Program Design Methods and Intro to Programming Python Final Project: Shogi Sim



**Student Information:**

Morris Kim / 2440074174

**Class Information:**

Class: L1BC  
Lecturer: Ida Bagus Kerthyayana / D5757 (might be wrong)

**Binus University International  
School of Computer Science  
2021**

Chapter I – Introduction

When this final project was first introduced to the class, the prospect of being able to make anything we want, limited only by our creativity and time seemed exciting. The only caveat was that an external library must be used. That first night, several hours was spent thinking of feasible and entertaining ideas. During that duration, 3 ideas came to mind: An RTS (Real Time Strategy) game where you control resources and produce units to defeat an enemy; An chatbot-like AI that one can talk to and hold somewhat coherent conversations with; Shogi.

It should now be noted that although the first 2 ideas were considerably challenging, Shogi was not chosen because it was simpler to do. By the time the project started its development, it was already mid-December. After watching a 7 hour tutorial video on TensorFlow, it was decided to perhaps leave the idea for a future project since there might not be enough time to finish things given my own inexperience with dealing with the subject. The first idea was also questionable when considering the time limit given all the preparation needed in creating the assets, design, sound, and etc.

Hence, in the end, Shogi was chosen. With the popularity of the recent release of Netflix’s historical drama *Queen’s Gambit* and everyone’s recommendation to go and watch it, it is only appropriate to work on a variant of chess that is lesser known to the general audience.

This project is completely open source, and can be accessed through my github repository linked here: <https://github.com/morris-necc/shogi_arcade>

Chapter II – Project Specifications

**Project purpose:**

This project’s purpose is to accurately represent a game of Shogi so that people can play the game with either themselves or other people. Another purpose is to introduce Shogi to more people as a fascinating board game.

**Project audience:**

This project’s audience is individuals who are interested in Shogi, or have played Shogi in the past and are interested to play it again.

**Project aim:**

This project’s aim is to simulate a game of Shogi. Every piece on the board must work like it should in the actual board game, and the mechanics of the game must be replicated exactly.

**Project Requirements:**

* Pieces objects, each with their own set of movements
* A promoted version of each piece, again with their own set of movements
* Calculations to highlight which moves are available for every piece
* Calculations for other mechanics such as pinning, checking, and checkmating
* Accurate sound effects featuring an angry Asian man calling you a disgrace for losing
* A simple design to make pieces readable for a western audience

Chapter III – Solution Design

**1. Overview**

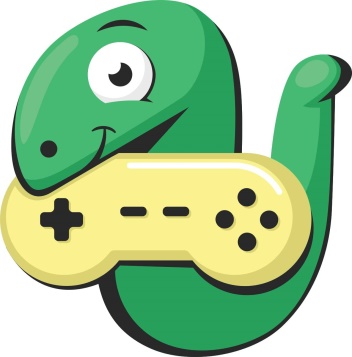
****

Image 1. Arcade logo  
(<https://arcade.academy/_images/arcade-logo.svg>)

*Shogi Sim* uses Python 3.8.1 using arcade 2.5.1 as an external library. It is used to make the window, draw objects on that window, detecting mouse clicks in that window, as well as to load and play the appropriate sounds.

