PLANNING PROCESS

BRAINSTORMING

Idea	What works	What doesn't work
Flashcard Game	Simple to explain Easy to implement in many different subjects	Too vague Not entertaining and engaging for people to play No way to create different difficulty levels
Math Game: Operations	Fun way to work on math skills & think outside the box Easy to implement different levels and track progress	Focuses on one subject only
Spelling Game	Simple to explain	Requires an audio portion Too many possible inputs

DRAFT

Chosen Game: Math Game: Operations

DRAFT #1

Game Name:

- XXX for Roman Numeral 30 because the numbers must equal to 30

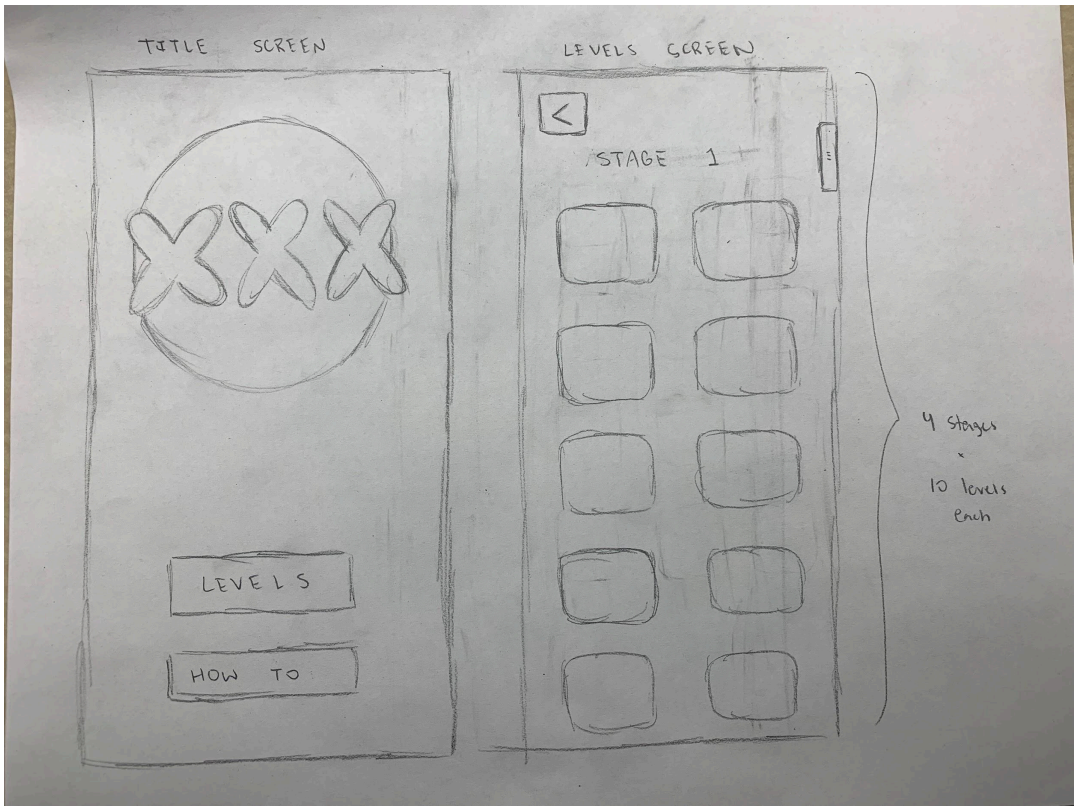
Game Instructions Draft:

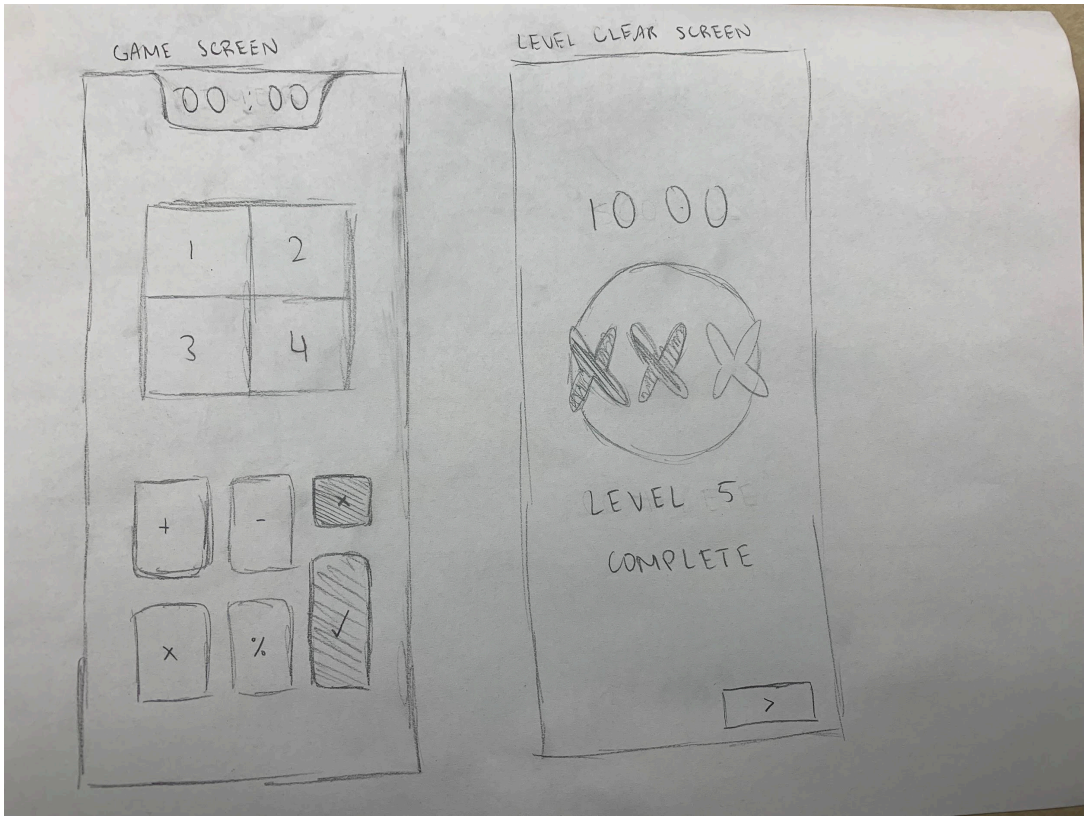
- 4 numbers must end up equal to 30
- operations must be chosen to manipulate the equation

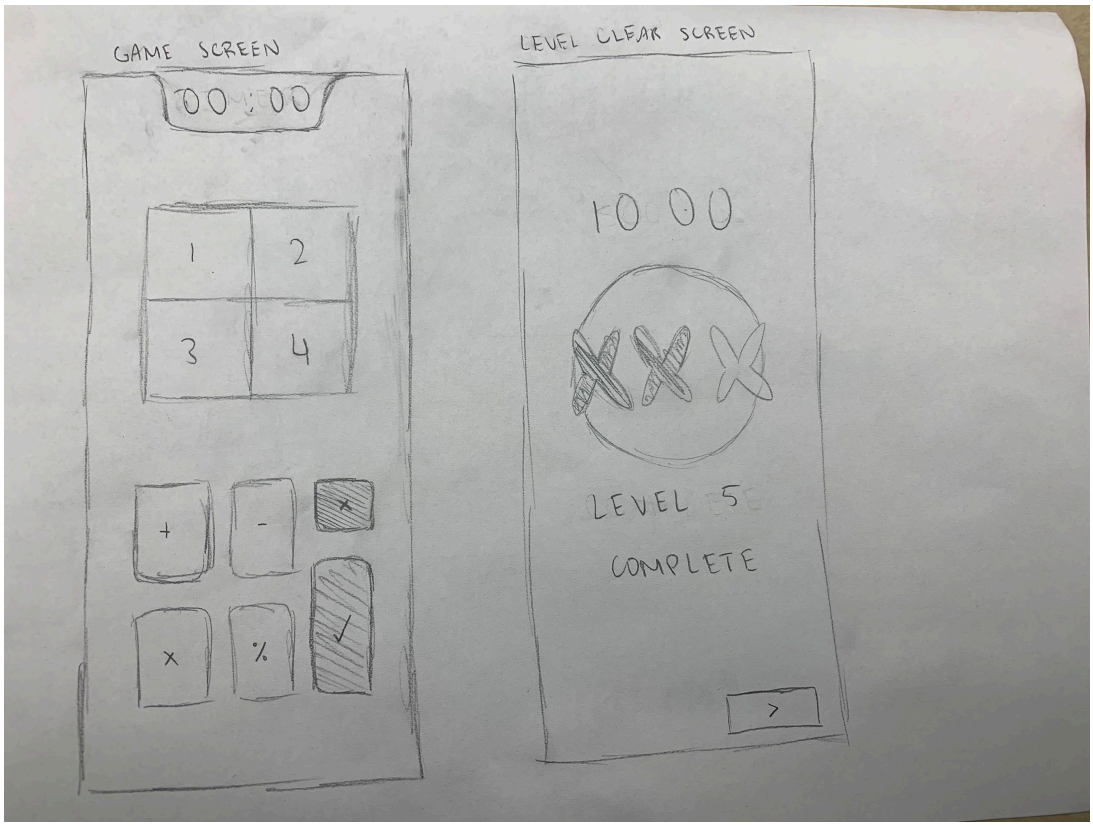
Game Functions:

- there will be a stopwatch
- the number of "stars" the player obtains after each level depends on how fast they complete the level
- must use all 4 numbers in equation
- there will be a pause button
- there will be 4 stages
    - each ascending stage will increase in difficulty (adds another operation type)
- there will be an instructions tab in the "how to" section

Game Design Draft:




GAME SCREEN
00 : 00
1
2
3
4
+
-
x
x
%
✓
LEVEL CLEAR SCREEN
10 00
LEVEL 5
COMPLETE
>


GAME SCREEN
00 : 00
1
2
3
4
+
-
x
x
%
✓
LEVEL CLEAR SCREEN
10 00
LEVEL 5
COMPLETE
>

DRAFT #2

Game Name

- XXV, which is 25 in Roman Numerals since the numbers must equal to 25
- Reason for change:
    - 25 is a more obscure number → makes the game more challenging

Game Instructions Draft:

- 4 numbers must end up equal to 25
- operations must be chosen to manipulate the equation

Game Functions:

- there will be a stopwatch
- the number of "stars" the player obtains after each level depends on how fast they complete the level
- must use all 4 numbers in equation
- there will be a pause button
- there will be 4 stages
    - each ascending stage will increase in difficulty (adds another operation type)
- there will be an instructions tab in the "how to" section

Game Design Draft:

- (same draft as the one in DRAFT #1; only change is the logo)

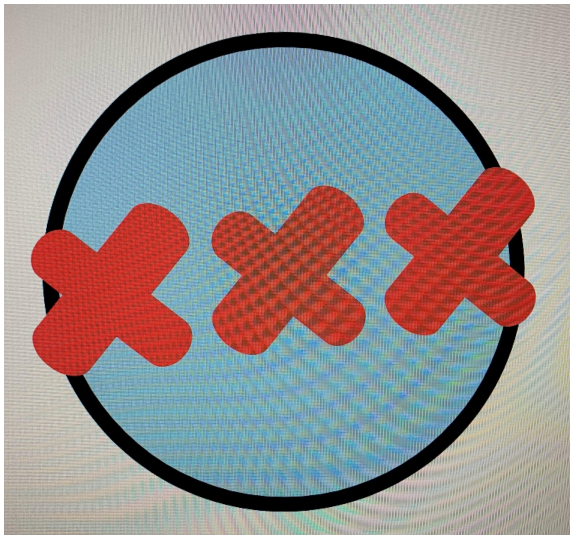
LOGO DESIGN

Application Used: Canva - https://www.canva.com

Logo Design Draft Progression:

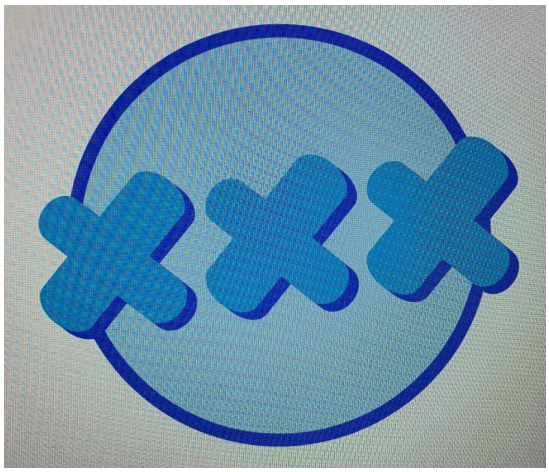
*Disclaimer: Some photos are taken off a computer monitor; drafts done BEFORE the name of the application changed from "XXX" to "XXV"

