

Phase 0

Name of Team: Mind Optimizers				
First name: Lu	Last name: Shen	York Email: shen6644@my.yorku.ca	Lecture Section: N	Lab Section: 3
First name: Menghua	Last name: Jiang	York Email: Monna521@my.yorku.ca	Lecture Section: M	Lab Section: 4
First name: Yiqing	Last name: Wu	York Email: yiqing98@my.yorku.ca	Lecture Section: N	Lab Section: 3
First name: Wanqin	Last name: Jiang	York Email: jwqlily@my.yorku.ca	Lecture Section: N	Lab Section: 4
Project Title: Memorization Game				
Project Description: For this project, our goal is to build a game app to improve the user's memorization ability. Our app will be able to store various colors and shapes, and as the game starts, the system will randomly show one or more colored shapes. The user needs to memorize the color and shape and later select the correct one from the given options. We'll include timing as one of our features since it's directly related to the user's final result. As the user progresses, the difficulty level of the game will gradually increase. The other feature would be the recording of the highest score achieved.				

Phase 1

Functional Requirements

Function 1:

User registration

On the welcome screen, there is a registration button that the user can click to create an account first to begin the game.

Function 2:

User login

Before running the game, the user is able to click on the button on the welcome page to login to her/his account on the welcome screen first.

Function 3:

Start game

After login, the new screen will display a "start game" button, and once being clicked, the program will automatically start the game.

Function 4:

Timing

Once the game starts, users get limited time to memorize the shapes and when the time is up, a timing clock will appear on the next new screen, which will automatically start recording the users' solving time on each game.

Function 5:

Display colored shapes

The program is able to store various colored shapes that need to be memorized as the game starts, it'll generate random colored shapes automatically.

Function 6:

Displaying all options

Once the time for memorization is up, the original colored shapes disappear and all options automatically appear on the screen, then the user needs to choose the correct ones in correct order.

Function 7:

Return the user performance and the points received

The program will calculate the user's score and generate a point to be returned to the user. For instance, if above 80%, return "great job!".

Function 8:

Increase of difficulty levels

The difficulty levels will automatically increase according to the points acquired by users. The difficulty level may increase by the number of shapes, colors, or less time for memorization.

Function 9:

Suspend game

The user is able to stop the game by pressing the "Pause" button at the top, if necessary and the program will save the current score at the back storage.

Function 10:

View high scores

Users can view their highest score by clicking the "show record" button on the home page.

Function 11:

End and save game

Users can end the game by clicking on the "End game" button and the app will automatically save all the related data.

Function 12:

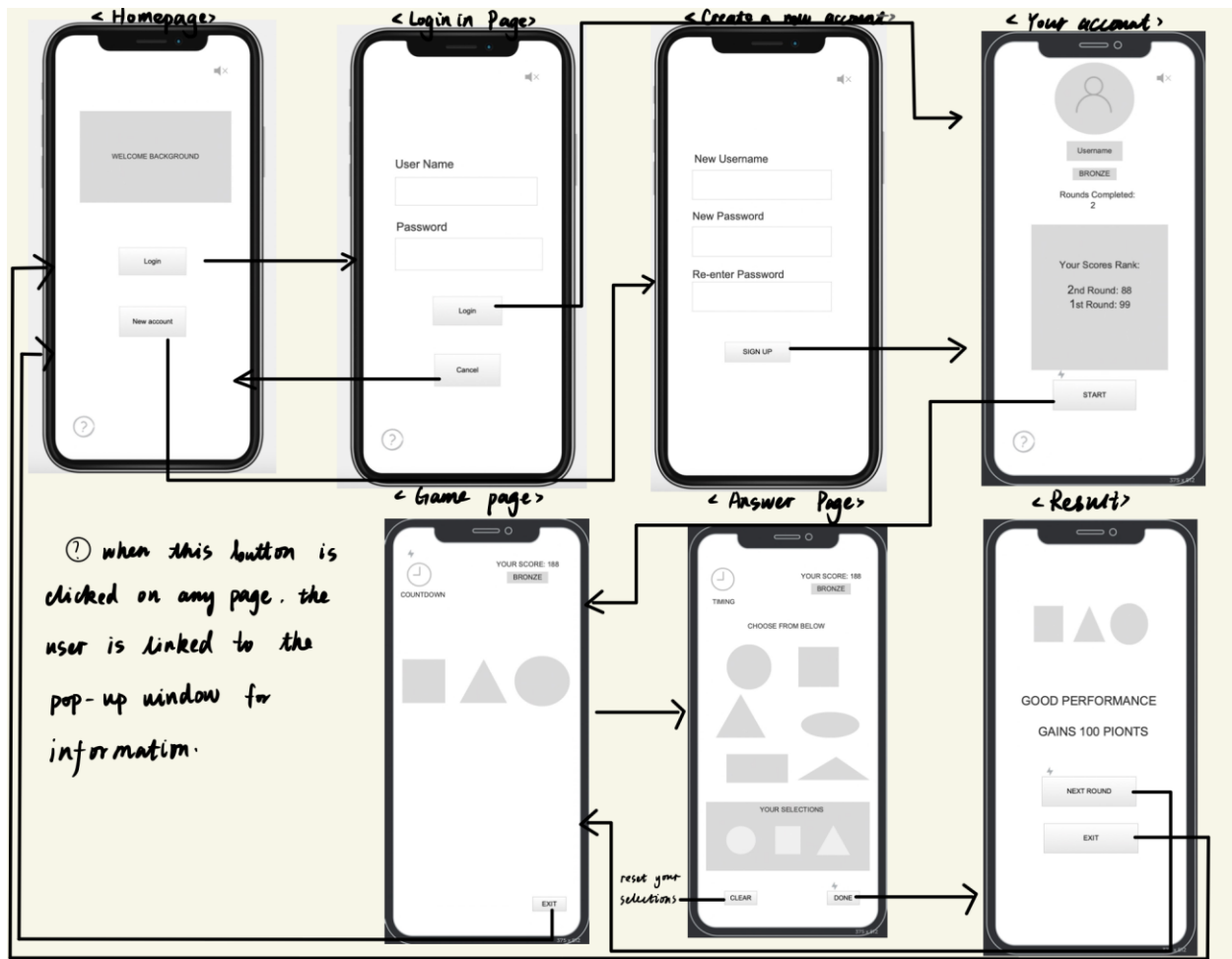
Resume saved game

Users can resume the saved game by tapping the resume button once re-entering the game.

Phase 2

Design

<https://github.com/users/jwqlily/projects/1>



Phase 3

1. Model

1.1 UerAccount Class

1.1.1 Instance Variable

- a. userName
- b. passWord
- c. points (an arrayList of points of each round)

1.1.2 Methods

1.1.2.1 Constructor

1.1.2.2 getTotalPoints

- Return the total points of all completed rounds
- Technology used: ArrayList, Iterator

1.1.2.3 getRankHistory

- Return the history of points gained in each round
- Technology used: ArrayList, Iterator

1.1.2.4 getCompletedRounds

- Return the total rounds completed

1.1.2.5 updatePoints

- Add the point into the List point after a round

1.1.2.6 getUsername

- Getter method for username

1.1.2.7 getPassWord

- Getter method for password

1.1.2.8 setUsername

- Setter method for username

1.1.2.9 setPassword

- Setter method for password

1.2 Game Class

1.2.1 Instance Variable

- a. shapes (an array of all the colored shapes in the game, R.drawable is used for each element)
- b. answer (an array of index of the shapes that need to be memorised)

1.2.2 Methods

1.2.2.1 getAnswer

- Create a random list of index for the shapes that need to be memorised in the game
- Technology used: HashSet, Random, Array

1.2.2.2 getOptions

- Create a random list of index for the shapes as options that are shown in the Answer Page
- Technology used: HashSet, shuffle of Collections, Random, Array

1.3 FileIO Class

1.3.1 Instance Variable

- a. filelines (an ArrayList of String, each element represents one line of txt file that contains user information)
- b. users (an ArrayList of UerAccount objects representing the users in the txt file)
- c. InputStream is (This is for reading the txt file in the resource)

1.3.2 Methods

1.3.2.1 Constructor

1.3.2.2 createFileStr

- Read the users info txt file and update the instance variable filelines
- Technology used: BufferedReader, InputStream, FileNotFoundException, ArrayList

1.3.2.3 createUsers

- Manipulate the strings of file content, extract user information and store the user information in the instance variable users

