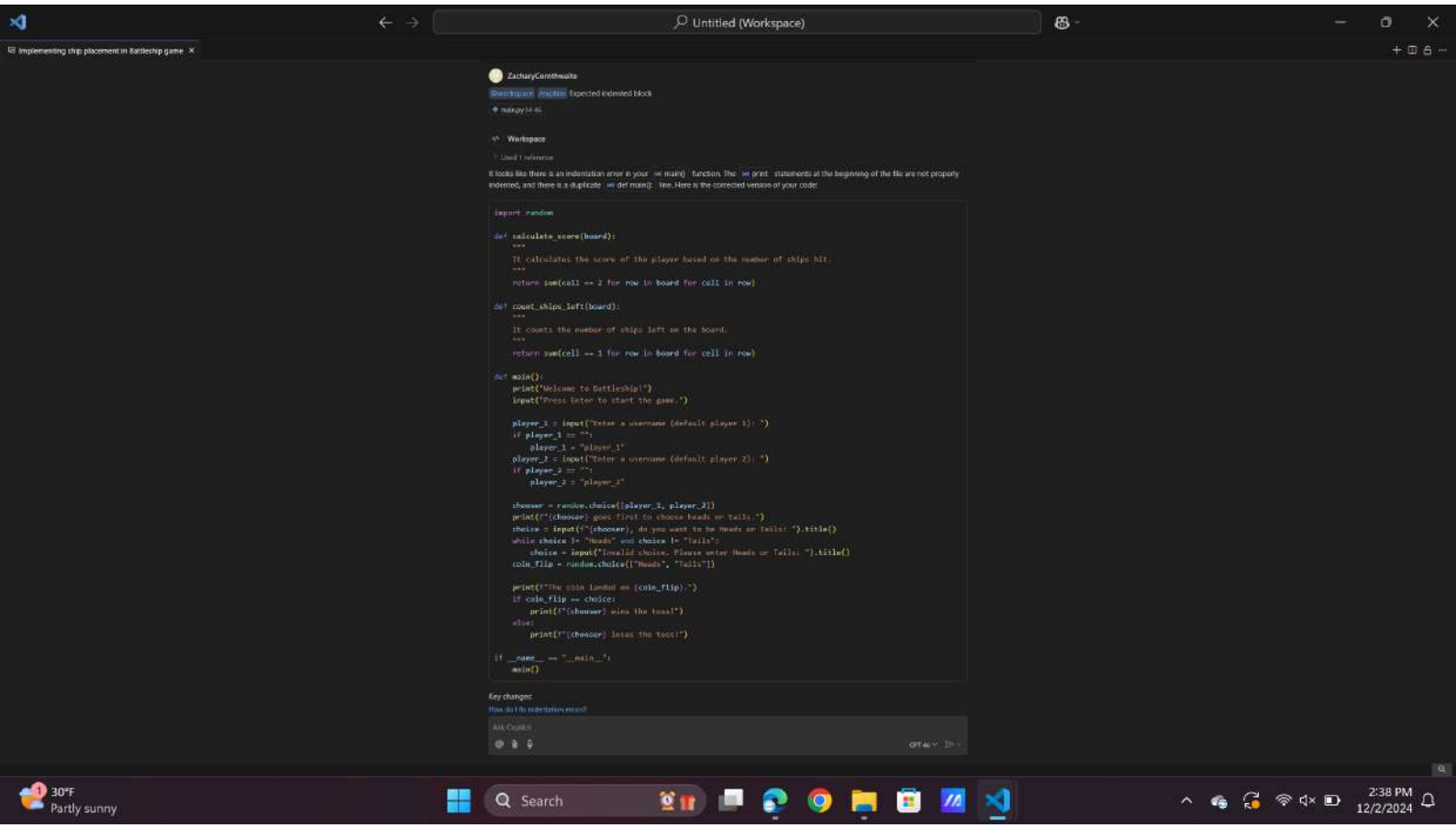
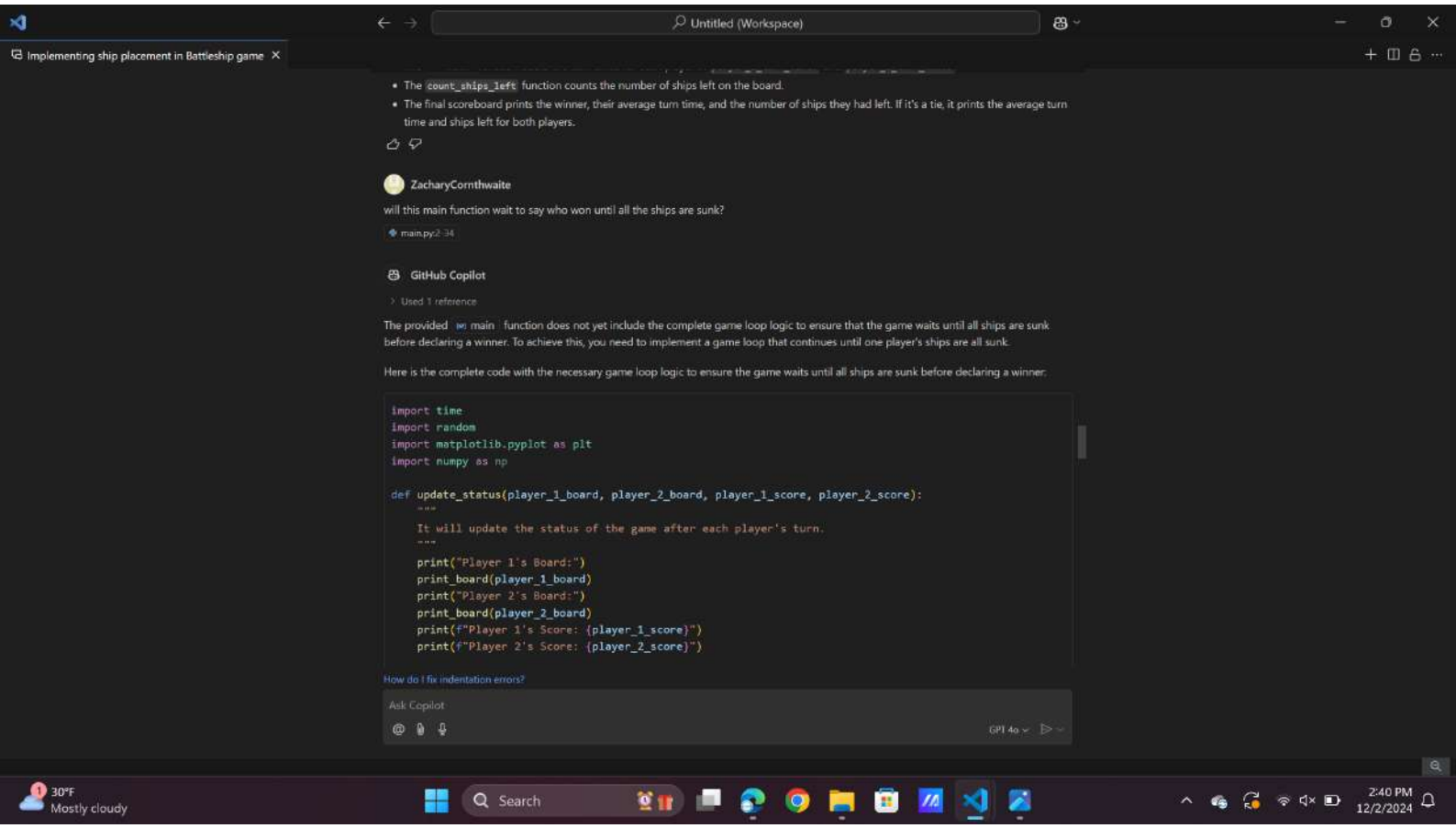
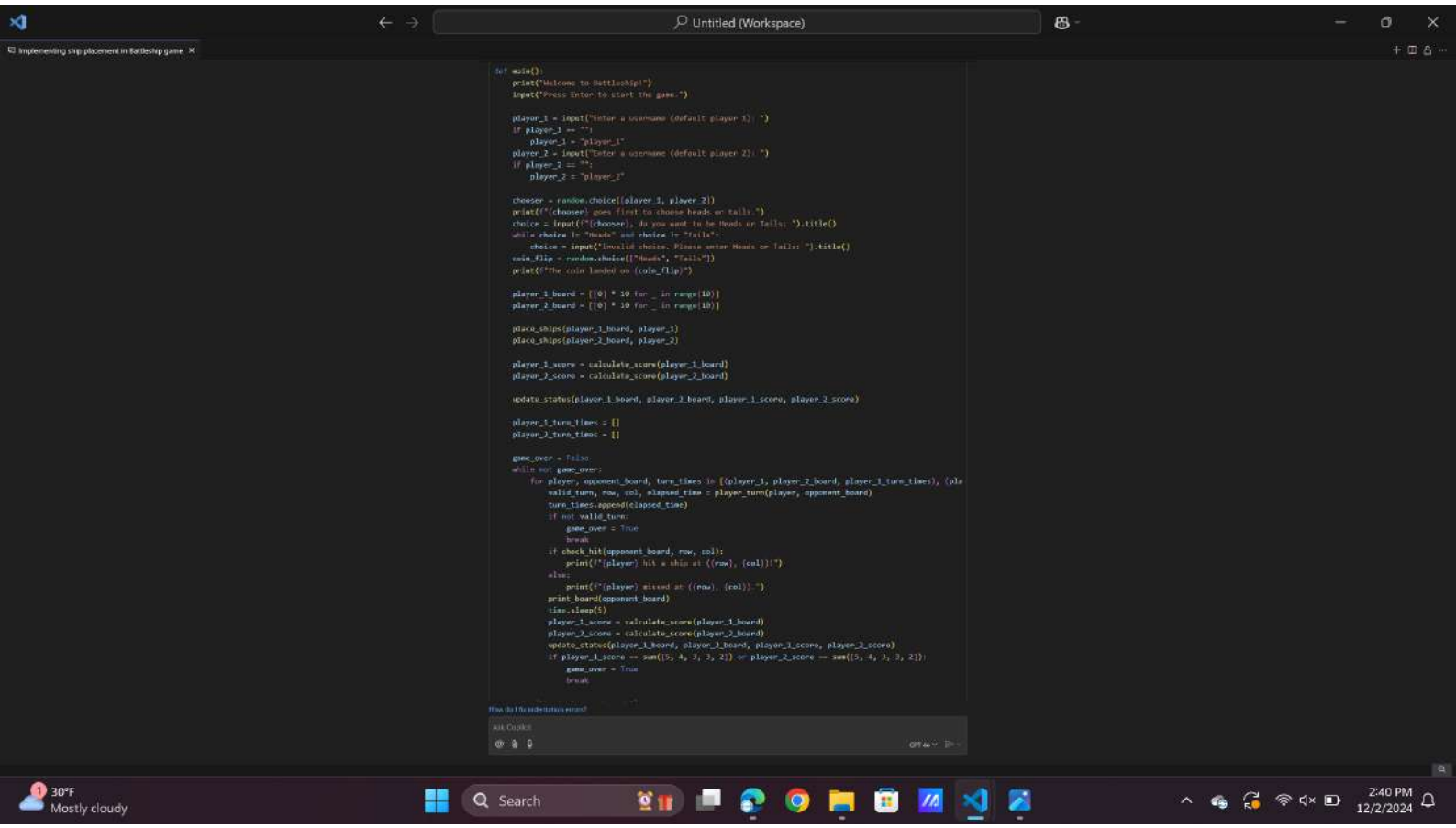
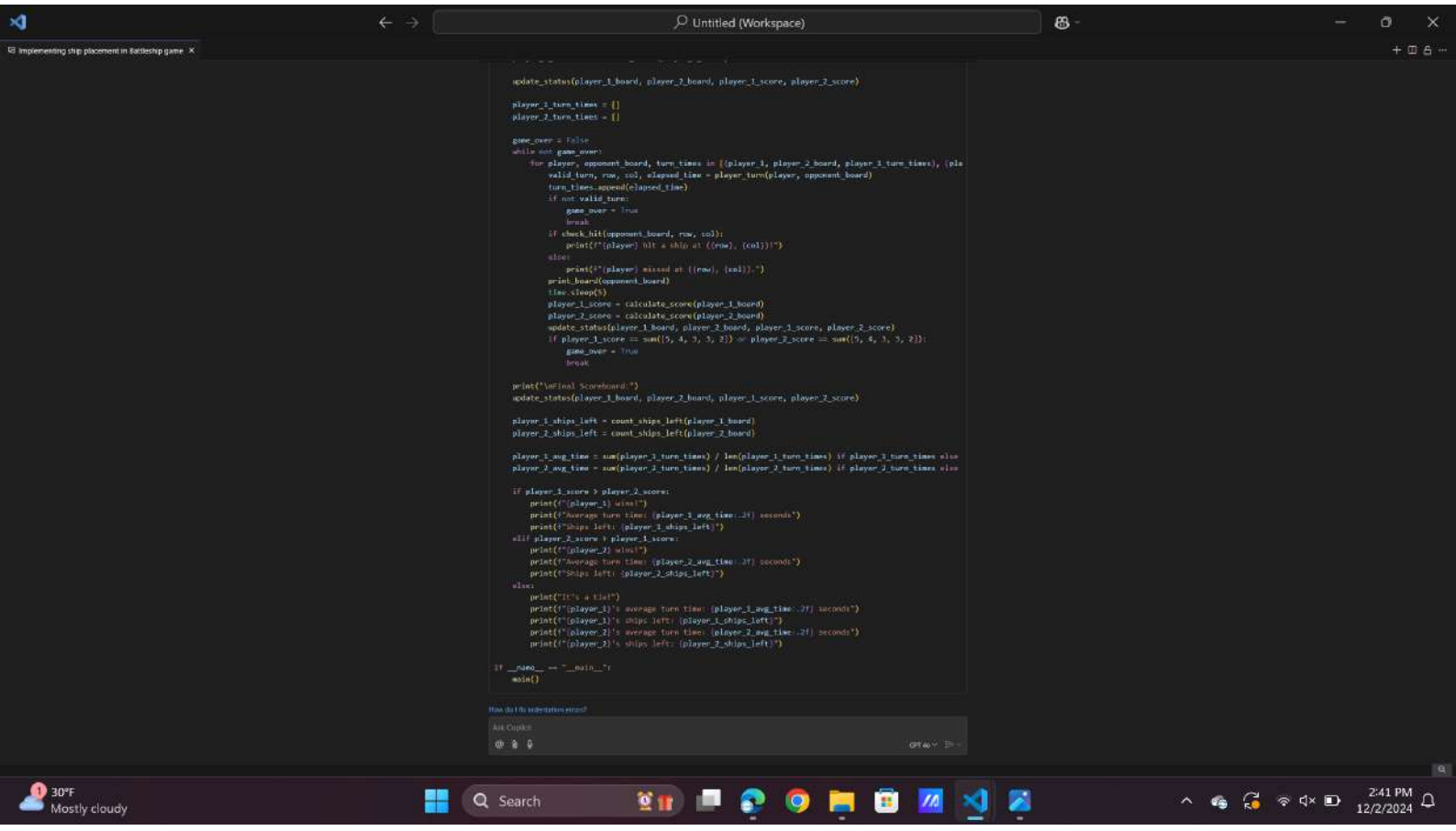
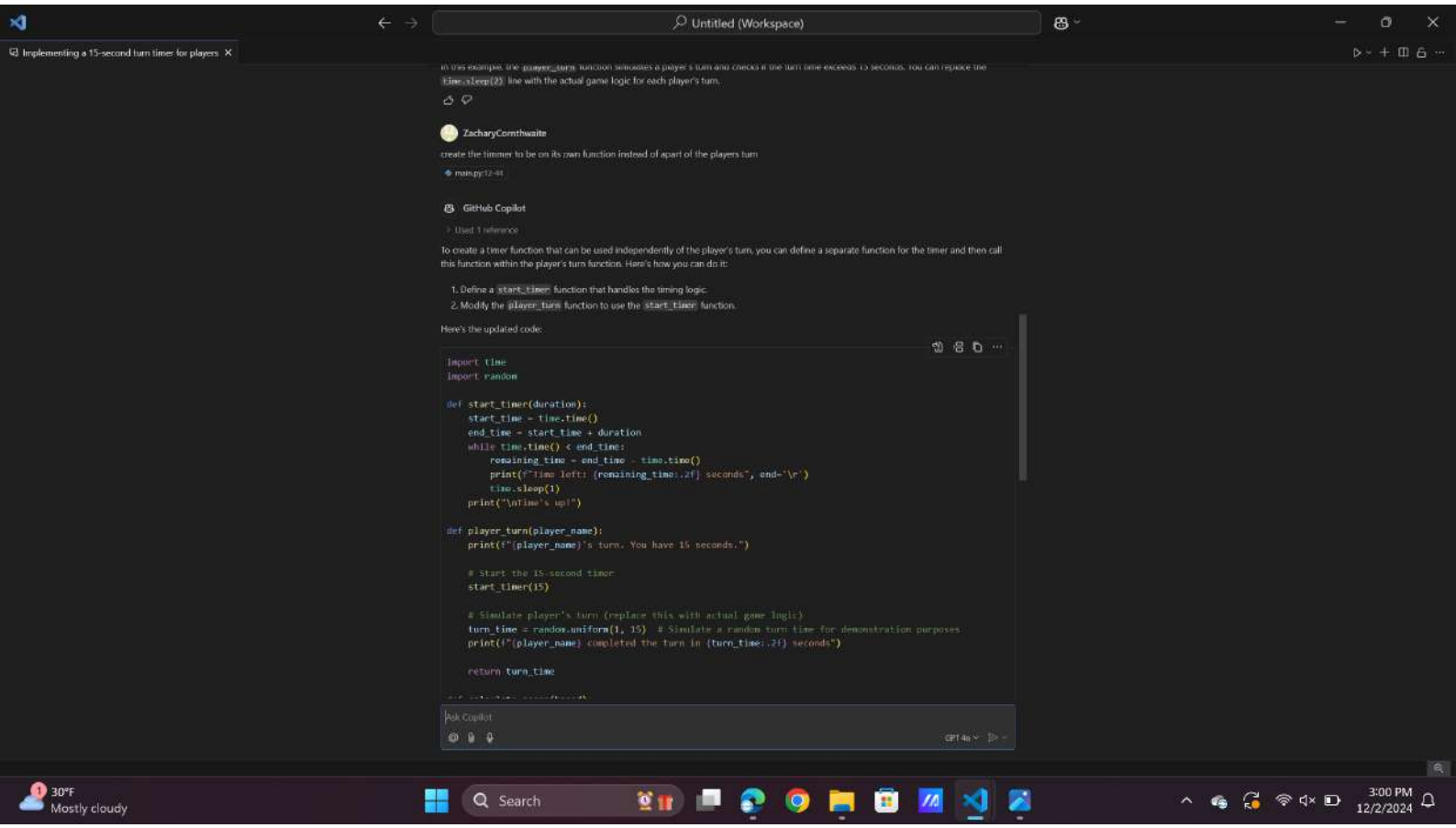
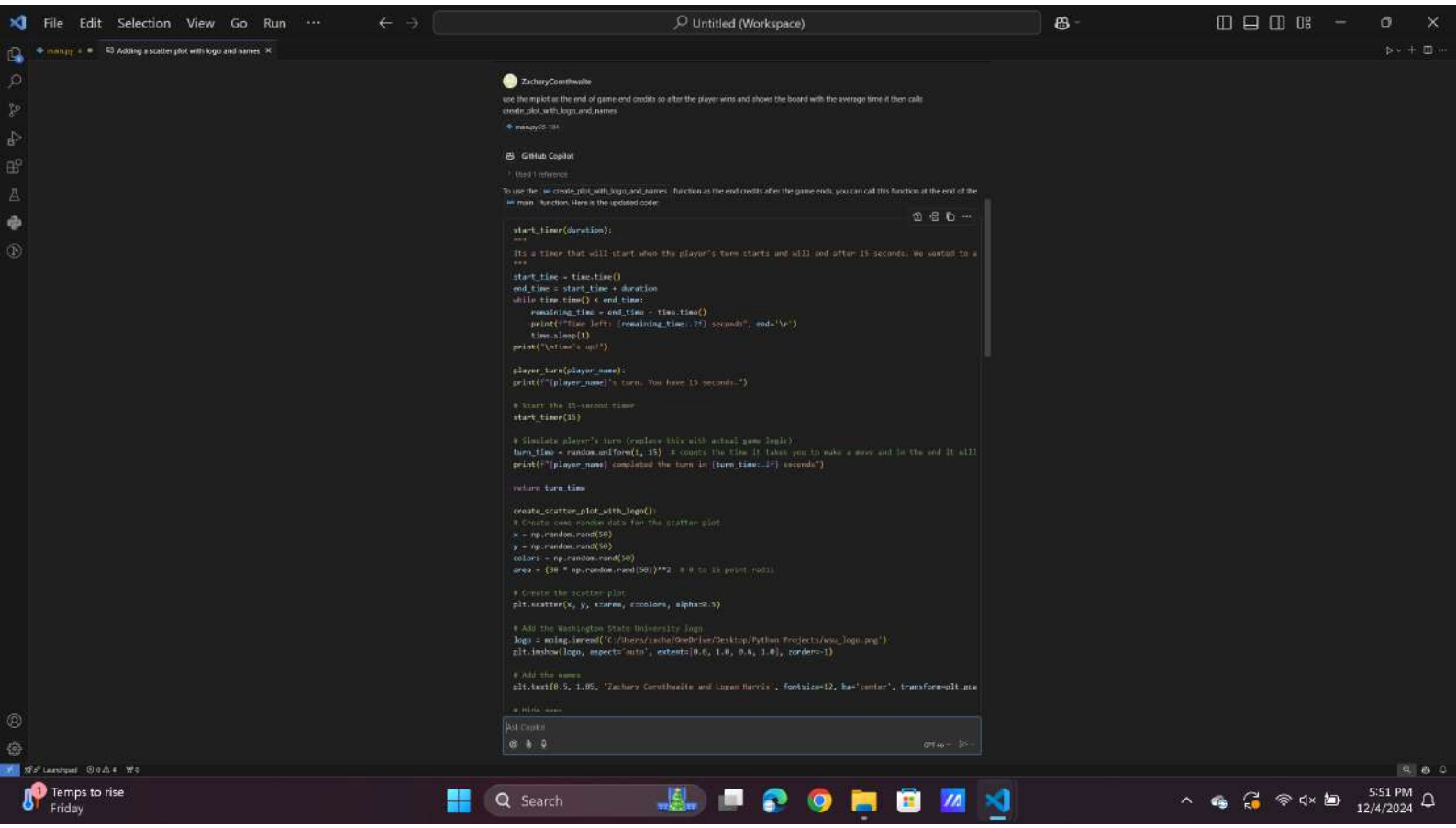

