Contents

[1 Aim 2](#_Toc531537474)

[2 Game types 2](#_Toc531537475)

[3 Design Considerations 2](#_Toc531537476)

[3.1 Board composition 2](#_Toc531537477)

[3.2 Piece detection 2](#_Toc531537478)

[3.2.1 Standard configuration 2](#_Toc531537479)

[3.2.2 Non-standard configuration 3](#_Toc531537480)

[3.3 Piece movement 4](#_Toc531537481)

[3.3.1 Electromagnet 4](#_Toc531537482)

[3.3.2 IR Sensor (Only for glass board) 4](#_Toc531537483)

[3.3.3 Hall effect sensor (all boards) 4](#_Toc531537484)

[3.3.4 Robotic apparatus 4](#_Toc531537485)

[3.3.5 Obstacle detection 5](#_Toc531537486)

[4 Artificial intelligence 5](#_Toc531537487)

[5 Power supplies 5](#_Toc531537488)

[6 Minimum viable product 5](#_Toc531537489)

[7 Limitations 6](#_Toc531537490)

# Aim

The aim of this project is to build a high-end chess application whereby chess pieces will move themselves automatically.

Pieces can be moved either with hands or via verbal commands.

A raspberry pi will be the central device controlling and managing the various peripherals in this project.

# Game types

This project will service three game types

1. 1 human vs 1 human
2. 1 human vs 1 AI
3. 1 AI vs 1 AI

# Design Considerations

## Board composition

The perception of a high-quality board is essential to the success of this project, as a result the board perception parameter in Table 1 is the main driver for the selection of the board composition.

Table 1: Comparison of board materials

|  |  |  |  |
| --- | --- | --- | --- |
|  | Finely threaded mesh | Wooden chessboard | Glass |
| Board perception | Low quality | High quality | Low to mid quality |
| Cost | Low | Medium | Medium |
| Piece sensor cost | Low | Medium-High | Low |

## Piece detection

### Standard configuration

This section assumes that the chess game starts in the standard configuration.

In the case where verbal commands are the only ones being given, the application doesn’t need to detect the pieces on the board, it will have a complete picture of where all pieces are on the board as all movements are conveyed to the application.

In the case where a player is moving pieces with his hand, the application needs to recognise what move this corresponded to. There are a few ways to do this. The options are outlined in the table below.

Table 2: Comparison of piece detection methods

|  |  |  |  |
| --- | --- | --- | --- |
|  | Pressure Sensor Array | Pi Camera module | Hall effect sensor |
| Supplier | Alibaba Chinese supplier | Raspberry pi foundation | Various |
| API | Unclear, vendor hasn’t provided details | Well known python libraries | Read input pins (python) |
| Cost | $300 + Driver cost (unknown) | $30 | $4 per sensor + $10 port extender + $70 PCB board |
| Mode of detection | Compare map of pressure hotspots before and after move | Take pictures before and after move and compare them for differences, the two spots with the greatest differences will correspond to the move. | Detect magnetic field of magnets embedded in pieces |
| Durability | Wear and tear over time may cause detection issues, build quality is also suspect due to the origin of the piece | Not coming in contact with any pieces, should last a long time. | Not coming in contact with any pieces, should last a long time. |
| Detection reliability | Questionable depending on configuration, pressure mat would need to be placed under the board and detect small perturbations | Questionable due to lighting conditions | Very reliable |
| Piece location precision | Questionable depending on configuration but should be quite precise | Questionable due to lighting conditions | Can’t get exact coordinates, it’s either there or not |
| Board types supported | Wooden, Glass, Plastic | Glass | Wooden, Glass, Plastic |

The detection reliability parameter is of paramount importance and as such is given a proportionally high weighting, as a result the hall effect sensor setup has been chosen.

### Non-standard configuration

This section assumes that the chess game starts in a random configuration.

This poses a few challenges as now the application has no knowledge about any of the pieces on the board, it needs to obtain information about all the pieces on the board, a few solutions are outlined below:

1. The application performs no validation of any piece moves, it blindly moves pieces to the requested positions.
2. The user manually inputs all the pieces on the board via an API
3. NFC chips are embedded into each piece. The application traverses the entire board, reads the NFC chip and constructs an idea of the board.

