

# Implementation of LUDO Game in 2D using Python

G. Keerthana<sup>1</sup> and <sup>2</sup>S. Manoj<sup>2</sup>

Faculty-IT, Inurture Education Solutions Ltd, Bangalore<sup>1</sup>

M.Sc, Data science, Christ University, Bangalore<sup>2</sup>

keerthiguru28@gmail.com and manojdashiv05@gmail.com

**Abstract:** *The Ludo Game project is composed in Python. The job data has python manuscripts (Ludo.py, Ludo-Game.py) as well as photo data. This is a simple GUI based method board game which is really understandable and make use of. Speaking about the gameplay, all the playing policies are the same much like real time LUDO. This is a GUI based 2D multiplayer game. The game also consists of a high score module where the winner is assigned specific points based on the ranking and is stored in a database. The individual score is parsed from the variable to the SQL database. After starting the game, a GUI Ludo board shows up, other rules coincide. Initially, the player needs to chance. The main thing in this GUI based game is that the player simply has to push "Roll" to chance. On top of the board, it displays a dice with the number. The needs to keep on rolling up until there's a possible pawn to relocate. All the game motions are to be done by hand by the player. The 2D GUI is designed using Tkinter python libraries. The gameplay and GUI is designed in such a way that the user has no difficulty in locating and comprehending the contents of the gameplay. The GUI utilizes various photos as well as GIF documents to provide real-time experience. At the end of the game a GUI High score table is displayed for semi-quantitative analysis of the player's performance.*

## 1. INTRODUCTION

### 1.1 Python

The game is developed using **Python**. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

### 1.2 MySQL

MySQL is a fast, easy-to-use efficient Relational Database Management System. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages. It uses a standard form of well-known SQL data language. It works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. It supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB). It is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

### 1.3 Tkinter

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

The Tk interface is located in a binary module named `_tkinter`. This module contains the low-level interface to Tk, and should never be used directly by application programmers. It is usually a shared library (or DLL), but might in some cases be statically linked with the Python interpreter. In addition to the Tk interface module, Tkinter includes a number of Python modules. The two most important modules are the Tkinter module itself, and a module called Tkconstants. The former automatically imports the latter, so to use Tkinter, all you need to do is to import one module.

#### **1.4 Ludo**

Ludo is a board game played by 2-4 players. Each player is assigned a specific color and given four pieces. In this board game 2 to 4, players race their tokens from start to finish according to the dice rolls. Various variations are seen in the way people play Ludo. A Ludo board is in the shape of a cross, each arm being divided into three adjacent columns of six squares. The middle squares form the home column for each color and cannot be landed upon by other colors. The middle of the cross forms a large square which is the 'home' area and which is divided into 4 home triangles, one of each color. At each corner are colored areas where the pieces are placed to begin. The typical colors seen in Ludo boards are bright yellow, green, red, and blue. Each player is assigned a color and has four tokens of matching color. The players put their tokens in the starting circle. The movement of the tokens is determined by the dice. During game play the token moves clockwise from the starting square around the perimeter of the board, and up the player's home column to the finishing square. Players are not allowed to move a token out of the starting area unless they roll a 6 on the dice. In this game, players have to carefully play their token or they run the risk of sending it back to the starting point to start all over again. When a player 1's token lands on square that is already occupied by Player 2's token of different color, Player 2's token is returned to its starting point. Player who brings all their tokens to the finish wins the game.

## **2. REVIEW OF LITERATURE**

**2.1 Artificial Intelligence: Game Techniques Ludo - A Case Study** by Veenus Chhabra1 and Kuldeep, M.Tech Student Department of CSE, NGF College of Engineering & Technology, Palwal,.

In this paper the artificial intelligence ludo provides environment for playing to the artificial agents. Its main purpose lies in the area of artificial intelligence. At one side, Ludo is instead a very simple game and is fully observable but on the other side it contains a few challenges due to the stochastic and multi agent environment. Therefore, it offers a good balance between simplicity and complexity and is able to attract a wide audience and not only professionals.

