



# Wrath Of The Fallen Queen - Documentation

## ▼ Tabelle

Aufgabe	Anfang	Ende
Themenfindung	13.02.2024	13.02.2024
Anpassung der Tilemap	14.02.2024	—
Wissen für JS Klassen	15.02.2024	15.02.2024
Anpassung der Tilemap - Information für 2D Collider	19.02.2024	19.02.2024
Block Collider - Zurücksetzung der Position	20.02.2024	—
Konzeptualisierte Hintergrund-Gesichte	21.02.2024	21.02.2024
Allgemeines Level Design	21.02.2024	21.02.2024
Konzept für das erste Level	21.02.2024	21.02.2024
Ausarbeitung des ersten Levels	22.02.2024	—

Digitale Map	22.02.2024	22.02.2024
Dash Prototyp	22.02.2024	22.02.2024
Level Design - Level 1-1, 1-2	23.02.2024	---
Level Design - Level 1-2, 1-3	26.02.2024	---
Level Design - Level 1-3	27.02.2024	---
Anpassung des Dashes	27.02.2024	27.02.2024
Media Query	27.02.2024	27.02.2024
Zusammensetzung aller Dateien	28.02.2024	28.02.2024
CSS für Herzen	28.02.2024	28.02.2024
Finalisierung Level 1	29.02.2024	29.02.2024
Konzeptdarstellung Level 2	01.03.2024	01.03.2024
Level Design - Level 2	04.03.2024	14.03.2024
Jumppad	05.03.2024	05.03.2024
Level Design - Level 2-1	05.03.2024	05.03.2024
Jumppad Überarbeitung	06.03.2024	06.03.2024
HTML-CSS Überarbeitung	07.03.2024	07.03.2024
HTML-CSS-JS Überarbeitung	08.03.2024	08.03.2024
HTML-CSS-JS Überarbeitung	11.03.2024	11.03.2024
Zusammenführung der Daten	12.03.2024	12.03.2024
Level Design - Level 2-2	13.03.2024	13.03.2024
Multiple Air Jumppad	13.03.2024	13.03.2024
Moving Damage Tile	13.03.2024	13.03.2024
Level Design - Level 2-3	13.03.2024	13.03.2024
Jumppad Handle Input	14.03.2024	14.03.2024
Coins	14.03.2024	14.03.2024
Healing Animation	14.03.2024	14.03.2024

Collisioncheck Update	15.03	15.03
Bewegende Spikes	19.03	19.03
Map Loading	20.03	20.03
Airjump pad Sprites Redesign	21.03	21.03
Coin Reset	21.03	21.03
Verfeinerung der Maps	21.03	21.03
Zusammensetzung der Maps	22.03	22.03
Überführung des Finalcodes	26.03	26.03
Hintergrund Anpassung	26.03	26.03
NPCs entfernen	26.03	26.03
Generelle Anpassung der Level	26.03	26.03
Fixierung der UI	26.03	26.03
Checkpoints	27.03	27.03
Game End Screen	27.03	27.03
Hauptmenü Button	27.03	27.03

## ▼ 13.02 - Themenfindung, Ideendarstellung & Programmcode

Setting des Spiels und Genre

>> Ägyptische Mythologie, Action Adventure.

Überarbeitung des Programmcodes nach Basis vom Unterrichtsmaterial jumpnrun.html

## ▼ 14.02 - Anpassung der Tilemap

Ansatz der Tilemap-Funktion durch einzelne PNGs statt Tileset.

>> Gedanke war weniger Rechenoperationen zu Programmieren.

Aufbau der Tilemap nach Basis vom Unterrichtsmaterial > Überarbeitung in eine Variante basierend aus Quellen im Internet und YouTube.

## ▼ **15.02- Aneignung von Wissen**

Grundlegende Informierung über Klassen in JavaScript für einen zukünftig leichteren Ablauf der Programmierung.

## ▼ **19.02 - Anpassung der Tilemap - Information für 2D Collider**

Rückfuhr auf Tilesets statt einzelne PNGs. > Einzelne PNGs würden doch mehr Zeit in Anspruch nehmen, als die Berechnung aus dem Tileset.

```
// Deklaration einer Konstante für einen Canvas
const canvas = document.getElementById('game');
// Deklaration einer Konstante getContext('2d') für canvas
const ctx = canvas.getContext('2d');
// Größe eines Tiles
const tile = 40;

// Initialisierung der .pngs
const background = new Image();
background.src = './img/background.png';

const tileset = new Image();
tileset.src = './img/tileset.png';

const playerSprite = new Image();
playerSprite.src = './img/player.png';

canvas.width = 1280;
canvas.height = 720;

const canHeight = canvas.height;
// Deklaration der Gravitation
const gravity = 0.5;

// Neuen Spieler erstellen
const player = new Player({
    position: {
        x: 100, y: 400
    }
});
```

Information über 2D Collider in JS durch YouTube, Stackoverflow, ChatGPT und Gemini.

## ▼ 20.02 - Block Collider - Zurücksetzung der Position

Einführung der Colliders mit Rücksetzung der Position, wenn der Spieler über die Position des Colliders hinausgeht.

(Verbesserungen werden noch vorgenommen)

>> Probleme mit Zurücksetzung der Position des Spielers. Der Spieler kann an der Wand klettern, durch die falsche Berechnung der Position und überlappt sich um wenige Pixel im Collider.

```

}

// Update-Methode zum ermitteln der neuen Position
update() {
    this.draw();
    // Nächste Position des Spielers berechnen
    const nextX = this.position.x + this.velocity.x;
    const nextY = this.position.y + this.velocity.y;
    // Flag für ob Spieler auf dem Boden ist = false
    let onGround = false;

    // Durch die Tiles der Map gehen um nach Kollisionen zu prüfen
    for (let y = 0; y < map.length; y++) {
        for (let x = 0; x < map[0].length; x++) {
            const texture_index = map[y][x];
            if (texture_index >= 0 && texture_index <= 7) {
                // Collision der aktuellen Blocker berechnen
                const tileX = x * tile;
                const tileY = y * tile;
                const tileRight = tileX + tile;
                const tileBottom = tileY + tile;

                // Prüfen ob der Spieler mit einem Blocker kollidiert
                if (
                    nextX < tileRight &&
                    nextX + this.width > tileX &&
                    nextY < tileBottom &&
                    nextY + this.height > tileY
                ) {
                    // Berechnung des Zentrums vom Spieler und Teil zur reposition
                    const playerCenterX = this.position.x + this.width / 2;
                    const playerCenterY = this.position.y + this.height / 2;
                    const tileCenterX = tileX + tile / 2;
                    const tileCenterY = tileY + tile / 2;

                    // Berechne die Entfernung von den beiden Zentren
                    const distanceX = playerCenterX - tileCenterX;
                    const distanceY = playerCenterY - tileCenterY;

                    // Berechne die Kombinierte "Halb"-Länge und -höhe
                    const combinedHalfWidths = this.width / 2 + tile / 2;
                    const combinedHalfHeights = this.height / 2 + tile / 2;

```

```
        if (!onGround) {
            this.velocity.y += gravity;
        }

        this.position.x += this.velocity.x;
        this.position.y += this.velocity.y;
    }
}
```

▼ 21.02- Konzeptualisierte Hintergrund-Geschichte

# Konzeptualisierung der Gesichte des Spiels

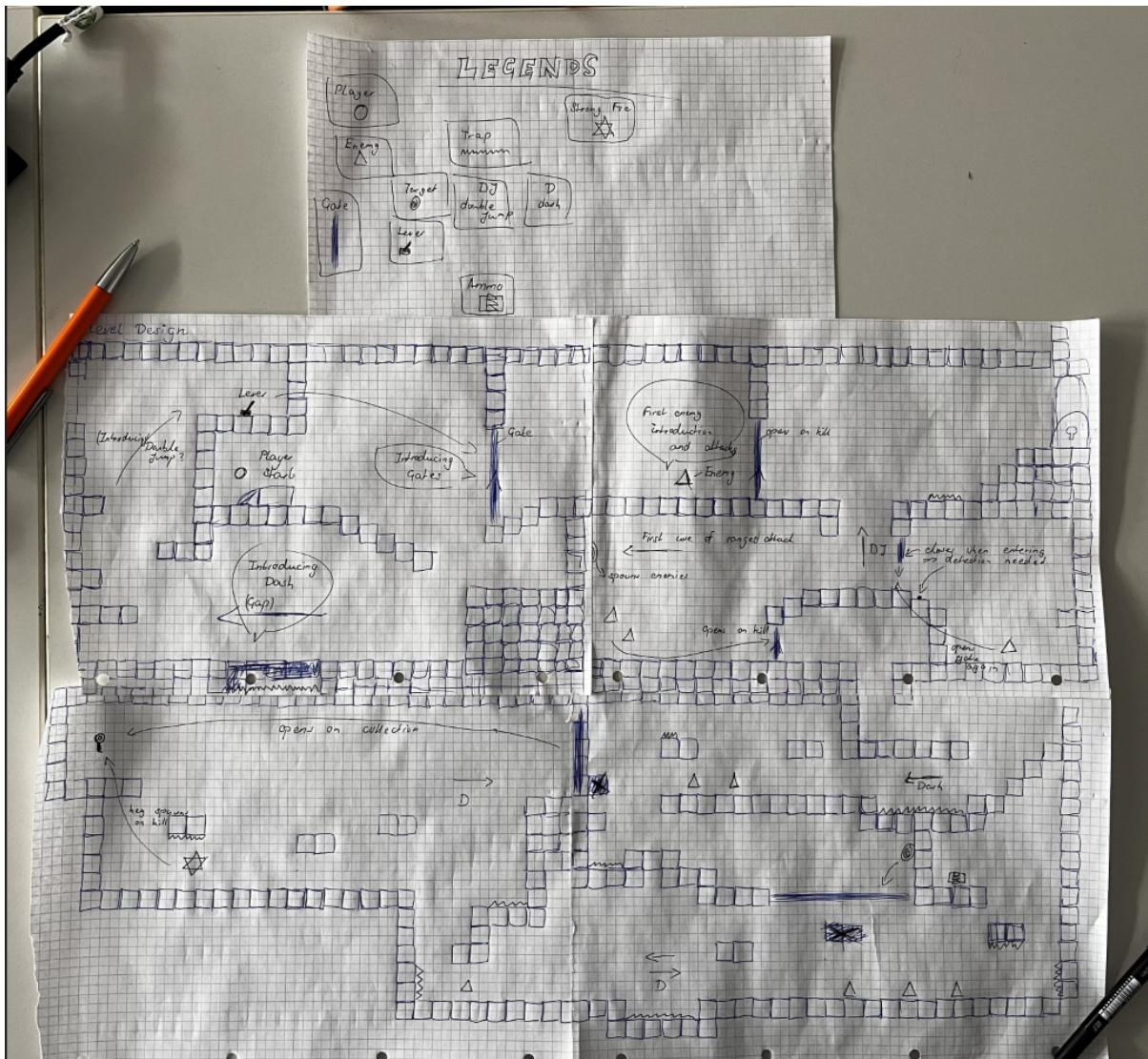
>> Gesichte wird noch expandiert.

Allgemeines Level Design im Metroidvania Still aufgestellt > Pyramiden sind nicht linear dementsprechend macht es Sinn, die Levels nicht linear zu gestalten.

