TITLE: Math Tic Tac Toe

Dairy Number: 15963/2023-CO/SW

TIC TAC TOE

Tic Tac Toe is a classic paper-and-pencil game played on a 3x3 grid. Two players take turns marking cells with their respective symbols, usually "X" and "O." The objective is to get three of your symbols in a row, either horizontally, vertically, or diagonally.

Here are the basic rules of the game:

- The game starts with an empty grid of 3x3 cells.
- Player 1 is usually assigned the symbol "X," and Player 2 is assigned "O."
- Players take turns, starting with Player 1.
- Each player chooses an empty cell and marks it with their symbol.
- The game continues until one of the following conditions is met:

a. One player gets three of their symbols in a row horizontally, vertically, or diagonally. This player wins the game.

b. All cells are filled, and no player has achieved three in a row. The game is then considered a draw.

If a winning condition is met, the game ends, and the player who achieved the winning move is declared the winner.

If the game ends in a draw, no player wins.

NOVELTY IN APPLICATION

Math helps children develop their critical thinking, problem-solving abilities, number sense, spatial reasoning, and arithmetic skills. It also improves their language and communication, fosters confidence, and sets them up for future academic success.

This application has been developed keeping in mind not just the congenial aspects of playing a game, but also the intellectual capacity of a child's brain. A child's mind is like a sponge - soaking up huge amounts of information from their environment. They are absorbing everything around them, effortlessly, continuously, and indiscriminately.

ve combine the fun aspect with intellectual activities, such as math games, puzzles, and learning, children become more engaged and motivated to explore and understand

mathematical concepts. This approach helps make Math enjoyable and promotes a positive attitude toward learning. By integrating fun and intellectual aspects, children can develop their math skills while having a great time, enhancing their overall learning experience.

WORKING

Each time a player plays a move, they will be asked a basic math question. The player needs to answer the question correctly for their symbol to get assigned to the particular block they clicked on. The math questions include one of the four basic math operations: addition, subtraction, multiplication or division. The questions will be of Multiple Choce Question (MCQ) type with three options, of which only one would be the correct answer.

If they choose the correct answer, their symbol will get placed on the clicked block and the turn will switch to the opposite player. But if they choose an incorrect answer, their symbol will **not** get placed on the clicked block and the turn will **still** switch to the opposite player. This means that an incorrect answer will cost the player a turn.

The game continues until one of the following conditions is met:

a. One player gets three of their symbols in a row horizontally, vertically, or diagonally. This player wins the game.

b. All cells are filled, and no player has achieved three in a row. The game is then considered a draw.

CODE

Please have a look at the **GitHub Link** for the extended working version of the code pasted below:

In the 'lib' folder, the following dart files will be found:

InterstitialAdPage.dart

```
import 'package:flutter/material.dart';
import 'package:google_mobile_ads/google_mobile_ads.dart';

stitialAdPage extends StatefulWidget {
erstitialAdPage({Key? key}) : super(key: key);
```

```
  @override
  State<InterstitialAdPage> createState() => _InterstitialAdState();
}

class _InterstitialAdState extends State<InterstitialAdPage> {

  @override
  void initState(){
    super.initState();
    initInterstitialAd();
  }
  late InterstitialAd interstitialAd;
  bool isAdLoaded = false;

  initInterstitialAd(){
    InterstitialAd.load(
        adUnitId: 'ca-app-pub-3940256099942544/1033173712',//test
        request: const AdRequest(),
        adLoadCallback: InterstitialAdLoadCallback(
            onAdLoaded: (ad){
              interstitialAd = ad;
              setState(() {
                isAdLoaded = true;
                print("hehe");
              });
            },
            onAdFailedToLoad: ((error) {
              interstitialAd.dispose();
              print("oh");
            }
            ),
        ),
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
        r: AppBar(
        const Text("Interstitial Ad"),
```

```
      ),
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            if(isAdLoaded) {
              print("YES");
              interstitialAd.show();
             }
            else  print("noo");
          },
          child: const Text('Task Completed'),
        ),
      ),
    );
  }
}
```


