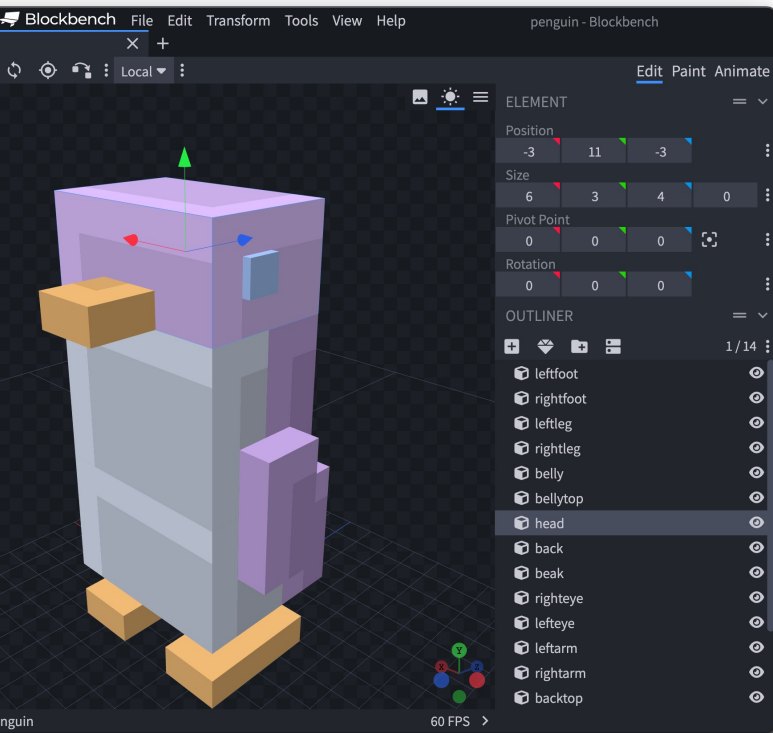


S4F Projekt von Daniel

Crossy Clone



```

1  const createGroup = model => {
2    const group = new Group();
3    const { elements, rotation, scale, meshOffset, groupOffset } = model;
4
5    if (!elements?.length) throw new Error("No elements found for " + model.name);
6
7    elements.forEach(element => {
8      const { name, from, to, color } = element;
9
10     // create material
11     const materialColor = parseInt(color, 16);
12     const material = new MeshStandardMaterial({ color: materialColor }); // convert from string hex to int
13
14     // create geometry
15     const [x1, y1, z1] = from.map(v => v / 16);
16     const [x2, y2, z2] = to.map(v => v / 16);
17
18     const dx = x2 - x1;
19     const dy = y2 - y1;
20     const dz = z2 - z1;
21
22     const geometry = new BoxGeometry(dx, dy, dz);
23
24     // create mesh
25     const mesh = new Mesh(geometry, material);
26     mesh.position.set(x1 + dx / 2, y1 + dy / 2, z1 + dz / 2);
27     mesh.name = name;
28
29     // offset mesh
30     if (meshOffset) mesh.position.add(new Vector3(...meshOffset));
31
32     // shadows
33     mesh.castShadow = true;
34     mesh.receiveShadow = true;
35
36     group.add(mesh);
37   });
38
39   if (rotation) group.rotation.set(0, rotation, 0);
40   if (scale) group.scale.set(scale, scale, scale);
41   if (groupOffset) group.position.set(...groupOffset);
42
43   const wrapper = new Group();
44   wrapper.add(group);
45
46   return wrapper;
47 };

