

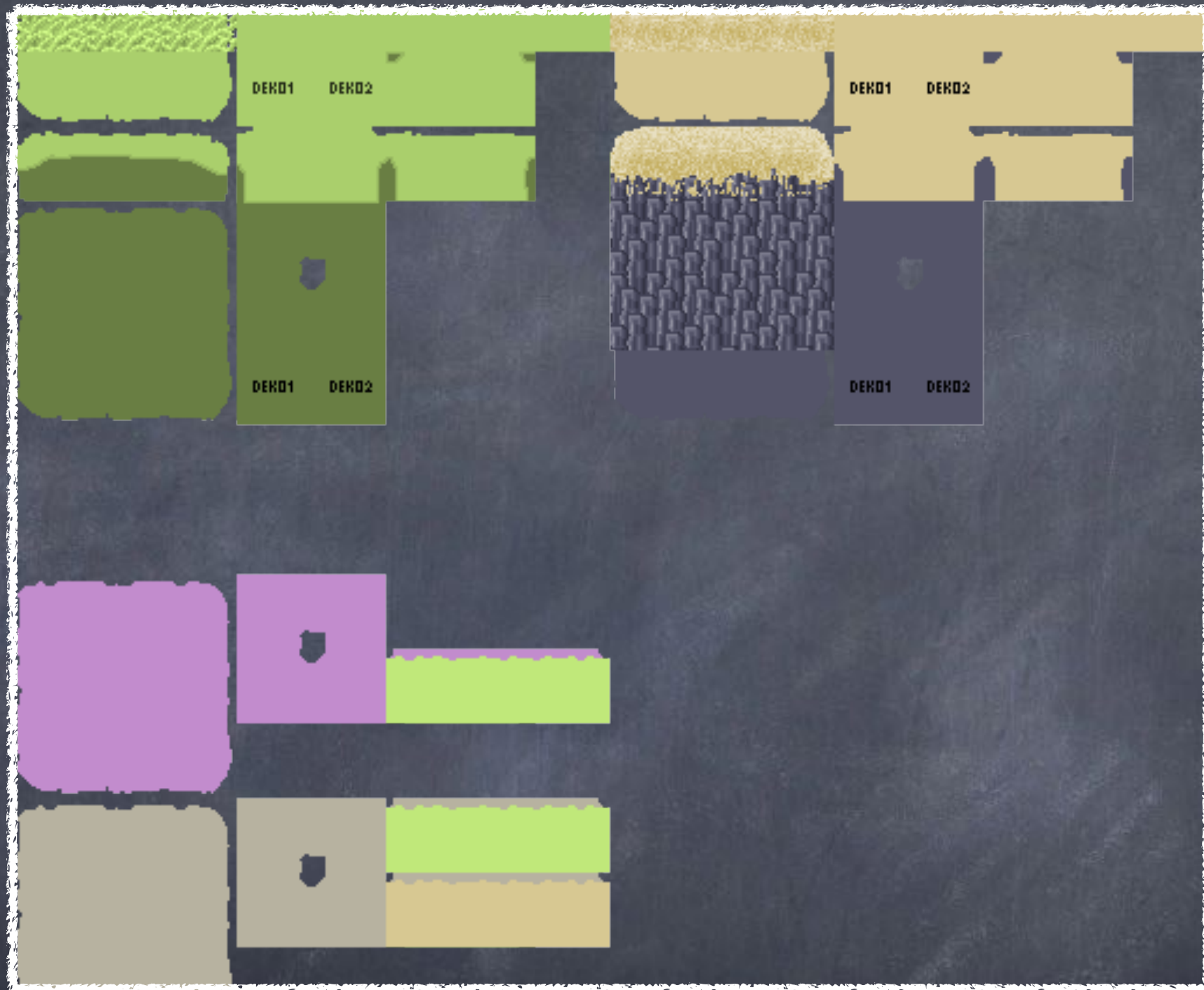
# Quallenfischen



# Probleme

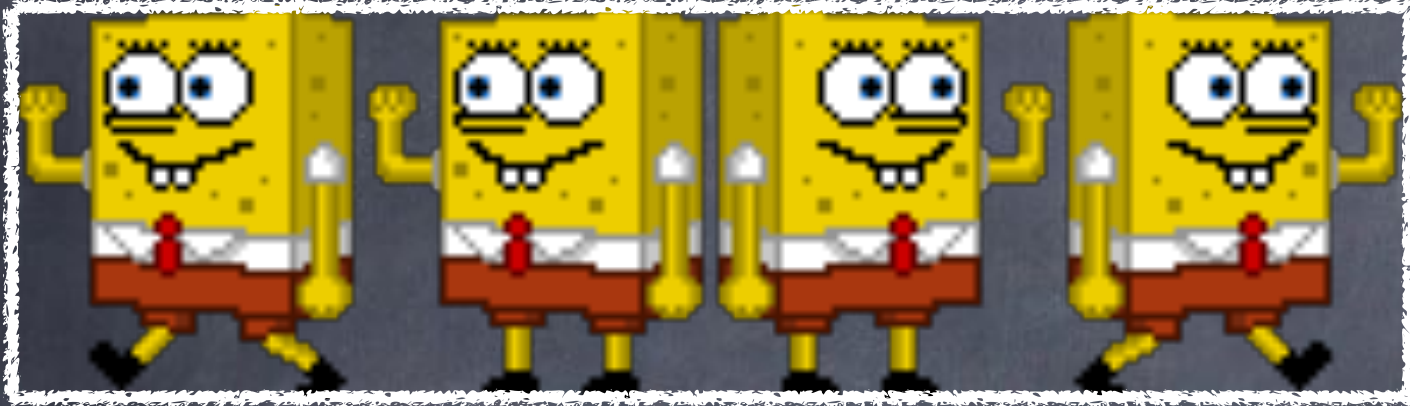
- Health-Anzeige nicht vollständig funktionsfähig
- Quallenposition ist noch random
- Kescher folgt Spongeboy nur auf dem Boden





Tiles

# Objekte und Sprites





Musik

```

37
38 //Funktion zum Erstellen des Spielers
39 function createPlayer(){
40     // Player wird erstellt
41     player = game.add.sprite(40,game.height +850, "player"); // 0,0 lässt Zwiebelkopf in die Welt hineinfallen
42     player.inputEnabled =true;
43     player.input.enableDrag();
44
45
46
47     // Player Physic wird geladen
48     game.physics.enable(player, Phaser.Physics.ARCADE);
49
50     // Pysic für den Player wird gesetzt
51     player.body.bounce.y = 0.1;
52     player.body.collideWorldBounds = true;
53
54     // Animation erstellen
55     player.animations.add("left",[0,1,2,3],10, true);
56     player.animations.add("right",[5,6,7,8],10,true);
57
58     // Die Kamera folgt dem Spieler
59     game.camera.follow(player);
60 }
61

```

# Probleme

```

132
133 function createCatcher(){
134
135     catcher = game.add.sprite(player.x , player.y-2 ,'catcher');
136     catcher.scale.setTo(0.1,0.1);
