EXERCISE GAME WITH POSE ESTIMATION

BY

Mr. Kobchok Paphanithisakul
Mr. Mohamed Arfan Mohamed Nayas
Mr. Pruetikorn Chirattitikarn

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF ENGINEERING IN ROBOTICS AND AI
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2022

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

PROJECT CERTIFICATE

Project Title	Exercise Game with Pose Estimation
Student Name	Mr. Kobchok Paphanithisakul
	Student ID. 62011141
	Mr. Mohamed Arfan Mohamed Nayas
	Student ID. 62011151
	Mr. Pruetikorn Chirattitikarn
	Student ID. 62011224
Degree	Bachelor of Engineering in Robotic and AI

Project Advisor Signed: _____

(Dr. Rutchanee Gullayanon)

Project Co-Advisor Signed: _____

(Dr. Pitikhate Sooraksa)

Project Title	Exercise Game with Pose Estimation
Student Name	Mr. Kobchok Paphanithisakul
	Student ID. 62011141
	Mr. Mohamed Arfan Mohamed Nayas
	Student ID. 62011151
	Mr. Pruetikorn Chirattitikarn
	Student ID. 62011224
Degree	Bachelor of Engineering in Robotics and AI
Project Advisor	Dr. Rutchanee Gullayanon
Project Co-Advisor	Dr. Pitikhate Sooraksa
Academic Years	2022

ABSTRACT

Numerous fitness games have flooded the market, offering users a fun and engaging way to exercise. However, these games often come with the added expense of specialized equipment. Our team's project aims to address this issue by developing an exercise game that eliminates the need for supporting hardware. Furthermore, this game is designed to detect body movements using a pose estimation model and a web camera. Python programming, specifically MediaPipe, is used to detect these movements and transfer the collected data through the Socket library to another section of the game. The Unity Game Engine is then utilized to create an immersive game environment and mechanics.

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to Dr. Rutchanee Gullayanon and Dr. Pitikhate Sooraksa, for inspiring and guiding us throughout the completion of our project. Even with all the modifications we made throughout the course, we were always given assistance.

We would also like to say thank you to our lecturers at King Mongkut's Institute of Technology Ladkrabang for supporting our work and believing in us throughout our journey to becoming engineers. Our project would not have been completed without all the knowledge we garnered from all of them and for that, we are incredibly thankful.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

Symbols/Abbreviations	Terms
AI	Artificial Intelligence
MP	MediaPipe
ML	Machine Learning
UI	User Interface

CHAPTER 1
INTRODUCTION

1.1 Background and significance

Throughout our studies and experiments in the field of robotics and AI engineering, we have learned several technologies regarding robotics and human interaction. From our experience and interest in digital creation, our members decided to create a project that involves creating digital content with the utilization of robotics and AI subjects to deliver a product that users can control with minimal equipment. Furthermore, the growth of the computer vision field is substantial which has led to us applying it in our project.

1.2 Research objectives

The development of exercise games has dramatically risen, there are several games related to muscle movement and calorie reduction which are published in the market, for instance, Ring Fit Adventure, and Wii Sport. However, they require additional hardware components, excluding computers and web cameras, to play their games to the extent. Thus, we planned to create an exercise game that utilizes only computers and web cameras to control the program's mechanism. Combined with AI, the program will apply a technique called pose estimation.

1.3 Scope of research

1. Design a playable game and set up a game mechanism.
2. Detect a body movement and body parts' location.
3. Control game mechanics with body movements.

1.4 Methods of conducting research

The research is primarily conducted in 3 phases, designing, building, and testing. In the design phase, we focus on utilizing the most suitable game engine that creates the game efficiently and supports AI algorithms. The build phase develops the game system, graphics, and program that uses AI to detect body location. Continuously, the testing phase is to receive feedback from the users and use the results to improve the game features.

1.5 Expected Benefits

Regarding the completion of the game development and testing phase, we expect our project to expand with many added features afterwards. In other words, it can include more poses and body movements to train muscles and burn calories more effectively. Furthermore, the inclusion of new maps and characters will always get the users excited and wanting more. With our objectives, the project can successfully reduce costs for users and detect body movement clearly.

CHAPTER 2
REVIEW OF LITERATURE

2.1 Related Theory

2.1.1 MediaPipe

MediaPipe is a framework developed by Google for building real-time multimedia processing pipelines. These pipelines consist of modular processing units called calculators that perform specific tasks. MP provides a set of pre-built calculators covering various multimedia processing tasks, and developers can also create their own custom calculators using libraries like TensorFlow or OpenCV. The framework supports the integration of machine learning models for tasks such as object detection or pose estimation. To use MP, you define the pipeline structure, configure the calculators, and provide input data, allowing the pipeline to process the data and generate the desired output results. Overall, MP offers a flexible and scalable platform for creating real-time multimedia applications.

Since it is a large framework, and we only needed the pose estimation aspect of it we used MP's Pose Estimation feature to achieve our goal. This is a feature of the MP framework that enables real-time human pose estimation from video or camera input. It uses a machine learning model trained to detect and track the positions of various body joints, such as the head, shoulders, elbows, and knees. By analyzing the spatial relationships between these joints, MP Pose Estimation can accurately estimate the pose of a person in the video. This information can be used in applications like motion tracking, gesture recognition, fitness analysis, and virtual try-on. MP Pose Estimation provides an efficient and accessible solution for developers to incorporate real-time pose estimation capabilities into their multimedia applications.

In computer science jargon, a graph consists of Nodes connected by Edges. Inside the MP Graph, the nodes are called Calculators, and the edges are called Streams. Every stream carries a sequence of Packets that have ascending time stamps.

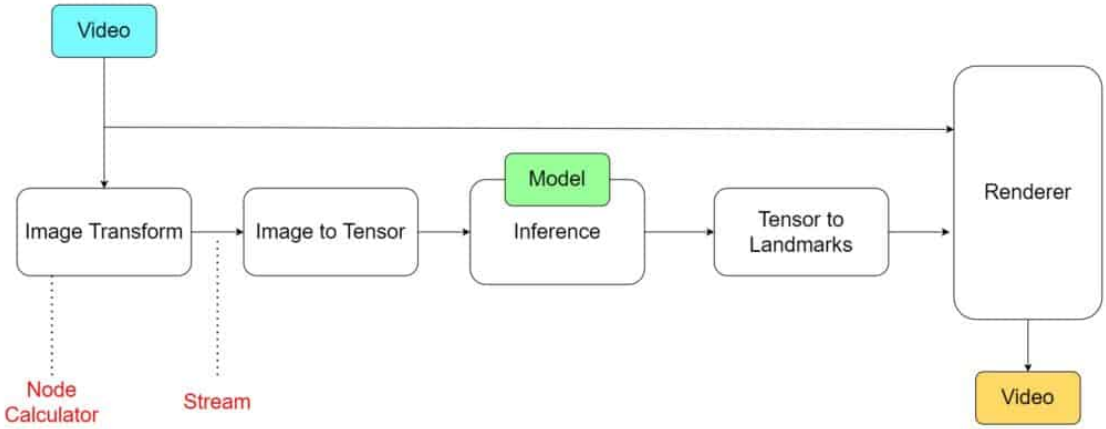


Figure 2.1 MediaPipe Solution Graph

In Figure 2.1, we have represented Calculators with rectangular blocks and Streams using arrows.

(1) MediaPipe Calculator

MediaPipe utilizes calculators as specific computation units responsible for processing tasks. These calculators, written in C++, handle the packets of data (such as video frames or audio segments) that enter and exit through designated ports. During initialization, a calculator specifies the packet payload type that will flow through the port. When a graph runs, the Framework executes the Open, Process, and Close methods within the calculators. The Open method initializes the calculator, the Process method runs repeatedly as packets enter, and the Close method finalizes the process after a complete graph run.

(2) Calculator types in MediaPipe

As an example, consider the first calculator shown Figure 2.1. The calculator, Image transform, takes an image at the input port and returns a transformed image in the output port. On the other hand, the second calculator, Image to tensor, takes an image as input and outputs a tensor. All the calculators shown above are built-in into MP. We can group them into four categories

(2.1) Pre-processing calculators are a family of image and media-processing calculators. The Image transform and Image to tensors in the graph above fall in this category.

(2.2) Inference calculators allow native integration with TensorFlow and TensorFlow Lite for ML inference.

(2.3) Post-processing calculators perform ML post-processing tasks such as detection, segmentation, and classification. Tensor to the landmark is a post-processing calculator.

(2.4) Utility calculators are a family of calculators performing final tasks such as image annotation.

