COMPX202 Assignment8 Documentations

**Authors**

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

**Project Name**

Rolling Ball

**Software Methodology**

Iterative and incremental approach

**Backlogs**

TE: Time Estimate

|  |  |  |  |
| --- | --- | --- | --- |
| Task ID | Description | Author | TE |
|  | App Icon Design | H | 2 hrs |
|  | Project Icon & name setting | J | 15 mins |
|  | Three screens setting | J | 15mins |
|  | Define functions in welcome screen | J | 20 mins |
|  | Wireframe design of welcome screen | J | 30 mins |
|  | Define icons & color for welcome screen | H | 15 mins |
|  | Layout the welcome screen with XML | H | 30 mins |
|  | Define functions in game screen | J | 30 mins |
|  | Define obstacles & targets | H | 1 hrs |
|  | Define ball movement & score | J | 1 hrs |
|  | Wireframe design of game screen | J | 30 mins |
|  | Visible objects hierarchy | J | 1hrs |
|  | Define icons & color for game screen | H | 15 mins |
|  | Layout the game screen with canvas | H | 1.5 hrs |
|  | Implement ball movement | J | 5 hrs |
|  | Test ball movement | H | 2 hrs |
|  | Implement obstacle effect | J | 5 hrs |
|  | Implement target effect | J | 3 hrs |
|  | Test obstacle effect | H | 1 hr |
|  | Test target effect | H | 1 hr |
|  | Implement score system | J | 3 hrs |
|  | Test score system | H | 2 hrs |
|  | Define functions in score screen | J | 15 mins |
|  | Wireframe design of score screen | J | 30 mins |
|  | Define icons & color for score screen | H | 15 mins |
|  | Display pseudo data in score screen | J | 1 hrs |
|  | Use flow in home screen | H | 1 hrs |
|  | Use flow in game screen | J | 3 hrs |
|  | Use flow in score screen | J | 1 hrs |
|  | Test Use flow between screens | H | 2 hrs |
|  | Transfer data from home to game | J | 1 hrs |
|  | Transfer data from game to score | J | 2 hrs |
|  | Display Top 5 in score screen | J | 2 hrs |
|  | Test Top 5 in score screen | H | 1 hrs |
|  | Test the whole game | H | 5 hrs |

**Recordings**

Task 1: Designing an icon

**TE**: 2hrs, **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**: 15mins, **Actual**: 5min

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

Task 3: Three screens setting

**TE**: 15mins, **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**: 20mins, **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**: 30mins, **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 interface more concise. The color of logo is in line with our main color.



Release v1.0: Welcome screen is finished

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.

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**:1hr, **Actual**: 1.5hrs

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**:1hr, **Actual**: 1hr

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**:30mins, **Actual**: 30mins

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



Task 12: Visible objects hierarchy

**TE**: 1hr, **Actual**: 40mins

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**: 15mins, **Actual**: 25mins

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

Task 14: Layout the game screen with canvas

**TE**: 1.5hrs, **Actual**: 5.5hrs

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



Task 15: Implement ball movement

**TE**: 5hrs, **Actual**: 2hrs

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**: 2hrs, **Actual**: 1hrs

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**: 5hrs, **Actual**: 1hrs

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**: 3hrs, **Actual**: 2hrs

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**: 1hr, **Actual**: 1.5hrs

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**: 1hr, **Actual**: 1.5hrs

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**: 3hr, **Actual**: 3hrs

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**: 2hr, **Actual**: 1hr

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 v2.0: xxx

1. Original Task 9 & 10 is now combined to Task 9, as we consider the targets just following the decisions on obstacles.
2. Original Task 11 & Task 12 is now combined to Task 10, as we consider the score sytem 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.
4. 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.

**References**