

Updating the Font Set in iDOCENT Application Client Package and Eclipse IDE

Andrew White

Thursday, December 8th, 2011

Abstract

The iDOCENT application can be downloaded onto an Android OS smart phone via USB cable. Within this application file are numerous coding files, as well as image data. The images files in mind, the r*.png font files for OpenGL masking of room numbers and other identifiers, are located in the ./res/drawable/ folder. OpenGL stands for Open Graphics Library and is a low-level Application Programming Interface (API) for embedded graphics. OpenGL can draw lines and triangles only, but any structure can actually be created from these primitives.

While using OpenGL and the iDOCENT code, these r*.png font files must be dimensions of powers of two for the length and the width; chosen dimensions for the masking in iDOCENT are 16x64. These files can be updated by replacing the r*.png files with newly created files. After reloading the project into Eclipse and running the application identifier as Android, the Emulator will exhibit the navigation map with the new font set.

Steps

First, the interested should download the Eclipse IDE and the Android SDK v2.1. With these tools installed, the project folder iDOCENT can be rooted by importing an existing project, pointing to the iDOCENT directory. Using the side explorer frame, the folder ./res/drawable/ can be opened and its content viewed.

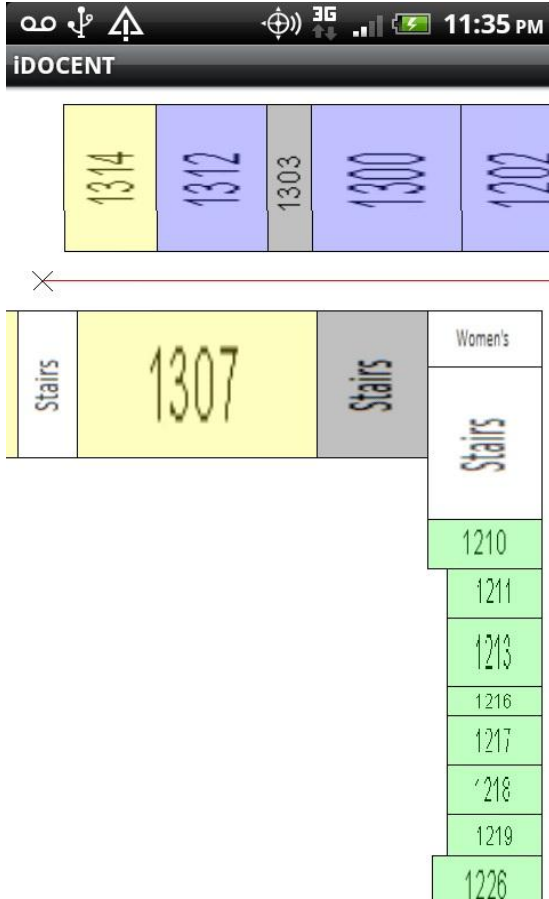
Opening the root directory in Windows Explorer, one can navigate to the same folder. This folder will contain the same information as the folder viewed from Eclipse. In it are 200 files, 199 of which are .png room images.

Examples:

1235 1237 Elevator Men's 1233 D

One will notice that the font choice is not uniform. The goal is to set a new font standard to replace all of the existing room number .pngs. By opening each of these files and rasterizing a new font that will fit in the same dimensions, which are 16x64 in this case. In the mask, any white in the image will be transparent, and any black will show. Any type of transparency in the .png files themselves will cause a memory allocation error, since OpenGL prefers masking rather than using inherent transparency qualities of .png images.

Here is a screen capture of the iDOCENT application using this image set:

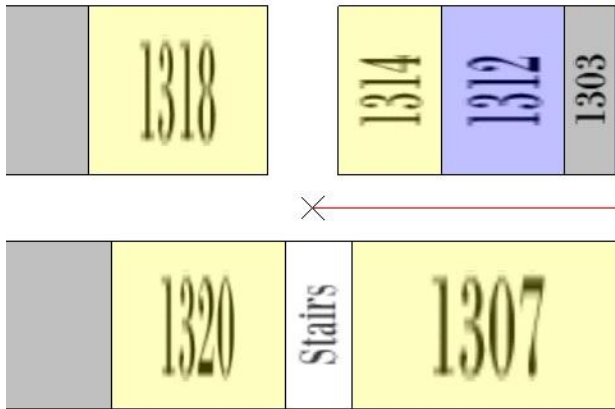
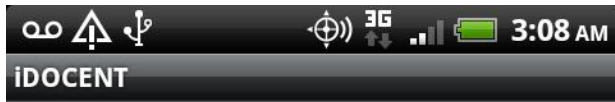


After the new .png image set has been created, they can be saved to replace every existing .png in the ./res/drawable/ folder. Our font now looks like this:

1235 1237 Elevator MENS 1233D

The new font .pngs are now uniform, adding a necessary aesthetic improvement.

Here is a screen capture of the new font set in iDOCENT:



Conclusion

The new font set updates the program to appeal to more users, hence adding to its universal design. The font set previously was not uniform in typeface or font size, which is a distraction for most users.

To further update the application to add aesthetic appeal, one could create thicker lines. This is done still by the OpenGL framework using the line primitives. A thicker line, however, requires the drawing of rectangles, and using commands to fill them in. One might expect there to exist a simple command to thicken the lines, but OpenGL works on primitives, so this is not the case. The method for increasing thickness does require some geometric aspects that can be worked out with some basic knowledge of how to draw using the Vertex method, as used in the iDOCENT application code.