

Advanced Blackjack Strategy Software Design Specification

5-31-2021 – v2.0
Connor Finch (cmf)
E. Mielke (enm)
E. Hsu (eh)
Y. Chen (yc)

Table of Contents

1. SDS Revision History	1
2. System Overview	2
3. Software Architecture	3
4. Software Modules	4
4.1. Strategy Practice	4
4.2. Blackjack Gameplay	5
4.3. Cards and Deck	6
4.4. Betting Module	8
4.5. Visual Interface	9
4.6. Card Counting	10
4.7. Betting Deviations	11
4.8. Basic Strategy	11
5. Dynamic Models of Operational Scenarios (Use Cases)	13
6. References	13
7. Acknowledgements	14

1. SDS Revision History

Date	Author	Description
5-9-2021	cmf	Created the initial document.
5-9-2021	cmf	Completed system overview description.
5-9-2021	cmf	Created Software Architecture Figure and began to provide module descriptions.
5-10-2021	cmf	Finished the rest of the module descriptions from the Software Architecture section.
5-10-2021	cmf	Updated the TOC with the names of software modules/submodules to match the Software Modules section.

5-10-2021	cmf	Started working on Section 4, Software Module.
5-10-2021	cmf	Removed “Basic Tutorial” module from Software Architecture figure and made the submodules the modules themselves.
5-10-2021	cmf	Finished revising all of Section 3.
5-10-2021	cmf	Finished the “Role and Primary Function” and “Interface Specification” sections for every module.
5-11-2021	cmf	Finished the “Design Rationale” and “Alternative Designs” sections for every module.
5-11-2021	cmf	Finished Diagrams for Use Case section.
5-11-2021	cmf	Created and added diagrams to all modules in section 4.
5-12-2021	cmf	Proofread and made final revisions.
5-12-2021	cmf	Updated TOC with correct section names and page numbers.
5-12-2021	cmf	Added colors, keys, and labels to the figures that needed it in this document.
5-12-2021	cmf	Added citations to “References” section.
5-12-2021	cmf	Finished “References” and “Acknowledgments” sections.

2. System Overview

The proposed system offers an abundance of methods and strategies that will give the user an edge while playing blackjack games. These features include the ability to simulate blackjack games, so the user can hone the blackjack strategies of their choice. The system gives the option to practice many types of strategies such as card counting, betting deviations, and basic strategy individually or simultaneously depending on the user’s skill level.

