CMPE360

Project 6

Hello WebGL

Section 02

Erkan Sancak

44293566706

İçindekiler

Part I.. 3

I. Local Web Server 3

Part II 4

II.Image.. 4

WebGL Setup and Templates:................................ 4

JavaScript Files in Common Folder:........................ 5

Diverse Color Palette:.. 5

Code Explanation: 5

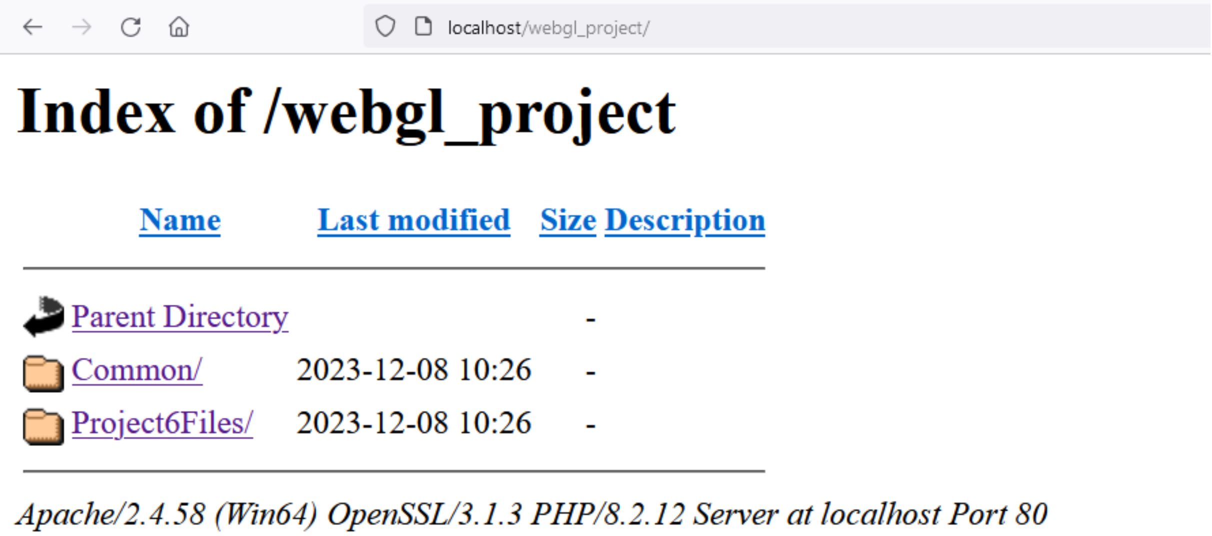
Challenges Faced:... 6

Image Without POINTS primitive........................... 6

Conclusion.. 6

Part I

I. Local Web Server

