

# *Amusement park*

*Computer Graphics  
mini project*



# *Group members*

**Lama Alfreah**

**4250107**

**F22**

**Asma Alsharif**

**4253618**

**F22**

# Introduction

## Scene Concept:

The amusement park includes a Ferris wheel, cable cars, balloons, and 3D characters.

## Key Elements:

- **Ferris Wheel:** Central attraction with rotating pods.
- **Cable Cars:** A series of suspended cars traveling along a wire.
- **Balloons:** Floating and shrinking as part of the animation.
- **Characters:** Represent visitors in the park, including a girl, a boy, and a cat.

# Challenges and Solutions

## 1. Object Positioning:

- **Challenge:** Identifying the right coordinates for accurate 3D positioning.
- **Solution:** Achieved through experimentation and transformations.

## 2. Smooth Animations:

- **Challenge:** Synchronizing motions.
- **Solution:** Used JavaFX transitions for seamless effects.

## 3. Realism:

- **Challenge:** Enhancing 3D depth and appeal.
- **Solution:** Applied textures, lighting, and antialiasing.

# Implementation Details

## 3D Shapes Used:

- Ferris Wheel: **Spheres** for the center, **boxes** for pods, and wire ropes.
- Cable Cars: **Boxes** arranged in a sequence with a supporting wire.
- Balloons: **Spheres** for large and small balloons with cylinders for ropes.
- Characters: **Box** shapes textured with images for the girl, boy, and cat.

## Transformations Applied:

- **Translation:** Positioning objects (e.g., balloons and characters).
- **Rotation:** Animating the Ferris Wheel continuously, while the cable cars and balloons rotate on their respective axes when triggered.
- **Scaling:** Reducing the size of balloons during animation.

# Implementation Details

## Textures Applied:

- Ferris wheel pods and cable cars are textured using images (ferrisWheelPod.png and cableCarPod.png).
- Balloons and characters are textured using images (largeBalloon.jpg, smallBalloon.jpg, girl.png, cat.png, and boy.png).

## Lighting:

- **Ambient Light:** Provides uniform lighting to the entire scene.
- **Point Light:** Adds a focal light source.

# Implementation Details

## Implemented Animations:

- Ferris Wheel Rotation: Continuous rotation.
- Cable Cars Movement: Smooth horizontal movement across the scene.
- Balloon Animation: Simultaneous upward movement and shrinking.

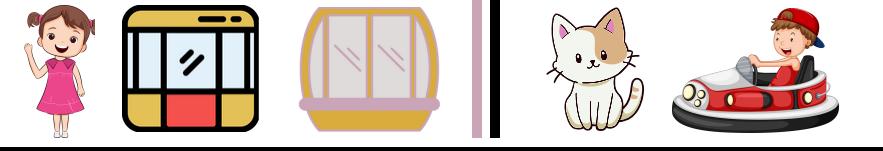
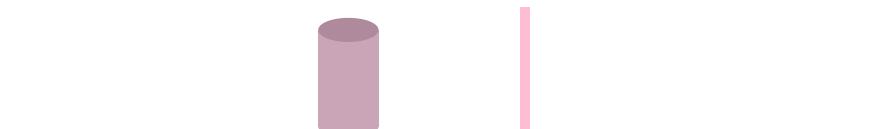
## Animation Details in Code:

- **RotateTransition** for the Ferris wheel.
- **TranslateTransition** for the cable cars and balloons.
- **ScaleTransition** for the shrinking effect of balloons.
- **ParallelTransition** to synchronize balloon animations.

# Implementation Details

2



Basic 3D shapes	Boxes	
	Spheres	
	Cylinders	
Translation		
3D Transformations		
Scaling		
Rotation		
Texture Mapping	AmbientLight	
	PointLight	
	Translation	
Basic Animation	Scaling	
	Rotation	

