

Assignment #2 – Flood-It

Date assigned: Friday, January 12, 2018

Date due: Flood-It UMLet Design, Sunday, 1/14/18 by Noon (10 pts)
Flood-It Implementation, Thursday, 1/18/18 by 1pm (50 pts)

The game of Flood-It can be found at <http://unixpapa.com/floodit/>. You are to design and implement a similar game for android. Here are the specifications:

1. Design your application to run on a Nexus 4 768x1280 device. If you want to make your application more general to run on any device, feel free. I will be grading on the Nexus 4.
2. You do not have to allow for a setting to change the number of squares on the screen, but you do need to add a menu option that allows the user to select a difficulty level where Easy is 3 colors, Medium is 4 colors, and Hard is 5 colors. The game initially comes up by default in 3 colors. Any time a difficulty level is changed, the game is reset at the selected difficulty level.
3. Randomize your color selection around the given ARGB colors from the last assignment. You do not need to reuse any of your code thus far and the application is Android only. If you want to add more than 8 ARGB colors, feel free.
4. Design the game so that both the color squares AND a score are displayed at the same time.
5. You need to load your game up on an actual device and show me that it works before 1pm on the day in which the final implementation is due.
6. The design is to be done in groups of two as follows:

Ethan & Jacob

Ryan & Joshua

Jeffrey & Joseph

Sarah & Dannie

Sagnik & Nicholas & Chance

Clio & Michael

7. The implementation of the project is to be done individually. I will put all designs on Grace (and maybe the Web) no later than noon on Sunday.

Goals for Flood-It:

1. Write an Android application using multiple classes
2. Use packages to better organize all classes
3. Use good OOP techniques in designing your solution
4. Use the Android API which has a rich library of routines
5. Include other projects in an existing project
6. Debug with LogCat
7. Have fun with basic 2D graphics, touch events, and problem solving

Specifics:

1. Create a folder called your PUNetIDFloodIt that contains your entire Flood-It project. When the assignment is due, drop the PUNetIDFloodIt folder into CS260-01Drop on grace.
2. If you come to me with a question regarding your solution, I will have you load your project onto a machine in the CS lab. I will not look at your code on your computer or on paper as it just takes me too long to get at the problem. Further, I want you to bring in your lecture notes in case I want you to look up something. Remember, I'm not just a tell you the answer guy. Make sure you understand how to use the developer tools and that you can run your program in Eclipse.
3. If you want help with a compiler error, you must be able to tell me exactly what statement you put in your code that caused the error and be able to isolate the error. If you have typed in a bunch of code and have not tested your code as you've gone along, I'm not going to help you sort out the mess. You've been warned!!