## JUMBLED WORDS GAME USING PYTHON

***Dissertation submitted to the***

**MAHENDRA ARTS & SCIENCE COLLEGE (AUTONOMOUS)**

### In partial fulfillment of the requirements for the award of the degree of BACHELOR OF SCIENCE IN COMPUTER SCIENCE

**Submitted by**

### SIVARAM Y

**(20BCS1061)**

***Under the Guidance of***

**Mrs.M.GOMATHI,Msc.,M.Phil.,**

**DEPARTMENT OF COMPUTER SCIENCE & APPLICATIONS MAHENDRA ARTS & SCIENCE COLLEGE**

## (Autonomous)

### Affiliated to Periyar University

**Accredited with Grade 'A++' by NAAC | Recognized u/s 2(f) & 12(B) of the UGC Act, 1956**

## Kalippatti (Po), Namakkal (Dt) - 637501

APR/MAY- 2023

MAHENDRA ARTS & SCIENCE COLLEGE (Autonomous)

Kalippatti

(Affiliated to Periyar University, Salem)



This is to certify that the project entitled

### JUMBLED WORDS GAME USING PYTHON

is the Bonafide record of project work done

by **SIVARAM Y (20BCS1061)**

A project submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science to the

Mahendra Arts & Science College (Autonomous)

**Mrs.M.GOMATHI,Msc.,M.Phil.,**

Assistant Professor

Dept. of Computer Science & Applications,

**Dr.R INDHUMATHI,MCA.,M.Phil.,Ph.D.,**

Head of the Department

Dept. of Computer Science & Applications

Mahendra Arts & Science College (Autonomous), Mahendra Arts & Science College (Autonomous), Kalippatti-637501. Kalippatti-637501

Submitted for viva – voce examination held on 07.06 .2023

Internal Examiner External Examiner

DECLARATION

I **SIVARAM Y** hereby declare that the project work, entitled “**JUMBLED WORDS GAME USING PYTHON**” submitted to the Mahendra Arts & Science College (Autonomous), Kalippatti in partial fulfillment of the requirements for the award of the degree of **Bachelor of Comeputer Scienc**e is a record of the original project work done by me under the supervision and guidance of **Mrs.M.GOMATHI,Msc.,M.Phil., Assistant Professor**, Department of Computer Science & Applications, Mahendra Arts & Science College (Autonomous), Kalippatti and it has not formed the basis for the award of any Degree / Diploma / Associate ship / Fellowship or other similar title to any candidate in any university.

Place: Kalippatti Signature of the Candidate Date: 07.06.2003 [SIVARAM Y]

### ACKNOWLEDGEMENT

I would like to express my deepest gratitude to **Shri. M. G. BHARATHKUMAR, M.A., B.Ed.,** Chairman of Mahendra Educational Trust for offering me an opportunity and providing me all the facilities to do my Project work.

I am grateful for you and your generosity **Smt.B.VALLIYAMMAL, M.A., B.Ed**., Secretary, Mahendra Educational Trust for providing excellent facilities.

I extended my sincere thanks to the Managing Directors of Mahendra Educational Trust

### Mr. Ba. MAHENDHIRAN and Mr. B. MAHA AJAY PRASATH.

I would like to convey my sincere gratitude and thanks to the Principal of Mahendra Arts & Science College, (Autonomous) **Dr. S. ARJUNAN, M.Sc., M.Phil., Ph.D** for providing me extremely useful and enlightening opportunity to inclusive this work.

I am ineffably indebted to **Dr. J. JOSEPHINE DAISY., M.Com., M.Phil.,MBA., Ph.D.,** the Controller of Examinations of Mahendra Arts & Science College, (Autonomous) for conscientious guidance and encouragement to accomplish this project work.

I express my profound thanks to **Dr.R INDHUMATHI,MCA.,M.Phil.,Ph.D., Head, Department of Computer Science & Applications,** for her advice and assistance in keeping my progress on schedule.

I would like to express my deep gratitude to **Mrs.M.GOMATHI,Msc.,M.Phil., , Department of Computer Science & Applications** for my project guide, for their patient guidance, enthusiastic encouragement and useful critiques of this work.

I would also like to expand my deepest gratitude to all those who have directly and indirectly guided us in writing this assignment work.

Finally, I wish to thank my parents for their support and encouragement throughout my study

CONTENTS

|  |  |  |
| --- | --- | --- |
| S.No. | PARTICULARS | PAGE No. |
|  | ABSTRACT | 6 |
| 1. | INTRODUCTION | 7 |
| 2. | SYSTEM SPECIFICATION | 9 |
| 2.1 HARDWARE SPECIFICATION |  |
| 2.2 SOFTWARE SPECIFICATION |  |
| 3. | SYSTEM STUDY AND ANALYSIS | 10 |
| 3.1 EXISTING SYSTEM |  |
| 3.2 PROPOSED SYSTEM |  |
| 3.3 FEASIBILITY STUDY |  |
| 4. | SOFTWARE DESCRIPTIONS | 13 |
| 4.1 FRONT END |  |
| 4.2 BACK END |  |
| 4.3 FEATURES |  |
| 5. | PROJECT DESCRIPTION | 20 |
| 5.1 PROBLEM DEFINITION |  |
| 5.2OVERVIEW OF THE PROJECT |  |
| 5.3 MODULES DESCRIPTION |  |
| 6. | SYSTEM DESIGN AND IMPLEMENTATION | 30 |
| 7. | CONCLUSION | 33 |
| 8. | FUTURE ENHANCEMENT | 35 |
| 9. | APPENDIX | 38 |
| 9.1 SOURCE CODE |  |
| 9.2 SCREEN SHOTS |  |
| 10. | REFERENCES | 41 |

ABSTRACT

This study aims at describing the implementation and the effectiveness of jumbledsentences as

a technique to enhance students’ skill in writing report text and finding out the factors influencing students’ skill in writing report text. This study was conducted in MAN 2 Yogyakarta that involved 24 tenth graders of IPA 3 in academic year 2018/2019. This study was a classroom action research. To collect the data, observation checklist, structured interview, and tests were used. The resultshowed that most of students could understand the subject easily and involve actively during the technique implementation. Besides, the test results showed students’ meanimprovement in writing report text from pre-test (3.00), post-test I (3.36), and post-test II (3.78). means thatjumbled-sentences technique is effective to enhance students’ skill in writing report text.Moreover, the factors influencing students’ writing skill were having lack of vocabulary, getting anxiety in making grammar errors, getting difficulties in understanding and translating words or sentences, and having limited classroom for learning-teaching process. Keywords: implementation; jumbled-sentences; students’ skill; writing; report text.

