



## Prototyping interaktiver Medien-Apps und Games

Wintersemester 2020/2021

vorgelegt von: Thyra Hultsch [263162]

eingereicht am: 19. Februar 2021

Betreuender Dozent: Prof. Jirka Dell'Oro-Friedl

# Inhaltsverzeichnis

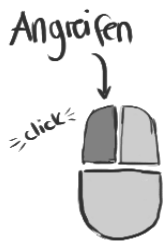
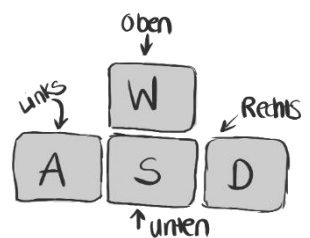
1.	Kurzbeschreibung und Gameplay.....	3
2.	Steuerung und Nutzerinteraktion.....	3
3.	Objektinteraktionen .....	3
4.	Objektanzahl variabel .....	4
5.	Szenenhierarchie .....	5
6.	Externe Daten.....	5
7.	Verhaltensklassen und Subklassen .....	5
8.	Maße und Proportionen .....	6
9.	Event-System.....	7
10.	Assets .....	7
11.	Sounds.....	10
12.	Checkliste/Zusammenfassung .....	12
13.	Quellenverzeichnis .....	14

## 1. Kurzbeschreibung und Gameplay

Der Avatar namens Thyra befindet sich in einer Arena, in welcher Geister der Vergangenheit erscheinen und den Avatar angreifen möchten. Die Spielfigur verfügt über eine Waffe, um sich gegen ihre Feinde zu wehren. Das Ziel ist es, möglichst viele Monster zu töten, Münzen zu sammeln und nicht zu sterben. Hat der Spieler 3 Leben verloren, verliert er das Spiel.

## 2. Steuerung und Nutzerinteraktion

Der Avatar kann durch die Benutzung der WASD-Tasten bewegt und gesteuert werden. Dadurch kann er vor Geistern fliehen, muss jedoch darauf achten, nicht mit Hindernissen wie zum Beispiel Bäumen, zu kollidieren.



Mithilfe der Maustaste kann der Avatar angreifen und somit einen „Rundumschlag“ ausführen, wobei er seine Gegner im Umkreis verletzen oder umbringen kann. Münzen kann der Avatar durch darüberlaufen einsammeln. Weitere Einstellungen wie zum Beispiel die Schwierigkeit seiner Gegner oder das Level kann der Nutzer im Menü bestimmen.

## 3. Objektinteraktionen

Zwischen dem Avatar und seiner Umgebung findet eine Vielzahl von Kollisionen statt. Manche davon sind für den Nutzer der Applikation sichtbar, indem sich der Avatar bei einer Kollision bewegt oder bestimmte Sounds abgespielt werden.

Das Grundprinzip der Kollision basiert auf der Überprüfung von Überschneidungen, also Intersections. Gibt es bei einer Überprüfung keine Überschneidungen zwischen dem Avatar und einem anderen Game Object ( $\text{intersection} == \text{null}$ ), gibt die Methode false zurück und spätere If-Abfragen werden zum Beispiel nicht ausgeführt. Findet eine Kollision statt, ist dieser jedoch vom Wert „true“.



