

SYNAPTIC SEQUENCE SAGA

MINOR PROJECT REPORT

By

SRISHTI PANDA (RA2211033010146)
AASTHA SINGH (RA2211033010158)
SESHADRI PATRA (RA2211033010182)

Under the guidance of

Mrs. Indumathi

In partial fulfilment for the Course

of

21CSC203P – ADVANCED PROGRAMMING PRACTICE

in Department of Computational Intelligence



FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this minor project report for the course **21CSC203P ADVANCED PROGRAMMING PRACTICE** entitled in "**Synaptic Sequence Saga**" is the bonafide work of **Srishti Panda (RA2211033010146)**, **Aastha Singh (RA2211033010158)** and **Seshadri Patra (RA2211033010182)** who carried out the work under my supervision.

SIGNATURE

Mrs .Indumathi

Teaching Associate

Department of Computational Intelligence

SRM Institute of Science and Technology

Kattankulathur

ABSTRACT

The project embarks on a journey into the intricate realm of memory enhancement and cognitive exploration. This is an area where memory, concentration, and entertainment converge in a delightful digital experience. This is a type of a Memory Puzzle Game which are exclusively designed to put your memory prowess to the test, challenging you to remember and match pairs of images, numbers, or symbols within a set of cards. And as you flip over cards one by one, the clock is ticking, and the challenge lies in remembering the positions of the cards you've seen and strategically uncovering their counterparts.

How the Game works:

The game board consists of sixteen cards arranged in a grid. The deck is made up of eight different pairs of cards, each with different symbols on one side. The cards are arranged randomly on the grid with the symbol face down. The gameplay rules are very simple: flip over two hidden cards at a time to locate the ones that match!

Each turn:

- The player flips one card over to reveal its underlying symbol.
- The player then turns over a second card, trying to find the corresponding card with the same symbol.

If both the cards matches then it's a match otherwise try again.

- **Memory Improvement:** The core objective is to enhance players' memory skills. By repeatedly engaging with the game and practicing the recall of various patterns and pairs, players can gradually improve their memory capacity and ability to remember visual information.

- **Pattern Recognition:** Memory puzzle games train players to recognize patterns and associations.
- **Concentration and Focus:** Players must maintain their concentration and focus throughout the game to remember card positions accurately.
- **Stress Relief:** Engaging with memory puzzle games can offer stress relief by diverting the mind from daily worries and tasks.
- **Entertainment and Relaxation:** While the primary objectives are cognitive in nature, memory puzzle games also aim to provide entertainment and relaxation.

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy**, for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal**, for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman**, for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professors Dr. Vadivu. G , Professor, Department of Data Science and Business Systems and Dr. Sasikala. E Professor, Department of Data Science and Business Systems and Course Coordinators** for their constant encouragement and support.

We are highly thankful to our my Course project Faculty **Mrs.Indumathi, Teaching Associate , Department of Computational Intelligence**, for his/her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HoD, Dr.R.Annie Uthra, Professor, Department of Computational Intelligence** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

TABLE OF CONTENTS

CHAPTER NO	CONTENTS	PAGE NO
1	INTRODUCTION	7
	1.1 Motivation	7
	1.2 Objective	7-8
	1.3 Problem Statement	8
	1.4 Challenges	8
2	LITERATURE SURVEY	9
3	REQUIREMENT	10
	ANALYSIS	
4	ARCHITECTURE &	11
	DESIGN	
5	IMPLEMENTATION	12
6	EXPERIMENT RESULTS	13-15
	& ANALYSIS	
7	CONCLUSION	16
8	REFERENCES	16

1. INTRODUCTION

Memory Puzzle Games – where memory, concentration, and entertainment converge in a delightful digital experience. Memory Puzzle Games are designed to put your memory prowess to the test, challenging you to remember and match pairs of images, numbers, or symbols within a set of cards. As you flip over cards one by one, the clock is ticking, and the challenge lies in remembering the positions of the cards you've seen and strategically uncovering their counterparts.

