Classic Solitaire Card Game

- **Xteam Number:** 245
- **Team members:**
    - Nada Elkordi: elkordi@wisc.edu
    - HAOZHAN YUAN: hyuan63@wisc.edu
    - Gabriel Kim Wilkinson: gwilkinson2@wisc.edu
    - Russell Ren-Shing Cheng: rrcheng@wisc.edu

- **Problem description:**

Adopted from: https://bicyclecards.com/how-to-play/solitaire/

- Solitaire is a popular and classic card game.

The objective:

- To combine and sort cards belonging into piles.
- A pile needs to be sorted based on card rank.

- Cards are *ranked* as follows:

King(high), Queen, Jack, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1 (low).

- Cards have two *colors*:

red and black.

- The game is *won* when the player builds four piles for each *suit*:

Hearts, Diamonds, Spades, and Clubs.

- The sorted piles are called *foundations*

- A full card set is partitioned as follows:

  1. *The Tableau*: Seven piles that start face down.
  2. The Foundation: Initially empty four piles for each suite
  3. *The Hand pile*: The remaining cards that are not in the Tableau. Initially face down
  4. *The waste pile*: Contains cards that cannot be placed in any of the above piles.

Rules of the game:

1. Each suit in the foundation should start with the lowest rank (1).

   - Cards are added to the foundation pile in ascending order of their rank.

2. Only a face down card from the tableau can be opened with a click to be face up.

3. You can move a face up card from one tableau pile (Card 1) to another tableau pile with (Card 2) if the following conditions apply.

   o Card 1 is of different color from Card 2

   o Card 1 is of a rank that is directly less than the rank of Card 2.

4. If a tableau pile is empty, you can move a face up King card to this tableau pile.

5. When there are no moves possible among the foundation piles, the player can open a card from the hand pile into the waste pile and put it under a tableau pile.

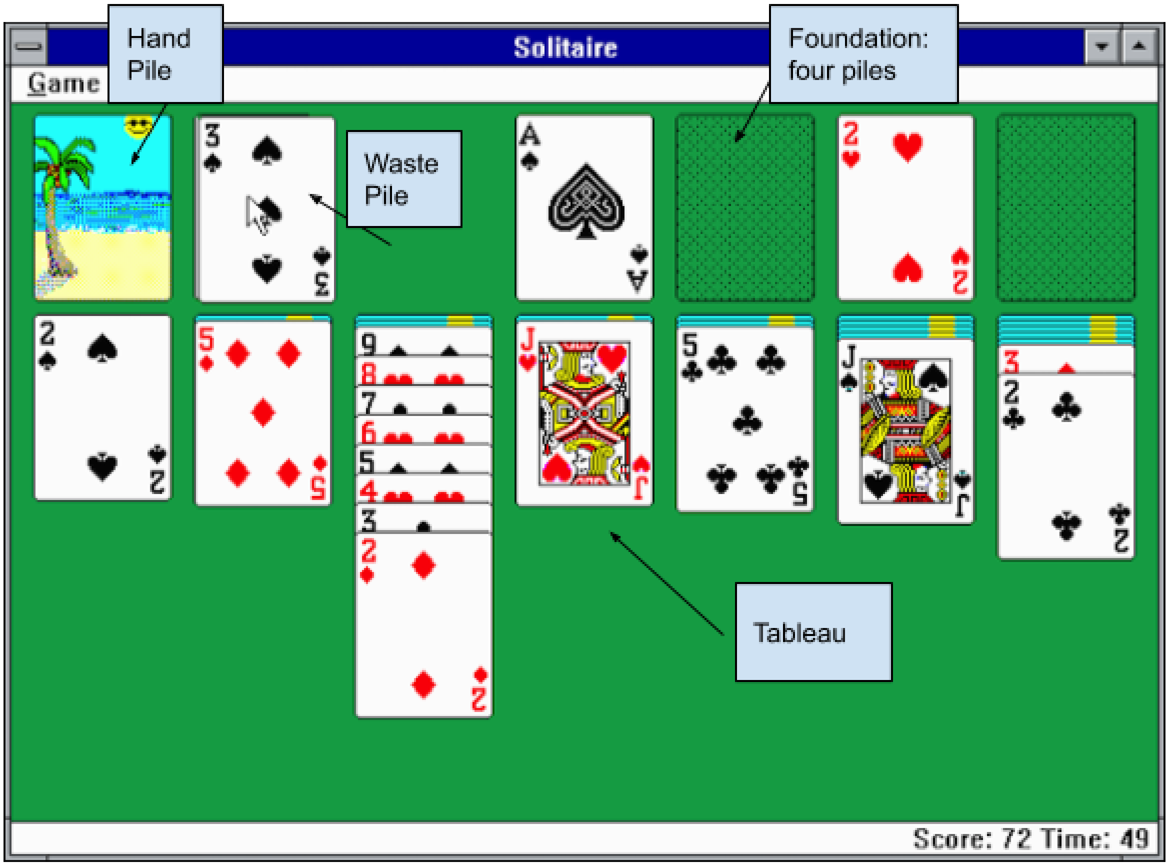
6. A card can go directly from the hand pile to any of the foundation piles.

