}

.

# Getting Ready: Online Blackjack Game

Understand the online Blackjack game problem and learn the questions to further simplify this problem.

**We'll cover the following**

* [Problem definition](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Problem-definition)
* [Expectations from the interviewee](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Expectations-from-the-interviewee)
  + [Players in the game](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Players-in-the-game)
  + [Point dynamics](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Point-dynamics)
  + [Card limit](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Card-limit)
* [Design approach](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Design-approach)
* [Design patterns](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Design-patterns)

## Problem definition

**Blackjack** is one of the most played games in casinos. In this game, several players play against the dealer. The objective of the game is to get closer to 21 than the dealer without exceeding the 21 points. This game is played with a deck of cards. Each card has a specific value associated with it, and these values are compared. The value of an ace card can be 1 or 11 points while cards of 10, Jack, Queen, and King value 10 points. Whereas, cards 2-9 have the same values as the number written on them. Hence, the player having one ace and a face card (King, Queen, or Jack card.) or a 10 card has 21 points. This is called Blackjack.

At the start of the game, a bet is placed by a player, and then the dealer will create hands (Each player and the dealer get two cards each.). All players have both cards exposed while the dealer has one card exposed and one card hidden. Now, both the dealer and the player can hit (Draw an additional card.) a card and ensure that they should not get over 21. Anyone exceeding 21 points, busts and loses the bet. The player can stand pat (Stop taking cards) at any time. When a player stops taking cards, if the player has more points than the dealer but less than 22, they win the bet.

The following slide represents how to play Blackjack:

Blackjack table setup

**1** of 20

## Expectations from the interviewee

There are several components in a Blackjack game, each with specific constraints and requirements placed on them. The following provides an overview of some of the main expectations that the interviewer will want to hear you discuss in more detail during the interview.

### Players in the game

To get a better understanding of the game and the players of the game, you may ask the interviewer the following set of questions:

1. How many players can play Blackjack?
2. Can players play the game against each other?

### Point dynamics

The rules related to the number of points also need to be clarified by the interviewer, therefore, you may ask questions listed below:

1. Upto how many points can the player or the dealer hit the card?
2. What will happen if the dealer and the player both get the same points?

### Card limit

The interviewer would expect you to ask a question related to the maximum number of cards the player can have. Therefore, you may ask a question listed below:

1. Is there a limit on the number of cards the player take?

## Design approach

We are going to design this Blackjack game system using the bottom-up design approach. For this purpose, we will follow the steps below:

* Identify and design the smallest components first–the card, and player.
* Use these small components to design bigger components, for example, the deck and controller.
* Repeat the steps above until we design the whole Blackjack game.

## Design patterns

It is always a good practice to discuss the design patterns that the online Blackjack game falls under, during the interview. Stating the design patterns will give the interviewer a positive impression and shows that the interviewee is well-versed in the advanced concepts of object-oriented design.

The following design patterns are used to design the online Blackjack game:

* Iterator design pattern
* State design pattern

Let's explore the requirements of the online Blackjack game in the next lesson.

Back

**Requirements for the Online Blackjack Game**

Learn about all requirements of the online Blackjack game.

**We'll cover the following**

* [Requirement collection](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Requirement-collection)

In this lesson, we’ll list the requirements of the online Blackjack game. This is a very crucial step since requirements define the scope of a problem, so getting them right from the interviewer and understanding them well will make the design of the rest of the system smooth and easy.

We’ll use the notational convention to identify each requirement with a unique label "Rn", where "R" is short for Requirement and "n" is a natural number.

**Requirement collection**

The requirements for the Blackjack design problem are defined below:

**R1:** The Blackjack game contains the shoe of cards which contains one or more decks of cards in it.

**R2:** The deck will consist of 52 cards in four suits, where each suit contains 13 cards: The ace, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, and King.

**R3:** Every card has points associated with it. The criteria to calculate the face value of the card is as follows:

|  |  |
| --- | --- |
| **Card** | **Face value** |
| Ace | 1 or 11 |
| From 2 to 10 | Equals the card number |
| Face cards (King, Queen, and Jack) | 10 |

**R4:** There can be two types of users that can play the Blackjack game, i.e., the dealer and the player.

**R5:** The player places a bet at the start of the game.

**R6:** The dealer will deal two cards to themselves and two to each player at the start of the game.

**R7:** The player will have both cards exposed, while the dealer has one card facing up and the other card down.

**R8:** The player can hit (Draw an additional card.)  if their hand has less than 21 points.

**R9:** The dealer can hit if their hand is less than 17.

**R10:** If a player or the dealer’s hand is more than 21, they bust and lose the game.

**R11:** The player can decide to not get a further card by standing pat.

**R12:** At the end, if the points of the player are more than that of the dealer but less than 21, then they win the game and get 100 percent profit.