Untitled (Workspace)
Implementing ship placement in Battleship game
ZacharyCornthwaite
Workspace
Used 1 reference
It looks like there is an indentation error in your main() function. The print statements at the beginning of the file are not properly indented, and there is a duplicate def main(): line. Here is the corrected version of your code:
import random
def calculate_score(board):
"""
It calculates the score of the player based on the number of ships hit.
"""
return sum(cell == 1 for row in board for cell in row)
def count_ships_left(board):
"""
It counts the number of ships left on the board.
"""
return sum(cell == 1 for row in board for cell in row)
def main():
print("Welcome to Battleship!")
input("Press Enter to start the game.")
player_1 = input("Enter a username (default player 1): ")
if player_1 == "":
player_1 = "player_1"
player_2 = input("Enter a username (default player 2): ")
if player_2 == "":
player_2 = "player_2"
chooser = random.choice([player_1, player_2])
print(f"{chooser} goes first to choose heads or tails.")
choice = input(f"{chooser}, do you want to be Heads or Tails: ").title()
while choice != "Heads" and choice != "Tails":
choice = input("Invalid choice. Please enter Heads or Tails: ").title()
coin_flip = random.choice(["Heads", "Tails"])
print(f"The coin landed on {coin_flip}.")
if coin_flip == choice:
print(f"{chooser} wins the toss!")
else:
print(f"{chooser} loses the toss!")
if __name__ == "__main__":
main()
Key changes:
How do I fix indentation errors?
Ask Copilot
GPT-4o
30°F
Partly sunny
Search
2:38 PM
12/2/2024