- LOGO 1 -



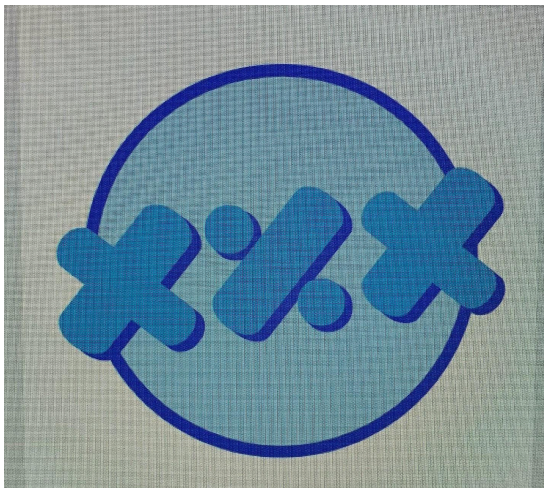
What worked	What didn't work
Simple & easily legible Compact & concise Good shape for an app icon Good concept - incorporating game name	Too bland & lacks dimension Colors do not match well Doesn't give the impression that it is a math-related game Too much negative space

- LOGO 2 -



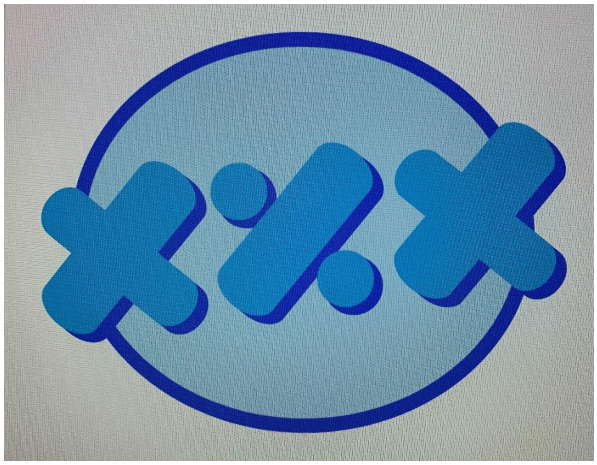
What worked	What didn't work
Simple & easily legible Better color palette More dimension & depth - the game name has a nice shadow that makes it pop out	Doesn't give the impression that it is a math-related game Too much negative space

- LOGO 3 -



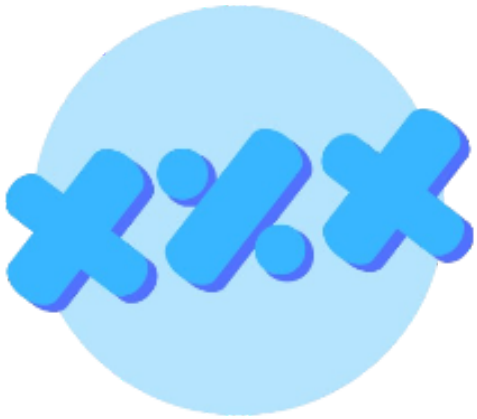
What worked	What didn't work
Relates the Logo to the type of game: Math game that tests players on their operation abilities	Color palette still not the most appealing

- LOGO 4 -



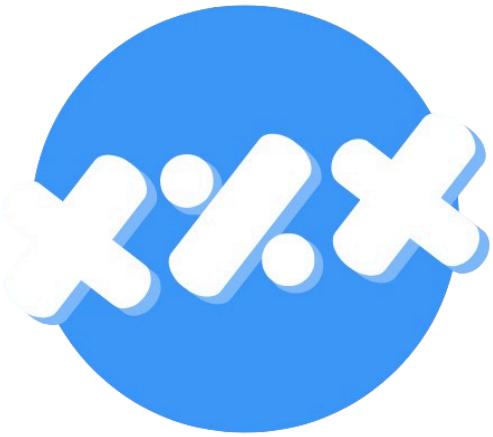
What worked	What didn't work
Less negative space	Shape is awkward

- LOGO 5 -



What worked	What didn't work
Better formatted (without border line) More legible (greater contrast between game name and background)	Colors do not match as well

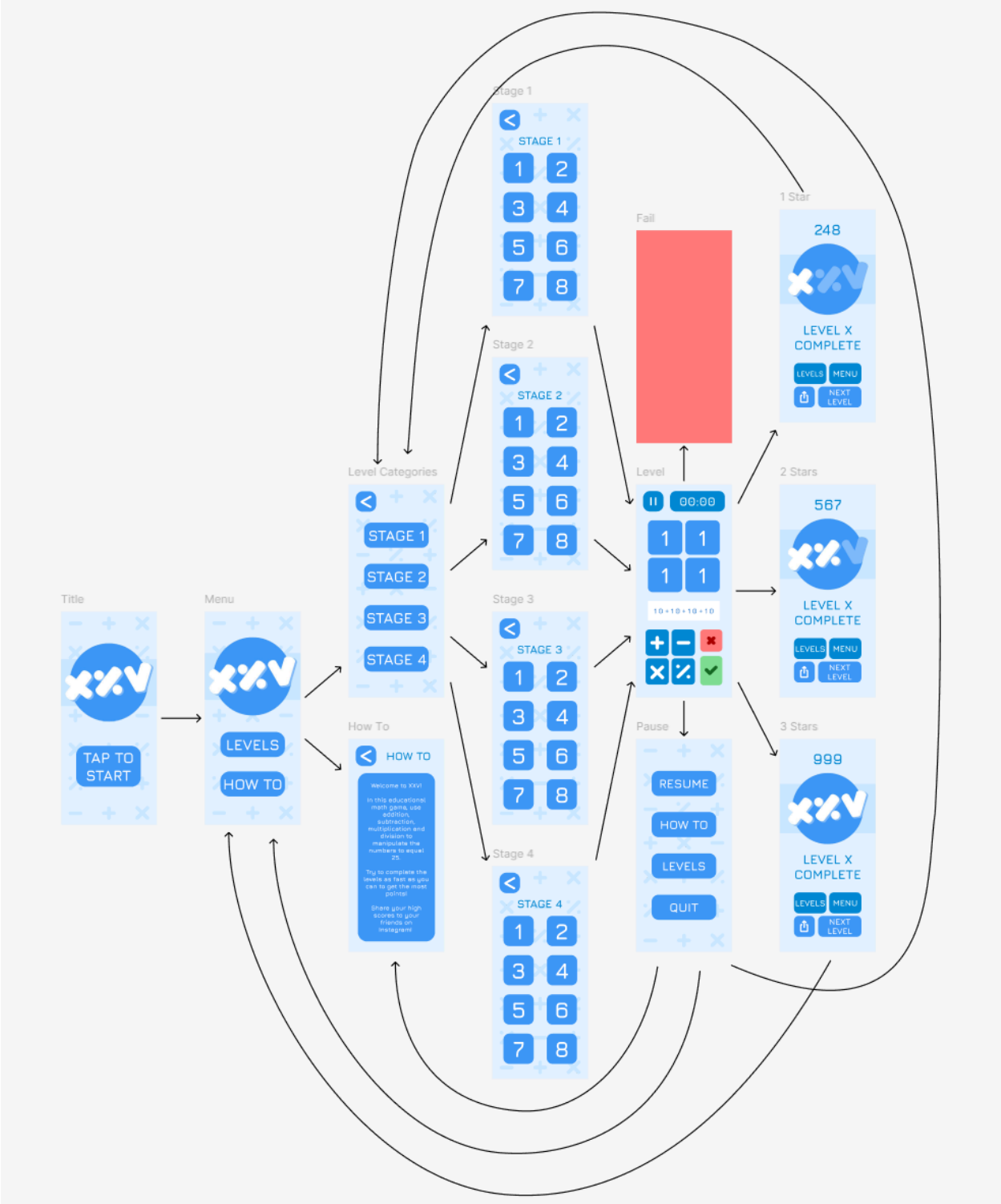
- LOGO 6 -



What worked	What didn't work
Better color palette (the background of the game will not be white so it will be legible)	Needs to change the last "X" into a "V"

- FINAL LOGO DESIGN -

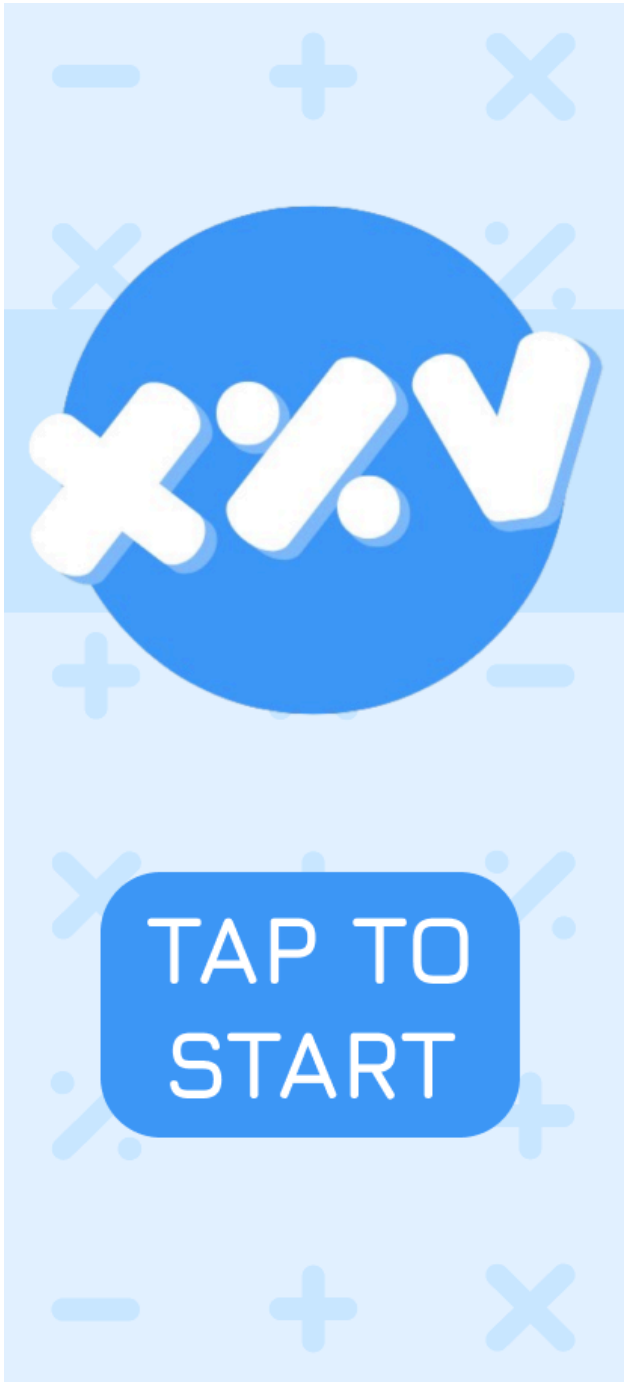
STORYBOARD


Title
TAP TO START
Menu
LEVELS
HOW TO
Level Categories
STAGE 1
STAGE 2
STAGE 3
STAGE 4
How To
HOW TO
Stage 1
STAGE 1
Stage 2
STAGE 2
Stage 3
STAGE 3
Stage 4
STAGE 4
Fail
Level
00:00
Pause
RESUME
HOW TO
LEVELS
QUIT
1 Star
248
LEVEL X COMPLETE
LEVELS
MENU
NEXT LEVEL
2 Stars
567
LEVEL X COMPLETE
3 Stars
999
LEVEL X COMPLETE

