Blackjack Design Document

Dawson Melin, Aamir Patel, Archi Amitkumar, Maulik Nagarji, Abraham Paikkattuvayalil

Sheridan College

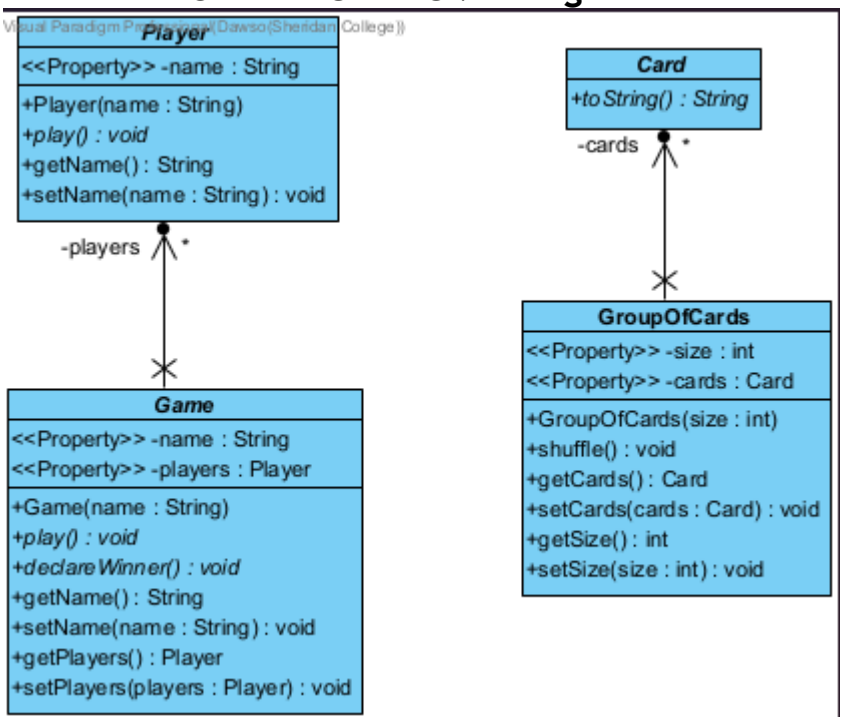
SYST17796: Software Design and Development

Sivagama Srivivasan

2024-02-14

Contents

Starter Code UML Diagram



Project Background and Description

Blackjack Rules: https://bicyclecards.com/how-to-play/blackjack/

Blackjack is a game of luck and strategy where the player must get as close to 21 without going over, while also having more than the dealer. The player can choose the desired amount to bet with each hand and if the player succeeds, they double their money or lose it. Additionally, if the player and dealer's hands result in a tie no one wins, and no money is gained or lost. Our group intends to have all the rules of blackjack implemented in our final design. Players will be able to choose an amount to 'cash in' with and if they double their money or more and cash out, they win.

The base code is written in Java and consists of four classes: Card, Game, GroupOfCards, and Player. The Card class is declared as abstract and includes a toString() method for card representation. The Game class is designed to model various games and includes properties, getters, and setters for the game's name, an array of objects for players, as well as methods for game play and determining the winner. The GroupOfCards class contains an ArrayList for the group of cards, a size attribute representing the number of cards, a constructor for GroupOfCards, getters and setters for cards and size, and a method for shuffling the cards. Finally, the Player class represents individual players, featuring an instance variable, a constructor for initializing a player with a unique identifier, getters and setters for the player's name, and a player method for specific player actions.

Project Scope

Dawson Melin will handle the game rules, logic, and mechanics for Blackjack. Aamir Patel will manage user input, such as how players interact with the game. Archi Amitkumar will work on the player class, focusing on player actions and status. Maulik Nagarji will take care of the dealer class, deciding when the dealer should hit or stand, similar to players. Abraham Paikkattuvayalil will handle the card deck, dealing cards, shuffling, and keeping track of them. Though tasks vary in complexity, everyone needs to work together for the game to run smoothly.

The finished project of the game will have a clear text representation of both the player and dealer’s cards and value, the ability for players to register and keep track of scores, the ability for players to bet, hit, stand, double down, split and cash-in/out, and finally, the game will have all the logic and rules of Blackjack.

High-level requirements

- Ability for the player to interact with the game (Stand, hit, split, double-down, bet, cash-in/out)
- Ability for users to register with the game.
- Ability to display the player's current balance.
- Ability to display the player and dealer cards and the total value of each hand.
- Ability to determine if the player has won (Player cashes out with double or more the amount cashed in with)
- Dealer acts according to rules

Implementation Plan

GitHub Repo: https://github.com/Dawson991/BlackJack-Junction.git

We intend to use this GitHub repository to save all changes using version control and collaborating as a team. In the main, there is a folder for UML diagrams, text documents, and the project code. Expectations of each member are to check the repository daily or weekly for changes so every member stays up to date on the changes made so one does not fall out of the loop.

