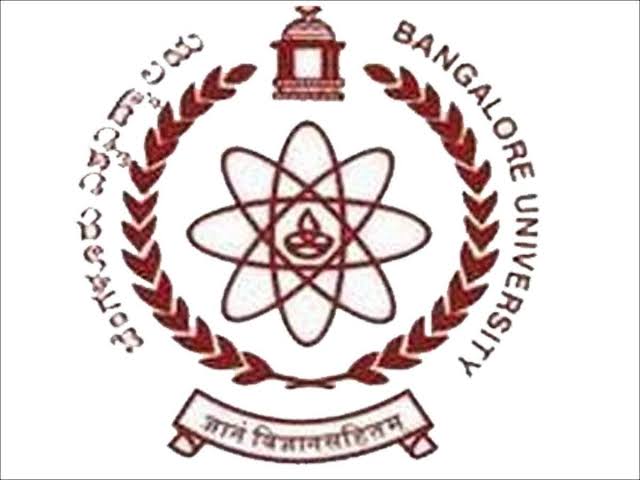
**Bangalore University**

## UNIVERSITY VISVESVARAYA COLLEGE OF ENGINEERING

**K R Circle, Bengaluru – 560001**



# Department of Computer Science and Engineering Computer Graphics Mini-Project

# on

## “MEMORY GAME”

**Submitted By**

**Aarya Subramanyam(19GANSE002) Ankur Singh(19GANSE006)**

**V SEM, B.Tech (ISE)**

**Under the Guidance of**

**Dr. Arunalatha J S Manjula S**

**Professor Guest Faculty**

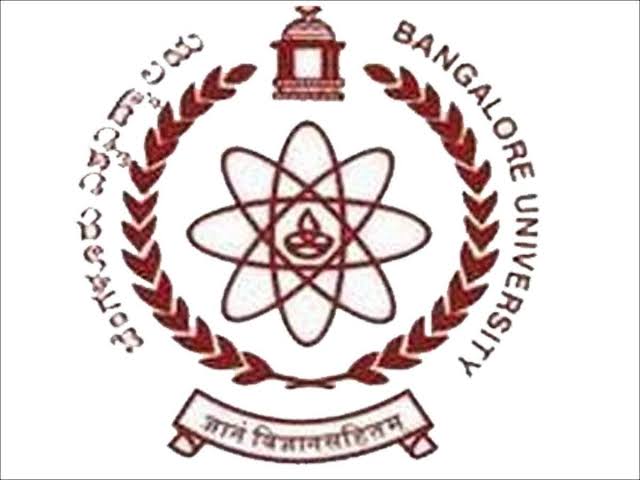
**Dept of CSE, UVCE Dept of CSE, UVCE**

**2021-2022**

**Bangalore University**

## UNIVERSITY VISVESVARAYA COLLEGE OF ENGINEERING

**K R Circle, Bengaluru – 560001**



**Department of Computer Science and Engineering**

# CERTIFICATE

This is to certify that **Mr. Aarya Subramanyam (19GANSE002)** and **Mr. Ankur Singh (19GANSE006)** students of V Semester, B. Tech, (Information Science and Engineering), has submitted the Computer Graphics Mini-Project Report on “**MEMORY GAME**”, in partial fulfillment for the Computer Graphics Lab, prescribed by the Bangalore University for the academic year 2021- 22.

**\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_**

|  |  |
| --- | --- |
| **Dr. Arunalatha J S** | **Dr. Dilip Kumar S M** |
| **Professor** | **Professor &Chairman** |
| **Department of CSE** | **Department of CSE** |
| **UVCE** | **UVCE** |

**Examiner 1: \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ Examiner 2: \_ \_ \_ \_ \_ \_ \_ \_ \_\_ \_**

# ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task is incomplete without the mention of people who made it possible, whose constant guidance and encouragement crown all efforts and success.

We are grateful to acknowledge the hon’ble Vice Chancellor of Bangalore University **Dr. Venugopal K R**, for his valuable suggestions and relentless support, which has sustained us throughout the course of the project.

We express our gratefulness to **Dr. H N Ramesh**, Principal, UVCE who has been leading our college with a brighter vision in technical education.

We express our profound gratitude to **Dr. Dilip Kumar S M**, Chairman, Department of Computer Science and Engineering, UVCE who has been a constant source of inspiration to the students work.

With extreme honour we express our deep sense of gratitude to our guide **Dr. Arunalatha J S**, Professor, Department of Computer Science and Engineering, UVCE for her valuable guidance and supervision in this course of project.

We also thank our guide **Manjula S**, Department of Computer Science and Engineering, UVCE.

We express our sincere thanks to all teaching and non-teaching staff, Department of Computer Science and Engineering, UVCE for all the facilities that they provided us to successfully complete this project.

We also thank our parents and friends for their continuous support and encouragement.

**Aarya Subramanyam**

**19GANSE002**

**Ankur Singh**

**19GANSE006**

# ABSTRACT

This report discusses the Computer Graphics project “Memory Game”. It also deals with the different functions used in the program to make the program execute successfully. It attempts to highlight OpenGL and GLUT. This report also describes the general theory for Drawing in OpenGL and other relevant topics. A graphics-based memory game is a great start for a student who starts learning computer graphics & visualization**.** The development of the game has large scope to learn computer graphics from scratch. This report lists the limitations in the project. It also tries to deal with the improvements that can be made in this project to make its graphics more detailed and finer.

# CONTENTS

|  |  |  |
| --- | --- | --- |
| **1** | **Introduction** | **1** |
| **2** | **Requirement Specifications** | **4** |
| **3** | **Design and Implementation** | **6** |
| **4** | **Results** | **9** |
| **5** | **Conclusions and Future Enhancement** | **14** |

**6 References 15**

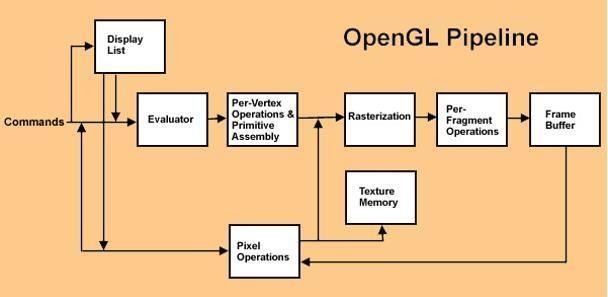
III

# CHAPTER 1

# INTRODUCTION

This report contains implementation of “Memory Game” using a set of OpenGL functions. The objects are drawn by using GLUT functions. This project has been developed using windows 10 (codeblocks) with OpenGL package. It is a game in which a few blocks have objects with different colors hidden inside. Move the focus on to an object and press enter to uncover it. As soon as you move to another block this uncovered block gets covered. You have to uncover two blocks with same object hidden behind to clear both the blocks. This is a time based game. Faster the board is cleared the higher the points.

OpenGL is considered a cross-language that provides us with a large set of

functions that can be used to manipulate graphics and images. OpenGL draws primitives: points, line segments, or polygons—subject to several selectable modes and can control modes independently of each other.

**Figure 1.1: The Principle Functions of OpenGL**

The OpenGL Utility Toolkit (GLUT) is a library of utilities for OpenGL

programs, which primarily perform system-level I/O with the host operating system.

Functions performed include window definition, window control, and monitoring of keyboard and mouse input. Routines for drawing a number of geometric primitives (both in solid and wireframe mode) are also provided, including cubes, spheres and the Utah teapot. GLUT also has some limited support for creating pop-up menus.

## Motivation

The main aim is to implement the skills learnt in computer graphics theory and computer graphics lab programs using OpenGL software which includes the concepts like: Lighting, Projection, Rotation, Motion, Clipping, Transformation, Viewing, Visible surface detection, Input and interaction, Curves and Computer Animation. The application of this particular project motivates to learn lots of concepts which demonstrate the motion of the ball which can be observed in the animation along with the camera effect that makes the viewer control the camera as they please to see the animation.

## Basic Idea

Project’s main goal is to design a Memory game using OpenGL. This project displays a 2D array of cubic blocks which display colored objects when selected. The basic idea of the game is to test the user’s photographic memory. The user has to remember the location of each object and match it with its pair. The game ends when all the objects have been successfully matched with their pairs and there are no more blocks left. The score is calculated based on the time taken to match all the objects. With the help of various OpenGL functions, it’s managed

* + - To create a 2D array of rectangular boxes.
    - To create colored objects that pop up when the boxes are selected.
    - To create a window for the game using OpenGL create window.

## Proposed Model

This proposed model/mini-project is basically done in two dimensional effects by using open GL software.

## Problem Statement

To create a 2D array of boxes hiding objects which pop up when the boxes are uncovered and to specify the boundaries and limits of different objects in the pre-determined window size. It’s also required to design entry and exit screens.

## Objectives

* The objective of the game is to test the user’s memory by matching objects hidden inside blocks in continuous two step. The user can select any box randomly and open it to reveal the object hidden inside. The object gets covered as soon as the user moves to another box. The object revealed should be matched with the object in another random box in the next immediate step for the boxes to get eliminated. Keyboard is used as the interface.
* The distribution of the objects among the boxes is random and the objects are present at a new location every time the game is restarted. The game has one single level and the game ends when the level is completed. Since the game is based on the user’s memory and random distribution, the difficulty of the game is completely subjective.
* The game will be over when all the objects have been matched with their pair.

## Scope

The player starts the game & can traverse 2-dimentionally from one box to another and select the boxes using the keyboard. The score is calculated based on the least number of selections made to match all the objects with their pair.

## Summary

## This chapter introduces the given problem and its solution and the introduction to required software OpenGl and its utility toolkit. This also explains the motivation, basic idea, proposed model, scope and objectives of the solution.

## 

# CHAPTER 2

# REQUIREMENT SPECIFICATION

The minimum system requirements for executing the proposed project are discussed below in this chapter.

* 1. **Hardware Requirements**
     + - Microprocessor: Preferably above Quad core or at least dual core required.
       - Processor speed: 2 GHz or above.
       - Main memory : 2 GB RAM
       - Hard Disk : 40 GB
       - Hard disk speed in RPM:5400 RPM
       - Monitor, Key board and Mouse

.

* 1. **Software Requirements**
     + Windows used, but platform independent given Code blocks with libraries present.
     + A C/C++ (integrated with OPEN GL) compiler like Code blocks is required for compiling the source code to make the executable file which can then be directly executed.
     + A built-in graphics library: glut.h is required for drawing the layout of the game and glut32.dll for running the application.
  2. **Functional Requirements**
     + **OpenGL APIs**

To have a control on the flow of program and to interact with the window system then OpenGL API’S are used.

### GL/glut.h

Readily available library called the OpenGL Utility Toolkit (GLUT), which provides the minimum functionality that should be expected in any modern windowing system is used.



* **Summary**

Many of the computers in present time contain all hardware requirements specified above by default and concentration should be mainly on the software requirements as they don’t come pre-installed and need to be installed carefully.



# CHAPTER 3

# Design and Implementation

In this proposed model, it contains only one algorithm named as **display\_function**, and which is used in the program to develop the memory game. The algorithm has input and output which is shown below

## FLOW CHART:

START

THE PLAYER TRIES TO MATCH THE OBJECTS IN CONSECUTIVE TWO STEP.

NO

ALL OBJECTS MATCHED?

YES

DISPLAY SCORE

STOP

**Figure 4.1: Flow Chart of display\_function**



**Algorithm 3.1: display\_function**

**Input**: Keyboard keys as navigator.

**Output**: Score obtained by the Player.

**Steps:**

**Step 1:** Initialize the position of the cubic boxes.

**Step 2:** Randomly distribute the fixed set of objects among the cubic boxes.

**Step 3:** Set the limits of the screen for the game.

**Step 4:** Display welcome screen. If player presses ‘space’ then go to game screen.

**Step 5:** Move the navigator frame from one box to another as per the user’s command through the keyboard and open the box to reveal the object when user presses ‘enter’.

**Step 6:** Cover the object when moving to the next box and eliminate the box if the next object matches with the previous object.

**Step 7:** Repeat step 5 and 6 till all the objects are matched.

**Step 8:** Display the score on the screen.

**User interface**

Keyboard Control is allotted which are listed below

**Start:** Press spacebar key to start the game

**Navigation:** Use the up, down, left and right arrow keys to move accordingly.

**Enter**: Use enter to uncover the box  
           
**Help:**Press x for help during the game play.  
  
**Quit:**Press Esc key to quit the game.

Page 7

## Summary

## This chapter mainly describes the algorithm used for implementation of the proposed project with flow of control and also the inputs and outputs of the project. Main part of the project is the display function whose working is designed above.

Page 8

# CHAPTER 4

# RESULTS

### In this chapter the working of the project is discussed with the screenshots step by step, and describe what is being done in each step.

### Welcome Screen

### 

### 

### Figure 4.1: Welcome Screen

### Once the player starts the game, the welcome screen appears displaying basic information. The game starts when the user presses the “space” key.

### Game starts

### 

**Figure 4.2: Game window**

Once the player presses ‘space’ and starts the game this screen appears. A 4X5 matrix of cyan colored cubic boxes is present on the screen. A red frame surrounds the first box. The frame indicates the user’s position.

### 1st box clicked by the user

### 

**Figure 4.3: The object inside the box is revealed**

As the player moves across the boxes, he can select a box to reveal the object inside by pressing the “enter” key. The object gets covered again when the player moves to another box.

Page 10

### 2nd box clicked by the user such that both objects are matching.

### 

### 

### Figure 4.4 and 4.5: when the objects match

### When the objects match in 2 consecutive selections, the boxes are eliminated from the screen.

Page 11

### Instructions

### 

### Figure 4.6: Displaying instructions.

### When the user presses the “x” key on the keyboard while in the game window, the instructions are displayed.

### Displaying score

### 

**Figure 4.7: the final score of the player**

This is the end of the game and it displays the score obtained by the player in the game.

Page 12

* **Summary**

This chapter shows the working of the project at different stages and instructions to play the game. The game starts with the welcome screen and shifts to the game window once the user presses the spacebar. The player uses the prescribed keyboard functions and matches the objects and the game comes to an end once all the objects are matched. Once all the objects are matched, the game ends and the score is displayed on the screen.

Page 13

# CHAPTER 5

# CONCLUSIONS AND FUTURE ENHANCEMENT

Project has been successfully implemented and all the required objectives are satisfied. In this chapter the conclusion of the project done and the opportunities for future enhancement are discussed.

## 5.1 Conclusion

The project successfully implemented a simple game using OpenGL. OpenGL supports enormous flexibility in the design and the use of OpenGL graphics programs. The presence of many built in classes methods take care of much functionality and reduce the job of coding as well as makes the implementation simpler. This game shows the use of computer graphics throughout an application especially when it comes to interaction of computers with humans. In this program, how alphabetical characters and stored data like scores are rendered on screen, how the game or colorful objects are created, and how objects can be hidden from view is observed.

## 5.2 Future Enhancement

The future scope of this Computer Graphics mini-project lies in developing the project. After “game over” a list can be displayed which shows the top ten scorers, a “play again” button can also be embedded. Welcome screen needs more modification. There is scope to embed buttons like “about”, “how to play”, “configuration”, “profiles”, etc... In future this project would be implemented in source code for better experience of playing this game.

**CHAPTER 6**

# REFERENCES

1. Edward Angel’s , “Interactive Computer Graphics”, Pearson Education, 5th Edition, 2008.
2. “Interactive computer Graphics --A Top-down Approach using Open GL” Edward Angle, 3rd Edition, 2008.
3. Donald D Hearn and M Pauline Baker, "Computer Graphics with OpenGL", 3rd Edition, 2007.
4. <https://www.cs.rit.edu/~jdb/cg1/openGLIntro.pdf>
5. <https://learnopengl.com/>
6. [http://www.cs.uccs.edu/~ssemwal/glman.html/](http://www.cs.uccs.edu/~ssemwal/glman.html)
7. [http://www.opengl.czweb.org/ewtoc.html/](http://www.opengl.czweb.org/ewtoc.html)
8. [http://www.opengl.org](http://www.opengl.org/)/
9. [http://www.en.wikipedia.org/wiki/OpenGL/](http://www.en.wikipedia.org/wiki/OpenGL)