Carnival in OpenGL

CS352: Computer Graphics & Visualization Lab

Project Report

Course Instructor:
Dr Somnath Dey

Submitted By:

Mihir Karandikar – 200001044
Mukul Jain – 200001050
Nilay Ganvit – 200001053

Introduction

A carnival is a popular destination for people of all ages, offering various attractions and entertainment. Our project aims to create a virtual carnival using OpenGL, a powerful graphics library that allows us to create 3D graphics and animations.

The carnival is implemented using OpenGL, a cross-platform graphics library that allows us to create 3D graphics and animations. The project utilises OpenGL's capabilities to create realistic models of the rides, stalls, and other elements. We have also used various lighting and texturing techniques to enhance the overall look and feel of the environment.

The carnival has several attractions, including roller coasters, around-the-world and a carousel. Each ride is modelled using 3D graphics, allowing users to experience the thrill and excitement of being in a real carnival. The park also has various stalls, a cafeteria, and food vendors, providing users with a complete experience.

Our team has designed and developed various objects and models using OpenGL primitives such as cubes, spheres, and cylinders. We have also created custom textures and shaders to enhance the visual quality of the project scene.

The project uses modern OpenGL techniques and technologies such as lighting, shading, and transformations. We have also incorporated user interaction and input through keyboard and mouse events.

Overall, our project aims to showcase the capabilities of OpenGL in creating a virtual carnival with realistic and immersive graphics and animations. We hope users will enjoy exploring the virtual park and experiencing the thrill of the various rides and attractions.

Specifications

Controls

- Mouse Drag for camera control
- Scroll Up for zoom out
- Scroll Down for zoom in
- ‘a’ look left
- ‘d’ look right
- 1 to start around the world ride
- 2 to start the roller coaster ride
- 3 to start the carousel ride
- ‘i’ to increase light intensity
- ‘k’ to decrease light intensity
- 6 to toggle spotlight 1
- 7 to toggle spotlight 2
- 8 to toggle spotlight 3
- 9 to toggle spotlight 4
- ‘s’ sit in the roller coaster ride
- ‘e’ to exit the roller coaster ride

Libraries Required

- g++
- libglu1-mesa-dev
- freeglut3-dev
- mesa-common-dev

How to Run Project (in Linux)

To run the Project, do the following

- git clone https://github.com/MihirK1212/cs352-carnival
- cd cs352-carnival
- ./run.sh

Functionalities Implemented

1) Roller Coaster Ride

We implemented a roller coaster ride with a customised track entirely from scratch. The shape of the track was defined using a parametric function (x(θ), y(θ), z(θ)). The poles supporting the track were also placed based on these parametric coordinates. The car of the roller coaster is aligned, and moves along the tangent of the track as theta varies between 0 and 360.
The function used to generate the track and its corresponding tangent is:

```
double x = 10*cos(theta);
double z = 10*sin(theta);
double y = (x*x*x + z*z - 3*x + 4*z)/100;

double dx = -10*sin(theta);
double dz =  10*cos(theta);
double dy = (3*x*x*dx + 2*z*dz - 3*dx + 4*dz)/100;
```

We want to align the track with the tangent, so we perform the inverse of the operation of aligning the tangent with the z-axis, as the track is initially generated along the z-axis. Hence, if we apply this inverse operation on the track, it will get aligned with the tangent.

Similarly, the roller coaster's car is aligned with the tangent based on its current theta position. The ride moves along the track as the theta position varies between 0 and 360 degrees.

2) Human

A minimalistic human has been implemented, and the user can control this human's motion. Arrow keys can be used to control its orientation and motion. The hands and legs of the human resemble walking motion as the human moves. The human is holding a balloon in his hand.

Initially, the body parts of the human are generated with respect to the origin, and the human is rotated about the y-axis based on his orientation. Finally, the human is translated to its current position in the coordinate system.

We have also added the ability for humans to sit on the roller coaster ride. The human sits in the roller coaster's car and moves along. To do this, we translated the human to the current position of the roller coaster and made his face along the tangent of the track.

