

DEPARTMENT OF ELECTRICAL ENGINEERING & ELECTRONICS

Project Plan for Over the board Chess AI opponent

Author: Edward Collins (201743271)

Project Supervisor: Dr Lee Devlin

Project Assessor: …name of assessor…

Declaration of academic integrity

By submitting this work, I confirm that I have read and understood the University’s Academic Integrity Policy.  
By submitting this work, I confirm that I have acted honestly, ethically and professionally in conduct leading to assessment for the programme of study.  
By submitting this work, I confirm that the work I am submitting is my own. I have not commissioned production of the work from a third party or used artificial intelligence software in an unacceptable manner to generate the work\*. I have not copied material from another person or source, nor committed plagiarism, nor fabricated, falsified or embellished data when completing the attached piece of work. I have not colluded with any other student in the preparation and/ or production of this work.  
\*Software applications include but are not limited to, ChatGPT, Bing Chat, DALL.E, Bard

## Tables of contents, figures, etc.

## **Project Scope**

## **Project Objectives**

## **Project Requirements**

## 4.1 Functional Requirements

## 4.2 Non-functional Requirements

## 4.3 Hardware Requirements

## 4.4 Software Requirements

## **Work Packages**

## WP1: Project Initialisation

## WP2: Select, Order and Assemble Physical Components

## WP3: Computer Vision Software

## WP4: AI Software

## WP5: Chess Engine Software

## WP6: Final Software Completion

## WP7: Project Completion

## **Skills and Resource Audit**

## 6.1 Skills Audit

## 6.2 Resource Audit

## **Appendices**

## Appendix 1: Gantt Chart

## Appendix 2: Risk Assessment Form

## Appendix 3: FYP Ethics Self-Assessment

## Project scope

This project aims to build an over the board chess AI opponent. This will be able to determine the position of the chess pieces on the board in real time using a camera mounted above the chessboard. The system will process the visual information to understand the current game state, identify the human player's move, and then use the Stockfish chess engine to suggest a move to reply with. The final output will be the AI's chosen move, displayed to the user with either standard algebraic notation, or a chessboard with the newly moved piece being highlighted.

The project scope is to focus on the software implementation of the vision and logic systems. The physical output is limited to a text-based or simple graphical user interface display of the AI's move.

## Project Objectives

1. To assemble a stable physical rig with a camera mounted above a standard chess board.
2. To develop and code a computer vision system that can reliably take a raw image of the board and output the game state in with at least 99% accuracy for piece location.
3. To create a software interface that allows the computer vision module to communicate with the Stockfish chess engine, sending the human's move and receiving the AI's calculated response.
4. To allow the AI to play with both the white and black pieces (not at the same time).
5. To Create a User Interface to allow the project to easily be played with. ???
6. To Test and Validate the system by testing the integrated system's performance, measuring its speed, accuracy, and reliability across a series of test games and board positions.
7. To produce a comprehensive final project report detailing the project's design, implementation, testing methodology, results, and conclusions.

## Project Requirements

|  |  |
| --- | --- |
| **Category** Functional requirements | |
| **Identifier** | **Description** |
| F1 | The system shall successfully detect the boundaries of the chessboard from a top-down camera image. |
| F2 | The system shall segment the detected board into 64 individual square images. |
| F3 | The system shall be able to determine if a square is occupied by a white piece, a black piece, or is empty. |
| F4 | The system shall detect a change in board state between two consecutive images to determine the human player's move. |
| F5 | The system shall use stockfish to suggest moves to reply with and will display them either in algebraic notation or with an image of the chessboard showing the piece moved |

|  |  |
| --- | --- |
| **Category** Non-functional requirements | |
| **Identifier** | **Description** |
| NF1 | The processing time from the human moving a piece to displaying the AI's move shall be less than 10 seconds. |
| NF2 | The square occupancy detection (empty vs. white vs black) shall have an accuracy of over 99%. |
| NF3 | The piece detection shall have an accuracy of over 95%. |
| NF4 | The system should be able to cope with variations in ambient lighting. (shall be tested in different environments) |

|  |  |
| --- | --- |
| **Category** Hardware requirements | |
| **Identifier** | **Description** |
| H1 | A standard chess set is required. |
| H2 | A camera and a mounting rig to hold the camera in a fixed, top-down position. |
| H3 | A computer capable of running Python, Stockfish, and OpenCV |

|  |  |
| --- | --- |
| **Category** Software requirements | |
| **Identifier** | **Description** |
| S1 | The programming language used will be python |
| S2 | The OpenCV library will be used for the computer vision tasks. |
| S3 | The Stockfish chess engine will be used to produce the AI response |

