TITLE PAGE

s

Introduction

The goal of this project is to implement a visual programming language where the method of input is through a gesture-based system. This can include head movement, hand or leg movement, facial expression, etc. This will all be done through accessing the user’s camera and tracking the actions that they do. The program will include different types of problems through programming such as: sequence, selection, iteration, and modules/sub-programs. A main goal of this project is to make the gesture-based system feel natural. This can be achieved through using standard HCI rules.

Background research

Human Computer Interaction (or, HCI) is a technology that has been developing at a fast rate in recent years after gaining traction around the 1980s (Interaction Design Foundation, 2021 <https://www.interaction-design.org/literature/topics/human-computer-interaction>) The concept of HCI is a multidisciplinary area of study that focuses on the design of computer technology. To be more specific, HCI is a technology that primarily focuses on the development of interaction between human and computer, giving it its name ‘Human Computer Interaction’. HCI’s main boom in research focus was around the time that devices such as the Apple Macintosh, and Commodore 64 became somewhat feasible for home and office use, due to their affordable price. The devices were capable of many tasks such as word processing or handling information through digital spreadsheets. While most people today believe these tasks to be simple tasks to complete, they have HCI to thank for it being that simple. A time before HCI was when computers were built for specific purposes. This meant that HCI was not important as the people using the computers were few and would also likely be knowledgeable within the current devices interface. However, when it came to distributing computer systems, it became vital to make the computer interface user friendly to the masses. People using the devices weren’t expected to be familiar with the computing devices and therefore the design of the interface would have to be simple, and efficient. (Adobe, 2020 <https://xd.adobe.com/ideas/principles/human-computer-interaction/>)

HCI has many different uses, but one of the main ones being developed is ‘Computer Vision’ this allows the computer to see images or video footage and determine what it is seeing. Presently, computer vision can see and identify objects. Things like self-driving cars are at the forefront of computer vision where sensors detect where the road is, danger of pedestrians, and traffic lights. The computer interacts with its surroundings to determine what it should do. Another rising example of this is the Amazon Go stores that have recently been put into trial for select members. The premise of this store is that there is no physical user interaction with any machines in the store. You simply pick up what you want, and leave. The money is deducted from your account when you leave the store. Go Store is the Computer Vision based Machine Learning that is used to seamlessly track and estimate the intention of everyone in the store. (<https://towardsdatascience.com/how-the-amazon-go-store-works-a-deep-dive-3fde9d9939e9>) Things like this make things more seamless for the end user and is what HCI is all about, making computer interactions more natural. Doing this also saves a significant amount of time as there is no need to be queueing up to pay for your items.

One of the many subcategories developed from HCI is Perceptual User Interface (or, PUI). The idea of a PUI is that it will allow a computer to interact with the user without the use of standard peripherals such as a mouse or a keyboard (J. Davis, 2001). This is done through using other peripherals such as a camera or a microphone. Using a camera can allow the computer to detect the user’s gestures through video footage using computer vision. This means that the user can interact with the computer through something that may seem more natural to a person without experience with standard peripherals. This works through the computer using gesture recognition and detecting what the user is doing. One example of this is the Xbox Kinect. The Xbox Kinect is a gesture-based interaction system that has been used by millions, mainly due to it being bundled in with the Xbox 360 (A popular video game console) upon its release (Makuch, 2013). The Kinect uses an infrared camera that can draw a digital skeleton of the user. It uses this skeleton to allow the user to navigate the interface. This is done using different gestures such as swiping in a direction as if you are dragging the interface with your arm. This is very intuitive and is something that can be seen as like swiping left or right on a mobile device, something most of the population is familiar with. PUIs have many different advantages to them, outside of just being a different way to interact with a machine. Different interfaces can help people with disabilities participate in using a computer. For example, Stephen Hawking, could use a keyboard using a special keyboard that had a sensor built in for scanning allowing the computer to see which letter should be processed. This interface allowed Stephen Hawking, someone with a severe motor disability to participate via computer in a brand-new way. (<https://www.researchgate.net/publication/344748456_Accessibility_of_Different_Natural_User_Interfaces_for_People_with_Intellectual_Disabilities>) While this is an extreme case, and the interface was built with the minority of people in mind, it is important to consider different disabilities when designing an interface.