dialogbox.dart

```
import 'package:flutter/material.dart';
import 'dart:math';


class MathDialog extends StatefulWidget {
  const MathDialog({Key? key}) : super(key: key);

  @override
  State<MathDialog> createState() => _MathDialogState();
}

class _MathDialogState extends State<MathDialog> {

  @override
  Widget build(BuildContext context) {

    var intValue1 = Random().nextInt(100);
    var intValue2 = 1 + Random().nextInt((10 + 1)-1);
    var randdiv = Random().nextInt(10);
          ivValue = 1 + Random().nextInt((10 + 1)-1);
          ıl = Random().nextInt(100);
```

```
var mcq2 = Random().nextInt(100);
int ans = 0;
num flag;
List<String> operators = ['+', '-', '*', '/'];

operators.shuffle();
String operator = (operators[0]);

if(operator == '+')
{
  ans = intValue1 + intValue2;
}
else if( operator == '-')
{
  ans = intValue1 - intValue2;
}
else if(operator == '*')
{
  ans = intValue1 * intValue2;
}
else if(operator == '/')
{
  intValue1 = randdiv * intValue2;
  ans = randdiv;
}

List<int> mcqs = [mcq1, mcq2, ans];

mcqs.shuffle();
var charmcqs0 = mcqs[0];
var charmcqs1 = mcqs[1];
var charmcqs2 = mcqs[2];

      return WillPopScope(

        onWillPop: () async => false,
        child: AlertDialog(

//title: Text('Welcome'),
```

```
// To display the title it is optional
    content: Text('$intValue1 $operator $intValue2'),
// Message which will be pop up on the screen
// Action widget which will provide the user to acknowledge the choice

actions: [
  FloatingActionButton(
    backgroundColor: Colors.green,
    foregroundColor: Colors.white,
    onPressed: () {
      flag = (charmcqs0 == ans) ? 1.0 : 0.0;
      Navigator.pop(context, flag);
    },
    child: Text('$charmcqs0'),
  ),
  FloatingActionButton(
    backgroundColor: Colors.green,
    foregroundColor: Colors.white,
    onPressed: () {
      flag = (charmcqs1 == ans) ? 1.0 : 0.0;
      Navigator.pop(context, flag);
    },
    child: Text('$charmcqs1'),
  ),
  FloatingActionButton(
    backgroundColor: Colors.green,
    foregroundColor: Colors.white,
    onPressed: () {
      flag = (charmcqs2 == ans) ? 1.0 : 0.0;
      Navigator.pop(context, flag);
    },
    child: Text('$charmcqs2'),

  ),
 ],
  )
 );
}
```

home.dart

```
import 'package:flutter/material.dart';
import 'package:google_mobile_ads/google_mobile_ads.dart';
import 'package:url_launcher/link.dart';
import 'dart:async';
import 'package:url_launcher/url_launcher.dart';


class Home extends StatefulWidget {
  const Home({super.key});

  State<Home> createState() => _HomeState();
}

class _HomeState extends State<Home> {

  int number = 1;


  Future<bool> showExitPopup() async {
    return await showDialog( //show confirm dialogue
      //the return value will be from "Yes" or "No" options
      context: context,
      builder: (context) => AlertDialog(
        title: Text('Exit App'),
        content: Text('Do you want to exit the game?'),
        actions:[
          ElevatedButton(
            onPressed: () => Navigator.of(context).pop(false),
            //return false when click on "NO"
            child:Text('No'),
          ),

          ElevatedButton(
            onPressed: () => Navigator.of(context).pop(true),
              eturn true when click on "Yes"
            ild:const Text('Yes'),
```

```
        ),

      ],
    ),
  )??false; //if showDialouge had returned null, then return false
}

@override
Widget build(BuildContext context) {


  return WillPopScope(
    onWillPop: showExitPopup,
    child: Stack(

      children: <Widget>[

       Align(
        child: Column(
         children: <Widget>[
          Expanded(
           child: Row(
            children: <Widget>[
             Expanded(
              child: Container(
               decoration: const BoxDecoration(
                image: DecorationImage(
                 image: AssetImage('assets/mathbg.jpg'),
                 fit: BoxFit.cover,
                ),
               ),
              ),
             ),
            ],
           ),
          ),
         ],
```

```
Align(

   alignment: Alignment.topLeft,
   child: Column(
      mainAxisAlignment: MainAxisAlignment.start,
      crossAxisAlignment: CrossAxisAlignment.center,
      children: <Widget>[

        Column(
      mainAxisAlignment: MainAxisAlignment.center,
      crossAxisAlignment: CrossAxisAlignment.center,
      children: <Widget>[

          /*ElevatedButton(
            onPressed: ()  {
              setState(() {
                number += 1;
              });
            },
            child: const Icon(Icons.add),
          ),*/
          DefaultTextStyle(
             style: const TextStyle(
               color: Colors.black87,
               fontSize: 50.0,
               fontFamily: 'EBGaramond',
             ),
            child: Container(
              padding: const EdgeInsets.all(0.0),
              margin: const EdgeInsets.fromLTRB(30.0, 150.0, 30.0, 180.0),
              child: const Text('MATH\n'
                  'TIC TAC TOE',
              textAlign: TextAlign.center,
              ),
            ),
          ),
        ]

         Row(
```

```
            mainAxisAlignment: MainAxisAlignment.center,
            crossAxisAlignment: CrossAxisAlignment.center,
            children: <Widget>[

              ElevatedButton(
                onPressed: () {
                  Navigator.pushNamed(context, '/location');
                },
                style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.black,
                    textStyle: const TextStyle(fontFamily: 'EBGaramond', fontSize: 40),
                    shadowColor: Colors.black87,
                elevation: 20),
                child: const Text('PLAY'),
              ),
            ]
        ),
        Row(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.end,
          children: <Widget>[
            Link(
              target: LinkTarget.blank,
              uri: Uri.parse('https://sites.google.com/view/mathtictactoe/home'),
              builder: (context, followLink) =>
                ElevatedButton.icon
            (
                onPressed:  followLink,           //Navigator.pushNamed(context, '/roo');
                icon: const Icon(
                    Icons.privacy_tip_outlined
                ),
                label: const Text(''),
                style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.white10),
              ),
            ),
            ]
        )
```

```
          )
        ]
    )
    );
  }
}
```