**2. A brief explanation of every piece’s movements**

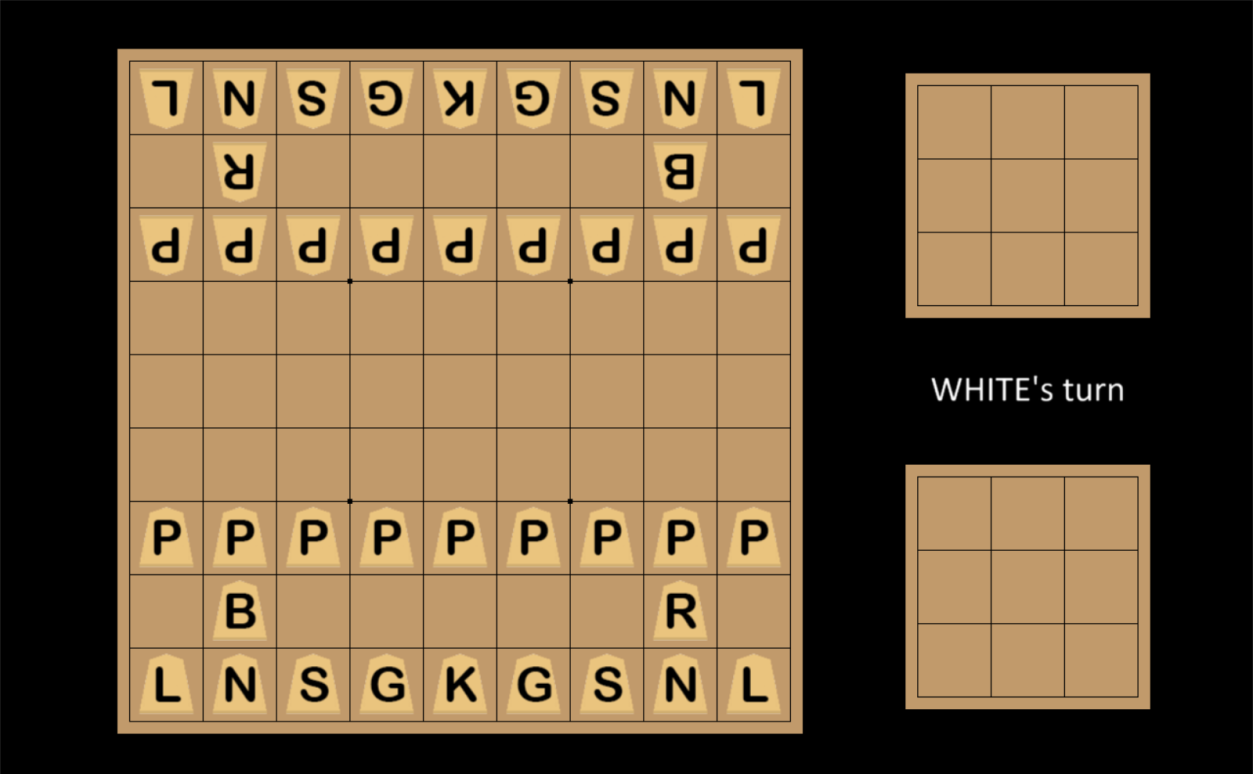


Image 2. The Shogi board

In Shogi, pieces are usually made of wood with a slightly tipped edge not unlike the program’s representation. These pieces usually have kanji characters to represent them, but for ease of understanding, the starting letter for the piece is used instead -- with the exception of the Knight piece (N), so it is not confused with the King.

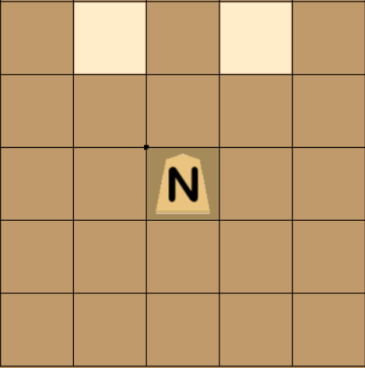
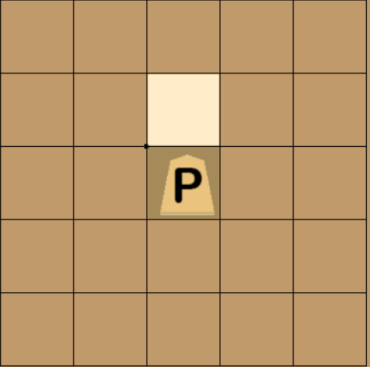


Image 3, 4. The pawn and the knight piece

Although the pawn’s movement in chess and Shogi are similar, pawns in Shogi cannot take on the diagonal and is only able to move vertically in any circumstances. Likewise, the knights in Shogi can only move in a vertical “L” motion and are therefore unable to flexibly jump around the board like its chess counterpart.