Motivation:-

- **Cognitive Enhancement:** Memory puzzle games provide a fun and engaging way to exercise and enhance memory skills
- **Challenge and Achievement:** Human nature thrives on challenges and achievements. Memory puzzle games offer a structured challenge where players can set goals, strive for high scores, and experience a sense of accomplishment when they improve their performance or beat their previous records
- **Competition and Social Interaction:** Many memory puzzle games offer multiplayer modes or leaderboards that allow players to compete with friends, family, or players from around the world. This competitive element adds an extra layer of motivation as players strive to outperform others and showcase their memory skills
- **Education and Learning:** Memory puzzle games are also used in educational settings to enhance learning and retention of information. They can be particularly effective for teaching concepts that require visual recognition and recall.

Idea/Objective:-

- **Memory Improvement:** The core objective is to enhance players' memory skills. By repeatedly engaging with the game and practicing the recall of various patterns and pairs, players can gradually improve their memory capacity and ability to remember visual information.
- **Pattern Recognition:** Memory puzzle games train players to recognize patterns and associations.
- **Concentration and Focus:** Players must maintain their concentration and focus

throughout the game to remember card positions accurately.

- **Stress Relief:** Engaging with memory puzzle games can offer stress relief by diverting the mind from daily worries and tasks.
- **Entertainment and Relaxation:** While the primary objectives are cognitive in nature, memory puzzle games also aim to provide entertainment and relaxation.

Challenges faced:-

1. **Complexity of Memory Concepts:-** Abstract Nature: Memory management concepts, such as allocation, deallocation, and fragmentation, can be abstract and challenging for learners to grasp. Designing a game that effectively communicates these concepts in an intuitive manner is a significant challenge.
2. **Educational Effectiveness:-** Balancing Fun and Education: Striking the right balance between creating an engaging game and ensuring educational effectiveness can be challenging. The game needs to be enjoyable to keep players interested while effectively teaching memory management principles.
3. **Graphics and Design:**
 - **Artistic Design:** Creating visually appealing and cohesive graphics, animations, and user interfaces can be challenging. Achieving a balance between aesthetics and performance is crucial.
 - **Consistency:** Maintaining a consistent art style and theme throughout the game can be challenging, especially when multiple team members are involved in the design process.
4. **Gameplay Mechanics:-** Balancing: Balancing the difficulty, pacing, and overall gameplay to keep players engaged without making the game too easy or too challenging is a continuous challenge.

5. LITERATURE SURVEY

1. Search Academic Databases:- Utilize databases such as IEEE Xplore, ACM Digital Library, PubMed, and Google Scholar to search for academic papers, articles, and surveys related to memory management games.

2. Keywords:- Use relevant keywords and phrases in your search, such as "memory management game," "educational games for memory management," "simulation for memory allocation," or "computer science learning games."

3. Filter by Publication Type:- Focus on academic journals, conference proceedings, and books. These sources are more likely to contain literature reviews, surveys, and in-depth analyses.

4. Review Relevant Journals:- Look into journals that publish articles related to computer science education, game-based learning, and software engineering. Examples include the Journal of Educational Technology & Society, Computers & Education, and the International Journal of Computer Applications.

5. Explore Conference Proceedings:- Conferences in the field of computer science and education often publish proceedings that include research on memory management games. Check proceedings from conferences such as SIGCSE (Special Interest Group on Computer Science Education) and relevant IEEE conferences.

6. Check Books and Book Chapters:- Some comprehensive reviews or surveys may be found in books or book chapters. Look for literature that discusses the intersection of education, gaming, and memory management.

7. Consult Education Research Databases:- Explore databases specifically focused on education research, such as ERIC (Education Resources Information Center), to find studies related to educational games and their effectiveness in teaching memory management.