# INTRODUCTION

Introduction to Jumbled Words Game:

The Jumbled Words Game is an interactive and entertaining word puzzle that challenges players to unscramble jumbled words and find the correct answers. This popular word game tests players' vocabulary, spelling, and problem-solving skills, making it a fun and engaging activity for people of all ages.

The objective of the game is simple: players are presented with a word that has been jumbled or scrambled, where the letters are rearranged in a random order. The player's task is to decipher the original word by rearranging the letters in the correct order to form a meaningful word. It requires careful observation, analysis, and a good grasp of language to solve the puzzles.

The Jumbled Words Game offers an excellent opportunity to enhance language skills and mental agility. By engaging in this game, players can expand their vocabulary, improve their spelling, and boost their cognitive abilities. It encourages players to think critically, creatively, and analytically as they work through the challenging puzzles.

### Features and Mechanics:

1. Random Word Selection: The game utilizes a vast database of words, from which it randomly selects jumbled words to present to the players. This ensures a diverse and varied gameplay experience.
2. Time Limit (Optional): To add an element of excitement and urgency, the game may incorporate a time limit within which players must solve each jumbled word. This feature tests their ability to think quickly and make accurate decisions under pressure.
3. Difficulty Levels: The game can offer multiple difficulty levels, catering to players of different skill levels. Beginners can start with easy or common words, while advanced players can tackle more complex and challenging jumbles.
4. Hint System (Optional): To assist players who may be stuck on a particular jumbled word, the game may provide hints or clues. These hints could be in the form of definitions, synonyms, or even partial rearrangements of the jumbled word.
5. Score Tracking: The game keeps track of players' scores, providing feedback on their performance. Points may be awarded based on the number of correct answers or the speed at which the words are unscrambled. This feature encourages players to improve their skills and compete against their own records or with others.

### Benefits of Playing the Jumbled Words Game:

1. Vocabulary Expansion: By deciphering jumbled words and finding their correct forms, players are exposed to new words and their meanings. This process helps improve vocabulary and language comprehension.
2. Cognitive Development: Solving jumbled word puzzles requires mental effort, concentration, and logical thinking. Regular engagement with the game can enhance memory, problem-solving abilities, and overall cognitive function.
3. Spelling Improvement: As players unscramble words, they develop a better understanding of letter combinations, spelling patterns, and correct word structures. This practice can contribute to improved spelling skills in everyday writing.
4. Fun and Entertainment: The Jumbled Words Game offers an enjoyable and challenging experience for players. It can be played individually, with friends, or in competitive settings, providing a source of entertainment and engagement.
5. Educational Tool: The game can be a valuable educational tool in schools and language learning environments. It combines learning with entertainment, making it an effective way to reinforce vocabulary and language skills in a fun and interactive manner.

### Conclusion:

The Jumbled Words Game is an exciting and intellectually stimulating word puzzle that engages players in unscrambling jumbled words. By providing opportunities for vocabulary expansion, spelling improvement, and cognitive development, this game offers a rewarding and enjoyable experience. Whether played for leisure, educational purposes, or as a means of friendly competition, the Jumbled Words Game is a delightful way to challenge one's linguistic skills and have a great time in the process.

* 1. **SYSTEM SPECIFICATIONS**
  2. **HARDWARE SPECIFICATION:**

PROCESSOR : AMD PRO A4-4350B,2.5GHZ

RAM : 4GB

HARD DISK : 500GB

KEYBOARD : STANDARD KEYBOARD

## SOFTWARE SPECIFICATION

OPERATING SYSTEMS : WINDOWS 10 PRO(64 BIT) FRONT END : PYTHON

IDLE FOR PYTHON : VERSION (3.11 64 BIT)

# SYSTEM STUDY AND ANALYSIS

## EXISTING SYSTEM:

In the existing system word search game is designed only for English language. Word search helps the people to easily learn the vocabulary words, places, historical figures and other theme-based topics. This game only available in online mode. These games also include timer facility that help the people to complete the levels within the given time. It consists of many stages starting from easy, medium, hard.

## PROPOSED SYSTEM:

In this proposed system, offline game design is formulated. This may induce the interest among the people of opting this offline game rather than the tedious online games.it includes offline multiplayer mode by using Bluetooth. Static slot table algorithm is used for the construction of word search game. Hidden words will be displayed at the bottom. The player needs to find the hidden words in the grid of jumbled letters. Timer facility is also added; the player needs to find the hidden word within the specified time. Once the player the finds the word, then next hidden new word will be displayed and the game process goes on. They have different theme Ain each stage. The score depends on the speed of finding the hidden word.

* 1. FEASIBILITY STUDY

### Introduction:

A feasibility study is a crucial step in determining the viability and potential success of a project or initiative. In the context of developing a Jumbled Words Game, conducting a feasibility study helps evaluate the practicality, market demand, technical requirements, and financial aspects associated with creating and launching the game. This comprehensive study aims to assess the feasibility of developing and implementing the Jumbled Words Game by considering various key factors.

### Market Analysis:

To begin the feasibility study, it is essential to analyze the market for word games and puzzle games. This involves researching the target audience, their preferences, and the current demand for such games. Conducting market surveys, studying competitor offerings, and analyzing user trends can provide valuable insights into the potential success of the Jumbled Words Game.

### Technical Considerations:

Developing a Jumbled Words Game requires assessing the technical requirements and capabilities. This includes evaluating the necessary software development tools, programming languages, and platforms suitable for creating the game. Additionally, considering factors such as cross-platform compatibility, user interface design, and gameplay mechanics is crucial to ensure a seamless and user-friendly gaming experience.

