CS 329E: Bulko
Programming Assignment 9
Moving Block

1 Problem Definition

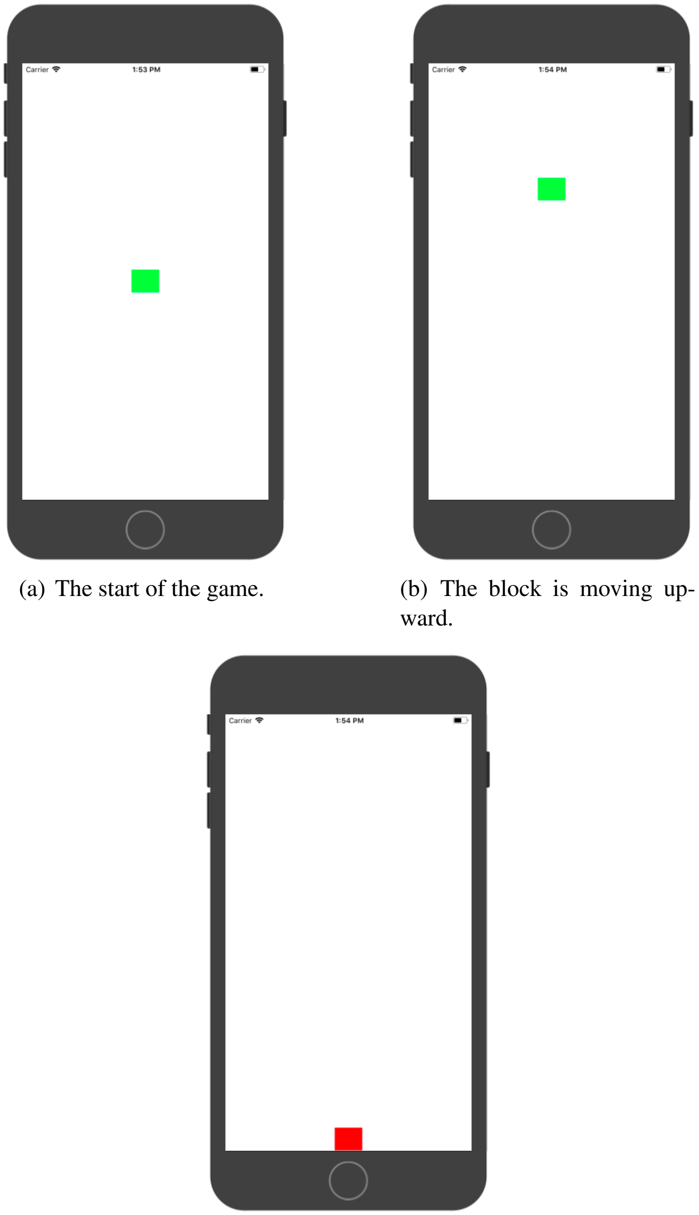
In this assignment, you are going to play around with gestures. There is only one Main VC in this project, as shown in Fig. 1(a). You will use a view object to represent a block which can move up, down, left, and right. Tapping the block will place it in the center of the screen and start the block moving in the down direction. Then iOS gestures will be used to change the direction of the block's movement. Initially, the color of the block is green. If the block attempts to move off the edge of the screen, it will stop and change its color to red, as shown in Fig. 1(c).

The block will move around in a "grid world". Imagine that the screen has been mathematically divided into 9 columns and 19 rows of rectangles. (The width and height of those rectangles in pixels will depend on the size of the device's screen.) Set the size of the view object representing the block so that it exactly fits in one of these rectangles. Those rectangles represent the 171 possible locations where the block could be.

Note: in this assignment, "screen" refers to the **safe area** of the screen. This will ensure your block is not obscured by the presence of a notch at the top of the simulator or phone.

2 Detailed Instructions

- Create a Single View application project named <lastName><firstName>-HW9.
- Programmatically create a view as the block. Set the color to be green, and set the size to be (1/9 the width of the screen) by (1/19 the height of the screen). Place the view in the center of the screen (i.e., the middle column and the middle row).
- Add 1 Tap Gesture Recognizer and 4 Swipe Gesture Recognizers. The 4 swift gesture recognizers will correspond to the 4 different directions up, down, left, and right.
  - When the Tap Gesture Recognizer is triggered, the block will be put in the center of the grid world and start moving downward one step for every 0.3 seconds. The step size should match the height of the grid cell.
  - When one of the Swipe Gesture Recognizers is triggered, the direction of the block's motion will change according to the direction of the swipe. For example, if the direction of the swipe is left, the block should start moving to the left one block width per tick.
  - The block will keep moving until it hits the boundary of the grid world. When the block hits the boundary of the grid world (i.e, the leading edge of the block lines up with the edge of the screen), it will stop and change its color to red. At this point, the game is over, and swiping no longer works.



(a) The start of the game.

(b) The block is moving upward.

(c) When the block hits the wall

Figure 1: Application demos

  - Tapping the block will then move the block back to the center, change the color back to green, and restart the block moving.

- Movement of the block should be a jump of one block width (left or right) or one block height (up or down). It should not be a smooth slide.

3 Grading criteria

1. The block is created programmatically. (10%)
2. The Tap Gesture Recognizer correctly initializes/reinitializes the position of the block and starts/restarts the movement in the down direction. (20%)
3. The block moves step by step with a time interval of 0.3 seconds. (30%)
4. The 4 Swipe Gesture Recognizers correctly change the direction of the block's movement. (20%)
5. When the block hits the edge of the screen, it stops and changes color to red. (20%)
6. **Note that if the app does not build and run, ZERO points will be given.**
7. The Coding Standard is followed. One point deducted for each violation.

4 General criteria

1. I will be looking for good documentation, descriptive variable names, clean logical structure, and adherence to all coding conventions expected of an experienced programmer, as well as those outlined in the Coding Standard document. There will be penalties for failure to meet these standards.
2. Your code must compile and run before submission.
3. Xcode will automatically generate standard headers to your .swift files. Add two lines to each Swift file so that the header includes the following:

   ```
   // Project: LastnameFirstname-HW9
   // EID: xxxxxx
   // Course: CS329E
   ```