CSE 201 Advanced Programming (Monsoon Semester)
Course Project Option-1
Snake vs Block Game

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. **Each group should choose their project latest by 23:59pm on 08/10. Absolutely no changes are allowed after this deadline. Either of the two members in any group should ensure they choose the project in case another group member is unavailable until this deadline.**
3. **You MUST have a PRIVATE git repository for your project and every group member should very frequently check in their code in this repository.**
4. **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 a 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 - Snakes vs Block.The objective of this game is to create a chain of balls that has enough power to break through numbered blocks. If your chain hits a block that it can't break, you die.

Rules for the gameplay are mentioned below -

1) The snake is a collection of n balls arranged linearly. (n >= 0)
2) Snake can move in 3 directions: forward, left, and right.
3) The snake always moves forward until it encounters a block.
4) Blocks and Walls appear randomly during the gameplay.
5) Each block has a value, which represents the points made by the player when the snake eats it. After eating the block the length of the snake decreases by the value of the block
6) Blocks may occur independently, or in a chain. Atleast one of the block in a chain should have a value less than the length of the snake.
7) Walls may occur in different lengths, but should not divide the screen in two halves at any time. That is, length of the wall should be strictly less than the length of screen.
8) Snake should not be able to move across the wall.
9) A snake may encounter 4 types of tokens during the gameplay.
    a) Ball: A ball has a value written over it. The length of the snake increases by the value of the ball if it eats the ball.

b) Destroy Blocks: Destroy all the blocks present on the screen. Value of each block destroyed is added to the score of the player.
c) Shield: A shield lets the player eat any block without decreasing the snake's length. Again, the value of the block eaten by snake is added to the total score of the player. The shield remains with snake for 5 seconds only.
d) Magnet: A magnet allows the snake to collect coins which are within a certain distance from the head of the snake. The distance should be chosen by the programmer but should not equal the width of the screen.

10) The frequency of ball is greater than the frequency of other 3 tokens.
11) Speed of snake in forward direction increases as its length increases
12) The game ends when the snake is not able to eat a block completely.

Note:

1) The game is endless and only ends when snake dies.
2) The frequency of tokens, bricks, walls is open to programmer's implementation, but should make the game challenging for the player.

You can watch the video of the game by clicking here. A similar game is available on playstore. (Gameplay/GUI of both the games mentioned above is a bit different. Try to make a game similar to that in the video.)

Basic Requirements:

1) **Main Page:**
   a) Start game button
   b) Resume game button (only occurs when the last played game wasn't completed.)
   c) Leaderboard button
2) **Leaderboard Page:** Top 10 scores of the player with date.
3) **During Gameplay**:
   a) A GUI similar to the game in video
   b) A drop down with two options
      i) Start the game again
      ii) Exit and go to Main Page
   c) Current score of the player should be shown at the corner of the screen.
4) **Animation:**
   a) You are not required to have a fluid like movement of snake in left or right direction as shown in the video. A smooth switch in either direction should be made. While moving left or right, the entire snake can move together.
   b) Whenever a brick is eaten/destroyed, a token is collected, a burst similar to that in the video should be made.
   c) A brick with value less than equal to five is instantly eaten by the snake. For bricks with value greater than 5, the snake takes time to eat it and is not able to move forward as shown in the video.

5) **Avatars:** The look of bricks, balls, snakes should be similar to that shown in the video. The bricks should occur in different colours. The look of 3 tokens (shield, magnet, destroy blocks) can be chosen by the programmer but should be easily identifiable.
6) Save the state of the game at any point of time.
7) Also save the state in abnormal conditions (user presses cross, presses home button (if any), etc.)
8) Generate very detailed documentation for each and every class/method/fields of your project by using the JavaDoc tool. You can get information on what/how to use JavaDoc on google.
9) At the end of the game, user should be taken to the main screen and the score of last game should be shown on the screen.

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 group has also come up with same additional functionality.

Project Deliverables:
1. Deadline 1 (Due on 17th October) – Submit detailed UML class diagrams and use case diagrams for your project. Submit pictures of your diagrams on backpack. We will use these pictures during demo of this deadline.
2. Deadline 2 (Due on 02nd November) – Show static GUI of your project and also some animations components. Submit this code on backpack just like lab deadlines.
3. Deadline 3 (Due on 17th November) – Submit complete project on backpack. Demo after end semester exams.