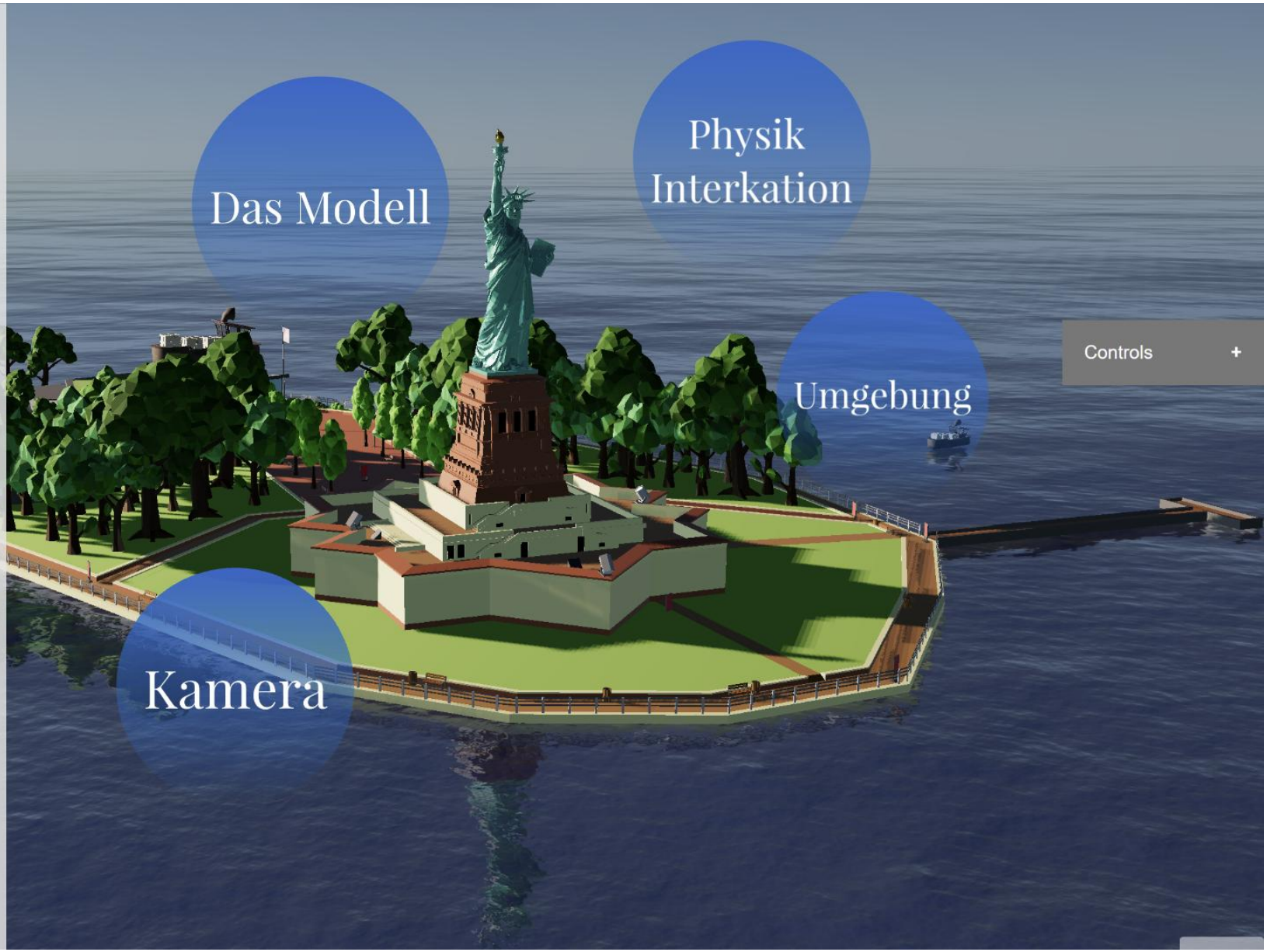
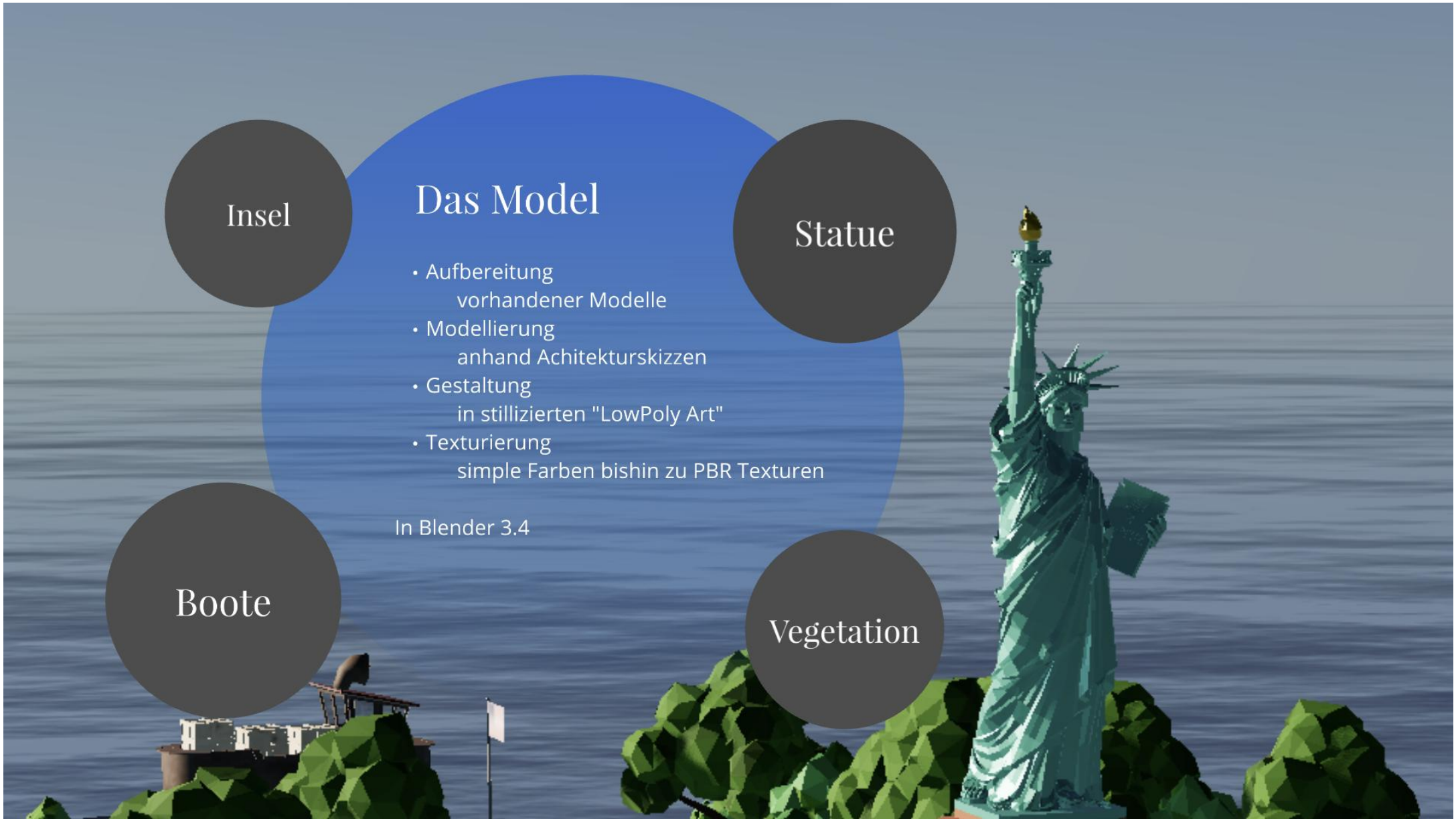