### Content and Word Database:

The success of a Jumbled Words Game heavily relies on the quality and variety of jumbled words presented to the players. Conducting research to build a comprehensive and diverse word database is essential. The feasibility study should address the sources of obtaining jumbled words, the process of creating new puzzles, and ensuring the accuracy and appropriateness of the words used in the game.

### Gameplay Features:

Assessing the potential gameplay features and mechanics is vital to make the Jumbled Words Game engaging and enjoyable for players. This involves considering options such as different difficulty levels, time limits, hint systems, scoring mechanisms, and multiplayer capabilities. Understanding the technical feasibility and user preferences for these features is essential to create a captivating gameplay experience.

### Monetization Strategies:

Evaluating the monetization potential of the Jumbled Words Game is crucial for the project's financial feasibility. Exploring various revenue streams such as in-app purchases, advertisements, and premium versions can help generate revenue. Additionally, conducting a cost-benefit analysis,

estimating development and maintenance costs, and projecting potential profits will provide insights into the game's financial viability.

### Technical Challenges and Risks:

Identifying potential technical challenges and risks associated with the development and implementation of the Jumbled Words Game is necessary. This includes assessing factors such as compatibility issues, performance optimization, security considerations

### Conclusion:

The feasibility study for the Jumbled Words Game provides a comprehensive analysis of the market demand, technical requirements, content considerations, gameplay features, monetization strategies, and potential challenges associated with the development and implementation of the game. Conducting this study allows project stakeholders to make informed decisions, allocate resources effectively, and ensure the successful creation and launch of an engaging and commercially viable Jumbled Words Game. By considering the various aspects outlined in the feasibility study, the development team can increase the chances of creating a successful and enjoyable gaming experience for the target audience.

# SOFTWARE DESCRIPTIONS

## FRONT END: PYTHON

Python is a widely-used, high-level programming language known for its simplicity, readability, and versatility. It was created by Guido van Rossum and first released in 1991, with the design philosophy of emphasizing code readability and promoting a clean and concise syntax. Python is an interpreted language, which Python means that it does not need to be compiled before execution, making it highly accessible and interactive.

Python is an open-source language, which means that its source code is freely available and can be modified and distributed by the user community. This has contributed to its rapid growth and popularity, as developers around the world have contributed to its development and created a vast ecosystem of libraries, frameworks, and tools.

### Key Features of Python:

1. Readability: Python's syntax is designed to be clear and expressive, focusing on simplicity and readability. Its use of whitespace indentation instead of curly brackets or semicolons enhances code readability and reduces the likelihood of syntax errors.
2. Easy to Learn: Python's straightforward syntax and extensive documentation make it an ideal language for beginners. It emphasizes simplicity and encourages good programming practices, such as code modularity and reusability.
3. Cross-platform Compatibility: Python is a cross-platform language, meaning it can run on different operating systems, including Windows, macOS, Linux, and more. This portability makes Python a versatile choice for developing applications that can be deployed on various platforms.
4. Large Standard Library: Python comes with a comprehensive standard library that provides a wide range of modules and functions, making it easier to perform common tasks without the need for external libraries. This library covers areas such as file handling, network programming, web development, data manipulation, and more.
5. Extensive Third-Party Ecosystem: Python has a vibrant and active community that has developed

an extensive collection of third-party libraries and frameworks. These include popular libraries like NumPy for numerical computing, Pandas for data analysis, Django for web development, TensorFlow for machine learning, and many others. This rich ecosystem enables developers to leverage existing solutions and accelerate development.

1. Object-Oriented Programming (OOP) Support: Python supports object-oriented programming, allowing developers to structure their code using classes, objects, and inheritance. This paradigm promotes code organization, modularity, and code reuse.
2. Dynamic Typing: Python uses dynamic typing, which means that variable types are determined at runtime. This flexibility allows for rapid prototyping and simplifies code development, as variables can change types as needed.
3. Strong Community and Support: Python has a strong and supportive community of developers who actively contribute to its development, create tutorials, participate in forums, and offer assistance. The availability of resources and community support makes Python an attractive choice for both beginners and experienced developers.

### Applications of Python:

Python's versatility has led to its widespread use in various domains and industries. Some of the common applications of Python include:

1. Web Development: Python is extensively used in web development frameworks like Django and Flask to build scalable and feature-rich web applications.
2. Data Analysis and Visualization: Python, along with libraries like Pandas, NumPy, and Matplotlib, is widely used in data analysis, scientific computing, and visualization. It is a popular choice among data scientists and analysts for processing and interpreting data.
3. Machine Learning and Artificial Intelligence: Python's simplicity and the availability of libraries such as TensorFlow, PyTorch, and scikit-learn make it a popular language for machine learning and artificial intelligence tasks. It is widely used for developing and training machine learning models.
4. Scripting and Automation: Python's ease of use and powerful scripting capabilities make it a preferred language for automating repetitive tasks, system administration, and writing scripts for various purposes.
5. Game Development: Python has libraries like Pygame that facilitate

## FEATURES:

Python is a versatile and popular programming language known for its simplicity, readability, and flexibility. It offers a wide range of features that make it a preferred choice for developers, data scientists, and beginners alike. In this document, we will explore the key features of Python in detail, highlighting its strengths and advantages.

## Readability:

Python's syntax is designed to be highly readable, emphasizing code clarity and reducing the cost of program maintenance. It uses whitespace indentation rather than brackets or braces, making the code visually appealing and easy to understand. The clean and elegant syntax enables developers to write clear and concise code, promoting better collaboration and faster development cycles.

## Easy to Learn and Use:

Python is widely regarded as one of the most beginner-friendly programming languages. Its simplicity and intuitive syntax make it easy to learn, even for individuals without prior programming experience. Python's extensive standard library provides ready-to-use modules and functions, reducing the need for additional code and simplifying development tasks.

## Large Standard Library:

Python comes with a comprehensive standard library that provides a vast collection of modules and functions for various tasks. This library covers areas such as file I/O, networking, web development, data manipulation, and more. The rich set of pre-built modules allows developers to quickly implement complex functionality without

reinventing the wheel, saving time and effort.

