

Configuration Manual

MSc Research Project
Msc in Artificial Intelligence

Saketh Reddy Banala
Student ID: x23327596

School of Computing
National College of Ireland

Supervisor: Taylou Maniganze

National College of Ireland
MSc Project Submission Sheet



School of Computing

Student Name: Saleth Reddy Banala

Student ID: X23327596

Programme: Msc in Artificial Intelligence

Year: 2025

Module: Research Practicum

Lecturer: Taylou Maniganze

Submission Due Date: 15/09/25

Project Title: Deep learning chess framework to synthesize language model reasoning with traditional engines

Word Count: 729

Page Count: 9

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Saketh Reddy Banala

Date: 15-09-25

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Saketh Reddy Banala
Student ID: x23327596

INTRODUCTION

This system compares different types of AI in chess games:

- **LLMs:** Claude 3.5 Sonnet, Gemini 1.5 Pro
- **Chess Engines:** Stockfish 17, Berserk 13, Caissa 1.22

It runs automated chess matches and generates research data comparing how language models vs traditional engines play chess.

Following Steps

Requirements

- **Node.js** (version 16+) - Download from <https://nodejs.org>
- **VS Code** (recommended) - Download from <https://code.visualstudio.com>
- **Internet connection** (for LLM APIs)
- **Windows/Mac/Linux** operating system

Setup Steps

1. Extract the Project

Unzip the downloaded file

Extract to a folder like: C:\chess-ai-testing\ or ~/chess-ai-testing/

2. Open in VS Code

Open VS Code

File → Open Folder

Select the extracted GameCode folder

3. Open Terminal in VS Code

In VS Code: Terminal → New Terminal

Or press Ctrl+` (backtick)

4. Navigate to Project

`cd GameCode`

5. Install Dependencies

npm install

6. Check Setup

bash

node validate-setup.js

Should show ☒ for all engines.

7. Start API Server (Required)

cd src/actions

node server.cjs

Keep this running. Shows: Proxy server is running on http://localhost:3001

8. Open New Terminal

In VS Code: Terminal → New Terminal (keep first one running)

Navigate back to GameCode folder

cd ..

cd ..

9. Run a Test

node run-single-combination.js C001

Test Options

ID	White	Black	Time
C001	Claude 3.5	Gemini 1.5	15 min
C002	Gemini 1.5	Claude 3.5	15 min
C003	Claude 3.5	Berserk 13	10 min
C004	Claude 3.5	Caissa 1.22	10 min
C005	Gemini 1.5	Berserk 13	10 min
C006	Gemini 1.5	Caissa 1.22	10 min
C007	Berserk 13	Claude 3.5	10 min
C008	Berserk 13	Gemini 1.5	10 min
C009	Caissa 1.22	Claude 3.5	10 min
C010	Caissa 1.22	Gemini 1.5	10 min
C011	Berserk 13	Caissa 1.22	5 min
C012	Caissa 1.22	Berserk 13	5 min

Run Tests

Single Tes

node run-single-combination.js C001

All Tests (6-8 hours)

node src/run-all-tests.js

Optional: Web Interface

Start Web Dashboard

Open **new terminal** in VS Code:

cd src/monitoring

node server.cjs

Open: <http://localhost:3000/manual-test-dashboard.html>

Start Chess Game UI

Open **another new terminal** in VS Code:

npm run dev

Open: <http://localhost:5173>

View Results

Results saved to: results-by-combination/C001/csv-files/

Combine all results:

node consolidate-all-results.js

Common Fixes

"Node.js not found" → Install Node.js from <https://nodejs.org>

"Command not recognized" → Use node before script names

"Cannot read properties" → Start API server first: cd src/actions && node server.cjs

"No UCI response" → Run node validate-setup.js

Empty CSV files → Restart API server, check internet connection

Wrong folder → Make sure you're in the GameCode folder: cd GameCode

Success Signs

- Terminal shows move-by-move progress
- CSV files appear in results folder
- No error messages

Start with C001 to test, then run others as needed.

