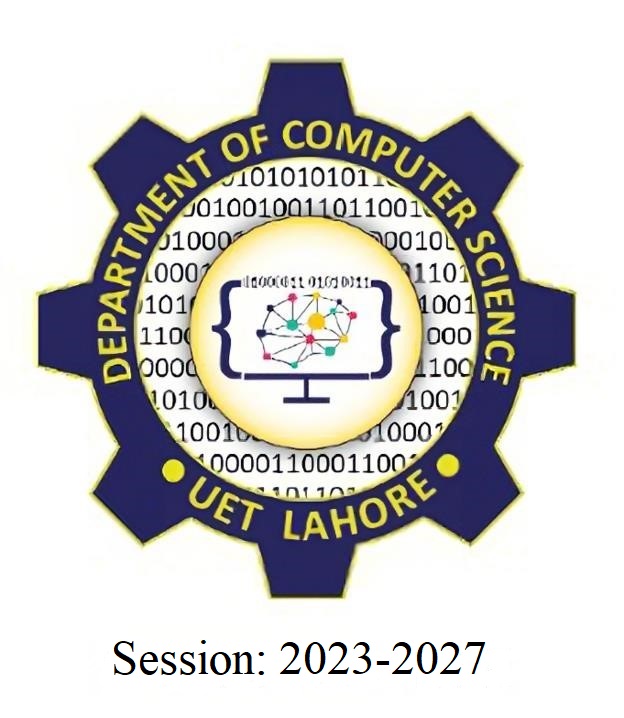
**Solitaire**



**Submitted by:**

Muhammad Fahad 2023-CS-124

**Supervised by:**

Sir Nazeef Ul Haq

Sir Waseem

**Department of Computer Science**

**University of Engineering and Technology Lahore Pakistan**

Table of Contents

1. Overview……………………………………………………………………………..3
2. Features………………………………………………………………………………4
3. Data Structures i. Stacks ii. Queues iii. Arrays………………………………………4
4. Wireframes of the Game……………………………………………………………..5

Table of Figures

Figure-1: MainGame………………………………………………………………………….5

Figure-2: RulesGame………………………………………………………………………….5

Figure-3: PlayGame…………………………………………………………………………...6

Figure-4: WinGame…………………………………………………………………………...6

# **Solitaire Game Report**

# **1. Overview**

# This project involved developing a classic Solitaire card game using Pygame. The game follows traditional Solitaire rules where players move cards across piles to build four foundation piles (one for each suit) in ascending order from Ace to King. The project highlights game development principles, object-oriented programming, and the practical application of data structures like stacks and queues for card management.

# **Objective:**

# Move all cards from the tableau to the foundation piles.

# **Mechanics:**

# Tableau: Consists of seven columns; the first card in each is face-up, while others are face-down.

# Stockpile: Draws cards to be used when no more moves are available.

# Foundation Piles: Build each pile by suit in ascending order, starting with Ace.

# **Moves:**

# Cards can be moved between tableau columns if they follow a descending, alternating color sequence.

# Kings can be placed in empty tableau columns.

# Cards from the waste pile can be placed on the tableau or foundation if valid.

# **Winning Condition:**

# Complete all four foundation piles from Ace to King.

# **2. Features**

# **Interactive UI:**

# Drag-and-drop functionality for card movement.

# Buttons to start a new game, access game rules, and navigate back to the main menu.

# **Score and Timer:**

# A timer to track the duration of gameplay.

# Scoring system that rewards strategic moves (e.g., moving a card to a foundation).

# **Game Logic:**

# Automated detection of valid moves.

# Comprehensive win detection and notification.

# **3. Data Structures**

**Data Structures**

**i. Stacks**

Used for LIFO operations in tableau, foundations, and waste pile.

**ii. Queues**

Potentially useful for reshuffling cards back into the stockpile.

**iii. Linked Lists**

Used to represent decks, piles, and for card management.

**iv. Custom Classes**

Card, Deck, Tableau, Foundation, and Stockpile classes represent card collections and their behavior.