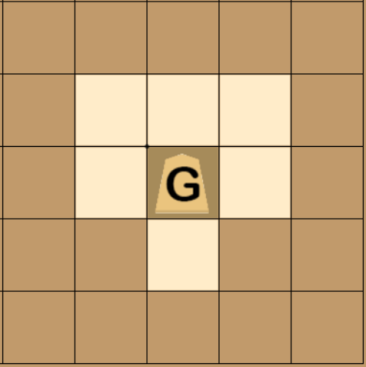
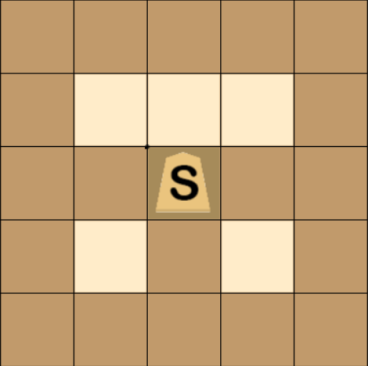
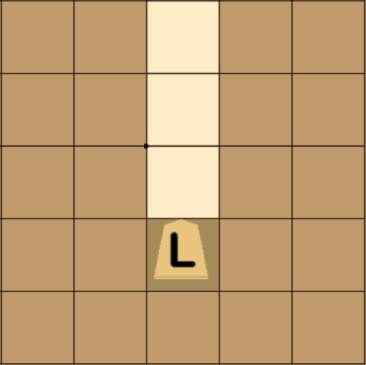


Image 5, 6, 7. The lance, silver general, and gold general

There are 3 new pieces in Shogi: the Lance, Silver General, and Gold General. The lance piece is able to move vertically forwards an indefinite amount of squares until another piece is met; the silver general is able to move similar to the king, but it is unable to move to its sides and the square below it. The gold general, like the silver general, also moves like the king but with the exception of its bottom two corners.

Although there are more differences between chess and shogi mechanically, such as the way its promotion works as well as the unique capability of players being able to place captured pieces back on the board, they will explained with greater depth in later sections.

**3. File dependency and general structure**

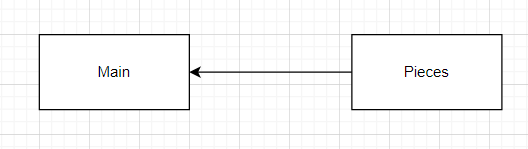
****

Image 8. The 2 files

Disregarding the sound and sprite files, only 2 python files are used in the development of the program. The actual relationship is a little more complicated than this, but this diagram can be used as a general idea of how it works.

As the name suggests, Pieces contain the classes for every single type of piece possible. The main file imports these classes and uses them to perform the logic of the board (which is mostly in the visual side) such as selecting pieces, updating their positions on the board, and giving the promotion prompt when the piece is in range among many other things.

To get a vague sense of what each of these files do, feel free to refer to the class structure diagrams below. MyGameWindow is a class inherited from arcade’s Window class, and it also doubles as the class representing the board, simultaneously giving the visuals and running the game’s logic.

