6.111 Project Proposal

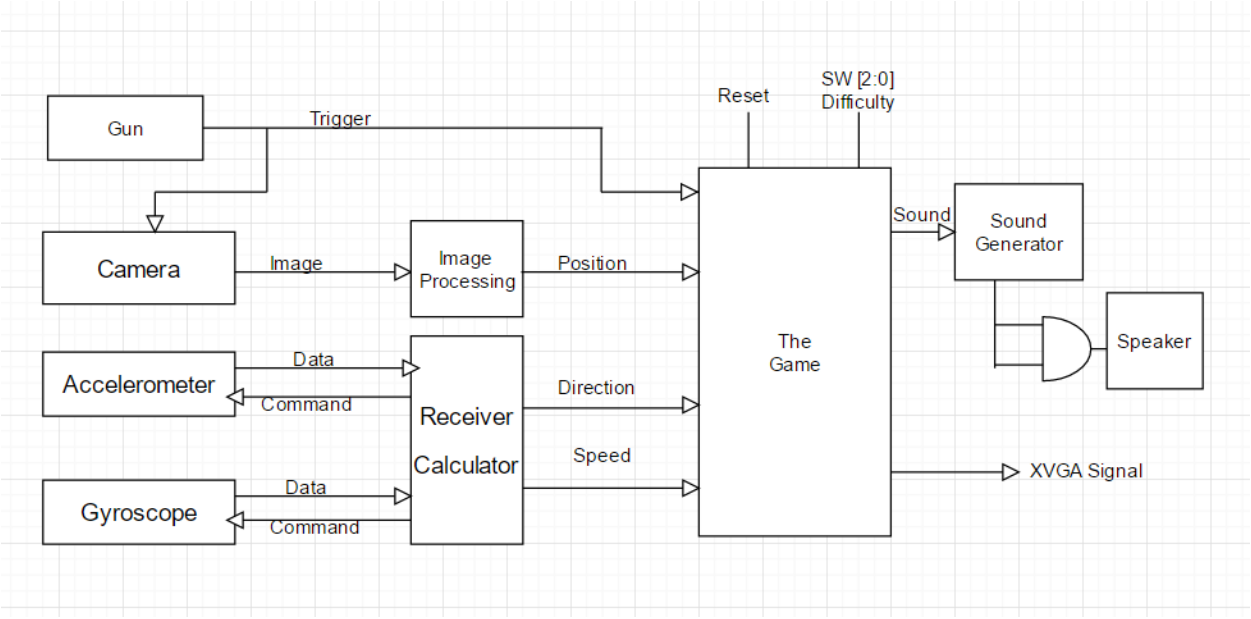
Space Invader with a Twist

Overview:

The goal of our project is to create a classic arcade game in the spirit of space invaders. In the game we will control a single spaceship which will battle against enemies by shooting at them. In order to make the game more interesting and challenging we will be implementing a shooting mechanism with camera-captured laser-pointing. The control of the ship will be handled making use of a gyroscope and accelerometer. The game will initially be a single stage which will increase in difficulty with increasing score and time. We plan on implementing several different enemy types using sprites.

Design:

High Level Block Diagram :



Module Description:

Accelerometer/ Gyroscope:

These two components will be sourced in the form of a LSM9DS1. Which includes both of them inside one package. To communicate with the module we will use I^2C protocol. This module will be mounted onto a control peripheral that will be used by the user to control the movement of the ship.

