LUDO APPLICATION DEVELOPMENT USING FLUTTER

Dhiraj Kumar Jaiswal1, Dr. Nitin Mishra2, Abhishek Chaudhary3, Pankaj Kumar4
School of Computer Science and Engineering, Galgotias University, Greater Noida, India
1dheeraj83721@gmail.com, 2nitin.mishra@galgotiasuniversity.edu.in
3abhishekchaudhary12dec@gmail.com, 4pk71094@gmail.com

ABSTRACT

In 21st century a digital world where people prefer technologies and innovation to do any kind of job. Weather, its playing games, making transactions, meetings, or even exams. It is a new world with full of opportunities to develop and grow. Gaming has become common for everyone, we can see its growing from boards and grounds to screens. Playing games have never been so easy before, we can play games even if we are not in a same room or even in the same country. It just needs a mobile and an app to enjoy with your loved ones. Ludo is played by two to four players. No internet is required as it is played in single app with multiple people like we use to play on board at our grandma’s. It is a cross-platform mobile application developed using flutter and dart on Android Studio and Vs Code. Flutter is an open-source SDK for developing Android and iOS applications. It has just-in-time compilation which executes the computer code that encompasses compiling during program execution at run time rather than preceding execution. It also has a feature called hot reload which helps developers to easily experiment, build UI, fix bug and add features while inserting the updated code into the running machine. When virtual machine updates the classes with new fields, functions and methods, Flutter framework implements the changes in widget tree, allowing you to see the new changes.

Keywords: Flutter, Dart.

I. INTRODUCTION

The number of mobile Application Being News has increased in our daily life. Since 2016 there is more mobile applications traffic network then compare to web applications.

Most of the users now a day mainly focus on the mobile applications that desires to adapt itself into two separate platforms Android and iOS. Interestingly the differences between these two platforms are so much that it will require a whole new study on it and for developing application it uses such as Java/Kotlin only for Android and Object-C/Swift for iOS. Thus, developers and organization often require more at dealing with complexity of developing cross-platform applications.

Facebook introduced React Native a JavaScript framework in March 2015 which is an open source cross-platform used by developers and organizations to build applications 23. On other hand, Google introduced flutter in May 2017 which is a mobile SDK to build applications which can run on both the platforms (Android and iOS). Thus, reducing the time and complexity of the application. Ludo App is completely build on flutter which makes it light weight and with enhanced UI. Ludo is a game which requires the basic knowledge for different moves that needs to be taken in order to win the game.

In the paper, we will describe the rules of the game, modelling, development and elements of the game. And in the end some future developments which can be added in this Application.
II. LITERATURE REVIEW

A. Flutter

Flutter is a framework which is a cross-platform used in developing high-performance and effective mobile applications. Flutter was released in 2016 by Google. Google choose flutter for application development framework in its next generation operating systems. Flutter is dependent on the device’s OEM widgets rather than consuming web views. It also uses high performance rendering engine to render each view component using its own. Thus, providing a way to build applications that are as high-performance and complex as native applications can be. In Flutter Hot reload is known as Stateful hot reload which makes easy development with reduction in time. Stateful hot reload works same as hot reload in react native where updated code is inserted into the running Dart virtual machine. When virtual machine updates the classes with new fields, functions and methods, Flutter framework implements the changes in widget tree, allowing you to see the new changes.

Architecture

Flutter is a layered designed system which has a numbers of independent libraries that depend on the underlying layer. No layer has the access to the layer below and framework levels are designed in a manner that it is optional and replaceable. An embedder provides entrypoint which coordinates with operating system to access the services like rendering surfaces, accessibility and event loop. The embedder is platform dependent currently used language is Java/C++ for Android, Objective-C/Objective-C++ for iOS/macOS, and C++ for windows and Linux. Flutter engine is written in C++ which provides the low-level implementation of API (graphics, text layout, file I/O, plugin).

![Flutter Architectural Diagram](image)

Dart

In Flutter, mobile Applications use programming language called Dart which was developed and maintained by Google. Initially it was only used by Google but now has been available for everyone to use and develop applications. It proved to develop high performance, effective, massive web applications, such as AdWords.

The main purpose to develop dart was to find a replacement and successor of JavaScript. Therefore, we can that it implements the important features of JavaScript’s next standard (ES7), like keywords “async” and “await”. Google played a bold move here as java was most used programming language, so it made dart with same styling and syntax with Java just to attract developers which were not familiar or never used JavaScript before. Dart uses static type checking to ensure that variable’s value matches its static type. This is also referred as sound typing. Dart has feature called sound null safety which protects us from null exceptions at runtime through static code analysis.
III. PROPOSED METHODOLOGY

A. System Overview

We developed a flutter based mobile application Ludo gaming app. Ludo is a board game played by two to four players. No internet is required as it is played in single app with multiple people like we use to play on board at our grandma’s. At the beginning of the game, each player's pawn is out of play and staged in one of the large corner areas of the board which player chooses as its color (called the player's yard/home). After the dice is thrown and number six is drawn then only players will enter their pawns one per time on their respective starting squares, and continue to race them in clockwise direction around the board along the game track. When reaching the square below his home column, a player continues by racing pawns up the column to the finishing square. The rolls of a dice control the movement of the pawns, and entry to the finishing square requires a precise roll from the player. The first to bring all their pawns to the finishing square wins the game. The others often continue play to determine second, third, and fourth place finishers.

