VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Belagavi – 590 018

A

Mini Project Report

On

"TIC TAC TOE"

Submitted in partial fulfillment of Bachelor of Engineering Degree

In

COMPUTER SCIENCE AND ENGINEERING

18CSMP68- Mobile Application Development Laboratory

Submitted by:

MOHAMMED SHAHZADUL QUADRI	**1HK20CS093**
MOHAMMED TAMHEED SHARIFF	**1HK20CS095**

Under the guidance of

Prof. Najmusher H

Assistant Professor, Department of Computer Science & Engineering

JULY 2023

Department of Computer Science and Engineering

HKBK COLLEGE *of* ENGINEERING

(Approved by AICTE & Affiliated to VTU)

No.22/1,Opp., Manyata Tech Park Rd, Nagavara, Bengaluru, Karnataka 560045

Email: info@hkbk.edu.in URL: www.hkbk.edu.in

HKBK COLLEGE *of* ENGINEERING

Nagawara, Bangalore–560 045

Approved by AICTE & Affiliated to VTU

Department of Computer Science and Engineering

Certificate

Certified that the Mini Project Work entitled **"TIC TAC TOE",** carried out by **MOHAMMED SHAHZADUL QUADRI (1HK20CS093)** and **MOHAMMED TAMHEED SHARIFF (1HK20CS095)** are bonafide students of **HKBK COLLEGE*of* ENGINEERING**, in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of the **Visvesvaraya Technological University**, Belgaum, during the year **2022–23**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of **18CSMP68- Mobile Application Development Laboratory** prescribed for the said Degree.

Prof. Najmusher H	**Dr. Smitha Kurian**	**Dr. Tabassum Ara**
Guide	HOD - CSE	Principal – HKBKCE

External Viva

Name of the Examiners	Signature with Date
1. _____	_____
2. _____	_____

ACKNOWLEDGEMENT

We would like to express our regards and acknowledgement to all who helped us in completing this mini project successfully.

First of all we would take this opportunity to express our heartfelt gratitude to the personalities, **Mr. C M Ibrahim**, Chairman, HKBKGI and **Mr. C M Faiz Mohammed**, Director, HKBKGI for providing facilities throughout the course.

We express our sincere gratitude to **Dr. Tabassum Ara**, Principal, HKBKCE for her support towards the attainment of knowledge.

We consider it as a great privilege to convey our sincere regards to **Dr. Smitha Kurian.**, Associate Professor and HOD, Department of CSE, HKBKCE, for her constant encouragement throughout the course of the project.

We would specially like to thank our guide **Prof. Najmusher H,** Assistant Professor, Department of CSE, HKBKCE for her vigilant supervision and her constant encouragement. She spent her precious time in reviewing the project work and provided many insightful comments and constructive criticism.

Finally, we thank Almighty, all the staff members of CSE Department, our family members and friends for their constant support and encouragement in carrying out the project work.

Mohammed Shahzadul Quadri (1HK20CS093)
Mohammed Tamheed Shariff (1HK20CS095)

ABSTRACT

Tic Tac Toe is a classic paper-and-pencil game that has been enjoyed by people of all ages for many years. It is played on a square grid usually consisting of nine equal squares arranged in a 3x3 layout. The game involves two players, traditionally referred to as "X" and "O," who take turns marking their respective symbols on the grid.

The objective of Tic Tac Toe is to be the first player to form a horizontal, vertical, or diagonal line of three of their symbols in a row. Players alternate turns, with "X" typically going first. They place their symbol in an empty square of their choice, with the goal of strategically positioning their symbols to achieve the winning pattern.

The game continues until one of the players succeeds in forming a winning line or all the squares are filled without a winner, resulting in a draw. A winning line can be achieved by placing three symbols consecutively in a row, column, or diagonal. If a player manages to form a winning line, they are declared the winner.

Tic Tac Toe is a relatively simple game, but it still requires strategic thinking and planning. Experienced players try to anticipate their opponent's moves and create opportunities for their own winning lines while blocking their opponent's progress. Since the grid is small and the number of possible moves is limited, the game can often end in a draw if both players play optimally.

The game's simplicity and accessibility have made it a popular choice for teaching basic logical reasoning and strategy to children. It helps develop critical thinking skills, pattern recognition, and decision-making abilities. Tic Tac Toe can be played on paper or a physical board, but it has also found its digital incarnation in various computer and mobile applications, allowing players to enjoy the game anytime and anywhere.

In conclusion, Tic Tac Toe is a classic game that has stood the test of time. Its simple rules, yet strategic gameplay, have made it a favorite pastime for generations. Whether played with pencil and paper or in a digital format, Tic Tac Toe provides an engaging challenge that can be enjoyed by players of all ages.

TABLE OF CONTENTS

LIST OF FIGURES:

CHAPTER 1:

INTRODUCTION

1.1 Overview

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014.The first stable build was released in December 2014, starting from version 1.0.

On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development.[13] Java is still supported, as is C++.

A specific feature of the Android Studio is an absence of the possibility to switch autosave feature off. The following features are provided in the current stable version:

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go; and Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that vary by platform version. “External projects backport some Java 9 features. While IntelliJ states that Android Studio supports all released Java versions, and Java 12,it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

Once an app has been compiled with Android Studio, it can be published on the Google Play Store. The application has to be in line with the Google Play Store developer content policy.

1.1 Statement of problem

The problem at hand is to implement a Tic Tac Toe game. Tic Tac Toe, also known as Noughts and Crosses, is a classic two-player game played on a 3x3 grid. The bjective of the game is to get three of your own symbols (either X or O) in a horizontal, vertical, or diagonal line, or to fill the entire grid without any three-symbol combinations.

The game should allow two human players to take turns placing their symbols on the grid until a player wins or the game ends in a draw.

The implementation should be user-friendly, providing clear instructions and visual representation of the game board. It should handle input validation and gracefully handle any errors or unexpected user inputs.

1.2 Objective of the problem

The objective of the Tic Tac Toe game is to strategically place your symbols (either X or O) on a 3x3 grid in order to create a winning combination. The game is played between two players, who take turns marking empty cells on the grid with their respective symbols.