3. Software Architecture

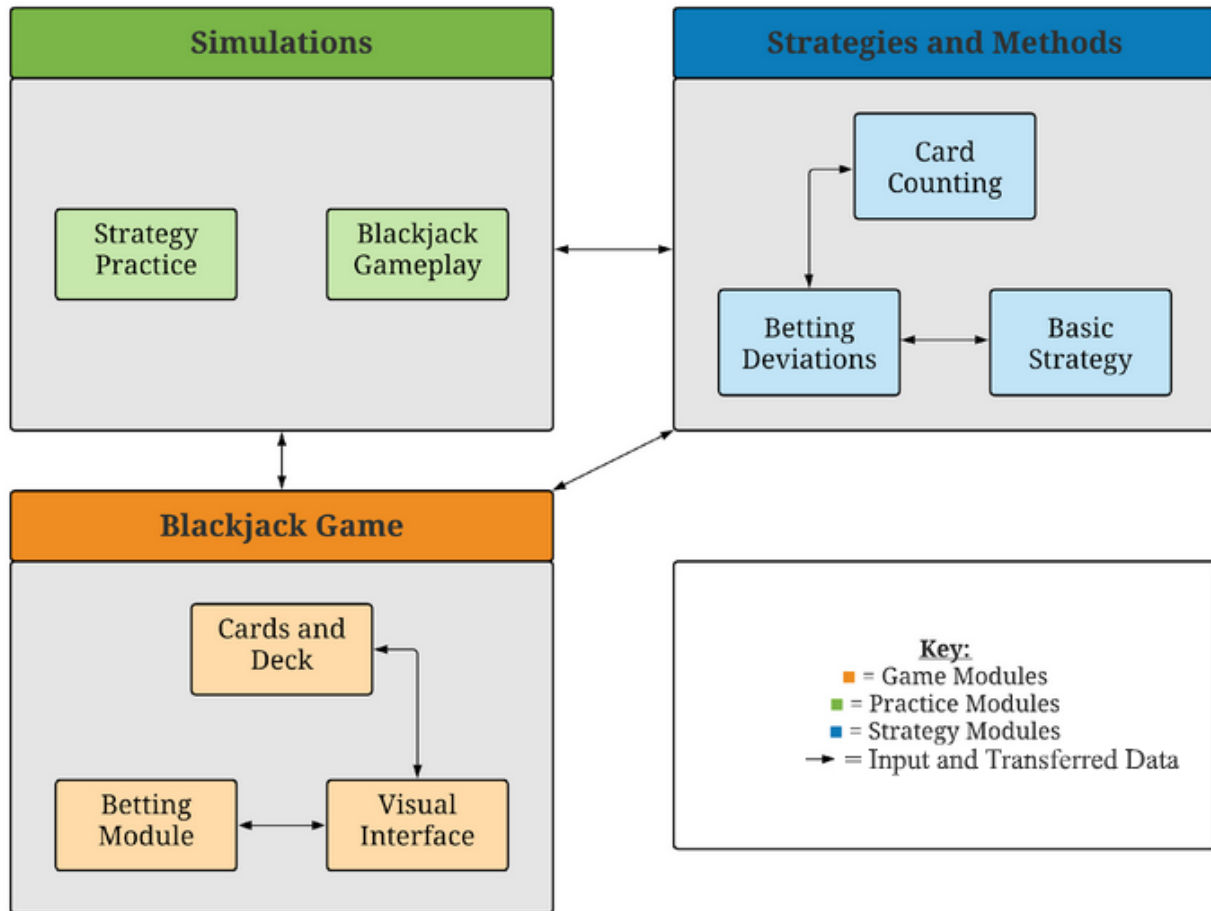


Figure 1: A software architectural model of the proposed system.

- **Simulations:** A group of modules that simulates different blackjack situations, so the user can either practice or test their skills.
 - Strategy Practice
 - This module simulates different blackjack hands, so the user can practice the blackjack strategy of their choosing. There will be no Betting Component module involved with this module because it is strictly meant for practicing. Though there will be a small reward for getting answers correct.
 - Blackjack Gameplay
 - This module is meant for the user to test their skills under pressure as the betting system will be in full effect. This module attempts to replicate playing a real game of blackjack except that a user can request hints and explanations if they forget the correct move during a game. However, the hints and explanation features will be turned off by default because the

main purpose of this module is to test the user's skills rather than help them practice.

- **Blackjack Game:** A group of modules that provide the most basic functions necessary to create a working blackjack game.
 - Cards and Deck Module
 - This module controls the card data and takes in input from the user and can return output as the dealer.
 - Betting Module
 - This module keeps track of a user's balance, net earnings, and win to loss ratio. Additionally, it keeps track of how much money should be rewarded or taken away from the user depending on the outcome of the blackjack game.
 - Visual Interface
 - This module provides a visual representation of the cards, deck, and user's balance while they play blackjack. It must interact with the Cards and Deck module in order to accurately represent the current state of the game.
- **Strategies and Methods:** A group of modules that can verify a user's accuracy to a specified blackjack strategy by receiving data from the Blackjack Game modules of the system.
 - Card Counting
 - This module keeps track of the card count for the specific type of card counting method that was selected by the user.
 - Betting Deviations
 - This module receives data from the Card Counting and Basic Strategy modules to measure the probability of winning a hand of blackjack. The higher the probability of winning, the more the user should bet.
 - Basic Strategy
 - This module performs a simple check whenever the user makes a move (i.e., hit, stand, double, or split) in order to determine if it correctly follows the Basic Strategy which is one of the most common blackjack strategies for beginners.

4. Software Modules

4.1. Strategy Practice

Role and Primary Function

The primary role of the Strategy Practice module is to provide a way for the user to practice a single blackjack strategy of their choosing without having to worry about other strategies or running out of money to bet with.

Interface Specification

This module will interact with the Cards and Deck as well as a Strategies module which would be Card Counting or Basic Strategy depending on the user's choosing.

Static and Dynamic Model

The Strategy Practice diagram is nearly identical to **Figure 2**; except, the Betting Deviations module would be absent since this module does not account for modules that interact with betting data.

Design Rationale

The reason behind designing this module is to provide the user a way to practice a single blackjack strategy without being distracted by other strategies or their active balance. It is necessary to separate this module from the Blackjack Gameplay module because the goal of this module is to help the user become better at a blackjack strategy rather than to simply test their skills.

