

# CS M152A Lab 4 Proposal

Michael Wu, Haoyu Yun, Jennie Zheng

## Objective

We will attempt to design a version of the game “Flappy Bird” on the FPGA board. This game consists of a character moving horizontally on a 2D plane through a series of columns with gaps in them. After passing through each column, the player will increase their score by one. The character undergoes constant downward acceleration that mimics gravity. By pressing a button, the character can “flap”, which sets their upward velocity to a positive value. In this manner the player can maneuver the character through the columns. If the character hits a column or the ground, this will end the game.

## Design Description

We will use the VGA output to display the locations of the character and the columns. The character will have a fixed horizontal position on the left side of the screen, while the columns move left in order to simulate the movement of the character right.

The gaps in the columns will be randomly generated in order to add variety to the game. We will keep track of where the columns are in order to detect whether the game has ended because the character has hit something. The game also ends if the character hits the bottom of the screen.

The character maintains a velocity variable, which has a fixed maximum magnitude. We will have one input button that will set the character’s velocity to a positive value when it is pressed. Another button will be used to reset the game.

We will show the player’s score on the seven segment display. The score will correspond to the number of columns the player can pass through without hitting anything.

As this is a real-time game, this will require clocks of varying frequencies to update the graphics, calculate position, and detect button presses.

## Grading Rubric

Flapping Button and Physics (25%) - When the player presses the flap button, the character will move up and down correctly. Also the character should undergo constant downwards acceleration, and not fall off the top or bottom of the screen. If the character hits the top his velocity should be set to zero.

Reset Button (10%) - When the reset button is pressed the game should reset to its initial state.

Column Movement (15%) - The columns should move towards the player and there should be infinite columns. A maximum of two columns will be visible at a time.

Random Column Generation (15%) - The columns should have a random height for the gap that is continuously generated.

Hit Detection (20%) - The game should end when the character hits either the ground or a column. The screen should then pause and no more movement should happen. The reset button will be required to start a new game.

Score display(15%) - The seven segment display should show how many columns the character is able to get through.