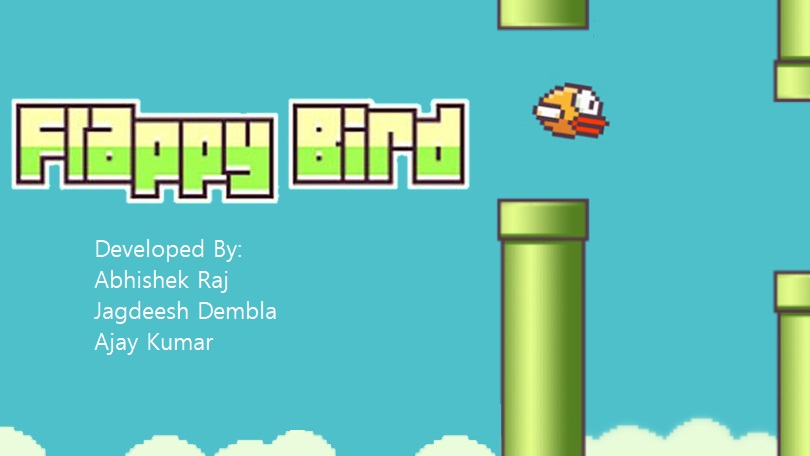
Second Semester OOP Java Project

**FLAPPY BIRD**

****

BY:

Abhishek Kumar

Jagdeesh Dembla

Ajay Kumar

**Overview**

In this project, we design and implement a Flappy Bird like video game on the SoCKit development board. Flappy Bird is a very popular mobile game on Android platform, driving a lot of people crazy. In this game, the player can control the vertical movement of bird ( every pressing on the keyboard makes the bird leap upward for a little bit, and the bird will fall freely without control ). As soon as the game begins, tubes will keep appearing from the right side of the screen and moving leftwards. (so that it seems like the bird flying forward). The goal of this game is to control the bird, dodging and passing the incoming tubes, as many as possible. The game is endless until the bird eventually hit one of the tubes, ground, or ceiling. Figure 1 is the start screen of Flappy Bird. The title "Flappy Bird" is shown in the middle of the uppers side of the screen. The bird is also displayed on the background.

****

**High Level Design**

Primary components that constitutes our game includes the ARM core (game logic), device driver, USB controller to control the input from keyboard, Sprite controller (control the display of sprites) , SDRAM ( store all the data needed in game logic). The game logic module interfaces with several other modules including the USB keyboard, by receiving the control signal; as well as the device driver in order to control the display of sprites, including the positions of pillars and birds, the length of the pillars and the score. Sprite controller is connected to VGA Controller, which is responsible for the display of all the images.

**Game Logic Controller**

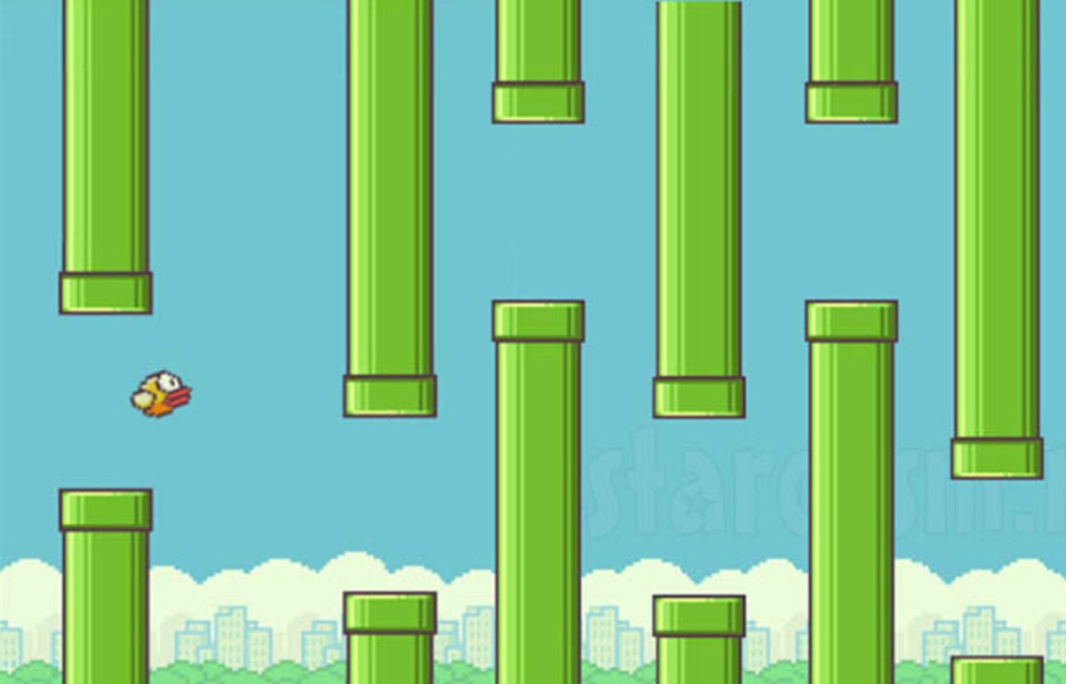
We implement the game logic by using Java programming language. The game logic controller should realize the functions which are indicated below: updating the location of the bird from the keyboard, implementing the game rule (whether the game is over or not, computing how many pillars the bird has passed), and controlling the generation of sprites. Based on the functions given above, there should be 3 submodules for the game logic controller.

1. Game rules: This is the core submodule of the game logic controller which interfaces with all of the other submodules, instructing them what to do based on the game rules. The rules are implemented by the updated position of bird from the keyboard, and the current position of the pillars. Appropriate audio is chosen corresponding to the rules (whether the game is over or not).

2. Sprite generator: 1. Pillars: This submodule keeps updating the X coordinates of the pillars that has already appeared on the screen (by decrementing them in every cycle), as well as the length of the upcoming pillar that is going to appear from the right side of the screen (which is actually the number of "partial" pillars that stack). The length of the pillar should be random, as long as the distance between the pillars is constant. Once the sprite moves out of the screen (in this case, x coordinate of any one of the pillars becomes zero), we reset the coordinate so that it can reappear from the right side of the screen. 2. Bird: Bird acts like in real world that its jump and fall will be affect by the gravity. When 5 we implement the object motion formula in our code, time calculation is an issue that we use a counter counting instead of using system clock. We put the delay in our loop and try a suitable count number being our time unit. In addition, we add a status variable to indicate if the bird status is rising or falling. It cooperates with our jumping and falling function with iteration loop supporting continuous jumping without multi-thread.

3. Score: Every time the bird passes one of the pillars, the "Game Rules" submodule sends a signal, which will make the score increment by 1. Since the sprite for displaying the score are separated into 3 parts, hundreds, tens and digits, we need to extract them from the score before sending them to the hardware.

4. Title: The display of the title "Flappy Bird" depends on whether the game starts. When the game is over, press "enter" to restart, and the title would be displayed instantly. Since each signal sent from software to hardware has 8 bits, we only use one of them as the control signal to display the title, so that we can use other bits for other purposes, which improve efficiency.



**Contribution**

* Abhishek Kumar From Computer Science.
* Jagdeesh Kumar From Computer Science.
* Ajay Kumar From Computer Science.