The Dueling Club: An Interactive Spell Casting Game

Marayna Martinez and Lotta Blumberg

November 6, 2016

1 Overview

A large component of the Harry Potter universe hinges on wizard dueling. Two wizards face off and cast spells, hoping to inflict damage on one another. The goal of this project is to bring that experience to the real world. A camera will track movements of a wand. Based on patterns that the wand traces, different spells will be cast. A visual output onto computer screen will show wand movements and spell responses. Spell responses can range from single player spells like levitating objects to two player spells like inflicting damage. A finite state machine will control competing spells to determine who wins and who loses. Two players will be able to brandish wands, cast spells, and fight to the finish.

2 Design

Overall, the game design consists of a input from the camera. This project will use a NTSC camera with an IR filter to track up to two wooden wands with an LED light attached to the end. The movements of the wand will be decomposed into x and y coordinates. The coordinates will be fed into the labkit. After the labkit has the coordinates, modules will process the data. First, the x and y coordinates will be translated to x and y coordinates on the screen. A sprite will show the wand strokes that trace out a spell. Based on the spell traced, a spell number will be assigned. Using a case structured module, the spell number will trigger a series of sprite motions on the computer screen that make up the spell response. A finite state machine will take into account both players spell number, then produce a reaction (either adding or subtracting points). When point threshold are met, the game will end and one player will be declared the winner and the other the loser.

2.1 Design Decisions

Several very important decisions were made while designing this project. First and foremost, a method of wand tracking had to be decided. There are two labkits with two camera options to be considered. The newer setup has better camera quality, but less documentation. The older setup has worse quality, but plenty of documentation from previous projects. This project uses the old labkit to take advantage of the previous documentation. The robust nature of the old labkit is more beneficial than better camera quality. Second, the wand could be tracked using color or IR. Tracking specific colors is easier to accomplish, but also easier to mess up if something in the background has the same color. IR is slightly more difficult to implement because extra hardware is needed, but more robust. For this project, IR tracking was chosen. IR tracking on the old labkit is a good starting point, but this project has the ability to expand in complexity by adding a second LED to the bottom of the wand. Then, additional processing can measure the distance between the two lights and solve for the angle of the wand.

Second, the two players could either be tracked by a single camera or by two separate camera. In the end, it was decided that adding a second camera and feeding both sets of data into one labkit for processing was not worth having a larger space to cast spells.

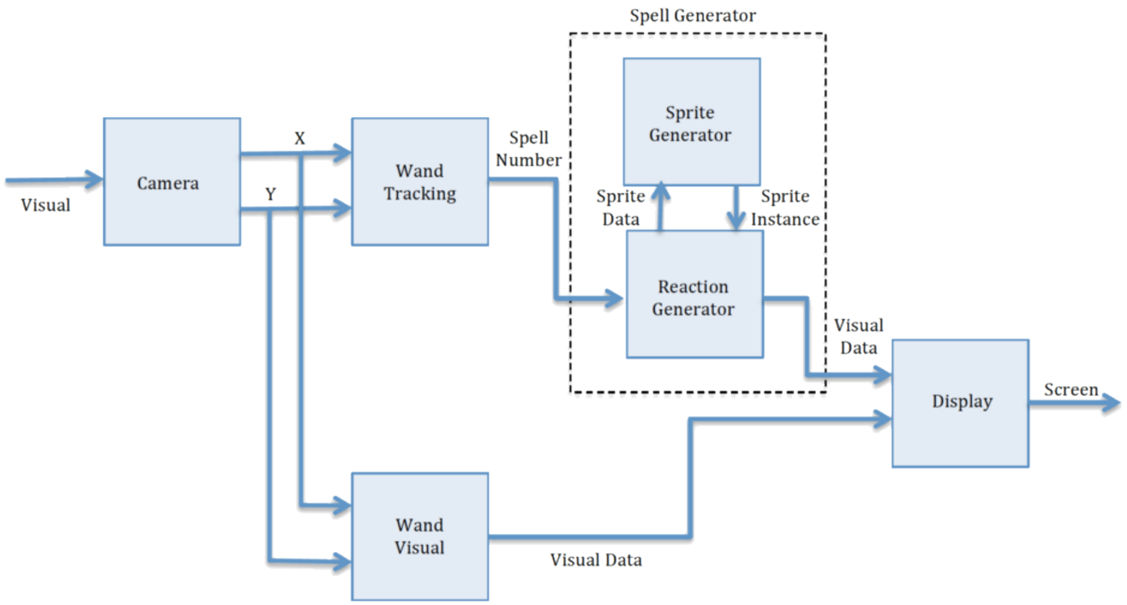
Finally, the last big decision was how to structure project deadlines. At first, the timeline for the project was going to be focused on a well developed single player game. Adding a second player was going to be a reach goal. The project timeline was amended to make two players a higher priority. Developing a more extensive bank of spells will be a late stage goal. Having a two player game adds more to complexity and makes the final project more enjoyable.

