

Ethan Mathieu

Template for Player

The Player Class uses the Template Design pattern. The Player class has a series of steps to execute a turn of the game. Each of the subclasses RandomPlayer and HeuristicPlayer override a step of the Player class, namely the `make_move()` method. This is a good use of the template design pattern because we can write generic player code in the Santorini class and create iterations of Player object that vary depending on the type of strategy they want to employ.

Singleton for Turn

Only one instance of the Turn class is ever created, during the initialization of the Santorini class. That single instance has its class variables manipulated as the game progresses from turn to turn. Being a singleton that is edited only by updating its class variables allows other files / classes to access the current turn of the game without accepting it as a parameter. This makes sense since there will only ever be one Turn going on at once.

Strategy for PlayGame

In the `main.py` file we set a game. This will allow other expansions of the codebase to include other games like potentially Chess, Checkers, etc, without having to change `main.py`. Each additional game would just need to inherit from PlayGame and implement the `play()` method.

Memento for Santorini. History. SantoriniState

The originator is the Santorini Class, the memento is SantoriniState and the caretaker is the History class. This is a classic implementation of memento, which makes sense since we're trying to support undo's and redo's.