Pick and Place Game with Mobile App using Robotic Arm

Interim Report

Chung Pui Yin 3035569160
Abstract

This project aims to make a robotic arm controlled by a mobile app to play Reversi Chess with a human player. Although there are many different robotic arms in use in the industry including product packaging and sports, this project aims to make robotic arm together with an interacting mobile app for entertainment. The program gets the chess game information by using computer vision called OpenCV for calculation of the best move. The program sends signal to control the robotic arm through Arduino board which takes input from the program and return output to the robotic arm. Currently, the manual control of the robotic arm has been completed by interacting with the mobile app. Then, extracting game information through the smartphone camera, automatic control and home positioning of the robotic arm will be implemented. Limitations and difficulties are encountered, a flipping mechanic is required to play Reversi chess which may involve external tools.
Table of Contents

List of Figures  I

List of Tables  II

1. Introduction  P.5-7
   1.1 Robotic Arm and Mobile App  P.5
   1.2 Objective and Deliverables  P.6-7
   1.3 Report Outline  P.7

2. Methodology  P.4-6
   3.1 Mobile App  P.7-8
   3.2 Robotic Arm  P.8-9
   3.3 Arduino  P.9
   3.4 Game  P.9-10

3. Results and Discussions  P.10-12

4. Schedule and Milestones  P.12-13

5. References  P.14
List of Figures

Figure 1: OpenCV separate the chess board into grid P.7
Figure 2: Principle to pick up and put down object P.9
Figure 3.1: Initial state of Reversi chess P.10
Figure 3.2: Available positions for white chess player P.10

List of Tables

Table 1 Schedule and milestones for the project progress P.12
1. Introduction

The following chapter introduces this progress report. Firstly, a brief background of the robotic arm and mobile app is presented. Then, the objective and deliverable are introduced. Finally, the outline of the report is listed.

1.1 Robotic Arm and Mobile App

There are different kinds of robotic arm and robotic tools to help making jobs easier nowadays. However, there is not much on the field of entertainment for humans. Currently, most of these robotic components are designed to help with humans’ work like packaging products in factories, performing certain step in the progress of making a product. Also, there are some but not much on interacting and entertaining humans. For example, a robotic arm had been made to play table tennis with human which can detect the ping pong ball and hit back the ball like a normal player. However, these are just on the level of production and sports entertainment.

Due to the fast progression of technology in the modern world, smartphones and tablets are very common. Apps installed are designed to be convenient and able to finish all kinds of tasks with simple controls. A large variety of apps are available on the internet including games, tools, social platforms. However, there is not much on the real-world environment interaction. Also, a significant advantage of using app is that it is portable. For example, while waiting for a bus, most people tend to wait while using apps on smartphone instead of waiting while using a computer. In this project, the product to me made which is the robotic arm controlled by the mobile app can act as a real player to play games with a human. The app can get the real-world information through the smartphone camera and control the robotic arm to pick up and place down objects like chess.
1.2 Objective and deliverables

In this project, the final product is a robotic arm and a mobile app for android phone. There are several functionalities that the robotic arm can perform which are movement, picking up an object and putting down an object. There also several functionalities the mobile app can perform which are capturing the game state, calculating the action to be performed by the robotic arm, sending signals to the robotic arm and acting as a control panel for the user to control the robotic arm manually.

The robotic arm can be controlled manually or automatically by the mobile app to move along 3 axes of directions in the 3D plane (X, Y, Z) which is forward, backward, left, right, upward and downward. For manual control, the user can access the mobile app to control the robotic arm by tabbing some virtual buttons in the app including moving, the robotic arm, grabbing and releasing object. For automatic control, the phone must be placed at a place which can capture the game board and the game objects clearly in order to identify the game state. The app will send signal to the robotic arm to control its actions including moving to a position, grabbing an object and releasing an object. The two modes can be switched by tabbing the buttons in the app.

The mobile app contains several functionalities, including capturing the game state, calculating the best move, sending signal to the robotic arm and acting as a control panel in manual control mode. Therefore, the device must be installed with a working camera. The camera will keep capturing the game board and game objects so that the app program has access to the current game state. Then, the program will calculate the best move according to the rules of that game which is pre-coded in the program. Therefore, the games available for this mobile app and robotic arm is limited. After the calculation
finishes, the corresponding controls will be sent to the robotic arm to perform movement and pick and place actions. For manual control mode, the app will display a control panel which contains virtual buttons for the user to control the robotic arm directly.

1.3 Report Outline

After the introduction, Chapter 2 states how the mobile app interacts with the robotic arm, followed by the game to play chosen for the project. Chapter 3 states the results and discussion of the app and robotic arm including difficulties encountered and future work. The schedule of the project is listed in Chapter 4 including the progress and future planning. Finally, Chapter 5 concludes the project.

2. Methodology

There are 2 main parts of this project which are the programming part of the mobile app(software) and the robotic arm(hardware).

2.1 Mobile app

For the programming part, the mobile app is coded using Android Studio in Java language. The app functionalities include a manual control mode to control the robotic arm manually, automatic mode to capture the game state and control the robotic arm remotely. To control the robotic arm using the app, socket connection must be established first. Then, the mobile app can send a string type message in G-code format as command to control the robotic arm by writing to output stream to the socket. The manual control is implemented with simple button interactions by Java coding to send signals to the robotic arm. When the user presses the “Open” button, the app will
establish the socket connection with the Arduino board. The “Steppers” dropdown menu controls the state of the steppers before moving the robotic arm. After establishing the connection and turning on the steppers, the movement panel including buttons to move the robotic arm in X, Y and Z directions allows the user to move the robotic arm in each axis for a short distance or a long distance. For automatic control, the game state is captured using the smartphone camera. Then the image is analysed using OpenCV which is a tool to help detecting and identifying various objects. For this project, it detects the chess board and identifies the location of the chess objects. For example, in the following picture, OpenCV separates the chess board into a grid. Such information is used by the app to calculate where to pick and put down the chess.

