

**Project and Professionalism**

**(6CS020)**

**A1: Project Proposal**

|  |  |
| --- | --- |
| Student Id | : NP03A190314 |
| Student Name | : Sashank Dulal |
| Supervisor | : Sangay Lama |
| Cohort/Batch | : 5 |
|  |  |
| Submitted on | : Wednesday, September 23,2021 |

**Acknowledgements**

**Table of Contents**

[**Statement of Proposal**](#_6fked771qjas) **4**

[**Project Title**](#_yywu7k97m0f2) **4**

[Academic Question](#_qsaz7gc3scrt) 4

[Aims and Objectives](#_alxgqq40nvxd) 4

[Artefacts](#_1m752tquz60k) 5

# 

# Statement of Proposal

## Project Title

The name of the proposed project is “Chessmate”.

* 1. **Project Statement**

Chessmate is a chess playing bot that engages in a chess match with the user on a physical chess board. It uses a robotic arm to move the chess pieces in the board and image processing techniques to detect movement of pieces on the board.

## Academic Question

* How will the bot get the best possible movement for the move produced by the user?
* How will the chess pieces be moved using the arm?
* What will be used to track the movement of chess pieces?

## Aims and Objectives

* + 1. **Aims**
* Research on similar systems and their features.
* Gather information on image processing techniques and object detection.
* Gain an understanding on how to create movements for a bot.
* Learn how to create/use chess engines or APIs.

* + 1. **Objectives**
* Develop a chess playing bot.
* Implement object detection to the bot.

## Artefacts

* **System to detect movements of chess pieces placed by the user on the board.**

This system is responsible for tracking the position of chess pieces on the board and detecting the moves of the user.

* **Chess Engine**

This system generates best possible moves based on the moves done by the opponent user.

* **Mechanism for movement of pieces in the board.**

This system takes in the recommendations made by the chess engine to physically move the chess pieces on the chess board.

* **Algorithm to find out the pathway of the movement.**

This system finds out the optimal pathway for the chess piece to move from one place to another.

1. **Project Proposal**
   1. **Introduction**

Chess is a 2-player strategy game played on a 8x8 board with 16 pieces for each player. With the aim of trapping the opponent’s king piece (Also called “Checkmate”), the players devise their own strategies and moves to get to that aim.

1. Problem Domain
2. No Proper method of training by playing physical match for newcomers

If a newcomer wants to gain proper experience at chess, he/she have to physically play that game and since they are not well-trained, playing with another professional can be discouraging for them and it is difficult to find the person who agrees to play with an unexperienced player.

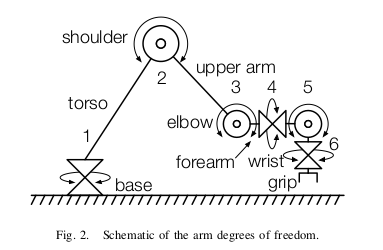
1. Costly trainings and coaches

In order to get good at chess, proper training and regular practice is required and the training programs and coaches are expensive to maintain.

1. Project as a Solution
2. Even the newest inexperienced players get to play with the standard opponent.
3. The cost of getting trained drastically decreases as it is a one-time investment. Once the bot has been bought, users can play whenever they like. Similarly, the bot would be constructed out of locally-sourced parts, repairing and even upgrading would not be difficult or expensive.
4. Research on Similar Systems
   1. Gambit: An Autonomous Chess-Playing Robotic System

This paper presents a 6-DoF chess playing robot system named Gambit which is capable of playing chess on a physical board against humans. It includes a low-cost sensor for perception, custom-made robot arm and learning algorithms for detection and recognition of objects on the board. The authors of the project mainly view this project as a way of exploring the perception and manipulation in a “noisy, less constrained real-world environment”. Gambit does not require the chess pieces to be exactly modelled or instrumented as it monitors the state of the board and tracks the movement of the pieces that the opponent has made and communicates with the human opponent using spoken-language interface.