```

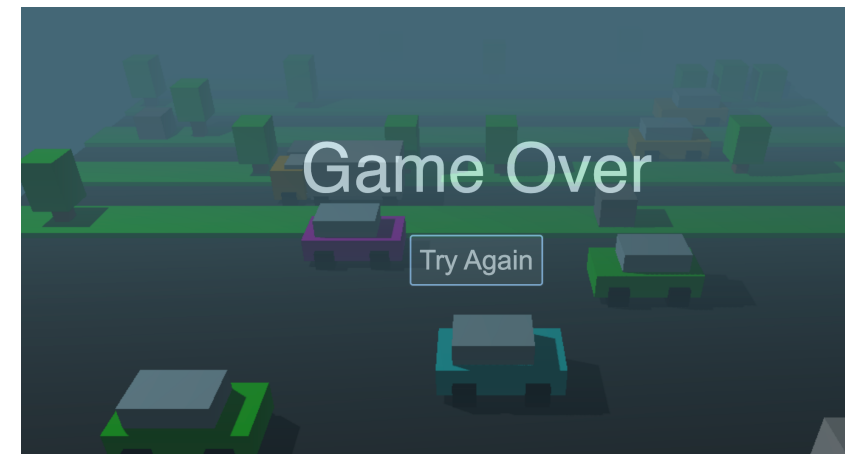
```

1  const laneManager = new LaneManager();
2  const player = new Player(laneManager);
3
4  player.onLaneAdvance = () => {
5    laneManager.addLane();
6    light.position.x += 1;
7    cameraController.jump();
8    document.querySelector("#score").innerHTML = `Score: ${player.score}`;
9  };
10
11  player.onFirstLaneAdvance = () => {
12    cameraController.enable();
13  };
14
15  cameraController.onCatchup = () => {
16    player.die();
17  };
18
19  player.onDeath = () => {
20    cameraController.disable();
21    gameOver();
22
23    if (player.score > highscore) {
24      localStorage.setItem("highscore", player.score);
25      document.querySelector("#highscore").innerHTML = `High: ${player.score}`;
26    }
27  };

```

Mein Projekt

- Crossy Road Clone mit JavaScript
- THREE.JS
- Blockbench
- Möglichst viel Code selber schreiben





Wo stehe ich jetzt?

- **Es funktioniert**
- Eigene 3D Modelle mit Blockbench
- Score / Highscore wird angezeigt und lokal gespeichert
- Deployment
- Keine Soundeffekten
- Keine Handysteuuerung
- Kein "Title Screen"
- Keine Partikel Effekte und Animationen
- Es gibt nur einen Charakter

Soll- Istvergleich

Crossy Clone			
Schritt	Soll	Ist	Erreicht
Dokumentation erstellen	04.04.22	04.04.22	ok
Repository einrichten auf GitHub	09.05.22	04.04.22	ok
Deployment einrichten	09.05.22	29.04.22	ok
Szene einrichten	23.05.22	28.04.22	ok
Generation von Fahrzeugspuren	04.07.22	28.04.22	ok
Spieler Controller	04.07.22	28.04.22	ok
3D Modelle für Fahrzeuge	13.06.22	29.04.22	ok
Collision detection	13.06.22	28.04.22	ok
Präsentation Zwischenstand	30.05.22		ok
Sound Design	27.06.22		ok
Game Over Screen	27.06.22	02.05.22	ok
Options Screen	11.07.22		ok
Schluss-Präsentation	27.06.22		ok
Dokumentation nachführen	11.07.22		ok

Deployment

- Webpack und GH-Pages
- <https://cuddlybunion341.github.io/crossy-road-clone/public/>

```
1  const path = require("path");
2  const TerserPlugin = require("terser-webpack-plugin");
3
4  module.exports = {
5    mode: "production",
6    entry: "./src/main.js",
7    output: {
8      path: path.resolve(__dirname, "public"),
9      filename: "bundle.js",
10   },
11   optimization: {
12     minimize: true,
13     minimizer: [new TerserPlugin({ extractComments: false })],
14   },
15   devServer: {
16     compress: true,
17     port: 9000,
18     hot: true,
19   },
20 };
21
```



Fazit

Positives

- Ich konnte viel neues mit THREE.JS lernen
- Ich habe code bundling und deployment von JavaScript gelernt
- Ich habe gelernt wie man mit npm arbeitet

Negatives

- Projekt ein bisschen zu einfach
- Projektplanung könnte besser sein