



L LOVELY
P ROFESSIONAL
U NIVERSITY

Transforming Education Transforming India

FINAL REPORT

For

JAVA PROJECT

Course Code:CSE310

Section: K21ZN

Project Title: LUDO GAME

Submitted By:

Name	Roll No.	Registration No.
Praveen Kumar Tiwari	RK21ZNB44	12104582
Rahul Singh Chouhan	RK21ZNB56	12112405
Prakash Sah	RK21ZNB62	12112811

Submitted To:

Professor Chandani Bhasin

ABSTRACT

This is a simple GUI based strategy board game which is very easy to understand and use. All the playing rules are the same just like we play in real time ludo. This is a simple 2D multi player game. After starting the game, a GUI ludo board appears, other rules are the same. First, the player has to roll the dice. The main thing in this simple GUI based game is that the player just has to press “Roll” to roll the dice. At the top of the board, it displays a dice with the number. The player has to keep on rolling until there’s a possible pawn to move.

All the game movements are to be performed manually by the player. A simple 2D GUI is provided for easy game play. The game play design is so simple that user won’t find it difficult to use and understand.

LITERATURE REVIEW

2D Ludo is a computer program that imitates the manual method of playing the ludo board game. The motivation behind this project work is the need to strengthen the understanding of processes and digital literacy and more specifically their reflective understanding of the video games .This system is achieved by writing a computer program that allows players to roll a dice randomly, take decisions and move the pawn based on the outcome of the dice on a well-designed graphical interface. The design of the adequate gaming simulation system helps and plays an important role supporting the organizational learning, changing mental models, fostering alternative interpretation patterns of the reality. It is also helpful in developing new communication and action patterns, and reconstructing the sociotechnical aspects of people. Furthermore, any game involving numbers can aid the children early mathematics development. 2D LUDO has two modules which are Player Module, and Game Module. First of all, Player’s Module is that players are ones to roll the die and to click on the coins to move them according to the number on the dice. The Game’s Module is the one that provides all the features like the dice, coins, directions which are necessary to play the game.

1. INTRODUCTION

Pachisi originated in India by the 6th century. In this project, the basic ludo that is present in the real world is computerized i.e., the program that is written in this project creates a board which is quite similar to that of the board that is seen everywhere. Special areas of the ludo board are typically coloured yellow, green, blue, red. Each player is given a colour and possesses four tokens(coins)of one colour in their game. The board is normally square with a cross-shaped game track, with each arm of the cross consisting of three columns with six squares per each column. The middle column consists of five coloured squares which represent the player's respective home column. A sixth coloured square which is not on the home column represents the player's starting square. At the centre of the board is a large finishing square often composed of triangles in the four colours a top the player's home column thus forming the arrows pointing to the finish.

1.1 Background

Ludo game systems have been around almost since the time cars were invented. In any area where there is a significant amount of traffic, there are car parking systems. Ludo game were developed in the early 20th century in response to the need for storage space for vehicles. A ludo game system automates a car parking system. It optimizes parking space and make processes efficient. It gives real time car parking information such as vehicle & slot counts, available slots display, reserved parking. pay-and-park options, easy payments, reports, and a host of other features.

1.2 Objectives

2 to 4 players begin by placing their respective pieces in their bases. Each takes turns throwing the die, and the player with the highest roll plays first. The players to the left follow in turn going clockwise. On each player's turn, the player rolls the die to determine a move. The goal of the game is to move all four of the player's pieces clockwise once around the board, up the home column, and into the home triangle. To begin, a player must roll a six to move a piece out of the base and onto the start position. That piece is then in play. The player cannot make any other moves until at least one piece is in play.

1.3 Applicability

Social media was abuzz today after it was reported that the Bombay High Court will decide whether Ludo is a game of skill or sheer luck. There was plenty of mockery going around but as it turns out, Ludo is a serious matter. If Ludo is declared a game of luck, then should stakes be involved, then it could be categorised as gambling. Muley had initially approached the local police in November of 2020 seeking action against the makers of an online Ludo game, Private Ltd, under the Gambling Act and under relevant sections of the IPC.

2. PROBLEM IDENTIFICATION

Generally, we observed that it is troublesome to play the game in real world as it is needed to do some of the physical works like rolling the dice, moving the coin.

