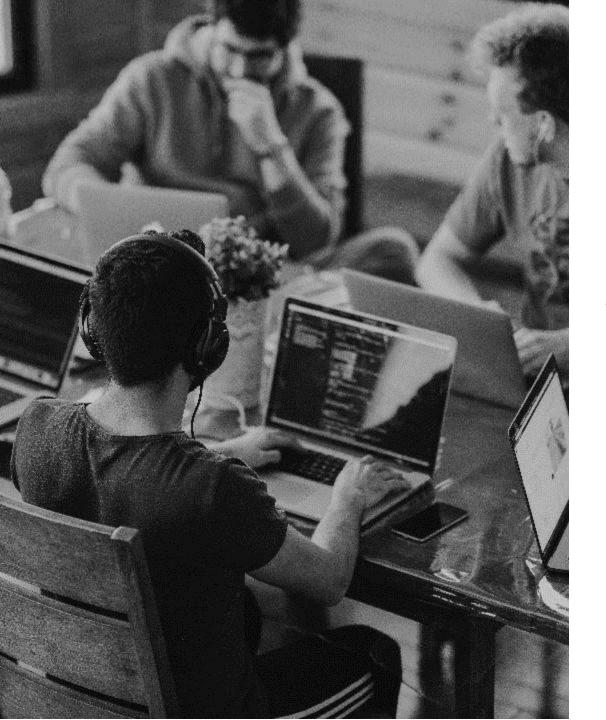# DATA STRUCTURES MINI PROJECT:
A GAME

PROBLEM STATEMENT

Name of the game – Hangman

What is it all about – The user has to guess a word, letter by letter, from a given hint. But the chances to guess this word are limited. Whenever a guess is wrong, the hangman starts getting drawn on the screen partly. Once this hangman is completely formed, it indicates all the chances are used and the game gets over. If the user guesses that word before the hangman is formed, she/he wins.

OBJECTIVE

To implement the data structure – Binary Search Tree, so as to decrease the running time for the operations to make the game faster and more efficient for the user.

USE

Used as a source of entertainment or fun

FUTURE SCOPE

Security systems where if the password is guessed correctly, the access to a system is granted.

IMPLEMENTATION

Data structure used – Binary Search Tree

Algorithm – Recursion

Programming language – Python

GUI developed with – PyGame

Other data structures used – Lists and Dictionaries.

WHY WE USED BINARY SEARCH TREE

This game is mainly about finding whether the input given by the user matches with any of the letters of the tree. Since the searching was predominantly occurring, BST was used. The search operation was completed quickly.

WHY WE USED PYGAME

Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language. Pygame is one of the most popular and reliable GUIs for python. It is specifically designed to support the graphics used for a game. It gives the options to display pictures, text and even plays sounds.

ALGORITHM

1. Start.
2. Create 3 lists of words for 3 levels (easy, medium, hard) from which a word will be randomly selected for the user to guess.
3. Create a dictionary which contains word as key and hint for that word as value.
4. Take the input from the user for level.
5. According to the level selected by the user, a word is randomly selected from one of the lists of words and is stored in a variable (say 'secretWord').
6. Maintain 2 different strings named 'missedLetters' for keeping track of letters which are wrongly guessed and 'correctLetters' for keeping track of all correctly guessed letters by the user.
7. Construct a BST for this secretWord where 1st letter of the secretWord would be the root value.

