**Title of the Project**

**A PROJECT REPORT**

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

**BACHELOR OF COMPUTER APPLICATION**

(SEMESTER)

Submitted by: Submitted to:

**Mamta Rawat Ms. Sugandhi Vij**

**(Assistant Professor)**

**Avviare Educational Hub, Noida (U.P)**

AVVIARE EDUCATIONAL HUB, NOIDA

(GLOCAL UNIVERSITY, SAHARANPUR, UTTARPRADESH

* 1. **Introduction to Project**

Welcome to Hangman Game !

Hangman is a classic word-guessing game that has entertained and challenged players for generations. In this digital adaptation, we've taken the traditional pen-and-paper game and brought it to life in an interactive and engaging format.

The objective of Hangman is simple: guess the hidden word before the hangman is fully drawn. With each incorrect guess, a new part of the hangman is added, bringing you one step closer to losing the game. But with each correct guess, letters are revealed, helping you unravel the mystery word.

**1.2 Project Category**

* Application Development

**1.3 Objectives**

* Develop an interactive Hangman Game application.
* Enhance vocabulary and word guessing skills.
* Provide an engaging and entertaining experience for users.

**1.4 Problem Formulation**

* Create a user-friendly interface for playing Hangman.
* Implement a mechanism for selecting random words.
* Design an efficient method for tracking and displaying guessed letters.

**1.5 Identification/Reorganization of Need**

* There's a need for an entertaining and educational game that can help users improve their vocabulary and cognitive skills.
* Hangman is a popular word guessing game that fulfills this need effectively.

**1.6 Existing System**

* Traditional pen-and-paper or physical Hangman game.
* Online Hangman games available on various platforms.

**1.7 Proposed System**

* Hangman Game 1.1 will be a digital application accessible on multiple platforms (web, mobile, desktop).
* Users can play solo or against others, either locally or online.
* The application will feature various difficulty levels and categories to choose from.

**1.8 Unique Features of the System**

* **Multi-platform Accessibility**: Available on web browsers, mobile devices, and desktop computers.
* **Customizable Settings**: Users can adjust difficulty levels, choose categories, and customize game settings.
* **Interactive User Interface**: Engaging graphics and animations to enhance the gaming experience.
* **Online Multiplayer Mode**: Challenge friends or random opponents in real-time.
* **Word Database**: Continuously updated word bank to keep the game fresh and challenging.

**Introduction to C Language:**

C is a powerful and widely-used programming language known for its efficiency and flexibility. Here's a brief overview of what you might cover when introducing C:

1. **Background and Significance:** Developed in the early 1970s by Dennis Ritchie at Bell Labs, C has since become one of the most widely-used programming languages. Its influence can be seen in many other languages, and it's particularly well-suited for system programming and applications where performance is critical.
2. **Basic Syntax and Structure:** Cover the fundamental aspects of C syntax, such as variables, data types (int, char, float), control structures (if-else, loops), functions, and arrays. Emphasize the importance of understanding memory management through concepts like pointers and dynamic memory allocation.
3. **Portability and Versatility:** One of the strengths of C is its portability, allowing programs written in C to be compiled and run on different platforms with minimal changes. Additionally, C's versatility makes it suitable for a wide range of applications, from operating systems to games.

**Introduction to VSCode IDE:**

Visual Studio Code (VSCode) is a popular and feature-rich code editor developed by Microsoft. Here's an overview of its relevance to the Hangman game project:

1. **Installation and Setup:** VSCode is available for various operating systems and is easy to install. You can guide users through the installation process and demonstrate how to set up the C/C++ extension for C programming support.
2. **Code Editing Features:** VSCode offers numerous features to enhance the coding experience, such as syntax highlighting, code completion, and IntelliSense for automatic code suggestions. These features can significantly improve productivity and help prevent common coding errors.
3. **Debugging Capabilities:** VSCode includes built-in debugging tools that allow developers to identify and fix issues in their code efficiently. You can demonstrate how to set breakpoints, step through code, and inspect variables during debugging sessions.
4. **Extensions and Customization:** VSCode has a vast ecosystem of extensions that provide additional functionality for various programming languages and tasks. You can explore relevant extensions for C development, such as linters, code formatters, and Git integrations.