SOLUTION FOR PROBLEM DEFINITION

A solution is designed for this problem. This project involves representing the board of ludo which is very similar to that of the board in the real world. Here the can be able to easily understand the rules of the game as these rules are quite similar to that of the rules of ludo game. All that the player has to do is to watch where he/she wants to go and to roll the dice by clicking the button which shows "Roll Die".

The player can easily play and win the game.

2.1 DRAWBACKS

Some of the disadvantages of board games that you must know about or must have experienced while playing these are:

Board games can become addictive and distracting: Long after the game is over, you may find yourself thinking about what you could do better or to win if lost the game. Also, you may feel drawn to play these again and again just to improve the scores.

Board games are too time-consuming: Many gameplays are very long and may extend for hours. If you do not play with cognition, you can feel absorbed in these the whole day without even realizing the time spent. Board games can be stressful sometimes: Especially when you are too touchy about losing you may find stress building up in mind while playing board games.

Can build an undue sense of competition when feeling is not controlled: The board games become fun only when you play these with a constructive frame of mind. Bringing egos in gameplay and using board games as a premise to prove supremacy can defeat the very idea behind playing these games. Such an attitude can also lead to fights that may cause discord in a meeting/party/class.

Board games are relatively costlier a bit than other alternatives: Some board games are quite pricey and do not deliver the value as compared to the size of the value these carry.

2.2 Limitations of the System

As every study has its limitation, the limitations of this study are time constraints, insufficient skill and the limited access to materials on the subject on ground.

*Security is somewhere an issue, there is no voice command encryption in this project.

*Background voice can interfere

*Misinterpretation because of accents and may cause inaccurate results.

* ELON cannot be called externally anytime like other traditional assistance .

CODE USED IN LUDO GAME :

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

//Player class to store player information
class Player {
    int height, width, status, coin;

    Pawn[] pa = new Pawn[4]; // Array of 4 pawns for each player

    public Player(int height, int width) {
        status = -1;
        coin = 0;
        for (inti = 0; i < 4; i++) { // Initializing pawns for each player
            pa[i] = new Pawn(height, width);
        }
    }

    public void draw(Graphics2D g) {
    }
}

//Pawn class to store pawn information
class Pawn {
    int x, y;
    int current;
    int height, width;

    public Pawn(int h, int w) { // Constructor to initialize pawn
        current = -1;
        x = -1;
        y = -1;
        height = h;
        width = w;
    }

    // Function to draw pawn on board
    public void draw(Graphics2D g, inti, int j, int play) {
        if (current == -1) { // If pawn is not on board
            int temp1 = 80 + (height / 2), temp2 = 50 + (width / 2); // Initial position of pawn
            x = i; // x coordinate of pawn
            y = j; // y coordinate of pawn
            if (play == 0) { // If player 1
                g.setColor(Color.RED); // Color of pawn
            }
        }
    }
}
```

```

elseif(play==1){ //Ifplayer2
g.setColor(Color.GREEN); //Colorofpawn
}
elseif(play==2){ //Ifplayer3
g.setColor(Color.YELLOW); //Colorofpawn
}
elseif(play==3){ //Ifplayer4
g.setColor(Color.BLUE); //Colorofpawn
}

g.fillOval(temp1+5+(i*width),temp2+5+(j*height),width-10,height-10); //Drawingpawnonboard
g.setStroke(newBasicStroke(2)); //Thicknessofborder
g.setColor(Color.BLACK); //Colorofborder
g.drawOval(temp1+5+(i*width),temp2+5+(j*height),width-10,height-10); //Drawingborderofpawn
}

else
{ //Ifpawnonboard
inttemp1=80,temp2=50; //Initialpositionofboard
x=Path.ax[play][current]; //xcoordinateofpawn
y=Path.ay[play][current]; //ycoordinateofpawn
if(play==0){
g.setColor(Color.RED);
}
elseif(play==1){
g.setColor(Color.GREEN);
}
elseif(play==2){
g.setColor(Color.YELLOW);
}
elseif(play==3){
g.setColor(Color.BLUE);
}

g.fillOval(temp1+5+(x*width),temp2+5+(y*height),width-10,height-10); //Drawingpawnonboard
g.setStroke(newBasicStroke(2)); //Thicknessofborder
g.setColor(Color.BLACK);
g.drawOval(temp1+5+(x*width),temp2+5+(y*height),width-10,height-10); //Drawingborderofpawn
}
}
}