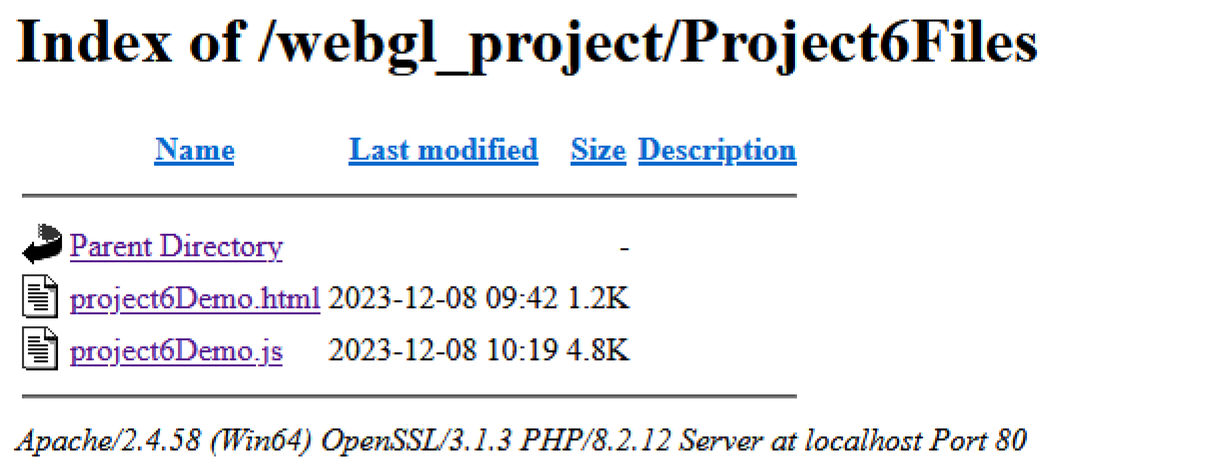


localhost/webgl_project/

Index of /webgl_project

Name	Last modified	Size	Description
Parent Directory		-	
Common/	2023-12-08 10:26	-	
Project6Files/	2023-12-08 10:26	-	

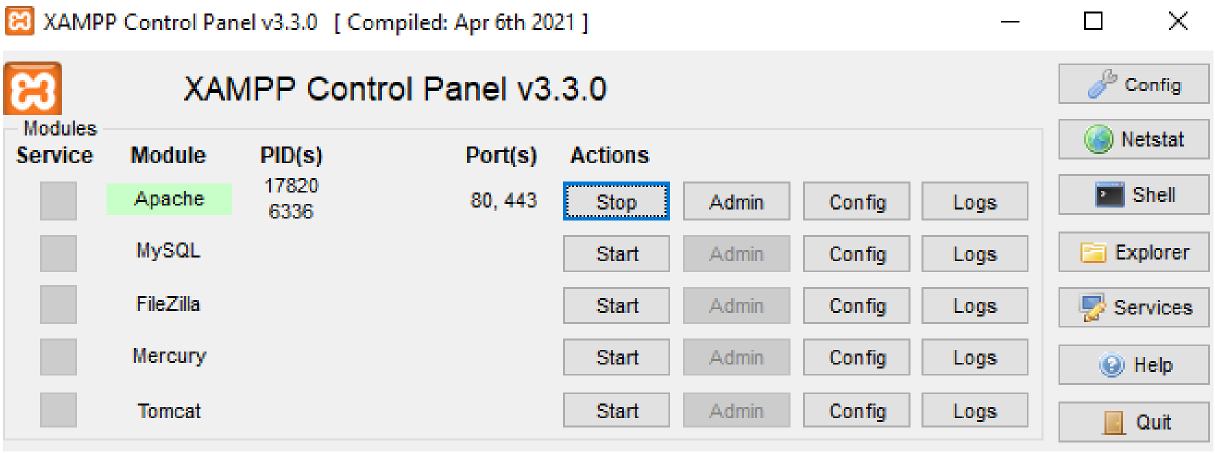
Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at localhost Port 80



Index of /webgl_project/Project6Files

Name	Last modified	Size	Description
Parent Directory		-	
project6Demo.html	2023-12-08 09:42	1.2K	
project6Demo.js	2023-12-08 10:19	4.8K	

Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at localhost Port 80



XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

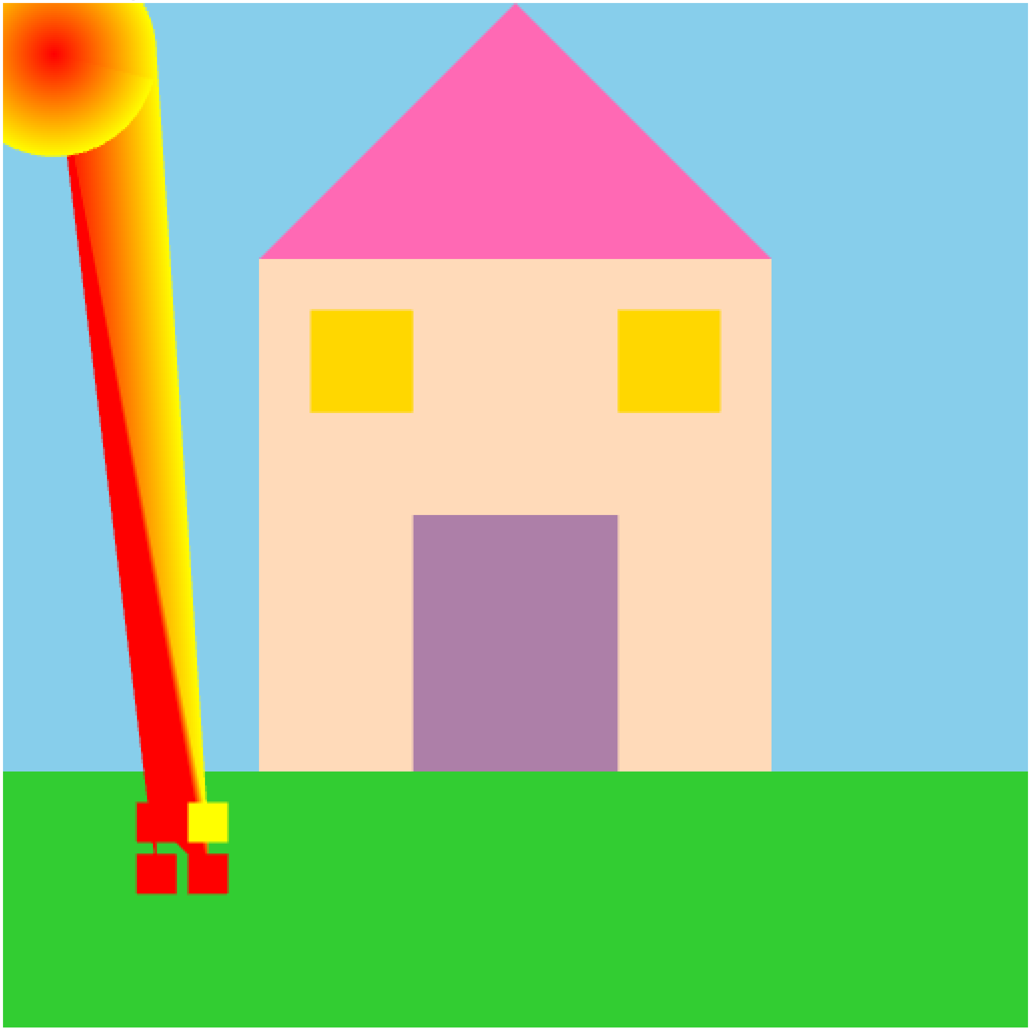
Modules

Service	Module	PID(s)	Port(s)	Actions			
□	Apache	17820 6336	80, 443	Stop	Admin	Config	Logs
□	MySQL			Start	Admin	Config	Logs
□	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
□	Tomcat			Start	Admin	Config	Logs

I used the XAMPP Apache service as the local web server. I was able to view my image by uploading my files to the local server.

Part II

II.Image



Utilized a combination of OpenGL primitives including TRIANGLE_STRIP, TRIANGLES, TRIANGLE_FAN, POINTS to introduce variety to the scene and satisfy the homework requirements.

WebGL Setup and Templates:

- Project commenced by setting up a WebGL environment, utilizing the provided HTML and JavaScript templates from the following links:
    - HTML Template
    - JS Template
- Used HTML template to construct the fundamental structure of the webpage. This served as a fundamental blueprint for the layout.
- Structured the JavaScript file using the provided JS template to create a organized code.

JavaScript Files in Common Folder:

- Utilized additional JavaScript files in a 'Common' folder:
  - utility.js for utility functions.
  - initShaders.js for shader initialization.
  - MVnew.js for matrix and vector operations.
  - flatten.js for flattening arrays.

Diverse Color Palette:

- Sky (background) represented by light blue.
- Grass(ground) with a green hue.
- House is peach, deep pink for the roof, lavender for the door, and gold for windows.
- Red for flowerpot and yellow for sun.