# User Guide

3

Start rotating  
the Ferris wheel



Stop the  
Ferris wheel



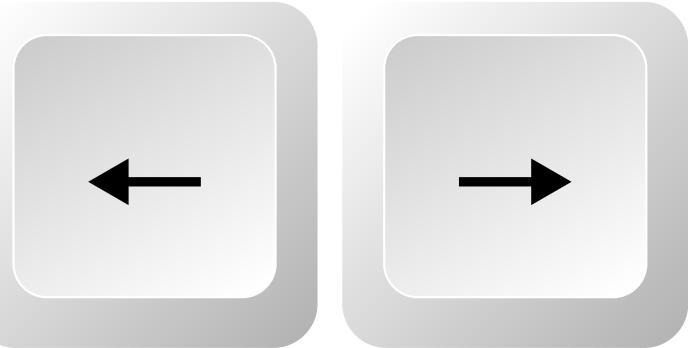
Move the  
cable cars



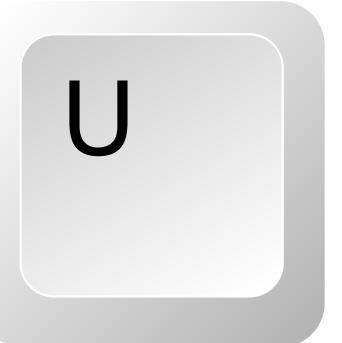
Move the  
boy  
forward



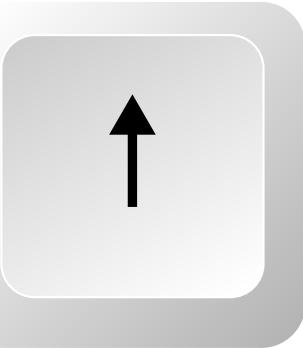
Move the cat  
horizontally



Rotates  
cable  
cars and  
balloons



Trigger the  
balloon animation  
(fly up and shrink)



# Conclusion

## Lessons Learned

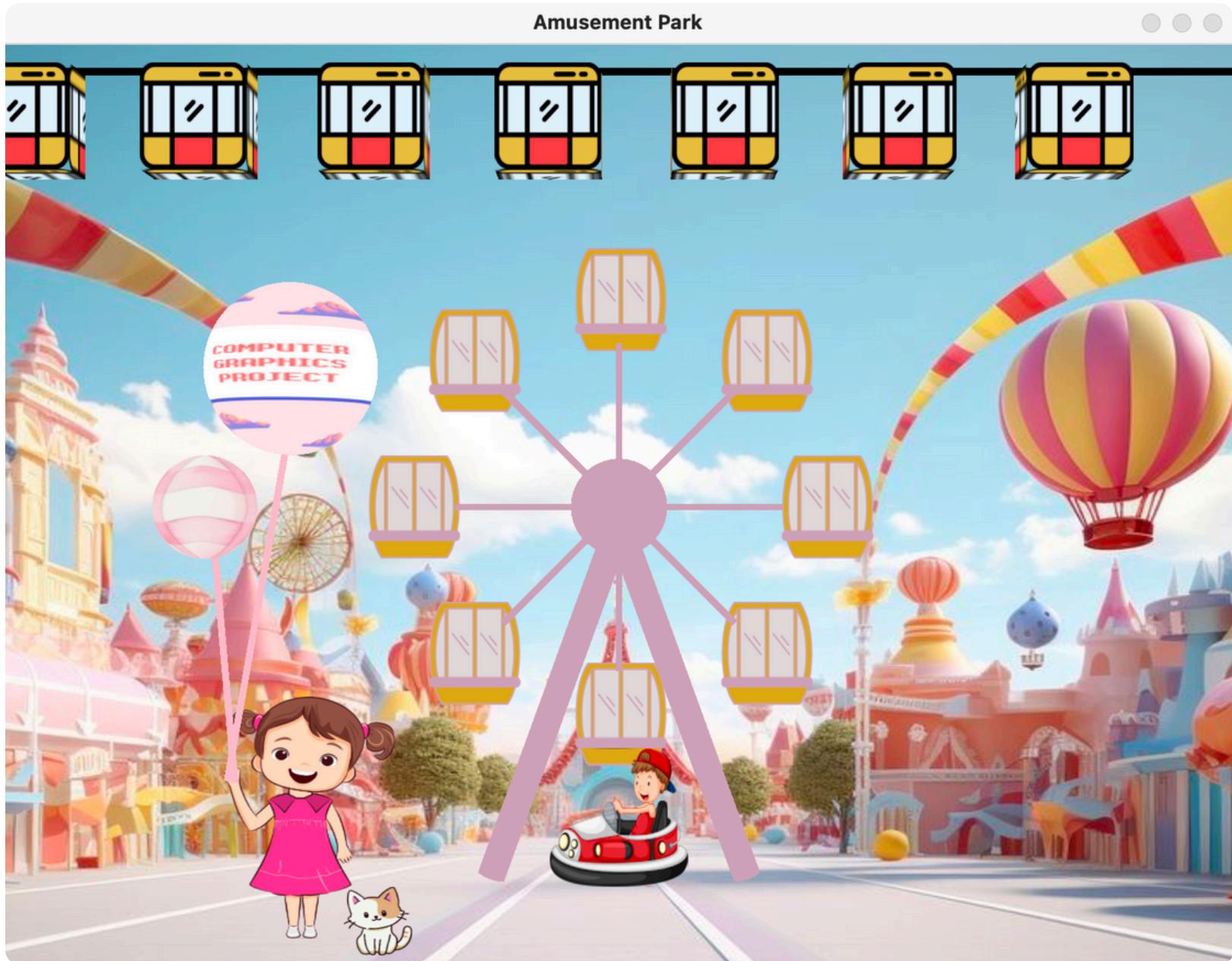
- Gained hands-on experience with JavaFX for 3D graphics.
- Understood the importance of lighting and textures in enhancing realism.
- Improved debugging and problem-solving skills during object placement and animation setup.
- Strengthened knowledge of user interaction in graphical applications.