## Fälle der Kollision:

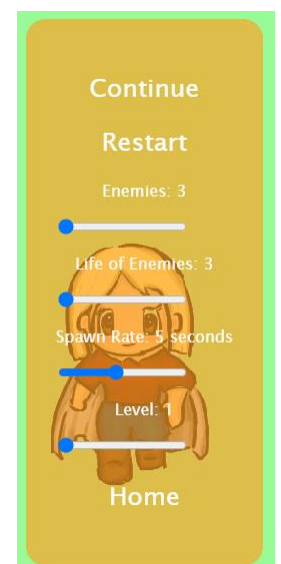
- Avatar und Hindernis/Außenbegrenzung:  
Findet eine Kollision zwischen dem Avatar und einem Hindernis statt, wird überprüft, von welcher Seite er auf das Objekt eintrifft.  
Anschließend wird die Position des Avatars, seines Rects und des AttackRadius um 0,2 Einheiten verschoben. Schlägt dies fehl und der Avatar schafft es, sich auf/in einem Objekt zu bewegen, wird dieser automatisch vom Hindernis heruntergeschoben.
- Avatar und Enemy:  
Eine Kollision von Avatar und Enemy wird interessant, sobald einer von ihnen angreifen möchte. Attackiert der Avatar den Enemy wird überprüft, ob der AttackRadius des Avatars eine Überschneidung mit dem Rect des Enemys hat. Ist dies der Fall, wird dem Enemy ein Leben abgezogen. Falls der Enemy nah genug an den Avatar heran gelangt und sich die Rects der beiden Game Objects überschneiden, wird dem Spieler ebenfalls ein Leben abgezogen.
- Avatar und Coin:  
Gibt es eine Überschneidung zwischen dem Rect des Avatars und dem Rect des Coins, verschwindet der Coin und die Anzahl der gesammelten Münzen erhöht sich und wird im User Interface aktualisiert dargestellt.
- Enemy und Hindernis:  
Hierbei findet keine Kollision statt. Die Enemies sind ja schließlich Geister.

## 4. Objektanzahl variabel

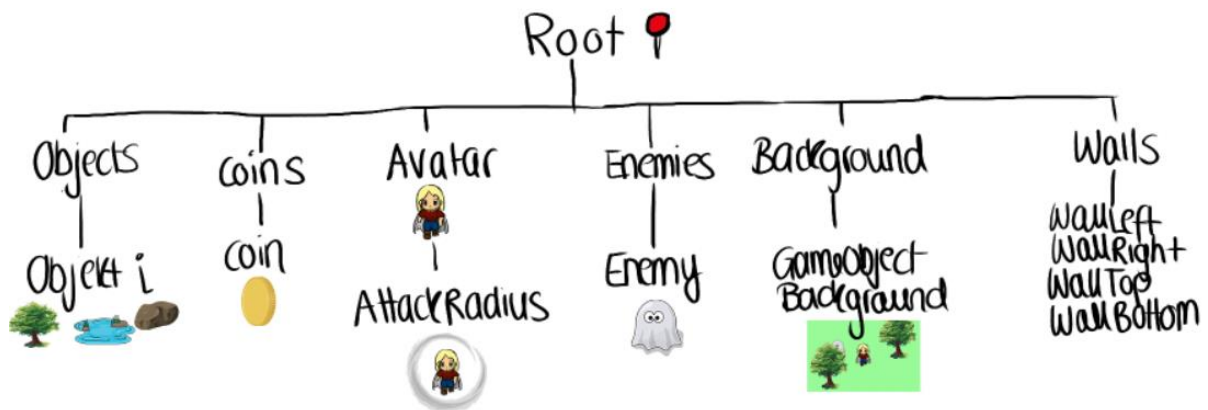
Die Objektanzahl kann in den Aspekten „Anzahl der Enemies“, „Leben der Enemies“, „Spawn Rate“ und dem Level zur Laufzeit im Menü verändert werden.

Durch Slider kann die Anzahl der Enemies, welche sich gleichzeitig auf dem Spielfeld befinden können, festgelegt werden. Der Default- und Minimalwert beträgt hierbei 3, der Maximalwert 10. Außerdem kann man so auch die Spawn Rate, also in welchem zeitlichen Abstand Enemies erzeugt werden, einstellen (Minimum: 1 Sekunde, Default: 5 Sekunden, Maximum: 10 Sekunden). Durch „Life of Enemies“ können minimal 3 und maximal 10 Leben für einen Geist festgelegt werden.

Im Code werden die Input-Werte der Slider ausgelesen und anschließend mithilfe eines EventListeners vom Typ „change“ interpretiert.



