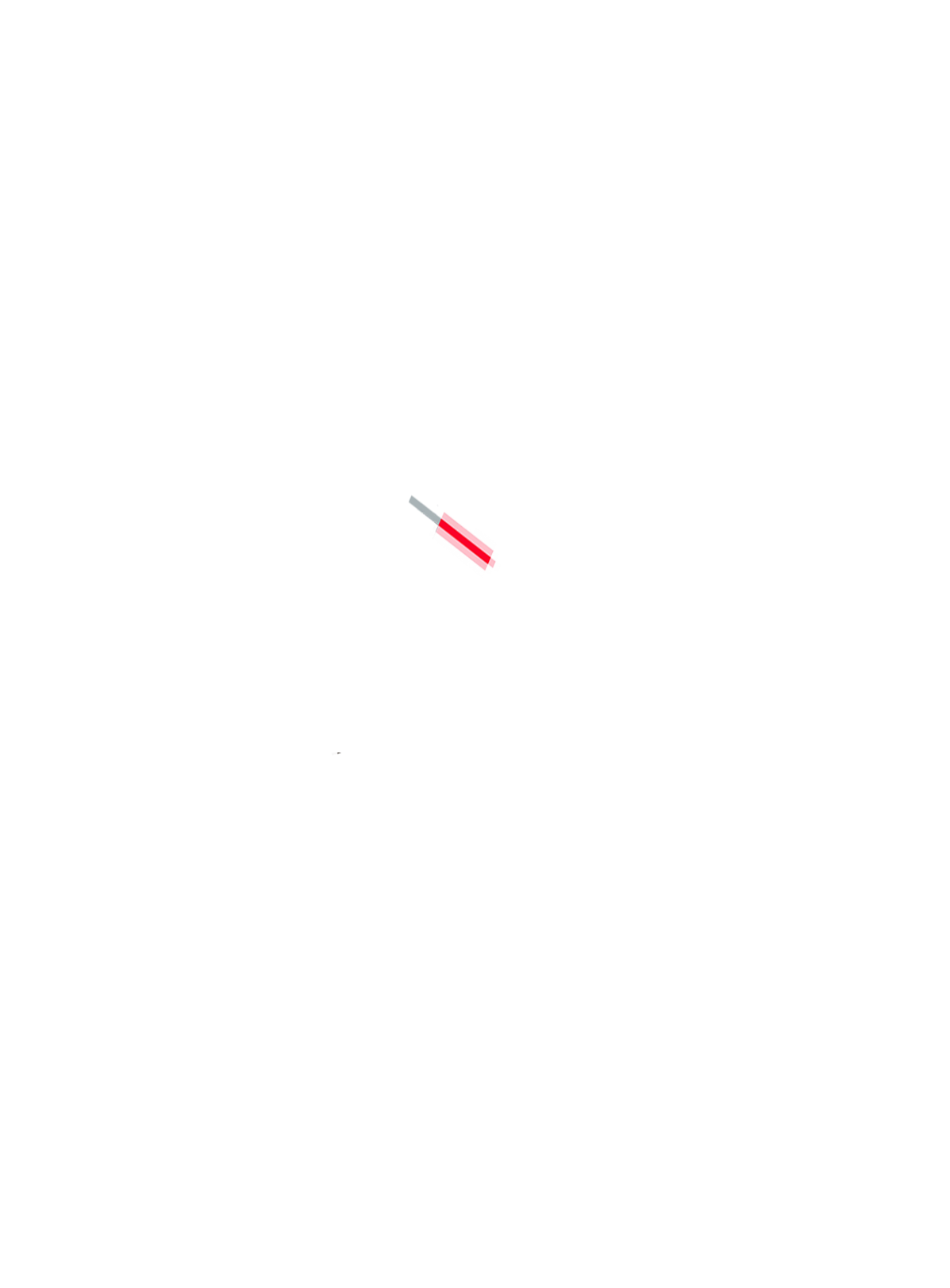
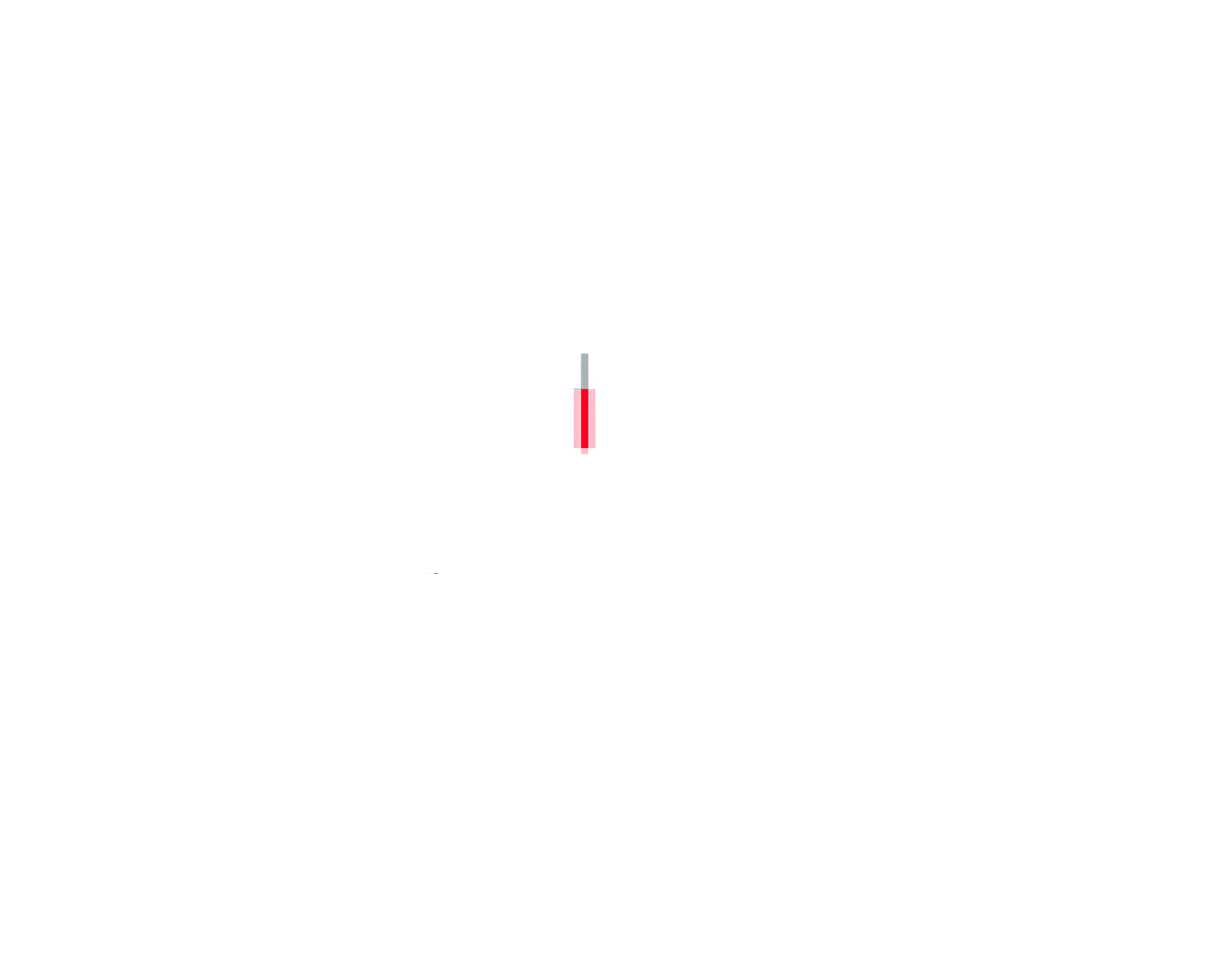
