COMP4801 FINAL PRESENTATION

Topic: Pick and Place Game App for 3D Printed Robotic Arm

Group: 21049
Supervisors: Dr. Chim T W, Mr. David Lee
Wan Tsun Wai, Alan (3035569017)
Wong Ka Ngai, Benny (3035568881)
01

PROJECT INTRODUCTION

Background
In 2017, an AI player AlphaGO defeated the best human player Ke Jie in the game GO.
The use of robotic arms can be traced back to 1954 when George Devol invented the first industrial arm.
01

PROJECT INTRODUCTION

Motivation
Motivation

- Connect Four is simple to play, but hard to master.
- Connect Four is a solved strategy game
- No existing library to use
- STEM education
PROJECT INTRODUCTION

Objective & Deliverable
Develop a mobile application for a 3D printed pick-and-place robotic arm that can play Connect Four with human opponents.
Deliverables

Controlling Arm: Arm will be controlled using the app via BlueTooth.

Seeing Board: Board status will be captured and analyzed.

Thinking Strategy: Mini-Max Algorithm will be used to construct the winning algorithm.
02

METHODOLOGY

Software
TOOLS

Android Studio:
IDE for app development

Firebase:
Cloud database for scoring system

Github & Git extension:
branching workflow, code collab
BOARD DETECTION

- Capture board image and store in OpenCV Mat format
- Trim and blur
- OpenCV HoughCircles() function to detect the 42 circular slots
- HSV color detection to check for discs
CONNECT FOUR AI

- Minimax is widely used in two player turn-based games
- Negamax simplifies Minimax (one’s loss = gain of another)
- Optimize with Alpha-Beta pruning
4,531,985,219,092

reachable states from the initial empty board state
OPTIMIZATION

- Follow Pascal Pons’s tutorials
  - Bitboard
  - Move exploration order
  - Transposition table
  - Pre-generated book
Scoring System

- Optimal move +5, non-optimal move +0
- Use hint +0
- Max score 105 (Player is optimal and wins)
- 100 for a tie game
Methodology

Hardware
**TOOLS**

**Arduino:** Modify microcontroller program

**Shapr3D, Solidworks:** 3D modeling

**Ultimaker Cura:** 3D printing

**CorelDraw:** Laser cutting
Robotic Arm

- Arduino MEGA 2560
- RAMPS1.4
- 3D printed components
- End-stop switch x3
- NEMA17 Stepper motor x3
- SPP-CA Bluetooth module
- Tower pro SG90 Servo motor
Florin Tobler designed the Robot geometry (3-DOF)

G-code commands to instruct the arm

Control the arm using the app by sending G-codes via Bluetooth
ENHANCEMENT

- Install end-stops switches for auto-homing (G28)
- Modify G-code behavior to be more intuitive
- Implement operation range in both Arduino and the app
DISC STACKER

- 3 10-degree ramps roll out discs
- Gripper can grip 21 discs continuously at the exit
- 3D printed joint clamps assemble laser cut plates and act as support base
PROJECT SETUP

Robotic Arm
picks and places discs

Disc Stacker
facilitates disc picking

App
detects board and controls the arm

Connect 4
Game Board
METHODOLOGY

System Workflow
The app captures image, OpenCV processes to get board state

JNI translates into C++ format

AI algorithm loads external book to table and then calculates result

The app receives the move and sends G-code to the arm

When the game finishes, the app sends player name and score to Firebase
APP DESIGN

UI and UX design
Followed suggestion from Material Design

Teal and Pink are the colours of the connect four discs

Teal represents the player

Pink represents the robot
MODERN MINIMALISTIC DESIGN AESTHETIC

- Additional colors (Blue, Grey)
- Gradient colors
- Rounded corners
- Shadows
- Sans-serif font (Roboto)
- Vector instead of image
UX DESIGN

- 3 game modes
  - Player VS AI mode
  - Arm controller
  - Building mode
- Before accessing them, use bluetooth button to connect to the arm
UX HONEYCOMB

- Useful
  - Solves the right problem
- Findable
  - Provides everything needed at home page
- Usable
  - Learnable (simple user flow)
  - Forgiving (minimizes user errors through instructions)
  - Satisfying (conveys connection status)
APP DESIGN

Main functions
PLAYER VS AI MODE

Play with the perfect AI on the Connect 4 game board

Earn points with optimal moves, use hints for assistance

Players learn from mistakes, advance skills and strive for better scores.
ARM CONTROLLER

The first half keypad moves the arm to predefined positions.

