COMPX202 Assignment8 Documentations

Authors

Xuanlin Jiang(J), 31711049; Ning Huang(H), 31711047

Project Name

Rolling Ball

Software Methodology

Iterative and incremental approach

Test Conditions

Emulator: Pixel 2, Real Device: Mi 6

Overall Descriptions

The game will keep or reenter the full screen mode. Player can input his name in the welcome screen and then start to play the game. In the game screen, a tip in the bottom will always pop up when game starts to remind the player to throw the light green for playing. For the core game play, player should throw the ball with proper angle to elude the black rectangle obstacles and hit more dark green circle targets, every hits to the targets will accumulate more scores, the fun factor is that

there is a small probably situation the ball will hit and rotate around the target to gain extra bonus scores, this is the chance for player to get higher scores. The ball will rebound when it hits both targets and screen edges, the game will end when the ball hits any obstacles. A alert will pop up to tell user the final score, after user dismiss the alert, the game will automatically restart. Player can check his personal top 5 score records by clicking the top right rank button. The score records will update after every play.

Initial Backlogs

TE: Time Estimate

Green stands for task modification Yellow stands for task insertion

Task ID	Description	Author	TE
1.	App Icon Design	H	2h
2.	Project Icon setting	J	15min
3.	Define functions in welcome screen	J	30min
4.	Wireframe design of welcome screen	J	30min
5.	Define icons & color for welcome screen	H	15min
6.	Layout the welcome screen with XML	H	30min

7.	Define functions in game screen	J	30min
8.	Define obstacles	H	30min
9.	Define targets	H	30min
10.	Define ball movement	J	30min
11.	Define how to calculate score	J	30min
12.	Wireframe design of game screen	J	30min
13.	Define icons & color for game screen	H	15min
14.	Layout the game screen with canvas	H	1.5h
15.	Implement ball movement	J	5h
16.	Test ball movement	H	2h
17.	Implement obstacle effect	J	5h
18.	Test obstacle effect	H	2h
19.	Implement score system	J	3h
20.	Test score system	H	2h
21.	Define functions in score screen	J	15min
22.	Wireframe design of score screen	J	30min

23.	Define icons & color for score screen	H	15min
24.	Display pseudo data in score screen	J	1h
25.	Use flow in home screen	H	1h
26.	Use flow in game screen	J	3h
27.	Use flow in score screen	J	1h
28.	Test Use flow between screens	H	2h
29.	Transfer data from home to game	J	1h
30.	Transfer data from game to score	J	2h
31.	Display Top 5 in score screen	J	2h
32.	Test Top 5 in score screen	H	1h
33.	Test the whole game	H	5h

Recordings

Task 1: Designing an icon

TE: 2h, **Actual**: 1h

My inspiration comes from the elastic ball. When it falls, it's very fast. I use the half-moon shape to show the speed when the elastic ball falls. The gray shadow of the ground also reflects the moment when the elastic ball falls. The main color I choose is green. Here are two different design schemes. After thinking, I chose the second logo of our app.

#128A43 #FFFFFF #9E9D9E

Task 2: Project Icon & name setting

TE: 15min, **Actual**: 5min

This stage not only set the app icon, also changed the app name to rolling ball.

Task 3: Three screens setting

TE: 15min, **Actual**: 15min

Three activities (welcome, game and score) with corresponding xml files are created and set in the project. Also all three activities inherits from the FullScreenActivity where the full screen display is enabled.

Task 4: Define functions in welcome screen

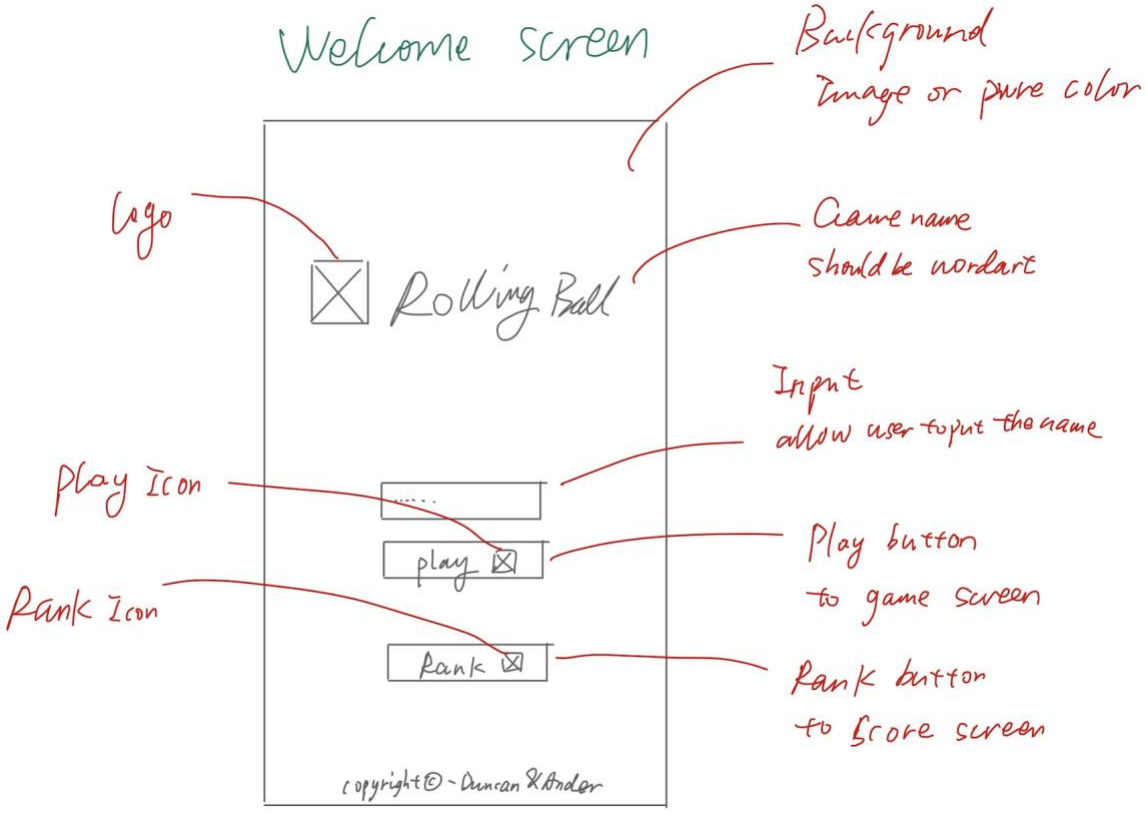
TE: 20min, **Actual**: 10min

The welcome screen is also the every first screen when user open the game, thus it must have following functions: 1) game brand identification: telling user what game he is playing. 2) A input allow user to type in his name. 3) A play button to start the game (go to the game screen). 4) A rank button to show user top 5 score ranks (go to the score screen). 5) Copyright info.

Task 5: Wireframe design of welcome screen

TE: 30min, **Actual**: 15min

According to Task4, the wireframe design of welcome screen is as follows:



Task 6: Define icons & color for welcome screen

TE: 15min, **Actual**: 25min

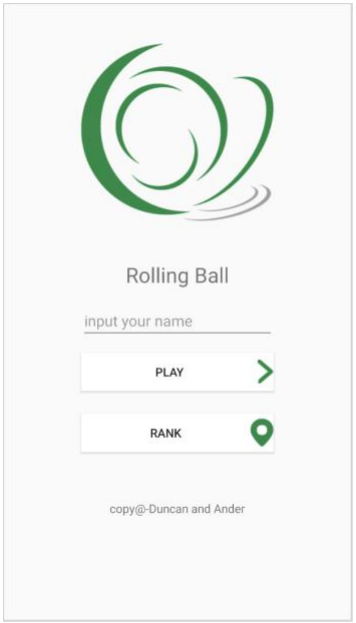
The colors I define and the choice of icon are based on the design of our logo, green. Icon belongs to the original icon, which is concise and clear to show the function of the button.

#128A43

Task 7: Define functions in game screen

TE: 30min, **Actual**: 50min

According to the sketch above, I made some minor adjustments in the layout. Make the overall welcome screen more concise. The color of logo is in line with our main color.



Release v1.0: Welcome screen is finished

Task ID	Description	Author	TE
1.	App Icon Design	H	2h
2.	Project Icon & name setting	J	15min
3.	Three screens setting	J	15min
4.	Define functions in welcome screen	J	20min
5.	Wireframe design of welcome screen	J	30min
6.	Define icons & color for welcome screen	H	15min
7.	Layout the welcome screen with XML	H	30min
8.	Define functions in game screen	J	30min
9.	Define obstacles	H	30 min
10.	Define targets	H	30 min
11.	Define ball movement	J	30 min
12.	Define how to calculate score	J	30 min
13.	Wireframe design of game screen	J	30 min
14.	Define icons & color for game screen	H	15 min

15.	Layout the game screen with canvas	H	1.5h
16.	Implement ball movement	J	5h
17.	Test ball movement	H	2h
18.	Implement obstacle effect	J	5h
19.	Test obstacle effect	H	2h
20.	Implement score system	J	3h
21.	Test score system	H	2h
22.	Define functions in score screen	J	15min
23.	Wireframe design of score screen	J	30min
24.	Define icons & color for score screen	H	15min
25.	Display pseudo data in score screen	J	1h
26.	Use flow in home screen	H	1h
27.	Use flow in game screen	J	3h
28.	Use flow in score screen	J	1h
29.	Test Use flow between screens	H	2h
30.	Transfer data from home to game	J	1h

31.	Transfer data from game to score	J	2h
32.	Display Top 5 in score screen	J	2h
33.	Test Top 5 in score screen	H	1h
34.	Test the whole game	H	5h

In this version we initialize the project including files, icons, logo, name, etc. Then we design and layout the welcome screen, also we evaluated some task modifications and improvements.

1. Task 2 is added with also setting the app name.
2. Task 3 (setting the three screens) is inserted, we actually need some initialized tasks before staring.
3. We now only have one main green color from Task 1, we may need more in the following tasks when making game play.
4. Time estimate for task 4 is shorter.

Task 8: Define functions in game screen

TE: 30min, **Actual**: 15min

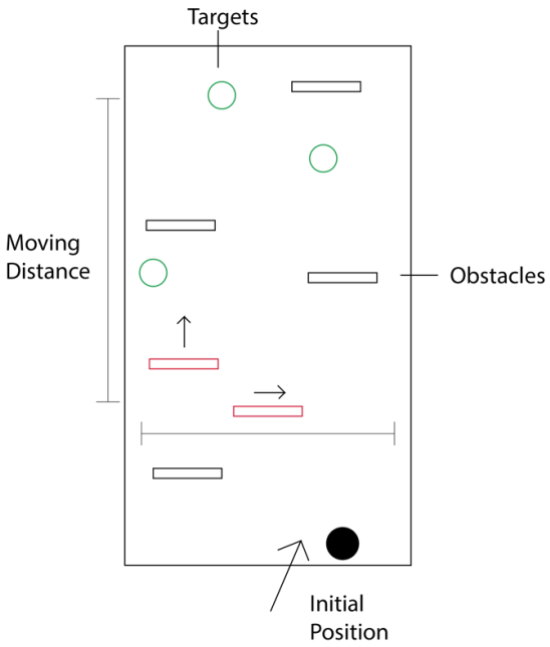
The game screen is the core screen that user plays game on, which not only needed game play function, but also some interactive functions, dTEils are as follows: 1) A tip message tells user the game is going to

start. 2) A cancel button which stops the current game and go back to home screen. 3) A replay button allows user to replay the game. 4) A rank button allows user to go to the score screen. 5) A message in the top showing real-time score. 6) A message box pop up after the game is finished, with the player name and final score. 7) Game play functions will be discussed in the following tasks.

Task 9: Define obstacles & targets

TE:1h, **Actual**: 1.5h

After discussion with the team member, we decided to use rectangle to represent obstacles, green circle to represent target, the game ended when touching obstacles, and increased scores when touching circles. The obstacles indicated in red can move, and the moving track and distance have been identified. Black represents the initial position of the ball.



Task 10: Define ball movement & score

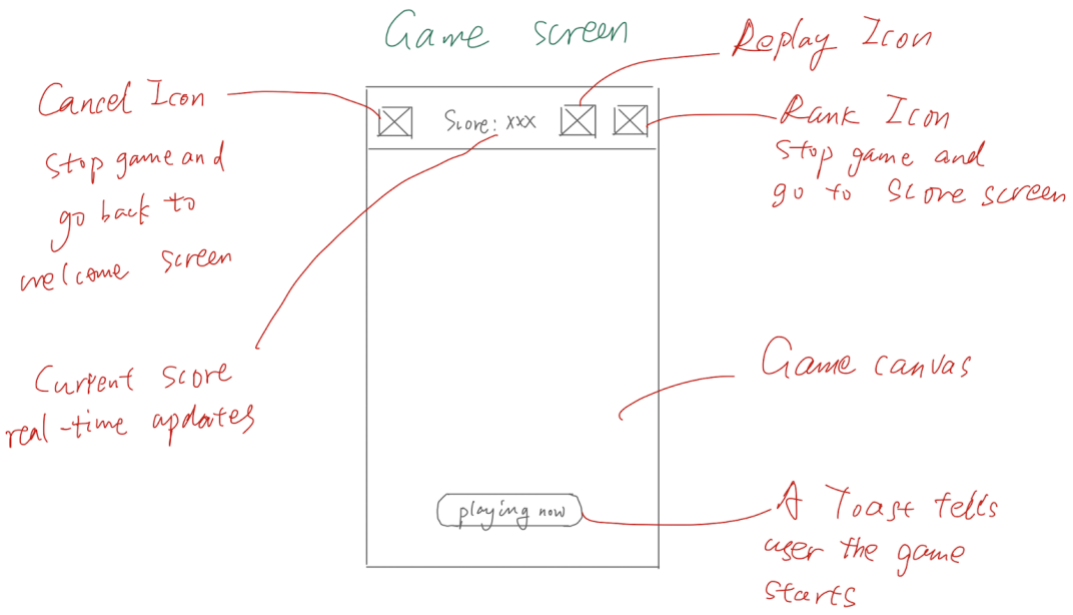
TE:1h, **Actual**: 1h

After discussion with the team member, we decided the ball in the start point is allowed to be threw, When the ball is activated by the Fling gesture, ball will move accelerate to the gesture direction, bouncing to opposite direction when it reach the screen edge and targets circles. For score system, three circle targets towards different points, the one nearer the top screen contains more points(3points, 2points, 1point). The total points will accumulated once the ball reach any targets. Finally, the game can end with two conditions: 1) user stops it manually. 2) Ball touches the obstacles.

Task 11: Wireframe design of game screen

TE:30min, **Actual**: 30min

According to Task 8, 9 & 10. The wireframe design of game screen is as follows:



Task 12: Visible objects hierarchy

TE: 1h, **Actual**: 40min

VisibleObject as parent class of visible objects in canvas is added in this task, Rectangle class represents obstacles and Circle class represents both targets and ball.

Task 13: Define icons & color for game screen

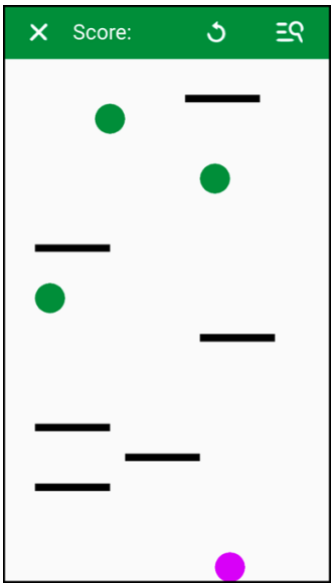
TE: 15min, **Actual**: 25min

Green is a very difficult color to match, so I still choose white as the bottom color of the game screen, and Pinball uses green complementary color, purple. Obstacles are still black to highlight the characteristics of obstacles. All the design of icon is done by myself

Task 14: Layout the game screen with canvas

TE: 1.5h, **Actual**: 5.5h

The main difficulty of layout lies in the distribution of obstacles in the screen. Once the distribution is not good, it will cause the game to be too simple or too difficult. Here is the final game screen.



Task 15: Implement ball movement

TE: 5h, **Actual**: 2h

The ball movement is implemented with following features: 1) only when touching close to the ball to fling, the ball will start move. If the touch point far away from ball, ball will not be moved. 2) Ball will move follow the direction exactly from fling gesture. 3) Fling gesture with any directions are acceptable to affect the ball movement. 4) Ball will move and rebound inside the canvas view rather exceeding it. 5) If user fling the ball with fast speed, the ball will also move fast forward, otherwise, with slow speed.

Task 16: Test ball movement

TE: 2h, **Actual**: 1h

1. open the Android virtual machine, and I try to click in the blank space and find that the ball will not move. When you touch it close to the ball, it starts to move. If the contact point is away from the ball, the ball will not move. Meet the requirements

2. I tried to throw the ball at multiple angles (up, down, left, right, etc.), and the ball can move according to the throwing angle. Because the

Android virtual machine is used to test, the ball's response is not sensitive.

3. After trying to move in different directions, the ejection direction of the ball is the same as the throwing direction, which meets the expected requirements.

4. I adjust it by dragging at different speeds (fast, slow, uniform). The ball's ejection speed moves according to my mouse dragging speed, which meets the expected standard.

Task 17: Implement obstacle effect

TE: 5h, **Actual**: 1h

The obstacle affect is implemented with following features: 1) one obstacle move horizontally in the specific range inside the view area. 2) another obstacle move vertically in the specific range inside the view area. 3) Once the ball interacts any edges of any obstacles, the ball will stop and be reset to the start point. 4) The reset ball could be flung again.

Task 18: Implement target effect

TE: 3h, **Actual**: 2h

The target affect is implemented with following features: 1) once the ball hits any of the target, the ball will rebound to other direction. 2) The small probability event may happen that the ball intersect and move around the target, then the ball will rebound, this situation provide a chance to make higher score.

Task 19: Test obstacle effect

TE: 1h, **Actual**: 1.5h

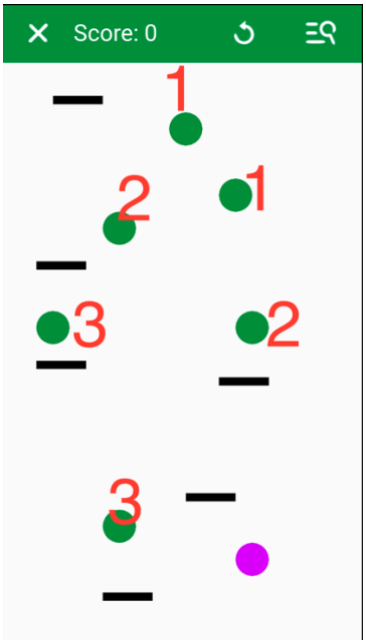
When testing obstacles, I found that the width of the obstacles is too large, which makes the ball easy to touch and makes scoring difficult. So I reduced the size and location of the obstacles. For mobile obstacles, I reduce their movement range and speed, because too difficult games will lead to a very poor sense of experience.

Task 20: Test target effect

TE: 1h, **Actual**: 1.5h

For the target ball, during the test, I found that the number of target balls was too small. It's very difficult for the purple ball to hit the green target ball, which makes scoring very difficult. So I increased the

number of goals and made it easier to score. I also changed the position of the green target ball to a certain extent. The top ball can have a continuous impact with the screen, making scoring easier. According to the degree of ease, I also mark the score of the ball, the more difficult it is to hit the ball, the higher the score.



Task 21: Implement score system

TE: 3h, **Actual**: 3h

The score system is implemented with following features: 1) When firstly the game screen is loaded or after the end of each play, a toast will appear with “play now” text, telling the user can play the game

now. 2) When the ball making a collision with any targets, the current score will be accumulated with the value of that target, the score will be timely updated in the top score view. 3) After game is over, an alert window pops up to show the final score. 4) After user clicking the OK button, the game will be reset, so as the score view.

Task 22: Test score system

TE: 2h, **Actual**: 1h

When testing the scoring system, we found that if our ball collides with any edge of any target ball, it will lead to extra points, so there will be a lot of extra points for a ball. But we kept the problem because for every ball, the difficulty of collision didn't decrease, so scoring didn't affect the difficulty. After continuous testing, the scoring system is in good condition without errors, and can complete scoring in time.

Release v1.1: Game screen is finished

Task ID	Description	Author	TE
1.	App Icon Design	H	2h
2.	Project Icon & name setting	J	15min
3.	Three screens setting	J	15min

4.	Define functions in welcome screen	J	20min
5.	Wireframe design of welcome screen	J	30min
6.	Define icons & color for welcome screen	H	15min
7.	Layout the welcome screen with XML	H	30min
8.	Define functions in game screen	J	30min
9.	Define obstacles & targets	H	1h
10.	Define ball movement & score	J	1h
11.	Wireframe design of game screen	J	30min
12.	Visible objects hierarchy	J	1h
13.	Define icons & color for game screen	H	15min
14.	Layout the game screen with canvas	H	1.5h
15.	Implement ball movement	J	5h
16.	Test ball movement	H	2h
17.	Implement obstacle effect	J	5h
18.	Implement target effect	J	3h
19.	Test obstacle effect	H	1h

20.	Test target effect	H	1h
21.	Implement score system	J	3h
22.	Test score system	H	2h
23.	Define functions in score screen	J	15min
24.	Wireframe design of score screen	J	30min
25.	Define icons & color for score screen	H	15min
26.	Display pseudo data in score screen	J	1h
27.	Use flow in home screen	H	1h
28.	Use flow in game screen	J	3h
29.	Use flow in score screen	J	1h
30.	Test Use flow between screens	H	2h
31.	Transfer data from home to game	J	1h
32.	Transfer data from game to score	J	2h
33.	Display Top 5 in score screen	J	2h
34.	Test Top 5 in score screen	H	1h
35.	Test the whole game	H	5h

In this version we finish the key game play of our game, including the ball movement, obstacle effects of ending the game, target effects of rebounding ball and accumulating scores and score system. Besides, the fun factor is highlighted that the ball may move around the targets after getting collisions in small probability situations which leads to the bonus for scores, this is the chance for user to get higher scores.

1. Original Task 9 & 10 is now combined to Task 9, as we consider the targets just following the decisions on obstacles, they are similar.
2. Original Task 11 & Task 12 is now combined to Task 10, as we consider the score system just following the decisions on ball movement.
3. Task 12 (Visible objects hierarchy) is inserted into before the original Task 12 (Define icons & color for game screen), as the objects hierarchy should be established before laying out the game screen.
5. Task 18 (Implement target effect) and Task 20 (Test target affect) are inserted, as we need to implement effects of both obstacles and targets before doing the score system.
6. Layout and icons should be improved in the later tasks.

Task 23: Define functions in score screen

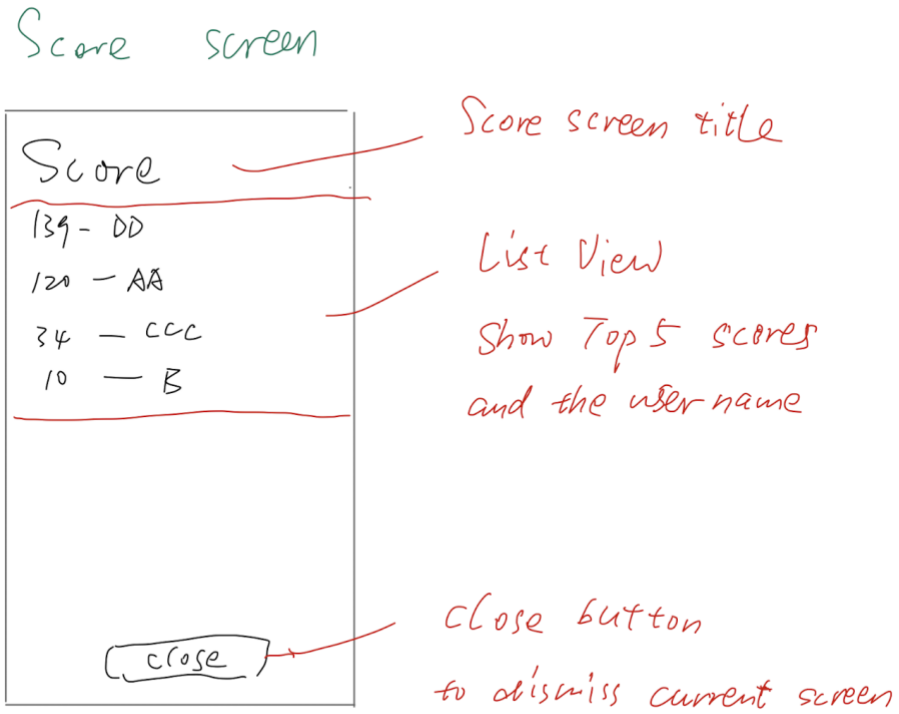
TE: 15min, **Actual**: 15min

The score screen functionalities are as follows: 1) A big title(e.g., Score) tells user what is the current screen. 2) A list view shows top 5 high score record with score and usernames in descending order. 3) A close button that can dismiss current screen, and then jump to home screen.

Task 24: Wireframe design of score screen

TE: 30min, **Actual**: 15min

According to Task 23, the wireframe design of welcome screen is as follows:



Task 25: Define icons & color for score screen

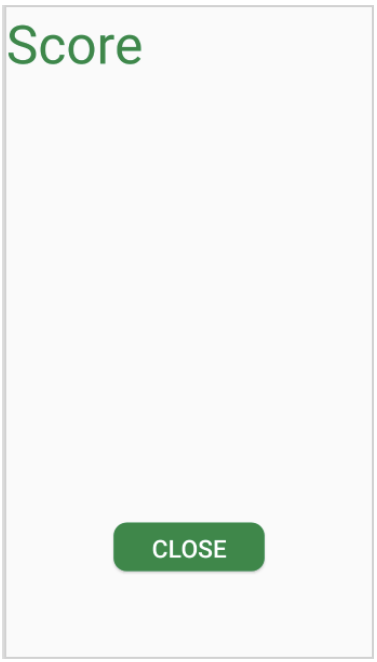
TE: 15min, **Actual**: 35min

The color I chose is still the main color of our app, light green. Button is a common round button, still light green.

Task 26: Layout the score screen

TE: 1h, **Actual**: 1h

The overall layout is linear from top to bottom. They are "score" and "List View" (to show the top five scores), and the close button.



Task 27: Use flow in home screen

TE: 1h, **Actual**: 1h

The use flow in home screen is implemented as follows: 1) User can type their name in the name input. 2) The keyboard showing up will exit the full screen mode when clicking the name input. When clicking down-arrow button in the navigation bar after finishing inputting the name, the app will enter full screen mode again. 3) User can click the play button to play the game which will lead to the game screen. 4) When the name input is empty, the app will not bring user to game screen after clicking the play button. 5) User can click the rank button to see the top 5 list which will lead to the score screen.

Task 28: Use flow in game screen

TE: 1h, **Actual**: 30min

The use flow in game screen is implemented as follows: 1) User can click the close button to dismiss the game screen which will lead to the welcome screen. 2) User can click the reset button to restart the current game only when the game is ongoing. 3) User can click the rank button to see the top 5 list which will lead to the score screen.

Task 29: Use flow in score screen

TE: 1hr, **Actual**: 10min

The use flow in score screen is implemented as follows: 1) User can click the close button to dismiss the score screen which will lead to the screen that is requested to show the score screen. For example, if user click the rank button in welcome screen, then the app will go back to welcome screen after dismissing the score screen.

Task 30: Test use flows among screens

TE: 2h, **Actual**: 1h

First of all, I enter the game from the "welcome" screen, and click the "play" button to find that I can't enter. I entered my name and successfully entered the game. I click the close button in the upper left corner, and the screen successfully returns from the game screen to the welcome screen. I enter the "game" screen again, click the button of ranking list, and the game successfully enters the "ranking" page. I click the close button at the bottom of the screen, and the game returns to the game screen successfully. I clicked the close button and the game successfully returned to the welcome page. In the "welcome" page, I click the ranking button, and the game enters the "ranking" page. The whole smooth operation, button to achieve perfect.

Task 31: Transfer data from home to game

TE: 1h, **Actual**: 30min

The only data needs to be transferred from welcome screen to game screen is the player name, it is also for the later on score recordings. In detail, the game screen will show up when the user had inputted his name. The game screen will show the tip in the bottom to remind to start the game, the tip contains the player name. When the game is over, a window pop up will display the final score with corresponding player name.

Task 32: Transfer data from game to score

TE: 2h, **Actual**: 2h

Once the game is ended in the game canvas, the game screen will get the latest score, then the Set is used to remove duplicated data and priority queue is used to get top five scores. Then the score array with up to five elements and player name are passed when starting the score screen.

Task 33: Display Top 5 in score screen

TE: 2h, **Actual**: 2h

The top 5 score display functionality has the following features: 1) If user gets 0 after play, the data will excluded from score record. 2) If currently no score is generated, the list view in score screen will show a tip to remind the player to play first. 3) The title in score screen contains the player's name. 4) The score record keeps until user dismiss the game screen. In detail, if user dismiss the score screen, and enter it again from game screen, the score records are still there. 5) The score list only displays top 5 scores that current user generates.

Task 34: Test Top 5 in score screen

TE: 1h, **Actual**: 1h

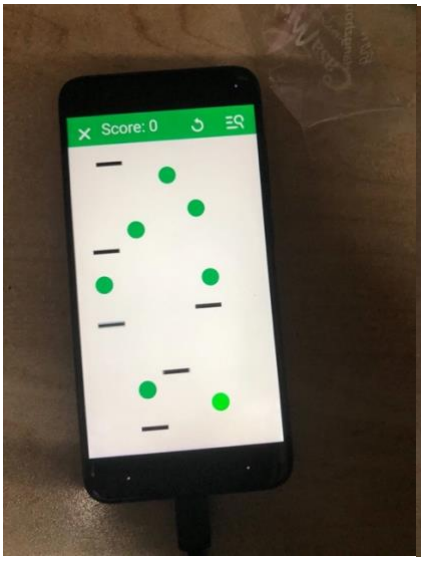
I tested the score system and found that my top 5 scores could be displayed perfectly in the score form. No matter how many games I play, the top five scores are still correctly arranged.

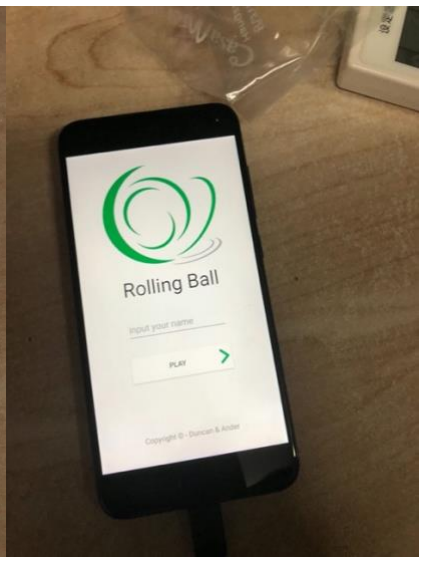
Task 35: Test the whole game

TE: 1h, **Actual**: 1.5h

I entered the game screen to input the game name, and found that the input was normal. I click the "play" button to enter the game screen

successfully. The name I typed is also correctly displayed on the game screen. I clicked the close button and successfully returned to the game screen. I entered the game again and began to operate the game. All functions are normal. But when I used my mobile phone to test, I found that the pinball was not sensitive, so I adjusted the touch sensitivity and successfully solved the problem. The game is running normally. I played six games and they all worked. I opened the "rank" expression, and my top 5 scores were successfully displayed on the form. I click the close button to return to the welcome screen. Everything is OK. All functions operate normally. Here is my test conditions.





Task 36: Layout and color adjustment

TE: 1h, **Actual**: 1.5h

I adjusted the color of the pinball to fit our theme better. At the same time, I also adjusted the position of the display bar to green. I've changed all the XML warnings to make the code more standard. There are now only two green colors besides the black and white colors. All the buttons are changed to rectangles while not rounded. Unused resources (rank button for welcome screen) is removed.

Task 37: Performance improvement

TE: 2h, **Actual**: 1h

In order to get the top 5 score records, complicated data structure and method are used in task 32, which is bad to the performance, thus this functionality is now optimized to only use a fixed array with the help of sort() in Arrays.

Task 38: Code reorganization and comment

TE: 1hr, **Actual**: 45min

The code is reorganized and comments are added to all the classes.

Task 39: Fix layout issue for different size displays

TE: 1hr, **Actual**: 30min

The sizes and positions of all visual objects are now modified to relative ratio, in the case that the size of game canvas are fit to the screen size. Thus the display bug is fixed. Now players with different screen size can play game perfectly.

Release v2.0: The Game project is finished

Task ID	Description	Author	TE
1.	App Icon Design	H	2h
2.	Project Icon & name setting	J	15min
3.	Three screens setting	J	15min
4.	Define functions in welcome screen	J	20min
5.	Wireframe design of welcome screen	J	30min
6.	Define icons & color for welcome screen	H	15min
7.	Layout the welcome screen with XML	H	30min

8.	Define functions in game screen	J	3min
9.	Define obstacles & targets	H	1h
10.	Define ball movement & score	J	1h
11.	Wireframe design of game screen	J	30min
12.	Visible objects hierarchy	J	1h
13.	Define icons & color for game screen	H	15min
14.	Layout the game screen with canvas	H	1.5h
15.	Implement ball movement	J	h
16.	Test ball movement	H	2h
17.	Implement obstacle effect	J	5h
18.	Implement target effect	J	3h
19.	Test obstacle effect	H	1h
20.	Test target effect	H	1h
21.	Implement score system	J	3h
22.	Test score system	H	2h
23.	Define functions in score screen	J	15min

24.	Wireframe design of score screen	J	30min
25.	Define icons & color for score screen	H	15min
26.	Layout the score screen	H	1h
27.	Use flow in home screen	J	1h
28.	Use flow in game screen	J	1h
29.	Use flow in score screen	J	1h
30.	Test use flows among screens	H	2h
31.	Transfer data from home to game	J	1h
32.	Transfer data from game to score	J	2h
33.	Display Top 5 in score screen	J	2h
34.	Test Top 5 in score screen	H	1h
35.	Test the whole game	H	1h
36.	Layout and color adjustment	H	1h
37.	Performance improvement	J	2h
38.	Code reorganization and comment	J	1h
39.	Fix layout issue for different size displays	J	1h

The game project is finished, modifications to tasks and app are as follows.

1. The rank button in welcome screen is removed, as the score data is are generated in the game activity, the app will never get the scores in welcome screen. So, the user can see his personal top 5 scores data in score screen by clicking the rank button in the game screen.
2. Reset and rank buttons in game screen are restricted, the game will be reset only when the game is being played. On the contrary, only when currently user is not playing the game, the rank button will jump to score screen.
3. Task 26 is changed only to layout the score screen, cause it is not necessary to show some pseudo data in the list view, instead we gave height to the list view when layout.
4. As mentions in previous release, the layout and color should be improved thus we insert Task 36.
5. Task 37 and Task 38 are inserted to improve the game better.
6. Thanks to Andrea and Sam pointing out the display issue in different screen, thus we insert Task 39.

Final Evaluation: Performance may improve

There are also some improvements could be made to the game for better user experiences. For example, since we have improved the performance in Task 37 when recording the scores, there is also other room for performance improvements. When calculating the collisions of ball, targets and obstacles, we calculated all of them, no matter where the ball is, actually many of the collisions detections shouldn't be done until the ball goes close to the corresponding objects. Thus we may divide the whole canvas to 5 or more areas vertically. Then the collision detections only be made area by area when the ball is going to pass through. This measurement will make the game more smooth and consume less battery.

References

The icons in all the screens are from:

https://www.iconfont.cn/collections/detail?spm=a313x.7781069.1998910419.de12df413&cid=16472