



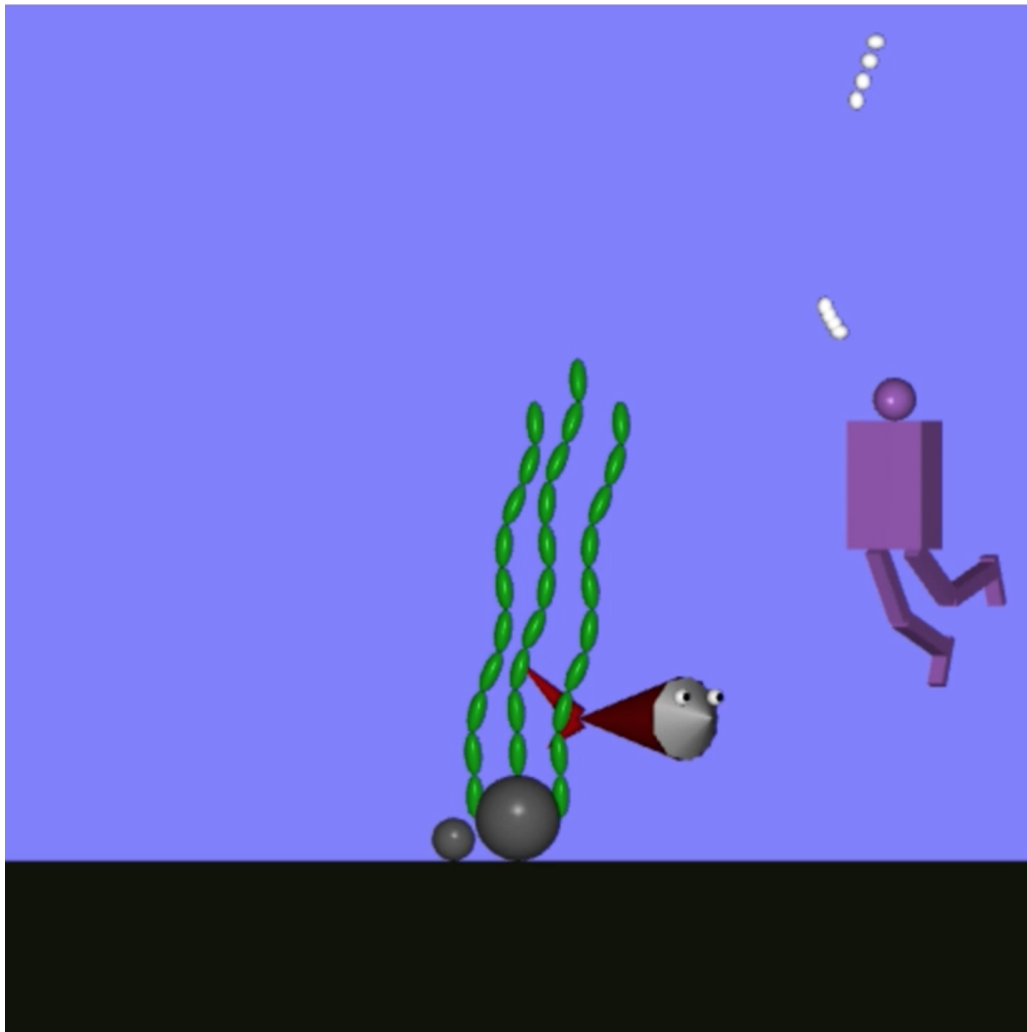
Assignment 1



Due Oct 14, 2022 11:59 PM

Computer Graphics is composed of many components to get us from object models to pixels on the screen. This assignment tests your understanding of transformations and hierarchical transformations that we have been learning in the beginning of the course.

Write a program in JavaScript/WebGL that draws the aquatic scene shown here. Template code is provided to handle many of the aspects necessary to get basic objects rendered and moving in a scene. Use the template code found here on the course webpage (an introduction to this base code is provided in Lab 3). For drawing objects, use only the procedures provided, e.g., `drawCube()`; for transforming the objects use the functions provide by the template code, e.g, `gRotate(theta,x,y,z);`. There is a video attached here, as well as a still frame below so you can see what is expected. Submit your code as a single zip file including everything needed. Please read the entire assignment and ensure you understand what should be submitted here. There are hints provided at the end.