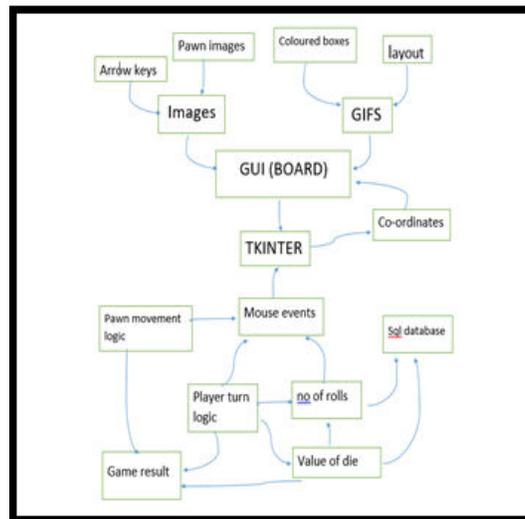
**2.2 TD ( $\lambda$ ) based Ludo player and also implemented a Q-learning based Ludo player using reinforcement learning** by Georgios N. Yannakakis and Carla P. Gomes

This paper explains a thorough description of Game Theory and its applications. He explained about AI and its four flagships, briefing about the terminology used currently in games and also gives connection between search algorithms with hard combinatorial problems. He grouped this connection under three things planning, duality and randomization.

**2.3 Complexity Analysis and Playing Strategies for Ludo and its Variant Race Games** by Faisal Alvi, Member IEEE, Moataz Ahmed.

In this paper the game is set up by designing classes representing entities involved in a typical Ludo game. Initially they tested the game setup by running numerous games with all four players selecting the random strategy. This was done to ensure that our game setup is correct and that each random player wins approximately equal number of times. They found that each random player wins  $25.0 \pm 1.0\%$  of the games and that this performance stabilized at nearly 5000 games. Running a higher number of games did not reduce the variation in results significantly, suggesting that 5000 games was an adequate number to observe reliable trends in strategy evaluation. The tests were also repeated for each strategy separately and identical results were obtained.

### 3. DESIGN METHODOLOGY OF PROPOSED WORK



**Figure 1:** Game board architecture

#### 3.1 Description of Game Board Architecture

- Board GUI: The interface of the Ludo game.
- Images and GIFS: Contains image files and gifs for the interface.
- Pawn movement logic: It contains the main game logic.
- Console: It displays value of die, co-ordinates and number of rolls

#### 3.2 Existing System

The current Ludo game is offline based multi-player game. Multiple players can play concurrently. The game interface is built in 2D format. The interface is an interactive window where the user's manipulations of pawns are handled. A console window is created as soon as the game starts and the console window maintains the records of current player's turn, value of dice and number of rolls and disclaimer after every turn. The player can interact with the game simply by using mouse events. The GUI is developed using Tkinter and various image files are used to create the interface. The interface is divided into various sections and each sections lead up to different functionalities.

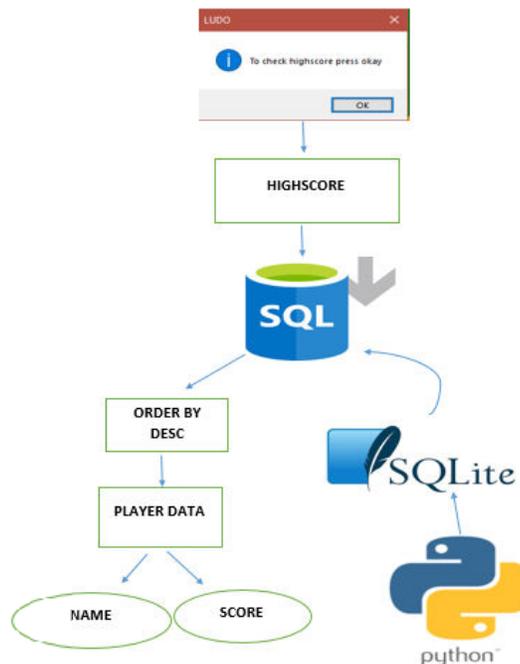
The limitations with the existing system is that the game has no dedicated storage allocated to store the user's information and performance statistics. The game has no information regarding the user so every time the game is executed it is like the game is running for the first time.