Code Explanation:

1. **Initialization (init function):**
   a. Sets up the WebGL environment, retrieves the canvas, and checks for WebGL 2.0 support.
   b. Initializes shaders and sets up the vertex and fragment programs.
2. **Vertex Data:**
   a. Defines arrays positions and colors representing vertex positions and corresponding colors for various elements in the scene.
3. **Circle Vertex Creation:**
   a. Dynamically adds vertices for a circle (representing the sun) to the positions and colors arrays.
4. **Buffer Creation:**
   a. Creates WebGL buffers (vertexBuffer and colorBuffer) and fills them with vertex data.
5. **Attribute Setup:**
   a. Sets up vertex attributes (vPosition and vColor) for position and color data.
6. **Rendering (render function):**
   a. Draws various elements using different WebGL primitives for example → gl.drawArrays(gl.TRIANGLE_STRIP, 0, 4) for drawing a background).

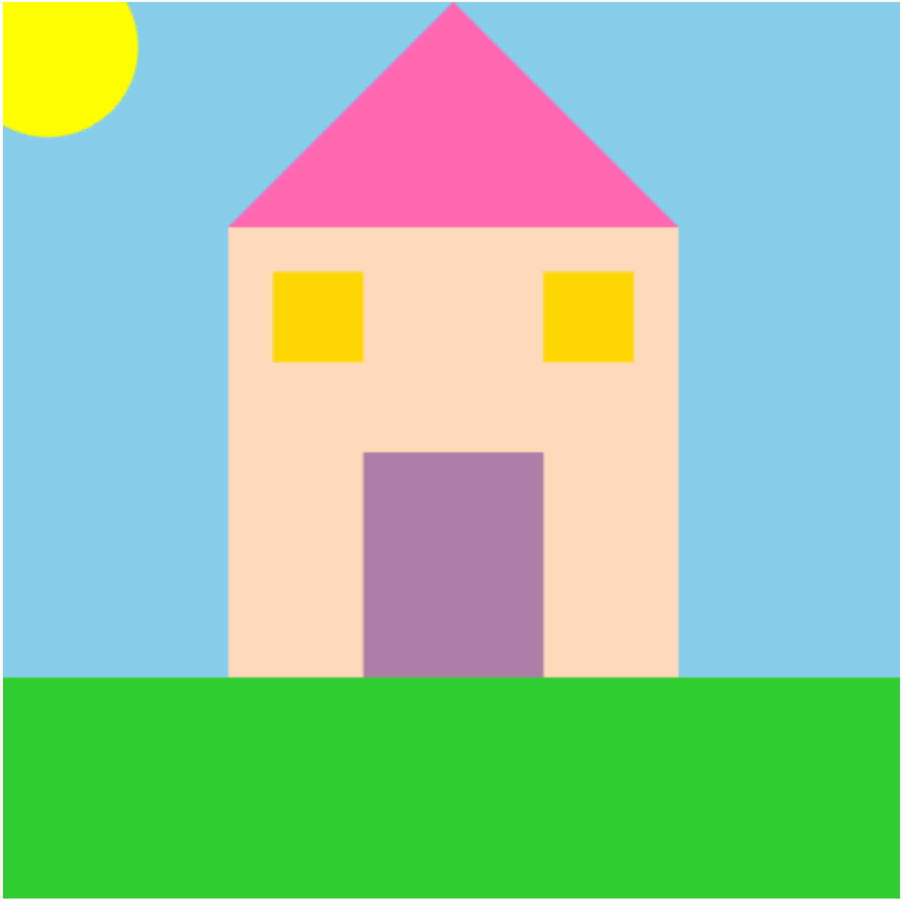
Challenges Faced:

1. **Issues with POINTS Primitive for the Sun:**
   Implementing the POINTS primitive for the flowerpot resulted in unexpected distortions, corrupting the intended circular shape for the sun which was made with TRIANGLE_FAN.

2. **Color Mixing in POINTS Primitive:**
   Colors of the flowerpot's POINTS and sun's TRIANGLE_FAN primitive were blending together, diminishing the clarity of individual colors.

3. **Issues with LINE Primitive Implementation:**
   Attempts to introduce LINE primitives resulted in unexpected distortions and disruptions to the intended composition of the image.

Image Without POINTS primitive



Conclusion

Despite the challenges encountered in the implementation of POINTS and LINE primitive types, the WebGL project has provided valuable insights into the complexities of real-time graphics programming. The pursuit of fulfilling the specified requirements in Step 3 yielded a partially completed result.