COMP 6909 Final Project-Team 16

A First-person Shooter Game Developed Using WebGL

Xiang Chen 201992538

Yue Xing 201466372

Memorial University of Newfoundland

A First-person Shooter Game Developed Using WebGL

1. **Introduction**

This project is a first-person shooter game developed using WebGL. The main functions of this game are as follows:

1. Navigate by mouse movement and keyboard keys, click the left mouse button to shoot the moving target.

2. Clicking the left mouse button will produce the sound effect of shooting.

3. Display of current score and time.

4. All objects in the scene have been added textures, such as the wall and the surface of the gun are posted with pictures.

5. The two light sources can be manipulated by keyboard keys and effective interfaces. Players can switch the color, position, and intensity of the light source separately.

6. The effect of fog is added. Players can control the intensity of fog in the scene through keyboard keys and effective interfaces.

7. Walls and obstacles (boxes) are added to the scene to increase the difficulty of the game. Players can control whether the box moves through keyboard keys and an effective interface.

8. The number of targets and moving speed at different levels are different.

2. User Manual

2.1 How to run the project

Please use sublime-text or other HTML editor to open the "Source Code-Final Project" folder, run the "Project-team16.html", all necessary js, CSS files should be located in the "js" folder, all audio effect mp3 files needed to load the game are located in the "music" folder.**(Please bring headphones to experience the sounds in the game.)**

Please note: This project needs to load texture images locally. When you use the Firefox browser, enter “about: config” in the address bar, then enter and select “security.fileuri.strict_origin_policy”, and set the corresponding parameter to “false”. **It is recommended to use the Firefox browser, the operation is relatively simple and can effectively load local images to display the project effect.**

When using the Google Chrome browser, due to security reasons, Chrome will not load local files by default.

Please get the URL of the Chrome Installation path:

E.g C:\Users\-your-user-name\AppData\Local\Google\Chrome\Application>

Launch the Google Chrome browser from the command line window with the additional argument‘-allow-file-access-from-files’:

E.g ‘path to the chrome installation\chrome.exe --allow-file-access-from-files’.

The temporary method you can use each time you are testing Copy the existing chrome launcher. Do as above and save it with a new name E.g chrome-testing

Alternatively, you can simply create a new launcher with the above and use it to start chrome.

2.2 Game operations and effects

The user can control the relative position of the gun through the mouse and W, A, S, D keyboard keys to complete the navigation. Point at the target and click the left mouse button to shoot. Different levels and different speeds of targets will appear on different walls. "Mark" shows the number of targets hit, "Time" shows the remaining time. The user can control the two corresponding lights to turn on or off through the keyboard keys "L" "R", and control the intensity of the lights through the left and right keys. Different light sources correspond to different colors, the left light source corresponds to red, and the right light source corresponds to green. By default, the red light is turned on. When the two lights are turned on at the same time, the scene light is displayed as yellow.

Besides, users can also use the up and down buttons to control the fog density in the scene. The up button corresponds to an increase, and the down button corresponds to a decrease. The obstacle-box in the scene can also move, and the keyboard button F controls whether the box moves. All the above functions can be controlled through the effective interface and keyboard keys.

3. Statement of source and reference

The main reference sources for this project are listed in "Sources and References". We mainly implemented all codes for functions such as light source control, object surface mapping, fog effect, gun movement, and obstacle boxes' movement. The main files involved include: “Project-team16.html”, “wall.js”, “box (1-5).js”, “gun.js”. The implementation codes

of the shader and all related attributes are defined in “Project-team16.html” and they are clearly annotated.

4. Novel Component

4.1 Sound loading

All sound files are stored in the “music” folder in mp3 format, corresponding to the sound of the shot and the prompt sound. We Defined a function called “callback” in the “sound.js” file to call each audio file in sequence to produce the corresponding game effect.

4.2 Fog effect

We are inspired by a basic equation: gl_FragColor = mix (originalColor, fogColor, fogAmount); Where fogAmount is a value from 0 to 1.The mix function mixes the originalColor and fogColor. When fogAmount is 0 mix returns originalColor. Then fogAmount is 1 mix returns fogColor. In between 0 and 1 we can get a percentage of both colors. In this project, an alternative equation for mix is applied: gl_FragColor = originalColor + (fogColor-originalColor) * fogAmount; u_fogintensity corresponds to fogAmount, and textureColor + vColor corresponds to originalColor. Finally, the user can control the density of the fog through the Up and Down. The effect of the fog is that the color of all objects in the scene becomes lighter.

4.3 Collision detection between objects

The effect of the bullet hitting the target is mainly achieved through collision detection. We created the framebuffer object “framebuffer”, and read the RGBA value of each point in

the framebuffer to determine whether the corresponding point is located on the target, thus implementing coordinate collision detection. The above is mainly implemented in the check function in target.js.

5. Screenshots of the application

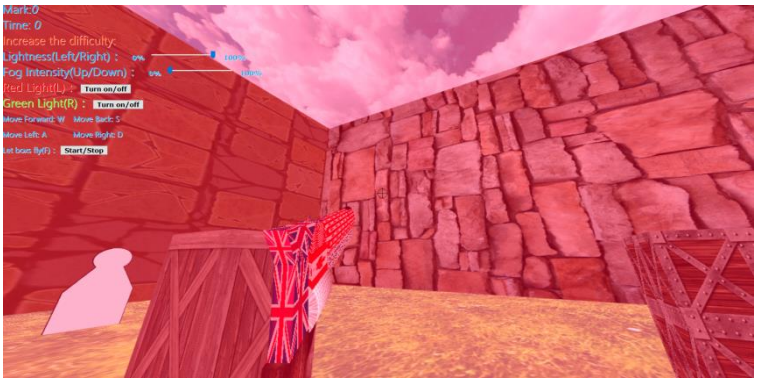

Figure 1: Only turn on the red light, the brightness is 100%

Figure2: Only turn on the green light, the brightness is 25%

Figure3:Turn off all lights, lightness is 0% (through the left and right key buttons)

Figure4:Click the keyboard button F or click let the box fly, the fog density is 75%

6. Source and Renference

[1] Original Project: https://github.com/vorshen/simpleFire

[2] Textbook's author:

https://www.cs.unm.edu/~angel/BOOK/INTERACTIVE_COMPUTER_GRAPHICS/SEVENTH_EDITION/CODE/06/shadedCube.html

[3] https://learnopengl.com/index.php?p=Lighting/Basic-Lighting (Learn how to affect the overall brightness of ambient light by setting the ambient light intensity coefficient.)

[4] WebGL Fog:https://webglfundamentals.org/webgl/lessons/webgl-fog.html

[5] Function loadTexture taken from:

https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Tutorial/Using_textures_in_WebGL