Toss n' Toe

Submitted to:

Mr. Vince Lorenz Dela Rea

Submitted by:

Genaro Howard S. Gaborni

Christian Fernandez

Ralph Aian B. Escote


7:38 PM
TOSS N' TOE
Tic Tac Toe
Coin Flip
Bye Bye!
A GAME BY:
GABORNI, GENARO HOWARD
FERNANDEZ, CHRISTIAN
ESCOTE, RALPH AIAN


7:25 PM
Tic-Tac-Toe
Player Wins: 2
AI Wins: 0
Draws: 1
Main Menu

7:25 PM

Coin Flip

Your choice is:

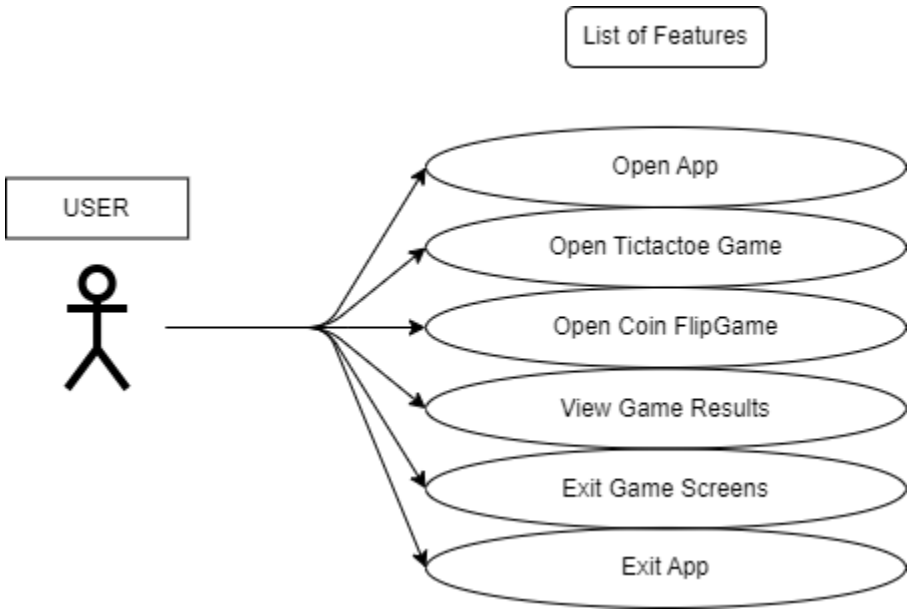
Heads | Tails

Reset

Main Menu

Functions of the application:

Toss n’ Toe is a simple app with 2 games, In Tictactoe you are against an ai by playing a game of tictactoe, you can see the results of your matches at the game screen below the board, and can exit back to the main menu. At the coin flip game, you can pick a choice between heads or tails, and depending on the randomization code in the app, it will compare its result with your choice and tell you that you win when your choices match with the result and you lose when it isn't. After you are done playing you can exit the app.



```dart
import 'dart:math';
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:flutter/services.dart';
import 'dart:async';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Gaborni,Fernandez,Escote',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primaryColor: Colors.teal.shade100,
      ),
      home: MainMenu(),
    );
  }
}
```

```dart
class MainMenu extends StatefulWidget {
  MainMenu({super.key});

  @override
  _MainMenuState createState() => _MainMenuState();
}

class _MainMenuState extends State<MainMenu> {
  String _currentBackground = 'assets/bgnamed.png';

  void exitApp(BuildContext context) {
    if (Platform.isAndroid) {
      SystemNavigator.pop();
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Stack(
        children: [
          // Background Image
          Positioned.fill(
            child: Image.asset(
              _currentBackground,
              fit: BoxFit.fill,
            ),
          ),
```

```
57              Center(
58                child: Padding(
59                  padding: EdgeInsets.symmetric(horizontal: 40.0),
60                  child: Column(
61                    mainAxisSize: MainAxisSize.min,
62                    children: [
63                      Image(image: AssetImage("assets/logo.png"),),
64                      SizedBox(height: 50),
65                      ElevatedButton(
66                        style: ElevatedButton.styleFrom(
67                            backgroundColor: Colors.transparent, // Transparent background
68                            foregroundColor: Colors.black,
69                            minimumSize: Size(120, 50),
70                            side: BorderSide(color: Colors.black, width: 2),
71                            shadowColor: Colors.transparent,
72                            shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(20.0))
73                        ),
74                        onPressed: () {
75                          Navigator.push(
76                            context,
77                            MaterialPageRoute(builder: (context) => GameScreen()),
78                          );
79                        },
80                        child: Text(
81                          "Tic Tac Toe",
82                          style: TextStyle(fontSize: 24, color: Colors.black),
83                        ),
84                      ),
```

```
85                      SizedBox(height: 20),
86                      ElevatedButton(
87                        style: ElevatedButton.styleFrom(
88                            backgroundColor: Colors.transparent, // Transparent background
89                            foregroundColor: Colors.black,
90                            minimumSize: Size(120, 50),
91                            side: BorderSide(color: Colors.black, width: 2),
92                            shadowColor: Colors.transparent,
93                          shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(20.0))
94                        ),
95                        onPressed: () {
96                          Navigator.push(
97                            context,
98                            MaterialPageRoute(builder: (context) => CoinFlipGame()),
99                          );
100                       },
101                       child: Text(
102                         "Coin Flip",
103                         style: TextStyle(fontSize: 24, color: Colors.black,),
104                       ),
105                     ),
106                     SizedBox(height: 70),
107                     Container(
108                       width: 100,
109                       height: 50,
110                       child: ElevatedButton(
111                         style: ElevatedButton.styleFrom(
112                           minimumSize: Size(double.infinity, 20),
```

```
                          shadowColor: Colors.black,
                          elevation: 5.0,
                        backgroundColor: Colors.cyanAccent,
      shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(20.0))
                      ),
                      onPressed: () => exitApp(context),
                      child: Text(
                        "Bye Bye!",
                        style: TextStyle(fontSize: 15, color: Colors.black),
                      ),
                    ),
                  )
                ],
              ),
            ),
          ),
        ],
      ),
    );
  }
}

class GameScreen extends StatefulWidget {
  GameScreen({super.key});

  @override
  State<GameScreen> createState() => GameScreenState();
}
```

```
class GameScreenState extends State<GameScreen> {
  List<String> board = List.filled(9, '');
  bool isPlayerTurn = true;
  int playerWins = 0;
  int aiWins = 0;
  int draws = 0;
  bool isGameOver = false;
  String resultMessage = '';

  @override
  void initState() {
    super.initState();
    loadScores();
  }

  Future<void> loadScores() async {
    final prefs = await SharedPreferences.getInstance();
    setState(() {
      playerWins = prefs.getInt('playerWins') ?? 0;
      aiWins = prefs.getInt('aiWins') ?? 0;
      draws = prefs.getInt('draws') ?? 0;
    });
  }

  Future<void> saveScores() async {
    final prefs = await SharedPreferences.getInstance();
    prefs.setInt('playerWins', playerWins);
```

```
    prefs.setInt('aiWins', aiWins);
    prefs.setInt('draws', draws);
  }

  void resetBoard() {
    setState(() {
      board = List.filled(9, '');
      isPlayerTurn = true;
      isGameOver = false;
      resultMessage = '';
    });
  }

  void playerMove(int index) {
    if (board[index] == '' && !isGameOver) {
      setState(() {
        board[index] = 'O';
        isPlayerTurn = false;
      });
      checkWinner();
      if (!isGameOver) {
        aiMove();
      }
    }
  }

  void aiMove() {
    int move = findBestMove();
```

```
    if (move != -1) {
      setState(() {
        board[move] = 'X';
        isPlayerTurn = true;
      });
      checkWinner();
    }
  }

  int findBestMove() {
    // 30% chance na random para manalo naman whahahha
    if (Random().nextDouble() < 0.3) {
      List<int> availableMoves = [];
      for (int i = 0; i < board.length; i++) {
        if (board[i] == '') {
          availableMoves.add(i);
        }
      }
      return availableMoves[Random().nextInt(availableMoves.length)];
    }

    int bestScore = -999;
    int move = -1;

    for (int i = 0; i < board.length; i++) {
      if (board[i] == '') {
        board[i] = 'X';
        int score = minimax(board, 0, false);
```

```
            board[i] = '';
            if (score > bestScore) {
              bestScore = score;
              move = i;
            }
          }
        }
        return move;
      }

      int minimax(List<String> newBoard, int depth, bool isMaximizing) {
        String winner = getWinner(newBoard);
        if (winner != '') {
          if (winner == 'X') return 10 - depth;
          if (winner == 'O') return depth - 10;
          return 0;
        }

        if (isMaximizing) {
          int bestScore = -999;
          for (int i = 0; i < newBoard.length; i++) {
            if (newBoard[i] == '') {
              newBoard[i] = 'X';
              int score = minimax(newBoard, depth + 1, false);
              newBoard[i] = '';
              bestScore = max(score, bestScore);
            }
          }
          return bestScore;
        } else {
          int bestScore = 999;
          for (int i = 0; i < newBoard.length; i++) {
            if (newBoard[i] == '') {
              newBoard[i] = 'O';
              int score = minimax(newBoard, depth + 1, true);
              newBoard[i] = '';
              bestScore = min(score, bestScore);
            }
          }
          return bestScore;
        }
      }

      String getWinner(List<String> boardToCheck) {
        List<List<int>> winPatterns = [
          [0, 1, 2],
          [3, 4, 5],
          [6, 7, 8],
          [0, 3, 6],
          [1, 4, 7],
          [2, 5, 8],
          [0, 4, 8],
          [2, 4, 6],
        ];