![Figure 1: OpenCV separate the chess board into grid](image1)

2.2 Robotic Arm

For the robotic arm, it is 3D printed and assembled with other mechanic parts like motors. It picks up objects by using air pressure but not friction. When the gripper of the robotic arm which is connected to a tube is placed on the object, the vacuum turns on to draw out the air inside the tube. It results in the air inside the tube has lower pressure compared to the air outside the tube. Since high pressure flows to low pressure, the air outside tends to flow inside the tube. Hence, the object is pressed against the tube. When the gripper moves...
upward, the object will be lifted. Reversely, when the vacuum turns off to balance the air pressure inside and outside the tube, the only force acting on the object is gravitational force. Therefore, the chess falls. The robotic arm receives commands from the mobile app through Arduino board to perform actions.

![Figure 2: Principle to pick up and put down object](image)

2.3 Arduino

The key for communication of the app and robotic arm is the Arduino board. The robotic arm itself only performs action and Arduino board receives commands. The Arduino board is installed with a Bluetooth chip to connect to Bluetooth devices. The app connects to the server socket of Arduino board via Bluetooth. Then the socket listens to G-code command string messages from the app and return physical outputs like turning on the steppers to move the robotic arm or turning on the vacuum to pick up objects.

2.4 Game

Another important element of this project is the game. As stated in the project title, the mobile app controls the robotic arm to play a game with the player. This project focus on playing the Reversi chess. It is a turn-based game with
one player playing against another player. Initially, there are 4 chesses on the chess board as shown in figure 3.1. The rule to place a chess in the player’s turn is that there must be at least one opponent’s chess placed between the turn player’s chesses. For example, figure 3.2 shows the possible locations (outlined in red) for placing the white chess. After placing the chess, the opponent’s chesses between the turn player’s chesses turns into the turn player’s chesses (flip them to change the colour). The game ends if both players cannot place any chess on the chess board, including the situations that the whole grid is filled or there are only chesses of one player.

![Figure 3.1: Initial state of Reversi chess](image1)

![Figure 3.2: Available positions for white chess player](image2)

### 3. Results and Discussion

The following includes the progress made so far and the limitations and the difficulties encountered.

#### 3.1 Initial Findings

For current progress, user interface design and manual control of the robotic arm has been finished. In semester 1, most of the time in participating the FYP
is used for learning Android Studio and coding. The user interface including interactable buttons is done by using Scroll View to store the constraint layout to display the buttons. By pressing the buttons, the user can set up connection to the server socket of the Arduino board, manually control the movements of the robotic arm and control it to pick up and put down object.

3.2 Limitations and Difficulties Encountered

While playing Reversi chess, players always need to flip the chesses after placing a chess. For the current robotic arm, the gripper is not designed to flip an object since it is supposed to move the object without changing its orientation. Therefore, external tools may be required without modifying the mechanics of the robotic arm. The prototype has been designed but testing is required for better modification. Figure 4 shows the cross-section of the tool. The chess flips after travelling the tunnel.

Another difficulty is that flipping is another frequently performed action in Reversi chess game. However, the robotic arm may not be able to perform such action quickly so it may take quite a long period of time.

3.3 Remaining Work

First, computer vision is required for the app to know what happens on the chess board. OpenCV will be implemented to analyse the image captured by the smartphone camera. The information returned will be used to calculate
where to pick up and put down the chesses. OpenCV is an open-source library to analyse image. For example, it can separate the chess board into grid so location of the chesses can be identified by the app.

Second, an AI is required to play the game as a real player. Minimax algorithm and alpha beta pruning will be implemented to calculate the best moves. Minimax algorithm is a backtracking algorithm which calculates all the possible results. Then, it backtracks to decide the best results. The process is separated into 2 steps which is minimum and maximum. For minimum, the least value is chosen which represents the opponent player makes the optimal move which is a disadvantage to the AI player. Opposingly, the maximum step chooses the greatest value which means that the AI player makes the optimal move which is an advantage to the AI player.

Finally, the robotic arm will be enhanced to implement automatic home positioning functionality. By adding end stop switches to the robotic arm, the starting position of the gripper can be recorded and the app will detect the end stop signal to determine whether the gripper is at the starting position.

4. Schedule and Milestones

Here is the schedule and milestones for the project progress.

<table>
<thead>
<tr>
<th>Date</th>
<th>Milestones</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/10/2021</td>
<td>Finished detailed project plan</td>
</tr>
<tr>
<td>11/1/2022</td>
<td>Finished the user interface design</td>
</tr>
<tr>
<td></td>
<td>Finished the manual control including socket connection, movement of robotic arm and picking and putting down objects</td>
</tr>
<tr>
<td>23/1/2022</td>
<td>Hand in interim report</td>
</tr>
</tbody>
</table>
5. Conclusion

The project aims to create a robotic arm controlled by the mobile app to play a game with a human player. The mobile app is coded using Android Studio with Java to control the robotic arm. The robotic arm can move along 3 axes and pick up and put down object. The Arduino board receives command from the app and controls the robotic arm.

However, there are several difficulties encountered in the progress, including the flipping action of the robotic arm. Therefore, it may take some time to figure out the solutions and test them. In the next phase, actions are to be taken to prepare all the deliverables before optimizing them. With improvements and tests done later, it is expected to be able to play game with human normally which performs better and able to make more accurate moves.
References


Figure 2: Self prepared

Figure 3.1: Self prepared

Figure 3.2: Self prepared

Figure 4: Self prepared