Alternative Designs

The group considered designing this module to represent more of a minigame where the user would try to see how many correct moves they make in a row. However, the group members felt the design was too ambitious and suggested a design that offered a way for the user to practice blackjack strategies without having to completely redesign the Blackjack Gameplay module into a separate minigame.

4.2. Blackjack Gameplay

Role and Primary Function

The primary role of the Blackjack Gameplay module is to offer a way for users to test their skills in order to see how prepared they are if they were to go and play with real money with friends or at a casino.

Interface Specification

This module will interact with every module apart of the Blackjack Game and Strategies and Methods groups because it aims to give the most realistic simulation of a blackjack game.

Static and Dynamic Model

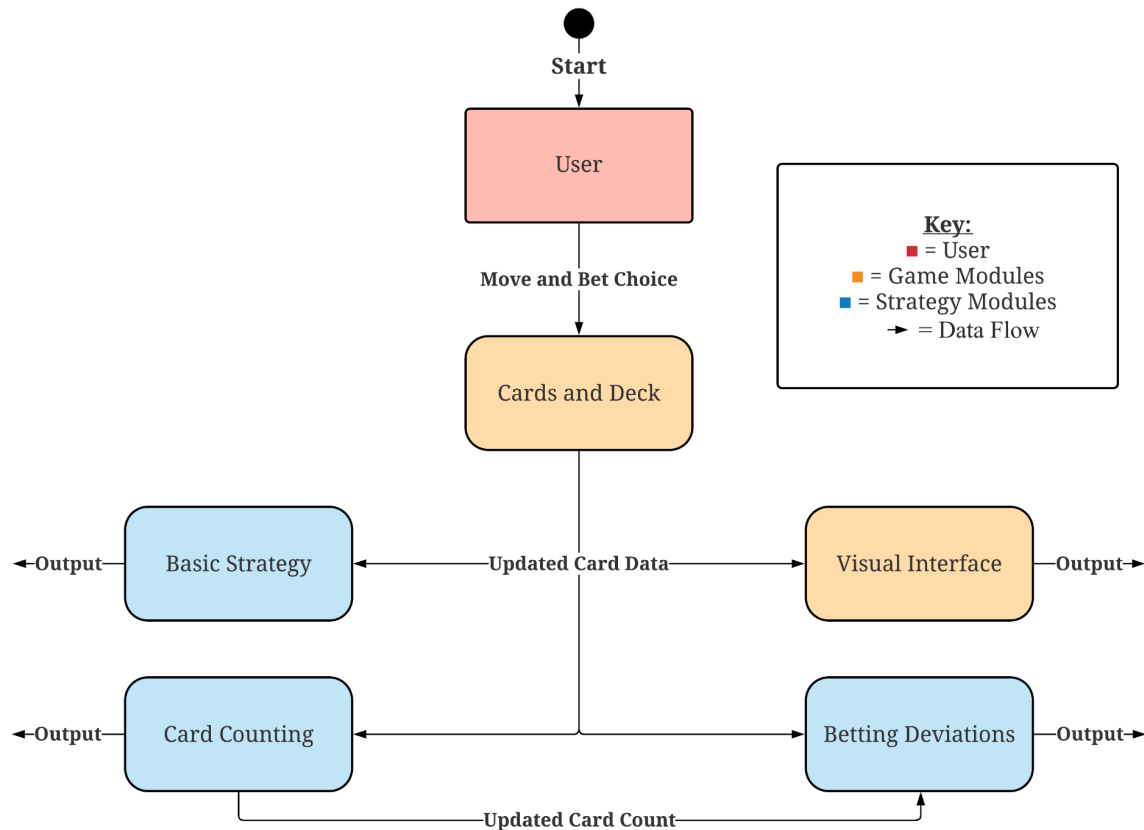


Figure 2: This figure shows the modules and their possible interactions while the Blackjack Gameplay module is in use.

Design Rationale

The reason behind this module is to offer a realistic simulation of blackjack games. This differs from the Strategy Practice module because the user must take in account for their balance, card count, and the Basic Strategy all at the same time. If they fail to do so, their score in each respective module will suffer and may even result in losing all their money.

Alternative Designs

The alternative designs of this module are not very different when it comes to function or purpose but rather in ways that they interact with the other modules of the systems. For instance, the group originally thought of having the Blackjack Game modules be apart of the Blackjack Gameplay module but quickly realized it would make it unnecessarily difficult to implement the Basic Strategy module's compare functions.

4.3. Cards and Deck

Role and Primary Function

The primary function of the Cards and Deck module is to provide the ability for the user to interact with the cards, so they can play a game of blackjack.

