IMPLEMENTATION CODE

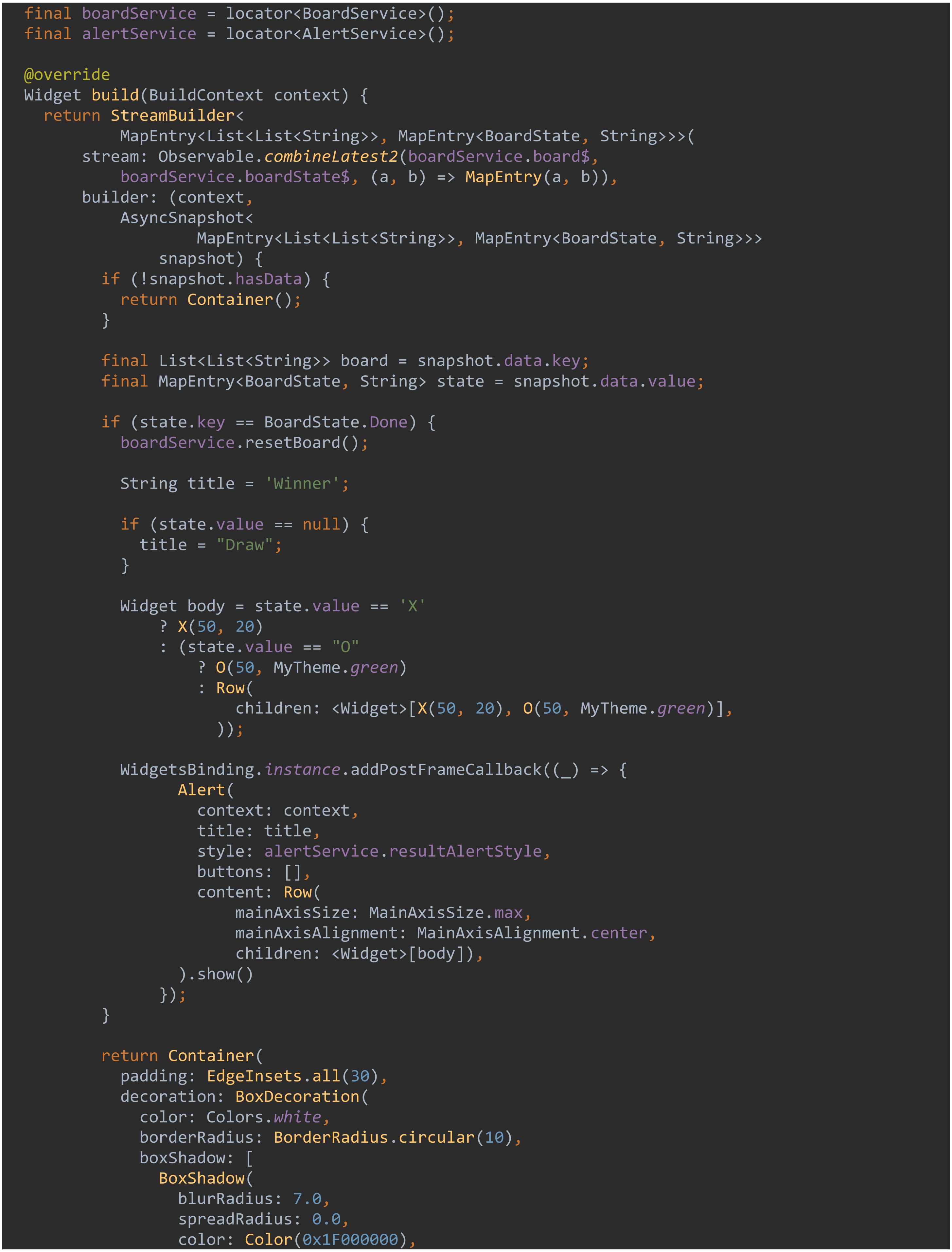
Main.dart

```
import 'package:flutter/material.dart';
import 'package:tic_tac/pages/start.dart';
import 'package:tic_tac/services/provider.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(fontFamily: 'Poppins'),
      debugShowCheckedModeBanner: false,
      home: StartPage(),
    );
  }
}
```

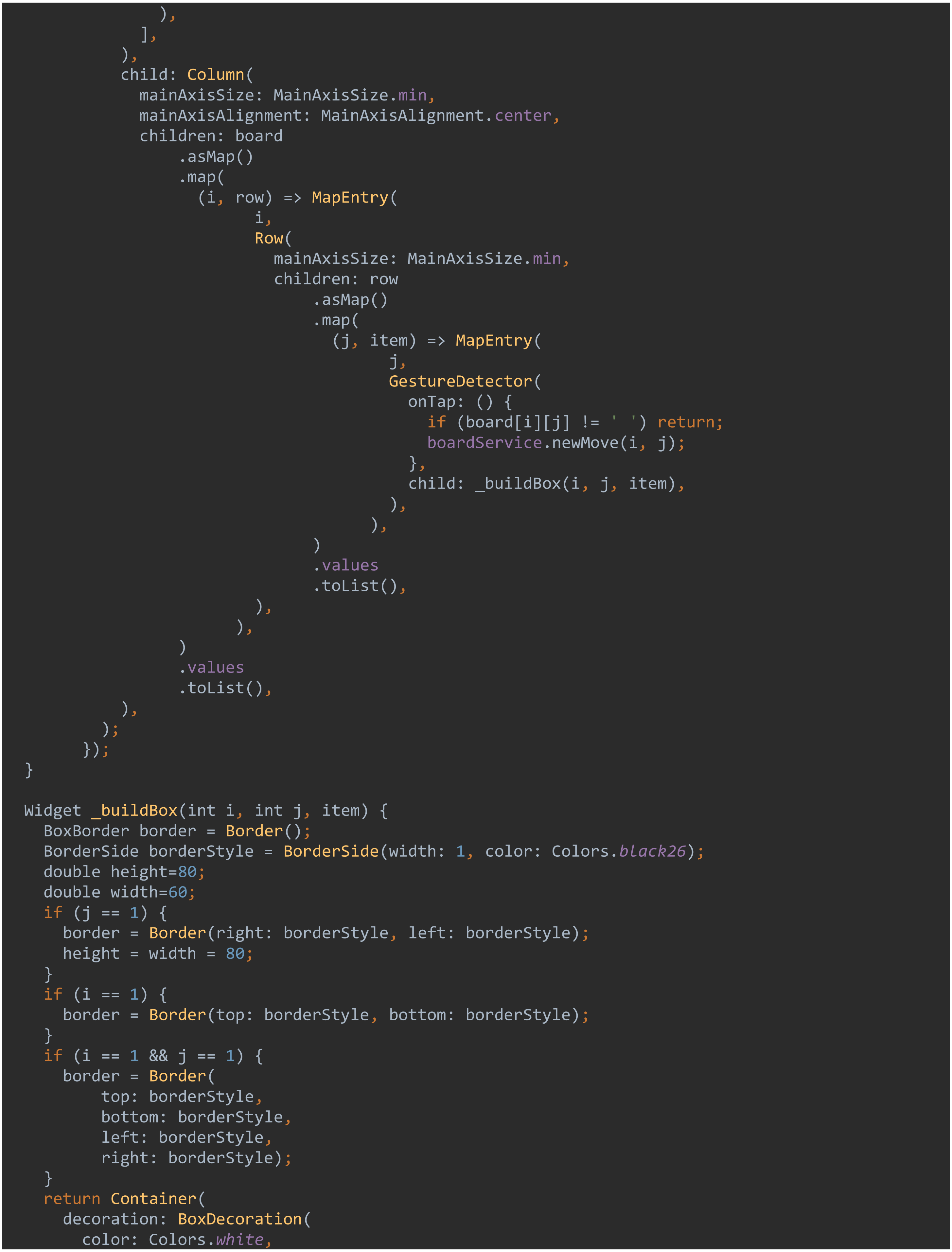
Components :

Board.dart

```
import 'package:flutter/material.dart';
import 'package:flutter/widgets.dart';
import 'package:rxdart/rxdart.dart';
import 'package:tic_tac/components/x.dart';
import 'package:tic_tac/services/alert.dart';
import 'package:tic_tac/services/board.dart';
import 'package:tic_tac/services/provider.dart';
import 'package:tic_tac/theme/theme.dart';
import 'package:rflutter_alert/rflutter_alert.dart';

import 'o.dart';

class Board extends StatefulWidget {
  Board({Key key}) : super(key: key);

  _BoardState createState() => _BoardState();
}

class _BoardState extends State<Board> {
```



```
  final boardService = locator<BoardService>();
  final alertService = locator<AlertService>();

  @override
  Widget build(BuildContext context) {
    return StreamBuilder<
            MapEntry<List<List<String>>, MapEntry<BoardState, String>>>(
        stream: Observable.combineLatest2(boardService.board$,
            boardService.boardState$, (a, b) => MapEntry(a, b)),
        builder: (context,
            AsyncSnapshot<
                    MapEntry<List<List<String>>, MapEntry<BoardState, String>>>
                snapshot) {
          if (!snapshot.hasData) {
            return Container();
          }

          final List<List<String>> board = snapshot.data.key;
          final MapEntry<BoardState, String> state = snapshot.data.value;

          if (state.key == BoardState.Done) {
            boardService.resetBoard();

            String title = 'Winner';

            if (state.value == null) {
              title = "Draw";
            }

            Widget body = state.value == 'X'
                ? X(50, 20)
                : (state.value == "O"
                    ? O(50, MyTheme.green)
                    : Row(
                        children: <Widget>[X(50, 20), O(50, MyTheme.green)],
                      ));

            WidgetsBinding.instance.addPostFrameCallback((_) => {
                  Alert(
                    context: context,
                    title: title,
                    style: alertService.resultAlertStyle,
                    buttons: [],
                    content: Row(
                        mainAxisSize: MainAxisSize.max,
                        mainAxisAlignment: MainAxisAlignment.center,
                        children: <Widget>[body]),
                  ).show()
                });
          }

          return Container(
            padding: EdgeInsets.all(30),
            decoration: BoxDecoration(
              color: Colors.white,
              borderRadius: BorderRadius.circular(10),
              boxShadow: [
                BoxShadow(
                  blurRadius: 7.0,
                  spreadRadius: 0.0,
                  color: Color(0x1F000000),
```