## 5. Szenenhierarchie



## 6. Externe Daten

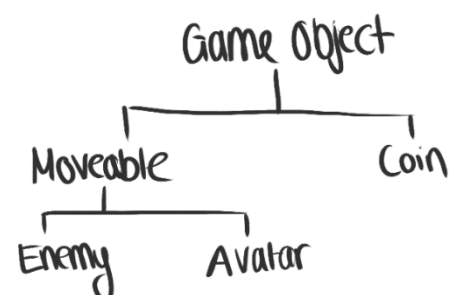
Externe Daten werden bei The Legend of Thyra verwendet, um den Aufbau des Levels zu variieren. Dabei befinden sich in einer JSON-Datei alle Hindernisse, welche pro Level verwendet werden. Diese werden mit dem Typen des Hindernisses, der X- und Y-Position und der Größe beschrieben.

In der Main.ts wird die Datei mithilfe einer fetch Funktion eingelesen, anschließend verarbeitet und in einem Array gespeichert. In der createLevel Methode werden die Objekte aus dem Array interpretiert und anhand des Typen eingeteilt und erstellt.

So können Level auch später noch hinzugefügt, einfach geändert und erweitert werden.

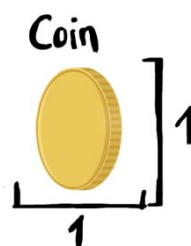
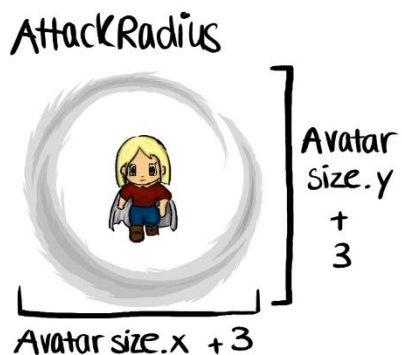
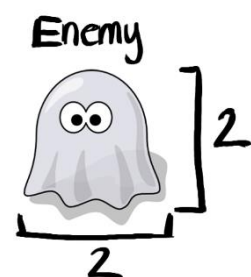
## 7. Verhaltensklassen und Subklassen

- GameObject
  - swapMaterial: Wechselt das Material von einem angegriffenen Enemy
  - swapAvatarMaterial: Wechselt das Material vom Avatar bei einer Bewegung
- Moveable
  - move und translate bewegen Enemies

