

Project #2. Fall 2021.

## #SPECIFICATIONS

For this project, students will also be allowed to work in pairs. This will provide them with an opportunity to collaborate, exchange ideas and check each other's work.

The project will consist in designing a program that can simulate a game of blackjack as described in the Wikipedia article on Blackjack at [Blackjack - Wikipedia](#) or [Blackjack - Simple English Wikipedia, the free encyclopedia](#) (this last one has an example of a game).

We will not simulate a strict version of blackjack. The version students will have to design will be a simplified version where the players can only "HIT" or "STAND". There will be a textbox to pick the seed that the RandomVariable's static random number generator will produce. Choosing different seeds (done through a reset button as in project #1) will allow the changing of the sequence of cards dealt as well as resetting the shuffling machine (reloading it with the decks)

A rough outline of the game is this:

The source of cards (The shuffling machine) can contain anywhere from 1 to 8 decks (a number selected from a combo-box labeled "# of Decks")

All cards drawn have to be shown on the screen (there are no face-down cards)

The payout is 3:2 meaning that if you bet \$10 and win, you get \$15. If you lose, you lose your bet of \$10

Players will start with a pot of \$X specified in a textbox labeled "Available \$" with a default of \$100.

The following parts 1) and 2) represent the "Initial Deal" depicted in the Wikipedia article (see the associated image)

1) The dealer (in this case the application) draws 2 cards from the shuffling machine to each "box" (player). In this project there is only 1 player (but there could be more in the last project).

2) The dealer then draws their first card.

3) The dealer then deals 1 card to each player still wishing to "HIT" (get a card) rather than "STAND"

4) Based on the cards they have, each player determines whether they want another card (HIT) or stop receiving cards (STAND). If the player's cards add up to more than 21, they have BUSTED and have lost their bet.

Players have a choice whether or not to get extra cards. Stand means to get no more cards and hit means to ask for another card. To stand, wave your hand back and forth.

To ask for another card, wave your hand toward you. You can ask for any number of cards unless you bust.

After all the boxes have finished betting (busted or a standing) go to the next step or else go back to step 3

5) After all the boxes have finished betting (busted or a standing), the dealer's hand is resolved by drawing cards until the hand achieves a total of 17 or higher (a dealer total of 17 including an ace valued as 11, also known as a "soft 17", must be drawn to in some games and must stand in others). The dealer never doubles, splits, or surrenders. If the dealer busts, all remaining player hands win. If the game mode selection is "H17" (for Hit 17), the dealer has to HIT (draw) on 17. If the mode is "S17" (for Soft 17) the dealer has to STAND on 17

6) Resolve if the player has won or lost that game and adjust their money amount. Number cards count as their number, the jack, queen, and king ("face cards" or "pictures") count as 10, and aces count as either 1 or 11 according to the player's choice. If the total exceeds 21 points, it busts, and all bets on it immediately lose.

Hands below the dealer's hand lose unless the dealer goes over 21 (busts). Hands equal to the dealer's keep their chips. If both the player and dealer bust, the dealer wins.

If the dealer does not bust, each remaining bet wins if its hand is higher than the dealer's and loses if it is lower.

A player total of 21 on the first two cards is a "natural" or "blackjack," and the player wins immediately unless the dealer also has one, in which case the hand ties. In the case of a tie ("push" or "standoff"), bets are returned without adjustment. But a player blackjack beats any dealer hand that is not a blackjack, even one with a value of 21 (see [What Is Natural Blackjack? – Remedies For All \(wordpress.com\)](https://www.wordpress.com/what-is-natural-blackjack-remedies-for-all/))

This project will have students build on their original project 1 by creating a shuffling machine or a shoe (class aShoe) containing potentially up to 8 decks of cards. The deck of card class is to be named aDeckOfCards and consists of 52 cards. The card class is to be named aCard and should contain its value, face and picture.

The shoe should must implement the IDrawCard interface which is defined as:

```
Interface IDrawCard
{
    aCard Draw();
}
```

You can check information on interfaces at [interface - C# Reference | Microsoft Docs](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/interfaces/)

**Every new class or interface must have its own file.** To do so, use the "Add → "New Item" menu option. This will add the new item in its own file with an appropriate name.

You will need to obtain images of the 52 cards present in a deck.

Your form should look something like this (but does not have to be exactly like it)

Fig. 1)

0517

Dealer Cards

Value

Some value or "BUST"

Total \$

100

Bet (\$)

5

Hit

Stand

Win/Lose/Tie

Win Lose or Tie

Reset

Play

Seed: 999

You should be able to play multiple games (multiple tables) at the same time. This means that this application will use multiple forms.

We also will want to be able to play multiple games on the same form (simulating having multiple players on the same table), but this will be for project 3. This specification is to give you a chance to start thinking about how you would satisfy that requirement. For this look-ahead of project 3, you will want to start thinking about how you will implement the "Basic Strategy" displayed at the bottom of [Blackjack - Wikipedia](https://en.wikipedia.org/wiki/Blackjack). This will be the strategy your simulated players will use while you use the "STAND" and "HIT" buttons. And, of course, you can already start asking in class, of me or your classmates, how to do so.

If you feel that you need more information about the project, please save your questions for when in class. We can then clarify things.

## MODIFICATIONS:

As mentioned in class on Monday, Nov. 1, you want to have all your different classes/interfaces in different files. We went over how to do that by adding either a new

item (for a new class) or an existing item from another project (for the reuse of an already designed class). Having multiple classes in a single file will cost you 25 points!

We also now want to have a simpler form as our starting form. The form should look something like this:

Fig. 2)

The image shows a hand-drawn form on a Microsoft Whiteboard grid. The form is a large rectangle containing three input fields and a button. The first field is labeled 'Decks' with the value '2'. The second field is labeled 'Seed' with the value '555'. The third field is a checkbox labeled 'S17' which is checked. A 'New Player' button is at the bottom. Handwritten labels with arrows point to the input fields: 'Text boxes' points to the 'Decks' and 'Seed' fields, and 'Check box' points to the 'S17' checkbox. The 'New Player' button is labeled 'Button'.

This new starting form will allow you to specify the number of decks in the shoe, the seed, the mode of play (Soft 17 or Hard 17). The “New Player ” button should then launch a new form which should look like the original one (shown in Fig. 1).

We showed in class how this can be done fairly easily.

Not following the new specification will cost you:

50 points If you do not have your starting form be something like Fig. 2) (as shown in class)

25 points If your code does not use a separate file for each class (as done in class)

30 points If your code doesn't allow multiple players to be launched.

The new specifications do bring into focus the role of the random number generator.

You will want to think about your implementation.

DELIVERABLE

Your deliverable must be your cleaned, zipped, solution folder which will contain project 1 and project 2. You will need to provide both projects in the solution so that when compiled, project 2 will have available to it all the classes or interfaces or resources it needs from project 1 (as discussed in class).

If you had to create multiple projects due to the modifications in the project, simply make the working project (the one that implements what is the 2<sup>nd</sup> project) the startup project. In this way, when the TA runs the solution (made up of multiple projects), it will run the appropriate one.