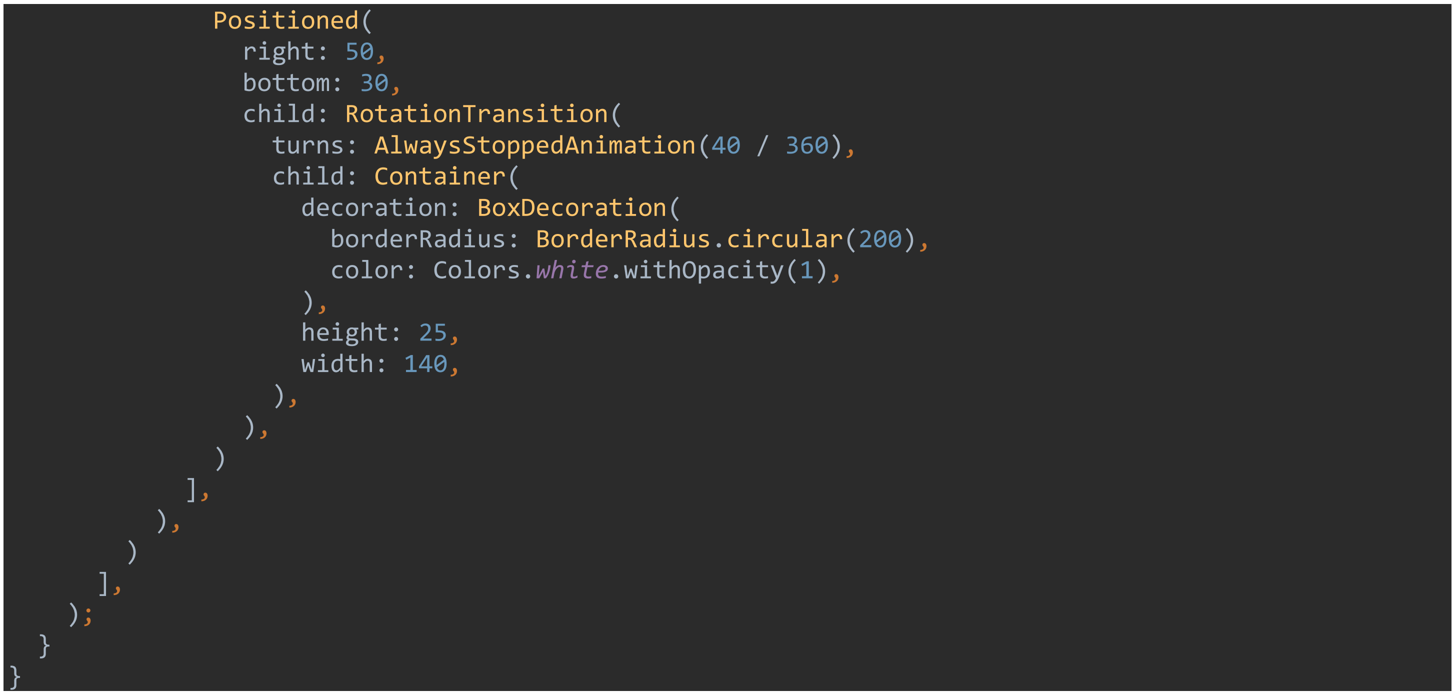
```
              ),
            ],
          ),
          child: Column(
            mainAxisSize: MainAxisSize.min,
            mainAxisAlignment: MainAxisAlignment.center,
            children: board
                .asMap()
                .map(
                  (i, row) => MapEntry(
                        i,
                        Row(
                          mainAxisSize: MainAxisSize.min,
                          children: row
                              .asMap()
                              .map(
                                (j, item) => MapEntry(
                                      j,
                                      GestureDetector(
                                        onTap: () {
                                          if (board[i][j] != ' ') return;
                                          boardService.newMove(i, j);
                                        },
                                        child: _buildBox(i, j, item),
                                      ),
                                    ),
                              )
                              .values
                              .toList(),
                        ),
                      ),
                )
                .values
                .toList(),
          ),
        );
      });
}

Widget _buildBox(int i, int j, item) {
  BoxBorder border = Border();
  BorderSide borderStyle = BorderSide(width: 1, color: Colors.black26);
  double height=80;
  double width=60;
  if (j == 1) {
    border = Border(right: borderStyle, left: borderStyle);
    height = width = 80;
  }
  if (i == 1) {
    border = Border(top: borderStyle, bottom: borderStyle);
  }
  if (i == 1 && j == 1) {
    border = Border(
        top: borderStyle,
        bottom: borderStyle,
        left: borderStyle,
        right: borderStyle);
  }
  return Container(
    decoration: BoxDecoration(
      color: Colors.white,
        border: border,
      ),
      height: height,
      width: width,
      child: Center(
        child:
            item == ' ' ? null : item == 'X' ? X(50, 13) : O(50, MyTheme.green),
      ),
    );
  }
}
```

Btn.dart

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

class Btn extends StatelessWidget {
  final List<Color> gradient;
  final Color color;
  final double height;
  final double width;
  final GestureTapCallback onTap;
  final double borderRadius;
  final Widget child;

  Btn(
      {Key key,
      this.gradient,
      this.color,
      this.onTap,
      this.child,
      this.borderRadius = 0,
      this.height,
      this.width})
      : super(key: key);

  @override
  Widget build(BuildContext context) {
    return GestureDetector(
      onTap: onTap,
      child: Container(
        height: height,
        width: width,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(borderRadius),
          color: color,
          gradient: this.gradient == null
              ? null
              : LinearGradient(
                  begin: Alignment.centerLeft,
                  end: Alignment.centerRight,
                  stops: [0.1, 0.8],
                  colors: gradient),
          boxShadow: [
            BoxShadow(
                color: Colors.black.withOpacity(.1),
                spreadRadius: 5,
                blurRadius: 10)
          ],
        ),
        child: Center(
          child: child,
        ),
      ),
    );
  }
}
```

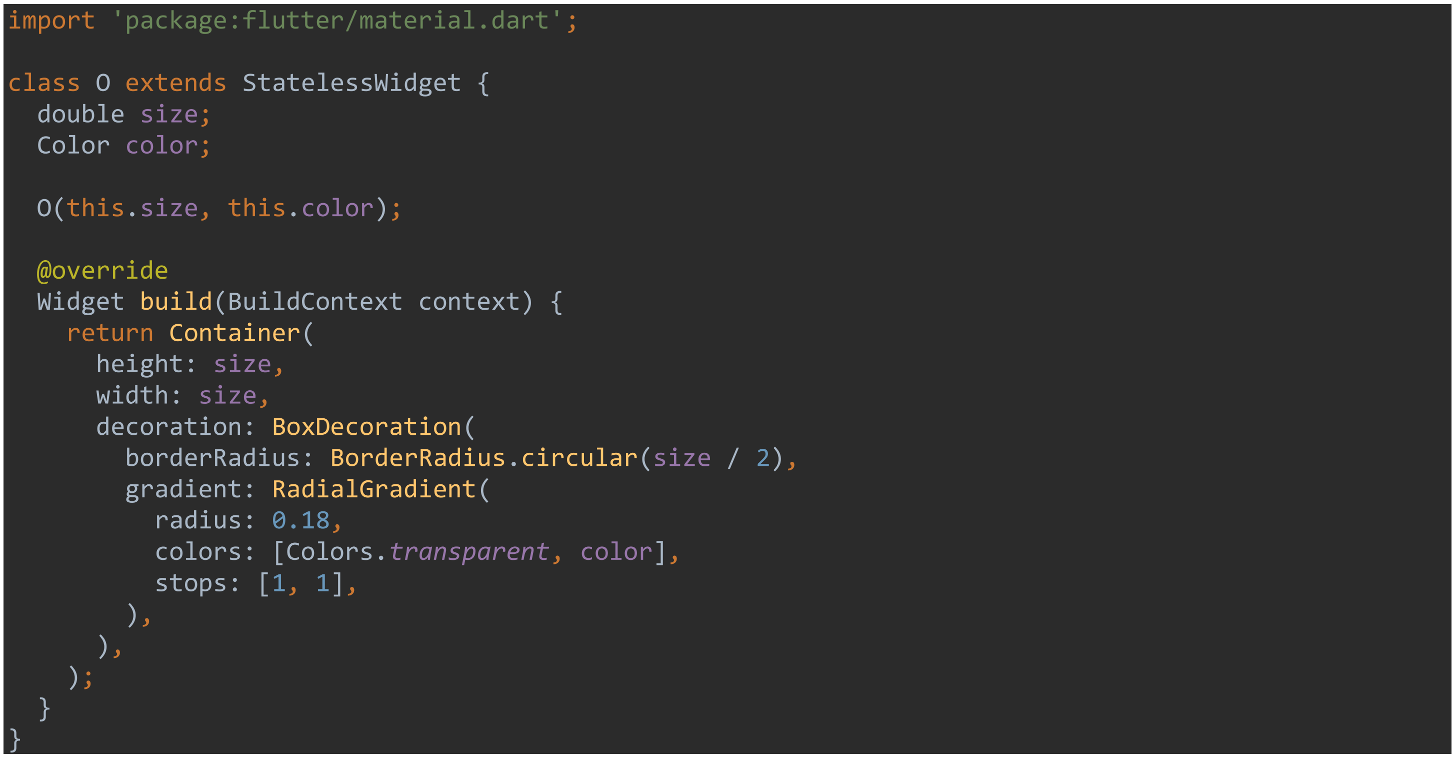
Logo.dart

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

class Logo extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Stack(
      children: <Widget>[
        Container(
          height: 150,
          width: 200,
          child: Stack(
            children: <Widget>[
              Positioned(
                right: 10,
                top: 70,
                child: Container(
                  height: 65,
                  width: 65,
                  decoration: BoxDecoration(
                    borderRadius: BorderRadius.circular(65 / 2),
                    gradient: RadialGradient(
                      radius: 0.18,
                      colors: [
                        Colors.transparent,
                        Colors.white.withOpacity(.35)
                      ],
                      stops: [1, 1],
                    ),
                  ),
                ),
              ),
              Positioned(
                left: 0,
                bottom: 50,
                child: RotationTransition(
                  turns: AlwaysStoppedAnimation(-50 / 360),
                  child: Container(
                    decoration: BoxDecoration(
                      borderRadius: BorderRadius.circular(200),
                      color: Colors.white.withOpacity(1),
                    ),
                    height: 25,
                    width: 200,
                  ),
                ),
              ),
```



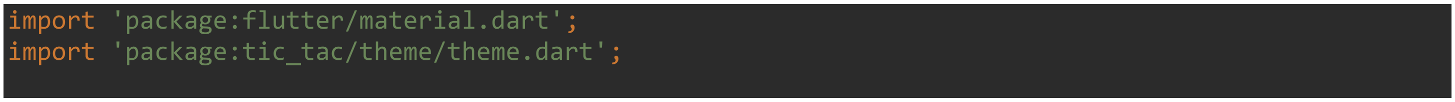
```dart
              Positioned(
                right: 50,
                bottom: 30,
                child: RotationTransition(
                  turns: AlwaysStoppedAnimation(40 / 360),
                  child: Container(
                    decoration: BoxDecoration(
                      borderRadius: BorderRadius.circular(200),
                      color: Colors.white.withOpacity(1),
                    ),
                    height: 25,
                    width: 140,
                  ),
                ),
              )
            ],
          ),
        )
      ],
    );
  }
}
```

