#### **LAB REPORT ON**

#### **MEMORY GAME**

#### A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

**Bachelor of Technology**

**in**

**Computer Science and Engineering (Data Science)**

Submitted by

**S.K.KAVYA SREE**

22H51A6758

Under the esteemed guidance of

**Mrs. U. HEMA LATHA**

(Assistant Professor)



**Department of Computer Science and Engineering (Data Science)**

### CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

\*Approved by AICTE \*Affiliated to JNTUH \*NAAC Accredited with A+ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

### 2022- 2026

**CMR COLLEGE OF ENGINEERING & TECHNOLOGY**

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)**



### CERTIFICATE

This is to certify that the Lab Project report entitled **"SPELL CORRECTOR"** being submitted by S.K.Kavya Sree(22H51A6758), B.Sai Preethi (23H55A6701), CH.Sai Swetha(23H55A6702) in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering (Data Science)** is a record of bonafide work carried out his/her under my guidance and supervision.

#### The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

**Mrs. U. Hema Latha Dr. L. Chandra Sekhar Reddy**

**Assistant Professor Associate Professor and HOD**

**Dept. of CSE(DS) Dept. of CSE(DS)**

**External Examiner**

### ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Mrs. U. Hema Latha**, **Assistant Professor**, Department of Computer Science and Engineering(Data Science) for his valuable technical suggestions and guidance during the execution of this projectwork.

We would like to thank **Dr. L. Chandra Sekhar Reddy**, Head of the Department of ComputerScience and Engineering(Data Science), CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Computer Science and Engineering(Data Science) for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care.

Finally, We extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

#### S.K.Kavya Sree

22H51A6758

|  |  |  |
| --- | --- | --- |
| **TABLE OF CONTENTS** | | |
| **CHAPTER**  **NO.** | **TITLE** | **PAGE NO.** |
|  |  |  |
|  | ABSTRACT | I |
| **1** | **INTRODUCTION** |  |
|  | 1.1 Problem Statement | 2 |
|  | 1.2 Objective | 2 |
|  | 1.3 Project Scope and Limitations | 3 |
| **2** | **PROPOSED SYSTEM** |  |
|  | 2.1. Advantages of Proposed System | 5 |
|  | 2.2 Implementation   * 1. Design   2. Code and output | 6  7  7-9 |
| **3** | **RESULTS AND DISCUSSION** |  |
|  | 3.1 Result And Discussion | 11 |
| **4** | **CONCLUSION**  4.1 Conclusion and Future Enhancement |  |
|  | 13 |
|  | **REFERENCES** | 15 |

**ABSTRACT**

The Python script presented here employs the Turtle graphics library to construct a captivating memory matching game with an intuitive graphical user interface. The game unfolds on a grid of square tiles, each concealing a hidden image or number. Players endeavor to uncover matching pairs by clicking on the tiles, with the graphical interface providing real-time updates on the score and a ticking timer. The game encompasses essential features, such as an aesthetically pleasing user interface crafted with Turtle graphics, dynamic game mechanics where matching pairs influence the score, and randomized initial placements of content for each gaming session. Additionally, a countdown timer introduces a temporal element, intensifying the challenge. As the timer approaches zero, a "Game Over" message is displayed, accompanied by the player's final score. This project showcases the fusion of Python programming and Turtle graphics, resulting in an interactive and visually engaging game that not only entertains but also puts players' memory and attention skills to the test. It serves as an illustrative example of leveraging Turtle graphics for the creation of uncomplicated yet entertaining games.

# CHAPTER 1

## INTRODUCTION

### CHAPTER 1 INTRODUCTION

##### Problem Statement

The Memory Matching Game project aims to address the need for a visually engaging and interactive memory-based gaming experience. The problem at hand involves designing an efficient and enjoyable memory matching game that not only tests and improves users' memory capabilities but also provides pleasing and user-friendly experience. The incorporation of randomized tile placements, a countdown timer, and real-time score updates aims to make the game more dynamic and challenging. By addressing these challenges, the project endeavors to deliver a well-rounded solution that combines the cognitive benefits of a memory game with an engaging and visually appealing user interface.