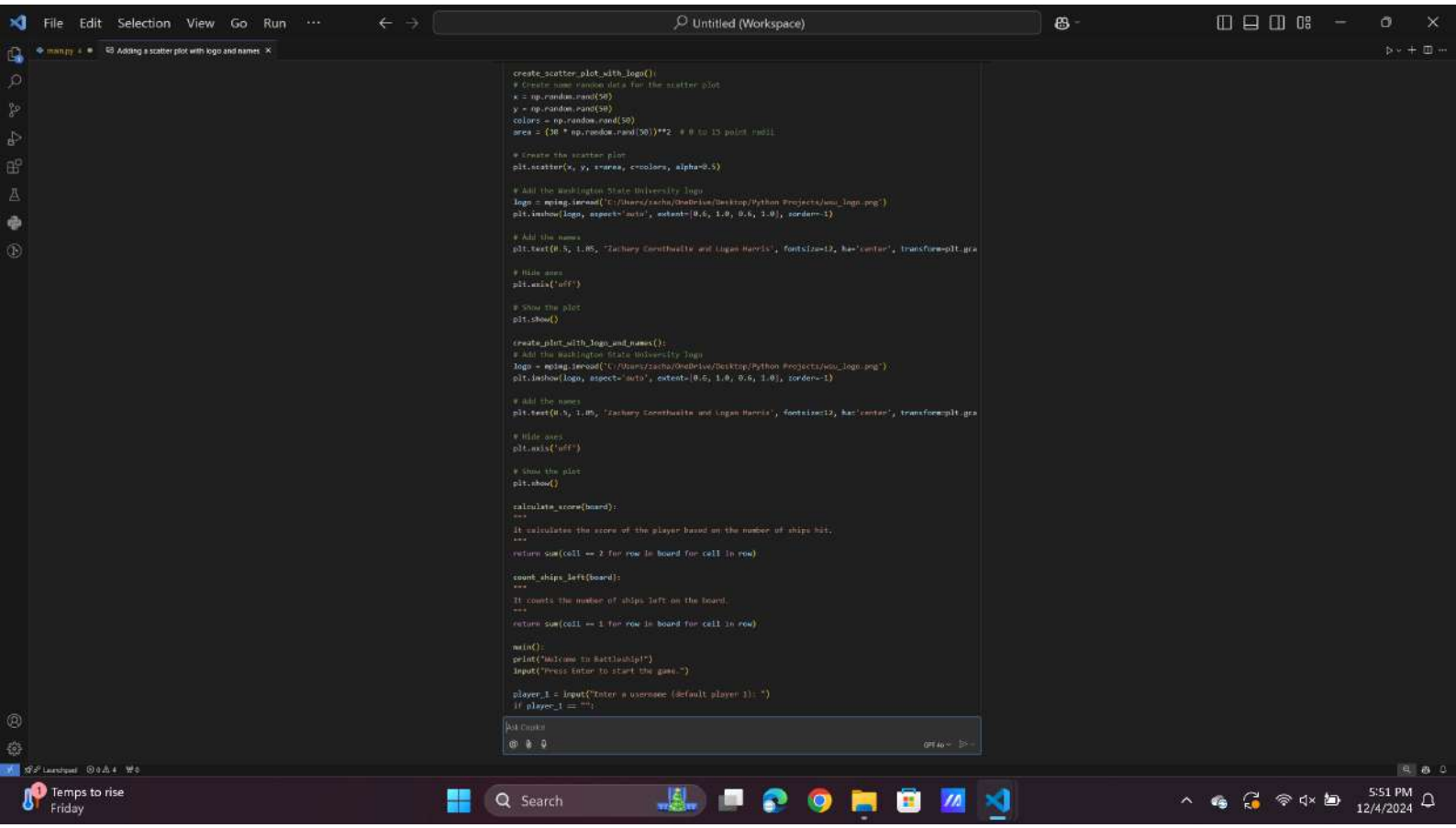

Untitled (Workspace)
Implementing ship placement in Battleship game
• The count_ships_left function counts the number of ships left on the board.
• The final scoreboard prints the winner, their average turn time, and the number of ships they had left. If it's a tie, it prints the average turn time and ships left for both players.
ZacharyCornthwaite
will this main function wait to say who won until all the ships are sunk?
main.py:2-34
GitHub Copilot
Used 1 reference
The provided main function does not yet include the complete game loop logic to ensure that the game waits until all ships are sunk before declaring a winner. To achieve this, you need to implement a game loop that continues until one player's ships are all sunk.
Here is the complete code with the necessary game loop logic to ensure the game waits until all ships are sunk before declaring a winner:
import time
import random
import matplotlib.pyplot as plt
import numpy as np
def update_status(player_1_board, player_2_board, player_1_score, player_2_score):
"""
It will update the status of the game after each player's turn.
"""
print("Player 1's Board:")
print_board(player_1_board)
print("Player 2's Board:")
print_board(player_2_board)
print(f"Player 1's Score: {player_1_score}")
print(f"Player 2's Score: {player_2_score}")
How do I fix indentation errors?
Ask Copilot
GPT 4o
30°F
Mostly cloudy
Search
2:40 PM
12/2/2024

