Bouncing Cube - By Jordan Ukawoko, Daniel Bodlund & Wimer Gunnarson

IST Java Programming Course

This is a project made for the Java IST Course in INSA LYON

- Object Orientated Programming
- Use of Abstract Classes & Interfaces
- Use of imported libaries for GUI

About

We created a game called Bouncing Cube. Have you heard of the game flappy birds? This fanmade game draws heavy inspiration from the smash hit mobile game released in 2013.

How to Play

- User must tap the screen to start the game
- Press space to keep the cube flying. There will be a score counter. Can you beat your high score?
- Try playing against friends to compare your scores
- The cube will fly over obstacles. If the cube hits the obtacle the game will end
- However the user has the ability to try again after hitting the obstacle.

Tech

We used Java.awt classes and methods for the foundation of the game. Abstract Classes such as ActionListener was extremely important for the cube to perform actions. The MouseListener abstract class was also very important so that the programme responds to the click of the mouse.

- We created two abstract interfaces classes. Jump and AddColumn
- The jump class was essential to allow our cube to jump
- The addColumn class was essential to add different columns / shapes to the environment

~~~
```
public void repaint(Graphics g) // We are repainting over here (abstract class for graphics - imported)
    {
        g.setColor(Color.blue); // we decided to go with blue
        g.fillRect(0, 0, WIDTH, HEIGHT);

        g.setColor(Color.red); // we decided to go with red
        g.fillRect(0, HEIGHT - 120, WIDTH, 120);

        g.setColor(Color.green); // we decided to go with green
        g.fillRect(0, HEIGHT - 120, WIDTH, 20);

        g.setColor(Color.yellow); // we decided to go with yellow
~~~

~~~~
        g.fillRect(cube.x, cube.y, cube.width, cube.height);
```
~~~~

This imported abstract class was important for settng the colours of our environment. The user can change the colours if they want. When you play the game you ill discover that the cube is yellow and the other colours are set accordingly

~~~~
```
public void actionPerformed(ActionEvent e) // abstract class
    {
        int velocity = 8;

        ticks++;
```
~~~~

The actionPerformance abstract class was used to set the speed of the cube. Again, this value is modifiable. For example if you were to change the velocity value to 2, the cube would move extremely slowly. If you were to change the velocity value to 40 for example, the cube will move extremely fast. This is a neat trick to change the difficulty of the game.

Contribution

- Jordan Ukawoko (worked on code, ReadMe, debugging and play-testing)
- Daniel Bodlund (worked on code, ReadMe, debugging and play-testing)
- Wimer Gunnarson (worked on code, ReadMe, debugging and play-testing)

This project was a group effort. We helped each other on knowledge we did not know.

Installation - To do in terminal

~~~~
```
cd src
java bouncingCube/BouncingCube
```
~~~~

References

- Abstract Classes & Methods tutorial - https://docs.oracle.com/javase/tutorial/java/IandI/abstract.html
- Jpanel tutorial - https://www.codejava.net/java-se/swing/jpanel-basic-tutorial-and-examples
- Java.awt imported libary tutorial - https://www.javatpoint.com/java-awt
- Java Swing tutorial - https://docs.oracle.com/javase/tutorial/uiswing/
- Snake game in Java tutorial (to get an idea of 2d game making) - https://zetcode.com/javagames/snake/
- Pong game in Java tutorial (to get an idea of 2d gaming making) - https://betterprogramming.pub/create-the-classic-ping-pong-game-3aea1f7b16e6