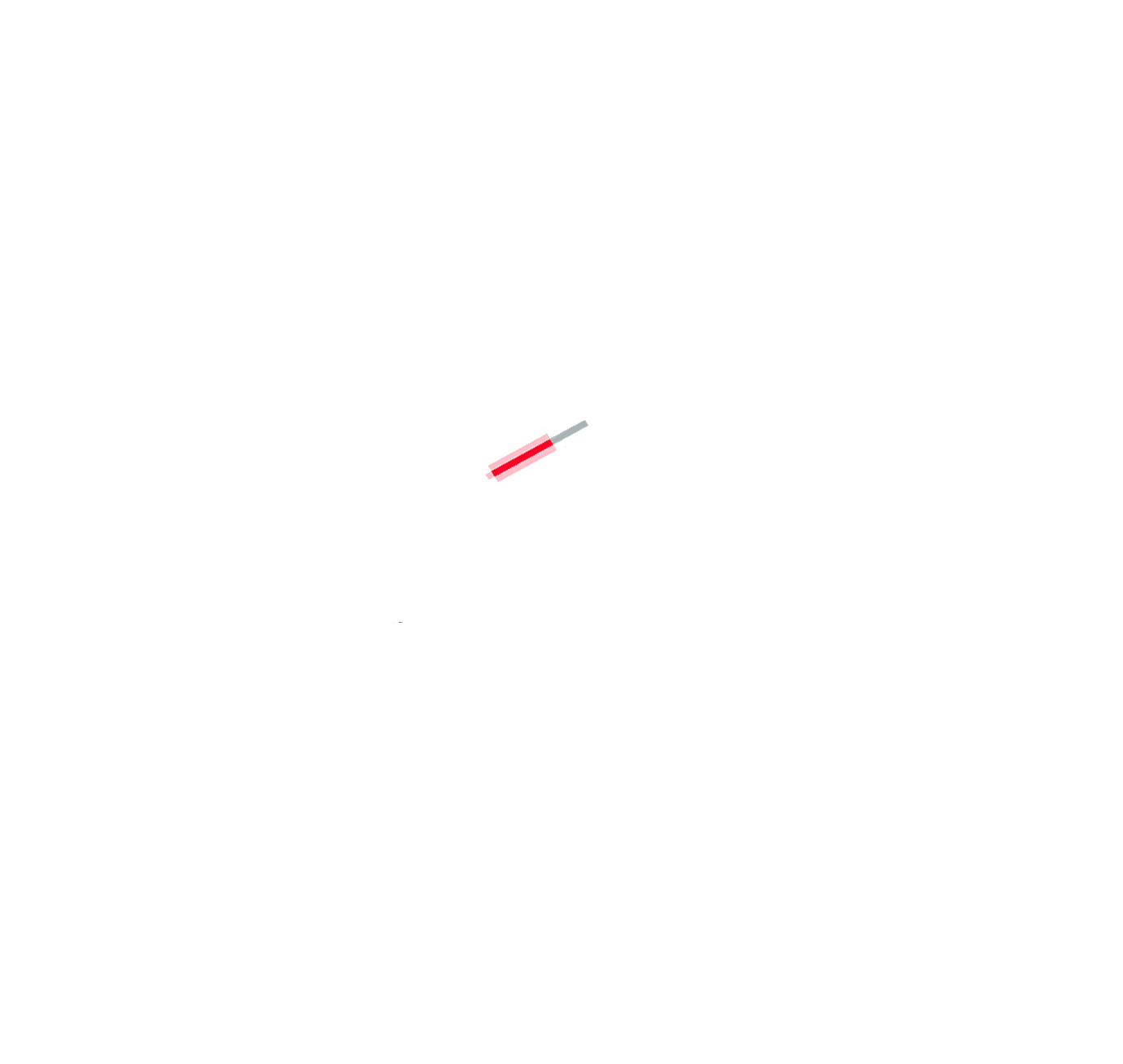
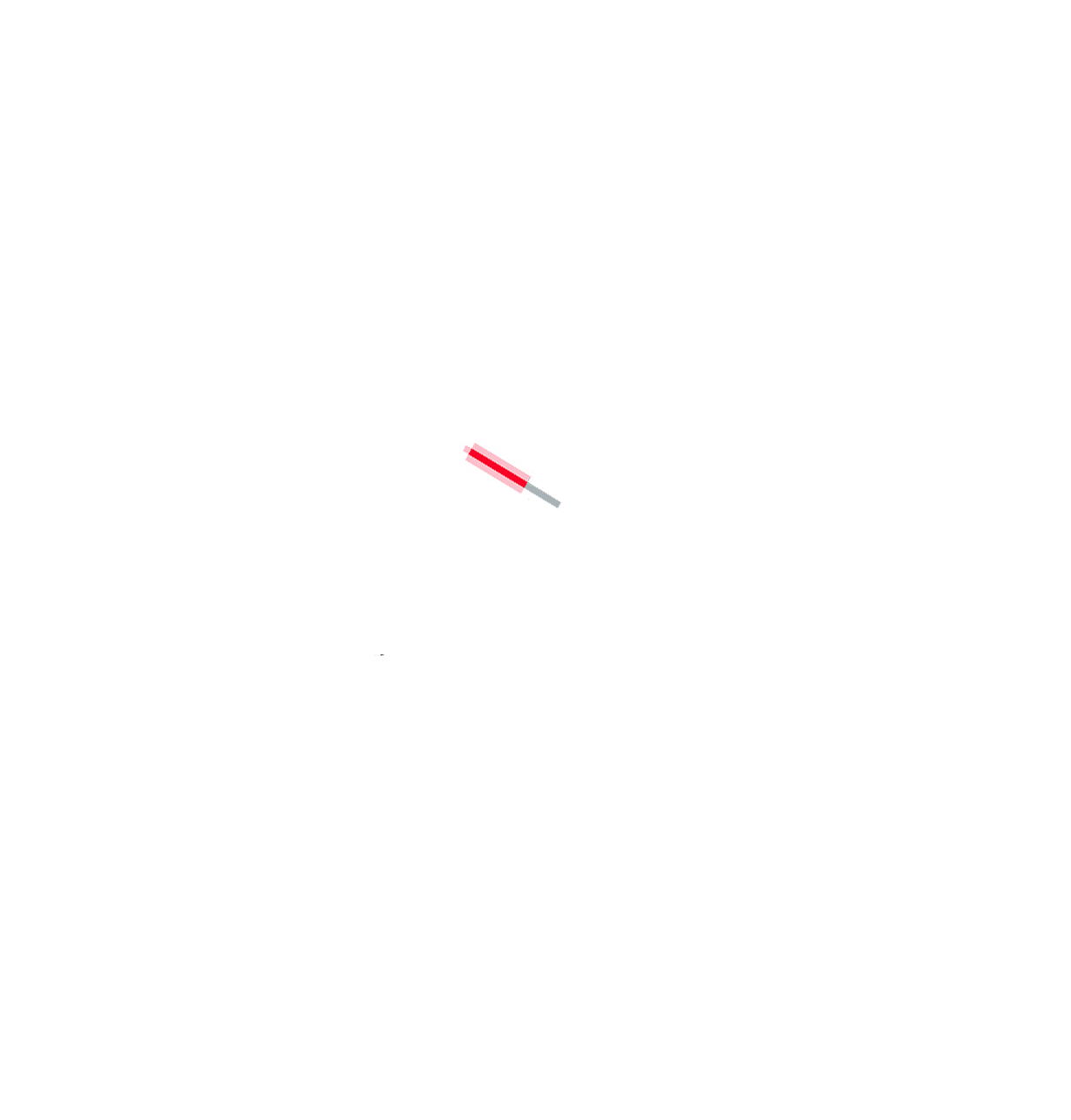
Für den Polizisten haben wir die gleichen drei Frames erstellt wie für den Charakter, wenn er am Rennen ist. Dieser Gegner/Objekt kann auch nichts ausser dem Charakter entgegen zu laufen, damit der Charakter über ihn springen muss.

