CGRA151 Project Report

Niamh Ferns
300 632 732
Pixel Chess

Vision

Game concept

A pixel-art chess game where the player is playing against either the computer or another player. The game will follow the standard rules of chess and will be made out of pixel-art. It will also have a menu at the beginning that allows the player to enable/disable a timer and select their colour.

Achievement

I was able to achieve a semi-polished pixel art chess game that implements an optional timer for players and the ability to rotate the board in play. Unfortunately I was unable to even start on the AI due to time constraints and there were some minor features in chess I could not implement like en passant, castling, and check/checkmate. Check and checkmate can however still be observed by players while playing, there is just no lock that forces it in the game.

Technical Challenges

The two largest challenges faced were movement checking and implementation of optional features like board rotation and timer.

I was able to overcome movement checking by making use of the state pattern by making each Chess piece type be a state of chess piece and then having a recursive `getPossibleMoves()` method that is called via dynamic dispatch on that piece when selected.

I was able to overcome the optional features challenge by having a handler API that gates access to the various optional features and adds them dynamically on resetting the board. All of these features are a GamePart and are added to the list of GameParts to be called while running the game when wanted. You can think of this conceptually as hook where each optional component is a hook that attaches to the game when needed.

Reflection

There are parts of this assignment that ended up being way easier/harder than expected. Movement checking was no where near as hard as I was thinking it would be but that's because I spent a lot of time on the groundwork for the game before attempting the main features. Handling access to the board and the various chess pieces was much harder than I though. Giving only one component the responsibility of handling chess pieces felt nearly impossible and it's not something I managed in the end. Given the chance to repeat this assignment, I'd have changed a lot about how I handle various aspects about the chess pieces locations and interaction. Currently they're tied to the cell they're in which makes things very hard. I would also completely redo how I handle *when* to check moves as well. My current implementation doesn't allow me to peak into the future making check/checkmate almost impossible to implement cleanly. I'd say my plan matched the reality fairly accurately.