## Cross-Platform Compatibility:

Python is a cross-platform language, meaning that Python programs can run on different operating systems without requiring significant modifications. Whether you are using Windows, macOS, Linux, or other platforms, Python offers consistent behavior and code portability. This feature enables developers to write code once and deploy it on multiple platforms, expanding the reach of their applications.

## Object-Oriented Programming (OOP):

Python supports object-oriented programming paradigms, allowing developers to create modular and reusable code. It offers classes, objects, inheritance, and polymorphism, facilitating the organization and structuring of complex applications. OOP in Python promotes code reusability, encapsulation, and the separation of concerns, leading to more maintainable and scalable software solutions.

## Dynamic Typing and Automatic Memory Management:

Python is dynamically typed, meaning that variable types are determined at runtime. Developers can assign values of different types to variables without explicit type declarations. This flexibility enables rapid prototyping, quick experimentation, and easier code maintenance. Python also features automatic memory management through garbage collection, relieving developers from manual memory management tasks and reducing the chances of memory leaks and segmentation faults.

## Extensive Third-Party Libraries and Ecosystem:

Python has a vibrant and active community that has developed a vast ecosystem of third-party libraries and frameworks. These libraries expand Python's capabilities and provide solutions for various domains such as web development (Django, Flask), scientific computing (NumPy, SciPy), data analysis (Pandas), machine learning (TensorFlow, PyTorch), and more. Leveraging these libraries, developers can accelerate development, enhance functionality, and tap into a wealth of pre-existing.

## Integration and Scripting:

Python can seamlessly integrate with other programming languages and systems, making it an excellent choice for scripting and automation tasks. It provides interfaces to C/C++, Java, .NET, and other languages, enabling developers to combine the strengths of multiple languages in a single application. Python's integration capabilities make it a valuable tool for system administration, data processing, and rapid prototyping.

## Scalability and Performance:

Although Python is an interpreted language, it offers various options for improving performance. Libraries like NumPy and PyPy provide efficient numerical operations, while tools like Cython allow developers to write Python code that can be compiled to highly performant C code. Moreover, Python's simplicity and ease of development contribute to faster time-to-market, making it an attractive choice for building scalable applications.

## Community and Support:

Python has a large and active community of developers, enthusiasts, and contributors. The community-driven nature of Python fosters collaboration, knowledge sharing, and continuous improvement. Developers can seek help, participate in forums, attend conferences, and access a wealth of learning resources. This strong community support ensures that developers can easily find solutions, get assistance, and stay up to date with the latest trends and best practices in Python development.

## Testing and Debugging:

Python provides robust tools and frameworks for testing and debugging code. The built-in unittest module offers a framework for writing test cases, running test suites, and performing assertions. Additionally, tools like pytest and coverage.py enhance testing capabilities and code coverage analysis. Python's debugging tools, such as pdb (Python Debugger), facilitate step-by-step debugging and code inspection, aiding

in identifying and resolving issues effectively.

## Documentation and Readability:

Python places a strong emphasis on documentation and readability. The official Python documentation is comprehensive and well-maintained, providing detailed explanations, examples, and guidelines. Additionally, Python's community encourages the use of docstrings (documentation strings) to document code within the code itself, making it easier for developers to understand and maintain the codebase.

## Rapid Prototyping and Development:

Python's ease of use, extensive libraries, and expressive syntax make it an ideal choice for rapid prototyping and development. Developers can quickly turn their ideas into working prototypes, test concepts, and iterate on designs. Python's agility and productivity features, such as dynamic typing and high-level data structures, contribute to faster development cycles and efficient code iteration.

## Security and Reliability:

Python prioritizes security and reliability. The language has a strong focus on code readability, reducing the chances of introducing common programming errors and vulnerabilities. Additionally, Python's standard library and third-party packages undergo rigorous testing and security reviews, ensuring a robust and reliable ecosystem for developers to build upon.

## Open Source and Community-Driven Development:

Python is an open-source language with an active and collaborative development process. The Python Enhancement Proposal (PEP) process allows community members to propose and discuss language improvements. This open and inclusive approach empowers developers to contribute to the language's evolution and shape its future direction.

In conclusion, Python's rich set of features, simplicity, readability, and extensive ecosystem make it a powerful and versatile programming language. Its wide range of applications, from web development and data analysis to scientific computing and machine learning, has contributed to its popularity and widespread adoption. Python's community-driven development, strong documentation, and robust support make it an excellent choice for beginners, experienced developers, and organizations seeking to build scalable and efficient software solutions.

# 5.PROJECT DESCRIPTION

## PROBLEM DEFINITION

### Introduction:

The Jumbled Words Game is an engaging and challenging word puzzle that requires players to rearrange scrambled letters to form meaningful words. This problem definition outlines the key features and requirements of the game, aiming to provide a comprehensive understanding of its design and implementation.

### Problem Statement :

The problem is to develop a Jumbled Words Game that generates a random set of scrambled words and challenges players to unscramble them within a specified time limit. The game should provide an interactive user interface, track player scores, and offer hints when needed. The goal is to create an entertaining and educational game that promotes vocabulary and cognitive skills while providing an enjoyable gaming experience.

### Functional Requirements:

* 1. **Word Generation**:
     1. The game should have a database of words from various categories (e.g., animals, countries, fruits).
     2. It should randomly select a word and scramble its letters to create a jumbled word.
     3. The jumbled word should maintain the same number of letters as the original word.

### Gameplay:

* + 1. The game should present the jumbled word to the player.
    2. The player must rearrange the letters to form a valid word.
    3. The game should provide a time limit for each word, and the player must solve it within that time.
    4. After submitting the answer, the game should check if it matches the original word and calculate the score accordingly.
    5. The game should display the correct word if the player fails to solve it within the time limit.

### User Interface:

* + 1. The game should have an intuitive and visually appealing interface.
    2. It should display the jumbled word, input area for the player's answer, and buttons for hints and submitting the answer.
    3. The game should show the player's score and the time remaining for each word.