OUTPUT:

```
> chessBattle@0.0.0 dev
> vite

Re-optimizing dependencies because lockfile has changed

VITE v5.4.11 ready in 439 ms

→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

```
Proxy server is running on http://localhost:5000
CORS enabled for http://localhost:5173
Chess engine API available at /api/engine/*
Available endpoints: /api/engine/initialize, /api/engine/move, /api/engine/analyze, /api/engine/cleanup
```

Chess AI Arena

Select two participants to compete in an intelligent chess battle

● **Player 1**

Select Participant

Select a participant ▼

● **Player 2**

Select Participant

Select a participant ▼

✕ **Game Information**

Format: Head-to-Head Battle Available Players: 2 AI Models + 2 Chess Engines Evaluation: Stockfish Analysis

Please select 2 different participants and provide required API keys

Start Chess Battle

Chess AI Arena

Select two participants to compete in an intelligent chess battle

• Player 1

Select Participant

Select a participant

Select a participant

Claude 3.5 Sonnet (AI Model)

Gemini 1.5 Pro (AI Model)

Berserk 13 (Chess Engine)

Caissa 1.22 (Chess Engine)

• Player 2

Select Participant

Select a participant

Players: 2 AI Models + 2 Chess Engines

Evaluation: Stockfish Analysis

Please select 2 different participants and provide required API keys

Start Chess Battle

Chess AI Arena

Select two participants to compete in an intelligent chess battle

• Player 1

Select Participant

Select a participant

• Player 2

Select Participant

Select a participant

Select a participant

Claude 3.5 Sonnet (AI Model)

Gemini 1.5 Pro (AI Model)

Berserk 13 (Chess Engine)

Caissa 1.22 (Chess Engine)

Game Information

Format: Head-to-Head Battle

Available Players: 2 AI Models + 2 Chess Engines

Please select 2 different participants and provide required API keys

Start Chess Battle

Chess AI Arena

Select two participants to compete in an intelligent chess battle

Player 1

Select Participant

 Claude 3.5 Sonnet (AI Model) ▼

API Key for Claude 3.5 Sonnet

Enter API key

Player 2

Select Participant

Select a participant ▼

Game Information

Format: Head-to-Head Battle

Available Players: 2 AI Models + 2
Chess Engines

Evaluation: Stockfish Analysis

Please select 2 different participants and provide required API keys

Start Chess Battle

White: Caissa 1.22 VS Black: Berserk 13

White's Turn



Controls

Stockfish 17

Start Game

Download Log

Stats

Moves

0

Balance

0

Players

White

Caissa 1.22

0.0s

Total Time

Black

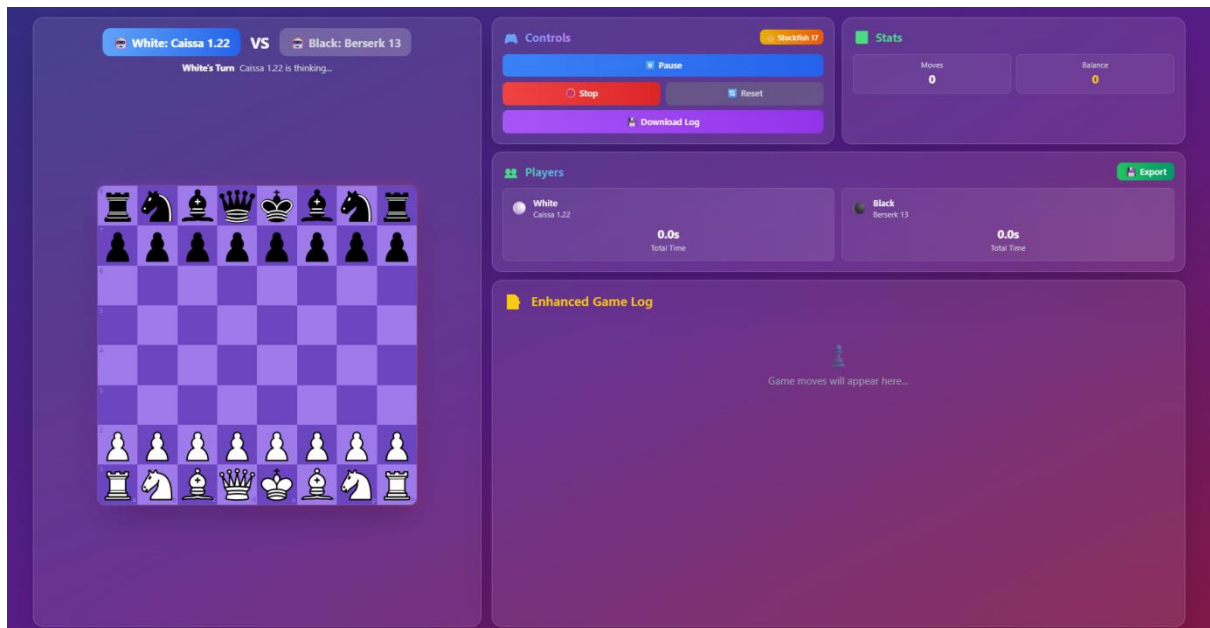
Berserk 13

0.0s

Total Time

Enhanced Game Log

Game moves will appear here...



REFERENCES:

<http://nodejs.org/en>

<https://github.com/microsoft/vscode>