

Design Dokumentation des Spiels

RED

Rogue Exterminate (un)Dead

Referent: Prof. Dipl.-Ing. Jirka R. Dell'Oro-Friedl

Vorgelegt am: 22.07.2022

Vorgelegt von: Matthias Ngo

Matthias.ngo@hs-furtwangen.de

Inhaltsverzeichnis

| | |
|-------------------------------|-----|
| Inhaltsverzeichnis | III |
| 1 Konzept..... | 1 |
| 2 Umsetzung..... | 3 |
| 2.1 Units and Positions | 3 |
| 2.2 Hierarchy | 6 |
| 2.3 Editor | 6 |
| 2.4 Scriptcomponents | 6 |
| 2.5 Extend..... | 7 |
| 2.6 Sound | 7 |
| 2.7 VUI..... | 7 |
| 2.8 Event-System | 8 |
| 2.9 External Data | 8 |
| 2.10 Light..... | 8 |
| 2.11 Physics | 10 |
| 2.12 State Machines..... | 11 |
| 2.13 Sprites | 12 |

1 Konzept

Genre: Rogue-like

Dimension: 2D

Environment: Top-Down

Fighting-Mechanic: Melee (Sword, Axe, etc.)

Game-Mechanic:

Ein Rogue-Like ist ein Genre, welches durch fehlende Lademöglichkeiten eines Speicherstands nach dem Tod gekennzeichnet ist. Der Spieler soll in diesem Spiel in verschiedene Levels mit erhöhtem Schwierigkeitsgrad versuchen, so weit wie möglich zu kommen. Der Score pro überwindet Level berechnet.

Kennzeichen:

- Schwierigkeitsgrad erhöht sich pro Level
- Mehr Gegner pro Level
- Soll nicht einfach sein

Ziel: Highscore

2 Umsetzung

In diesem Abschnitt wird die Umsetzung des Spiels anhand der Kriterien erläutert.

2.1 Units and Positions

Das Spiel ist ein 2D Spiel und verwendet die X und Y-Achsen. Der Mittelpunkt des Spielfelds ist der Nullpunkt und gleichzeitig der Spawn Point des Spielers. Mit dem Gedanken, dass 1 Unit etwa 1 Meter entspricht, ist der Spieler etwa auf 1,7 Meter skaliert.