# Conclusion

## Potential Improvements:

- Add more rides (e.g., a carousel).
- Introduce moving crowds or more interactive characters.
- Use advanced textures and materials for ultra-realistic effects.
- Enhance the camera system for user-controlled zoom and rotation.

# Visuals



## Ferris wheel

### Components:

Central sphere, 8 pods, wire ropes, supporting columns.

### Animation:

R: Rotate Ferris Wheel.  
S: Stop rotation.



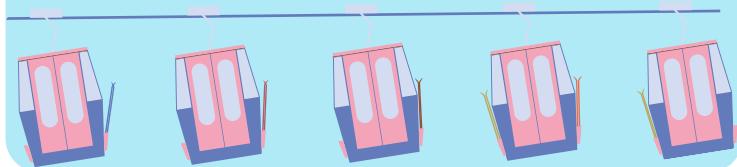
## Cable car

### Components:

20 textured cars on a black wire rope.

### Animation:

G: Move cable cars horizontally.  
U: Rotates around the X-axis.



## Balloons

### Components:

Large and small balloons with pink ropes.

### Animation:

Arrow Up: Fly upward and shrink.  
U: Rotate around the Y-axis.



## Characters

### Components:

Girl, boy, and cat.

### Interaction:

Arrow Keys: Move cat left/right.  
Z: Move boy forward.



# Visuals



In this screenshots, the Ferris Wheel has rotated fully, and the cable cars have moved along the wire and rotated. The balloons have flown upward, shrunk, and rotated, while the characters (boy and cat) are moving, with the boy moving forward and the cat shifting horizontally.

# Implementation

## Code

## Background

```
// Set up background image and center it in the scene
Image backgroundImage = new Image(getClass().getResourceAsStream("amusementPark.JPG"));
ImageView amusementPark = new ImageView(backgroundImage);
amusementPark.setFitWidth(500);
amusementPark.setFitHeight(300);
amusementPark.getTransforms().add(new Translate(-250, -150, 0));
```

## Building the components of the ferris wheel

```
// The start of building the components of the ferris wheel
// Centre for ferris wheel
Sphere ferrisWheelCentre = new Sphere(14); // Sphere (radius)
ferrisWheelCentre.setMaterial(new PhongMaterial(Color.web("#caa5b8"))); // Set 3D shape material

// Set surfaces texture materials for ferris wheel pod
PhongMaterial ferrisWheelPod = new PhongMaterial();
ferrisWheelPod.setDiffuseMap(new Image(getClass().getResourceAsStream("ferrisWheelPod.png")));

// Ferris wheel pods and wireropes
Group ferrisWheelPodsAndWireropes = new Group(); // Group to hold all pods and wires
for (int i = 0; i < 8; i++) {
    // Loop to create 8 pods and their wires
    // Create a wire rope (thin box)
    Box wireropes = new Box(95, 2, 2); // Box (width, height, depth)
    wireropes.setMaterial(new PhongMaterial(Color.web("#caa5b8"))); // Set 3D shape material
    wireropes.setRotate(i * -45); // Rotate the wires to align with the pods
    // Create a pod as a small box
    Box pod = new Box(30, 30, 0); // Box (width, height, depth)
    pod.setMaterial(ferrisWheelPod); // Set 3D shape material
    double angle = i * 45; // 360° / 8 = 45° // Calculate the angle for each pod and wire
    // Calculate the position of the pods
    double x = 60 * Math.cos(Math.toRadians(angle));
    double y = 60 * Math.sin(Math.toRadians(angle));
    pod.getTransforms().add(new Translate(x, y, 0));
    // Add the pods and wires to the group
    ferrisWheelPodsAndWireropes.getChildren().addAll(wireropes, pod);
}

// Create a support columns for the Ferris Wheel
Cylinder column2 = new Cylinder(4, 100); // Cylinder (radius, height)
column2.setMaterial(new PhongMaterial(Color.web("#caa5b8"))); // Set 3D shape material
column2.setTranslateY(60);
column2.setTranslateX(-20);
column2.setRotate(20);

Cylinder column3 = new Cylinder(4, 100); // Cylinder (radius, height)
column3.setMaterial(new PhongMaterial(Color.web("#caa5b8"))); // Set 3D shape material
column3.setTranslateY(60);
column3.setTranslateX(20);
column3.setRotate(-20);
// Add all Ferris Wheel components to the root
root.getChildren().addAll(amusementPark, column2, column3, ferrisWheelCentre, ferrisWheelPodsAndWireropes);
// The end of building the components of the ferris wheel.
```