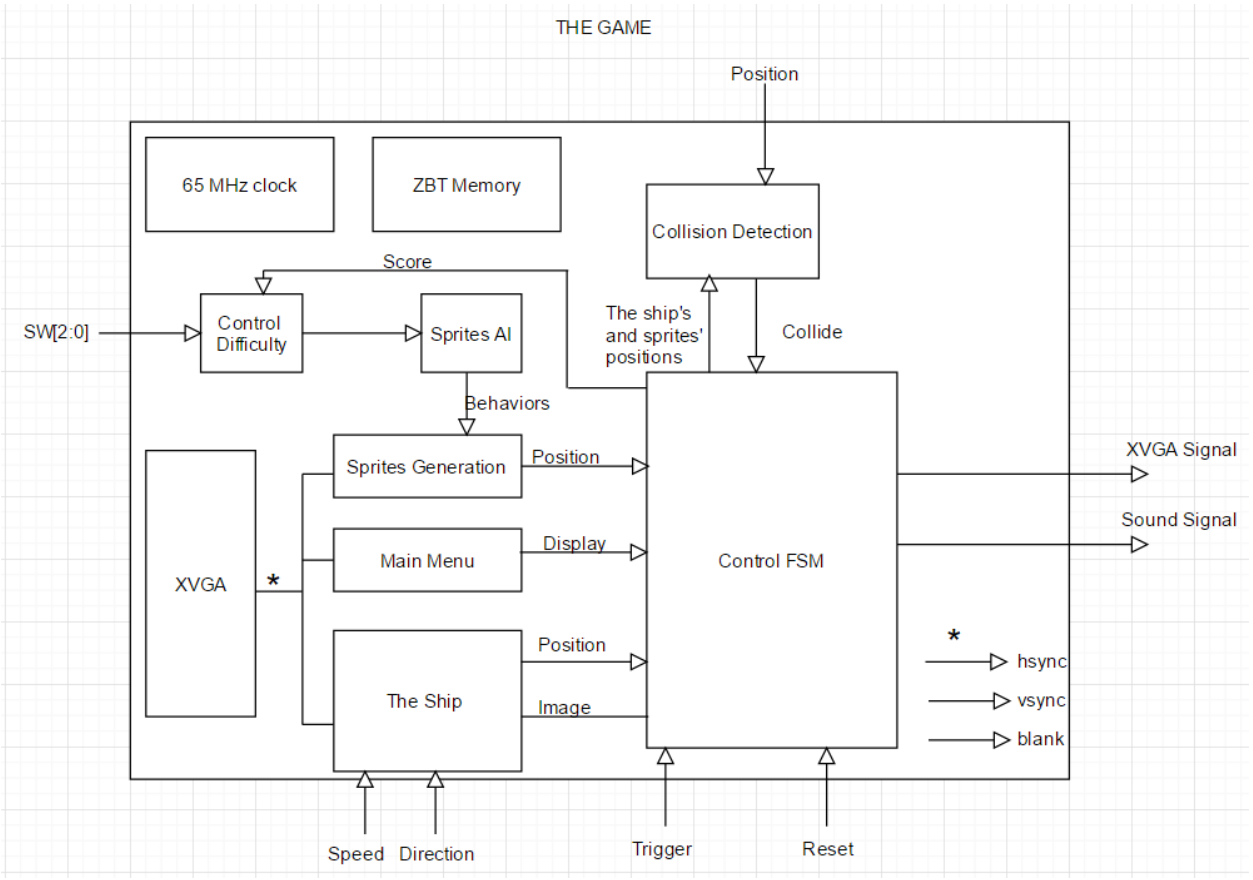
Receiver Calculator:
This module will recieve the signals coming out of the LSM9DS1 and convert them into a direction and speed command to be inputted into the game module. Since we are creating a 2D game, we will only use the x-axis and y-axis of the IMU chip. The signal coming from the chip is already digitized and has start and stop bit with acknowledgement bit between data. To determine the direction of the moving ship we would use the gyroscope and to determine the speed of the ship we will use the accelerometer. The gyroscope will determine the rotation velocity which we can then determine the direction the ship is changing (i.e a positive x-rotation will be correspond to the negative y direction). The velocity in each direction can be extracted by integrating the accelerometer data.

Camera/Gun:
The “gun” will be a laser pointer coupled with a camera to determine the location of the shot. The trigger on the gun fires the laser onto the screen and signals the camera to capture an image.By doing this we will eliminate the constraint of having a single enemy like the original Duck Hunt game and the targeting system will be more robust compare to the original idea of having a phototransistor.

Image Processing:
This module is in charge of taking the raw image from the camera and figuring out the relative position of the laser dot on the screen. This position will be transmitted to the game FSM which will then compare it to the position of the enemies and calculate the collision.