##### Research Objective

##### The research objectives for the Memory Matching Game project encompass assessing the cognitive impact of the game on memory, concentration, and pattern recognition skills through empirical studies. The study aims to analyze user engagement with the graphical user interface created using Python's Turtle graphics library, evaluating interaction patterns and overall satisfaction. Additionally, the effectiveness of the randomization strategy for tile placements will be explored to understand its role in sustaining player interest and challenging memory skills. The project also investigates the influence of temporal pressure introduced by a countdown timer on player performance, aiming to find the optimal balance between challenge and enjoyment. Ultimately, the research seeks to contribute to the formulation of design guidelines for educational memory games, providing insights into creating effective and engaging learning experiences.

* 1. **Scope**
* **Spell Cognitive Skill Enhancement:**

The project aims to contribute to cognitive skill development by providing an engaging platform for users to enhance their memory, concentration, and pattern recognition abilities through interactive gameplay.

* **Educational Value:**

The game's design incorporates educational elements, making it suitable for use in educational settings to supplement traditional learning methods. It offers a fun and interactive way for users, especially students, to reinforce cognitive skills.

* **User-Friendly Interface:**

The graphical user interface created with Turtle graphics emphasizes user-friendliness, ensuring an intuitive and visually appealing experience. This makes the game accessible to a diverse audience, including individuals with varying levels of technical proficiency.

* **Adaptability and Customization:**

The project's design allows for easy adaptation and customization. Developers can modify the game's content, difficulty levels, and visual elements to suit different educational contexts, age groups, or specific learning objectives.

* **Research and Analysis Platform:**

The project provides a platform for conducting research on the cognitive impact of memory games, user engagement with interactive interfaces, and the effectiveness of game mechanics such as randomization and time constraints. Researchers can use the game as a tool for studying cognitive psychology and human-computer interaction.

##### Limitation

* **Limited Complexity:**

The game focuses on a simple memory matching concept, and while this simplicity enhances accessibility, it might limit its appeal to users seeking more complex or varied gameplay experiences.

* **Static Content:**

The current design involves static content on tiles, which may lead to reduced replay value as users might quickly memorize the patterns after repeated gameplay sessions.

* **Graphics Dependency:**

The project heavily relies on the Turtle graphics library for its graphical user interface. This might limit its compatibility with certain systems or require additional adaptations for environments without graphical capabilities.

* **Single-Player Focus:**

The game is primarily designed for single-player interaction, and the absence of a multiplayer or collaborative mode limits its social and interactive potential.

* **Limited Educational Depth:**

While the game introduces educational elements, it may lack the depth required for in-depth learning experiences compared to more specialized educational software.

# CHAPTER 2

## PROPOSED

## SYSTEM

### CHAPTER 2

**PROPOSED SOLUTION**

##### Advantages of proposed System

* **Improved Cognitive Skill Enhancement:**

The game provides an engaging platform for users to enhance their memory, concentration, and pattern recognition skills through interactive gameplay, contributing to cognitive skill development.

* **Educational Value:**

With incorporated educational elements, the game serves as a valuable tool for reinforcing cognitive skills in an entertaining manner, making it suitable for educational settings and self-directed learning.

* **User-Friendly Interface:**

The graphical user interface, created using Python's Turtle graphics library, is designed to be intuitive and visually appealing, ensuring a user-friendly experience for players of varying technical proficiencies.

* **Adaptability and Customization:**

The game's design allows for easy adaptation and customization, enabling developers to modify content, difficulty levels, and visual elements to suit different educational contexts, age groups, or specific learning objectives.

* **Research and Analysis Platform:**

The project provides a platform for researchers to study the cognitive impact of memory games, user engagement with interactive interfaces, and the effectiveness of game mechanics such as randomization and time constraints.

* **Open-Source Potential:**

With proper documentation and code organization, the project has the potential to be open-sourced, fostering collaboration and allowing developers to contribute improvements, additional features, or alternative educational content.

##### Implementation:

### Step 1: Step 1: Importing Modules

### The project begins by importing the necessary Python modules, specifically the 'random' module for shuffling numbers and the 'turtle' module for creating the graphical user interface.

### Step 2: Hiding Turtle Cursor and Setting Up Screen

### The turtle cursor is hidden, and the screen is set up using the Turtle graphics library. The background color of the screen is chosen to be yellow for an aesthetically pleasing interface.

### Step 3: Defining the Square Function

### A function named 'Square' is defined to create a square section for the game. This function takes coordinates and uses the Turtle graphics to draw a white-filled green square at those coordinates.

### Step 4: Defining the Numbering Function

### The 'Numbering' function is created to map the coordinates of a click on the screen to an index number. This function is crucial for keeping track of the tiles and their states.

### Step 5: Defining the Coordinates Function

### The 'Coordinates' function is defined to convert an index number back into coordinates. This function aids in the visualization of the game grid on the screen.

### Step 6: Defining the Click Function

### The 'click' function is implemented to handle user interactions. It checks for matching pairs of tiles based on the user's clicks, updates the game state accordingly, and increases the score when a pair is found.

### Step 7: Drawing the Game

### The 'draw' function is designed to visually represent the game on the screen. It utilizes Turtle graphics to draw the square tiles and updates the interface to reflect the current state of the game, including the score and timer.

### Step 8: Setting Initial Timer Value and Game State

### The initial timer value is set to 80 seconds, and the initial state of the game, including tile values, visibility, and score, is initialized.

### Step 9: Shuffling Tiles and Initiating Game Loop

### The tiles containing numbers are shuffled to provide a randomized placement for each gaming session. The game loop is initiated, which listens for user clicks and continuously updates the game interface.

### Step 10: Starting the Timer Countdown

### A countdown function is implemented to decrement the timer by one second at regular intervals using the 'ontimer' method. This creates a time-constrained gaming experience.

### Step 11: Running the Main Loop

### The main loop is started, ensuring that the window remains open until the user clicks on it to close. This loop keeps the game running, allowing users to interact with the Memory Matching Game until the end of the countdown or until they choose to exit.

##### 2.3 DESIGN:

**Environment Setup:**

* Import necessary modules (random and turtle).

**Hide the turtle cursor.**

* Set the background color of the game screen to yellow.

**Square Drawing and Game Logic:**

* Square(x, y): Draws a white square at specified coordinates.
* Numbering(x, y): Calculates the index number based on x, y coordinates.
* Coordinates(count): Converts an index number to x, y coordinates.
* click(x, y): Handles user clicks, checks for matching pairs, and updates the score.
* draw(): Draws the game interface, including squares, numbers, score, and timer.

**Game Initialization and Tile Shuffling:**

* Initialize game variables: timer, game state, tiles, hide list, and score.
* Shuffle the tiles randomly.
* Turn off Turtle animation (tracer(False)).
* Set up click handling (onscreenclick(click)) and draw the initial game interface.

**Timer Countdown and Main Game Loop:**

* Start the timer countdown (countdown()).
* Enter the main game loop (mainloop()).

**Timer Countdown Functionality:**

* countdown(): Updates the timer every second and displays a game over message when the timer reaches zero.

**Game Over Message:**

* Display a game over message on a new panel with a white background, showing the final score.