//Pathclasstostorepathinformation
classPath{

```

```

static int[][] ax = { //x coordinates of path for each player
    { 1,2,3,4,5,6,6,6,6,7,8,8,8,8,9,10,11,12,13,14,14,13,12,11,10,9,8,8,8,8,7,6,6,6,6,6,5,4,3,2,1,0,0,1,2,3,4,5,6 },
    { 8,8,8,8,9,10,11,12,13,14,14,13,12,11,10,9,8,8,8,8,7,6,6,6,6,6,5,4,3,2,1,0,0,1,2,3,4,5,6,6,6,6,6,7,7,7,7,7,7 },
    { 13,12,11,10,9,8,8,8,8,8,7,6,6,6,6,6,5,4,3,2,1,0,0,0,1,2,3,4,5,6,6,6,6,6,7,8,8,8,8,9,10,11,12,13,14,14,13,12,11,10,9,8 },
    { 6,6,6,6,5,4,3,2,1,0,0,0,1,2,3,4,5,6,6,6,6,6,7,8,8,8,8,9,10,11,12,13,14,14,13,12,11,10,9,8,8,8,8,7,7,7,7,7,7 }
};

static int[][] ay = { //y coordinates of path for each player
    { 6,6,6,6,5,4,3,2,1,0,0,0,1,2,3,4,5,6,6,6,6,7,8,8,8,8,9,10,11,12,13,14,14,13,12,11,10,9,8,8,8,8,7,7,7,7,7,7 },
    { 1,2,3,4,5,6,6,6,6,7,8,8,8,8,9,10,11,12,13,14,14,13,12,11,10,9,8,8,8,8,7,6,6,6,6,6,5,4,3,2,1,0,0,1,2,3,4,5,6 },
    { 8,8,8,8,9,10,11,12,13,14,14,13,12,11,10,9,8,8,8,8,7,6,6,6,6,6,5,4,3,2,1,0,0,1,2,3,4,5,6,6,6,6,6,7,7,7,7,7,7 },
    { 13,12,11,10,9,8,8,8,8,8,7,6,6,6,6,6,5,4,3,2,1,0,0,0,1,2,3,4,5,6,6,6,6,6,7,8,8,8,8,9,10,11,12,13,14,14,13,12,11,10,9,8 }
};

static int[][] initialx = { //x coordinates of initial position of each player
    { 1,1,3,3 },
    { 10,10,12,12 },
    { 10,10,12,12 },
    { 1,1,3,3 }
};

static int[][] initialy = { //y coordinates of initial position of each player
    { 1,3,1,3 },
    { 1,3,1,3 },
    { 10,12,10,12 },
    { 10,12,10,12 }
};

}

//Layout class to draw board
class Layout{
    int x,y,width,height; //x,y coordinates and width and height of board
    public Layout(int xi,int yi){
        x=xi;
        y=yi;
        width=30;
        height=30;
    }

    //Method to draw board
    public void draw(Graphics2D g){
        g.setColor(Color.WHITE); //Color of background of board
        g.fillRect(x,y,15*width,15*height); //Drawing background of board
        for(int i=0;i<6;i++){ //drawing all the inner squares

```

```

g.setColor(Color.RED);
g.fillRect(x+(i*width),y,width,height);
g.fillRect(x,y+(i*height),width,height);
g.fillRect(x+(i*width),y+(5*height),width,height);
g.fillRect(x+(5*width),y+(i*height),width,height);
g.setColor(Color.GREEN);
g.fillRect(x+((i+9)*width),y,width,height);
g.fillRect(x+(9*width),y+(i*height),width,height);
g.fillRect(x+((i+9)*width),y+(5*height),width,height);
g.fillRect(x+(14*width),y+(i*height),width,height);
g.setColor(Color.YELLOW);
g.fillRect(x+((i+9)*width),y+(9*height),width,height);
g.fillRect(x+(9*width),y+((i+9)*height),width,height);
g.fillRect(x+((i+9)*width),y+(14*height),width,height);
g.fillRect(x+(14*width),y+((i+9)*height),width,height);
g.setColor(Color.BLUE);
g.fillRect(x+(i*width),y+(9*height),width,height);
g.fillRect(x,y+((i+9)*height),width,height);
g.fillRect(x+(i*width),y+(14*height),width,height);
g.fillRect(x+(5*width),y+((i+9)*height),width,height);
}