- Technology used: Array, String method, ArrayList, try...catch, NumberFormatException

1.3.2.4 getUserIndex

- Given a user name, return the corresponding index of UerAcount object in the ArrayList users. If the user name is not found, return -1
- Technology used: ArrayList

1.3.2.5 getUser

- Given a user name, return the corresponding UerAcount object in the ArrayList users. If the user name is not found, throw an exception
- Technology used: ArrayList, illegalArgumentException

1.3.2.6 match

- Check if the given username and password match the record in the user info txt file
- Technology used: ArrayList, String method

1.4 MusicService Class

1.4.1 Instance Variable

- a. isplay (a Boolean variable representing whether the music is played)
- b. player (a MediaPlayer Object)

1.4.2 Methods

1.4.2.1 Constructor

1.4.2.2 onCreate

- Override the onCreate method and create a MediaPlayer object
- Technology used: MediaPlayer

1.4.2.3 onStartCommand

- Override the onStartCommand method and start playing the music
- Technology used: MediaPlayer

1.4.2.4 onDestroy

- Override the onDestroy method and stop playing the music
- Technology used: MediaPlayer

2. Control

2.1 MainActivity Class

- Show the dialog box when clicking on the image button question mark
- Turn to login page when clicking on the login button
- Turn to create account page when clicking on the create account button
- Turn on/off the music, adjust the volume
- Technology used:
- Button
- Image button
- Intent

2.2 LoginActivity Class

- Turn to home page when clicking on the cancel button
- Show the dialog box when clicking on the image button question mark
- Turn on/off the music, adjust the volume
- Technology used:
- Button
- Image button
- Intent

2.3 CreateAccountActivity Class

- Get the user input from the textbox
- Give notification message when user gives the wrong input
- Store the user information and turn to account page when clicking on the sign up button
- Turn to home page when clicking on the cancel button
- Show the dialog box when clicking on the image button question mark
- Turn on/off the music, adjust the volume
- Technology used:
- Button
- Image button
- Intent

2.4 LoginActivity Class

- Get the user input from the textbox
- Read the txt file of user info and check the input username and password when clicking on the login button

- Give notification message when the input username and password doesn't match
- If the input username and password match, store the user information and turn to account page
- Turn to home page when clicking on the cancel button
- Show the dialog box when clicking on the image button question mark
- Turn on/off the music, adjust the volume
- Technology used:
- openRawResources
- Intent
- Button

2.5 AccountActivity Class

- Make the Labels show the user information such as rounds completed, rank history, username, and so on
- Turn to game page when clicking on the start button
- Show the dialog box when clicking on the image button question mark
- Turn on/off the music, adjust the volume
- Technology used:
- Button
- Image button
- Intent
- TextView

2.6 GameActivity Class

- Make the Textview show the user's current points
- Call getAnswer methods of Game Class and make the Imageview show the randomly generated colored shape
- Set the countdown timing and when time is up, turn to the answer page
- Turn to account page when clicking on the exit button
- Show the dialog box when clicking on the image button question mark
- Turn on/off the music, adjust the volume
- Technology used:
- Button
- Image button
- Intent
- TextView
- Chronometer
- ImageView

2.7 AnswerActivity Class

- Call getOption methods of Game Class and make the Image button show the randomly generated colored shape as options
- Set the ImageView in the answer section as exactly the colored shapes that the user has clicked on
- When the user has clicked on more than 3 shapes, show the notification message
- Clear the ImageView in the answer section when clear button is clicked
- When the done button is clicked on, save the user answer and turn to result page
- Set the countup timing and record how long it takes for the user to answer
- Turn to account page when clicking on the exit button
- Make the Textview show the user's current points
- Turn on/off the music, adjust the volume
- Technology used:
 - Button
 - Image button
 - Intent
 - TextView
 - Chronometer
 - ImageView

2.8 ResultActivity Class

- Compare the user selections and the correct answer and after comparison, show corresponding message about performance in the pointsGain Label and update the instance variable points for this user object
- Make the ImageView show the correct colored shapes
- Turn to account page when clicking on the exit button
- Turn to game page when clicking on the next round button, if the points user has obtained reach a 100, the difficulty level up.
- Turn on/off the music, adjust the volume
- Technology used:
 - Button
 - Intent
 - TextView
 - ImageView

