Contents

1. Introduction
2. Control flow of the app
   1. Control flow of each activity fragment
3. Summary/reference of each class

Purpose of this document

The purpose of this document is to describe, easily and accessibly, the inner workings of the Onerous App. This is to allow future developers to easily be able to maintain the code, add features/functions, upgrade the UI and perhaps even port the app to another platform such as iOS.

What this document will not do is explain the Android or Java framework. It will not detail how methods or functions operate but will summarise their job and input/outputs. If further technical detail is required, the commented source code can be found in the project files.

Control Flow

General overview

This a repeatable, three-step process is a typical way a user could navigate the app whilst operating it.

Variables Tab

Simulation Tab

Results Tab

Below is a brief summary of what role each of the three tabs fulfil in the overall operation of the app.

1. Variables Tab: allows the user to view the current simulation variables and to change them if desired.
2. Simulation Tab: allows the user to begin the simulation with the set variables. They can choose to view the graphics and animations or choose to toggle them off for an instant run.
3. Results Tab: allows viewing of the statistics and graphs generated from the most recent simulation.

**The Variables Tab**

**Class references**

**ActivityDistribution**

The ActivityDistribution class is a user defined type that holds values that will be used to calculate how long each ‘activity’ (e.g. engine removal, engine refit, bad engine transit etc.) takes.

**EventData**

The EventData class is a user defined type that is used to hold two bits of data: the time of an ‘event’ and the ‘kind’ of event.

**Events**

The class Events contains the methods used for performing the actions required when a certain scheduled event occurs.

**MainActivity**

The MainActivity class is the very first class that launches when the app is launched. It is responsible for initialising the UI view and setting up the tabbed fragments that will hold the majority of the UI elements.

**PlaceholderFragment**

Tabbed placeholder fragment that can hold the place of a tab in a tabbed activity.

**ResultsData**

The ResultsData is a user defined type that is used to hold data and statistics when the simulation runs.

**ResultsFragment**

The ResultsFragment class initialises the UI elements of the ‘Results’ tab, and contains the methods required for populating the results table and the graph with the appropriate data.

**SimAnimations**

The SimAnimations class contains all the methods for performing the various animations when visually displaying the simulation running.

**SimMethods**

The SimMethods class contains much of the backbone for actually simulating the scenario.

**SimulationFragment**

The SimulationFragment class initialises the UI elements of the “Simulation” tab.

**Stats**

**Variables**

**VariablesFragment**