# Implementation Code

## *Building the components of the cable car*

```
// The start of building the components of the cable car
// Set surfaces texture materials for cable car pods
PhongMaterial cableCarPod = new PhongMaterial();
cableCarPod.setDiffuseMap(new Image(getClass().getResourceAsStream("cableCarPod.png")));

// Create the car pods
Group cableCarPods = new Group();
for (int i = 0; i < 20; i++) {
    Box pod = new Box(30, 30, 30); // Box (width, height, depth)
    pod.setMaterial(cableCarPod);
    // Position the cable car pods
    pod.getTransforms().add(new Translate(130 - (i * 50), -110, 0));
    cableCarPods.getChildren().add(pod);
}
// Create the wire rope for the cable cars.
Box wirerope = new Box(2000, 2, 2); // Box (width, height, depth)
wirerope.setMaterial(new PhongMaterial(Color.BLACK)); // Set 3D shape material
wirerope.getTransforms().add(new Translate(0, -126, 0));

// Add the cable cars and wire to the root
root.getChildren().addAll(wirerope, cableCarPods);
// The end of building the components of the cable car
```

# Implementation

## Code

### *Texture materials for balloons and characters*

```
// Set surfaces texture materials for balloons and characters
PhongMaterial largeBalloonMaterial = new PhongMaterial();
largeBalloonMaterial.setDiffuseMap(new Image(getClass().getResourceAsStream("largeBalloon.jpg")));
PhongMaterial smallBalloonMaterial = new PhongMaterial();
smallBalloonMaterial.setDiffuseMap(new Image(getClass().getResourceAsStream("smallBalloon.jpg")));
PhongMaterial boy = new PhongMaterial();
boy.setDiffuseMap(new Image(getClass().getResourceAsStream("boy.png")));
PhongMaterial girl = new PhongMaterial();
girl.setDiffuseMap(new Image(getClass().getResourceAsStream("girl.png")));
PhongMaterial cat = new PhongMaterial();
cat.setDiffuseMap(new Image(getClass().getResourceAsStream("cat.png")));

// Girl
Box girlTemplate = new Box(50, 70, 0); // Box (width, height, depth)
girlTemplate.setMaterial(girl); // Set 3D shape material
girlTemplate.setTranslateY(90);
girlTemplate.setTranslateX(-90);

// Cat
Box catTemplate = new Box(20, 20, 0); // Box (width, height, depth)
catTemplate.setMaterial(cat); // Set 3D shape material
catTemplate.setTranslateY(120);
catTemplate.setTranslateX(-70);

// Boy
Box boyTemplate = new Box(40, 40, 0); // Box (width, height, depth)
boyTemplate.setMaterial(boy); // Set 3D shape material
boyTemplate.setTranslateY(90);
```

# Implementation

## Code

### *Building the components of the balloons*

```
// The start of building the components of the balloons

// Large balloon
Sphere largeBalloon = new Sphere(25); // Sphere (radius)
largeBalloon.setMaterial(largeBalloonMaterial); // Large balloon texture
largeBalloon.setTranslateX(-95);
largeBalloon.setTranslateY(-40);

// Small balloon
Sphere smallBalloon = new Sphere(15); // Sphere (radius)
smallBalloon.setMaterial(smallBalloonMaterial); // Small balloon texture
smallBalloon.setTranslateX(-120);
smallBalloon.setTranslateY(0);

// Balloons ropes
Cylinder rope1 = new Cylinder(1, 80); // Cylinder (radius, height)
rope1.setMaterial(new PhongMaterial(Color.PINK)); // Set 3D shape material
rope1.setTranslateX(-115);
rope1.setTranslateY(40);
rope1.setRotate(-5);

Cylinder rope2 = new Cylinder(1, 100); // Cylinder (radius, height)
rope2.setMaterial(new PhongMaterial(Color.PINK)); // Set 3D shape material
rope2.setTranslateX(-105);
rope2.setTranslateY(30);
rope2.setRotate(10);

// Create a group to contain all parts of balloons
Group balloons = new Group();
balloons.getChildren().addAll(largeBalloon, smallBalloon, rope1, rope2);

// Add the balloons to the root
root.getChildren().addAll(balloons, girlTemplate, catTemplate, boyTemplate);
// The end of building the components of the balloons
```

