Homework 3

CSU0021: Computer Graphics

This homework extends the last homework (from 2D to 3D) and covers 3D rendering, illumination, 3D hierarchical transformation, and user interaction.

You will implement a WebGL program to render a scene that contains 1 ground, one robot, at least one object, and one point light.

Scene:

- 1 point light: you should set a point light and draw a small cube or sphere at its location. (let us be able to see where it is)
- 1 robot: The robot consists of multiple 3D primitives (it is ok if all primitives are the same 3D geometric shape), and they are connected by multiple joints (at least 3). The robot should be a nice-look robot (such as a human-like robot or an excavator-like car).
- 1 object: The object consists of at least three 3D primitives (it is ok if all primitives are the same 3D geometric shape) and two joints that connect the three primitives.
- 1 ground (e.g., a big flat cuboid): Initially, you should put all the robot and object(s) on this ground correctly.

Illumination:

- You should implement a correct and nice-looking local illumination (ambient+difusse+specular and phong shading) for everything in your scene (the ground, robot, and object)

User Interaction:

- In order to observe the scene from different view angles and to test the correctness of the implementation of the illumination, we should allow users to (1) rotate everything in the environment (2) except for the light source (change the relative locations between the light source and objects) by mouse.
- Zoom in and out: Users can zoom in and out by either the mouse scroll wheel or a slider.
- Control your robot:
  - You should allow users to move the robot on the ground.
  - You should also create interfaces for users to rotate every joint individually.
  - You should allow users to grab and move the object. (1) It is not necessary to implement the real ”grab” behavior. You can just detect whether the ”hand” of the robot touches the object to grab. (2) However, detecting whether the ”hand” of the robot and the object touch each other is not easy either. You can choose to detect whether a point of the object is within a distance of a point of the robot’s hand. If so, the object is in the ”grabbable” status. (3) When the object is in the grabbable status, you should either

change the object or the robot's color to give users a hint. (4) Allow users to press keys or mouse buttons to grab and release the object. (4) You do not have to implement the "gravity". You can just let the object float in the air when it releases the object.

- Control the object: You should also create interfaces for users to rotate every joint individually. Users can rotate the joints no matter whether the object is moving by the robot or not.

Submission:

- You have to submit your program to moodle before the deadline. Otherwise, late submission penalty will be applied.
- You have to put all files (index.html, js) in a folder, zip the folder, rename the zip file to your student ID (e.g., 407470888s.zip), and submit this zip file to moodle. Ensure that TA can unzip your zip file and drag index.html to the browser to run without any extra work. If you do not follow this rule, your homework will be penalized.
- **You have to schedule time with TA to demonstrate your homework (you will not receive any points if you don't):**
  - Please book a 5 minutes time slot here before moodle submission deadline: `https://tinyurl.com/44nvpmzv` (Please check and sign up this at this form before May 1st)
  - You are welcome to bring your laptop for this demonstration. **If you will not bring your laptop, make a note when you book the time slot.**
  - make sure you arrive on time
  - TA office: Room 109 Applied Science Building.
  - TA email: 40747022s@gapps.ntnu.edu.tw
  - **If you submit the homework late, you still have to email TA and book a time for demonstration again. Otherwise, you will not receive any points.**