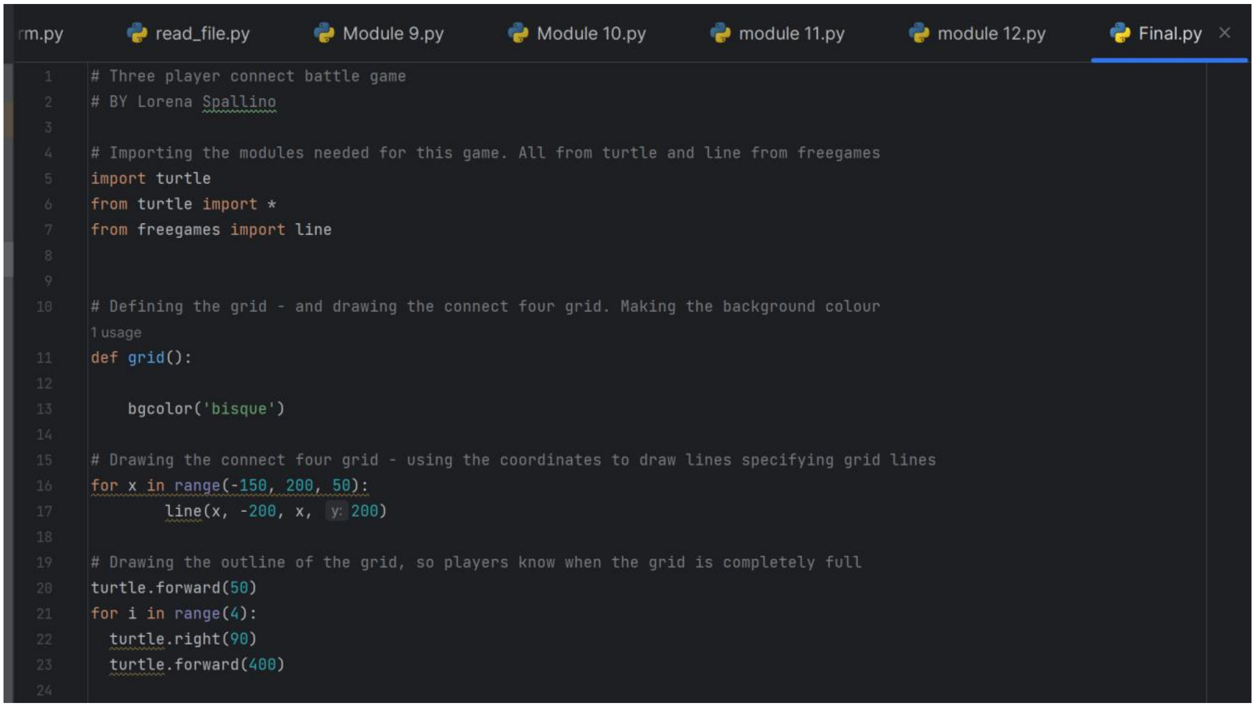
Python Game – Lorena Spallino

One great way to showcase one's experience and knowledge of python is by creating a simple game using python methods, functions, and libraries. To showcase my expertise using Python, I have created a simple connect game. It sounds very simple, and not very interesting however, there is a catch. This is no ordinary connect four games, this is a three-player connect battle. The purpose of this game is to allow three people to play on one computer, filling the entire grid with coins. Each player alternates turns, dropping one coin in at a time, by the end of the game, when the entire board if filled up, each player counts the longest uninterrupted line, either vertical, horizontal or diagonal, and the player with the longest streak is declared the winner! Below are the snapshots of the entire code using Python's IDE called PyCharm.

As evidently shown in the screenshots to the right, this game required a fair bit of python coding. The comments explain what each bit of code is, and why it is utilized for this game. As seen at the beginning, this game not only uses python functions, lists, loops, and dictionaries, but also makes use of 2 python modules, turtle and freegames.



```
# Three player connect battle game
# BY Lorena Spallino

# Importing the modules needed for this game. All from turtle and line from freegames
import turtle
from turtle import *
from freegames import line


# Defining the grid - and drawing the connect four grid. Making the background colour
1 usage
def grid():

    bgcolor('bisque')

# Drawing the connect four grid - using the coordinates to draw lines specifying grid lines
for x in range(-150, 200, 50):
        line(x, -200, x,  y: 200)

# Drawing the outline of the grid, so players know when the grid is completely full
turtle.forward(50)
for i in range(4):
  turtle.right(90)
  turtle.forward(400)