8.Other letters will get inserted into the BST after getting compared with the root using recursion. If value < root.value then value gets inserted into left subtree
   Else value gets inserted into right subtree.
9. Execute a while loop i.e.
    While true:
    a. Display the correctLetters and missedLetters.
    b. User guesses the letter which gets stored in a variable 'guess'. The guess should be single alphabet.
    c. Search the guessed letter in the BST
    d. If the letter exists in BST Then
        i.   Add the guessed letter into correctLetters.
        ii.Check if all letters are guessed, then game is over and display a message that the user has won and the secretWord.
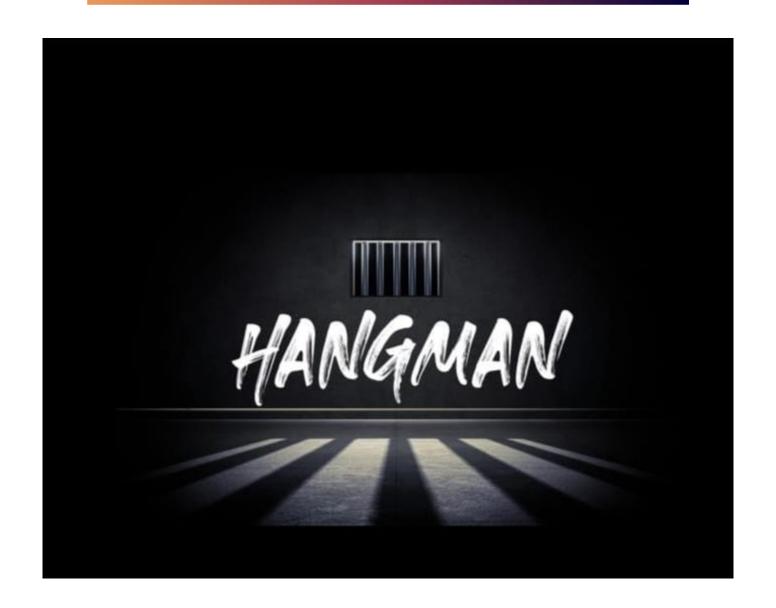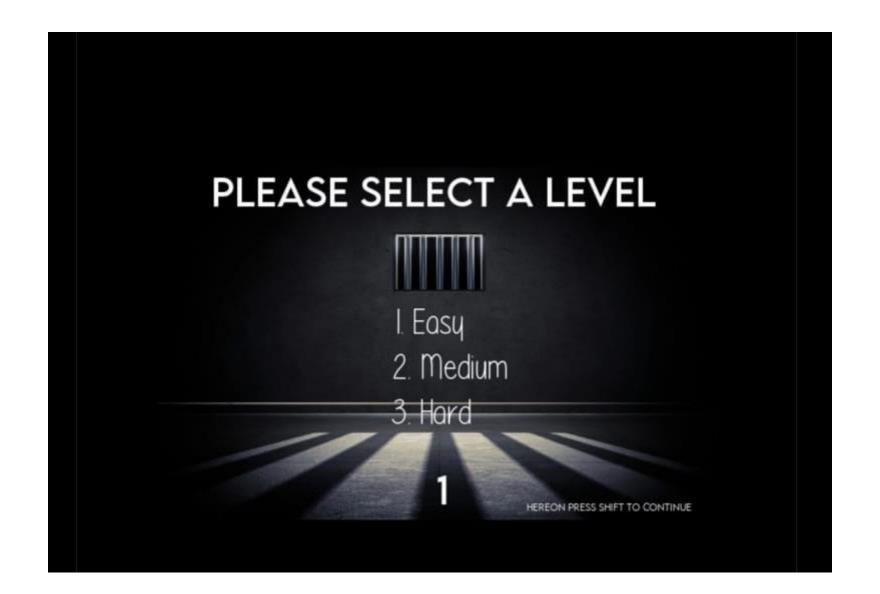
ALGORITHM

e. Else

    i. Add guess letter in missedLetters.

    ii. Print the hangman pictures partly. If the chances are over then print a msessage saying that the game is over and display the correctLetters and the secretWord.

f. If game is over, ask user whether to continue playing or not.

g. If user want to continue with game, then reset missedLetters, correctLetters to empty strings and go to step 3 and start executing again.

h. If user want to end the game then exit.
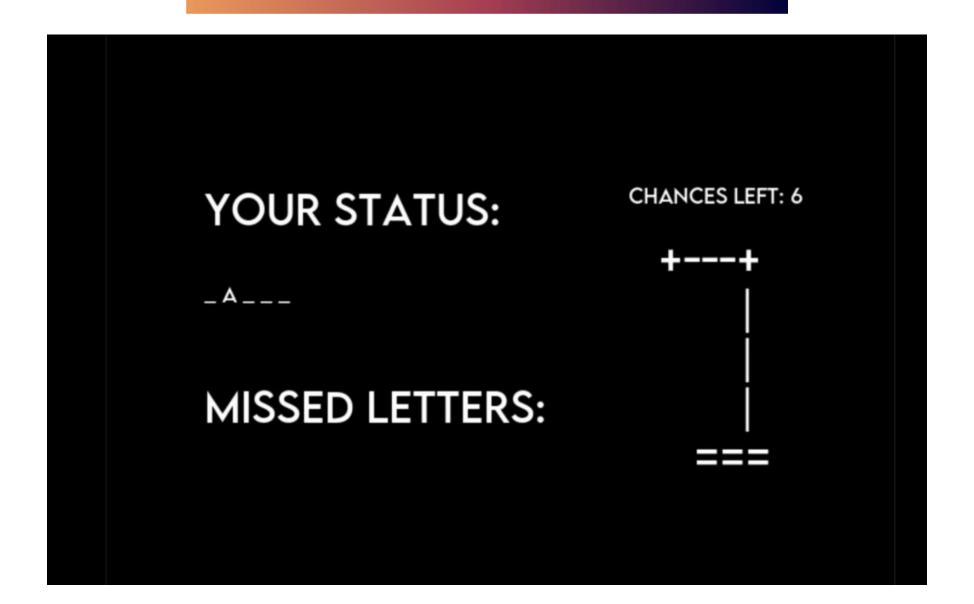
10. Exit.

LEVEL SELECTION

HINT

HINT
A DEVICE WHOSE MODE OF COMMUNICATION IS THROUGH WAVES

GUESS A LETTER!

STATUS

YOUR STATUS:

CHANCES LEFT: 6

_ A _ _ _

```
+---+
    |
    |
    |
===
```

MISSED LETTERS:

YOUR STATUS:

_ A _ I _

MISSED LETTERS:

S

CHANCES LEFT: 5

```
+---+
  O   |
      |
      |
     ===
```

SUCCESS

FAILURE

REPLAY

```
        T              Y
         /              /
        H              O
       /  \             \
      A    N             U
            /
           K

        THANK YOU!
```