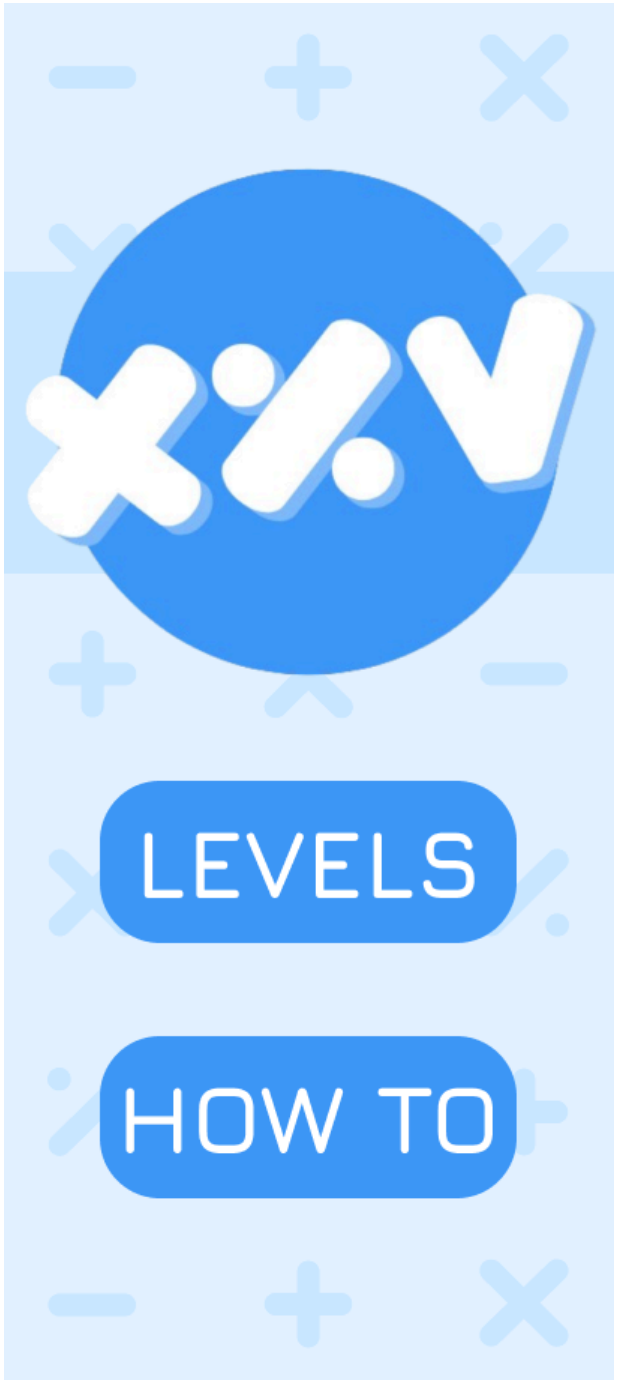
APPLICATION PAGE DESIGNS

Application used: Figma - https://www.figma.com

TITLE SCREEN



MENU



LEVELS CATEGORIES

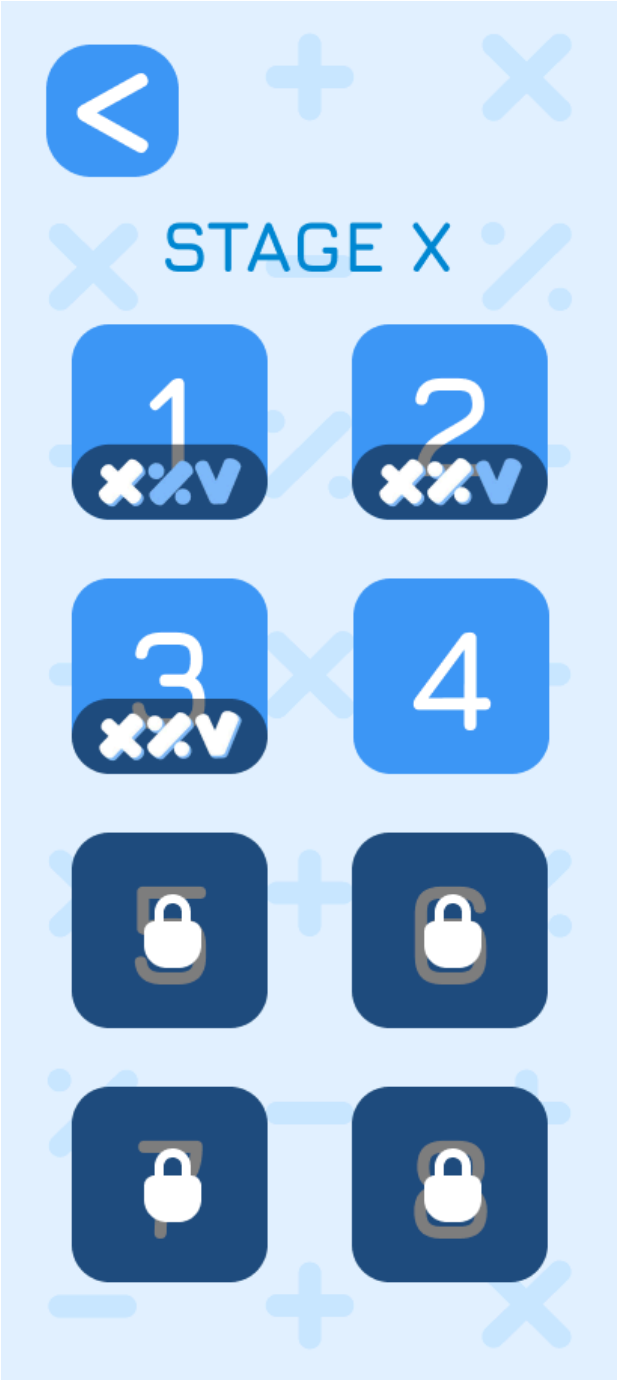
LEVELS: STAGE 1

LEVELS: STAGE 2

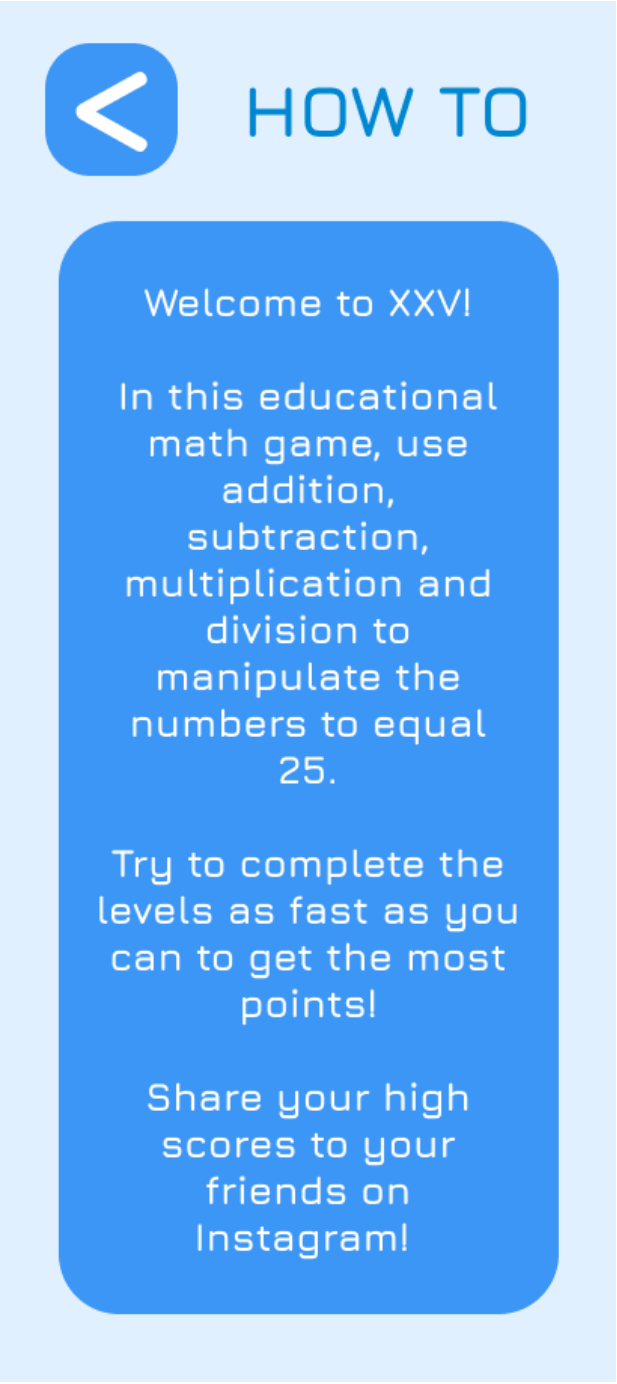
LEVELS: STAGE 3

LEVELS: STAGE 4

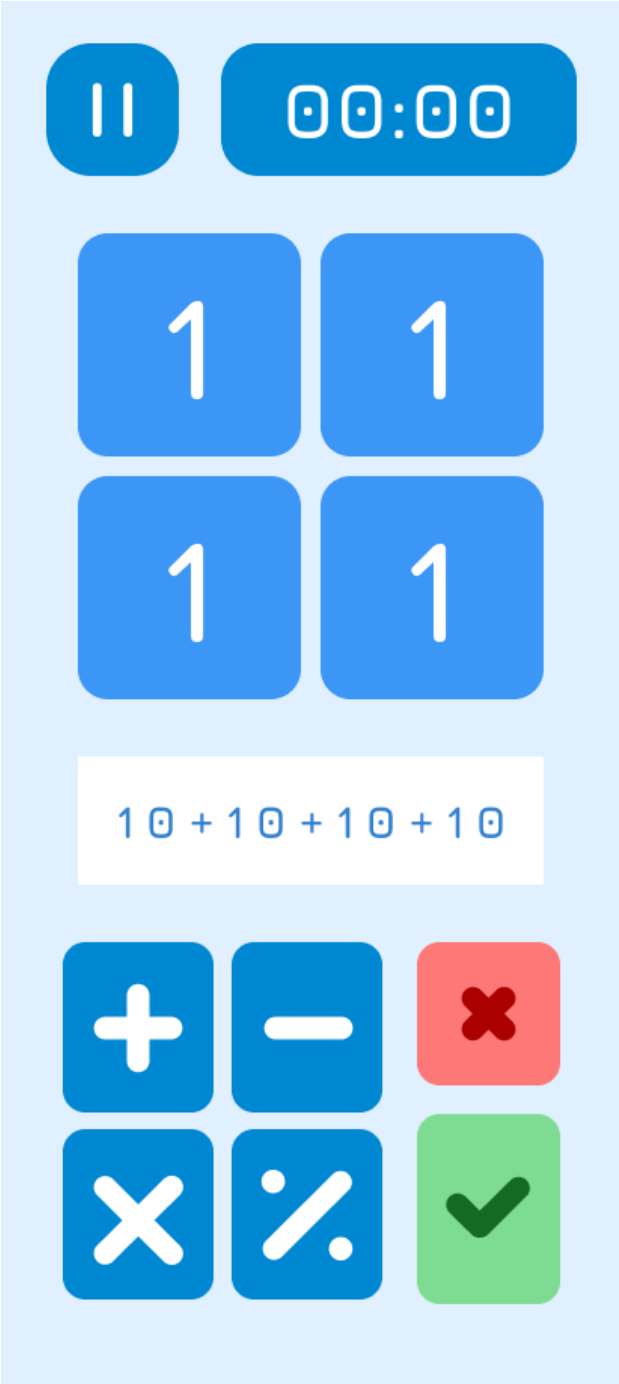
LOCKED & COMPLETED STAGES



HOW TO SCREEN

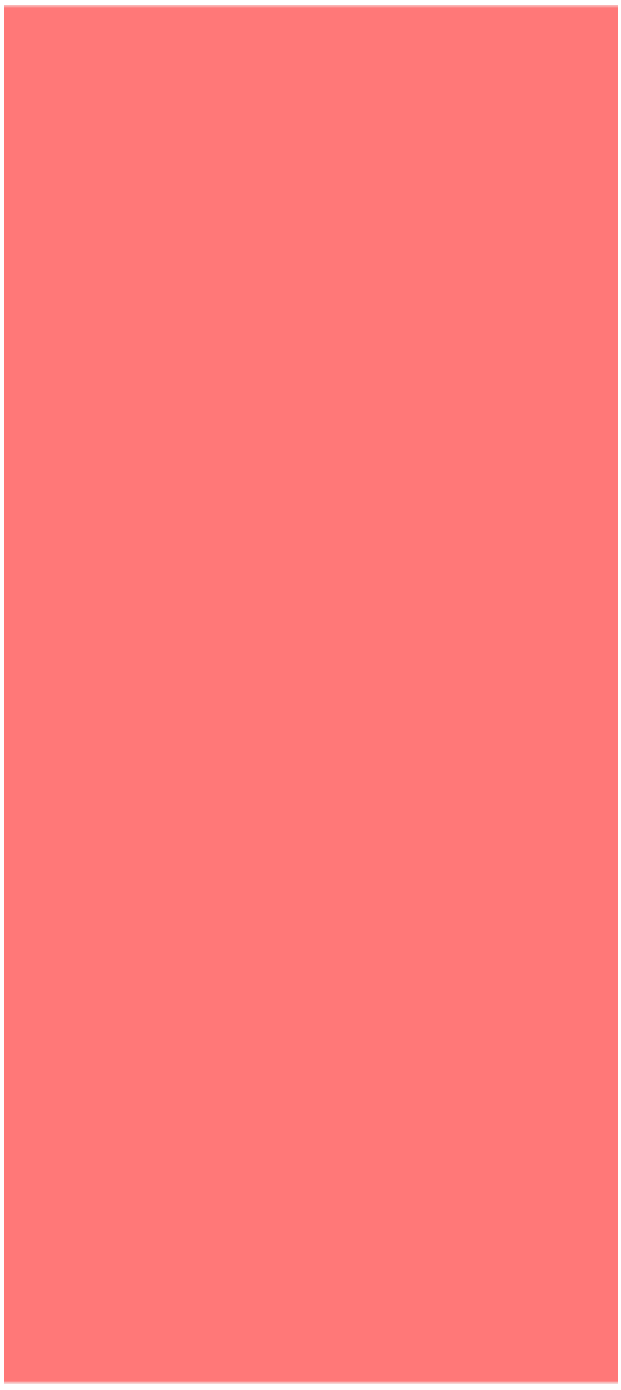


LEVEL SCREEN

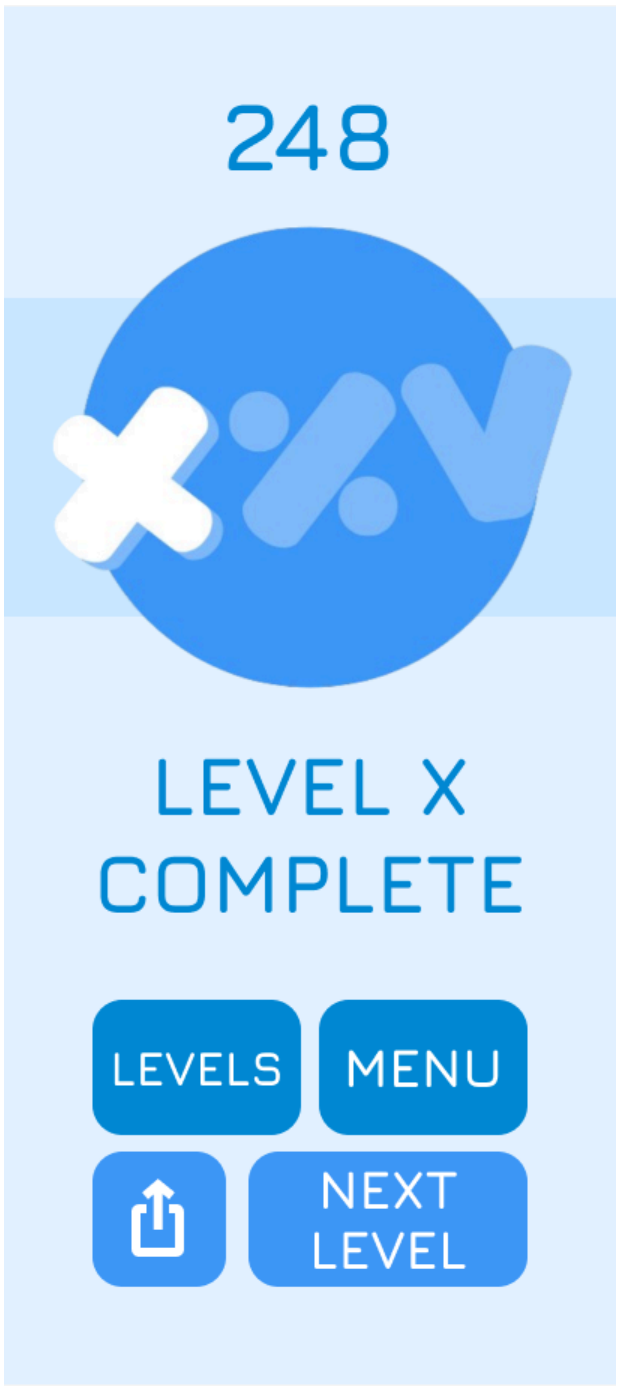


PAUSE SCREEN

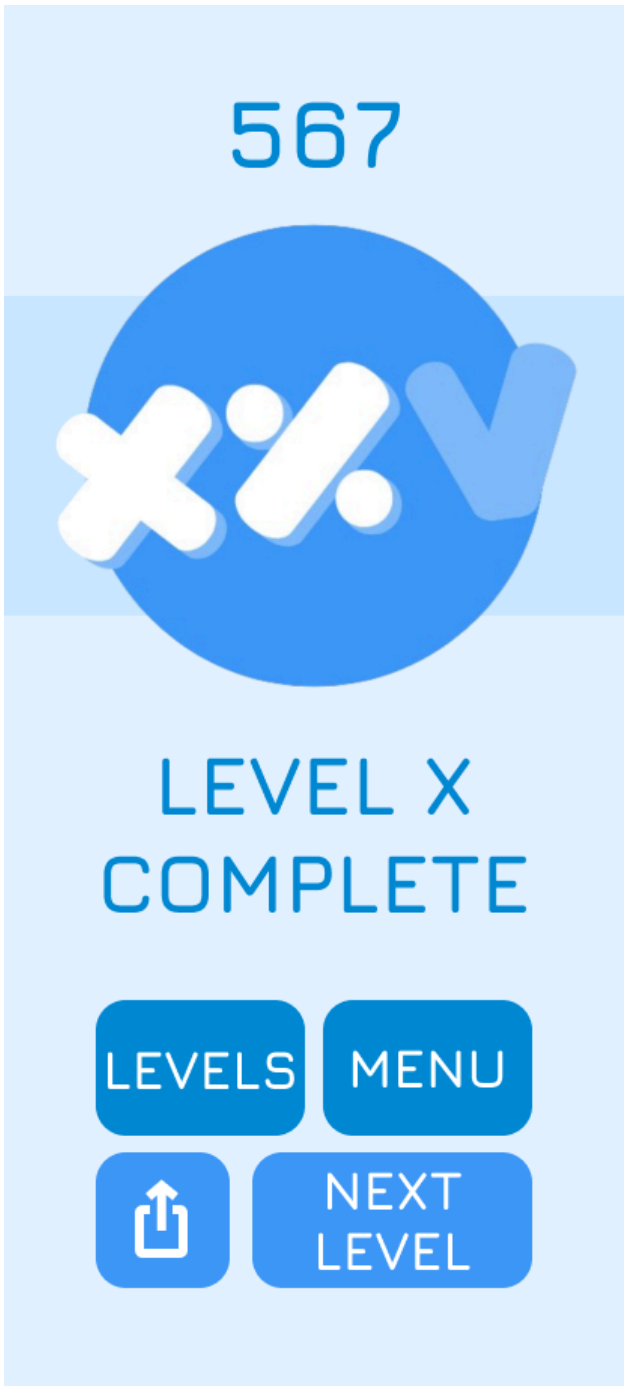
FAIL SCREEN



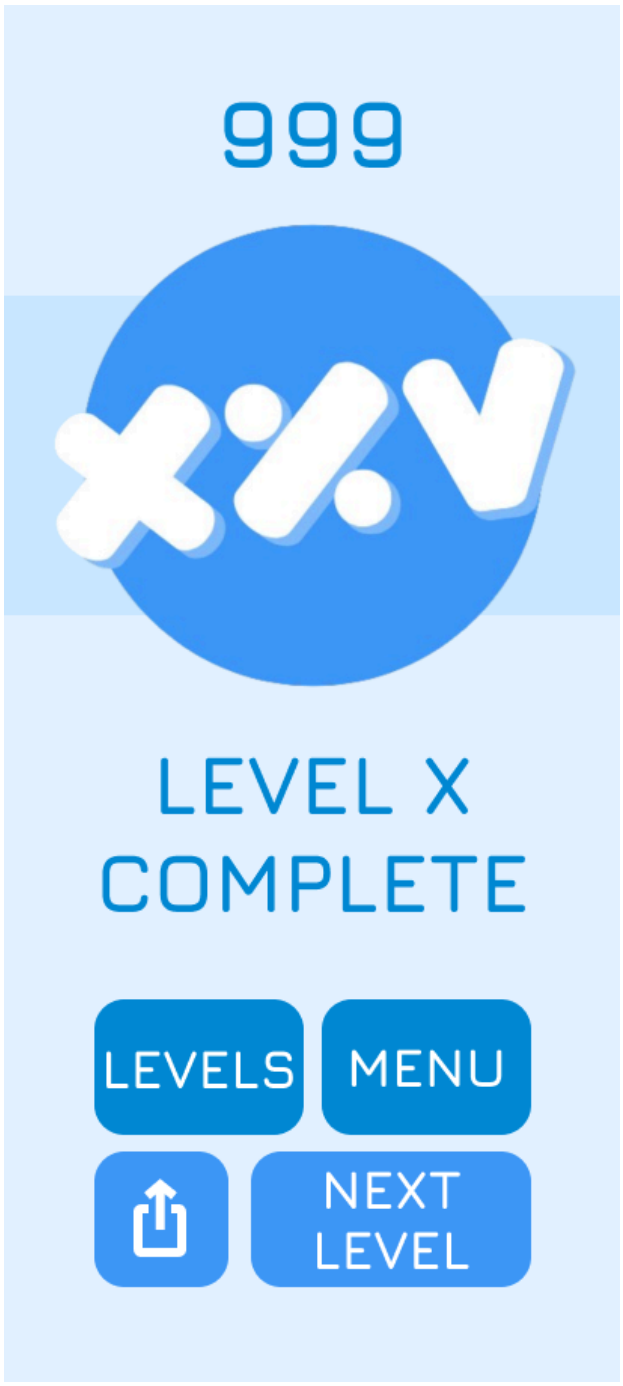
LEVEL CLEAR: 1 STAR



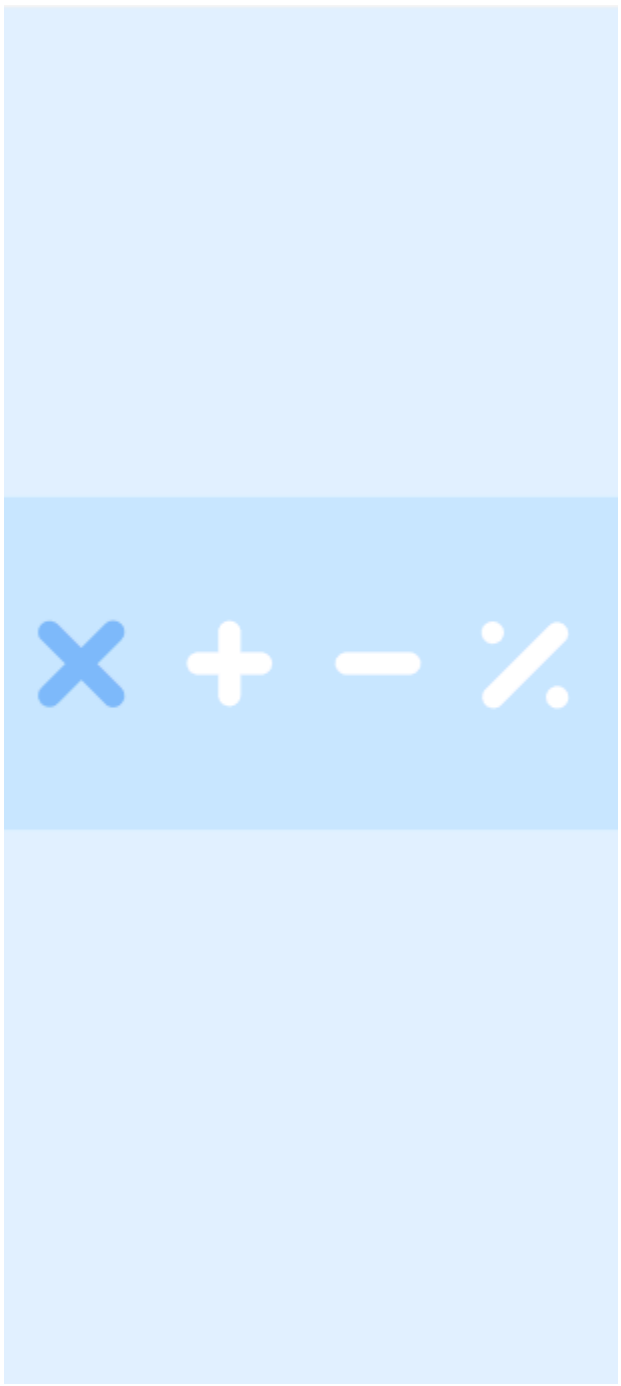
LEVEL CLEAR: 2 STAR



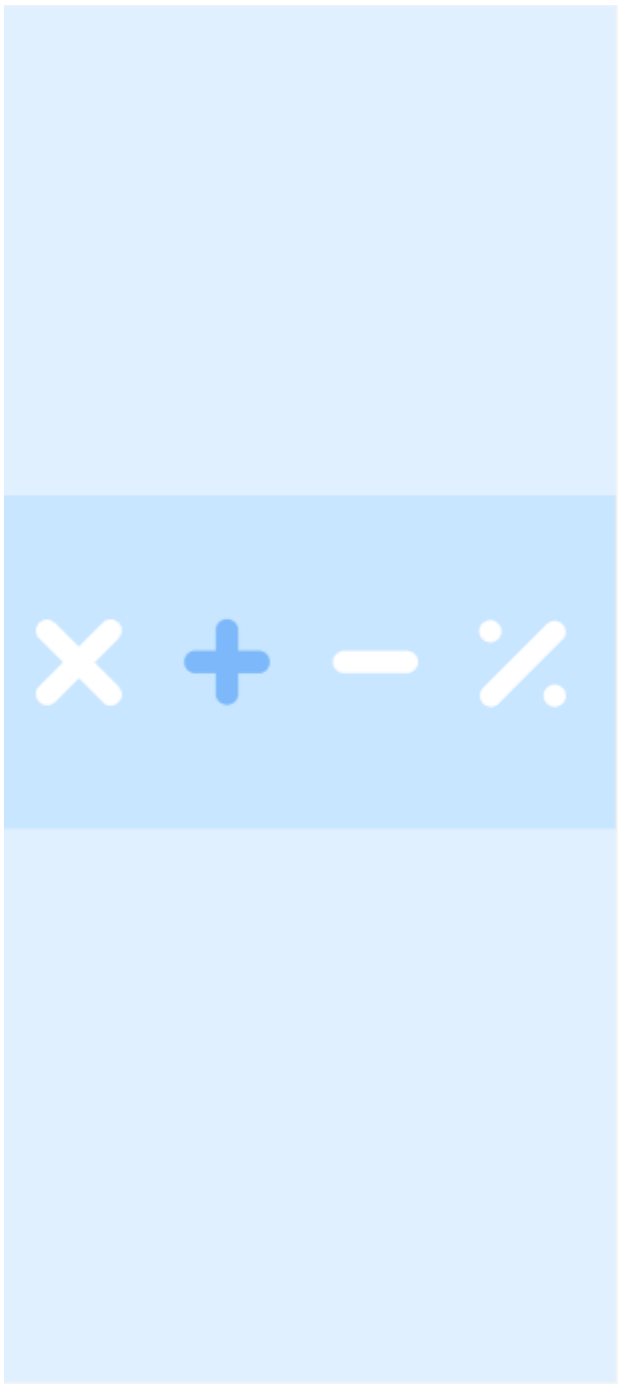
LEVEL CLEAR: 3 STAR



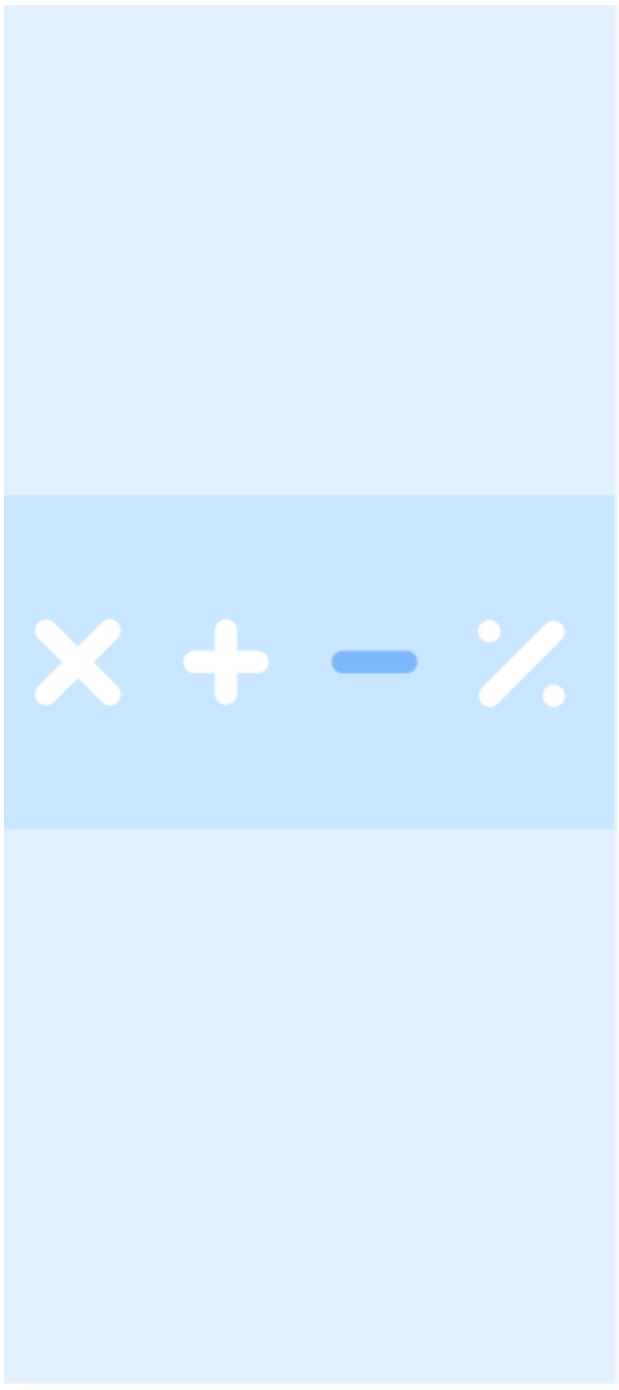
LOADING SCREEN 1



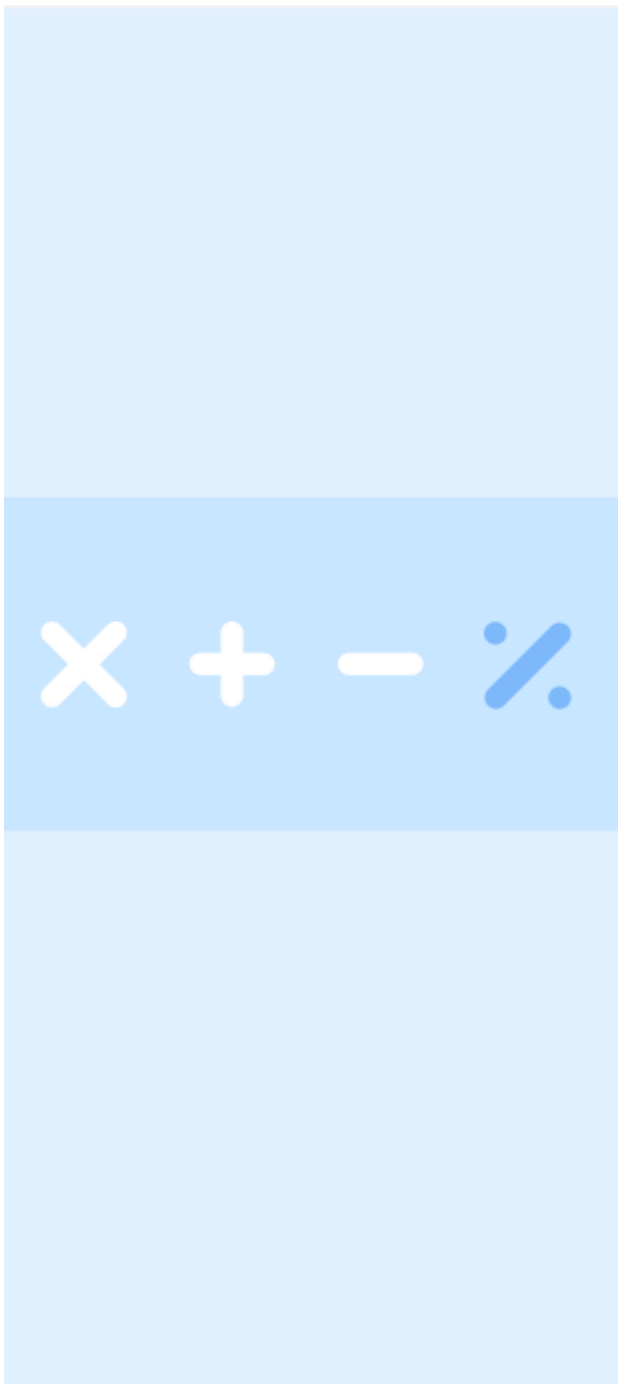
LOADING SCREEN 2



LOADING SCREEN 3



LOADING SCREEN 4



CODE

Programming Language: Java

Application Used: Visual Studio Code (Flutter & Dart Extension) - https://code.visualstudio.com

TITLE SCREEN

```
class HomeScreen extends StatelessWidget {
  const HomeScreen({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Stack(
        children: <Widget>[
          // Background image
          Container(
            decoration: const BoxDecoration(
              image: DecorationImage(
                image: AssetImage(assetName: 'assets/background-title.png'), // Replace with your image path
                fit: BoxFit.cover,
              ), // DecorationImage
            ), // BoxDecoration
          ), // Container
          // Blue rectangle
          Align(
            alignment: const Alignment(x: 0.0, y: 0.5),
            child: GestureDetector(
              onTap: () {
                Navigator.push<dynamic>(
                  context: context,
                  route: MaterialPageRoute<dynamic>(builder: (BuildContext context) => MenuScreen()),