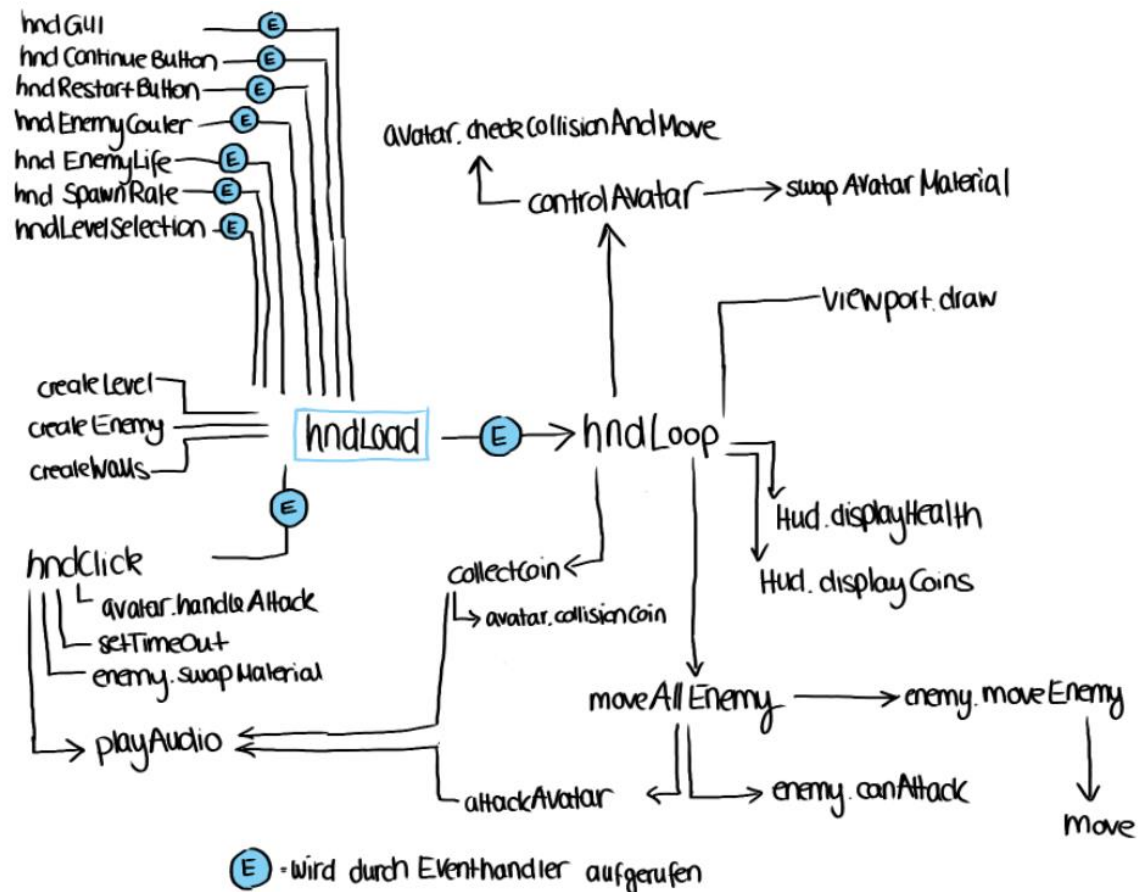


- Avatar
  - checkCollisionAndMove: auf Kollision mit Hindernissen überprüfen und bewegen
  - handleAttack: auf Kollision mit Enemy überprüfen
  - collisionCoin: auf Kollision mit Münze überprüfen
- Enemy
  - moveEnemy: bewegt Enemy in Richtung Avatar
  - canAttack: überprüfen, ob Enemy den Avatar angreifen kann
- Coin
- Hud
  - displayHealth: Anzahl der Leben des Avatars im User Interface anzeigen
  - displayCoins: Anzahl der gesammelten Münzen im User Interface anzeigen

## 8. Maße und Proportionen



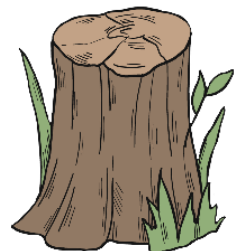
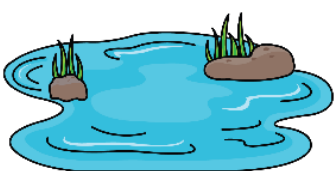
## 9. Event-System



## 10. Assets

### Hindernisse und Spielfeld

Das Spielfeld wird aus der Vogelperspektive angezeigt und stellt einen Ausschnitt einer Wiese mit vielen verschiedenen Naturelementen dar. Die Hindernisse dienen dazu, dem Spieler Wege abzuschneiden und das fliehen vor den Geistern zu erschweren. Zu den Hindernissen zählen Bäume, Steine, Hecken, Baumstämme oder Teiche. Die Größe, Position, und Textur wird in der jeweiligen JSON-Datei eines Levels gespeichert und in der Main.ts ausgelesen und interpretiert.







## Avatar

Der Avatar names Thyra befindet sich auf dem Spielfeld und wird dort zu Beginn des Spieles in dessen Mitte gespawnt. Die Aufgabe des Avatar ist es, möglichst viele Geister zu töten und die von ihnen fallengelassenen Münzen aufzusammeln. Dafür wird ein Rundumschlag verwendet, welcher durch eine kreisförmig aussehende Textur angezeigt wird.

Der Avatar wird abhängig von der Steuerung des Nutzers, aus verschiedenen Positionen angezeigt. Er kann von Geistern angegriffen werden und verliert mit jeden Schlag eines der insgesamt 3 Leben. Die Anzahl der Leben und der Münzen wird im User Interface angezeigt.