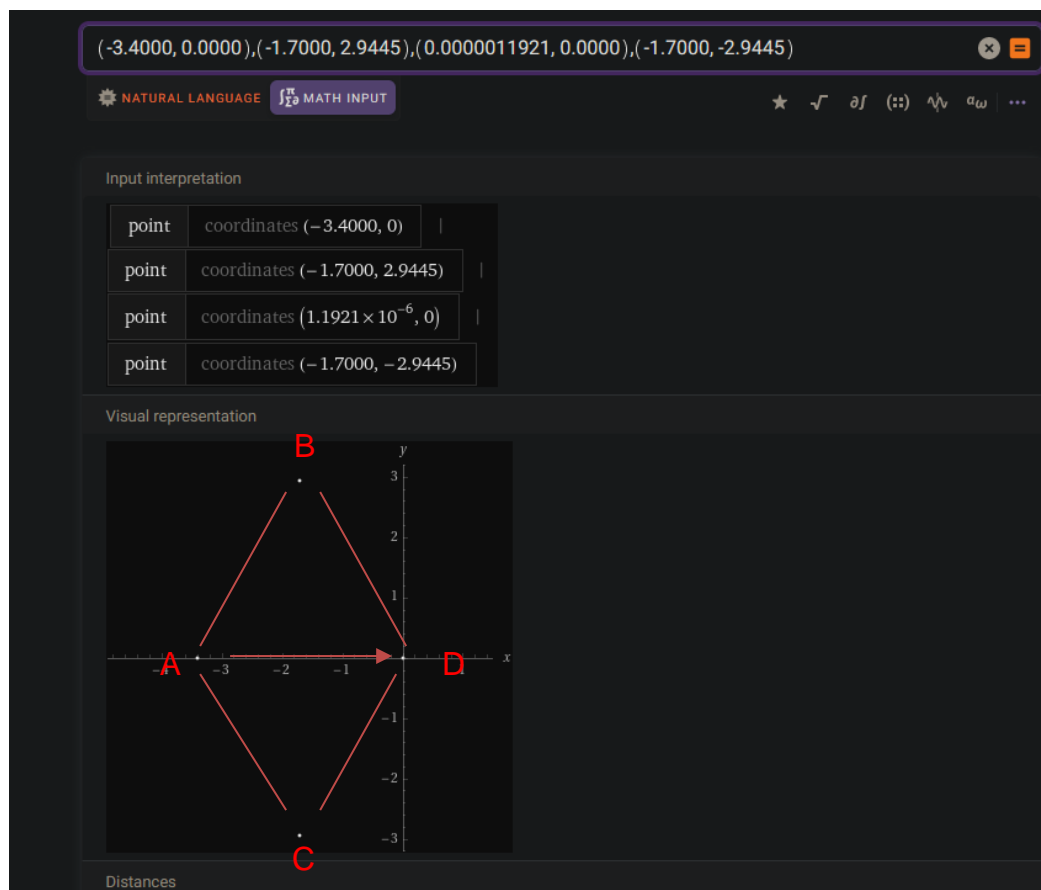
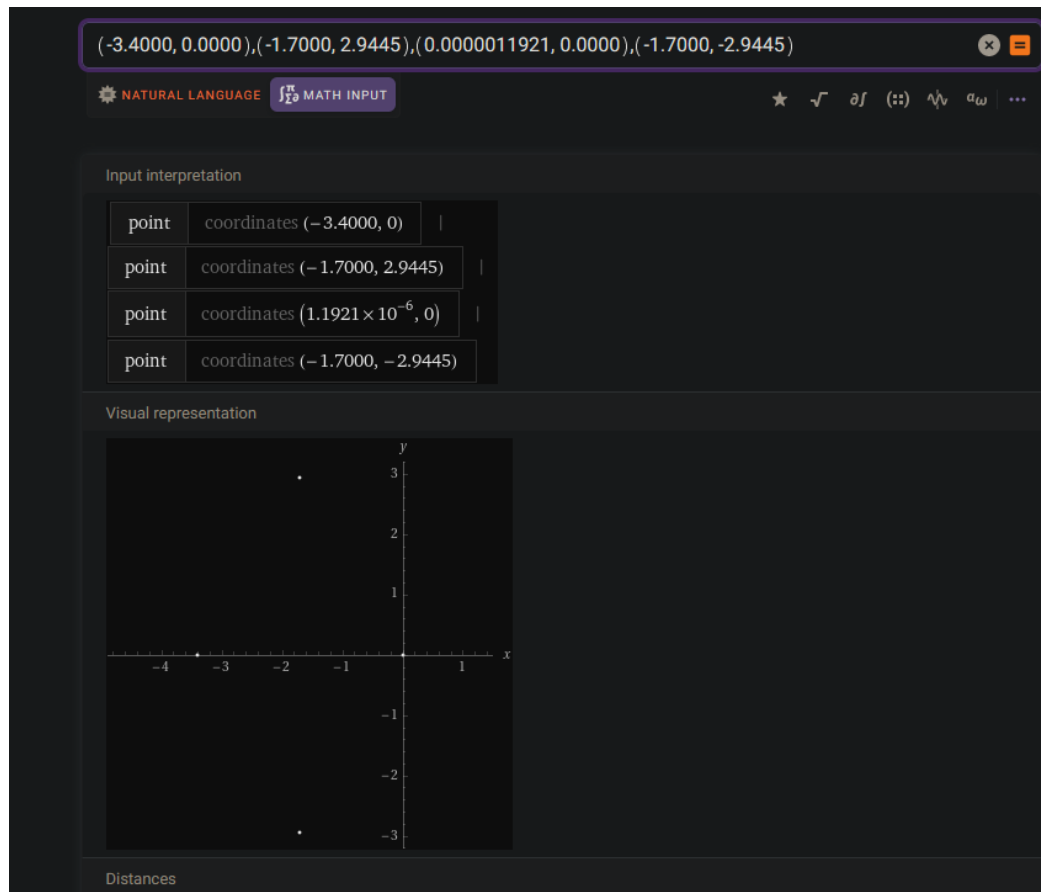


