

## Aufgabe 1

### Lorem Ipsum



Abbildung 1: Ergebnis Aufgabe 1

- **Aufgabe:**

- Laden Sie die Datei *Beispielprojekt Woche 4* aus Moodle herunter welche bereits das Inselobjekt beinhaltet.
- Legen Sie eine Ordnerstruktur an, welche der Folgenden entspricht:

```
Insel
├── common
│   ├── initShaders.js
│   └── objects3D.js
├── index.html
└── main.js
```

- **Tipps:**

- Achten Sie darauf, die objects3D.js Datei in der HTML Datei durch

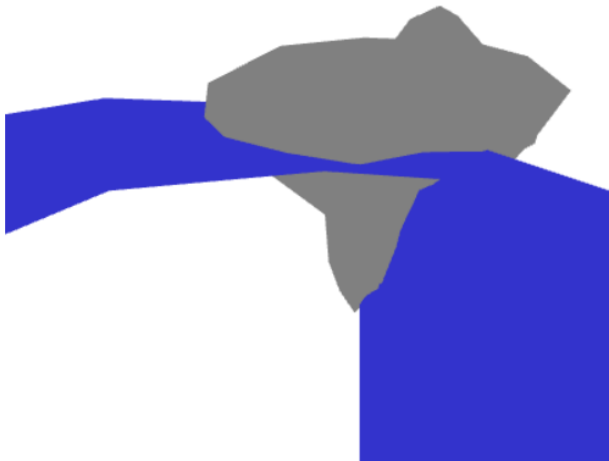
```
<script type="text/javascript" src="common/objects3D.js"></script>
```

einzubinden.

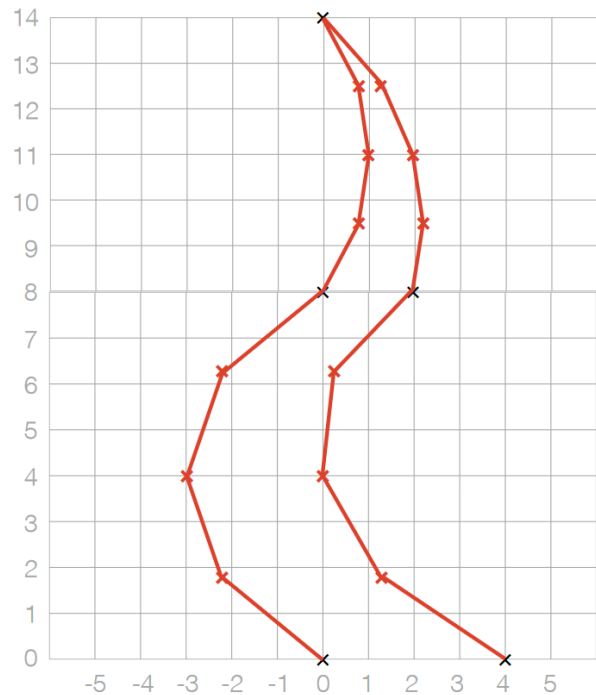
---

## Aufgabe 2

### Lorem Ipsum



(a) Ergebnis Aufgabe 2



(b) Der Fluss für eure südsee Insel

- **Aufgabe:**

- Die in Aufgabe 1 heruntergeladene Zip-Datei beinhaltet bereits das Insel- und das Fluss-Objekt, allerdings wird bis jetzt nur die Insel dargestellt. Schreiben Sie den fehlenden Programmcode der Funktionen **initBuffers2** und **render2** um das Flussobjekt zu visualisieren.

- **Tipps:**

- Versuchen Sie den Initialisierungs- und Render-Code der Insel zu verwenden um den Fluss zu rendern.
- Achten Sie auch darauf nicht den Vertex- / Index- Buffer der Insel zu überschreiben und die Buffer der jeweiligen Objekte vor dem Rendern zu binden.

- **Wichtige Funktionen:**

```
gl.bindBuffer(...);  
gl.enableVertexAttribArray(...);  
gl.vertexAttribPointer(...);  
gl.drawElements(...);
```

---

## Aufgabe 3

- **Aufgabe:**

- Um Codeduplikation zu vermeiden, soll nun eine Basisklasse **Object3D** geschrieben werden, welche sich um das Speichern und Rendern der Objekte kümmert. Diese soll formal folgende Schnittstelle bieten:

- \* Einen Konstruktor, welcher die benötigten OpenGL Buffer allokiert.
- \* Eine Methode *InitBuffers*, welcher die Vertex- und Indexdaten an OpenGL übergibt.
- \* Eine Methode *Render* um das Objekt darzustellen.

Die geschriebene Klasse soll dann folgenden Test-Case erfüllen um einen Würfel rendern.

```
// objects3d.js
class Cube extends Object3D {
  constructor() {
    super();

    this.positions = [
      -1, -1, -1, // index 0
      -1, -1,  1, // index 1
      -1,  1, -1, // index 2
      -1,  1,  1, // index 3
      1, -1, -1, // index 4
      1, -1,  1, // index 5
      1,  1, -1, // index 6
      1,  1,  1, // index 7
    ];
    this.indices = [
      1, 7, 3, 1, 5, 7, // Front
      5, 6, 7, 5, 4, 6, // Right
      4, 2, 0, 4, 6, 2, // Back
      0, 3, 2, 0, 1, 3, // Left
      0, 5, 1, 0, 4, 5, // Bottom
      3, 6, 2, 3, 7, 6, // Top
    ];
    this.colors = [
      0, 0, 0, 1, // index 0
      1, 0, 0, 1, // index 1
      0, 1, 0, 1, // index 2
      0, 0, 1, 1, // index 3
      1, 1, 0, 1, // index 4
      1, 0, 1, 1, // index 5
      0, 1, 1, 1, // index 6
      1, 1, 1, 1, // index 7
    ];
    this.InitBuffers();
  }
}

// main.js
function main() {
  ...

  // Only clear once
  gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);

  let cube = new Cube();
  cube.render();
}
```

- **Tipps:**

- Denken Sie daran vor dem Rendern der Objekte alle zu verwendenden Buffer mit `gl.bindBuffer(...)` korrekt zu binden

- **Wichtige Funktionen:**

```
gl.bindBuffer(...);
gl.enableVertexAttribArray(...);
gl.vertexAttribPointer(...);
gl.drawElements(...);
```

---

## Aufgabe 4

- **Aufgabe:**

- Implementieren Sie nun die Insel und den Fluss als einzelne Klassen welche von Objekt3D erben. Die Implementierung soll dabei so erfolgen, dass alle Objekte in einer Liste gefügt und dann mittels dieser gerendert werden.

- **Tipps:**

```
// main.js
let objects = [];
function main() {
    ...

    objects.push(new Island());
    objects.push(new River());
}

function render() {
    // clear buffers
    ...
    // Call render function of each scene object
    for(let object of objects) {
        ...
    }
}
```

## Aufgabe 5 (Optional)

- **Aufgabe:**

- Überlegen Sie sich die Vor- und Nachteile der Implementierung aus Aufgabe 4.
- Überlegen Sie sich zusätzlich, wie sich die Implementierung verbessern lässt in dem die Vertex- / Indexdaten unabhängig von der OpenGL Implementierung in einer *Mesh*-Klasse gespeichert und mittels einer separaten *Render* Klasse visualisiert werden, so dass keine Vererbung mehr für neue Objekte benötigt wird.
- Implementieren Sie die von Ihnen gemachten Überlegungen für die Insel und den Fluss.