loading.dart

```
import 'package:appy/InterstitialAdPage.dart';
import 'package:flutter/material.dart';
import 'package:appy/win.dart';
import 'package:appy/dialogbox.dart';
import 'package:google_mobile_ads/google_mobile_ads.dart';

class Loading extends StatefulWidget {
  const Loading({Key? key}) : super(key: key);

  @override
  State<Loading> createState() => _LoadingState();
}

class _LoadingState extends State<Loading> {

  final BannerAd myBanner = BannerAd(
    adUnitId: 'ca-app-pub-5523026977500112/8151031145',
    size: AdSize.banner,
    request: const AdRequest(),
    listener: const BannerAdListener(),
  );


  @override
  void initState() {
    super.initState();
    myBanner.load();
  }
```



```
lue = 'Screen 1 Data';
rn = 'X';
```

```
String turntext = '';
bool keeptrack = true;
int track = 1;
late num flag;
int isPressedNum = 0;
late String button;
String button1 = '';
String button2 = '';
String button3 = '';
String button4 = '';
String button5 = '';
String button6 = '';
String button7 = '';
String button8 = '';
String button9 = '';
bool _isPressed1 = false;
bool _isPressed2 = false;
bool _isPressed3 = false;
bool _isPressed4 = false;
bool _isPressed5 = false;
bool _isPressed6 = false;
bool _isPressed7 = false;
bool _isPressed8 = false;
bool _isPressed9 = false;

int i = 0;
int j = 0;
List<String> operators = ['+', '-', '*', '/'];

_navigatetohome(turn)async{
  await Future.delayed(const Duration(milliseconds: 750), () {});
  Navigator.of(context).push( MaterialPageRoute(builder: (context) =>
  Win(value: turn),
 ));
}

int checkwin()
{
          on1 == button2 && button2 == button3 && (button3 == 'X' || button3 ==  'O'))
```

```
 return 1;
}
else if (button4 == button5 && button5 == button6 && (button6 == 'X' || button6 ==  'O'))
{
 return 1;
}
else if (button7 == button8 && button8 == button9 && (button9 == 'X' || button9 ==  'O'))
{
 return 1;
}
else if (button1 == button4 && button4 == button7 && (button7 == 'X' || button7 ==  'O'))
{
 return 1;
}
else if (button2 == button5 && button5 == button8 && (button8 == 'X' || button8 ==  'O'))
{
 return 1;
}
else if (button3 == button6 && button6 == button9 && (button9 == 'X' || button9 ==  'O'))
{
 return 1;
}
else if (button1 == button5 && button5 == button9 && (button9 == 'X' || button9 ==  'O'))
{
 return 1;
}
else if (button3 == button5 && button5 == button7 && (button7 == 'X' || button7 ==  'O'))
{
 return 1;
}
else if (track == 10)
{
 return 0;
}
else
{
 return  - 1;
}
```

```
Future<String> _myCallback(isPressedNum) async {

  flag = await showDialog(
    barrierDismissible: false,
    context: context,
    builder: (context) => const MathDialog(),
  );

  setState(() {
    button = '';

    if(flag == 1.0) {
      track += 1;
      print('Trackkkkkkk: $track');

      button = (keeptrack) ? 'X' : 'O';
      if(isPressedNum == 1)
      {
        _isPressed1 = true;
        button1 = button;
      }
      else if(isPressedNum == 2)
      {
        _isPressed2 = true;
        button2 = button;
      }
      else if(isPressedNum == 3)
      {
        _isPressed3 = true;
        button3 = button;
      }
      else if(isPressedNum == 4)
      {
        _isPressed4 = true;
        button4 = button;
      }
      else if(isPressedNum == 5)

        ;Pressed5 = true;
```

```
  button5 = button;
 }
 else if(isPressedNum == 6)
 {
  _isPressed6 = true;
  button6 = button;
 }
 else if(isPressedNum == 7)
 {
  _isPressed7 = true;
  button7 = button;
 }
 else if(isPressedNum == 8)
 {
  _isPressed8 = true;
  button8 = button;
 }
 else if(isPressedNum == 9)
 {
  _isPressed9 = true;
  button9 = button;
 }

 i = checkwin();


 if (i == 1) {
  turn = 'Winner is: $button!!!';

  _navigatetohome(turn);
 }
 else if (i == 0) {
  turn = 'DRAW...';

  _navigatetohome(turn);
 }

}
    rack = !keeptrack;
```

```
return button;
}

@override
Widget build(BuildContext context) {

  turn = (keeptrack) ? 'Turn: X' : 'Turn: O';

  return Stack(


        children: <Widget>[
          Align(
            child: Column(
              children: <Widget>[
                Expanded(
                  child: Row(
                    children: <Widget>[
                      Expanded(
                        child: Container(
                          decoration: const BoxDecoration(
                            image: DecorationImage(
                              image: AssetImage('assets/wallpaper.jpg'),
                              fit: BoxFit.cover,
                            ),
                          ),
                        ),
                      ),
                    ],
                  ),
                ),
              ],
            ),
          ),

          Align(

            lignment: Alignment.topLeft,
```

```
child: Column(
   mainAxisAlignment: MainAxisAlignment.start,
   crossAxisAlignment: CrossAxisAlignment.center,
   children: <Widget>[

        DefaultTextStyle(
          style: const TextStyle(
            color: Colors.lime,
            fontSize: 69.0,
            fontFamily: 'EBGaramond',
          ),
          child: Container(
            padding: const EdgeInsets.symmetric(horizontal: 1.0, vertical: 10.0),
            margin: const EdgeInsets.fromLTRB(98.0, 100.0, 1.0, 30.0),


            child: Text(turn),
          ),
        ),
   ]
  )
),

Align(

   alignment: Alignment.topLeft,
   child: Column(
      mainAxisAlignment: MainAxisAlignment.end,
      crossAxisAlignment: CrossAxisAlignment.center,
      children: <Widget>[



       Row(
         mainAxisAlignment: MainAxisAlignment.center,
         children:  <Widget>[

          SizedBox(
            height:100, //height of button
            width:100,
```

```
                child: ElevatedButton(


                  onPressed: () async => _isPressed1 == false ? button1=await _myCallback(1):
null,

                 style: ElevatedButton.styleFrom(
                                backgroundColor: Colors.redAccent, disabledForegroundColor:
Colors.redAccent, disabledBackgroundColor: Colors.redAccent, //background color of button
                    side: const BorderSide(width:3, color:Colors.black87), //border width and
color
                    elevation: 5, //elevation of button
                ),
```



```
                  child: DefaultTextStyle(
                    style: const TextStyle(
                      color: Colors.white,
                      fontSize: 40.0,
                      fontFamily: 'EBGaramond',
                    ),
                    child: Text(button1),

                  ),


                ),

              ),
              SizedBox(
                height:100, //height of button
                width:100,

                child: ElevatedButton(
                  onPressed: () async => _isPressed2 == false ? button2=await _myCallback(2):
null,

                  style: ElevatedButton.styleFrom(
```

```
                                backgroundColor: Colors.redAccent, disabledForegroundColor:
Colors.redAccent, disabledBackgroundColor: Colors.redAccent, //background color of button
                                    side: const BorderSide(width:3, color:Colors.black87), //border width and
color
                                  elevation: 5, //elevation of button
                                ),

                                child: DefaultTextStyle(
                                  style: const TextStyle(
                                    color: Colors.white,
                                    fontSize: 40.0,
                                    fontFamily: 'EBGaramond',
                                ),
                                child: Text(button2),
                                ),
                              ),

                            ),
                            SizedBox(
                              height:100, //height of button
                              width:100,

                              child: ElevatedButton(