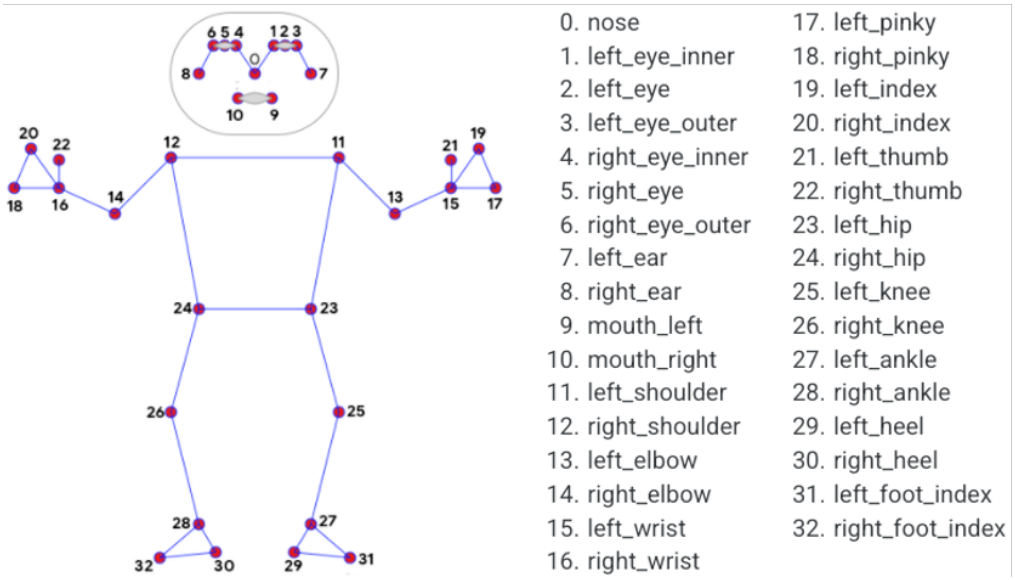


Figure 2.2 MediaPipe Pose Estimation KeyPoints

2.1.2 OpenCV

OpenCV (Open-Source Computer Vision Library) is a popular open-source library in Python for computer vision and image processing tasks. It provides a wide range of functions and algorithms to handle image and video data, including image manipulation, feature detection, object recognition, and camera calibration. OpenCV supports various image and video file formats and allows for real-time video capture and processing. It also offers tools for image filtering, transformation, and geometric operations. With its comprehensive set of functions and easy-to-use Python bindings, OpenCV is widely used in fields such as robotics, augmented reality, facial recognition, and surveillance systems, making it a powerful tool for computer vision applications.

2.1.3 Socket (Python)

The socket package in Python provides a low-level interface for network communication. It allows Python programs to create sockets and establish network connections for sending and receiving data over different protocols, such as TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). Sockets can be used to create both client and server applications, enabling bidirectional communication between them. The socket package offers functions and classes for configuring socket settings, binding sockets to specific addresses and ports, sending and receiving data, and handling various network-related errors. It serves as a foundation for building network applications in Python, facilitating the implementation of network protocols and communication between different devices and services.

2.2 Related Technology

2.2.1 Software

2.2.1.1. Unity

Unity is a cross-platform game engine developed by Unity Technologies and released in June 2005. The engine has since been gradually extended to support a variety of desktop, mobile, console, and virtual reality platforms. The engine can be used to create 2D, 3D, VR, and AR games or applications, as well as interactive simulations and other experiences. In this software, the coding language used for development is C#.

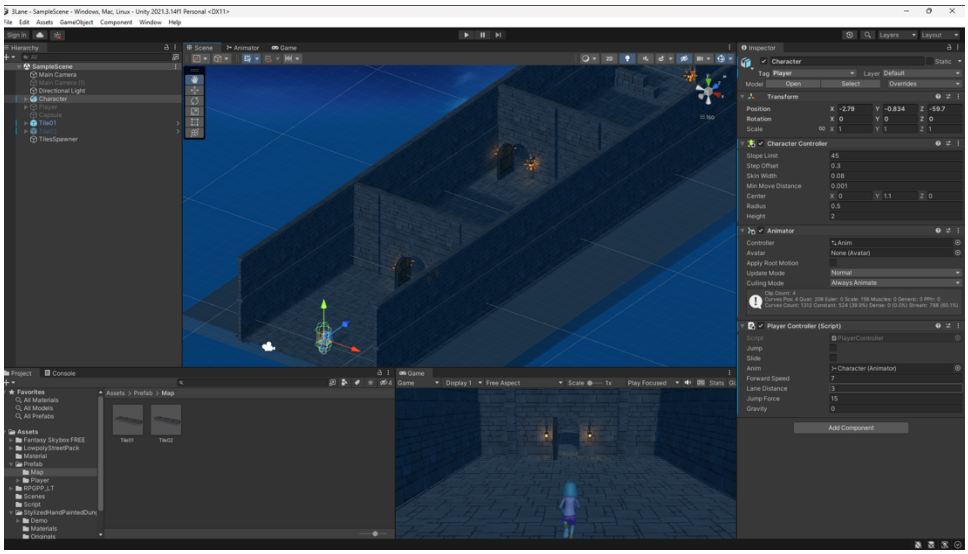


Figure 2.3 Create stage scene by using Unity

2.2.1.2. Python Programming

Python is a high-level, interpreted programming language known for its simplicity and readability. It emphasizes code readability and provides an extensive standard library, making it versatile and suitable for a wide range of applications. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming. It has a large and active community, with numerous third-party libraries and frameworks available for various domains such as web development, data analysis, machine learning, and more. Python's syntax and structure make it easy to learn for beginners, while its scalability and flexibility make it a preferred choice for professional developers.

2.2.1.3. MediaPipe

MediaPipe is an open-source libraries and tools for artificial intelligence and machine learning solution. It is contributed by Google developers for users to apply artificial intelligence model into the application swiftly and customize independently. It can be utilized across multi-platform and several frameworks, for instance, JavaScript, Android, and iOS. Furthermore, it has a wide range of subdivision task of each MediaPipe depended on purposes.

Figure 2.4 Result from Mediapipe

2.2.1.4. User Datagram Protocol (UDP)

UDP is a communication protocol used to establish fast and tolerant connections between internet applications. It allows for quick data transfer without waiting for confirmation from the receiving party, making it useful for time-sensitive communications like VoIP, DNS lookup, and media playback. UDP is an alternative to TCP, with differences such as enabling process-to-process communication and operating in a best-effort mode without guarantees or retransmission features. UDP provides port numbers for request differentiation and an optional checksum for data integrity verification.

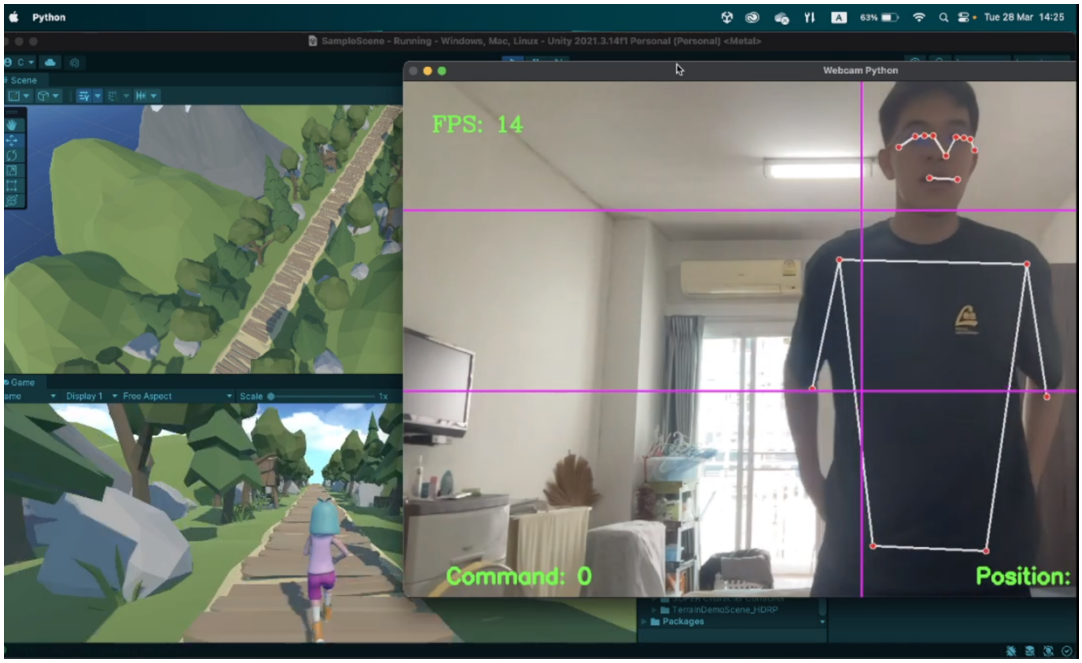


Figure 2.5 Connection between MediaPipe to Unity

2.2.2 3D Object Model Packages

2.2.2.1 RPG Poly Pack Lite

This is one of the Unity assets that we used to create our project. This package provided us with the necessary prefabs and textures to develop the maps of our game. Furthermore, some of the obstacles present in our game used the available models from this package.

Figure 2.6 Map environments (Forest)

2.2.2.2 RPG Monster Duo PBR Polyart

This is one of the Unity assets that we used to create our project. This package provided us with animations as a moving mechanism for our characters in the game, these include running, jumping, slide and walk. We also used some of the materials and prefabs to further improve our game.