Konzept für das erste Level des Spiels aufgestellt

>> Neue Mechaniken bedacht:

>>> Fallen, Targets zum Abschießen, Türen, Detectors zum Schließen dieser Türen, Hebel, Items, Dash, Double Jump und Schlüssel



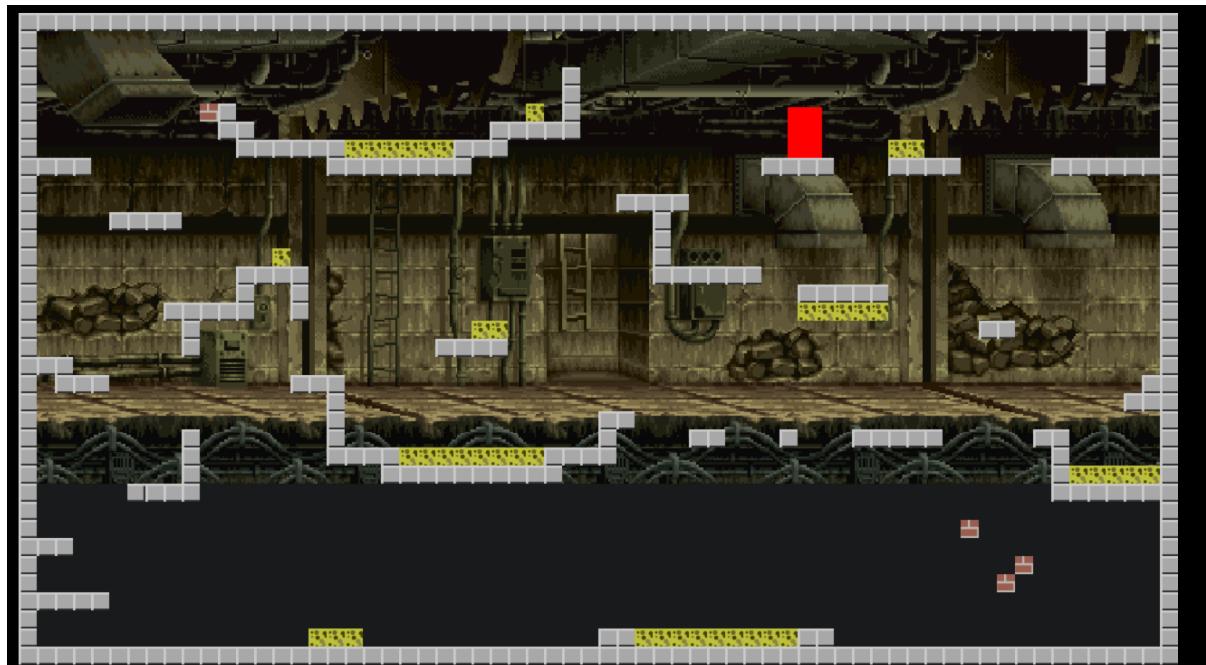
## ▼ 22.02 - Ausbau des ersten Levels

Weitere Ausarbeitung des Level-Designs.

Erstellung einer digitalen Map für leichtere Konzepte.



Bildung des Levels im Code zur Testung der Sprungkraft und Geschwindigkeit



Ein Level besteht aus vier Tilemaps, die geladen werden sobald der Spieler einen Detector erreicht zum Laden des nächsten Abteils im Level.

Erstellung eines Dash Prototypen. Erfordert noch Anpassung

>> Der Dash ist momentan ein Blink (Teleport) und keine Bewegung. Spieler kann durch Wände durch, wenn er dasht.

```
91 // Keyup-Event checkt welche Taste gedrückt wurde => true
92 window.addEventListener('keydown', (event) => {
93     switch(event.key){
94         case 'a': keys.a.pressed = true; break;
95         case 'd': keys.d.pressed = true; break;
96         case 'w': keys.w.pressed = true; break;
97         case ' ':keys.space.pressed = true; break;
98     }
99 });
100 // Keydown-Event checkt welche Taste losgelassen wurde => false
101 window.addEventListener('keyup', (event) =>{
102     switch(event.key){
103         case 'a': keys.a.pressed = false; break;
104         case 'd': keys.d.pressed = false; break;
105         case 'w': keys.w.pressed = false; break;
106         case ' ': keys.space.pressed = false; break;
107     }
108 });

// ERSTELLEN EINER Player-KLASSE
```

```
class Player{
    constructor({position}){
        this.position = position;
        this.velocity = {
            x: 0,
            y: 1
        }
        this.height = 56;
        this.width = 38;
        this.onGround = false;

        this.dashSpeed = 20;
        this.isDashing = false;
    }
}
```

```

40 | // Logik für den Dash
41 | if (keys.space.pressed && !this.isDashing) {
42 |     // Gebe dashSpeed zur Velocity dazu
43 |     this.velocity.x *= this.dashSpeed;
44 |     this.isDashing = true;
45 |     setTimeout(() => {
46 |         // Zurücksetzen nach dem Dash
47 |         this.velocity.x /= this.dashSpeed;
48 |         this.isDashing = false;
49 |     }, 1000);
50 |
51 }

```

## ▼ 23.02 - Level Design für Level 1-1, 1-2

Weiterer Ausbau des Levels

Level 1-1 und Level 1-2





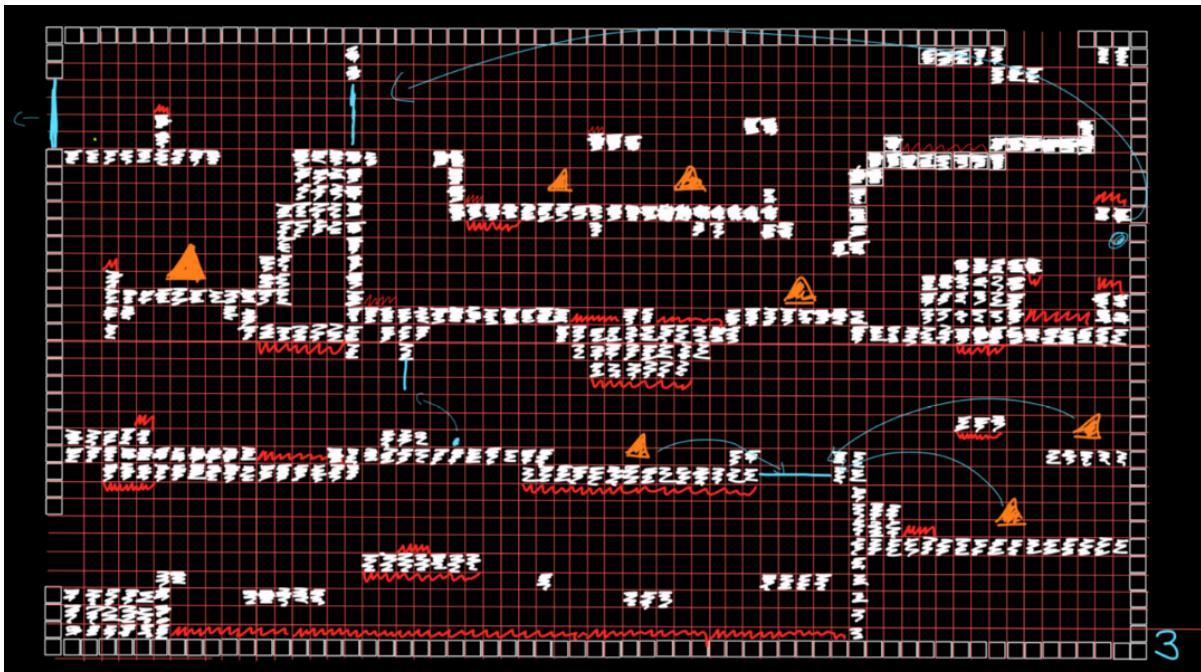
## ▼ 26.02 - Level Design für Level 1-2, 1-3

Weiterer Ausbau des Levels.

Level 1-2 und Level 1-3

Gedanken gemacht, wie man die Levels gestalten kann. Platforming und Gegner-Platzierung.





## ▼ 27.02 - Level Design für Level 1 - Dash - Media Query

Weiterer Ausbau des Levels.

Level 1-3

Anpassung des Dashes

>> Der Dash ist nun eine Bewegung, die die Velocity des Spielers nutzt, und nicht mehr eine Rechenoperation, die einen Punkt ausrechnet.

```

65 // Bewegungsoperation für den Dash
66 if (keys.space.pressed) {
67     // Wenn sich Spieler nach links bewegt und dasht, dann Dash nach links
68     if (this.velocity.x < 1) {
69         this.velocity.x = -10;
70     // Wenn sich Spieler nach rechts bewegt und dasht, dann Dash nach rechts
71     } else if (this.velocity.x > 1) {
72         this.velocity.x = 10;
73     }
74 }
75

```

```

32 // Setzt jeden Key standartmäßig auf false
33 const keys = {
34     a: {pressed: false},
35     d: {pressed: false},
36     w: {pressed: false},
37     space: {pressed: false}
38 }
39
40 let spacePressed = false;

```

```

93 // Keyup-Event checkt welche Taste gedrückt wurde => true
94 window.addEventListener('keydown', (event) => {
95     switch(event.key){
96         case 'a': keys.a.pressed = true; break;
97         case 'd': keys.d.pressed = true; break;
98         case 'w': keys.w.pressed = true; break;
99         case ' ':keys.space.pressed = true; break;
100    }
101 });
102 // Keydown-Event checkt welche Taste losgelassen wurde => false
103 window.addEventListener('keyup', (event) =>{
104     switch(event.key){
105         case 'a': keys.a.pressed = false; break;
106         case 'd': keys.d.pressed = false; break;
107         case 'w': keys.w.pressed = false; break;
108         case ' ': keys.space.pressed = false; break;
109    }
110 });

```

Einführung der Media Query für Responsivität des Spiels.

```
39 | @media only screen and (max-width: 1600px) {  
40 |     #game {  
41 |         display: flex;  
42 |         justify-content: center;  
43 |         align-items: center;  
44 |         transform: translateY(7px);  
45 |         scale: 1.1;  
46 |     }  
47 | }  
48 |  
49 >     @media only screen and (max-width: 1500px) { ...  
57 | }  
58 |  
59 >     @media only screen and (max-width: 1400px) { ...  
67 | }  
68 |  
69 >     @media only screen and (max-width: 1350px) { ...  
77 | }  
78 |  
79 >     @media only screen and (max-width: 1250px) { ...  
87 | }  
88 |  
89     @media only screen and (max-width: 1100px) {  
90 |         #game {  
91 |             display: flex;  
92 |             justify-content: center;  
93 |             align-items: center;  
94 |             transform: translateY(7px);  
95 |             scale: .7;  
96 |         }  
97 |     }
```

## ▼ 28.02 - Dateien Zusammensetzung - CSS Herzen - Finalisierung Level 1

Die Dateien aller Teammitglieder wurden in einen allgemeinen Ordner verfasst.  
>> Ordner im Ordner wurden erstellt für eine einfachere und bessere  
Gliederung der Dateien.

Name	Änderungsdatum	Typ	Größe
CSS	28.02.2024 11:28	Dateiordner	
JS	28.02.2024 11:22	Dateiordner	
media	28.02.2024 11:24	Dateiordner	
pages	28.02.2024 11:45	Dateiordner	
.gitattributes	28.02.2024 11:57	Git Attributes-Que...	1 KB
prefix.html	28.02.2024 13:54	HTML-Quelldatei	3 KB
Startbildschirm.html	28.02.2024 11:48	HTML-Quelldatei	1 KB

Die <div> Herzcontaienr hat CSS Eigenschaften bekommen, damit es als Overlay im Spiel dargestellt wird.