7. The game ends when all cards are sorted into the foundation pile

Primary stakeholder:

This game attracts users beginning from 7 years old. It is a very popular game.

Figure1: Graphical User interface:



Data:

- The data that will be used in the implementation:

  1- Stack ADT: every pile of cards will be represented by a stack

  2- Integers variables:

  a. Card dimension: width, height

  3- Enum variables

  a. Card suit: heart, spade, diamond, and club

  b. Card rank: King, Queen, Jack, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1.

  c. Card color: Red, and black

- Images
  - The game requires need to have 52 images of the cards and the back of the card

Classes:

I did read the following URL: http://web.engr.oregonstate.edu/~budd/Books/oopintro3e/info/chap09.pdf

1- Class Card:

- This class represents the main class in the program. It will contain:

  ·Variables that represents the card dimensions, suit, color, rank, xlocation, ylocation on the interface.

·Methods: isFaceUp(), getRank(), getSuit(), flip(), getColor(), draw()

·Image of card

·Image of back of card

2- Class Pile:

- Every pile will be represented using a Stack ADT. This class will override all the methods from the Stack ADT: top(), pop(), isEmpty(), size() and peek().
- PileNode is an inner class to represent the stack node.
- The stack will be implemented using a linked list.

All the following classes will extend the Pile class:

3- Class FoundationPile:

- This class represents a foundation pile of cards at the top of the playing surface, the pile will be built in suit from 1 to king.
- This class will have a variable representing its suit, a method getSuite(), addCard().

4- Class WastePile

- Used to contain the waste pile. Has method moveAllToHandPile() to move all cards from the waste pile to the HandPile

5- Class HandPile:

- To contain the hand pile
- To implement the draw method and to check if a card from the hand pile can go to any of the foundation piles
- The class has a method called checkAddFoundation()

6- Class Tableau:

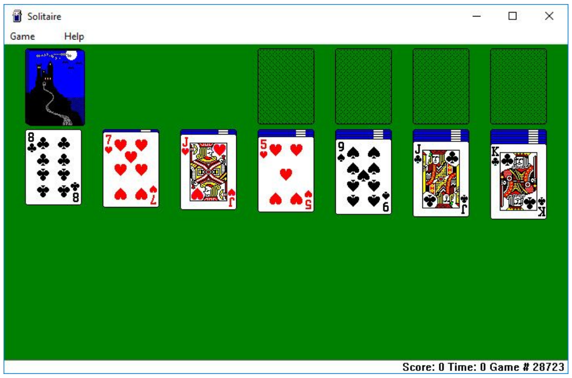
- Contains a pile of tableau cards.
- The cards will have some face up and others face down.

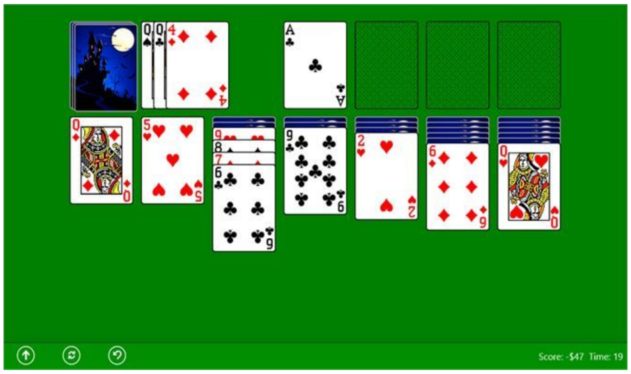
7- Class SolitaireGame:

- This class represents the game.
- It will contain:
  ·Variables for HandPile and WastePile
  ·Arrays representing FoundationPiles, and Tableau.
  ·Main method to randomly distribute cards over the piles.
  ·Methods
    odraw() to draw the entire game.
    oreset() to reinitialize the game.

Input Data File Format:

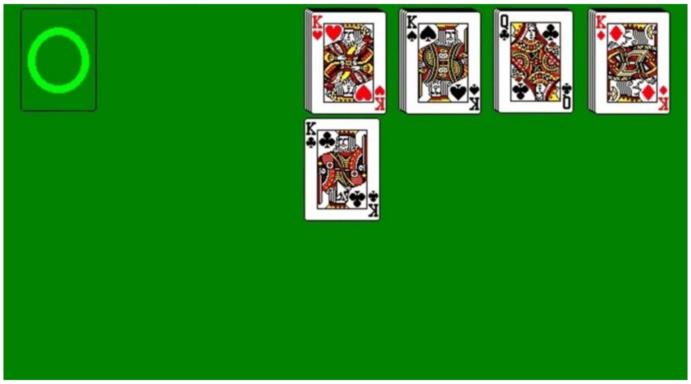
- Images of the game
- User input clicks