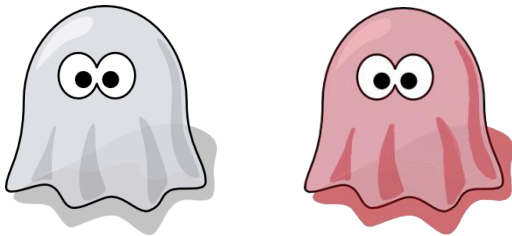
## Enemy

Die Geister der Vergangenheit versuchen Thyra, den Avatar immer wieder aufs neue anzugreifen. Dabei werden sie von keinen Objekten aufgehalten und können diese einfach überwinden. Ihr Ziel ist es, den Avatar anzugreifen, sodass dieser alle Leben verliert und das Spiel beendet wird. Die Geister werden dabei in der linken, oberen Ecke des Spielfeldes gespawnt und kommen von dort aus auf den Avatar zu.



Kommen sie dem Avatar zu nahe, verliert dieser ein Leben. Nachdem ein Geist angegriffen hat, besitzt er einen Angriffs-Cooldown von 3 Sekunden. Ist dieser abgelaufen, kann der Avatar erneut angegriffen werden.

Die Geister besitzen ebenfalls 3 Leben, diese Anzahl kann jedoch variiert werden. Wird der Geist getroffen wechselt sein Aussehen zu einer roten Version der originalen Textur. Die Geister verlieren außerdem mit einer Wahrscheinlichkeit von 50% eine Münze.



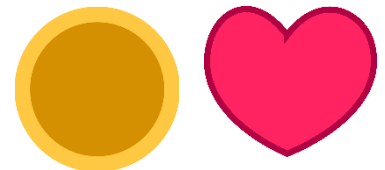
## Münzen

Eine Münze erscheint mit einer Wahrscheinlichkeit von 50%, wenn ein Geist getötet wurde. Sie wird an der ehemaligen Position des Geistes erzeugt und kann dort vom Avatar durch darüberlaufen eingesammelt werden. Die Anzahl der gesammelten Münzen wird ebenfalls im User Interface angezeigt. Die Münze ist das kleinste Spielelement und somit 1x1 Einheiten groß.



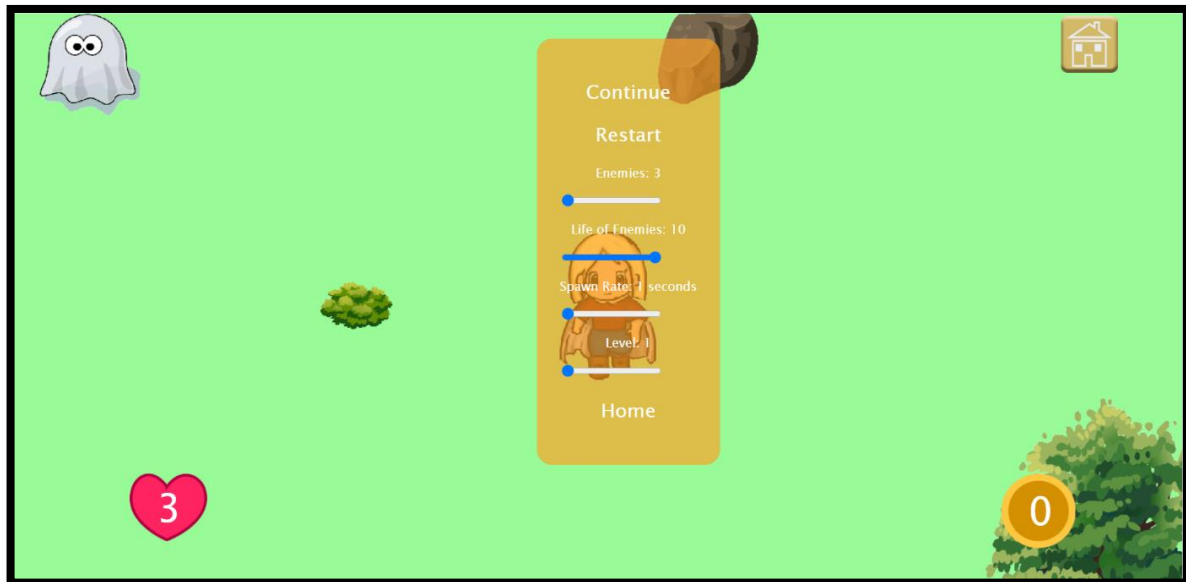
## User Interface

Im Userface werden die wichtigsten Informationen über den Avatar angezeigt: seine Gesundheit und seine gesammelten Münzen. Diese Anzeigen befinden sich in den jeweiligen unteren Ecken.