### Hints:

* + 1. The game should provide hints to assist players in solving difficult jumbled words.
    2. Hints may include revealing one letter or providing a clue related to the word's category.

### Levels and Progression:

* + 1. The game should have multiple levels of increasing difficulty.
    2. Each level should include a set of jumbled words from different categories.
    3. Players should be able to unlock higher levels by successfully completing lower ones.

### Non-functional Requirements:

* 1. **Performance**:
     1. The game should respond quickly to user actions and provide a seamless experience.
     2. The word generation and checking mechanisms should be efficient to ensure smooth gameplay.

### Compatibility:

* + 1. The game should be compatible with various platforms, such as desktops, mobile devices, and tablets.
    2. It should support popular web browsers and operating systems.

### Security:

* + 1. The game should not collect or store any personal information from the players.
    2. It should adhere to standard security practices to prevent unauthorized access or data breaches.

### Conclusion:

The Jumbled Words Game aims to provide an entertaining and educational experience for players. By implementing the outlined requirements, the game will challenge players' word-solving skills while offering a user-friendly interface, various levels, and hints for assistance. By incorporating these features, the game will create an engaging experience that promotes vocabulary building and cogninative development.

## OVERVIEW OF THE PROJECT

### Introduction:

The Jumbled Words Game is an interactive and challenging word puzzle project that aims to develop a game where players unscramble jumbled letters to form meaningful words. This project overview provides a comprehensive understanding of the game's objectives, features, and implementation details, highlighting its educational and entertainment value.

### Objectives :

The primary objective of the Jumbled Words Game project is to create an engaging and educational word puzzle game that promotes vocabulary building, cognitive skills, and problem-solving abilities. The specific objectives include:

* 1. Developing a word generation mechanism to generate random jumbled words from various categories.
  2. Designing a user-friendly interface that facilitates easy interaction with the game.
  3. Implementing a scoring system to track and display players' performance.
  4. Incorporating hints to assist players in solving difficult jumbled words.
  5. Creating multiple levels of increasing difficulty to provide a progression system.
  6. Ensuring compatibility across different platforms and devices.

### Features and Gameplay:

* 1. **Word Generation:**

The game will include a comprehensive database of words from different categories (e.g., animals, countries, fruits). It will randomly select a word from the database and scramble its letters to create a jumbled word. The jumbled word will maintain the same number of letters as the original word.

### Gameplay:

Players will be presented with a jumbled word and must rearrange the letters to form a valid word. The game will provide a time limit within which the player needs to solve the word. After submitting an answer, the game will check if it matches the original word and calculate the player's score accordingly. If the player fails to solve the word within the time limit, the correct word will be displayed.

### User Interface:

The game will feature an intuitive and visually appealing interface. It will display the jumbled

word, provide an input area for the player's answer, and include buttons for hints and submitting the answer. The interface will also show the player's score and the time remaining for each word.

### Hints:

To assist players in solving challenging jumbled words, the game will offer hints. Hints may include revealing one letter of the correct word or providing a clue related to the word's category.

Players can choose

to use hints to gain assistance when needed.

### Levels and Progression:

The game will have multiple levels of increasing difficulty. Each level will consist of a set of jumbled words from different categories. Players will unlock higher levels by successfully completing lower ones, providing a sense of progression and accomplishment.

### Implementation Details

The Jumbled Words Game will be developed using suitable programming languages and frameworks, such as HTML, CSS, and JavaScript. The game's word generation mechanism will be built using algorithms to select random words from the database and scramble their letters. The user interface will be designed using HTML and CSS, ensuring responsiveness and compatibility with various devices.

The scoring system will be implemented using JavaScript to track and calculate players' scores based on correct word submissions. The game's logic will handle the checking of player answers and display the correct word if the time limit is exceeded.

To ensure compatibility, the game will be designed to work seamlessly on desktops, mobile devices, and tablets. It will be tested on popular web browsers and different operating systems to ensure a smooth and consistent user experience.

### Conclusion:

The Jumbled Words Game project aims to create an engaging and educational word puzzle game. By implementing features like word generation, gameplay mechanics, user interface, hints, levels, and progression, the game will provide an entertaining and challenging experience. The project will leverage programming languages and frameworks to develop the game's

## 5.3 MODULE DESCRIPTION

### tkinter:

* + The tkinter module provides a Python interface to the Tk GUI toolkit.
  + It allows the creation of graphical user interfaces (GUIs) for desktop applications.
  + The module includes classes, functions, and methods to create and manage various GUI elements such as windows, buttons, labels, entry fields, etc.

### random:

* + The random module provides functions for generating random numbers and selecting random items from sequences.
  + It includes functions like random(), randint(), choice(), shuffle(), etc., which are useful for implementing randomization and generating random elements in a program.

### messagebox:

* + The messagebox module is part of the tkinter library and provides a set of functions for displaying pop-up message boxes.
  + It allows the display of different types of message boxes, including information, warning, error, and question boxes.
  + These message boxes can be used to show messages, ask for user input, or display notifications to the user.

### Label:

* + The Label class is a widget in the tkinter module used to display text or images on a window or frame.
  + It provides various options to customize the appearance of the text, such as font, color, alignment, etc.
  + Labels are commonly used to display static text or provide information to the user.

### Entry:

* + The Entry class is a widget in the tkinter module used to create a single-line input field.
  + It allows users to enter and edit text or data in a GUI application.
  + The Entry widget supports various options and methods for controlling the input and retrieving the entered value.

### Button:

* + The Button class is a widget in the tkinter module used to create a clickable button in a GUI application.
  + It provides options to customize the appearance of the button, such as text, font, color, size, etc.
  + Buttons can be associated with functions or methods to perform specific actions when clicked.

### StringVar:

* + The StringVar class is a special variable type in the tkinter module used to store and manipulate string values.
  + It is commonly used to associate a variable with an Entry widget to retrieve or set its value.
  + StringVar objects can be traced, allowing for automatic updates of associated widgets when the variable value changes.