>> Responsivität muss beigefügt werden.

```

105 |     <div id="health-bar">
106 |       
107 |       
108 |       
109 |     </div>

```

## ▼ 29.02 - Level 1 Finalisierung

### ▼ Finalisiertes Design von Level 1

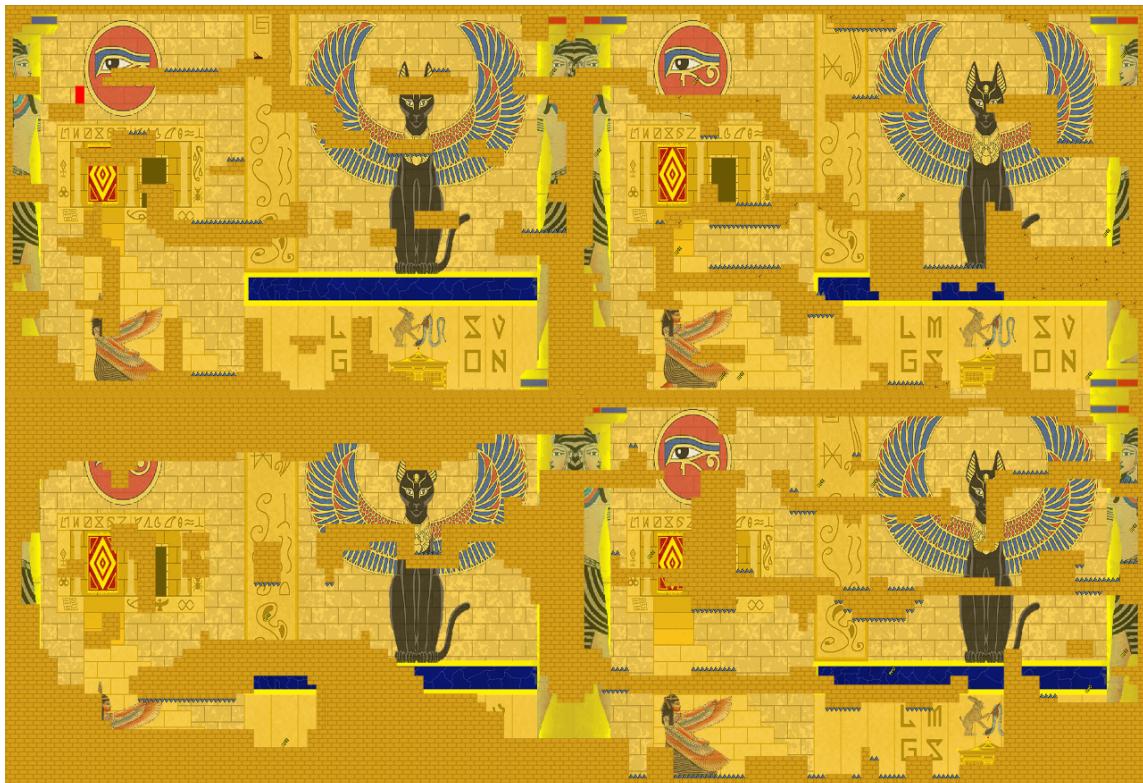
Die vier Tilemaps der einzelnen Level Abschnitte wurden alle in eine Datei gepackt.



- ▼ Canvas um 160px in Höhe vergrößert
  - >> Hintergrund muss um 160px angepasst werden.



>>> Hintergrund erweitert und vervierfacht für das Level



## ▼ 01.03 - Level 2 Konzept

Planung der Gestaltung vom zweiten Level des Spiels

Design gestrichen und mit anderem Design ersetzt

## ▼ 04.03 - Design für Level 2

Große Anpassungen im Design des zweiten Levels.

>> Es werden keine vier Level aneinander und untereinander gereiht

>>> Das zweite Level wird aus größtenteils vertikalen Passagen bestehen, wie in Ice Climbers.

Ansatz des zweiten Levels erstellt

>> Jumppads für eine Abwechslung im Platforming werden eingeführt.

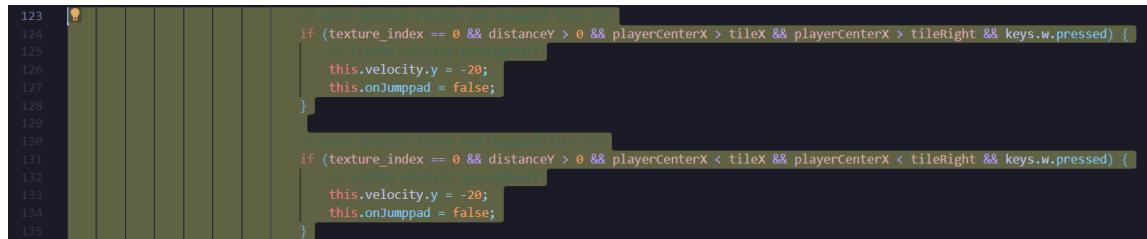
## ▼ 05.03 - "Jumppad" - Level 2-1

### ▼ Programmierung eines "Jumppads" für das angepasste Level Design vom zweiten Level

>> Probleme beim Aufbau des Jumppads:

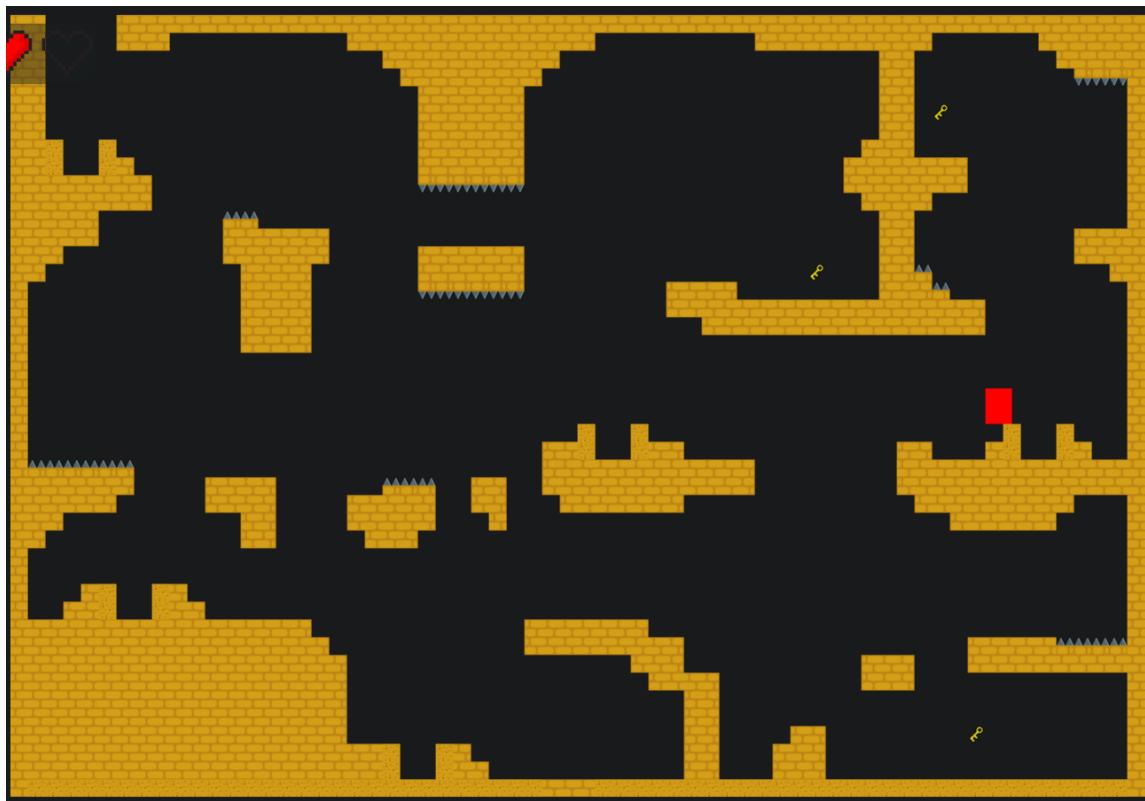
>>> Die Kollision von oben wird nicht von keys übernommen > Der neue Wert der Sprungkraft wird nicht übernommen

>>> Jumppad funktioniert nur von links und / oder rechts



```
123 // wenn Spieler rechts vom Jumppad steht ist  
124 if (texture_index == 0 && distancey > 0 && playerCenterX > tilex && playerCenterX < tileRight && keys.w.pressed) {  
125     // erhöhe nächste Sprungkraft  
126     this.velocity.y = -20;  
127     this.onJumppad = false;  
128 }  
129 // wenn Spieler Links vom Jumppad steht ist  
130 if (texture_index == 0 && distancey < 0 && playerCenterX < tilex && playerCenterX > tileRight && keys.w.pressed) {  
131     // erhöhe nächste Sprungkraft  
132     this.velocity.y = -20;  
133     this.onJumppad = false;  
134 }  
135 }
```

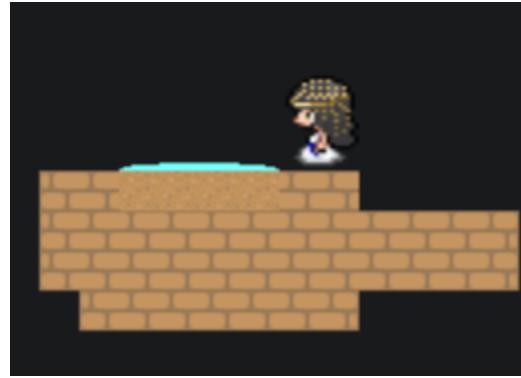
### ▼ Aufbau von Level 2-1



## ▼ 06.03 - Jumppad Überarbeitung - Air Jumppads 3

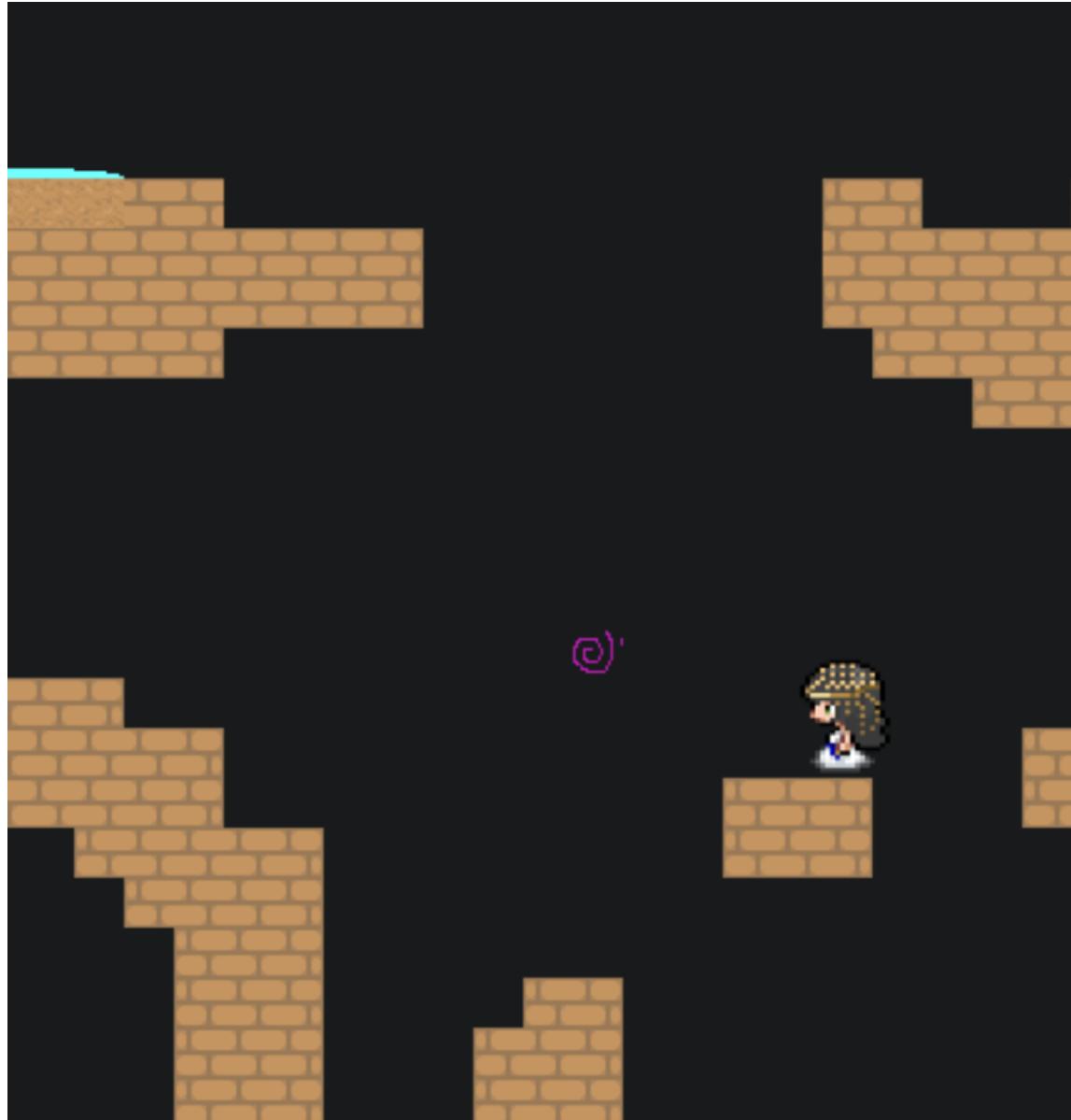
▼ Der Code für das Jumppad würde überarbeitet

```
if (texture_index != 0 || texture_index < 0) {  
    this.onJumppad = false;  
    console.log("DONT JUMP");  
}  
  
if (texture_index == 0 && keys.space.pressed && keys.w.pressed) {  
    this.onJumppad = true;  
}
```



