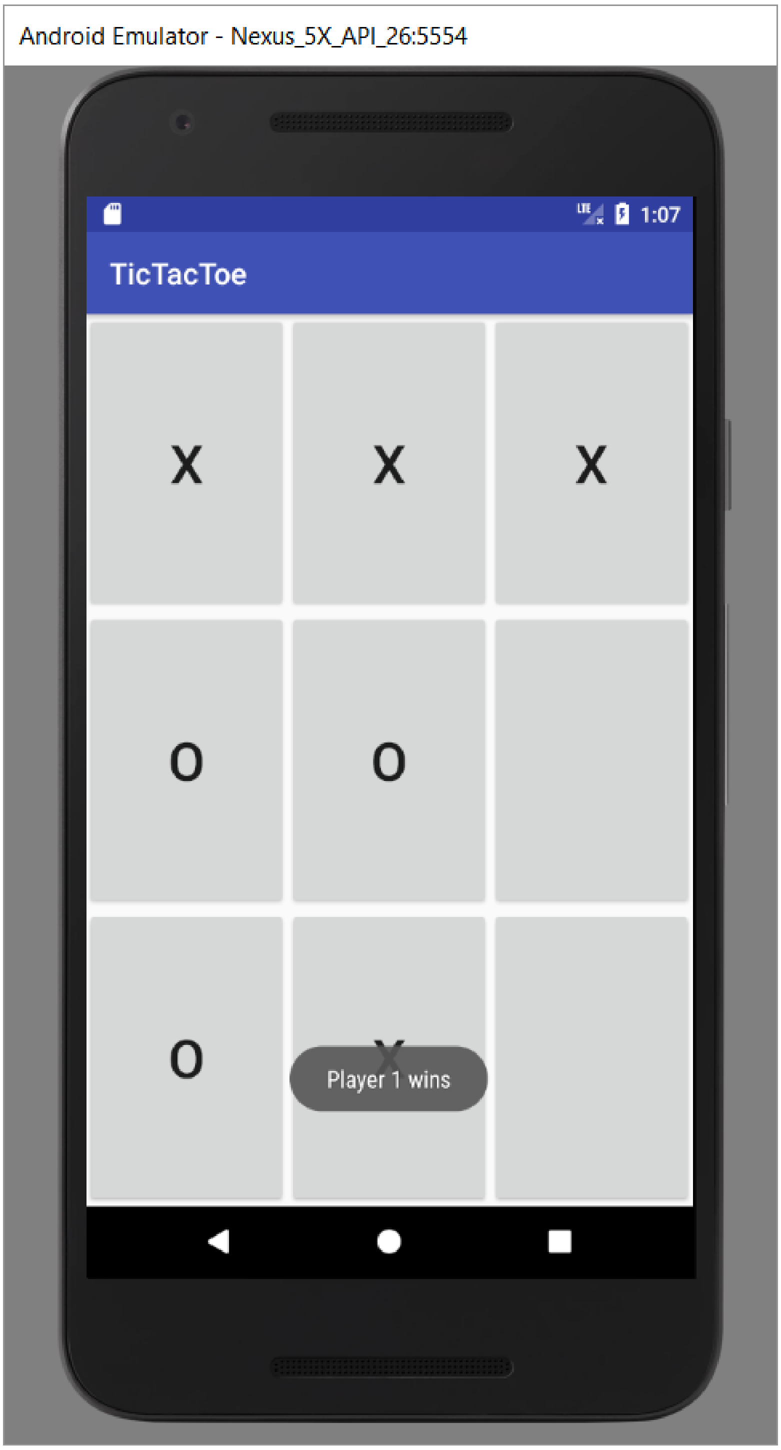
Lab 7: Tic Tac Toe MVP

In this lab we will be implementing the Tic Tac Toe applying the MVP architecture.