```

```
280         for (var pattern in winPatterns) {
281           String a = boardToCheck[pattern[0]];
282           String b = boardToCheck[pattern[1]];
283           String c = boardToCheck[pattern[2]];
284           if (a == b && b == c && a != '') {
285             return a;
286           }
287         }
288
289         if (!boardToCheck.contains('')) {
290           return 'draw';
291         }
292
293         return '';
294       }
295
296       void checkWinner() {
297         String winner = getWinner(board);
298         if (winner != '') {
299           setState(() {
300             isGameOver = true;
301             if (winner == 'O') {
302               resultMessage = 'You Win!';
303               playerWins++;
304             } else if (winner == 'X') {
305               resultMessage = 'You Lose!';
306               aiWins++;
```

```
307             } else {
308               resultMessage = "It's a Draw!";
309               draws++;
310             }
311             saveScores();
312           });
313           showResultDialog();
314         }
315       }
316
317       void showResultDialog() {
318         showDialog(
319           barrierDismissible: false,
320           context: context,
321           builder: (context) => AlertDialog(
322             title: Text(
323               resultMessage,
324               textAlign: TextAlign.center,
325             ),
326             content: ElevatedButton(
327               style: ElevatedButton.styleFrom(
328                 shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(5.0)),
329                 backgroundColor: Colors.teal,
330               ),
331               onPressed: () {
332                 Navigator.of(context).pop();
333                 resetBoard();
```

```
            },
            child: Text('Play Again',
            style: TextStyle(color: Colors.black,shadows: [
              Shadow(
                blurRadius:1.0,  // shadow blur
                color: Colors.white, // shadow color
                offset: Offset(.5,.5), // how much shadow will be shown
              ),
            ],),)
          ),
        ),
      );
    }

    Widget buildBoard() {
      return AspectRatio(
        aspectRatio: 1,
        child: GridView.builder(

          itemCount: board.length,
          gridDelegate:
          SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 3),
          padding: EdgeInsets.all(16.0),
          itemBuilder: (context, index) {
            return GestureDetector(
              onTap: () => playerMove(index),
              child: Container(
                decoration: BoxDecoration(
```

```
                  border: Border.all(color: Colors.white),
                ),
                child: Center(

                  child: Text(
                    board[index],
                    style: TextStyle(
                      color: board[index] == 'O' ? Colors.blue : Colors.red,
                      fontSize: 64,
                      fontWeight: FontWeight.bold,
                    ),
                  ),
                ),
              ),
            ),
          );
        },
      ),
    );
  }

  Widget buildScoreBoard() {
    return Padding(
      padding: EdgeInsets.symmetric(vertical: 24.0),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text(
```

```
              'Player Wins: $playerWins',
              style: TextStyle(fontSize: 24),
            ),
            SizedBox(height: 8),
            Text(
              'AI Wins: $aiWins',
              style: TextStyle(fontSize: 24),
            ),
            SizedBox(height: 8),
            Text(
              'Draws: $draws',
              style: TextStyle(fontSize: 24),
            ),
          ],
        ),
      );
    }

    @override
    Widget build(BuildContext context) {
      return Scaffold(
        backgroundColor: Colors.teal[100],
        appBar: AppBar(
          title: Text('Tic-Tac-Toe'),
          centerTitle: true,
          automaticallyImplyLeading: false,
        ),
```

```
        body: Stack(
          children: [
            Positioned.fill(
            child: Image.asset(
              "assets/bg.png",
              fit: BoxFit.fill,
            ),
          ),
            Center(
              child: Column(
                children: [
                  buildBoard(),
                  buildScoreBoard(),
                  ElevatedButton(
                    style: ElevatedButton.styleFrom(
                        shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(5.0))
                    ),
                    onPressed: () {
                      Navigator.pop(context);
                    },
                    child: Text('Main Menu',style: TextStyle(fontSize: 15, color: Colors.black),),
                  ),
                ],
              ),
            )
          ],
        ),
```

```
444       }
445     }
446
447     // CoinFlipGame
448     class CoinFlipGame extends StatefulWidget {
449       CoinFlipGame({super.key});
450
451       @override
452       _CoinFlipGameState createState() => _CoinFlipGameState();
453     }
454
455     class _CoinFlipGameState extends State<CoinFlipGame> {
456       String result = '';
457       String outcomes = '';
458       bool _isButtonDisabled = false;
459       String _imagePath = 'assets/neutral.png'; // Start with the neutral image
460       String chos = '';
461
462       void _flipCoin(String choice) {
463         setState(() {
