COMP 2005

Group 8

Design patterns

Our group decided to implement the GRASP approach for our project. The design patterns we used were as follows:

Information Expert: We used the information expert pattern to assign responsibilities to the classes which had the most information available to fulfill it. For example, our GameSetup was responsible for setting up the board as it took user input which directly told us what settings would be implemented in our game.

Creator: By choosing what class should be responsible for creating instances of other classes, we were able to reduce coupling. Our GameSetup class aggregates a lot of GameBoard objects, so we selected it to create the GameBoard instances.

Low coupling: By using the Creator pattern we were able to reduce coupling by selecting the best class to create instances of other classes. We used multiple different classes to reduce the amount that our main GameSetup class had to do. By doing so we can change GameSetup easier without it impacting the rest of our code as much as it would have if we used a lower number of classes.

High cohesion: By creating focused classes on their specific job in the game we were able to increase cohesion in our project. An example of this is our GameBoard class that is created by GameSetup. By giving GameBoard a well-defined task of creating the board, we can split up responsibilities of the classes and make them more well defined and cohesive.

In the future if we had more time we would further implement even more classes to make it even more cohesive and create even less coupling. An example of this would be adding classes for hints, save, load, and similar options in the game. This would further reduce the responsibility of each individual class and allow us to make changes to classes more freely.