```

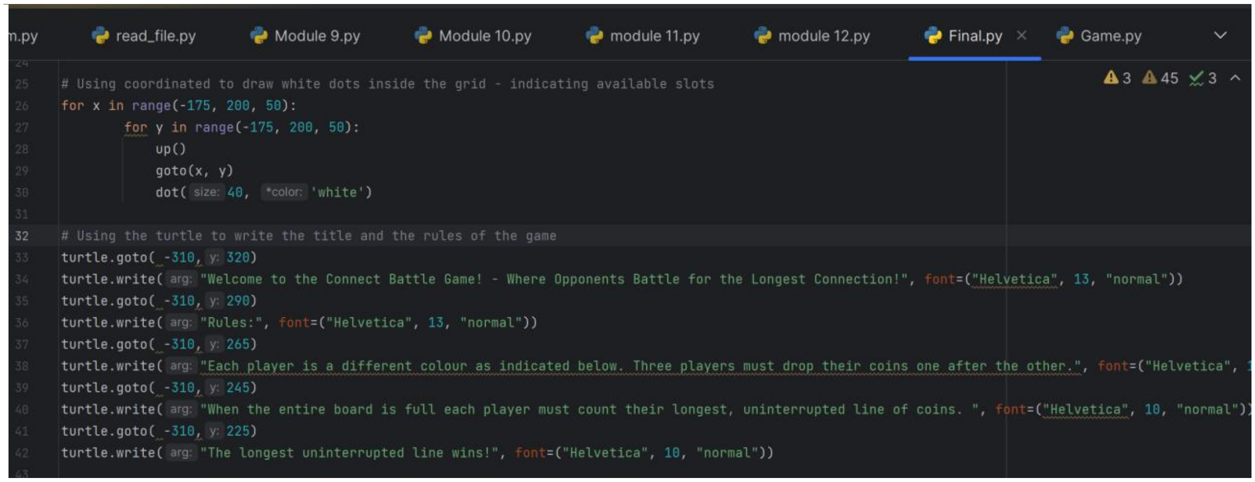
Furthermore, these snapshots of code display the setup of the game board, or grid using the ‘line’ aspect of the freegames module and the turtle module. Using loops, I have drawn a board on the screen, and included white ‘dots’ to indicate where the players can drop their coins.

Then, I have printed the intended rules of the game on top of the board, and displayed each player’s colour below.

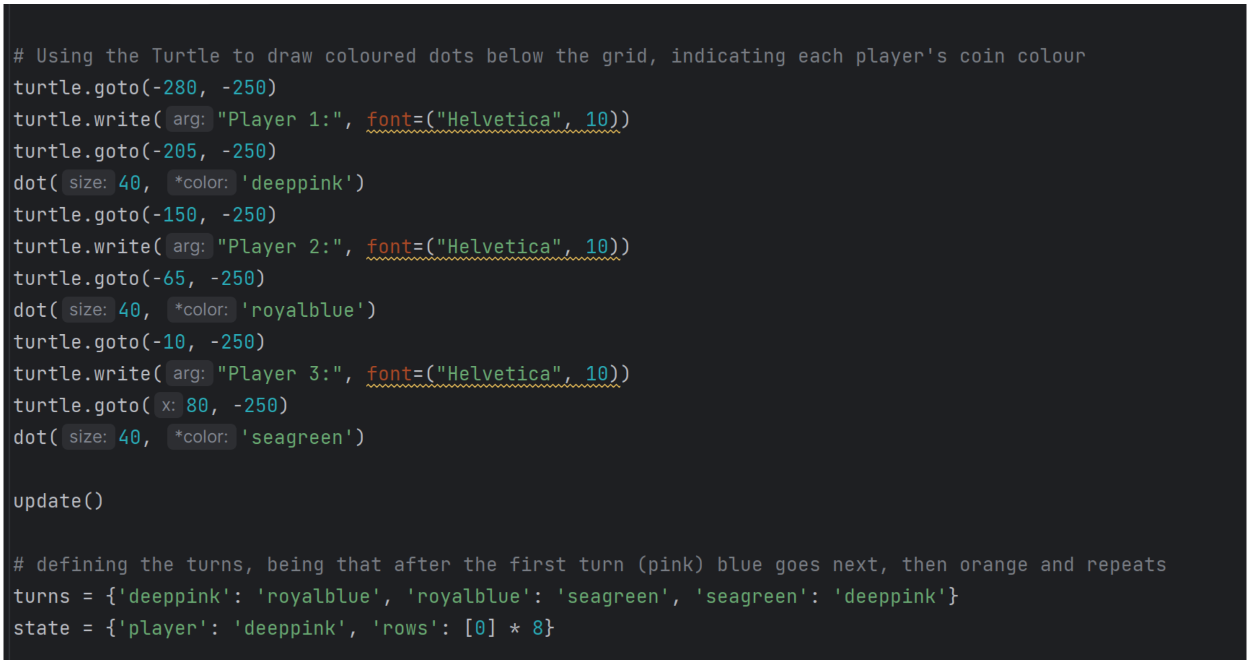
After that has been updated, the functions responding to the player’s clicks can begin.

