DES 206 : Prototyping Interactive Systems

Prepared by Group 22 :

Aajay Ayyappan Devaraj - 2021001
aajay21001@iiitd.ac.in

Anshuman Bunga - 2021016
anshuman21016@iiitd.ac.in

Dhvanil Sheth - 2021040
dhvanil21040@iiitd.ac.in

Shivam Dwivedi - 2021352
shivam21352@iiitd.ac.in

Full-Motion Gaming Controller

Mini Project - Mid Semester Assignment

Contents

An Introduction to the concept

Part One

The gaming industry has seen significant advancements in technology over the years, with game developers constantly pushing the boundaries to provide users with more immersive and engaging experiences. The evolution of gaming controllers has been a significant contributor to this trend, with new technologies being developed to allow for more intuitive and responsive control of in-game characters.

With most of the group members being avid gamers ourselves, when tasked with the objective of creating unique game controllers, provided the Makey Makey Kit and Machine learning applications such as "Teachable Machine" by Google, we were motivated to create an innovative product that made gaming more interactive and entertaining experience.

After rigorous ideation and designing, We decided to utilize a combination of Makey Makey controller inputs and video camera inputs to allow for full-body movement control in an old but popular game, Cactus McCoy. We also made sure that the entire gaming controller could easily be mapped to control and play any game of the user's choice.

To accomplish this, we had three laser-cut wooden pads on the floor and a handheld wearable "brass knuckle" type object to which the Makey Makey controller inputs were fed to.

We also have a Python script that runs during the gaming process that takes in video input that recognizes and translates gestures and poses identifiable by the model trained by the Teachable Machine. The code would also provide precise text-to-speech translation of each input provided by the user to help them recognize their input and control their movements accordingly.

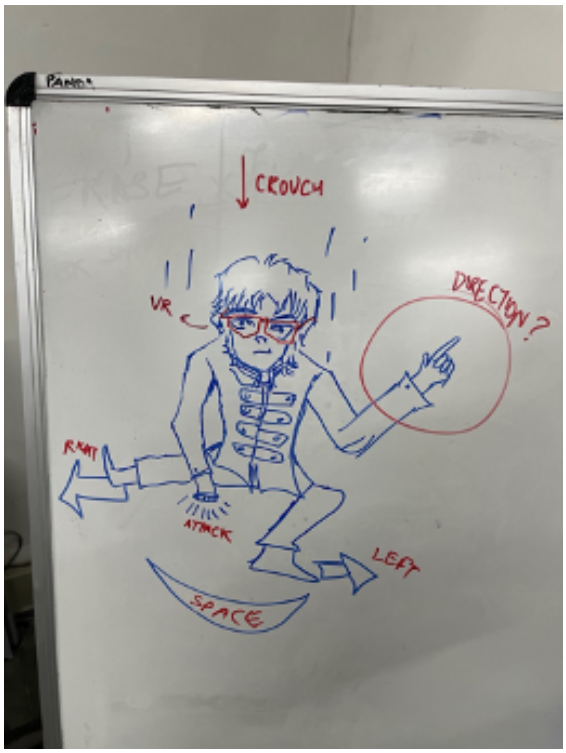
Each component of the entire gaming controller set could be mapped to any game's controls depending on the specific application that the user desires to play.
In the context of the game "Cactus McCoy" though, the pads on the floor are used for moving around the map, such as moving left, right, and jumping up, which the user will control using their feet. The brass knuckle wearable is worn on the right hand and is used as a trigger to attack enemies or fire a weapon as the user clenches their fist. Finally, the video inputs recognize the direction in which the user points their brass knuckle towards which the character in-game aims their weapon, the program also inputs the user crouching to make the in-game character crouch or climb down.

With all the components of the controller in play together, the gaming experience of the user is elevated to the next degree, and by simchanging the assignment of the keyboard keys to the various inputs in the Python program, the user can use the controller to upgrade their gaming experience with any game of their choice.

