Project Title: Blackjack Game + Learner

Contents (will update with headings when finished)

[My idea and where it came from: 3](#_Toc209439133)

[Research into existing, similar games: 4](#_Toc209439134)

[Example 1 – “Blackjack” by MSM Games: 5](#_Toc209439135)

[Example 2 – “Blackjack” by FreeGames: 6](#_Toc209439136)

[Example 3 – “Blackjack” by The Washington Post Games: 7](#_Toc209439137)

[Survey Questions, Results and Analysis 8](#_Toc209439138)

[Survey Results: 9](#_Toc209439139)

[Requirements for my game 12](#_Toc209439140)

[Identifying Limitations 15](#_Toc209439141)

[Hardware and Software Requirements 16](#_Toc209439142)

[Hardware Requirements: 16](#_Toc209439143)

[Software Requirements: 16](#_Toc209439144)

[Computational Methods 17](#_Toc209439145)

[Development Plan Stages: 18](#_Toc209439146)

[The Deck of Cards 18](#_Toc209439147)

[Player Moves 18](#_Toc209439148)

[GUI 18](#_Toc209439149)

[Menu 19](#_Toc209439150)

[Betting Chips (& money) 19](#_Toc209439151)

[Structure Diagram (for Development Plan Stages): 20](#_Toc209439152)

[Parts of the Game: 20](#_Toc209439153)

[GUI design: 21](#_Toc209439154)

[The Main Game: 21](#_Toc209439155)

[Cards: 21](#_Toc209439156)

[Chips: 22](#_Toc209439157)

[Buttons: 22](#_Toc209439158)

[Table: 22](#_Toc209439159)

[The Menu: 23](#_Toc209439160)

[Close Menu: 23](#_Toc209439161)

[‘MENU’ Title: 23](#_Toc209439162)

[Menu Options: 24](#_Toc209439163)

[Extra Details: 24](#_Toc209439164)

[Player Moves 25](#_Toc209439165)

[Flowchart: 25](#_Toc209439166)

[Betting Chips 27](#_Toc209439167)

[Flowchart: 27](#_Toc209439168)

[TO DO List / Feedback summary: 29](#_Toc209439169)

# My idea and where it came from:

For my project I came up with the idea to create a Blackjack game / learner, where you play against a computer operated dealer. This game would be used for multiple purposes; It could be used as a game for entertainment/testing your skill, learning how to play, and practicing certain methods such as perfect strategy and card counting.

The game is also free to play and does not use real money to gamble. This means people can play Blackjack without the risk of losing any real money and therefore does not encourage gambling. Additionally, because it is not gambling real money, it is legal for under 18-year-olds to play it.

I got this idea as I often play, and enjoy playing, Blackjack with my friends and family, where we do not gamble with real money. I currently do not know how to card count or play perfect strategy off by heart, which gave me the idea of implementing these strategies into the game as this would help me (and others) practice how to.

For my project, I will research similar, existing games of Blackjack and their different rules to figure out what I will, and will not, implement into my game.

Stakeholders of my game:

The stakeholders for my game are people that are teenagers or adults, however it can be played by people of any age, who already know how to play Blackjack. This is because it will not teach them the basic rules of Blackjack.

Out of the age range of my target audience, I feel that young and middle-aged adults will be the main users of my game; This is because people aged 36-50 statistically play Blackjack the most (35.1%), followed by people aged 21-25 (32%). *[I got these statistics from ‘*[*The Relationship of Demographics to Gaming Preferences and Behavior*](https://core.ac.uk/download/pdf/216977702.pdf)*’ Written by Michelle Millar and Seyhmus Baloglu in 2008].* Because of the age of my target audience, the design of game will not be overly simple (because it is not aimed for children) and will be reasonably detailed as older audiences can handle this.

This game is not aimed at any specific gender or race and can be played by any; However, this game is aimed at English speakers as it will be in English.

I believe my stakeholders will use my game for entertainment, practicing, or learning.

# Research into existing, similar games:

I have researched existing, similar games to assist in my plan for making my game. I have then noted what features I would, and would not, like to implement in my game from them.