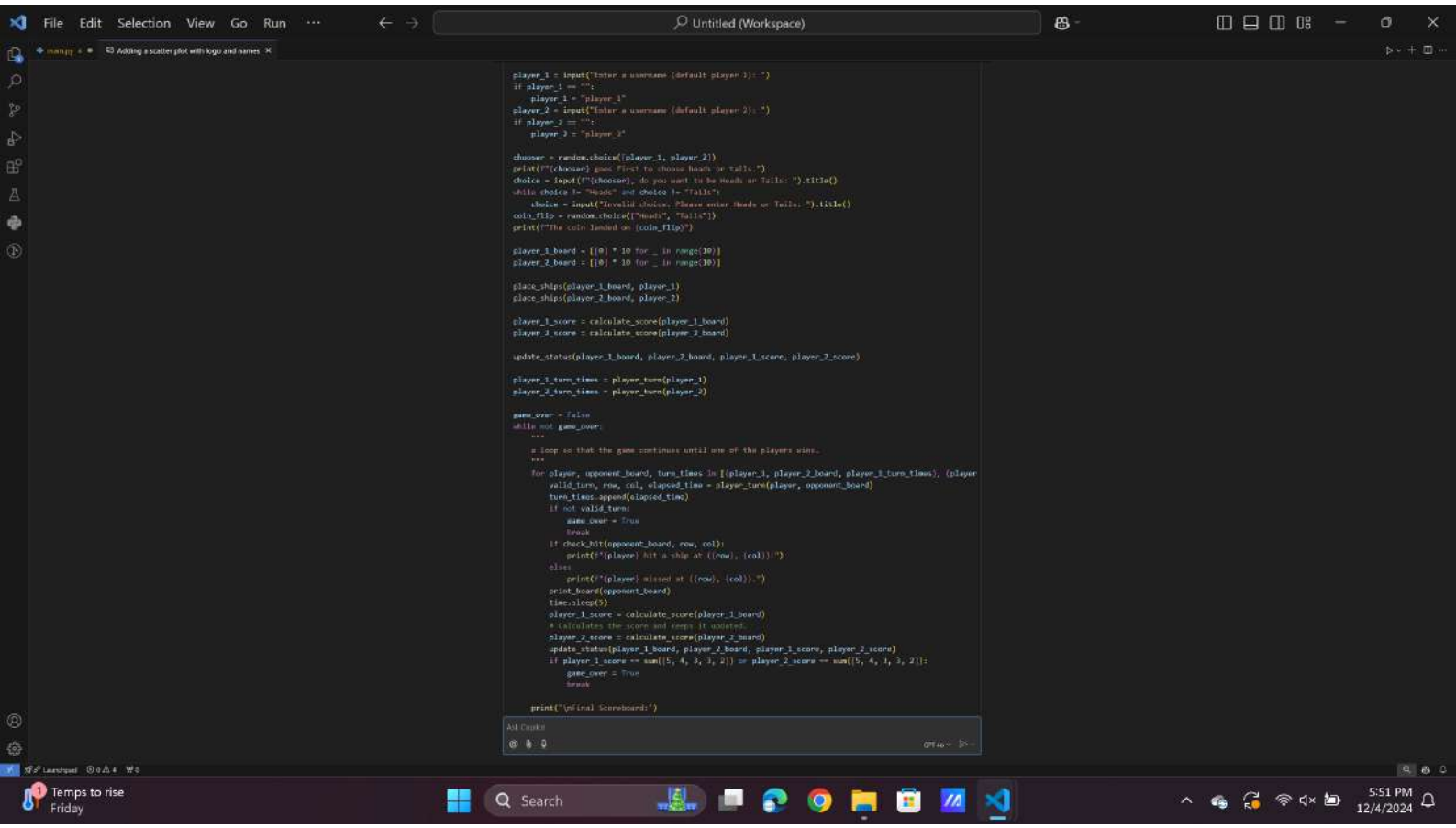

Untitled (Workspace)
Implementing ship placement in Battleship game
def main():
print("Welcome to Battleship!")
input("Press Enter to start the game.")
player_1 = input("Enter a username (default player 1): ")
if player_1 == "":
player_1 = "player_1"
player_2 = input("Enter a username (default player 2): ")
if player_2 == "":
player_2 = "player_2"
chooser = random.choice([player_1, player_2])
print(f"{chooser} goes first to choose heads or tails.")
choice = input(f"{chooser}, do you want to be Heads or Tails: ").title()
while choice != "Heads" and choice != "Tails":
choice = input("Invalid choice. Please enter Heads or Tails: ").title()
coin_flip = random.choice(["Heads", "Tails"])
print(f"The coin landed on {coin_flip}")
player_1_board = [[0] * 10 for _ in range(10)]
player_2_board = [[0] * 10 for _ in range(10)]
place_ships(player_1_board, player_1)
place_ships(player_2_board, player_2)
player_1_score = calculate_score(player_1_board)
player_2_score = calculate_score(player_2_board)
update_states(player_1_board, player_2_board, player_1_score, player_2_score)
player_1_turn_times = []
player_2_turn_times = []
game_over = False
while not game_over:
for player, opponent_board, turn_times in [(player_1, player_2_board, player_1_turn_times), (pla
valid_turn, row, col, elapsed_time = player_turn(player, opponent_board)
turn_times.append(elapsed_time)
if not valid_turn:
game_over = True
break
if check_hit(opponent_board, row, col):
print(f"{player} hit a ship at ({row}, {col})!")
else:
print(f"{player} missed at ({row}, {col}).")
print_board(opponent_board)
time.sleep(5)
player_1_score = calculate_score(player_1_board)
player_2_score = calculate_score(player_2_board)
update_states(player_1_board, player_2_board, player_1_score, player_2_score)
if player_1_score == sum([5, 4, 3, 3, 2]) or player_2_score == sum([5, 4, 3, 3, 2]):
game_over = True
break
How do I fix indentation errors?
Ask Copilot
GPT-4o
30°F
Mostly cloudy
Search
2:40 PM
12/2/2024

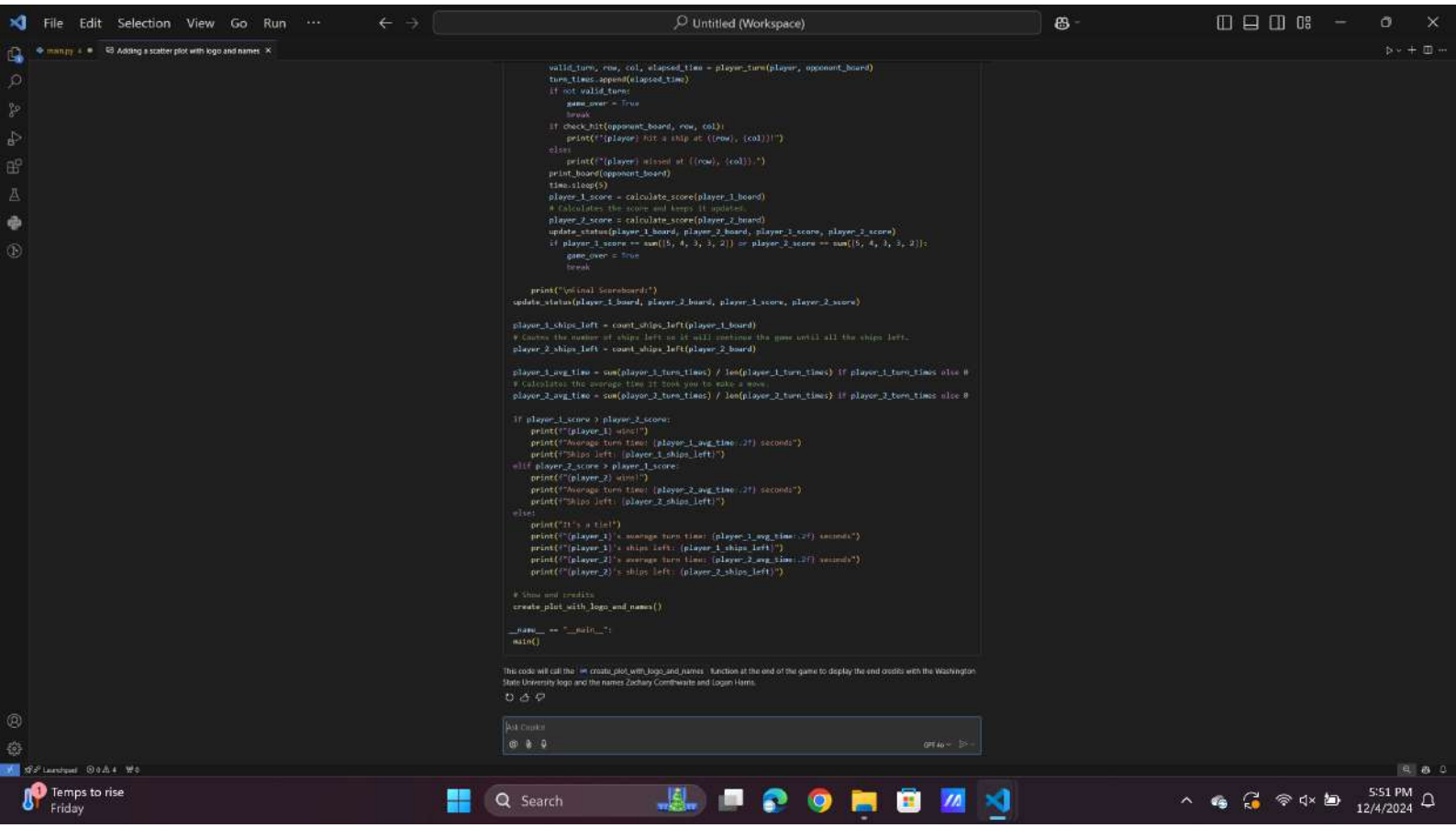

Untitled (Workspace)
Implementing ship placement in Battleship game
update_status(player_1_board, player_2_board, player_1_score, player_2_score)
player_1_turn_times = []
player_2_turn_times = []
game_over = False
while not game_over:
for player, opponent_board, turn_times in [(player_1, player_2_board, player_1_turn_times), (pla
valid_turn, row, col, elapsed_time = player_turn(player, opponent_board)
turn_times.append(elapsed_time)
if not valid_turn:
game_over = True
break
if check_hit(opponent_board, row, col):
print(f"{player} hit a ship at ({row}, {col})!")
else:
print(f"{player} missed at ({row}, {col}).")
print_board(opponent_board)
time.sleep(5)
player_1_score = calculate_score(player_1_board)
player_2_score = calculate_score(player_2_board)
update_status(player_1_board, player_2_board, player_1_score, player_2_score)
if player_1_score == sum([5, 4, 3, 3, 2]) or player_2_score == sum([5, 4, 3, 3, 2]):
game_over = True
break
print("\nFinal Scoreboard:")
update_status(player_1_board, player_2_board, player_1_score, player_2_score)
player_1_ships_left = count_ships_left(player_1_board)
player_2_ships_left = count_ships_left(player_2_board)
player_1_avg_time = sum(player_1_turn_times) / len(player_1_turn_times) if player_1_turn_times else
player_2_avg_time = sum(player_2_turn_times) / len(player_2_turn_times) if player_2_turn_times else
if player_1_score > player_2_score:
print(f"{player_1} wins!")
print(f"Average turn time: {player_1_avg_time:.2f} seconds")
print(f"Ships left: {player_1_ships_left}")
elif player_2_score > player_1_score:
print(f"{player_2} wins!")
print(f"Average turn time: {player_2_avg_time:.2f} seconds")
print(f"Ships left: {player_2_ships_left}")
else:
print("It's a tie!")
print(f"{player_1}'s average turn time: {player_1_avg_time:.2f} seconds")
print(f"{player_1}'s ships left: {player_1_ships_left}")
print(f"{player_2}'s average turn time: {player_2_avg_time:.2f} seconds")
print(f"{player_2}'s ships left: {player_2_ships_left}")
if __name__ == "__main__":
main()
How do I fix indentation errors?
Ask Copilot
GPT-4o
30°F
Mostly cloudy
Search
2:41 PM
12/2/2024