Interface Specification

This module will interact with every module in the system in some way or another because the Cards and Deck make up the blackjack game data. However, the cards and deck themselves will need to communicate to one another in order to be successfully called by the functions from other modules.

Static and Dynamic Model

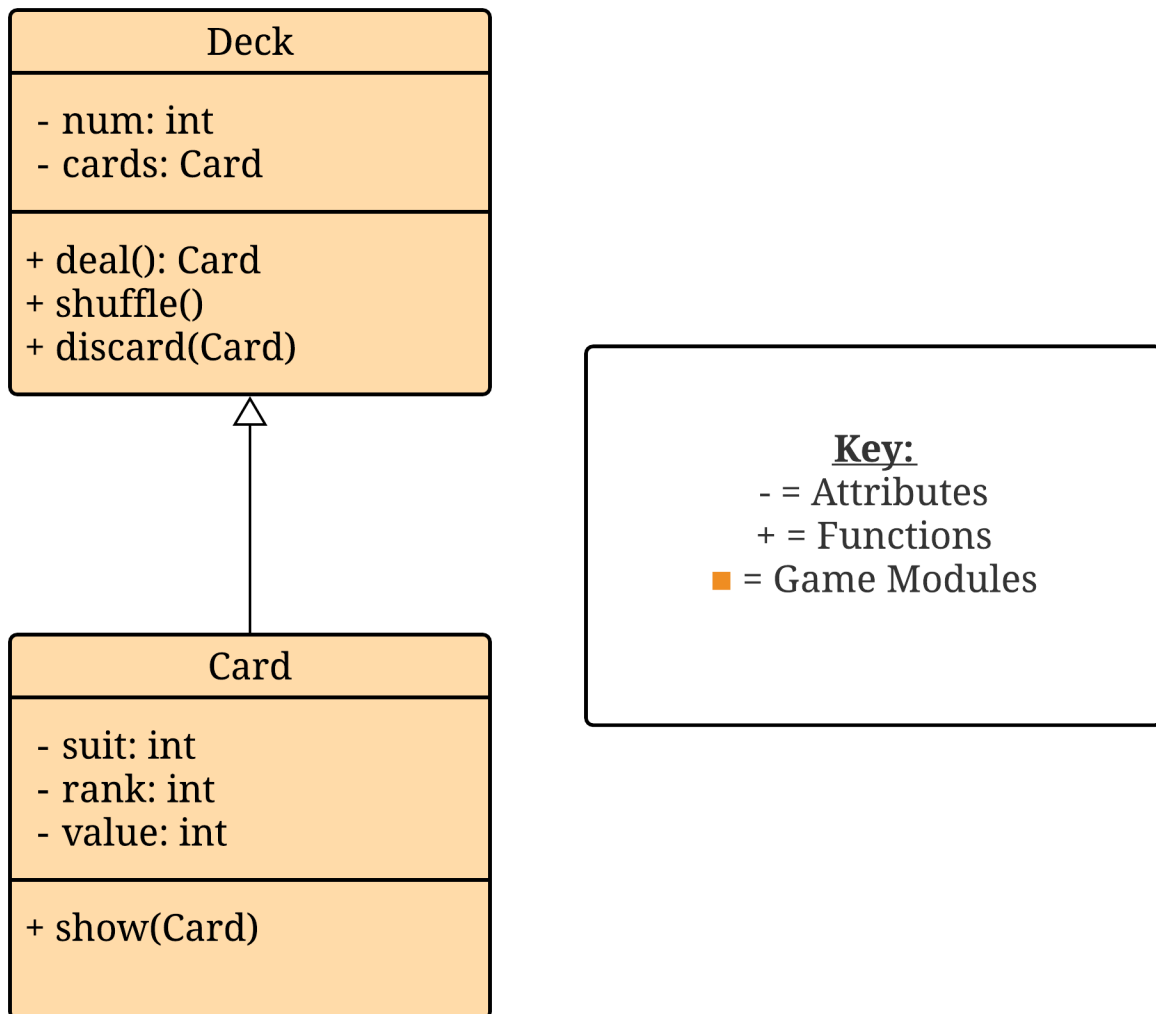


Figure 3: This figure shows a vertical UML class diagram for the relationship between the Card and Deck classes which makes up the Cards and Deck module (in addition, a key for color coordinated modules)

Design Rationale

The reason behind this module is to provide the baseline components, so a blackjack game can be played. These components represent the data of the system in the form of playing cards which provides the other modules with a form of information that can be computed.