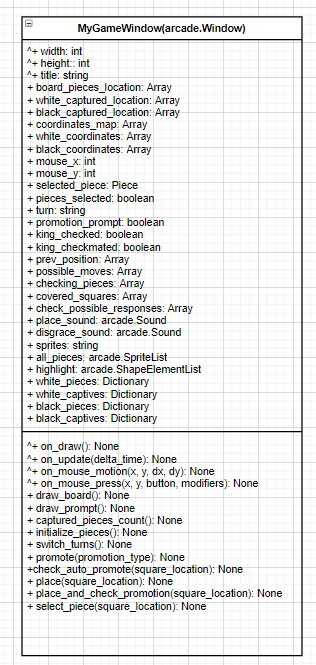


Image 9. MyGameWindow class structure

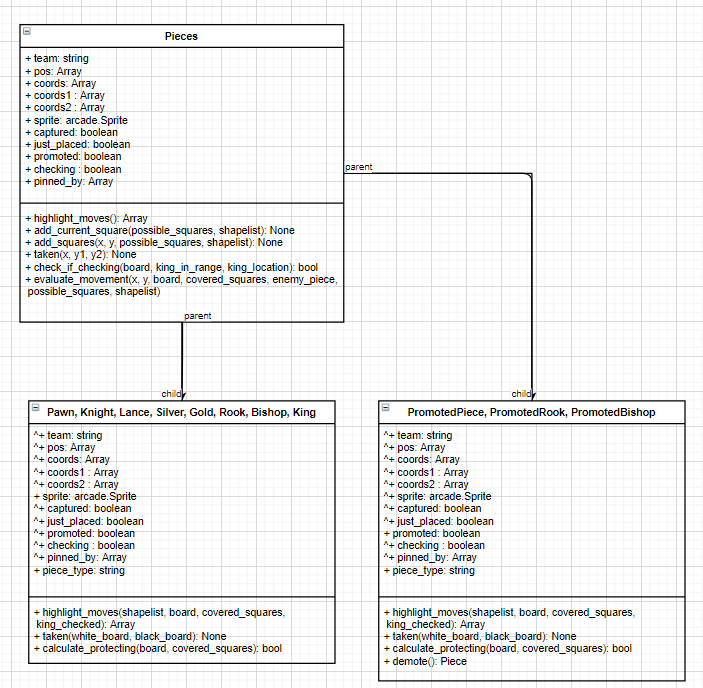


Image 9. Pieces class structure diagram

In order to simplify the diagram, the Pawn, Knight, Lance, Silver, Gold, Rook, Bishop, and King are placed under the same box and PromotedPiece, PromotedRook, and PromotedBishop in another box. They are all mostly similar with the exception of their movement patterns, which are different for every piece.

Another thing to note here is that although King is grouped into with all the unpromoted pieces, it is lacking the taken method. It is placed there as the boxes are quite large and inserting another one will affect the ability to read the diagram without the need to squint.

**4. Visually representing the board and its pieces**

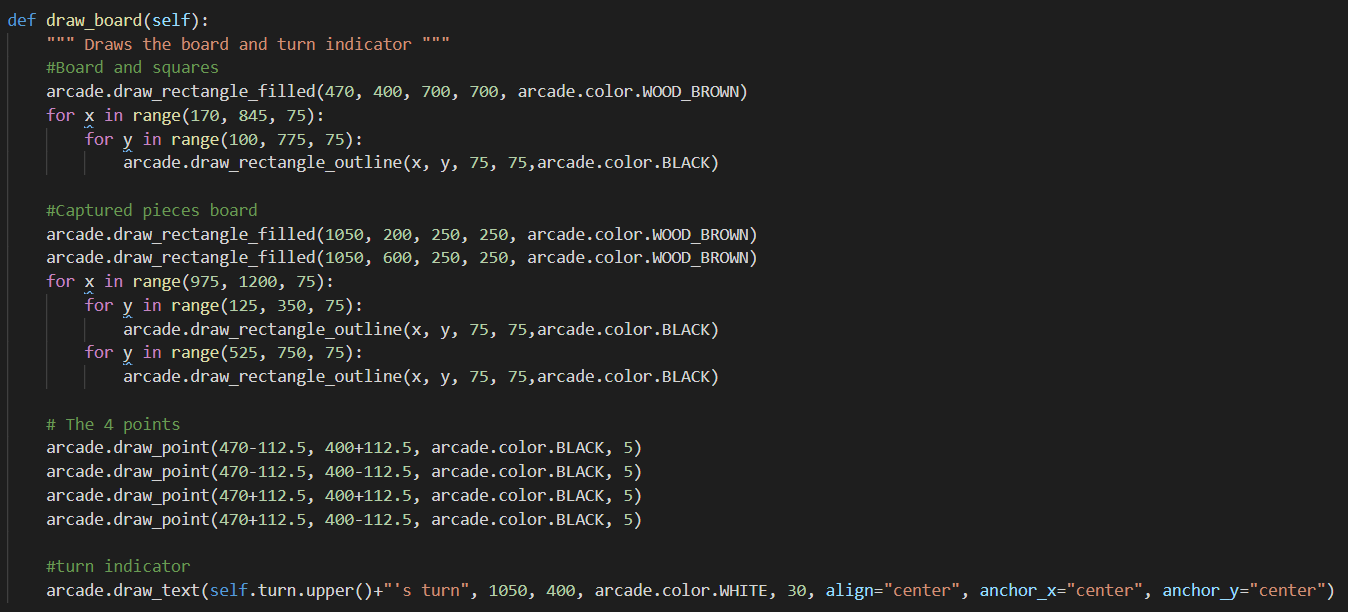
****

Image 8. Drawing the board

Taking advantage of the simple design of the shogi board, this program fully utilizes arcade’s draw functions in order to visualize the board. The board’s background is drawn as a filled rectangle with the preset WOOD\_BROWN color. In addition to that, it might be noted that instead of drawing lines to form the grid, the method of drawing rectangle outlines to form the grid is used instead. This is intended in order to ease development process and make repositioning the grid faster and more efficient. This same process is used to create the smaller boards that are used to keep captured pieces.

The 4 points around the center is calculated with a simple formula of being 1 and a half squares away from the center point of the board where one square has side length of 75 pixels. The turn indicator is put at the side in between the opposing player’s small boards in order to not obstruct the screen.