By playtesting these games, I found out some rules (and features) for Blackjack I previously was not aware of which I need to include in my game:

* **Push:** When the player and the dealer stand on the same amount, it is called a push, and the player gets their money back.
* **Dealer hits on soft 17:** Unless the dealer has a hard 17 (the first two cards dealt total to 17), the dealer can keep hitting on a soft 17 (not a hard 17).
* **Insurance:** A side bet offered to players when the dealer's upcard (face up of their two cards) is an Ace, on whether they think the dealer will hit 21. I will make mine pay 2:1 (which is normally what casinos will payout)

Here are the games I researched (see next page):

## Example 1 – “Blackjack” by MSM Games:

A screenshot of a video game

AI-generated content may be incorrect.

Figure 1 - 'Blackjack' by MSM Games - [Play Blackjack in your browser | Games from MSN](https://www.msn.com/en-gb/play/games/blackjack/cg-9mt4hk2d6rcn?ocid=cganswer&cgfrom=cg_ans_game_playbtn)

This version of Blackjack feels the most realistic and cleanest with its design and layout; This makes it easy to see what is going on. However, this game does not have the option for multiplayer or a Perfect Strategy cheat sheet, which are things I will attempt to include in mine.

Out of the three existing similar games, the design and layout of this version is my favourite as:

* The buttons for what move you can do and what chips you can bet are constantly on screen (while non obstructive).
* Clear instructions in the top left corner from the “dealer” which also show who won the round.
* Clear numbers next to the chips to show how much you have bet.
* Clear numbers next to the cards to show the total you have.
* Clear counter/scoreboard on how much money the player has.
* Visually appealing with animations and seeing what you have bet in chips stacked up.
* The colour of the table changes (I could implement customisation).
* Game rules on the table.

I am going to base my design and layout similarly to this and implement some of these features.

## Example 2 – “Blackjack” by FreeGames:

A screenshot of a video game

AI-generated content may be incorrect.

Figure 2 - 'Blackjack' by FreeGames - [Play Blackjack | 100% Free Online Game | FreeGames.org](https://freegames.org/blackjack/)

The design of this version feels too childlike which is not my target audience. The reason why it feels like this is because of the round colourful buttons. The layout of the game also feels too squished rather than spaced out. These are features I will not include in mine as it will be aimed for teenage and adult audiences. This version also has quite simple designs and sound effects.

Features of the game I would like to implement in my game:

* Simplistic design.
* Sound effects.

This game also has long pauses between rounds. This is something I will try to avoid when creating my game as it can ruin the flow of the gameplay. Furthermore, you are only able to bet 20, 100, 500 or 1000 in this game, whereas I want the players of my game to be able to bet any amount (like ‘*Figure 1’*) as it is more realistic.

## Example 3 – “Blackjack” by The Washington Post Games:

A screenshot of a game

AI-generated content may be incorrect.A screenshot of a game

AI-generated content may be incorrect.

Figure 3 - 'Blackjack' by The Washington Post Games - [BlackJack | Instantly Play BlackJack Online for Free!](https://games.washingtonpost.com/games/blackjack)

This version of Blackjack feels the most simplistic layout wise as there are not many things on screen; However, it is clear on what the player can choose from. This game also does not have the option of multiplayer or a Perfect Strategy cheat sheet. I would like these options in my game therefore I will not be using this feature of this game.

This version also has the option to “Cash out and leave” which results to a win/loss screen (see above) which displays different stats; This is different to the other games which only allow you to exit the game. For my game I will allow the player to exit the game, but it will not display stats.

Features of the game I would like to implement in my game:

* Very clear on who won the round.
* It is clear when the player goes bust.
* The ability to exit the game.
* 6 card decks in the Shoe.
* Help menu explaining the rules of Blackjack (If I have time to).

# Survey Questions, Results and Analysis

To find out what I should prioritise when creating my game, I created a survey and sent it out to my stakeholders. Here are the questions I asked and why:

Q1: “*Have you played Blackjack before? (Also known as “21”)”* - I asked this question to find out how many of my stakeholders have already played Blackjack before (and would likely know the rules).