#### 3.3 Proposed System

To overcome the limitations of the existing system a database is created and a database connection is setup between the python scripts and the database. The user's information and performance statistics like individual's win score is retrieved from python script and the values are passed through SQL workbench to the database. The stored data is monitored and updated every time a game is played and now the game has information about the users. A high score table is displayed in a window at the end of each game where user's name and his individual score can be viewed.

#### 3.4 Description of Proposed System Architecture

- **Splash screen:** This screen prompts the user to check high score.
- **SQL,connector:** This is used to connect to the database
- **Player data table:** The user information is stored in this table.
- **SQL:** The database contains the player data table.



**Figure 2:** Proposed System Architecture

To overcome the problem of lack of storage of game information is to describe an external SQL database and parse variable values into the records of the table in the database. So the player's score values can be stored in a database in their respective rows and the scores can be retrieved and updated the next time the game is opened. A window is created to display the high score of all the individual players.

### A. Ludo game logic

A class is created to develop game logic and various methods are created to control pawn movements. The class contains numerous methods like turn, roll, clear, move check kill etc. First, the player's turn is determined. Then roll function is called to that player's color and the player can roll the die. Then the value of the die is checked. If it is 6, the player is asked to roll the die again. Else move method is called and the control of the die is sent to next color. The move method enables the user to manipulate the pawns when the method is called in roll method. Left Click is a method for mouse click events and it is called every time left click is clicked on valid co-ordinates. The mouse events is used to move the pawns by selecting on the valid pawns.

### B. Ludo Board Interface setup

To create the board first all the necessary image files are imported into the code and size for each images on the board is initialized. For each color different co-ordinates are set in the board. The pawn images are also given set of rules where it cannot lie on illegal co-ordinates. Mouse events ensured the pawns are moved by the user on the board in real time. A separate frame right next to board is created for user to roll the die by clicking on a button. A random number is defined every time the roll button is clicked. The interface opens up with a start dialog box to prompt the user to start the game. Then the game interface is called. At the end of the game once the user has won, it displays which user has won the game.

### C. Database Connectivity

To establish a connection with a SQL database, a platform needs to be installed so that the code can interact easily with the database. SQL workbench is installed to allow the easy interaction. A python library called sql.connector is

imported which can setup the connection with the database. First an Object is created insql class. Then the host, user id and password of the database account is passed as a parameter in connect method. The database is also chosen. When the database is chosen, a table is created and linked with the code. A SQL command is executed every game is finished, the command increments the winner's color's score value by 5 and the table is displayed in descending order.

### **3.5 Features of Proposed System**

- All the player's data will be stored and won't be deleted once the game is terminated
- The user can view their individual scores.
- High score table is displayed to compare the score with other players.
- A separate database is created for storing of user data
- The data stored cannot be deleted from the game.
- A dedicated window to display the stored data.

## **4. RESOURCE AND TOOLS**

### **4.1. Hardware Requirements**

- Processor: Intel i3 processor
- Hard Disk: 20GB
- Memory: 2GB RAM

### **4.2 Software Requirements**

- Python 3 or higher
- Python Package Installer
- JetBrainsPyCharm IDE
- MySQL Server
- MySQL workbench
- Sublime text Editor
- Tkinter Python Libraries

### **4.3 Tools and Platform**

#### **PyCharm IDE**

**PyCharm** provides smart code completion, code inspections, on-the-fly error highlighting and quick-fixes, along with automated code refactoring's and rich navigation capabilities.

