**Unit Test and Refactoring Questions**

1. Write a program using TDD to do the following (20 marks):
2. This program will generate a random number between 1 to 100. User will be asked to guess the number or enter ‘q’ to quit the program. The program will keep asking the user to guess the number until the user guesses it correctly or has quitted. Once the user has guessed it correctly or quitted, the program will tell the user how many attempts have been taken.
3. Refactor your code – Code smell can give indications that there is some issue with the codes and can be solved by refactoring. Identify issues and their related code smells in your code. Include solutions to fix up the issues and implement them.
4. Create a Git directory for your assignment (including word or pdf documents and programming code)
5. Capture screenshots and write briefly the steps you have taken to create this program

**Answer**

* Requirement 01 - Input validation
* Description - Here I have written 4 test cases to check the user’s input is valid or not.
* Test cases -
  1. When input is above 100
  2. When input is 0
  3. When input is between 0 – 100
  4. When input is empty
* Refactoring - The requirement integrates to Main activity. And set to display the toast message “Please enter a number between 0-100”, If the input is invalid. Here I initialized variables and UI components in Main activity.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

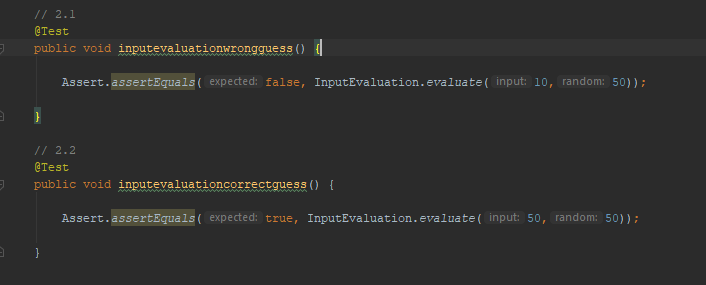
A screenshot of a cell phone

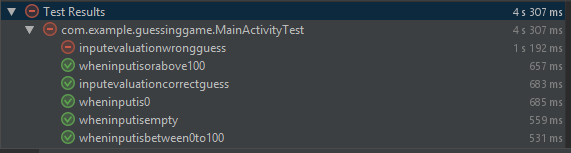
Description automatically generated

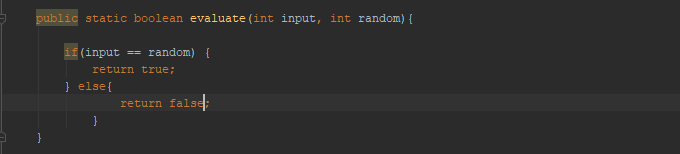
* Requirement 02 - Input evaluation.
* Description - Check whether the input is correct or not.
* Pre requests – Input is a valid input.
* Test cases -

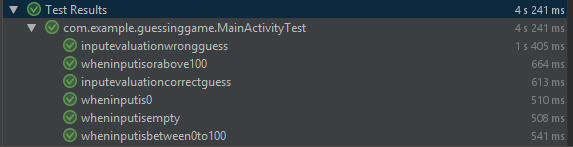
2.1 Check for wrong guess

2.2 Check for correct guess









Requirement 03 – Display results

Description - Check whether the application correctly display results.

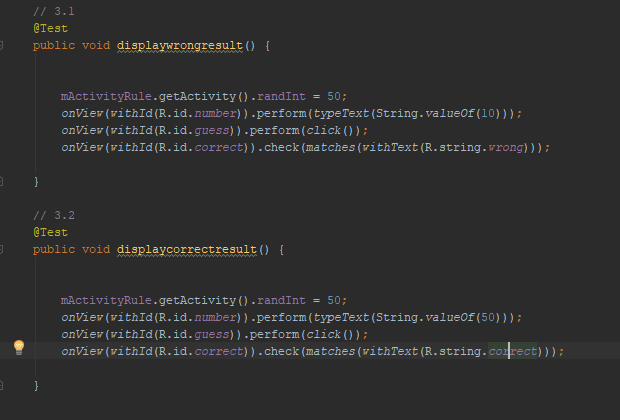
Pre-requests -

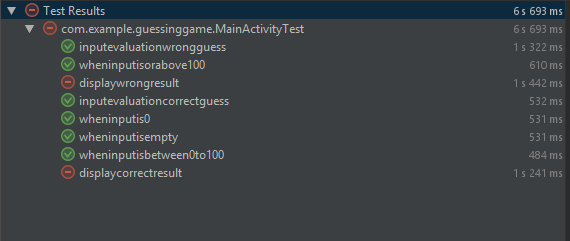
* 1. Input is a valid input.
  2. Input evaluation was done.

