CS 246 - Final Project (Assignment 5)

Spring 2016

Watopoly

PLAN OF ATTACK

Contributors:

Shakil Kanji (20437442)

Ziqi Zhou (20600496)

Steven Yang (20610147)

Breakdown of the project

Task	Date	Person in charge
Discussion about the project	July 11th	Together
Plan of attack	July 12th - July 13th	Zhou
Answer questions	July 12th - July 13th	Shak
UML	July 12th - July 13th	Steven
Submit to Marmoset - Due 1	**July 15th**	**Steven**
Header files (Player)	July 13th - July 15th	Steven
Header files (Game & Text Display)	July 13th - July 15th	Zhou
Header files (Squares)	July 13th - July 15th	Shak
Implementation (Player)	July 15th - July 19th	Steven
Implementation (Game & Text Display)	July 15th - July 19th	Zhou
Implementation (Squares)	July 15th - July 19th	Shak
Debugging and Integration	July 20th - July 23th	Together
Bonus Features	July 23th - July 24th	Together
Submit to Marmoset - Due 2	**July 25th**	**Steven**

Questions

Question. After reading this subsection, would the Observer Pattern be a good pattern to use when implementing a gameboard? Why or why not?

Yes, the observer pattern can work well for the gameboard implementation. Similar to Assignment 4 question 3a, the observer pattern can allow the gameboard class to notify different classes when the state of the game changes. First, a text display class can be notified by the gameboard class when the state of the game changes, so that the text display can update itself and present itself to the user. Second, the squares of the board can be notified when a player lands on them, in order to take the necessary action for that player.

Question. Suppose that we wanted to model SLC and Needles Hall more closely to Chance and Community Chest cards. Is there a suitable design pattern you could use? How would you use it?

The Singleton pattern would be a suitable design pattern to implement Chance and Community Chest cards. Since all players will be drawing from the same pile and never at the same time, one instance of the Chance/Community Chest class can be created, and a global point of access to this instance can be provided through the Gameboard class. The Chance/Community Chest class can implement a function to randomly decide on an action to take on the player who landed on the square.

Question. Is the Decorator Pattern a good pattern to use when implementing Improvements? Why or why not?

Using the Decorator Pattern to implement Improvements would be unnecessary. The Decorator pattern would be useful if a player had more choices of functionality to implement on their property such as Christmas lights or the ability to hand out chocolate. However, since Improvements merely change the tuition rate for players who land on that square, a much simpler implementation can be done within the Square's class.