BBM 414 - Computer Graphics Laboratory

HACETTEPE UNIVERSITY
Department of Computer Engineering

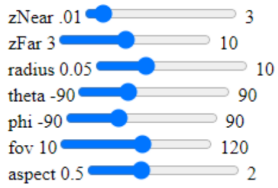
Programming Assignment 4
Fall 2023-2024
Assoc. Prof. Dr. Ufuk Çelikcan

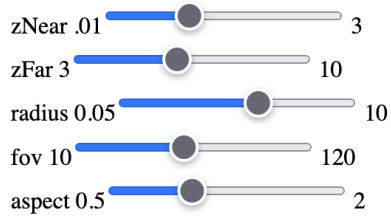
Due Date: 23:59 pm on Friday, December 15th, 2023

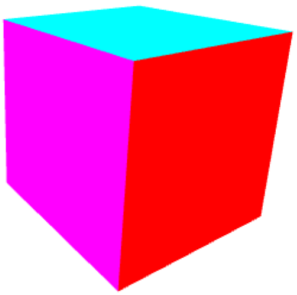
WebGL2 Model Drawing and Viewing

Part 1

In this part, you will modify the given project by downloading from Piazza resources and obtain the shape of form given in Figure 1.







(a) Input

(b) Result

Figure 1: Part 1.

- Download the source code perspective2 example from Piazza.
- Change the theta and phi with respect to the mouse movements (hint: pointer lock api).
- You will use 'p' to activate and deactivate the pointer lock api.

Part 2

In this part, you will get familiar with 3 dimensional shapes with first person camera viewing. You will load two objects and create a controllable camera on 3d scene. The steps are explained in the following statements:

1. You will create a scene by placing a gray "cat" on a green "terrain" provided on resources of Piazza (see Figure 2).
2. Read the cat and terrain model files in your code and place it on the given figure 2.