Im oberen rechten Eck hat der Nutzer die Möglichkeit das Spiel zu pausieren. Klickt er auf das Menü-Feld, öffnet sich ein Menü, in welchen man verschiedene Einstellungen ändern kann. Das Spiel kann von hier aus mit einem Klick auf das Menü-Feld oder mit der Return Funktion fortgesetzt werden oder auch mithilfe von Restart neu gestartet werden. Außerdem kann hier die Schwierigkeit der Gegner durch verschiedene Faktoren beeinflusst werden. Dazu werden Slider verwendet, um die gewünschte Anzahl von Gegnern, Leben der Gegner und die Spawn Rate einzustellen. Um das Level zu ändern kann der Slider verändert werden, anschließend muss das Spiel neu gestartet werden. Außerdem kann man von hier aus auf den Startbildschirm zurückkehren.





## 11. Sounds

### Münzen aufsammeln

Läuft der Avatar über eine Münze, sammelt er diese auf und erhält ein auditives Signal in Form eines Retro-Sounds. Dieser wird vom Nutzer sofort mit dem Aufsammeln einer Münze assoziiert.

### Enemy-Hit und Enemy-Tod

Greift der Avatar einen Geist an, erscheint neben einer anderen Geister-Textur auch ein Sound. Hat der Enemy kein Leben mehr, verschwindet er und ein weiterer Sound wird abgespielt.

### Avatar-Hit

Wird ein Avatar von einem Geist erfolgreich angegriffen, ertönt der Sound eines Punches.

### Avatar-Angriff

Greift der Avatar einen Geist an, wird dies neben dem visuellen Zeichen des Kreises um den Avatar durch das Geräusch eines Schwert-Schwunges unterstützt.

### Hintergrundmusik

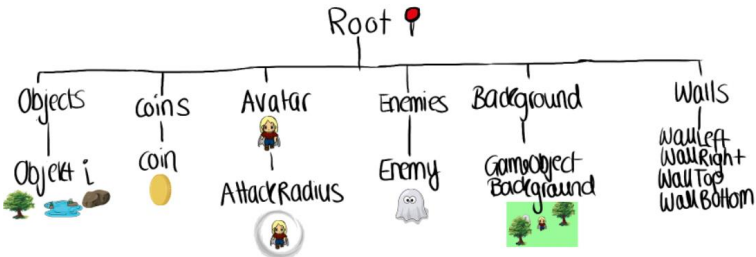
Zu Beginn des Spiels wird im Hauptbildschirm eine ruhige und freundliche Musik abgespielt. Startet der Nutzer nun ein Spiel, wechselt sie zu einem actionreichen und fesselnden 8-bit-Retro-Track.

*Achtung: Aufgrund der Google Chrome Policy kann beim Starten einer Applikation kein Autoplay von Audio und Video stattfinden. Möchte man die wunderschöne Musik trotzdem hören kann man dies tun, indem man ein Spiel startet und sich anschließend vom Pausenmenü aus sich zum Startbildschirm zurücknavigiert.*

## **Game Over**

Wird der Avatar drei Mal von einem Geist getroffen ertönt ein Game Over Sound und das Spiel kann neu gestartet werden.