Schwierigkeit und Lösung:

Eine Schwierigkeit stellte sich bei Erkennung von Schaden. Der Spieler kann sich nicht nur noch auf den Achsen bewegen, sondern auch Diagonal. Da der Gegner den Spieler verfolgt und somit den kürzesten Weg zum Spieler findet, soll er sogar die Möglichkeit haben, von allen Winkeln anzugreifen.

Hierzu bietet die folgende Lehre über Rotation in einem 2-dimensionalen Raum eine große Hilfe:

https://matthew-brett.github.io/teaching/rotation_2d.html



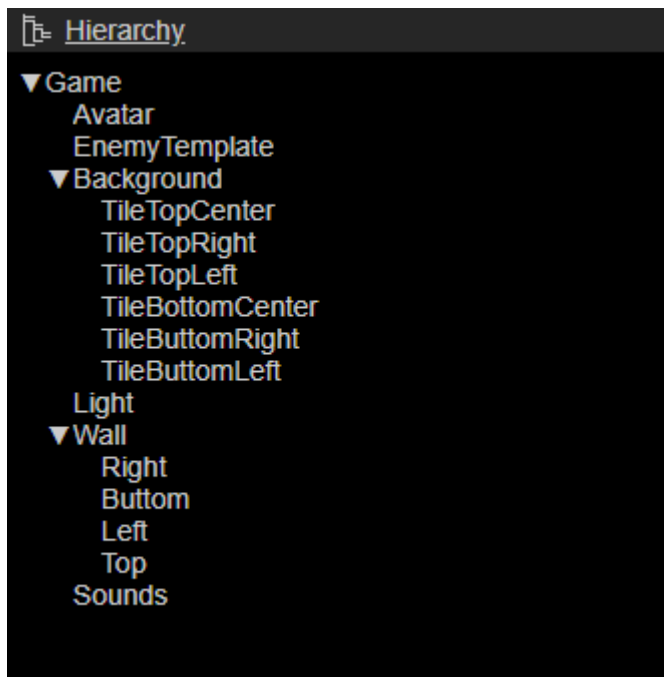
Soll Punkt **A** die Position des Spielers darstellen, so werden Punkte **B** und **C** mithilfe einer Formel berechnet. **D** kann einfach mithilfe von Attack Range berechnet werden. Hier im Bild ist war die Attack Range auf 3,4 eingestellt und hatte „fast“ den Nullpunkt erwischt.



Die Winkel von B und C kann eingestellt werden.

2.2 Hierarchy

Die Hierarchie ist sehr flach angelegt. Alle Nodes worauf zugegriffen werden, sind weiter oben in der Hierarchie, um ein schneller Zugriff zu erlauben. Hintergrund und Wände, worauf nicht über Code zugegriffen werden und die aus mehreren Teilen bestehen, sind weiter unten in der Hierarchie.



2.3 Editor

Hintergrund, Wände, Licht und Sounds wurden im Editor erstellt und hinzugefügt. Es ist einfacher, im Editor den statischen Wänden eine rigid Body Komponente hinzuzufügen. Beleuchtung konnte gut im Editor angepasst werden, da das Ergebnis sofort sichtbar war. Der Rest wurde per Code gelöst.

