CG1 WS 20/21 - Exercise 0: Set up

Technische Universität Berlin - Computer Graphics
Date 05. November 2020 **Deadline** 11. November 2020
Prof. Dr. Marc Alexa, Ugo Finnendahl, Max Kohlbrenner

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. 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 a 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, than 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, Safari etc). For editing Typescript a simple text editor (e.g. Visual Studio Code, Atom or Notepad++), or IDEs (e.g. WebStorm, Eclipse or Netbeans) can be used.

Setup the skeleton (0 points)

The most used packet manager of 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 `ex0-basics.zip` file. Run `npm install` in the root directory of the project to install all required dependencies. The command `npm run start` starts a webserver on port 8080 combined with a watcher that recompiles every changed (Typescript) file within this folder. It will also refresh the browser on every change. 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 (1.5 points)

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 change the application layout to use the whole space of the browser and add a keyboard input that allows to rotate the object around the x-axis using the left and right arrow keys. (*0.5 points*)
- Then update the interface and add the additional functionality: (*1 point*)
  - The *App name* section in the interface should define the document title.
  - The checkbox should toggle between two different colors of the shown object. (*Hint:* Look at `THREE.MeshPhongMaterial`)
  - It should be possible to use the slider to adjust the objects scale. (*Hint:* You can set properties of `THREE.Object3D` using `.set` e.g. `position.set`)

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 **11. November 2020, 23:59**.
- *Naming convention*: {firstname}_{lastname}_cg1_ex{#}.zip (for example: jane_doe_cg1_ex0.zip).
- The zip must be standalone and include the whole code and all libraries, including html and js files.