### Geometry Management:

* + The root.geometry() method sets the dimensions and position of the root window on the screen.
  + It allows customization of the window's width, height, and position by specifying the desired values.
  + The geometry is defined in the format "widthxheight+xposition+yposition".

### Title and Background Configuration:

* + The root.title() method sets the title text for the root window.
  + It allows customization of the window's title displayed in the title bar.
  + The root.configure() method is used to configure the background and other attributes of the root window.

### mainloop:

* + The root.mainloop() method is the main event loop of the tkinter application.
  + It continuously listens for events and updates the GUI accordingly.
  + The mainloop() function keeps the window active and responsive to user interactions until the window is closed.

### Pack Geometry Manager:

* + The Pack geometry manager in tkinter is a simple layout manager that allows widgets to

be organized in a top-to-bottom or left-to-right fashion. It is used to pack widgets within a window or frame. The Pack() method is called on each widget to specify its placement and stacking order.

### Event Handling:

* + user actions or system events. For example, in the jumbled word game, the Check button has a command associated with it that calls the checkans() function when clicked. This function handles the logic for checking the answer and displaying appropriate messages. By defining event handlers, developers can create interactive applications that Event handling in tkinter involves defining functions or methods that are triggered in response to respond to user input.

### Score Tracking:

* + The jumbled word game keeps track of the player's score using variables and labels. The variables 'c' and 'd' store the correct answers and the total attempts, respectively. The score is displayed using the Label widget, which is updated dynamically with the current score information. This allows players to see their progress and encourages them to continue playing.

### Reset Functionality:

* + The game includes a reset functionality that allows the player to start a new round. The reset() function is called when the Reset button is clicked. It generates a new random word, clears the input field, and resets the score variables. This feature provides a way for the player to restart the game and attempt new jumbled words.

### Visual Customization:

* + The tkinter components, such as labels, buttons, and entry fields, can be customized in terms of their appearance. Properties like font, color, size, and alignment can be adjusted to create visually appealing interfaces. In the jumbled word game, the Label, Entry, and Button widgets are customized with specific fonts, colors, and sizes to enhance the game's visual presentation.

### User Feedback:

* + The game provides user feedback through message boxes. When the player submits an answer, a message box is displayed informing them whether their answer is correct or incorrect. This feedback enhances the gaming experience by providing instant information and enabling the player to learn from their attempts.
  + By leveraging these features and functionalities provided by the modules and components, the jumbled word game becomes an engaging and interactive experience for the players.

### Random Word Selection:

* + The game uses the `random` module to select a random word from the `words` list. The

`random.randrange()` function is used to generate a random index within the range of the list. This ensures that each time the game is played, a different jumbled word is presented to the player, adding variety and challenge to the game.

### Answer Checking:

* + The `checkans()` function compares the player's input with the correct answer for the jumbled word. If the answer matches, a congratulatory message is displayed using the

`showinfo()` function from the `messagebox` module. If the answer is incorrect, an error message is shown using the `showerror()` function. This provides immediate feedback to the player, helping them evaluate their progress and accuracy.

### Score Calculation:

* + The game keeps track of the player's score by incrementing the `c` variable whenever a correct answer is given and incrementing the `d` variable for every attempt, correct or incorrect. The score is displayed as a string using the `str()` function and is updated dynamically in the `Label` widget. This allows players to see their score in real-time and encourages them to improve their performance.

### Responsive GUI:

* + The game's graphical user interface (GUI) is built using tkinter, which provides a responsive and interactive experience. GUI elements such as labels, buttons, and entry fields are designed to be intuitive and easy to use. The use of event handlers ensures that the game responds to user actions in a timely manner, enhancing the overall gameplay experience.

### Game Reset:

* + The `reset()` function allows the player to reset the game by generating a new jumbled word, clearing the input field, and resetting the score variables. This feature enables players to play multiple rounds without having to restart the entire game. It adds

replayability and encourages players to continue playing and improving their performance.

### Visual Design:

* + The game's GUI is visually designed using a combination of font styles, colors, and layout arrangements. The choice of fonts and colors contributes to the game's aesthetic appeal and readability. The layout of GUI elements, such as labels and buttons, is organized to provide a clean and intuitive interface, making it easy for players to navigate and interact with the game.

### Cross-Platform Compatibility:

* + The jumbled word game built with tkinter is cross-platform compatible, meaning it can run on different operating systems such as Windows, macOS, and Linux. This allows the game to reach a wider audience and ensures that players can enjoy the game regardless of their preferred operating system.

### Educational Value:

* + Jumbled word games are not only entertaining but also have educational benefits. They can improve vocabulary, language skills, and cognitive abilities. By engaging players in solving jumbled words, the game promotes learning in a fun and interactive way.

**DATA FLOW DIAGRAM:**

Start

Display score

Repeat until user quits End

Update score

NO

Generate

Random Word

Display Jumbled Word

Check

User Input

Get User Input

Initialize variable

YES

# 6.SYSTEM DESIGN AND IMPLEMENTATION

### Introduction:

The Jumbled Words Game is a word puzzle that challenges players to unscramble jumbled words and find the correct answers. To create an effective and engaging game, a well-designed system architecture and careful implementation are crucial. This section will provide a comprehensive overview of the system design and implementation aspects of the Jumbled Words Game.

### System Design:

1. **User Interface:**
   * The game should have an intuitive and visually appealing user interface (UI). It should include components such as a word display area, input box for user answers, buttons for actions like checking answers and resetting the game, and a score display.
   * The UI can be designed using a GUI framework like Tkinter, PyQt, or wxPython. This allows for easy creation of windows, buttons, labels, and other UI elements.

### Word Database:

* + The game requires a database of jumbled words and their corresponding correct answers. The database can be stored in a file or a database management system (DBMS) like SQLite or MySQL.
  + The database should include a large collection of words of varying difficulty levels. Each word should have its jumbled form and the correct unscrambled answer.

### Game Logic:

* + The game logic involves selecting a random word from the word database and presenting it to the player in its jumbled form.
  + The player's input needs to be validated against the correct answer. If the answer is correct, the player's score increases; otherwise, appropriate feedback is provided.
  + The game logic should handle actions such as resetting the game, displaying scores, and

providing hints or clues if desired.