Figure 2.7 Monster in the wild

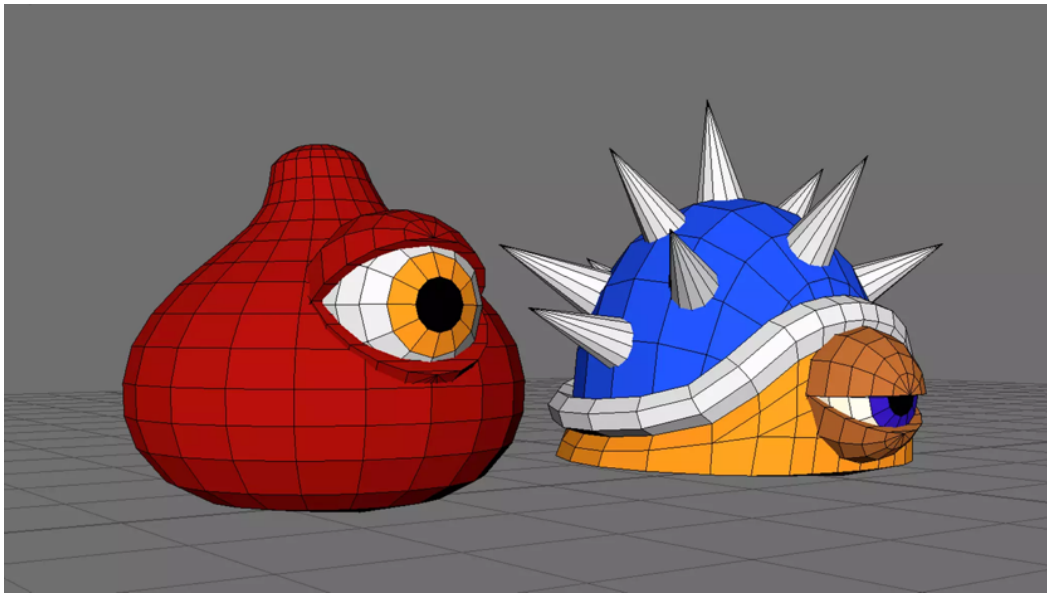


Figure 2.8 Object from the RPG Monster Duo Package

2.2.2.3 StylizedHandPaintedDungeon

We used this asset to create a castle-like map environment that look, where players are required to choose the correct door to avoid obstacles and go through the next stage.

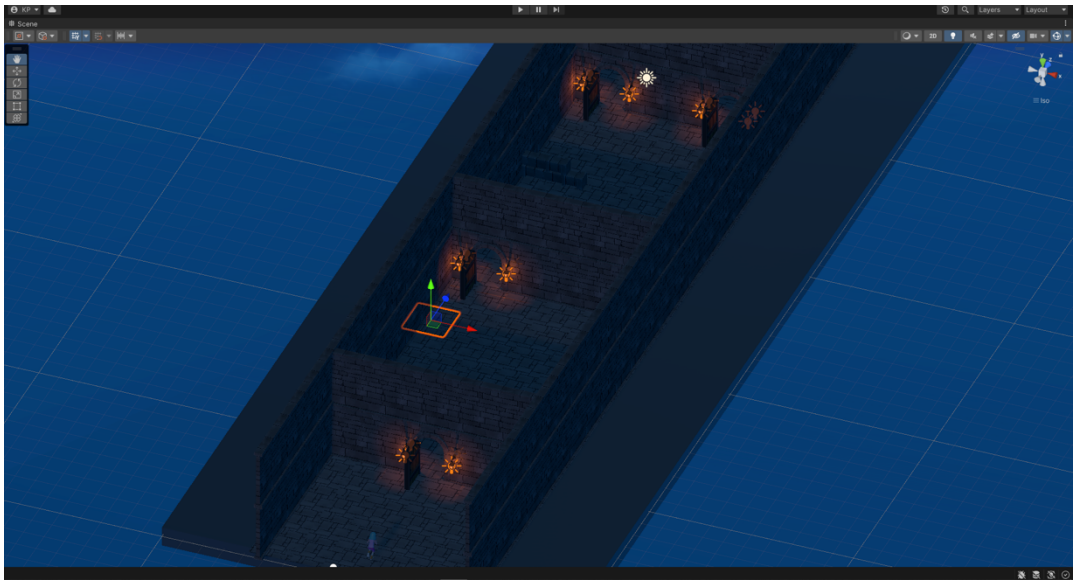


Figure 2.9 Map environment (Castle)

CHAPTER 3
METHODOLOGY

3.1 Introduction

In this chapter, we included a deeper insight into the process of our project design and development. The initial phase goes into the steps taken in regard to the project plan. In the further section, we describe more about the problems we have encountered and resolved for. Furthermore, we discuss the development and design of other types of game for exercising.

3.2 Interesting Problems

To discover a meaningful purpose for our project, we plan to tackle existing problems that are affecting the gaming industry, especially games with body motion tracking. In the following sections, we will be discussing the problems:

3.2.1 Additional Cost

Most existing exercising games in the market require additional hardware components to support game mechanics. One such example includes Ring Fit Adventure that needs a controller called Nintendo Switch, Wii Sport, and specially made cameras or cameras with sensors to detect distant objects. These components are of high price, which some users with sufficient budget find difficult to justify.

3.2.2. Precision

Some exercise games have a particular issue, where they are unable to detect body motion properly. Based on few reviews, users sometimes cannot fully control their game character, or the game processes the wrong command to a character, which results in frustration for the users.

3.2.3. Inaccessibility

Many people find it difficult to include exercising a daily part of their schedule due to many reasons. Plenty of locations do not have a gym or fitness area in proximity for people to travel to every day. Most gyms and fitness areas are also very crowded, making it very difficult for people to perform all the necessary exercises, in addition to the lack of available equipment.

3.2.4. Lack of Motivation

Many people simply do not go to gyms because it involves and takes a lot of users' motivation just to get up and workout. Exercising in public areas can seem very discouraging and lonely for many people as it only involves yourself for the most part.

3.2.5. Time Efficiency

Most people also find it difficult to have time to meet their weekly exercise needs. This can be because of reasons such as tight schedules, busy with family, and work, etc.

3.3 Project Planning

Our proposed solution was to solve earlier mentioned problems and for that we created an exercise game that uses machine learning models to predict and detect body part locations, including body movements. The prediction then will be processed to control game mechanics and game characters. In this section, we will be discussing the preparation phase of the project, including design, development, and testing phase.

3.3.1 Preparation phase

This is the beginning phase of our project where we discussed the tasks and steps that we should be setting out to complete. The month of Febuary 2023 was to assemble and think of possible ideas and features that our app should be composed of in the final stages of development. This phase included the discussion of utilization between game engines, software programming, and artificial intelligence models.

We determined that Unity Engine, and Python programming would be suitable most for our needs in the app as it had the features to create a game and support machine learning processes respectively. Moreover, the model that is incorporated in the project is MediaPipe, an open-source machine learning model developed by Google. It provides libraries and tools to support machine learning solutions and has several machine learning rojects, including pose estimation.

Furthermore, for the preparation of the exercise game, we concluded that the 3D models for maps and characters are downloaded from the website and Unity Package Store. Due to the limited time and budget, this decision was able to save budget and time, and, in the market, there are handful of model assets.

Lastly, we decided on creating a computer or desktop app because it is very easy to integrate and there is a large number of users already in the market. Moreover, a desktop app is suitable for testing and observing.

3.3.2 Design and Development phase

During the months of March and April of 2022, we started designing and developing the Unity game app. The creation of our app involves us separating our workflow to ensure our final product comes to fruition quickly. This meant we had to divide it into steps, which are listed below, that we need to complete in order.

1. After selecting the software program and programming language, we needed to code the algorithm to detect body movement by Python programming language. Since Unity game engine does not fully support machine learning libraries, we used Python programming, imported with MP, and sent the data through local network.
2. We then need to either design our own 3D model or find the right 3D model on the web. Then, we imported models into Unity game for designing and generating maps.

3. The third step involved us creating a full function that is able to control character with inputs from keyboard initially and spawned maps and obstacles endlessly.
4. In the fourth step, we primarily focused on communication between Python program and Unity game. We implemented it with the local network interface, or socket.
5. After the completion of the network communication, we started the development of game mechanics, where users can control the character with body movement and body location from model prediction. In initial design, we expected the Python program to determine the state of actions among standing, jumping, and squatting.
6. The final step involved us designing the user experience where users understand the state during gameplay and after the game over, and refining the UI of the app so users can easily understand how our app functions.

3.3.3 Testing Phase

The project's integration testing phase took up the first two weeks of May 2022. This period was preoccupied with us ensuring every feature in our app works uniformly and cooperatively. We also started fixing bugs that occurred when using different panels in our app.

3.4 Game Development

For the full completion of our game, we had to use Unity Engine to develop all the gaming elements. We have gone into further details about all the development aspects of our game that were needed.

3.4.1 Maps

We decided to create maps into 2 stage first is 1 lane in forest theme and second one is 3 lanes in castle theme.