                );
              },
              child: Container(
                width: 278,
                height: 176,
                decoration: BoxDecoration(
                borderRadius: BorderRadius.circular(radius: 40),
                color: const Color(value: 0xff3d97f9)), // BoxDecoration
                alignment: const Alignment(x: 0.0, y: 0.0), // Center text inside the box
                child: const Text(
                  data: 'TAP TO START', // Your text
                  textAlign: TextAlign.center,
                  style: TextStyle(
                      color: Colors.white, // Text color
                      fontSize: 50, // Font size
                      fontWeight: FontWeight.w900, // Bold text
                  ) // TextStyle
```

MENU

```
class MenuScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Stack(
        children: <Widget>[
          // Background image
          Container(
            decoration: const BoxDecoration(
              image: DecorationImage(
                image: AssetImage(assetName: 'assets/backgroundmenu.png'), // Replace with your image path
                fit: BoxFit.cover,
              ), // DecorationImage
            ), // BoxDecoration
          ), // Container
          // Blue rectangle
          Align(
            alignment: const Alignment(x: 0.0, y: 0.23),
            child: GestureDetector(
              onTap: () {
                Navigator.push<dynamic>(
                  context: context,
                  route: MaterialPageRoute<dynamic>(builder: (BuildContext context) => StagesScreen()),
                );
              },
              child: Container(
                width: 278,
                height: 108,
                decoration: BoxDecoration(
                borderRadius: BorderRadius.circular(radius: 40),
