

Airplane Battle Game Embedded System Project Proposal

Zhaomeng Wang (zw2801), Jakob Stiens (jts2176), Yongmao Luo (yl4893), Tristan Saidi (tls2160)

GroupName - Water-Raid-Project

1. Project Overview

For our project, we want to achieve a third-person airplane battle game based loosely off of River Raid. When playing the game, the player should control the position of the plane on the screen such as being able to move it left or right. Additionally, the plane will be able to shoot targets in front of it and will need to avoid colliding with obstacles. The score will increase when the player hits the right target and based on the amount of time the player has been alive. The game world will consist of a scrolling world where rows are generated at the top and then are moved down at a consistent rate to generate the appearance that the world below is moving. Movement of the plane will be independent of the movement of the rows for the generated world.

2. System Description

The system can be divided into two parts, software and hardware. For the hardware part, we will use the SoC board as the control unit, SD card to store and load any static game files like the scenario map and a screen monitor to show the game video. We will also develop the VGA interface in hardware using SystemVerilog, and if more RAM memory is needed, SRAM can be developed in SystemVerilog as well. A keyboard as the system will be used as input, and if time permits a speaker to play audio (optional).

On the software side, based on Linux, We use C to develop several drivers to implement the input and output for the keyboard, video, and audio. We are looking to use protocols like VGA for the video. We will also be using C to program the game logic.

3. Main Potential Challenges

As far as we can see, we think there are three main potential challenges we may encounter in our project:

1. Figuring out if the RAM capacity is large enough to run the game, if not, we will need to consider whether or not loading the file from SD card will cause high latency (We currently believe that the RAM will be large enough for the game files, and will only need the SD card for files that will be loaded into RAM once the game is launched.).
2. We are unsure of how the FPGA board will behave while trying to handle large amounts of matrix manipulations especially in real time. We don't know where the FPGA limitations are in this area.
3. Possibly, it may be very difficult for us to implement the user mode/kernel mode switching program (driver) under the operating system.

4. We predict that the scrolling map mechanism may be difficult to generate and store in memory.

4. Similar Game Images

Images of games we would base our game around:



5. Milestone Goals

Milestone 1:

On the hardware side, we hope to finish the VGA implementation in SystemVerilog. On the software side, we hope to finish initial game logic.

Milestone 2:

By this milestone we wish to finish peripheral drivers for the keyboard, audio and monitor. We also want to begin displaying initial graphics on VGA monitor.

Milestone 3:

Stitching the game together, adding audio logic and finishing up graphics and mechanics.