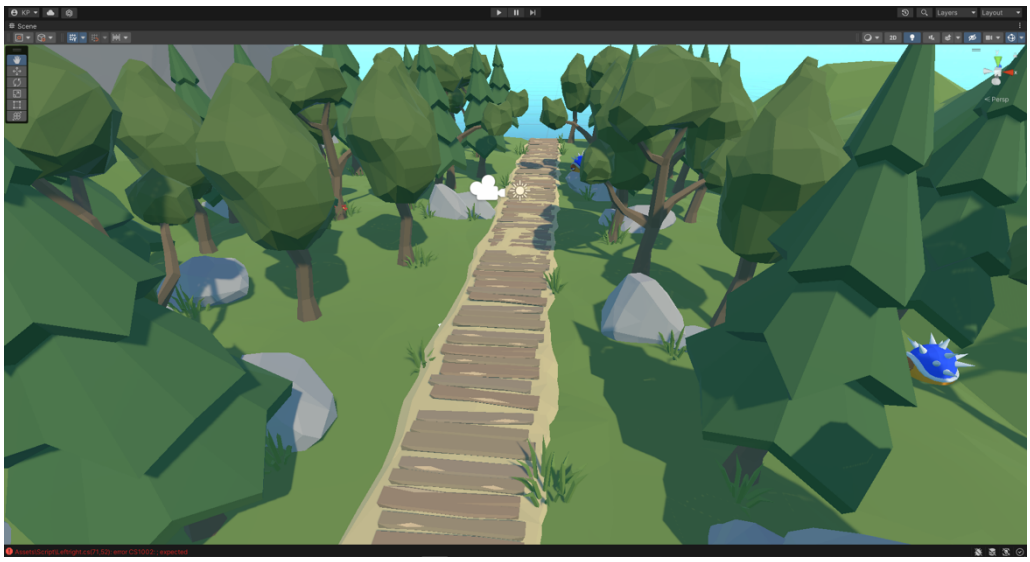


Figure 3.1 Forest stage

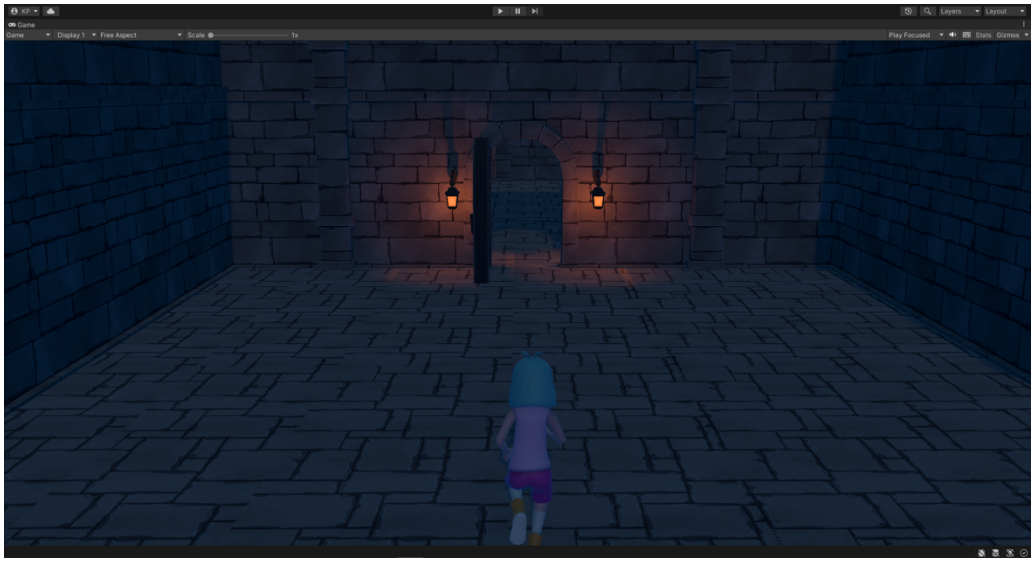


Figure 3.2 Castle stage

Both stages have different difficulty levels. For the Forest stage player will use actions jump and squat for dodging obstacles, whereas, in Castle stage player need to move right and left for changing the lanes and also jump and squat for dodging obstacle.

3.4.2 Character Development

For the character model and animations, we downloaded from Adobe Mixamo because we wanted to focus on gameplay and the character control process, moreover, we used ML to detect human bodies to control the character while moving through the scene environments.

3.4.3 UDP Connection

Since we needed data to be transferred from the Python script to Unity so that the character can be controlled, we set up an UDP connection. This was done by following the steps down below:

1. Setting up the UDP Connection:

In Unity, we created a C# script that handles the UDP communication. This script created a UDP socket and bind it to a specific port for communication. We also specified the IP address and port of the Python script's UDP server.

2. Sending Data from Python:

In our Python script, we imported the socket module and created an UDP socket. We then used the send to method to send data to the IP address and port that our Unity script is listening on.

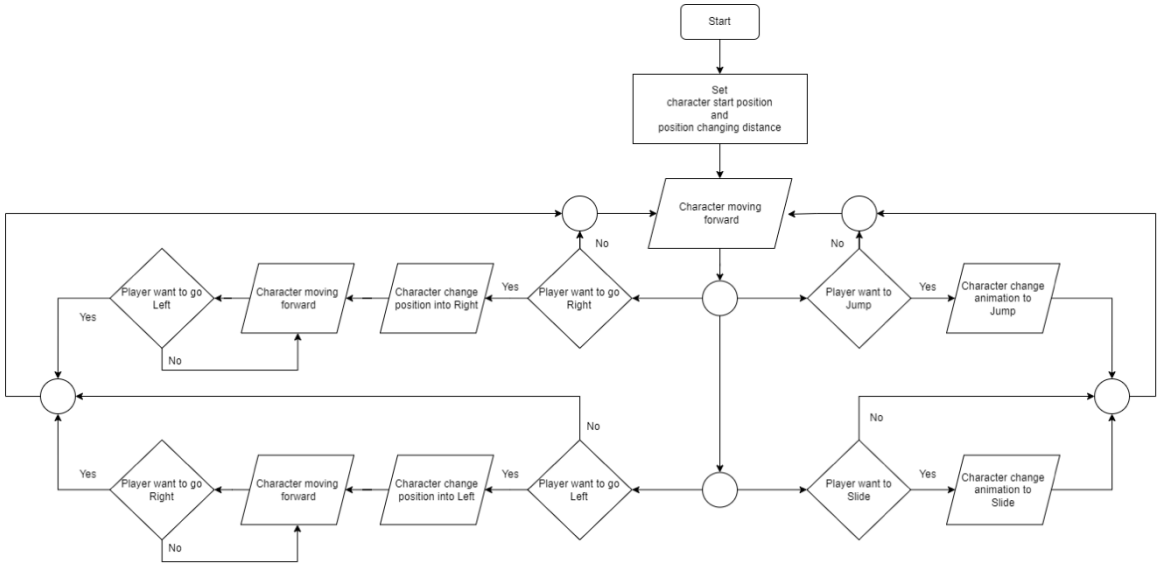
3. Receiving in C#

Receive data in Unity (C#). In the UDP server script in Unity, we used the socket's Receive () method to listen for incoming data. Processing and handling the received data within our Unity application as needed.

3.4.4 Programming Script

3.4.4.1 PlayerController Script

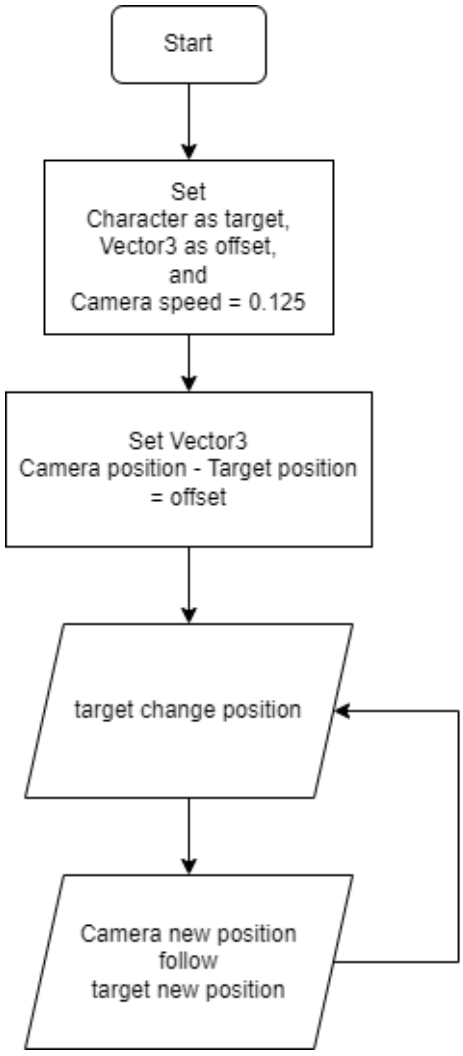
Initially, the character is set to be located at a certain position, or the center of pathway. Then, the character moves toward a Z axis and the script will receive an input from the data. If the data describes that users move to the left side or right side, the character's movement will follow the command. Furthermore, if it receives that player jump and character will follow the command, in addition player's squatting.



