Top Gun

Team Mavericks: Eashan Sapre [es4069], Kuraloviyan Senthilnathan [ks4065], Aparna Muraleekrishnan [am5964]

1. INTRODUCTION

We attempt to design and develop a game titled Top Gun. This is a side-scroller game where the player controls a fighter jet, attempting to avoid mountains and missiles. If the player's aircraft touches any of the obstacles on screen, the game ends. The fighter jet ascends each time the player presses a key, else it descends. The player earns points based on the distance covered by the jet. A high score is maintained in memory and displayed during all game play.

1.1 GAME RULES

- The game starts when the "Start" button is pressed, the game can be paused by another button press.
- The player controls the jet via keyboard input, pressing the space bar to cause the jet to ascend vertically. The player has no control over the horizontal coordinates of the aircraft.
- Points can be scored by successfully maneuvering around obstacles.
- Crashing into obstacles causes the game to end. High score is updated if the player beats the previous record.
- There will be three difficulty levels, with more missiles/mountains appearing on the screen as points increase above set thresholds. The screen scroll speed also increases with increasing difficulty.

2. GAME CONTROL

2.1 Game Logic

This is the core submodule of the game logic which interfaces with all of the other submodules, instructing them what to do based on the game rules.The game should constantly update the screen by supplying the graphic generator with the location of the fighter jet, the number of mountains and frequency of missiles generated based on the score, as well as the judgment on whether the game is over. This module also updates the difficulty level and decides the scroll speed and obstacle frequency accordingly.

2.2 Tracker

The function keeps track of the movement of the jet by the previous location, instructions from the keyboard and also a sub function that controls the automatic descent of the aircraft. Once the space bar is pressed, the jet should ascend with a vertical upward speed. When the space bar is not pressed, the aircraft descends due to the gravitational pull. We only care about the vertical location of the aircraft.

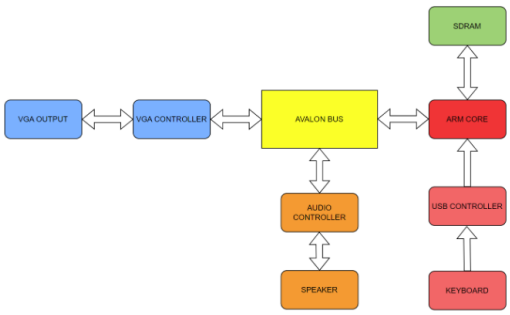

Figure 1: Hardware Design Components

2.3 Judge

This function calculates the real-time score. The function will compare the X/Y coordinates of the fighter jet with that of all obstacles. The speed of the obstacles remain constant and we use a counter to calculate the score. As long as the game is not over, the counter increments at the positive edge of a clock cycle depending on the distance between adjacent obstacles, and the speed of screen scroll. The score appears on the top right of the screen. This function also determines whether the game is over or not. If the coordinates of the jet overlaps with any obstacle, the counter stops. If so, the jet blasts(graphically) and the sub module will send a message to the graphic controller, to generate the "Game over" screen with the final score and a "Restart" button.

2.4 Graphics generator

This sub module receives all data required for the generation of graphics, including the coordinates of the jet and the mountains and missiles and the signal whether the game is over or not. These coordinates are stored in memory and updated according to the game logic. The graphic controller uses addressing to access memory-mapped data and then displays the necessary graphics components on the screen.

2.5 Audio generator

The audio sounds needed in the game, including the background music, the Top Gun Anthem music, as well as the

sound when the jet crashes, are encoded inside the audio generator. This sub module should instruct the audio controller which one to play, based on the game logic.

2.6 Milestones

The following are the milestones for this project:

1. Implement hardware ports, controllers and drivers.
2. Implement graphics display with jet and obstacles and test game screen with input control.
3. Implement game logic at low speed. Implement start/-pause/restart buttons.
4. Add difficulty levels and integrate audio output.