                color: const Color(value: 0xff3d97f9)), // BoxDecoration
                alignment: const Alignment(x: 0.0, y: 0.0), // Center text inside the box
                child: const Text(
                  data: 'LEVELS', // Your text
                  textAlign: TextAlign.center,
                  style: TextStyle(
                      color: Colors.white, // Text color
                      fontSize: 50, // Font size
                      fontWeight: FontWeight.bold, // Bold text
                  ) // TextStyle
                ) // Text
                  ), // Container
```

```
            ), // GestureDetector
          ), // Align
          Align(
            alignment: const Alignment(x: 0.0, y: 0.60),
            child: GestureDetector(
              onTap: () {
                Navigator.push<dynamic>(
                  context: context,
                  route: MaterialPageRoute<dynamic>(builder: (BuildContext context) => HowToScreen()),
                );
              },
              child: Container(
                width: 278,
                height: 108,
                decoration: BoxDecoration(
                borderRadius: BorderRadius.circular(radius: 40),
                color: const Color(value: 0xff3d97f9)), // BoxDecoration
                alignment: const Alignment(x: 0.0, y: 0.0), // Center text inside the box
                child: const Text(
                  data: 'HOW TO', // Your text
                  textAlign: TextAlign.center,
                  style: TextStyle(
                      color: Colors.white, // Text color
                      fontSize: 50, // Font size
                      fontWeight: FontWeight.bold, // Bold text
                  ) // TextStyle
                  ) // Text
                    ), // Container
              ), // GestureDetector
          ), // Align
        ],
      ), // Stack
    ); // Scaffold
  }
}
```