for(int i=1;i<6;i++){ //Drawingtheinnerboxes
g.setColor(Color.RED);
g.fillRect(x+(i*width),y+(7*height),width,height);
g.setColor(Color.YELLOW);
g.fillRect(x+((8+i)*width),y+(7*height),width,height);
g.setColor(Color.GREEN);
g.fillRect(x+(7*width),y+(i*height),width,height);
g.setColor(Color.BLUE);
g.fillRect(x+((7)*width),y+((8+i)*height),width,height);
}

g.setColor(Color.RED);
g.fillRect(x+(1*width),y+(6*height),width,height);
g.setColor(Color.YELLOW);
g.fillRect(x+((13)*width),y+(8*height),width,height);
g.setColor(Color.GREEN);
g.fillRect(x+(8*width),y+(1*height),width,height);
g.setColor(Color.BLUE);
g.fillRect(x+((6)*width),y+((13)*height),width,height);
int temp1=x+45,temp2=y+45;
for(int i=0;i<2;i++){ //Drawingtheouterboxes

```



```

for(int j=0;j<2;j++){
    g.setColor(Color.RED);
    g.fillRect(temp1+(2*i*width),temp2+(2*j*height),width,height);
    g.setColor(Color.YELLOW);
    g.fillRect(temp1+(2*i*width)+9*width,temp2+(2*j*height)+9*height,width,height);
    g.setColor(Color.GREEN);
    g.fillRect(temp1+(2*i*width)+9*width,temp2+(2*j*height)+0*height,width,height);
    g.setColor(Color.BLUE);
    g.fillRect(temp1+(2*i*width)+0*width,temp2+(2*j*height)+9*height,width,height);
}
}

g.setColor(Color.RED);
intxpoints0[]={x+(6*width),x+(6*width),x+15+(7*width)};
intypoints0[]={y+(6*height),y+(9*height),y+15+(7*width)};
intnpoints=3;
g.fillPolygon(xpoints0,ypoints0,npoints);//Drawingthetriangleinthecenter
g.setColor(Color.YELLOW);
intxpoints1[]={x+(9*width),x+(9*width),x+15+(7*width)};
intypoints1[]={y+(6*height),y+(9*height),y+15+(7*width)};
intnpoints1=3;
g.fillPolygon(xpoints1,ypoints1,npoints1);//Drawingthetriangleinthecenter
g.setColor(Color.GREEN);
intxpoints2[]={x+(6*width),x+(9*width),x+15+(7*width)};
intypoints2[]={y+(6*height),y+(6*height),y+15+(7*width)};
intnpoints2=3;
g.fillPolygon(xpoints2,ypoints2,npoints2);//Drawingthetriangleinthecenter
g.setColor(Color.BLUE);
intxpoints3[]={x+(6*width),x+(9*width),x+15+(7*width)};
intypoints3[]={y+(9*height),y+(9*height),y+15+(7*width)};
intnpoints3=3;
g.fillPolygon(xpoints3,ypoints3,npoints3);//Drawingthetriangleinthecenter
g.setStroke(newBasicStroke(2));
g.setColor(Color.BLACK);
for(int i=0;i<3;i++){ //Drawingtheouterboxes
    for(int j=0;j<6;j++){
        g.drawRect(x+((i+6)*width),y+(j*height),width,height);
        g.drawRect(x+((j)*width),y+((i+6)*height),width,height);
        g.drawRect(x+((i+6)*width),y+((j+9)*height),width,height);
        g.drawRect(x+((j+9)*width),y+((i+6)*height),width,height);
    }
}

```

```

g.drawRect(x+((1)*width),y+(1*height),4*width,4*height);
g.drawRect(x+((10)*width),y+(1*height),4*width,4*height);
g.drawRect(x+((1)*width),y+(10*height),4*width,4*height);
g.drawRect(x+((10)*width),y+(10*height),4*width,4*height);
g.drawRect(x,y,15*width,15*height);
for(int i=0;i<2;i++){
for(int j=0;j<2;j++){
g.drawRect(temp1+(2*i*width),temp2+(2*j*height),width,height);
g.drawRect(temp1+(2*i*width)+9*width,temp2+(2*j*height)+9*height,width,height);
g.drawRect(temp1+(2*i*width)+9*width,temp2+(2*j*height)+0*height,width,height);
g.drawRect(temp1+(2*i*width)+0*width,temp2+(2*j*height)+9*height,width,height);
}
}

