**fLOPPY Bird**

Group name: APOLLO

Group members: Sigodi Lutho , Ntsaluba Abona , Makgaloa Tshephang , Tantsi Okoyama Nomafaku

Group mentor: SINENJABULO GOGWANA

Project title: FLAPPY BIRD 2.0

Project page: <https://github.com/g24m3630/Epic-car>

Instructions for use (allowable actions): For Windows devices a player has to tab spacebar to make the bird flop its wing and fly upwards, however if the user stops taping the spacebar key the bird will fall and the game ends. And for mobile phone the player has to tap on the screen to make the bird flop its wings and fly upwards.

Tools used:

* “java.awt” – This contains the classes for creating user interface as well as painting graphics and images.

The classes used for this game are:

- color : manages the colours in the graphical interface.

-image: representation of the graphical images.

-font

-dimension: specifies the size of the images.

* “java.awt.event” – Provides the necessary clauses and interfaces for event handling.

The classes used:

-ActionEvent: indicates that a component – defined action has occurred.

- ActionListener: Adds interactivity in java applications by handling user actions.

- KeyListener: allows program to respond to user input from the users keyboard.

- KeyEvent: encapsulates information about the key event.

* “javax.swing” – provides lightweight components that are used to create window-based applications.

Classes used:

* JFrame: represents a top level window with a title border.
* JButton: implements events when clicked.
* JPanel: hold and organises other components.
* Timer: implements action events in intervals to assist in repetitive tasks.
* “java.util” – provides the collections framework, random number generations and other utility classes.

Classes used:

* Random: generates random numbers assisting in the random generation of the pipes in the game.
* ArrayList: stores the lists for Pipe objects

Concepts used:

* Variables
* Data types
* Classes and objects
* Recursion
* Files
* Methods
* Exception Handling
* Key events
* Inheritance
* Object-oriented programming
* Event-driven programming
* Graphical user interface
* Data Structures

Assumptions: [State ANY assumptions about how you expect the system will be used or run]

Limitations: The game does not include a save/load system or a way to store high scores or game progress between sessions. The program is developed for desktop systems and does not have mobile or tablet support.