****

**North South University**

**Project Report**

**CSE 427**

Title: Tick-Tac-Toe

**Submitted By:**

|  |  |  |
| --- | --- | --- |
| Muhammad Sakib Khan | 1520016042 | muhammad.sakib@norhsouth.edu |

GitHub Project Link:

<https://github.com/nsuspring2019cse427/Group10>

**Submitted to:**

**Shaikh Shawon Arefin Shimon**

Lecturer,

Department of Electrical and Computer Engineering

North South University

Table of Contents

[Project Description 3](#_Toc7478571)

[Introduction 3](#_Toc7478572)

[Testing Aspects 3](#_Toc7478573)

[Tools/Frameworks Used: 3](#_Toc7478574)

[Description of Input Space Partitioning 4](#_Toc7478575)

[Graph Partitioning 5](#_Toc7478576)

[Class: GameEngine.java 5](#_Toc7478577)

[Methods: 5](#_Toc7478578)

[Class: MainActivity.java 11](#_Toc7478579)

[Methods: 11](#_Toc7478580)

# Project Description

## Introduction

Classic games that were once made for desktop computers at first are now commonly developed for smartphones. Android Operating System is highly used in modern smartphones. In this project, testing of one of such classic games on Android known as Tic-Tac-Toe has been done. The application has been developed using Java Programming Language in backend hence testing was done in Java Programming Language.

The project that has been tested by me was initially developed and available publically at: <https://www.ssaurel.com/blog/learn-to-create-a-tic-tac-toe-game-for-android/>

## Testing Aspects

The following testing aspects have been implemented:

* Unit Testing methods
* Input Space Partitioning
* Graph Partitioning
* Integration Testing
* UI Testing

## Tools/Frameworks Used:

* Android Studio
* JUnit4
* Espresso (For UI Testing)

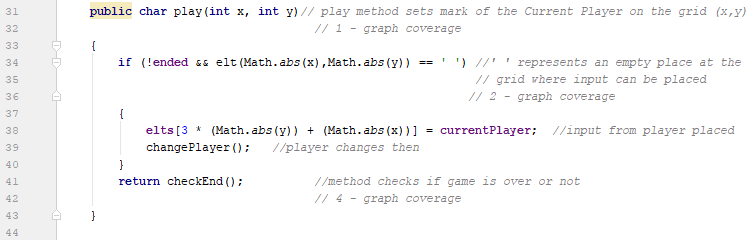
# Description of Input Space Partitioning