## Work packages

|  |  |  |  |
| --- | --- | --- | --- |
| **WORK PACKAGE DESCRIPTION** | | | |
| **WP Title**: | Project initialisation | **WP number** | 1 |
| **Start Event**: | Start of semester 1 | | |
| **Duration**: | 3 weeks | | |
| **Objectives:**   * Submit completed project plan. * Become more familiar with required python libraries. | | | |
| **Task Description:**   * Create project plan with objectives and requirements. * Research existing projects on chess recognition. * Produce a Gantt chart. * Submit risk assessment. * Learn and practise using OpenCV, TensorFlow, and Stockfish. | | | |
| **Inputs:**   * Initial specification. * Project plan template. | | | |
| **Interfaces/links with other tasks or WPs:**   * This is required before any other work packages can begin. | | | |
| **Deliverables:**   * Risk assessment will be completed and approved. * Project plan will be completed. * Sufficient knowledge of python libraries will be attained. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WORK PACKAGE DESCRIPTION** | | | |
| **WP Title**: | Select, order and assemble physical components | **WP number** | 2 |
| **Start Event**: | WP1 is completed | | |
| **Duration**: | 2-3 weeks (depending on delivery times) | | |
| **Objectives:**   * To assemble the camera and the camera stand. | | | |
| **Task Description:**   * Choose a chess set, camera stand, and camera. * Complete and submit component order form. * Assemble Camera and Camera stand and ensure that it is stable and safe. | | | |
| **Inputs:**   * Physical components are required. | | | |
| **Interfaces/links with other tasks or WPs:**   * Risk assessment must be approved first (WP1). * Required for software from WP3 to be tested. | | | |
| **Deliverables:**   * All physical components will be obtained and assembled. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WORK PACKAGE DESCRIPTION** | | | |
| **WP Title**: | Computer vision software | **WP number** | 3 |
| **Start Event**: | WP 1 is completed | | |
| **Duration**: | 4 weeks | | |
| **Objectives:**   * To develop the software code that processes an image of the chessboard. | | | |
| **Task Description:**   * Ensure that the python code is able to access the image from the camera. * To create software that will detect the edges of the board and warp the board image so that an image of the board as a square is formed. * If detecting edges of the board is unreliable due to chessboards often having a boarder, then the above will be done using the squares on the board. * To split the resulting image into 64 small images of each square of the board for WP4. | | | |
| **Inputs:**   * Image output from camera | | | |
| **Interfaces/links with other tasks or WPs:**   * Required for WP4 * Can be started in parallel with WP2, however cannot be tested and therefore completed until WP2 is completed. | | | |
| **Deliverables:**   * Software created to detect the edges of the chessboard. * To output 64 small images of the board | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WORK PACKAGE DESCRIPTION** | | | |
| **WP Title**: | AI software | **WP number** | 4 |
| **Start Event**: | WP 3 is completed | | |
| **Duration**: | 5 weeks | | |
| **Objectives:**   * To develop the software to recognise a piece from an image. | | | |
| **Task Description:**   * To gather a large set of training data using the code in WP3 for taking pictures of the pieces. * Labelling the above pictures (either manually or automatically). * Creating an AI model which will use the training data. * To incorporate a prediction function which returns which piece it believes is most likely to be shown in a given input picture. * To test the accuracy of this to ensure that it meets the requirements set out above. | | | |
| **Inputs:**   * Images from WP3 | | | |
| **Interfaces/links with other tasks or WPs:**   * WP3 must be completed first to provide training data. | | | |
| **Deliverables:**   * To develop the software to recognise a piece from an image with 95% accuracy and to have a 99% accuracy at detecting whether a square is vacant or not. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WORK PACKAGE DESCRIPTION** | | | |
| **WP Title**: | Chess engine software | **WP number** | 5 |
| **Start Event**: | WP1 is completed | | |
| **Duration**: | 1 week | | |
| **Objectives:**   * To set up the Stockfish engine. * Ensure it can communicate with Python. | | | |
| **Task Description:**   * Install the Stockfish chess engine. * Use the python-chess library to create a communication interface. * Write a script that can push a human move to the engine and request a move from the engine. * Ensure that the Elo rating of the machine can be set. | | | |
| **Inputs:**   * None | | | |
| **Interfaces/links with other tasks or WPs:**   * Required for WP6. * Can be done in parallel with WP2, 3, or 4. | | | |
| **Deliverables:**   * A standalone script that can play a game of chess against Stockfish (via text command inputs). | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WORK PACKAGE DESCRIPTION** | | | |
| **WP Title**: | Final software completion | **WP number** | 6 |
| **Start Event**: | WP 3, 4, & 5 are all completed | | |
| **Duration**: | 2 weeks | | |
| **Objectives:**   * To have a fully functional Over the board Chess AI opponent. | | | |
| **Task Description:**   * Combine the software produced by WP 3, 4, & 5. * Test to ensure that all parts of the software work together. | | | |
| **Inputs:**   * WP 3, 4, & 5 | | | |
| **Interfaces/links with other tasks or WPs:**   * Integrates with WP 3, 4, & 5 to form the complete system. | | | |
| **Deliverables:**   * Complete all software relating to the project. * To have a fully functional Over the board Chess AI opponent which meets the requirements set out above. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **WORK PACKAGE DESCRIPTION** | | | |
| **WP Title**: | project completion | **WP number** | 7 |
| **Start Event**: | Project is fully functional | | |
| **Duration**: | 3 weeks | | |
| **Objectives:**   * Project abstract will be written. * Bench inspection will be completed. * Project report will be written | | | |
| **Task Description:**   * Write project abstract. * Write project report. * Ensure that the machine works well in the location and with similar lighting as where the bench inspection will take place. | | | |
| **Inputs:**   * Rest of the project. | | | |
| **Interfaces/links with other tasks or WPs:**   * Mostly requires WP6, though report and abstract may be started earlier. | | | |
| **Deliverables:**   * All documentation completed. | | | |

