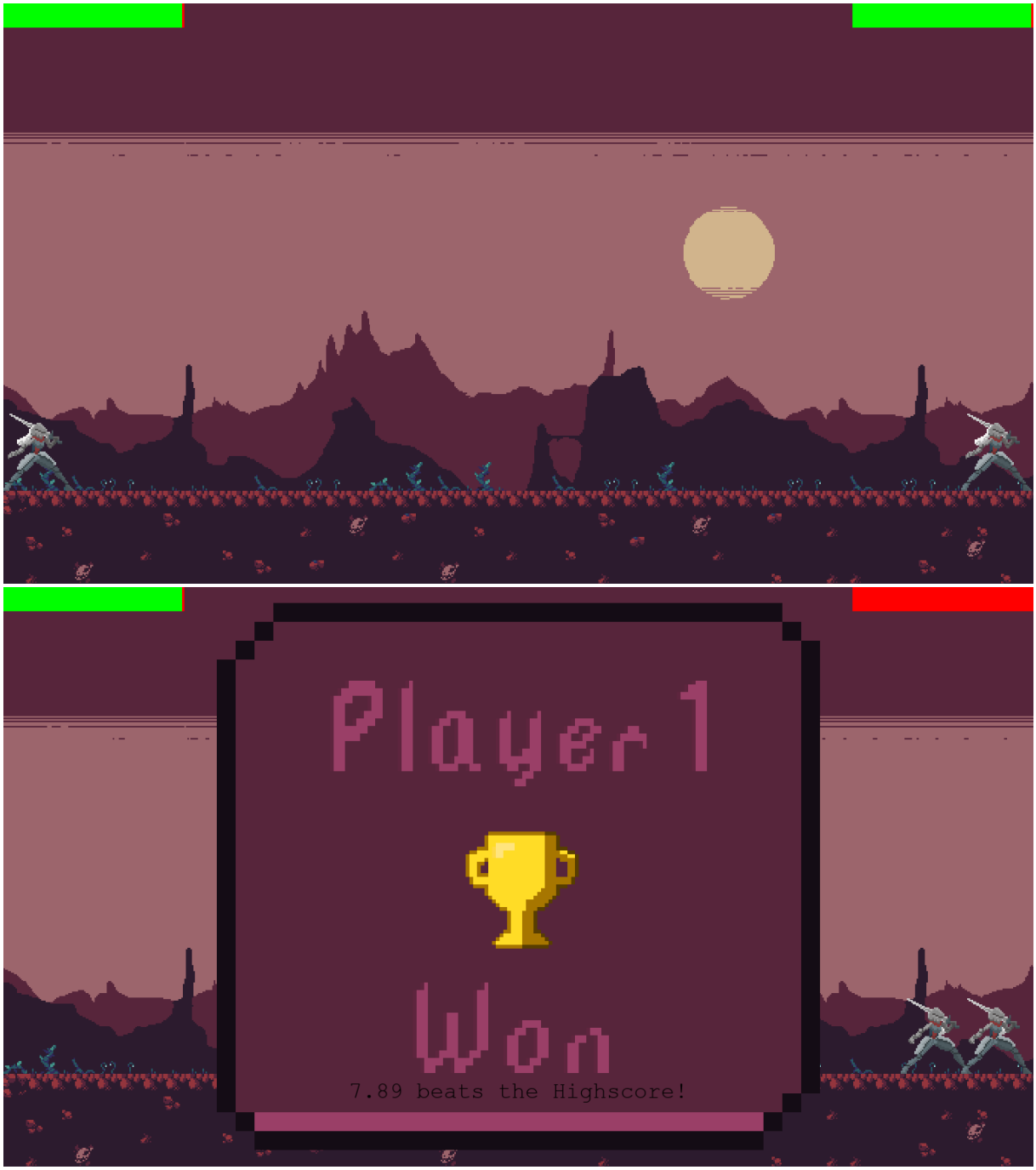
# Knight fight

**Projektgruppe:** Lia Louise Natter und Luis Kaufmann
**Klasse:** 1AHIF
**Jahr:** 2024


Player 1
Won
7.89 beats the Highscore!

