Semester Work

1. Goal

In the context of this specific work, a simple website will be implemented that will host a 3D Graphics scene and will offer user interaction capabilities. For its implementation, the WebGL/ Javascript platform should be used and the final result should be fully functional in Mozilla Firefox which on the one hand offers native compatibility with the required WebGL capabilities and on the other hand is available in the (physical) laboratory. For the processing of matrices in the code, if library functions are desired, the glMatrix library (ver.2.0 or later) which was also used in the laboratory should be used.

2. Organization

2.1 The work will be carried out by teams of 2 people who will be declared when it is assigned. Single-member teams are acceptable but should be avoided due to the workload of 2 people.

2.2. The deadline for submitting the assignment is the end of the course examination day in the Spring Semester 2023-2024 examination.

2.3. Deliverables will only be submitted to e-class as a file in the format epwnymo1_am1_epwnymo2_am2.rar or .zip or .7z and must include:

- The four versions of the scene with all the necessary files to be directly executable (libraries, texture images, code files).
- A short 1-2 page documentation on any assumptions made, implementation difficulties, solutions adopted, etc. where the full names of the team members and the AMs will appear.

2.4. The final grade of the course is determined by 50% by the theory exam, 25% by the laboratory exam and 25% by the grade of the assignment. For the successful completion of the course, the final grade must be ÿ5 **and** the theory exam grade must be ÿ5 **and** the laboratory exam grade must be ÿ5. The grade of the assignment does not have to be ÿ5 (it can also be 0 or it may not be submitted, in which case it is counted as 0 in the total grade).

2.5 For the implementation of the work, **ONLY** the libraries that were used in the laboratory exercises and are also available via e-class can be utilized:

- webgl-debug.js for WebGL debugging messages,
- gl-matrix-min.js for creating and managing matrices and transformations.

3. Object

Step 1 (5%). Use a dark shade of gray for the background color and draw a 2-edge cube so that it is exactly in the center of the scene, in other words its center is at (0,0,0) and its sides are parallel to the x,y,z axes. Give each face of the cube a different shade of blue, uniform for each face. Use whatever policy you prefer for defining the triangles of the object. *Tip: For each face of the cube, use different vertices for the triangles that compose it, even if they are vertices with the same coordinates, between different faces. This will make it easier to give each face a different uniform color, giving the vertices that participate in each face the color you want.*

Step 2 (5%). Place the camera at position (9,9,9), facing the center of the scene and oriented upwards, parallel to the z-axis. Set the perspective to a viewing angle of 80 degrees, an aspect ratio of 1, a near visibility threshold of 0.01, and a far visibility threshold of 1000.

Step 3 (5%). A. Add a text box to enter the viewing angle (in degrees) viewAngle.

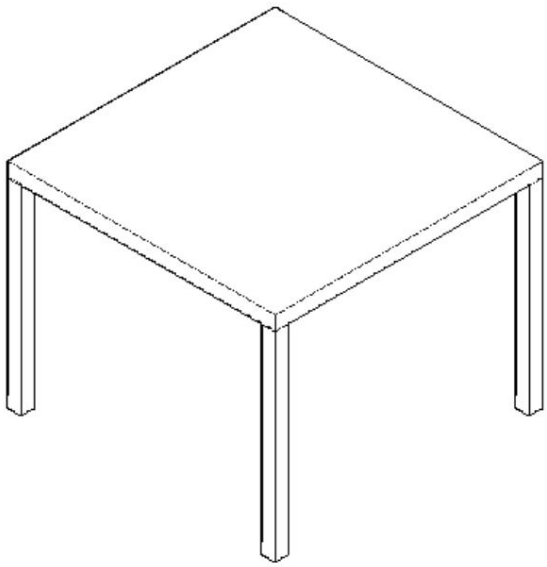
B. Add a text box to enter the orthogonal distance (common to all axes) of the camera from the origin of the viewDistance axes.
C. Add a group of radio buttons for predefined options for the camera position according to the table below (in practice, left-right refers to x'x, front-back to y'y, and bottom-top to z'z – the "ground" as in the lab will be considered to be the xy plane). The xy plane, in the current version of the scene, passes through the cube horizontally at mid-height
of:

Verbal	**Camera position coordinates corresponding to the literal (x,y,z)**
Left-Front-Top	(-viewDistance,viewDistance,viewDistance)
Left-Front-Bottom	(-viewDistance,viewDistance,-viewDistance)
Left-Back-Top	(-viewDistance,-viewDistance,viewDistance)
Left-Back-Bottom	(-viewDistance,-viewDistance,-viewDistance)
Right-Front-Top	(viewDistance,viewDistance,viewDistance)
Right-Front-Bottom	(viewDistance,viewDistance,-viewDistance)
Right-Back-Top	(viewDistance,-viewDistance,viewDistance)
Right-Back-Bottom	(viewDistance,-viewDistance,-viewDistance)

Step 4 (5%). Add a button to redraw taking into account the contents of the text boxes and radio buttons and always using one hundred times the viewDistance as the new far visibility threshold on each redraw (from this step until Step 7) for the perspective.

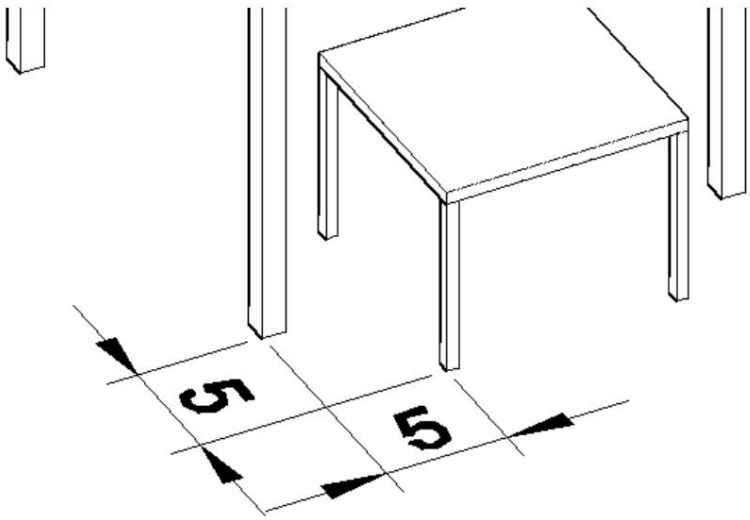
At this point save the first version of the scene (up to Step 4).

Step 5 (10%). Using scale and translate transformations on the cube and redrawing each version of it after the transformations (similar to the way we did it in the tetrahedron lab, without having to use a loop), draw a table like the one in the figure. Its top is dimensioned in x,y,z to



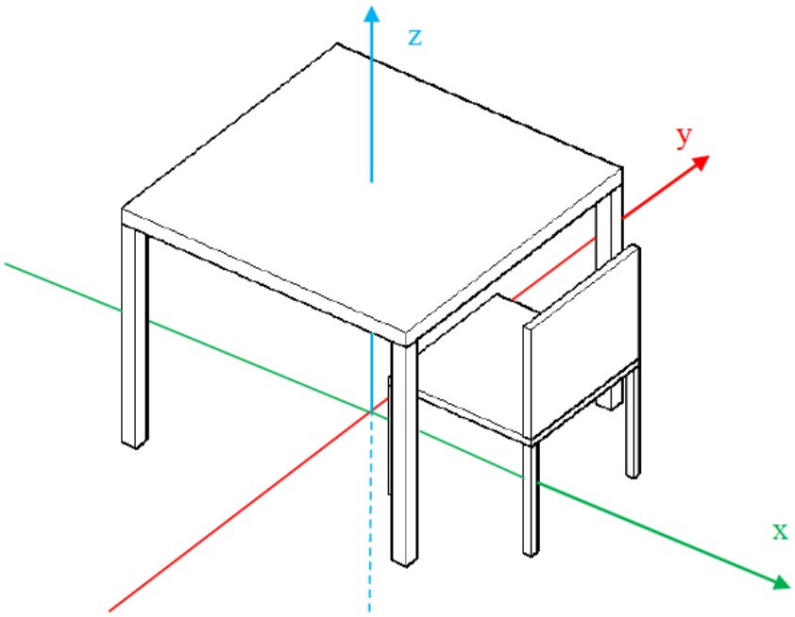
20ÿ20ÿ1 and each leg is 1ÿ1ÿ15 (all dimensions here and in the following are given in terms of x,y,z). Place it so that 0,0,0 is at the center of its base (i.e. the “floor” is the xy plane and each leg “steps” on a quadrant of xy – see the figure on the next page). Use uniform shades of blue for each part of the table (e.g. shades of blue for one leg, shades of red for the other, etc.).