Q2: *“Did you enjoy it?”* – I have asked this question to find out how many of my stakeholders enjoyed playing the game.

Q3: *“Which would you prefer: A more detailed game, or a more simplistic designed game?”* – I asked this question to find out whether I should make my game more detailed (not super realistic but a more polished feel) or more simplistic.

Q4: *“What is more important to you: Animations or sound effects?”* – I asked this question to find out whether I should focus on the animations or sound effects for my game.

Q5: *“Please rank these uses of the game, of which you would be most likely to use it for: Playing, Practicing, Learning.”* – I asked this question to find out what my stakeholders would use my game for the most.

Q6: *“Please rank these features of the game in importance:*

* *Being able to exit the game*
* *Help menu (for basic rules of Blackjack*
* *Visual chip stack*
* *Toggleable Card Count counter*
* *Toggleable Perfect Strategy cheat sheet*
* *Having 6 cards in the shoe*
* *Instructions/messages from the (computer) dealer*
* *Option to ask the dealer for a hint”*

– I asked this question to find out what features my stakeholders think are the most important.

Q7: *“Finally, would you play this game?”* – I asked this question to find out whether my stakeholders would play my game after hearing what it is about and what will be included.

# Survey Results:

##### Q1: A screenshot of a computer AI-generated content may be incorrect.

##### Q2: A white background with black text AI-generated content may be incorrect.

##### Q3: A screenshot of a computer AI-generated content may be incorrect.

##### Q4: A screenshot of a computer AI-generated content may be incorrect.

##### Q5: A screenshot of a computer AI-generated content may be incorrect.

##### Q6: A screenshot of a computer game AI-generated content may be incorrect.

##### Q7: A white background with black text AI-generated content may be incorrect.

# Requirements for my game

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Feature** |  | **Sub-features** | **Explanation** | **Justification** | **Importance** |
| 1. | Menu/options | a | Being able to exit the game. | A button that allows the user to exit the game. | Without this, the game would continue for an endless loop. | Important |
| b | Toggleable Perfect Strategy cheat sheet. | A button that toggles an on-screen cheat sheet. | This would allow for a learning aspect to my game which is what I intended it to have. | Important |
| c | Start button | A button presented to the player when they are not in a game. | Without this, the game would start automatically. By having a start button, it makes the game feel more polished. | Not important |
| d | Title screen | When the player is not in a game, a title screen is presented to them. | This would make the game feel more polished and finished. A polished game is more likely to be played than an unpolished game. | Moderately important |
| e | Toggleable Card Count counter | A button that toggles an on-screen Card Count counter. | This would further add to the learning aspect of my game. | Slightly important |
| 2. | Help Menu |  |  | Navigational menu on the basic rules of Blackjack. May be part of the main menu or will be a separate menu. | Players who do not already know the rules will need this to play the game. Furthermore, people who know the rules may need reminding. | Important |
| 3. | Stacked poker chips |  |  | Purely visual. See what you have bet, stacked in front of you. | This would be purely visual as there is already a counter on-screen for the total value you have bet. | Not important |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Feature** |  | **Sub-features** | **Explanation** | **Justification** | **Importance** |
| 4. | Toggleable cheat sheet |  |  | Toggleable perfect strategy cheat sheet. | This feature is important as it is what will make my game stand out from other existing similar games. | Important |
| 5. | Moves the player can make | a | Stand | The player’s turn ends with what upcards they currently have. | This feature is needed for the game to function correctly. | Essential |
| b | Hit | The player gets dealt another card. | This feature is needed for the game to function correctly. | Essential |
| c | Split | When the player’s (2) upcards are equal, they have the option to split their hand into multiple hands. | This feature is needed for the game to function correctly. | Essential |
| d | Double down | The player doubles their initial bet and commits to taking only **one** additional card. This move is only available before the player has ‘hit’. | This feature is needed for the game to function correctly. | Essential |
| e | Bust | The player’s upcard total goes over 21, they are removed from the round and lose their poker chips. | This feature is needed for the game to function correctly. | Essential |
| f | Insurance | Side bet where the player bets that the dealer will get 21. | This feature is not needed for the game to function correctly; however, it is a feature of the game Blackjack. | Not important |
| **No.** | **Feature** |  | **Sub-features** | **Explanation** | **Justification** | **Importance** |
| 6. | GUI | a | Current bet total counter | The total amount that the player has currently bet will be displayed on screen. | This will help the player see more clearly how much they have bet. | Essential |
| b | Current upcard total counter for the player | The total amount that the player’s upcards add up to will be displayed on screen. | This will help the player see more clearly what their running total is. | Essential |
| 7. | Game rules on the table | a | Blackjack pays 3:2 | When a player gets a Blackjack, they will be paid 3:1. | This rule is needed to ensure the amount the player wins is a fair amount. | Essential |
| b | Dealer must stand on 17 and draw to 16 | If the dealer’s upcards add to 17, the dealer has to stand. If the dealer does not, the dealer has to draw until they have 16. | This rule is needed to make the game more balanced. | Essential |
| c | Insurance pays 2:1 | If the player wins the insurance side bet, they get paid 2:1. | This feature is needed for the game to function correctly (If I include insurance as a feature). | Important  (if I include Insurance) |