Alternative Designs

In a previous design, the Cards and Deck module did not consist of a regular number of cards because most blackjack games in casinos are played using four to eight decks at a time. The design was changed to its current implementation because the group realized that this module should be as simple as possible and leave more detailed decisions up to the user or developers.

4.4. Betting Module

Role and Primary Function

The primary function of the Betting Module is to give a way for the user to track their progress and see how much money they have won or loss. It is also required by the Betting Deviations module as it needs to know a user's current balance in order to calculate the best amount to bet.

Interface Specification

This module will interact with the Betting Deviations and Visual Interface modules. The Betting Deviations will use the user's current balance (which is handle by the Betting Component) to determine the most optimal bet amount based on a fraction of the user's total balance and information sent from the Card Counting module. The Visual Interface requires the betting component because it needs to display the user's current balance, bet amount, and money won or loss during a turn.

Static and Dynamic Model

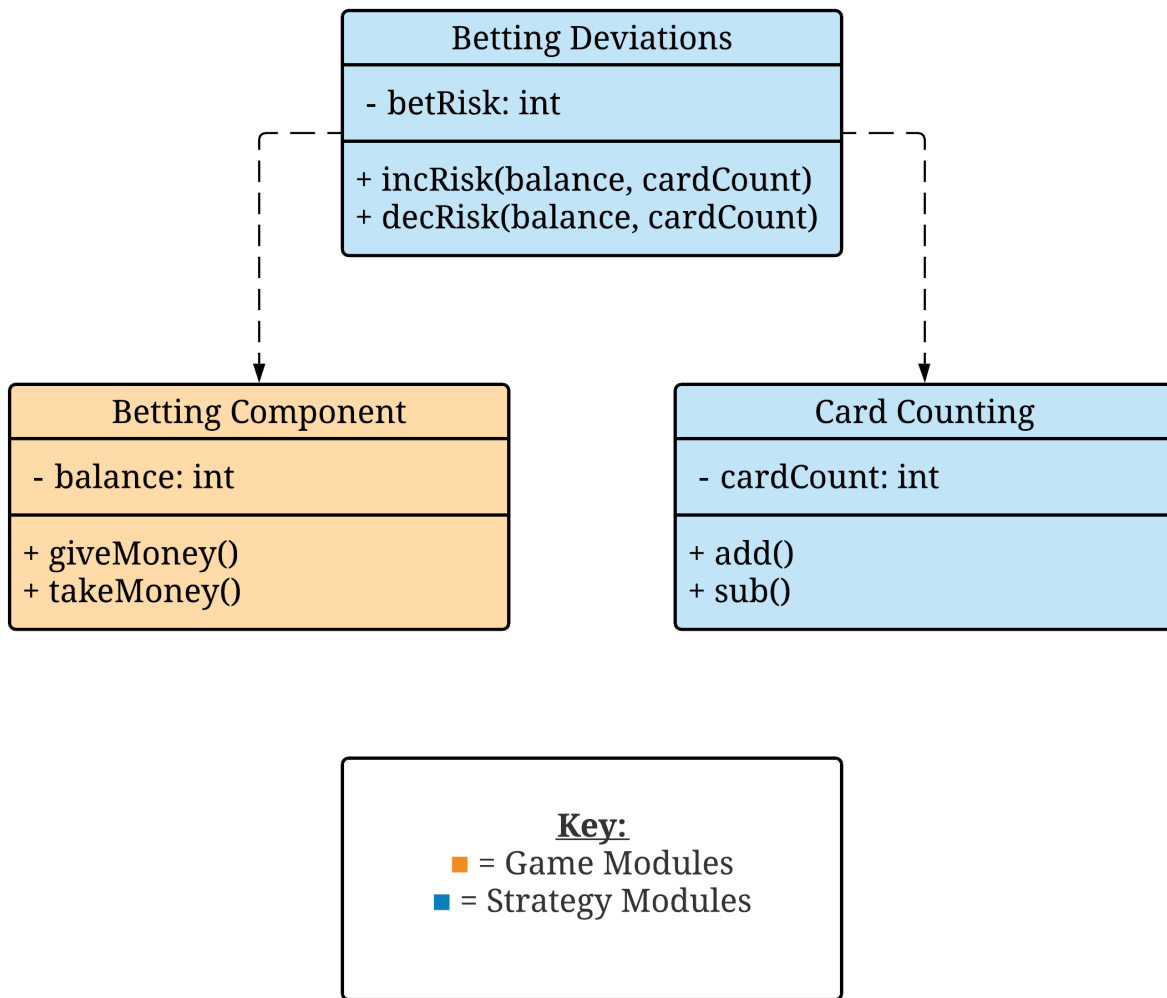


Figure 4: This figure shows how the Betting Component module is required by the Betting Deviations module via UML object diagram + a key for color coded modules.