«1. Bild» «2. Bild» «3. Bild»

### Animation: Fliegendes Laser-Schwert

Da wir mit dem Polizisten ein Gegner haben, über welchen der Charakter springen kann haben wir das fliegende Laserschwert noch einzeln eingebaut damit wir ein Element haben, unter welchem der Charakter unten durch rutschen muss. Das Schwert soll mit einer rotierenden Bewegung fliegen. Aus diesem Grund haben wir das Schwert in 6 verschiedenen Winkel gezeichnet und diese dann entsprechend aneinandergehängt.

🡺 🡺 🡺 🡺 🡺 🡺 🡺 🡺 🡺 🡺 🡺 🡺

### Animation: Sterben des Charakters

Wenn der Charakter in einen Gegner läuft wird immer folgende Animation abgespielt, da er der Spieler nun verloren hat.

«1. Bild» «2. Bild» «3. Bild»

## Soundtrack

Für den Soundtrack haben wir das Programm FL Studio 20.  
Wir haben uns alle für ein Sample entschieden, welches wir dann so bearbeitet haben, dass es zu dem Spiel passt. Das Sample zu unserem Soundtrack war «Sagittarius» von Daniel Kandi. Zu dem Sample haben wir dann ein einfaches Kick Snare Kick Snare Layout mit Hi-Hats hinzugefügt. Zu Schluss haben wir den Pitch des Samples erhöht und dieses neue Sample mit der Grundmelodie kombiniert. Das ganze Lied wurde dann noch mit einem Equalizer beschmückt welcher am Anfang des Tracks automatisiert wurde.



Und der fertige Soundtrack sieht in FL Studio 20 so aus:



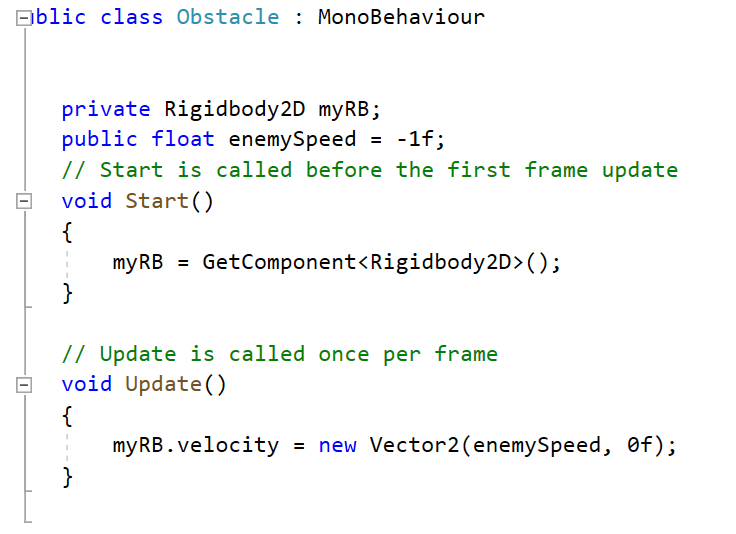
## Code: Player Controller

Beim Code des Player Controller, mussten wir die Funktionen einbauen. Die Funktionen sind in unserem Fall die Animationen Springen und Rutschen. Diese Animation konnten wir erstellen, in dem wir einen «unsichtbaren» Player erstellt haben, welchen wir an unseren Main Charakter gehängt haben. Dieser Player reagiert nun auf die geschriebenen Animationen und der Main Charakter führt diese dann aus.