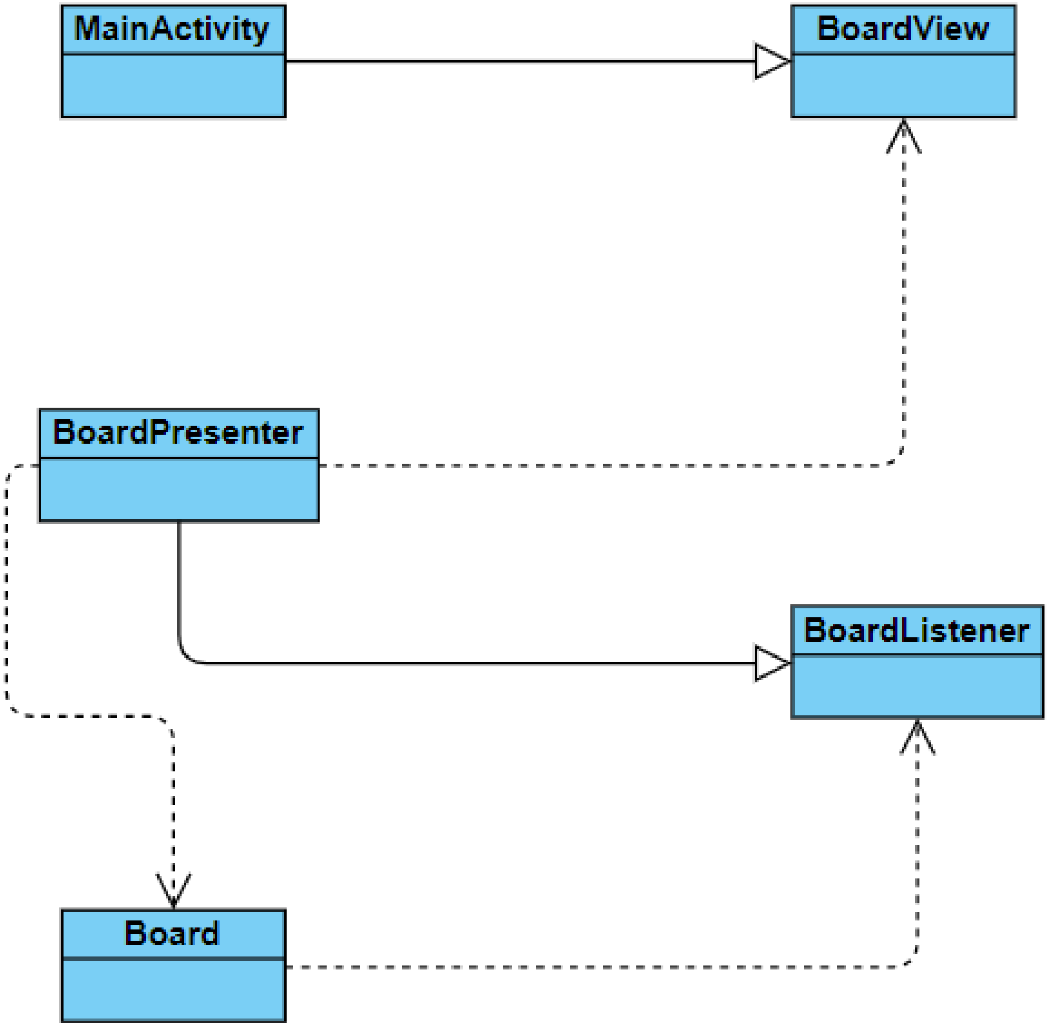
Android Emulator - Nexus_5X_API_26:5554

TicTacToe

X	X	X
O	O	
O	X	

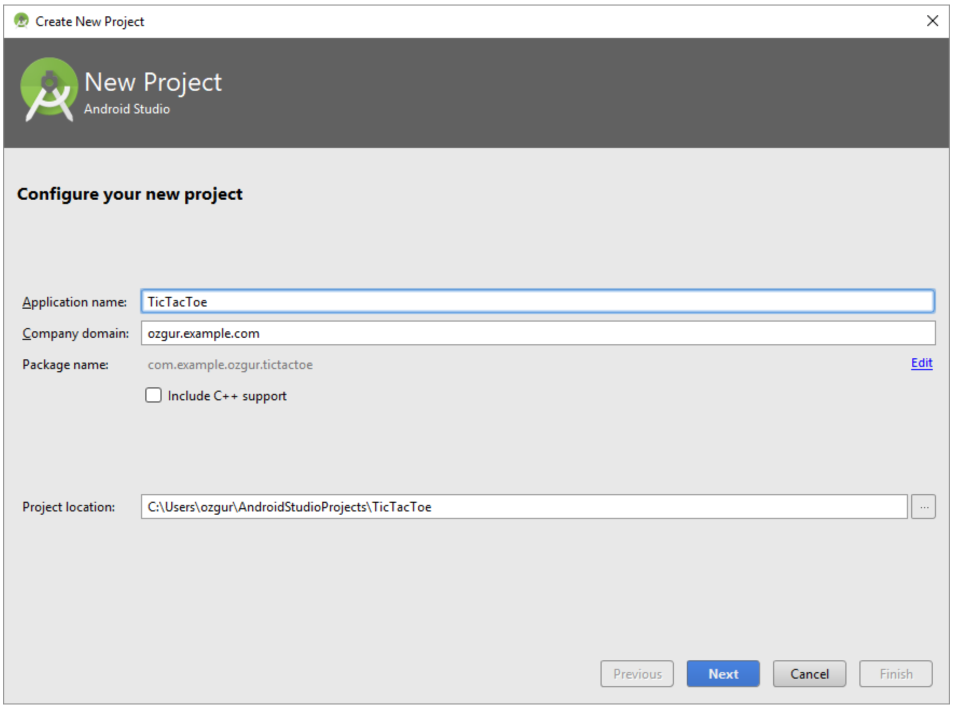
Player 1 wins

Class Diagram



Create Project Tic Tac Toe

1. Create new project in Android Studio.



Create New Project

New Project
Android Studio

Configure your new project

Application name: TicTacToe

Company domain: ozgur.example.com

Package name: com.example.ozgur.tictactoe

□ Include C++ support

Project location: C:\Users\ozgur\AndroidStudioProjects\TicTacToe

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

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:id="@+id/board"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   tools:context="tr.edu.mu.ceng.gui.tictactoe.MainActivity">
   <TableRow
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_weight="1">
      <Button
         android:id="@+id/btn00"
         android:layout_width="wrap_content"
         android:layout_height="match_parent"
         android:layout_weight="1"
         android:textSize="36sp" />
      <Button
```

```
            android:id="@+id/btn01"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:textSize="36sp" />
        <Button
            android:id="@+id/btn02"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:textSize="36sp" />
    </TableRow>
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1">
        <Button
            android:id="@+id/btn10"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:textSize="36sp" />
        <Button
            android:id="@+id/btn11"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:textSize="36sp" />
        <Button
            android:id="@+id/btn12"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:textSize="36sp" />
    </TableRow>
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1">
        <Button
            android:id="@+id/btn20"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:textSize="36sp" />
        <Button
            android:id="@+id/btn21"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:textSize="36sp" />
        <Button
            android:id="@+id/btn22"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:textSize="36sp" />
    </TableRow>
</TableLayout>
```

3. Create an interface for View as shown below

```
public interface BoardView {

    char PLAYER_1_SYMBOL = 'X';
    char PLAYER_2_SYMBOL = 'O';

    byte DRAW = 0;
    byte PLAYER_1_WINNER = 1;
    byte PLAYER_2_WINNER = 2;

    void newGame();

    void putSymbol(char symbol, byte row, byte col);

    void gameEnded(byte winner);
}
```

4. Create an Interface for BoardListener

```
public interface BoardListener {
    byte NO_ONE = 0;
    byte PLAYER_1 = 1;
    byte PLAYER_2 = 2;

    void playedAt(byte player, byte row, byte col);

    void gameEnded(byte winner);
}
```

5. Create the Board class as shown below, note that this is a primitive implementation and we will complete it during the lab hour.