The ultimate goal is to be the first player to successfully create a line of three of their symbols either horizontally, vertically, or diagonally. By placing symbols strategically, players aim to prevent their opponent from achieving a winning combination while simultaneously working towards their own victory.

The game requires critical thinking, spatial awareness, and anticipation of the opponent's moves. Players must carefully consider each move and assess potential future consequences, attempting to create opportunities for their own victory while blocking their opponent's progress. Once a player successfully creates such a line, they are declared the winner of the game. Conversely, if the entire grid becomes filled with symbols and no player has formed a winning combination, the game ends in a draw

The objective of Tic Tac Toe extends beyond simply winning the game. It encourages players to develop critical thinking skills, strategic planning, and the ability to adapt their moves based on the changing game state. The simplicity of the game rules allows players to focus on the strategic aspect while still providing entertainment and a sense of achievement.

CHAPTER 2:

SYSTEM REQUIREMENTS SPECIFICATIONS

SYSTEM REQUIREMENTS SPECIFICATION

A software requirement definition is an abstract description of the services which the system should provide, and the constraints under which the system must operate. It should only specify the external behavior of the system.

Functional Requirements

Functional Requirements define the internal working of the software. The following conditions must be taken care of: The ability to perform correct operations when corresponding keys are pressed.

Non-functional Requirements

Non-functional requirements are requirements which specify criteria that can be used to judge the operation of the system, rather than specific behaviors. This should be contrasted with functional requirements that specify specific behavior or functions. Typical non-functional requirements are reliability and scalability. Non- functional requirements are “constraints”, “quality attributes” and “quality of service requirements”.

Types of non-functional requirements

- **Volume:** Volume of a system (the total space occupied) varies depending on how the component assemblies are arranged and connected.
- **Reliability:** System reliability depends on component reliability but unexpected interactions can cause new types of failure and therefore affect the reliability of the system.
- **Security:** The security of the system (its ability to resist attacks) is a complex property that cannot be easily measured. Attacks maybe devised that were not anticipated by the system designers and may use default built- in safeguards.
- **Repairability:** This property reflects how easy it is to fix a problem with the system once it has been discovered. It depends on being able to diagnose the problem, access the components that are faulty and modify or replace these components.
- **Usability:** This property reflects how easy it is to use the system. It depends on the technical system components, its operators and its operating environment.

Software Requirements:

Operating System : Windows

Front End : Android Studio

Coding Language : Java

Hardware Requirements

System : Intel® Core™ i3 – 6006U CPU @ 2.00GHz

Hard Disk : 30 GB or above

Monitor : 15 VGA color

RAM : 4GB or above

Introduction to Environment

Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go;[20] and Android Studio 3.0 or later supports Kotlin[21] and "all Java 7 language features and a subset of Java 8 language features that vary by platform version."[22] External projects backport some Java 9 features.[23] While IntelliJ states that Android Studio supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.A specific feature of the Android Studio is an absence of the possibility to switch autosavefeature off.

The following features are provided in the current stable version

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations[18]
- Support for building Android Wear app

- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine[19]
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

The Android Emulator has additional requirements beyond the basic system requirements for Android Studio, which are described below:[30]

- SDK Tools 26.1.1 or higher;
- 64-bit processor;
- Windows: CPU with UG (unrestricted guest) support;
- Intel Hardware Accelerated Execution Manager (HAXM) 6.2.1 or later (HAXM 7.2.0 or later recommended).

The use of hardware acceleration has additional requirements on Windows and Linux:

- Intel processor on Windows or Linux: Intel processor with support for Intel VT-x, Intel EM64T (Intel 64), and Execute Disable (XD) Bit functionality;
- AMD processor on Linux: AMD processor with support for AMD Virtualization (AMD-V) and Supplemental Streaming SIMD Extensions 3 (SSSE3);
- AMD processor on Windows: Android Studio 3.2 or higher and Windows 10 April 2018 release or higher for Windows Hypervisor Platform (WHPX) functionality.

CHAPTER 3:
DESIGN

DESIGN

High level design

Requirement analysis encompasses all the tasks that go into the instigation, scoping and definition of a new or altered system. Requirement analysis is an important part of the design process. Here we identify the needsor requirements. Once the requirements have been identified the solution for the requirements can be designed.

We design a project with specific goals, tasks and outcomes. The more specific and the more closely it is aligned. UI (User Interface) and UX (User Experience) are the two vital components of your mobile app design. The former is responsible for the look and appeal of your application, whereas the latter facilitates the interaction between the design elements. While preparing a design for your mobile app, two primary considerations should be your project's scope and budget.

The time required for designing cannot be specified as it may take only a couple of hours to a few days. Another factor that impacts your app designing time is the experience of the developers from your mobile app development services provider. It is a multistep process that needs to be done carefully to ensure that the outcome provides a clear vision of your business idea.

CHAPTER 4:

IMPLEMENTATION

IMPLEMENTATION:

Implementation is an act or instance of implementing something that is, the process of making something active or effective. Implementation must follow any preliminary thinking in order for something to actually happen. In the context of information technology, implementation encompasses the process involved in getting newsoftware or hardware to operate properly in its environment. This includes installation, configuration, running, testing and making necessary changes.