O.dart

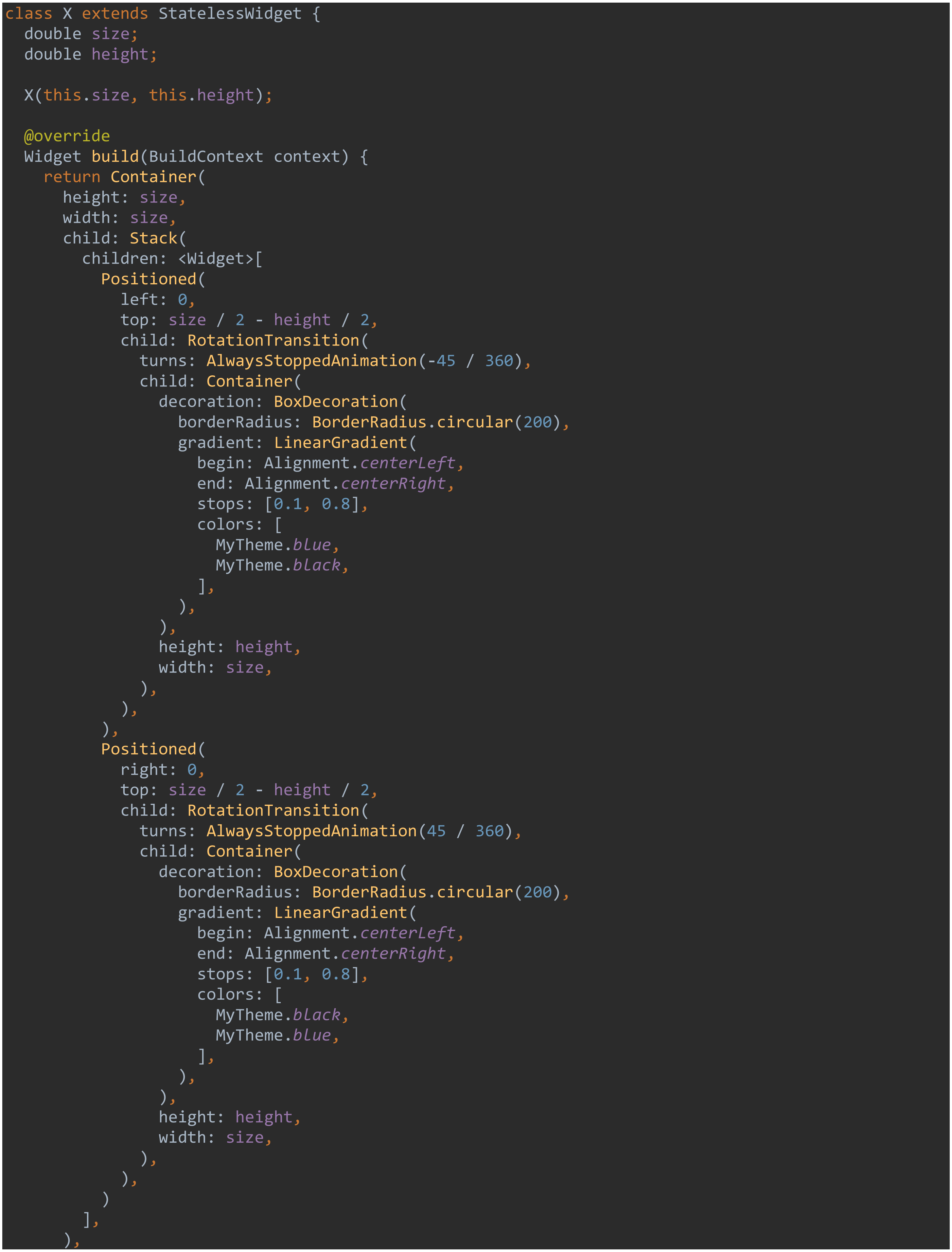


```dart
import 'package:flutter/material.dart';

class O extends StatelessWidget {
  double size;
  Color color;

  O(this.size, this.color);

  @override
  Widget build(BuildContext context) {
    return Container(
      height: size,
      width: size,
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(size / 2),
        gradient: RadialGradient(
          radius: 0.18,
          colors: [Colors.transparent, color],
          stops: [1, 1],
        ),
      ),
    );
  }
}
```

X.dart



```dart
import 'package:flutter/material.dart';
import 'package:tic_tac/theme/theme.dart';
```

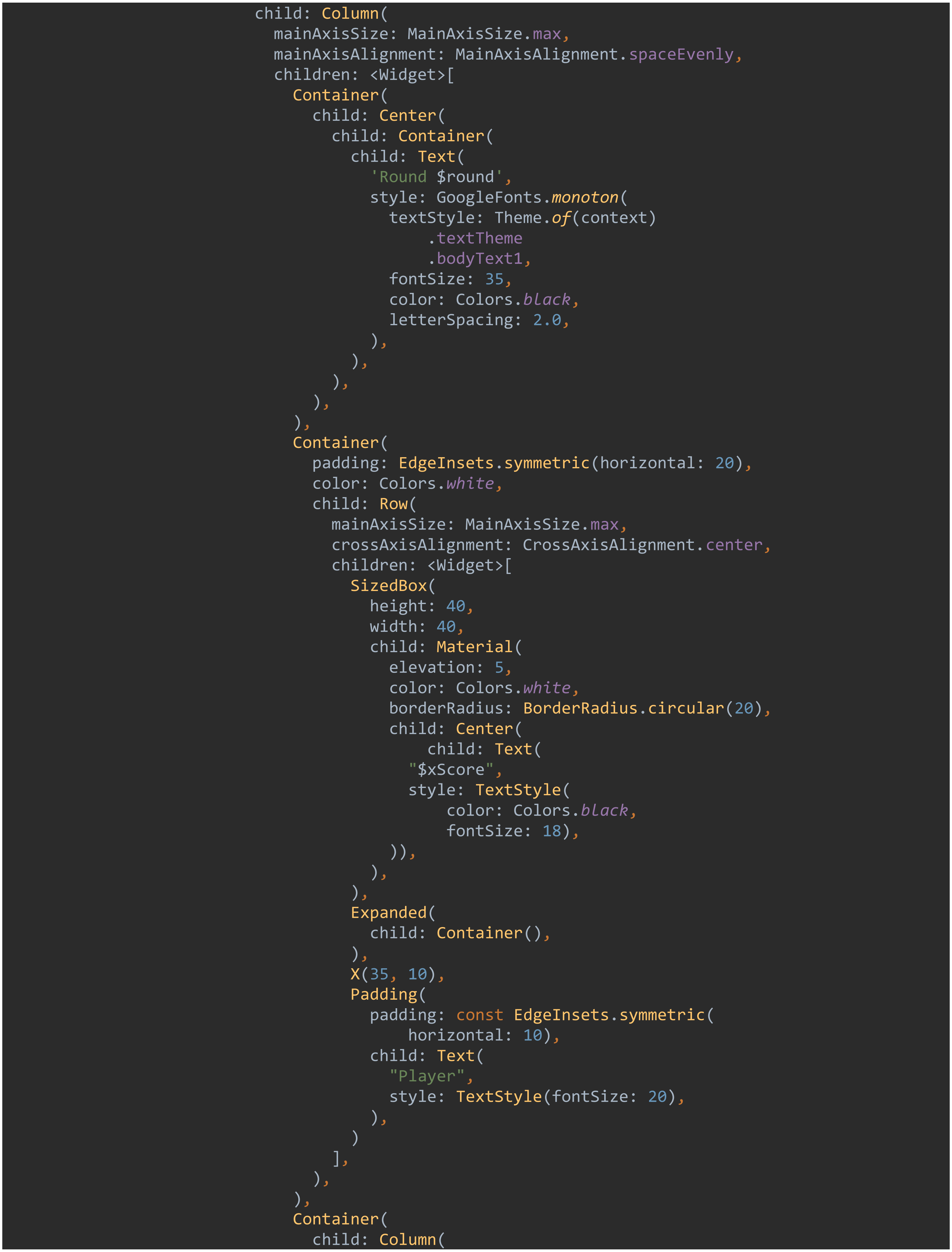


```
class X extends StatelessWidget {
  double size;
  double height;

  X(this.size, this.height);

  @override
  Widget build(BuildContext context) {
    return Container(
      height: size,
      width: size,
      child: Stack(
        children: <Widget>[
          Positioned(
            left: 0,
            top: size / 2 - height / 2,
            child: RotationTransition(
              turns: AlwaysStoppedAnimation(-45 / 360),
              child: Container(
                decoration: BoxDecoration(
                  borderRadius: BorderRadius.circular(200),
                  gradient: LinearGradient(
                    begin: Alignment.centerLeft,
                    end: Alignment.centerRight,
                    stops: [0.1, 0.8],
                    colors: [
                      MyTheme.blue,
                      MyTheme.black,
                    ],
                  ),
                ),
                height: height,
                width: size,
              ),
            ),
          ),
          Positioned(
            right: 0,
            top: size / 2 - height / 2,
            child: RotationTransition(
              turns: AlwaysStoppedAnimation(45 / 360),
              child: Container(
                decoration: BoxDecoration(
                  borderRadius: BorderRadius.circular(200),
                  gradient: LinearGradient(
                    begin: Alignment.centerLeft,
                    end: Alignment.centerRight,
                    stops: [0.1, 0.8],
                    colors: [
                      MyTheme.black,
                      MyTheme.blue,
                    ],
                  ),
                ),
                height: height,
                width: size,
              ),
            ),
          )
        ],
      ),
    );
  }
}
```