All in all, through multiple rounds of ideation and redesign, our group has created a prototype that is entertaining and engaging. Involving the motion of hands, legs, and body as a whole, we've developed a comprehensive product that possibly, with further development and investment, could be a product that many gamers can enjoy and upgrade their gaming experience with.

Design Process and Exploration

Part Two



The design process and exploration are crucial steps in creating innovative and user-friendly products. The gaming controller was developed through a series of rigorous ideation and design, finding ways to take advantage of the Makey Makey Kit and Machine Learning applications such as "Teachable Machine."

This section will dive into the detailed description of the design process and exploration undertaken to create this innovative product.

The first step in the design process was ideation. As a team of avid gamers, we wanted to create a unique product that would enhance the gaming experience for users. We considered various ideas and requirements, from which body parts to use and ways of interaction, to the games we want to focus our product towards. We decided that we'd like to incorporate our whole body to try and make it as engaging and immersive as possible.

We also wanted to make sure our gaming controller could be used for a variety of games. And as we browsed through the abundant number of online games we grew up playing. We picked some of the many old classics that would accurately demonstrate the full capabilities of our game controller.

We start with the basics: how do you move in-game? What are ways in which you could control your movement in-game?
As we went through the various ideas, thinking about the basic and boring joysticks to camera-detecting motion, We wanted it to be simple and dependable to use while also making the gaming experience immersive and requiring you to move your body, so we decided why not just use our legs, and place some pads on the floor on which you can step on to move in the corresponding direction.

This implementation would be straightforward with the help of the Makey-Makey circuit. We borrowed some leftover laser cutout waste from the DI Lab that seemed like we could use instead of cutting new ones unnecessarily [reuse-reduce-recycle and all that :)].
We assigned a Cut-out for left, right, and jump, respectively, covering the pads in aluminum foil and connecting them to the circuit according to their respective controls.

inspired from the design of these arcade classics

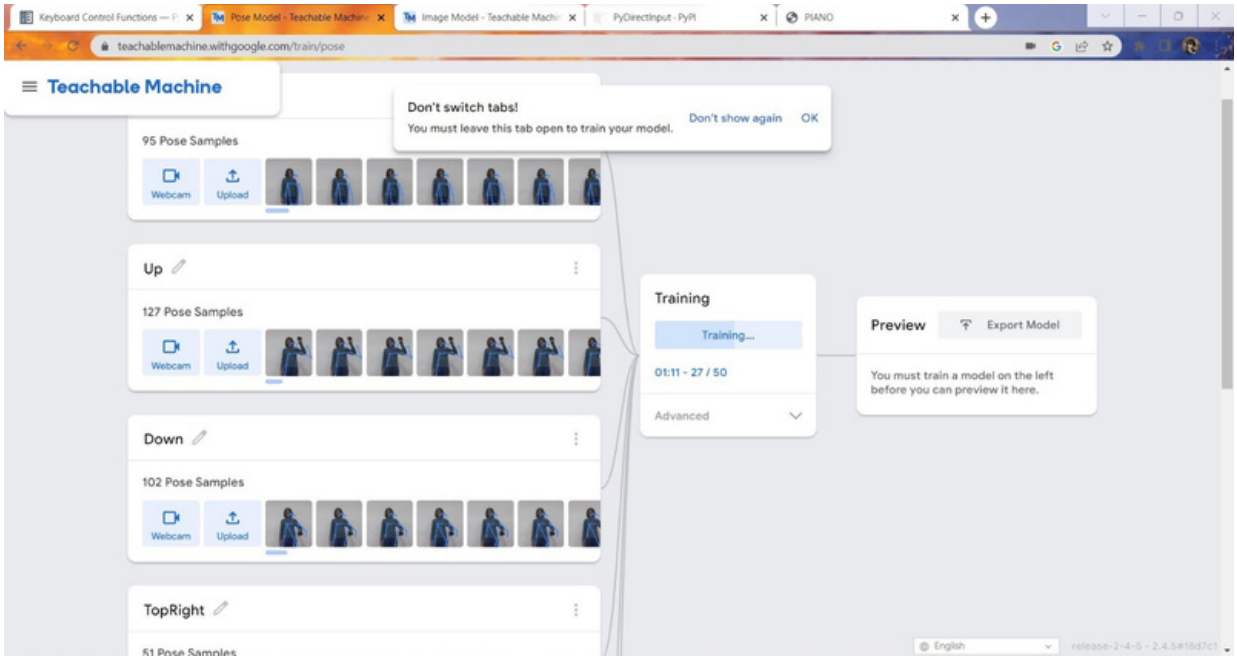
Next in line was a crucial part of the game controller. It was to figure out a way you would perform an action, say attack, open a box, or shoot in-game, and any other in-game interaction of that kind.