LEVELS CATEGORIES

```dart
class StagesScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Stack(
        children: <Widget>[
          // Background image
          Container(
            decoration: const BoxDecoration(
              image: DecorationImage(
                image: AssetImage(assetName: 'assets/backgroundnorm.png'), // Replace with your image path
                fit: BoxFit.cover,
              ), // DecorationImage
            ), // BoxDecoration
          ), // Container
          Positioned(
            top: 80, // Adjust as needed for padding.
            left: 40,
            child: GestureDetector(
              onTap: () {
                Navigator.push<dynamic>(
                context: context,
                route: MaterialPageRoute<dynamic>(builder: (BuildContext context) => MenuScreen()),
                );
              },
              child: Container(
                width: 65,
                height: 65,
                decoration: BoxDecoration(
                  color: const Color(value: 0xff3d97f9),
                  borderRadius: BorderRadius.circular(radius: 20),
                  boxShadow: const <BoxShadow>[
                  BoxShadow(
                    color: Colors.black26,
                    blurRadius: 10,
                    offset: Offset(dx: 0, dy: 5),
                  ), // BoxShadow
                ],
                ), // BoxDecoration
                padding: const EdgeInsets.all(value: 8),
                child: const Text(
                data: '<',
```

```
            style: TextStyle(
              color: Colors.white, // Text color.
              fontSize: 33,
              fontWeight: FontWeight.bold,
            ), // TextStyle
            textAlign: TextAlign.center,
          ), // Text
          ), // Container
        ), // GestureDetector
      ), // Positioned
    // Blue rectangle
    Align(
      alignment: const Alignment(x: 0.0, y: -0.55),
      child: GestureDetector(
        onTap: () {
          Navigator.push<dynamic>(
            context: context,
            route: MaterialPageRoute<dynamic>(builder: (BuildContext context) => Stage1()),
          );
        },
        child: Container(
          width: 278,
          height: 118,
          decoration: BoxDecoration(
            boxShadow: const <BoxShadow>[
              BoxShadow(
                color: Colors.black26,
                blurRadius: 10,
                offset: Offset(dx: 0, dy: 5),
              ), // BoxShadow
            ],
          borderRadius: BorderRadius.circular(radius: 25),
          color: const Color(value: 0xff3d97f9)), // BoxDecoration
          alignment: const Alignment(x: 0.0, y: 0.0), // Center text inside the box
          child: const Text(
            data: 'STAGE 1', // Your text
            textAlign: TextAlign.center,
            style: TextStyle(
              color: Colors.white, // Text color
              fontSize: 50, // Font size
```

```
                fontWeight: FontWeight.bold, // Bold text
              ) // TextStyle
              ), // Text
            ), // Container
          ), // GestureDetector
        ), // Align
        Align(
          alignment: const Alignment(x: 0.0, y: -0.15),
          child: GestureDetector(
            onTap: () {
              Navigator.push<dynamic>(
                context: context,
                route: MaterialPageRoute<dynamic>(builder: (BuildContext context) => Stage2()),
              );
            },
            child: Container(
              width: 278,
              height: 118,
              decoration: BoxDecoration(
                boxShadow: const <BoxShadow>[
                  BoxShadow(
                    color: Colors.black26,
                    blurRadius: 10,
                    offset: Offset(dx: 0, dy: 5),
                  ), // BoxShadow
                ],
              borderRadius: BorderRadius.circular(radius: 25),
              color: const Color(value: 0xff3d97f9)), // BoxDecoration
              alignment: const Alignment(x: 0.0, y: 0.0), // Center text inside the box
              child: const Text(
                data: 'STAGE 2', // Your text
                textAlign: TextAlign.center,
                style: TextStyle(
                    color: Colors.white, // Text color
                    fontSize: 50, // Font size
                    fontWeight: FontWeight.bold, // Bold text
                ) // TextStyle
                ), // Text
            ), // Container
          ), // GestureDetector
```

```
        ), // Positioned
      // Blue rectangle
      Align(
        alignment: const Alignment(x: 0.0, y: -0.55),
        child: GestureDetector(
          onTap: () {
            Navigator.push<dynamic>(
              context: context,
              route: MaterialPageRoute<dynamic>(builder: (BuildContext context) => Stage1()),
            );
          },
          child: Container(
            width: 278,
            height: 118,
            decoration: BoxDecoration(
              boxShadow: const <BoxShadow>[
                BoxShadow(
                  color: Colors.black26,
                  blurRadius: 10,
                  offset: Offset(dx: 0, dy: 5),
                ), // BoxShadow
              ],
            borderRadius: BorderRadius.circular(radius: 25),
            color: const Color(value: 0xff3d97f9)), // BoxDecoration
            alignment: const Alignment(x: 0.0, y: 0.0), // Center text inside the box
            child: const Text(
              data: 'STAGE 1', // Your text
              textAlign: TextAlign.center,
              style: TextStyle(
                  color: Colors.white, // Text color
                  fontSize: 50, // Font size
                  fontWeight: FontWeight.bold, // Bold text
            ) // TextStyle
              ), // Text
          ), // Container
        ), // GestureDetector
      ), // Align
      Align(
        alignment: const Alignment(x: 0.0, y: -0.15),
        child: GestureDetector(
```

```
            onTap: () {
              Navigator.push<dynamic>(
                context: context,
                route: MaterialPageRoute<dynamic>(builder: (BuildContext context) => Stage1()),
              );
            },
            child: Container(
              width: 278,
              height: 118,
              decoration: BoxDecoration(
                boxShadow: const <BoxShadow>[
                  BoxShadow(
                    color: Colors.black26,
                    blurRadius: 10,
                    offset: Offset(dx: 0, dy: 5),
                  ), // BoxShadow
                ],
              borderRadius: BorderRadius.circular(radius: 25),
              color: const Color(value: 0xff3d97f9)), // BoxDecoration
              alignment: const Alignment(x: 0.0, y: 0.0), // Center text inside the box
              child: const Text(
                data: 'STAGE 1', // Your text
                textAlign: TextAlign.center,
                style: TextStyle(
                  color: Colors.white, // Text color
                  fontSize: 50, // Font size
                  fontWeight: FontWeight.bold, // Bold text
                ) // TextStyle
                  ), // Text
              ), // Container
            ), // GestureDetector
          ), // Align
          Align(
            alignment: const Alignment(x: 0.0, y: -0.15),
            child: GestureDetector(
              onTap: () {
                Navigator.push<dynamic>(
                  context: context,
                  route: MaterialPageRoute<dynamic>(builder: (BuildContext context) => Stage2()),
                );
```

```
        },
      child: Container(
        width: 278,
        height: 118,
        decoration: BoxDecoration(
          boxShadow: const <BoxShadow>[
            BoxShadow(
              color: Colors.black26,
              blurRadius: 10,
              offset: Offset(dx: 0, dy: 5),
            ), // BoxShadow
          ],
        borderRadius: BorderRadius.circular(radius: 25),
        color: const Color(value: 0xff3d97f9)), // BoxDecoration
        alignment: const Alignment(x: 0.0, y: 0.0), // Center text inside the box
        child: const Text(
          data: 'STAGE 1', // Your text
          textAlign: TextAlign.center,
          style: TextStyle(
              color: Colors.white, // Text color
              fontSize: 50, // Font size
              fontWeight: FontWeight.bold, // Bold text
        ) // TextStyle
          ), // Text
        ), // Container
      ), // GestureDetector
    ), // Align
    Align(
      alignment: const Alignment(x: 0.0, y: -0.15),
      child: GestureDetector(
        onTap: () {
          Navigator.push<dynamic>(
            context: context,
            route: MaterialPageRoute<dynamic>(builder: (BuildContext context) => Stage2()),
          );
        },
      child: Container(
        width: 278,
        height: 118,
        decoration: BoxDecoration(
```

```
            boxShadow: const <BoxShadow>[
                BoxShadow(
                  color: Colors.black26,
                  blurRadius: 10,
                  offset: Offset(dx: 0, dy: 5),
                ), // BoxShadow
              ],
            borderRadius: BorderRadius.circular(radius: 25),
            color: const Color(value: 0xff3d97f9)), // BoxDecoration
            alignment: const Alignment(x: 0.0, y: 0.0), // Center text inside the box
            child: const Text(
              data: 'STAGE 1', // Your text
              textAlign: TextAlign.center,
              style: TextStyle(
                  color: Colors.white, // Text color
                  fontSize: 50, // Font size
                  fontWeight: FontWeight.bold, // Bold text
              ) // TextStyle
              ), // Text
            ), // Container
          ), // GestureDetector
        ), // Align
        Align(
          alignment: const Alignment(x: 0.0, y: -0.15),
          child: GestureDetector(
            onTap: () {
              Navigator.push<dynamic>(
                context: context,
                route: MaterialPageRoute<dynamic>(builder: (BuildContext context) => Stage2()),
              );
            },
          child: Container(
            width: 278,
            height: 118,
            decoration: BoxDecoration(
              boxShadow: const <BoxShadow>[
                  BoxShadow(
                    color: Colors.black26,
                    blurRadius: 10,
                    offset: Offset(dx: 0, dy: 5),
```

```
            ), // BoxShadow
          ],
        borderRadius: BorderRadius.circular(radius: 25),
        color: const Color(value: 0xff3d97f9)), // BoxDecoration
        alignment: const Alignment(x: 0.0, y: 0.0), // Center text inside the box
        child: const Text(
          data: 'STAGE 2', // Your text
          textAlign: TextAlign.center,
          style: TextStyle(
              color: Colors.white, // Text color
              fontSize: 50, // Font size
              fontWeight: FontWeight.bold, // Bold text
          ) // TextStyle
          ), // Text
        ), // Container
      ), // GestureDetector
    ), // Align
    Align(
      alignment: const Alignment(x: 0.0, y: 0.25),
      child: GestureDetector(
        onTap: () {
          Navigator.push<dynamic>(
            context: context,
            route: MaterialPageRoute<dynamic>(builder: (BuildContext context) => Stage3()),
          );
        },
      child: Container(
        width: 278,
        height: 120,
        decoration: BoxDecoration(
          boxShadow: const <BoxShadow>[
              BoxShadow(
                color: Colors.black26,
                blurRadius: 10,
                offset: Offset(dx: 0, dy: 5),
              ), // BoxShadow
          ],
        borderRadius: BorderRadius.circular(radius: 25),
        color: const Color(value: 0xff3d97f9)), // BoxDecoration
        alignment: const Alignment(x: 0.0, y: 0.0), // Center text inside the box
```

```
            child: const Text(
              data: 'STAGE 3', // Your text
              textAlign: TextAlign.center,
              style: TextStyle(
                  color: Colors.white, // Text color
                  fontSize: 50, // Font size
                  fontWeight: FontWeight.bold, // Bold text
              ) // TextStyle
                ), // Text
            ), // Container
          ), // GestureDetector
        ), // Align
        Align(
          alignment: const Alignment(x: 0.0, y: 0.67),
          child: GestureDetector(
            onTap: () {
              Navigator.push<dynamic>(
                context: context,
                route: MaterialPageRoute<dynamic>(builder: (BuildContext context) => Stage4()),
              );
            },
            child: Container(
              width: 278,
              height: 123,
              decoration: BoxDecoration(
                boxShadow: const <BoxShadow>[
                  BoxShadow(
                    color: Colors.black26,
                    blurRadius: 10,
                    offset: Offset(dx: 0, dy: 5),
                  ), // BoxShadow
                ],
              borderRadius: BorderRadius.circular(radius: 25),
              color: const Color(value: 0xff3d97f9)), // BoxDecoration
              alignment: const Alignment(x: 0.0, y: 0.0), // Center text inside the box
              child: const Text(
                data: 'STAGE 4', // Your text
                textAlign: TextAlign.center,
                style: TextStyle(
```

```
                    color: Colors.white, // Text color
                    fontSize: 50, // Font size
                    fontWeight: FontWeight.bold, // Bold text
                ) // TextStyle
                  ), // Text
              ), // Container
            ), // GestureDetector
          ), // Align
        ],
      ), // Stack
    ); // Scaffold
  }
}
```

LEVELS: STAGE 1

```dart
class Stage1 extends StatelessWidget {
  final List<String> levelTraits = <String>[
    '1_star', '2_stars', '3_stars', 'locked',
    '1_star', '2_stars', '3_stars', 'unlocked'
  ];

  @override
  Widget build(BuildContext context) {
    final Color themeColor = const Color(value: 0xff3d97f9); // Reusable theme color

    return Scaffold(
      body: Stack(
        children: <Widget>[
          // Background image
          Positioned.fill(
            child: Image.asset(
              name: 'assets/backgroundnorm.png',
              fit: BoxFit.cover,
            ), // Image.asset
          ), // Positioned.fill
          // Main content
          Padding(
            padding: const EdgeInsets.symmetric(horizontal: 16.0, vertical: 20.0),
            child: Column(
              children: <Widget>[
                // Back arrow and title
                Row(
                  mainAxisAlignment: MainAxisAlignment.spaceBetween,
                  children: <Widget>[
                    GestureDetector(
                      onTap: () {
                        Navigator.pop<Object?>(context: context); // Navigate back
                      },
                      child: Icon(
                        icon: Icons.arrow_back,
                        size: 30,
                        color: themeColor,
                      ), // Icon
                    ), // GestureDetector
                    Text(
                      data: 'STAGE 1',
                      style: TextStyle(
```

```
                fontSize: 35,
                fontWeight: FontWeight.bold,
                color: themeColor,
              ), // TextStyle
            ), // Text
            const SizedBox(width: 30), // Spacer for alignment
          ],
        ), // Row
        const SizedBox(height: 20),
        // Level buttons in rows
        Expanded(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
              // First row of buttons
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: <Widget>[
                  _buildLevelButton(level: 1, trait: levelTraits[0]),
                  _buildLevelButton(level: 2, trait: levelTraits[1]),
                ],
              ), // Row
              const SizedBox(height: 15),
              // Second row of buttons
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: <Widget>[
                  _buildLevelButton(level: 3, trait: levelTraits[2]),
                  _buildLevelButton(level: 4, trait: levelTraits[3]),
                ],
              ), // Row
              const SizedBox(height: 15),
              // Third row of buttons
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: <Widget>[
                  _buildLevelButton(level: 5, trait: levelTraits[4]),
                  _buildLevelButton(level: 6, trait: levelTraits[5]),
                ],
              ), // Row
```

```
                    const SizedBox(height: 15),
                    // Fourth row of buttons
                    Row(
                      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                      children: <Widget>[
                        _buildLevelButton(level: 7, trait: levelTraits[6]),
                        _buildLevelButton(level: 8, trait: levelTraits[7]),
                      ],
                    ), // Row
                  ],
                ), // Column
              ), // Expanded
            ],
          ), // Column
        ), // Padding
      ],
    ), // Stack
  ); // Scaffold
}

// Helper method to build a level button
Widget _buildLevelButton(int level, String trait) {
  String imagePath;

  // Determine overlay image based on the level's trait
  switch (trait) {
    case '1_star':
      imagePath = 'assets/1_star.png';
      break;
    case '2_stars':
      imagePath = 'assets/2_stars.png';
      break;
    case '3_stars':
      imagePath = 'assets/3_stars.png';
      break;
    case 'locked':
      imagePath = 'assets/locked.png';
      break;
    case 'unlocked':
    //default:
```

```
    //imagePath = null;
  }

  return GestureDetector(
    onTap: trait == 'locked'
        ? null // Disable interaction if locked
        : () {
            print(object: 'Level $level selected');
          },
    child: Stack(
      alignment: Alignment.center,
      children: <Widget>[
        // Background button
        Container(
          width: 135, // Fixed width for smaller size
          height: 135, // Fixed height for square shape
          decoration: BoxDecoration(
            color: const Color(value: 0xff3d97f9),
            borderRadius: BorderRadius.circular(radius: 12.0),
            border: Border.all(
              color: Colors.white.withOpacity(opacity: 0.8),
              width: 1.5,
            ), // Border.all
          ), // BoxDecoration
        ), // Container
        // Overlay image
        Positioned.fill(
          child: Opacity(
            opacity: 0.8, // Adjust transparency if needed
            child: Image.asset(
              name: imagePath,
              fit: BoxFit.cover,
            ), // Image.asset
          ), // Opacity
        ), // Positioned.fill
        // Text overlay
        Text(
          data: '$level',
          style: const TextStyle(
            fontSize: 20,
```

```
            fontWeight: FontWeight.bold,
            color: Colors.white,
            shadows: <Shadow>[
              Shadow(
                blurRadius: 2.0,
                color: Colors.black,
                offset: Offset(dx: 0.5, dy: 0.5),
              ), // Shadow
            ],
          ), // TextStyle
        ), // Text
      ],
    ), // Stack
  ); // GestureDetector
 }
}
```

