Project Overview:

Our project utilizes OpenCV and your laptop’s webcam to translate the movements of your hand into the movements of your mouse cursor. By changing the physical location of your hand in space, you can move your mouse, click on objects, and interact with your computer kinesthetically.

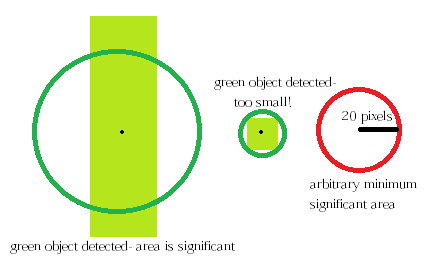
Results:

We succeeded in implementing OpenCV to track the movements of an object. However, due to lack of time, we chose to track hands by color, not through finding a convex shell. In order to increase background contrast, we chose to track the color green, since it was relatively scarce in the environment.



*Figure 1: The webcam output video. The red point represents the center of the detected green object.*

Once the webcam located a significant amount of green, (defined as having an area equal or greater to that of some arbitrary circle of radius 20 pixels) our program calibrates itself by finding the exact radius of the object’s circle.

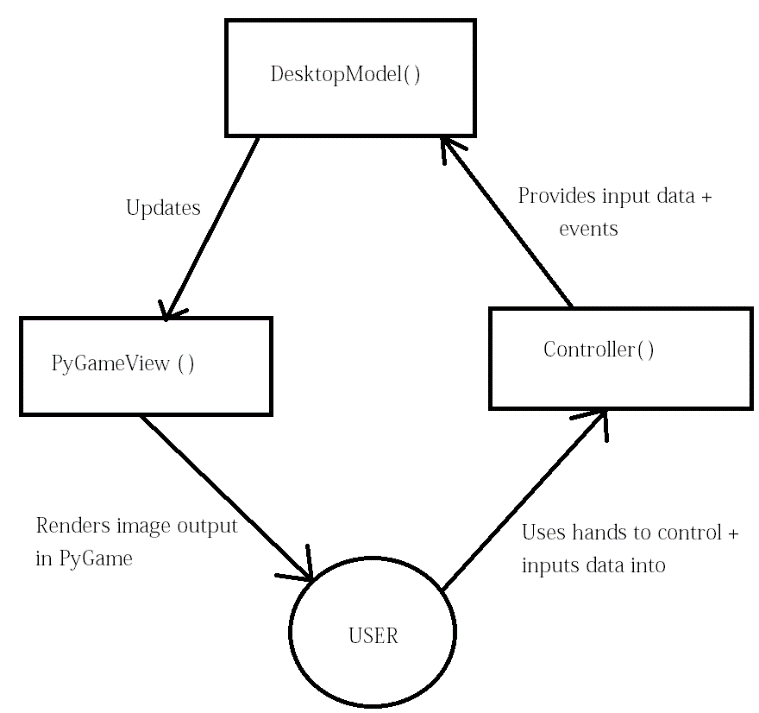


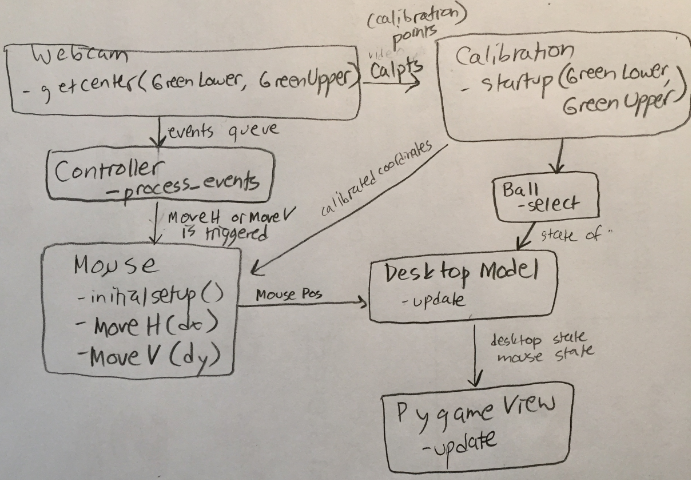
*Figure 2: Determining the significance of a green object*

Later, this value can be compared to calculated minimum and maximum threshold values to allow our program to detect when the object has become significantly larger or smaller in size. This indicates movement of the object towards and away from the camera. By detecting if an object’s area has crossed one of these thresholds, our program can direct a circle in pygame to change color. Future work would involve implementing actual mouse click functionality in Linux, not just in Pygame.

Implementation:

We eventually decided to use a Model-View-Controller design to organize and implement our code. This allowed our code to run much more efficiently, and was a lot easier to understand.





Reflection:

We were able to quickly implement the functionality that we wanted using Python and OpenCV. However, we spent a lot of time reorganizing and rewriting code because we didn’t use classes or objects. After rewriting our code, it was able to run much more efficiently, without as much lag. Our unit testing was adequate, thankfully, though we could have probably commented our code more thoroughly. If we were to do anything differently next time, we would likely begin using classes and objects immediately, rather than waiting until we had a functional implementation.

Our project was appropriately scoped. Anna and I finished implementing our desired features by the deadline, but we were unable to work on any of our stretch goals, and we were a little pressed for time on the written work. We planned to meet together occasionally to pair program, but mainly work independently and push to GitHub, since we were both pretty busy. Our meetings were very productive, and we worked well together. There was a slight hiccup at the beginning of our project, where I couldn’t code for half of a week or so because I didn’t have webcam functionality in VirtualBox. However, once that issue was resolved, work progressed pretty smoothly.