**WebGL 3D Scene**

John Kucera

Prof. Amitava Karmaker

CMSC 405 6380 Computer Graphics (2218)

07 October 2021

**Included files**: WebGL3D.html, basic-object-models-IFS.js, gl-matrix-min.js, trackball-rotator.js, textures folder: burning+hot+lava.png, clear+sea+water.png, hedge+wall.png

**Texture sources** (also in References at end of document):

Lava: <https://www.the3rdsequence.com/texturedb/texture/259/burning+hot+lava/>  
 Water: <https://www.the3rdsequence.com/texturedb/texture/44/clear+sea+water/> Leaves: <https://www.the3rdsequence.com/texturedb/texture/70/hedge+wall+/>

**Purpose of program**: To create a unique 3D graphics scene using WebGL that contains 10 different objects, uses multiple light effects and different materials, uses multiple textures, is at least 640x480 pixels, includes widgets that can turn on/off certain animation components, and uses frame buffers to organize memory resources that are needed to render the scene.

***10 DIFFERENT OBJECTS***: Boat, Lifebuoy, Lilypad, Grass, Water, Sun, 2 Clouds, 2 Fish

**Descriptions of methods** (in WebGL3D.html):

* Shaders:
  + Vertex shader and Fragment shader. These are used to create graphics on the web, including shape transformations and rendering shape attributes like colors.
  + Two fragment shaders: One that provides color and one that provides texture. Shaders are switched depending on what user selects in the drop down menu.
* function *draw()*: Sets background of canvas to be GRAY. Calls on all lights and objects functions so they can be created and drawn.
* function *lights()*: Creates and positions two lights. Lights are only on if CHECKBOXES are checked.
  + Sunlight: YELLOW, shines from ABOVE. Always in position of the rotating sun.
  + Water light: BLUE, shines from BELOW where the water is.
* function *setLightPosition()*: Used to efficiently position lights.
* function *boat()*: Creates boat with flagpole and flag. Bobs up and down on the water, represented by ANIMATED TRANSLATION.
  + Boat: DARK YELLOW, EMISSIVE color, composed of CUBE objects.
  + Flagpole: RED, DIFFUSE color, composed of a CYLINDER object.
  + Flag: WHITE, DIFFUSE color, composed of a CUBE object.
* function *lifebuoy()*: Creates lifebuoy. PINK, EMISSIVE color, composed of a TORUS object. Bobs up and down on the water, represented by ANIMATED TRANSLATION.
* function *lilypad()*: Creates lilypad. LIGHT GREEN, EMISSIVE color, composed of a DISK object. Bobs up and down on the water, represented by ANIMATED TRANSLATION.
* function *grass()*: Creates 4 blades of grass. GREEN, EMISSIVE color, composed of CONE objects.
* function *fish()*: Creates 2 fish. One is ORANGE, EMISSIVE color, and the other is PURPLE, EMISSIVE color. They are composed of DISK objects. They swim in clockwise circles, represented by ANIMATED ROTATION.
* function *water()*: Creates block of water. BLUE, EMISSIVE color, composed of a CUBE object.
* function *sun()*: Creates sun. YELLOW, EMISSIVE color, composed of a SPHERE object. It rotates counterclockwise above the water, represented by ANIMATED ROTATION.
* function *clouds()*: Creates 2 clouds. GRAY, EMISSIVE color, composed of SPHERE objects. They rotate clockwise above the water, represented by ANIMTED ROTATION.
* functions *pushMatrix()* and *popMatrix()*: Pushes and restores modelview matrices onto matrix stack.
* function *createModel()*: Base function for creating a basic object.
* function loadTexture(): Loads image as texture. Textures to choose from: LAVA, WATER, LEAVES.
* function *createProgram()*: Creates program for use in WebGL context, initializes shaders
* function *initGL()*: Initializes WebGL context
* function *frame()*: Contains incremental object positions for each animated frame, including fish rotation, sun rotation, cloud rotation, and bob distance in y component.
* function *setAnimating()*: Responds to animation RADIO BUTTONS to turn animation on and off.
* function textureChoose(): Chooses which shader and texture to initialize based on what user has selected in HTML drop down menu
* function *init()*: Initializes HTML and canvas when page is loaded.
* HTML code: Contains a header for the page, instructions on how to rotate the scene, radio buttons to turn all animation on or off, checkboxes to turn lights on and off, and drop down menu to choose texture of scene. Canvas is 700x700.