**Tools and Technologies for Implementation:**

To implement the Hangman game project using C and VSCode, you'll need the following tools and technologies:

1. **C Compiler:** Choose a C compiler such as GCC (GNU Compiler Collection) or Clang, which will translate your C code into machine-readable instructions. These compilers are freely available and widely used in the C programming community.
2. **Text Editor or IDE:** VSCode serves as the primary development environment for writing and managing your C code. Its features and extensions enhance the development experience and improve productivity.
3. **Terminal or Command Prompt:** You'll use the terminal or command prompt to compile and execute your C programs. This is where you'll enter commands to compile your code using the chosen C compiler and run the resulting executable.
4. **Version Control System:** Consider using a version control system like Git to manage your project's source code, track changes, and collaborate with other developers. Platforms like GitHub or GitLab provide hosting for Git repositories and offer additional collaboration features.

Source code

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#include <time.h>

#include <string.h>

#define WORDS 10

#define WORDLEN 40

#define CHANCE 6

bool srand\_called = false;

int i\_rnd(int i) {

if (!srand\_called) {

srand(time(NULL) << 10);

srand\_called = true;

}

return rand() % i;

}

char\* decrypt(char\* code) {

int hash = ((strlen(code) - 3) / 3) + 2;

char\* decrypt = malloc(hash);

char\* toFree = decrypt;

char\* word = code;

for (int ch = \*code; ch != '\0'; ch = \*(++code))

{

if((code - word + 2) % 3 == 1){

\*(decrypt++) = ch - (word - code + 1) - hash;

}

}

\*decrypt = '\0';

return toFree;

}

void printBody(int mistakes, char\* body) {

printf("\tMistakes :%d\n", mistakes);

switch(mistakes) {

case 6: body[6] = '\\'; break;

case 5: body[5] = '/'; break;

case 4: body[4] = '\\'; break;

case 3: body[3] = '|'; break;

case 2: body[2] = '/'; break;

case 1: body[1] = ')', body[0] = '('; break;

default: break;

}

printf("\t \_\_\_\n"

"\t| |\n"

"\t| %c %c\n"

"\t| %c%c%c\n"

"\t| %c %c\n"

"\t| \n"

"\t| ", body[0], body[1], body[2],

body[3], body[4], body[5], body[6]);

}

void printWord(char\* guess, int len) {

printf("\t");

for (int i = 0; i < len; ++i)

{

printf("%c ", guess[i]);

}

printf("\n\n");

}

int main() {

printf("\n\t Be aware you can be hanged!!.");

printf("\n\n\t Rules : ");

printf("\n\t - Maximum 6 mistakes are allowed.");

printf("\n\t - All alphabet are in lower case.");

printf("\n\t - All words are name of very popular Websites. eg. Google");

printf("\n\t - If you enjoy continue, otherwise close it.");

printf("\n\t Syntax : Alphabet");

printf("\n\t Example : a \n\n");

char values[WORDS][WORDLEN] = {"N~mqOlJ^tZletXodeYgs","gCnDIfFQe^CdP^^B{hZpeLA^hv","7urtrtwQv{dt`>^}FaR]i]XUug^GI",

"aSwfXsxOsWAlXScVQmjAWJG","cruD=idduvUdr=gmcauCmg]","BQt`zncypFVjvIaTl]u=\_?Aa}F",

"iLvkKdT`yu~mWj[^gcO|","jSiLyzJ=vPmnv^`N]^>ViAC^z\_","xo|RqqhO|nNstjmzfiuoiFfhwtdh~",

"OHkttvxdp|[nnW]Drgaomdq"};

char \*body = malloc(CHANCE+1);

int id = i\_rnd(WORDS);

char \*word = decrypt(values[id]);

int len = strlen(word);

char \*guessed = malloc(len);

char falseWord[CHANCE];

memset(body, ' ', CHANCE+1);

memset(guessed, '\_', len);

char guess;

bool found;

char\* win;

int mistakes = 0;

setvbuf(stdin, NULL, \_IONBF, 0);

do {

found = false;

printf("\n\n");

printBody(mistakes, body);

printf("\n\n");

printf("\tFalse Letters : ");

if(mistakes == 0) printf("None\n");

for (int i = 0; i < mistakes; ++i)

{

printf("%c", falseWord[i]);

}

printf("\n\n");

printWord(guessed, len);

printf("\tGive me a alphabet in lower case : ");

do {scanf("%c",&guess);} while ( getchar() != '\n' );

for (int i = 0; i < len; ++i)

