Bryan Nguyen

Kevin Yan

Nam Lam

The main key concepts from CS 151 that our team applied to this project included the Object

Oriented Design Process, interfaces, the use of GUI and patterns, and inheritance. During the Object Oriented Design Process, our group decided to start out with creating multiple use cases to plan out our design. By using use cases, we were able to plan out all the necessary classes we needed to start implementation. To make our program simpler, we used what we learned in class about interfaces. We realized that our classes shared a common goal in achieving a certain task. By creating the interface, we were able to quickly implement our methods while saving an extensive amount of code. By taking what we learned in class about GUI, we created JButtons, JFrames, JPanels, JLabels, and action listeners. During our implementation, we also had to strategize our classes in a specific way so that it would match what we learned about the strategy, mvc, and template method patterns. The strategy pattern helped us create two GUI interfaces that had the same methods, but just had a different view in each for the user. The mvc pattern allowed us to group methods in specific classes so each class would have a specific goal in the program. The template method pattern helped us simplify the methods that were relatively the same in some classes and break down our interface. By using inheritance for our classes, we took the methods from our abstract class and created concrete classes.

Throughout this project, our group had to revisit multiple topics such as GUI, GUI patterns, abstract classes, and the Object Oriented Design Process. The reason we revisited GUI at first was because we needed to know how to create frames and then stack panels on top of it. Our group at first had trouble figuring out the layout of our view and it was necessary to look up topics on creating the view of our program. We next revisited GUI patterns because our group wanted to understand exactly what format the project required us to program in. Understanding the patterns earlier as a group, helped us program efficiently and prevented our codes from clashing. While trying to figure out our interface, we realized that in order to break down our interface class. In order to do this, we had to revisit abstract classes and teach ourselves how to create an abstract class to accommodate methods that both our two view classes implemented. Both our view classes were concrete, however, they had similar methods that could have been pulled to the abstract class in order to save code. The last topic we revisited was because the topic was presented earlier within the class. We self studied the objected oriented design process and taught ourselves how to create class diagrams, use cases, and sequence diagrams. Some other miscellaneous topics that we self studied included how to copy an array so that we could perform our undo button and closing a frame.