

Self Review

By AbdelRahman



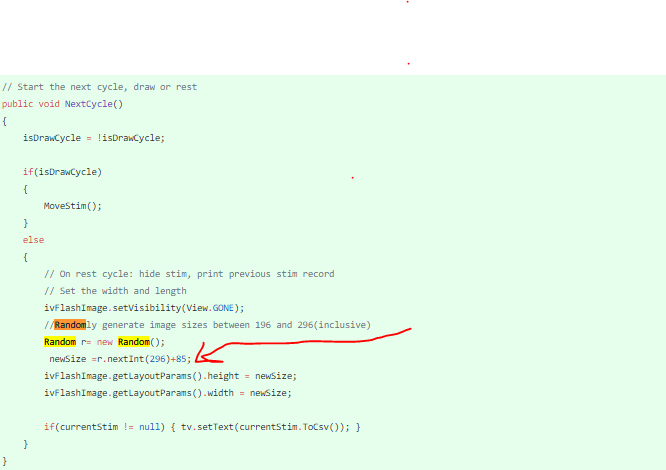
# Technical Proficiency

Project1 in the community group consisted of a lot of coding and being exposed to an array of different languages to which all posed a challenge and means to grow one vocabulary of languages. Moreover, the progressive nature of project1 was that it allowed us to gradually ease into the software development environment that involved much group work and intragroup communication. Coming from a somewhat simple

Programming background, this paper meant that there was so much more to learn. As Part of the client needs there 4 assigned tasks that are the following: the Visual Scan tool, Aya App, Internet of things Database API and a VR rendition of the Visual Scan tool. These tasks will form the bases for answering the self-review questions.

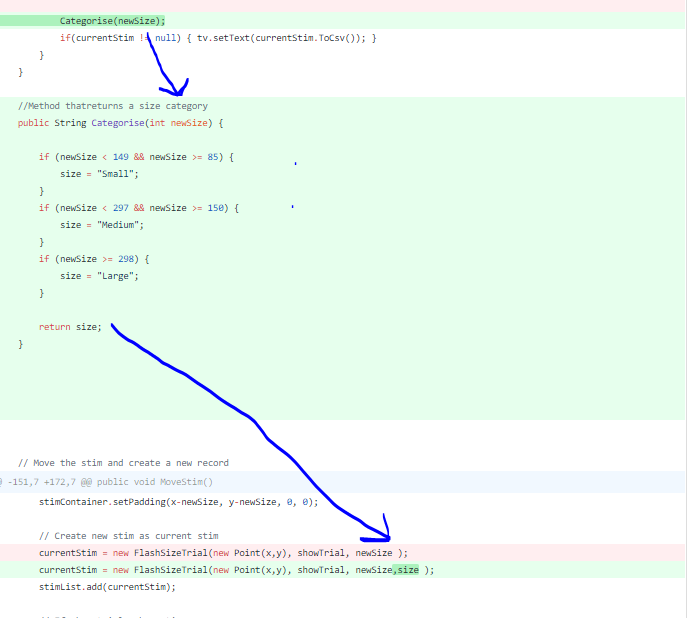
## Question: What is the overall quality of your code like?

### Task1: The visual Scan tool-FlashSizeImage.

As an initial transition to project1 paper, my task was to code the random generation of Image sizes that would appear on the screen in the visual scan flash tool. Integration of a new class flashSize fragment class with the required functionality needed another class /object called the FlashSizeTrial that defined its movement, and CSV file output. I injected a random generator and stored it in a variable called newSize. This value was used to set both the height and width of the image view drawn to the screen. This enabled the size of the image view to be controlled by changing the random upper and lower limit parameters.

<https://github.com/OtagoPolytechnic/CommSoftTasks/commit/e6353acb6070a6dd08533d57dbc559924d2950f1>