## 12. Checkliste/Zusammenfassung

Nutzerinteraktion	<ul style="list-style-type: none"> <li>– Steuerung des Spielers durch WASD (Bewegung) und Klicken (Angreifen),</li> <li>– Benutzung des GUIs <ul style="list-style-type: none"> <li>• Spiel fortsetzen</li> <li>• Spiel neu starten</li> <li>• Anzahl der Enemies einstellen</li> <li>• Leben der Enemies einstellen</li> <li>• Spawn Rate der Enemies einstellen</li> <li>• Level auswählen</li> <li>• Zurück zum Hauptmenü</li> </ul> </li> </ul>
Objektinteraktion	<ul style="list-style-type: none"> <li>– Kollision bei Spielrandbegrenzung, und Objekten wie Bäume, Hecken und Steine usw.</li> <li>– Angriffe auf Gegner</li> <li>– Gegnerische Angriffe auf Avatar</li> <li>– Münzen sammeln</li> </ul>
Objektanzahl variabel	<ul style="list-style-type: none"> <li>– Variable Objekte:</li> <li>– Anzahl der gespawnten Enemies</li> <li>– Anzahl der Leben der Enemies</li> <li>– Spawn Rate der Enemies</li> </ul>
Szenenhierarchie	 <pre> graph TD     Root[Root] --&gt; Objects     Root --&gt; Coins     Root --&gt; Avatar     Root --&gt; Enemies     Root --&gt; Background     Root --&gt; Walls     Objects --&gt; Object1[Object 1]     Coins --&gt; Coin     Avatar --&gt; AttackRadius     Enemies --&gt; Enemy     Background --&gt; GameobjectBackground[Gameobject-Background]     Walls --&gt; WallLeft     Walls --&gt; WallRight     Walls --&gt; WallTop     Walls --&gt; WallBottom     </pre>
Sound	<ul style="list-style-type: none"> <li>– Gegner wird verletzt</li> <li>– Gegner stirbt</li> <li>– Spielfigur wird verletzt</li> <li>– Angreifen (Schwertgeräusch)</li> <li>– Hintergrundmusik im Startbildschirm und während des Spielens</li> <li>– Münzen aufsammeln</li> <li>– Game Over</li> </ul>
GUI	<ul style="list-style-type: none"> <li>– Leben der Spielfigur</li> <li>– Gesammelte Münzen</li> <li>– Einstellungen/Pause <ul style="list-style-type: none"> <li>• Spiel fortsetzen</li> <li>• Spiel neu starten</li> <li>• Anzahl der Enemies einstellen</li> <li>• Leben der Enemies einstellen</li> <li>• Spawn Rate der Enemies einstellen</li> <li>• Level auswählen</li> <li>• Zurück zum Hauptmenü</li> </ul> </li> </ul>

Externe Daten	<ul style="list-style-type: none"> <li>– Levelaufbau als JSON-Datei</li> </ul>
Verhaltensklassen	<ul style="list-style-type: none"> <li>– Avatar <ul style="list-style-type: none"> <li>• checkCollisionAndMove</li> <li>• handleAttack</li> <li>• collisionCoin</li> </ul> </li> <li>– Enemy <ul style="list-style-type: none"> <li>• moveEnemy</li> <li>• canAttack</li> </ul> </li> <li>– GameObjects <ul style="list-style-type: none"> <li>• swapMaterial</li> <li>• swapAvatarMaterial</li> </ul> </li> <li>– Hud <ul style="list-style-type: none"> <li>• displayHealth</li> <li>• displayCoins</li> </ul> </li> <li>– Moveable <ul style="list-style-type: none"> <li>• Move</li> <li>• Translate</li> </ul> </li> <li>– Coins</li> </ul>
Subklassen	<ul style="list-style-type: none"> <li>– GameObject <ul style="list-style-type: none"> <li>• Moveable <ul style="list-style-type: none"> <li>○ Enemy</li> <li>○ Avatar</li> </ul> </li> <li>• Coin</li> </ul> </li> </ul>
Maße und Positionen	<ul style="list-style-type: none"> <li>– Münze (1x1)</li> <li>– Enemy (2x2)</li> <li>– Avatar (2x3)</li> <li>– AttackRadius (5x6)</li> <li>– Spielfeld (39x25)</li> </ul>
Event-System	<p>The diagram illustrates the event system architecture. On the left, a list of event handlers (E) is shown: hndGui, hndContinueButton, hndRestartButton, hndEnemyCounter, hndEnemyLife, hndSpawnRate, and hndLevelSelection. These are connected to a central 'hndLoad' box. Below 'hndLoad', there are three event triggers: 'createLevel', 'createEnemy', and 'createWalls'. These triggers lead to a 'hndClick' box, which then branches into 'avatar.handleAttack', 'setTimeOut', and 'enemy.swapMaterial'. These actions lead to 'playAudio'. From 'playAudio', the flow goes to 'collectCoin', which then triggers 'avatar.collisionCoin'. This leads to 'moveAllEnemy', which then triggers 'enemy.canAttack'. This leads to 'attackAvatar', which then triggers 'avatar.checkCollisionAndMove'. This leads to 'controlAvatar', which then triggers 'swapAvatarMaterial'. Finally, 'swapAvatarMaterial' leads to 'viewport.draw'. The diagram also shows 'Hud.displayHealth' and 'Hud.displayCoins' being triggered from the 'hndLoop' box. A legend at the bottom indicates that a blue circle with an 'E' represents an event handler.</p> <p>• wird durch Eventhandler aufgerufen</p>

