Rifle Arcade Game

Austin Phillips-Brown
Casey Wessel

Introduction

Project Overview

We will be making a virtual target shooting game similar to a shooting video game you would play in an arcade. The standard version that you would find in an arcade is bulky - the gun is permanently attached to the machine and costs thousands of dollars to manufacture as well as to maintain. Our version is a low-cost alternative that uses an FPGA and an Arduino to do the "heavy lifting" of the game. What makes our "light" version of the game possible are the methods that we use to track the movement of the gun - a combination of image tracking and gun angle calculations. The game itself would first begin by displaying options for game levels of 'beginner', 'intermediate', and 'expert'. The player will have a physical plastic rifle, which they will aim at the computer monitor. The player will select their game level by aiming at the desired level and pulling the trigger. The screen will display a target, its size will be based on the level selected. Higher levels will have smaller targets which correspond to a distance farther away. The player will aim the rifle at the displayed targets. A combination of image tracking and a gyroscope will calculate where the player is aiming on the screen and will display an image of a sight on the screen so that the player can see where they are aiming. To fire, the player will push a button on the rifle, which will be wired to the FPGA. The rifle will also include a "safety" switch, which can be used to pause the game. The display will show any bullet holes if the target is hit, and if the player has accumulated enough points, the game will advance to the next level.

Motivation

Our main motivation behind this project is that we wanted to create something fun to play and test in the lab. We thought a virtual game would be a great way to accomplish this goal.We picked a shooting game in particular because we both have taken rifle classes and enjoy the challenge of testing our aim. One of the challenges of practicing shooting with a gun is the restriction on practicing in a safe environment, like a shooting range. We thought implementing this game on the FPGA would allow us to simulate practicing a sport we enjoy, in a more readily accessible environment.

Implementation

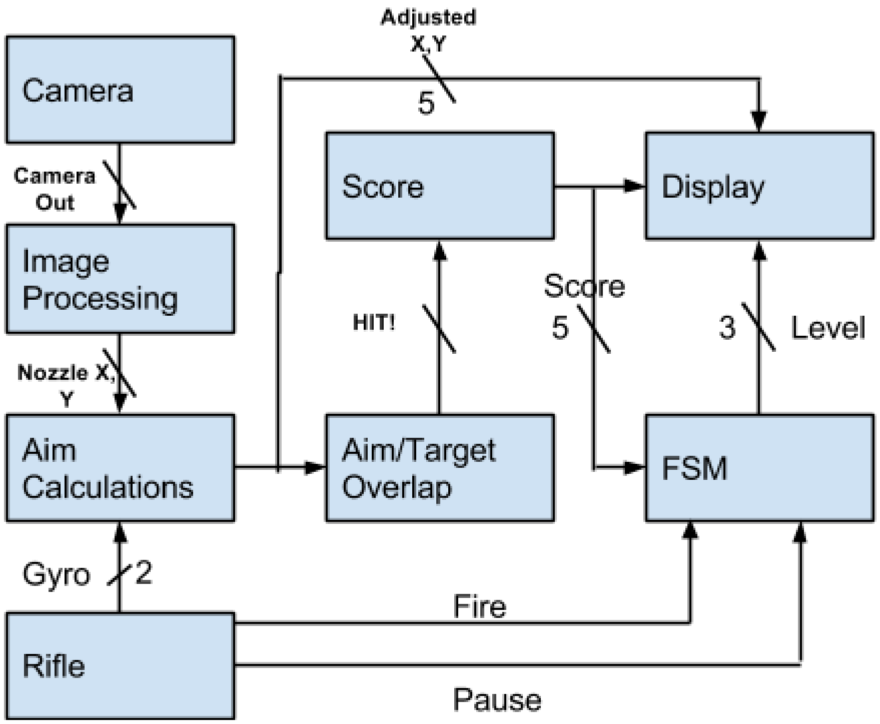
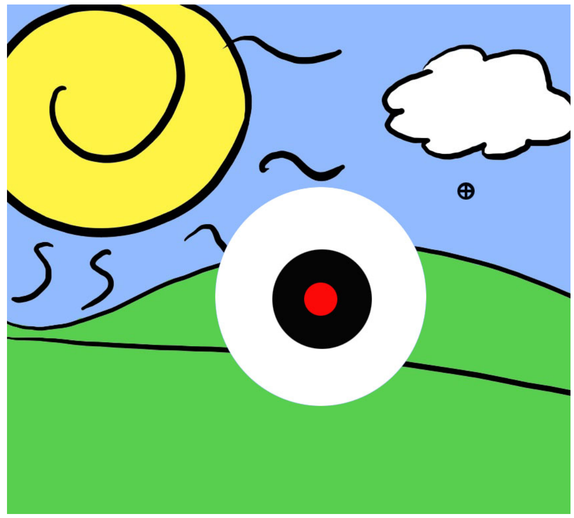