Figures 3.3 Flowchart show PlayerController script

3.4.4.2 CameraController Script

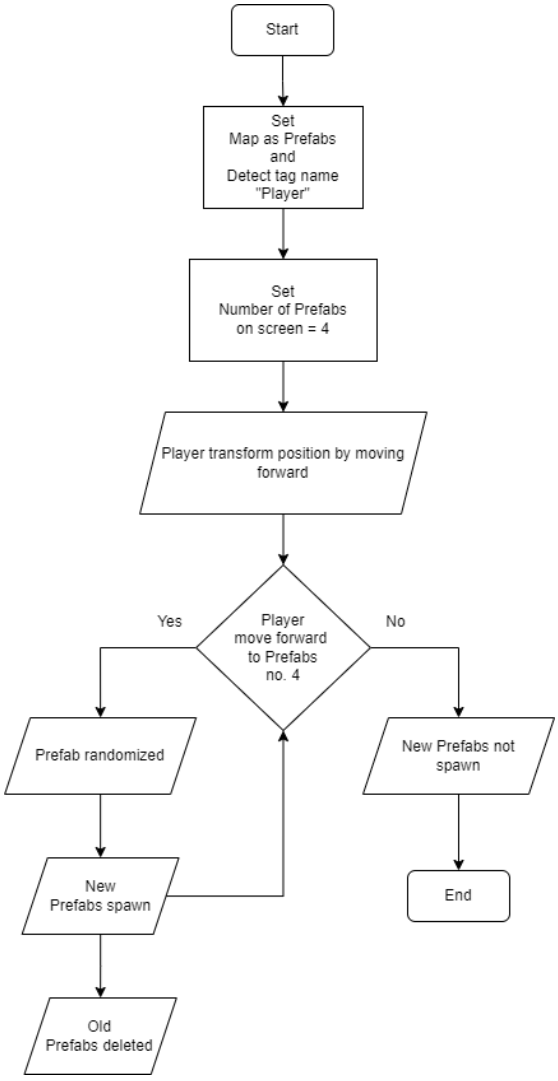
The game camera is set to be positioned behind the character with the offset defined in the program. When the character moves forward, it will follow through a constant speed.



Figures 3.4 Flowchart show CameraController script

3.4.4.3 MapSpawning Script

The map in game scene is initially spawned for 4 times. After the character moves forward, the script will receive the character's position. If the character's position is at the fourth map, the script will spawn the next one infinitely until the character stops running.

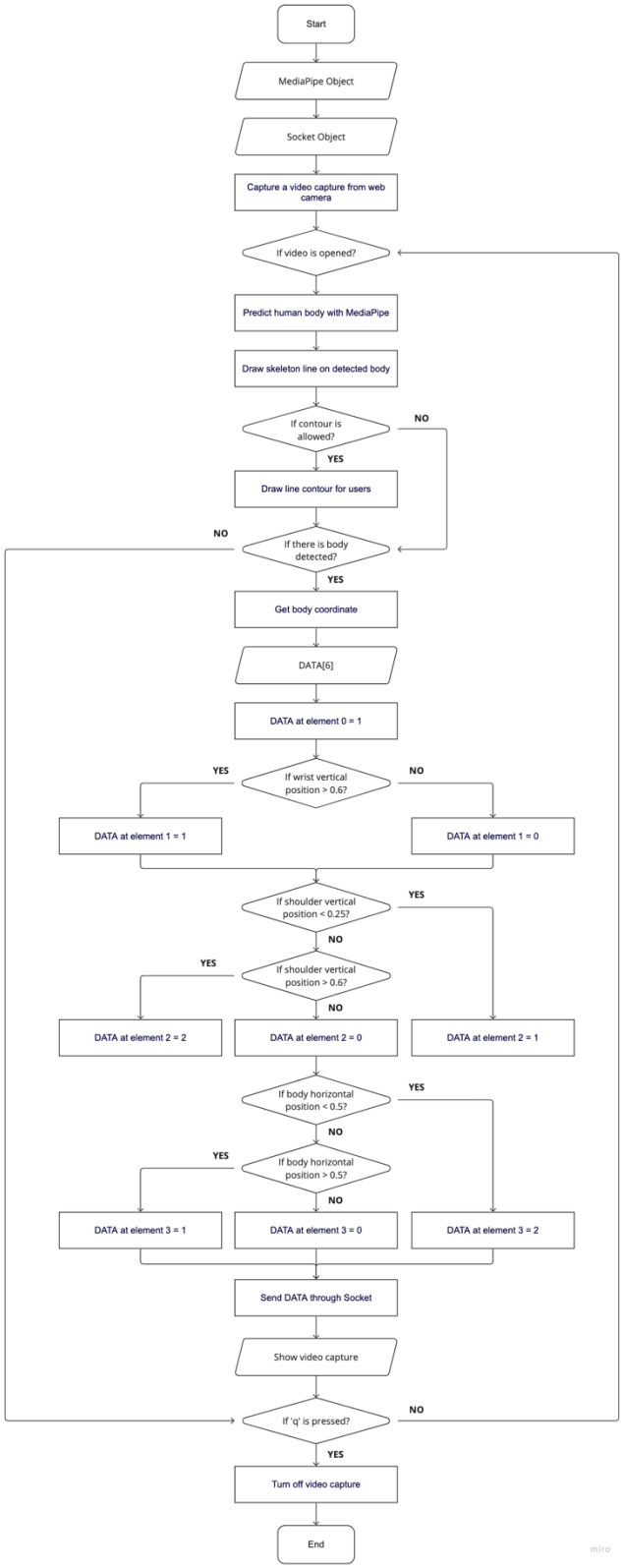


Figures 3.5 Flowchart show MapSpawning script

3.5 Python Script

In this script, its main function is to detect body position and send data to Unity game. The program starts by defining MediaPipe object and Socket package. It then initializes the web camera and, the ML model reads the frame to detect the body. If the body is perceived by the program, it will draw a skeleton line, aligned with the body. The data is defined in an array, and it is used to collect the body states.

The first element of an array is to determine whether it is detected or not, and number 1 indicates that it is detected. The next element represents the vertical location of the body. The value is set according to the state where body position is, among standing, jumping, and squatting. And lastly, the third element indicates the state of horizontal position. If the body stays at the center point, the value is set to 0, and set to different numbers if the body does not. Finally, the array is converted to data to send to Unity game.



Figures 3.6 Flowchart show Python script

3.6 Summary

In this chapter, we discussed the methodology involved in the completion of our testing phase. We discussed the problems our target market are facing and how to solve them. Furthermore, we outlined the project plan that we followed throughout and the app design of our project.

CHAPTER 4
EXPERIMENTAL RESULT

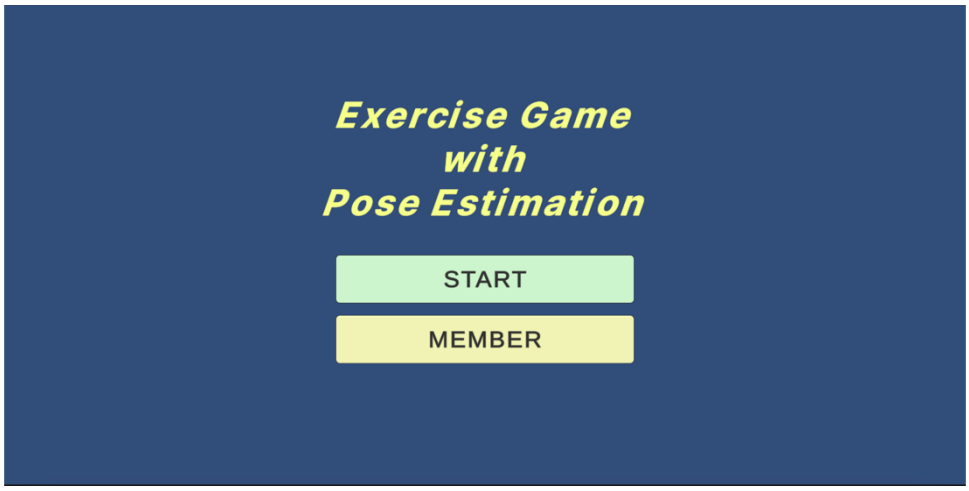
4.1 Introduction

In this chapter, we will be giving a deeper insight into the final version of the application that we created. We will be showcasing the UI of our application and other modifications that we had completed for it to be ready to be deployed. In the further section, we will be going through the evaluation process for this project.

4.2 Unity Scenes

4.2.1 Start Menu Page

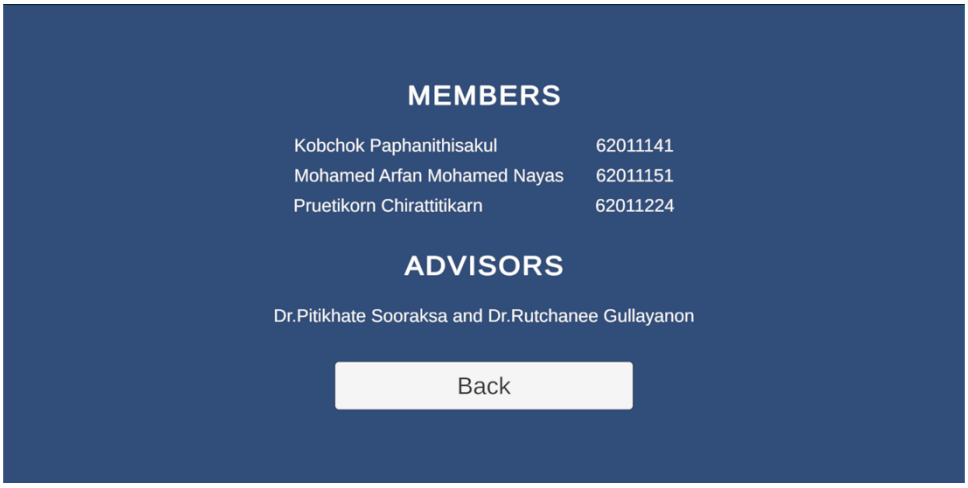
On the Start Menu page, we created the buttons which transfers the screen to other scenes. The first button, or upper one, is a Start button, which opens the Map selection page, and another one switches to Member page. On top of the buttons, the title of the project aligns at the center of the screen.



