



Anne Lam &lt;anne.v.lam@gmail.com&gt;

---

## Unit Testing

2 messages

---

**Mando Garcia** <garcia.mando@yahoo.com>

Mon, Apr 27, 2015 at 6:11 PM

Reply-To: Mando Garcia <garcia.mando@yahoo.com>

To: Sanan Aamir <sananaamir@gmail.com>, Anne Lam <anne.v.lam@gmail.com>, Hieu Tran <hdtran89@yahoo.com>, James Rowe <jrrowe55@yahoo.com>

Hey guys,

So Anne, Sanan and I briefly discussed in Software Engineering what we expect to have completed before our next meeting held on Wednesday. The main points of discussion was which framework we should use for Unit Testing, Hieu's email, and the "Software Engineering Minutes" document that Dr. Stringfellow is adamant about us keeping track of.

**Point #1) Unit Testing...** this is our last week to have this project completed so it would be wise of us to use *NUnit* being that Anne's presentation is a good resource for us to use, and we were assigned homework which covered it. Therefore, we all have some type of knowledge of how to use the *NUnit* framework. On Wednesday, those of us that are doing Unit Testing should bring a list of test they believe should be completed, or we can use *Google Docs* to keep track of this information. If anyone feels as though we should use a different framework respond back to this email with the framework you believe will be more beneficial.

**Point #2) Hieu's email**

So we were all a bit confused about the first test Hieu tried that broke the code. It is shown below..

"The application broke when i try to put in a different directory with .bmp images or .tif images. I created a test folder and added a test image extension .tif and another folder with .bmp"

First and foremost, I think we all can agree that nothing the user enters into our tool should break the code...

After talking to Hieu, he tried random images that DID NOT include a needle and a base, and as expected the code broke since our tool tried to process an image that was not in the scope of the data. So the question is.. should we leave it up to the user to enter the correct set of data, or should/can we have the tool only process the "correct images" within the remaining time allotted? By "correct images" I mean images containing a needle and a base.

Every other test in his email, we should be able to fix.

**Point #3) Software Engineering Minutes**

I speak for myself when I say I haven't been keeping up with this document and to be honest today was my first time seeing it ); heck I couldn't even find it. By the way it is located in this directory --->

***ImageProcessing/A Team Documents/Software Engineering Minutes***

If anyone can think of a day and time they met with another group member, it would be nice to have it in writing by Wednesday so we can add it to this document. If not then it's all good.

The format is shown below...

**Date:**

**Time:**

**Location/Communication via:**

**Team Members Present:**

**Agenda/Discussion:****This project is almost complete!! Good Job A Team!**

Thanks,

Romando Garcia

p.s. I won't be in class Wednesday so I'll have my work completed by tomorrow

---

**Anne Lam** <anne.v.lam@gmail.com>

Mon, Apr 27, 2015 at 8:51 PM

To: Mando Garcia &lt;garcia.mando@yahoo.com&gt;

Cc: Sanan Aamir &lt;sananaamir@gmail.com&gt;, Hieu Tran &lt;hdtran89@yahoo.com&gt;, James Rowe &lt;jrrowe55@yahoo.com&gt;

What a wonderful summarizing email! Great interim leadership, Mando!

I think I fixed all issues in Hieu's email except the tif, bmp exception stuff.

I also produced code to create the processed images in a new folder that is located in the same place as the selected images folder, with "\_processed" appended to it. Problem is, it drastically slows down at the end because of this.. See the code at...

```
Parallel.ForEach(dropletImages, dropletImage =>
{
    dropletImage.DetermineCentroid();
    /*****COMMENT THE FOLLOWING LINES IF YOU DONT WANT TO CREATE
IMAGES****/
    //path for newImageFile = "newDirectory(originalFolderName_processed)" + originalImageName
    string newImageFile = newDirectory + "/" + dropletImage.GetImageName();
    CreateProcessedImageFile(newImageFile);
    //Save the processed image to newImageFile
    dropletImage.GetBlackWhitelImage().Save(newImageFile);

});
```

So I may need some help improving the speed on that. Images look pretty good for the most part - the circumference is now showing in **black** around the drop, but some extra **red** lines near the needle show for some images. However that doesn't affect the calculations, since the circumference points are the only points used to calculate centroid.

I added this images to Github - sorry for the download time on your next pull! but I thought y'all might want to see what this new code does so we can improve it, get feedback.

And lastly, that's my bad on the minutes thing, I never mentioned it... Haha, but if you can, do note any work you've done. I'm not sure what all she wants, but she did want us to discuss bugs discovered, maybe solutions found, idk.. something like that.

Anne

Go ATeam

[Quoted text hidden]