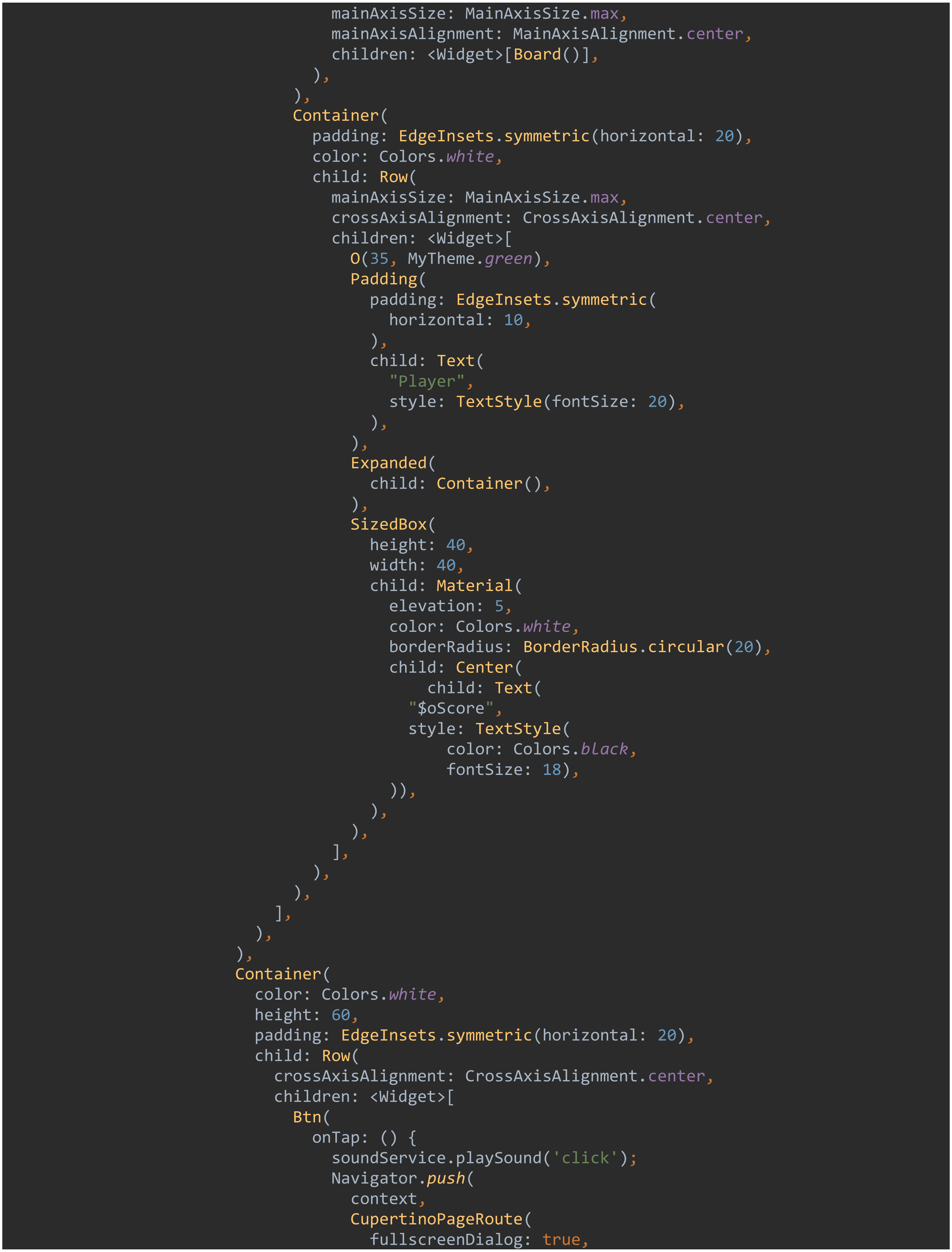
Pages:

Game.dart

```
import 'dart:async';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutter/widgets.dart';
import 'package:tic_tac/components/board.dart';
import 'package:tic_tac/components/o.dart';
import 'package:tic_tac/components/x.dart';
import 'package:tic_tac/services/board.dart';
import 'package:tic_tac/services/provider.dart';
import 'package:tic_tac/theme/theme.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:tic_tac/components/btn.dart';
import 'package:tic_tac/pages/settings.dart';

class GamePage extends StatefulWidget {
  GamePageState createState() => GamePageState();
}

class GamePageState extends State<GamePage> {
  final boardService = locator<BoardService>();
  @override
  Widget build(BuildContext context) {
    return WillPopScope(
      onWillPop: () {
        boardService.newGame();
        return Future.value(true);
      },
      child: SafeArea(
        child: Scaffold(
          backgroundColor: Colors.white,
          body: SafeArea(
            child: StreamBuilder<MapEntry<int, int>>(
                stream: boardService.score$,
                builder: (context, AsyncSnapshot<MapEntry<int, int>> snapshot) {
                  if (!snapshot.hasData) {
                    return Container();
                  }
                  final int xScore = snapshot.data.key;
                  final int oScore = snapshot.data.value;

                  var round = xScore + oScore + 1;
                  return Container(
                    // color: Colors.red,
                    width: MediaQuery.of(context).size.width,
                    child: Column(
                      mainAxisSize: MainAxisSize.max,
                      children: <Widget>[
                        Expanded(
```



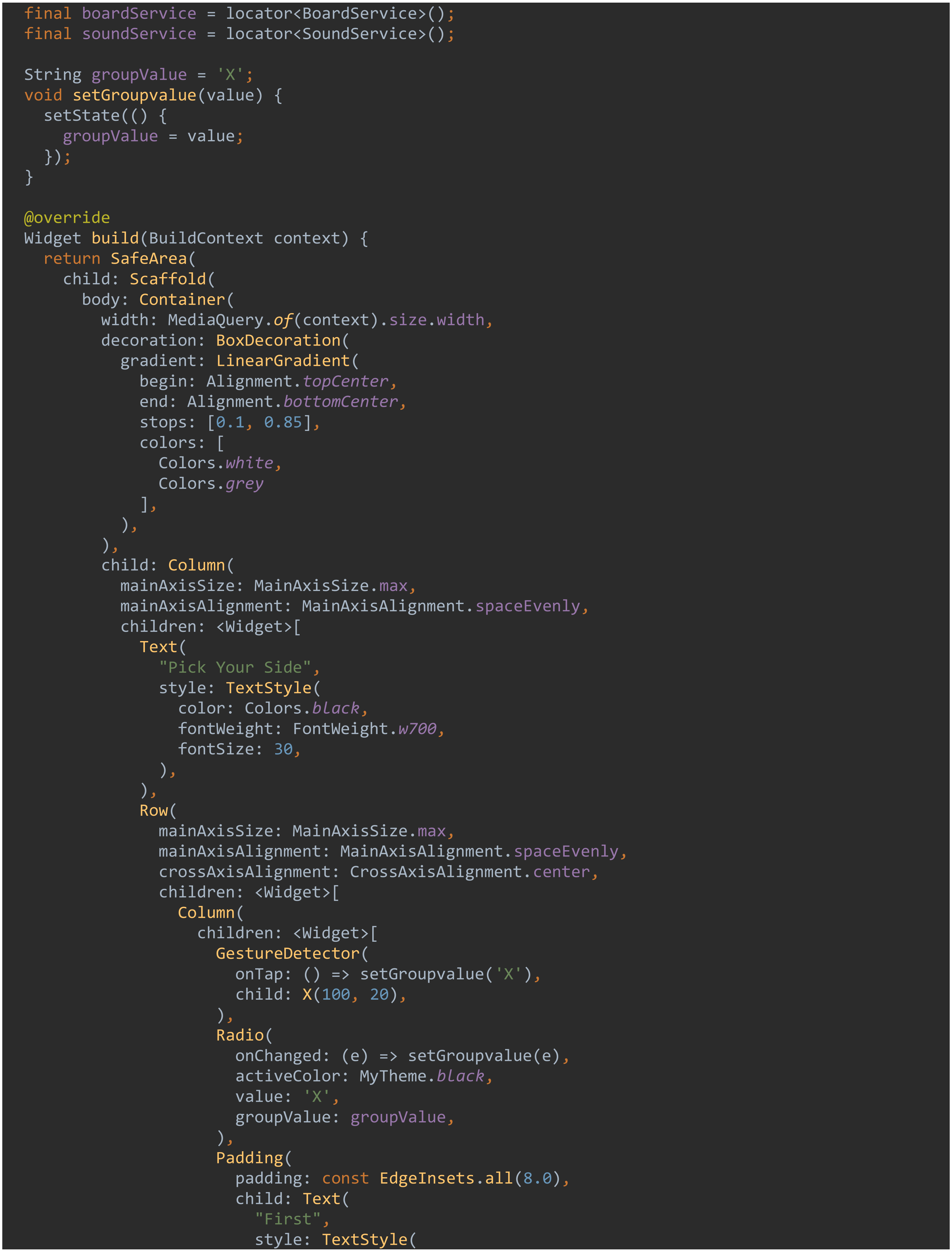
```
child: Column(
  mainAxisSize: MainAxisSize.max,
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: <Widget>[
    Container(
      child: Center(
        child: Container(
          child: Text(
            'Round $round',
            style: GoogleFonts.monoton(
              textStyle: Theme.of(context)
                  .textTheme
                  .bodyText1,
              fontSize: 35,
              color: Colors.black,
              letterSpacing: 2.0,
            ),
          ),
        ),
      ),
    ),
    Container(
      padding: EdgeInsets.symmetric(horizontal: 20),
      color: Colors.white,
      child: Row(
        mainAxisSize: MainAxisSize.max,
        crossAxisAlignment: CrossAxisAlignment.center,
        children: <Widget>[
          SizedBox(
            height: 40,
            width: 40,
            child: Material(
              elevation: 5,
              color: Colors.white,
              borderRadius: BorderRadius.circular(20),
              child: Center(
                  child: Text(
                "$xScore",
                style: TextStyle(
                    color: Colors.black,
                    fontSize: 18),
              )),
            ),
          ),
          Expanded(
            child: Container(),
          ),
          X(35, 10),
          Padding(
            padding: const EdgeInsets.symmetric(
                horizontal: 10),
            child: Text(
              "Player",
              style: TextStyle(fontSize: 20),
            ),
          )
        ],
      ),
    ),
    Container(
      child: Column(
```