- ▼ Einführung von Air Jumppads um in der Luft einen zweiten Sprung ausführen zu können

```
checkItemCollision(texture_index, y, x){  
    // Wenn texture_index 10 ist =>  
    if(texture_index === 13){  
        // füge Leben hinzu  
        this.addHealth();  
        // setze texture_index auf -1  
        map[y][x] = -1;  
    }  
    if(texture_index === 10){  
        this.collectItem('key');  
        map[y][x] = -1  
        // console.log(this.items);  
    }  
  
    if (texture_index == 17 && keys.space.pressed) {  
        this.velocity.y = -10;  
    }  
}
```



## ▼ 07.03 - HTML-CSS Überarbeitung

### ▼ Komplette Überarbeitung und Anpassung vom CSS

>> Animationen mit Transitions wurden hinzugefügt sowohl wie die Lesbarkeit der Tasten und Infos





```

1  <!DOCTYPE html>
2  <html lang="de">
3      <head>
4          <meta charset="utf-8" />
5          <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6          <title>Startbildschirm</title>
7          <link rel="stylesheet" href="../CSS/styles.css">
8          <script src="../Script/script.js"></script>
9      </head>
10
11     <body>
12         <div id="curtain" class="curtain"></div>
13         <div class="moving_svg"></div>
14
15         <div id="game" class="game_background">
16             <div class="main_heading"> Wrath of the Fallen Queen</div>
17             <div id="containerButtons" class="container_buttons">
18                 <div id="startButton" class="start" href="prefix.html"> Start </div>
19                 <div id="settingsButton" class="settings" href="#"> Steuerung </div>
20                 <div id="creditsButton" class="credits" href="#"> Credits </div>
21             </div>
22         </div>
23
24         <div class="container_info">
25             <div class="arrow"></div>
26             <div id="entwicklung" class="entwicklung"> Entwicklung </div>
27             <div id="level" class="level"> Level </div>
28             <div id="info" class="info"> Info </div>
29             <div id="story" class="story"> Geschichte </div>
30         </div>
31
32     </body>
33
34     <script>
35         let game = document.getElementById("game");
36         let start = document.getElementById("startButton");
37         let buttons = document.getElementById("containerButtons")
38         let curtain = document.getElementById("curtain")
39
40         let entwicklung = document.getElementById("entwicklung");
41         let level = document.getElementById("level");
42         let info = document.getElementById("info");
43         let story = document.getElementById("story");
44
45         start.addEventListener("click", function() {
46             curtain.classList.toggle("curtain_CLICKED");
47         })
48

```

```
48
49     entwicklung.addEventListener("click", function() {
50         curtain.classList.toggle("curtain_CLICKED");
51     })
52
53     level.addEventListener("click", function() {
54         curtain.classList.toggle("curtain_CLICKED");
55     })
56
57     info.addEventListener("click", function() {
58         curtain.classList.toggle("curtain_CLICKED");
59     })
60
61     story.addEventListener("click", function() {
62         curtain.classList.toggle("curtain_CLICKED");
63     })
64
65     </script>
66 </html>
```

```

1   * {
2     padding: 0;
3     margin: 0;
4   }
5
6   body {
7     width: 100vw ;
8     height: 100vh ;
9     background: linear-gradient(0deg, #rgba(134,219,188,1) 0%, #rgba(46,185,169,1) 51%, #rgba(76,149,156,1) 100%);
10    background-color: black;
11    user-select: none;
12    overflow: hidden;
13    font-family: Algerian ;
14    cursor: url('../media/img/egyptian-custom-cursor.png'), auto;
15    a { cursor: url('../media/img/egyptian-custom-cursor.png'), pointer;}
16  }
17
18  .curtain {
19    display: flex;
20    width: 100%;
21    height: 0%;
22    position: absolute;
23    top: 0;
24    background-color: blue;
25    transition: 1s ease all;
26    z-index: 10;
27  }
28
29  .curtain_CLICKED {
30    height: 100%;
31  }
32
33  .moving_svg {
34    display: flex;
35    position: absolute;
36    width: 200vw ;
37    height: 200vh ;
38    transform: translate(-50%, -50%);
39    background-image: url("../media/img/svg/mmmotif.svg");
40    background-size: cover;
41    background-repeat: no-repeat ;
42    background-position: center center ;
43    animation: ANIMATION_moving_svg 360s linear infinite;
44  }

```

```

46    @keyframes ANIMATION_moving_svg {
47        0% {transform: translate(-50%, -50%);}
48        50% {transform: translate(0, 0);}
49        100% {transform: translate(-50%, -50%);}
50    }
51
52    .game_background {
53        display: flex;
54        width: 110%;
55        height: 130%;
56        position: absolute;
57        top: 50%;
58        left: 50%;
59        transform: translate(-65%, -66.5%);
60        z-index: 3;
61        border-radius: 20px;
62        box-shadow: 0px 0px 30px 15px □rgba(0, 0, 0, 0.719);
63        background-image: url("../media/img/backgrounds/main.png") ;
64        background-repeat: no-repeat ;
65        background-size: cover;
66        background-position: center center;
67        scale: 0.75;
68        transition: 3s ease all, scale 3s ease ;
69    }
70
71    .game_background_CLICKED {
72        scale: 2;
73        transform: translate(-25%, -25%);
74    }
75
76    .main_heading {
77        display: flex;
78        justify-content: center;
79        align-items: center;
80        text-align: center ;
81        width: 100%;
82        height: 25%;
83        position: absolute;
84        z-index: 5;
85        border-radius: 20px 20px 0px 0px;
86        background-color: □rgba(73, 73, 73, 0.322);
87        color: ■rgb(229, 255, 0) ;
88        font-size: 7rem ;
89        backdrop-filter: blur(5px);
90    }

```

```
92     .container_buttons {
93         display: flex;
94         justify-content: center;
95         align-items: center;
96         align-self: center;
97         flex-direction: column;
98         width: 600px;
99         height: 500px;
100        position: absolute;
101        left: 50%;
102        top: 50%;
103        z-index: 5;
104        border-radius: 10px;
105        background-color: □rgba(0, 0, 0, 0.01);
106        transform: translate(-50%, -50%);
107        backdrop-filter: blur(0px);
108        transition: .5s ease all, scale 1s ease;
109    }
110
111    .container_buttons_CICKED {
112        display: none;
113    }
114
115    .container_buttons:hover {
116        background-color: □rgba(0, 0, 0, .5);
117        backdrop-filter: blur(3px);
118        box-shadow: 0px 0px 3px 10px □white;
119    }
120
121    .start,
122    .settings,
123    .credits {
124        display: flex;
125        text-align: center;
126        position: absolute;
127        font-size: 3em ;
128        color: □yellow ;
129        font-weight: bold ;
130        text-align: center ;
131        transition: .3s ease all;
132        text-decoration: none;
133    }
134
135    .start:hover {
136        font-size: 6rem;
137    }
```

```
159     .settings:hover,
160     .credits:hover {
161         font-size: 3.5em
162     }
163
164     .start {
165         transform: translateY(-10rem);
166         font-size: 5em ;
167         color: black;
168     }
169
170     .settings {
171         transform: translateY(0rem);
172     }
173
174     .credits {
175         transform: translateY(5rem);
176     }
177
178     .container_info {
179         display: flex;
180         justify-content: center;
181         align-items: center;
182         width: 300px;
183         height: 300px;
184         position: absolute;
185         left: -14%;
186         top: 50%;
187         z-index: 9;
188         transform: translateY(-50%);
189         border-radius: 10px;
190         box-shadow: 0px 0px 100px 10px;
191         background-color: black;
192         overflow: hidden;
193         transition: 1s ease all;
194     }
195
```

```

176    .arrow {
177        display: flex;
178        width: 50px;
179        height: 50px;
180        position: absolute;
181        right: -10%;
182        top: 50%;
183        z-index: 10;
184        scale: 2;
185        background-image: url("../media/img/svg/arrow-right.svg");
186        background-repeat: no-repeat;
187    }
188
189    .container_info:hover {
190        left: .3rem;
191    }
192
193    .entwicklung,
194    .level,
195    .info,
196    .story {
197        display: flex;
198        position: absolute;
199        scale: 1.3;
200        transition: .5s ease all;
201    }
202
203    .entwicklung {
204        top: 4rem;
205    }
206
207    .level {
208        top: 7rem;
209    }
210
211    .info {
212        top: 10rem;
213    }
214
215    .story {
216        top: 13rem;
217    }

```

## ▼ 08.03 - HTML-CSS-JS Überarbeitung - Gedanken

### ▼ Weitere Überarbeitung des CSS sowie HTML mit Einführung von JS



▼ Verfeinerung und Überarbeitung des Gesamtcodes.

```

2 <html lang="de">
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Startbildschirm</title>
7     <link rel="stylesheet" href="../CSS/styles.css">
8     <script src="../Script/script.js"></script>
9   </head>
10
11 <body>
12   <div id="curtain" class="curtain"></div>
13   <div id="curtain2" class="curtain2"></div>
14   <div class="moving_svg"></div>
15
16   <div id="game" class="game_background">
17     <div id="settingsScreen" class="settings_screen">
18       <div id="settingsBack"> BACK </div>
19       <div class="settings_heading"> Steuerung </div>
20     </div>
21     <div id="creditsScreen" class="credits_screen">
22       <div id="creditsBack"> BACK </div>
23       <div class="credits_heading"> Mitwirkende </div>
24     </div>
25     <div id="mainHeading" class="main_heading"> Wrath of the Fallen Queen </div>
26     <div id="mainContainer" class="main_container">
27       <div class="line"></div>
28       <div class="line2"></div>
29       <a id="startButton" class="start" href="prefix.html" onclick="delayRedirect(event)">
30         <div class="line"></div>
31       </a>
32       <div id="settingsButton" class="settings" href="#">
33         <div class="line"></div>
34       </div>
35       <div id="creditsButton" class="credits" href="#">
36         <div class="line"></div>
37       </div>
38     </div>
39     <div id="moreLessButton" class="more_less_button"> Mehr
40       <div class="line"></div>
41       <div class="line2"></div>
42     </div>
43     <div id="moreLessBG" class="more_less_BG"></div>
44   </div>
45   <div id="entwicklung" class="entwicklung"> Entwicklung ...
46   </div>
47   <div id="level" class="level"> Level ...
48   </div>
49   <div id="info" class="info"> Info ...
50   </div>
51   <div id="story" class="story"> Geschichte ...
52   </div>
53 </div>
54 </body>
55
56
57
58
59
60
61

```

## Neue JavaScript Elemente

```
169 let hasPlayed = false;
170
171 document.getElementById("mainContainer").addEventListener("mouseover", function() {
172     if (!hasPlayed) {
173         let theLetters = "abcdefghijklmnopqrstuvwxyz%&+=-€ƒ†šŒžšæžŸŒ€¥|§ø°±µø¾¾;?ÆÇÅØÖ×Ø÷þø×";
174         let content = "Start";
175         let speed = 50;
176         let increment = 4;
177
178         let clen = content.length;
179         let si = 0;
180         let stri = 0;
181         let block = "";
182         let fixed = "";
183
184         function rustle(i) {
185             setTimeout(function() {
186                 if (--i) {
187                     rustle(i);
188                 }
189                 nextFrame(i);
190                 si = si + 1;
191             }, speed);
192         }
193
194         rustle(clen * increment + 1);
195
196         function nextFrame(pos) {
197             for (let i = 0; i < clen - stri; i++) {
198                 let num = Math.floor(Math.random() * theLetters.length);
199                 let letter = theLetters.charAt(num);
200                 block = block + letter;
201             }
202             if (si == (increment - 1)) {
203                 stri++;
204             }
205             if (si == increment) {
206                 fixed = fixed + content.charAt(stri - 1);
207                 si = 0;
208             }
209             document.getElementById("startButton").innerHTML = fixed + block;
210             block = "";
211         }
212
213         hasPlayed = true;
214     }
215 });
});
```

## Weitere CSS Anpassungen

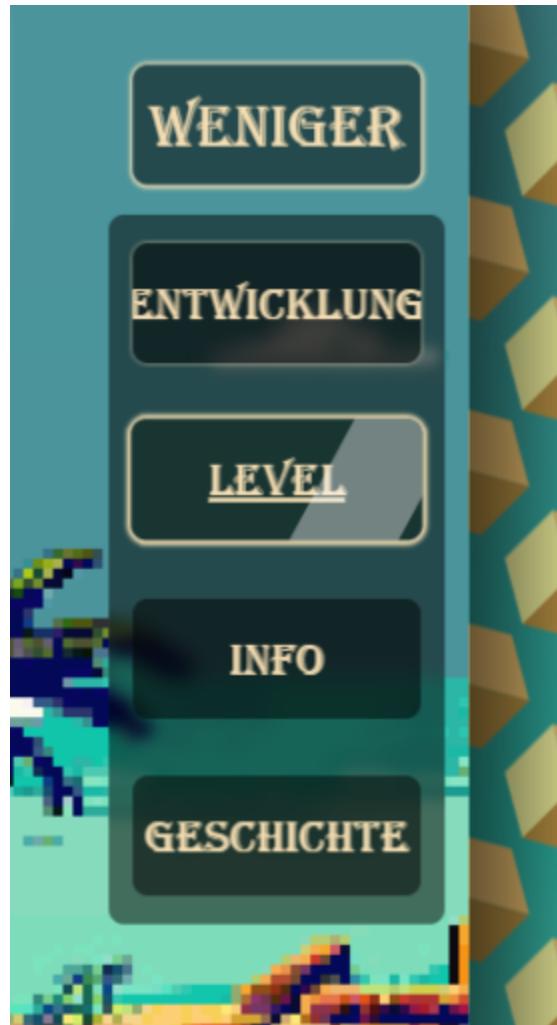
469 lines

>> Weitere Arbeit ist nötig.