Figures 4.1 Application's Start Menu page

4.2.2 Member Page

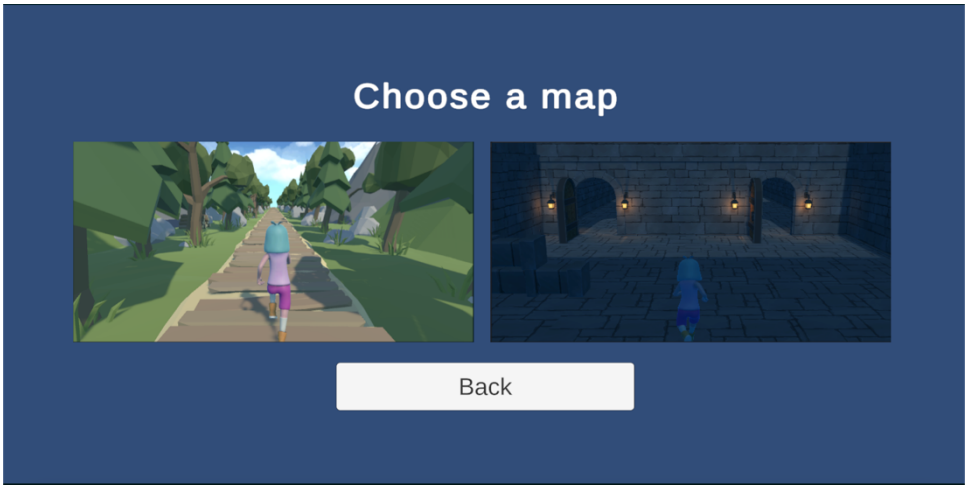
On the Member page, we created a screen where it mentions the list of members who have been involved in the project development.



Figures 4.2 Application's Member page

4.2.3 Map Selection Page

For the Map Selection page, we created buttons so users can choose between two maps. There are 2 maps in the projects, one at a forest-like environment and the other at a castle-like environment. At the bottom of the page, there is 'Back' button which leads users back to Start Menu page.



Figures 4.3 Application's Map Selection page

4.2.4 Gameplay page

When users press the map button, the screen will switch to the main gameplay. On the screen, the game character will be placed at the center, surrounded by 3D environmental objects. There is a pathway below the character to showcase the running track. The score, created to give users mental prize and encouragement, is shown at the top left conner and increases as the time users have been alive. The screen at the gameplay page is shown in distinguished states depending on the data received from Python code and game state.

Figure 4.4 Application's Gameplay page

4.2.4.1 Game over scene

When the character collides with the obstacles, the Game Over scene will appear and hide the gameplay page. The words 'Game Over' are aligned at the center of screen to clarify game state to the users. At the bottom of the screen, there are 2 texts to guide users to the next steps afterward. When users press 'r' on the keyboard, it will restart the gameplay page, and pressing 'space' will switch page to Start menu page.

Figure 4.5 Application's Game Over scene

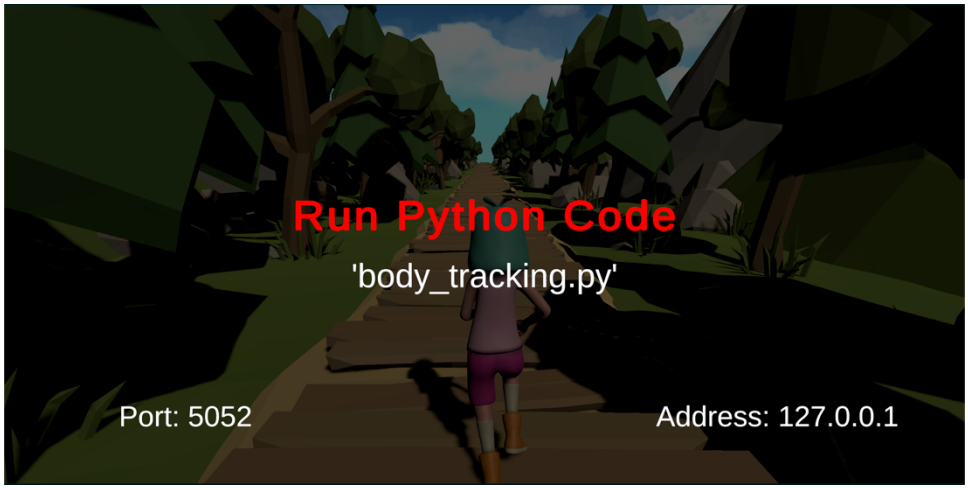
4.2.4.2 Countdown scene

At the start of the game, the countdown scene will appear before entering the gameplay page. It is used for users to prepare themselves for exercise before playing and confirm that they have executed the Python code. The timer will count down for 5 seconds and, later, the scene continues to gameplay page.

Figure 4.6 Application's Countdown scene

4.2.4.3 Warning scene

If users have not executed Python code yet or the application does not receive the data from Python code, the warning scene will appear first and pauses the gameplay page. On the screen, it is mentioned the name of the Python file that needs to be executed, including port number and address for the data receiver. When the application receives the data, the screen will disappear.

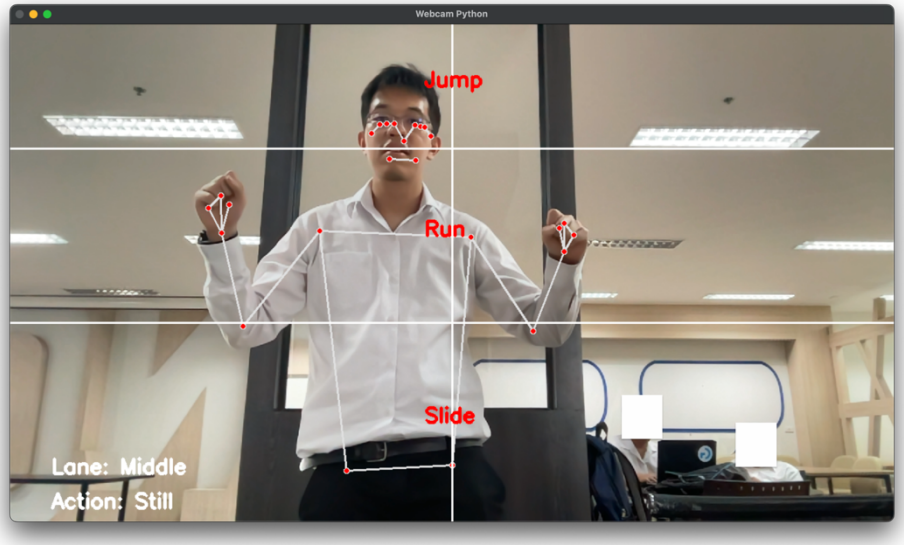


Figures 4.7 Application's Warning scene

4.3 Python Code with MediaPipe

In the beginning of the code, the Python code will define the objects of socket connection and MP model for sending data and detecting body gestures respectively. It initializes the web camera, and the ML model predicts the locations of body parts. The results will be saved in an array to send the data through Socket library.

After executing the code, a video will be presented on the screen. If the body is detected, it will show a skeleton line on the screen, aligned with the body. On the screen, the horizontal lines indicate the state of the movement, among running, jumping, and squatting. The vertical line represents the position of the body horizontally. At the bottom left conner, there are 2 texts mentioning the horizontal position and the movement state of the body.



Figures 4.8 Video Capture resulted from Python Code

4.4 User Feedback

4.4.1 Product Tester Feedback

Our product tester is an acquaintance of one of the project members. He is familiar with game and application development. He has been using our product since the first draft, and his role has been important to our project's development. After finishing the project, he gave his impression of the project by mentioning that he felt he exercised during the game and was left tired after playing.

1.) Recommendation

"After I experimented with this game during the development period, I felt the game was very good. It helped with exercises such as increased heart rate (cardio) and also really helped with weight loss, because after the experiment came to an end, I felt I had more energy and my weight was slightly reduced, but I would like you to try to add more new features because the ones at this moment are too few and may make players feel bored easily."