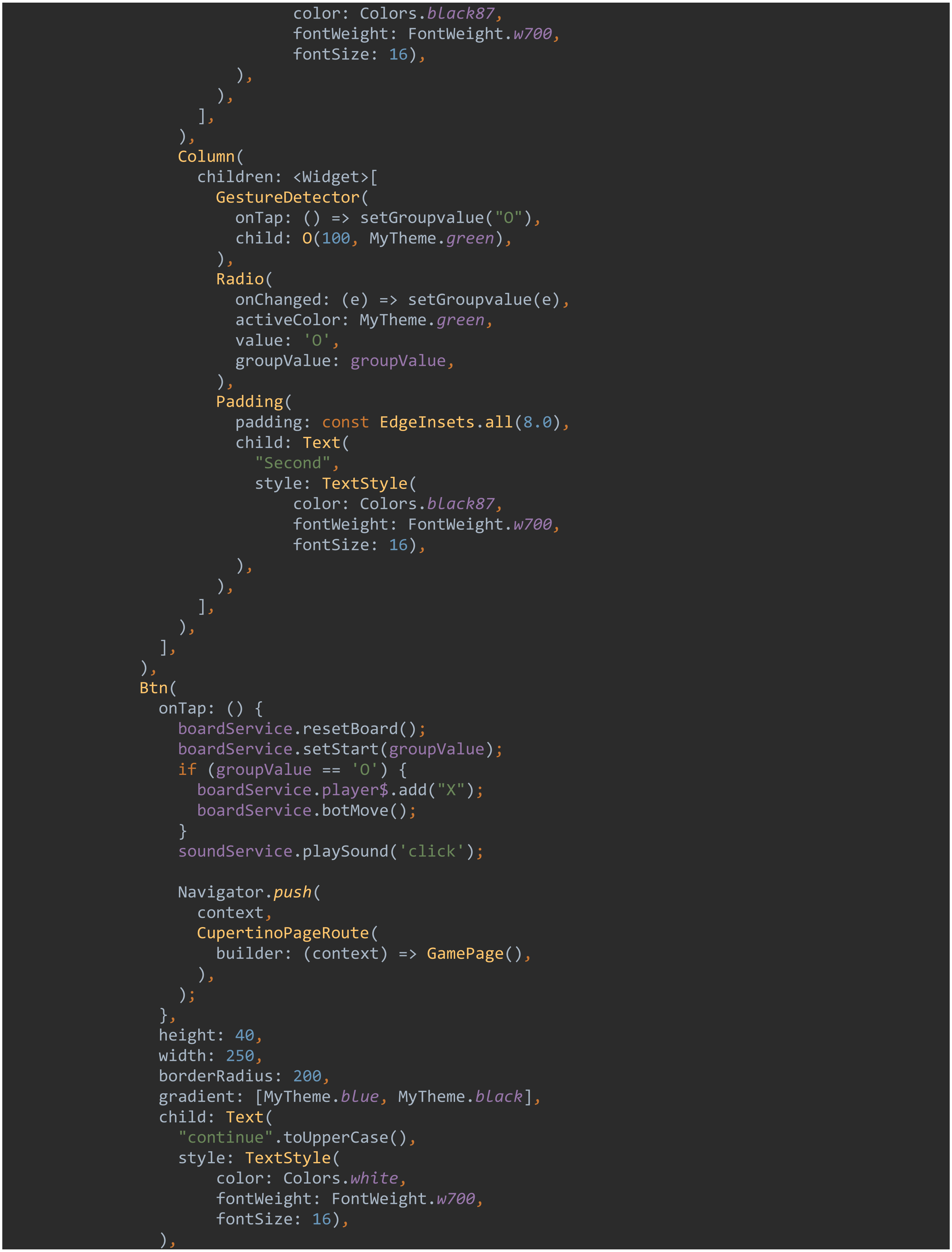
```
                                  mainAxisSize: MainAxisSize.max,
                                  mainAxisAlignment: MainAxisAlignment.center,
                                  children: <Widget>[Board()],
                                ),
                              ),
                              Container(
                                padding: EdgeInsets.symmetric(horizontal: 20),
                                color: Colors.white,
                                child: Row(
                                  mainAxisSize: MainAxisSize.max,
                                  crossAxisAlignment: CrossAxisAlignment.center,
                                  children: <Widget>[
                                    O(35, MyTheme.green),
                                    Padding(
                                      padding: EdgeInsets.symmetric(
                                        horizontal: 10,
                                      ),
                                      child: Text(
                                        "Player",
                                        style: TextStyle(fontSize: 20),
                                      ),
                                    ),
                                    Expanded(
                                      child: Container(),
                                    ),
                                    SizedBox(
                                      height: 40,
                                      width: 40,
                                      child: Material(
                                        elevation: 5,
                                        color: Colors.white,
                                        borderRadius: BorderRadius.circular(20),
                                        child: Center(
                                            child: Text(
                                          "$oScore",
                                          style: TextStyle(
                                              color: Colors.black,
                                              fontSize: 18),
                                        )),
                                      ),
                                    ),
                                  ],
                                ),
                              ),
                            ],
                          ),
                        ),
                        Container(
                          color: Colors.white,
                          height: 60,
                          padding: EdgeInsets.symmetric(horizontal: 20),
                          child: Row(
                            crossAxisAlignment: CrossAxisAlignment.center,
                            children: <Widget>[
                              Btn(
                                onTap: () {
                                  soundService.playSound('click');
                                  Navigator.push(
                                    context,
                                    CupertinoPageRoute(
                                      fullscreenDialog: true,
                                      builder: (context) => SettingsPage(),
                                    ),
                                  );
                                },
                                color: Colors.greenAccent,
                                height: 50,
                                width: 50,
                                borderRadius: 25,
                                child: Icon(Icons.settings),
                              ),
                              Expanded(
                                child: Container(),
                              ),
                              Btn(
                                onTap: () {
                                  boardService.newGame();
                                  Navigator.of(context)
                                      .popUntil((route) => route.isFirst);
                                },
                                color: Colors.blueAccent,
                                height: 50,
                                width: 50,
                                borderRadius: 25,
                                child: Icon(Icons.home),
                              ),
                            ],
                          ),
                        ),
                      ],
                    ),
                  );
                }),
          ),
        ),
      ),
    );
  }
}
```

Pick.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:tic_tac/components/btn.dart';
import 'package:tic_tac/components/o.dart';
import 'package:tic_tac/components/x.dart';
import 'package:tic_tac/services/board.dart';
import 'package:tic_tac/services/provider.dart';
import 'package:tic_tac/services/sound.dart';
import 'package:tic_tac/theme/theme.dart';

import 'game.dart';

class PickPage extends StatefulWidget {
  _PickPageState createState() => _PickPageState();
}

class _PickPageState extends State<PickPage> {
```



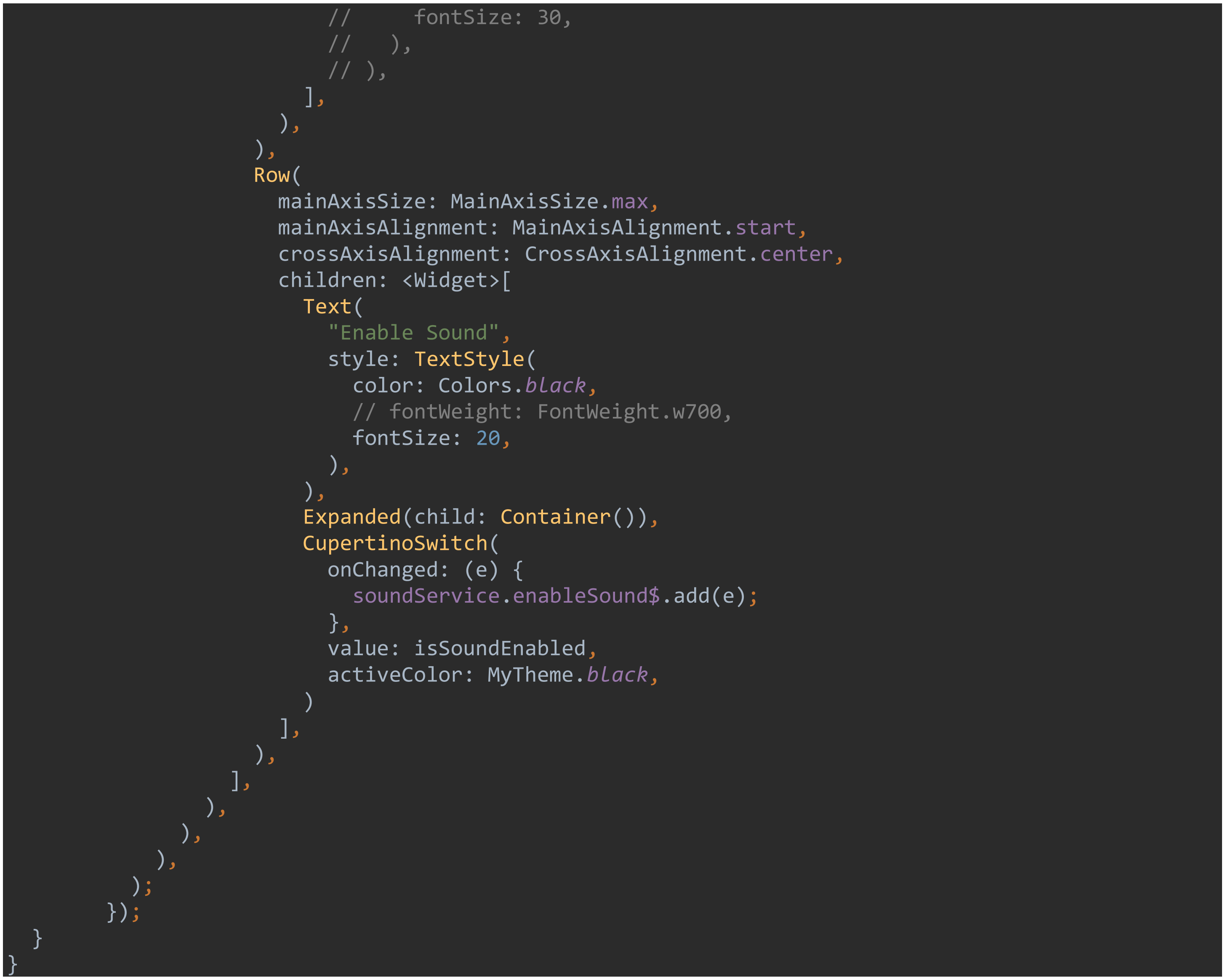
```
  final boardService = locator<BoardService>();
  final soundService = locator<SoundService>();

  String groupValue = 'X';
  void setGroupvalue(value) {
    setState(() {
      groupValue = value;
    });
  }

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        body: Container(
          width: MediaQuery.of(context).size.width,
          decoration: BoxDecoration(
            gradient: LinearGradient(
              begin: Alignment.topCenter,
              end: Alignment.bottomCenter,
              stops: [0.1, 0.85],
              colors: [
                Colors.white,
                Colors.grey
              ],
            ),
          ),
          child: Column(
            mainAxisSize: MainAxisSize.max,
            mainAxisAlignment: MainAxisAlignment.spaceEvenly,
            children: <Widget>[
              Text(
                "Pick Your Side",
                style: TextStyle(
                  color: Colors.black,
                  fontWeight: FontWeight.w700,
                  fontSize: 30,
                ),
              ),
              Row(
                mainAxisSize: MainAxisSize.max,
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                crossAxisAlignment: CrossAxisAlignment.center,
                children: <Widget>[
                  Column(
                    children: <Widget>[
                      GestureDetector(
                        onTap: () => setGroupvalue('X'),
                        child: X(100, 20),
                      ),
                      Radio(
                        onChanged: (e) => setGroupvalue(e),
                        activeColor: MyTheme.black,
                        value: 'X',
                        groupValue: groupValue,
                      ),
                      Padding(
                        padding: const EdgeInsets.all(8.0),
                        child: Text(
                          "First",
                          style: TextStyle(
```



```
               color: Colors.black87,
               fontWeight: FontWeight.w700,
               fontSize: 16),
            ),
          ),
        ],
      ),
      Column(
        children: <Widget>[
          GestureDetector(
            onTap: () => setGroupvalue("O"),
            child: O(100, MyTheme.green),
          ),
          Radio(
            onChanged: (e) => setGroupvalue(e),
            activeColor: MyTheme.green,
            value: 'O',
            groupValue: groupValue,
          ),
          Padding(
            padding: const EdgeInsets.all(8.0),
            child: Text(
              "Second",
              style: TextStyle(
                color: Colors.black87,
                fontWeight: FontWeight.w700,
                fontSize: 16),
            ),
          ),
        ],
      ),
    ],
  ),
  Btn(
    onTap: () {
      boardService.resetBoard();
      boardService.setStart(groupValue);
      if (groupValue == 'O') {
        boardService.player$.add("X");
        boardService.botMove();
      }
      soundService.playSound('click');

      Navigator.push(
        context,
        CupertinoPageRoute(
          builder: (context) => GamePage(),
        ),
      );
    },
    height: 40,
    width: 250,
    borderRadius: 200,
    gradient: [MyTheme.blue, MyTheme.black],
    child: Text(
      "continue".toUpperCase(),
      style: TextStyle(
          color: Colors.white,
          fontWeight: FontWeight.w700,
          fontSize: 16),
    ),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

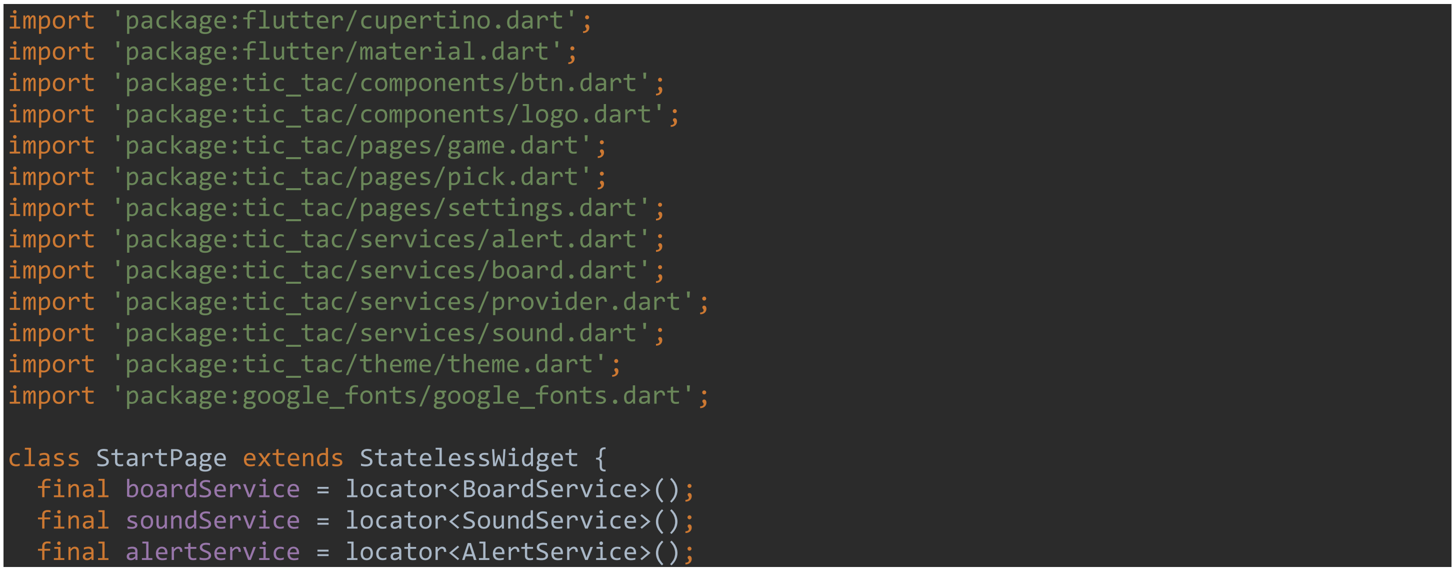
Settings.dart

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:tic_tac/services/provider.dart';
import 'package:tic_tac/services/sound.dart';
import 'package:tic_tac/theme/theme.dart';

class SettingsPage extends StatefulWidget {
  SettingsPage({Key key}) : super(key: key);

  SettingsPageState createState() => SettingsPageState();
}

class SettingsPageState extends State<SettingsPage> {
  final soundService = locator<SoundService>();

  @override
  Widget build(BuildContext context) {
    return StreamBuilder<bool>(
        stream: soundService.enableSound$,
        builder: (context, AsyncSnapshot<bool> snapshot) {
          if (!snapshot.hasData) {
            return Container();
          }

          final bool isSoundEnabled = snapshot.data;

          return Scaffold(
            body: Container(
              width: MediaQuery.of(context).size.width,
              child: Padding(
                padding:
                    const EdgeInsets.symmetric(vertical: 15, horizontal: 20),
                child: Column(
                  mainAxisSize: MainAxisSize.max,
                  mainAxisAlignment: MainAxisAlignment.start,
                  children: <Widget>[
                    Padding(
                      padding: const EdgeInsets.symmetric(vertical: 20),
                      child: Row(
                        mainAxisSize: MainAxisSize.max,
                        mainAxisAlignment: MainAxisAlignment.start,
                        children: <Widget>[
                          // Text(
                          //   "Settings".toUpperCase(),
                          //   style: TextStyle(
                          //     color: Colors.black,
                          //     fontWeight: FontWeight.w700,
```

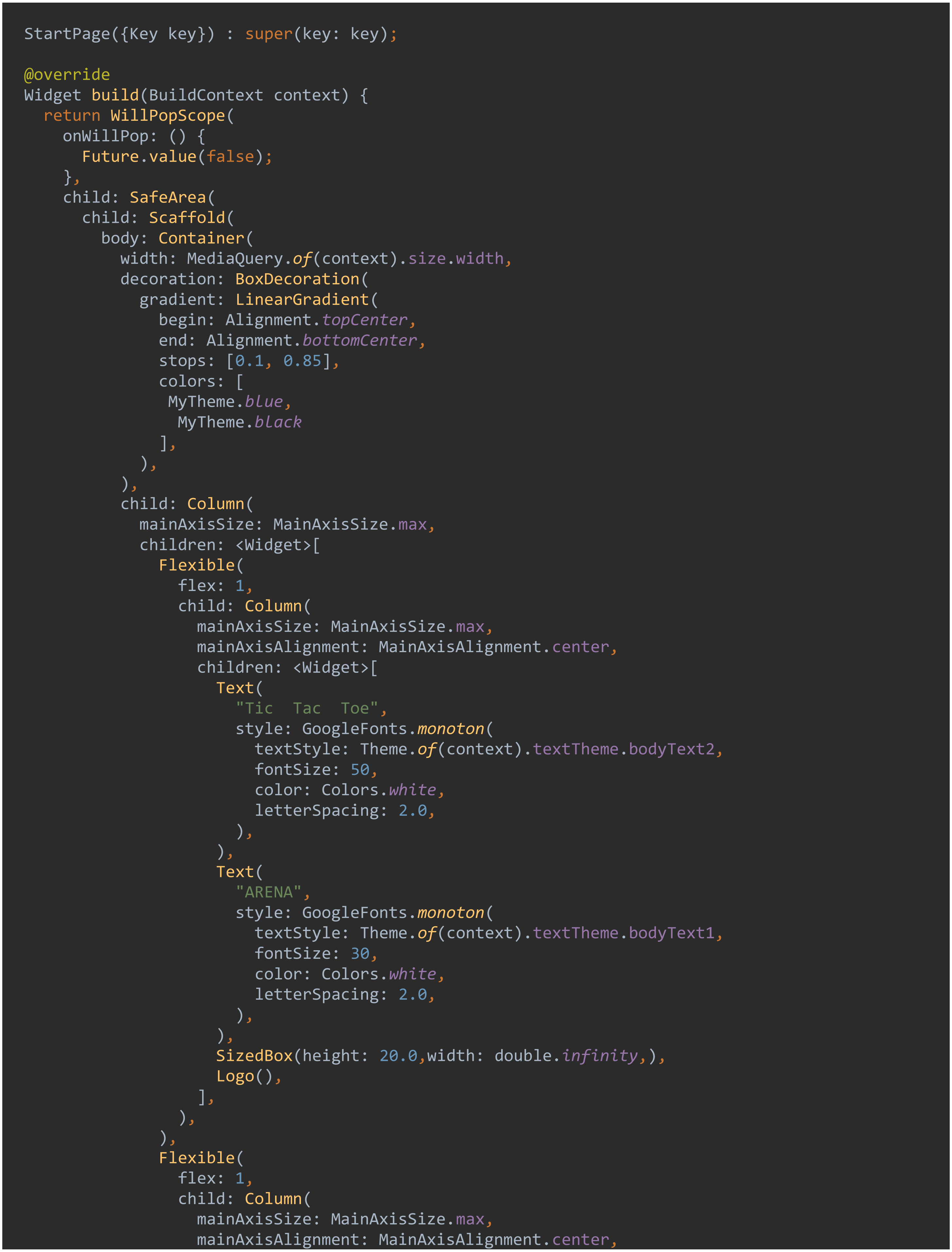


```
                          //     fontSize: 30,
                          //   ),
                          // ),
                        ],
                      ),
                    ),
                    Row(
                      mainAxisSize: MainAxisSize.max,
                      mainAxisAlignment: MainAxisAlignment.start,
                      crossAxisAlignment: CrossAxisAlignment.center,
                      children: <Widget>[
                        Text(
                          "Enable Sound",
                          style: TextStyle(
                            color: Colors.black,
                            // fontWeight: FontWeight.w700,
                            fontSize: 20,
                          ),
                        ),
                        Expanded(child: Container()),
                        CupertinoSwitch(
                          onChanged: (e) {
                            soundService.enableSound$.add(e);
                          },
                          value: isSoundEnabled,
                          activeColor: MyTheme.black,
                        )
                      ],
                    ),
                  ],
                ),
              ),
            ),
          );
        });
  }
}
```

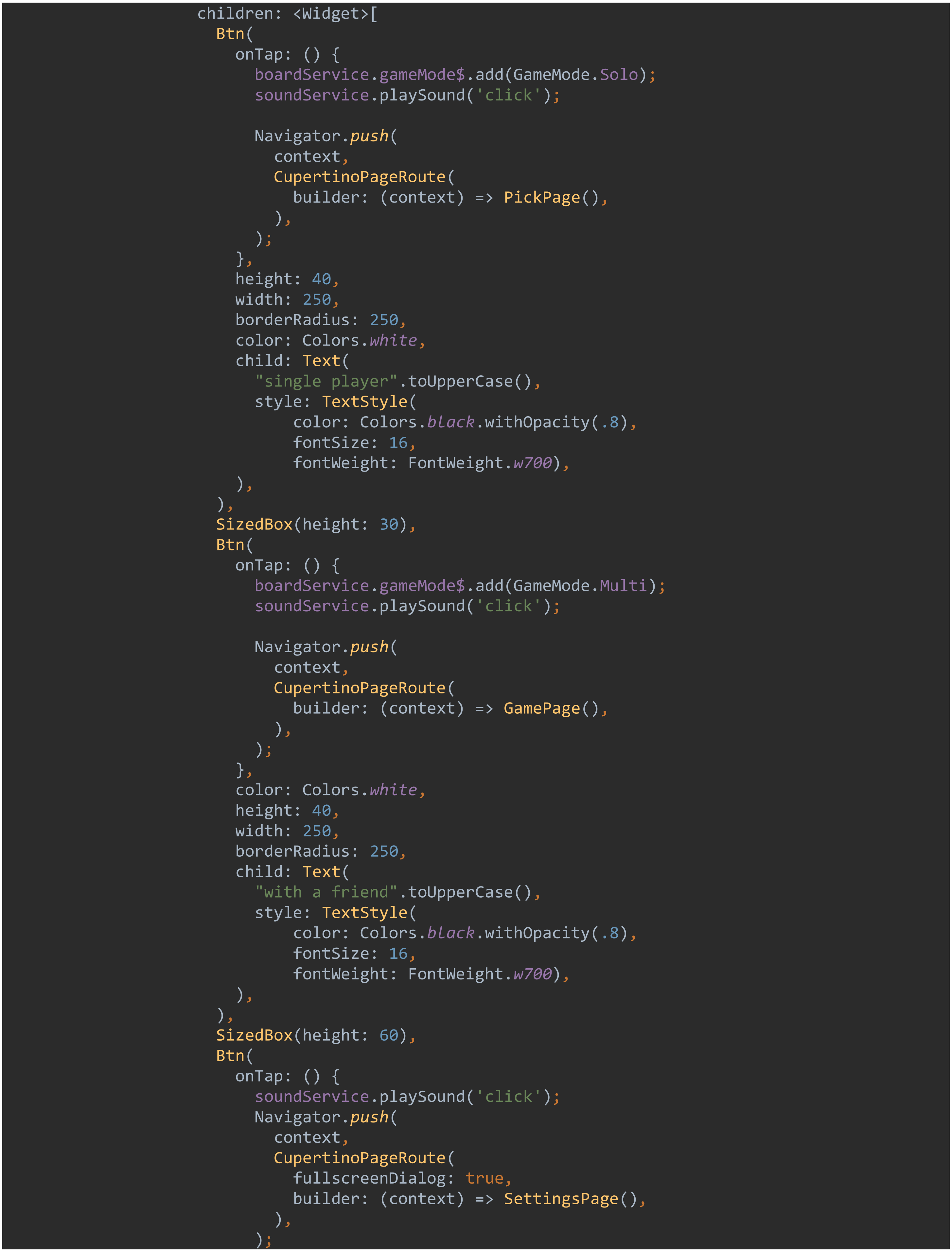
Start.dart