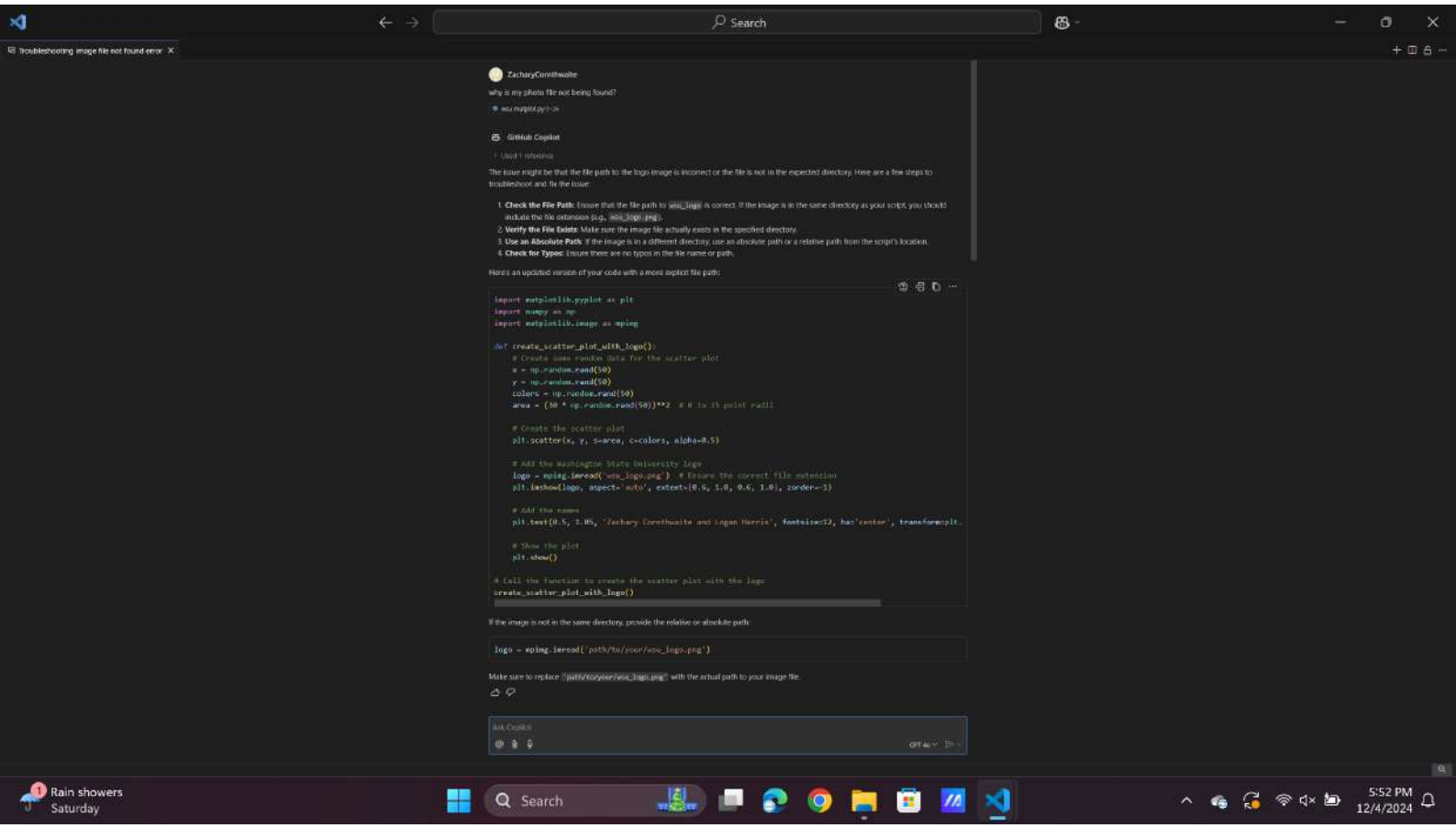

Untitled (Workspace)
Implementing a 15-second turn timer for players
time.sleep(2) line with the actual game logic for each player's turn.
ZacharyCornthwaite
create the timmer to be on its own function instead of apart of the players turn
main.py:12-44
GitHub Copilot
Used 1 reference
To create a timer function that can be used independently of the player's turn, you can define a separate function for the timer and then call this function within the player's turn function. Here's how you can do it:
1. Define a start_timer function that handles the timing logic.
2. Modify the player_turn function to use the start_timer function.
Here's the updated code:
import time
import random
def start_timer(duration):
start_time = time.time()
end_time = start_time + duration
while time.time() < end_time:
remaining_time = end_time - time.time()
print(f"Time left: {remaining_time:.2f} seconds", end="\r")
time.sleep(1)
print("\nTime's up!")
def player_turn(player_name):
print(f"{player_name}'s turn. You have 15 seconds.")
# Start the 15-second timer
start_timer(15)
# Simulate player's turn (replace this with actual game logic)
turn_time = random.uniform(1, 15) # Simulate a random turn time for demonstration purposes
print(f"{player_name} completed the turn in {turn_time:.2f} seconds")
return turn_time
Ask Copilot
GPT 4o
30°F
Mostly cloudy
Search
3:00 PM
12/2/2024

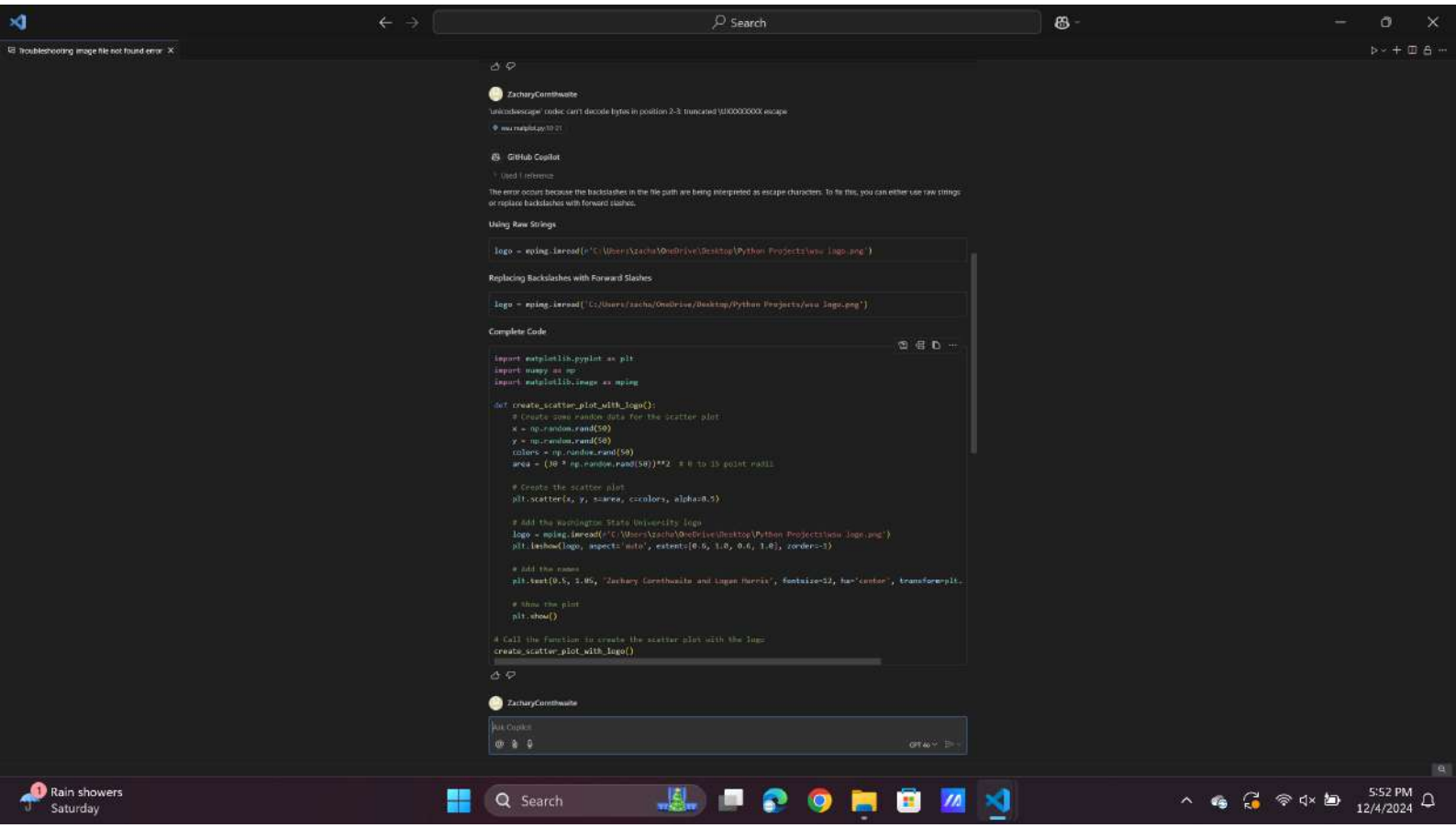

File Edit Selection View Go Run
Untitled (Workspace)
main.py
Adding a scatter plot with logo and names
ZacharyCornthwaite
use the mplot as the end of game end credits so after the player wins and shows the board with the average time it then calls create_plot_with_logo_and_names
main.py:55 - 184
GitHub Copilot
1 reference
To use the create_plot_with_logo_and_names function as the end credits after the game ends, you can call this function at the end of the main function. Here is the updated code:
start_timer(duration):
"""
Its a timer that will start when the player's turn starts and will end after 15 seconds. We wanted to a
"""
start_time = time.time()
end_time = start_time + duration
while time.time() < end_time:
remaining_time = end_time - time.time()
print(f"Time left: {remaining_time:.2f} seconds", end="\r")
time.sleep(1)
print("\nTime's up!")
player_turn(player_name):
print(f"{player_name}'s turn. You have 15 seconds.")
# Start the 15-second timer
start_timer(15)
# Simulate player's turn (replace this with actual game logic)
turn_time = random.uniform(1, 15) # counts the time it takes you to make a move and in the end it will
print(f"{player_name} completed the turn in {turn_time:.2f} seconds")
return turn_time
create_scatter_plot_with_logo():
# Create some random data for the scatter plot
x = np.random.rand(50)
y = np.random.rand(50)
colors = np.random.rand(50)
area = (30 * np.random.rand(50))**2 # 0 to 15 point radii
# Create the scatter plot
plt.scatter(x, y, s=area, c=colors, alpha=0.5)
# Add the Washington State University logo
logo = mpimg.imread('C:/Users/zache/OneDrive/Desktop/Python Projects/wsu_logo.png')
plt.imshow(logo, aspect='auto', extent=[0.6, 1.0, 0.6, 1.0], zorder=-1)
# Add the names
plt.text(0.5, 1.05, 'Zachary Cornthwaite and Logan Harris', fontsize=12, ha='center', transform=plt.gca
Hide axes
Ask Copilot
GPT-4o
Temps to rise
Friday
Search
5:51 PM
12/4/2024


File Edit Selection View Go Run
Untitled (Workspace)
main.py
Adding a scatter plot with logo and names
create_scatter_plot_with_logo():
# Create some random data for the scatter plot
x = np.random.rand(50)
y = np.random.rand(50)
colors = np.random.rand(50)
area = (30 * np.random.rand(50))**2 # 0 to 15 point radii
# Create the scatter plot
plt.scatter(x, y, s=area, c=colors, alpha=0.5)
# Add the Washington State University logo
logo = mpimg.imread('C:/Users/zacha/OneDrive/Desktop/Python Projects/wsu_logo.png')
plt.imshow(logo, aspect='auto', extent=[0.6, 1.0, 0.6, 1.0], zorder=-1)
# Add the names
plt.text(0.5, 1.05, 'Zachary Corathwaite and Logan Harris', fontsize=12, ha='center', transform=plt.gca
# Hide axes
plt.axis('off')
# Show the plot
plt.show()
create_plot_with_logo_and_names():
# Add the Washington State University logo
logo = mpimg.imread('C:/Users/zacha/OneDrive/Desktop/Python Projects/wsu_logo.png')
plt.imshow(logo, aspect='auto', extent=[0.6, 1.0, 0.6, 1.0], zorder=-1)
# Add the names
plt.text(0.5, 1.05, 'Zachary Corathwaite and Logan Harris', fontsize=12, ha='center', transform=plt.gca
# Hide axes
plt.axis('off')
# Show the plot
plt.show()
calculate_score(board):
"""
It calculates the score of the player based on the number of ships hit.
"""
return sum(cell == 2 for row in board for cell in row)
count_ships_left(board):
"""
It counts the number of ships left on the board.
"""
return sum(cell == 1 for row in board for cell in row)
main():
print("Welcome to Battleship!")
input("Press Enter to start the game.")
player_1 = input("Enter a username (default player 1): ")
if player_1 == "":
Ask Copilot
GPT-4o
Launchpad
Temps to rise
Friday
Search
5:51 PM
12/4/2024