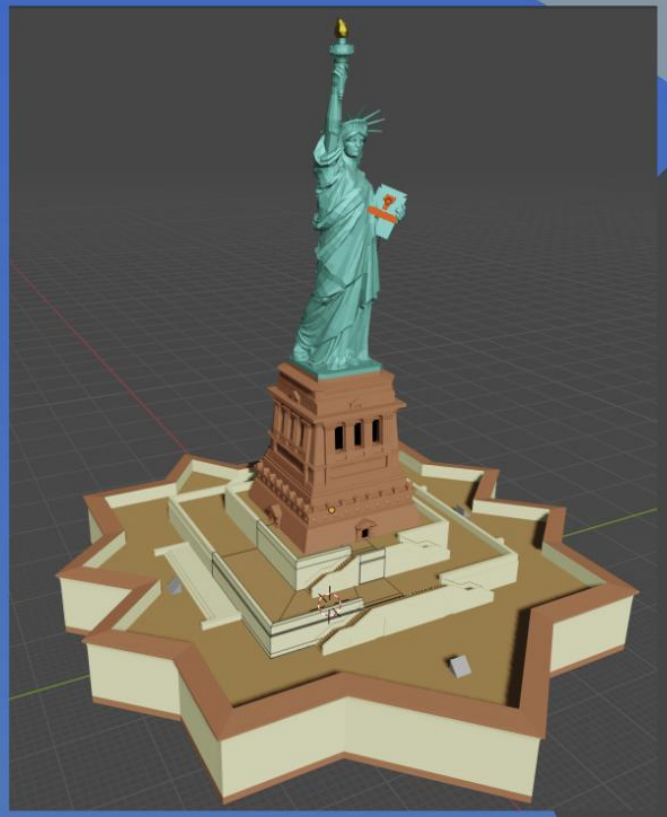
Liberty Island