We went through many ideas on this matter, whether you would want the user to hold a cut-out of a gun, a sword, or maybe not hold anything at all. But in the end, we decided that we would want to make the user hold some type of object or wearable in his hand, which he could then interact with to shoot or attack in-game, which would make the experience much more immersive, making the user feel like they’re actually holding something they can use.

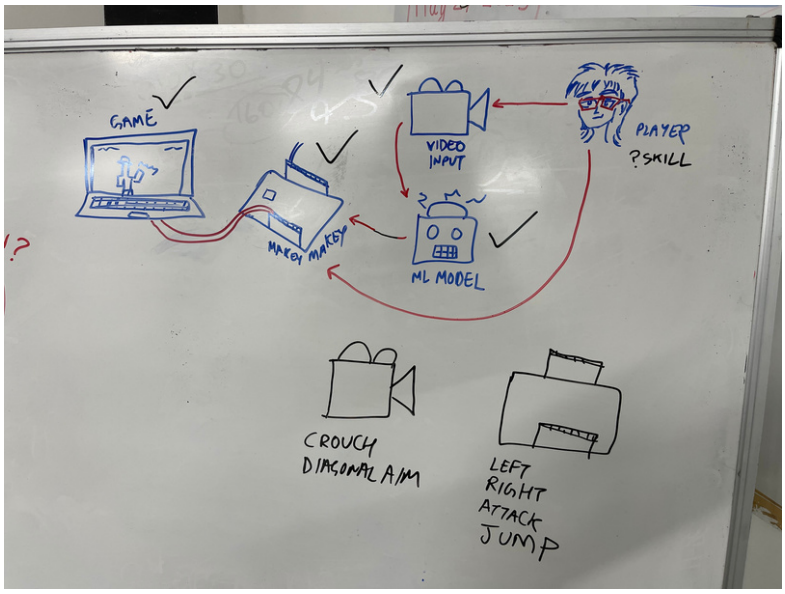
Now instead of making the item to be held specific, such as a gun or a sword, we wanted it to be something more generic instead. Since this controller could be used in many different games, it'd be a bad idea to force the shape of the item to look like one particular weapon that is too game-specific. Thus, we ended up deciding on making the user wear a brass knuckle-type cut-out that one could slip onto their fingers, and made the trigger action as clenching the hand. So whether it be shooting a gun, attacking an enemy with an axe, opening a chest or a door, clenching your fist onto the grip of the brass knuckle cut-out would perform that action.

We implemented the above idea by getting a laser cut-out of a brass knuckle and attaching the aluminium foil to the desired trigger point. We also used a glove with parts of it cut out to insulate some sections of the hand from getting into unnecessary contact with the trigger point, improving the performance and reducing the error of the prototype.

Next to plan out was how one would aim in-game. Here, the members unanimously agreed to use the camera to analyse and distinguish where the user is aiming and translate the same in-game. It was a novel concept that almost everyone had thought of in the past. We took the help of Teachable Machine and then wrote a Python script to make this possible. After continuous debugging and rigorous testing, we created a program that was able to accurately calculate the direction in which the user aims and make the in-game character aim the same way, thus incorporating full-body movement and making the whole experience immersive and fun.