### Score Tracking:

* + The system should include a mechanism to track and store players' scores. This can be implemented using variables or by storing scores in a file or a database.
  + The scoring mechanism can assign points based on factors such as the number of correct answers, time taken to solve each word, and bonus points for consecutive correct answers.

### Difficulty Levels:

* + To provide a varied and challenging gameplay experience, the game can include multiple difficulty levels.
  + Difficulty levels can be determined based on factors such as word length, complexity of jumbling, and rarity of words.
  + The system should have the flexibility to adjust the difficulty level based on the player's preferences or progression in the game.

### Hint System:

* + A hint system can be implemented to assist players who may struggle with certain words. Hints can provide partial solutions, synonyms, or definitions related to the jumbled word.
  + The hint system can be designed to have limited uses or to require the player to earn hints through achievements or scoring milestones.

### Implementation:

1. **User Interface:**
   * Utilize a GUI framework like Tkinter to create the game's UI.
   * Design and implement the UI components, such as labels, buttons, and entry boxes, using the framework's features and functions.
   * Ensure proper layout and alignment of UI elements to create an aesthetically pleasing and user-friendly interface.

### Word Database:

* + Create or import a database of jumbled words and their corresponding correct answers.
  + Implement a mechanism to randomly select a word from the database for each game session.
  + Consider organizing the database in a structured format, such as a CSV file, JSON file, or a relational database, for efficient retrieval of words and answers.

### Game Logic:

* + Develop the game logic to handle the main functionalities, such as selecting a word, validating user input, and providing feedback.
  + Implement the scoring mechanism to track and update the player's score.
  + Include functions for actions like resetting the game, displaying scores, and providing hints if applicable.

### Score Tracking:

* + Create variables or data structures to store and update the player's score.
  + Implement functions or methods to calculate and display the score on the UI.
  + Determine the scoring system based on the game's rules and assign points accordingly.

### Difficulty Levels:

* + Design and implement a mechanism to set and adjust the difficulty level.
  + Define rules for each difficulty level, such as the range of word lengths, complexity of jumbling, and frequency of rare or common words.
  + Modify the word selection process to adhere to the chosen difficulty level.

### Hint System:

* Implement a hint system that provides assistance to players.
* Create functions or methods to generate hints based on predefined criteria, such as revealing a few letters or providing synonyms.
* Integrate the hint system with the UI, allowing players to request hints when needed.

### Conclusion:

The successful implementation of the Jumbled Words Game requires careful system design and meticulous implementation. By considering factors such as user interface design, word database management, game logic, score tracking, difficulty levels, and hint systems, developers can create an engaging and enjoyable game. With a well-designed system architecture and efficient implementation, the Jumbled Words Game can provide an entertaining and intellectually stimulating experience

# CONCLUSION

The Jumbled Words Game is a captivating word puzzle that combines entertainment with education, providing players with an enjoyable and challenging experience. Throughout this comprehensive egame, ved into its features, mechanics, and the various benefits it offers to players of all ages. As we conclude, let us reflect on the game's significance, impact, and potential for personal growth.

First and foremost, the Jumbled Words Game serves as an effective tool for expanding one's vocabulary. By presenting players with jumbled words, the game encourages them to engage in word unscrambling, deciphering hidden meanings, and discovering new words. Through this process, players encounter unfamiliar terms, allowing them to broaden their lexicon and improve their overall language proficiency. This vocabulary expansion is not limited to specific domains but encompasses a wide range of words, enabling players to enhance their communication skills in various contexts.

Moreover, the Jumbled Words Game promotes the improvement of spelling abilities. As players unjumble words, they must pay close attention to letter combinations, identify correct word structures, and apply their understanding of spelling rules. By repeatedly engaging in this process, players reinforce their knowledge of proper spelling and develop a heightened sensitivity to spelling patterns and conventions. Consequently, these improved spelling skills transfer to other areas of written communication, enhancing written expression and accuracy in everyday life.

Beyond vocabulary and spelling, the Jumbled Words Game fosters cognitive development by stimulating critical thinking, problem-solving, and analytical skills. Players must analyze the jumbled letters, search for meaningful patterns, and mentally rearrange them to unveil the correct word. This mental exercise challenges memory, concentration, and cognitive flexibility. Regular engagement with the game can enhance cognitive abilities such as memory recall, pattern recognition, and strategic thinking. These skills are not only valuable in word puzzles but are transferable to other academic disciplines and real-life situations.

One notable aspect of the Jumbled Words Game is its potential for customization and adaptability. The game can be tailored to suit different skill levels, ensuring an inclusive experience for players of varying abilities. Beginners can start with simpler jumbled words, gradually progressing to more challenging puzzles as they build confidence and proficiency. Additionally, the game's optional features, such as time limits and hint systems, further contribute to its adaptability, allowing players to customize the gameplay based on their preferences and desired level of difficulty.

Furthermore, the Jumbled Words Game offers numerous benefits beyond pure intellectual development. Its interactive and engaging nature makes it an excellent recreational activity for individuals, families, and friends. Playing the game together not only promotes social bonding but also encourages healthy competition and a sense of achievement. Additionally, the game's appeal spans across generations, making it an ideal choice for educational institutions, language learning centers, and even casual gatherings.

In educational settings, the Jumbled Words Game can be employed as an effective teaching tool. Its ability to combine learning with entertainment makes it an engaging method for reinforcing vocabulary, spelling, and language skills. Educators can incorporate the game into their lesson plans, creating an interactive and dynamic learning environment that captivates students and enhances their linguistic abilities. The game's adaptability and customizable features allow teachers to align the challenges with the curriculum, ensuring a meaningful and effective learning experience.

In conclusion, the Jumbled Words Game is a remarkable fusion of entertainment, learning, and cognitive development. Through unscrambling jumbled words, players embark on an intellectually stimulating journey that expands their vocabulary, enhances spelling proficiency, and boosts cognitive skills. Beyond its educational value, the game promotes social interaction, encourages healthy competition, and provides a source of amusement for individuals and groups alike. Whether as a recreational pastime, a teaching aid, or a means of personal growth, the

Jumbled Words Game offers a rewarding and enjoyable experience that nurtures linguistic abilities and stimulates the mind.

