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 are going to create a grid world and a block (a label or view, actually) which can move up, down, left and right in the grid world. A tap gesture is used to put the block in the center of the grid world and start the block moving in the down direction. Then iOS gestures are 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).

2 Detailed Instructions

- Create a Single View application project named <lastName><firstName>-HW9.
- Divide the screen into 9 columns and 19 rows to define a “grid world” of $9 \times 19$ cells.
- Programmatically create a label or view as the block. Set the color to be green, and set the size to match the size of one grid cell.
- Add 1 Tap Gesture Recognizer and 4 Swipe Gesture Recognizers to the VC. 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 size of the grid cell. Use multithreading to implement this.
  - 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.
  - The block will keep moving until it hits the boundary of the grid world. When the block hits the boundary of the grid world (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.
  - The tap gesture can then be used to put the block back to the center, change the color back to green and restart the moving process.

3 Grading criteria

1. The block is created programmatically. (10%)
Figure 1: Application demos

(a) The start of the game.
(b) The block is moving upward.
(c) When the block hits the wall.
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 movement process is implemented by multithreading, such that 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 that list your EID and the course number, so that the header looks like the following:

```
//
// Filename: LastnameFirstname-HW9
// EID: xxxxxx
// Course: CS371L
//
// Created by xxxxxx on x/xx/20.
// Copyright 2020 xxxxxx. All rights reserved.
//
```