▼ 11.03 - HTML-CSS-JS Überarbeitung

## ▼ Verlinkung von Pages und Änderungen am CSS





▼ JS geordnet

```

68 <script>
69 // Definitionen
70 let game = document.getElementById("game");
71
72 let mainHeading = document.getElementById("mainHeading");
73
74 let mainContainer = document.getElementById("mainContainer");
75 let containerLine = document.getElementById("containerLine");
76 let startButton = document.getElementById("startButton");
77
78 let settingsButton = document.getElementById("settingsButton");
79 let settingsScreen = document.getElementById("settingsScreen");
80 let settingsBack = document.getElementById("settingsBack");
81
82 let creditsButton = document.getElementById("creditsButton");
83 let creditsScreen = document.getElementById("creditsScreen");
84 let creditsBack = document.getElementById("creditsBack");
85
86 let mainCurtain = document.getElementById("mainCurtain")
87 let curtainDev = document.getElementById("curtainDev");
88 let curtainLevel = document.getElementById("curtainLevel");
89 let curtainInfo = document.getElementById("curtainInfo");
90 let curtainStory = document.getElementById("curtainStory");
91
92 let moreLessButton = document.getElementById("moreLessButton");
93 let moreLessBG = document.getElementById("moreLessBG");
94
95 let entwicklung = document.getElementById("entwicklung");
96 let level = document.getElementById("level");
97 let info = document.getElementById("info");
98 let story = document.getElementById("story");
99
100 let hasPlayedMouse = false;
101 let hasPlayedMainContainer = false;
102 let lettersStartGenerated = false;
103 let lettersSettingsGenerated = false;
104 let lettersCreditsGenerated = false;
105 let lettersMoreGenerated = false;
106
107 // Funktionen
108 > function mouseReveal() { ...
316 mouseReveal();
317
318 > function delayRedirect(event) { ...
324
325 > function eventListeners() { ...
381 eventListeners();
382 </script>

```

## ▼ 12.03 - Zusammenführung der Daten und Anpassung des Codes

Planung der Gestaltung vom zweiten Level des Spiels

Design gestrichen und mit anderem Design ersetzt

▼ **13.03 - Design für Level 2-2 - Level 2-3 - "Multiple Air Jumppad - Moving Damaging Tiles"**

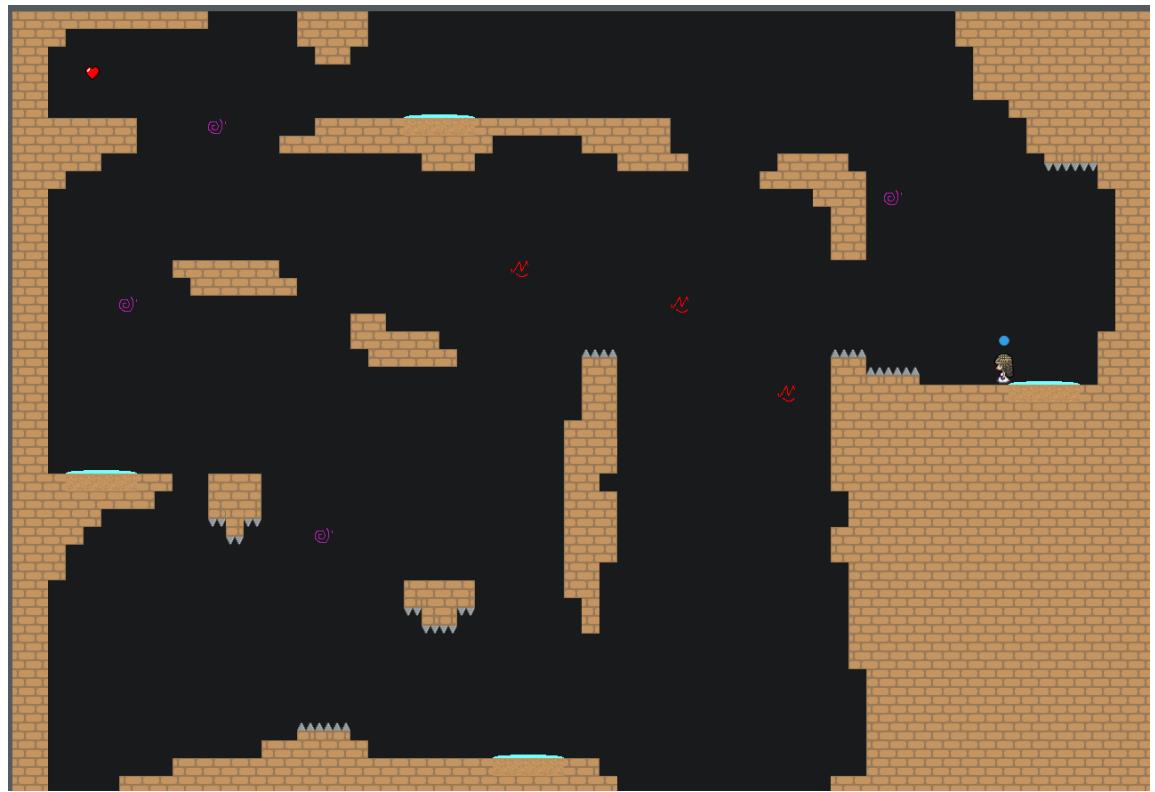
▼ **Einführung neuer Mechanik: "Multiple Air Jumppad"**



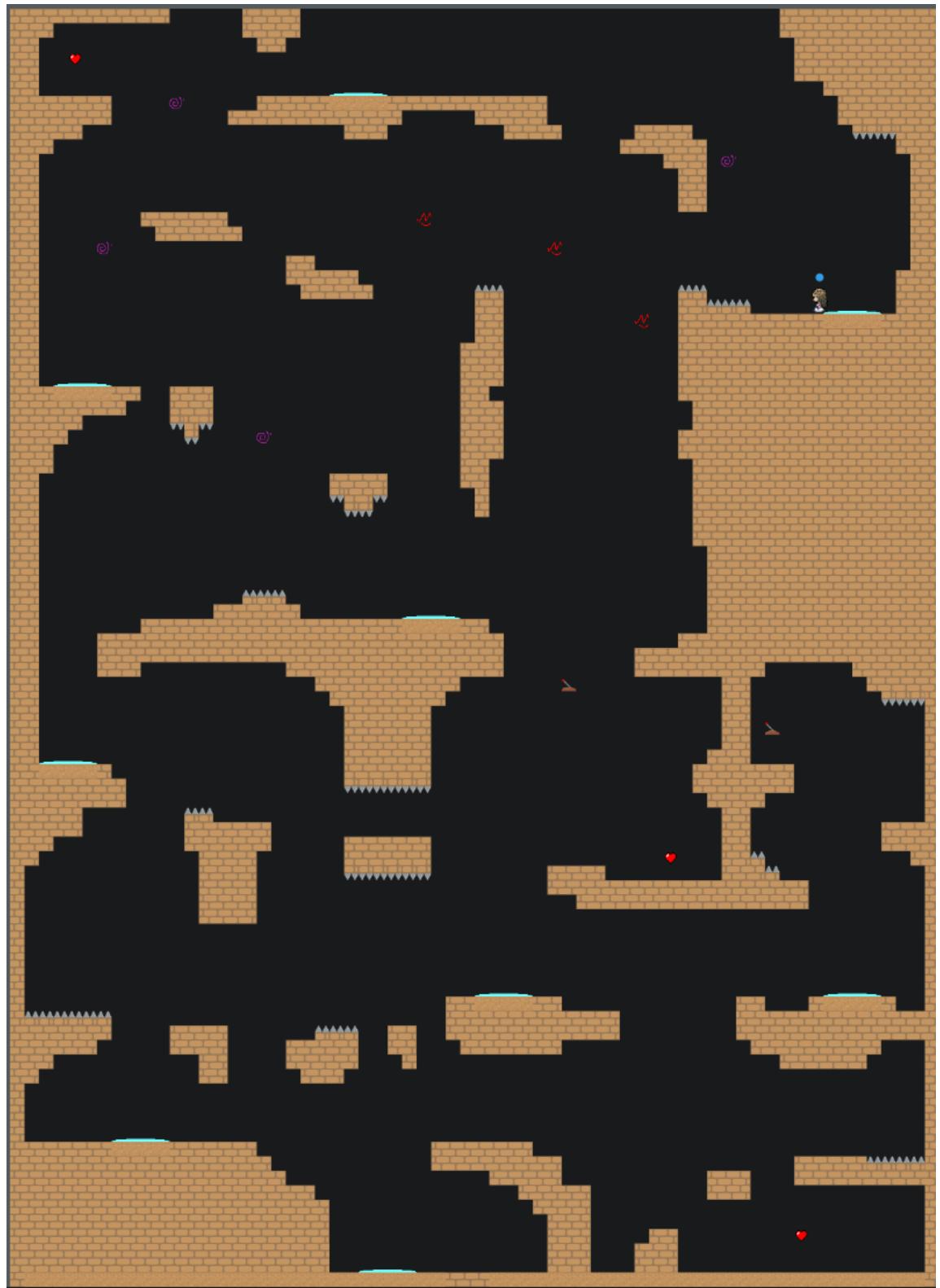
```
270
271     if (texture_index == 18 && keys.w.pressed) {
272         this.velocity.y = -9.2;
273     }
274
275 }
```

>> Erlaubt es dem Spieler dieses Jumppad mehr als nur einmal in der Luft zu verwenden

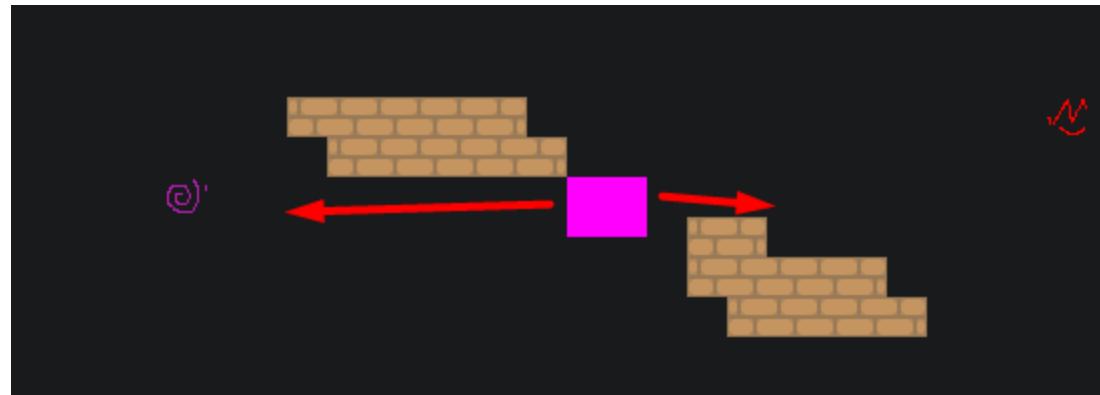
▼ **Level Design für Level 2-2**



▼ Anpassung des Canvas zum Betrachten des Levels in seiner größeren Form:



## ▼ Einführung eines neuen Schadenstypen



Blöcke / Tiles die sich in der X-Achse bewegen und dem Spieler bei Kollision Schaden zufügen.