The second half gamepad allows users to control the gripper and arm movement manually.
BUILDING MODE

Offers the gaming experience of playing with a claw machine.

Randomly gives a pattern to player. Player builds the pattern within 10 min using the gamepad.
03

APP DESIGN

Demo
ARM CONTROLLER
BUILDING MODE
PLAYER VS AI MODE (GAMEPLAY 1)

1.5x speed
PLAYER VS AI MODE (GAMEPLAY 2)
PLAYER VS AI MODE (DRAW)

20x speed
04

EXPERIMENTS

Board State Detection
APPROACH 1:
TENSORFLOW LITE NEURAL NETWORK

- YOLOv5s model for supervised learning
- Capture board and disc images and label manually
- Construct the board state by identifying all discs in the image
RESULT

- Too sensitive to recognize irrelevant objects
- Detect multiple objects in the same area
- Miss some discs on the board
- Performance is unstable
**APPROACH 2: OPENCV CIRCLE DETECTION**

- HoughCircles() function to find circle radius and center
- Detect 42 circles in a 7x6 grid to identify the board
- Valid circle if its coordinate is detected in multiple snapshots
- Valid row if 7 valid centers with similar y coordinate
- Identify the board with 6 valid rows
RESULT

- Performance is reliable
- Trimming and blurring applied to reduce noises
- Fine-tune input parameters, e.g. Reduce the range of radii
04

EXPERIMENTS

Disc Detection
RGB Color Space

- Combine red, green and blue lights
- Correlated to light intensity
- Poor performance
- Fail to classify the same disc with different light direction and position
HSV COLOR SPACE

- Hue represents the color
- Saturation represents the amount of grey mixed
- Value represents brightness
- Colors are in degrees, check the color range of teal and pink
- Robust to the change in environment
04

EXPERIMENTS

Connect 4 AI
STEPS TO MAKE THE CONNECT 4 AI

Minimax with Alpha-Beta Pruning
Bitboard
Move Ordering
Transposition Table
Transforming into an Android Library
MINIMAX SCORE

- If the player can win (+ve)  
  22 – the total number of discs needed to win

- If the player can only lose (-ve)  
  the total number of discs needed to defeat the player by the opponent – 22

- If the player can draw (null)

The current Red player can win with his 4th disc,  
22 - 4 = 18
MINIMAX WITH ALPHA-BETA PRUNING

- Maximizer VS Minimizer
- Cut off branches to reduce computational time
- Negamax variant
- Can only solve positions with more than 14 moves

the rightmost branch is cut as 5>C is guaranteed
- Encode board position with bitmap
- Bitwise operation for disc placing, alignment checking, etc.
MOVE ORDERING

MIDDLE COLUMN FIRST
- Explore from the middle to the edge columns
- Higher chances to produce alignments

AVOID BAD MOVES
- Place disc to block the opponent’s winning
- Do not place under the opponent’s winning slot
- Nothing to do if his winning slot >= 2

EVALUATE MOVES
- Moves with open-ended 3-disc alignments are generally better
**TRANSPOSITION TABLE**

- Store calculated key-value pairs in cache
- Avoid re-computing the explored board positions
- 40MB with $2^{23}$ entries
# Benchmark by Pascal Pons