Computergrafiken

Christoph Gründer
Yannic Grafwallner
Gabriel Wuwer




Insel
Das Model
Statue
• Aufbereitung
vorhandener Modelle
• Modellierung
anhand Achitekturskizzen
• Gestaltung
in stillizierten "LowPoly Art"
• Texturierung
simple Farben bishin zu PBR Texturen
In Blender 3.4
Boote
Vegetation

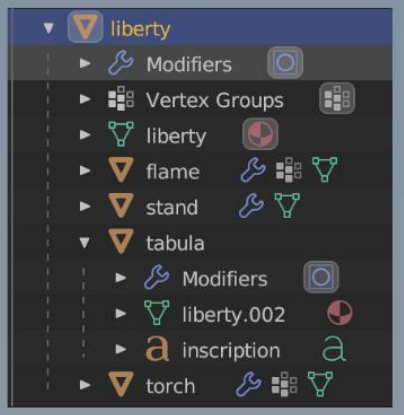


Statue

1. Nachberbeitet
   Gesicht, Robe, Fackel,
   Form, Normals, Gruppierung,

2. Modellieren
   Sockel, Podest, Tabula,
   Mirror Modifier,

3. Materialien
   Metallisch: Korrodiertes Kupfer, Gold
   Matt: Sandstein, Beton, Steinplatten



Vegetation

1. LowPoly
   Mit geringer Polygonanzahl modelliert
   und kleinen Texturen texturiert

2. Partikel Effekt
   Für vielseitige plazierung und Baumart