## ▼ Level Design für Level 2-3



## ▼ 14.03 - Jumppad - Coins - Healing Animation

### ▼ Jumppad

Handle Input für Jumppad und Shift Bedingung entfernt

```
● 165  ✓ handleInputJumppad() {  
 166  ✓   if(this.onJumppad){  
 167  |     return;  
 168  }  
 169  ✓   if(keys.a.pressed) {  
 170  |     this.velocity.x = -3.5;  
 171  |     this.isLeft = true;  
 172  ✓   } else if(keys.d.pressed) {  
 173  |     this.velocity.x = 3.5;  
 174  |     this.isLeft = false;  
 175  }  
 176  ✓   if(keys.w.pressed){  
 177  |     if(this.onGround) {  
 178  |       this.velocity.y += -9.2;  
 179  |       this.onGround = false;  
 180  |     }  
 181  }  
 182  ✓   if(!this.isDashing)  
 183  {  
 184  |     this.dash();  
 185  }  
 186  }  
 187  
 188  
 189  
 190  
 191  
 192  
 193  
 194  
 195  
 196  
 197  
 198  
 199  
 200  
 201  
 202  
 203  
 204  
 205  
 206  
 207  
 208  
 209  
 210  
 211  
 212  
 213  
 214  
 215  
 216  
 217  
 218  
 219  
 220  
 221  
 222  
 223  
 224  
 225  
 226  
 227  
 228  
 229  
 230  
 231  
 232  
 233  
 234  
 235  
 236  
 237  
 238  
 239  
 240  
 241  
 242  
 243  
 244  
 245  
 246  
 247  
 248  
 249  
 250  
 251  
 252  
 253  
 254  
 255  
 256  
 257  
 258  
 259  
 260  
 261  
 262  
 263  
 264  
 265  
 266  
 267  
 268  
 269  
 270  
 271  
 272  
 273  
 274  
 275  
 276  ✓   if (texture_index == 0 && keys.w.pressed) {  
 277  |     this.onJumppad = true;  
 278  }  
 279  // Stoppt die vertikale Geschwindigkeit
```

### ▼ Coins

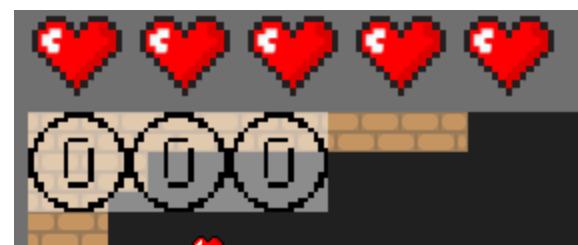
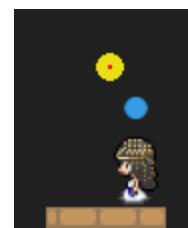
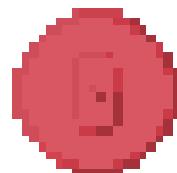
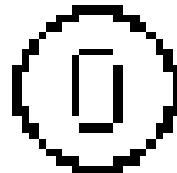
▼ Einführung von Coins in das Spiel für ein Objekt zum Sammeln

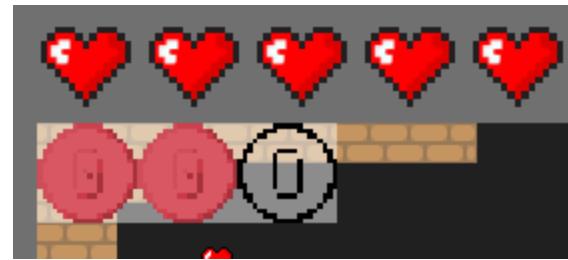
```
100 |     </div>
101 |     <div id="coin-bar">
102 |         
103 |         
104 |         
105 |     </div>
      <canvas id="game"></canvas>
```

```
1  class Player {
2      isDashing = false;
3      constructor({position}) {
4          this.position = position;
5          this.velocity = {
6              x: 0,
7              y: 1
8          }
9          this.height = 40;
10         this.width = 30;
11
12         this.onGround = false;
13         this.onSpike = false;
14         this.onJumppad = false;
15         this.jumppadCD = false;
16
17
18         this.maxHealth = 5;
19         this.health = this.maxHealth;
20
21         this.maxCoin = 3;
22         this.coin = 0;
23
24         this.doorTileLimit = 3;
25         this.items = [];
26
27         this.lastFrameTime = 0;
28         this.frame = 0;
29
30         this.hasCollided = false;
31         this.doorTileCount = 0;
32
33         this.dashTime = 100;
34         this.dashPause = false;
35         this.isLeft = true;
36     }
```

```
91 | // Coin UI
92 | updateCoinUI() {
93 |     for (let i = 0; i <= 3; i++) {
94 |         const coin = document.getElementById(`coin${i}`);
95 |         if (i <= this.coin) {
96 |             heart.src = '../img/coin_empty.png';
97 |         } else {
98 |             heart.src = '../img/coin.png';
99 |         }
100 |     }
101 |
102 }
```

## ▼ Sprites





### ▼ Healing Animation



Allgemeine Einführung für zukünftige Animationen

### ▼ Code

!>> Animation läuft zu schnell ab

```
16
17 const spritesheetHeal = new Image();
18 spritesheetHeal.src = "../img/spritesheet_heal.png";
19
```

```

48     draw(texture_index){
49         if(new Date().getTime() > this.lastFrameTime + 100){
50             this.frame++;
51             if(this.velocity.x === 0){
52                 this.frame = 0;
53             } else if(this.velocity.x < 0){
54                 this.direction = -1;
55             } else {
56                 this.direction = 1;
57             }
58             if(this.frame > 3) this.frame = 0;
59             this.lastFrameTime = new Date().getTime();
60         }
61
62         let row = 0;
63         switch (this.direction) {
64             case -1:
65                 row = 0;
66                 break;
67             case 0:
68                 row = 0;
69                 break;
70             case 1:
71                 row = 1;
72                 break;
73         }
74
75         if (texture_index === 10) {
76             ctx.drawImage(spritesheetHeal,64*this.frame,row*64,64,64, this.position.x-5, this.position.y, 40, this.height);
77         } else {
78             ctx.drawImage(spritesheet, 64 * this.frame, row * 64, 64, 64, this.position.x - 5, this.position.y, 40, this.height);
79         }
80
81         if (texture_index === 10) {
82             if (this.isLeft) {
83                 ctx.drawImage(spritesheetHeal, 64 * this.frame, 1 * 64, 64, 64, this.position.x - 5, this.position.y, 40, this.height);
84             } else {
85                 ctx.drawImage(spritesheetHeal, 64 * this.frame, 0 * 64, 64, 64, this.position.x - 5, this.position.y, 40, this.height);
86             }
87         }
88     }

```

## ▼ 15.03 - Collisioncheck Update

### ▼ Collisioncheck Update

Collisioncheck bei NPC und/oder movingBlock auf einen Wert stellen z.B. 99, und auf der Karte diesen Wert eingeben für invisible Colliders, die den Spieler nicht beeinträchtigen.

```

49     checkBlockTiles(texture_index, tileX, tileY) {
50         if(texture_index >= 0 && texture_index == 99) {
51             const playerCenterX = this.position.x + this.width / 2;
52             const playerCenterY = this.position.y + this.height / 2;
53             const tileCenterX = tileX + tile / 2;
54             const tileCenterY = tileY + tile / 2;
55
56             const distanceX = playerCenterX - tileCenterX;
57             const distanceY = playerCenterY - tileCenterY;
58
59             const combinedHalfWidths = this.width / 2 + tile / 2;
60             const combinedHalfHeights = this.height / 2 + tile / 2;
61
62             if(Math.abs(distanceX) < combinedHalfWidths + 1 && Math.abs(distanceY) < combinedHalfHeights + 1) {
63                 this.hasCollided = true;
64
65                 const overlapX = combinedHalfWidths - Math.abs(distanceX);
66                 const overlapY = combinedHalfHeights - Math.abs(distanceY);
67
68                 if(overlapX >= overlapY) {
69                     this.collisionCheckY(texture_index, distanceY, distanceX, overlapX, overlapY);
70                 } else {
71                     this.collisionCheckX(overlapX, tileCenterX);
72                 }
73             }
74         }
75     }

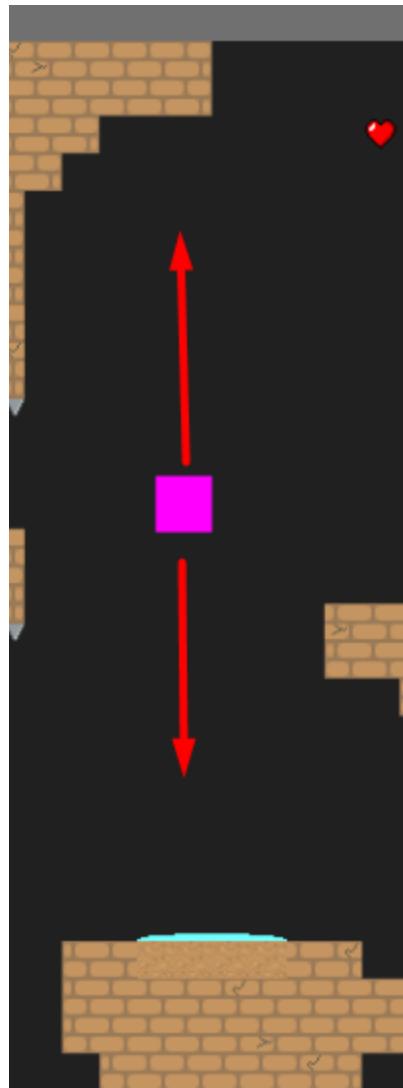
```



## ▼ 19.03 - Bewegende Spikes

### ▼ Einfügung von Spikes die sich von oben nach unten Bewegen

Selber Code wie Moving Blocks



## ▼ 20.03 - Map Loading

### ▼ Einführung von Map Loading im Spiel selbst

- >> Eine neue Tilemap wird geladen, wenn ein bestimmter texture\_index berührt wird
- >> Überarbeitung der Check Item Collision und allgemeine Collision zum Übernehmen der neuen Map Positionen
- >> Eine neue Variable für die maps selbst musste erstellt werden zum Laden der verschiedenen Maps

```

80  function drawMap() {
81      const currentMap = player.mapLoad ? map : map2;
82      canvas.width = currentMap[0].length * tile;
83      canvas.height = currentMap.length * tile;
84      // Clear the canvas if player.mapLoad is false
85      if (player.mapLoad == false) {
86          ctx.clearRect(0, 0, canvas.width, canvas.height);
87          // Draw the second map
88          for (let y = 0; y < map2.length; y++) { // Für jede Reihe
89              for (let x = 0; x < map2[0].length; x++) { // Gehe jede
90                  const texture_index = map2[y][x]; // Der map-index
91                  if (texture_index < 0) continue; // Wenn map-index
92                  ctx.drawImage ( // Tiles einzeichnen (tileset, aus
93                      tileset, // img => tileset
94                      tile * texture_index, // sx => x-Koordinate des
95                      0, // sy => y-Koordinate des Tiles im Tileset
96                      tile, // s-width => Tile-Breite
97                      tile, // s-height => Tile-Höhe
98                      tile * x, // dx => x-Koordinate des Tiles auf
99                      tile * y, // dy => y-Koordinate des Tiles auf
100                     tile, // d-width => Breite des Tiles auf Canvas
101                     tile // d-height => Höhe des Tiles auf Canvas
102                 );
103             }
104         }
105     } else {
106         for (let y = 0; y < map.length; y++) {
107             for (let x = 0; x < map[0].length; x++) { ...
108             }
109         }
110     }
111     player.draw();
112 }
113

