**Design Idea For Procedural Paradigm:**

A diagram of a diagram

Description automatically generated with medium confidence

Created a diagram of how the system would run for hangman to find any potential repeated code that could be put into procedures.

**Beginning Of Development:**

A screenshot of a computer

Description automatically generated

I was checking that the user can successfully choose a data, but an error occurred where only Invalid Input would be outputted even though the user didn’t input anything.

A screenshot of a computer

Description automatically generated

I inputted breakpoints on the lines and saw that there wasn’t a value within level so the input was not being read and stored by the program. I fixed this by adding scanf under printf so that the input would be scanned and saved into level.

A screenshot of a computer

Description automatically generated

The input is successfully stored and level is chosen.

A screenshot of a computer

Description automatically generated

When creating the word arrays for the levels, there was an error due to not being able to declare variable length arrays at file scope. I used the AI in replit to help me overcome this issue as it recommended me to use the define method, so I did.

A screen shot of a computer program

Description automatically generated

I then created all the word arrays for the easy, medium, hard level. Then I created a random selection method. This selection method takes in the address of the parameter “selectedWord” which is a variable found in the main program. It chooses a random number as the index and stores the wordarray[index] as selectedword.

A screenshot of a computer

Description automatically generated

I added a call to the word selection procedure within the if selection statements. I then outputted the selected word to check if a random word was being selected. However, I outputted each level twice but both times the word was the same.

A screenshot of a computer

Description automatically generated

I investigated this but inputting the easy level 4 times as the code for the procedures are all the same just different arrays called within. I deduced that the random number was producing the same number each time so I turned to google for some help.

A screenshot of a computer program

Description automatically generated

<https://www.javatpoint.com/random-function-in-c>

This website used the line srand(time(0)); which seeds the random number generator used by the rand.

A screenshot of a computer

Description automatically generated

I inputted srand in and outputted easy level 5 times and the selected word was different.

A screen shot of a computer code

Description automatically generated

The next step In the diagram was to display the letters so I created a procedure for this as the system will need to display the letters every time the user guesses. The procedure takes in the address of the selectedword and the guessed letters given by the user. As the max word length was 15, some words will have white space at the end of them so this needs to be removed before displaying. The procedure loops through the letters of the selectedword, as long as selectedword[i] is not null (white space). Then all the letters in the array of guessed letters are compared to the selectedword[i] and if it matches, the letter is shown else \_ is shown.

A screenshot of a computer

Description automatically generated

I then called the display word procedure and the word is displayed as \_.

A screen shot of a computer program

Description automatically generated

The user will have tries to guess the letter, so I set the tries value as 0 as they haven’t yet tried a guess. I didn’t set the max tries as each level has a different max which is set within the if selection statements.

A screen shot of a computer program

Description automatically generated

While statement to iterate as long as max guess has not been reached. The number of tried remaining and the word are outputted. User is asked for a letter, and this letter is checked against all letters in the guessed array to stop user from re-guessing a letter. Otherwise the letter is added onto the array of guessed letters. Checks if the letter is within the word if yes it will just loop again, if no tries is incremented and loops again.

A screenshot of a computer program

Description automatically generated

Code checked, is working correctly, stops user from re-guessing. However, when I tried to guess the word feather it did the word letter by letter. Isn’t a problem but does lead to a lot of outputs to user.

A screen shot of a computer

Description automatically generated

Word guessed correctly but it is asking for a letter. I am going to stop this by adding a procedure to stop the loop once all letters are guessed.

A screen shot of a computer code

Description automatically generated

This loops through all letters in selected word and checks each letter with every letter in guessed letters to see if all letters were correctly guessed.

A screen shot of a computer code

Description automatically generated

Calling procedure within loop, if correct breaks loop otherwise continues.

A screenshot of a computer program

Description automatically generated

Loop successfully broken out of if word is correct.

A screenshot of a computer

Description automatically generated

If tries= max tries then loop is exited so code written to check if this happens and as shown in console, it works.

The final step Is to add a timer. Each level has a different amount of time due to level of difficulty.

A screenshot of a computer program

Description automatically generated

I was unsure of how to do the countdown timer so I used the suggestion given by AI as a starting point.

A screen shot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

A screenshot of a computer screen

Description automatically generated

Using the suggestion from AI and adding a few tweaks, The timer stops the program and outputs the users word.

A screen shot of a computer

Description automatically generated

I tested the game for a final time to see if it was working as intended. The game continues on the timer even though the user has guessed the word correctly.

A screenshot of a computer

Description automatically generated

I realised the issue was when the user completed the word, it exited the loop but didn’t stop the program. To fix this I replaced break with abort and the game should stop after the user guesses the word right, As shown on the console.