Greg Beca
Daniel Sheaffer
Andy Vanek

Brief details about our project

Team 4 decided to create the basic image editor using C# as the primary programming language and Windows Forms in Visual Studio as the primary GUI creator. In using these two items in tandem, we feel that the creation of the GUI will not be the most time-consuming part of the project due to the power of Windows Forms. Instead, we can focus our time and efforts on creating quality image manipulation algorithms, enhancing the usability and feature-richness of the end product.

Shopped Subversion Information

Our project currently has a Google Code page which can be found here:

http://code.google.com/p/capstone2009/

Here you can find the basic details of our project, a wiki, an issues page (similar to Trac, TargetProcess, etc.) and a subversion repository to check-out and view our current code base. To check out our code, you can do the following with subversion:

```
svn checkout http://capstone2009.googlecode.com/svn/trunk/
```

The trunk currently has two directories in it. The CapstoneF2009_Test directory contains our "spike" (proof-of-concept) for the project, in this case a very basic GUI that opens and saves an image. The shopped_iteration2 directory contains our current iteration (milestone) code base.

Goals for this iteration

Our primary goal for this milestone/iteration is to add undo/redo functionality to our program. As with any image editing program the undo/redo feature is essential. We have an idea of how to implement this effectively, but no code has been written yet.

Another goal we have in mind is to add basic filters to our program. Some ideas we had were the manipulation of the colors in the image (e.g. grayscale conversion) and adjusting the exposure of the image (brightness and contrast).

Member Contribution

We still believe that this project is not quite big enough to work on individually because of the lack of independent parts. If two people were working on this project at the same time there could either be code corruption or repeated effort. We are currently using the "pair programming" paradigm. Andy has been the main coder while Greg and Dan research possible algorithms for our current focus. We have also found that having someone watch over the person typing the code leads to better, more efficient code, and less time spent debugging simple problems because they were caught before compilation.