```

```

385
386     if (texture_index == 17) {
387         this.mapLoad = false;
388         console.log("map");
389     }
390

```

```

38
39     this.mapLoad = true;

```

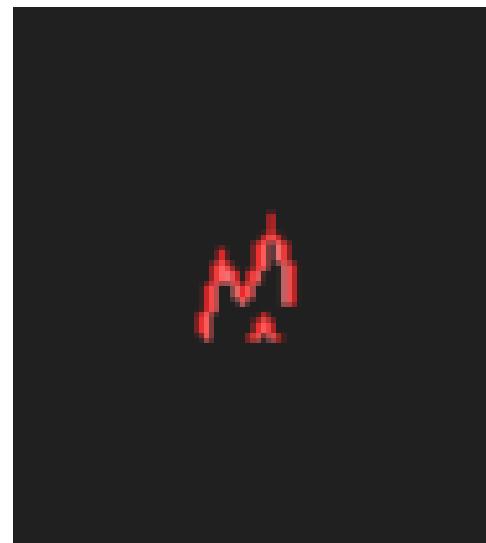
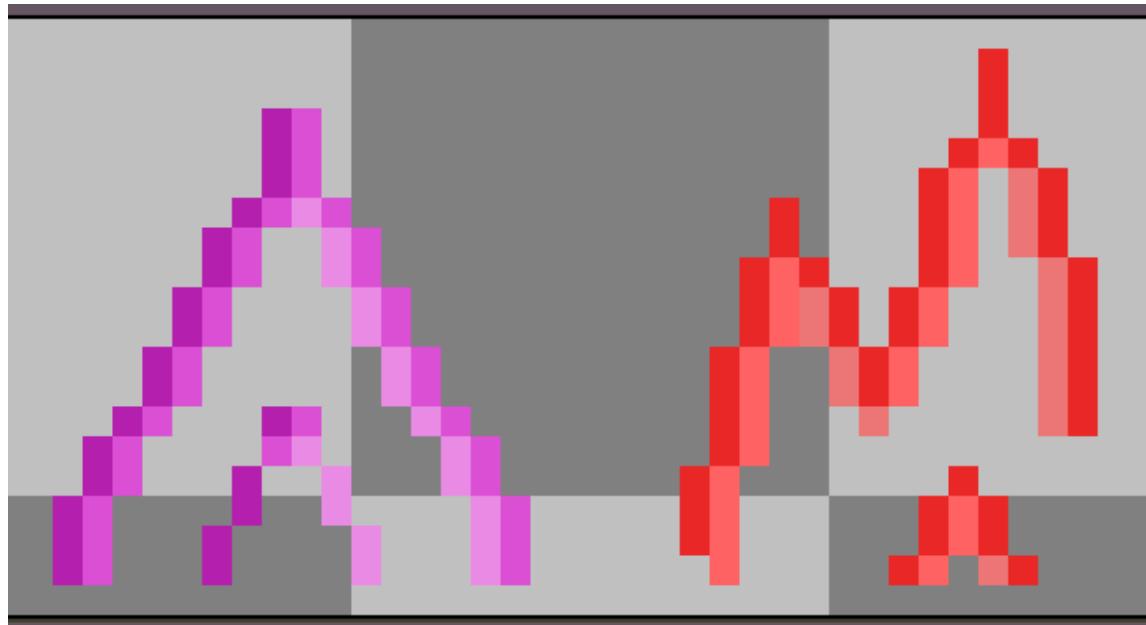
```

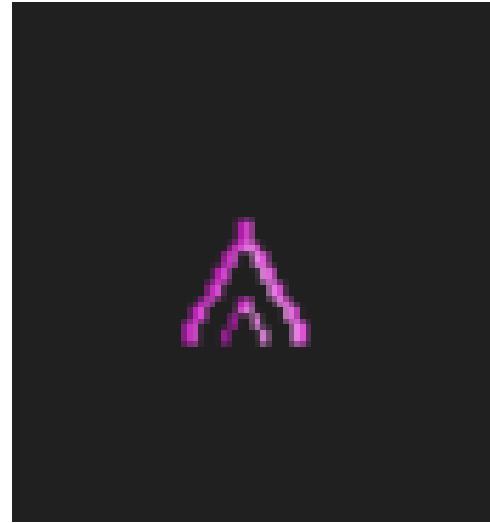
356     checkItemCollision(texture_index, y, x){
357         const currentCollisionMap = this.mapLoad ? map : map2;
358

```

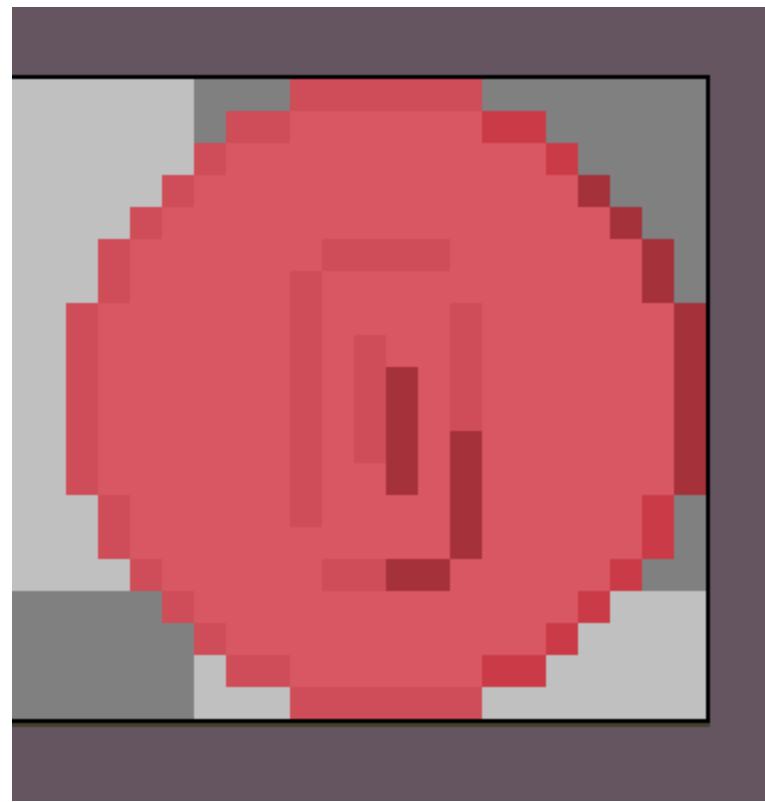
▼ 21.03 - Sprite Überarbeitung - Coin Reset - Verfeinerung der Maps

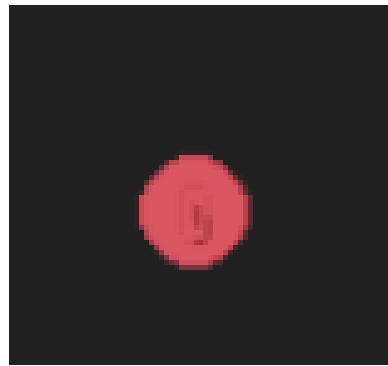
▼ Sprites für Airjump pads überarbeitet





▼ Coin Sprite Anpassung zum UI Icon





### ▼ Reset Coins bei Map Load

```
182     resetCoin() {  
183         while (this.coin != 0) {  
184             this.coin--;  
185         }  
186         this.updateCoinUI();  
187     }
```

```
384 // MAP LOAD  
385 if (texture_index == undefined) {  
386     this.mapLoad = false;  
387     this.resetCoin();  
388     console.log("map");  
389 }  
390 }
```

### ▼ Verfeinerung der Maps

Die Maps wurden grundlegend verfeinert und finalisiert. Visuelles wurde angepasst, texture\_index für Gegner wurde platziert, sinnvolle Platzierung der Herzen, Checkpoints und Coins

### ▼ 22.03 - Zusammensetzung der Maps

- ▼ Neue, überarbeitete Mpas wurden zusammengeführt zu deren jeweiligen Leveln.

- ▼ 25.03 -

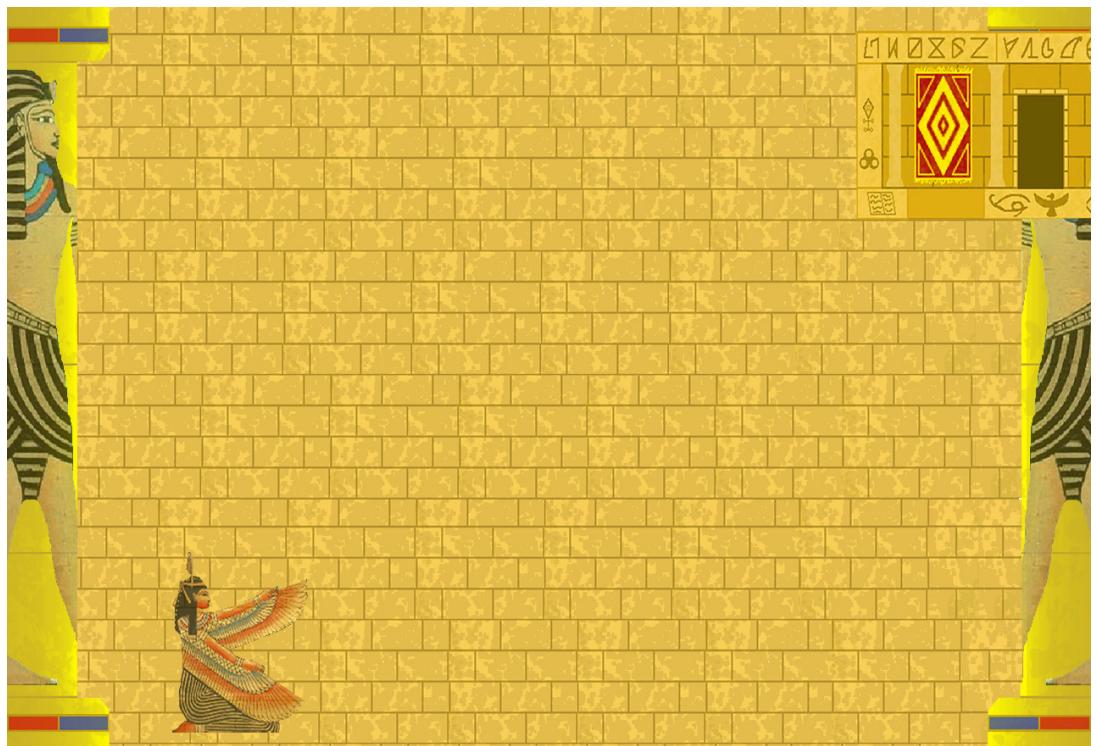
- ▼ 26.03 - Finalsetzung

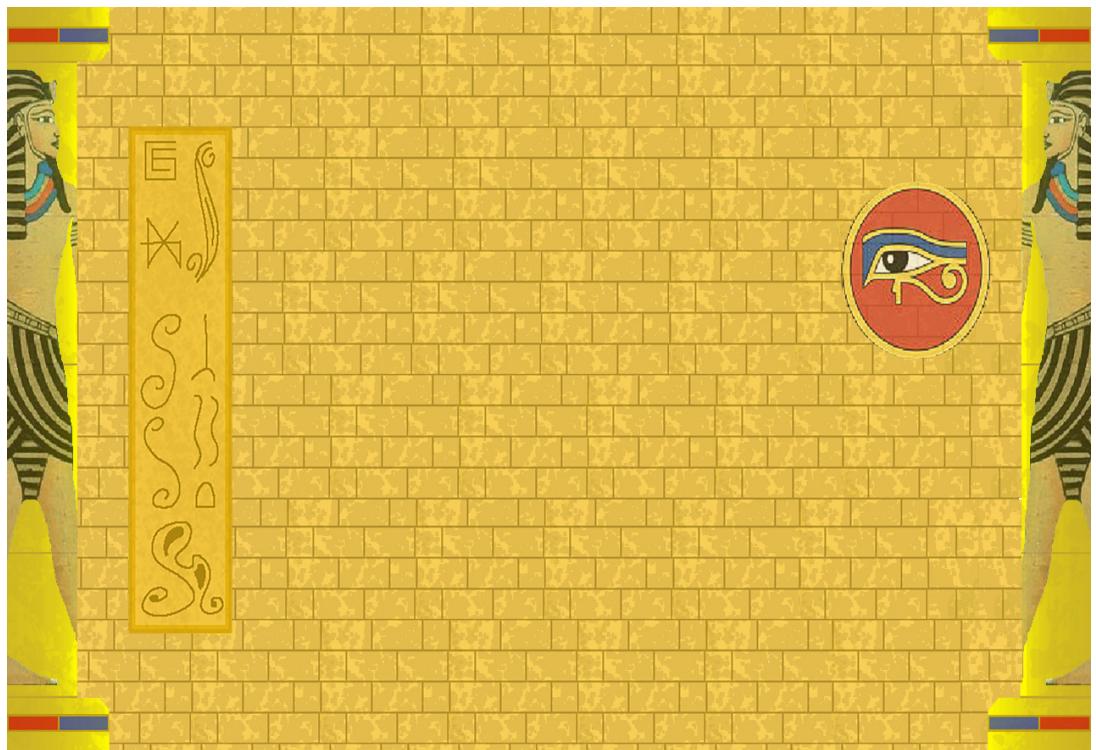
- ▼ Überführung des Finalcodes in ein Komplex