2.9 GameLv2Activity Class

- It is similar to GameActivity Class but intended for higher level of game difficulty
- 6 colored shapes is used for user memorization instead of 3 colored shapes in game difficulty 1

2.10 AnswerLv2Activity Class

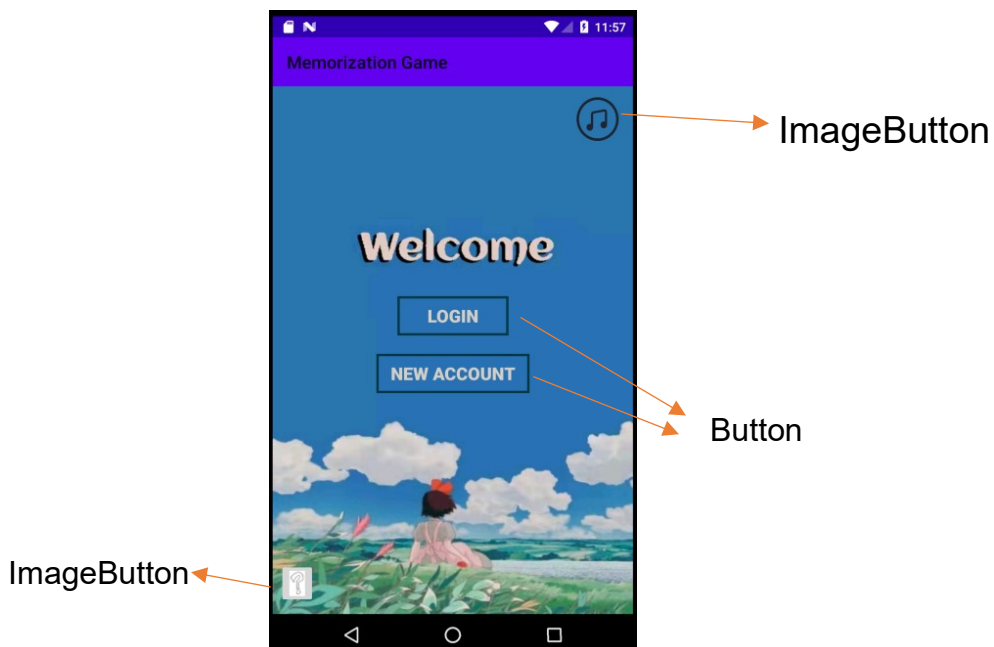
- It is similar to AnswerActivity Class but intended for higher level of game difficulty
- 12 colored shapes is used for options instead of 6 colored shapes in game difficulty 1 , and user need to choose 6 correct colored shapes

2.11 ResultLv2Activity Class

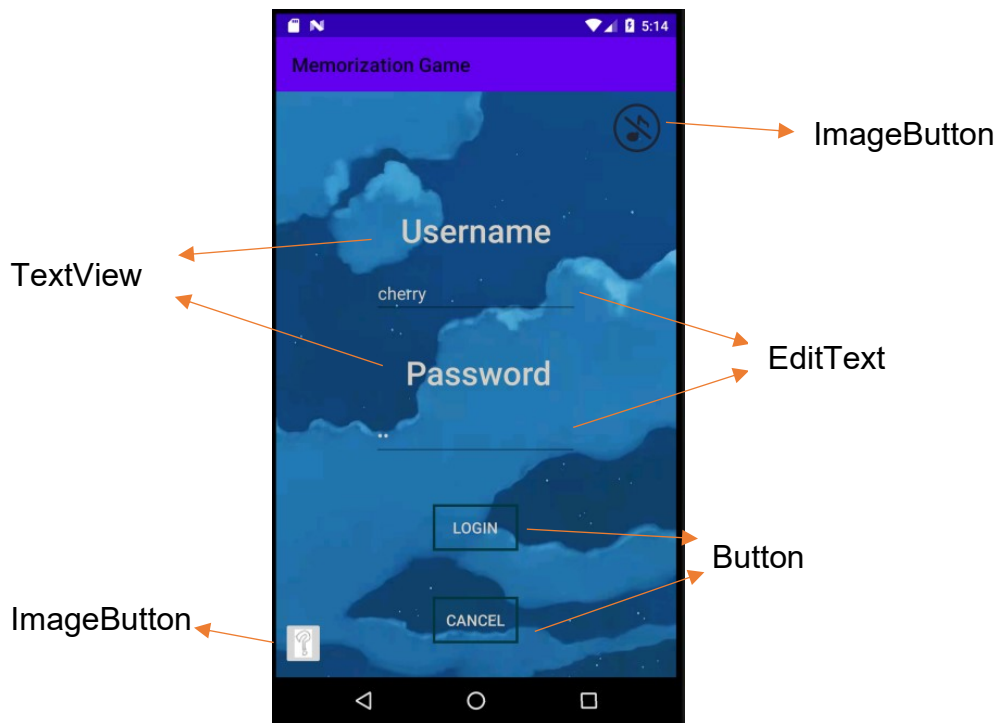
- It is similar to ResultActivity Class but intended for higher level of game difficulty
- 6 correct colored shapes is shown instead of 3 colored shapes in difficulty level 1

