CS 3330 Final Project Documentation
Greg Ryterski, gjr7dz
12/11/2020

This project was created using JavaFX. This document provides the location and examples of where the project meets the criteria outlined in the Final Project Document.

1. Object-Oriented Elements:
   a. Classes:
      i. FlappyBirdFinal.java
      ii. MainModel.java
      iii. AbstractModel.java
      iv. StartFlappyBird.java
      v. Game.java
      vi. Login.java
      vii. Score.java
      viii. Sound.java
      ix. Bird_Movement.java
      x. Pipe_Movement.java
   b. Sub-Classes:
      i. MainController.java
      ii. AboutController.java
   c. Abstract-Class:
      i. AbstractModel.java
   d. Interface:
      i. SwitchScene.java
2. Code elements Utilized:
   a. Collection Classes Utilized:
      i. MainModel.java
         1. Utilizes ArrayList<String> & ArrayList<Integer> for gaining the top 5 players for the leaderboard. <Lines 68-77>
      ii. Game.java
         1. Utilizes ArrayList<String> & ArrayList<Integer> for gaining the top 5 players for the leaderboard and storing their values to be later used <Lines 73-74>
   b. Exception Handling:
      i. FlappyBirdFinal.java
         1. Method: start()
            a. Try-Catch ultiized to catch IOException <Line 37>
      ii. MainModel.java
         1. Method: loadLeaderBoard()
            a. Throws clause utilized for FileNotFoundException, IOException <Line 38>

 iii. StartFlappyBird.java
   1. Method: start()
     a. Try-Catch utilized to catch ClassNotFoundException, InstantiationException, IllegalAccessException, or UnsupportedLookAndFeelException <Line 29>
 iv. MainController.java
   1. Method: switchToAbout()
     a. Try-Catch utilized to catch IOException <Line 94>
   2. Method: playGame()
     a. Throws clause IOException <Line 109>
 v. Game.java
   1. Method: startShock()
     a. Try-Catch utilized to catch Exception <Line 293>
 vi. Score.java
   1. Method: writeToFile()
     a. Try-Catch utilized to catch IOException <Line 238>
 vii. Sound.java
   1. Method: skip()
     a. Try-Catch utilized to catch FileNotFoundException <Line 35>
     b. Try-Catch utilized to catch Interrupted Exception, JavaLayerException, or FileNotFoundException <Line 42>
   2. Method: drop()
     a. Try-Catch utilized to catch FileNotFoundException <Line 65>
     b. Try-Catch utilized to catch Interrupted Exception, JavaLayerException, or FileNotFoundException <Line 72>
   3. Method: shock()
     a. Try-Catch utilized to catch FileNotFoundException <Line 94>
     b. Try-Catch utilized to catch Interrupted Exception, JavaLayerException, or FileNotFoundException <Line 101>
   4. Method: points()
     a. Try-Catch utilized to catch FileNotFoundException <Line 121>
     b. Try-Catch utilized to catch Interrupted Exception, JavaLayerException, or FileNotFoundException <Line 128>
 viii. Pipe_Movement.java
   1. Method: run()
     a. Try-Catch utilized to catch InterruptedException <Line 67>
 ix. Bird_Movement.java
   1. Method: run()

a. Try-Catch utilized to catch InterruptedException <Line 47>
            2. Method: jump1()
                        a. Try-Catch utilized to catch InterruptedException <Line 90>
            3. Method: jump2()
                        a. Try-Catch utilized to catch InterruptedException
                           <Line 119>
3. Clearly Defined Model:
    a. Model Classes Utilized:
        i. MainModel.java
4. Multiple Scenes with dynamic scene display:
    a. MainView.fxml
        i. Play & About buttons will visit different pages
        ii. New User button will send user to create a username otherwise defaulted
            to "Guest"
    b. AboutView.fxml
        i. Back button returns to the MainView.fxml
    c. The scene is constantly changing during the game located on a JPannel in
       Game.java
5. About Page:
    a. Click the 'About' button on the MainView.fxml will bring up the About Page.
       Provides brief information about myself, how to play the game, and the purpose
       behind the application.
6. Saving and Loading Data
    a. MainModel.java
        i. Method: loadLeaderBoard()
            1. Loading past users names and scores and separating the top 5
               users from the rest <Lines 39-49>
    b. Score.java
        i. Method: writeToFile()
            1. Saves current users name and score to the already existing
               'names.txt' file <Lines 231-237>