Ein weiterer Vorteil bei der Bearbeitung von Beleuchtung war die das Ausprobieren von verschiedenen Shadern um den richtigen Shader auszuwählen. Zum Beispiel benötigt eine Textur den „ShaderGouraudTexture“ Shader damit Licht Effekte richtig auf der Fläche angezeigt werden. Scriptcomponents

2.4 Extend

Spieler und Gegner sind von der Klasse Actor abgeleitet. Actor wiederum ist von der Node Klasse abgeleitet. Damit sind Spieler und Gegner Nodes und es ist einfacher, Komponente an die Nodes anzuhängen.

2.5 Sound

Angriffe, Laufen und Treffer des Avatars können akustisch wahrgenommen werden. Für den Gegner wurde entschieden das nur ein Treffer beim Gegner abgespielt wird. Bei der Akustik soll der Spieler sich auf den Avatar konzentrieren. Je nach Level erscheinen auch mehr Gegner. Deswegen ist es sinnvoller die Sounds beim Gegner so gering zu halten wie möglich.

2.6 VUI

Der Spieler wird über Health, Level und Score per Visual User Interface informiert. Ein Custom Font wurde für das Spiel verwendet.



2.7 Event-System

Ein Keyboard Event hilft bei der Erkennung von Angriffen (SpaceBar).

2.8 External Data

Der Spieler kann im Config File die Maximale Geschwindigkeit sowie die Beschleunigung des Avatars festlegen.

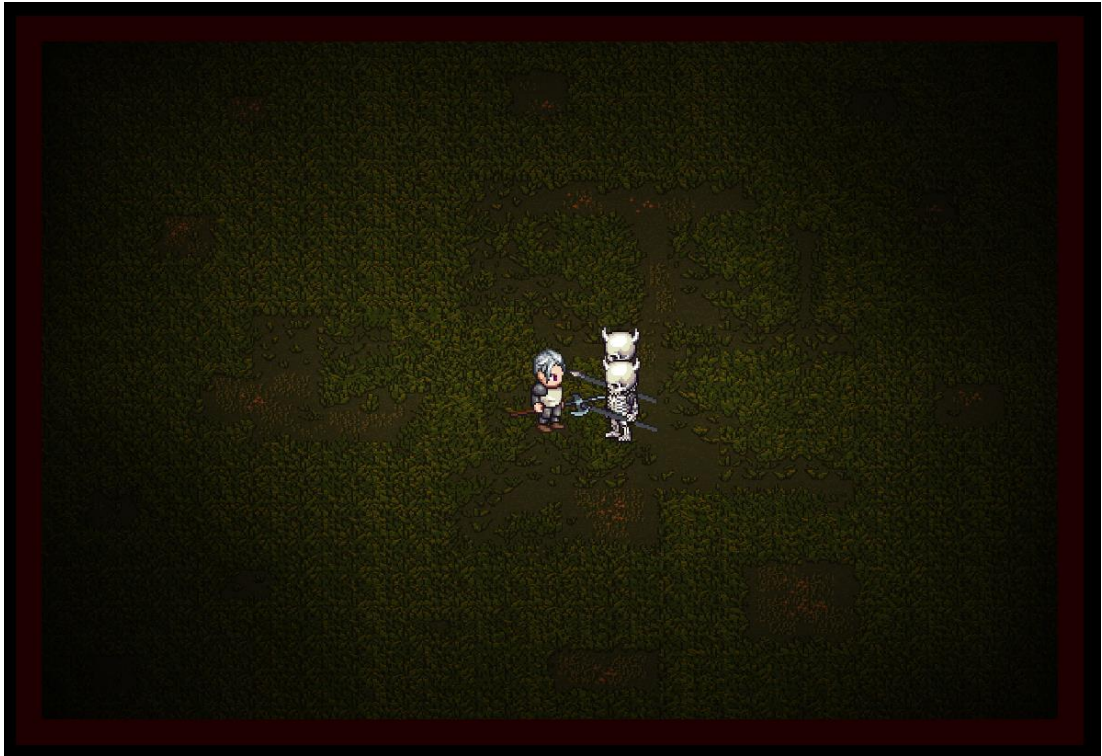
2.9 Light

Licht wird verwendet, um das Spielfeld zu beleuchten und je nach Health des Spielers wird das Spielfeld rötlicher durch die Beleuchtung.