# Identifying Limitations

Below are some limitations and extra features that I am not going to include within my game, explanations for each of them, and reasoning:

|  |  |
| --- | --- |
| **Limitations** | **Explanation** |
| Multiple card decks in the shoe | Although this would make my game more realistic and increase the difficulty, this would be too time consuming for me to do in this project. Because of this, I will be using one card deck (as opposed to a desired 6). This will ensure than I finish my project in time and to a good standard. |
| Language | My game will be in English. I will not implement language options; This means non-English speakers will not have a translated game. The reason I have decided this is because I will not have enough time to add this to my game. |
| Customisation | For my game, I am not going to implement customisation. As much as I enjoy customisation in games, I believe my time will be better spent on other parts of the game. If I were to implement customisation, some examples of what I would add would be:   * Card back design + colour * Table felt design + colour * Table edge design + colour   Another idea would be having unlockable cosmetics that you can purchase. |
| Leaderboards and Personal scoreboard | Leaderboards comparing different players’ stats/data.  Personal scoreboard with stats such as:   * Highest amount of money * Total Blackjacks * Saved customisation.   I am not going to add these features to my game. Although it will improve it, I think it would be better to use my time to improve the main aspects of my game. |
| Save data | The ability to save player data.  I am not going to add this feature as it is quite complex to implement. I would prefer to spend my time on the main parts of my game as opposed to this. |
| Multiplayer | For my game I originally wanted to implement multiplayer, however, I am not going to add this feature as it is quite complex and time-consuming. |

# Hardware and Software Requirements

I have chosen to code my game and GUI in Greenfoot. This is because I have used this software before and feel comfortable coding a game using it.

## Hardware Requirements:

Any device with a touchscreen user interface or mouse inputs such as:

* Mobile phone
* Tablet/iPad
* Laptop
* Desktop PC

These devices won’t need to be very powerful as my game is quite basic with what will need to be loaded.

The device may require speakers if in-game sound effects/music is to be played.

A keyboard (virtual/physical) will **not** be needed as the game does not require text input.

## Software Requirements:

A device that uses Windows, macOS, or Linux, or a device that supports a recent Java Virtual Machine (JVM). This is because my game will be made using Greenfoot and has these requirements to play the game I will make.

# Computational Methods

I believe that this problem is a computational problem. By having it as a game that is played digitally, it is more convenient and accessible to play. This is because you do not need a physical table to play on, a deck of cards, or poker chips, and allows for portability.

To make my game, I will use **decomposition** to break down the different sections of my game into more manageable, smaller sub-problems such as:

* The betting aspect
* The deck of cards
* The player’s moves (I will then break this down into the individual moves themselves such as: Stand, Hit, Split, Double Down)
* Calculating the winner

I will also use **abstraction** for my game; The user will only be presented with necessary information on-screen needed to play the game such as:

* The player’s cards and their total value.
* The dealer’s cards and their total value.
* The player’s bet total.
* The player’s possible moves for the game, they won’t be presented with options such as: “sit at the table”, “get up from table”.