**2.4 Code :**

from tkinter import \*

from textblob import TextBlob

import enchant

# Function to clear both the text entry boxes

def clearAll():

word1\_field.delete(0, END)

word2\_field.delete(0, END)

# Function to get all possible words

def all\_possible\_words():

input\_word = word1\_field.get()

d = enchant.Dict("en\_US")

possible\_words = [word for word in d.suggest(input\_word)]

suggestions = ', '.join(possible\_words)

word2\_field.insert(5, suggestions)

# Function to get a corrected word

def correction():

input\_word = word1\_field.get()

blob\_obj = TextBlob(input\_word)

corrected\_word = str(blob\_obj.correct())

word2\_field.insert(10, corrected\_word)

# Driver code

if \_\_name\_\_== "\_\_main\_\_":

root = Tk()

root.configure(background='light green')

root.geometry("400x200")

root.title("Spell Corrector")

headlabel = Label(root, text='Welcome to Spell Corrector Application', fg='black', bg="red")

label1 = Label(root, text="Input Word", fg='black', bg='dark green')

label2 = Label(root, text="Result", fg='black', bg='dark green')

headlabel.grid(row=0, column=1)

label1.grid(row=1, column=0)

label2.grid(row=3, column=0, padx=10)

word1\_field = Entry()

word2\_field = Entry()

word1\_field.grid(row=1, column=1, padx=10, pady=10)

word2\_field.grid(row=3, column=1, padx=10, pady=10)

button1 = Button(root, text="Correction", bg="red", fg="black", command=correction)

button1.grid(row=2, column=1)

button2 = Button(root, text="All Possible Words", bg="orange", fg="black", command=all\_possible\_words)

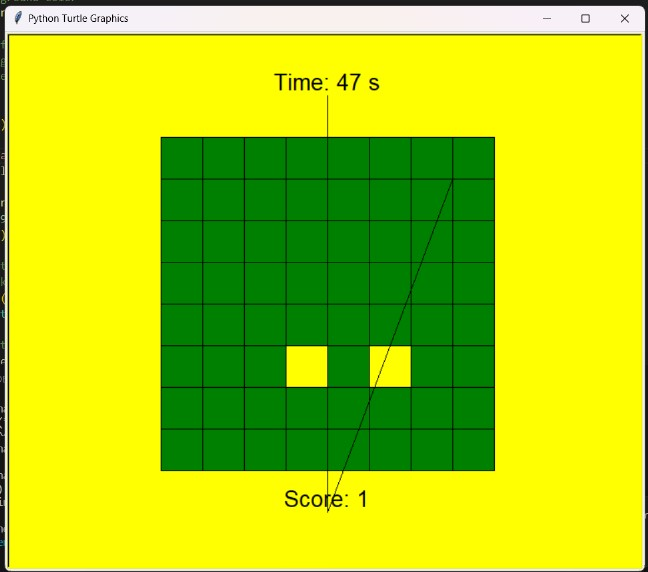
button2.grid(row=4, column=1)

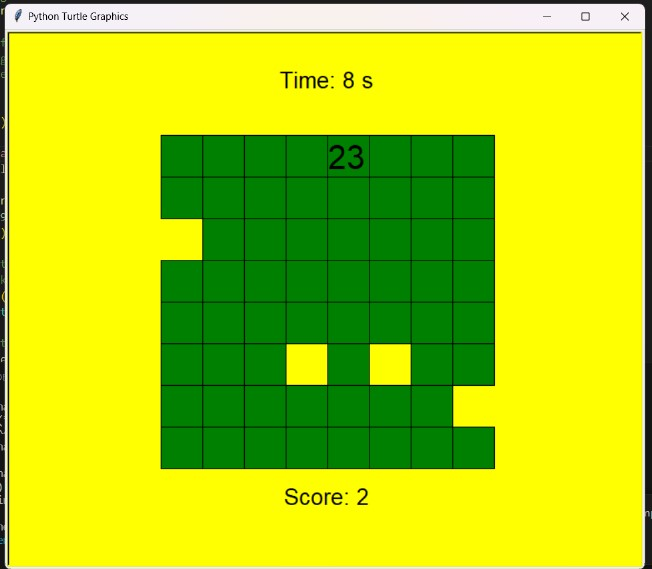
button3 = Button(root, text="Clear", bg="red", fg="black", command=clearAll)

