```python
def Setup():
    for i in range(MaximumNoOfGueses):
        for j in range(Pegs):
            Guesses[i].append(Black)
            ListOfPegs[i].append(Black)
```

This subprogram appends black to the guesses and list of pegs which basically sets a blank canvas for the game to swap the coloured pegs with.

```python
def SetUpTheBoard():
    replit.clear()
    divider = BG + Pixel * (1 + Pegs) + Pixel + Black + Pixel + BG + Pixel * (1 + Pegs)+ Pixel + End
    print(divider)
    for i in range(MaximumNoOfGueses):
        to_print = BG + Pixel
        for j in Guesses[i]:
            to_print += j + Pixel
        to_print += BG + Pixel + Black + Pixel + BG + Pixel
        for j in ListOfPegs[i]:
            to_print += j + Pixel
        to_print += BG + Pixel + End
        print(to_print)
        print(divider)
    print()
    divider2 = BG + Pixel * (2 + Pegs) + End
    if GuessNum == MaximumNoOfGueses or GameEnded:
        to_print = BG + Pixel
        for i in TheSequence:
            to_print += i + Pixel
        to_print += BG + Pixel + End
    else:
        to_print = BG + Pixel + Black + Pixel * Pegs + BG + Pixel + End
    to_print += Pixel
    for i in Colours:
        to_print += i + Pixel + End + " "
    print(divider2 + Pixel + "Available Colours:")
    print(to_print)
    print(divider2 + Pixel + "R  O  Y  LG DG LB DB P")
```

This subprogram decides whether the guesses are the right colour and position. It then decides what red and white pegs it needs to display to the user.

```python
def UserGuess():
    global Guesses
    for i in range(Pegs):
        Guessed = False
        while not Guessed:
            SetUpTheBoard()
            print()
            print("Please enter the color of the next Peg of your guess")
            Peg = input("using the Abreviations in the chart of available Colours:\n").lower().strip()
            if Peg not in Abreviations:
                print("You did not enter a valid input. Please try again.")
                time.sleep(1)
                continue
            else:
                Guessed = True
                Guesses[GuessNum][i] = Colours[Abreviations.index(Peg)]
    SetUpTheBoard()
    time.sleep(0.5)
```

This subprogram runs another subprogram that sets up the board but it mainly focuses on your input and whether its valid and if it is valid to continue the program and add the colour pixel to the board or get the input to do so.

```python
def CorrectOrNot():
    global ListOfPegs
    global Won
    RedsToAdd = 0
    WhitesToAdd = 0
    CurrentGuess = Guesses[GuessNum][:]
    TheSequenceclone = TheSequence[:]
    count = 0
    while count < len(CurrentGuess):
        if CurrentGuess[count] == TheSequenceclone[count]:
            RedsToAdd += 1
            del CurrentGuess[count]
            del TheSequenceclone[count]
        else:
            count += 1
    count = 0
    while count < len(CurrentGuess):
        if CurrentGuess[count] in TheSequenceclone:
            WhitesToAdd += 1
            TheSequenceclone.remove(CurrentGuess[count])
            CurrentGuess.remove(CurrentGuess[count])
        else:
            count += 1
    if RedsToAdd == Pegs:
        Won = True
    for i in range(Pegs):
        if RedsToAdd > 0:
            RedsToAdd -= 1
            ListOfPegs[GuessNum][i] = Red
        elif WhitesToAdd > 0:
            WhitesToAdd -= 1
            ListOfPegs[GuessNum][i] = White
        else:
            break
```

This Subprogram sets up the playing board by individually placing and colouring the pixels and using variables to link the inputted values to appear as pixels in the board. It also prints the display for the available colours.

```python
def EndOfGame():
    global GameEnded
    GameEnded = True
    SetUpTheBoard()
    if Won:
        print("Well done, you beat Mastermind")
    else:
        print("Good Try, but you lost")
    print("\n" * 2)
```

This subprogram prints a victory or defeat message depending on whether the amount of guesses have been used up or if the player has guessed the correct sequence.

```python
def Loop():
    global GuessNum
    UserGuess()
    CorrectOrNot()
    GuessNum += 1
    SetUpTheBoard()
    if Won or GuessNum == MaximumNoOfGueses:
        EndOfGame()
    else:
        Loop()
```

