

Gilberto E. Ruiz, Kyle Maiorana  
Dr. Senem Velipasalar  
ELE 400 Video and Image Processing  
October 6, 2022

## Project Proposal

The hallmark of a good final project is an engaging and entertaining demonstration that an audience can interact with. To accomplish this for the final project in Image and Video Processing, we wanted to create a real-world variation of one of our favorite interactive video games. There have been several hit implementations of dueling swordfighting gamemodes over the years, though they primarily use infrared tracking to capture movements. For this project, we will instead use a camera feed to capture a player's real-time movements of a foam sword, while algorithmically defending their strikes by generating a virtual opponent's blade. The software will also overlay on the screen a blue line indicating where the real-life sword is, and when an attack is being made, a red line will appear that shows what an opponent would do to counter. In order to recognize when a strike is happening (versus the player standing around), the velocity of the sword will be tracked in a data structure and software will determine if the movement is above a determined threshold for what constitutes an offensive movement. We can introduce varying threshold values to increase or decrease the sensitivity of the program, as well as potentially introduce a form of gameplay that has staggered levels of difficulty. Another possible form of processing to introduce would be the opposite situation, where a line appears on the screen and the player has to orient their sword to block the virtual attack. The program would be capable of recognizing the reaction time and effectiveness of the movement.

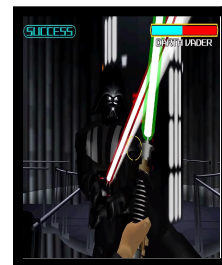
It will likely be a good idea to fashion a sword for the player that has indicating markers on it, similar to what is used by motion-capture systems. This will help the software identify and orient the weapon, even in a low light or noisy situation.



Figure: Proposed Sword for the project



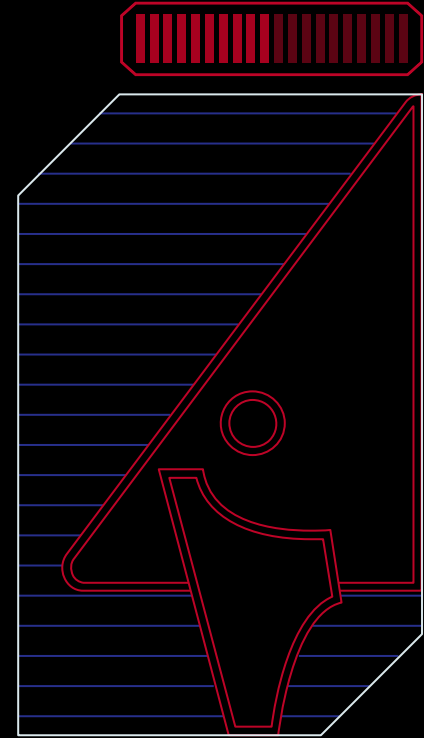
Samurai Game on Wii Sports Resort (2009)



Star Wars Arcade Game: Duel with Vader (1998)

# OpenCV Duel Final Project Demo

Kyle Maiorana and Gilberto Ruiz





## PROJECT RECAP

A camera feed captures a player's real-time hand movements to let them defend against a virtual opponent's generated blade.

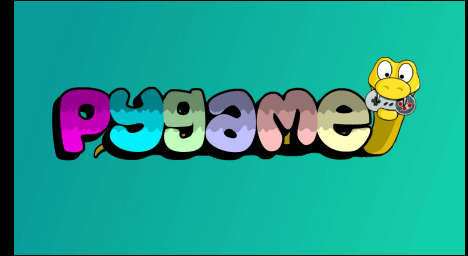
The software overlays a red blade on the screen indicating where the opponent's sword is, and when an attack is being made, a blue saber line will appear that shows the player's motions to defend.



# USEFUL TECHNOLOGY FOUND

## PYGame

- The PyGame Library is an open-source package for Python intended to help structure games and other multimedia applications.



## Menu System made by BaralTech

- BaralTech on YouTube created an open-source menu system with the PyGame library.
- Consists of 3 screens that can be accessible through buttons that've been programmed in the code.



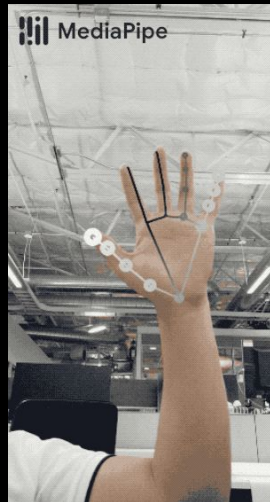
# USEFUL TECHNOLOGY FOUND CONT.