B. Proposed System

On opening an App, the Splash screen with the Ludo app logo will appear. Then we will be able to see the main Screen where different options will be available to take the action like to play game or the option menu or close game option. Play game option will start the game. Option menu will have different actions like sound adjustments. Quit game will exit the game and close the app.

![Diagram]

Figure 2: Architectural Diagram

IV. GAME THEORY

A. Rules

This game can be played by two to four players. The pawns under each player are out of play at the start and stay in one of the large corner portion of the board in the color chosen by each player referred as player's yard/home. At every point in time, it is the turn of one of the players. Turns rotate according to the cyclic order of players. The player whose turn it is, rolled number six on dice then only pawn is allowed to move accordingly.

Entry: If number six appears on dice and the player pawns are at home and the player’s entry field is not occupied by its own pawn then player must move one pawn from his home to his entry field.

Forward: If all the pawns of the particular players are already out and no entry move is possible then player must select one of his pawns on the board and move it forward by the exact number rolled on the dice in a clockwise direction tracing the complete board until he reaches the winning squares.
If in moving forward your pawn gets the target field of the opponent’s pawn, then the opponent’s pawn is kicked and it is returned to its home where player will start again with that pawn only.

If the player rolls the dice and number six appears then player gets another turn to roll the dice and then the next player in order will get his turn 9.

**B. Strategies**

In doing anything we need some strategies to do it in the most efficient way so here also we have to determine the efficient way to win the game. There are many global strategies which do not change during the game.

**Aggressive:** Giving priority to kick opponents pawn so that he/she will start again.

**Cautious:** Giving priority to avoid kicking opponents pawn so that it does not anger the opponent and with the possibility that he/she may not kick your pawn.

**Defensive:** Giving priority to the move to a target field where pawn cannot be kicked.

**Movie-First:** Giving priority in moving the foremost pawn.

**Movie-Last:** Giving priority in moving the hindmost pawn.

More sophisticated strategies can be defined by taking the moves of opponents into account during the game to counter the movie and winning the game.

**C. Elements**

**Player:** Their will be two to four players at a time. Players will have their own color which will identify their home. Players turn will come in cyclic order.

**Pawn:** Four pawns are allotted to the particular player with same color chosen by the player. Pawn will move forward according to the dice rolled in clockwise direction.

**Field:** Once player rolls number six then only pawn is allowed to come out in entry field and in the cyclic turn of players they move forward to get to the winning squares or goal fields.

**Board:** Board consists of all the squares, fields and yards connecting them.

**Dice:** The dice is rolled to move the pawn in clockwise direction.

**Strategy:** Player makes strategy to move its pawn from entry field to winning squares while avoiding pawn being kicked by opponents.
V. CONCLUSION

The main aim of this study was to get knowledge for the Ludo application development in flutter with giving brief description of the game theory. During the whole procedures, theory foundation of flutter development and its features were discussed and introduced to get the full extent of the experiment. Flutter being the cross-platform framework is loved by developers with having consistency, clean syntax, SDK level and widgets boosting the performance and the development cycle. People can play the game at any time anywhere with their loved ones and enjoy. In this digital world we will be able to save more trees by not using paper boards to play the games. Some of the advantages where it proved to be good when having mental health issues and in this COVID-19 period where everybody has to be in their home, it comes out to be the most played game ever.

FUTURE SCOPE

The current Ludo game Application is very optimized but still there are chance of improvements by adding some more features like playing with opponents from different place with internet connections or some cool points and rank level can also be implemented in the future work. Also, we can also implement where the game can start by not only getting “6” at dice but also with “1”. And there is continuous rolling of dice after getting “6” as a bonus move which we can also restrict that to “two” only. As we can get continuous “three” “6” and the turn is lost without any movement of pawn 10.

REFERENCE

1. Desktop / Mobile Share Worldwide, Desktop / Mobile Market Share Worldwide, Accessed 05 Jan 2021
2. A JavaScript Library, Open Source (facebook.com), Accessed 05 Jan 2021
4. Flutter VS React Native, Flutter vs React Native, Accessed 2 Jan 2021
5. Flutter Documentation, Flutter documentation - Flutter, Accessed 10 Jan 2021
6. Revolutionary about Flutter, , Accessed 05 Jan 2021
7. Flutter Architecture, Flutter architectural overview - Flutter, Accessed 19 Jan 2021
8. Dart Documentation, Language samples | Dart, Accessed 05 Jan 2021
11. Flutter FAQ, Flutter + Redux = Fludex, Accessed 05 Jan 2021