The first two are dictionaries indicating how the turns work – player one then two then three – then the state.

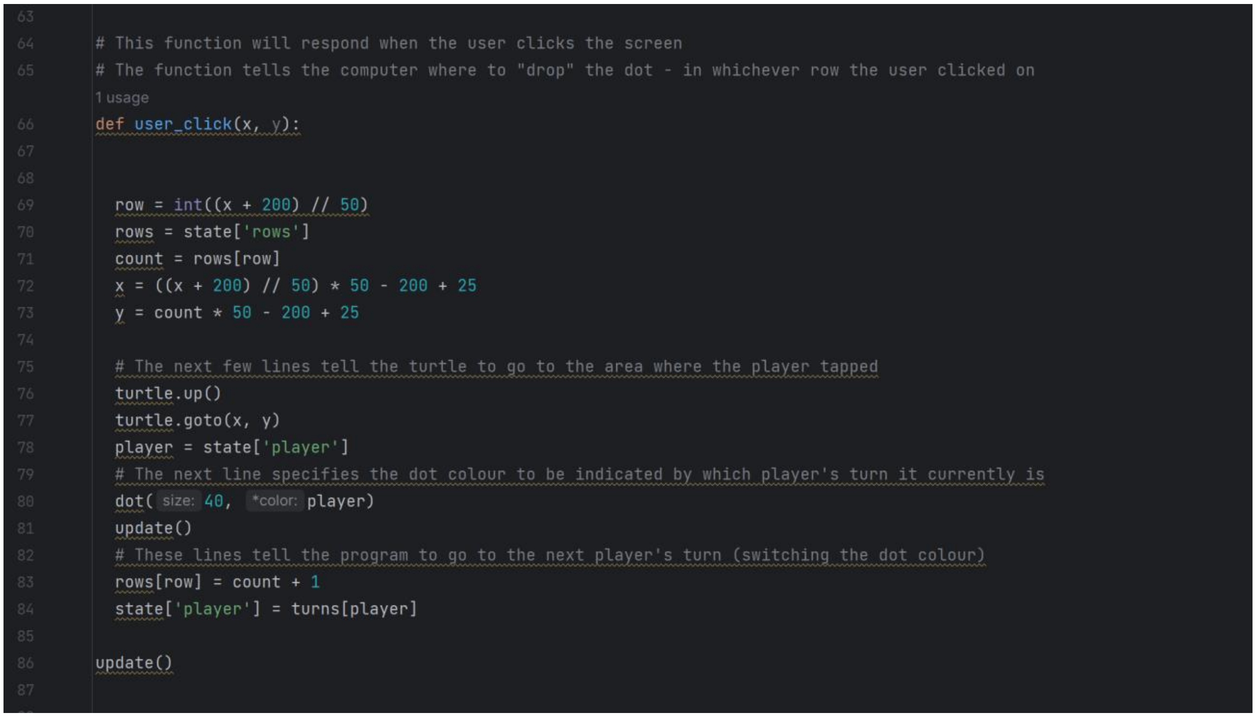
Then comes the function called user_click, which responds to the players tap by dropping a dot, or coin in the row where the user clicked.



```
# Using coordinated to draw white dots inside the grid - indicating available slots
for x in range(-175, 200, 50):
        for y in range(-175, 200, 50):
            up()
            goto(x, y)
            dot(40, 'white')

# Using the turtle to write the title and the rules of the game
turtle.goto(-310, 320)
turtle.write("Welcome to the Connect Battle Game! - Where Opponents Battle for the Longest Connection!", font=("Helvetica", 13, "normal"))
turtle.goto(-310, 290)
turtle.write("Rules:", font=("Helvetica", 13, "normal"))
turtle.goto(-310, 265)
turtle.write("Each player is a different colour as indicated below. Three players must drop their coins one after the other.", font=("Helvetica", 1
turtle.goto(-310, 245)
turtle.write("When the entire board is full each player must count their longest, uninterrupted line of coins. ", font=("Helvetica", 10, "normal"))
turtle.goto(-310, 225)
turtle.write("The longest uninterrupted line wins!", font=("Helvetica", 10, "normal"))
```