464           _isButtonDisabled = true;
465           _imagePath = 'assets/neutral.png'; // Show neutral image during flipping
466         });
467
468         Timer(Duration(seconds: 1), () {
469           final random = Random();
470           final outcome = random.nextBool() ? 'Heads' : 'Tails';
```

```
472           setState(() {
473             outcomes = outcome;
474             result = outcome == choice ? 'You Win!' : 'You Lose!';
475             _imagePath = outcome == 'Heads' ? 'assets/flip-Heads.gif' : 'assets/flip-Tails.gif';
476             _isButtonDisabled = false;
477             chos = choice;
478           });
479         });
480       }
481
482       void _resetGame() {
483         setState(() {
484           outcomes = '';
485           result = '';
486           chos = '';
487           _imagePath = 'assets/neutral.png'; // Reset to neutral image
488         });
489       }
490
491       @override
492       Widget build(BuildContext context) {
493         return Scaffold(
494           backgroundColor: Colors.teal[100],
495           appBar: AppBar(
496             title: Text('Coin Flip'),
497             centerTitle: true,
498             automaticallyImplyLeading: false,
499           ),
```

```
      body: Stack(
        children: [
          Positioned.fill(child: Image.asset("assets/bg.png",fit: BoxFit.fill,),),
          Center(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Image.asset(
                  _imagePath,
                  height: 150,
                ),
                SizedBox(height: 20),
                Text(
                  outcomes,
                  style: TextStyle(fontSize: 50, fontWeight: FontWeight.bold,
                    shadows: [
                      Shadow(
                        blurRadius:20.0,  // shadow blur
                        color: Colors.white, // shadow color
                        offset: Offset(1.0,1.0), // how much shadow will be shown
                      ),
                    ],
                  ),
                ),
                SizedBox(height: 20),
                Text(
                  result,
                  style: TextStyle(fontSize: 32, fontWeight: FontWeight.bold),
```

```
                ),
                SizedBox(height: 40),
                Container(
                  child: Row(
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: [
                      Text("Your choice is: "),
                      Text("$chos", style: TextStyle(decoration: TextDecoration.underline),),
                    ],
                  ),
                ),
                Container(
                  child: Row(
                    mainAxisAlignment: MainAxisAlignment.center,
                    children: [
                      ElevatedButton(
                        style: ElevatedButton.styleFrom(
                            shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(5.0))
                        ),
                        onPressed: _isButtonDisabled ? null : () => _flipCoin('Heads'),
                        child: Text('Heads',style: TextStyle(fontSize: 15, color: Colors.lightBlueAccent,
                          shadows: [
                            Shadow(
                              blurRadius:1.0,  // shadow blur
                              color: Colors.black, // shadow color
                              offset: Offset(1.0,1.0), // how much shadow will be shown
                            ),
                          ],),),),
```

```
                    ),
                    SizedBox(width: 10),
                    ElevatedButton(
                      style: ElevatedButton.styleFrom(
                          shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(5.0))
                      ),
                      onPressed: _isButtonDisabled ? null : () => _flipCoin('Tails'),
                      child: Text('Tails',style: TextStyle(fontSize: 15, color: Colors.cyanAccent,
                        shadows: [
                          Shadow(
                            blurRadius:1.0,  // shadow blur
                            color: Colors.black, // shadow color
                            offset: Offset(1.0,1.0), // how much shadow will be shown
                          ),
                        ],
                      ),
                      ),
                    ),
                  ],
                ),
              ),
              SizedBox(height: 20),
              ElevatedButton(
                style: ElevatedButton.styleFrom(
                    shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(5.0))
                ),
                onPressed: _resetGame,
```

```
              ElevatedButton(
                style: ElevatedButton.styleFrom(
                    shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(5.0))
                ),
                onPressed: _resetGame,
                child: Text('Reset',style: TextStyle(color: Colors.redAccent),),
              ),
              SizedBox(height: 40),
              ElevatedButton(
                style: ElevatedButton.styleFrom(
                    shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(5.0))
                ),
                onPressed: () {
                  Navigator.pop(context);
                },
                child: Text('Main Menu',style: TextStyle(fontSize: 15, color: Colors.black),),
              ),
            ],
          ),
        )
      ],
    ),
  );
 }
}
```