6. REQUIREMENTS

3. Software Requirement Analysis

From the given scenario, we draw the following requirements in **GUI**:

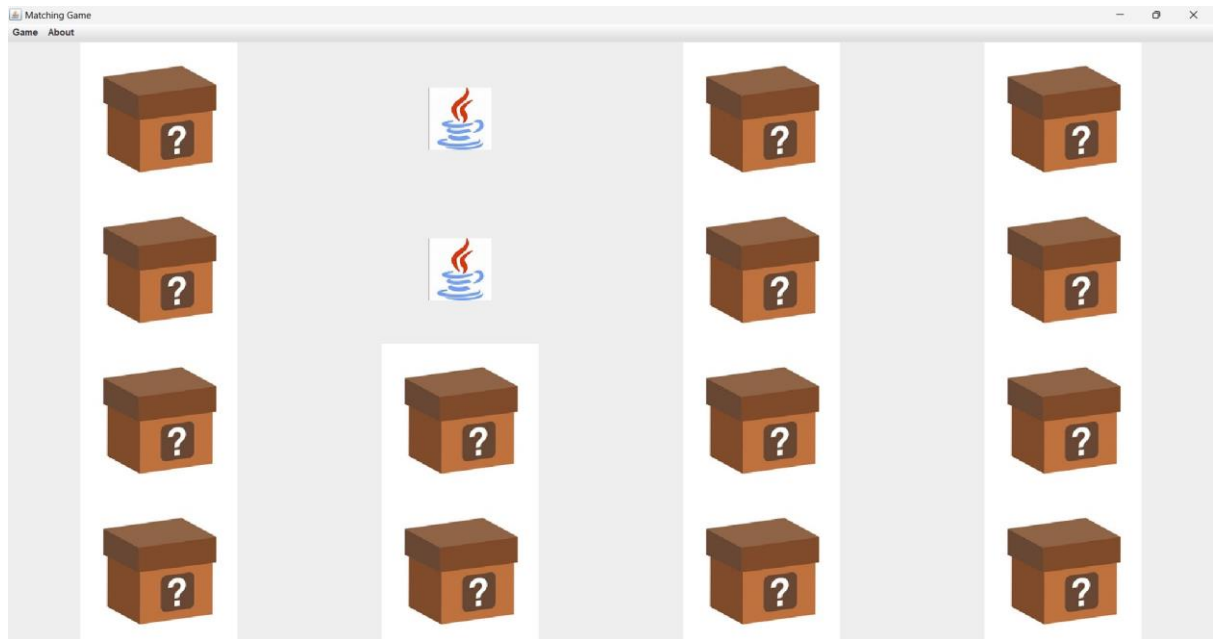
1. **Graphics and User Interface:** Define the graphical elements of the game, including characters, scenes, and user interface components.
2. **Game Logic:** Implement the logic of the game, including how the game progresses, user interactions, and scoring.
3. **Event Handling:** Set up mechanisms to handle user inputs and events within the game.
4. **Game Loop:** Implement a game loop that continuously updates the game state and redraws the screen.

From the given scenario, we draw the following requirements in **JDBC**:

- 1) **JDBC Driver:** Include the JDBC driver for the database you are connecting to. Different databases have different JDBC drivers (e.g., MySQL, PostgreSQL, Oracle).
- 2) **Connection Establishment:** Write code to establish a connection to the database using the JDBC driver and provide necessary connection details (URL, username, password).
- 3) **Query Execution:** Implement methods to execute SQL queries and updates against the database. This includes SELECT statements to retrieve data and INSERT, UPDATE, DELETE statements to modify data.
- 4) **Exception Handling:** Implement proper exception handling to manage potential errors during database interactions.
- 5) **Resource Management:** Ensure that resources like database connections, statements, and result sets are properly closed to prevent memory leaks.