Step 6 (5%). Using additional transformations, draw a stool that is exactly half the dimensions of the table (e.g. each leg is 0.5ÿ0.5ÿ7.5) and place it symmetrically, as shown in the figure below, between the legs of the table that rest on the xy and xy' quadrants (as shown in the figure on the next page). Use shades of green for the stool.



Step 7 (5%). Upgrade the stool to a chair by adding a back in shades of red with dimensions 0.5ÿ10ÿ7.5.

At this point save the second version of the scene (up to Step 7).



Βήμα 8 (10%). Προσθέστε μια animated σπειροειδή περιφορά της κάμερας παρόμοια με αυτήν που είδαμε στο εργαστήριο: η κάμερα περιστρέφεται γύρω από τη σκηνή "κοιτάζοντας" πάντα προς το κέντρο της. Προσθέστε δύο buttons για την εκκίνησή και την παύση του animation. Δεν είναι απαραίτητο να καθορίζει ο χρήστης τα βήματα περιστροφής και ύψους, μπορείτε να χρησιμοποιήσετε σταθερές τιμές αν θέλετε.

Βήμα 9 (10%). Χρησιμοποιήστε υφή αντί για χρώμα στα αντικείμενα: υφή ξύλου στο τραπέζι και υφάσματος στην καρέκλα.

<u>Στο σημείο αυτό αποθηκεύστε την τρίτη έκδοση της σκηνής (έως και το Βήμα 9).</u>

Βήμα 10 (10%). Προσθέστε ένα skybox, δηλαδή έναν μεγάλο κύβο 1000×1000×1000 με κέντρο στο (0,0,0) που περιβάλλει όλη τη σκηνή και έχει υφή ουρανού και ένα πάτωμα διαστάσεων 50×50 με τα ονόματα και τους ΑΜ των μελών της ομάδας *(χρησιμοποιήστε σαν υφή μία εικόνα που περιλαμβάνει αυτές τις πληροφορίες).*

Βήμα 11 (10%). Ελέγξτε το animation μέσω του ποντικιού, με τρόπο παρόμοιο με την αντίστοιχη άσκηση του εργαστηρίου (εκτός της ροδέλας): κίνηση δεξιά-αριστερά με πατημένο κουμπί ποντικιού ελέγχει την περιστροφή της κάμερας ενώ κίνηση πάνω-κάτω με πατημένο κουμπί ποντικιού ελέγχει το ύψος της κάμερας. (*<u>Ο έλεγχος με το ποντίκι θα πρέπει να λειτουργεί και με σταματημένο το animation.)</u>*