2.) Tester Information

Name: Jirawat Kodchaporn

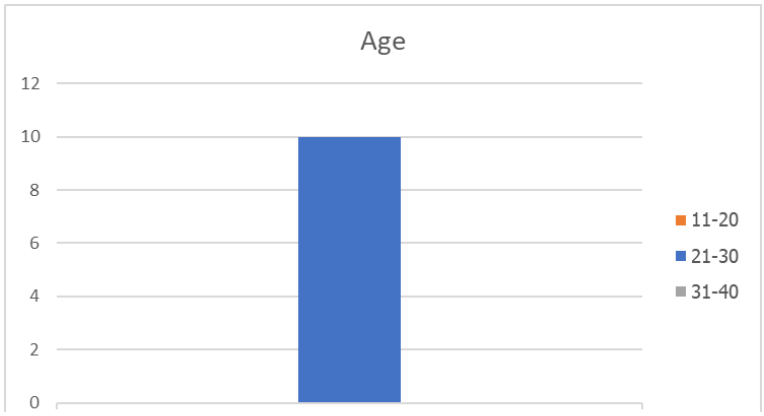
Email: job_jirawat@hotmail.com

Telephone Number: (+66) 90 025 0737

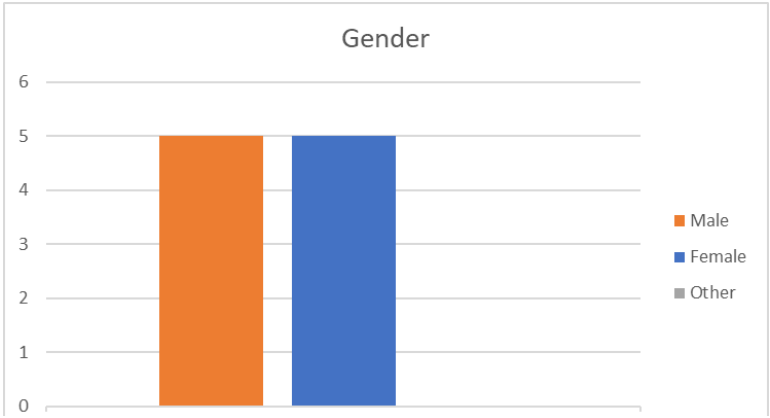
4.4.2 Survey

We created a survey to receive feedback from users who have played our game. According to the graph, there are 5 males and 5 females, whose ages range from 21- to 30-year-old, participating in this survey, and the bar charts show positive result.

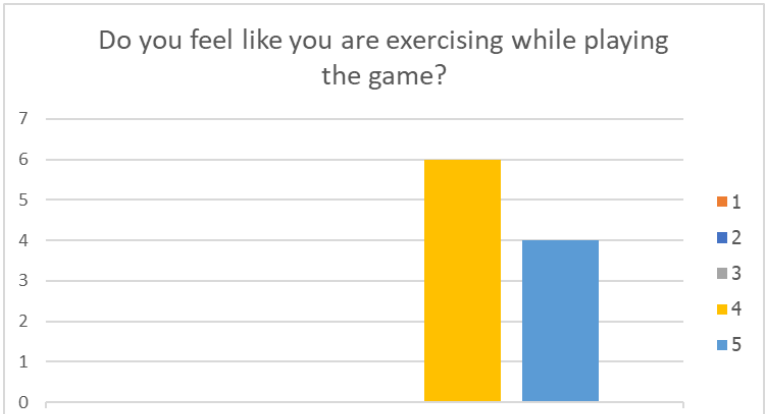
The survey is composed of 6 questions related to application's benefits and user experience. The questions inquire about users' satisfaction of the end-product and receives score in linear values ranging from 1 to 5. The value of 1 represents the most negative feedback. On the other hand, the value of 5 means users satisfied most with projects' features. According to the bar charts below, most users are pleased with our project as most of them selected a value of 3 or more, referring to positive opinions. Furthermore, around 90 percent of participants would like to play our game again.



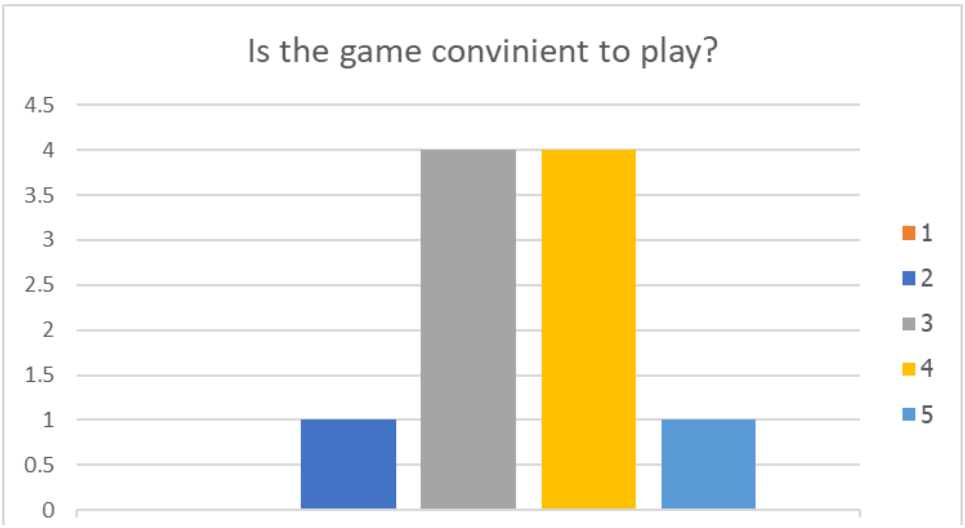
Figures 4.9 Chart show feedback from testing user 1



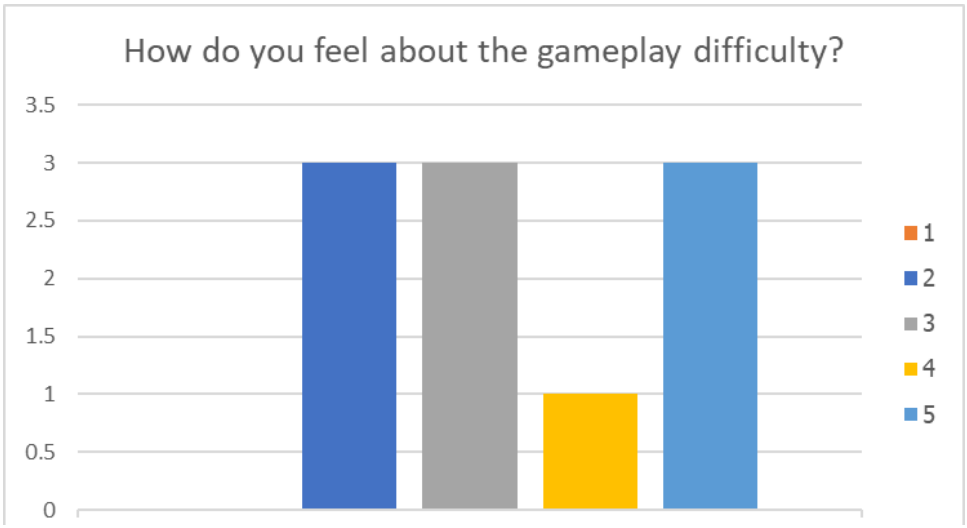
Figures 4.10 Chart show feedback from testing user 2



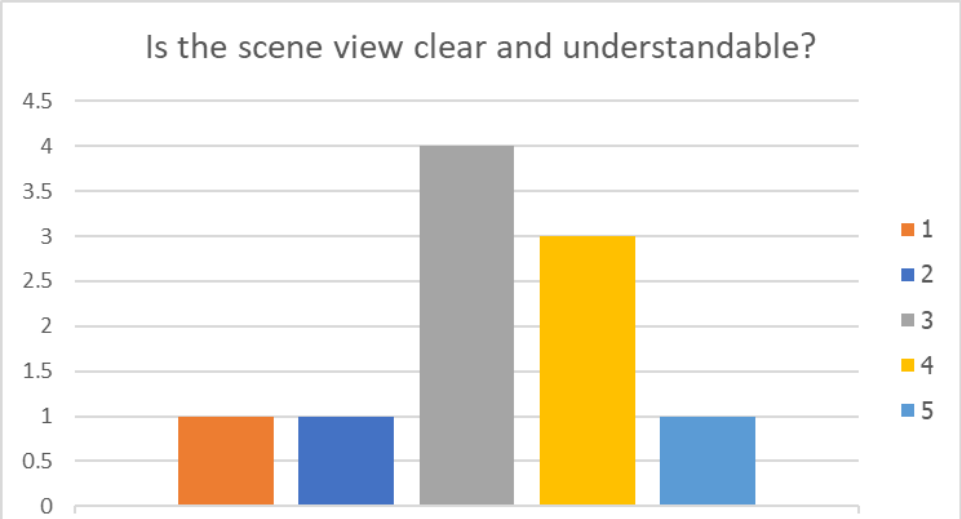
Figures 4.11 Chart show feedback from testing user 3



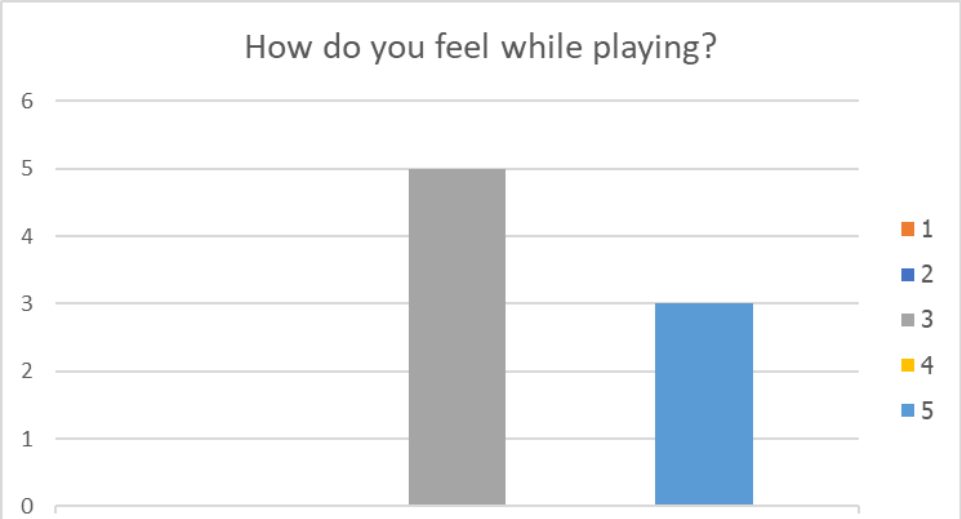
Figures 4.12 Chart show feedback from testing user 4