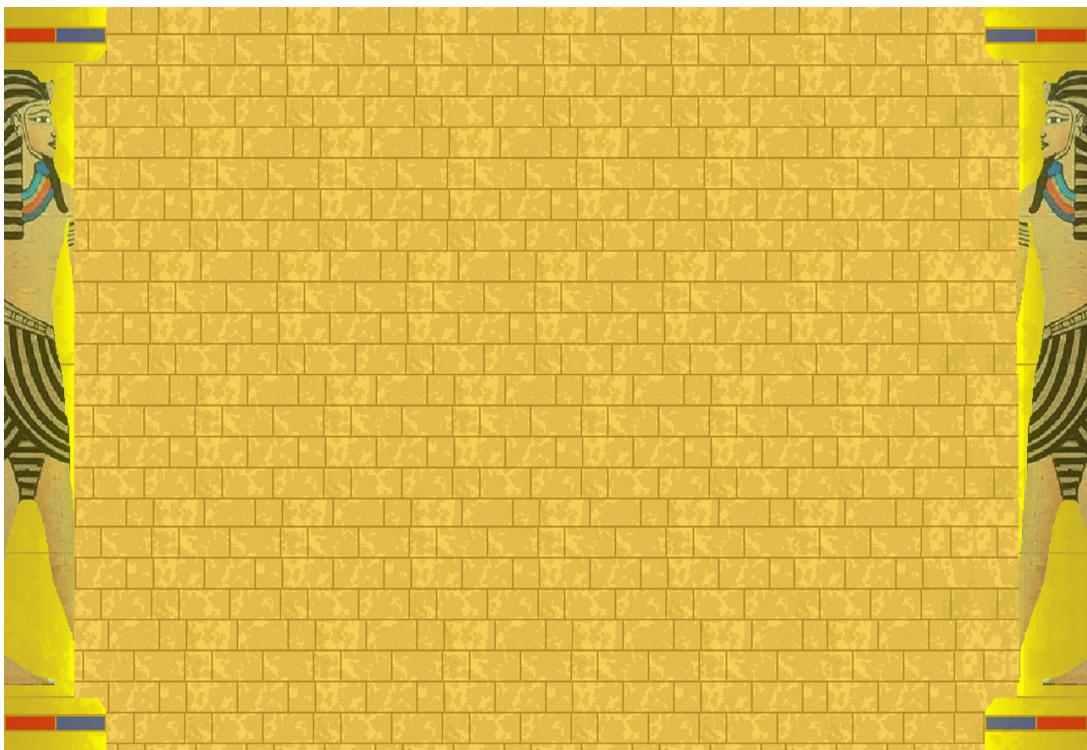
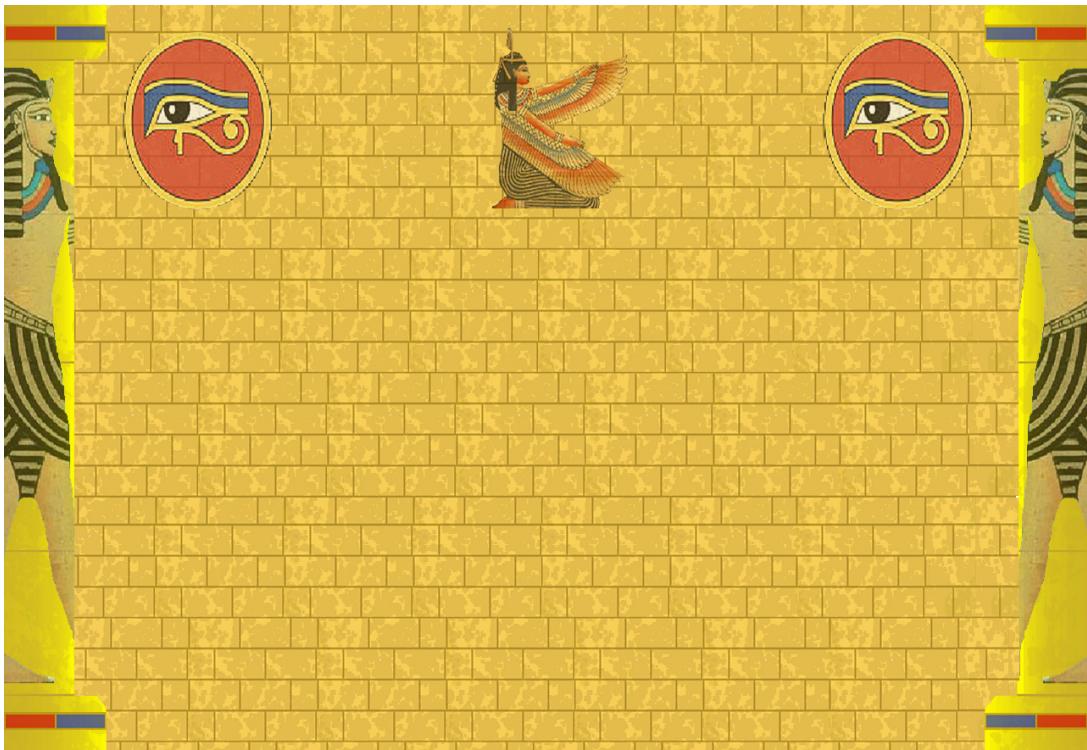
- ▼ Anpassung der Hintergründer für die verschiedenen Level-Abschnitte

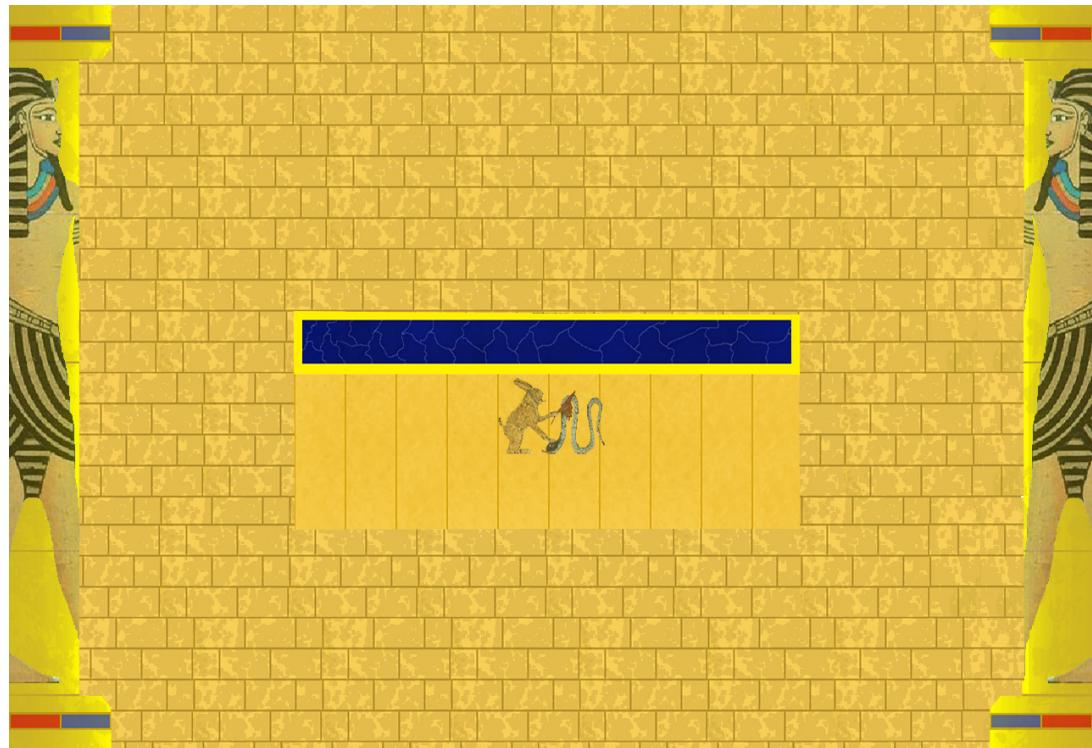
- ▼ Hintergründe











## ▼ Methode zum Entfernen der NPCs im Map Load

```
197     player.draw();  
198  
199     // Gegner und Spikes einzeichnen im neuen Level  
200     if (player.mapLoad) {  
201         npc12one.draw();  
202         npc12two.draw();  
203         npc12three.draw();  
204         npc12four.draw();  
205         npc13one.draw();  
206         npc13two.draw();  
207     }  
208     // Gegner und Spikes entfernen im neuen Level  
209     if (!player.mapLoad) {  
210         // npc24one.draw();  
211         // npc24two.draw();  
212     }  
213 }  
214 }  
215 }
```

## ▼ Generelle Anpassung der Level

### ▼ Fixierung der UI im Screen und CSS Eigenschaften

```
9   * {  
10    user-select: none;  
11  }  
12
```

```
18  canvas{  
19    background-color: transparent;  
20    cursor: none;  
21  }  
22
```

```
23  #health-bar {  
24    position: fixed;  
25    top: 10px;  
26    left: 10px;  
27    width: fit-content;  
28    background-color: □rgba(255, 255, 255, 0.127);  
29    border-radius: 20px;  
30  }  
31  
32  #coins {  
33    position: fixed;  
34    top: 10px;  
35    right: 10px;  
36    width: fit-content;  
37    background-color: □rgba(255, 255, 255, 0.127);  
38    border-radius: 20px;
```

## ▼ 27.03 - Checkpoints - Game End- Titelbildschrim Button

### ▼ Checkpoints

Tileset der Checkpoints wird jetzt verwendet und Checkpoints wurden auf der Map platziert

```
426
427     // Checkpoint De-
428     if(texture_index === 25){
429         respawnX = tileX;
430         respawnY = tileY - tile;
431         // Checkpoint Activated
432         currentCollisionMap[y][x] = 26;
433     }
434
```

## ▼ Game End Screen

```
523
524     gameEnd() {
525         document.getElementById("gameEnd").style.display="flex";
526     }
527
599
400     if (texture_index === 23 && this.items.some(item => item.name === 'key2')) {
401         this.resetCoin();
402         this.gameEnd();
403         this.pauseGame();
404     }
405
```

## ▼ Hauptscreen Button

Beim Spielende und Gameover hat man die Möglichkeit zurück zum Hauptbildschirm zu gelangen

```
58             <div class="game-over" id="gameOverScreen" style="display: none;">
59                 <div class="game-over-inner">
60                     
61                     <div class="game-over-buttons">
62                         <button class="button" onclick="restartGame()">Wiederholen</button>
63                         <button class="button" onclick="returnToMainMenu()">Zurück zum Hauptmenü</button>
64                     </div>
65                 </div>
66             </div>
67
```

```
306  
307 | // Funktion für die Rückkehr zur Levelauswahl  
308 | function returnToMainMenu() {  
309 |     isGameOver = false;  
310 |     window.location.href = "../pages/index.html"  
311 | }  
312 | setTimeout(checkGameOver, 5000);|
```