Design Rationale

This module exists so that users have a way to keep track of their progress and see if they have made or lost more money than what they started with.

Alternative Designs

The group members of the proposed system are still currently debating how much money a user should start with. This is an important decision because it directly influences one of the three major Strategies and Methods modules provided by the system.

4.5. Visual Interface

Role and Primary Function

This module provides a more user-friendly way to interact and visualize what is currently happening in the user's blackjack game. It displays the user's balance, hand, and dealer's up card.

Interface Specification

This module interacts with the Cards and Deck and Betting Component modules. In receives data from both these modules so that it can visually represent it in a user-friendly way.

Static and Dynamic Model

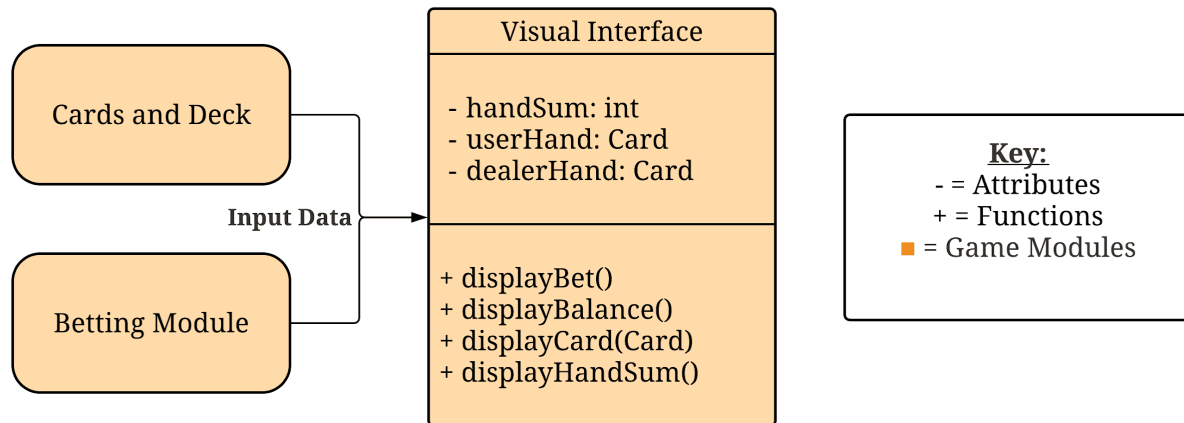


Figure 5: This figure shows the data required by the Visual Interface module in order to be displayed and where that data comes from as an adaptation of an UML Object Diagram.

Design Rationale

This module exists to provide a more user-friendly way for the user to interpret the data of the blackjack game. In this interface, a user will be able to see the cards, deck, and current balance without having to interact with a command line in order to access the information.

Alternative Designs

Currently, the group has decided to design the module to represent the blackjack data as a string of ASCII characters. Originally, members thought of using scans or images of real cards to represent the data but ended up choosing to prioritize the more urgent modules of the system.

4.6. Card Counting

Role and Primary Function

The primary role of this module is to keep track of the card count for a specific type of card counting method, so it can be compared to the user's card count.

Interface Specification

This module will interact with the blackjack game modules; specifically, the Cards and Deck because it needs to keep track of the card count based on the cards played so far.

Static and Dynamic Model

The simplest attributes of the Card Counting module can be seen in **Figure 4**. The number that the Card Counting class increments and decrement its card count by differs depending on the type of card counting method in use.

Design Rationale

This module exists so that the system has a way to compare the user's performance to that of the specific card counting method in order to determine the user's accuracy.

Alternative Designs

Since there are many types of card counting methods, the group originally thought to keep a live card count for every method but found that this was overkill and largely unnecessary because a user will only have one card count at a time. Furthermore, it would be extremely inefficient and not worth the resources even if it provided the user with more features.

4.7. Betting Deviations

Role and Primary Function

The primary role of this module is to determine the best amount of money to bet based on the user's balance and the previous cards that have been seen.

Interface Specification

This module interfaces with the Card Counting, Basic Strategy, and Betting Component modules. It uses data from each of these three modules to calculate the most optimal bet size based on the provided data.