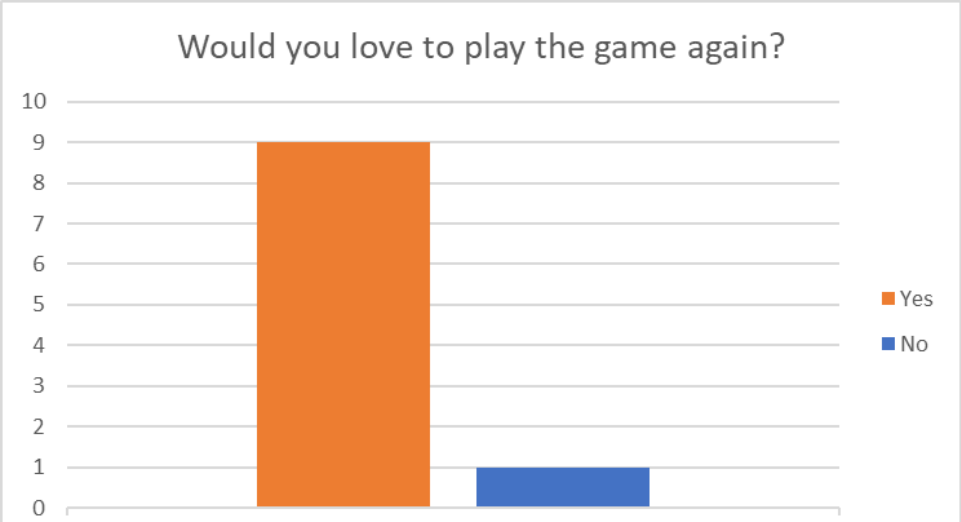
Figures 4.13 Chart show feedback from testing user 5



Figures 4.14 Chart show feedback from testing user 6



Figures 4.15 Chart show feedback from testing user 7



Figures 4.16 Chart show feedback from testing user 8

CHAPTER 5
CONCLUSION

5.1 Introduction

In this chapter, we have summarized and described our work in each chapter of the project, then we wrote conclusions about how our work have developed, and recommendations to increase the development's flexibility and speed.

5.2 Summary

Chapter 1

Description of our background, significance, and research objectives.

Chapter 2

Description of theories related to our project and technology that we have used in our project.

Chapter 3

Description of the problems we are trying to find a solution for, the project planning, and the development phase of our app.

Chapter 4

Description of the Unity scenes in our project, the features in our app, and the user interface.

Chapter 5

Outline of all the chapters and description of possible recommendations and additions to our existing project.

5.3 Conclusion

Throughout the course of this report, we went into further details of our project, starting from the thought process to the completion of our application. We have provided the necessary details to allow the readers to understand the depth and backend of our project.

5.4 Recommendation

Before this app becomes deployable to users, we recommend the addition of features and obstacles to increase efficiency of exercise. In other words, we should add functionalities such as better body movement detection and include varied types of obstacles, to support more exercising poses and increased calorie reduction. Furthermore, using Unity game for gameplay and Python code for ML is inconvenient in practical use, hence, we need to combine game functions and pose estimation in one single app in the future.

REFERENCES

Python Software Foundation (April 15, 2023). The Python Tutorial. Retrieved from: https://docs.python.org/3/tutorial/index.html

Kukil, LearnOpenCV (Mar 1, 2022) *Introduction to MediaPipe.* Retrieved from: https://learnopencv.com/introduction-to-mediapipe/

Balaji, S. (Oct 17, 2020) *Introduction of Game Engines*. Retrieved from: https://medium.com/techiepedia/introduction-to-game-engines-8dd8cd8af03e

TechTarget (Oct, 2021) *Definition of User datagram protocol*. Retrieved from: https://www.techtarget.com/searchnetworking/definition/UDP-User-DatagramProtocol#:~:text=User%20Datagram%20Protocol%20(UDP)%20is,provided%20by%20the%20receiving%20party.

Gaudenz Boesch, Viso.ai (2023). *MediaPipe: Google's Open-Source Framework for ML solutions (2023 Guide).* Retrieved from: https://viso.ai/computervision/mediapipe/#:~:text=Computer%20Vision%20Teams-,What%20is%20MediaPipe%3F,currently%20in%20alpha%20at%20v0

Python Software Foundation (April 15, 2023). *Socket – Low level networking interface*. Retrieved from: https://docs.python.org/3/library/socket.html

[Murtaza's Workshop - Robotics and AI]. (March 25, 2022). *3d Hand Tracking in Virtual Environment | Computer Vision* [Video]. Computervision.Zone. https://www.youtube.com/watch?v=RQ-2JWzNc6k&t=2195s

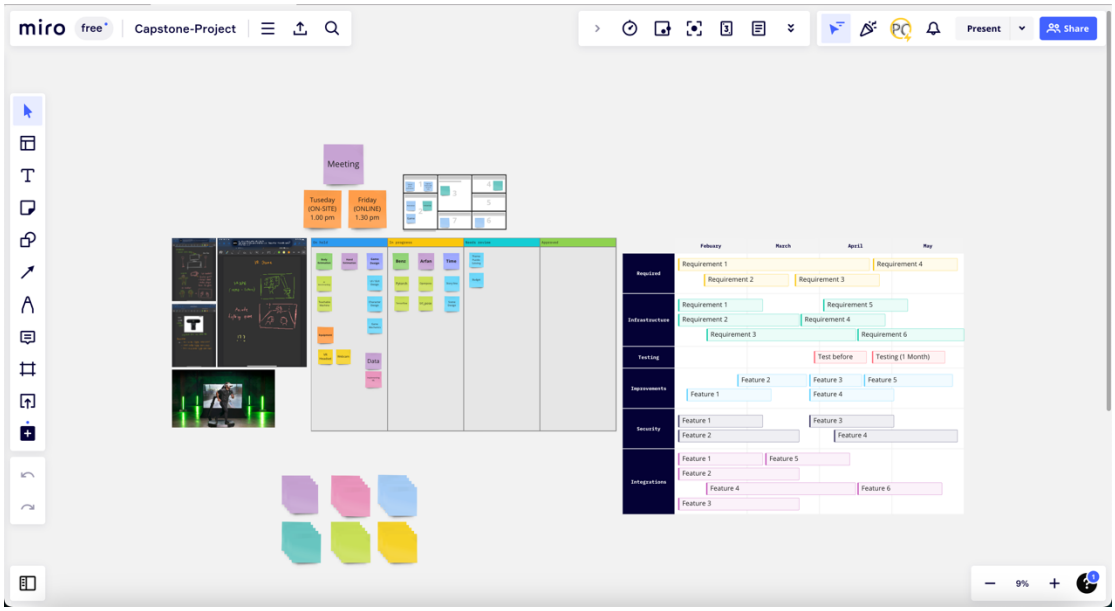
Doxygen (2023). *Introduction to OpenCV-Python Tutorials*. Retrieved from: https://docs.opencv.org/4.x/d0/de3/tutorial_py_intro.html

Google for Developers (2023). *MediaPipe | Google for Developers*. Retrieved from: https://developers.google.com/mediapipe

APPENDICES

APPENDIX

Overview of project planning and game design



BIOGRAPHY

Kobchok Paphanithisakul

Kobchok Paphanithisakul is currently a Robotics and AI student in the Faculty of School of International & Interdisciplinary Engineering at King Mongkut's Institute of Technology Ladkrabang (KMITL) in Thailand. He joined an internship for 2 months as a software developer at ASEFA to develop Face recognition and a 360-degree VR factory tour.

Mohamed Arfan Mohamed Nayas

Mohamed Arfan Mohamed Nayas is currently a Robotics and AI student in the Faculty of School of International & Interdisciplinary Engineering at King Mongkut's Institute of Technology Ladkrabang (KMITL) in Thailand. He has been a Teaching Assistant for numerous courses taught at KMITL. He obtained an internship at CMKL University for 4 months, where he garnered more experience in Machine Learning and Artificial Intelligence. With this, he continued his journey in gaining more experience and landed another internship at AIT AI Center to further improve his skills.

Pruetikorn Chirattitikarn

Pruetikorn Chirattitikarn is currently a Robotics and AI student in the Faculty of School of International & Interdisciplinary Engineering at King Mongkut's Institute of Technology Ladkrabang (KMITL) in Thailand. He has been a Teaching Assistant for numerous courses taught at KMITL. He has an experience of an internship at Japanese College for 1 month and at CMKL University for 6 months. Continuously, he enrolled an exchange program to study computer science at Belgium University, Thomas More University.