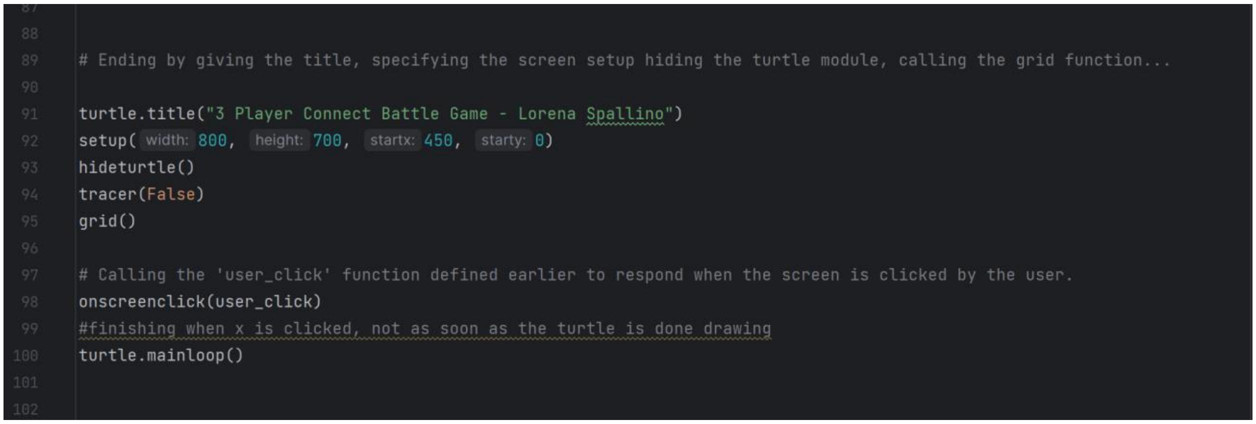
```
# Using the Turtle to draw coloured dots below the grid, indicating each player's coin colour
turtle.goto(-280, -250)
turtle.write("Player 1:", font=("Helvetica", 10))
turtle.goto(-205, -250)
dot(40, 'deeppink')
turtle.goto(-150, -250)
turtle.write("Player 2:", font=("Helvetica", 10))
turtle.goto(-65, -250)
dot(40, 'royalblue')
turtle.goto(-10, -250)
turtle.write("Player 3:", font=("Helvetica", 10))
turtle.goto(80, -250)
dot(40, 'seagreen')

update()

# defining the turns, being that after the first turn (pink) blue goes next, then orange and repeats
turns = {'deeppink': 'royalblue', 'royalblue': 'seagreen', 'seagreen': 'deeppink'}
state = {'player': 'deeppink', 'rows': [0] * 8}
```



```
# This function will respond when the user clicks the screen
# The function tells the computer where to "drop" the dot - in whichever row the user clicked on
def user_click(x, y):


  row = int((x + 200) // 50)
  rows = state['rows']
  count = rows[row]
  x = ((x + 200) // 50) * 50 - 200 + 25
  y = count * 50 - 200 + 25

  # The next few lines tell the turtle to go to the area where the player tapped
  turtle.up()
  turtle.goto(x, y)
  player = state['player']
  # The next line specifies the dot colour to be indicated by which player's turn it currently is
  dot(40, player)
  update()
  # These lines tell the program to go to the next player's turn (switching the dot colour)
  rows[row] = count + 1
  state['player'] = turns[player]

update()
```