137     catcher.anchor.set(0.5);
138     catcher.inputEnabled = true;
139     catcher.input.enableDrag();
140     game.physics.enable(catcher, Phaser.Physics.ARCADE); //Problem
141     //catcher.body.collideWorldBounds = true;
142
143
144 }
145

```



```
65  
66 function update() {  
67  
68     game.physics.arcade.collide(player, layer);  
69     updatePlayerControl();  
70  
71  
72  
73     //game.physics.arcade.collide(jellyfish, wLayer);  
74  
75     // game.physics.arcade.collide(jellyfish, layer);  
76     game.physics.arcade.overlap(catcher, jellyfish, collectJellyfish, null, this); //problem  
77     game.physics.arcade.overlap(player, jellyfish, killPlayer, null, this);  
78     catcher.x = Math.floor(player.x); //Positioniert den Catcher vor dem Dude --> DER ÜBELTÄTER!!! >.<  
79     catcher.y = Math.floor(player.y);  
80  
81
```

```

246 function createCatcher(){
247
248     catcher = game.add.sprite(40, game.height+450,"catcher");
249     catcher.scale.setTo(0.1,0.1);
250     catcher.anchor.setTo(0.5);
251     game.physics.enable(catcher, Phaser.Physics.ARCADE); //Problem
252     catcher.body.bounce.y = 0.1;
253
254     //catcher.immovable = true;
255
256     catcher.body.collideWorldBounds = true;
257 }

