Cross The Road Game

Team Members:

1-Ahmed Nasser Emam Ibrahim (Section 1)
2-Aya Ragab Hussien Ali (Section 2)
3-Esraa Mohamed Salem (Section 1)
4-Mohamed Yehia Amin (Section 5)
5-Mahmoud Sameh Abd EL-Aal(Section 5)

ENG: Mai Ashraf
Instructor: DR Orphain

Abstract

Cross The Road Game which is programmed using the OpenGL library. The Cross Road game is a classic arcade-style game where the player controls a character attempting to cross a busy road. The game features various levels of difficulty, with increasing traffic and obstacles as the player progresses.

The documentation provides a detailed explanation of the game's features, including the game mechanics, game controls, scoring system, and graphics. The documentation also includes instructions for installing and running the game, as well as troubleshooting tips for common issues. Overall, this documentation serves as a comprehensive guide for anyone interested in playing or developing the Cross Road game using the OpenGL library.

Start Screen

This code defines a function named “startscreen” that draws a start screen for a game using the OpenGL graphics library in C++.

The “glClear” function is used to clear the color and depth buffers to prepare for drawing the new screen.

The “glColor3f” function sets the color of the text that will be displayed.

“glRasterPos3f” sets the position where the text will be displayed in 3D space.

“glutBitmapCharacter” is used to draw each character of the title and the messages on the screen. It takes two parameters: the font type and the character to be drawn. In this code, different font types are used for different messages.

Finally, “glutSwapBuffers” is called to swap the front and back buffers, displaying the newly drawn frame on the screen.

Overall, this function creates a start screen for the game that displays the title and instructions for starting or exiting the game. The user can start the game by pressing the space bar or exit the game by pressing the ESC key.

3D Space

this function responsible for rendering the game scene.

The glClear function is used to clear the color and depth buffers to prepare for drawing the new screen.

The glLoadIdentity function is used to reset the current model-view matrix to the identity matrix.

the gluLookAt function is sets up the camera or the eye position in the 3D space.

Calls the function (drawBall, car,coliision, attempts**) to actual run**

The load function to load the image (road(1), wall(2),background(3)**) and use the texture to show image 2D in the wall and road of the game**

glutSwapBuffers (): Swap the front and back buffers to display the rendered scene.

Draw Ball and Lightning

This code defines a function named “draw Ball” that draws a ball in a 3D space using the OpenGL graphics library in C++.

The function first enables lighting and sets the properties of a point light source, including its position, diffuse color, and specular color.

Then, it sets the material properties of the ball, including its ambient, diffuse, and specular colors, as well as its shininess.

The function then saves the current matrix state using “glPushMatrix”, translates the ball to its position in 3D space using “glTranslatef”, and draws a solid sphere with a radius of 3 units and 30 subdivisions in both the horizontal and vertical directions using “glutSolidSphere”.

Finally, the function disables lighting, pops the matrix stack to restore the previous state of the scene, and returns.

Overall, this function draws a ball with lighting and material properties in a 3D space at a specified position.

Draw Cars

The car () function is a part of a larger project that simulates the movement of cars on five different roads in 3D space. The function uses the OpenGL library to create 3D cubes that represent cars. Each cube is made up of six faces that are colored differently to give the appearance of a real car.

The movement of the cars on the different roads is controlled by a for loop and conditional statements. The cars on roads 1 and 3 move from left to right, while the cars on roads 0, 2, and 4 move from right to left.

The function considers the position of each car on the road and moves it according to the values of car1 and car2. These variables control the position of the cars on the X-axis and allow them to move along the road.

Overall, the car () function provides a visual representation of the movement of cars on different roads. It is a part of a larger project that simulates traffic flow and can be used to analyze and optimize traffic patterns.

Collision

This function checks for collisions between a ball and three different cars that move on roads 0, 1, and 3. The function is called continuously during the game to check for collisions and update the game state accordingly.

The function checks if the ball is between the values of -10 and -60 on the Z-axis, which indicates that the ball is moving from right to left. If this is the case, the function checks which car the ball is colliding with based on the position of the ball and the cars on the road.

If the ball is colliding with car1 or car2, the function checks if the ball is in front of the car or not. If the ball is in front of the car, the function returns the ball to its original position and decreases the number of attempts by one. If the number of attempts reaches three, the attempts () function is called to end the game.

If the ball is colliding with car3, the function checks if the ball is in front of the car or not. If the ball is in front of the car, the function returns the ball to its original position and decreases the number of attempts by one. If the number of attempts reaches three, the attempts () function is called to end the game.

Overall, the collision () function provides collision detection for the game and updates the game state based on collisions.

Texture

The load function: receive the image number and based on it use stb_image header file to load the image and store its width, height, and color system. then generate texture using the check function.

Check function: receive the image, generate texture and binding it. specifies the texture target to GL_TEXTURE_2D, texture format to RGB. use texture wrapping GL_REPEAT to repeat the image on x-axis and y-axis. use texture filtering GL_LINEAR takes an interpolated value from the texture coordinate's neighboring texels, approximating a color between the texels. The smaller the distance from the texture coordinate to a texel's center, the more that texel's color contributes to the sampled color.