Coding standards we intend to implement include using commenting to explain complex code, formatting (Using whitespace after operands and proper indentation), naming conventions (functions should use camelCase and class names should use PascalCase), and writing modular code for reusability. Tools we will be using include NetBeans, GitHub, and Visual Paradigm.

Design Considerations

This base code effectively demonstrates encapsulation using instance variables and getter and setter methods. In the Player class, the name variable is made private and accessed through the getName() and setName() methods. Similarly, in the GroupOfCards class, encapsulation is evident with the private instance variables cards and `size`, which are accessed by other classes through the getCards() and getSize() methods.

Delegation is illustrated in both the GroupOfCards and Game classes. In the GroupOfCards class, the `shuffle()` method delegates the responsibility of shuffling the cards to the Collections.shuffle() method without necessitating updates to other classes. Similarly, in the `Game` class, delegation occurs with methods like playGame() and declareWinner(), which could be delegated to other classes, such as the Player class for determining scores and conditions.

The base code also demonstrates flexibility and maintainability. For instance, the Player class allows different types of players to register with the game by extending it to other classes for use. Additionally, the Game class encapsulates all game-related logic, showcasing flexibility and maintainability. If new features need to be added, the class can be extended or modified without impacting existing functionality, ensuring ease of maintenance and adaptability

Team Contract

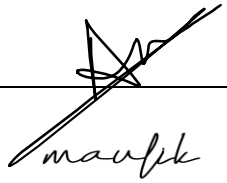
SYST 17796 TEAM PROJECT

Team Name: **Blackjack Junction**

Please negotiate, sign, scan and include as the first page in your Deliverable 1.

Please note that if cheating is discovered in a group assignment each member will be charged with a cheating offense regardless of their involvement in the offense. Each member will receive the appropriate sanction based on their individual academic integrity history.

Please ensure that you understand the importance of academic honesty. Each member of the group is responsible to ensure the academic integrity of all of the submitted work, not just their own part. Placing your name on a submission indicates that you take responsibility for its content.

Team Member Names (Please Print)	Signatures	Student ID
Project Leader: Dawson Melin		991701686
Archi Patel		991706059
Aamir Patel		991715970
Abraham Cherian		991716004
Maulik Nagarji Thakor		991687463

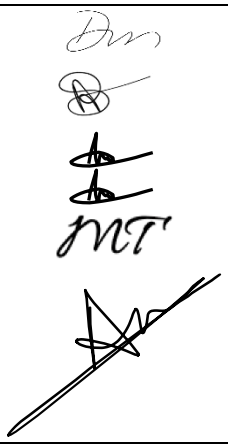
For further information, read Academic Integrity Policy here :
https://caps.sheridancollege.ca/student-guide/academic-policies-and-procedures.aspx

By signing this contract, we acknowledge having read the Sheridan Academic Integrity Policy

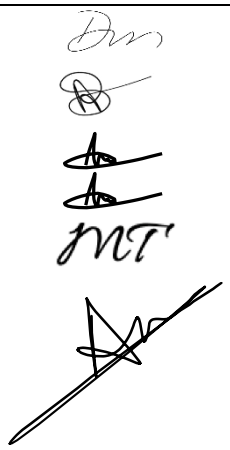
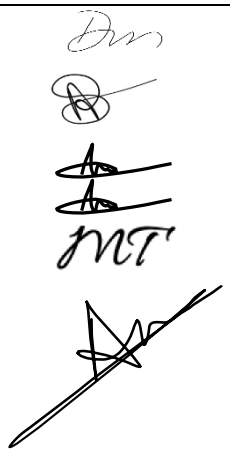
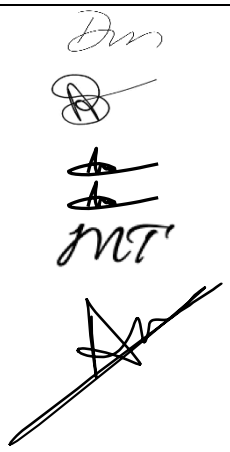
Responsibilities of the Project Leader include:

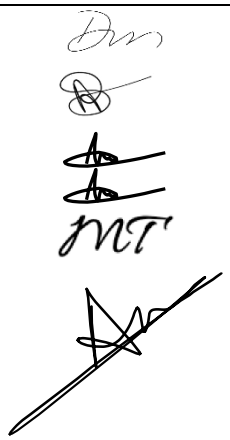
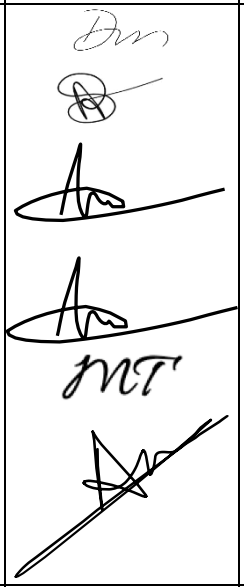
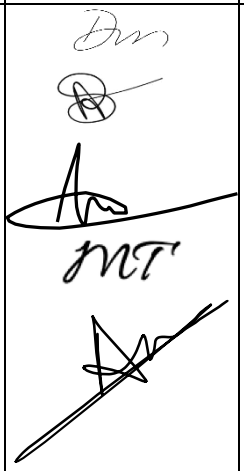
- Assigning tasks to other team members, including self, in a fair and equitable manner.
- Ensuring work is completed with accuracy, completeness and timeliness.
- Planning for task completion to ensure timelines are met.
- Notifying the professor of any issues in a timely manner so that corrective measures can be taken.
- Any other duties as deemed necessary for project completion.

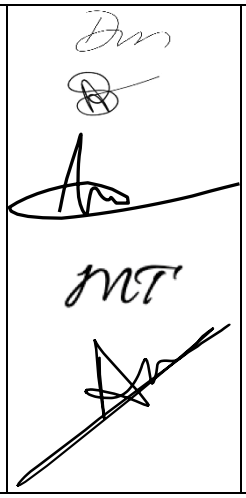
What we will do if . . .

Scenario	**Accepted initials**	**We agree to do the following (Put an X corresponding to your choice in each box)**
Team member does not regularly attend team meetings and/or does not respond to communications in a timely manner.		Project leader emails the student citing the concerns and cc's the professor so they are aware of the situation at the very onset _X_ (**Mandatory**). a) ___In addition to above, the leader/team will (add your own content here):
Team member does not deliver component on time due to severe illness or extreme personal problem.		a) Team absorbs workload temporarily X b) Team seeks advice from professor ___ c) Team shifts target date if possible ___ d) ___Other (specify):

Scenario	**Accepted initials**	**We agree to do the following (Put an X corresponding to your choice in each box)**

Team member has difficulty delivering component on time due to lack of understanding or ability.		a) Team reassigns component ___ b) Team helps member X c) Team member must ask professor for help d) ___Other (specify):
Team member does not deliver component on time due to lack of effort.		a) Team absorbs workload ___ b) Team member(s) ask professor to request a Participation Form from all team members. This *may* result in individualized grades being awarded for a deliverable __ c) Both a. and b. above X d) ___Other (specify):
Team cannot achieve consensus leaving one or more member(s) feeling that their voice(s) is/are not being heard in a decision which affects everyone.		a) Team agrees to abide by majority vote X b) Team seeks advice from the professor ___ c) ___Other (specify):
Scenario	**Accepted initials**	**We agree to do the following (Put an X corresponding to your choice in each box)**

Team members do not share expectations for the quality of work on a particular deliverable.		a) Team members will draw on each other's strengths to help bring the quality of the deliverable to a minimal acceptable level ___ b) Team votes on each submission's quality X c) Team member(s) ask professor to request a Participation Form from all team members, which may result in individualized grades being awarded for a deliverable _____ d) ___Other (specify):
Team member behaves in an unprofessional manner, e.g. being rude, uncooperative and/or making one or more member(s) feel uncomfortable.		a) Team agrees to avoid use of all vocabulary inappropriate to a business/college setting ___ b) Team attempts to resolve the issue by airing the problem at a team meeting _____ c) Team requests a meeting with the professor to discuss further X d) ___Other (specify):
There is a dominant team member who insists on making all decisions on the team's behalf leaving some team members feeling like subordinates rather than equal members		a) Team will actively solicit consensus on all decisions which affect project direction by asking for each member's decision and vote X b) Team will express subordination feelings and attempt to resolve issue _____ c) Team seeks advice from the professor ___ d) ___Other (specify):
Scenario	**Accepted initials**	**We agree to do the following (Put an X corresponding to your choice in each box)**

Team has a member who refuses to participate in decision making but complains to others that s/he wasn't consulted		a) Team forces decision sharing by routinely voting on all issues _____ b) Team routinely checks with each other about perceived roles X c) Team discusses the matter at team meeting