Jason Lee

OOP - HW6 Write-up

In my implementation of the board game Santorini, I made use of multiple patterns. One of the patterns I used is the template pattern when designing the different types of players in the game (human, random, heuristic). There is an abstract Player object that implements some of the necessary methods such as the ones for calculating move_score or initializing. The three different types of players, HumanPlayer, RandomPlayer, and HeuristicPlayer are all subclasses of the Player class, but some methods are implemented differently. For example, each type of player has their own version of the make_move() function to move one of their workers on the board. However, all three types of players share the same move_score() function that resides in the Player parent class.

The second pattern I decided to use is the memento pattern. I utilized this pattern to implement the undo/redo functionality of the game. At each backup, the current game is saved into a concrete memento, and then added to the caretaker's history. The undo function reverts the game back to the previous move, and the redo function updates the game state to the most recent undo.

The third pattern I used is the command pattern. This pattern is prominent in event driven programming, and I use this pattern in both the HumanPlayer class and with the undo/redo functionality of the game. One of the HumanPlayer's methods invokes the user for inputs on the command-line interface for a worker, direction, and a direction to build, and from these input values, the relevant HumanPlayer methods behave accordingly. The same thing happens with undo/redo/next functionality of the game. From the user's decision, the program behaves a certain way.

The fourth pattern I used is the singleton pattern. The singleton pattern restricts the instantiation of a class and ensures that only one instance of the class exists. This pattern is present with the Game class I implemented. There is only one Game going on at a time, and even if there may be Game instances saved as states in the memento class for the undo/redo functionality, there is no more than one game being played at one time.