                                 onPressed: () async => _isPressed3 == false ? button3=await _myCallback(3):
null,

                                  style: ElevatedButton.styleFrom(
                                                backgroundColor: Colors.redAccent, disabledForegroundColor:
Colors.redAccent, disabledBackgroundColor: Colors.redAccent, //background color of button
                                    side: const BorderSide(width:3, color:Colors.black87), //border width and
color
                                  elevation: 5, //elevation of button
                                ),

                                child: DefaultTextStyle(
                                  style: const TextStyle(
                                    color: Colors.white,
                                    fontSize: 40.0,
                                    fontFamily: 'EBGaramond',
```

```
                ),
                child:  Text(button3),

              ),
            ),

          ),
        ],
      ),

      Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children:  <Widget>[

          SizedBox(
          height:100, //height of button
          width:100,

          child: ElevatedButton(
             onPressed: () async => _isPressed4 == false ? button4=await _myCallback(4):
null,

           style: ElevatedButton.styleFrom(
                          backgroundColor: Colors.redAccent, disabledForegroundColor:
Colors.redAccent, disabledBackgroundColor: Colors.redAccent, //background color of button
                 side: const BorderSide(width:3, color:Colors.black87), //border width and
color
            elevation: 5, //elevation of button
           ),

           child: DefaultTextStyle(
            style: const TextStyle(
              color: Colors.white,
              fontSize: 40.0,
              fontFamily: 'EBGaramond',
            ),
            child:  Text(button4),

           ),
          ),
```

```
                ),
                SizedBox(
                  height:100, //height of button
                  width:100,

                  child: ElevatedButton(
                     onPressed: () async => _isPressed5 == false ? button5=await _myCallback(5):
null,

                    style: ElevatedButton.styleFrom(
                                backgroundColor: Colors.redAccent, disabledForegroundColor:
Colors.redAccent, disabledBackgroundColor: Colors.redAccent, //background color of button
                          side: const BorderSide(width:3, color:Colors.black87), //border width and
color

                    elevation: 5, //elevation of button
                    ),

                    child: DefaultTextStyle(
                      style: const TextStyle(
                        color: Colors.white,
                        fontSize: 40.0,
                        fontFamily: 'EBGaramond',
                      ),
                      child:  Text(button5),

                    ),
                  ),

                ),
                SizedBox(
                  height:100, //height of button
                  width:100,

                  child: ElevatedButton(
                     onPressed: () async => _isPressed6 == false ? button6=await _myCallback(6):
null,

                    style: ElevatedButton.styleFrom(
```

```
                                backgroundColor: Colors.redAccent, disabledForegroundColor:
Colors.redAccent, disabledBackgroundColor: Colors.redAccent, //background color of button
                                    side: const BorderSide(width:3, color:Colors.black87), //border width and
color
                                  elevation: 5, //elevation of button
                                ),

                                child: DefaultTextStyle(
                                  style: const TextStyle(
                                    color: Colors.white,
                                    fontSize: 40.0,
                                    fontFamily: 'EBGaramond',
                                ),
                                child: Text(button6),

                                ),
                              ),

                            ),

                           ],
                          ),
                          Row(
                           mainAxisAlignment: MainAxisAlignment.center,
                           children:  <Widget>[

                            SizedBox(
                              height:100, //height of button
                              width:100,

                              child: ElevatedButton(
                                  onPressed: () async => _isPressed7 == false ? button7=await _myCallback(7):
null,

                                style: ElevatedButton.styleFrom(
                                                backgroundColor: Colors.redAccent, disabledForegroundColor:
Colors.redAccent, disabledBackgroundColor: Colors.redAccent, //background color of button
                                    side: const BorderSide(width:3, color:Colors.black87), //border width and

                                  elevation: 5, //elevation of button
```

```
                ),

                child: DefaultTextStyle(
                  style: const TextStyle(
                    color: Colors.white,
                    fontSize: 40.0,
                    fontFamily: 'EBGaramond',
                  ),
                  child:  Text(button7),

                ),
              ),
            ),
            SizedBox(
              height:100, //height of button
              width:100,

              child: ElevatedButton(
                 onPressed: () async => _isPressed8 == false ? button8=await _myCallback(8): null,

                 style: ElevatedButton.styleFrom(
                              backgroundColor: Colors.redAccent, disabledForegroundColor: Colors.redAccent, disabledBackgroundColor: Colors.redAccent, //background color of button
                        side: const BorderSide(width:3, color:Colors.black87), //border width and color
                   elevation: 5, //elevation of button
                 ),


                 child: DefaultTextStyle(
                          style: const TextStyle(
                          color: Colors.white,
                          fontSize: 40.0,
                          fontFamily: 'EBGaramond',
                          ),
                          child: Text(button8),

                   ),
```

```
                ),

              ),
              SizedBox(
                height:100, //height of button
                width:100,

                child: ElevatedButton(
                  onPressed: () async => _isPressed9 == false ? button9=await _myCallback(9):
null,

                  style: ElevatedButton.styleFrom(
                                backgroundColor: Colors.redAccent, disabledForegroundColor:
Colors.redAccent, disabledBackgroundColor: Colors.redAccent, //background color of button
                    side: const BorderSide(width:3, color:Colors.black87), //border width and
color

                    elevation: 5, //elevation of button
                  ),

                  child: DefaultTextStyle(
                        style: const TextStyle(
                        color: Colors.white,
                        fontSize: 40.0,
                        fontFamily: 'EBGaramond',
                        ),
                        child:  Text(button9),

                        ),
                ),
              ),
            ],
          ),


          Container(
            padding: const EdgeInsets.fromLTRB(1.0, 170.0, 1.0, 0.0),

            child: SizedBox(
              height: 70.0,
              width: 540.0,
```

```
                    child: AdWidget(ad: myBanner, )
                )
                )
            ]
            )
          ),
        ]
    );
  }
}
```


main.dart

```
import 'package:appy/InterstitialAdPage.dart';
import 'package:appy/splash.dart';
import 'package:flutter/material.dart';
import 'package:appy/loading.dart';
import 'package:appy/win.dart';
import 'package:appy/home.dart';

void main() => runApp(MaterialApp(
  initialRoute: '/splash',
  routes: {
    '/splash': (context) => const Splash(),
    '/home': (context) => const Home(),
    '/location': (context) => const Loading(),
    '/win': (context) => const Win(value: ''),
    '/iap': (context) => const InterstitialAdPage(),
  },
   ),
);
```

splash.dart

```
import 'package:animated_splash_screen/animated_splash_screen.dart';
      ckage:flutter/material.dart';
      ckage:appy/home.dart';
```

```
class Splash extends StatefulWidget {
  const Splash({Key? key}) : super(key: key);

  @override
  State<Splash> createState() => _SplashState();
}

class _SplashState extends State<Splash> {
  @override
  void initState() {
    super.initState();
    _navigatetohome();
  }

  _navigatetohome()async{
    await Future.delayed(const Duration(milliseconds: 2500), () {});
    Navigator.pushReplacement(context, MaterialPageRoute(builder: (context)=>const Home()));
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: AnimatedSplashScreen(
        splash: Icons.home,
        backgroundColor: Colors.blueGrey,
        nextScreen: const Home(),
        splashIconSize: 150.0,
        duration: 5000,
        splashTransition: SplashTransition.scaleTransition,
        curve: Curves.easeInBack,
      ),
    );
  }
}
```