While training the model to recognize the aiming directions, we also came up with the idea of training the model to recognize the user “crouching”, which is a famous control of the in-game character in various games. Thus, whether it’s a first-person shooter or a stealth game, the user has the intense experience of having to crouch in real life to crouch in-game.



We were happy with how all the components of our controller eventually synced to create a fully engaging and immersive experience with the games we played. Growing up playing games ourselves, it was a fun experience pouring all our ideas and inspirations together to create this product.

Heavily inspired by the experience of a VR game set and the recent strides of massive innovation that the gaming industry is taking in the direction of VR and AR. We tried to innovate at the forefront of this, a simple, inexpensive, yet effective product prototype that gamers can use to get an immersive experience.

Unfortunately, the final and major step that we couldn't execute was implementing a VR experience along with our gaming set. A fantastic idea that was discussed many times, despite hours of effort, either due to paywalls or mere software incompatibility, did not manifest. We weren't able to display the games in our cardboard VR headset to take the immersive experience to the next level.

Despite the shortcomings, we were finally able to design and develop a well-performing and versatile gaming controller that we were able to use and enjoy various games in ways we could never have before.

Ideation and Brainstorming Resources

Part Three

Platforms used :

- Teachable Machines

- Python Libraries such as :
  - OpenCV
  - Tensorflow
  - Numpy
  - Python TextToSpeech
- Easy Object Detection Article on hackster.io

- ChatGPT for reading Keras Code in Python

Links to the platforms used :

https://teachablemachine.withgoogle.com
https://www.tensorflow.org
https://pypi.org/project/opencv-python/
https://pypi.org/project/pyttsx3/
https://www.hackster.io/mjdargen/easy-object-detection-with-teachable-machine-python-d4063b
https://chat.openai.com/chat

Prompt 1

Novel Game Controllers Using Makey Makey Kit and Machine Learning

The Makey Makey kit is a tool that allows users to turn everyday objects into touchpads, which can then be used to control a computer or other electronic device. With the addition of machine learning, it's possible to create novel game controllers that respond to a wide range of inputs and provide a unique gaming experience.

Here are some ideas for novel game controllers using the Makey Makey kit and machine learning:

1. Fruit drum kit: Use different fruits as drum pads to play virtual drums. By using machine learning, the controller can recognize different fruit sounds and trigger different drum sounds accordingly.
2. Yoga mat game controller: Create a game that involves yoga poses, where the player has to hold a specific pose for a certain amount of time to score points. The yoga mat can be used as a touchpad, and machine learning can recognize different poses and provide feedback to the player.
3. Musical stairs: Turn a staircase into a musical instrument by using the Makey Makey kit and machine learning. Each step can be programmed to play a different note, creating a musical scale that the player can play by walking up and down the stairs.
4. Air guitar: Use a cardboard guitar as a controller to play virtual guitar in a game. By using machine learning, the controller can recognize different hand gestures and trigger different guitar sounds accordingly.
5. Gesture-based game controller: Create a game that responds to hand gestures, where the player can control the game by waving their hands in front of the Makey Makey kit. Machine learning can recognize different gestures and trigger different actions in the game accordingly.

Overall, the possibilities are endless when it comes to using the Makey Makey kit and machine learning to create novel game controllers. With some creativity and programming skills, you can create a truly unique gaming experience that's both fun and educational.

Prompt 2

Write a python script to detect to map an image detected to a key on the keyboard

To map an image to a key on the keyboard, we first need to detect the image using computer vision techniques. We can use the OpenCV library for this purpose. Once the image is detected, we can use the pyautogui library to simulate a keypress on the keyboard.
Here's a Python script that detects an image and maps it to a key on the keyboard:

```
import cv2
import numpy as np
import pyautogui

# Load the image
img = cv2.imread('image.png')

# Convert the image to grayscale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Load the template image
template = cv2.imread('template.png', 0)

# Find the template in the image
res = cv2.matchTemplate(gray, template, cv2.TM_CCOEFF_NORMED)

# Define a threshold
threshold = 0.8

# Get the coordinates of the template in the image
loc = np.where(res >= threshold)

# Loop through each match and simulate a keypress
for pt in zip(*loc[::-1]):
    pyautogui.press('a') # Replace 'a' with the desired key to be pressed
```

Note that this script assumes that the template image is a single object that is easily distinguishable from the background. If the template is more complex, additional preprocessing steps and more sophisticated computer vision techniques may be required.