LEVELS: STAGE 2

```
class Stage2 extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text(data: 'Stage 2')),
      body: const Center(
        child: Text(
          data: 'Welcome to the Stage 2!',
          style: TextStyle(fontSize: 24),
        ), // Text
      ), // Center
    ); // Scaffold
  }
}
```

LEVELS: STAGE 3

```dart
class Stage3 extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text(data: 'Stage 3')),
      body: const Center(
        child: Text(
          data: 'Welcome to the Stage 3!',
          style: TextStyle(fontSize: 24),
        ), // Text
      ), // Center
    ); // Scaffold
  }
}
```

LEVELS: STAGE 4

```dart
class Stage4 extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text(data: 'Stage 4')),
      body: const Center(
        child: Text(
          data: 'Welcome to the Stage 4!',
          style: TextStyle(fontSize: 24),
        ), // Text
      ), // Center
    ); // Scaffold
  }
}
```

LOCKED & COMPLETED STAGES

HOW TO SCREEN

```dart
class HowToScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        color: const Color(value: 0xffE3F3FF), // Set the background color to blue.
        child: Stack(
          children: <Widget>[
            // Back Button
            Positioned(
              top: 80, // Adjust as needed for padding.
              left: 40,
              child: GestureDetector(
                onTap: () {
                  Navigator.push<dynamic>(
                  context: context,
                  route: MaterialPageRoute<dynamic>(builder: (BuildContext context) => MenuScreen()),
                  );
                },
                child: Container(
                  width: 65,
                  height: 65,
                  decoration: BoxDecoration(
                    color: const Color(value: 0xff3d97f9),
                    borderRadius: BorderRadius.circular(radius: 20),
                    boxShadow: const <BoxShadow>[
                    BoxShadow(
                      color: Colors.black26,
                      blurRadius: 10,
                      offset: Offset(dx: 0, dy: 5),
                    ), // BoxShadow
                  ],
                  ), // BoxDecoration
                  padding: const EdgeInsets.all(value: 8),
                  child: const Text(
                  data: '<',
                  style: TextStyle(
                    color: Colors.white, // Text color.
                    fontSize: 33,
                    fontWeight: FontWeight.bold,
                  ), // TextStyle
                  textAlign: TextAlign.center,
```

```
          ), // Text
          ), // Container
        ), // GestureDetector
      ), // Positioned
      const Positioned(
        top: 70, // Position the title below the back button.
        left: 170,
        right: 0,
        child: Text(
          data: 'HOW TO',
          style: TextStyle(
            color:  Color(value: 0xff3d97f9),
            fontSize: 60,
            fontWeight: FontWeight.bold,
          ), // TextStyle
        ), // Text
      ), // Positioned
      // Centered Instructions Box
      Positioned (
        top: 180, // Distance from the top of the screen.
        left: 40, // Distance from the left of the screen.
        right: 40,
        child: Container(
          width: 275,
          height: 676, // Width of the rectangle.
          padding: const EdgeInsets.all(value: 20), // Padding inside the rectangle.
          decoration: BoxDecoration(
            color:  const Color(value: 0xff3d97f9),
            borderRadius: BorderRadius.circular(radius: 40),
            boxShadow: const <BoxShadow>[
              BoxShadow(
                color: Colors.black26,
                blurRadius: 10,
                offset: Offset(dx: 0, dy: 5),
              ), // BoxShadow
            ],
          ), // BoxDecoration
          child: const Text(
            data: 'Welcome to XXV! \n \nIn this educational math game, use addition, subtraction,

 multiplication and division to manipulate the numbers to equal 25. \nTry to complete the levels as fast

as you can to get the most points! \nShare your high scores with your friends on Instagram!',
```

```
                    style: TextStyle(
                      color: Colors.white, // Text color.
                      fontSize: 27,
                      fontWeight: FontWeight.bold,
                    ), // TextStyle
                    textAlign: TextAlign.center,
                  ), // Text
                ), // Container
              ), // Positioned
            ],
          ), // Stack
        ), // Container
      ); // Scaffold
    }
```

LEVEL SCREEN

```
class TimerScreen extends StatefulWidget {
 const TimerScreen({Key? key}) : super(key: key);
 //final double levelDecimal;
 @override
 _TimerScreenState createState() => _TimerScreenState();
}

class _TimerScreenState extends State<TimerScreen> {
 Timer? _timer;
 int _seconds = 0;
 bool _isPaused = false;
 String _textBoxContent = '';

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

 void _startTimer() {
   _timer?.cancel();
   _timer = Timer.periodic(const Duration(seconds: 1), (Timer timer) {
     setState(() {
       _seconds++;
     });
   });
```

```
}

String _formatTime(int seconds) {
  final int minutes = seconds ~/ 60;
  final int remainingSeconds = seconds % 60;
  return '${minutes.toString().padLeft(2, '0')}:${remainingSeconds.toString().padLeft(2,
'0')}';
}

void _updateTextBox(String content) {
  setState(() {
    _textBoxContent += content;
  });
}

bool _showRedOverlay = false; // To track when to show the red screen
bool _showGreenOverlay = false;

void _flashRed() {
  setState(() {
    _textBoxContent = ''; // Reset the text box
  });

  OverlayEntry overlay = OverlayEntry(
    builder: (context) {
      return Positioned.fill(
        child: Container(
          color: Colors.red,
        ),
      );
    },
  );

  Overlay.of(context).insert(overlay);

  Future.delayed(const Duration(milliseconds: 300), () {
    overlay.remove(); // Remove the red overlay
  });
}

bool isValidExpression(String input) {
```

```
  // List of allowed numbers
  final allowedNumbers = ['8', '2', '10', '5'];

  // Count occurrences of each allowed number in the input
  final numberCounts = <String, int>{};
  for (var number in allowedNumbers) {
    numberCounts[number] = RegExp('\\b$number\\b').allMatches(input).length;
  }

  // If any number appears more than once, the expression is invalid
  return numberCounts.values.every((count) => count == 1);
}

void _evaluateExpression() {
  if (!isValidExpression(_textBoxContent)) {
    // Show an error message if the expression is invalid
    print("Invalid input: Each number can only be used once.");
    _flashRed(); // Flash red to indicate invalid input
    return;
  }

  try {
    // Evaluate the expression
    final result = _calculateExpression(_textBoxContent);
    if (result == 25) {
      _timer?.cancel();

      if (_seconds < 30) {
        levelTraits1[0] = '3_stars';
      } else if (_seconds < 60) {
        levelTraits1[0] = '2_stars';
      } else {
        levelTraits1[0] = '1_star';
      }
      if (levelTraits1[1] == 'locked') {
        levelTraits1[1] = 'unlocked';
      }
      // Navigate to the new screen with the timer rvalue
      Navigator.push(
        context,
        MaterialPageRoute(
```

```
          builder: (context) => ResultScreen(
            timerValue: _seconds,
          ),
        ),
      );
    } else {
      _flashRed(); // Flash red if the result is not 25
    }
  } catch (e) {
    _flashRed(); // Flash red if there is an error in the expression
  }
}

double _calculateExpression(String expression) {
  try {
    // Replace custom symbols with standard math symbols
    final sanitizedExpression = expression
        .replaceAll('×', '*') // Replace multiplication symbol
        .replaceAll('÷', '/'); // Replace division symbol

    // Use the math_expressions library to parse and evaluate the expression
    Parser parser = Parser();
    Expression exp = parser.parse(sanitizedExpression);
    ContextModel contextModel = ContextModel();

    // Evaluate the expression and return the result
    return exp.evaluate(EvaluationType.REAL, contextModel);
  } catch (e) {
    throw Exception("Invalid Expression");
  }
}

@override
void dispose() {
  _timer?.cancel();
  super.dispose();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
```

```
    backgroundColor: const Color(0xFFE3F3FF),
    body: Stack(
      children: [
        Column(
          children: [
            // Timer and Pause Button
            Padding(
              padding: const EdgeInsets.only(
                  top: 65.0, left: 40.0, right: 2.0, bottom: 0.0),
              child: Row(
                mainAxisAlignment: MainAxisAlignment.start,
                children: [
                  // Pause Button
                  Container(
                    width: 80, // Circle width
                    height: 80, // Circle height
                    decoration: BoxDecoration(
                      color: const Color(0xFF0087D6), // Background color
                      borderRadius:
                          BorderRadius.circular(20), // Makes it circular
                    ),
                    child: IconButton(
                      onPressed: () {
                        setState(() {
                          if (!_isPaused) {
                            _timer?.cancel(); // Stop the timer
                            _isPaused = true; // Set pause state
                          }
                        });
                        // Navigate to the Pause screen
                        Navigator.push(
                          context,
                          MaterialPageRoute(
                            builder: (context) => PauseScreen(),
                          ),
                        ).then((value) {
                          // Resume the timer when coming back from the pause screen
                          setState(() {
                            _isPaused = false; // Reset the pause state
                            _timer = Timer.periodic(
                                const Duration(seconds: 1), (timer) {
```

```
                setState(() {
                  _seconds++;
                });
              });
            });
          });
      },
      icon: Icon(
        _isPaused
            ? Icons.play_arrow
            : Icons.pause, // Play or Pause icon
        color: Colors.white, // Icon color
        size: 60, // Icon size
      ),
    ),
  ),

  const SizedBox(
      width: 75), // Spacing between button and timer

  // Timer Display
  Container(
    width: 160, // Fixed width
    height: 78, // Fixed height
    alignment:
        Alignment.center, // Ensures the text stays centered
    decoration: BoxDecoration(
      color: const Color(0xFF0087D6), // Background color
      borderRadius:
          BorderRadius.circular(12), // Rounded rectangle
    ),
    child: Text(
      _formatTime(_seconds), // Timer text
      style: const TextStyle(
        fontSize: 50,
        fontWeight: FontWeight.bold,
        color: Colors.white, // Timer text color
      ),
      textAlign:
          TextAlign.center, // Centers text horizontally
    ),
```

```
          ),
        ],
      ),
    ),
    // 4 Large Buttons in 2x2 Grid
    Padding(
      padding: const EdgeInsets.symmetric(
          horizontal: 30.0, vertical: 20.0),
      child: Column(
        mainAxisSize: MainAxisSize
            .min, // Ensures the column does not take extra vertical space
        children: [
          Wrap(
            spacing: 7, // Horizontal spacing between buttons
            runSpacing: 7, // Vertical spacing between rows of buttons
            children: [8, 2, 10, 5].map((number) {
              return GestureDetector(
                onTap: () => _updateTextBox('$number'),
                child: Container(
                  width: 150, // Adjust size as needed
                  height: 150, // Ensure square buttons
                  decoration: BoxDecoration(
                    color: const Color(0xFF3D97F9),
                    borderRadius: BorderRadius.circular(16),
                  ),
                  child: Center(
                    child: Text(
                      '$number',
                      style: const TextStyle(
                        fontSize: 80,
                        fontWeight: FontWeight.bold,
                        color: Colors.white,
                      ),
                    ),
                  ),
                ),
              );
            }).toList(),
          ),
        ],
      ),
```

```
            ),
            // Fixed Size Text Box
            Center(
              child: Container(
                width: 310, // Fixed width
                height: 80, // Fixed height
                padding:
                    const EdgeInsets.symmetric(horizontal: 20, vertical: 0),
                decoration: BoxDecoration(
                  color: Colors.white,
                  borderRadius: BorderRadius.circular(5),
                  boxShadow: const [
                    BoxShadow(
                      color: Colors.black26,
                      blurRadius: 10,
                      offset: Offset(0, 5),
                    ),
                  ],
                ),
                child: Container(
                  alignment: Alignment
                      .center, // Centers the Text within the Container
                  padding: const EdgeInsets.only(bottom: 5),
                  child: Text(
                    _textBoxContent,
                    style: const TextStyle(
                      fontSize: 40,
                      fontWeight: FontWeight.bold,
                      color: Color(0xFF3583D6),
                    ),
                    textAlign: TextAlign.center,
                  ),
                ),
              ),
            ),
            // 6 Smaller Buttons with Icons
            Padding(
              padding: const EdgeInsets.only(
                  left: 20.0, right: 20, top: 30, bottom: 10),
              child: Wrap(
                spacing: 8, // Horizontal spacing between buttons
```

```
                runSpacing: 8, // Vertical spacing between rows of buttons
                children: [
                  {
                    'icon': Icons.add,
                    'symbol': '+',
                    'color': const Color(0xFF0087D6),
                    'iconColor': Colors.white
                  },
                  {
                    'icon': Icons.remove,
                    'symbol': '-',
                    'color': const Color(0xFF0087D6),
                    'iconColor': Colors.white
                  },
                  {
                    'icon': Icons.arrow_back,
                    'symbol': 'clear',
                    'color': const Color(0xFFFF7A7A),
                    'iconColor': const Color(0xFFAC0000)
                  },
                  {
                    'icon': Icons.close,
                    'symbol': '×',
                    'color': const Color(0xFF0087D6),
                    'iconColor': Colors.white
                  },
                  {
                    'icon': Icons.percent,
                    'symbol': '÷',
                    'color': const Color(0xFF0087D6),
                    'iconColor': Colors.white
                  },
                  {
                    'icon': Icons.check,
                    'symbol': null,
                    'color': const Color(0xFF81DF92),
                    'iconColor': const Color(0xFF186927)
                  },
                ].map<Widget>((buttonData) {
                  return GestureDetector(
                    onTap: () {
```

```
                    if (buttonData['symbol'] == 'clear') {
                      // Remove last input from the text box
                      setState(() {
                        if (_textBoxContent.isNotEmpty) {
                          // Check if the last two characters form a two-digit number
                          final regex = RegExp(
                              r'\d{2}$'); // Matches two digits at the end
                          if (regex.hasMatch(_textBoxContent)) {
                            // Remove the last two characters
                            _textBoxContent = _textBoxContent.substring(
                                0, _textBoxContent.length - 2);
                          } else {
                            // Remove only the last character
                            _textBoxContent = _textBoxContent.substring(
                                0, _textBoxContent.length - 1);
                          }
                        }
                      });
                    } else if (buttonData['symbol'] != null) {
                      setState(() {
                        // Validate the input
                        _textBoxContent += buttonData['symbol'] as String;
                      });
                    } else if (buttonData['icon'] == Icons.check) {
                      _evaluateExpression(); // Evaluate the expression
                    }
                  },
                  child: Container(
                    width: 110, // Adjust size for consistent button layout
                    height: 110, // Ensure square buttons
                    decoration: BoxDecoration(
                      color: buttonData['color'] as Color,
                      borderRadius: BorderRadius.circular(12),
                    ),
                    child: Icon(
                      buttonData['icon'] as IconData,
                      color: buttonData['iconColor']
                          as Color, // Dynamic icon color
                      size: 65, // Icon size
                    ),
                  ),
```

```
                  );
                }).toList(),
              ),
            ),
          ],
        ),
        // Red Overlay
        if (_showRedOverlay)
          Container(
            color: Colors.red,
          ),
        // Green Overlay
        if (_showGreenOverlay)
          Container(
            color: Colors.green,
          ),
      ],
    ),
  );
 }
}
```