```
public class Board {
    private static byte PLAYER_1_SYMBOL = 1;
    private static byte PLAYER_2_SYMBOL = 2;
    private boolean player1Turn = true;
    byte [][] board = new byte [3][3];
    BoardListener boardListener;
    public Board(BoardListener listener) {
        boardListener = listener;
    }
    public void move(byte row, byte col) {
        if (player1Turn){
            board[row][col] = PLAYER_1_SYMBOL;
            boardListener.playedAt(BoardListener.PLAYER_1,row,col);
        }else{
            board[row][col] = PLAYER_2_SYMBOL;
            boardListener.playedAt(BoardListener.PLAYER_2,row,col);
        }
        player1Turn = !player1Turn;
    }
}
```

6. Modify the Main Activity as shown below

```
public class MainActivity extends Activity implements BoardView{

    BoardPresenter presenter;
    TableLayout boardView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
       presenter = new BoardPresenter(this);
        boardView = findViewById(R.id.board);
        //register button listeners
        for (byte row=0; row<3; row++){
            TableRow tableRow = (TableRow)boardView.getChildAt(row);
            for (byte col=0; col<3; col++){
                Button button =  (Button)tableRow.getChildAt(col);
                BoardPresenter.CellClickListener clickListener = new
BoardPresenter.CellClickListener(presenter,row,col);
                button.setOnClickListener(clickListener);
                presenter.addCellClickListener(clickListener);
            }
        }
    }
    @Override
    public void newGame() {
        TableLayout boardView = findViewById(R.id.board);
        for (int row=0; row<3; row++){
            TableRow tableRow = (TableRow)boardView.getChildAt(row);
            for (int col=0; col<3; col++){
                Button button =  (Button)tableRow.getChildAt(col);
                button.setText("");
                button.setEnabled(true);
            }
        }
    }
    @Override
    public void putSymbol(char symbol, byte row, byte col) {
        TableRow tableRow = (TableRow)boardView.getChildAt(row);
        Button button =  (Button)tableRow.getChildAt(col);
        button.setText(Character.toString(symbol));
    }
    @Override
    public void gameEnded(byte winner) {
        for (int row=0; row<3; row++){
            TableRow tableRow = (TableRow)boardView.getChildAt(row);
            for (int col=0; col<3; col++){
                Button button =  (Button)tableRow.getChildAt(col);
                button.setText("");
                button.setEnabled(false);
            }
        }
        switch (winner){
            case BoardView.DRAW :
                Toast.makeText(this,"Game is Draw", Toast.LENGTH_LONG).show();
                break;
            case BoardView.PLAYER_1_WINNER :
                Toast.makeText(this,"Player 1 Wins", Toast.LENGTH_LONG).show();
                break;
            case BoardView.PLAYER_2_WINNER :
                Toast.makeText(this,"Player 2 Wins", Toast.LENGTH_LONG).show();
                break;
        }
    }
}
```

7. Create BoardPresenter class as shown below

```
class BoardPresenter implements BoardListener {
   private BoardView boardView;
   private Board board;
   private List<CellClickListener> cellClickListeners = new ArrayList<>();
   public BoardPresenter(BoardView view) {
      this.boardView = view;
      board = new Board(this);
   }
   private void move(byte row, byte col) {
      board.move(row,col);
   }
   public void addCellClickListener(CellClickListener listener){
      cellClickListeners.add(listener);
   }
   @Override
   public void playedAt(byte player, byte row, byte col) {
      if (player == BoardListener.PLAYER_1){
         boardView.putSymbol(BoardView.PLAYER_1_SYMBOL, row, col);
      }else if(player == BoardListener.PLAYER_2){
         boardView.putSymbol(BoardView.PLAYER_2_SYMBOL, row, col);
      }
   }
   @Override
   public void gameEnded(byte winner) {
      switch (winner){
         case BoardListener.NO_ONE :
            boardView.gameEnded(BoardView.DRAW);
         case BoardListener.PLAYER_1 :
            boardView.gameEnded(BoardView.PLAYER_1_WINNER);
         case BoardListener.PLAYER_2 :
            boardView.gameEnded(BoardView.PLAYER_2_WINNER);
      }
   }
   static class CellClickListener implements View.OnClickListener{
      BoardPresenter presenter;
      byte row;
      byte col;
      public CellClickListener( BoardPresenter presenter, byte row, byte col){
         this.row = row;
         this.col = col;
         this.presenter = presenter;
      }
      @Override
```

```
        public void onClick(View view){
            Log.d("CellClickListener", "at" + row + ", " + col);
            presenter.move(row,col);
        }
    }
}
```