7. ARCHITECTURE AND DESIGN

The game is as follows:



The architecture consists of three major parts:

- Player
- Cards
- Result

8. IMPLEMENTATION

Game theory is a branch of mathematics that studies strategic interactions between different players. Memory management, on the other hand, is a critical aspect of computer science that involves keeping track of each byte in a computer's memory and managing the allocation and deallocation of memory space as needed by different programs. To implement memory management using game theory, you would need to define the players, strategies, and payoffs in the context of memory allocation.

Here's an overview of how you might use JDBC in a memory game like Synaptic Sequence Saga:

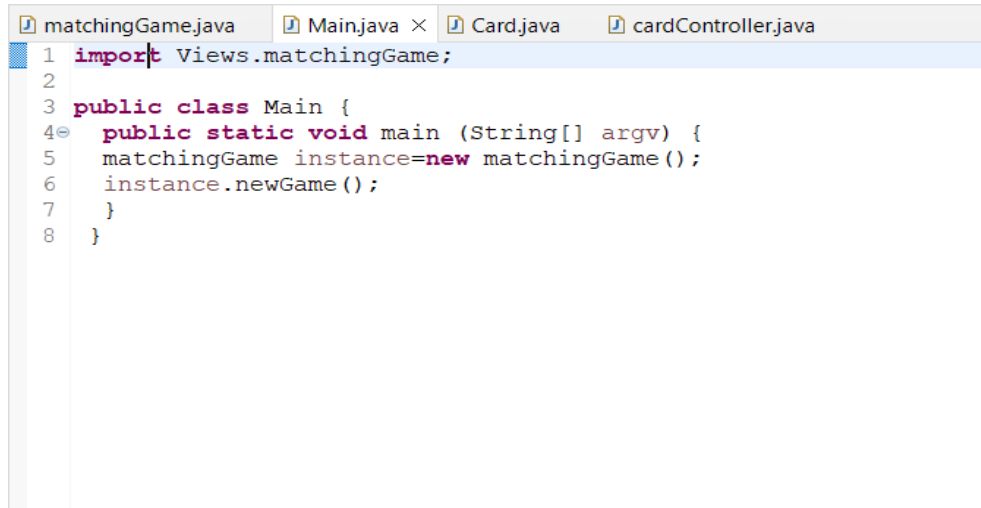
1. **Database Structure:** You would need a database to store the game state. This could be a simple table with columns for the game ID, the current state of the game, the player's score, and any other relevant information.
2. **Game State Management:** The game state could be represented as a string or a binary blob, depending on the complexity of your game. This state would be updated every time the player makes a move.
3. **Connection Management:** You would use JDBC to connect to your database. This involves loading the appropriate JDBC driver, creating a connection using a connection string, and then creating statements to execute SQL queries.
4. **Data Access Objects (DAOs):** You would create DAOs (Data Access Objects) to handle the database operations related to your game. This could include methods to get the current state of a game, save the current state of a game, update the player's score, etc.
5. **Error Handling:** You would need to handle SQL exceptions that might occur when executing queries. This could involve retrying the operation, logging the error for later analysis, or showing an error message to the player.
6. **Performance Considerations:** Depending on the number of players and the complexity of your game, you might need to consider performance. This could involve using prepared statements to speed up frequently executed queries or using a connection pool to manage your database connections.

Creating a memory game like "Synaptic Sequence Saga" involves a graphical user interface (GUI) for the game board, card flipping animations, logic for checking matches, and managing the game state. This code creates a simple memory-matching game with a grid of cards. Images are loaded and shuffled, and the player can flip two cards at a time to find matching pairs. If a pair is found, the cards stay flipped; otherwise, they flip back after a short delay. The game continues until all pairs are matched.

9. RESULTS AND DISCUSSION

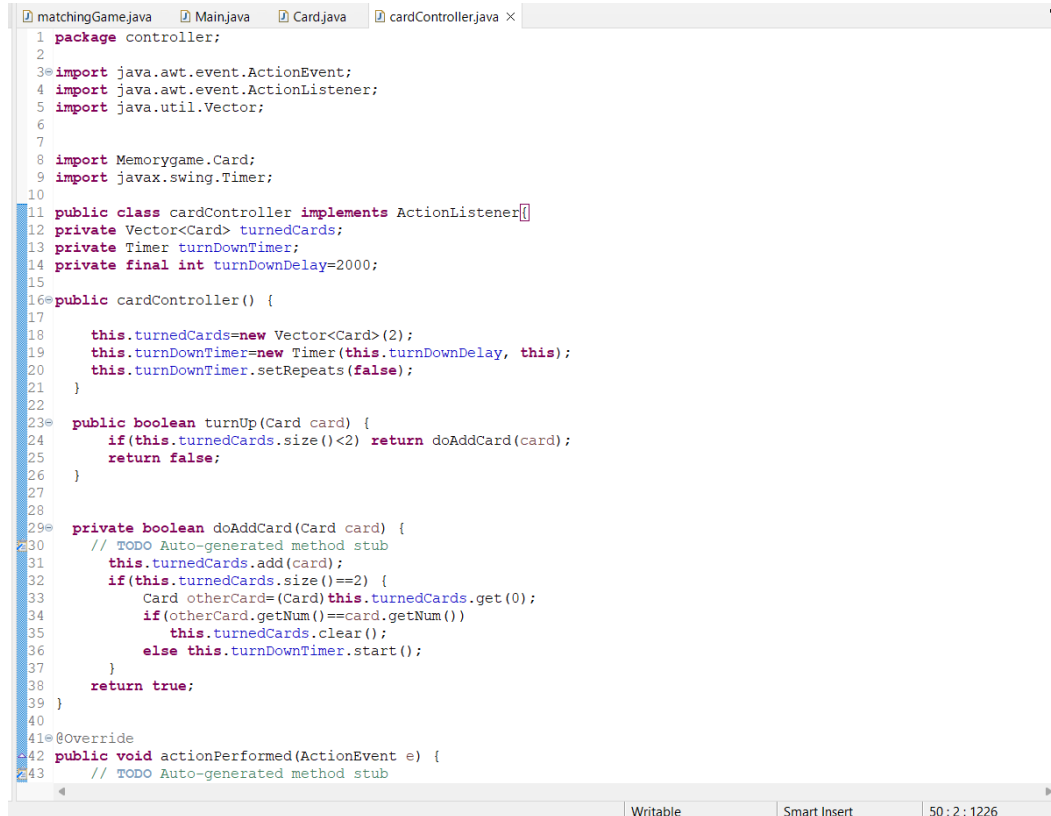
1.Code snippets in GUI:-

- The synaptic sequence saga was run by main class:



```
1 import Views.matchingGame;
2
3 public class Main {
4     public static void main (String[] argv) {
5         matchingGame instance=new matchingGame();
6         instance.newGame();
7     }
8 }
```