**Betreuer:in:** Lukas Diem

**Kurzbeschreibung:**

Lokaler 1-gegen-1-Kampfspiel, das sowohl mit Controller als auch mit Tastatur gespielt werden kann. Es ist jedoch optimal für Controller ausgelegt, weshalb es etwas umständlich ist, zu zweit an einer Tastatur zu spielen.

Inhaltsverzeichnis

# 1 Projektzeitplan

## 1.1 Projektzeitplan: Lia Natter

| Datum | Aufgabe | Status (%) |
|---|---|---|
| 30.04 2024 | Ordner Struktur (Main.py ...) erstellen | 100% |
| 16.5 2024 | Player Class | 100% |
| 17.5 2024 | Links-Rechts-Bewegung | 100% |
| 21.5 2024 | Hintergrund erstellen | 100% |
| 22.5 2024 | Springen | 100% |
| 29.5 2024 | Gravitation | 100% |
| 4.6 2024 | Player Spawnpunkte | 100% |
| 12.6 2024 | Startmenü | 100% |
| 16.6 2024 | Deathscreen | 100% |

## 1.2 Projektzeitplan: Luis Kaufmann

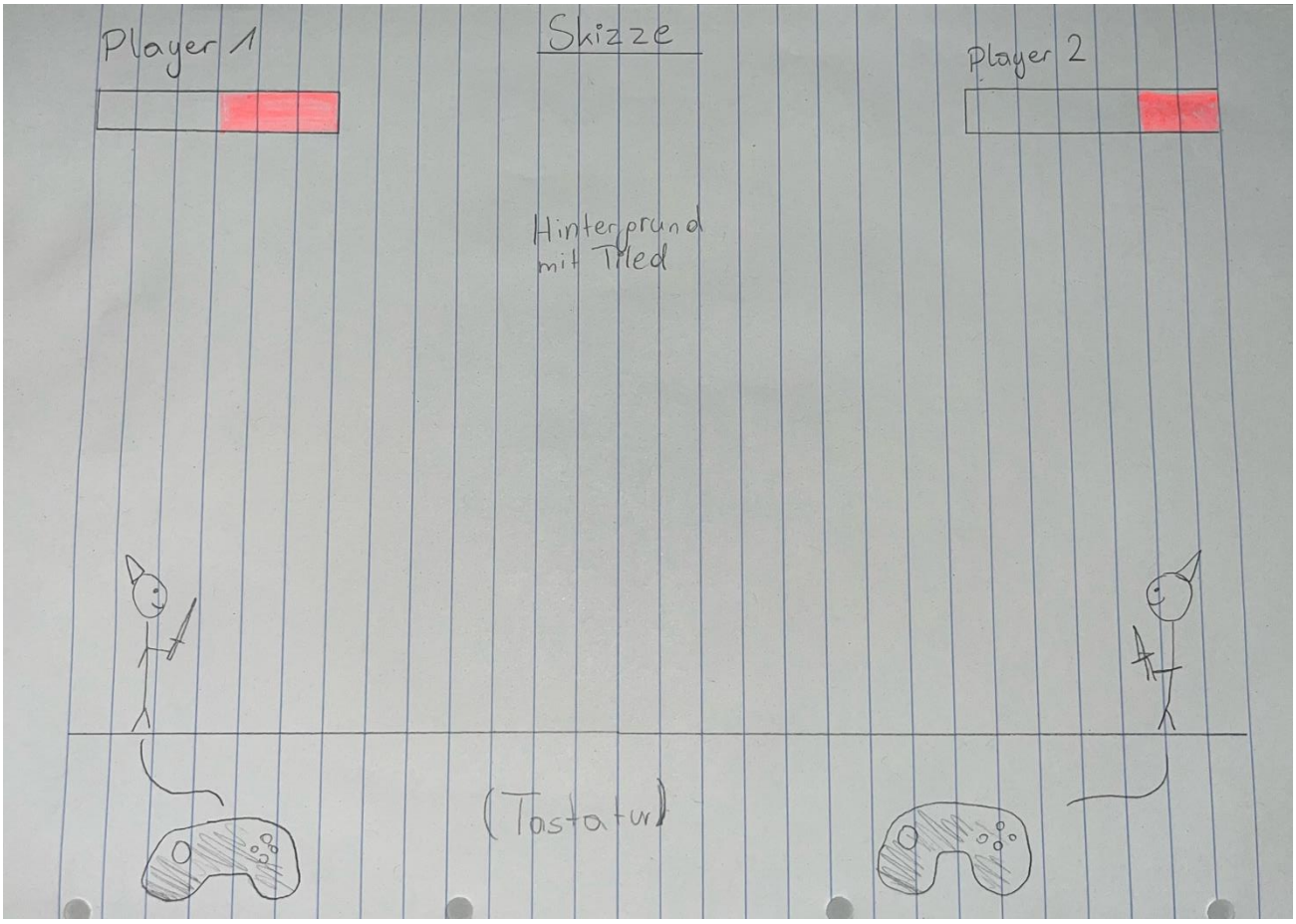
| Datum | Aufgabe | Status (%) |
|---|---|---|
| 30.4 2024 | Allgemeine Initialisierung VCS | 100% |
| 12.5 2024 | Player Class | 100% |
| 16.5 2024 | Start neuer Class für Spritsheets und ihre Animation | 100% |
| 21.5 2024 | Wechsel der Art wie die Player Class initialisiert wird und der Art wie Inputs verarbeitet werden | 100% |
| 22.5 2024 | Das Tmx File Verarbeitung | 100% |
| 06.06 2024 | Collision with use of Tmx | 100% |
| 11.06 2024 | Schlag funktion | 100% |
| 16.6 2024 | Animation | 100% |