## Skills and Resource Audit

### Skills Audit:

|  |  |  |
| --- | --- | --- |
| **Skill** | **Level** | **Training needs** |
| Python programming | intermediate | Learning using w3schools |
| OpenCV Library | intermediate | Learning using online OpenCV documentation |
| Chess | basic | Completed |
| Stockfish | basic | Learning using online Stockfish documentation |
| Tensorflow library | intermediate | Learning using w3schools |

### Resource Audit:

|  |  |  |
| --- | --- | --- |
| **Resource** | **Availability** | **Alternatives** |
| Camera | Available to buy online | Use my own |
| Camera stand | Available to buy online | Make a camera stand |
| Chess set | Available to buy online | Use my own |

## Appendix 1. A Gantt chart (1 Page)

## Appendix 2. The risk assessment form.

**A blue and yellow text on a black background

AI-generated content may be incorrect.Risk Assessment Report**

**Risk Assessment ID:** sgecoll4\_2025-09-29-11-25-07

**Expiry Date:** 29/09/2026 *(unless environment has changed)*

**Risk Assessor Details:**

**MWS Username:** sgecoll4

**Name:** Edward Collins

**Email:** [sgecoll4@liverpool.ac.uk](mailto:sgecoll4@liverpool.ac.uk)

**Location Details:**

**School:** School of Electrical Engineering, Electronics and Computer Science

**Department:** Electrical Engineering and Electronics

**Building:** Electrical Engineering-Block A

**Space Ref:** 4th floor lab space

**Location Notes:**

**Task/Activity Details:**

**Task/Activity:** A camera will be mounted on a stand which could potentially fall over and cause minor injury, this risk will be minimised by ensuring the stand has a sturdy base to ensure it does not fall over.

The camera may fall off the stand and have similar consequences, the camera will be securely attached to the stand to minimise this risk.

**Person(s) Affected:** people close to the project.

**Supervisor Details:**

**Research Group:** other

**Supervisor Name:** Dr. Lee Devlin (ljdevlin) **Supervisor Email:** [ljdevlin@liverpool.ac.uk](mailto:ljdevlin@liverpool.ac.uk) **Additional Notes:**

**Colour Key:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Likelihood** | **5** | 5 | 10 | 15 | 20 | 25 |
| **4** | 4 | 8 | 12 | 16 | 20 |
| **3** | 3 | 6 | 9 | 12 | 15 |
| **2** | 2 | 4 | 6 | 8 | 10 |
| **1** | 1 | 2 | 3 | 4 | 5 |
|  | | **1** | **2** | **3** | **4** | **5** |
| **Consequence** | | | | |

**Likelihood/Consequence:**

|  |  |  |
| --- | --- | --- |
| **Likelihood** |  | **Consequence** |
| Very unlikely | **1** | Insignificant – no injury |
| Unlikely | **2** | Minor – minor injuries needing first aid |
| Fairly likely | **3** | Moderate – up to seven days absence |
| Likely | **4** | Major – more than seven days absence; major injury |
| Very likely | **5** | Catastrophic – death; multiple serious injury |

Likelihood(L) x Consequence(C) = Result(R)

**Result/Action:**

|  |  |
| --- | --- |
| **Result** | **Action to be taken** |
| 1-4 Acceptable | No further action but ensure controls are maintained |
| 5-9 Adequate | Look to improve at next review. |
| 10-16 Tolerable | Stop activity and review current controls |
| 17-25 Unacceptable | Stop activity with immediate effect, and rethink the entire experiment |

Likelihood(L) x Consequence(C) = Result(R) Additional Controls Required(Y/N)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identified Hazard** | **Control Measures** | **L** | **C** | **R** | **Y/N** |
| stand falling over | A large/heavy stand will be used to make the camera stand more steady | 3 | 1 | 3 | N |
| camera falling off stand | The camera will be securely attached to the stand | 2 | 1 | 2 | N |

**No Additional Controls required at this time.**

No attached files for this Risk Assessment.

**Local Area Rules:**

**Fire Evacuation Location:**

In front of Harold Cohen Library

**Emergency Arrangements:**

No Emergency Arrangements identified at this time.

**Risk Assessor Notes:**

Apologies if I've made any mistakes

**Supervisor Notes:**

No issues, this is a very low risk project.

**Risk Assessment Status:**

Approved by Supervisor.

## Appendix 3. FYP ethics self-assessment.

A screenshot of a computer

AI-generated content may be incorrect.