Design is one of the key factors when building an interface. There are many HCI guidelines out there to help build one. Following an already given guideline is affective as it means that there will already be other interfaces following the same design. This means that if other users come to use an interface that follows the same design, the gesture-based system will be natural to them. One example of these types of guidelines are the Kinect User Interface Guidelines. (REFERENCE THIS IDK) This guidelines document acts as the requirements for all developers creating something for the Kinect on Xbox. This means that if you were to use a software on Xbox that uses the Kinect, they will always have similar methods of navigation and interaction. The document discusses many ways in which the HCI should work in the case of Kinect. Firstly, it states that the software should ‘Enable users to accomplish tasks quickly and intuitively.’ This is opposed to making the user do an inappropriate input that doesn’t make that much sense for a user to be doing. It states that an input method should never feel ‘Forced, unnatural, awkward, or tedious’ this is very important when designing an interface, as it being natural is the primary appeal of the technology. It is also important that gestures aren’t too similar when designing an interface as well. If a gesture is too similar, it could get mixed up with something else the user is doing resulting in not the outcome that the user intended to get through using the interface. This can overall lead to dissatisfaction and dissuade people from using the interface over something like a keyboard and mouse. Visual feedback is another essential feature when designing an interface. When the user successfully completes a gesture, there should be some sort of signal that what they did was what they wanted and has been successfully completed. Lastly, some sort of configuration should be done to start with an interface like this. Making sure that the device is working correctly rather than having the user move their arms at something that won’t work. This can be done by asking the users to do certain gestures to see if they work before allowing the user to continue. To summarise, there are many things that need to be considered when designing an interface. This project aims implement some of these into the methodology used when creating the program to allow for a simple and efficient, user experience.

Implementation

For this project the program will use the OpenCV library. This library is built to collect camera input along with the ability to track movement using Haar Cascades.

There are four output windows for this program. The one on the far left is the camera output. This shows the user what exactly the camera is seeing. It can also help the user figure out if the gestures are working using the feedback that the window gives when completing a gesture. The middle window is the main window. This is where the questions will appear allowing the user to choose between two options to finish the code. The user must select which answer they would like to submit into the code using the gesture-based system. The code can be seen in a small window with part of the answer blurred out. Once submitted the user will then be informed whether their answer works in the code using the final window, the output window. The output window is where the results of the code will be displayed for the user to see their code in action. If the code is wrong it will simply not progress the user any further while if it is correct, it will display the output of the code that the user has created through the gesture-based system.

A screenshot of a video game

Description automatically generated with medium confidence

The triangle on the user’s camera is defaulted to the colour white. When completing a gesture (either smiling or creating a fist motion) the colour of the triangle will change to either red or green. This corresponds to the colours that are shown on the main window with the purpose of giving the user feedback and allowing them to see if their inputs are successfully being picked up by the application.

The program uses Haar Cascades and the OpenCV library to be able to identify different gestures.

Text

Description automatically generated

The cascades that this program uses are ones that have already been created by people prior and contain a good amount of accuracy. Having to do this without premade cascades would take much longer as it would have to learn what exactly it needs to detect for each gesture. This system works by taking each frame of the camera’s output and checking it for any of the gestures. This is repeated indefinitely until the code comes to a stop when the user completes the smile gesture when prompted to end the program.

A screenshot of a computer

Description automatically generated with medium confidence

This is done by creating text to prompt the user that they have finished all the coding challenges. After they complete the gesture the ‘guess’ will break the loop and destroy all the created windows, ending the program without the use of a different peripheral.

Text

Description automatically generated

Above is how exactly the question system works for the program. At the start of the program currentQuestion is set to 1 allowing the program to start from the first question. The correctGuess value determines whether a fist motion or a smile motion is the correct gesture. In this case it is 2 which makes the smile gesture the correct answer. After that, all of the text is placed onto the main UI window. This displays: the question, the answers, and the gesture prompts for each of the answers meaning that the users do not have to remember which gesture does which. The code will then also show the code snippet that the question is working with showing the code and the blurred-out section that the user is attempting to fill in. The final section of the code takes place when the user successfully gets the correct answer. First a new window will be outputted showing the output of the program when the correct code is selected. After this it will then change the currentQuestion value to itself plus 1. In this case that means that the program would then move onto question 2 and run the same process again with different questions and answers.

REFERENCES

DAVIS, J. and VAKS, S., 2001. A perceptual user interface for recognizing head gesture acknowledgements. *Proceedings of the 2001 workshop on Percetive user interfaces - PUI '01* [online]. [Accessed 10 March 2022].