

ECE 385

Fall 2022

Lab Report

Final Project Proposal

Jiabao Shen (jiabaos3)

Chuanrui Chen (cc86)

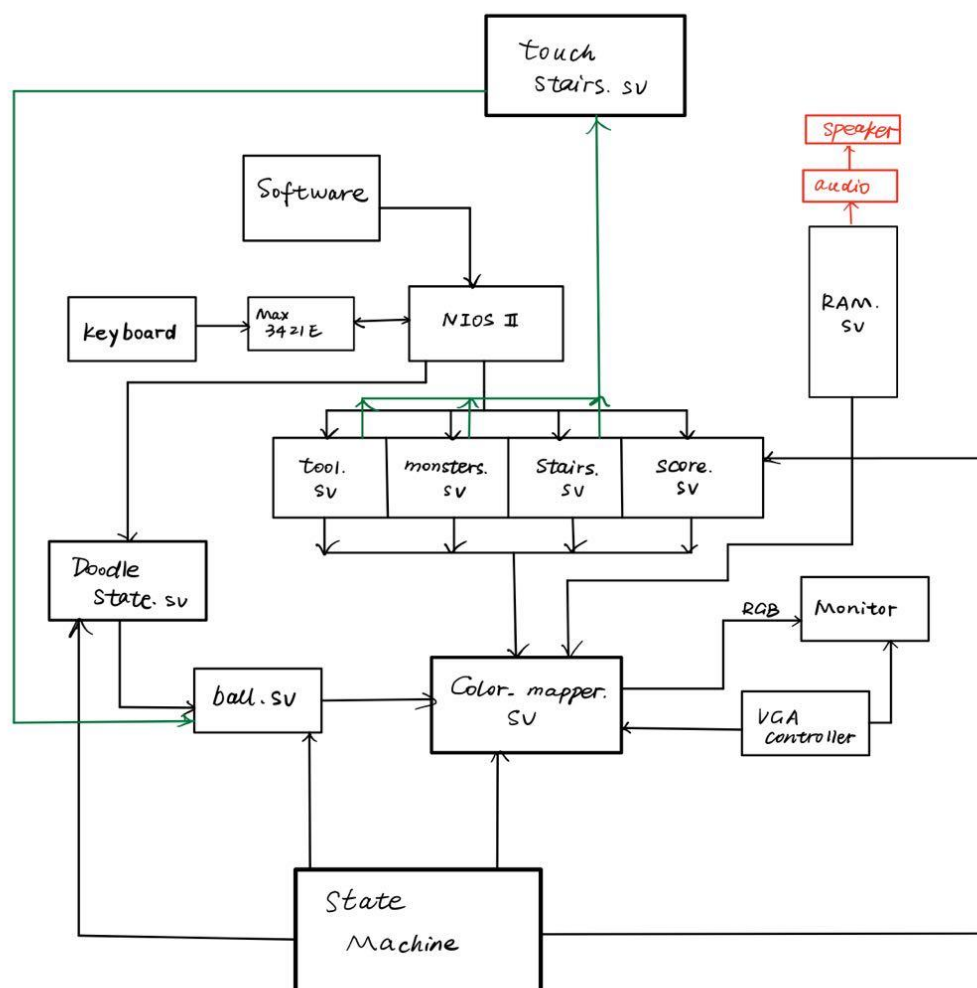
TA: Hongshuo Zhang

1. Idea and Overview

We decide to build the game “Doodle Jump”. Doodle Jump is a game with one character Doodle, who jumps to higher stairs to get scores. Every stair has some horizontal and tangential distance with each other, the player should use “A” or “D” keyboard to control Doodle and let it jump to a new stair. If it fails, game is over. There are also springs, monsters and moving stairs in the game. The springs can let Doodle jumps more stairs at one time. Monsters can capture the Doodle if they meet, and game is over. Doodle can shoot bullets in three directions using three keys on keyboard to kill monster or jump on monster to kill it. The game has no end if Doodle does not fail, and the accumulative score will show on the screen. There are also beginning page and ending page to design.

We will use color mapper, VGA, FPGA, SRAM, keyboard, Avalon bus, USB interface and so on to achieve our design function. We will use NIOS II CPU that is similar to what we did on USB interface in Lab6.2 to allow user uses the keyboard as input and controls the motion of Doodle.

2. Block Diagram



3. List of Features

3.1 Baseline Feature

There will be a start page to remind player to press enter key to start the game and there will be a ending page including the scores. The Doodle can move left or right when jumping and can touch a stair and stay on it. The Doodle can pass from left boundary to right boundary. The Doodle can shoot or stay on the head of monster to kill the monster, if not, the Doodle will die. The Doodle can use spring on some of stairs to pass more stairs at one jump. There are moving stairs so that Doodle is difficult to jump. Moving is controlled by A and D on keyboard and shooting is controlled by three arrow key on keyboard. The score is accumulating on the screen when playing.

3.2 Additional Features

We will add stairs which will be broken as soon as Doodle arrives, and Doodle will fail to increase the difficulty. Besides, we will add music every time Doodle is jumping and failing if the baseline features work well.

4. Expected Difficulty and Justification

The difficulty level of this project is about 5-6 points. Our goal is to implement the basic functions of the original game Doodle Jump, including jumping, obstacles (monsters, breaking boards, etc.), shooting, additional tools (springs, jet packs, etc.). Unlike the original game, we will use the keyboard to control the movement of the character and the shooting of bullets. The challenges are controlling the movement of characters and backgrounds, designing responses to different obstacles, and how the hardware and software interacted.

Based on the ball movement in Lab 6.2, we need to extend the modification of sv and C files and apply them to the map background, the jumping of figures and the firing of bullets. In order to be close to the original game, we will use physical models such as the gravity model for the movement of the characters, which will increase the difficulty of the project. We will also make use of the design in Lab 7 to score. These small features are more complex than games like Full Featured Breakout, Tetris, Missile Commands, but easier to implement than rhythm games, so we consider this project to be of medium difficulty. In addition, we are planning to create sound effects for the game. Besides the background music, there are also detailed sounds such as jumping and firing bullets, which we think will increase the difficulty of our project by about 0.5-1 point.

5. Proposed Timeline

Week 1 (11/7): Search and understand the requirement for the game. Have a basic structure of the project. Based on Lab 6.2 and Lab 7, define the PIO blocks for the Qsys.
Week 2 (11/14): Build the base of the game. Design the state machine and implement the basic functions of the game.

Checkpoint (11/18): Realize the basic functions of jumping, shooting, moving, scoring on a still background.

Week 3 (11/28): Finish all the basic design of the game. Try some additional functions like background music if we still have time.

Week 4 (12/5): Make sure that all features work well. Fix bugs if any. Prepare for the demo and the final report.