CSE 201 Advanced Programming (Monsoon Semester 2020)
Course Project
Color Switch Game

IIIT-Delhi. 21st October 2020. Due by 11:59pm on 6th Nov 2020 (1st Deadline)

IMPORTANT Instructions:

1. **It's mandatory that you attend all the deadlines in this project as per the schedule. No request for rescheduling the demo will be entertained. In case of any unavoidable circumstances you have to take email approval from me well in advance.**
2. **You MUST have a PRIVATE git repository for your project and every group member should frequently check in their code in this repository.**
3. **No extensions will ever be provided. Any submission after the deadline will not be evaluated. If you see any ambiguity or inconsistency in a question, please seek clarification from the teaching staff.**

Plagiarism: All submitted deliverables are expected to be the result of your individual effort. You should never misrepresent someone else's work as your own. In case any plagiarism case is detected, it will be dealt as per new plagiarism policy of IIITD that was also discussed in the lecture.

Create a JavaFX application for the game - **Color Switch**. The objective of the game is to collect as many stars as possible by manoeuvring the obstacles. If you hit any obstacle, you can continue with your earned stars or the game ends and you have to start from the beginning. This is a high score game. We will be developing only the endless (infinity) game mode of the classic gameplay. It is important that you play the game (*available on both Android and iOS*) and make yourself well-familiar with it before reading this document any further. A youtube video of the game can be found at this link.

Some of the rules for the game play are mentioned below. However, the exact set of rules you can find yourself after playing the game. In your implementation you should follow all those rules.

1) There are many modes and levels in this game. We are only concerned with **"Endless"** game mode denoted with an infinity symbol in the actual game's homescreen in which you are required to design the **"Classic"** gameplay alone.

2) The player has to use a single input to keep the ball afloat as well as move upward. A constant amount of distance is travelled each time the input is given.
3) The ball will fall if no input is given, as if it is moving against gravity.
4) The player has to collect the "Stars" placed along the gameplay. This will be counted towards the score, hence it should be provided at frequent intervals.
5) A minimum of 4 colors has to be used to create obstacles and color the game ball.
6) Make use of Color switches to change the ball color and make the game interesting. They need to be placed appropriately so that the game always progresses and there is no dead end.
7) As the game progresses, the game needs to become difficult. You could achieve this by creating different kinds of obstacles.
8) The Stars collected can be used to resurrect the ball in case the ball hits an obstacle. Choose an appropriate number of stars to use to let the player resume if she/he feels so or let them start afresh.
9) To save the state of game at any point in time, the game must store the following objects:
    a) Store the stars collected
    b) Store the exact position of the ball
    c) Store the position of the obstacles and their orientation
10) A player must be able to save as well as load any saved game.
11) Game must allow multiple save/load games.

Basic Requirements:

1) **Main page:**
    a) New game button
    b) Resume game button
    c) Exit game button
2) **Resume game button:** It should lead to a screen showing a list of saved games.
3) **In-game options**
    a) (on hitting an obstacle/losing) Restart game, continue to play (by using earned stars), exit to main menu
    b) During gameplay - a pause input, save the state of the game
4) GUI should be designed using JavaFX and should be similar to the gameplay video referred above.
5) Minimum number of colors (for ball and obstacles) used should be 4.
6) Show current score at an appropriate position.
7) Command line output will not be considered part of the game. GUI should be the sole interface for interaction.
8) It is not mandatory to have a fluid-like animation as shown in the game.

Bonus:

Although we have specified the basic requirements, if you are able to come up with some more interesting features then you would be "eligible" for bonus marks. Although this eligibility we will decide based on factors such as how many other groups have also come up with the same additional functionality, and how significant is that functionality.

Project Deliverables:

1. Deadline 1 (Due on 6th November) – Submit detailed UML class diagrams and use case diagrams for your project. Submit pictures of your diagrams on google classroom. We will use these pictures during the demo of this deadline.
2. Deadline 2 (Due on 20th November) – Show static GUI of your project and also some animations components. Submit this code on google classroom just like lab deadlines.
3. Deadline 3 (Due on 5th December) – Submit complete project on google classroom.