```

```

259 function updateCatcherControl(){ // Idee für weitere Bearbeitung: updatePlayerControl für die Gruppe armedPlayer überarbeiten
260     // Resetet die Player Velocity
261     catcher.body.velocity.x = 0;
262
263     if(cursors.left.isDown)
264     {
265         // Bewegt sich nach links
266         catcher.body.velocity.x = -150;
267     }
268
269     else if (cursors.right.isDown)
270     {
271         // Bewegt sich nach rechts
272         catcher.body.velocity.x = 150;
273     }
274
275     else
276     {
277         //Still stehenbleiben
278         catcher.animations.stop();
279         catcher.frame = 2;
280     }

```

```

259 function updateCatcherControl(){ // Idee für weitere Bearbeitung: updatePlayerControl für die Gruppe armedPlayer überarbeiten
260     // Resetet die Player Velocity
261     catcher.body.velocity.x = 0;
262
263     if(cursors.left.isDown)
264     {
265         // Bewegt sich nach links
266         catcher.body.velocity.x = -150;
267     }
268
269     else if (cursors.right.isDown)
270     {
271         // Bewegt sich nach rechts
272         catcher.body.velocity.x = 150;
273     }
274
275     else
276     {
277         //Still stehenbleiben
278         catcher.animations.stop();
279         catcher.frame = 2;
280     }
281
282     // Erlaubt das Springen mit der Pfeiltaste ODER Leertaste, wenn der boden berührt wird.
283     if ((sprungButton.isDown || cursors.up.isDown && catcher.body.onFloor()))
284     {
285         catcher.body.velocity.y = -200; //spring noch zu hoch :/
286     }
287
288 }

```



```
66 function update() {  
67  
68     game.physics.arcade.collide(player, layer);  
69     updatePlayerControl();  
70  
71  
72  
73     //game.physics.arcade.collide(jellyfish, wLayer);  
74  
75     // game.physics.arcade.collide(jellyfish, layer);  
76     game.physics.arcade.overlap(catcher, jellyfish, collectJellyfish, null, this); //problem  
77     game.physics.arcade.overlap(player, jellyfish, killPlayer, null, this);  
78     catcher.x = Math.floor(player.x); //Positioniert den Catcher vor dem Dude --> DER ÜBELTÄTER!!! >.<  
79     catcher.y = Math.floor(player.y);  
80  
81
```

```
370 //ERSTELLT DEN SCORE & HEALTH TEXT OBEN LINKS  
371 function createScoreBar(){  
372  
373     scoreText = game.add.text(60, 16, 'Score: 0', { fontSize: '32px', fill: '#000' });  
374     scoreText.anchor.set(0.5);  
375     scoreText.fixedToCamera = true;  
376  
377     lifeText = game.add.text(92, 40, 'Health: 100 %', { fontSize: '32px', fill: '#000' });  
378     lifeText.anchor.set(0.5);  
379     lifeText.fixedToCamera = true;  
380 }
```

# Pflicht (Muss)

- Tastenfunktionen Kescher
- Anpassung des Hintergrundes
- Einbinden der Sprunganimation
- Level 2
- Einbinden der Todesanimation
- Game-Over-Bildschirm
- Startbildschirm
- Animation des Keschers
- Einbinden des Tutorials
- Einfügen eines Lautstärke- und Pausebuttons
- Sounds: Wassersound, Sound bei Spongeboys tot, Blubbersound, Krabbenburgersound, Gameoversound, Keschersound, Mr. Cancer Sound
- Einfügen der Funktion, um das Spiel gewinnen zu können



# Optional (Kann)

- Tastenführung auf den WASD-Tasten
- dynamische Backgrounds
- Level 3

# Nicht

- Multiplayer-Game
- Mehr als 3 Level



# Demnächst

- Problemlösung mit Kescher und Player
- Verbesserung der Bewegung/Animation der Quallen (Schweben statt Bouncen)
- Verbesserung der Health-Anzeige bei Reset
- Bearbeitung der Projekt-Pflichtziele
- Einbinden des Krosse-Krabbe-Sprites
- Geschwindigkeit von Spongeboy

Score: 3  
Health: 100 %

49