PAUSE SCREEN

```
class PauseScreen extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
   return Scaffold(
     body: Stack(
       children: [
         // Background image
         Container(
           decoration: const BoxDecoration(
             image: DecorationImage(
               image: AssetImage(
                   'assets/backgroundnorm.png'), // Replace with your image path
               fit: BoxFit.cover,
             ),
           ),
         ),
```

```
        // Blue rectangle
        Align(
          alignment: const Alignment(0.0, -0.68),
          child: GestureDetector(
            onTap: () {
              Navigator.pop(context); // Close pause screen
            },
            child: Container(
              width: 278,
              height: 118,
              decoration: BoxDecoration(
                  boxShadow: const [
                    BoxShadow(
                      color: Colors.black26,
                      blurRadius: 10,
                      offset: Offset(0, 5),
                    ),
                  ],
                  borderRadius: BorderRadius.circular(25),
                  color: const Color(0xff3d97f9)),
              alignment:
                  const Alignment(0.0, 0.0), // Center text inside the box
              child: const Text('RESUME', // Your text
                  textAlign: TextAlign.center,
                  style: TextStyle(
                    color: Colors.white, // Text color
                    fontSize: 50, // Font size
                    fontWeight: FontWeight.bold, // Bold text
                  )),
            ),
          ),
        ),
        Align(
          alignment: const Alignment(0.0, -0.23),
          child: GestureDetector(
            onTap: () {
              Navigator.push(
                context,
                MaterialPageRoute(builder: (context) => HowToScreen()),
              );
            },
```

```
            child: Container(
              width: 278,
              height: 118,
              decoration: BoxDecoration(
                  boxShadow: const [
                    BoxShadow(
                      color: Colors.black26,
                      blurRadius: 10,
                      offset: Offset(0, 5),
                    ),
                  ],
                  borderRadius: BorderRadius.circular(25),
                  color: const Color(0xff3d97f9)),
              alignment:
                  const Alignment(0.0, 0.0), // Center text inside the box
              child: const Text('HOW TO', // Your text
                  textAlign: TextAlign.center,
                  style: TextStyle(
                    color: Colors.white, // Text color
                    fontSize: 50, // Font size
                    fontWeight: FontWeight.bold, // Bold text
                  )),
            ),
          ),
        ),
        Align(
          alignment: const Alignment(0.0, 0.23),
          child: GestureDetector(
            onTap: () {
              Navigator.push(
                context,
                MaterialPageRoute(builder: (context) => StagesScreen()),
              );
            },
            child: Container(
              width: 278,
              height: 120,
              decoration: BoxDecoration(
                  boxShadow: const [
                    BoxShadow(
                      color: Colors.black26,
```

```
                    blurRadius: 10,
                    offset: Offset(0, 5),
                  ),
                ],
                borderRadius: BorderRadius.circular(25),
                color: const Color(0xff3d97f9)),
            alignment:
                const Alignment(0.0, 0.0), // Center text inside the box
            child: const Text('LEVELS', // Your text
                textAlign: TextAlign.center,
                style: TextStyle(
                  color: Colors.white, // Text color
                  fontSize: 50, // Font size
                  fontWeight: FontWeight.bold, // Bold text
                )),
          ),
        ),
      ),
      Align(
        alignment: const Alignment(0.0, 0.71),
        child: GestureDetector(
          onTap: () {
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) => MenuScreen()),
            );
          },
          child: Container(
            width: 278,
            height: 123,
            decoration: BoxDecoration(
                boxShadow: const [
                  BoxShadow(
                    color: Colors.black26,
                    blurRadius: 10,
                    offset: Offset(0, 5),
                  ),
                ],
                borderRadius: BorderRadius.circular(25),
                color: const Color(0xff3d97f9)),
            alignment:
```

```
                    const Alignment(0.0, 0.0), // Center text inside the box
                child: const Text('QUIT', // Your text
                    textAlign: TextAlign.center,
                    style: TextStyle(
                      color: Colors.white, // Text color
                      fontSize: 50, // Font size
                      fontWeight: FontWeight.bold, // Bold text
                    )),
              ),
            ),
          ),
        ],
      ),
    );
  }
}
```

RESULTS SCREEN

```
class ResultScreen extends StatelessWidget {
 final int timerValue;
```

```
 const ResultScreen({Key? key, required this.timerValue}) : super(key: key);
```

```
 @override
 Widget build(BuildContext context) {
   // Calculate the result
   final double calculation = 999 - 11.1 * timerValue;
   final int roundedResult = max(calculation.toInt(), 0); // Convert to integer
```

```
   // Choose background image based on timer value
   String backgroundImage;
   if (timerValue < 30) {
     backgroundImage = 'assets/clear3stars.png';
   } else if (timerValue < 60) {
     backgroundImage = 'assets/clear2stars.png';
   } else {
     backgroundImage = 'assets/clear1star.png';
```

```
  }
```

```
  return Scaffold(
    body: Stack(
      children: [
        // Background image
        Container(
          decoration: BoxDecoration(
            image: DecorationImage(
              image: AssetImage(backgroundImage),
              fit: BoxFit.cover,
            ),
          ),
        ),
        // Text box displaying the calculated result
        Positioned(
          top: 60, // Adjust as needed for placement
          left: 20,
          right: 20,
          child: Container(
            height: 70, // Height of the text box
            padding: const EdgeInsets.symmetric(horizontal: 16),
            child: Center(
              child: Text(
                '$roundedResult',
                style: const TextStyle(
                  fontSize: 60,
                  fontWeight: FontWeight.bold,
                  color: Color(0xFF0087D6),
                ),
              ),
            ),
          ),
        ),
        Padding(
          padding: const EdgeInsets.only(top: 200.0),
          child: Container(
            alignment: Alignment.center,
            padding:
                const EdgeInsets.symmetric(horizontal: 50.0, vertical: 10.0),
```

```
            child: const Text('LEVEL 1.1 COMPLETE',
                style: TextStyle(
                  fontSize: 60,
                  height: 1.25,
                  fontWeight: FontWeight.bold,
                  color: Color(0xFF0087D6),
                ),
                textAlign: TextAlign.center),
          ),
        ),
        Padding(
          padding: const EdgeInsets.only(top: 640.0, left: 45.0, right: 20.0),
          child: Wrap(
            spacing: 10.0, // Horizontal spacing between buttons
            runSpacing: 15.0, // Vertical spacing between rows
            alignment: WrapAlignment.center, // Aligns buttons to the center
            children: [
              // Levels Button
              GestureDetector(
                onTap: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(builder: (context) => StagesScreen()),
                  );
                },
                child: Container(
                  width: 150, // Button widt
                  height: 80, // Button height
                  decoration: BoxDecoration(
                    color: const Color(0xFF0087D6), // Blue color
                    borderRadius: BorderRadius.circular(12),
                  ),
                  child: const Center(
                    child: Text(
                      'LEVELS',
                      style: TextStyle(
                        fontSize: 40,
                        fontWeight: FontWeight.bold,
                        color: Colors.white,
                      ),
                      textAlign: TextAlign.center,
```

```
          ),
        ),
      ),
    ),
    // Menu Button
    GestureDetector(
      onTap: () {
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => MenuScreen()),
        );
      },
      child: Container(
        width: 150,
        height: 80,
        decoration: BoxDecoration(
          color: Color(0xFF0087D6),
          borderRadius: BorderRadius.circular(12),
        ),
        child: const Center(
          child: Text(
            'MENU',
            style: TextStyle(
              fontSize: 40,
              fontWeight: FontWeight.bold,
              color: Colors.white,
            ),
            textAlign: TextAlign.center,
          ),
        ),
      ),
    ),
    // Replay Button
    GestureDetector(
      onTap: () async {
        final String instagramURL = "https://www.instagram.com/";
```

```
        // Try to open Instagram
        try {
          final Uri uri = Uri.parse(instagramURL);
```

```
                if (await canLaunchUrl(uri)) {
                  await launchUrl(uri,
                      mode: LaunchMode.externalApplication);
                } else {
                  // Handle the case where Instagram is not available
                  print("Instagram is not installed or URL is invalid.");
                  ScaffoldMessenger.of(context).showSnackBar(
                    const SnackBar(
                        content: Text("Instagram app is not available!")),
                  );
                }
              } catch (e) {
                print("Error launching Instagram: $e");
                ScaffoldMessenger.of(context).showSnackBar(
                  const SnackBar(
                      content: Text("Unable to open Instagram!")),
                );
              }
            },
            child: Container(
              width: 60,
              height: 80,
              decoration: BoxDecoration(
                color: Color(0xFF3D97F9),
                borderRadius: BorderRadius.circular(12),
              ),
              child: const Center(
                child: Icon(
                  Icons.share,
                  color: Colors.white,
                  size: 45,
                ),
              ),
            ),
          ),
        ),
        // Next Level Button
        GestureDetector(
          onTap: () {
            Navigator.push(
```

```
                    context,
                    MaterialPageRoute(builder: (context) => TimerScreen2()),
                  );
                },
                child: Container(
                  width: 240,
                  height: 80,
                  decoration: BoxDecoration(
                    color: Color(0xFF3D97F9),
                    borderRadius: BorderRadius.circular(12),
                  ),
                  child: const Center(
                    child: Text(
                      'NEXT LEVEL',
                      style: TextStyle(
                        fontSize: 40,
                        fontWeight: FontWeight.bold,
                        color: Colors.white,
                      ),
                      textAlign: TextAlign.center,
                    ),
                  ),
                ),
              ),
            ],
          ),
        ),
      ],
    ),
  );
 }
}
```

APPLICATION

- Mobile Application can be downloaded and run across all Android devices