Furthermore, by using abstraction, I will not have to show the deck of cards being shuffled, or the dealer itself.

Development

# Development Plan Stages:

### The Deck of Cards

I need to figure out how to and create a functioning deck of 52 cards.

These cards will need to be able to be ‘**shuffled’**, whether it be a random card from the 52 is selected, or if they are put into a random order and **dealt** in that order. Cards will need to be **dealt** as **face-up and face-down** cards. Cards will also need to be **discarded** at the end of a round.

### Player Moves

The player will have a choice of moves in which they can do on a turn based on what cards they have. These moves will be:

***Stand*** – Stick with the cards that have been dealt to you so far and end your turn.

***Hit*** – Have another card dealt to you.

***Split*** – If your starting hand is two cards of equal value, the player can split these and play them as separate hands.

***Double******Down*** – Put more money down, and be dealt one final card, your turn is then ended after this.

***Bust*** – The player’s total upcard values add up to over 21. Their turn ends and they lose the round.

### GUI

A GUI that the player can see and interact with. This includes:

***‘Current bet total counter’*** displays the players total bet in that round.

***‘Current upcard total counter’*** displays the player’s total upcard value.

***Face-up & Face-down Playing cards*** on the table.

***Game rules on the table***, such as “*Blackjack pays 3:2*”, “*Dealer stands on 17 and draws to 16*”, and “*Insurance pays 2:1*”.

***Visually stacked poker chips*** that the player has bet (purely visual as opposed to the other GUI features).

### Menu

A menu where the player can choose from several options:

***Exit to title screen*** – The player can exit the game to the title screen; this is also where they start when the game itself starts.

***Start button*** – Starts a game of Blackjack.

***‘Toggle Perfect Strategy Cheat Sheet’*** – Toggles an on-screen Perfect Strategy cheat sheet.

***‘Toggle Card Count counter’*** – Toggles an on-screen Card Count counter.

***Help Menu*** – Shows the basic rules to Blackjack.

### Betting Chips (& money)

Bet-able poker chips that the player bets each round.

These **chips will be clickable** by the player to bet them. There will be a **selection of different valued chips**: ‘*£1, £5, £10, £25, £100, £200’*. I have chosen these values as they are accurate to what valued chips you may have at a casino. The player won’t be able to bet more money than they have. When the player runs out of money the game ends and they lose.

# Structure Diagram (for Development Plan Stages):

*A screenshot of a computer screen

AI-generated content may be incorrect.*

Figure 4 – ‘Project Structure Diagram 2.0’

I have created a structure diagram for my game so I can clearly see what parts to be made and what features each part has. By breaking my game into parts using decomposition, it will make it easier for me to work on my game in parts.

## Parts of the Game:

*Deck of Cards Player Moves GUI Betting (Poker) Chips Menu*

**Player Moves** and **Deck of Cards** are the most important part of my game as they are needed to function, therefore I will be making them simultaneously first alongside my **GUI** (as the **GUI** is needed for the game to be used).

**Betting Chips** will be made next as betting is an important part of Blackjack (despite not being needed for the game to function).

Finally, I will make the **Menu** and title screen and add any extra features (depending on how much time I have left).

I have chosen this order as it is in order of importance for the game to function correctly. The stages I will be spending the most time on is ‘***Deck of cards***’, ‘***Player Moves***’, and ‘***GUI***’, this is because these stages are vital for the game to function. Whereas ‘***Menu***’ and ‘***Betting Chips***’ are not vital for the game’s functionality.

# GUI design:

## The Main Game:

A screenshot of a game

AI-generated content may be incorrect.

Figure 5 – ‘Blackjack Game Concept GUI’

### Cards:

For the cards I have decided to position them where they would be on a real Blackjack table. This is to add to the realism and so that it is in a layout players may be used to.

I have also decided to have the face of the cards be a plain white so that the values are easy to see. This would also be good for people with poor eyesight, potentially due to age or certain conditions, and therefore makes my game more accessible.

I have coloured the card backs red so it is clear it is not the card face; however, I may change these or give them a more designed back when creating the assets.