   Zeitersparrung bei der Platzierung

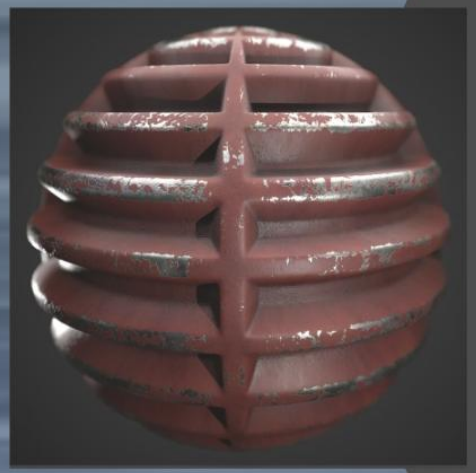
Insel

Starke orientierung an realer Insel mit gestalterischer Freiheit

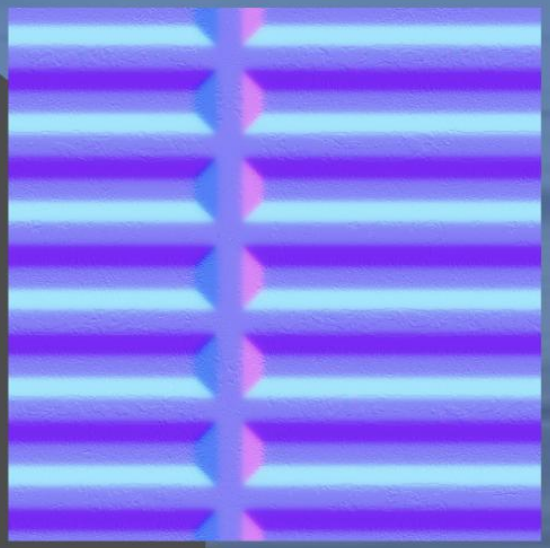
"One Object" Modell

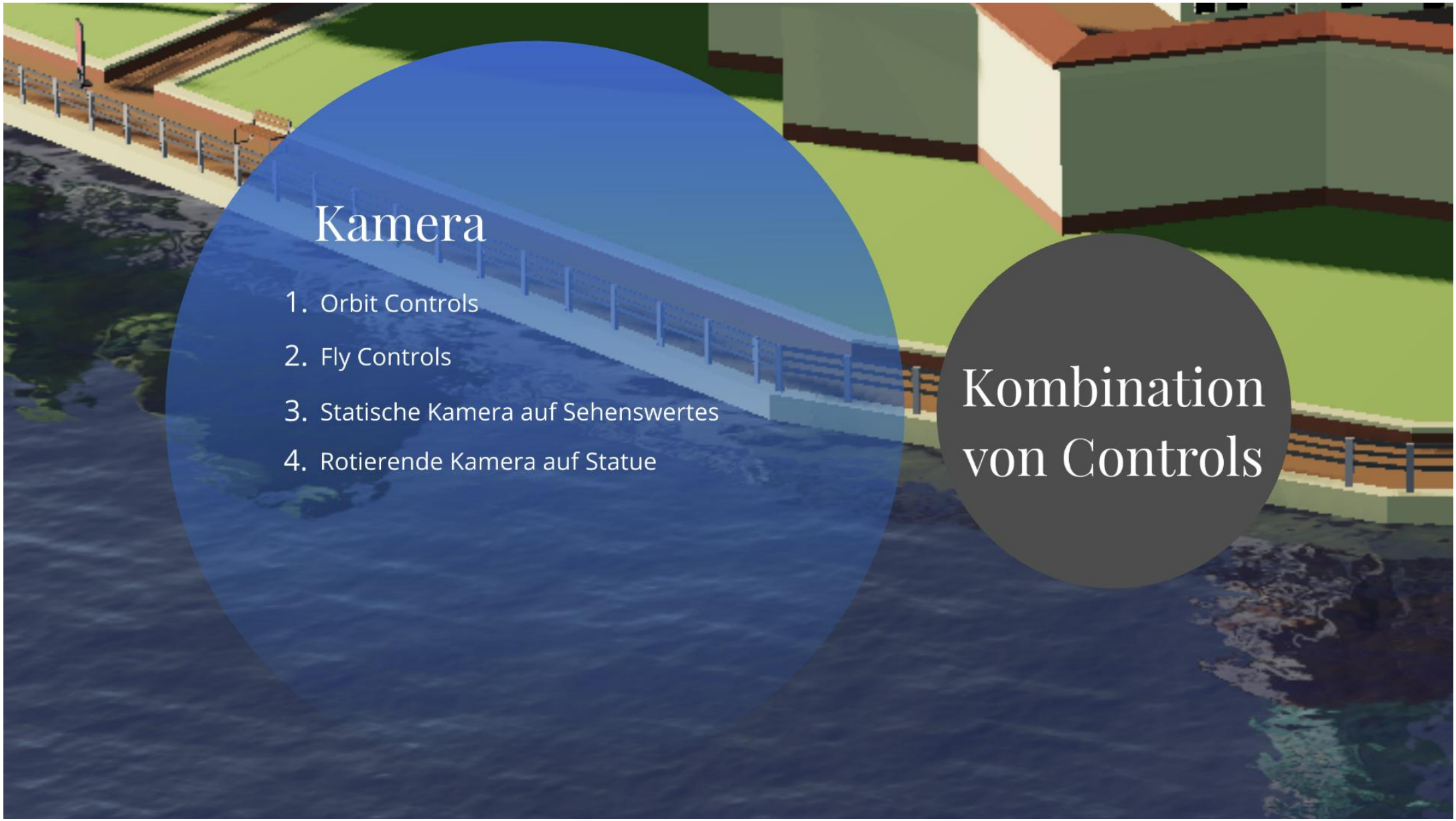
Details seperat modelliert

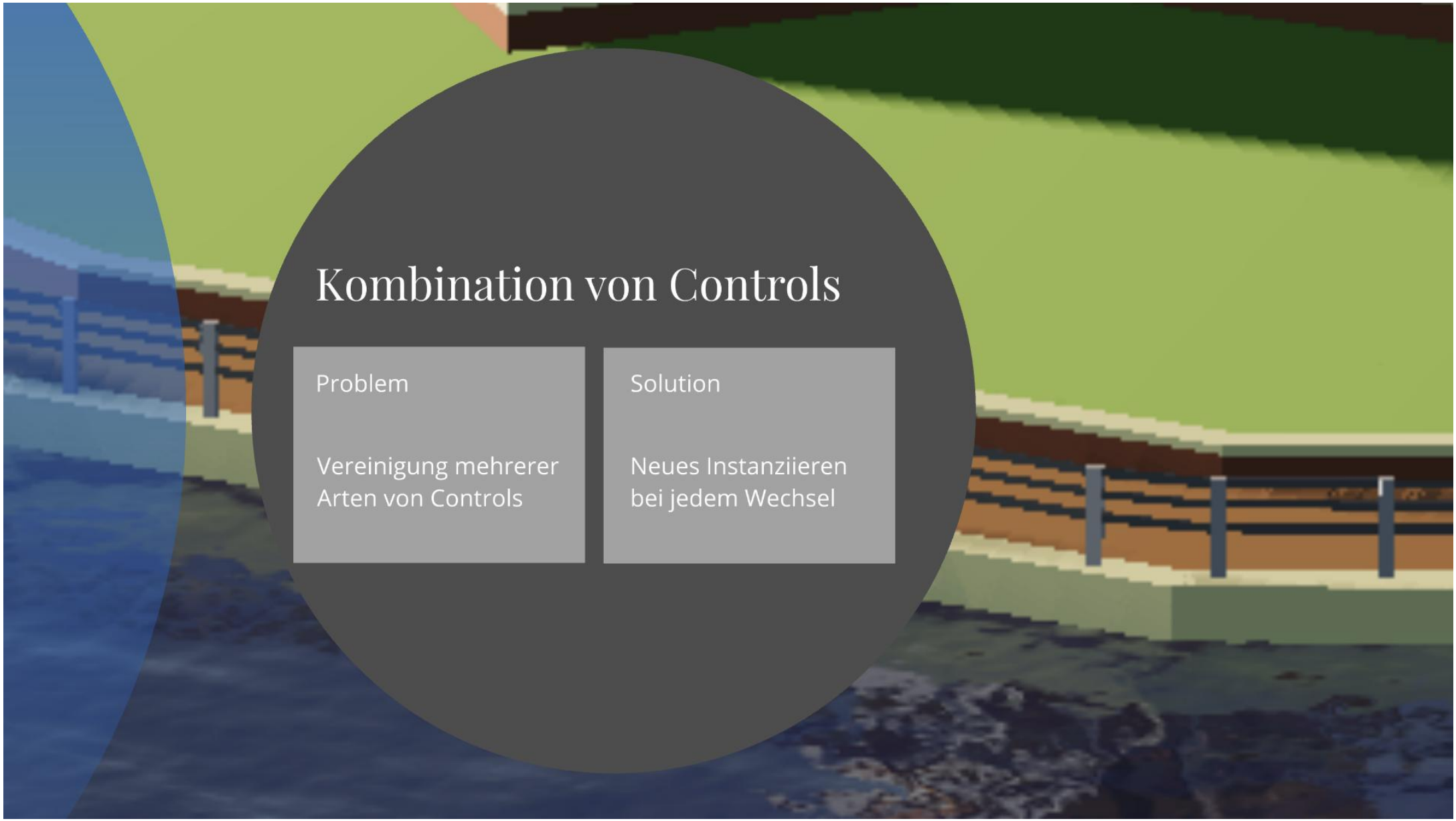
Boote

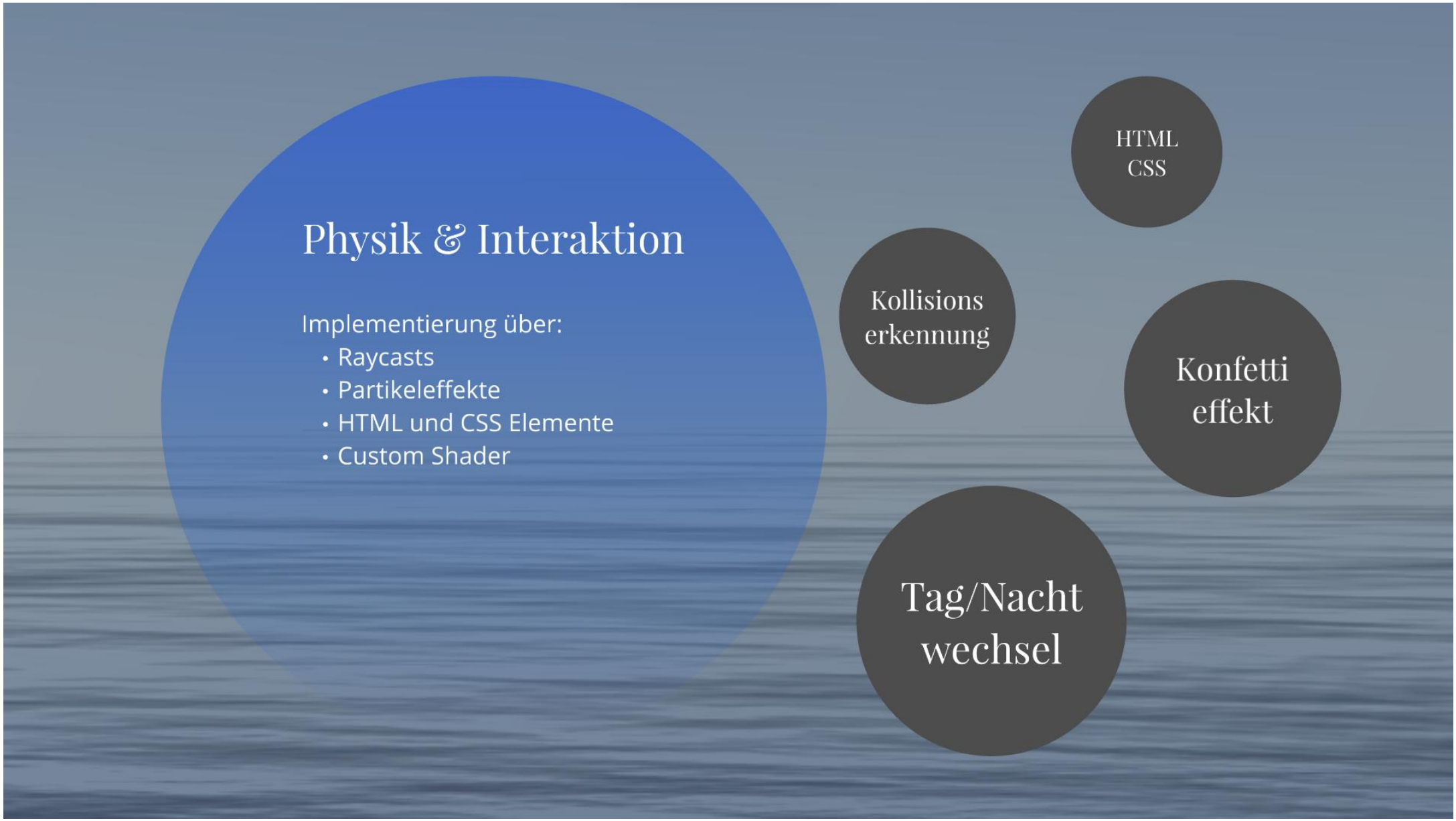


1. Modellierung mit Flächen und Zylindern
2. Animation in ThreeJS
3. Texturierung mit PBR Texturen




Kamera
1. Orbit Controls
2. Fly Controls
3. Statische Kamera auf Sehenswertes
4. Rotierende Kamera auf Statue
Kombination von Controls


Kombination von Controls
Problem
Vereinigung mehrerer Arten von Controls
Solution
Neues Instanziieren bei jedem Wechsel


Physik & Interaktion
Implementierung über:
• Raycasts
• Partikeleffekte
• HTML und CSS Elemente
• Custom Shader
HTML
CSS
Kollisions
erkennung
Konfetti
effekt
Tag/Nacht
wechsel

Kollisionserkennung

Entfernung von Objekten vor Kammera mithilfe von Raycasts gemessen

Bei Unterschreitung, nur noch zurück fliegen

```
function collisionDetected(freeCollisionKey) {
    controls.enabled = false;
    controls.dispose();
    console.log("Press " + freeCollisionKey + " to free the camera");
    document.getElementById("status").textContent="Tap S";

    // Wait for keypress s before enabling controls again
    document.addEventListener("keydown", function freeCollision(event) {
        if (event.key === freeCollisionKey) {
            timeAfterCollision = Date.now();
            controls = initControls(camera, renderer, controls);
            controls.enabled = true;
            document.removeEventListener("keydown", freeCollision);
            document.getElementById("status").textContent="Flight";
        }
    });

}
```

```
function animate() {
    const delta = clock.getDelta();
    if (controls instanceof FlyControls) {
        ray.setFromCamera(new THREE.Vector3(0, 0, 0), camera);
        var collisionResults = ray.intersectObjects(scene.children, true);
        if (collisionResults.length > 0 && collisionResults[0].distance < 10 && Date.now() - timeAfterCollision
            collisionDetected("s");
```