# 2 Lastenheft (Kurzbeschreibung, Funktionsumfang, Skizzen)

## 2.1. Kurzbeschreibung

Lokaler 1-gegen-1-Kampfspiel, das sowohl mit Controller als auch mit Tastatur gespielt werden kann. Es ist jedoch optimal für Controller ausgelegt, weshalb es etwas umständlich ist, zu zweit an einer Tastatur zu spielen.

## 2.2. Skizzen



## 2.3. Funktionsumfang

Main.py

- Die Main (=Hauptfunktion) führt das File aus.

Player.py

- "**__init__**" initialisiert den Player und übernimmt die Bereitstellung aller erforderlichen Daten.
- **"x_movement"** verarbeitet alle Bewegungssignale, unabhängig davon, ob sie vom Controller oder den jeweiligen Tasten stammen.
- **"jump"** verarbeitet die Eingabe des Springens.
- Die Funktion **„Gravity"** sorgt dafür, dass der Spieler nach dem Springen wieder sicher auf dem Boden landet.
- **"Hitting"** ist dafür verantwortlich, dass ein Rect in die Richtung erstellt wird, in die gezeigt wird.
- **"Being_hit"** überprüft, ob das Rect aus "Hitting" den Player trifft. Falls dies der Fall ist, wird dem Spieler entsprechend Leben abgezogen. Zudem wird geprüft, ob der Spieler bereits tot ist.
- **"HP_bar"** erstellt ein Rechteck für die Lebensanzeige und ein weiteres für die Anzeige des Schadens. Die Größe des zweiten Rechtecks wird dabei so verändert, dass die Lebensanzeige dargestellt werden kann.
- Die Funktion **"Animat"** gewährleistet, dass die Animation in bestimmten Zeitabständen hier 140 Ticks Frame wechselt.
- **"Displayer"** zeichnet alle erforderlichen Daten auf den Bildschirm auf.
- **"Update"** führt alle erforderlichen Updates durch und führt alle Funktionen aus. Somit ist dies die einzige Funktion, die in main.py aufgerufen werden muss.

Map.py

- **“Map_lister”** extrahiert aus einem Tmx-File eine übersichtliche Liste und liefert diese einmal in der komplexen Version und einmal in der Liste zurück.

- **“Map_drawer”** übernimmt die Darstellung der Karte aus dem Tmx-File und zeichnet sie.