## Google MediaPipe Hands

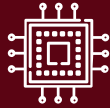
- Cross-platform, customizable ML solutions for live and streaming media, including a model for hand recognition.
- It employs machine learning to infer 21 different 3D landmarks of a hand from just a single frame.
- Regression model was trained using truth data of 30k manually annotated hand images.

## MediaPipe Documentation

- Helped us get the hang of mapping a drawn image to specific hand landmarks for the lightsaber.



# WHAT WE ALTERED



## Adding player and computer blades at the same time

Kyle used the hand landmark points to anchor the virtual lightsabers for gameplay, and predetermined opponent swing coordinates were programmed. Now the game can cycle through swings and determine when contact is made using coordinate intersections.



## Customizing screens and adding background music/sounds

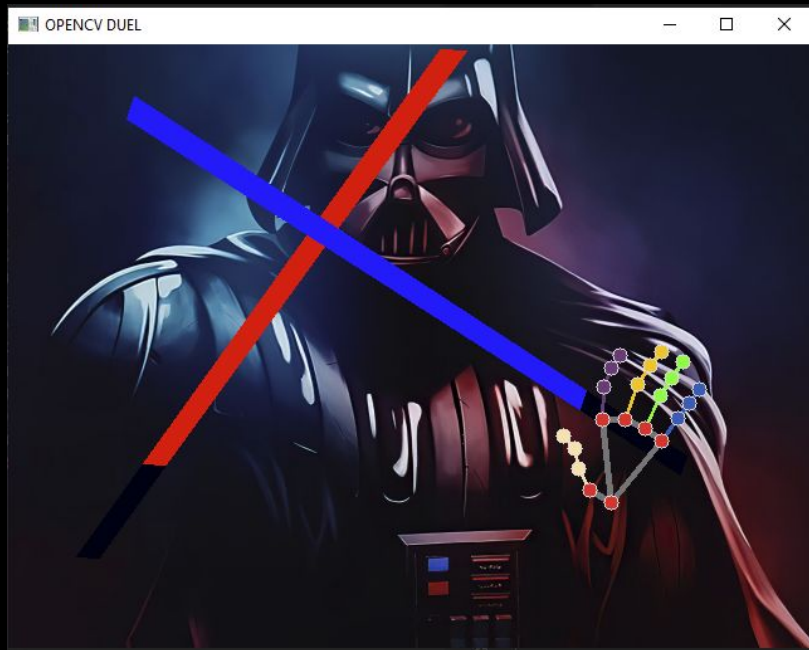
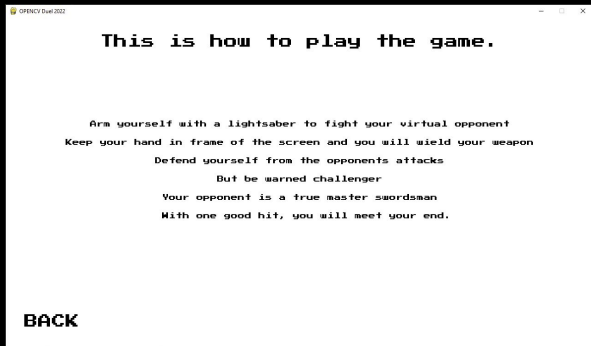
For BaralTech's main menu code, Gilberto customized one of the screens to be about the game's instructions and added background music. Game flow was discussed as well.



## Integrating all program components to play the game

The processing code was inserted into the gameplay driver so that it feels like you are playing a video game.

# Current State of Program



# WORKS CITED

- <https://www.osti.gov/biblio/4746348>
- [https://docs.opencv.org/3.4/d9/db0/tutorial\\_hough\\_lines.html](https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html)
- <https://www.geeksforgeeks.org/opencv-c-plus-plus-program-to-create-a-single-colored-blank-image/?ref=rp>
- [Zago M, Luzzago M, Marangoni T, De Cecco M, Tarabini M and Galli M \(2020\) 3D Tracking of Human Motion Using Visual Skeletonization and Stereoscopic Vision. \*Front. Bioeng. Biotechnol.\* 8:181. doi: 10.3389/fbioe.2020.00181](#)
- <https://google.github.io/mediapipe/solutions/hands>
- <https://github.com/baraltech/Menu-System-PyGame>



ANY  
QUESTIONS  
?