# Graph Partitioning

## Class: GameEngine.java

### Methods:

#### play (int x, int y)



A

B

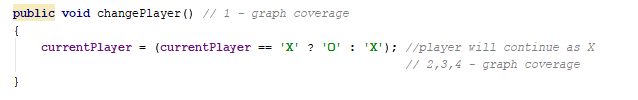
C

A !A

B

C

#### changePlayer ( )



C

A

B

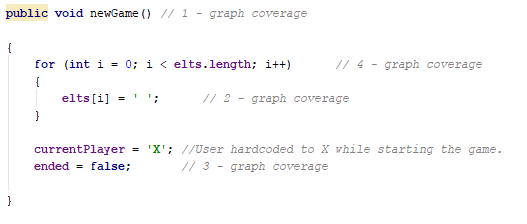
A B

C

C

Hardcoded to X

#### newGame ( )



D

B

C

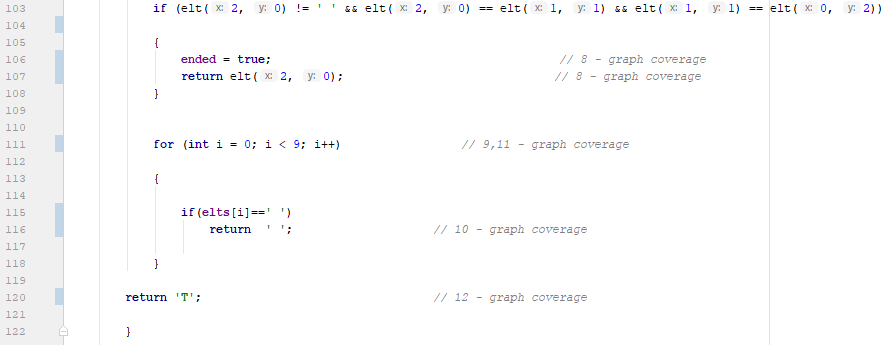
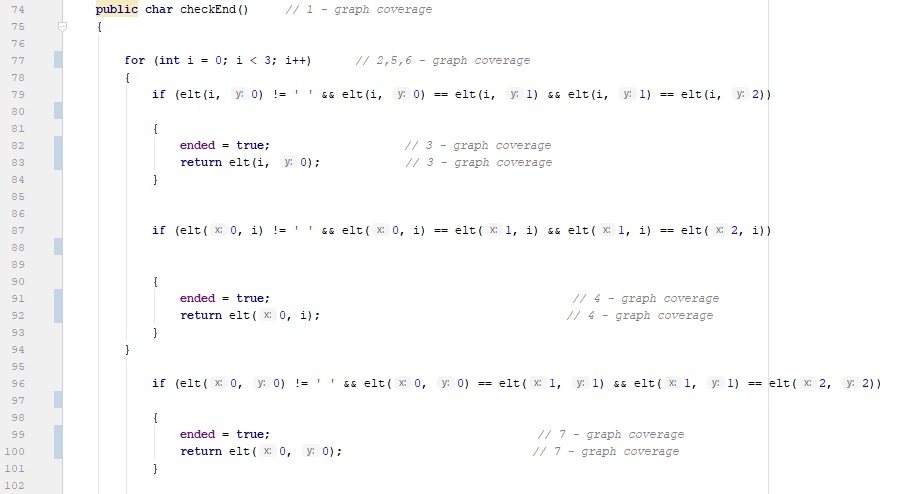
A

A !A

B D

C

#### checkEnd ( )



P

O

N

M

L

K

A

J

I

H

G

F

E

D

C

B

A H J L

B E I K M !M

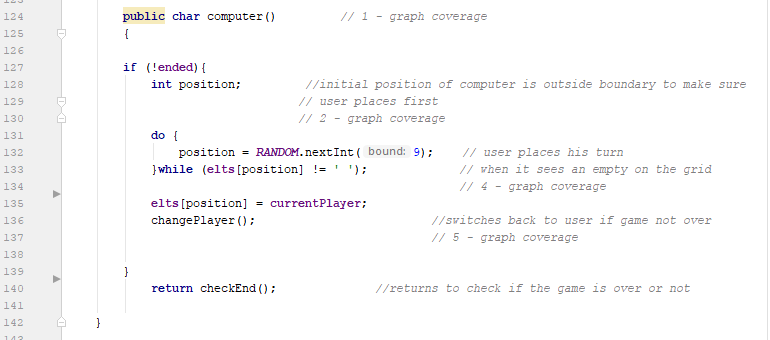
C F

N P

D G

O

#### computer ( )



D

F

E

C

B

A

A !A

B D

C D

F

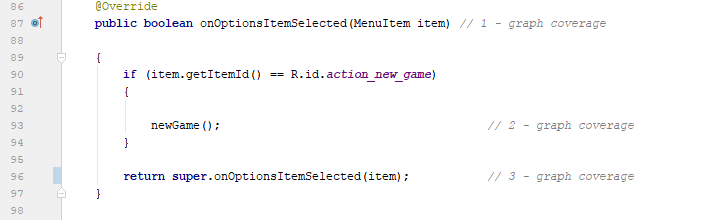
D!

E

## Class: MainActivity.java

### Methods:

#### onOptionsItemSelected (MainItem item)



C

B

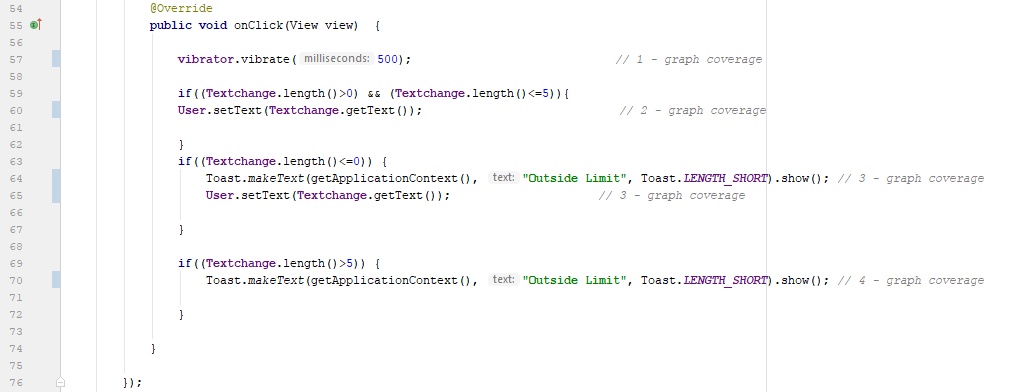
A

A !A

B C

#### 

#### OnClick (View view)



F

G

D

A

E

C

B

A

B D F

C G  
 E