

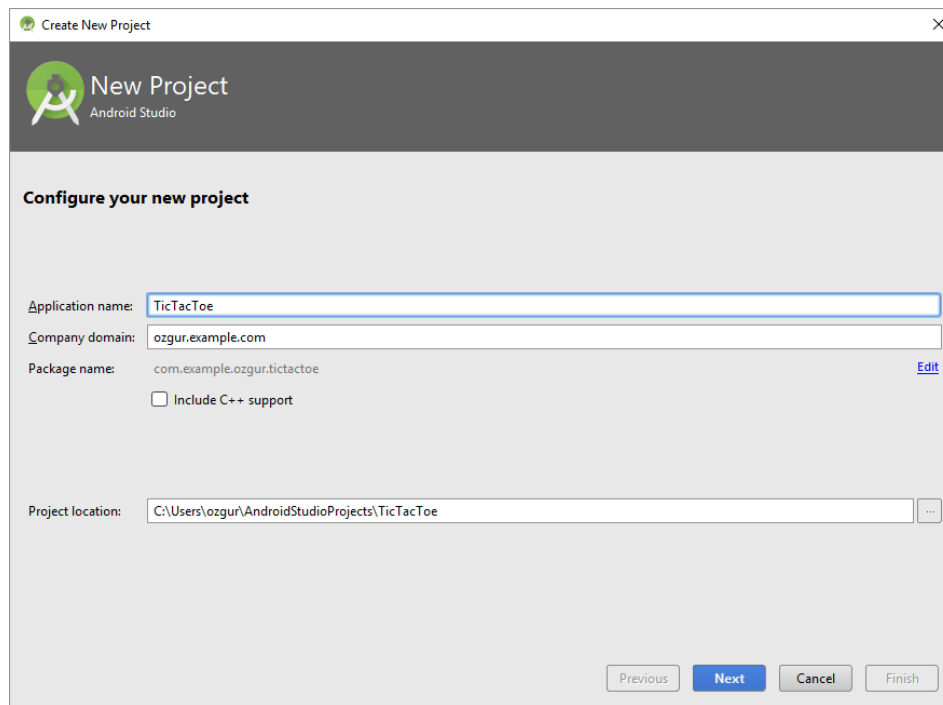
## Lab 5: Tic Tac Toe

In this lab we will be implementing the Tic Tac Toe application.

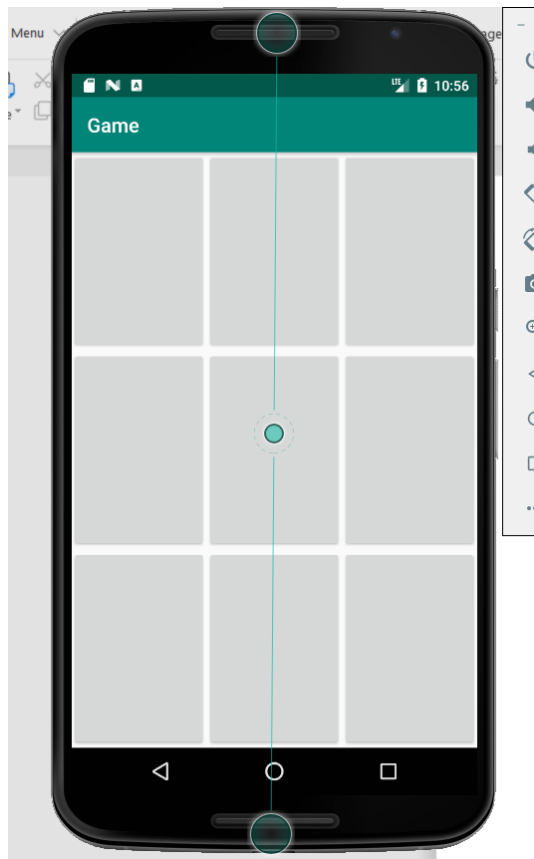


## Create Project Tic Tac Toe

1. Create new project in Android Studio.



2. Add 3 rows and 3 columns of buttons with the same size as show below using the designer



3. Define constants for player symbols and boolean for turn

```
static final String PLAYER_1 = "X";  
static final String PLAYER_2 = "O";  
boolean player1Turn = true;
```

4. Define a data structure to hold the board state

```
byte[][] board = new byte[3][3];
```

5. Create a listener to listen button events

```
class CellListener implements View.OnClickListener{  
    int row,col;  
  
    public CellListener(int row, int col){  
        this.row = row;  
        this.col = col;  
    }  
  
    @Override  
    public void onClick(View v) {  
        if (player1Turn) {  
            ((Button)v).setText(PLAYER_1);  
            board[row][col] = 1;  
        }else{  
            ((Button)v).setText(PLAYER_2);  
            board[row][col] = 2;  
        }  
    }  
}
```

6. Register listeners to buttons

```
TableLayout table = findViewById(R.id.board);  
for (int i = 0; i<3; i++) {  
    TableRow row = (TableRow) table.getChildAt(i);  
    for (int j = 0; j < 3; j++) {  
        Button btn = (Button) row.getChildAt(j);  
        btn.setOnClickListener(new CellListener(i, j));  
    }  
}
```

7. Check valid moves by implementing the following method and calling it accordingly

```
public boolean isValidMove(int row, int col){  
    //check valid move  
    return false;  
}
```

8. Check game ended by implementing the following method and calling it accordingly

```
public int gameEnded(int row, int col){  
    //check columns  
  
    //check rows  
  
    //check diagonals  
  
    return -1;  
}
```

9. Disable the board and show a toast message informing the game result to the user when game ends

10. Add a menu to restart the game

