CSC200 – Data Structure & Algorithms

Mid Term Project Guidelines

Guidelines:

- Gitlab Repositories should be created before next lab and add nazeefulaq.uet@gmail.com as a collaborator.
- Progress on project will be tracked from Gitlab accounts
- Project will be evaluated based on rubrics described in the documents
- Name of repository should be in this format: CSC200M24PIDXX e.g. if project id is 9 then repository name should be CSC200M24PID09. Note that project ID is your roll number.
- This project is individual.
- The deadline of this project will be November 07, 2024.
- You have to push your code in your GitLab repository on daily/ continuous basis whenever you update your code.

Section	Details
Project Scope	Develop a Solitaire game following Klondike rules, with features like card moves, shuffling, undo/redo, and win conditions. You can play the game using this link https://solitaired.com/ for better understanding that how it works.
Key Features	Card representation, game setup, valid moves, undo/redo system, and win condition.
Data Structures to Use	
Stack (LIFO)	Purpose: Represent tableau columns and foundation piles for card movement.
	Implementation: Use stacks to model piles where only the top card is visible and others are face down.
Queue (FIFO)	Purpose: Manage the stockpile (draw pile) where cards are drawn one or three at a time.
	Implementation: FIFO queue ensures the proper order of cards for drawing.
Array/List	Purpose: Store the deck of 52 cards before shuffling and dealing.
	Implementation: Efficiently shuffle and access cards.
Linked List	Purpose: Represent tableau piles for easy manipulation of cards.
	Implementation: Manipulate cards within the tableau using linked lists.
HashMap/Dictionary	Purpose: Track card positions, states (face-up or face-down), and pile locations.
	Implementation: Use dictionaries to map tableau and foundation piles to cards for quick access.
Game Mechanics	
Initialization	Shuffle the deck and deal seven tableau piles with increasing numbers of cards (1 to 7) with the last

	card face-up.
Moves	Implement moves between tableau piles, to the foundation, and from the stockpile.
Foundation	Build foundation piles from Ace to King by suit.
Victory Condition	The game is won when all cards are moved from the tableau and stockpile to the foundation piles.
Project Phases	
Phase 1: Card and Deck	Design a card class with attributes (suit, rank, face-up/down). Shuffle the deck.
Phase 2: Tableau and Foundation	Design the tableau and foundation piles using stacks and linked lists. Implement card movement functions.
Phase 3: Game Logic	Implement move validation, stockpile drawing, card flipping, and undo functionality.
Phase 4: User Interface (Optional)	Create a text-based or graphical interface to interact with the game. Display the tableau, stockpile, and foundation.
Validation and Testing	
Legal Moves	Ensure only legal moves are made (alternating colors, sequential ranks).
Card Flipping	Test card flipping after a card is moved from a tableau pile.
Win Condition	Verify that the game correctly identifies when the player has won.
Game Flow	Test the flow from initialization to game completion to ensure rules are followed.
Bonus Features (Optional)	
Undo/Redo System	Use stacks to track moves and enable undo/redo functionality.
Hint System	Suggest valid moves to the player.
Timer and Scoring	Add a timer and scoring mechanism for player performance tracking.
Submission Requirements	
Code	Well-documented code with comments explaining key parts.
ReadMe	Explain the project, how to run the game, and list dependencies.
Test Cases	Provide test cases that demonstrate game functionality.
Report	A comprehensive PDF report of project