{

if(word[i] == guess) {

found = true;

guessed[i] = guess;

}

}

if(!found) {

falseWord[mistakes] = guess;

mistakes += 1;

}

win = strchr(guessed, '\_');

}while(mistakes < CHANCE && win != NULL);

if(win == NULL) {

printf("\n");

printWord(guessed, len);

printf("\n\tCongrats! You have won : %s\n\n", word);

} else {

printf("\n");

printBody(mistakes, body);

printf("\n\n\tBetter try next time. Word was %s\n\n", word);

}

free(body);

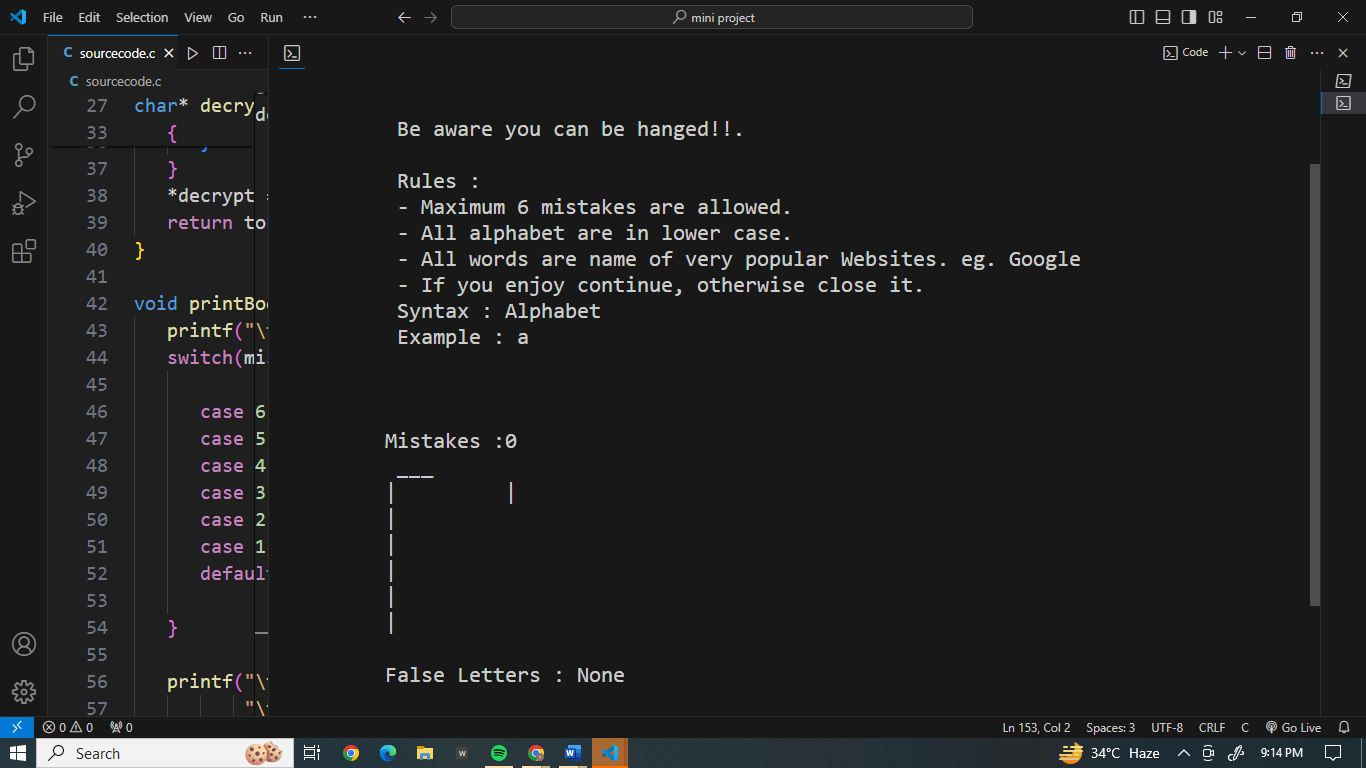
free(word);

