

Computer Science 209 – Software Development

Programming Project 6b

Code due on Github 11:59PM Friday 20 November

In this project, your team will continue to develop the data model for the game of *Freecell* and view it in a GUI-based application.

The GUI for this program should show the cells and a **New Game** button in a manner similar to the example program. When the application launches, it should instantiate the data model. The model automatically creates the cells and deals cards to the tableaux. The application then instantiates the main window of the GUI, passing it the model. The GUI creates a panel for each cell and adds it using an appropriate layout manager. The button is then added, and a listener is attached to it that will create a new game when the button is pressed. Thus, the user should be able to view the cards and reset this view with a new deal of cards. That's it!

The Project

1. Your top-level model class (let's call it **FreeCellGame**) contains the cells. During its instantiation, **FreeCellGame** instantiates its cells and a deck and deals the cards to the tableaux. This class should also include a **public** method to reset the model (to be run when the button is pressed). This method restores the existing model to its initial state. **FreeCellGame** should also include methods to obtain the *i*th free cell, the *i*th home cell, and the *i*th tableau. The main window class will call these methods after the model has been instantiated to obtain the cells to pass to its new panels.
2. Add the appropriate classes to represent the panels. The panels in the top row show just one card, whereas the panels in the bottom row can show multiple stacked cards. Therefore, you should define at least two distinct panel classes and organize them under an abstract class if that is appropriate. Each panel should contain a reference to the cell it displays. The only method to implement is **paintComponent**. This method should display the wire frame if the cell is empty. Otherwise, depending on the type of panel it is, it should display either the top card or the entire stack.
3. Add a main window class named **AppView** and an application class named **FreeCellApp**. These classes are similar in structure and function to the **MainView** and **GUIApp** classes you have already seen. The latter just includes a **main** method to run a Java application. As always, the application class instantiates the model and the view. The main window class adds the model's cells to the appropriate panels. Much of your work in this class will be to arrange the layout of the panels in the appropriate areas of the main window. Also add a command button for **New Game** with a listener.

4. After you complete steps 2 and 3, you can run the GUI application to view the cards and create new games.