**R13:** If a player gets 21 points by winning an ace and a face card or 10, then they are called a Blackjack and get 150 percent profit.

**R14:** If the player and the dealer have the same points at the end of the game, the player can take their bet money back or can replay the game.

**R15:** If the player left a game in the middle of the game, the dealer will win the game.

We've identified our requirements for the problem, and in the next lesson, we will define different use cases for the online Blackjack game design.

# Use Case Diagram for the Online Blackjack Game

Learn how to define use cases and create the corresponding use case diagram for the Blackjack game.

**We'll cover the following**

* [System](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#System)
* [Actors](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Actors)
  + [Primary actors](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Primary-actors)
  + [Secondary actors](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Secondary-actors)
* [Use cases](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Use-cases)
  + [Player](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Player)
  + [Dealer](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Dealer)
* [Relationships](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Relationships)
  + [Associations](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Associations)
  + [Include](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Include)
  + [Extend](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Extend)
* [Use case diagram](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Use-case-diagram)

Let's build the use case diagram of the Blackjack game and understand the relationship between its different components.

First, we’ll define the different elements of our Blackjack game, followed by the complete use case diagram of the system.

## System

Our system is a "Blackjack game."

## Actors

Now, we’ll define the main actors of our Blackjack game.

### Primary actors

* **Player:**This actor is the main player of the game. This actor plays the game, which includes placing a bet, hit or stand, or quitting the game. This can also create, edit, or update its account.

### Secondary actors

* **Dealer:** This is the secondary actor of the game, who manages the Blackjack table, dealing cards, and completing the payout of earnings. This actor can also manage the accounts and memberships of members.

## Use cases

In this section, we will define the use cases for the Blackjack game. We have listed the use cases according to their respective interactions with a particular actor.

**Note:**You will see some use cases occurring multiple times because they are shared among different actors in the system.

### Player

* **Join a game:**To join a new Blackjack game
* **Place bet:** To place a bet for the game
* **View open games:** To view open games that are waiting for the player to start
* **Resigns a game:** To leave a game in the middle of the game
* **Hit:** To request an extra card from the dealer
* **Stand:** To hold your cards and skip your turn
* **Create account:**To create a new account in the Blackjack game
* **Update account:**To update account information or password in the Blackjack game
* **Reset password:**To update the account password in the Blackjack game
* **Cancel membership:** To cancel membership in the Blackjack game
* **Login/Logout:** To log in and out of the Blackjack game

### Dealer

* **Create a new game:**To start a new Blackjack game
* **View open games:** To view open games that are waiting for the player to start
* **Create hands:** To give two cards to the player and two cards to the dealer
* **Draw card:**To add cards from the deck to the player and dealer's hand
* **Collect or payout:** To give or take money after the game ends, depending upon the game's result
* **Block member:** To block any member to ensure they can't play the game
* **Create account:**To create a new account in the Blackjack game
* **Update account:**To update account information or password in the Blackjack game
* **Reset password:**To update the account password in the Blackjack game
* **Cancel membership:** To cancel membership in the Blackjack game
* **Login/Logout:** To log in and out of the Blackjack game

## Relationships

This section describes the relationships between and among actors and their use cases.

### Associations

The below table shows the association relationship between actors and their use cases.

|  |  |
| --- | --- |
| **Player** | **Dealer** |
| Join a game | Create a new game |
| Place bet | View open games |
| View open games | Create hands |
| Resigns a game | Draw card |
| Hit | Collect or payout |
| Stand | Block member |
| Create account | Create account |
| Update account | Update account |
| Reset password | Reset password |
| Cancel membership | Cancel membership |
| Login/Logout | Login/Logout |

