CS6301 Human-Computer Interactions

Fall 2020

Assignment 6 – Write a Reaction Time Test Game

Your assignment is to write a relatively simple game. The game play will display shapes of various colors, and the object is to touch only the correct ones to make them disappear. A shape will stay on the screen until it is touched. In the text below I sometimes refer to the shapes as “balloons,” and touching them as popping them. In more detail:

1. Bring up a screen that describes game play to the user. For example, it may say, “Touch only the red circles,” or “Touch only purple squares.” Your game will have only circles and squares as shapes, and using only the colors Red, Orange, Yellow, Green, Blue, Purple, and White. These will be shown on a dark background. They will be different randomly-chosen sizes, ranging from 32dp to 64dp, but you can experiment with this. For circles, this is the diameter. For squares, this is the length of a side. The shape and color are also chosen randomly when the program starts. A general rule is that shapes **never** overlap.
2. When the user presses the OK button, bring up a screen (a new Activity) that shows a random number of circles and squares in different sizes and colors at random positions on the screen. However, make sure the entire shape is visible, not just part of it. The total number of shapes should be between 6 and 12. The program can display a mix of squares and circles. Again, choose the mix randomly.
3. Objects will stay on the screen until touched or their lifetime expires (see below.) When an object is touched, remove it and generate a new object.
4. Shapes have a lifetime, randomly chosen. If a correct shape disappears before you touch it, its lifetime is added to your total time. Experiment with lifetimes between 3 and 7 seconds, assigned randomly.
5. Shapes move every second by 20 pixels in a randomly-chosen direction, as long as they don’t overlap with another shape. This makes the game a little more difficult.
6. The playing field is from the bottom of the screen to 2/3 of the screen height. You can have other things above that, such as a running score, timer, etc. Outline the playing field with colored lines.
7. If you touch a correct shape the time is taken from the MotionEvent. The touch must be the coordinates at the ACTION\_DOWN. That is, sliding your finger around on the screen won’t work. Thus the reaction time is the difference between the time the shape is touched minus the time it was created.
8. The game ends after 10 correct shapes have been touched. (This number is a suggestion; you can experiment with different values.)
9. Score is the total time it took to touch all of the correct shapes. Obviously, shorter times are better, since this tests reaction time. Touching an incorrect shape makes it disappear but has no effect on the score. At the end of the game, show the total score and the number of balloons that should have been popped but that went off the screen before the player got them. If this is a high score, the user can press a button to enter it into the high scores list. The shorter the time the better, so low numbers are good. Show minutes, seconds, and two decimal places for tenths and hundredths.

You will need to use a timer to drive this. On every timer tick, recompute the positions of the shapes and redisplay the screen.

Use the High Scores screen you wrote for assignment 5 to track high scores for this game. You’ll have to modify your main Activity a little. There will be four Activities in the game:

1. The starting activity with the Start button and instructions.
2. The game play screen
3. The high scores display screen from the previous assignment
4. The high scores entry screen from the previous assignment, but you cannot enter the score manually; it will come from the game.

**Do not** use a surface holder; just create a custom View and override onDraw.

Remember that there is timer code you can use on my Web site, under Programming Tips, and possibly other Android code. **You may not use any other code. This is an individual assignment.**

**To hand in:**

Your complete Android Studio project as a Zip file. The file must be named <NetID>Asg6.zip. You will be required to show your working program running on your device to the TA, as well.

|  |  |
| --- | --- |
| Project Grading | |
| Meets the above requirements | 70 |
| Clean, object-oriented code. There must be at least one base class you define and two subclasses of it. | 20 |
| Program comments and naming conventions | 10 |

**Additional grading criteria:**

Using a surface holder. -30

Game screen does not update correctly. This could be various problems, including the wrong number of shapes, shapes not moving at all, popping shapes that have not been touched, shapes overlapping other shapes, etc. -10 to -20

Scoring not integrated with the game. -10

Using images instead of drawn shapes for the shapes. -30

Not following good object-oriented design. -10 to -20

No header comments in even one module: -5