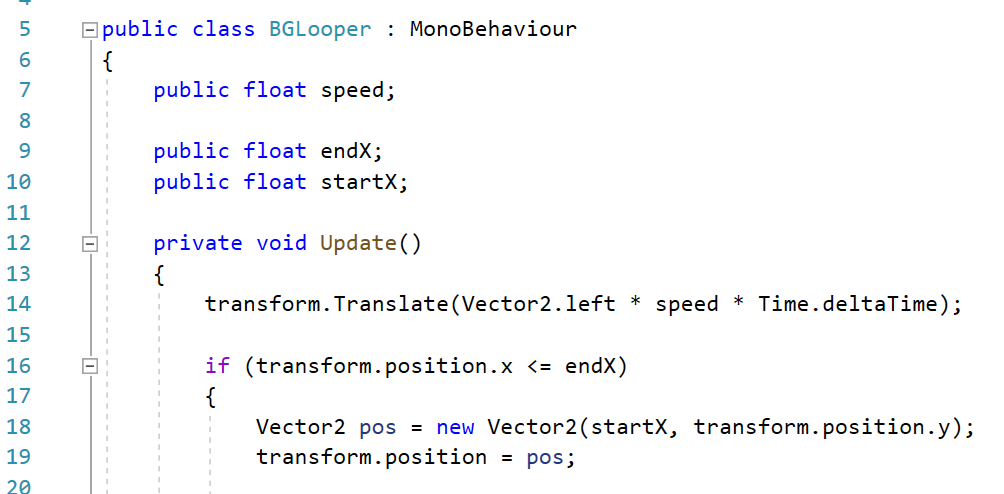


## Code: Obstacle

Beim Code Obstacle geht es darum, dass der Spieler wie auch die Hindernisse / Gegner Physisch da sind. Das meint, dass der Main Charakter auch getroffen werden kann oder der Maincharakter auch Ausweichen kann. Dazu haben wir noch die Geschwindigkeit der Gegner wie des Main Charakters festgelegt.



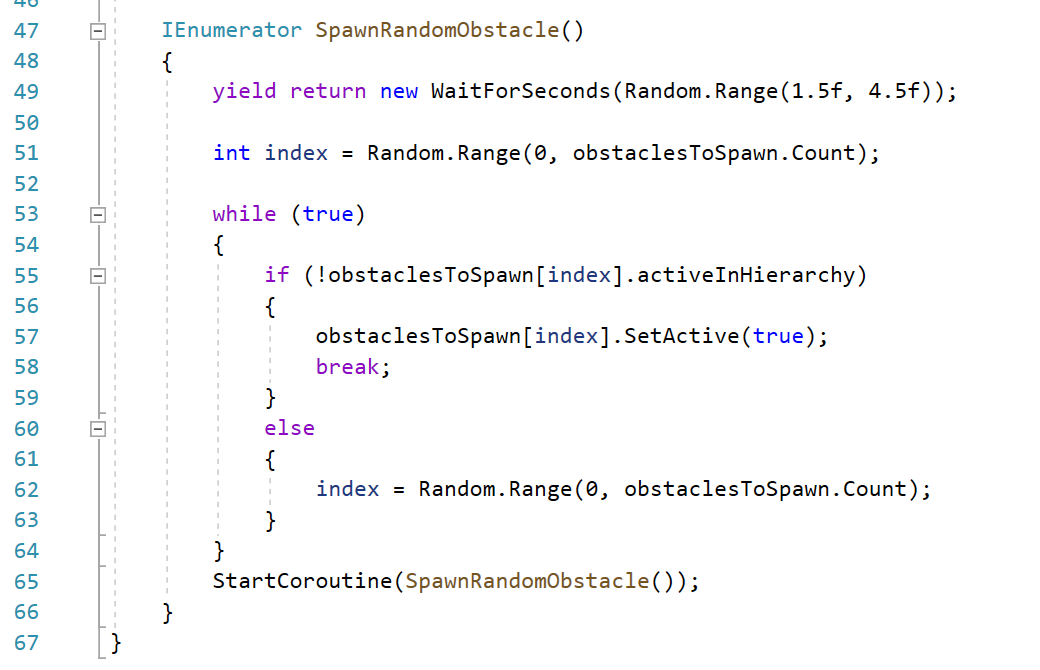
## Code: Background Looper

Beim Background Looper geht es um den Parallax Background. Wie bereits oben erklärt ist das ein Hintergrund, welcher aus mehreren Teilen besteht und sich bewegt. Beim Code mussten wir schauen, dass die 3 Hintergründe, die sich auch bewegen, sich zur richtigen Zeit an den richtigen Ort bewegen. Um die Orte festzulegen konnten wir mit der X - und Y - Achse arbeiten. Dazu mussten wir auch angeben wann sich der Hintergrund bewegen muss.