Challenges and Difficulties

Part Four

One of the most significant challenges that we encountered during our project was bridging the gap between the model we generated through Teachable Machines, which relied on full-body movements, and the corresponding keyboard inputs we needed to map for our selected game.

Despite the wealth of available resources and documentation, we found it difficult to establish a clear link between the generated Keras code and labels and the actual movements we needed to detect through camera-based methods.

This presented a significant obstacle for our team, as we needed to ensure that our model could accurately detect and respond to user movements in real-time to control the game effectively. However, with persistence and extensive research, we were able to identify a document that provided guidance on connecting the Keras code to Python code, which we could then use to map the required keyboard inputs.
By utilizing this resource, we were able to bridge the gap between the trained model and the character movements in the game.

Subsequently, we encountered another challenge in bringing the control functions to our chosen game. Specifically, we found that while we could successfully press the Space bar for a jump in the real world using our Teachable Machine model, this action was not being translated to the game itself. Despite extensive searching, we were unable to find any relevant libraries or resources to resolve this issue.

However, through continued exploration and experimentation, we were able to identify a python library, which proved invaluable in enabling us to move the necessary functions to the actual app where they were required.

By leveraging these libraries, we were able to integrate the Teachable Machine model with the game and ensure that it could accurately detect and respond to user movements in real-time. Ultimately, these efforts allowed us to overcome these significant hurdles and achieve our project goals.

Similar Works

Part Five

- https://www.oculus.com/experiences/quest/

- https://www.amazon.in/Sony-Move-Motion-Controller-PS3/dp/B002I0J51U

- MSI Liberator programmable gaming foot pedal

Novelty

Part Six

Our product represents a significant innovation in the field of human-computer interaction and machine learning. By leveraging the capabilities of Teachable Machines, OpenCV, and Text-to-Speech libraries, we were able to create a unique solution that enables users to control a game using their body movements in real-time.

This approach represents a significant departure from traditional methods of game control, which typically rely on keyboard inputs or other forms of manual control. Instead, our product leverages the latest developments in machine learning and computer vision to enable users to interact with the game in a more intuitive and immersive way.

Furthermore, our product represents a significant step forward in the development of assistive technologies that can help individuals with mobility or motor impairments to interact with digital systems more easily. By providing a more accessible and intuitive method of game control, our product has the potential to improve the lives of individuals with disabilities and empower them to engage more fully with digital systems.

Overall, our product represents a significant novelty in the field of human-computer interaction, demonstrating the potential of machine learning and computer vision to create innovative solutions that can transform the way we interact with digital systems.

Individual Contribution

Part Seven

1. Aajay Ayyappan Devaraj - 2021001 (CSE)

- Design of the product
- Makey Makey Circuit building and Laser cut design of the product
- Program to access the gaming application and mapping inputs

2. Anshuman Bunga - 2021016 (CSE)

- Game selection
- Volunteering for the demonstration
- Amendments to the code of user input through the camera

3. Dhvanil Sheth - 2021040 (CSE)

- Program to take the video camera input and map it to gaming application
- Documentation (parts 4 to 8)
- Laser Cutting of products (.dxf file)