Figure 1. System Block Diagram
Outside hardware - Camera and Rifle Inputs, FPGA Modules - Aim/Target Overlap, Score, Display, FSM
Arduino Modules - Image Processing, Aim Calculations

Display

Author:Austin Phillips-Brown and Casey Wessel

For this portion, we will aim to design methods that create the specific targets for our game. In general, we will have three standard target sprites: A short distance target, a long-distance target, and a medium distance target. These methods will be called at the start of each level. Each target will have three rings of different colors. The number of points the player receives will be dependent on which ring the player hits. The target sprites will move around the screen during the round, similar to the pong game, making it more difficult for the player to hit the target. In addition to the targets getting smaller, the speed of movement will increase as the player reaches more difficult levels.



Picture 1. Example of a Level Layout
Picture composed of Background, Target, and Aiming Aid (crosshairs)

The Display Module will take, as an input, the position of where the bullet would hit the target - we refer to this as Aim Location. Based upon the input value for the Aim Location, the Display Module will place a Sight sprite at a the corresponding location on the monitor. The Sight sprite will represent where the person is "aiming" and where the bullet itself would, ignoring wind, likely hit. If a person hits a target, a version of the target with an "X" shape at the Aim Location will appear. Rather than storing the images in RAM, we will write methods to produce sprites for the targets, "X", and pointer. This will be very similar to sprites we created when we implemented the game Pong on the FPGA.

To make the game a little prettier, we will also implement a scenic background behind the moving target. Using a MATLAB script, we can convert the JPEG image to RGB values to be displayed via the FPGA.

Camera and Image Processing

Author: Casey Wessel

The camera's purpose is to find the general location of the Rifle nozzle. First, we will need to use the provided Verilog code to take the data from the camera and convert it into a color image. That color image is then processed on the FPGA in the following manner: We search the pixels for the color of the neon orange that we set on the rifle nozzle and mark the locations where we see that image as the location of the nozzle. After finding all the locations, we choose one as a center point; this does not really need to be extremely accurate for our purpose as the camera will likely only have a few pixels containing the rifle nozzle.

The resolution of the camera may not necessarily be the resolution of the monitor so we will be forced to adjust our point's location to fit on the VGA screen. For this, we plan to set a ratio of the resolutions from the camera to the VGA display - for example, if we use 600 x 480 pixels for the display and the camera is 1024 x 1024, the x, y coordinates that we find will be "adjusted" by the general equation:

$$\frac{Position_{Camera}}{Res_{Camera}} * Res_{Display}$$

Rifle and Aim Calculations

Author: Austin Phillips-Brown

The Rifle module has three key aspects, the first two of which are relatively simple. We will have a button/switch system on the rifle that will relay the Fire and Pause commands to the FPGA respectively. The button will be normally disconnected and will connect 5v from the FPGA to the io port of the FPGA when pressed. The setup for the Pause function will be very similar but using a switch instead of a button. The idea behind the switch for the Pause function is that it is similar to a safety switch on a rifle.

The third task will be finding the angle of the rifle with respect to the x and y axis. For this portion, we plan on using a gyroscope with the analog outputs for the x and y angle being supplied to an Arduino. The Arduino will send the data to the FPGA via the serial port. The Aim Calculation Module will also take pixel coordinate from the Image Processing Module. Using the angle from the gyroscope and the pixel coordinate from the Image Processing Module, we can use simple geometry to figure out where the bullet would hit the screen.

The location where the bullet would hit the screen is then sent to another module named Aim/Target Overlap. This module will take in the location of the bullet hit and the display image. It will detect if the location of the bullet hit was within the target, and depending on which section of the target the bullet hits, it will output a certain number of points to a Score Module, which will keep tally of the players score for that level.

Timeline

For our timeline, we separated what we believe to be “high-risk” material from the rest of the tasks. The “high-risk” activities will require large amounts of time and it is possible that we will not be able to effectively implement them - these tasks include the Image Processing Module and the Aim Calculation Module. These tasks are given a week each so that we can be sure to either complete them or have enough time to consider alternative strategies. We’ve also put the sound module at the end because it will add a fun element to our game, but is not necessary for the overall operation.

Due Date	**Task Complete**
10/31	Moving Target and Background Image
11/7	Gun wired, FSM module written, Score module
11/14	Image Processing Module
11/23	Aim Calculation Module
12/10	Modules interfaced together, Sound module

Supplies

Part	Cost	Status
NTSC Camera	Free	Acquired
Plastic Rifle	$20	Acquired
Arduino Mega Board	$32	Purchased
Gyro Breakout Board	$12.50	Purchased
Button, Switch	Free	Acquired

Table 1. Project Supplies Table