## Code: Obstacle Spawner

Beim Code Obstacle Spawner, geht es um die Erstellung der Gegner / Hindernisse während des spielen. Wir haben alle 3 Hindernisse hinterlegt. Dann haben wir einen Generator erstellt, welche diese Hindernisse vorzu erstellt. Dann brauchten wir noch einen Zufälligen Zähler, welcher immer ein Hindernis von 3 auswählt und dieses immer zwischen 1.5 und 4 Sekunden Abstand spawnt (erstellt). Mit diesen Funktionen können wir darstellen, dass immer Zufällig ein Kegel, ein Polizist oder nur das Lichtschwert erscheint.

So sieht der Generator aus:



# Kontrolliren

## Testprotokoll

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Testfall | Resultat | Datum | Tester | Bemerkung | Unterschrift |
| 1 | Passed | 25.11.2020 | B.Krbanjevic |  |  |
| 2 | Passed | 25.11.2020 | B.Krbanjevic |  |  |
| 3 | Passed | 25.11.2020 | B.Krbanjevic |  |  |
| 4 | Passed | 25.11.2020 | B.Krbanjevic |  |  |
| 5 | Not passed | 25.11.2020 | B.Krbanjevic | Wir haben die Soundeffekte ausgelassen |  |
| 6 | Not passed | 25.11.2020 | B.Krbanjevic | Wir konnten das Spiel nicht auf das Handy laden |  |
| 7 | Not passed | 25.11.2020 | B.Krbanjevic | Wir konnten das Spiel nicht auf das Handy laden |  |
| 8 | Passed | 25.11.2020 | B.Krbanjevic |  |  |
| 9 | Not passed | 25.11.2020 | B.Krbanjevic | Wir konnten das Spiel nicht auf das Handy laden |  |
| 10 | Not passed | 25.11.2020 | B.Krbanjevic | Die Karte ist immer die gleiche, die Hindernisse Spawnen in verschiedenen Abständen |  |

# Auswerten

## Fazit zum Projekt

Wir haben das Projekt, Erstellung eines Handyspiels, gewählt. Am Anfang des Projektes hatten wir keinerlei Vorwissen, was das Projekt um einiges erschwert. Schliesslich haben wir uns zu Beginn sehr viel informiert und dazu sehr viel Neues kennengelernt. Unser Wissen haben wir in dem Projekt um ein Vielfaches erweitert, was uns schliesslich einen vertieften Einblick in die Programmierung eines Spiels erhielten. Dazu ist es ein sehr grosses und aufwendiges Projekt, was dazu führte, dass wir die Projektzeit um eine Woche verlängern mussten. Schliesslich haben wir die Aufgaben den Stärken der einzelnen Personen zugeteilt und bearbeitet. Im Team konnten wir gut miteinander arbeiten und haben uns bei Schwierigkeiten gegenseitig unterstützt und Feedback gegeben. Leider konnten wir das Spiel nicht komplett fertigstellen, da die Zeit nicht reichte und der Aufwand ziemlich gross war. Schlussendlich haben wir trotzdem ein Spiel auf die Beine gebracht, welches auf dem Computer funktionsfähig ist, was uns als Team sehr freut. Dazu war es eine sehr spannende Thematik und wir konnten sehr viel hinter einem Spiel kennenlernen und sind erstaunt, was alles damit möglich ist.

## Reflexion Luca Duttwiler

Bei diesem Projekt war ich im ersten Drittel für das Artwork und danach für alles was mit Unity zusammenhing zuständig.   
   