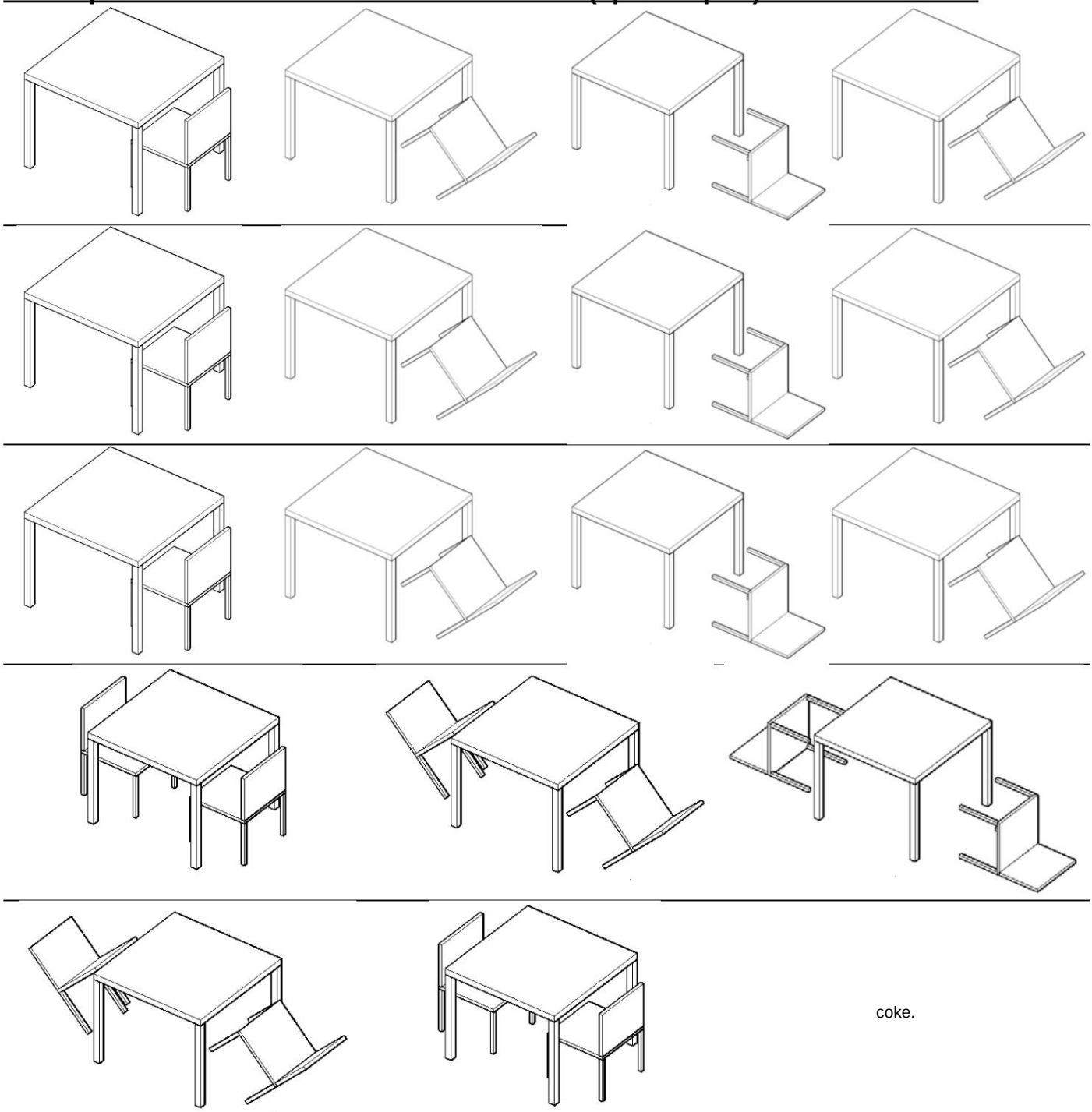
Βήμα 12 (10%). Κάντε την καρέκλα να ανατρέπεται προς τα πίσω και να ξανασηκώνεται ανάλογα με την κίνηση της ροδέλας του ποντικιού: κίνηση της ροδέλας προς τα επάνω ανεβάζει την καρέκλα προς τα επάνω μέχρι την όρθια θέση, κίνηση της ροδέλας προς τα κάτω τη ρίχνει στην πλάτη της. *<u>(Ο έλεγχος με το ποντίκι θα πρέπει να λειτουργεί και με σταματημένο το animation.)</u>*

Υπόδειξη: σε κάθε βήμα επανασχεδίασης θα πρέπει να <u>μετακινείτε</u> την καρέκλα ώστε η πίσω άκρες των πίσω ποδιών της να βρεθούν στον άξονα yy' (μετακίνηση μόνο ως προς x), να την <u>περιστρέφετε</u> τη γύρω από τον y και να τη <u>μετακινείτε</u> πίσω στη θέση της (αντίθετη μετακίνηση ως προς x). Η συνολική γωνία περιστροφής θα μεταβάλλεται μέσω

a (variable) step that will depend on how much the disc was rotated, making sure that it does not exceed the limits of the upright position (like the position in the first image) and the horizontal position (like the position in the third image).

Step 13 (10%). Easter Egg: If the user flips the chair all the way down *three times,* an additional chair appears opposite the original one, as soon as the latter returns to the upright position (see the sequence below). From this point on, the new chair also "obeys" the overturning and resetting via the washer normally, falling in the same way on its back and returning to the upright position. *(The mouse control and this response should work even when the animation is stopped.)*

At this point save the fourth version of the scene (up to Step 13).



coke.