#### **Features**

- Intelligent Code Editor
- Smart Code Navigation
- Fast and safe Refactorings.
- Debugging, Testing and Profiling
- VCS, Deployment and Remote Development
- Database Tools

#### **MySQL Workbench**

MySQL Workbench is a graphical tool for working with MySQL. MySQL Workbench provides an easy to use interface for performing the many tasks involved when working with databases. It integrates SQL development, administration, database design, creation and maintenance into one visual integrated development environment. MySQL Workbench is similar to SQL Server's SSMS, which is used for administering SQL Server

#### **Features:**

- SQL Editor

- SQL Code completion
- SQL Code formatter
- SQL Syntax highlighting
- SQL Code generation
- SQL Snippets
- SQL History

### **Tkinter**

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task. The Tk interface is located in a binary module named `_tkinter`. This module contains the low-level interface to Tk, and should never be used directly by application programmers. It is usually a shared library (or DLL), but might in some cases be statically linked with the Python interpreter. In addition to the Tk interface module, Tkinter includes a number of Python modules. The two most important modules are the Tkinter module itself, and a module called Tkconstants. The former automatically imports the latter, so to use Tkinter, all you need to do is to import one module:

### **Sublime Editor**

Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

- Multiple selections
- Command palette
- Powerful API and package ecosystem
- Customize anything
- Split editing
- Instant Project Switch
- Cross Platform
- Performance

## **5. IMPLEMENTATION AND RESULTS**

### **5.1 IMPLEMENTATION**

The game will be opened by executing the python script. Then the user will be asked to start the game by a prompt screen. Then the game interface is shown in a window and players can start playing the game.

### **5.2 CONFIGURATIONS**

- Python 3.7
- PIP
- MySQL workbench
- Sublime tools
- Sql.connector

### **5.3 TESTING**

In unit testing, the program modules that make up the system are tested individually. Unit testing focuses to locate errors in the working modules that are independent to each other. This testing is also used to ensure the integrity of the data stored. The various routines were checked by passing the inputs and the corresponding output is tested. Test cases used in the project as follows:

### **5.3 RESULTS**

When the game is opened, the user prompts the game to be started. Then the game interface is displayed and by default it is red's turn. The player must roll the die by clicking on the button in the top right. Until the value is 6, no pawns can be moved. Once it is 6, the same player gets another chance to roll and that value can be used to move the pawn to that many checks ahead. Then the control of the die shifts to the next color and same process goes on. When a player's pawn overlaps the other pawn, the latter pawn is sent to home box and the former pawn's players gets an another chance to roll the die. The one who ensures all the pawns get inside the center wins the game. As soon as the game is won, A dialog box appears showing who won the game and another window is opened which displays the score of each player's after the recent game. This is done by setting up a database connection with a table. The game's winner's score will be incremented by 5 in the table and the table will be displayed in descending order.

### **6. CONCLUSION**

This project implements a LUDO game in 2D GUI format. This application allows the user to play the game by simply opening the python script and prompting the game to start. The game interface is very simple and is easy to understand by the user. Since the game is built in 2D the game performance is fast and efficient . The game can also store user's score information by setting up a database connection from the python script. The individual's scores are displayed in a new window called highscore window where the players can compare each other's score.

### **7. FUTURE SCOPE**

As of now, the game can only store data in a database which can be accessed offline. The database can be set up to different host and the game can be played from any system but the score in the table will be updated in the database. The table can also be extended by creating more attributes like number of rolls, Probability of winning of each players. The table can also made to view by accessing specific IP address.

### **REFERENCES**

- [1]. ISSN 2325-4270 © TD( $\lambda$ ) and Q-learning based Ludo players
- [2]. ISSN 2325-4289 © Complexity analysis and playing strategies for Ludo and its variant race games