**1. Multiple Game Modes**:

# FUTURE ENHANCEMENT

* + Time Attack Mode: Introduce a mode where players race against the clock to solve as many jumbled words as possible within a given time limit. This adds a sense of urgency and challenges players to think quickly.
  + Multiplayer Mode: Implement a multiplayer feature that allows players to compete against each other in real-time. They can either race to solve the same set of jumbled words or take turns challenging each other with new puzzles.

### Levels of Difficulty:

* + Easy, Medium, and Hard Levels: Introduce different difficulty levels to cater to players of varying skill levels. Easy levels may contain common words with simpler jumbles, while hard levels could include more complex words with challenging arrangements.
  + Customizable Difficulty: Allow players to adjust the difficulty settings based on their preferences. They can choose the length of the words, the complexity of the jumbles, or even specific word categories they want to focus on.

### Word Categories and Themes:

* + Categorize Words: Group jumbled words into different categories such as animals, fruits, countries, or sports. This adds variety and allows players to focus on specific topics of interest.
  + Themes and Visuals: Enhance the game's visual appeal by incorporating themes related to the word categories. For example, using nature-themed backgrounds for words related to plants or landscapes.

### Hint System and Power-Ups:

* + Hint System: Provide players with the option to receive hints when they are stuck on a jumbled word. Hints could include revealing a single letter, rearranging some of the letters, or providing a definition or synonym.
    - Power-Ups: Introduce power-ups that players can earn or purchase to help them in challenging situations. These power-ups could include options to shuffle the jumbled letters, skip a particularly difficult word, or gain extra time.

### Achievement and Rewards:

* + Achievement System: Implement an achievement system that rewards players for reaching specific milestones or accomplishing challenging tasks. These achievements could be based on accuracy, speed, consecutive correct answers, or completing specific word categories.
  + In-Game Rewards: Offer in-game rewards, such as virtual currency or unlockable content, that players can earn by achieving high scores or completing challenges. These rewards can be used to customize the game's appearance, unlock additional word categories, or access premium features.

### Word Database Expansion:

* + Increase Word Database: Continuously update and expand the word database to include a broader range of words, ensuring that players are presented with fresh and diverse jumbled words in every game session.
  + User-Generated Content: Allow players to contribute their own jumbled words and share them with the game's community. This user-generated content can be moderated and incorporated into future game updates.

### Statistics and Leaderboards:

* + Player Statistics: Keep track of players' performance statistics, such as the number of words solved, accuracy rate, and average time taken per word. This allows players to track their progress and improvement over time.
  + Online Leaderboards: Introduce online leaderboards where players can compete with others globally. This fosters a sense of competition and provides an opportunity for players to showcase their skills.
  + By implementing these future enhancements, the Jumbled Words Game can become even more captivating, challenging, and enjoyable for players. These

additions will provide a more personalized and dynamic experience, catering to different skill levels and preferences while encouraging continuous engagement and skill development.

* 1. APPENDIX
  2. SOURCE CODE

import tkinter

from tkinter import \* import random

from tkinter import messagebox root = tkinter.Tk()

answers=["apple","mango","banana",'achieve','kolkata','evening','servant','receiver','london','ferrari', 'hollow','horror','master','morning','bottle','pen','router','copy','narrow','wide','dive','love','block','right'

,'simple','deaf','single','knight','hope'] words=['plpea','gnoma','annaba','hveeica','lkaatko','egvnine','aestnrv','iceever','lndono','rrreifa','wllho o','oohrrr','rtemsa','nnrgimo','lttobe','enp','ourrte','ypco','rraonw','wdie','ievd','elov','klboc','ightr','plms ie','dfea','glneis','ghtkni','opeh']

num=random.randrange(0,len(words),1) c=0

d=0

s = ""

l = Label(root)

def reset():

global words, answers, num num=random.randrange(0,len(words),1) label.config(text = words[num]) e1.delete(0, END)

def default():

global words,answers,num label.config(text = words[num])

def checkans():

global words, answers, num, c, d, s, l d=int(d)+1

var = e1.get() #getting the answer

if var == answers[num]: #checking if the answer is right or not messagebox.showinfo("Congratulations", "It's the correct answer!!") c = int(c)+1

else:

messagebox.showerror("Sorry", "It's not the correct answer.") s = 'Score :' + str(c) + '/' + str(d)

l.forget()

l = Label(root, font=("Verdana", 20), text=s, bg="#000000", fg="#fff", ) l.pack(side=LEFT)

reset()

root.geometry("500x500+500+150") root.title("Jumbled word game") root.configure(background="#000000")

Label(root,text="JUMBLED WORD GAME",font = ("Verdana",28),bg = "#000000", fg = "#fff").pack(pady = 5)

label = Label(root,font = ("Verdana",22),bg = "#000000", fg = "#fff")

label.pack(pady = 30,ipady=10,ipadx=10) # created a label that will be shown on the box and the pack() helps in executing what i wrote in Label

ans = StringVar() #defining that this is a string variable e1 = Entry(root,font = ("Verdana",20),textvariable = ans,) e1.pack(ipady=5,ipadx=5) #created an input box

#ipad=internal padding in x is called ipadx and y is called ipady

Button(root,text = "Check",font = ("Comic sans ms",20),width = 10,bg="#333945",fg="#45CE30",relief = GROOVE,command = checkans,).pack(pady = 40) #created a submit button

Button(root,text = "Reset",font = ("Comic sans ms",20),width = 10,bg="#777E8B",fg="#E1DA00",relief = GROOVE,command = reset).pack() #created a reset button

default()

root.mainloop() #its like the main function

* 1. SCREEN SHOTS





* 1. **REFERENCES**

1.<https://docs.python.org/> 2.<https://docs.python.org/3/library/tkinter.html>

3. <https://stackoverflow.com/> 4. (<https://realpython.com/>)

5. <https://www.pygame.org/>