Assignment 2

10/14/2024

Starting with program 1 when implementing smart pointers I decided to implement shared pointers. The reason for this is that I wanted to be able to access the object from anywhere in the program. Using shared pointers lets me create multiple instances of this pointer to the object. I then created a complex 2d object class that creates multiple instances of the portate class to draw them in the correct orientation they should be displayed. It was able to do this with an overwritten version of the draw function that would then draw each part of the complex object.

With program two we began to work with animation. With my first, I started to get the road to display to the screen. This was done through a new class I called Road. This class's main purpose was to draw the road and also manage the cart on the road. In the private section of the class, we had a few variables such as the road type, a shared smart pointer to the cart object to manage it, and the current direction that the cart was moving in. We also had two functions that would be used to generate the road. These are two equations that would then be graphed to the screen as the road. Along with the actual equation functions we calculated and used the first derivative of these functions in order to get the slope at each current x coordinate. In the constructor, we set which road that we wanted to use. Along with this, we had a few helper functions such as get the correct y value at the corresponding x coordinate by passing the x value into the equation function. The same idea was implemented to get the slope with the derivative function. We also call a draw function to draw the road to the screen. Lastly, we had our cart modification functions assigned consisting of creating the cart object, moving the cart, drawing the cart, and flipping the cart. The road was drawn by using GL_LINE_STRIP and plugging the full scale of the window every 0.1f by plugging it into the function for the road. When it comes to actually moving the cart I am able to use the road class helper functions to manage the current cart values. The cart class is used to store the variables used to draw the cart to the screen. When the cart has to be moved we use the move cart function in the road class. This function gets passed the speed and then calculates the needed slope of the cart and the current position that the cart needs to be drawn to the screen. Once these calculations are completed the cart instance is then moved by calling its sub-functions to update the position and orientation. Along with this it checks to make sure the cart is still in bounds in the world and turns it around if needed. Other than storing the cart's values in regards to position the cart class also has a draw function and a rotate wheel function to calculate how the wheels need to be rotated. To get the orientation we calculate the slope of the equation by using the first derivative. I had no issues implementing the speed component of the program. The cart's movement is calculated by repeatedly adding the cart speed value to its current position, adjusting based on direction (left or right). These updates occur regularly through a timer function that triggers the recalculations. The road's curve is computed by evaluating a mathematical function for a series of x-values and calculating corresponding y-values to define the road’s shape. We are able to make these calculations by normalizing the tangential direction vector. The cart's movement follows the slope of the road, so the function computes a unit vector that is tangential to the road

at the current point. To calculate the new x position we account for the speed and direction of the cart by doing newX=currentX+cartDirection_×speed×dx. The new y coordinate is calculated by passing the x value into the road equation. To calculate the orientation we use atan2(dy, dx) to calculate the angle of the cart relative to the horizontal axis. We then convert this value to radians. To calculate the wheel speed we take the speed traveling and divide it by the radius of the wheels.