I will also have a total card value counter next to the player’s cards to assist in the game’s ease of use (see above ‘K♦️ card’).

I have also made a discard pile (top left) and draw pile (top right) and put them towards the edges, so they are not obtrusive. I will have text by the discard pile, so people do not get confused as to why there are two piles of cards.

### Chips:

I have made poker chips for values: 1, 5, 10, 25, 100, and 200, and coloured them accurately to poker chips in real life to add to the realism and immersion.

I have also added a total bet counter to further improve the games ease of use as it allows the player to see exactly how much they’ve bet.

Furthermore, I have added a total money counter so the player can see how much money they have overall. This money counter is a warm yellow, this is because it stands out clearly, without being overbearing, and yellow is often associated with money.

The poker chips, once bet, are ‘placed’ in a yellow circle at the bottom of the table, this is so the player can see what they’ve bet (making it more immersive). I decided on the colour yellow because it stands out against the green table. Once there are more than 5 (maybe 10 depending on how the final GUI looks), another pile of chips will start to the side, this is so they do not get in the way of the cards if the player was to bet a lot of money.

### Buttons:

There are several buttons the player can press. These buttons have been coloured, so they stand out. The player’s moves (pale pink) are a different colour to the ‘reset’ button (warm yellow) as it is to do with the poker chips rather than the player’s moves.

There is a menu button that can take the player to the menu, here they will be able to access settings and help. This button is white, this is so it not only stands out on the table, but also from the other buttons.

The player’s moves also have buttons. These include Hit, Stand, Double down, and Split.

As extra buttons (and clickables), the player can click on poker chips to bet them, there is a reset button in case the player bets the wrong more/misclicks any poker chips. Finally, there is a ‘deal’ button, this starts the round, and the ‘dealer’ deals out cards.

### Table:

The table is a green colour, I have chosen this because Blackjack tables in real life, often have a green felt top. I have tried to make mine a similar colour, so the game is more immersive.

I have added the rules on the table: “Blackjack pays 2:1”. However, this looks slightly off and is missing some rules as they did not fit, I will need to fix these problems when I make my assets.

## The Menu:

A screenshot of a game

AI-generated content may be incorrect.

Figure 6 - ‘Blackjack Menu Concept GUI’

### Close Menu:

I have put the ‘Close Menu’ in the same place as the ‘Menu’ button in the main game (see figure 5) as that is normal for some games and will be easy to find. I have made the button white so that it stands out clearly against the green table.

### ‘MENU’ Title:

I have made the title ‘M E N U’ using the most vibrant poker chips from the main game (see figure 5) and changed the values to the letters ‘M’, ‘E’, ‘N’, ‘U’. The vibrant poker chips make it stand out and look appealing.

### Menu Options:

There a four menu options: ‘Perfect Strategy Cheat Sheet’, ‘Card Count Counter’, ‘Exit game’, and ‘Basic Rules’. I have designed them to look like playing cards shape and colour wise, and each have a playing card suit on them (as there are four menu options).

#### Perfect Strategy Cheat Sheet

Displays a perfect strategy cheat sheet

#### Card Count Counter

Displays the current card count

#### Exit Game

Exits the game

#### Basic Rules

Displays the basic rules to Blackjack

### Extra Details:

I have also put a stack of face down cards; these are roughly in the same position and colour of the face down stack in the main game (see figure 2). I have made all the card shapes larger to fill out the emptiness of the table.

Iterative Development

# Player Moves

## Flowchart:

A screenshot of a computer

AI-generated content may be incorrect.

Figure 7 - 'Player Moves - Flowchart'

A screenshot of a computer

AI-generated content may be incorrect.

Figure 8 - 'Player Moves - Flowchart'

I have made this flowchart as a rough overview of how the Player’s Moves will work in my game. I have coloured the flowchart for easier viewing, with subroutines and related code different colours to each other.

When it is the player’s turn, if they click on certain buttons, they will run different subroutines. These subroutines include:

* ‘hit(hand)’
* ‘stand(hand)’
* ‘split(hand)’
* ‘double\_down(hand)’

These subroutines take ‘hand’ as a parameter because players can have more than one card hand (because of the option of doubling down).

