Final Project Submission

5004 Fall 21

1. **Original Proposal:**

I am planning to create the game Freecell. I am thinking that we could use list to implement this game. Because of the limitation of time and knowledge, I will just randomly deal the cards so there may be some unsolvable deals. I felt I may have to do a lot of research to have the GUI for this project.

1. **All design documentation:**

Original project design: https://docs.google.com/document/d/11yFfinIziY2UI9tB56QyFO2BuL0zvm\_Bax13rgYZyms/edit?usp=sharing

My final product meets all my design and objectives I made. I created the freecell game to move one card or multiple cards. Also, I had created the restart function to let the player to restart the current game. In addition, I used Command Pattern to implement the undo function.

1. **How to use your program:**

Compile and run the ModelDriver Class. The project window will pop up and you can click New Game button to play the game.

GitHub link: https://github.com/yiyi-wang1/5004-freecell.git

1. **Code Walk-through:**

I tried to explain all the important parts in my project but the time is longer than 10 mins so I had to speed up my video. Please let me know if anything is not clear.

Video link:

https://youtu.be/q\_RFfQHsSXc

1. **Concept Mapping and Objective Review**

|  |  |  |  |
| --- | --- | --- | --- |
| Concept Level(Basic, Intermediate, Advanced) | Concept | Where to find it | How it was demonstrated |
| Basic | Basic Java programming skills, standard variable usage and casting, basic class construction and use, logical method use, toString overriding | All classes in the file | The base of the project |
| Intermediate | Logical class construction and use, data encapsulation | All classes in the file | The base of the project |
| Intermediate | Testing | FreecellModelTest | Junit Test to test the model |
| Intermediate | Exception handling | View Class and FreecellModel Class | Use try and catch to handle exception |
| Intermediate | Inheritance | PileView Class, EmptyPile Class,  CardImage Class | These three classes extend the JPanel |
| Intermediate | Dynamic dispatching | PileView Class, EmptyPile Class,  CardImage Class | Overriding the paint methods |
| Intermediate | ArrayList usage | FreecellModel Class | The design of the model use arraylist |
| Advanced | MVC design | The whole project | Use MVC to design the project |
| Advanced | GUI construction | The View Class,  PileView Class, EmptyPile Class,  CardImage Class | Create the GUI for freecell game |
| Advanced | Strategy Pattern | The Command interface, MoveOneCard Class and MoveMultipleCard Class | Create different command based on the situation. Both class will have execute and undo method |
| Researched | Command Pattern | The Command interface and The CommandManager Class | Use stack to store the command and previous state of game, pop the stack when the player wants to undo |

Table 1 : Concept mapping for grading consideration

Since I had not started the project when I wrote the Project plan and objectives, I was not sure how much I could do in the project. After I started, I firstly finished the model part and finished the auto collect function (and its helper function). Then I tried to design the multiple cards move methods. After the GUI lecture, I figured out the way to draw the piles. Also, I did some research on the Command Pattern and lastly I implemented the undo function.

|  |  |  |  |
| --- | --- | --- | --- |
| Objective | Met(Y or N) | How it was met | Why it wasn’t met |
| Primary: implement all the basic functions and pass all the tests. | Y | Use ArrayList to implement the model.  Move card is pop the last card from the ArrayList and add the card to another ArrayList. |  |
| Secondary: add the auto-complete function when the cards in cascades can be moved to foundations. Add View for this game. | Y | Auto-collect: use for loop to check each foundation and use for loop again to check cascades and freecells.  View: use the information from lecture to draw the piles. Researched on mouse event and analyzed the location of mouse to get the input from user. |  |
| Tertiary: Have the undo function and allow player to move multiple cards. | Y | Multiple card move:  Analyzed the x and y coordinate to determine which card the player wants to move. If it is not the last card of cascades, then get the card list to analyze if the card list could be moved.  Undo: Researched on the Command Pattern |  |

Table 2 : Objective reflection