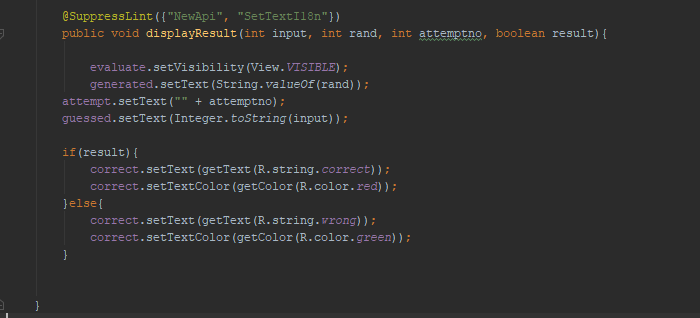
Test cases

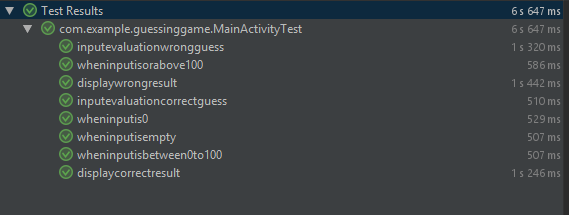
3.1 Check for wrong guess

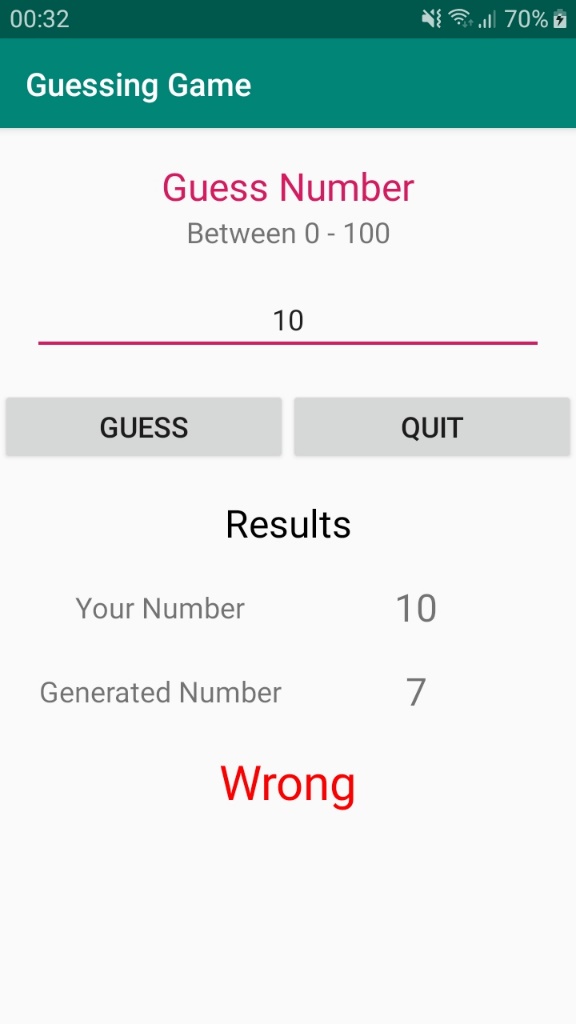
3.2 Check for correct guess











Requirement 04 – Display the no of attempts

Description – The no of attempts should display after the correct guess or after quit the game.

Pre-requests -

1. Input is a valid input.
2. Input evaluation was done and correct guess.

Test cases

4.1 Display no of attempts – when guess is correct

4.2 Display no of attempts – when click quit button

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Requirement 05 – Check the guess button will disable after the correct guess or quit game

Description – The guess button should disable if the guess is correct or after clicking the quit.

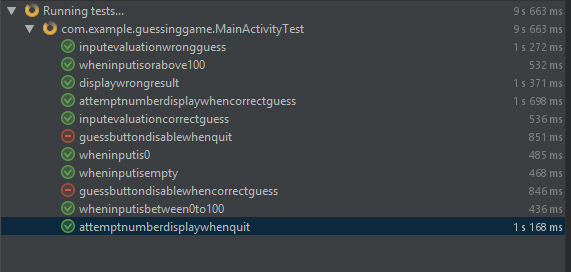
Pre-requests – If the guess is correct or click quit button

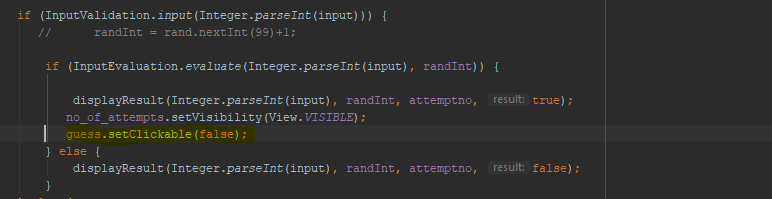
Test cases

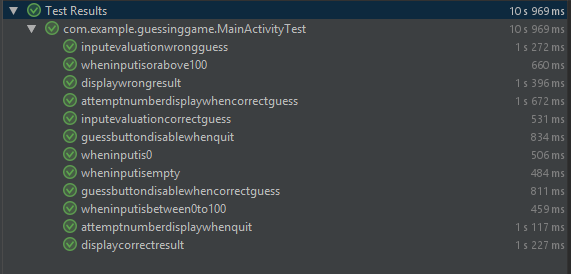
4.1 Disable guess button – when guess is correct

4.2 Disable guess button – when click quit button









Requirement 06 – Check the application will close after clicking quite button twice

Description - Application will close after clicking the quit button twice.

Pre-requests – If the guess is correct or click quit button

Test cases -

4.1 Close application – after clicked the quit button twice