**Program running in Visual Studio Code, with HTML Page in Mozilla Firefox**:

*(For Javascript/HTML projects, I prefer using Visual Studio Code over Netbeans)*

Text

Description automatically generated with medium confidence

Figure 1: Program running in Visual Studio Code with HTML Page in Mozilla Firefox

**Test Plan & Results**

|  |  |
| --- | --- |
| **Plan** | |
| After pressing Run and the HTML page is visible, first make sure that dimensions are 700x700, the widgets are in proper position, and that shapes are properly displayed in the canvas. Drag mouse across canvas to test trackball rotation functionality. Test widget functionality and shape animation. Canvas should update for each transformation. | |
| **Test # + Description** | **Result** |
| **(SUCCESS)**  **Test 1: Initial page**. Page title, header, and other HTML text are as intended. With the radio buttons, “Animation Off” is the default setting. The checkboxes are all initially filled in. No Texture has yet been selected. Canvas is 700x700. *All Shapes are drawn, colored, and positioned perfectly.* No animation is occurring yet. | Figure 2: Test 1 |
| **(SUCCESS)**  **Test 2: Trackball rotation functionality**. Dragged across the canvas and the scene rotated in the intended direction(s). | Figure 3: Test 2 |
| **(SUCCESS)**  **Test 3: Animation On/Off Radio button**. Pressed “Animation On” Radio button and all shapes started performing animated transformations. When I press “Animation Off”, ALL animation is stopped in its current position.  *(Animations: Rotating sun/clouds/fish, Translating boat/lifebuoy/lilypad)* | Figure 4: Test 3 |
| **(SUCCESS)**  **Test 4: Sunlight On/Off Checkbox**. When “Sunlight On / Off” checkbox is checked, yellow light is properly shined from the direction of the sun and onto the other objects. When I uncheck the checkbox, the sunlight disappears and no longer shines on the other objects.  (FIRST SCREENSHOT HAS SUNLIGHT ON, SECOND SCREENSHOT HAS SUNLIGHT OFF)  (WATCH FOR YELLOW LIGHT ON TOP OF CLOUDS) | Figure 5: Test 4 |
| **(SUCCESS)**  **Test 5: Water light On/Off Checkbox.** When “Waterlight On / Off” checkbox is checked, blue light is properly shined from the direction of the water and onto the other objects. When I uncheck the checkbox, the water light disappears and no longer shines on the other objects.  (FIRST SCREENSHOT HAS WATERLIGHT ON, SECOND SCREENSHOT HAS WATERLIGHT OFF)  (WATCH FOR THE BLUE LIGHT UNDER THE CLOUDS) | Figure 6: Test 5 |
| **(SUCCESS)**  **Test 6: Textures drop down menu (None).** When “None” is selected in the Textures menu, normal colors of shapes are shown. | Figure : Test 6 |
| **(SUCCESS)**  **Test 7: Textures drop down menu (Lava).** When “Lava” is selected in the Textures menu, all shapes have lava texture. | Figure : Test 7 |
| **(SUCCESS)**  **Test 8: Textures drop down menu (Water).** When “Water” is selected in the Textures menu, all shapes have water texture. | Figure : Test 8 |
| **(SUCCESS)**  **Test 9: Textures drop down menu (Leaves).** When “Leaves” is selected in the Textures menu, all shapes have leaves texture. | Figure : Test 9 |

**References**

*Burning hot lava - download royalty free texture*. the3rdSequence. (n.d.). Retrieved October 9, 2021, from <https://www.the3rdsequence.com/texturedb/texture/259/burning+hot+lava/>

*Clear sea water - download royalty free texture*. the3rdSequence. (n.d.). Retrieved October 9, 2021, from <https://www.the3rdsequence.com/texturedb/texture/44/clear+sea+water>

*Hedge wall - download royalty free texture*. the3rdSequence. (n.d.). Retrieved October 9, 2021, from <https://www.the3rdsequence.com/texturedb/texture/70/hedge+wall+/>