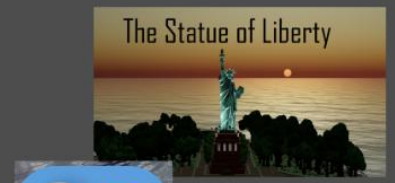
HTML,CSS

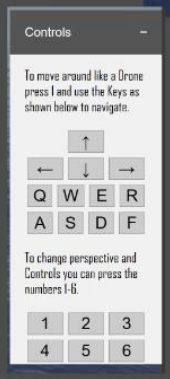
Überschrift

Tag/Nacht switch

Statusbutton

Controlpanel





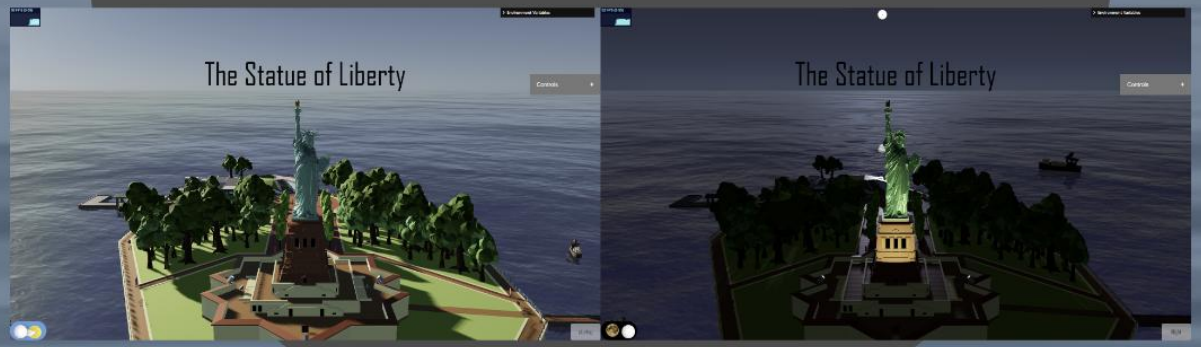
Konfetti Effekt

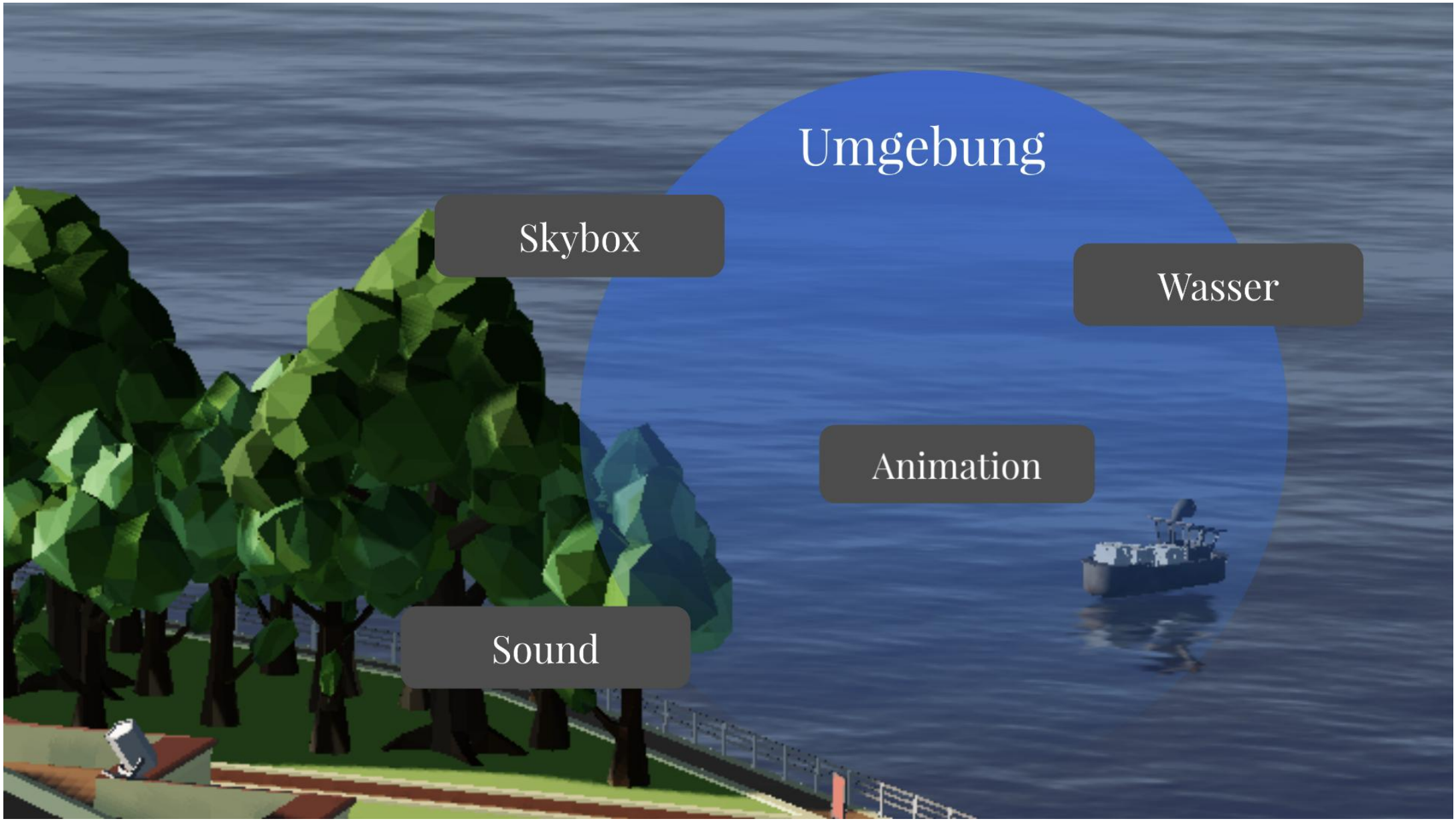
- Umgesetzt über BufferGeometry
- gewichtete Random Bewegung
- Zufällige Farben und Startpositionen

```
document.addEventListener( type: 'keydown', listener: function(event : KeyboardEvent ) {
    if (event.code === 'Space') {
        raycaster.setFromCamera( pointer, camera );
        intersects = raycaster.intersectObjects( scene.children );
        let position = new THREE.Vector3(intersects[0].point.x, intersects[0].point.y, intersects[0].point.z);
        confettiParticles.push(createConfetti(position,confetti));
        if (confettiParticles.length > 30){
            confetti.remove(confetti.children[0]);
            confettiParticles.shift();
        }
    }
});
```