# Implementation

## Code

### Keyboard controls

```
// Create a rotation animation for the Ferris Wheel pods and wire ropes.  
RotateTransition rotateWheel = new RotateTransition(Duration.seconds(20), ferrisWheelPodsAndWireropes);  
// Create a horizontal movement animation for the cable cars.  
TranslateTransition moveRightCableCar = new TranslateTransition(Duration.seconds(10), cableCarPods);  
// Create a vertical movement animation for the balloons to fly upwards.  
TranslateTransition flyUp = new TranslateTransition(Duration.seconds(20), balloons);  
// Create a scaling animation to shrink the balloons.  
ScaleTransition shrink = new ScaleTransition(Duration.seconds(10), balloons);  
// Combine the upward movement and shrinking animations into a parallel animation.  
ParallelTransition flyAndShrink = new ParallelTransition(flyUp, shrink);  
  
// Keyboard controls  
scene.setOnKeyPressed(event -> {  
    if (event.getCode() == KeyCode.R) {  
        // Rotation for Ferris wheel  
        rotateWheel.setAxis(Rotate.Z_AXIS); // Set the rotation axis for the Ferris Wheel to the Z-axis  
        rotateWheel.setByAngle(360); // Rotate the Ferris Wheel 360 degrees  
        rotateWheel.setCycleCount(RotateTransition.INDEFINITE); // Set the rotation to repeat indefinitely  
        rotateWheel.play(); // Start the rotation animation  
    } else if (event.getCode() == KeyCode.S) {  
        rotateWheel.stop(); // Stop the Ferris Wheel rotation  
    } else if (event.getCode() == KeyCode.G) {  
        moveRightCableCar.setByX(250); // Move the cable cars 250 units along the X-axis  
        moveRightCableCar.setCycleCount(TranslateTransition.INDEFINITE); // Set the cable cars to move continuously  
        moveRightCableCar.play(); // Start the cable car movement animation  
    } else if (event.getCode() == KeyCode.SPACE) {  
        flyUp.setByY(-200); // Move the balloons upward by 200 units along the Y-axis  
        flyUp.setCycleCount(1); // Play the upward movement animation once  
        // Shrink the balloons to 50% of their original size in all dimensions  
        shrink.setToX(0.5);  
        shrink.setToY(0.5);  
        shrink.setToZ(0.5);  
        shrink.setCycleCount(1); // Play the shrinking animation once  
        flyAndShrink.play(); // Start both the upward movement and shrinking animations together  
    } else if (event.getCode() == KeyCode.LEFT) {  
        catTemplate.setTranslateX(catTemplate.getTranslateX() - 5); // Move the cat 5 units to the left along the X-axis  
    } else if (event.getCode() == KeyCode.RIGHT) {  
        catTemplate.setTranslateX(catTemplate.getTranslateX() + 5); // Move the cat 5 units to the right along the X-axis  
    } else if (event.getCode() == KeyCode.Z) {  
        boyTemplate.setTranslateZ(boyTemplate.getTranslateZ() - 20); // Move the boy 20 units forward (negative Z direction)  
    } else if (event.getCode() == KeyCode.U) {  
        cableCarPods.setRotationAxis(Rotate.X_AXIS); // Set rotation axis to X-axis  
        cableCarPods.setRotate(cableCarPods.getRotate() + 10); // Rotate +10 degrees along X-axis  
  
        largeBalloon.setRotationAxis(Rotate.Y_AXIS); // Set rotation axis to Y-axis  
        largeBalloon.setRotate(largeBalloon.getRotate() + 10); // Rotate +10 degrees along X-axis  
  
        smallBalloon.setRotationAxis(Rotate.Y_AXIS); // Set rotation axis to Y-axis  
        smallBalloon.setRotate(smallBalloon.getRotate() + 10); // Rotate +10 degrees along X-axis  
    }  
});
```

# Resources

Official documentation for working with 3D shapes, transformations, and animations in JavaFX.

[JavaFX 3D Graphics Documentation](#)

Detailed reference for transitions like RotateTransition, TranslateTransition, and ParallelTransition.

[JavaFX Animation APIs](#)

*The end :)*

*Thank*

*you*

*for*

*listening*