<table>
<thead>
<tr>
<th>Level Type</th>
<th>Moves Condition</th>
<th>Remaining Condition</th>
<th>Mean Time</th>
<th>Mean Nb of Pos</th>
<th>K Pos/s</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>End Easy</strong></td>
<td>(28 &lt; \text{moves})</td>
<td>(\text{remaining} &lt; 14)</td>
<td>4.722 μs</td>
<td>54.93</td>
<td>11,630</td>
</tr>
<tr>
<td><strong>Middle Easy</strong></td>
<td>(14 &lt; \text{moves} \leq 28)</td>
<td>(\text{remaining} &lt; 14)</td>
<td>39.90 μs</td>
<td>517.4</td>
<td>12,960</td>
</tr>
<tr>
<td><strong>Middle Medium</strong></td>
<td>(14 &lt; \text{moves} \leq 28)</td>
<td>(14 \leq \text{remaining} &lt; 28)</td>
<td>3.736 ms</td>
<td>48,450</td>
<td>12,970</td>
</tr>
<tr>
<td><strong>Begin Easy</strong></td>
<td>(\text{moves} \leq 14)</td>
<td>(\text{remaining} &lt; 14)</td>
<td>275.5 μs</td>
<td>3,693</td>
<td>13,400</td>
</tr>
<tr>
<td><strong>Begin Medium</strong></td>
<td>(\text{moves} \leq 14)</td>
<td>(14 \leq \text{remaining} &lt; 28)</td>
<td>113.4 ms</td>
<td>1,459,000</td>
<td>12,870</td>
</tr>
<tr>
<td><strong>Begin Hard</strong></td>
<td>(\text{moves} \leq 14)</td>
<td>(28 \leq \text{remaining})</td>
<td>5.667 s</td>
<td>72,490,000</td>
<td>12,790</td>
</tr>
</tbody>
</table>
PRE-GENERATED BOOK

- A 32MB byte sequence file storing key-value pairs
- Load into the transposition table at initialization
- Save time for calculations at early game states
Transforming into an Android Library

- Convert jstring into std::string with JNI
- Configure CMake
- Add functions for JNI to call, replacing the original main function
- Put the book at assets file, change extension to avoid AAPT compression
- AssetManager instead of std::ifstream

https://github.com/bennywong3/c4solver-android-library
04

EXPERIMENTS

Disc Stacker
**DISC STACKER**

**REQUIREMENT**

- Pick discs at the same position
- Contain at least 21 discs
- Roll out another disc automatically when one disc is picked
- Able to stand alone without toppling

Sketch design of the disc stacker
VERSION 1

- 5-degree slope, 3cm gap
- Vertically printed due to build plate size, support removal caused damage to the ramps
- Discs fail to travel through the gap
VERSION 2

- 5-degree slope, larger gap, horizontally printed
- Shallow exit, sometimes discs roll to the exit and then pop out
- 5-degree ramps fail to provide enough gravitational force
- Warping
VERSION 3 LASER CUTTING

Shortcomings of 3D printing
● Build volume constraints
● Time-consuming
● Many PLAs are wasted if the print fails

Laser cutting can be used for simple shapes like the disc stacker
● 10-degree slope
● Deeper exit
FINAL RESULT

- 3D printed joint clamps
- Laser cut disc stacker layers
- Clamps combine layers and provide rigid base support
LIMITATION & FUTURE WORK
SOFTWARE LIMITATION

- No iOS support
- Necessary to install a black paper at the back for better board recognition
HARDWARE LIMITATION

- Fixed distance between the robotic arm, game board and disc stacker
- Cannot pick up laying down discs and a disc stacker is necessary
- Inaccurate positioning due to a non-rigid connection between robotic arm components
FUTURE WORK

- Enhance board detection
- Add a manual calibration function to Arm Controller
- Implement a difficulty system in Player VS AI Mode
CONCLUSION
CONCLUSION

- Accomplish the objective with extra features (perfect AI)
- Offer game modes other than Connect 4 in the app
- Demonstrate the robustness of AI in strategic thinking and object recognition
- Useful in STEM education
THANKS!

Do you have any questions?