Sound Generator:
This will be implemented in a similar fashion to the alarm in lab four. The module would receive a signal coming in from the game which would indicate a shot fired, shot hit, death, reset, spawn, etc. The input signal will be wide enough to encompass all the effect wanted. The individual sounds will be initially simple. Additional more complex music and effects will be added later making use of the ZBT memory, which would be instantiated inside of the module.

“The Game” module:



Internal Modules:

Control FSM:

The control FSM is in charge of analyzing all of the incoming data from the user (direction,speed,trigger, shot position) as well as internal calculations (collision, difficulty, AI, etc) and transition between states correctly to run the game. It should have at the very minimum the following states:

> *Reset*: game in main menu, triggered by making use of the gun. Later instances of the design may implement more features in the menu which will take other input commands.
>
> *Dead*: Player collision with enemy or projectile. Will transition to reset state upon the assertion of reset signal. This state should clearly display to the player their failure.

I*n-game*: Most complex state in which we take in the user inputs for the ship control and calculate output control signals for the individual on screen components (ship, enemy, projectiles, etc) These signals are dependent on the difficulty setting, AI module, ship collision with enemy. The game transitions to dead state upon a ship collision with enemy or projectile. Otherwise it may transition to Shot Fired upon the user's input trigger. This state is also in charge of displaying the score of the player.

Shot Fired: The game receives a trigger signal from the laser gun and actuates the camera to take an image and send the position over. This position is compared to the enemies and either results in the defeat of the enemy or no hit. In any case the game returns to an updated In-game state.

In general the FSM is in charge of conveying the on screen modules, information regarding their location on screen and the direction/speed so that they can update them. Their video signal is then correctly generated and outputted to the monitor. When it comes to switching between the game, menu, and dead state we should just use a muxing of the video signals like in lab 3.

Clock Module:
This module in in charge of creating the master clock and any division clocks that may be needed to ensure correct operation. It will likely make use of the divider modules we have used in previous labs.

Control Difficulty:
The module will mainly be controlled by three switches. It will determine the speed and the behaviors of the enemies and the number of enemies on screen.

XVGA Module:
Initializes the XVGA video signals which is then fed to the individual on screen blocks to be modified.The final video signal is then correctly selected by the FSM to be outputted to the video out signal on the diagram, out to the screen.

Main Menu:
The module will display the beginning of any game with a start command and if we are to incorporate music into the game, there will also be an option to turn up or down the volume. This module is also in charge of producing the onscreen image representing the dead state.

Collision Detection:
The module will take the position of the ship and the enemies sprites into consideration. When there are overlap in the images, it will send an appropriate command to the game to determine the next state. It will also do the job of telling the player whenever the player manage to get a hit using the laser on the enemies.

The Ship:
Is in charge of reading the sprite data from the memory and update its position given the signals coming from the LSM9DS1 through the receiver calculator module. This calculated positional information is fed to the FSM and the Collision detection module.

Sprite Generation:
Reads enemy and projectiles sprites from the ZBT memory creates the necessary on screen objects and updates their position according to the AI Module calculations. It also removes the enemies from the screen when defeated. It is also in charge of creating and updating the position of the enemies projectiles.

ZBT Memory:

This module is in charge of holding all of the sprite information for the game ship, enemy, and projectile models. The data for the memory will be written to it making use of the vivado interface. We are still working out the details of how to use the memory and assessing its limitations for the complexity of the game.

Separation of Work:

Tuan:

- Accelerometer, gyroscope, camera, gun, sound generation, main menu display. Will help Edwin's when tasks finished.

Edwin:

- The game and the internal modules associate with the FSM.

Resources:
For our project, we will use a camera to processing the location of the gun beam. We will use a LSM9DS1 for the accelerometer and the gyroscope measurements. For the sound generation, we will use a speaker and the 74HC08 chip (similar to lab 4). We will be using the ZBT memory from the labkit. We will display our result on the monitor we have in lab and the labkit.

Conclusion
We want to create a game that has a very interesting mix of inputs. In order to ensure that the game is fun, we want the difficulty of the game to scale linearly with the length of time/ skill of the player. We also want to ensure that we have as many enemies on screen as possible by the end. We hope that once the dust settles, we won't have just a barebones game, but a fun and challenging experience that lives up to the name of the classic Space Invaders.