A comparison of these approaches is performed below:

Table 3: Comparison of options for starting from a non-standard configuration

|  |  |  |  |
| --- | --- | --- | --- |
|  | No validation | Manual user input | NFC Chips |
| User friendliness | Just place the boards | Cumbersome and error prone | Easy |
| Timeliness | Easy |  |  |
| Materials cost | $0 | $0 | $185 total ($5 per tag and $25 for NFC reader) |
| Programming required | Minor amount, code paths are required that perform no validation | None, existing infrastructure can be leveraged | Interaction |
| Other considerations |  |  | NFC chips would need to be embedded into each piece, they would need to be secured. This is an additional cost |

## Piece movement

In order to move the chess pieces, magnetic material is attached to the underside of each chess piece. An electromagnet will then be used to magnetically attract those pieces and move them to their desired positions. To prevent jittering the very middle of the piece should be grasped.

### Electromagnet

A grove electromagnet was chosen for use due to its low cost, this electromagnet connects to a GrovePi+ board which integrates directly with the raspberry Pi. The GrovePi+ facilitates the use of many sensors simultaneously.

### IR Sensor (Only for glass board)

An IR sensor can be used to detect where a piece is on the board square, a Sharp GP2Y0A41SK0F IR sensor provides a cost-effective way to do this. If this sensor is rotated a series of data points can be obtained to provide a better estimate of the centre of the piece.

### Hall effect sensor (all boards)

A hall effect sensor can be used to detect magnetic fields at a distance, if each piece is embedded with a magnet and each board square has a corresponding hall sensor, then detecting piece movements is a trivial task.

### Robotic apparatus

Off the shelf solutions for the robotic apparatus have been evaluated. A comparison below shows the differences between two low-cost alternatives:

Table 4: Comparison of low-cost robotic arms

|  |  |  |
| --- | --- | --- |
|  | EleksLaser-A3 pro 2.5W | Makeblock XY stage |
| Cost (AUD) | $315 | $250 |
| Working area | 30x40cm | 31x39cm |
| Electrical requirements | 12V, 5A | 12V, 3A |
| Weight | 3.4kg | 3.17kg |
| Reviews | Good reviews about use | Complaints about jittering, difficulty of construction and outdated documentation |

The EleksLaser-A3 pro is being selected largely due to its reviews.

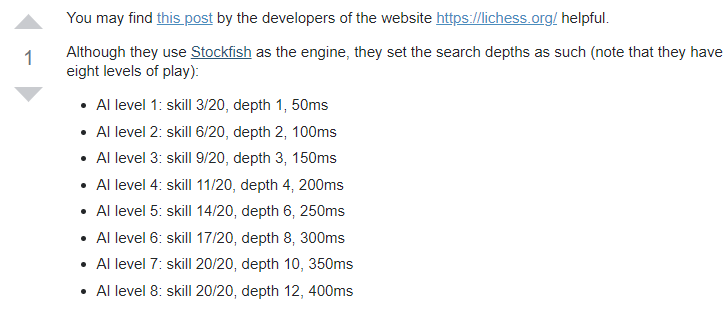
The EleksLaser can be programmed with GRBL software which has python compatible libraries, this permits seamless use with the raspberry pi

### Obstacle detection

When pieces are being moved optimal paths need to be selected for efficiency and to avoid obstacles. The A\* algorithm has high performance and accuracy.

# The AI player

The AI selected must be able to play at 10 difficulty levels. Consensus online is to use the open-source Snapfish engine. The stackoverflow article located at <https://chess.stackexchange.com/questions/14770/gull-engine-which-depth-value-to-set-6-difficulty-levels/17698> provides the following pertinent information.



# Power supplies

The application should be rechargeable with one plug powering all the necessary devices.

A device will need to be programmed to supply 12V to the motor system and 5V to the raspberry pi

# Minimum viable product

The minimum viable product needs to fulfil the following criteria

* Only operate in standard mode
* Allow a human player to either use their hand or issue audio commands to move pieces.
* Pieces will move without touching each other
* Permit the selection of 10 difficulty levels.
* Recognise conditions such as check/checkmate
* Have a single electrical plug that will be rechargeable and supply 12V to the motor system and 5V to the raspberry pi.
* Handle some misregistered moves

The cost outline of this is outlined below

|  |  |
| --- | --- |
|  | Cost (AUD) |
| Raspberry pi | 41 |
| GrovePi+ | 42 |
| Hall effect sensors | 4 x 64 |
| Raspberry pi port expander | 10 |
| PCB for hall sensors | 70 |
| EleksLaser-A3 pro 2.5W | 315 |
| Grove electromagnet | 15 |
| 12V supply | 31 |
| 10000 mAh supply | 30 |
|  |  |
| **Total:** | 810 |
| **Total + 15%:** | 931.5 |

# Limitations

The following cases are actively not being handled

* Resetting of the board