3. View

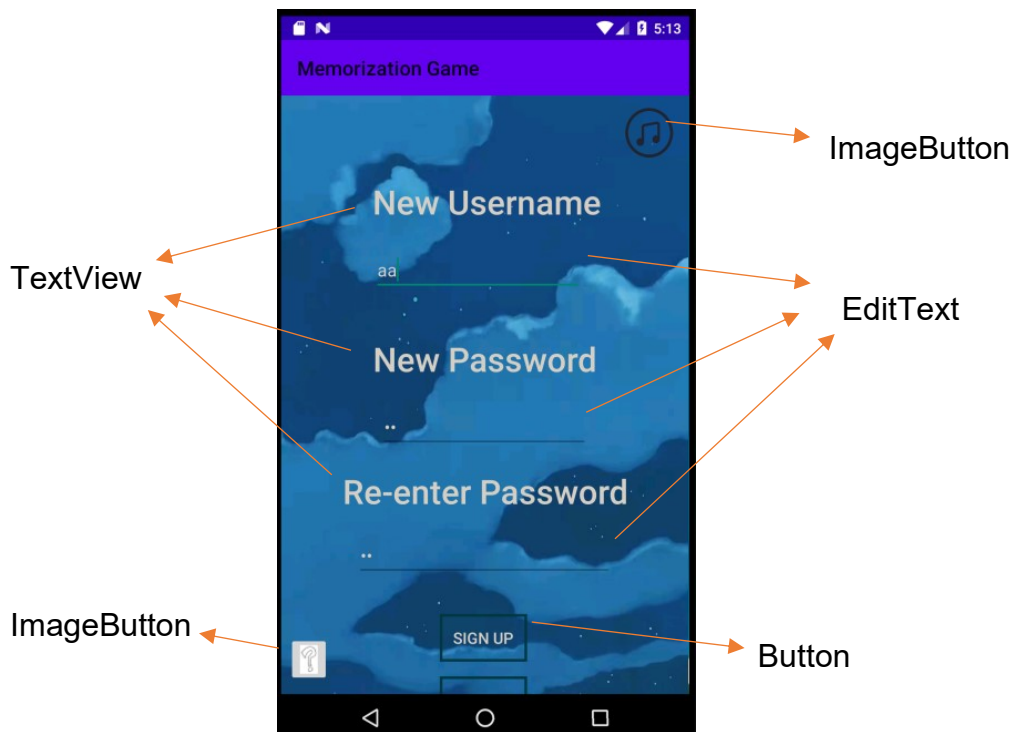
3.1 Homepage



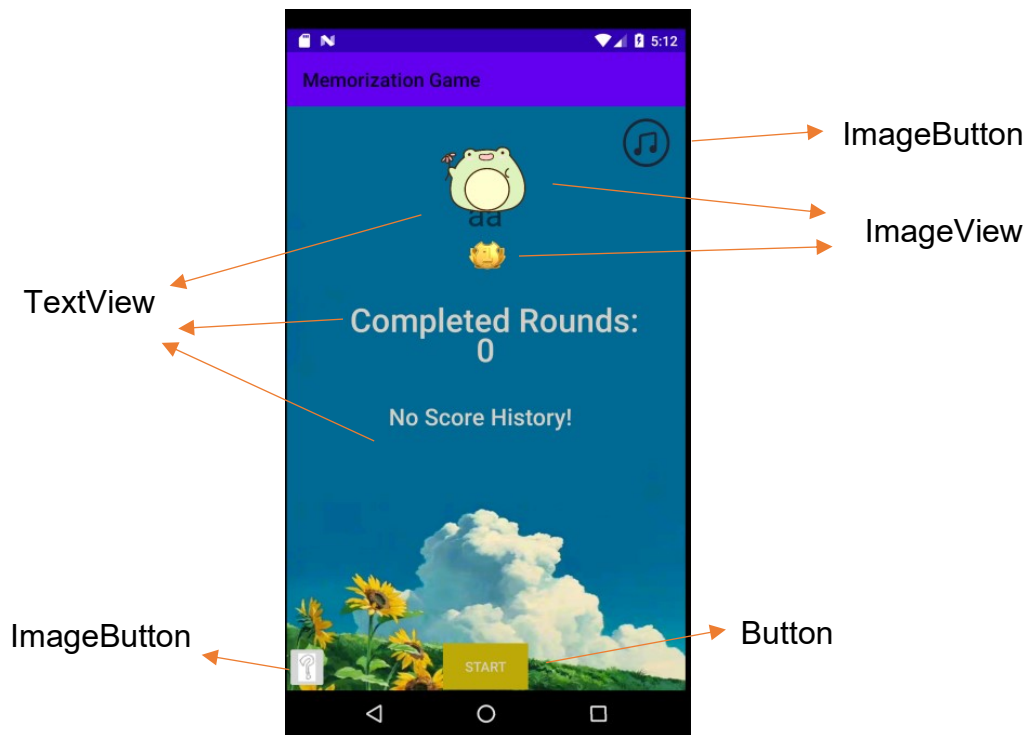
3.2 Loginpage



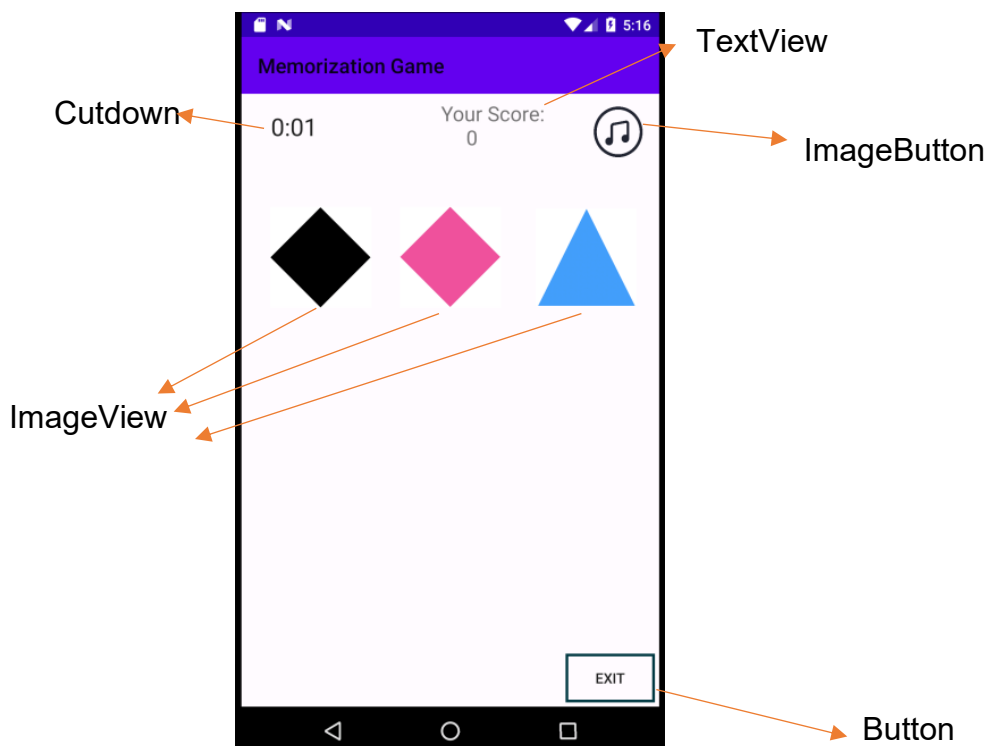
3.3 Create Account Page



3.4 Account page



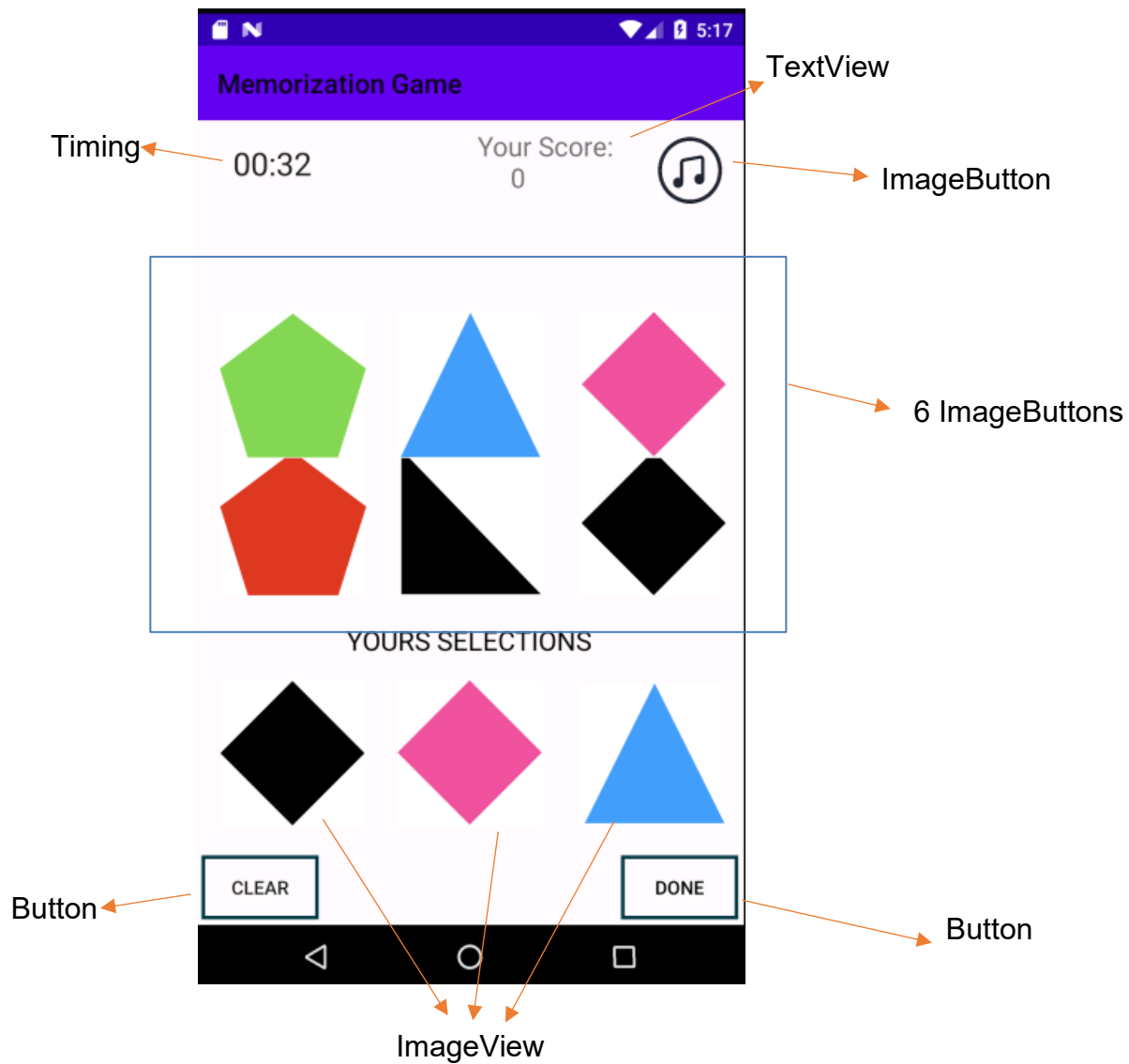
3.5 Game Page



Technology used:

- The Countdown is used for countdown
- The imageView is used for showing randomly generated colored shapes.

3.6 Answer Page



Technology used:

- The timing widget is used

- The 6 ImageButtons represent the randomly generated options that the user can choose by clicking on
- The imageView is used for synchronously showing the colored shapes selected by user.

3.7 Result Page