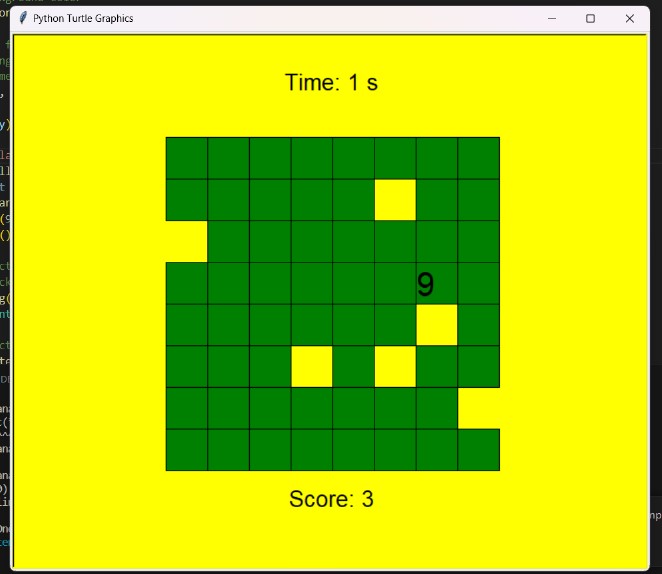
button3.grid(row=5, column=1)

root.mainloop()

**OUTPUT:**







# CHAPTER 3

## RESULTS AND DISCUSSION

### CHAPTER3

### RESULTS AND DISCUSSION

The Python code orchestrates a captivating memory matching game using Turtle graphics, featuring a grid of concealed squares with numbers. The game objective is to unveil matching pairs within a designated time frame. The Turtle cursor is concealed, and the game screen background is set to an appealing yellow.

Key functions, including Square, Numbering, Coordinates, and click, handle square drawing, index calculation, and user clicks. The central draw function manages the overall game interface, incorporating squares, numbers, the score, and the timer.

Upon initialization, variables for the timer, game state, tiles, hide list, and score are set. Random tile shuffling and the deactivation of Turtle animation (tracer(False)) enhance the gaming experience. User clicks are captured with on screen click, and the initial game interface is drawn using draw.

The game introduces an engaging countdown mechanism, urging players to match pairs within the specified time. The countdown function manages the timer, creating a sense of urgency. As the countdown concludes, the game displays the player's final score on a white-panel background, providing a conclusive end to the gaming.

# CHAPTER 4

## CONCLUSION

### CHAPTER 4

### CONCLUSION

##### Conclusion and Future Enhancement:

* + 1. **Conclusion:**

In summary, the Python code delivers a captivating memory matching game using Turtle graphics. The game challenges players to uncover matching pairs of numbered squares within a set time, fostering engagement and strategic thinking. Through well-defined functions like Square and draw, the code ensures a visually appealing interface, seamlessly updating elements such as squares, numbers, the score, and the timer. The incorporation of a countdown mechanism adds a sense of urgency, enhancing the overall gaming experience. The project successfully balances aesthetic appeal, interactivity, and strategic gameplay, offering users an entertaining and memorable engagement with the memory matching game.

##### Future Enhancement:

For future enhancements to your memory matching game project, consider implementing multiple difficulty levels, such as easy, medium, and hard, with different grid sizes and time constraints to cater to a broader audience. Introduce a high score tracking system to foster friendly competition and provide players with a goal to achieve. Enhance user customization by incorporating theme options and allowing users to personalize the game's appearance. To elevate the gaming experience, consider adding sound effects for successful matches, button clicks, and a background soundtrack, with an option for users to toggle sound settings. Introduce unique elements like power-ups or special tiles that can grant advantages during gameplay, adding a strategic layer to the experience. Lastly, ensure a responsive design that optimizes the game for various screen sizes and orientations, enhancing accessibility and user satisfaction. These enhancements collectively contribute to a more engaging, customizable, and dynamic memory matching game.

## REFERENCES

### REFERENCES

### <https://www.geeksforgeeks.org/flipping-tiles-memory-game-using-python3/>

### <https://projectgurukul.org/memory-game-in-python/>