Tag/Nacht Wechsel

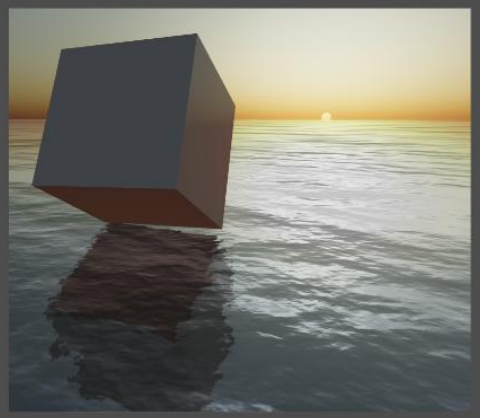
- Steuerbar über Toggle-Button
- Einschalten der Spotlights
- Anpassung der Parameter von Sonnenstrahlen zu Mondschein




Umgebung
Skybox
Wasser
Animation
Sound

Wasser

- Umgesetzt mit ThreeJS Water Klasse
- Wassertextur mit angepasster Farbe und Verzerrungswert



Skybox

- Umgesetzt über die ThreeJS Sky Klasse
- Zusätzliche Lichtquelle
  - Selbe Richtung wie Sonne
- Anpassung bei Tag und Nacht

Sound

- Möwengeschrei und Meeresrauschen
- Bei Nacht Musik
- Umgesetzt über ThreeJS Audio Klasse

```
let aliciaKeys=startSound(camera,
    path: 'public/Sounds/Empire State of Mind (Part II) Broken Down.mp3',
    audioloader,
    listener ,
    playwithload: false);
```

Animation

- Animation der Boote in ThreeJs umgesetzt
- Pfadberechung durch mathematische Funktion + synchrone Drehung

```
function render() {
    const time = performance.now();
    rotate(sailboat, -time, radius: .4, Math.PI, startX: -100,
        startY: 0.6, startZ: 0);
    rotate(cargoship, renderZaehler: time * 2 / 3, radius: .9, startausrichtung: 0, startX: 150,
        startY: 1, startZ: -50);
    rotate(yanBoat, renderZaehler: time / 8, radius: 9.5, Math.PI, startX: 50,
        startY: 1.1, startZ: -100);
    water.material.uniforms['time'].value += 1.0 / 60.0;
    renderStats()
    renderer.render(scene, camera);
}
```

```
export function rotate(obj, renderZaehler, radius, startausrichtung, startX,startY, startZ){
    if(obj===null)return;
    obj.rotation.y = renderZaehler/1000+startausrichtung;
    obj.position.y=startY;
    obj.position.x = startX+Math.sin( x: renderZaehler/1000+startausrichtung)*radius*25;
    obj.position.z = startZ+Math.sin( x: renderZaehler/1000+Math.PI/2+startausrichtung)*radius*20
}
```