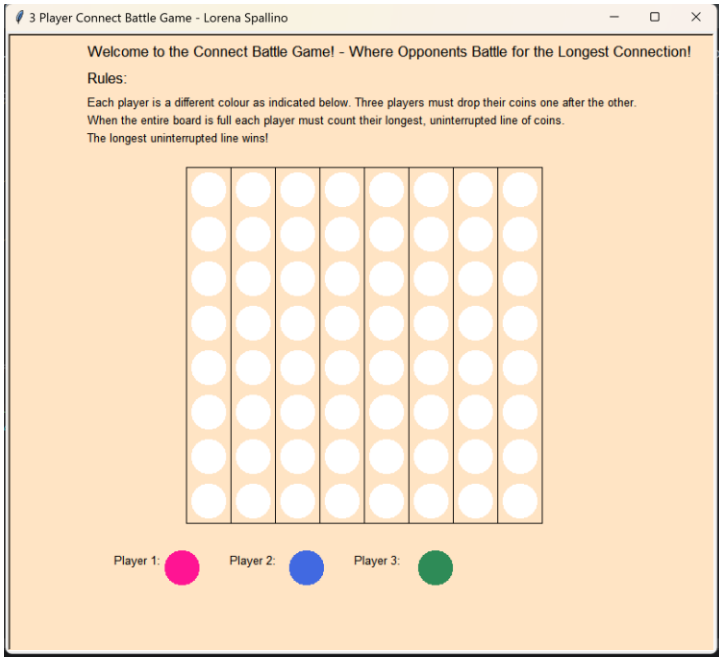
Then, the function increases in players, or turns meaning that the dot will change colour according to the dictionary above.

Finally, it is updated, and the game setup is complete. The bottom methods show the title of the game, the screen setup dimensions, and a few other functions specifying how the turtle module is to react.



```
# Ending by giving the title, specifying the screen setup hiding the turtle module, calling the grid function...

turtle.title("3 Player Connect Battle Game - Lorena Spallino")
setup( width: 800,  height: 700,  startx: 450,  starty: 0)
hideturtle()
tracer(False)
grid()

# Calling the 'user_click' function defined earlier to respond when the screen is clicked by the user.
onscreenclick(user_click)
#finishing when x is clicked, not as soon as the turtle is done drawing
turtle.mainloop()
```

Evidently, these lines of code are lengthy, and detailed. All is worthwhile when the code is run, and a   a beautiful screen displays the connect grid. Then when the user clicks on it and the screen responds as intended, the game is a success. Shown below are a few snapshots of the output.



3 Player Connect Battle Game - Lorena Spallino

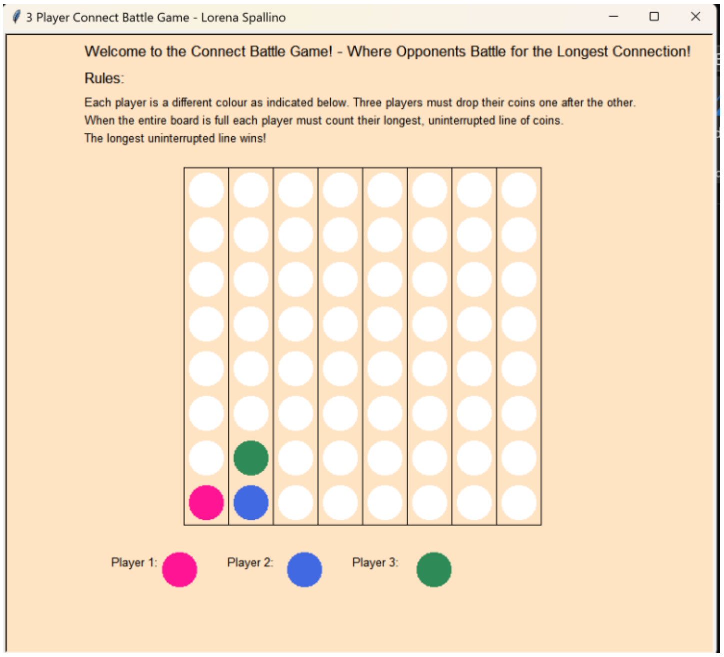
Welcome to the Connect Battle Game! - Where Opponents Battle for the Longest Connection!