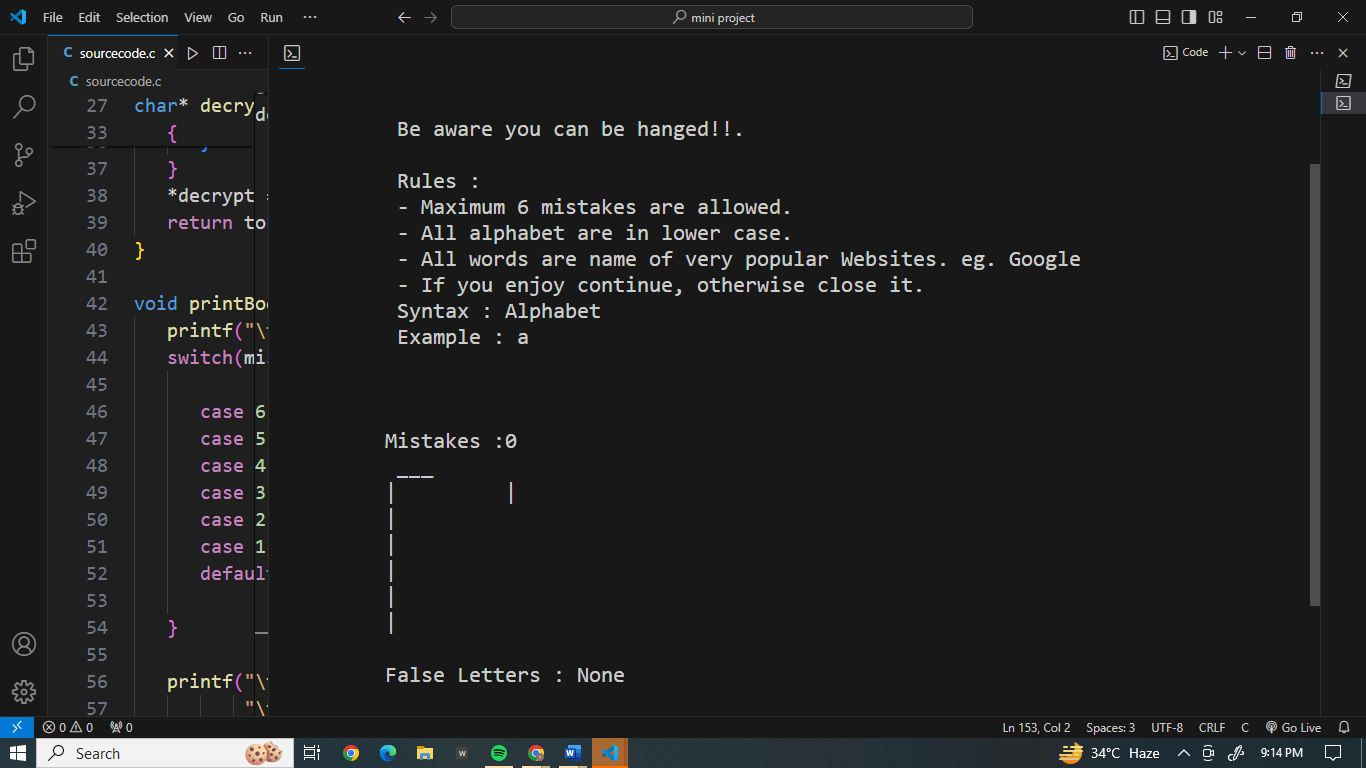
free(guessed);

return EXIT\_SUCCESS;

}

Output





Conclusion

In conclusion, the Hangman game mini-project serves as an excellent introduction to C programming, providing an opportunity for developers to apply their skills in a fun and interactive manner. Through this project, developers gain hands-on experience with fundamental concepts such as data types, control structures, functions, and memory management, while also honing their problem-solving and debugging skills.

Future Scope

1. **Graphical User Interface (GUI):** Consider adding a graphical interface using libraries like GTK, Qt, or SDL. This would provide a more visually appealing experience for players.
2. **Multiple Difficulty Levels:** Implement different difficulty levels with varying word lengths or categories (e.g., easy, medium, hard) to cater to players of different skill levels.
3. **Scoring System:** Introduce a scoring system based on factors like the number of guesses taken, time taken to guess the word, or difficulty level. Players can compete to achieve high scores and compare them with others.
4. **Word Database:** Expand the word database to include a larger variety of words, phrases, or even themes. This adds replay value and keeps the game fresh for players.
5. **Hangman Animation:** Add animations or graphics to depict the progress of the hangman as players make incorrect guesses, making the game more immersive.

Reference and Bibliography

<https://www.geeksforgeeks.org>

<https://www.onlinegdb.com>