# Cameron Cottam

1901441

Fruit Machine Assignment.

# How I planned it out

# Pseudocode

# V1:

User Enters balance

Slots Rotate

Win check

Continue/Quit

Separate each section into functions then adding it all in the int main.

# V2:

Additional features:

Slow

Holding and pushing down one chosen slot.

Display Characters

# V3:

Adding fixes to the stability to prevent crashes.

Bonus game mode.

I first started my program by creating a function called Logic. Which basically the user inputs. So for example, their balance, bet, how many lines they wanted to play for. One of the problems I occurred was that if the user wants to continue playing the game, it would make them re-enter their balance in the current session. So, to counter this I created a a variable called Account Attempts which basically after the user enters their balance the first time. It will not ask them to re-enter their balance.

I created a while statement that would execute if the users balance was less than 0. To prevent the user from entering a negative value. I did the same for the bet variable by making the max bet £5.0 and the minimum £0.10. One issue I discovered was that the user could enter their bet that was greater than their balance (which would leave the balance as a negative). So, to stop this, I created an if statement that if the user’s bet was greater than the balance it would instantly end the game. However, I did create a message to display to the user warning them that their balance should be greater than their bet.

The next part of my program was the slots and rotating them. I created three char arrays that contained the same symbols with the same positions and another three arrays with the same symbols but in a different order. I had an issue where the same symbol would display the same symbol in the same column. To fix this issue, I assigned each position of the array to the three shuffled arrays. For example, Column [0] = Mix1[index % 3] Column [1] = Mix2[index % 3]

The use of index is as a counter in a for loop while the modulus is used to divide it by 3 and take the remainder and add it to the array.

# Functionality

I’ve added the functionality for the user to play for multiple lines. So middle, top, bottom, diagonals etc. I also added the ability for the user to hold twos and push one slot of their choice down. I have also a slow mode. Which allows the user to slow down the rotating to give the user a chance to predict the slots. However, whenever it is press to slow down, I have a deductions counter which by the results stage will deduct the users balance by 10.

I have also added in an optional bonus mode for the user, if they’ve played the game 3 times. It’s known as the All-In mode. Where the user puts all their money into a bet and slots spin with the middle line only active and the user can only win the jackpot if all of the symbols are matching.

# Security

I’ve used cin.fail() to prevent the game from crashing or constantly printing a message, if the user enter a wrong data type. So for example, if the user was asking for a balance and enter the letter a it’s used to ask the user to re-enter their balance with the correct balance.

# Ending

After the Results have been conducted, the game will ask the user if they want to continue. If the user enters yes, it will reset the bets. If the user does not enter yes it will end the game and display a message saying ‘Game Over’