File Edit Selection View Go Run ···
Untitled (Workspace)
main.py
Adding a scatter plot with logo and names
player_1 = input("Enter a username (default player 1): ")
if player_1 == "":
player_1 = "player_1"
player_2 = input("Enter a username (default player 2): ")
if player_2 == "":
player_2 = "player_2"
chooser = random.choice([player_1, player_2])
print(f"{chooser} goes first to choose Heads or Tails.")
choice = input(f"{chooser}, do you want to be Heads or Tails: ").title()
while choice != "Heads" and choice != "Tails":
choice = input("Invalid choice. Please enter Heads or Tails: ").title()
coin_flip = random.choice(["Heads", "Tails"])
print(f"The coin landed on {coin_flip}")
player_1_board = [[0] * 10 for _ in range(10)]
player_2_board = [[0] * 10 for _ in range(10)]
place_ships(player_1_board, player_1)
place_ships(player_2_board, player_2)
player_1_score = calculate_score(player_1_board)
player_2_score = calculate_score(player_2_board)
update_status(player_1_board, player_2_board, player_1_score, player_2_score)
player_1_turn_times = player_turn(player_1)
player_2_turn_times = player_turn(player_2)
game_over = False
while not game_over:
"""
a loop so that the game continues until one of the players wins.
"""
for player, opponent_board, turn_times in [(player_1, player_2_board, player_1_turn_times), (player
valid_turn, row, col, elapsed_time = player_turn(player, opponent_board)
turn_times.append(elapsed_time)
if not valid_turn:
game_over = True
break
if check_hit(opponent_board, row, col):
print(f"{player} hit a ship at ({row}, {col})!")
else:
print(f"{player} missed at ({row}, {col}).")
print_board(opponent_board)
time.sleep(3)
player_1_score = calculate_score(player_1_board)
# Calculates the score and keeps it updated.
player_2_score = calculate_score(player_2_board)
update_status(player_1_board, player_2_board, player_1_score, player_2_score)
if player_1_score == sum([5, 4, 3, 3, 2]) or player_2_score == sum([5, 4, 3, 3, 2]):
game_over = True
break
print("\nFinal Scoreboard:")
Ask Copilot
GPT-4o
Temps to rise
Friday
Search
5:51 PM
12/4/2024


File Edit Selection View Go Run ···
Untitled (Workspace)
main.py
Adding a scatter plot with logo and names
valid_turn, row, col, elapsed_time = player_turn(player, opponent_board)
turn_times.append(elapsed_time)
if not valid_turn:
game_over = True
break
if check_hit(opponent_board, row, col):
print(f"{player} hit a ship at ({row}, {col})!")
else:
print(f"{player} missed at ({row}, {col}).")
print_board(opponent_board)
time.sleep(5)
player_1_score = calculate_score(player_1_board)
# Calculates the score and keeps it updated.
player_2_score = calculate_score(player_2_board)
update_status(player_1_board, player_2_board, player_1_score, player_2_score)
if player_1_score == sum([5, 4, 3, 3, 2]) or player_2_score == sum([5, 4, 3, 3, 2]):
game_over = True
break
print("\nFinal Scoreboard:")
update_status(player_1_board, player_2_board, player_1_score, player_2_score)
player_1_ships_left = count_ships_left(player_1_board)
# Counts the number of ships left so it will continue the game until all the ships left.
player_2_ships_left = count_ships_left(player_2_board)
player_1_avg_time = sum(player_1_turn_times) / len(player_1_turn_times) if player_1_turn_times else 0
# Calculates the average time it took you to make a move.
player_2_avg_time = sum(player_2_turn_times) / len(player_2_turn_times) if player_2_turn_times else 0
if player_1_score > player_2_score:
print(f"{player_1} wins!")
print(f"Average turn time: {player_1_avg_time:.2f} seconds")
print(f"Ships left: {player_1_ships_left}")
elif player_2_score > player_1_score:
print(f"{player_2} wins!")
print(f"Average turn time: {player_2_avg_time:.2f} seconds")
print(f"Ships left: {player_2_ships_left}")
else:
print("It's a tie!")
print(f"{player_1}'s average turn time: {player_1_avg_time:.2f} seconds")
print(f"{player_1}'s ships left: {player_1_ships_left}")
print(f"{player_2}'s average turn time: {player_2_avg_time:.2f} seconds")
print(f"{player_2}'s ships left: {player_2_ships_left}")
# Show end credits
create_plot_with_logo_and_names()
__name__ == "__main__":
main()
This code will call the create_plot_with_logo_and_names function at the end of the game to display the end credits with the Washington State University logo and the names Zachary Cornthwaite and Logan Harris.
Ask Copilot
GPT-4o
Temps to rise
Friday
Search
5:51 PM
12/4/2024


Search
Troubleshooting image file not found error
ZacharyCornthwaite
why is my photo file not being found?
#file:matplot.py:1-26
GitHub Copilot
1 Used 1 reference
The issue might be that the file path to the logo image is incorrect or the file is not in the expected directory. Here are a few steps to troubleshoot and fix the issue:
1. Check the File Path: Ensure that the file path to wsu_logo is correct. If the image is in the same directory as your script, you should include the file extension (e.g., wsu_logo.png).
2. Verify the File Exists: Make sure the image file actually exists in the specified directory.
3. Use an Absolute Path: If the image is in a different directory, use an absolute path or a relative path from the script's location.
4. Check for Typos: Ensure there are no typos in the file name or path.
Here's an updated version of your code with a more explicit file path:
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.image as mpimg
def create_scatter_plot_with_logo():
# Create some random data for the scatter plot
x = np.random.rand(50)
y = np.random.rand(50)
colors = np.random.rand(50)
area = (30 * np.random.rand(50))**2 # 0 to 15 point radii
# Create the scatter plot
plt.scatter(x, y, s=area, c=colors, alpha=0.5)
# Add the Washington State University logo
logo = mpimg.imread('wsu_logo.png') # Ensure the correct file extension
plt.imshow(logo, aspect='auto', extent=[0.6, 1.0, 0.6, 1.0], zorder=-1)
# Add the names
plt.text(0.5, 1.05, 'Zachary Cornthwaite and Logan Harris', fontsize=12, ha='center', transform=plt.
# Show the plot
plt.show()
# Call the function to create the scatter plot with the logo
create_scatter_plot_with_logo()
If the image is not in the same directory, provide the relative or absolute path:
logo = mpimg.imread('path/to/your/wsu_logo.png')
Make sure to replace 'path/to/your/wsu_logo.png' with the actual path to your image file.
Ask Copilot
GPT-4o
Rain showers
Saturday
Search
5:52 PM
12/4/2024


Search
Troubleshooting image file not found error
ZacharyCornthwaite
'unicodeescape' codec can't decode bytes in position 2-3: truncated \UXXXXXXXX escape
wsu matplot.py 10 21
GitHub Copilot
1 Used 1 reference
The error occurs because the backslashes in the file path are being interpreted as escape characters. To fix this, you can either use raw strings or replace backslashes with forward slashes.
Using Raw Strings
logo = mpimg.imread(r'C:\Users\zacha\OneDrive\Desktop\Python Projects\wsu logo.png')
Replacing Backslashes with Forward Slashes
logo = mpimg.imread('C:/Users/zacha/OneDrive/Desktop/Python Projects/wsu logo.png')
Complete Code
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.image as mpimg
def create_scatter_plot_with_logo():
# Create some random data for the scatter plot
x = np.random.rand(50)
y = np.random.rand(50)
colors = np.random.rand(50)
area = (30 * np.random.rand(50))**2 # 0 to 15 point radii
# Create the scatter plot
plt.scatter(x, y, s=area, c=colors, alpha=0.5)
# Add the Washington State University logo
logo = mpimg.imread(r'C:\Users\zacha\OneDrive\Desktop\Python Projects\wsu logo.png')
plt.imshow(logo, aspect='auto', extent=[0.6, 1.0, 0.6, 1.0], zorder=-1)
# Add the names
plt.text(0.5, 1.05, 'Zachary Cornthwaite and Logan Harris', fontsize=12, ha='center', transform=plt.
# Show the plot
plt.show()
# Call the function to create the scatter plot with the logo
create_scatter_plot_with_logo()
ZacharyCornthwaite
Ask Copilot
GPT-4o
Rain showers
Saturday
Search
5:52 PM
12/4/2024

Troubleshooting image file not found error

ZacharyCornthwaite

how can i take the graph part of for the matplot to where its just the names and the photo

wsu matplot.py:1-26

GitHub Copilot

Used 1 reference

To display only the names and the photo without the scatter plot, you can remove the code related to creating the scatter plot. Here's the updated function:

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

def create_plot_with_logo_and_names():
    # Add the Washington State University logo
    logo = mpimg.imread('C:/Users/zacha/OneDrive/Desktop/Python Projects/wsu_logo.png')
    plt.imshow(logo, aspect='auto', extent=[0.6, 1.0, 0.6, 1.0], zorder=-1)

    # Add the names
    plt.text(0.5, 1.05, 'Zachary Cornthwaite and Logan Harris', fontsize=12, ha='center', transform=plt.

    # Hide axes
    plt.axis('off')

    # Show the plot
    plt.show()

# Call the function to create the plot with the logo and names
create_plot_with_logo_and_names()
```

This code will display only the logo and the names without the scatter plot.

ZacharyCornthwaite

why are there color dots on the photo?

wsu matplot.py:1-12

Ask Copilot

GPT 4o

Search

Rain showers
Saturday

5:52 PM
12/4/2024

Search

Troubleshooting image file not found error

This code will display only the logo and the names without the scatter plot.

ZacharyCornthwaite

why are there color dots on the photo?

wsu matplot.py:1-12

GitHub Copilot

1 Used 1 reference

The color dots are appearing on the photo because the scatter plot is being drawn on the same figure as the image. To remove the scatter plot and only display the image and text, you need to remove the scatter plot code.

Here's the updated function to display only the image and the text:

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

def create_plot_with_logo_and_names():
    # Add the Washington State University logo
    logo = mpimg.imread('C:/Users/zacha/OneDrive/Desktop/Python Projects/wsu_logo.png')
    plt.imshow(logo, aspect='auto', extent=[0.6, 1.0, 0.6, 1.0], zorder=-1)

    # Add the names
    plt.text(0.5, 1.05, 'Zachary Cornthwaite and Logan Harris', fontsize=12, ha='center', transform=plt.

    # Hide axes
    plt.axis('off')

    # Show the plot
    plt.show()

# Call the function to create the plot with the logo and names
create_plot_with_logo_and_names()
```

This code will display only the Washington State University logo and the names without any scatter plot dots.