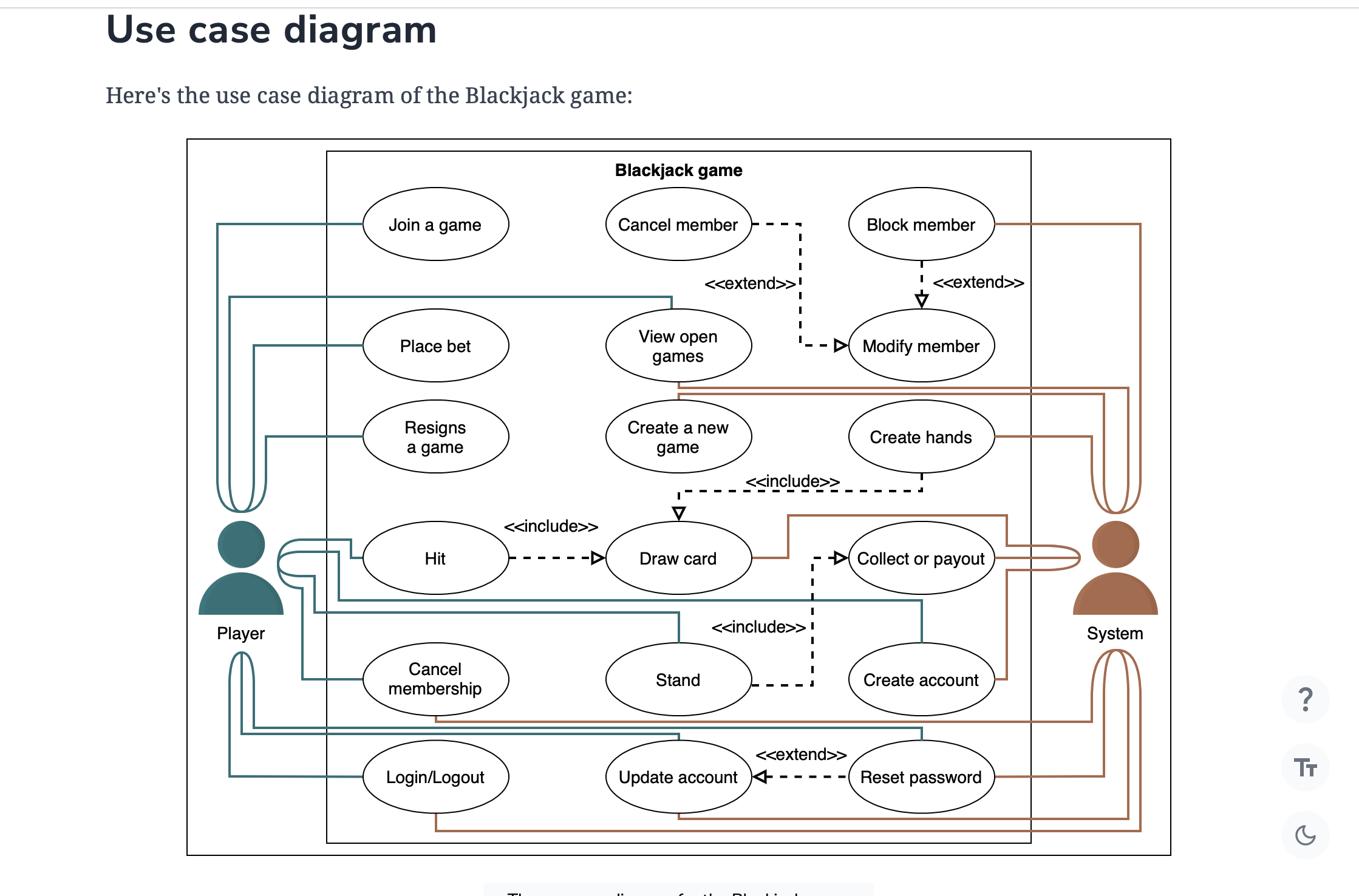
### Include

* The "Create hand"use case and the "Hit" use case has an include relationship with the "Draw card" use case because, at the start of the game, we draw cards to create hands, and in the middle, we draw cards when the player chooses to "Hit."
* Whenever a player chooses to "Stand," their turn is skipped. Then, if the dealer card total is more than 16, the dealer and player's card total is compared with each other. According to the result of the game, the dealer either collects or pays the money. So the "Stand" use case has an include relationship with the "Collect or payout" use case.

### Extend

* The "Cancel membership"use case and "Block member" use case have an extend relationship with the "Modify member" use case, because whenever we cancel someone's membership or block someone, we modify the member status.
* Whenever the account password is changed, the account is updated. Hence, the "Reset password" use case has an extend relationship with the "Update account" use case.

## Use case diagram



The use case diagram for the Blackjack game

In the next lesson, we will discuss the class diagram with a detailed explanation of all classes and their relationship with each other.

# Class Diagram for the Online Blackjack Game

Learn to create a class diagram for the Blackjack game using the bottom-up approach.