g.drawPolygon(xpoints0,ypoints0,npoints);
g.drawPolygon(xpoints1,ypoints1,npoints1);
g.drawPolygon(xpoints2,ypoints2,npoints2);
g.drawPolygon(xpoints3,ypoints3,npoints3);
g.drawOval(x+5+(6*width),y+5+(2*height),width-10,height-10); //Drawing the circles in the center
g.drawOval(x+5+(12*width),y+5+(6*height),width-10,height-10); //Drawing the circles in the center
g.drawOval(x+5+(8*width),y+5+(12*height),width-10,height-10);
g.drawOval(x+5+(2*width),y+5+(8*height),width-10,height-10);
g.setFont(new Font("InkFree",Font.BOLD,40));
g.drawString("RED",90,35); //Drawing the text on top of the boxes
g.drawString("GREEN",370,35);
g.drawString("Blue",90,540);
g.drawString("Yellow",370,540);
g.setFont(new Font("MVBoli",Font.BOLD,30));
g.drawString("Press enter to roll dice",550,150);
g.drawString("Click on the token to move",550,200);
}
}

```

```

//This is the class for initializing the player positions
class Build_Player{
Player[] pl=new Player[4];
int[][] initialX={ //This is the array for the initial x position of the players
{1,1,3,3},
{10,10,12,12},
{10,10,12,12},
{1,1,3,3}
};
}

```

```

int[][]initialy={ //Thisisthearrayfortheinitialypositionsoftheplayers

    { 1,3,1,3 },

    { 1,3,1,3 },

    { 10,12,10,12 },

    { 10,12,10,12 }

};

//Thisistheconstructorforinitializingtheplayerpositions
publicBuild_Player(intheight,intwidth){

for(inti=0;i<4;i++){

    pl[i]=newPlayer(height,width);

}

}

//Thisisthemethodfordrawingtheplayers
publicvoiddraw(Graphics:Dg){

for(inti=0;i<4;i++){

for(intj=0;j<4;j++){

    pl[i].pa[j].draw(g,initialx[i][j],initialy[i][j],i);

}

}

}

}

//Thisistheclassforthehegamemoves
classGameMovesextendsJPanelimplementsKeyListener,ActionListener,MouseListener{

    privatestaticfinallongserialVersionUID=1L;

    Layoutla;//Thisistheobjectforthelayoutclass

    Build_Playerp;//Thisistheobjectforthebuildplayerclass

    Timertime;//Thisistheobjectforthetimerclass

    intdelay=10;//Thisisthedelayforthetimer

    intcurrent_player,dice;//Thisisthevariablefortheplayerandthedice

    intflag=0,roll,kill=0;//Thisistheflagforthedicerollandthekillvariable

    publicGameMoves(){

        setFocusTraversalKeysEnabled(false);

        requestFocus();

        current_player=0;//firstplayerissetored

        la=newLayout(80,50);

        p=newBuild_Player(la.height,la.width);

        dice=0;

        flag=0;

```

```

roll=0;
kill=0;
}

//Thisisthemethodfordrawingthegame

publicvoidpaint(Graphicsg){
la.draw((Graphics2D)g);
p.draw((Graphics2D)g);
if(p.pl[current_player].coin==4){ //Thisistheconditionforthewinner
g.setColor(Color.WHITE);
g.fillRect(590,100,380,130);
if(current_player==0){
g.setColor(Color.RED);
}
elseif(current_player==1){
g.setColor(Color.GREEN);
}
elseif(current_player==2){
g.setColor(Color.YELLOW);
}
elseif(current_player==3){
g.setColor(Color.BLUE);
}

g.setFont(newFont("serif",Font.BOLD,40));
g.drawString("Player"+(current_player+1)+"wins.",600,350);
g.drawString("Congratulations.",600,400);
current_player=0;
la=newLayout(80,50);
p=newBuild_Player(la.height,la.width);
dice=0;
flag=0;
roll=0;
kill=0;
}

elseif(dice!=0){ //Thisistheconditionforthediceroll

g.setColor(Color.pink);
g.fillRect(590,300,260,200);
if(current_player==0){
g.setColor(Color.RED);

```

```

    }

    elseif(current_player==1){
    g.setColor(Color.GREEN);
    }

    elseif(current_player==2){
    g.setColor(Color.YELLOW);
    }

    elseif(current_player==3){
    g.setColor(Color.BLUE);
    }

    }

    g.setFont(newFont("MVBoli",Font.BOLD,30));
    if(current_player==0){
    g.drawString("RED's turn:",600,350);
    }

    elseif(current_player==1){
    g.drawString("GREEN's turn:",600,350);
    }

    elseif(current_player==2){
    g.drawString("YELLOW's turn:",600,350);
    }

    elseif(current_player==3){
    g.drawString("BLUE's turn:",600,350);
    }

    }

    //drawdice
    g.setColor(Color.BLACK);
    g.drawRect(670,370,100,100);
    g.setFont(newFont("MVBoli",Font.BOLD,50));
    g.drawString(""+dice,700,440);
    }

    if(flag==0&&dice!=0&&dice!=6&&kill==0){
    current_player=(current_player+1)%4;
    }

    kill=0;
    }

    publicvoidkeyPressed(KeyEvent){
    if(e.getKeyCode()==KeyEvent.VK_ENTER&&flag==0){ //ifenterispressedthediceisrolled
    roll=0;

```

```

dice=1+(int)(Math.random()*6);//random number is generated for the dice roll
repaint();
for(int i=0;i<4;i++){
if(p.pl[current_player].pa[i].current!=-1&&p.pl[current_player].pa[i].current!=56&&(p.pl[current_player].pa[i].current+dice)<=56){
flag=1;
break;
}
}

if(flag==0&&dice==6){
for(int i=0;i<4;i++){
if(p.pl[current_player].pa[i].current==-1){
flag=1;
break;
}
}
}
}
}

public void mouseClicked(MouseEvent e){
if(flag==1){
int x=e.getX();
int y=e.getY();

x=x-80;
y=y-50;
x=x/30;
y=y/30;

int value=-1;

//System.out.println(x+" "+y);

if(dice==6){ //if dice is 6
for(int i=0;i<4;i++){
if(p.pl[current_player].pa[i].x==x&&p.pl[current_player].pa[i].y==y&&(p.pl[current_player].pa[i].current+dice)<=56){
value=i;
flag=0;
break;
}
}

if(value!=-1){ //if the player has a coin on the clicked position
p.pl[current_player].pa[value].current+=dice;
if(p.pl[current_player].pa[value].current==56){
p.pl[current_player].coin++;

```

```

}

intk=0;

inthou=p.pl[current_player].pa[value].current;

if((hou%13)!=0&&(hou%13)!=8&&hou<51)

{

for(inti=0;i<4;i++){

if(i=current_player){

for(intj=0;j<4;j++){

inttem1=Path.ax[current_player][p.pl[current_player].pa[value].current],tem2=Path.ay[current_player][p.pl[current_player].pa
[value].current];

if(p.pl[i].pa[j].x==tem1&&p.pl[i].pa[j].y==tem2){

p.pl[i].pa[j].current=-1;

kill=1;

k=1;

break;

}

}

}

if(k==1)

break;

}

}

}

else{//iftheplayerdoesnothaveacoinontheclickedposition

for(inti=0;i<4;i++){

if(p.pl[current_player].pa[i].current===-1){

p.pl[current_player].pa[i].current=0;

flag=0;

break;

}

}

}

else{//ifdiceisnot6

for(inti=0;i<4;i++){

if(p.pl[current_player].pa[i].x==x&&p.pl[current_player].pa[i].y==y&&(p.pl[current_player].pa[i].current+dice)<=56){

value=i;

flag=0;

break;

}

}

}

if(value!=-1){//iftheplayerhasacoinontheclickedposition

```

```

p.pl[current_player].pa[value].current+=dice;
if(p.pl[current_player].pa[value].current==56){
p.pl[current_player].coin++;
}

intk=0;

inthou=p.pl[current_player].pa[value].current;

if((hou%13)!=0&&(hou%13)!=8&&hou<51)
{
for(inti=0;i<4;i++){
if(i=current_player){
for(intj=0;j<4;j++){

inttem1=Path.ax[current_player][p.pl[current_player].pa[value].current],tem2=Path.ay[current_player][p.pl[current_player].pa
[value].current];

