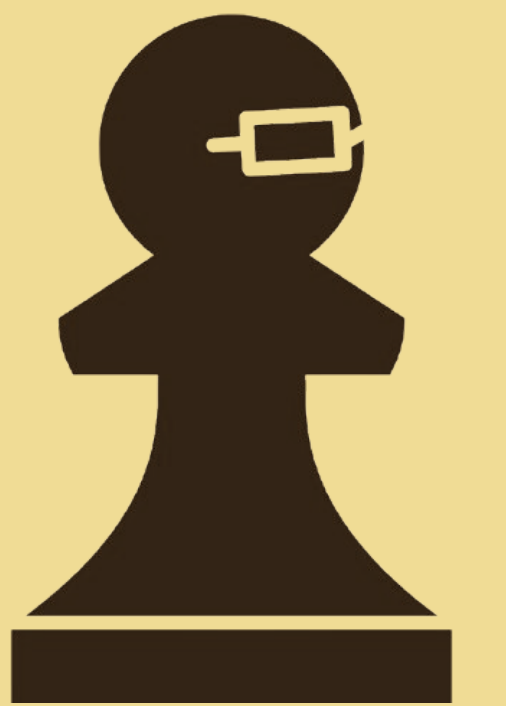


# FluxChess

## A Real-Time Dynamic Chess Tutor That Teaches Through Personalised Feedback

By Callum Farrell and Sanket Sridhar, Stratford College



### Introduction

#### Problem

We found that existing fixed-difficulty chess bots were **predictable** and often **too easy** or **too hard** to play against, and that feedback provided by existing chess sites was **too vague**.

#### Aim

Our aim was to design a chess bot that could:

- **Adapt** its difficulty in real time
- **Explain** player mistakes and good decisions
- Act as a “**personalised training partner**”

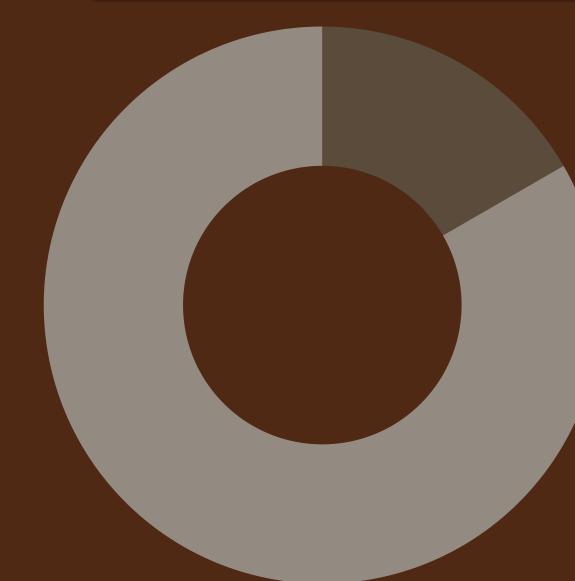
#### Hypothesis

Considering our problem, we wanted to see *if a chess bot dynamically adjusts its difficulty and provides real-time feedback would create a more effective chess training experience than fixed-difficulty bots.*

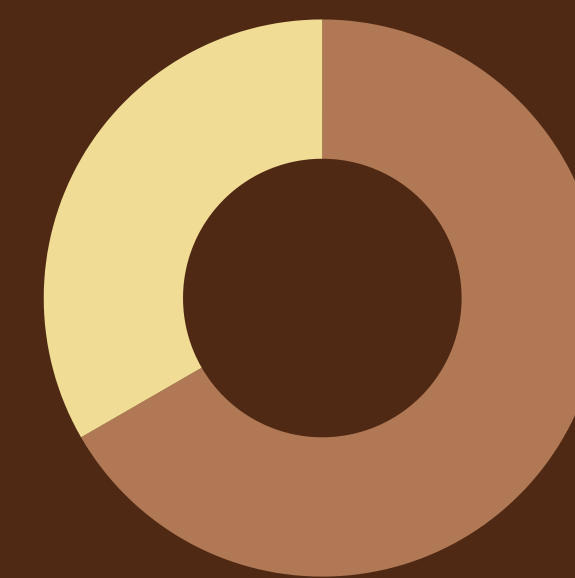


### Results

We asked questions to a group of chess players about how they felt about the bot after playing it. These are 2 of the questions we asked respondents.



*How well did the adaptive difficulty match your skill level?*  
17% answered “Balanced”  
83% answered “Too hard”  
*6 respondents*



*Did the bot help you understand your mistakes better than a standard chess bot?*  
33% answered “Yes”  
66% answered “About the same”  
*6 respondents*

#### Key Findings

- Users reported better understanding of mistakes compared to standard chess bots.
- Adaptive difficulty was noticeable, though some users found the bot itself challenging.
- Real-time feedback increased engagement during play.

### Analysis & Discussion

#### Interpretation of results

Majority of users preferred adaptive difficulty over fixed bots, with many also finding the given feedback to be useful, but most found it to be too difficult.

#### Why these results occurred

The engine we used for our project isn’t suited for lower skill players, as like most engines, it’s suited towards very high skill play. Although mitigation methods were implemented, the used engine remains optimised for high-level play, which contributed to perceived difficulty.

#### Feasibility and practical use

From our findings, we think our bot could be effective for **intermediate learners**, **school chess clubs**, and **self-directed** learners.

Since everything is ran locally, there are no cloud dependencies, and no risks of data sharing.

#### Limitations and improvements

- The survey we did had a very small sample size
- Difficulty calibration needs refinement
- Feedback quality depends on engine evaluation accuracy

With future iterations, we would focus on improved calibration and larger-scale testing.

### Methodology

#### System Overview

The program is built from five main components: the Chess Engine, the adaptive difficulty controller, the move analysis system, the feedback generation system, and the graphical user interface (via PyGame).

#### Adaptive difficulty method

Player moves are evaluated using centipawn (CP) analysis, these CP changes are used as a proxy for move quality, meaning the bot’s target ELO changes based on how well the user is performing. The engine’s strength is then adjusted through thinking time and candidate move limitations.

#### Feedback generation method

A locally ran LLM generates feedback based on engine evaluation change and move context, this feedback is short and explanatory, with the requests being handled asynchronously to avoid effecting gameplay.

#### Testing method

We tested the bot with members of our school’s chess team and other chess enthusiasts, where they either observed or played against it - and then completed a survey about the usefulness of the bot. We collected data from survey responses and logged games with the bot.

### Conclusion

#### What we achieved

- A working adaptive chess bot
- Real-time difficulty adjustment
- Context-aware feedback generation

#### What we learned

- Adaptive systems can increase engagement
- Difficulty calibration is critical to enjoyment
- Feedback improves perceived learning

Our project demonstrates that adaptive difficulty and real-time feedback can create a more engaging and personalised chess training experience than traditional fixed-strength bots.