- The game was controlled by cardController Class:-

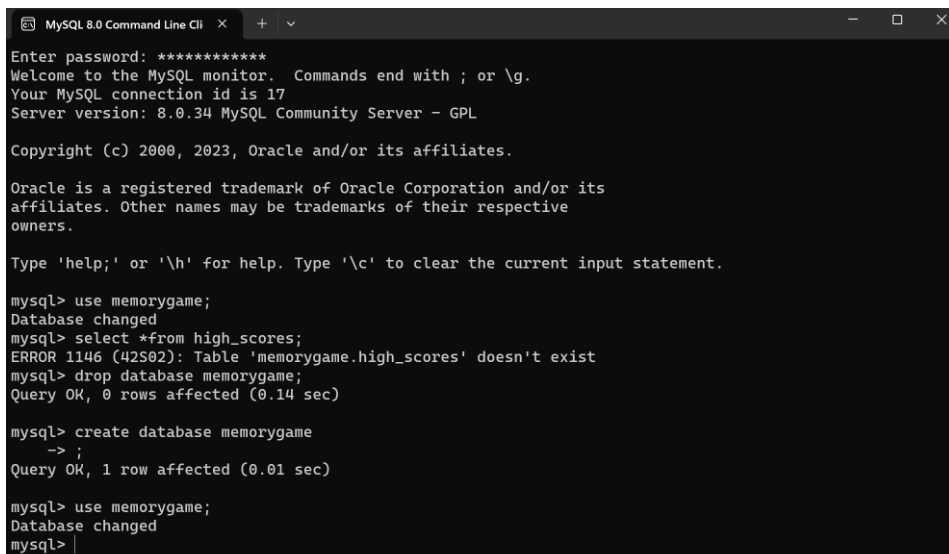


```
1 package controller;
2
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import java.util.Vector;
6
7 import Memorygame.Card;
8 import javax.swing.Timer;
9
10
11 public class cardController implements ActionListener{
12     private Vector<Card> turnedCards;
13     private Timer turnDownTimer;
14     private final int turnDownDelay=2000;
15
16     public cardController() {
17
18         this.turnedCards=new Vector<Card>(2);
19         this.turnDownTimer=new Timer(this.turnDownDelay, this);
20         this.turnDownTimer.setRepeats(false);
21     }
22
23     public boolean turnUp(Card card) {
24         if(this.turnedCards.size()<2) return doAddCard(card);
25         return false;
26     }
27
28
29     private boolean doAddCard(Card card) {
30         // TODO Auto-generated method stub
31         this.turnedCards.add(card);
32         if(this.turnedCards.size()==2) {
33             Card otherCard=(Card) this.turnedCards.get(0);
34             if(otherCard.getNum()==card.getNum())
35                 this.turnedCards.clear();
36             else this.turnDownTimer.start();
37         }
38         return true;
39     }
40
41     @Override
42     public void actionPerformed(ActionEvent e) {
43         // TODO Auto-generated method stub
44     }
45 }
```

2 Code snippets in JDBC:-

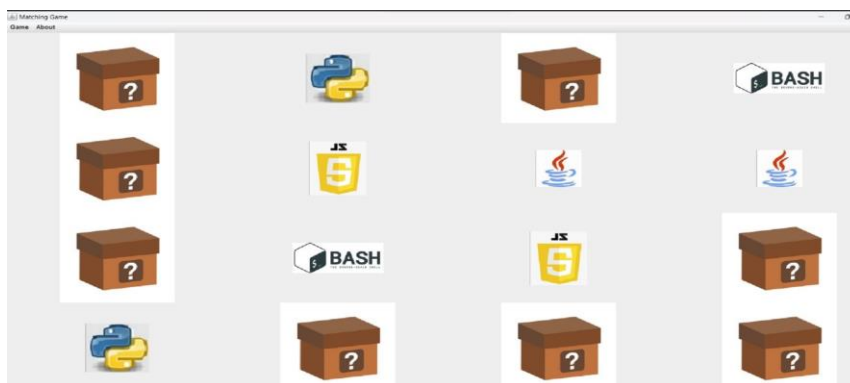
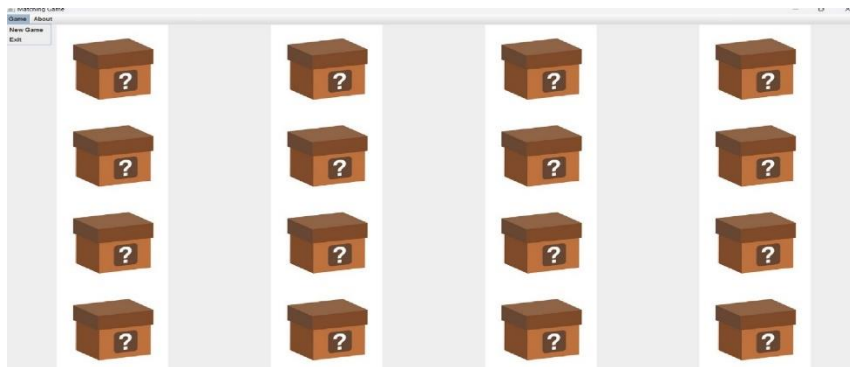
```
private void initGame() {  
    // Establish JDBC connection  
    String url = "jdbc:mysql://localhost:3306/memorygame";  
    String username = "root";  
    String password = "Sris@3991208";  
  
    try (Connection connection = DriverManager.getConnection(url, username, password)) {  
        System.out.println("Database connected!");  
  
        // Create a Statement object  
        Statement stmt = connection.createStatement();  
    }  
}
```

```
// SQL query to create a table for high scores  
String sql = "CREATE TABLE newscores (" +  
            "ID INT AUTO_INCREMENT, " +  
            "PlayerName VARCHAR(255), " +  
            "Score INT, " +  
            "PRIMARY KEY (ID))";  
  
// Execute the query  
stmt.executeUpdate(sql);  
System.out.println("newscores table created successfully");
```



```
MySQL 8.0 Command Line Cli  x  +  v  
Enter password: *****  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 17  
Server version: 8.0.34 MySQL Community Server - GPL  
  
Copyright (c) 2000, 2023, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> use memorygame;  
Database changed  
mysql> select *from high_scores;  
ERROR 1146 (42S02): Table 'memorygame.high_scores' doesn't exist  
mysql> drop database memorygame;  
Query OK, 0 rows affected (0.14 sec)  
  
mysql> create database memorygame  
-> ;  
Query OK, 1 row affected (0.01 sec)  
  
mysql> use memorygame;  
Database changed  
mysql> |
```

