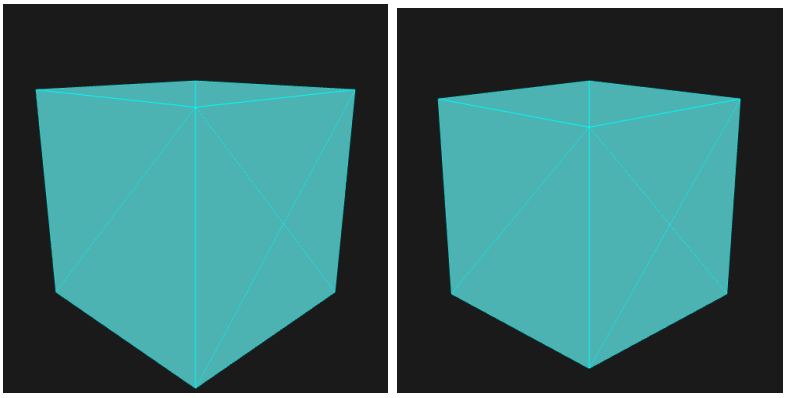
Computer Graphics Assignment1

In this assignment, our aim is to get insight into the components of 3D viewing pipeline such as various transformations in different coordinate systems. Your goal is to create a program for the interactive display and manipulation of a 3D robot.

1: Projection (25%)



(a) Perspective (b) Orthographic

Implement the orthographic and perspective projection functions in `01_projection.js` to return the correct matrices. You can view the results by opening `01_Projection.html` in a browser, and experiment with different field-of-view values, and near/far planes with the sliders provided.

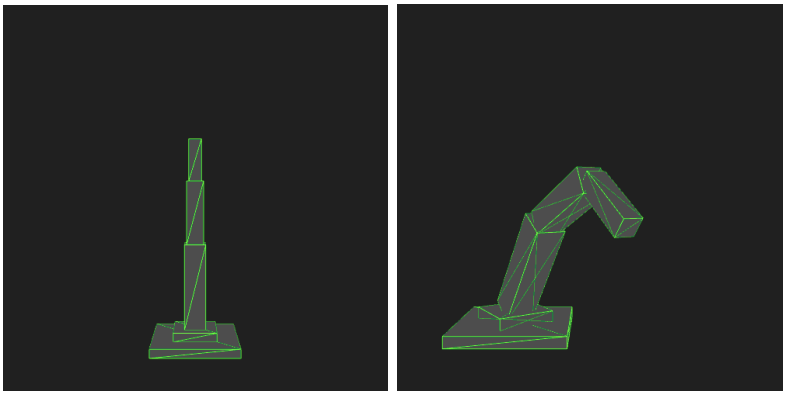
NOTE: To open with Google Chrome, add the command-line flag: `-allow-file-access-from-files` to ensure the 3D cube model file can be loaded.

2: Transformations (25%)

Implement the transformation functions in `02_transformations.js` so that they return the correct translation, rotation, or scaling matrices. You can view the re-

sults by opening `02_Transformations.html`, and experiment with different transformation parameters using the provided sliders.

3: Hierarchical Transformations (25%)



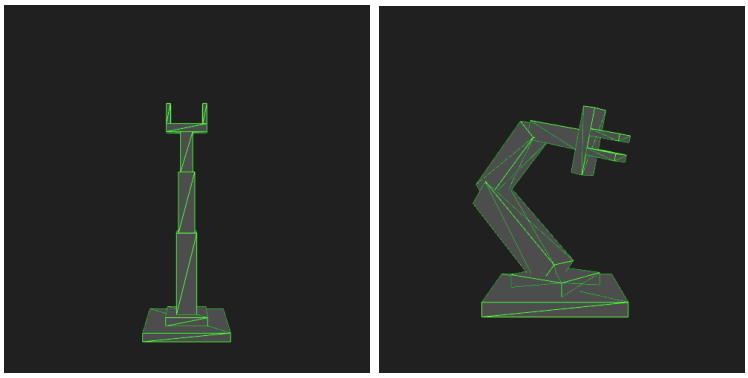
(c) Default (d) Transformed

Here, we will render a robot arm, which is composed of a hierarchy of `RobotSegment` objects. Each segment is composed of:

- A cuboid shape to render, with size and centre-coordinates defined in the local coordinate frame.
- A local transformation matrix (which may be altered by various sliders). This is returned via `getLocalTransform()`.
- A parent segment, representing the robot part the current one is attached to in the chain. When the parent moves, the child segment should move with it. If the segment is the base of the robot, its parent is `null`.
- A transformation matrix to define the segment's local coordinate frame in relation to its parent's coordinate frame (`localToParentMatrix`). This defines what the local coordinate system before any local transformations occur.

Fill in the missing code for the `getLocalToGlobalMatrix()` function of `RobotSegment` in `03_robot.js` to return a matrix to convert from the part's local coordinate frame to the global coordinate frame by traversing the segment hierarchy and combining their local, local-to-parent, and parent's transformations. When finished, the resulting model should look like the image above, and be movable via the sliders in `03_Robot.html`.

4: Adding Components (25%)



(e) Default (f) Transformed

Fill in the `addGripper(robotSegments)` function in `03_robot.js` to give the robot arm a gripper attachment. The segment/s creating this should be appended to the robotSegments given array. Your gripper should resemble the image above, and its segments should be positioned and scaled to avoid clipping into each other.

The gripper's segments should have local transformations such that it can be opened and closed via the slider in `03_Robot.html`. You may wish to use either translation or rotation matrices for this.

5: Submitting your assignment:

You can submit your assignment using the `submit` command on a DICE machine:

```
submit cg 1 ./folderWithYourAnswerFiles
```