Static and Dynamic Model

In **Figure 4**, it can be seen how the Betting Deviations module is dependent on the Betting Component and Card Counting modules.

Design Rationale

This module exists so the system can determine the most appropriate amount of money the user should bet in each blackjack game.

Alternative Designs

Originally, the group thought of having this module calculate an exact amount of money to bet based on the user's balance and card counting method but realized that might be too difficult to implement correctly for a wide variety of users. This is because a college kid might be willing to risk more money than an adult who is taking a more serious approach while they play blackjack. This led the group to decide to change the module where it would only suggest to the user the most optimal times to bet either a large, normal, or small amount of money. By doing so, it left the user to be able to interpret for themselves what they felt a large, normal, and small bet looked like.

4.8. Basic Strategy

Role and Primary Function

The primary role of this module is to compare the user's move to the move recommended by the Basic Strategy.

Interface Specification

This module interacts with betting deviations and simulation modules. It provides the betting deviations module to compare the value of the hand to the basic strategy. The basic strategy module also informs the simulation modules of the correct move if the user requested assistance for the given blackjack game/turn.

Static and Dynamic Model

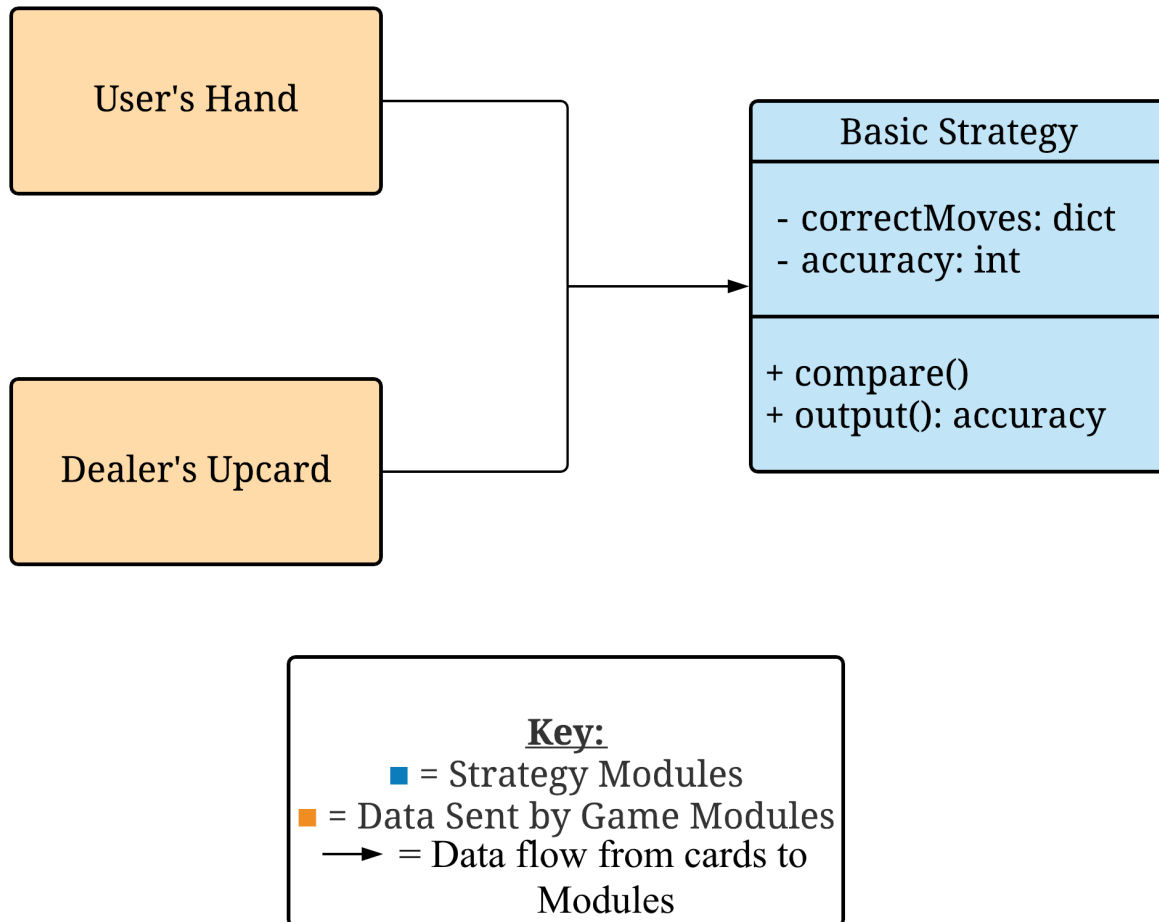


Figure 6: This figure shows the two card data inputs that are sent from the Blackjack Game modules to the Basic Strategy module as an adaptation of an UML Object diagram.