win.dart

```
ckage:flutter/material.dart';
ckage:appy/loading.dart';
```

```
import 'dart:io';
import 'package:google_mobile_ads/google_mobile_ads.dart';
import 'package:appy/InterstitialAdPage.dart';

class Win extends StatefulWidget {
  final String value;

  const Win({
    Key? key,
    required this.value,
  }) : super(key: key);

  @override
  State<Win> createState() => _WinState();
}

class _WinState extends State<Win> {

  @override
  void initState(){
    super.initState();
    initInterstitialAd();
  }
  late InterstitialAd interstitialAd;
  bool isAdLoaded = false;

  initInterstitialAd(){
    InterstitialAd.load(
      adUnitId: 'ca-app-pub-5523026977500112/8244182877',//real
      //adUnitId: 'ca-app-pub-3940256099942544/1033173712',//test
      request: const AdRequest(),
      adLoadCallback: InterstitialAdLoadCallback(
        onAdLoaded: (ad){
          interstitialAd = ad;
          setState(() {
            isAdLoaded = true;
            print("hehe");
          });
```



```
        lFailedToLoad: ((error) {
```

```
        interstitialAd.dispose();
        print("oh");
       }
       ),
     ),
   );
}


Future<bool> showExitPopup() async {
  int count = 0;

    Navigator.popUntil(context, (route) {
      return count++ == 2;
    });
  Navigator.pushNamed(context, '/location').then((_) {
    setState(() {
      // Call setState to refresh the page.
    });
  });
  return Future.value(true);
}

@override
Widget build(BuildContext context) {
  return WillPopScope(

      onWillPop: showExitPopup,
      child: Stack(


      children: <Widget>[


        Align(
          child: Column(
            children: <Widget>[
              Expanded(
                child: Row(
                  children: <Widget>[
```

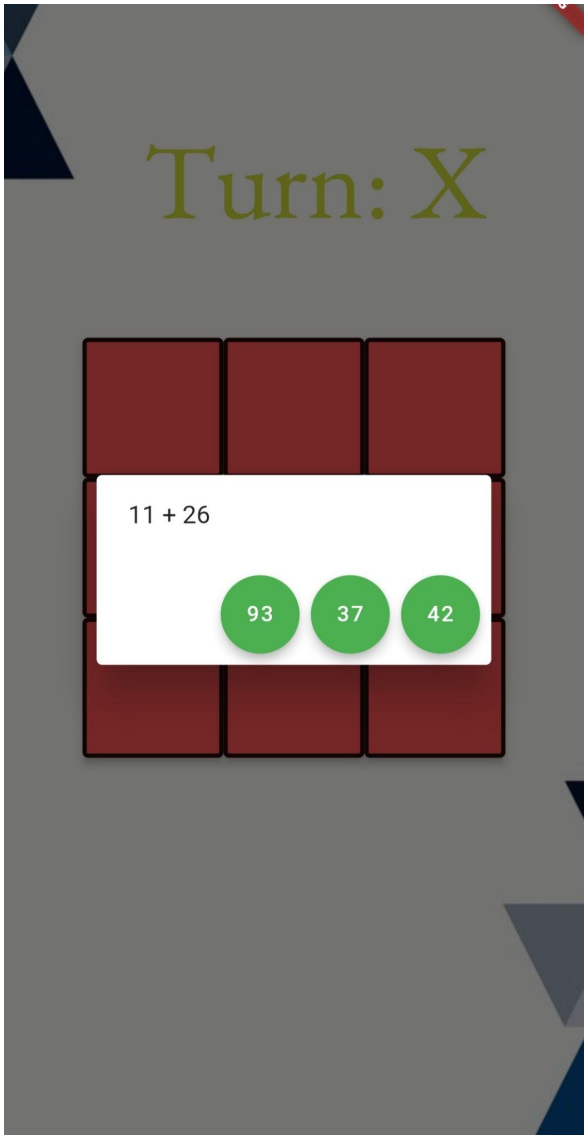
```
                Expanded(
                  child: Container(
                    decoration: const BoxDecoration(
                      image: DecorationImage(
                        image: AssetImage('assets/wallpaper.jpg'),
                        fit: BoxFit.cover,
                      ),
                    ),
                  ),
                ),
              ],
            ),
          ],
        ),
      ),
```