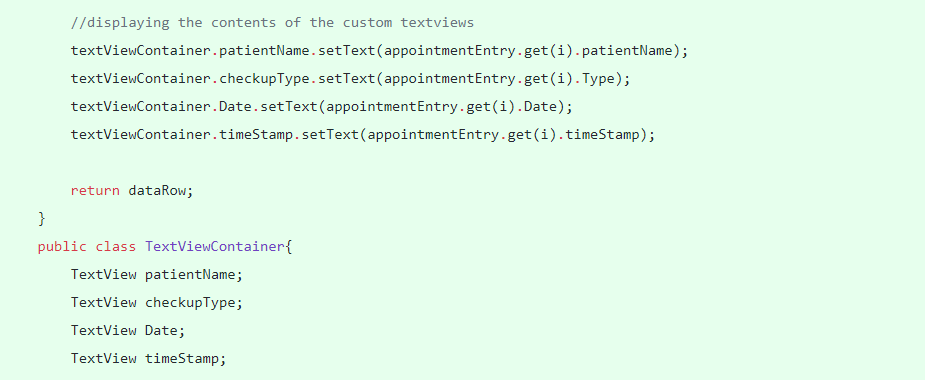
After the first initial meeting, the client needed information in the CSV to be more meaningful. This was remedied by adding functionality to display the size criteria of the imaged tapped. A simple method that checks the passed in random size integer and returns the corresponding size string data type. Following the preparation of the flashTrial constructor and its overriding toCSV toString to accept a size parameter, the categorize methods size property is seamlessly included in the class instances. The Simplicity of this code meant that it can be reusable and easily incorporated into additional classes with similar functionality.

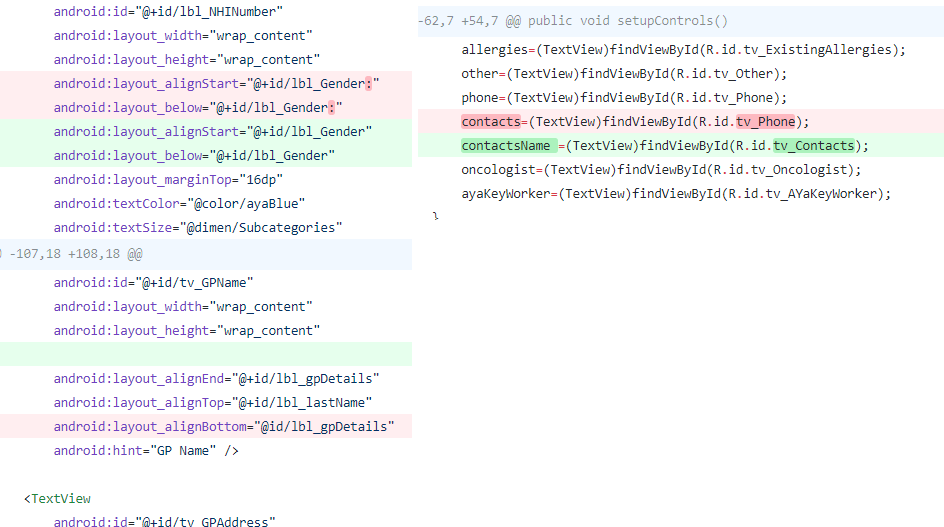


### <https://github.com/OtagoPolytechnic/CommSoftTasks/commit/d00b2da6c6c746cd9c4a14fef259b4945cbc5225>

### Task2: The Aya App.

Progressing from the Visual Scan tool, it was time for the group to onboard a medical passport app that was presented by the District Health Board. Fortunately, design students drafted page layouts for the application.

In the instance of coding the Appointments functionality that enabled users to add/edit and delete appointments with timestamps, we needed to inherit the gitView public method from the public interface adapter to populate a custom listview containing the patient's information. This custom adapter required a list of textViews that could have been declared as a global variable restricting its use within the parent activity. This is resolved by creating a simple Textview container class (courtesy of Samantha and adapted for this class) instantiated inside the method and its elements set by the retrieved XML appointment entries. This increases code modularity and tidiness as the classes sole purpose is to hold elements. As the AYA Apps activities increase, so is the need for a more efficient way to switch between intents. Having a separate button handler for each intent could prove cumbersome when adding more activities. This code conveniently creates an instance of the Intent when switching on the XML elements. As a result, the class method startActivity is executed according to the intent variable.

Also, the app consisted of the Edit Health Information registration form that had many text field and text view elements. Because of this, there was much need for a trivial- uniform labelling of their element ids. Having the id name and element they constituted separated by an underscore meant for easy search and fixes.

### Task3: Internet of Things database API.

Later in the semester, a client needed a means to monitor their hardware inventory and location. An ASP.Net database API was the best solution to this.

An API get request was needed to view all the items returned. However, database dependencies in the items table with the type model and subTypeModel tables made its difficulties. This was solved by this iterating through the items table and selecting its properties based on its ID’s. foreaching over the collection, this Loops through and connects the tables to the intermediary Item table. And because of the itemModel Table is a foreign key of newSubtype, that becomes its parameter. This is believed to be good quality code as the get method accepts an ID, makes a database query and selects on where the ID in the url get request and maches it with the items table ID. This contains a dynamically populated list of values in the Item table. 

### Task4: Unity Virtual Scan Wander Tool.