Design Rationale

This module exists so that the system can compare the user's accuracy of the Basic Strategy method.

Alternative Designs

Since the Basic Strategy is a predefined strategy that does not require any information from previous blackjack games played by the user, the goal of this module should be nearly

identical to alternative designs. However, there may be more efficient ways to compare and store playing card data than with a python dictionary. The group considered creating a python class that could use magic methods to compare card types instead of the dictionary that is currently implemented.

5. Dynamic Models of Operational Scenarios (Use Cases)

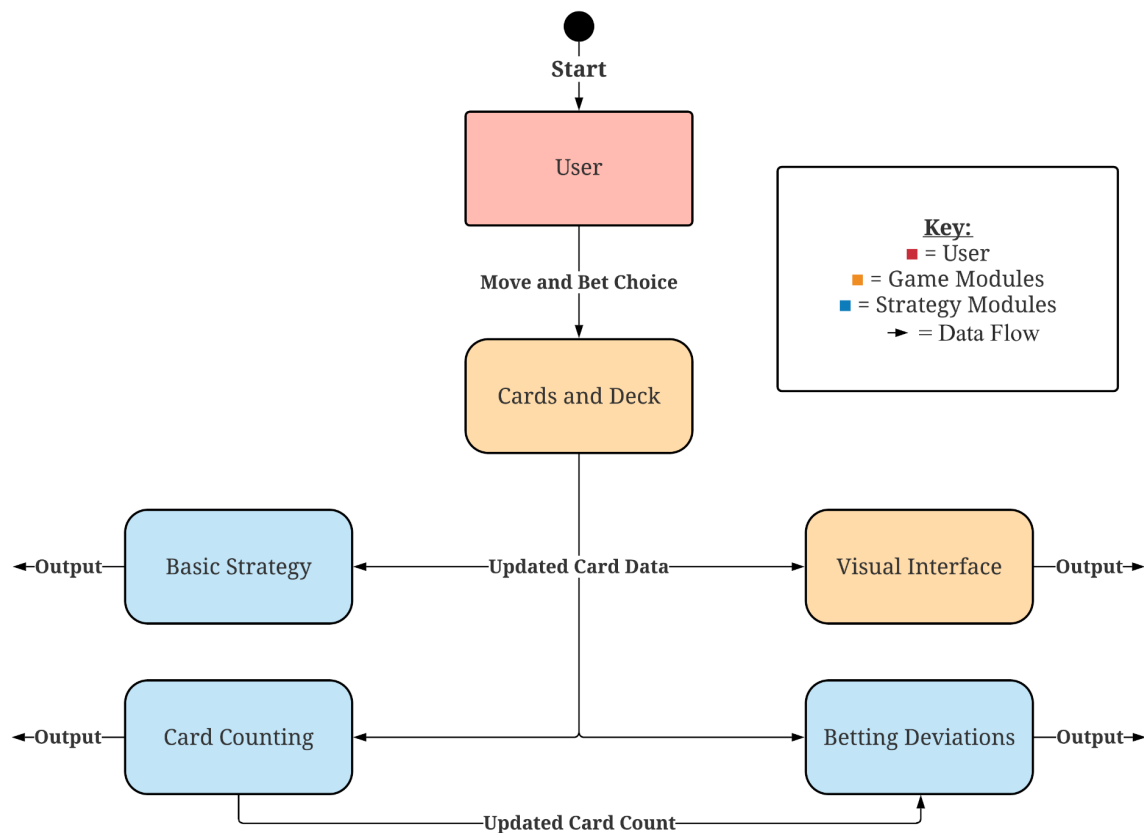


Figure 7: A UML Activity diagram that shows the possible interactions throughout the system while a user utilizes the Blackjack Gameplay module.

6. References

Fakhroutdinov, K. (n.d.). The Unified Modeling Language. UML Diagrams - overview, reference, and examples. <https://www.uml-diagrams.org/>.

Wattenberger, N. (n.d.). Card Counting. Card Counting - Strategies.
<https://www.qfit.com/card-counting.htm>.

7. Acknowledgements

This format builds heavily on Anthony Hornof's SDS template.