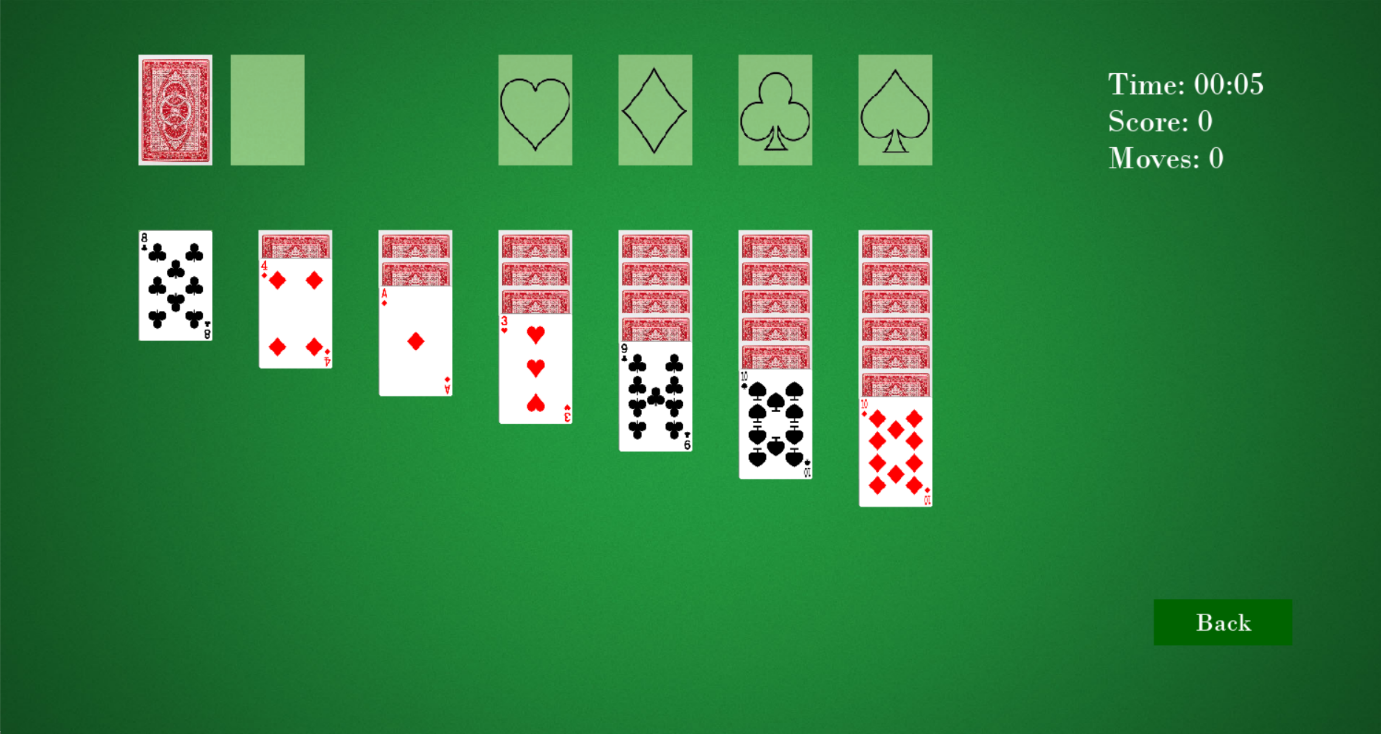
**4. Wireframes of the Game:**



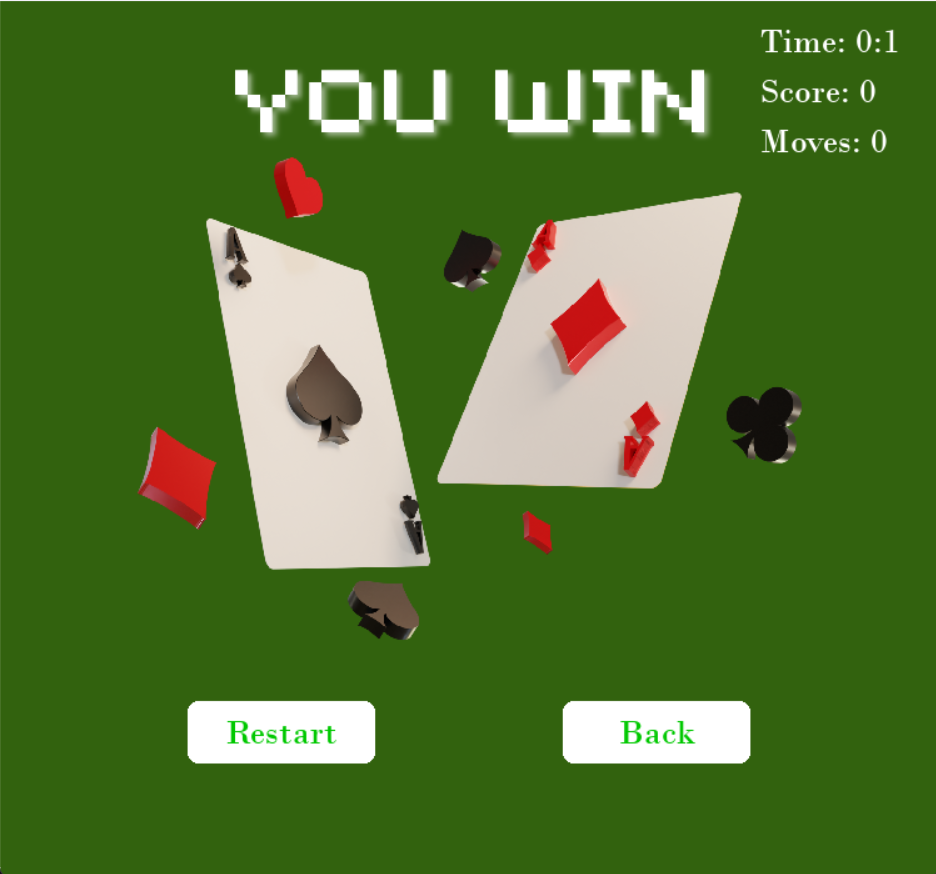
*Figure-1: MainGame*

**

*Figure-2: RulesGame*

**

*Figure-3: PlayGame*

**

*Figure-4: YouGame*