# Project Design Document: [Insert the name of your game]

(see Project Phase 2 instructions for more details about this template

The purpose of this document is to outline how you will go about designing your project

## Description of Game / What Playing it will be like

Write a short description (a few sentences) of how the gameplay will work. This should include details about how the user will play. For example, how many players? What will a sequence of the game appear like to the user? What will happen if the user enters invalid values? What happens when a game finishes? You may find it useful to add sample screenshots or sketches (drawn on paper for example or sample text typed into a text editor) as to how the game play will work.

* The game of Blackjack is played with 2 to a maximum of 7 and a dealer (Where dealer is dealing cards and playing at the same time)
* Gameplay loop
  1. Dealer Handouts 2 cards to each player and themselves
  2. Bet amount money
  3. Then Flip Card and add number up together (Ace being either 1 or 11, All face cards are all 10, All other cards are its pip value) Trying to get less than or equal to 21
  4. Then you get to decide to hit or stand (hitting means you get dealt another card and stand you keep your cards and don’t get dealt another card) Repeat until Stand
  5. Then Dealer hits until they get a number higher then 17
  6. Higher number wins between you and dealer wins the bet and collects the reward
  7. And just repeat for as long as you want to play
* When you get an invalid value which will be an int for the bet and a string playing again and to stand or hit. It will ask again
* When game finished you will be asked to play or finish game if you play again then gameplay loop will be repeated until you have no more betting money. If you say finish, then it will end game

If there are specific rule changes you intend to make, you can also include them here.

## UML Class Diagram

[Here you should insert a class diagram of how you anticipate coding your game. This is not a final decision but is meant to be a starting point!]

Diagram

Description automatically generated

**Remember that there are several classes listed in the Project Phase two document that are required**

## List of classes: descriptive

For each class in the class diagram, you should write a sentence or two outlining the purpose

**Card Class:**

getNumber: to get the Number when needed

getSuit: to get the suit when needed

String toString: to print out card

**DynamicCardArray:**

toString: to print out card array

add: to add card to hands  
remove: to remove a card from hand

cardRemoveFromTop: to remove card from top of deck so that the card isn’t there anymore

length: to get the length of the array

contains: checks if it contains a certain card

get: Validation to check and return the value at the index given

set: sets a number in the array to another value

shuffle: shuffles the deck in a random position in a new array

sum: adds the card values of all the cards in dealer and player hand and gives back number. And if ace is a one or an eleven  
**Deck:**

Deck: Creates an array of 52 cards

**Player:**

PlayerHand: creates an array of size 11 cause the smallest the greatest number of cards one can have

toString: prints out player’s hand

**Dealer:**

DealerHand: creates an array of size 11 cause the smallest the greatest number of cards one can have

toString: prints out player’s hand

**Blackjack (Application):**

Main(String[]args): where all the main program is running and programmed

## Risks/Challenges:

List here what you think will be the hardest parts of coding the game and how you hope to address them (feel free to add more columns to the table!).

|  |  |  |
| --- | --- | --- |
| Challenge | How to address? | Effect if we don't address? |
| Creating the function where ace equals either 1 or 11 depending on what player chooses | To create a switch in the sum method after an input by player changing the ace value to 1 or 11 | Hardcode ace to be a 1 and hard code jack as 11 and all other face cards as 10 |
| Creating betting system | To create an input where you give value to an input value creating a method that is a Boolean and it can double your value if you win and false if you lose | Hardcode a money as input and just lose all and win double depending on result |
|  |  |  |
|  |  |  |

## Task List:

Create a list of *tasks* that you need to do to code the game. For each task, you should include the *priority* of the task (P0/P1/P2). Please see the phase 2 instructions document for more information about the priorities. **You should separate out tasks related to optional rules so that way you have an estimate of the required tasks.**

When you make your estimates, be sure to include time for *debugging* and *testing* your program. The time estimate should also include time spent ensuring that your code is *readable and modular*! You should expect to spend **at least** half of your time debugging/testing your program.

Note: There are 3 weeks of time between when you start the project and when it is due. You will have each lab period during this time to work on the project. This means you have six lab periods worth of in class time. You're also expected to spend 3 hours per week on homework, so you'll have an equivalent of six lab periods of outside class time. To have some buffer, you should ensure that your P0 tasks take up **at most** 6 "lab units" worth of time. (It can be less). Your P0+P1 tasks should be at most 9 "lab units" worth of time.

No task should be greater than 1.5 lab units worth of time in the table below! If it is larger, you should split it into smaller tasks!

|  |  |  |
| --- | --- | --- |
| **Task** | **Priority (P0/P1/P2)** | **Number of Lab Periods** |
| Define/test BlackJack main class | P0 | 1 |
| Define/test DynamicCardArray class | P0 | 1.5 |
| Define/test Player class | P0 | 0.5 |
| Define/test Dealer class | P0 | 0.5 |
| Define/test Deck class | P0 | 0.5 |
| Define/test Card class | P0 | 0.5 |
| Code the input system for the betting | P1 | 0.75 |
| Code and test Ace being 1 or 11 | P1 | 1 |
| UI game design look and how it is printed | P2 | 1 |
| Update sum class to be smoother and use more things such as maybe a switch case | P1 | 1 |
| Creating method where the dealer must hit again | P0 | .75 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |