Course Name: **OBJECT ORIENTED PROGRAMMING (C++)**

Project Name: **LUDO GAME**

Group Members Name:

HARSHAL CHAVAN (111903036)
NIKHIL REPALE (111903050)
NIKITA TIPULE (111903051)

1. In General

1.1. What does program do?

-It creates an UI for playing the LUDO game.

-It asks for the number of players that wanted to play(2,3,4).

-It declares the winner accordingly.

1.2. Main structure

-Main class ludo under which path_init, play, movement , flood_fill and fill_color functions are defined.

-Structure pawn and user are used to store respective data.

-UI is created under the functions flood_fill, fill_color, dice, circle, star and ludo constructor.

2. Explanations of the Structures

2.1. Pawn

-int pos stores the position of pawn (where it is located).
-int status will stores 0 if pawn is reached to the home else 1.
-int sum stores the number of blocks that pawn has gone through.

2.2. User

-Array of pawns will store 4 pawns for each user.
-int def, stores the starting position of the pawn.
-int color, stores the color code of the pawn.
-int win, stores the number of pawns that have been reached to the home.

3. Explanations of class LUDO

3.1 Ludo constructor

This function will create an UI for the Game. It also fills color in rectangle, circle, star and diamond shaped figures.
We have done calculations before locating the positions of each figure in UI and after that we have called the corresponding function.

3.2 Star and circle functions

There are 6 lines in a star and we have given their location w.r.t. x and y axis to the function line and then called this function 8 times in ludo so that 8 stars will be seen in UI.

Under the cir function circle is called 4 times so that 4 pawns of the give color for user will be seen in UI.

3.3 flood_fill and flood_filll

flood_fill function takes x and y coordinate ,where we want fill the color, then it replaces the old color with new color. This function is basically for filling the color in layout.

Flood_filll takes the same arguments and it performs the function of filling color in the dice.

3.4 Fill_horizontal and Fill_vertical

Fill_horizontal and fill vertical is for filling color in horizontal and vertical boxes of 30 X 30 respectively.

3.5 path_init

This function will initialize the path of pawns i.e. it will store x and y co-ordinate of the path into an array.
There are 18 x and y co-ordinates for each pawn. We have created for loops to take their co-ordinates for each horizontal and vertical line.

3.6 home_asgn

This function will check for the pawns color and accordingly creates a path towards its home.

Four for loops and four array are there to store x and y co-ordinate of the path for four different players.

3.7 home

This function will move the pawns to its home area.

Checks for the pawn color then moves it to the home and increases the win for the corresponding player.

3.8 dice

It is the function to draw dice and fill it with appropriate color.

Takes two arguments prev and curr which are previous and current dice outputs respectively.

According to the dice output it draws the circle on dice and fills color in it.

3.9 movement

This function takes two arguments viz. initial position of the pawn and the final position of pawn where the pawn is supposed to move.

The function is basically for moving the pawn.

3.10 Play

Actual function where does it ask for number of players and accordingly creates player by allocating memory.

Then it initialize the player. It generates random number from one to six.
Now we want to move the pawn.

1. If the number generated is not six for the 1st time then we can't move the pawn.
2. If the pawn is already at home then we can't move the pawn.
3. If the pawn is moved in such a place where already a different colored pawn is present then it will eat (except the position of stars). And the second pawn will again come to the initial position.
4. It will keep on increasing the count of the pawn for the boxes it had gone through. And if the count becomes greater than 50 then it will assign it to the home.
5. It also checks the pawn we want to move is came out or not.
6. According to the number of pawns in home , it decides the winner.