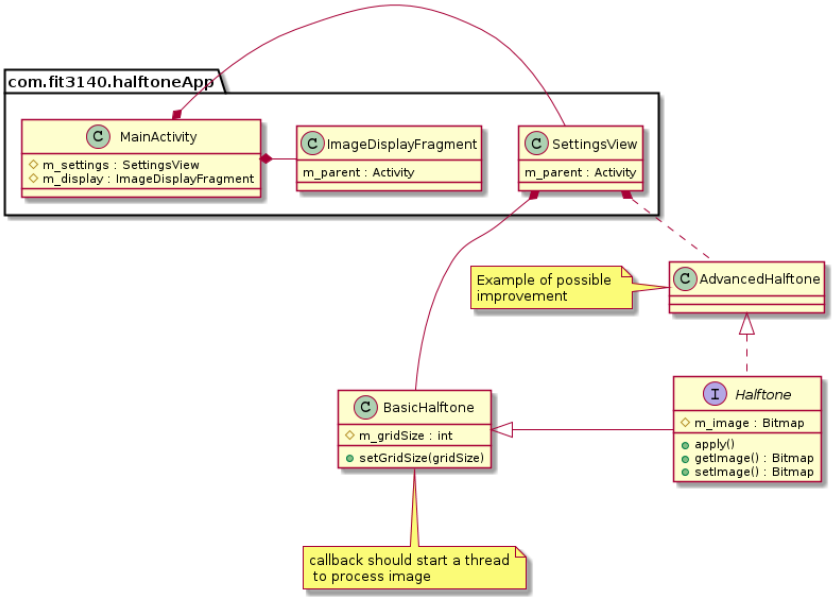
uml

Jack Hosemans

April 25, 2014

Contents

1 UML diagram



2 Descriptions of classes

2.1 MainActivity

2.2 ImageDisplayFragment

This fragment should show the user images before and after halftoning.

Right now it does not matter how it is done, but something nice should be made so that swapping back to the original image should be painless and automatic when settings are changed.

2.3 SettingsView

This is what displays the settings for the current halftoning mode.

Should have sliders or number input so that the user can change the settings for halftoning.

Probably the best way to do this will be a ViewPager so that swapping between the different setting screens will be easy.

2.4 Halftone

This is an interface that all halftoning methods should implement. Basic stuff.

2.5 BasicHalftone

This is the meat of the app, applys the halftoning that we all know and love(?).

Should be able so set the grid size, there really shouldn't be much more than that in this implementation.

Side note:

2.5.1 Proposal:

Halftoning usually takes a long time (>10s for large images.) A solution should be made for this.

- multiple threads could be used to work around the slowness of the application.
  - This would give the advantage of not slowing down the GUI
  - However, it would be harder to code. Maybe another spike is in order for implementation?
- A hard limit on the grid size would also be sufficient
  - "worse" than the threaded way because the application would appear to freeze when being halftoned. This is VERY bad.
  - Worse output due to each dot being bigger than any other way.
  - Not using the full power of the processor (We need all we can get)