2.2 Motivations

The primary motivation behind this project was to create a fun product that was also attainable. A game system with visual output fulfilled the fun criteria. The idea of wand movements was inspired the air drum kit project from a previous year, so the basis of the project was attainable. The project also needed to be buildable so that minimum and reach goals could be established. Having a bank of increasingly complex spells allows the project to scale in complexity.

3 Implementation

3.1 Block Diagram



3.2 Modules

Camera Block: This block takes in visual data from a camera. It looks for the location of the wand tip, the LED, and outputs the x and y coordinate of the location of the wand tip.
Wand Visual Block: This block takes in the x and y location of the wand tip and creates visual data of a sprite on a display at the corresponding location.

Tracking Block: This block is a complicated one that takes in the position of the wand over time and decides whether the wand has activated a spell. The way this works is that it splits the screen into several chunks and then tracks which chunk the wand is in. There is then an FSM that figures out whether the series of chunks corresponds to a spell. If so, the block outputs that spell number.

Spell Maker Block: This block takes in a spell number and creates some response that depends on the spell. It outputs visual data shows that the spell was successful. The precise visual data will range from moving sprites on the screen to a health bar that tracks points during two player game play.

A possible single player spell bank could include:

- Accio: Bring sprite onto screen
- Wingardium Leviosa: Book sprite moves up the screen
- Engorio: Scales the sprite larger
- Lumos: LED on labkit turns on
- Nox: LED on labkit turns off
- Aparecium: makes a message appear onscreen

- Avis: Shoots a bird from the end of the wand

Display Block: This block takes in visual data (location and motion of sprites) and processes it in order for it to appear on the screen. This block is very similar to the Pong game infrastructure.

4 Timeline

The project is split into 5 parts, with the first two being necessary, the 3rd and 4th being target goals, and the last being a stretch goal.

1. Wand-Tracking
   Due Date : November 11
   Team Member: Lotta Blumberg
   Build a wand and whatever is necessary to have the camera successfully track the wand tip and send the data in the form of x-y coordinates to the labkit. This module will be tested by reading out whatever is being fed to the labkit.

2. Make a Spell
   Due Date : November 11
   Team Member: Marayna Martinez
   Create a mechanism for recognizing when a spell is activated and have this displayed on screen while also displaying the wand movements. This module will be tested by using a manuel input from labkit buttons.

3. Add a Second Player
   Due Date : November 18
   Team Member: Marayna Martinez
   Split the screen in half and make it so that both players wands can be tracked. This module will be tested the same way as modules one and two.

4. Create a Dueling game
   Due Date : December 14
   Team Member: Marayna Martinez and Lotta Blumberg
   Make a game in which two players interact. They can attack each other with hexes and protect themselves with charms. Each player has a life-bar that when decreased all the way causes them to lose. This module will be tested incrementally. It will start with just outputting numbers with each input cell. Then, the health bar sprite will be developed and tested manually with buttons. Finally, the two pieces of testing will be put together.

5. Add Spells
   Due Date : November 18
   Team Member: Lotta Blumberg
   Develop enough spells and interesting responses to them so that the game is fun. Testing will depend on what each spell will do.