Rules:

Each player is a different colour as indicated below. Three players must drop their coins one after the other.
When the entire board is full each player must count their longest, uninterrupted line of coins.
The longest uninterrupted line wins!

Player 1: Player 2: Player 3:

As shown, the output is exactly as expected. A beautiful screen setup with a grid drawn by the turtle module of the specified dimensions. The rules are clearly shown above the board and the players, and their colours are clearly indicated below. The white dots clearly show the available slots still on the board, since no one has begun to play yet, the board is empty.



3 Player Connect Battle Game - Lorena Spallino

Welcome to the Connect Battle Game! - Where Opponents Battle for the Longest Connection!

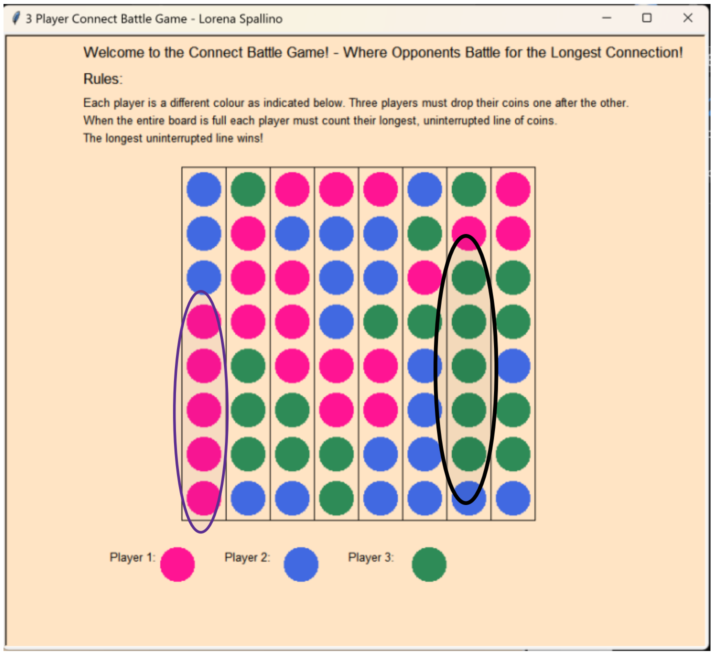
Rules:

Each player is a different colour as indicated below. Three players must drop their coins one after the other.
When the entire board is full each player must count their longest, uninterrupted line of coins.
The longest uninterrupted line wins!

Player 1: Player 2: Player 3:

This second screenshot shows the results after the first round of turns. Once each player has gone once, first pink, then blue then green. After this first round the turn will repeat, therefore the next player being pink, then blue again, then green.

Finally, when the board is entirely full the game is done. The players must count their longest uninterrupted lines, and the player or players with the longest line is/ are the victors! As shown, I have counted that the longest line is a streak of 5 coins, that means that both player one, pink, and player three, green, are the champions of the battle, and now the game is over!



3 Player Connect Battle Game - Lorena Spallino

Welcome to the Connect Battle Game! - Where Opponents Battle for the Longest Connection!

Rules:

Each player is a different colour as indicated below. Three players must drop their coins one after the other.
When the entire board is full each player must count their longest, uninterrupted line of coins.
The longest uninterrupted line wins!

Player 1: ● Player 2: ● Player 3: ●

To conclude, python has a variety of applications. Simple games are an excellent use of the knowledge, and skills that I have learned throughout this course. This three player connect game is an excellent representation of both a really fun and enjoyable game, but also some of the skills in python that I have learned in he past few weeks!

References

Aref, A. (2021, August 16). *Build Simple Python Games. Python developers never never get tired | by Ali Aref | Medium*. Ali Aref. Retrieved December 5, 2023, from https://aliarefwriorr.medium.com/build-simple-python-games-544eb68110d1

Developing a function to print a grid in python. (2020, March 25). Stack Overflow. Retrieved December 5, 2023, from https://stackoverflow.com/questions/60842728/developing-a-function-to-print-a-grid-in-python

Nguyen, Q. (n.d.-a). Hands-on application development with PyCharm: Accelerate your python applications using practical coding techniques in PyCharm. computer software.