

# Samurai Piledrivers

Matt Kaiser  
Tanner Quigley  
Kevin Flynn  
Nicholas Fentekes  
Robert Perez

## Final Project Proposal

May 4th, 2018

### Goals

When we first started this project, our main goal was to create a game controller that anyone could customize to their needs. We wanted to introduce a device that fit a diverse range of abilities and disabilities; removing the necessity of a uniform controller/mouse and keyboard. This idea formed from the lack of research and accessibility that is involved in games today. Mostly all games utilize a controller of some sort and rarely are any of these accessible. Our idea was that we could utilize the accessible features of machine learning in order to create a controller of our own. This would then be connected to a ui that allowed the user to customize their process and gestures. These would map to buttons and combos in our game and the user would then be able to play a game that they were unable to before.

### Design

For our final design, we ended up going with a very linear process for our UI. You begin by being welcomed to our project with an introduction on our home screen. After that you're taken to our Build New Schema screen. This allows you select all of the buttons and combos that are allowed to be trained for our controller. You will then be able to map simple gestures to buttons and then train a combo of simple gestures for the actual game combos. Some of the

tradeoffs that we had to do was a lot of functionality aspects of the game. One of our main problems was the when using the microbit to control movement left and right, it would interrupt some of our script combos that were triggered by the myoband. Also we were unable to save recordings of wekproj and then successfully load them into wekinator. This is something that could have been successful with more time or a different machine learning platform. In the end, the myoband and wekinator ended up working very well for our end goal. These two things meshed well together and allowed an easy running of our program to make a controller.

## Architecture

In order for our project to successfully run, there were many different moving parts that we had to connect together. This included our UI, wekinator, inputs, outputs, emulator, and python scripts controlling the emulator. The one thing that brought all of this together was our UI. This is what essentially controlled all of the moving parts. Our UI is able to record and train our gesture inputs by sending/receiving all of the data through OSC messages, as well as launch and control our game through different scripts.

## Machine Learning

For our process, we began by sending our input from our myoband into a classification model. This model is what we used in order to create our simple gestures. These were gestures like, open hand, wrist extension, resting position, and making a fist. Although these are the gestures that we used, you're able to create any kind of gesture you want to map to the controller. This works much better if the gesture itself is distinguishable, but whatever you can think of can work. Inside of this, we used the k-nearest neighbor for our model type. This proved to be the most accurate at judging our simple gestures. To make the whole process and data received even better, the data from the first model is then sent to a smoothing process. This was just a smoothing average of the data received in order to make everything more accurate.

From there, we took our first wekinator model that was run through our smoothing average and sent this data to another model. The next model was a Dynamic time warping model. What this allowed the user to do was make a combo out of the simple gestures that were created. They're able to record a series of gestures like open hand, fist, wrist extend and then map it to one of our combos. This makes it a little more difficult and fun for someone to execute a street fighter command/combo.

## Results

Our final outcome was very much what we had envisioned from the start and the results we were able to achieve showed this. In our game, you are able to consistently beat a computer by using our game controller and combos. This is essentially what we had envisioned once we chose street fighter as our main game to focus on. The user can now successfully train their own gestures and execute multiple combos for three different characters in the game. Although we don't have any real evaluation data, we're able to play the game at a competitive level which in itself is a strong result.

## Lessons Learned

There were many different things learned during the whole process of creating our project. This included simple things of just how our team works best together to more complicated aspects of the project like getting all of our moving parts to mesh. Some of our biggest problems came from working with wekinator and the noise that comes with it. We learned how to deal with this noise and the best smoothing processes to use for our game. As well, we learned that it isn't possible or very difficult to run processing from the shell command. This just led to more complications in our project that was needed. If we were to do this whole project over, we would have decided to either stick with just python or just processing. We spent a lot of time just trying to get easy things working that were made much more difficult by using a combination of python/processing.

## Future Work

If another team were to pick up from where we left off, there are many improvements that this could lead to. First of all, currently our UI has all the practical features that we need, but if a team desired they could add a lot more design/css to our project to make it more visually appealing. Along with this there are many functional issues with the UI that could be streamlined. Things like controlling everything inside of our UI and not having to launch any outside programs would make things easier. For example, loading the actual game from the UI, saving/loading .wekproj from the UI, and having character select be done from here. As well as this, there could be an option that allows the user to create their own combos for other games. This option would allow the user to input some type of combo/series of inputs, that could then be mapped to the myoband or microbit output. This would make our project much more customizable than it currently is. One idea that would make things more fun for our process would be adding a two player mode to our UI and game. This could be accomplished with the use of another myoband. Overall, we have a lot of ideas that we could add to this if it was a semester or year long project.

## Bibliography/Video

<https://youtu.be/G5DaMzXrlmA>

Foundation, Processing. *Processing.org*, 4 May 2018, [processing.org/](https://processing.org/).

*Wekinator*, 4 May 2018, [www.wekinator.org/](http://www.wekinator.org/).

"Myo Gesture Control Armband." *Myo Gesture Control Armband | Wearable Technology by Thalmic Labs*, 4 May 2018, [www.myo.com/](http://www.myo.com/).

"Micro:Bit Educational Foundation." *Micro:Bit*, 4 May 2018, [microbit.org/](https://microbit.org/).