if(p.pl[i].pa[j].x==tem1&&p.pl[i].pa[j].y==tem2){
p.pl[i].pa[j].current=-1;
kill=1;
k=1;
break;
}
}
}
if(k==1)
break;
}
}
}

repaint();
}
}

//thesefunctionsarenotusedbutarecompulsorytobedefinedastheyareabstractfunctionsofMouseListener
publicvoidactionPerformed(ActionEvent){
}

publicvoidkeyReleased(KeyEventarg0){
}

publicvoidkeyTyped(KeyEventarg0){
}

```



```
publicvoidmouseEntered(MouseEventarg0){  
}
```

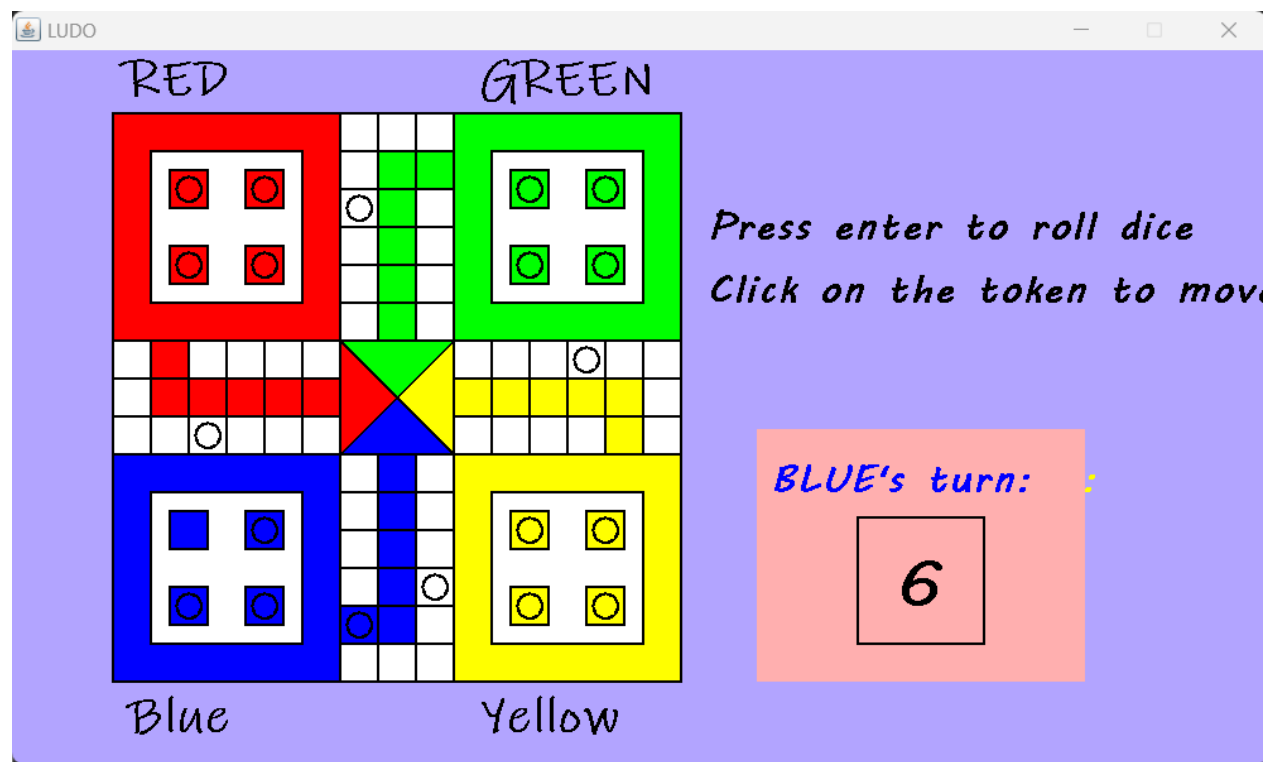
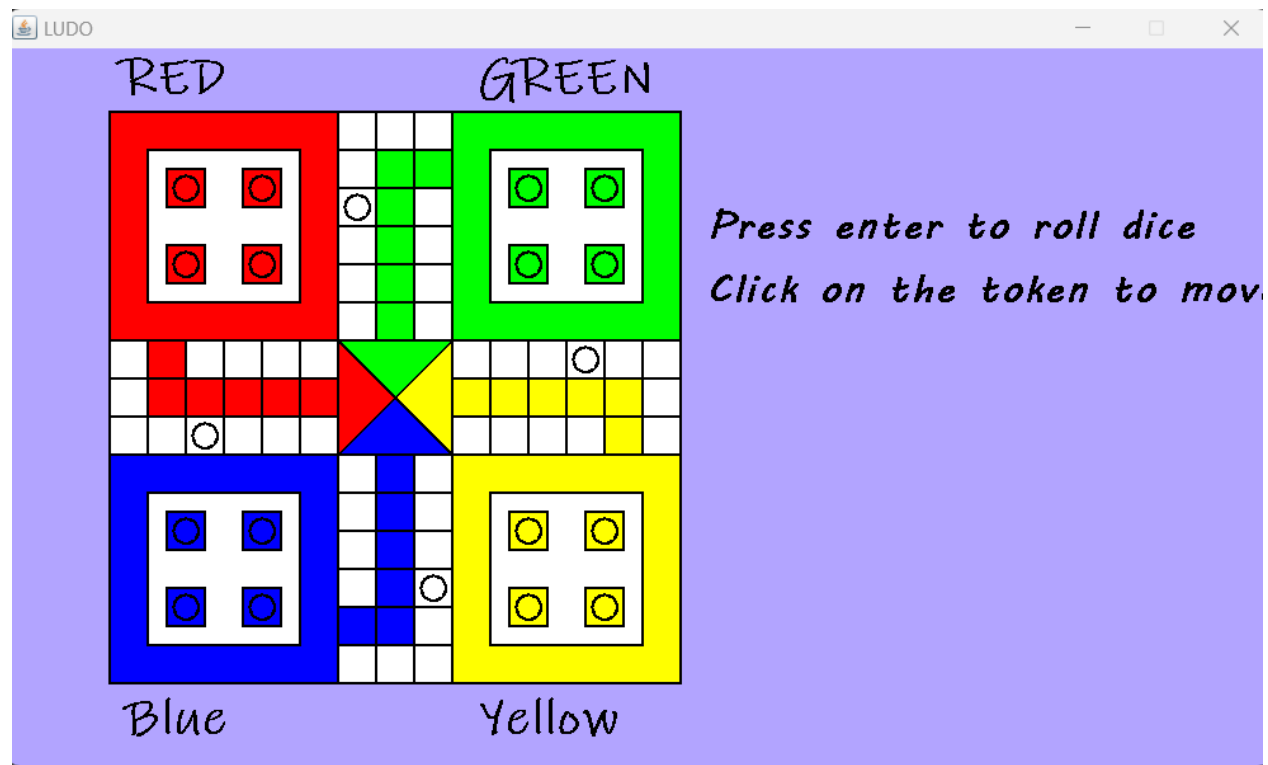
```
publicvoidmouseExited(MouseEventarg0){  
}
```