```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:tic_tac/components/btn.dart';
import 'package:tic_tac/components/logo.dart';
import 'package:tic_tac/pages/game.dart';
import 'package:tic_tac/pages/pick.dart';
import 'package:tic_tac/pages/settings.dart';
import 'package:tic_tac/services/alert.dart';
import 'package:tic_tac/services/board.dart';
import 'package:tic_tac/services/provider.dart';
import 'package:tic_tac/services/sound.dart';
import 'package:tic_tac/theme/theme.dart';
import 'package:google_fonts/google_fonts.dart';

class StartPage extends StatelessWidget {
  final boardService = locator<BoardService>();
  final soundService = locator<SoundService>();
  final alertService = locator<AlertService>();
```



```
StartPage({Key key}) : super(key: key);

@override
Widget build(BuildContext context) {
  return WillPopScope(
    onWillPop: () {
      Future.value(false);
    },
    child: SafeArea(
      child: Scaffold(
        body: Container(
          width: MediaQuery.of(context).size.width,
          decoration: BoxDecoration(
            gradient: LinearGradient(
              begin: Alignment.topCenter,
              end: Alignment.bottomCenter,
              stops: [0.1, 0.85],
              colors: [
               MyTheme.blue,
                MyTheme.black
              ],
            ),
          ),
          child: Column(
            mainAxisSize: MainAxisSize.max,
            children: <Widget>[
              Flexible(
                flex: 1,
                child: Column(
                  mainAxisSize: MainAxisSize.max,
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: <Widget>[
                    Text(
                      "Tic  Tac  Toe",
                      style: GoogleFonts.monoton(
                        textStyle: Theme.of(context).textTheme.bodyText2,
                        fontSize: 50,
                        color: Colors.white,
                        letterSpacing: 2.0,
                      ),
                    ),
                    Text(
                      "ARENA",
                      style: GoogleFonts.monoton(
                        textStyle: Theme.of(context).textTheme.bodyText1,
                        fontSize: 30,
                        color: Colors.white,
                        letterSpacing: 2.0,
                      ),
                    ),
                    SizedBox(height: 20.0,width: double.infinity,),
                    Logo(),
                  ],
                ),
              ),
              Flexible(
                flex: 1,
                child: Column(
                  mainAxisSize: MainAxisSize.max,
                  mainAxisAlignment: MainAxisAlignment.center,
```



```
                    children: <Widget>[
                      Btn(
                        onTap: () {
                          boardService.gameMode$.add(GameMode.Solo);
                          soundService.playSound('click');

                          Navigator.push(
                            context,
                            CupertinoPageRoute(
                              builder: (context) => PickPage(),
                            ),
                          );
                        },
                        height: 40,
                        width: 250,
                        borderRadius: 250,
                        color: Colors.white,
                        child: Text(
                          "single player".toUpperCase(),
                          style: TextStyle(
                              color: Colors.black.withOpacity(.8),
                              fontSize: 16,
                              fontWeight: FontWeight.w700),
                        ),
                      ),
                      SizedBox(height: 30),
                      Btn(
                        onTap: () {
                          boardService.gameMode$.add(GameMode.Multi);
                          soundService.playSound('click');

                          Navigator.push(
                            context,
                            CupertinoPageRoute(
                              builder: (context) => GamePage(),
                            ),
                          );
                        },
                        color: Colors.white,
                        height: 40,
                        width: 250,
                        borderRadius: 250,
                        child: Text(
                          "with a friend".toUpperCase(),
                          style: TextStyle(
                              color: Colors.black.withOpacity(.8),
                              fontSize: 16,
                              fontWeight: FontWeight.w700),
                        ),
                      ),
                      SizedBox(height: 60),
                      Btn(
                        onTap: () {
                          soundService.playSound('click');
                          Navigator.push(
                            context,
                            CupertinoPageRoute(
                              fullscreenDialog: true,
                              builder: (context) => SettingsPage(),
                            ),
                          );
                        },
                        color: Colors.white,
                        height: 50,
                        width: 50,
                        borderRadius: 25,
                        child: Icon(Icons.settings),
                      ),
                    ],
                  ),
                ),
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```

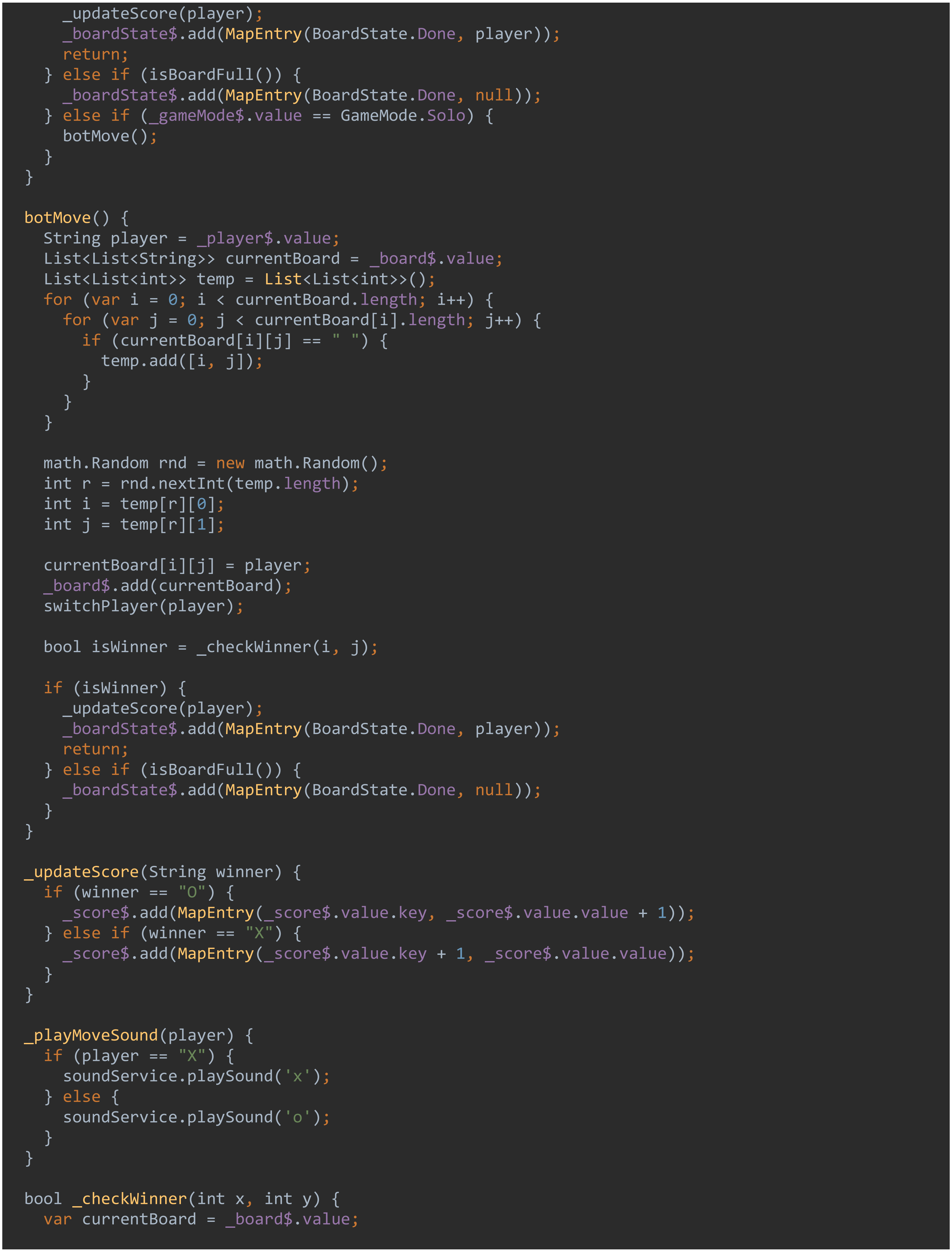
Services:

Alert.dart

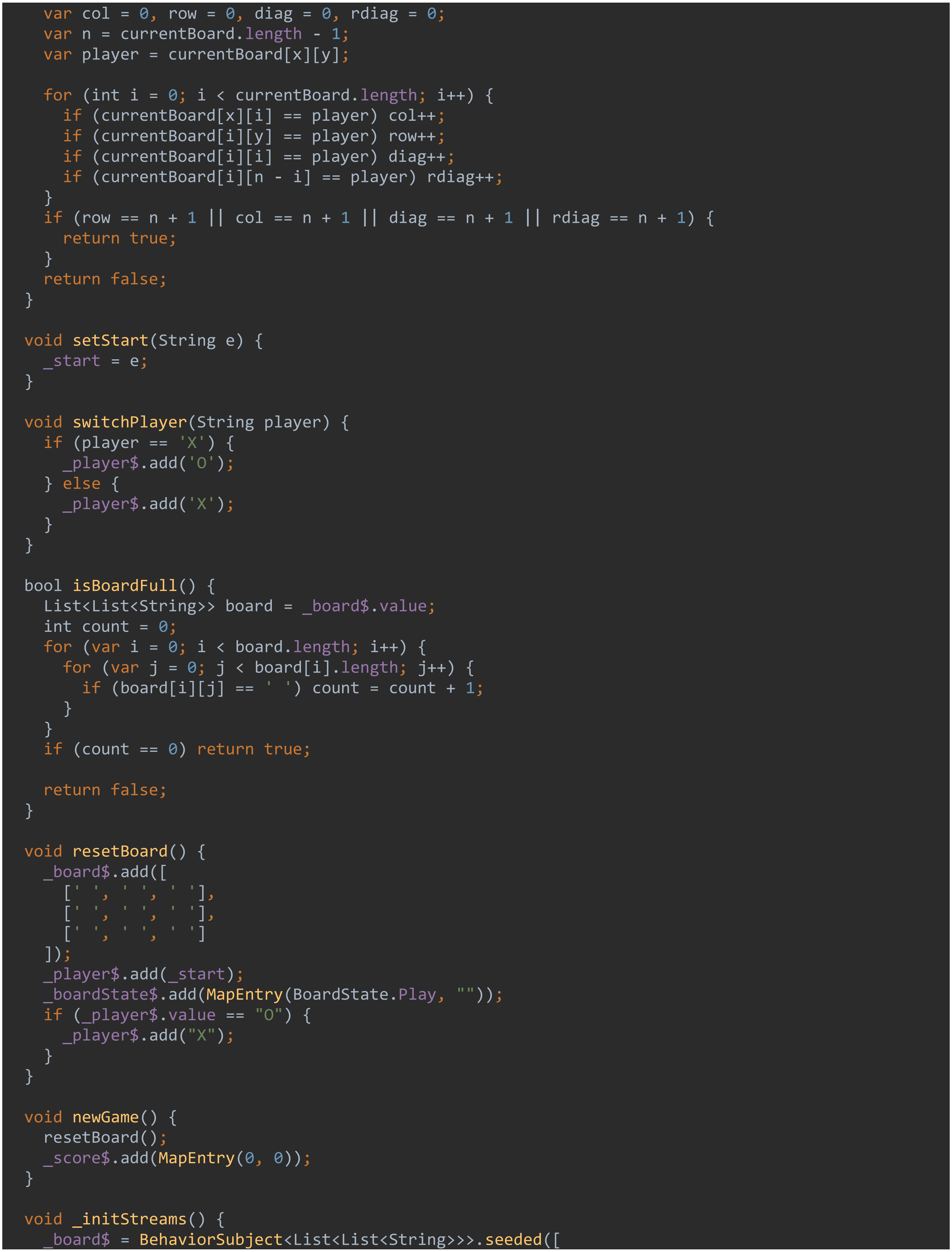
```
import 'package:flutter/material.dart';
import 'package:rflutter_alert/rflutter_alert.dart';

class AlertService {
  AlertStyle _resultAlertStyle;
  AlertStyle get resultAlertStyle => _resultAlertStyle;

  AlertStyle _settingsAlertStyle;
  AlertStyle get settingsAlertStyle => _settingsAlertStyle;

  AlertService() {
    _resultAlertStyle = AlertStyle(
      animationType: AnimationType.grow,
      isCloseButton: false,
      isOverlayTapDismiss: true,
      titleStyle: TextStyle(
          color: Color(0xff111111), fontWeight: FontWeight.w700, fontSize: 25),
      descStyle: TextStyle(fontWeight: FontWeight.bold),
      animationDuration: Duration(milliseconds: 300),
      buttonAreaPadding: EdgeInsets.all(12),
      overlayColor: Colors.black.withOpacity(.7),
      constraints: BoxConstraints(maxHeight: 200, maxWidth: 250),
      alertBorder: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10.0),
      ),
    );
    _settingsAlertStyle = AlertStyle(
      animationType: AnimationType.fromBottom,
      isCloseButton: false,
      isOverlayTapDismiss: true,
      titleStyle: TextStyle(
        color: Color(0xff111111),
        fontWeight: FontWeight.w700,
        fontSize: 25,
      ),
      // animationDuration: Duration(milliseconds: 300),
      buttonAreaPadding: EdgeInsets.all(12),
      // overlayColor: Colors.black.withOpacity(.5),
      // constraints: BoxConstraints(maxHeight: 200, maxWidth: 250),
      alertBorder: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(10.0),
      ),
    );
  }
}
```

Board.dart

```
import 'package:rxdart/rxdart.dart';
import 'dart:math' as math;

import 'package:tic_tac/services/provider.dart';
import 'package:tic_tac/services/sound.dart';

final soundService = locator<SoundService>();

enum BoardState { Done, Play }
enum GameMode { Solo, Multi }

class BoardService {
  BehaviorSubject<List<List<String>>> _board$;
  BehaviorSubject<List<List<String>>> get board$ => _board$;

  BehaviorSubject<String> _player$;
  BehaviorSubject<String> get player$ => _player$;

  BehaviorSubject<MapEntry<BoardState, String>> _boardState$;
  BehaviorSubject<MapEntry<BoardState, String>> get boardState$ => _boardState$;

  BehaviorSubject<GameMode> _gameMode$;
  BehaviorSubject<GameMode> get gameMode$ => _gameMode$;

  BehaviorSubject<MapEntry<int, int>> _score$;
  BehaviorSubject<MapEntry<int, int>> get score$ => _score$;

  String _start;

  BoardService() {
    _initStreams();
  }

  void newMove(int i, int j) {
    String player = _player$.value;
    List<List<String>> currentBoard = _board$.value;

    currentBoard[i][j] = player;
    _playMoveSound(player);
    _board$.add(currentBoard);
    switchPlayer(player);

    bool isWinner = _checkWinner(i, j);

    if (isWinner) {
```



```
    _updateScore(player);
    _boardState$.add(MapEntry(BoardState.Done, player));
    return;
  } else if (isBoardFull()) {
    _boardState$.add(MapEntry(BoardState.Done, null));
  } else if (_gameMode$.value == GameMode.Solo) {
    botMove();
  }
}

botMove() {
  String player = _player$.value;
  List<List<String>> currentBoard = _board$.value;
  List<List<int>> temp = List<List<int>>();
  for (var i = 0; i < currentBoard.length; i++) {
    for (var j = 0; j < currentBoard[i].length; j++) {
      if (currentBoard[i][j] == " ") {
        temp.add([i, j]);
      }
    }
  }

  math.Random rnd = new math.Random();
  int r = rnd.nextInt(temp.length);
  int i = temp[r][0];
  int j = temp[r][1];

  currentBoard[i][j] = player;
  _board$.add(currentBoard);
  switchPlayer(player);

  bool isWinner = _checkWinner(i, j);

  if (isWinner) {
    _updateScore(player);
    _boardState$.add(MapEntry(BoardState.Done, player));
    return;
  } else if (isBoardFull()) {
    _boardState$.add(MapEntry(BoardState.Done, null));
  }
}

_updateScore(String winner) {
  if (winner == "O") {
    _score$.add(MapEntry(_score$.value.key, _score$.value.value + 1));
  } else if (winner == "X") {
    _score$.add(MapEntry(_score$.value.key + 1, _score$.value.value));
  }
}

_playMoveSound(player) {
  if (player == "X") {
    soundService.playSound('x');
  } else {
    soundService.playSound('o');
  }
}

bool _checkWinner(int x, int y) {
  var currentBoard = _board$.value;
```



```
    var col = 0, row = 0, diag = 0, rdiag = 0;
    var n = currentBoard.length - 1;
    var player = currentBoard[x][y];

    for (int i = 0; i < currentBoard.length; i++) {
      if (currentBoard[x][i] == player) col++;
      if (currentBoard[i][y] == player) row++;
      if (currentBoard[i][i] == player) diag++;
      if (currentBoard[i][n - i] == player) rdiag++;
    }
    if (row == n + 1 || col == n + 1 || diag == n + 1 || rdiag == n + 1) {
      return true;
    }
    return false;
  }

  void setStart(String e) {
    _start = e;
  }

  void switchPlayer(String player) {
    if (player == 'X') {
      _player$.add('O');
    } else {
      _player$.add('X');
    }
  }

  bool isBoardFull() {
    List<List<String>> board = _board$.value;
    int count = 0;
    for (var i = 0; i < board.length; i++) {
      for (var j = 0; j < board[i].length; j++) {
        if (board[i][j] == ' ') count = count + 1;
      }
    }
    if (count == 0) return true;

    return false;
  }

  void resetBoard() {
    _board$.add([
      [' ', ' ', ' '],
      [' ', ' ', ' '],
      [' ', ' ', ' ']
    ]);
    _player$.add(_start);
    _boardState$.add(MapEntry(BoardState.Play, ""));
    if (_player$.value == "O") {
      _player$.add("X");
    }
  }

  void newGame() {
    resetBoard();
    _score$.add(MapEntry(0, 0));
  }

  void _initStreams() {
    _board$ = BehaviorSubject<List<List<String>>>.seeded([
      [' ', ' ', ' '],
      [' ', ' ', ' '],
      [' ', ' ', ' ']
    ]);
    _player$ = BehaviorSubject<String>.seeded("X");
    _boardState$ = BehaviorSubject<MapEntry<BoardState, String>>.seeded(
      MapEntry(BoardState.Play, ""),
    );
    _gameMode$ = BehaviorSubject<GameMode>.seeded(GameMode.Solo);
    _score$ = BehaviorSubject<MapEntry<int, int>>.seeded(MapEntry(0, 0));
    _start = 'X';
  }
}
```

Provider.dart

```
import 'package:get_it/get_it.dart';
import 'package:tic_tac/services/alert.dart';
import 'package:tic_tac/services/board.dart';
import 'package:tic_tac/services/sound.dart';

GetIt locator = new GetIt();

void setupLocator() {
  locator.registerSingleton(BoardService());
  locator.registerSingleton(SoundService());
  locator.registerSingleton(AlertService());
}
```

Sound.dart

```
import 'package:audioplayers/audio_cache.dart';
import 'package:audioplayers/audioplayers.dart';
import 'package:rxdart/rxdart.dart';

class SoundService {
  BehaviorSubject<bool> _enableSound$;
  BehaviorSubject<bool> get enableSound$ => _enableSound$;
  AudioPlayer _fixedPlayer;
  AudioCache _player;

  SoundService() {
    _enableSound$ = BehaviorSubject<bool>.seeded(true);
    _fixedPlayer = AudioPlayer(mode: PlayerMode.LOW_LATENCY);
    _player = AudioCache(fixedPlayer: _fixedPlayer);
    _player.loadAll(['x.mp3', 'o.mp3', "click.mp3"]);
  }

  playSound(String sound) {
    bool isSoundEnabled = _enableSound$.value;
    if (isSoundEnabled) {
      _player.play("$sound.mp3");
    }
  }
}
```

Theame:

Theme.dart

```dart
import 'dart:ui';

class MyTheme {
  static Color black = Color(0xff000000);
  static Color blue = Color(0xff00507C);
  static Color green = Color(0xfff7db5c);
}
```