**Design Idea For Imperative Paradigm:**

**Pseudocode**

User input for level

If easy/ medium/ hard:

Choose random easy/ medium/ hard word

Loop (if tries< max Tries) && (timer isn’t up):

Display easy/ medium/ hard word

User guess letter

Check if letter in word

If yes:

loop again

Else

tries – 1

if word is guessed

Exit loop

End loop

If max Tries = tries

output (“max tries reached “and word)

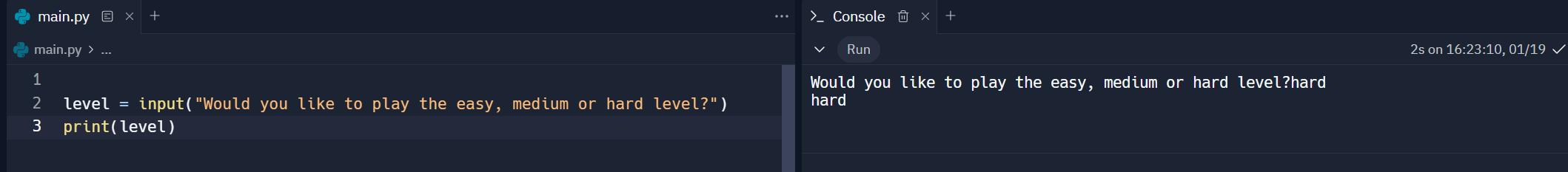
if timer is up

output (“ran out of time” and word)

else

output (“Invalid Input”)

**Beginning Of Development:**



I started off by taking in the level from the user.

A screenshot of a computer

Description automatically generated

I then used if selection statements to separate levels.

A screenshot of a computer

Description automatically generated

I created the word arrays and imported random to use to choose a random word in the array. I then tested this, and a random word was chosen each time.

A screenshot of a computer

Description automatically generated

I created the display word code, but an error occurred when I was running. It kept outputting \_.

A screenshot of a computer

Description automatically generated

I added another print outside of the If statement which adds a line in between the words so the word is being outputted correctly, but it just keeps looping through. The letters are not in a single line either so this needs to be fixed.

A screenshot of a computer program

Description automatically generated

A blue and black background

Description automatically generated with medium confidence

I asked AI for help on how to get the outputs on the same line and it suggested adding end to the print statements and as shown, it worked.

A screen shot of a computer

Description automatically generated I thought that the code kept looping through because only the display code is within the loop so there’s no way to exit the loop. So I added another line of code to check this and it was correct.

A screenshot of a computer

Description automatically generated

Takes in letter and checks if the user has already guessed it, if yes, they need to try again and program continues otherwise, letter is added to guessed letter array. It works as shown in console.

A screenshot of a computer

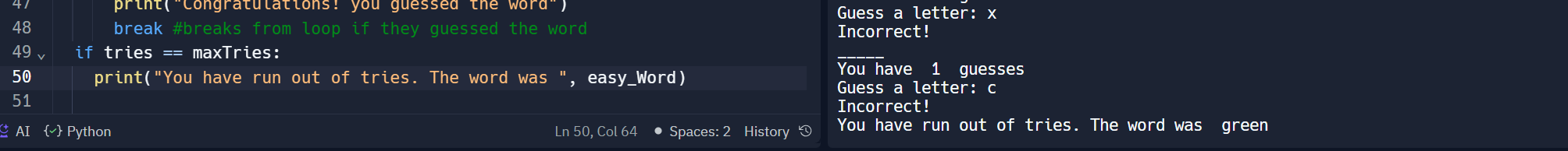
Description automatically generated

Line 39 – 43, letter is checked in easy word, if it is not there then tries is incremented. I tested this and it works in the console as intended.

A screenshot of a computer

Description automatically generated

45 – 48 checks if the word has been guessed. Loops through every letter in easy word and checks if the letters are in guessedLetters, outputs and breaks from loop If correct.



Outside of the while loop, I added an if statement to check tries = max Tries and outputs. As shown in console it works.

A screen shot of a computer program

Description automatically generated

Creating timer in the game by importing time module. Start time is the current time when game starts, then timePassed is difference between starttime and current time. Max time is how long game can run for.

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

Breaks out of loop if max time reached and then outside of loop is the output.

A blue background with white text

Description automatically generated

Tested and it works.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Copied the code over to the other levels but changed the values where needed.