**We'll cover the following**

* [Components of Blackjack](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Components-of-Blackjack)
  + [Card](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Card)
  + [Deck](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Deck)
  + [Shoe](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Shoe)
  + [Hand](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Hand)
  + [Players](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Players)
  + [Blackjack controller](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Blackjack-controller)
  + [Blackjack game view](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Blackjack-game-view)
  + [Blackjack game](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Blackjack-game)
  + [Enumerations and custom data types](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Enumerations-and-custom-data-types)
* [Relationship between the classes](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Relationship-between-the-classes)
  + [Association](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Association)
  + [Aggregation](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Aggregation)
  + [Composition](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Composition)
  + [Inheritance](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Inheritance)
* [Class diagram for the Blackjack game](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Class-diagram-for-the-Blackjack-game)
* [Design pattern](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Design-pattern)

We’ll create the class diagram for the Blackjack game. In the class diagram, we will first design the classes, and then identify the relationship between classes according to the requirements for the Blackjack game problem.

## Components of Blackjack

In this section, we’ll define the classes for the Blackjack game. As mentioned earlier, we are following the bottom-up approach to designing a class diagram for the Blackjack game.

### Card

Card belongs to a suit and has a face value. The face value of the card is according to the card number. For example, if we have a number card 5, its face value is also 5. The face value for the King, Queen, and Jack is 10, and 1 or 11 for the Ace, depending on the situation.

A screenshot of a computer

Description automatically generated

### A screenshot of a computer Description automatically generatedPlayers

Player is an abstract class. There are two types of players: BlackjackPlayer and Dealer. These classes can be derived from the Player class.

BlackjackPlayer: They place the first wager and update the stake with winning and losing sums. They can choose between hit and stand options.

Dealer: They are primarily in charge of dealing cards and following the Blackjack rules.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A diagram of a blackjack

Description automatically generated

A screenshot of a computer

Description automatically generated

A diagram of a game

Description automatically generated

## Design pattern

The Iterator design pattern can be applied, since cards are assigned to the players from the deck of cards by just iterating through the list of cards.

We can also use the State design pattern for our online Blackjack game because this game has a finite number of states. Some of these states are as follows:

* Shuffle the deck
* Draw a card and give it to the dealer
* Draw a card and give it to the player
* Deal cards
* Player hit
* Player stand

We have completed the class diagram of the Blackjack game according to the requirements. Now, let's design the activity diagram of the Blackjack game in the next lesson.

# Activity Diagram for the Online Blackjack Game

Create an activity diagram for the Blackjack game.

**We'll cover the following**

* [Blackjack gameplay](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Blackjack-gameplay)
  + [States](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#States)
  + [Actions](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Actions)

Activity diagrams are a great way to visualize the flow of messages from one activity to the other in the system. There can be different activity diagrams that we can create for our Blackjack game. In this lesson, we will create an activity diagram for the steps to play Blackjack.

## Blackjack gameplay

The following are the states and actions that will be involved in this activity diagram.

### States

**Initial state:**The player places the bets and the dealer deals the cards among themselves and the player.

**Final state:**There are four final states present in this activity diagram, shown below:

* The player wins with 3:2 of the bet.
* The player wins with an equal bet.
* The match is tied.
* The player loses.

### Actions

* The player places the bet.
* The dealer deals two cards among themselves and two for the player.
* If the total of the player's card is 21:
  + It is a Blackjack, and the player will win and get 150% of the bet.
* If the total is less than 21:
  + The player will lose, and the dealer will collect the bet.
* If the total of the player card is less than 21:
  + The player will have two options to either "Stand" or "Hit."
* If a player decides to "Hit":
  + The player gets another card from the dealer, and then we will check the player's card total again.
* If the player decides to "Stand":
  + The dealer's card total will be checked.
* If it is less than 17:
  + The dealer will get another card until its card total is more than 16.
* When the total is more than 16, we will compare the totals of the player and dealer’s cards.
* Whoever gets more wins, and if both cards' total is the same, it will be a tie.

Based on the order above, the activity diagram of a Blackjack game is given below:

A diagram of a game

Description automatically generated

We've looked at the activity diagrams of the Blackjack game. In the next lesson, we will present the code for our designed classes in some of the most popular languages.

Back

# Code for the Online Blackjack Game

Write the object-oriented code to implement the design of the Blackjack game problem.

**We'll cover the following**

* [Blackjack game classes](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Blackjack-game-classes)
  + [Enumerations and custom data type](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Enumerations-and-custom-data-type)
  + [Card](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Card)
  + [Deck and shoe](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Deck-and-shoe)
  + [Hand](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Hand)
  + [Players](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Players)
  + [Blackjack controller and game view](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Blackjack-controller-and-game-view)
  + [Blackjack game](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Blackjack-game)
* [Wrapping up](https://www.educative.io/order-confirmation/stripe/subscription-buy?payment_intent=pi_3O1FFeKhXp6R50hI1xakUX3g&payment_intent_client_secret=pi_3O1FFeKhXp6R50hI1xakUX3g_secret_EDla1wT2fgsm34I6up6bRJM7s&transaction_id=1d4a8dd5-516d-4906-aa95-9e9effdf447d#Wrapping-up)

We've covered different aspects of the Blackjack game and observed the attributes attached to the problem using various UML diagrams. Let us now explore the more practical side of things where we will work on implementing the Blackjack game using multiple languages. This is usually the last step in an object-oriented design interview process.

