3 Use Cases:

Group 5A (Connie Urie, Cloette Owensby, and Bolun Zhang)

CASE 1: Starting Solitaire

1. *Variation 1*
   1. User Goal: To begin a game of Perpetual Motion
   2. User Action:
      1. User clicks on the application icon, then the main menu interface appears
      2. User clicks on the button labeled “Perpetual Motion”, which launches a new window once they click on it.
   3. Software:
      1. Loads the main menu interface
      2. Upon the user clicking “Perpetual Motion”, launches the selected game type (Perpetual Motion).
2. *Variation 2*
3. User Goal: To begin a game of Baroness with Card Mage type cards
4. User Action:
   1. User clicks on the application icon, then the main menu interface appears
   2. User clicks on the CardMage radio button.
   3. User clicks on the button labeled “Baroness”, which launches a new window once they click on it.
5. Software:
   1. Loads the main menu interface
   2. Upon the user clicking the CardMage radio button, passes the selected card type (CardMage) to the class that reads in the card images.
   3. Upon the user clicking “Baroness”, launches the selected game type. (In this case, “Baroness”.)

*3. Variation 3*

1. User Goal: To begin a game of Baroness with a blue background
2. User Action:
3. User clicks on the application icon, then the main menu interface appears
4. User clicks on the “Choose background color” button twice, then selects a blueish color.
5. User confirms this is the color they would like by clicking “Ok”.
6. User clicks on the button labeled “Baroness”, which launches a new window once they click on it.
7. Software:
8. Loads the main menu interface
9. Upon the user double clicking the background color button, launches a window for the user to select their desired background color and saves that color to a variable, which is passed on to the game drivers.
10. Upon the user clicking “Baroness”, launches the selected game type. (In this case, “Baroness”.)

CASE 2: Playing Solitaire

1. *Variation 1*
2. User Goal: To move a card from one stack of cards to another. (Baroness)
3. User Action:
   * 1. User clicks and drags on the card they would like to move from the current stack the card is in to another stack location.
     2. User sees that nothing happens because the move is incorrect.
     3. User clicks and drags on the same card again and drags it to a different stack, the card appears at the desired stack and is no longer highlighted.
4. Software:
   * 1. Tests the value of the card and the values of the top card in the stack the user is trying to move the card to. If the move is against the organizing rule, it rejects it and the board does not change. If the move is accepted, the card moved by the user and the previous top card on the other deck disappear.
     2. Since this first move is invalid, it is rejected, and the board does not change.
     3. Tests the value of the card and the values of the top card in the stack the user is trying to move the card to. If the move is against the organizing rule, it rejects it and the board does not change. If the move is accepted, the card moved by the user and the previous top card on the other deck disappear.
     4. Since this move is valid, the card moved by the user and the previous top card on the other deck disappear.
     5. Checks to see if the user has won the game by checking the score. If the score is at the maximum, the winning sequence is executed.
5. *Variation 2*
6. User Goal: To undo a move they made and then redo that move. (Baroness)
7. User Action:
   1. User clicks on the “Undo” button.
   2. The user observes other possible choices, concludes that they did make the best choice and clicks the”Redo” button.
8. Software:
   1. Fetches the latest record of moves committed by the user (in moveRecord) and copies the latest move in a separate location, then reverses the last move made by the user. Also copies that move into the Redo record.
   2. Goes to the separate location, retrieves the movement information from the Redo record, and performs the last move that was undone.

CASE 3: Finishing Solitaire

1. *Variation 1*
2. User Goal: To finish the game of Solitaire (Perpetual Motion Type)
3. User Action:
   * 1. User moves the last available card, achieving the maximum score.
     2. The program displays a “Congratulations! You won!” message.
4. Software:
5. Checks the current score with the maximum possible score.
6. If the maximum possible score has been reached, displays a “Congratulations! You won.” message.
7. *Variation 2*
8. User Goal: To quit the game of Solitaire (Perpetual Motion Type) before they are out of cards because they know they are losing and are frustrated.
9. User Action:
   1. User clicks on the “x” button, to close the window.
10. Software:
    1. Closes the window.