3) Lighting

To add light to the scene, we have used the inbuilt **glLightfv** function and passed to it the following paraments:

GL_POSITION: to specify the position of the light
GL_AMBIENT: for ambient light
GL_DIFFUSE: for diffused light
GL_SPECULAR: for specular light

```
GLfloat light_ambient[] = {1.0, 1.0, 1.0, 1.0};
GLfloat light_diffuse[] = {1.0, 1.0, 1.0, 1.0};
GLfloat light_specular[] = {1.0, 1.0, 1.0, 1.0};
GLfloat light_position[] = {20.0, 50.0, 0.0, 1.0};

glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
```

Based on these parameters, we can add lighting and shading effect to the matrix space of OpenGL. We also added keyboard controls to vary the intensity of the diffused light to produce a gradual and progressive lighting effect.

4) Carousel

The carousel is a familiar ride in which horses are attached to several poles arranged in a circle, and the system rotates around its axis. The carousel we created has a conical top, a flat cylindrical base, and horses attached to the poles circularly.

To create the horse, we created separate body parts using the **gluCylinder, gluSphere, and gluCone** functions and translated them to the appropriate position. Finally, we applied some translation in the y-axis using the sinusoidal function to give a wavy motion to the position of the horses.

5) Camera Control

The camera had no mouse control in our reference code, and everything had to be viewed using keyboard keys. We added **mouse control** so that the park can be viewed easily.

In OpenGL, the camera view is defined using three parameters:

- the **'eye'** coordinates: which indicate the position of the viewer's eyer
- the **'reference'** coordinates: which indicate the position of the point at which the eye is looking
- the **'up'** axis for the view reference coordinate system

For the world view camera control, we detected the mouse button press and mouse button release events, and in between, we measured the mouse's motion. Accordingly, we varied the values of the 'eye' coordinates. The user will still have to use keyboard keys to look left and right, i.e., to move the 'reference' coordinates left and right. We can also zoom in and out by scrolling the motion of the mouse/keypad.

We have added a camera angle that moves behind the human, similar to first-player games. The 'eye' coordinates are always behind the human, and the 'reference' coordinates are always in front of him.

One final camera angle allows you to move along with the roller coaster. It simulates the actual first-person view as the roller coaster moves along the track.

6) Around the World Ride

Around the World is a ride where the seats are initially arranged circularly around the central rod and are at their bottommost position. On starting the ride, the seats rotate about the z-axis and go up. At their peak position above, the seats rotate about the central axis.

The motion of this ride was simulated by maintaining the atwTheta and atwAlpha values, which are the angles for rotation about the y and z axes, respectively.

7) Sky

We implemented the sky as a bounding box of 4 faces, each with a texture. The sky is made to give a pleasant feeling to the viewer corresponding to the beach.

8) Beach

The Carnival is set on a sandy beach which is actually a circular ground with a sand texture. The texture repeats itself again and again to fill the ground.

9) Texture

Several custom textures are implemented on the objects to give them a more personalised look. Textures have been added using the **'LoadTexture'** function, which uses the **'sgi'** file format to load the texture, and allows us to use them using GL_LOAD_TEXTURE.

10) Boundary Wall

Boundary has been created around the Carnival, and to set it apart, it has been given the IIT Indore Logo as the texture.

11) Flags

Flags have been positioned around the boundary of different colours, made using a cylindrical pole and rectangle.

12) Swimming Pool

A swimming pool has been created with textures along the sides. We tried to make the textures transparent to give a more wavy/watery look. However, due to our implementation of lighting effects, the OpenGL library doesn't support making the textures transparent.

13) Scary House

A Scary House has been created behind the Cafeteria, giving viewers varied rides and a haunting experience. It is a Cuboid created using textures along the walls and the roof.

14) Cafeteria

Cafeteria is a place with many small objects implemented along with one another. Firstly, a **Table** is created using a cuboid whose base is a cylinder on which it lies. An **umbrella** is constructed on the top of the cylinder, which is created using a cylinder converted into a cone with a small radius at the top and a large radius at the bottom. That cone is itself made from several segments. The **chair** has been created using many small cuboids, which create the chair's legs, seat and back. Similarly, a **Teapot** has been created using inbuilt GLU functions. All these things are replicated 12 times into 3 rows and 4 columns.

