LMU

LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

INSTITUT FÜR INFORMATIK
MEDIENINFORMATIK
PROF. DR. ANDREAS BUTZ, CHANGKUN OU, DENNIS DIETZ
COMPUTERGRAFIK 1, SOMMERSEMESTER 2022

Assignment 3: Camera

Submission Period: 07.06.2022 12:00:00 - 21.06.2022 12:00:00 (Central European Summer Time)

General Information

- This is one of the graded assignments. In this assignment, one can collect a maximum of 20 points. **A submission that cannot be compiled will not be graded.**
- There are two different aids for that may be helpful for accomplishing the assignment. 1) a certain visual effect that is described in detail in the assignment, and can be verified visually using `npm start`; or 2) provided tests that can be executed using command `npm test`.
- Note that **passing all provided tests, or achieving a similar visual effect does not represent one can collect all points**. The evaluation of submissions will also run additional tests that are not provided to check the general robustness and soundness of a submission, such as possible edge cases. We recommend carefully considering an implementation if desire more points.
- **Dependent tasks assume previous results to be correct**. This means a subsequent implementation that depends on a previous incorrect implementation is also considered incorrect.
- It is prohibited to exchange solutions for the assignments with other students during the examination period. You must work on the assignments alone and independently and submit your own solution. If we discover any fraud or plagiarism in the submission, **both parties will be excluded** from the exam.
- If you found the task description ambiguity or potential mistakes in the provided code skeleton, please contact cg1ss22@medien.ifi.lmu.de or ask in the tutorial class for further clarification.

Erklärung über die eigenständige Bearbeitung

Ich erkläre hiermit, dass ich die vorliegende Arbeit vollständig selbstständig angefertigt habe. Quellen und Hilfsmittel über den Rahmen der Vorlesungen/Übungen hinaus sind als solche markiert und angegeben. Ich bin mir darüber im Klaren, dass Verstöße durch Plagiate oder Zusammenarbeit mit Dritten zum Ausschluss von der Veranstaltung führen.

Task: The Camera Viewing Pipeline (20 Points, Medium)

In this task, we are going to implement the camera viewing pipeline for the bunny that we loaded from last assignment.

As we discussed in the class, the viewing transformation consists of a series of transformation matrices: $\mathbf{T}_{\text{model}}$ as the model transformation matrix, $\mathbf{T}_{\text{view}}$ as the view transformation matrix, $\mathbf{T}_{\text{proj}}$ can be either $\mathbf{T}_{\text{persp}}$ or $\mathbf{T}_{\text{ortho}}$ as projection transformation matrices, $\mathbf{T}_{\text{viewport}}$ as the viewport transformation matrix.

Look for `// TODO:` comments in the `src/camera/camera.ts` and `src/geometry/object.ts` files. Note that it is not allowed to introduce any new dependencies or to use APIs from `three.js`. The implementation is graded based on the following requirements:

- Implement scaling (2p), translation (2p), and rotation (2p) transformations.
- Implement model matrix (1p)
- Implement view transformation (4p)
- Implement perspective projection transformation (3p)
- Implement orthographic projection transformation (3p)
- Implement viewport transformation (3p)

There are some addtional hints:

- You can run the project by 1) installing all dependencies using `npm i` then 2) start and execute the project using `npm start`. Your browser will open a tab automatically. 3) A few additional unit tests are provided and can be used partially for testing your implementation using `npm test`.
- We recommend implementing the tasks along the pipeline from left to right. Making sure earlier steps work properly is the key to proceeding to the next successful stage.
- An overview of the implemented pipeline is presented in Figure 1, or interactive demo can be found here: `https://www.medien.ifi.lmu.de/lehre/ss22/cg1/demos/camera/`

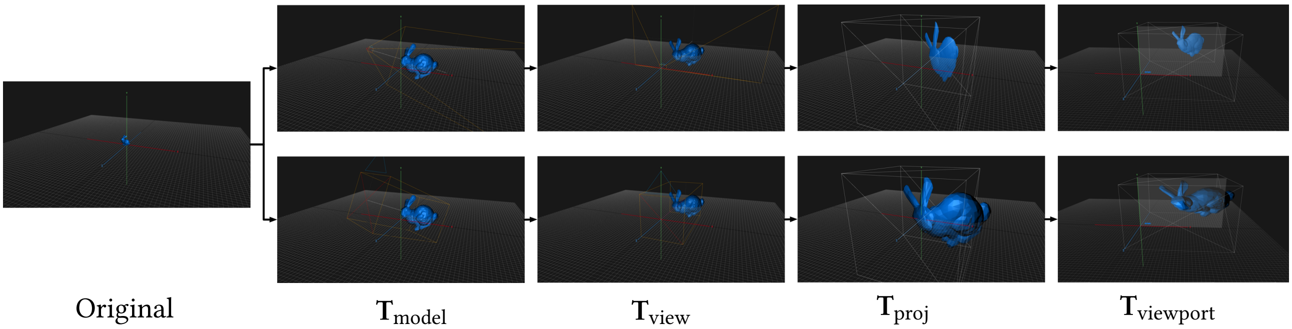
Original
$\mathbf{T}_{\text{model}}$
$\mathbf{T}_{\text{view}}$
$\mathbf{T}_{\text{proj}}$
$\mathbf{T}_{\text{viewport}}$

Figure 1: The camera viewing pipeline when all transformations are implemented. The upper branch uses a perspective camera, the lower branch uses an orthographic camera.

Submission Instructions

Please use the provided submission template and follow the submission instruction below to submit your solution to Uni2Work.

- Delete the two folders: **`node_modules`**, and **`build`**.
- Rename your folder to `cg1-assignment3-<your matriculation number>`, and compress everything as a single `.zip` file. For example, if your matriculation number is 12345678, then the zip-file's filename should be `cg1-assignment3-12345678.zip`.

  ✓ `cg1-assignment3-12345678.zip`
  ✗ `cg1-assignment3-<12345678>.zip`

  Your folder structure should be exactly like this (except the matriculation number):

```
cg1-assignment3-12345678/
  ├── .eslintignore
  ├── .eslintrc.json
  ├── .prettierrc.js
  ├── .vscode
  │   └── settings.json
  ├── README.pdf
  ├── assets
  │   └── bunny.obj
  ├── jest.config.js
  ├── package-lock.json
  ├── package.json
  ├── src
  │   ├── camera
  │   │   └── camera.ts
  │   ├── geometry
  │   │   ├── mesh.ts
  │   │   └── object.ts
  │   ├── main.ts
  │   ├── math
  │   │   ├── mat4.ts
  │   │   ├── quaternion.ts
  │   │   ├── utils.ts
  │   │   └── vec4.ts
  │   └── renderer.ts
  ├── test
  │   ├── camera.test.ts
  │   └── utils.ts
  ├── tsconfig.json
  └── webpack.config.js
```