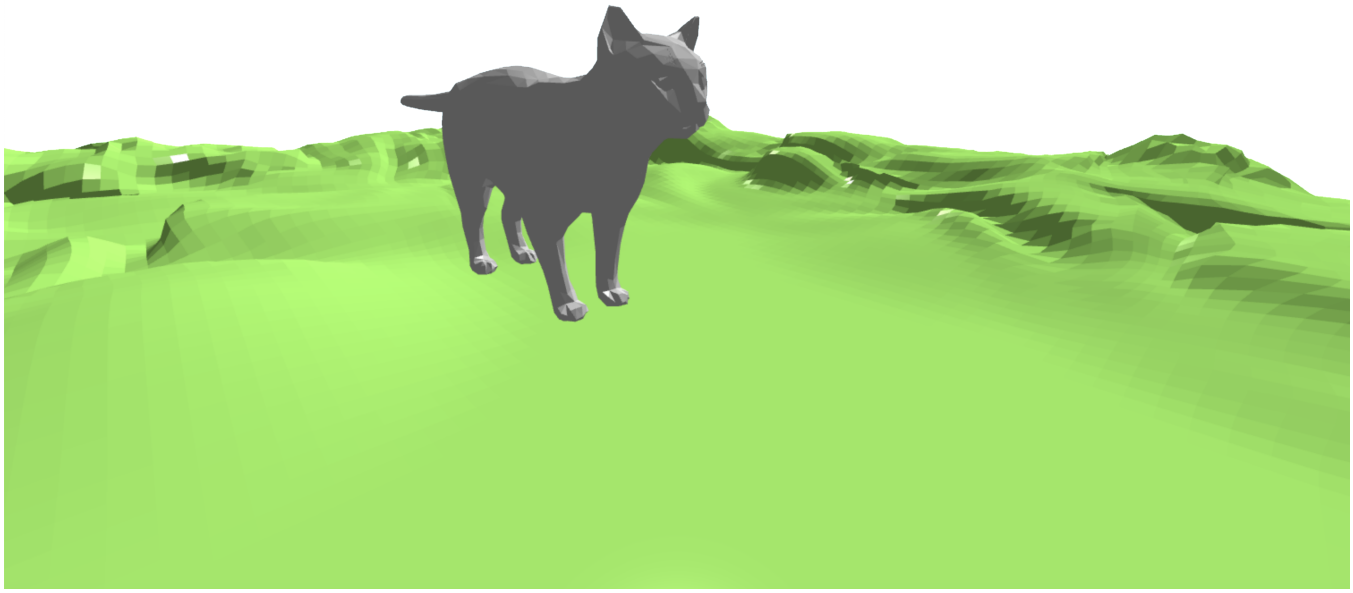


Figure 2: Gray cat on the green terrain

3. You do not need to implement any lighting (so the scene does not need to look as polished as in the figure). Just make sure that the user can distinctly see the cat placed over the terrain and they are in different colors.
4. Place your camera in the scene so that it sees the scene with a similar angle and distance as depicted in the figure above.
5. You will create a controllable camera.
   (a) Implement the keyboard input so that the user can navigate the camera:
      i. "Up Arrow" button: camera moves forward (positive way) along the global z-axis direction
      ii. Down Arrow" button: camera moves backward (negative way) along the global z-axis direction
      iii. "Right Arrow" button: camera moves to the right (positive way) along the global x-axis direction
      iv. "Left Arrow" button: camera moves to the left (negative way) along the global x-axis direction
      v. "Page Up" button: camera moves to the upward (positive way) along the global y-axis direction
      vi. "Page Down" button: camera moves to the left (negative way) along the global x-axis direction
   (b) Implement the mouse input so that the user can rotate the camera anyway s/he likes.
      i. Vertical movement of the mouse rotates the camera around its sideways axis.
      ii. Horizontal movement of the mouse rotates the camera around its y-axis.
      iii. Also note that mouse cursor should be invisible inside the drawing window.
   (c) The camera should see the scene using Perspective (fovy, aspect, zNear, zFar) function
   (d) The camera should adapt to resizing of the drawing window, so if the window is made smaller, the cat and the terrain should become smaller without getting distorted (and vice-versa).

The Implementation Details

1. Implement your homework using **WebGL2**. All programming assignments must use the shader-based functionality of **WebGL2**: at least one vertex shader and one fragment shader.
2. The assignment must be original work. Turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else. **Detection of such plagiarism**

in a submission will automatically void the submission and establish grounds to trigger an official disciplinary investigation. General discussion of the assignment among peers is allowed, but do not share answers, algorithms or source codes. **Also using other resources (example source code, book, webpage etc.) as a code and javascript libraries (except jquery, Angel's book) are not allowed.**

3. Do not write the scripts into the html file. Reference your scripts in html.

4. You should use Netbeans or Webstorm as IDE for your projects.

Hint: Search for pointer lock api to implement mouse input, and loading mesh in webgl.

The Report

You will write a report on latex for this assignment. You will explain the code parts and algorithm for part 1 and part 2.

What to Hand In

You should submit entire Netbeans or Webstorm project directory including javascript files and html file in a zip file extracted from IDE. Submission file structure is as given in below:

- b<studentNumber>.zip

  |–Experiment3_2023
  |——Part 1**(The whole Netbeans or Webstorm project)**
  |——Part 2**(The whole Netbeans or Webstorm project)**
  |–report.pdf

Archieve this folder as **b<studentNumber>.zip** and send via Piazza Private Post.

Grading

The assignment will be graded out of 100:

- PART 1 - CODE:0 (no implementation)
  20 (correct solution).

- PART 2 - CODE: 0 (no implementation)
  15 (terrain loading)
  15 (cat loading)
  15 (keyboard settings for camera)
  15 (mouse inputs for camera)

- REPORT: 20

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

References

[1] `https://github.com/esangel/WebGL`