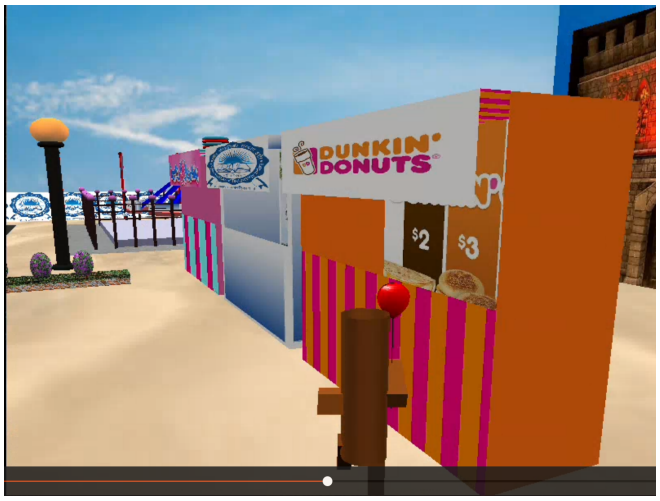
A **bush** has been created along the sides using a sphere and a spherical Texture. These bushes are placed on a **brick wall** which is a cuboid, along with textures. Finally, **3 different shops/eateries** are also implemented and given appropriate colour schemes to look real.

Output

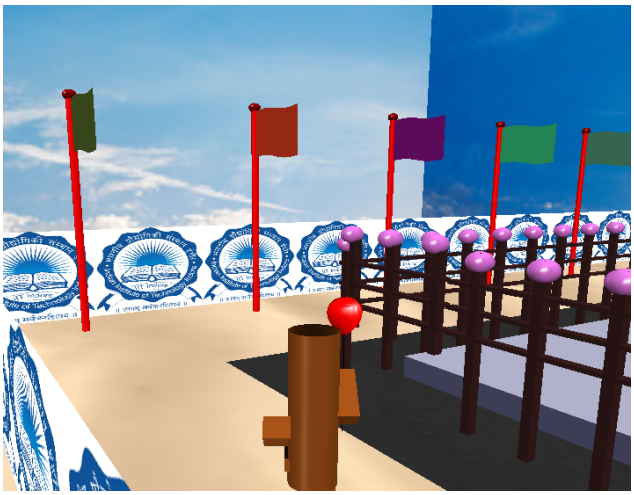
Changing Light Intensity, Evening Time in the left image and Daytime in the right image.

Roller coaster in the left image and Around the World ride in the right image.

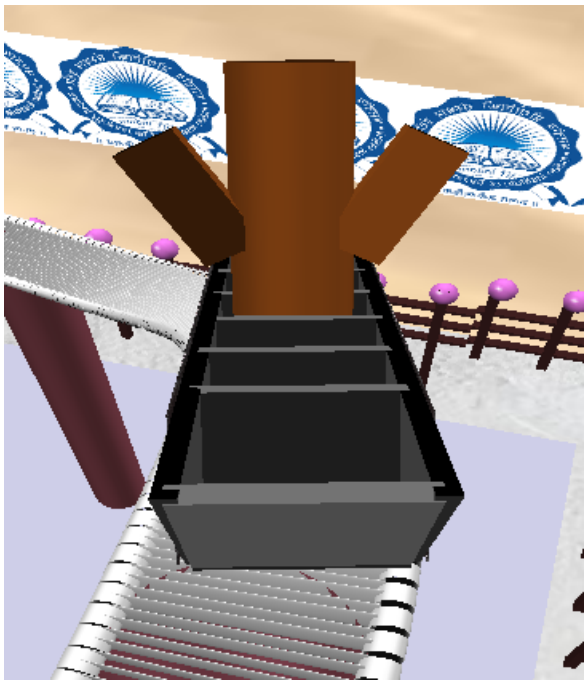

Carousel Image in the left image and Cafeteria in the right image.



3 Shops can be seen in the left image. On the right, we can see lighting due to lamp posts, the swimming pool and a close look of the sitting area in the Cafeteria.

Scary House and the effect of lighting by Lamp posts can be seen in the left image. On the right side, we can see the boundary walls and the flags



In this image, the human is riding the Roller Coaster and the camera view changes based on that.

References

- https://github.com/n-gauhar/3D-Amusement-Park