Output Example:

- Change in The GUI with every event click
- Game Progress: Win or Lose



Milestones Description:

- Milestone1:
  - Initialize GUI class (Taken from a GUI library) to draw Game window
  - Assemble library images and other prerequisites

- Milestone2:
  - Implement Card class
  - Implement Pile class
  - Implement the piles:

    1- Class FoundationPile
    2- Class WastePile
    3- Class HandPile
    4- Class Tableau

- Milestone3:
  - Implement the game application class (Game Logic): Class SolitaireGame

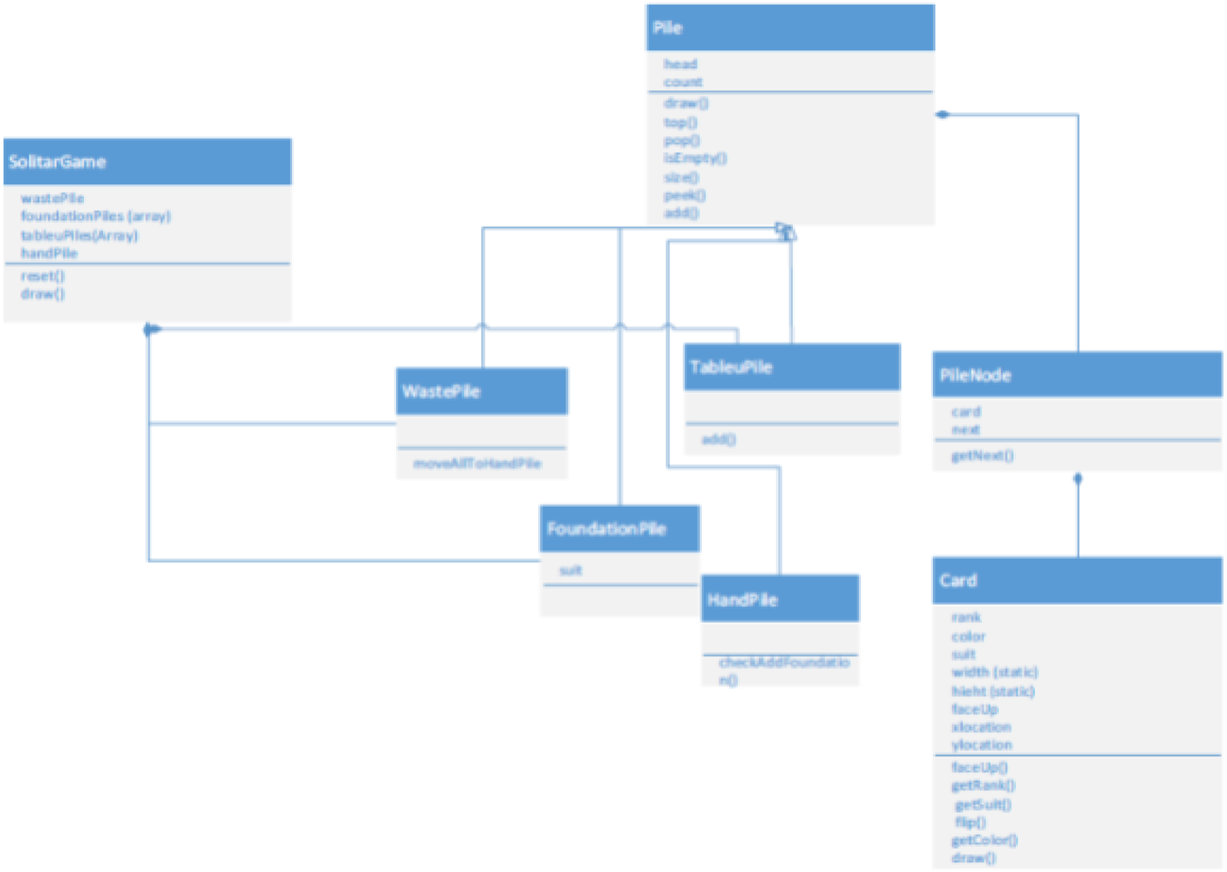
- Milestone4:
  - Implement events:
    - User inputs: clicks
    - GUI changes due to user inputs (e.g., card will be face up instead of face down after a click)

- Milestone5:
  - Result of the game: Output message
    - Win: “Congratulations! You win”
    - Lose: “You lose! Try again”

Assign tasks:

- Task1:
  - Milestone2: Russell
- Task2:
  - Milestone3: Gabriel
- Task3:
  - Milestone1 and Milestone5: Nada
- Task4:
  - Milestone4: Haozhan

UML Diagram:



GitHub URL:

https://github.com/Haozhan-Yuan/Xteam245