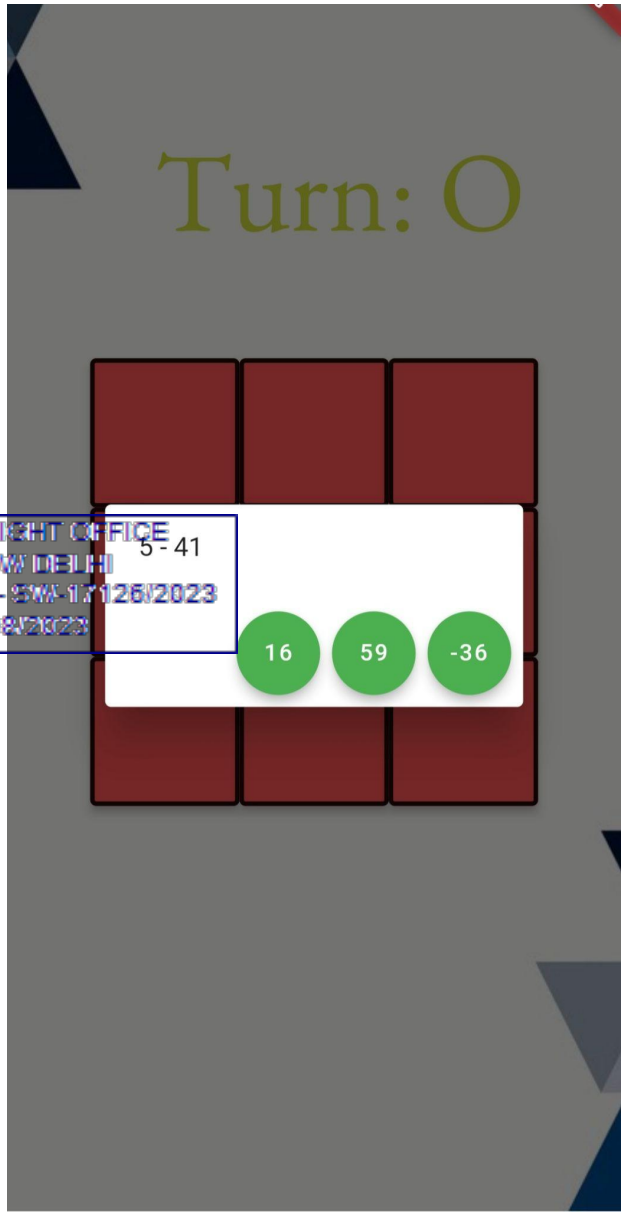


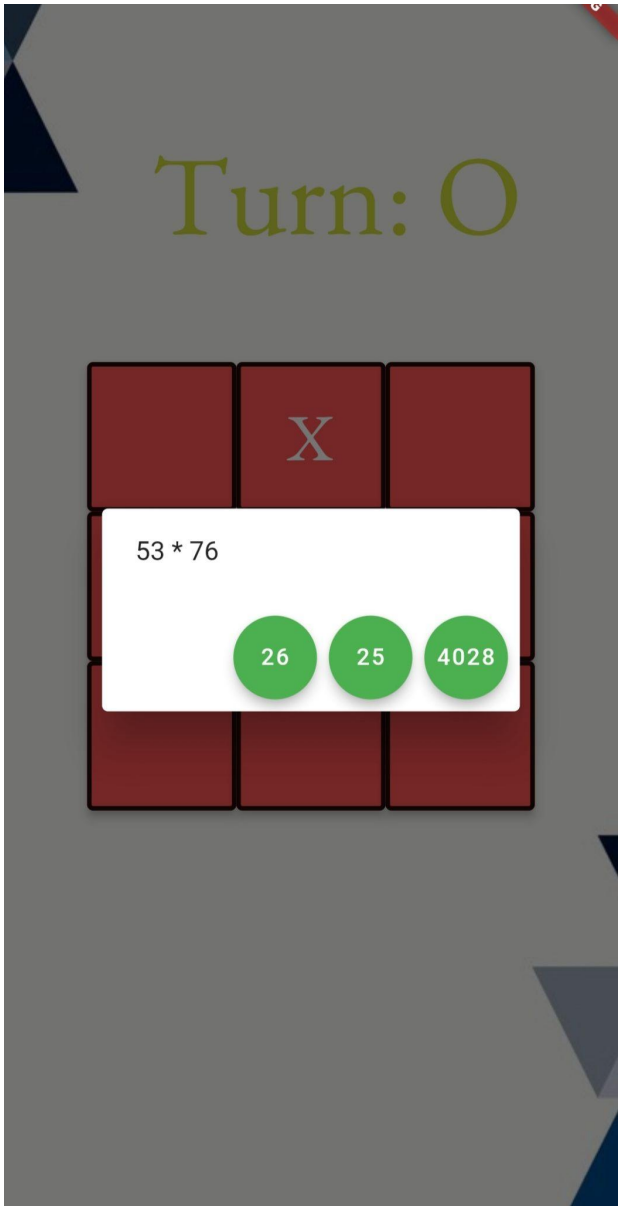
```
      Align(

        alignment: Alignment.topLeft,
        child: Column(
          mainAxisAlignment: MainAxisAlignment.start,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: <Widget>[
            Row(
              children: <Widget>[
                DefaultTextStyle(
                  style: const TextStyle(
                    color: Colors.lime,
                    fontSize: 39.0,
                    fontFamily: 'DancingScript',
                  ),
                  child: Container(
                    padding: const EdgeInsets.symmetric(
                        horizontal: 1.0, vertical: 10.0),
                    margin: const EdgeInsets.fromLTRB(
                        98.0, 100.0, 1.0, 30.0),

                    child: Text(widget.value),
                  ),
```