We have chosen the following languages to write the skeleton code of the different classes present in the Blackjack game:

* Java
* C#
* Python
* C++
* JavaScript

## Blackjack game classes

In this section, we will provide the skeleton code of the classes designed in the class diagram lesson.

**Note:** For simplicity, we are not defining getter and setter functions. The reader can assume that all class attributes are private and accessed through their respective public getter methods and modified only through their public method functions.

### Enumerations and custom data type

The following code provides the definition of the enumeration and custom data type used in the Blackjack game.

Suit: We need to create an enumeration to keep track of the suit of the card, whether it is diamond, spade, heart, or club.

AccountStatus: We need to create an enumeration to keep track of the status of the account, whether it is active, canceled, closed, blocked, or none.

The Person class is used as a custom data type. The implementation of the Person class can be found below:

**Note:** JavaScript does not support enumerations, so we will be using the Object.freeze() method as an alternative that freezes an object and prevents further modifications.

Ja

// Enumeration

enum Suit {

HEART,

SPADE,

CLUB,

DIAMOND

}

enum AccountStatus {

ACTIVE,

CLOSED,

CANCELED,

BLACKLISTED,

NONE

}

// Custom Person data type class

public class Person {

private String name;

private String streetAddress;

private String city;

private String state;

private int zipCode;

private String country;

}

### Card

This class contains the playing cards or cards used in the Blackjack game.

Java

C#

Python

C++

JavaScript

public class Card {

private Suit suit;

private int faceValue;

public Card(Suit suit, int faceValue);

}

### Deck and shoe

Shoe is a device to hold multiple Deck and a Deck has 52 cards of four suits. One suit contains nine number cards (2–10) and four face cards (King, Queen, Jack, and Ace).

public class Deck {

private List<Card> cards;

public Deck();

public List<Card> getCards();

}

public class Shoe {

private List<Deck> decks;

private int numberOfDecks;

public Shoe(int numberOfDecks, List<Deck> decks) {

// 1. createShoe();

// 2. shuffle();

}

public void createShoe();

public void shuffle();

public Card dealCard();

}

### Hand

The Hand class represents a Blackjack hand used in this game and contains multiple cards.

public class Hand {

private List<Card> cards;

public Hand(Card card1, Card card2);

public int getScores();

public void addCard(Card card);

}

### Players

The Player is an abstract class and the BlackjackPlayer and Dealer classes extend the Player class.

* BlackjackPlayer: They place the first wager and update the stake with winning and losing sums. They can choose between the hit and stand options.
* Dealer: They are primarily in charge of dealing cards and following the Blackjack rules.

public abstract class Player {

private String id;

private String password;

private double balance;

private AccountStatus status;

private Person person;

private Hand hand;

public void addHand(Hand hand);

public void removeHand(Hand hand);

public abstract boolean resetPassword();

public void addToHand(Hand hand);

}

public class BlackjackPlayer extends Player {

private int bet;

private int totalCash;

public BlackjackPlayer(Hand hand);

public void placeBet(int bet);

public boolean resetPassword(){

// definition

}

}

public class Dealer extends Player {

private Hand hand;

public int getTotalScore();

public boolean resetPassword(){

// definition

}

}

### Blackjack controller and game view

The BlackjackController class validates the actions (hit or stand) and responds accordingly. The BlackjackGameView class represents the game view.

public class BlackjackController {

public boolean validateAction();

}

public class BlackjackGameView {

public void playAction(String action, Hand hand);

}

### Blackjack game

The BlackjackGame class represents how we can play this game or its basic sequence of play.

public class BlackjackGame {

private Player player;

private Dealer dealer;

private Shoe shoe;

private final int MAX\_NUM\_OF\_DECKS = 4;

public BlackjackGame(BlackjackPlayer player, Dealer dealer);

public void playAction(String action, Hand hand);

public void hit(Hand hand);

public void stand();

public void start();

}

**Wrapping up**

We've explored the complete design of the Blackjack game in this chapter. We've looked at how a basic Blacljack game can be visualized using various UML diagrams and designed using object-oriented principles and design patterns.