Collective Contribution

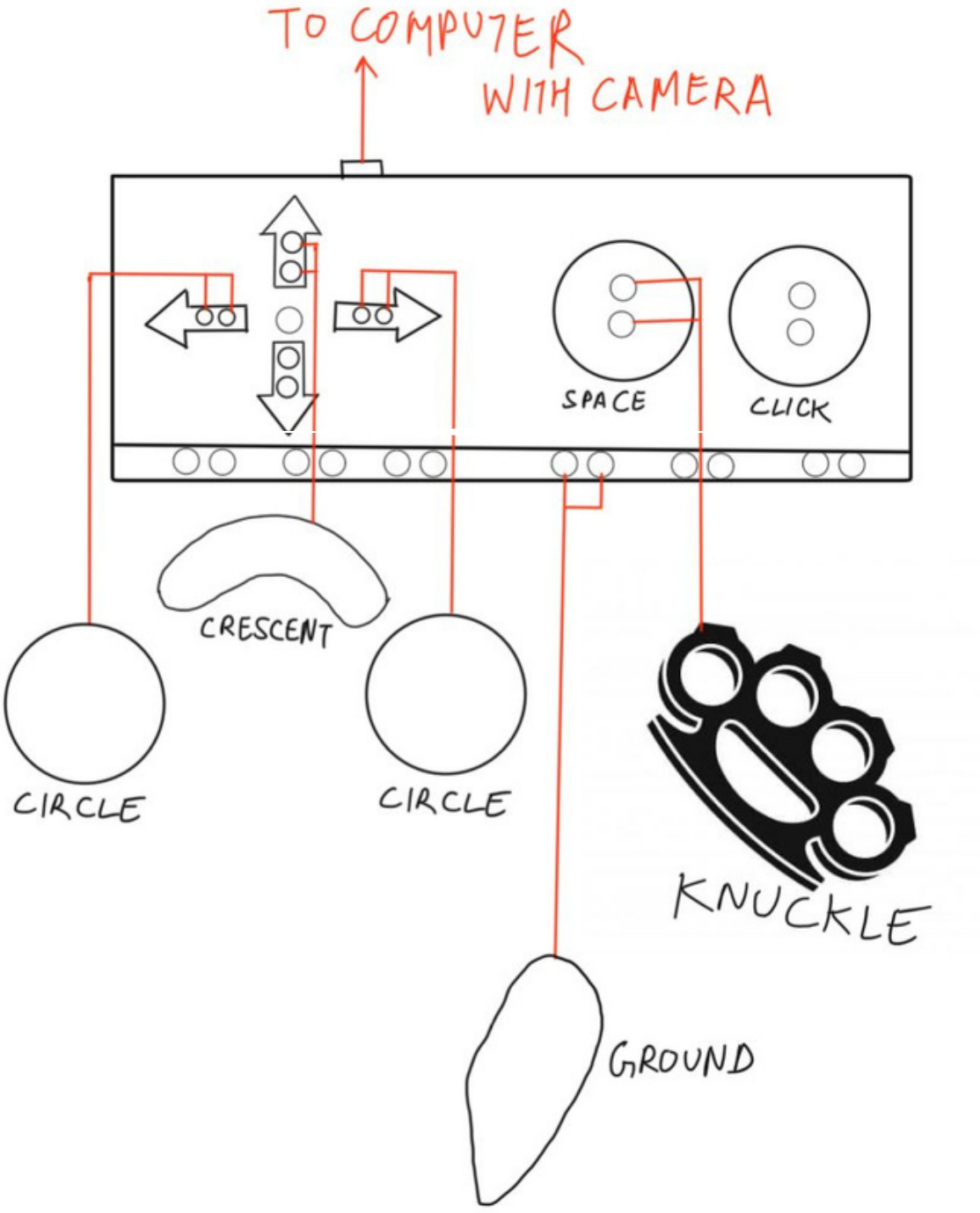
- Ideation of product
- Training Teachable Machine model
- Documentation

4. Shivam Dwivedi - 2021352 (CSD)

- NONE

Circuit, Code and Video Demo

Part Eight



Circuit, Code and Video Demo

Part Nine

Link to the video and Code :

Navigate to the file Group22 and click on the CameraDetection.py program to run the code on any laptop after the relevant libraries are installed

https://drive.google.com/drive/folders/1R0p_NpsrX2Ws8F5QKKfHkmmlqKtmKERG?usp=sharing

Thank You