Marking Scheme

Total Marks: 45

1. You have to use real-time to synchronize your animations. **(2 Marks)**.
2. Ground box, **2 (Marks)**.
3. Two rocks (spheres), **(4 Marks)**.
4. Seaweed modelling: each strand has 10 ellipses. **(4 Marks)**.
5. Seaweed animation **(4 Marks)**.
6. Seaweed positioning (3 strands) **(3 Marks)**.
7. Fish modelling: 2 eyes with pupils, 1 head, 1 body, 2 tail fins, **(6 Marks)**.
8. Fish animation: The fish must swim in a circle around the seaweed. It should always be aligned with the tangent of the circle. **(4 Marks)**.
9. Model a human character with no arms. **(4 Marks)**.
10. The character should move in the x and y world directions. **(2 Marks)**.
11. The legs of the character should kick (hips, knees rotate) as shown in the video. Note, the feet do not move. **(4 Marks)**.
12. You do not have to match the exact motion or dimensions of the objects shown in the examples. However, your scene should fit in the window (see 14) and be qualitatively and visually similar to the sample **(4 Marks)**.
13. Programming style (comments, functions) **(2 Marks)**.

14. The scene should be 512x512. **(-2 Marks if it is not).**
15. You have to submit a SINGLE file called <firstname-lastname>.zip that includes all the necessary files. **(-2 Marks if you do not).**
16. You *have to include a readme.txt* file that describes in full detail which of the required elements you have implemented successfully and which ones you have not. **(-4 Marks if you do not).**

Requirements/Policies

Collaboration

None. Do not collaborate this is an individual assignment.

Original Work

The assignment must be done from scratch. Apart from the template provided, you should not use code from any other source, including any previous offering of the class.

Zero Mark

If the code does not run, no objects appear in the window, or only the template code is running properly, no partial marks will be given.

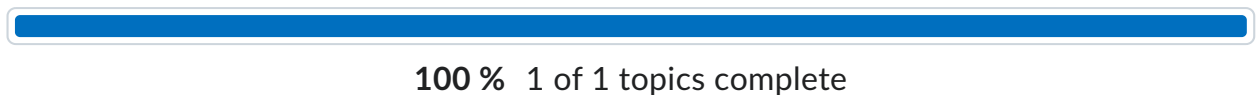
Clarifications

- Note that you must use the model stack correctly. That is, some students have distorted space and simply moved around in that space's units drawing and moving coordinates systems at the same time. Students will also sometimes apply the simple inverse for our elementary affine transformations instead of pushing and popping off the stack. Do not do this. It will lead to partial removal of marks for all of the modelling aspects of the assignment. The stack makes the assignment, and graphics in general, much easier!
- Make sure you rotate the body parts around the correct point, i.e. where they touch the parent body parts (at the joints), so bodies do not appear to break apart.
- The three seaweed strands are identical and move exactly the same way.
- The fish is rotating around a vertical axis and also moves up and down.
- Start your code in `main.js:render()` and feel free to write additional functions

Hints

- The render function is set as the window animation callback. There are details on how this works on MDN. This is the code:
 - `window.requestAnimationFrame(render);`

- This callback will also pass a time since the application started timestamp in real-time in milliseconds.
- There is already code for capturing the change in time per rendered frame, i.e., dt
- You will want to use timestamp and dt to sync your animations to real-time
- Use setColor(r,g,b) to set the desired colors.
- For the motions, you may want to use functions such as
 - Wave functions
 - $x(\text{timestamp}) = A \cdot \text{Math.cos}(w \cdot \text{timestamp} + h)$
 - Where timestamp is real-time, A is amplitude, Math.cos is cosine function, w is the angular frequency, h is the phase
 - Euler integration for movement
 - $x(t + dt) = x(t) + x'(t)dt$
 - Next position in time $x(t + dt)$ is just the current position $x(t)$ plus velocity $x'(t)$ scaled by the change in time dt.



AssignmentBaseCode
Zip Compressed File

A1_example
Video

Assignment 1
Assignment



Due Oct 14, 2022 11:59 PM

Assignment details and marking scheme can be found on the Assignment 1 course content page : Assignment 1.