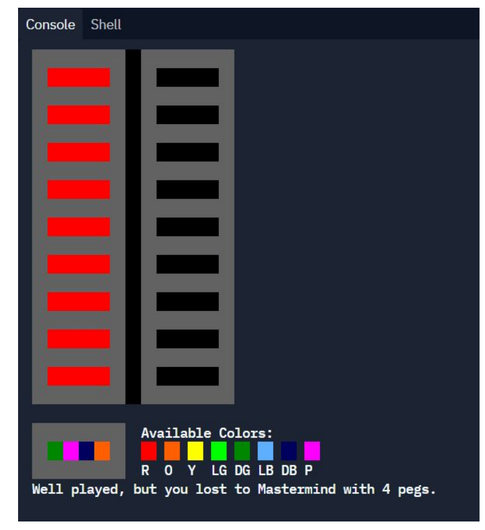
```python
import random
import time
import replit
global Pegs
global MaximumNoOfGueses
global TheSequence
global Guesses
global ListOfPegs
Red = "\033[48;5;196m"
Orange = "\033[48;5;202m"
Yellow = "\033[48;5;226m"
LightGreen = "\033[48;5;46m"
DarkGreen = "\033[48;5;28m"
LightBlue = "\033[48;5;75m"
DarkBlue = "\033[48;5;17m"
Pink = "\033[48;5;201m"
Black = "\033[48;5;16m"
White = "\033[48;5;231m"
BG = "\033[48;5;241m"
Pixel = "  "
End = "\033[0m"
Colours = [Red, Orange, Yellow, LightGreen, DarkGreen, LightBlue, DarkBlue, Pink]
TheSequence = []
Pegs = 0
MaximumNoOfGueses = 0
GuessNum = 0
Guesses = []
ListOfPegs = []
GameEnded = False
Won = False
Abreviations = ["r", "o", "y", "lg", "dg", "lb", "db", "p"]
print(Red + "Hello and welcome to Mastermind The Game, where the player tries To uess a the sequence of 4 colored Pegs.  The player fills up a row with their guess, then submits their guess n If they have one in the correct color and position, eg dsplayed.\n If it' in the wrong place but is the right color, a white peg is displayed.\n If the player can guess the the sequence fully without a set number of moves, they win!")
print()
print(Pink + "Press 'ENTER' to continue")
input()
Pegs = 4
MaximumNoOfGueses = 3 * (Pegs - 1)
for i in range(Pegs):
    TheSequence.append(random.choice(Colours))
for i in range(MaximumNoOfGueses):
    Guesses.append([])
    ListOfPegs.append([])
print()
Setup()
SetUpTheBoard()
Loop()
```

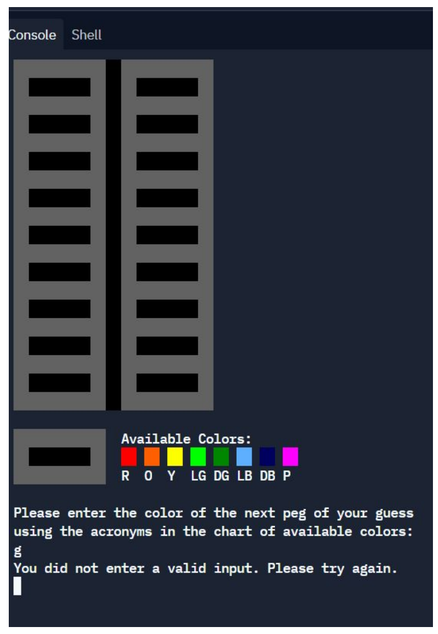
This part of the main code is all of the variables and colours being initialised. It also welcomes the player to the game and prints the rules.

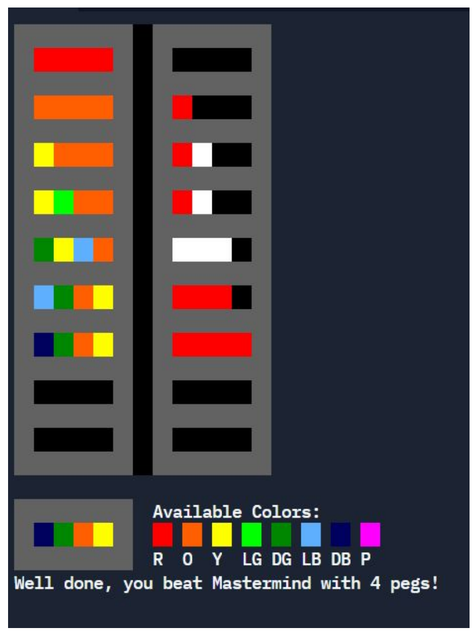
These for statements randomise the colours and add to the amount of guesses you've had which therefore allows the game to move on to the next row in the table

This subprogram makes the game loop and checks for win conditions.

Tests







Sam Easterbrook
A Level Computer Science
Year 12