MainActivity.java

```
package com.example.tictactoe;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity
    { ActivityMainBinding binding;
    private int totalSelectedBoxes = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding =
        ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
        combinationList.add(new int[] {0,1,2});
        combinationList.add(new int[] {3,4,5});
        combinationList.add(new int[] {6,7,8});
        String getPlayerOneName = getIntent().getStringExtra("playerOne");
        String getPlayerTwoName = getIntent().getStringExtra("playerTwo");

        binding.image1.setOnClickListener(new View.OnClickListener()
            { public void onClick(View view) {
                if (isBoxSelectable(0)){
                    performAction((ImageView) view, 0);
                }
            }
        });
```

```
        binding.image2.setOnClickListener(new View.OnClickListener()
            { @Override
            public void onClick(View view) {
                if (isBoxSelectable(1)){
                    performAction((ImageView) view, 1);
                }
            }
        });
        binding.image3.setOnClickListener(new View.OnClickListener()
            { @Override
            public void onClick(View view) {
                if (isBoxSelectable(2)){
                    performAction((ImageView) view, 2);
                }
            }
        });
    }

    private void performAction(ImageView  imageView, int selectedBoxPosition) {
        boxPositions[selectedBoxPosition] = playerTurn;

        if (playerTurn == 1) {
            imageView.setImageResource(R.drawable.ximage);
            if (checkResults()) {
                ResultDialog resultDialog = new ResultDialog(MainActivity.this,
binding.playerOneName.getText().toString()+ " is a Winner!", MainActivity.this);
                resultDialog.setCancelable(false);
                resultDialog.show();
            }
        }else {
                changePlayerTurn(1);
                totalSelectedBoxes++;
        }
    }
```

a

```
    private boolean checkResults(){
        boolean response = false;
        for (int i = 0; i < combinationList.size(); i++){
            final int[] combination =
            combinationList.get(i);

            if (boxPositions[combination[0]] == playerTurn && boxPositions[combination[1]] == playerTurn &&
                    boxPositions[combination[2]] == playerTurn) {
                response = true;
            }
        }
        return response;
    }

    private boolean isBoxSelectable(int boxPosition)
        { boolean response = false;
        if (boxPositions[boxPosition] == 0) {
            response = true;
        }
        return response;
    }

    public void restartMatch(){
        boxPositions = new int[] {0,0,0,0,0,0,0,0,0}; //9
        zero playerTurn = 1;
        totalSelectedBoxes = 1;

        binding.image1.setImageResource(R.drawable.white_box);
        binding.image2.setImageResource(R.drawable.white_box);
        binding.image3.setImageResource(R.drawable.white_box);
        binding.image4.setImageResource(R.drawable.white_box);
    }
}
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="@color/lavender"
   tools:context=".MainActivity">


   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:gravity="center"
      android:layout_above="@id/container"
      android:layout_alignParentTop="true"
      android:orientation="horizontal">

         <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:text="Player One"
            android:id="@+id/playerOneName"
            android:textSize="20sp"
            android:gravity="center"
            android:textStyle="bold"
            android:textColor="@color/lavender"
            android:maxLines="1"/>

         <ImageView
            android:layout_width="40dp"
            android:layout_height="40dp"
            android:src="@drawable/ximage"
            android:layout_marginTop="20dp"
            android:layout_marginBottom="20dp"
            android:layout_gravity="center"/>

      </LinearLayout>


   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:id="@+id/container"
      android:layout_alignParentBottom="true"
      android:layout_marginStart="10dp"
```

a

```
    android:layout_marginEnd="10dp"
    android:layout_marginBottom="50dp"
    android:orientation="vertical">

      <ImageView
        android:layout_width="0dp"
        android:layout_height="115dp"
        android:id="@+id/image1"
        android:layout_weight="1"
        android:background="@drawable/white_box"
        android:layout_marginTop="10dp"
        android:layout_marginEnd="5dp"
        android:layout_marginStart="10dp"
        android:adjustViewBounds="true"
        android:padding="20dp"/>

      <ImageView
        android:layout_width="0dp"
        android:layout_height="115dp"
        android:id="@+id/image3"
        android:layout_weight="1"
        android:background="@drawable/white_box"
        android:layout_marginTop="10dp"
        android:layout_marginEnd="5dp"
        android:layout_marginStart="10dp"
        android:adjustViewBounds="true"
        android:padding="20dp"/>

  </LinearLayout>

</RelativeLayout>
```

CHAPTER 5:

SNAPSHOTS

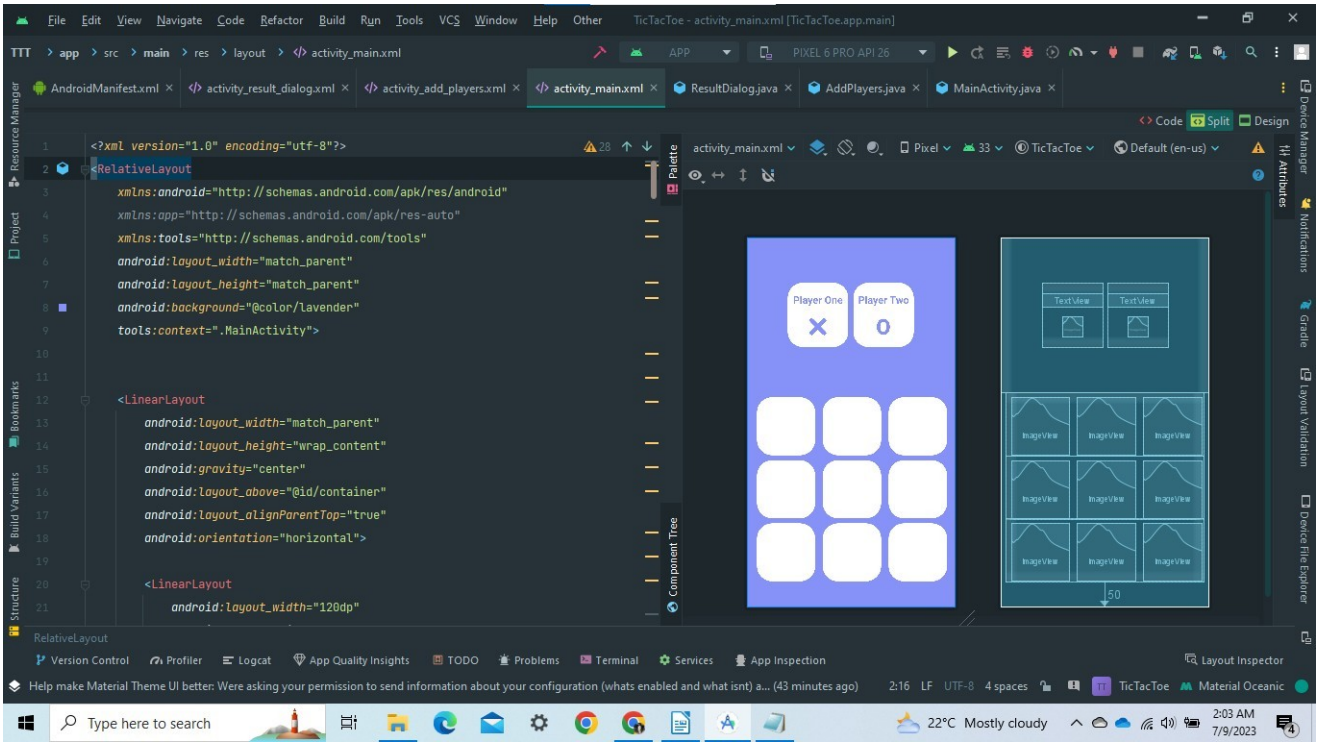

Fig 5.1 Activity_main.xml

file is an important component in Android development that defines the layout and appearance of the main user interface (UI) for an Android app. It is typically created as an XML (Extensible Markup Language) file and is associated with the main activity of an app.

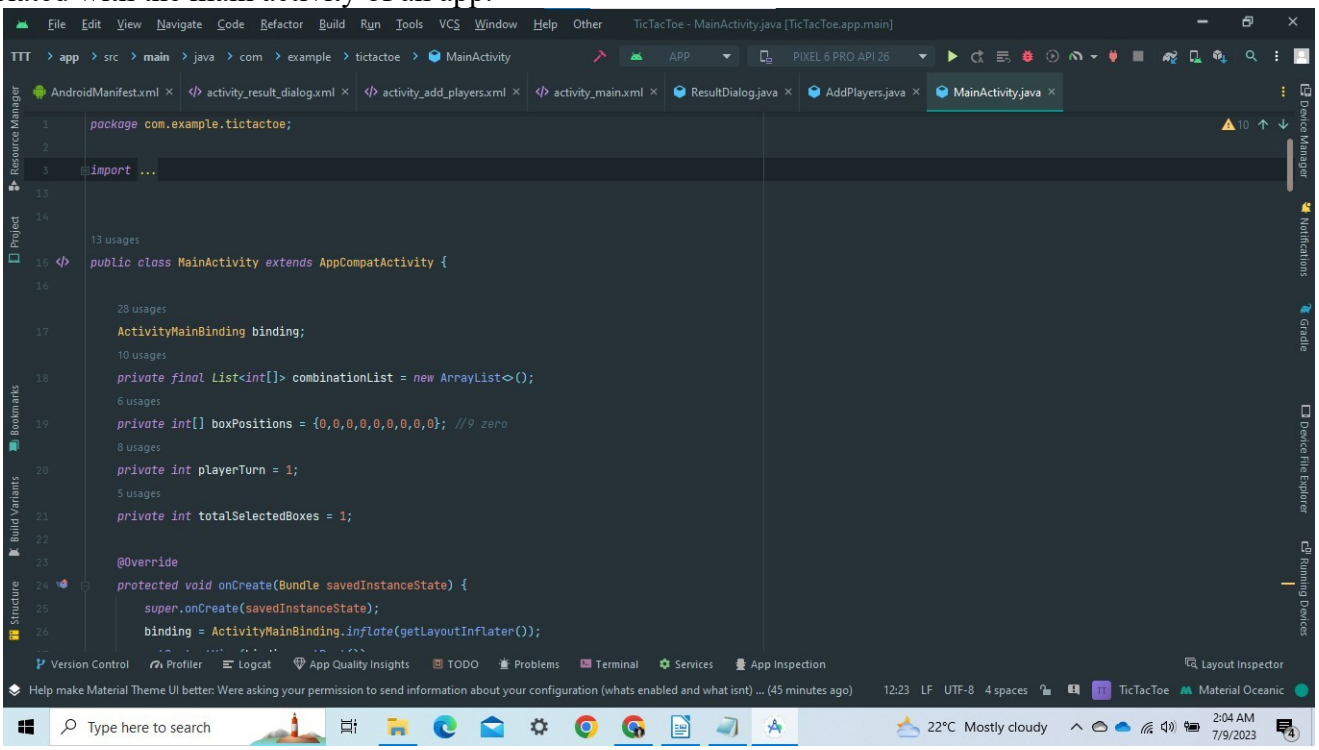

Fig 5.2 MainActivity.java

The MainActivity.java file is a Java class that serves as the main entry point for an Android app. It is responsible for controlling the behavior and logic of the main activity, which represents the primary user interface for the app.

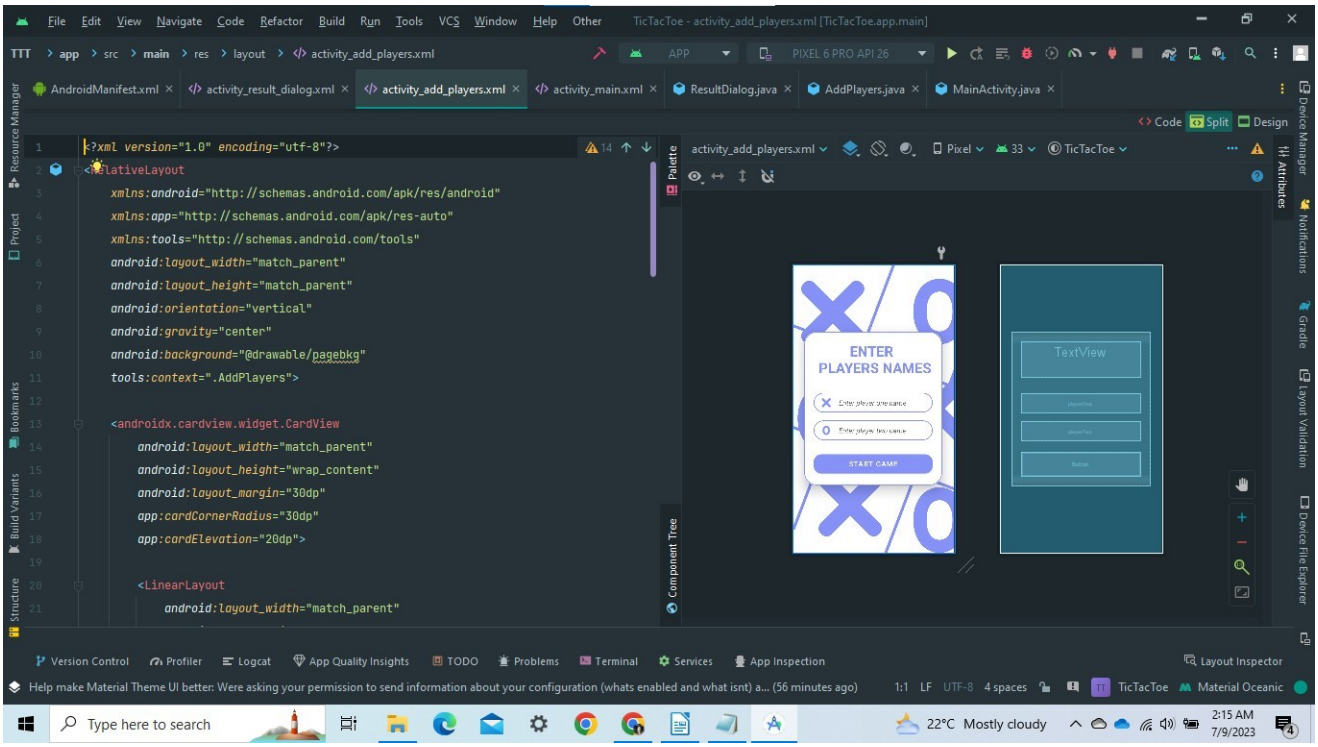

Fig 5.3 Add Player.xml

It allows to add player names, as is it a two player game.

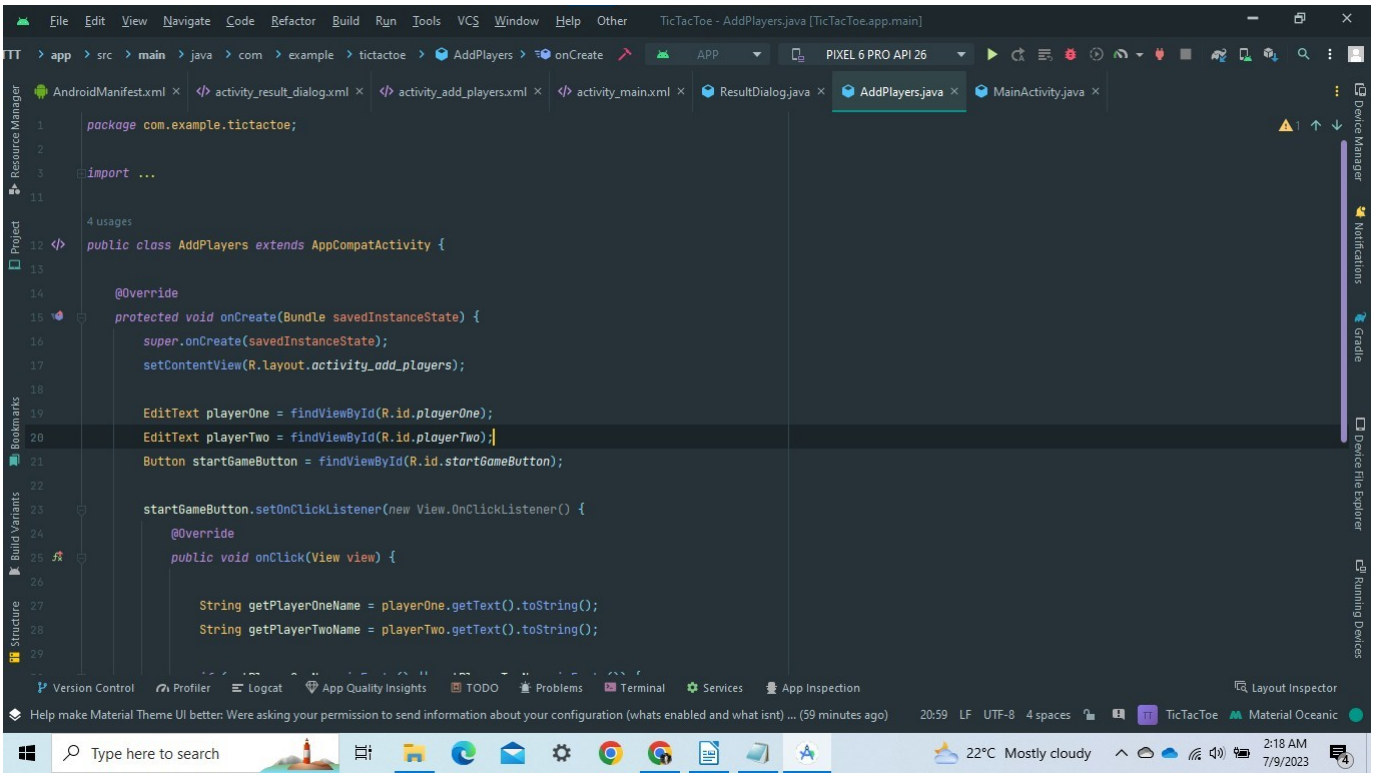

Fig 5.4 AddPlayer.java

The names are stored and each player is asigned with either ‘X’ or ‘O’

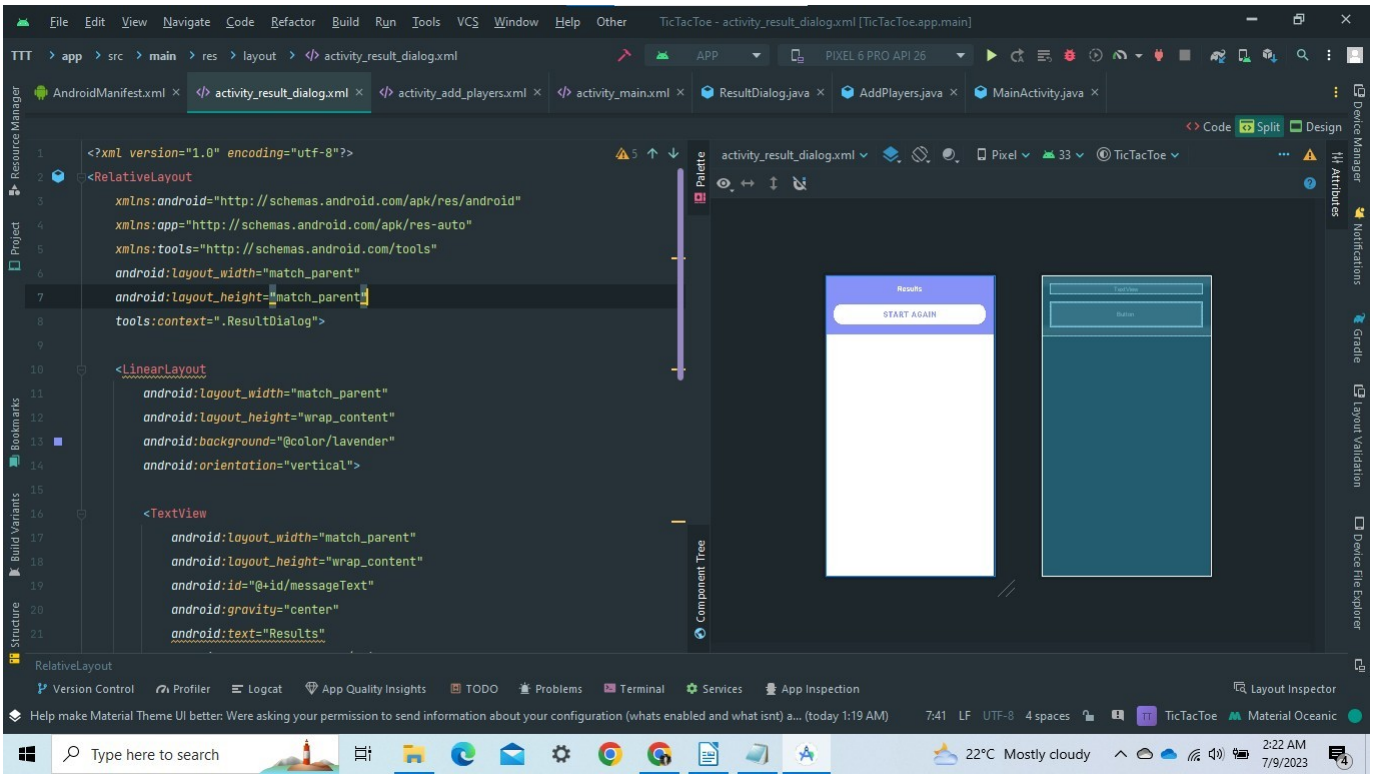

Fig.5.5 Result.xml

It is the design layout for the messages that displays who won the game.

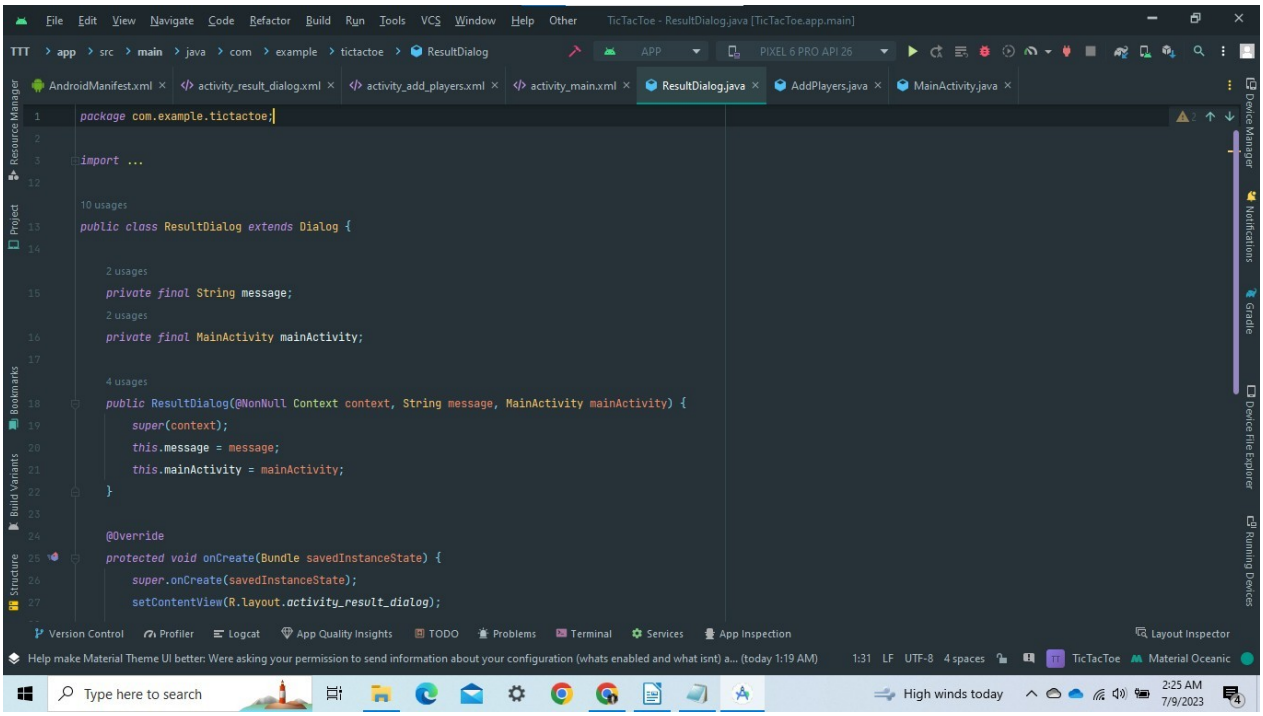

Fig .5.6 Result.java

Using the java program the result is declared after checking all the rules.

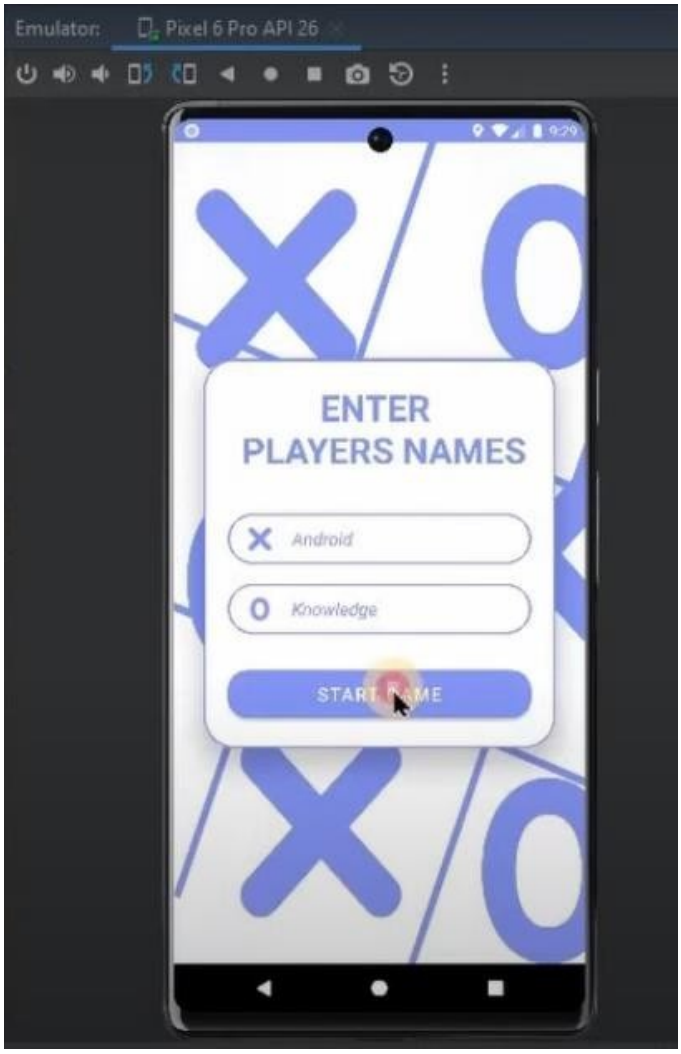

Fig 5.7 Output(1)

In a Tic Tac Toe game, two players engage in a friendly battle of wits and strategy. Each player takes turns, carefully considering their moves and analyzing the game state to gain an advantage. The players immerse themselves in the competition, striving to outsmart their opponent and claim victory.

As the game progresses, the players exhibit their unique playing styles and approaches. Some players might adopt an aggressive strategy, aiming to quickly form winning combinations and put their opponent on the defensive. They seize opportunities, seeking ways to maximize their chances of success while keeping their opponent off balance.

On the other hand, players may opt for a more defensive approach, focusing on blocking their opponent's moves and disrupting potential winning lines. They carefully analyze the board, identifying critical positions and actively countering their opponent's strategies. Such players demonstrate patience and resilience, relying on their tactical thinking to turn the game in their favor.

Throughout the game, players engage in a silent battle of wits, interpreting their opponent's moves, and predicting their next actions. They attempt to decipher their opponent's patterns and tendencies, looking for weaknesses to exploit or unexpected moves to surprise them. It becomes a mental game of anticipation and adaptability, as each player strives to stay one step ahead.

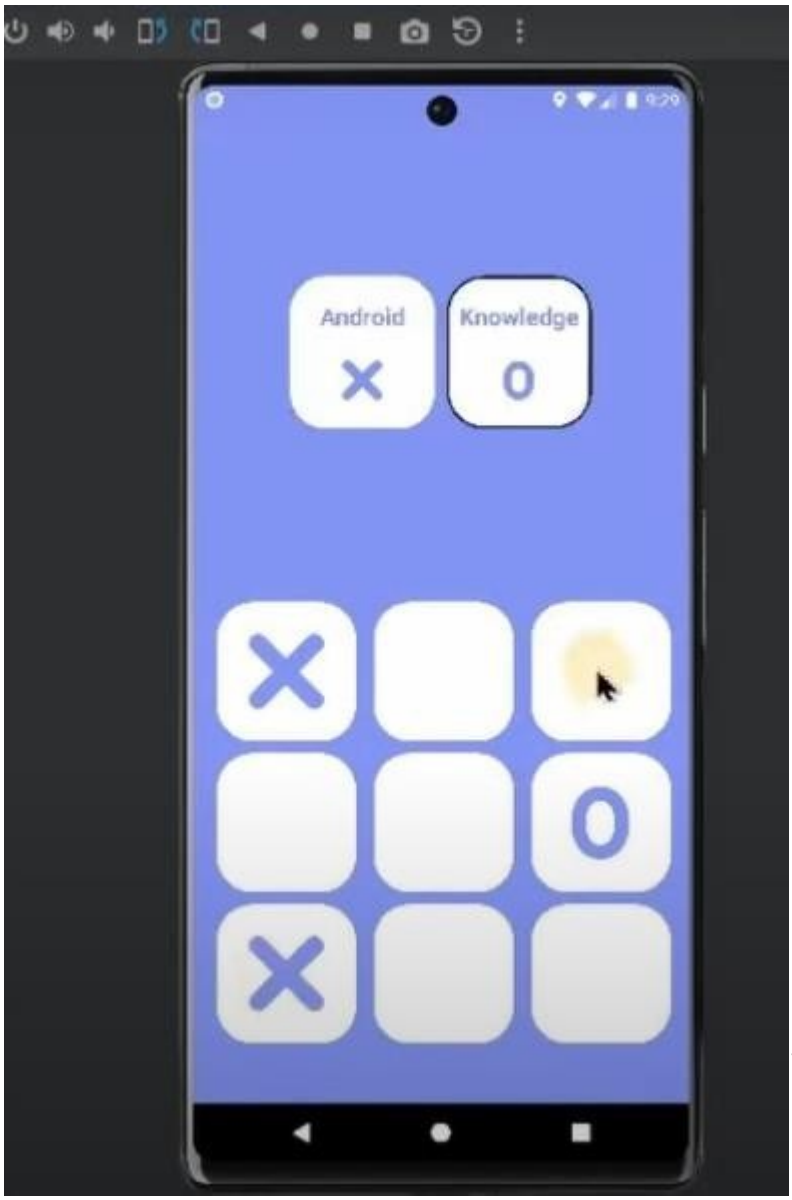

Fig 5.8 Output(2)

- Personalization and Contextual Understanding: The user can adapt to specific domains or user preferences, tailoring their responses and recommendations to provide a personalized experience. They strive to grasp the context of conversations, considering previous interactions to provide accurate and relevant responses.

- Knowledge Base Integration: Players leverage extensive knowledge bases and databases to access a wide range of information, allowing them to provide accurate answers and insights to user queries.

- Learning and Improvement: Players continuously learn from user interactions, feedback, and data to improve their conversational abilities, refining their responses and becoming more effective over time. They can adapt and enhance their capabilities through machine learning techniques.

CHAPTER 6:

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion:

In conclusion, the game of Tic Tac Toe is a timeless and engaging pastime that combines strategy, critical thinking, and social interaction. It challenges players to analyze the game state, anticipate their opponent's moves, and strategically place their symbols on the grid. The objective of the game is to create a winning combination of three symbols in a row, column, or diagonal, while also blocking the opponent from doing the same. Tic Tac Toe is not only a source of entertainment but also promotes cognitive skills such as problem-solving, pattern recognition, and decision-making. It is a game that transcends age barriers, allowing people of all generations to come together and enjoy a friendly competition. Whether played for fun or as a way to sharpen one's strategic acumen, Tic Tac Toe continues to captivate and entertain players around the world.

6.2 Future enhancements

These enhancements and scope expansions have the potential to make the Tic Tac Toe game more diverse, engaging, and appealing to a wider audience.

- AI Opponent: One potential enhancement is to introduce an artificial intelligence (AI) opponent. This would allow players to compete against a computer-controlled player of varying difficulty levels. Implementing AI algorithms, such as minimax or alpha-beta pruning, would enable the AI opponent to make intelligent and challenging moves, providing players with a more dynamic and unpredictable gaming experience.

- Board Size Customization: Adding the ability to customize the grid size beyond the standard 3x3 could offer a new dimension to the game. Allowing players to choose larger grids, such as 4x4 or 5x5, would introduce additional complexity and strategic possibilities, making the game more challenging and engaging.

- Tournament and Leaderboard Integration: Implementing tournament functionality and integrating leaderboards would add a competitive aspect to the game. Players could participate in ranked tournaments, earn points based on their performance, and compete for top positions on global or friend-specific leaderboards.

BIBLIOGRAPHY:

1. Erik Hellman, “Android Programming – Pushing the Limits”, 1st Edition, Wiley India Pvt Ltd, 2014. ISBN-13: 978-8126547197

2. Dawn Griffiths and David Griffiths, “Head First Android Development”, 1st Edition, O’Reilly SPD Publishers, 2015. ISBN-13: 978-9352131341

3. Bill Phillips, Chris Stewart and Kristin Marsicano, “Android Programming: The Big Nerd Ranch Guide”, 3rd Edition, Big Nerd Ranch Guides, 2017. ISBN-13: 978-0134706054

Links:

Google Developer Training, "Android Developer Fundamentals Course – Concept Reference”, Google Developer Training Team, 2017. https://www.gitbook.com/book/google-developer-training/android-developer- fundamentals- course-concepts/details

HKBK College of Engineering
No.22/1, Opp., Manyata Tech Park Rd, Nagawara, Bengaluru, Karnataka 560045.
Approved by AICTE & Affiliated by VTU

Department of Computer Science & Engineering

DEPARTMENT MISSION AND VISION

VISION

To advance the intellectual capacity of the nation and the international community by imparting knowledge to graduates who are globally recognized as innovators, entrepreneur and competent professionals.

MISSION

M-1. To provide excellent technical knowledge and computing skills to make the graduates globally competitive with professional ethics.

M-2. To involve in research activities and be committed to lifelong learning to make positive contributions to the society. Institute

INSTITUTE MISSION AND VISION

VISION

To empower students through wholesome education and enable the students to develop into highly qualified and trained professionals with ethics and emerge as responsible citizen with broad outlook to build a vibrant nation.

MISSION

M - 1. To achieve academic excellence through in-depth knowledge in science, engineering and technology through dedication to duty, innovation in teaching and faith in human values.

M - 2. To enable our students to develop into outstanding professionals with high ethical standards to face the challenges of the 21st century

M-3. To provide educational opportunities to the deprived and weaker section of the society, to uplift their socio-economic status.

PROGRAM OUTCOMES(PO'S)

PO-1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO-2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. **PO-3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO-4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO-5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations. **PO-6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice. **PO-7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO-8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO-9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO-10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO-11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO-12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO-1. Problem-Solving Skills: An ability to investigate and solve a problem by analysis, interpretation of data, design and implementation through appropriate techniques, tools and skills.

PSO-2. Professional Skills: An ability to apply algorithmic principles, computing skills and computer science theory in the modelling and design of computer-based systems.

PSO-3. Entrepreneurial Ability: An ability to apply design, development principles and management skills in the construction of software product of varying complexity to become an entrepreneur.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

PEO-1. To provide students with a strong foundation in engineering fundamentals and in the computer science and engineering to work in the global scenario.

PEO-2. To provide sound knowledge of programming and computing techniques and good communication and interpersonal skills so that they will be capable of analyzing, designing and building innovative software systems.

PEO-3. To equip students in the chosen field of engineering and related fields to enable him to work in multidisciplinary teams.

PEO-4. To inculcate in students professional, personal and ethical attitude to relate engineering issues to broader social context and become responsible citizen.

PEO-5. To provide students with an environment for life-long learning which allow them to successfully adapt to the evolving technologies throughout their professional career and face the global challenges.

COURSE OUTCOMES(COs)

CO-1. Create, test and debug Android application by setting up Android development environment.

CO-2. Implement adaptive, responsive user interfaces that work across a wide range of devices.

CO-3. Infer long running tasks and background work in Android applications.

CO-4. Demonstrate methods in storing, sharing and retrieving data in Android applications.

CO-5. Analyse performance of android applications and understand the role of permissions and security.

CO-6. Describe the steps involved in publishing Android application to share with the world.