



CG1 WS 23/24 - Exercise 0: Set up

Technische Universität Berlin - Computer Graphics

Date 19. October 2023 **Deadline** 25. October 2023

Prof. Dr. Marc Alexa, Tobias Djuren, Hendrik Meyer, Markus Worchel

In this very first exercise you will learn how to setup npm, basic features of TypeScript, how to write a function in it and the structure of the exercises. The exercises will be written in TypeScript using the 3D visualization library ThreeJS. That means we build a web application. Since CG1 is not about web development we try to abstract those parts with the provided skeleton as much as possible.

Web development is based on the server client model. The web server serves an HTML file, that is rendered using a browser as client. The rendered page can be altered programmatically using JavaScript. TypeScript is a super set of JavaScript and extends it by adding types. Since every modern browser only supports JavaScript, TypeScript must be transpiled to JavaScript first. This results in a pipeline that transpiles TypeScript to JavaScript, then serves that as web server for a browser to interpret the code. For this pipeline to work you need to install node.js, npm (<https://nodejs.org/>) and a browser supporting WebGL (Firefox, Chrome, Edge, etc.). For editing TypeScript a simple text editor (e.g. Visual Studio Code or Notepad++), or IDEs (e.g. WebStorm, Eclipse or Netbeans) can be used.

Setup the skeleton (0 points)

The most used packet manager for JavaScript/TypeScript is npm. We use it to download required packages and run scripts. These scripts compile TypeScript and run the web server.

We provide a code skeleton in the `cg1_exercise_0.zip` file. Run `npm install` in the root directory of the project to install all required dependencies. The command `npm run dev` starts a webserver on port 5173 combined with a watcher that recompiles every changed (TypeScript) file within this folder. Open the address `127.0.0.1:5173` in your browser. This page will be refreshed whenever you change a file. The main entry point is `src/main.ts`, additional global utilities can be found in `src/lib/`. Exercise depended code can be found in `src/helper.ts`. In future assignments, we will provide additional files in `src/` that can be used. You can base your solutions for the upcoming exercises on your solution to this assignment. *Important:* Submissions are a `.zip` file of your `src` folder.

Basic Functionality (1 point)

The code skeleton provides a basic view on geometric objects as well as a simple graphical user interface (using the `dat.gui` library) with incomplete functionality. This task helps to get a feeling for the behavior and I/O operations in a WebGL project.

- First, carefully read the file `main.ts` including all comments and get a feeling for the program structure and control flow. (0 points)
- Change the application layout to use the whole space of the browser. Add a keyboard input, so that a random background color is assigned to the scene when the spacebar is pressed. (*Hint:* For the second task, have a look at `document.addEventListener`) (0.25 points)
- Then update the interface and add the additional functionality:
 - The **Scale** slider should adjust the object's scale. (*Hint:* Use `helper.scale`. We disabled the ThreeJS convenience functions for setting properties of `THREE.Object3D`, e.g. `scale.set`) (0.25 points)
 - The **Disco** checkbox should toggle between two different colors of the shown object. (*Hint:* Look at `mesh.material`, which is an instance of the class `THREE.MeshPhongMaterial`) (0.5 points)

Requirements

- Exercises must be completed individually. Plagiarism will lead to exclusion from the course.
- Submit a `.zip` file of the `src` folder of your solution through ISIS by **25. October 2023, 23:59**.
- *Naming convention*: `{firstname}_{lastname}_cg1_ex{#}.zip` (for example: `jane_doe_cg1_ex0.zip`).
- You only hand in your `src` folder, make sure your code works with the rest of the provided skeleton.