



BAHRIA UNIVERSITY KARACHI CAMPUS

Department of Software Engineering

COURSE: CEL 220 COMPUTER ARCHITECTURE & LOGIC DESIGN PROJECT PROPOSAL

CLASS: BSE – 3B (FALL - 2023)

Flappy Bird

Group Members

S. No.	Name	Enrollment #
01	RAJA MUHAMMAD HAMMAD (Team lead)	02-131222-039
02	ABDUL AHAD KHAN	02-131222-101
03	ABDULLAH	02-131222-099

Submitted to:

Course Instructor: Engr. Ramsha Mashood

Lab Instructor: Engr. Noor us Sabah

Date: 04/12/2023

1. INTRODUCTION & BACKGROUND

Flappy Bird is an arcade-style game in which the player controls the bird, which moves persistently to the right. The player is tasked with navigating the bird through pairs of pipes that have equally sized gaps placed at random heights. It automatically descends and only ascends when the player taps the 'f' button on the keyboard.

2. PROBLEM STATEMENT:

Creating a version of the Flappy Bird game in MIPS assembly language with bitmap graphics poses a unique set of challenges. The game involves controlling a bird that moves continuously to the right, navigating through pairs of pipes with randomly placed gaps. The bird automatically descends and ascends only when the player taps the 'f' button on the keyboard. The challenge lies in implementing game logic, user input handling, and bitmap graphics representation within the constraints of the MIPS architecture.

3. PROPOSED SOLUTION:

The proposed solution is to develop a Flappy Bird game in MIPS assembly language using bitmap graphics. This involves designing the game's architecture, implementing game logic for bird movement and pipe generation, handling user input, and integrating bitmap graphics for visual representation. The solution aims to provide an engaging and functional version of Flappy Bird while adhering to the limitations of the MIPS architecture and bitmap graphics.

3.1. FEATURES OF THE PROJECT

In our game, the image moving is the most important function we need to implement. Besides the background horizontal moving, the vertical jumping combining keyboard input is our second challenge. The bird briefly flaps upward each time the player taps the 'f' button; if the button is not tapped, the bird falls due to gravity.

3.2. METHODOLOGY:

Game Logic Design:

Define the game rules, including bird movement, pipe generation, and user input handling.

Implementation for continuous rightward movement, automatic descent, and ascension upon the 'f' button tap.

Bitmap Graphics Representation:

Explore MIPS assembly language capabilities for handling bitmap graphics.

Design and create bitmap images for the bird, pipes, and other relevant game elements.

Develop functions to load, display, and update bitmap graphics on the screen.

User Input Handling:

Establish mechanisms for detecting 'f' button taps and responding to user input.

Ensure accurate and responsive control of the bird's ascension based on user actions.

Collision Detection:

Implement collision detection to identify interactions between the bird and pipes.

Define outcomes of collisions, such as game over scenarios.

Game Loop and Flow:

Construct a game loop to iterate through key steps in continuous gameplay.

Manage the flow of the game, including start, pause, and restart functionalities.

3.3. TECHNOLOGIES TO BE USED

Mars Simulator was employed in our project. Our project was created using the MIPS Assembly language bitmap.

4. PROJECT SCOPE

It's an easy game with endless levels. This game is playable by players of nearly age above 5 years. They can have some enjoyment from it throughout their free time. Additionally, it is a simple game that doesn't require complicated instructions to play. One can play this game for hours on end because it is so addictive. This game can be played on an Android device.

5. PROJECT ABSTRACT

In the arcade-style game Flappy Bird, the player controls a bird that keeps moving to the right. The objective is to guide the bird through pairs of pipes with randomly positioned, equal-sized gaps. It moves downward automatically and only rises when the player presses the keyboard's "f" button.

6. MODULE DISTRIBUTION

➤ **RAJA MUHAMMAD HAMMAD**

Implementation of pipe, coding of top refresh and bot refresh, Coding of bird, using loop.

➤ **ABDUL AHAD KHAN**

Designing of bitmap display, coding of reset of game and exit screen.

➤ **ABDULLAH**

Coding of jump Method in the game

7. REFERENCES:

Patterson, D. A., & Hennessy, J. L. (2017). Computer Organization and Design: The Hardware/Software Interface (5th ed.). Morgan Kaufmann.

MIPS Assembly Language Programming.

[Link For Reference](#)

MIPS Assembly Language Programming which we learn in the Lab.

Teacher's Signatures: _____

Remarks: _____