```
publicvoidmousePressed(MouseEvente){  
}
```

```
publicvoidmouseReleased(MouseEventarg0){  
}  
}
```

```
classGameScreen{  
//Runtheapplicationfromhere  
publicstaticvoidmain(String[]args){  
JFramejframe=newJFrame();  
jframe.setBounds(10,10,1000,600);  
jframe.setBackground(newColor(0XB2A4FF));  
jframe.setTitle("LUDO");  
jframe.setResizable(false);  
jframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
GameMovesgm=newGameMoves();  
gm.setFocusable(true);  
gm.addKeyListener(gm);  
gm.addMouseListener(gm);  
jframe.add(gm);  
jframe.setVisible(true);  
}  
}
```

OUTPUT OF THE CODE :



Future Scope of the Project

Ludo mobile game is one of the hottest mobile games right now. We all played this iconic game in our childhood as a kid, at that time it was popular as a board game but with the digital revolution and the people's demand for new emerging mobile technologies made several app development companies start developing ludo mobile applications. This is why the ludo mobile game app development has become one of the demanded things right now. Other board games, like, Ludo, Snakes and Ladders have been very notable since decades as 'family game". While this snappy paced life doesn't take into account the family to sit together and play these games, it directly comes in the propelled arrangement as a convenient mobile. These future scope will be great use.

Conclusion

The game of Ludo provides a platform for people of varied age groups to interact with each other. The young ones learn to count as they move their tokens in their turns. Children learn to strategize and make decisions to win the game. Each player is assigned a colour and has four tokens of matching colour. 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 .