- **“Colider”** verarbeitet das Tmx-File erneut in eine Liste. Dabei werden nur die Rects berücksichtigt, die sich in einem bestimmten Layer befinden, hier **“floor”**. Anschließend werden aus diesen Rects neue Rects erzeugt. Dies ist eine praktische Lösung für Kollisionen, da die Map ohne weitere Änderungen verändert werden kann.

Spritsheet.py

-> Klasse zum Verarbeiten von Sprit Sheets, das macht die Animation sehr einfach.

- **“Frames”** nimmt die Maße auf und speichert alle Frames der Animation in der “frame_list”. Anschließend kann die Animation einfach animiert werden.

Menu.py

- **“File_reader”** liest den Highscore aus einem File aus.

- **“Deathscreen”** erscheint mit einer Aussage darüber, welcher Spieler gewonnen hat und ob der Highscore gebrochen wurde.

- **“Options Menu”** erzeugt das Options Menü.

- **“Main_menu”** erzeugt das Startmenu.

## 2.4. Must-Haves und Nice-To-Haves

Must-Haves:

- Steuerung mit Maus und Tastatur
- 2 Player
- Lifebar
- Mit Tiled arbeiten
- Minimales Menü

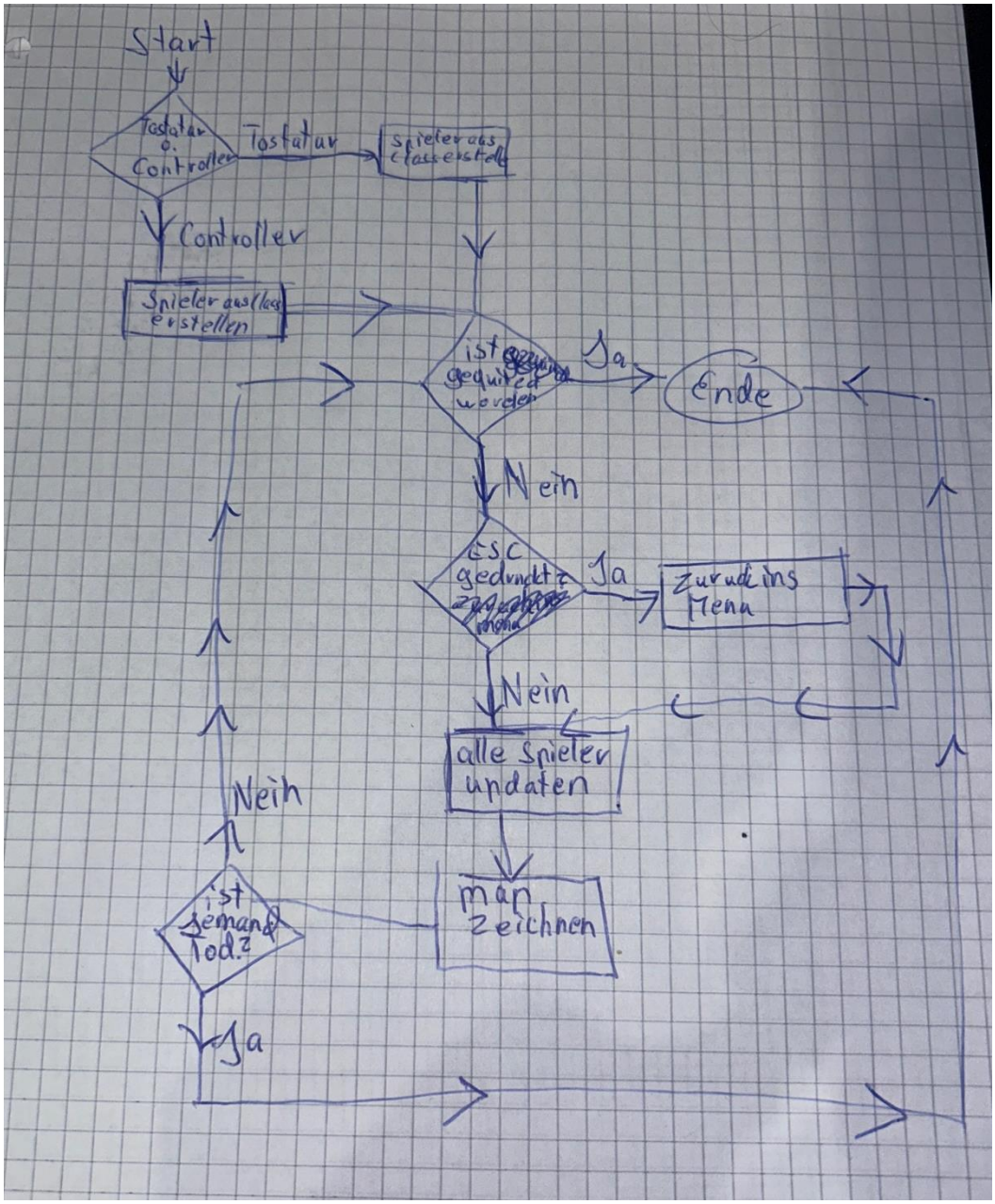
Nice-To-Haves:

- Erweitertes Menü (mehrere Optionen, ...)
- Animationen
- Highscore
- Musik

# 3 Pflichtenheft

## 3.1 Interner Programmaufbau (Programmlogik)

## 3.2 Flussdiagramm



## 3.3 Umsetzungsdetails

Nach jedem Durchlauf wird eine Prüfung auf die Verbindung eines Controllers durchgeführt. Bei einer erfolgreichen Verbindung wird ein neuer Player erstellt, der eine eigene Lebeleiste und alle erforderlichen Funktionen beinhaltet. Bei Auswahl der Option „Tastatur“ im Menü wird nicht auf bestehende Verbindungen geachtet, sondern direkt zwei Player erstellt, die dann mit den dazugehörigen Tasten gesteuert werden.

## 3.4 Ergebnisse, Interpretation (Tests)

Das Programm läuft fast einwandfrei, allerdings lässt sich der bereits erstellte Player nicht löschen. Das bedeutet, dass bei wiederholtem Ein- und Ausstecken einer Person für jedes Mal eine neue Figur erstellt wird.

# 4 Anleitung

## 4.1 Installationsanleitung

- Python 3.12
- Pygame 2.5.2
- PyTMX 3.32

Für Installation einfach in CMD pip install und dann der Name eingeben

## 4.2 Bedienungsanleitung

Sobald das Spiel gestartet wir erscheint das Startmenü bei dem als aller erstes unten ausgesucht werden kann, ob man mit Controllern oder mit Tastatur spielen möchte, standardmäßig ist aber Tastatur eingestellt


Wenn mit Controller gestartet wird, kann er jetzt eingesteckt werden. Wenn die Controller bereits eingesteckt gewesen sein sollten, müssen sie jetzt noch einmal ausgesteckt und wieder eingesteckt werden. Ab diesem Zeitpunkt kann die Nutzung beginnen.

Da das Button-Layout bei Controllern verschieden ist, wird hier als Beispiel der Xbox-Controller verwendet. Mit dem linken Joystick wird gesprungen, mit B wird geschlagen und mit X wird ein Gegenstand aufgenommen. Mit dem Optionsknopf kann jederzeit in das Startmenü zurückgekehrt werden.

Bei der Tastatur ist das Vorgehen anders. Es ist abhängig davon, ob man Spieler 1 oder Spieler 2 ist. Spieler 1 bewegt sich mit WASD, wobei er mit W springt und mit C schlägt.

Spieler 2 hingegen bewegt sich mit IJKL, springt mit I und schlägt mit N.

# 5 Bekannte Bugs, Probleme

Spieler können nicht gelöscht werden, das hätte mehr Zeit gebraucht.

Der Knopf ist von der Controller-Marke abhängig. Das heißt, es ist für jeden spielbar, aber mit verschiedenen Knöpfen.

# 6 Quellen

tileset : https://quintino-pixels.itch.io/wasteland-plataformer-tileset
player : https://luizmelo.itch.io/hero-knight
keyboaard buttons : https://dreammix.itch.io/keyboard-keys-for-ui#google_vignette
controller buttons : https://zrghr.itch.io/the-buttons