100% Health:



50% Health:



Tod:



2.10 Physics

Physics wird verwendet, um bei den Wänden ein Bounce Effekt zu erzeugen, falls der Spieler gegen die Wand läuft.



Läuft man leicht gegen die Wand, so wird der Spieler von der Wand weggedrückt.



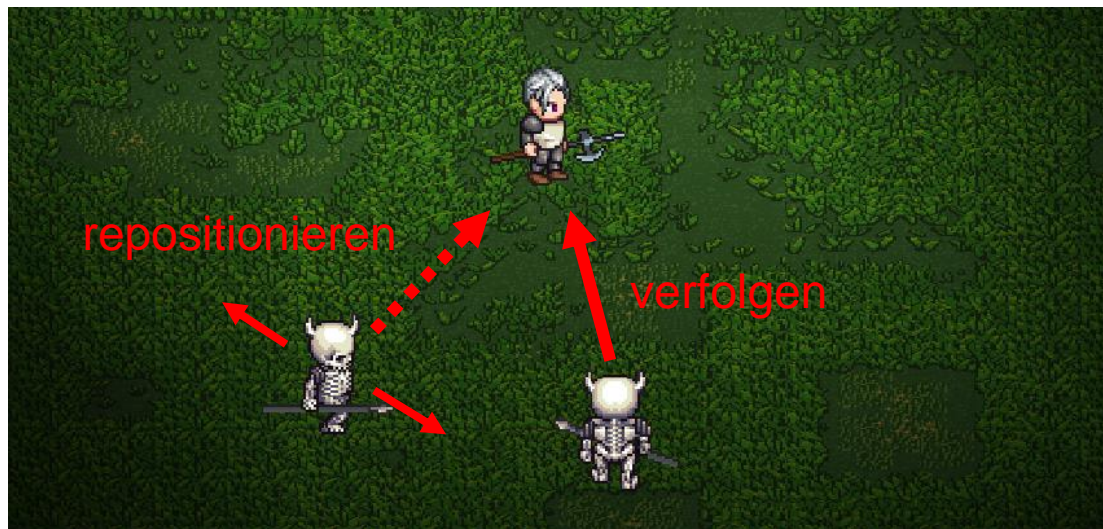
Schwierigkeit und Lösung:

Nur Spieler und Wände besitzen Rigid Body. Die Gegner jedoch nicht. Das Spiel litt unter Performance, sobald mehrere Gegner mit Rigid Body erschienen. Um Physics zu entlasten, werden Gegner ohne Rigid Body erzeugt. Dies führte jedoch dazu, dass Gegner durch Wände laufen konnten, weshalb das Spielfeld zusätzlich noch abgegrenzt wird.

2.11 State Machines

State Machines wird bei den Gegnern verwendet um zwischen den Zuständen "verfolgen" oder "repositionieren" zu wechseln

Beispiel:

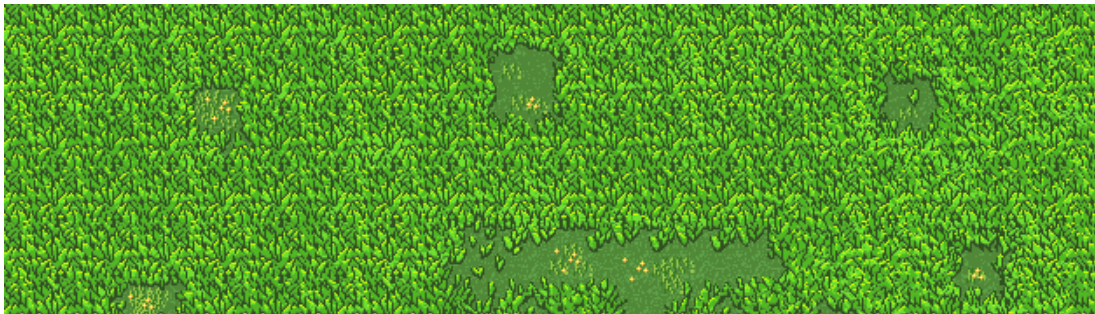


Vorteil: Bei niedrigem Level, wechselt die Gegner zwischen „repositionieren“ und „verfolgen“. Dies hilft dabei, dass der Spieler zu Beginn nicht von jedem Gegner angegriffen wird. Mit höherer Schwierigkeit, erhöht sich die Wahrscheinlichkeit dass der Gegner einen verfolgt. Je weiter man kommt, desto schwieriger wird es. Bei höherem Level verfolgen alle Gegner den Spieler.

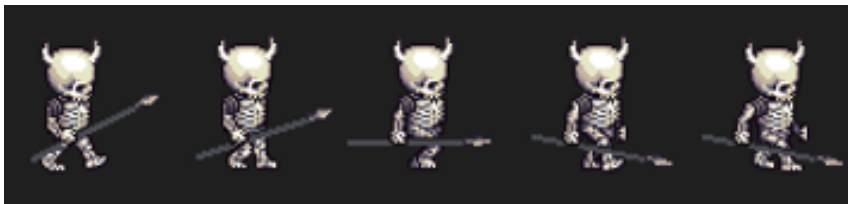
2.12 Sprites

Die Sprites wurden in einem zusätzlichen Editor (aus GitHub) für LibreSprites (ähnlich wie Asprite) erstellt. Hintergrund wurde aus dem Internet ausgesucht. Alle Credits befinden sich im Assets Ordner des Projekts.

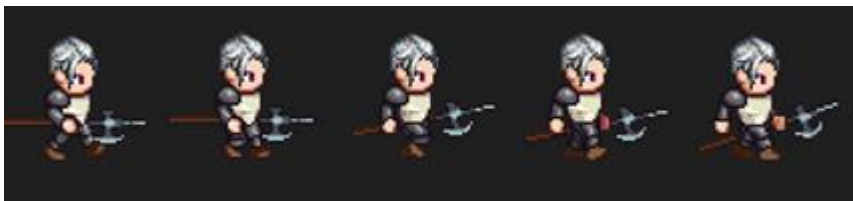
Hintergrund (Ausschnitt):



Gegner (Ausschnitt):



Spieler (Ausschnitt):



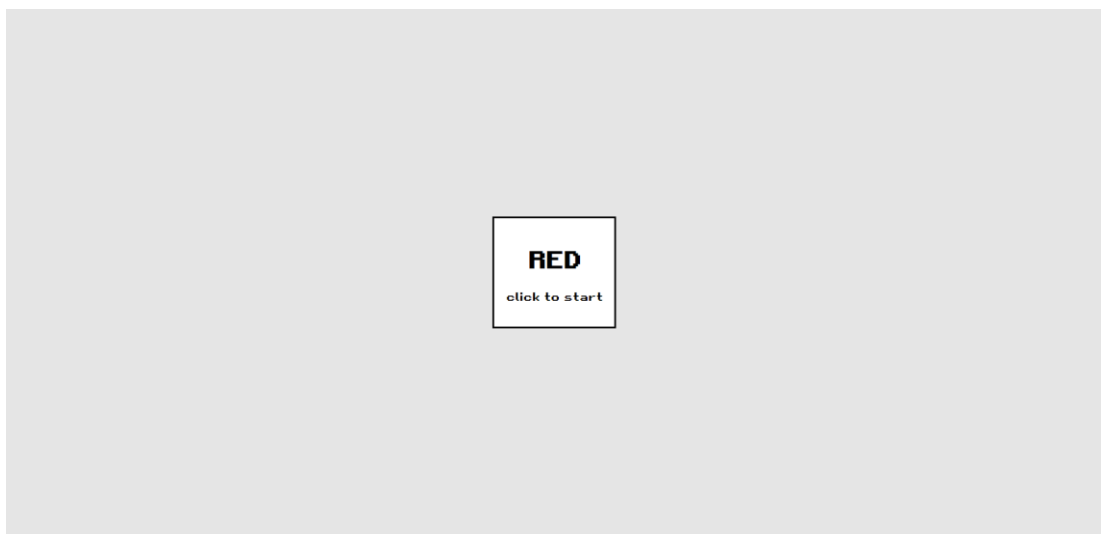
Schwierigkeit und Lösung:

Das Spiel erfordert viele verschiedene Sprites, da es ein Top-Down Spiel ist. Der Spieler sowie der Gegner können nicht nur auf der Horizontale Angreifen aber auch auf der Vertikale. Damit ist es nicht möglich mit einem Sprit und mehrere Spiegelungen das Problem zu lösen. Hierzu wurden mehrere Sprite Nodes erstellt die je nach Input den passenden Spirte Node aktivieren, während der Rest deaktiviert ist.

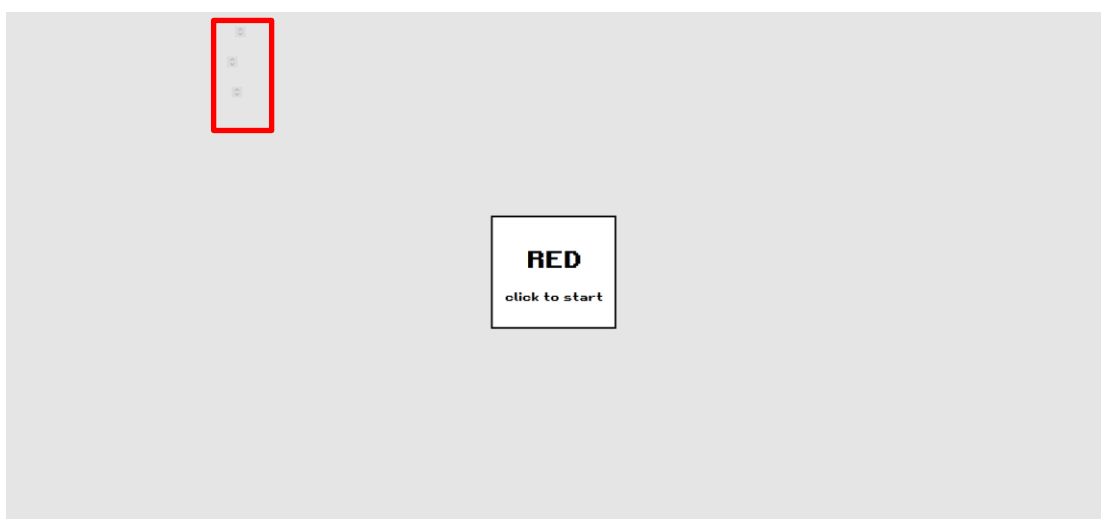
3 Probleme

Ein Problem stellt sich bei der Verwendung von verschiedenen Browsern. Das Spiel wurde größtenteils mit dem Browser Google Chrome getestet. Jedoch war bei der Verwendung des Browsers Mozilla Firefox Bugs zu erkennen, die bei Google Chrome nicht erkennbar waren:

Startbildschirm bei Google Chrome:



Startbildschirm bei Mozilla Firefox:



Im Spiel bei Google Chrome:



Im Spiel bei Mozilla Firefox:



Des Weiteren war auch ein Performanceunterschied zwischen den Browsern zu erkennen. Je besser die Performance, desto „schneller“ läuft das Spiel.