

Write Up:

When working on this assignment there were a number of features we created that had clear elements of design patterns we have talked about inside this course. One of the clear design patterns we implemented in our code is the template pattern. The means in which we implemented this is when we designed our Player class. IN this class we have a number of methods that are passed are passed and different subclasses of the player class implement these methods depending on the type of class it is. The players Greedy, Random, and Human, inherit from this template class that has the basic frame of what a player should do, and these subclasses override the general use of the player class, depending on what kind of player the user inputs into the command line.

We also use a chain of responsibility design pattern when designing our command line processing for the types of players playing the game. This is done by having processing in our code that basically checks the position the type of player entered into the command line is in and basically hands off the processing to different parts of the code depending on the type and position of the kind of player the user wants playing the game.

We also use a memento design pattern for the undo and redo functionality. This is pretty obvious given the nature of the requirements for undo and redo but essentially we do this through constant storage of the state of our board after every turn/movement and the different variables we utilize to check for wins and draws in the game. Through this storage whenever we call undo and redo we have the ability to restore previous states of the game that we have had.

We utilize an observer design pattern in our check for wins/draws, by having checking methods for wins and draws that notify different objects in our code of if any of the criteria for wins/draws have been met and those objects then output the correct things to the terminal and perform the right calls to end the game.

All in all, the use of these design patterns has made it easier for us to utilize our code for chess through our ability to do things like reusing our relationship with players but now tweaking it to have interaction with the different computer players we are implementing for chess. We can also reuse our checks for if a game is won but just replace it with a checkmate rather than just having missing all pieces denote a win. Our memento design pattern can easily be reused because it just autosaves the state of the board. All we need to do is update the variables we save to denote turns.

Samuel Tigistu:

While working on this project I feel a number of my programming skills improved. I saw drastic organizational improvements in my code. Prior to this project I feel the organizational benefits of object oriented programming kind of went over my head. However while implementing the different design patterns in the code, I really learned how useful object

oriented programming is for things like code reuse. Also, prior to this project I had never built an AI, which although simple I felt was a very valuable piece of information.

Paulos Thomas:

Working on this assignment was worthwhile because it was an opportunity to attempt to put into practice concepts we learned in class. I learned that structuring your plan for the project before starting to write the actual code is important and helps organize your ideas, and also helps reduce the amount of problems or difficulties encountered during the project.