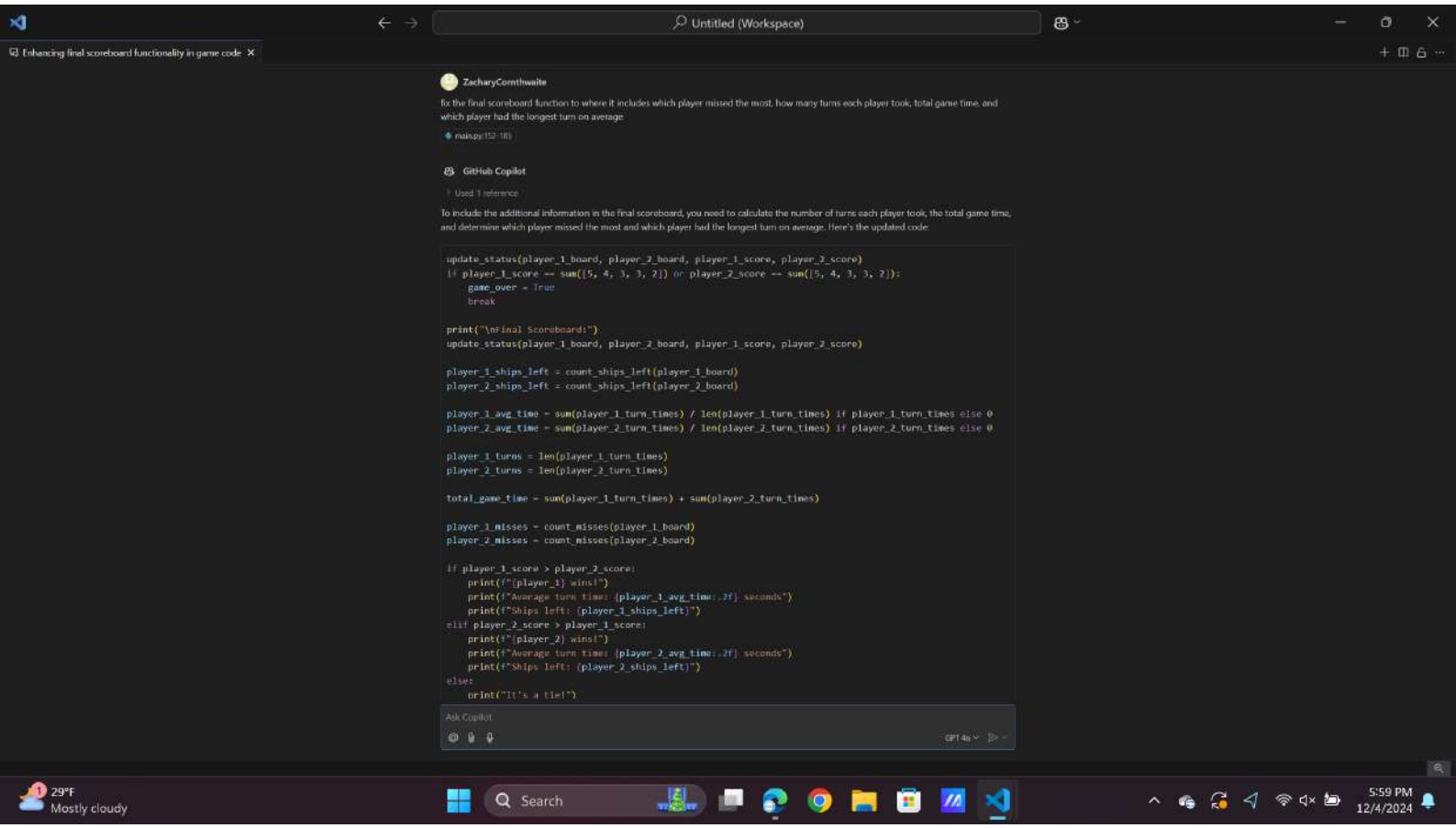
Ask Copilot

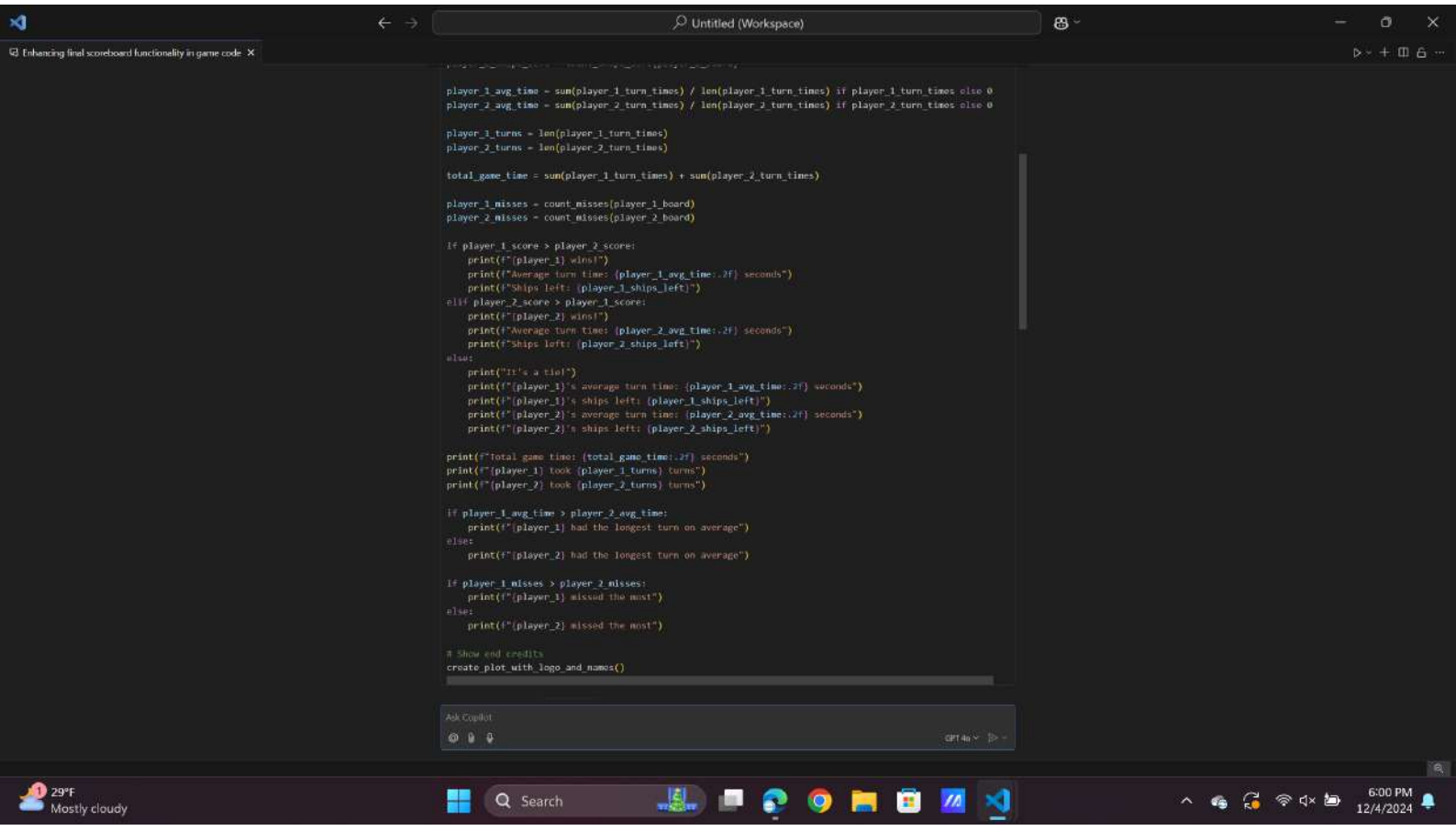
GPT 4o

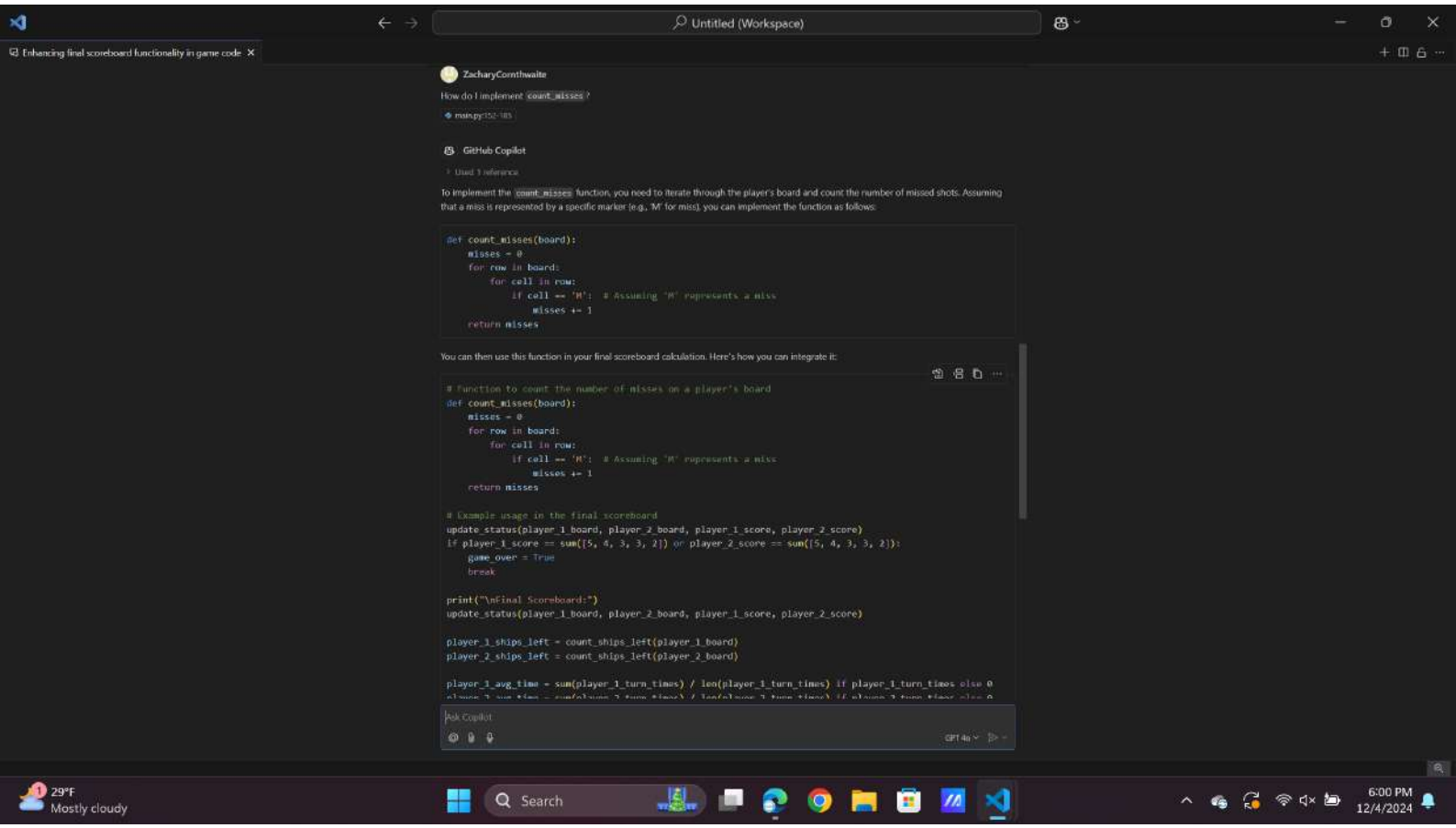
28°F
Mostly cloudy

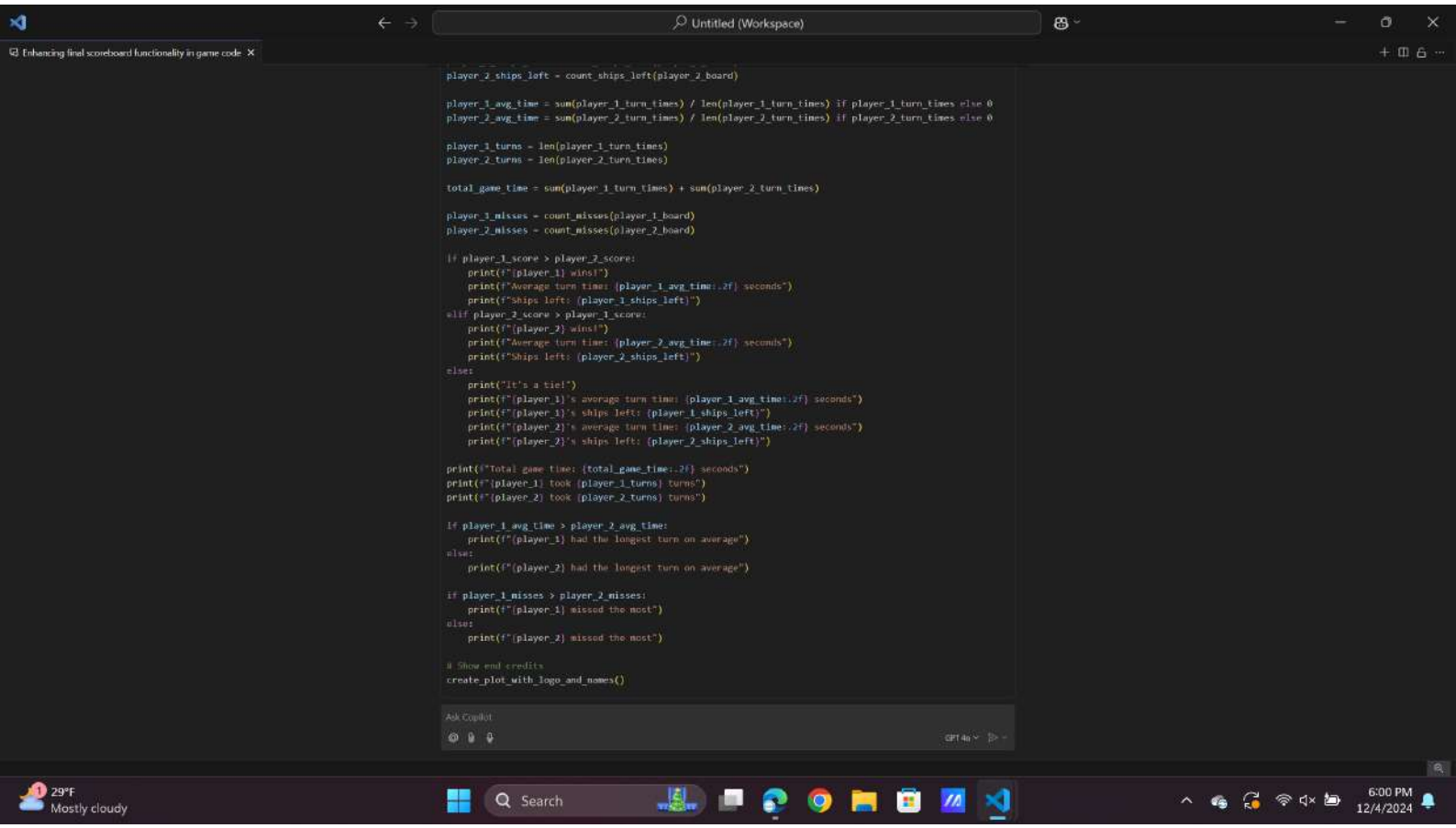
Search

5:52 PM
12/4/2024


Untitled (Workspace)
Enhancing final scoreboard functionality in game code
ZacharyCornthwaite
fix the final scoreboard function to where it includes which player missed the most, how many turns each player took, total game time, and which player had the longest turn on average
main.py:152-181
GitHub Copilot
Used 1 reference
To include the additional information in the final scoreboard, you need to calculate the number of turns each player took, the total game time, and determine which player missed the most and which player had the longest turn on average. Here's the updated code:
update_status(player_1_board, player_2_board, player_1_score, player_2_score)
if player_1_score == sum([5, 4, 3, 3, 2]) or player_2_score == sum([5, 4, 3, 3, 2]):
game_over = True
break
print("\nFinal Scoreboard:")
update_status(player_1_board, player_2_board, player_1_score, player_2_score)
player_1_ships_left = count_ships_left(player_1_board)
player_2_ships_left = count_ships_left(player_2_board)
player_1_avg_time = sum(player_1_turn_times) / len(player_1_turn_times) if player_1_turn_times else 0
player_2_avg_time = sum(player_2_turn_times) / len(player_2_turn_times) if player_2_turn_times else 0
player_1_turns = len(player_1_turn_times)
player_2_turns = len(player_2_turn_times)
total_game_time = sum(player_1_turn_times) + sum(player_2_turn_times)
player_1_misses = count_misses(player_1_board)
player_2_misses = count_misses(player_2_board)
if player_1_score > player_2_score:
print(f"{player_1} wins!")
print(f"Average turn time: {player_1_avg_time:.2f} seconds")
print(f"Ships left: {player_1_ships_left}")
elif player_2_score > player_1_score:
print(f"{player_2} wins!")
print(f"Average turn time: {player_2_avg_time:.2f} seconds")
print(f"Ships left: {player_2_ships_left}")
else:
print("It's a tie!")
Ask Copilot
GPT 4o
29°F
Mostly cloudy
Search
5:59 PM
12/4/2024


Untitled (Workspace)
Enhancing final scoreboard functionality in game code
player_1_avg_time = sum(player_1_turn_times) / len(player_1_turn_times) if player_1_turn_times else 0
player_2_avg_time = sum(player_2_turn_times) / len(player_2_turn_times) if player_2_turn_times else 0
player_1_turns = len(player_1_turn_times)
player_2_turns = len(player_2_turn_times)
total_game_time = sum(player_1_turn_times) + sum(player_2_turn_times)
player_1_misses = count_misses(player_1_board)
player_2_misses = count_misses(player_2_board)
if player_1_score > player_2_score:
print(f"{player_1} wins!")
print(f"Average turn time: {player_1_avg_time:.2f} seconds")
print(f"Ships left: {player_1_ships_left}")
elif player_2_score > player_1_score:
print(f"{player_2} wins!")
print(f"Average turn time: {player_2_avg_time:.2f} seconds")
print(f"Ships left: {player_2_ships_left}")
else:
print("It's a tie!")
print(f"{player_1}'s average turn time: {player_1_avg_time:.2f} seconds")
print(f"{player_1}'s ships left: {player_1_ships_left}")
print(f"{player_2}'s average turn time: {player_2_avg_time:.2f} seconds")
print(f"{player_2}'s ships left: {player_2_ships_left}")
print(f"Total game time: {total_game_time:.2f} seconds")
print(f"{player_1} took {player_1_turns} turns")
print(f"{player_2} took {player_2_turns} turns")
if player_1_avg_time > player_2_avg_time:
print(f"{player_1} had the longest turn on average")
else:
print(f"{player_2} had the longest turn on average")
If player_1_misses > player_2_misses:
print(f"{player_1} missed the most")
else:
print(f"{player_2} missed the most")
# Show end credits
create_plot_with_logo_and_names()
Ask Copilot
GPT 4o
29°F
Mostly cloudy
Search
6:00 PM
12/4/2024


Untitled (Workspace)
Enhancing final scoreboard functionality in game code
ZacharyCornthwaite
How do I implement count_misses ?
main.py:152-185
GitHub Copilot
Used 1 reference
To implement the count_misses function, you need to iterate through the player's board and count the number of missed shots. Assuming that a miss is represented by a specific marker (e.g., 'M' for miss), you can implement the function as follows:
def count_misses(board):
misses = 0
for row in board:
for cell in row:
if cell == 'M': # Assuming 'M' represents a miss
misses += 1
return misses
You can then use this function in your final scoreboard calculation. Here's how you can integrate it:
# Function to count the number of misses on a player's board
def count_misses(board):
misses = 0
for row in board:
for cell in row:
if cell == 'M': # Assuming 'M' represents a miss
misses += 1
return misses
# Example usage in the final scoreboard
update_status(player_1_board, player_2_board, player_1_score, player_2_score)
if player_1_score == sum([5, 4, 3, 3, 2]) or player_2_score == sum([5, 4, 3, 3, 2]):
game_over = True
break
print("\nFinal Scoreboard:")
update_status(player_1_board, player_2_board, player_1_score, player_2_score)
player_1_ships_left = count_ships_left(player_1_board)
player_2_ships_left = count_ships_left(player_2_board)
player_1_avg_time = sum(player_1_turn_times) / len(player_1_turn_times) if player_1_turn_times else 0
Ask Copilot
GPT 4o
29°F
Mostly cloudy
Search
6:00 PM
12/4/2024


Untitled (Workspace)
Enhancing final scoreboard functionality in game code
player_2_ships_left = count_ships_left(player_2_board)
player_1_avg_time = sum(player_1_turn_times) / len(player_1_turn_times) if player_1_turn_times else 0
player_2_avg_time = sum(player_2_turn_times) / len(player_2_turn_times) if player_2_turn_times else 0
player_1_turns = len(player_1_turn_times)
player_2_turns = len(player_2_turn_times)
total_game_time = sum(player_1_turn_times) + sum(player_2_turn_times)
player_1_misses = count_misses(player_1_board)
player_2_misses = count_misses(player_2_board)
if player_1_score > player_2_score:
print(f"{player_1} wins!")
print(f"Average turn time: {player_1_avg_time:.2f} seconds")
print(f"Ships left: {player_1_ships_left}")
elif player_2_score > player_1_score:
print(f"{player_2} wins!")
print(f"Average turn time: {player_2_avg_time:.2f} seconds")
print(f"Ships left: {player_2_ships_left}")
else:
print("It's a tie!")
print(f"{player_1}'s average turn time: {player_1_avg_time:.2f} seconds")
print(f"{player_1}'s ships left: {player_1_ships_left}")
print(f"{player_2}'s average turn time: {player_2_avg_time:.2f} seconds")
print(f"{player_2}'s ships left: {player_2_ships_left}")
print(f"Total game time: {total_game_time:.2f} seconds")
print(f"{player_1} took {player_1_turns} turns")
print(f"{player_2} took {player_2_turns} turns")
if player_1_avg_time > player_2_avg_time:
print(f"{player_1} had the longest turn on average")
else:
print(f"{player_2} had the longest turn on average")
if player_1_misses > player_2_misses:
print(f"{player_1} missed the most")
else:
print(f"{player_2} missed the most")
# Show end credits
create_plot_with_logo_and_names()
Ask Copilot
GPT 4o
29°F
Mostly cloudy
Search
6:00 PM
12/4/2024