The perception system tracks the position of pieces on the board and the board is not fixed relative to the board and the board is calibrated continuously throughout the game. On the other hand, the system makes use of an open-source arm design as it was moderate in cost (about $18000 for parts) and their major aim was to enable smooth motion and interaction. It consisted of a 6-DoF arm with a gripper attached ad one end. As shown in figure. 1, the DoFs 1,2 and 3 are used for positional control whereas 4,5 and 6 are used for orientation control with the help of roll-pitch-roll spherical wrist. (remaining)

****

For sensing purposes, the system consisted of a shoulder-mounted depth-sensing camera (similar to Xbox Kinect) along with a camera attached to the gripper. The depth-sensing camera provided color information along with depth of each pixel and worked within a range of 0.5m to 5m. Since the minimum working range was very high, it presented a problem that the camera had to be mounted high but the authors solved this issue by mounting the camera facing backwards tn the torso of the arm so that the distance is increased.

The driver software for the system ran on a dedicated Intel Atom net-top PC with CAN and RS-485 PCI cards whereas controlling of the arm was done using a separate computer that handled ROS operations. And the system for detection and recognition of board consisted of four hierarchical classifiers mainly for detecting squares, pieces/backgrounds and two for recognition of types of chess pieces. Each piece was labelled out of pre-defined pieces {B,N,K,P,Q,R}. The hierarchy of the vision algorithm is depicted in the figure below.

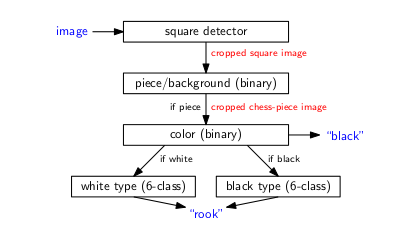


Figure 1: Hierarchy of the chess-piece recognition algorithms

1. Artefact

System to detect and track movement of chess pieces on the board

This system will be responsible for recognition and detection of movements on the chess board. It will comprise of a sensor for vision purposes and a processing hardware for computational tasks. An image processing library such as opencv or tensorflow will be used to develop the tracking algorithm. Similarly, the sensor will be fixed to a point to avoid any errors due to calibration. Once the system detects movement, it will transfer the information to the system handling chess gameplay.

System to generate chess moves

This system generates valid and competitive chess moves based on the moves placed by the user. When the user have already placed a move and the system detects a move, it passes the move to this system and the validity of the move is checked at first. If the move is valid, then the move is passed through the chess API to generate best possible move against that particular move placed by the user. On the other hand, the chess API will either be an open-source API or will be developed from scratch for our own purposes.

System to move the chess piece for the bot

Under this system, a hardware that can physically move the chess pieces in the board will be developed. The system will be able to move the pieces based on the moves provided by the chess API and the pathway will be calculated by a different algorithm. Different methods of movement such as mechanical arm or magnetic method have been considered for this project and will be decided by analyzing their strengths and weaknesses.

Mechanical Arm

An arm with 4-DOF will be constructed using servos and LEGO-style blocks and the arm will be capable of picking the chess piece and move it around the chess board.

Magnetic Method

A platform that lies below the chess board that resembles a typical CNC machine will be constructed and a magnet will be attached to the moving part of the machine and a magnet will be glued on the bottom part of each and every chess pieces/

System to declare pathway for moving chess pieces

This system will be responsible for generating pathway for the movement of chess pieces on the board for the bot. The algorithm to find out the optimal pathway for the chess piece to move will be based on the type of hardware to be developed based on the analysis. For instance, if a mechanical arm is to be constructed, an algorithm such as inverse kinematics would be developed in order to find out the angle required for each servos to get from one point to another whereas if the magnetic method is to be considered, an algorithm involving G-code or any other methods will be used.

Tools and technologies required

Hardware Requirements

* Mid-range PC (Laptop or Desktop) with at least Intel i5 or similar processor, 8GB RAM, 256GB SSD is required for SSH connection to Raspberry Pi.
* Hardwares such as Raspberry PI along with camera module, high torque servos.
* Chess Board

Technical Requirements

* Knowledge of Embedded systems such as Arduino and Raspberry PI.
* Clear understanding of image processing techniques and libraries.
* Experience of working with linux environments (especially SSH/VNC and scripting).

Other Requirements

Development Cycle

For our purposes, Rapid Application Development (RAD) model will be used since the project will be done in an individual basis and the requirements are

Chess statistics reference: <https://www.statista.com/statistics/809953/global-chess-market-size/>

2021