Es hat mir bestimmt ein paar graue Haare beschert, aber ich würde jederzeit noch mehr davon in Kauf nehmen, denn die der Dopaminrausch nach einem gelösten Problem war besser als damals das Tor der Schweizer gegen die Spanier.   
   
Am meisten Sorge hatte ich definitiv damit das Framework des ganzen zu verstehen, es gab vieles zu beachten, oft hatte der Code dutzende Fehler und die Assets haben sich gefühlt gegen uns verschworen. Doch wir haben durchgepflügt wie ein Bauer im Frühling und haben bis auf eines für alles eine Lösung gefunden.

Dieses eine Hindernis war der Code für die „Button“, welcher nötig gewesen wäre damit sich das Spiel auf einem Touchscreen spielen lassen würde.   
Die Physik liess sich da draufschrauben, nicht aber die Animationen.   
   
Ich bin zu guter sehr zufrieden mit unserer Arbeit, viele Erfahrungen, die ich gesammelt habe, werde ich mir privat weiterhin zu Nutze machen.

## Reflexion Shahin Afshar

Ich habe mit meiner Gruppe das Projekt Erstellung eines Handyspiels gewählt. Anfangs war die Arbeit nicht so einfach, da ich keinerlei Erfahrungen im Bereich Programmieren hatte. Schliesslich habe ich mich viel Informiert und dementsprechend viel gelernt. Ein Spiel selbst zu erstellen ist eine Tätigkeit, welche ich schon immer bearbeiten wollte. Der Projekttag ist dafür perfekt geeinigt, was und schliesslich auf die Idee brachte eines zu erstellen. Das wichtigste war meiner Meinung nach einer guten Idee zu finden, wie wir das Spiel aufbauen und welche Elemente darin enthalten sind. Schliesslich kamen in Laufe des Projekts immer mehr Ideen und Vorschlägen dazu, welche wir als Team versuchten zu bewältigen. Ich habe sehr viel Neues gelernt wie die Arbeit mit Photoshop, mit dem ich mich Hauptsächlich in dem Projekt beschäftigt habe. Ich konnte lernen, wie ich Grafiken erstelle und wie ich PNG für eine Animation erstelle. Schlussendlich kann ich sagen, dass das Projekt sehr spannend war und ich viel lernen konnte. Leider konnten wir unser Spiel nicht fertigstellen, da die Zeit es nicht zugelassen hat. Trotz alles bin ich erstaunt, was wir auf die Beine gestellt haben und freue mich schon auf das nächste Projekt.

## Reflexion Bogdan Krbanjevic

Nun haben wir es ja wirklich geschafft ein eigenes Spiel zu erstellen, mit eigenen Animationen, Gegnern und eigener Musik, von mir Komponiert. Ich war zuständig für Die Anforderungsanalyse, Testplanung, Testspezifikationen, das Testprotokoll, den Soundtrack und doch ein paar Dinge. Worauf ich richtig stolz bin ist der Soundtrack, ich habe zuerst nach einem Sample gesucht welches zum Stil unseres Spiel passte. Dieses habe ich dann angepasst und noch einige Elemente hinzugefügt. Für die restlichen Dinge war ich auch schon bei vorherigen Dokumentationen und Projekten verantwortlich also ging, das ohne Probleme. Ich wollte schon in der Quarantäne ein Spiel erstellen weil mir so langweilig war und jetzt haben wir es in einer Gruppe hingekriegt.

## Reflexion Gregory Bobst

Ich bin selber überrascht, dass wir ein Vollständiges Spiel erstellen konnten. Ich habe sehr viel gelernt von diesem Modul. Einerseits war es eine Herausforderung im Programmieren hat aber gleich gepasst, da wir Momentan das Modul C# bearbeiten und das Spiel auch in dieser Sprache ist. Am meisten profitiert habe ich vom Dokumentieren, da ich das erste mal einen Vollständigen Code dokumentieren musste. Schade ist, dass das Spiel noch nicht auf dem Handy spielbar ist. Jedoch werden wir ausserhalb des Projektes den Code weiterverfolgen bis es funktioniert.