## Data Dictionary:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data type | Description | Validation |
| playerLost | Boolean | Stores whether the player has lost the round or not. | ‘true’ or ‘false’ |
| handTotal | Integer | Stores the total value of the cards in a player’s hand | >= 0 |
|  |  |  |  |

## Test Table:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Betting Chips

## Flowchart:

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 9 - 'Betting Chips - Flowchart'

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 10 - 'Betting Chips - Flowchart'

I have made this flowchart to summarise how the poker chips betting system will work. I have also coloured this flowchart for easier viewing. The player can click on 6 different poker chips, depending on what one(s) they bet, they will bet the corresponding amount of money.

The player can also press ‘Reset’ which resets their bet, and ‘Deal’ which will start the game with the amount of money they have bet, as long as they have bet an amount.

# Deck of Cards

## Flow Chart:

A screenshot of a computer

AI-generated content may be incorrect.

Figure 11 - "Deck of Cards - Flowchart"

## Developing the Class:

I have several options for making my game:

1 – When cards are dealt, it is a randomly generated card, meaning you could get one card dealt repeatedly forever (despite the probability being extremely low). This would mean the game has an ‘infinite’ number of cards in the shoe which is not good.

2 – Have an array of what cards have been played already. When a new card is dealt, it is randomly generated. However, it will first check the generated card to see if it is in the list of played cards, if it is, it will attempt to generate a new card until it is unique. I would need to use a search algorithm to do this.

3 – Having an array of cards that can be mixed, and cards removed from as they are dealt.

*I am going to go with option 2, as this is a solution that is not too complex but will still function the same way as I planned for my game.*

## Coding the Class:

### Defining variables:

A screenshot of a computer

AI-generated content may be incorrect.

Figure 12 – Defining variable for the DeckOfCards class

For the **DeckOfCards** class, I have created an Array List, **playedCards**, to store the cards that have already been dealt in the game. This is because if they have already been dealt, I can check a generated card against this list, and they should not be able to be dealt again.

As this could potentially be a large list of cards (or a small list), it made more sense to me to have a non-static data structure to store them in. Because of this, I have chosen to use an Array List, however I have not coded them in Java before, so I had to research how to do it. I used this website: [Java ArrayList](https://www.w3schools.com/java/java_arraylist.asp). To use an Array List, I had to import ‘java.util.ArrayList’.

1. Define the class
2. Define instance variables
3. Create constructors – define object in the system
4. Create get/set methods
5. Create any custom methods

Used:

[Java Arrays](https://www.w3schools.com/java/java_arrays.asp) – W3Schools

*I used this to remind myself how to create an array in Java. I needed to create an array for the possible suits and values the cards in my game could have.*

<https://www.greenfoot.org/topics/63841/0>

*I used this to remind myself how to make a random number in Greenfoot. This was needed so I could generate random numbers to then be used to generate random cards. One random number will correspond to a card value, and another random number will correspond to a card suit.*

<https://www.w3schools.com/java/java_switch.asp>

*I used this to remind myself how to use Switch Case in Java.*

<https://stackoverflow.com/questions/5071040/java-convert-integer-to-string>

*I used this to remind myself how to cast an integer into a string in Java.*

# TO DO List / Feedback summary:

(In vague order of importance)

* Create concept GUI for the menu (Could be drop down or a whole screen, it should not be interactable when it is not visible either by having a variable that makes them non-clickable or move off screen). Fixed – Made the concept GUI for the menu (see figure 6).
* “The design plan could include an approximate time frame for the completion of the stages while leaving room for testing to make sure that you will get it all done in time.” I need to add time frames for the completion of the stages. This is important, as said, so that I get the project completed on time.
* Potentially make the poker chip mini menu into a pull out/drop down menu.
* Money boxes on structure diagram are cluttered. Fixed – Replaced old structure diagram (Project Structure Diagram) with the updated version (Project Structure Diagram 2.0) see figure 4.

This has less clutter and has white boxes with black text (as opposed to the black boxes with white text), as these changes make it easier to read/view & understand.