## 13. Quellenverzeichnis

### Assets

**Avatar + Schlaganimation:** Selbst gezeichnet

**Münze:** <https://www.pinterest.de/pin/766245324089410087/>

**Enemy:** <https://de.cleanpng.com/png-jhkzil/>

**Hindernisse:**

- **Baum:** [https://pngtree.com/freepng/dense-tree-cartoon-illustration\\_4595420.html](https://pngtree.com/freepng/dense-tree-cartoon-illustration_4595420.html)
- **Stein 1:** [https://www.pngitem.com/middle/hRixhhT\\_transparent-cartoon-stone-hd-png-download/](https://www.pngitem.com/middle/hRixhhT_transparent-cartoon-stone-hd-png-download/)
- **Stein 2:** [https://pngtree.com/freepng/grass-stone\\_3816494.html](https://pngtree.com/freepng/grass-stone_3816494.html)
- **Teich:** <https://www.cleanpng.com/png-drawing-clip-art-image-cartoon-pond-6801403/preview.html>
- **Hecke:** [https://pngtree.com/freepng/tree-for-cartoon-isolated-on-white-background\\_4846991.html](https://pngtree.com/freepng/tree-for-cartoon-isolated-on-white-background_4846991.html)
- **Baumstumpf:** <https://creazilla.com/nodes/12933-tree-stump-clipart>

**User Interface:**

- **Menü:** <https://simple-wares.itch.io/gui-65-button-asset-pack?download>
- **Health:** <https://www.pinclipart.com/maxpin/TwmJb/>
- **Coins:** Selbst gezeichnet

### Sounds

**Münzen aufsammeln:** <https://www.zapsplat.com/music/retro-8-bit-video-game-coin-collect-or-pick-up-4/>

**Enemy-Hit:** <https://www.zapsplat.com/music/ghost-voice-evp-eerie-fearful-breathing-4/>

**Enemy-Tod:** <https://www.zapsplat.com/music/large-horror-ghostly-breath-with-reverb-4/>

**Avatar-Hit:** <https://www.zapsplat.com/music/designed-fist-punch-thump-hit-impact-with-vocal-1/>

**Avatar-Angriff:** <https://www.zapsplat.com/music/sword-swing-fast-whoosh-in-air-with-metal-sound-of-blade-swiping-version-5/>

**Hintergrundmusik Startbildschirm:** <https://www.purple-planet.com/cute>

**Hintergrundmusik Spiel:** <https://www.zapsplat.com/music/game-music-action-retro-8-bit-style-bouncy-hard-dance-track-with-electronic-synths-and-drums/>

**Game Over:** <https://www.zapsplat.com/music/8-bit-game-over-80s-arcade-simple-alert-notification-for-game-1/>