Mouse & Keyboard Handling

There are two type of keyboard function:

1. **special Keyboard**

   that have not ASCI code on keyboard like f1,f2,f3 … f9 and arrows

2. **Keyboard**

   That have ASCI code on keyboard like Esc, m,n,w,s,d,a.

We use tow type of keyboard:

1 Keyboard

- **M -> go to main page**
- **N -> New game**
- **L -> next level**
- **D -> move ball to right**
- **A -> move ball to left**
- **W -> move ball to forward**
- **S -> move ball to back**
- **Esc -> to exit.**
- **Space -> to start the game.**

2 special Keyboard

- **f1 -> make the screen in the center.**
- **left -> move the coordinates of the point the camera is looking to left by decrease the value of x and z**
- **right -> move the coordinates of the point the camera is looking to right by increase the value of x and z**
- **up -> move the position of the camera to up by decrease the value of z and increase the value of z**
- **down -> move the position of the camera to up by increase the value of y and z**

mouse:

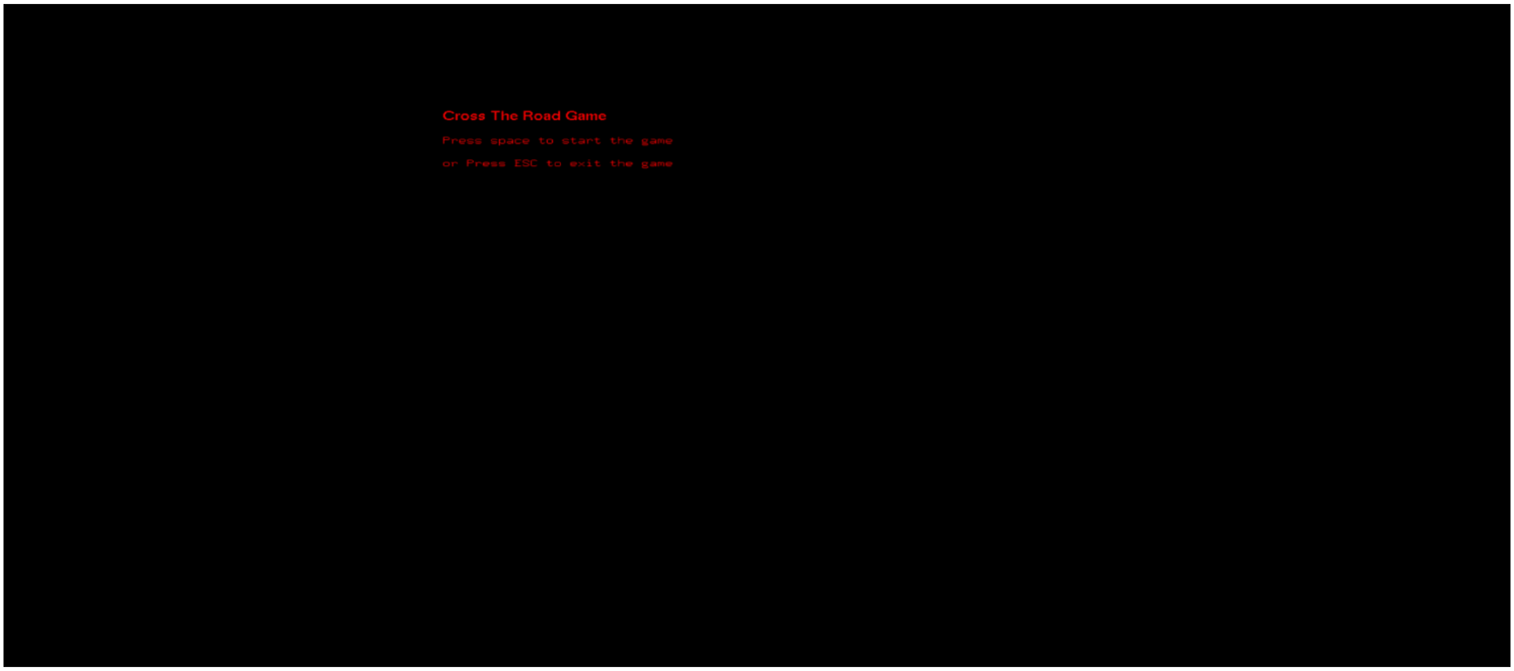
to change the color, depend on the button left or right and the state is down or up.

Show Result &Calculate Attempts

- **The attempts () function displays messages on the screen based on the value of attempts. If attempts is 0, the function displays a "Game Over" message and prompts the user to press "m" or "M" to return to the main page or "n" or "N" to start a new game. If the value of ball_z is less than or equal to -60, the function displays a "You Won, congrats!" message and prompts the user to press "l" or "L" to move to the next level.**
- **The messages are displayed using the glRasterPos3f function to set the position on the screen and the glutBitmapCharacter function to display each character of the message. The messages are displayed in red using the glColor3f function.**
- **Overall, the attempts () function provides a user interface for the game by displaying messages on the screen based on the game state.**

Images For
Run

Start Screen



Cross The Road Game

Press space to start the game

or Press ESC to exit the game

Game Screen