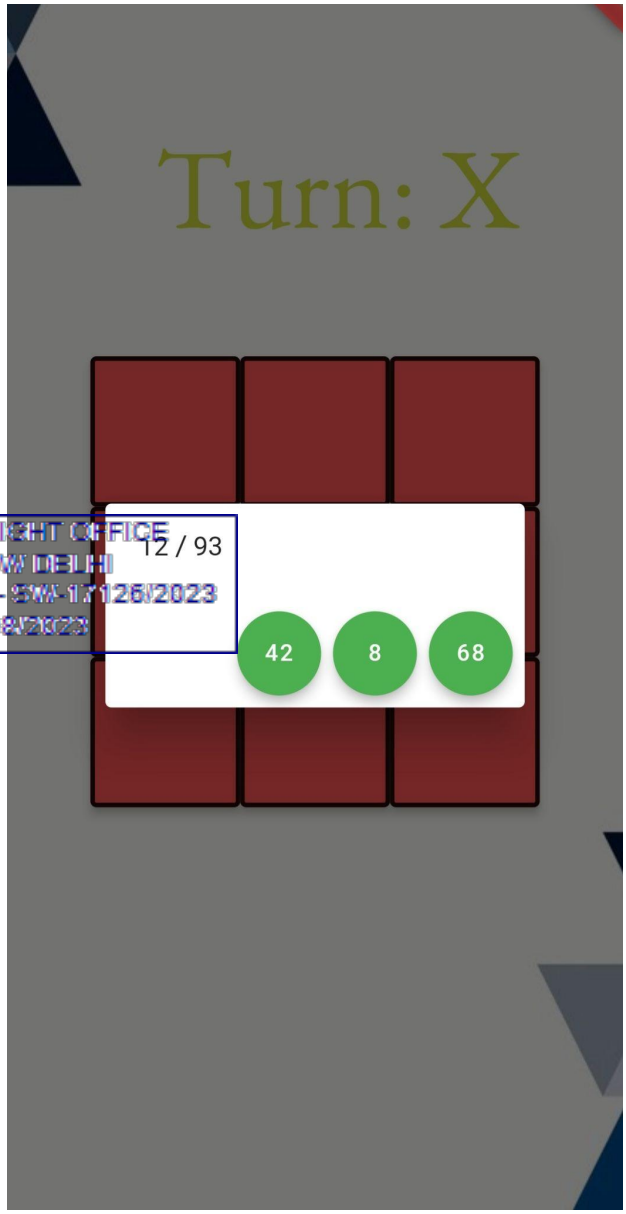
```
            ),
          ]
        ),
        ElevatedButton.icon(
          onPressed: () {
            if(isAdLoaded) {
              print("YES");
              interstitialAd.show();
            }
            else  print("noo");
            //Navigator.pushNamed(context, '/roo');
          int count = 0;
          Navigator.popUntil(context, (route) {
          return count++ == 2;
          });
          Navigator.pushNamed(context, '/location').then((_) {
              // This block runs when you have returned back to the 1st Page from 2nd.
          setState(() {
                // Call setState to refresh the page.
          });
        });
            },

            icon: const Icon(
              Icons.add_circle_outline
            ),
            label: const Text('Play new game'),
            style: ElevatedButton.styleFrom(
              backgroundColor: Colors.amber),
        ),

      ]

    )
  ),
 ]
 )
);
```

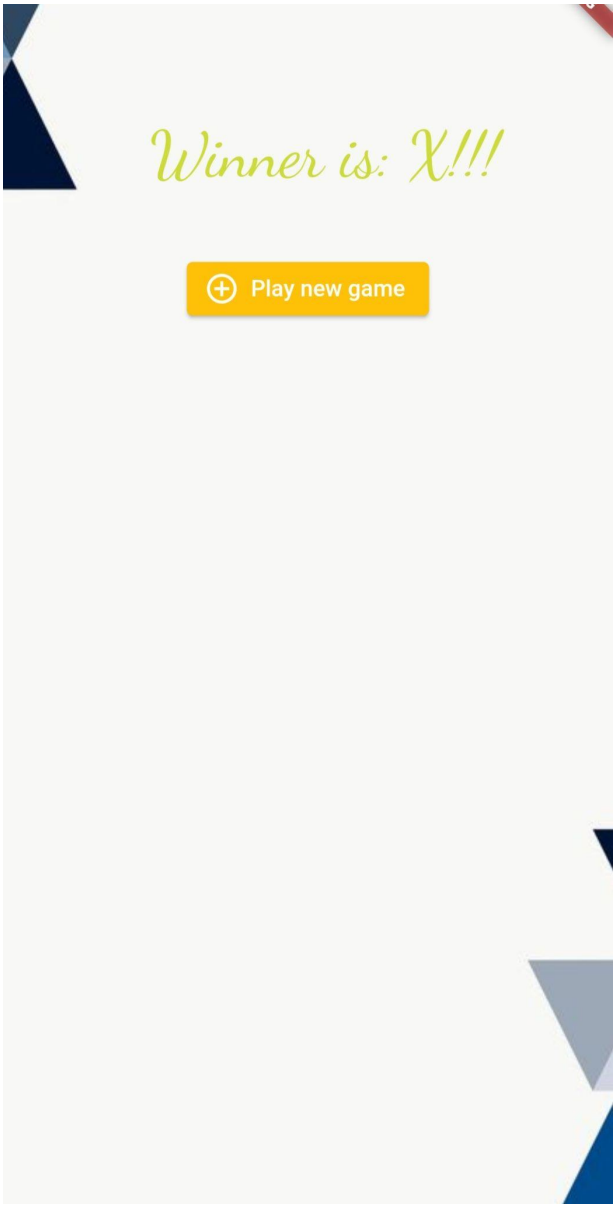
SCREENSHOTS OF THE APPLICATION











CONCLUSION

Math Tic Tac Toe is an application published and released on Play Store on 1st February 2023, with a subsequent update on 13th February, 2023. It is a fun tic tac toe game with basic math questions that aims to help improve children's creative as well as logical aspects equally.

Author:
Esha Ajit Chaugule