Java Report-OOP-Group20

Flappy Bird Game

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Assignments | Assignees | Time |
| 1 | Selection of the research topic, preparation of a research proposal, allocation of tasks among group members and writing report | Quân | 1 day |
| 2 | Game interface/game panel | Duy | Week 1 |
| 3 | The bird | Quân | Week 1 |
| 4 | Animation | An | Week2-4 |
| 5 | Background and instructions | Cường | Week 1 |
| 6 | Key interaction | Ân | Week 2-4 |
| 7 | Action Event | An | Week 2-4 |
| 8 | Starting and random positions features | Ân | Week 2-4 |
| 9 | Spawn pipe | Ân | Week 2-3 |
| 10 | Loop Pipe | Ân | Week 2-4 |
| 11 | Collide with pipe | An | Week 2-4 |
| 12 | Different deaths | Cường | Week 5 |
| 13 | Score calculation | Duy | Week 5 |
| 14 | Spawn pipe randomly | Duy | Week 4 |
| 15 | Skins for Bird | Quân | Week 6 |

INTRODUCTION: Flappy Bird is a popular mobile game with a simple interface but high level of difficulty. In this game, the player controls a bird flying over a series of tall pipes filled with water. The player earns points by successfully passing through each pair of pipes.

1.2 Features:

- Interface creation feature

- Add bird feature

- Add animation feature

- Add background feature

- Spawn bird at starting point feature

- Spawn ground at starting point feature

- Spawn pipes at random positions feature

- Spawn pipes at random positions feature

- Loop pipe feature

- Collide with pipe feature

- Different deaths feature

- Score calculation feature

1.Game interface/game panel

|  |
| --- |
| **import** javax.swing.JFrame;  **import** javax.swing.JPanel;  **import** java.awt.Graphics; |
|  |
|  |

4.Keying

2.8 Repetition of pipe generator

The “Repetition of pipe generator” function in this Flappy Bird game relates to using a continuous pipe that the bird has to get through. This was done by creating a new pipe on the right side of the screen every time a pipe disappeared through the left side of the screen

|  |
| --- |
| if(scroll % 90 == 0) {  Rectangle r = new Rectangle(WIDTH, 0, GamePanel.PIPE\_W, (int) ((Math.random()\*HEIGHT)/6f + (0.2f)\*HEIGHT));  int h2 = (int) ((Math.random()\*HEIGHT)/6f + (0.2f)\*HEIGHT);  Rectangle r2 = new Rectangle(WIDTH, HEIGHT - h2, GamePanel.PIPE\_W, h2);  rects.add(r);  rects.add(r2);  } |

9. Collision Detection:

The code includes collision detection logic to check if the bird intersects with any of the obstacles or gets out of bounds. If detected, the game ends and the player loses. To implement this function, we need to check for collisions between the bird and the pipes after each frame.

|  |
| --- |
| for(Rectangle r : rects) {  r.x-=3;  if(r.x + r.width <= 0) {  toRemove.add(r);  }    if (r.intersects(Math.round(bird.x - (bird.getWidth() / 2)), Math.round(bird.y - (bird.getHeight() / 2)), bird.getWidth(), bird.getHeight())) { bird.playHitSound();  bird.playHitSound();  showingResult1 = true;  showingResult2 = false;  showingResult3 = false;  game = false;  }  }  rects.removeAll(toRemove);  time++;  scroll++;    if (bird.y > HEIGHT) {  bird.playHitSound();  showingResult1 = false;  showingResult2 = true;  showingResult3 = false;  game = false;    }  if (bird.y + bird.RAD < 0) {  bird.playHitSound();  showingResult1 = false;  showingResult2 = false;  showingResult3 = true;  game = false;  } |

10. Scoring

The scoring system helps the player to see their current progression in the game. The getScore method returns the value representing the player's score based on the duration of their gameplay.

|  |
| --- |
| public int getScore() {  return time;  } |

11. Random Pipe Generation

The function "random pipe generation" in the Flappy Bird game in Java involves creating pipes at random positions on the screen. This creates unpredictability and challenges for the player, making the game more engaging. To implement this function, we need to create new pipes at random positions after a certain period of time or after a specific pipe has moved off the screen. We can use the Math.random() class in Java to generate random numbers and use them to determine the positions of the new pipes.

Example:

|  |
| --- |
| if(scroll % 90 == 0) {  Rectangle r = new Rectangle(WIDTH, 0, GamePanel.PIPE\_W, (int) ((Math.random()\*HEIGHT)/6f + (0.2f)\*HEIGHT));  int h2 = (int) ((Math.random()\*HEIGHT)/6f + (0.2f)\*HEIGHT);  Rectangle r2 = new Rectangle(WIDTH, HEIGHT - h2, GamePanel.PIPE\_W, h2);  rects.add(r);  rects.add(r2);  } |

12. Different deaths

|  |
| --- |
| if (r.intersects(Math.round(bird.x - (bird.getWidth() / 2)), Math.round(bird.y - (bird.getHeight() / 2)), bird.getWidth(), bird.getHeight())) { bird.playHitSound();  showingResult1 = true;  showingResult2 = false;  showingResult3 = false;  game = false;    }  }  rects.removeAll(toRemove);  time++;  scroll++;  if (bird.y > HEIGHT) {  bird.playHitSound();  showingResult1 = false;  showingResult2 = true;  showingResult3 = false;  game = false;    }  if (bird.y + bird.RAD < 0) {  bird.playHitSound();  showingResult1 = false;  showingResult2 = false;  showingResult3 = true;  game = false;    } |

15. Skins for bird

|  |
| --- |
| String[] colors = {"red", "blue", "green"};  String color = (String) JOptionPane.*showInputDialog*(**null**, "Choose a color for the bird:", "Color Selection", JOptionPane.***QUESTION\_MESSAGE***, **null**, colors, colors[0]);  **try** {  **if** (color.equals("red")) {  img = ImageIO.*read*(**new** File("flappybirdred.png"));  } **else** **if** (color.equals("blue")) {  img = ImageIO.*read*(**new** File("flappybirdblue.png"));  } **else** **if** (color.equals("green")) {  img = ImageIO.*read*(**new** File("flappybirdgreen.png"));}  } **catch** (IOException e) {  e.printStackTrace();  } |

Conclusion

Analysis:

- Simple and easy to play: The Flappy Bird game is developed using Java and has a simple and user-friendly interface. The gameplay is straightforward, and the controls are easy to understand. The player only needs to control the bird's flight by tapping the screen to make it fly over a series of water pipes without colliding with them.

- High level of challenge: Despite its simplicity, the game presents a high level of challenge, keeping the player engaged.

- Customizable settings: The game allows players to customize various settings such as sound effects, difficulty level, etc., providing a personalized gaming experience.

- Leaderboard: Players can compare their scores with other players through the leaderboard, which adds an extra layer of motivation for them to achieve higher scores.

Limitations:

- Simple graphics: The game has 2D graphics that are not very sophisticated or visually appealing compared to other modern games.

- Limited features: The game has limited features, focusing primarily on controlling the bird's flight over water pipes. It does not have multiple levels or modes that allow players to explore different environments or challenges.

- Limited replay value: Each time the player plays, they will repeat the same level, making the game less varied and less replayable compared to other games with multiple levels or modes.