1.Game Results in GUI:-



2.Game results in JDBC:-

```
Problems | Javadoc | Declaration | Console | Data Source Explorer | SQL Results
<terminated> MainFrame [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (18-Oct-2023, 1:05:26 am - 1:05:43 am) [pid: 372616]
Database connected!
newscores table created successfully
```

```
Problems | Javadoc | Declaration | Console | Data Source Explorer | SQL Results
APP_Project [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (18-Oct-2023, 12:09:16 pm) [pid: 384456]
| null | null | null | null | null | null | null | null |
Enter index of first card to flip:
1
| null | C | null | null | null | null | null | null |
Enter index of second card to flip:
3
Sorry, those cards don't match.
| null |  | null |  | null | null | null | null |
Enter index of first card to flip:
```

10.CONCLUSION

A well-designed memory management game can play a crucial role in the educational landscape, offering a dynamic and engaging way for individuals to learn, practice, and master memory-related concepts. As technology evolves, integrating gamified learning experiences into computer science education can provide a valuable bridge between theory and application. Here are key points to consider:

1) **Educational Value:**

- **Reinforcement of Concepts:** A memory management game can serve as an effective tool for reinforcing theoretical concepts related to memory allocation, deallocation, and fragmentation in computer systems.
- **Hands-On Learning:** By providing an interactive environment, the game allows users to practically apply memory management principles, enhancing their understanding through hands-on experience.

2) **Skill Development:**

- **Problem-Solving Skills:** Memory management games often present challenges that require strategic thinking and problem-solving skills. Players must optimize memory usage to achieve specific goals.
- **Adaptability:** Dynamic scenarios in the game encourage players to adapt their memory management strategies, fostering adaptability and flexibility in approaching changing conditions.

3) **Real-World Application:**

- **Alignment with Industry Practices:** Understanding memory management is crucial in professional settings, especially for software developers and system administrators. A game that simulates real-world memory scenarios prepares individuals for challenges they may encounter in their careers.
- **Practical Insight:** Users gain practical insights into how their decisions impact system performance, providing a bridge between theoretical knowledge and real-world application.

11.REFERENCES

<https://www.geeksforgeeks.org/memory-game-in-java/>

<https://www.educative.io/projects/build-a-memory-game-in-java-swing>

<https://github.com/topics/memory-game?l=java>

https://en.wikipedia.org/wiki/Memory_management