

CSCE 117: Project 2 ~ The Magical Pointillist Bird Gallery

You walk through the imposing iron gates, as purple mist swirls in the courtyard, and climb the 24 steps to ring the heavy knocker on the front door of the Gothic Victorian mansion. The door opens, as if by magic, and a commanding female voice summons you to enter. From the foyer, through a half-closed door to the right, you can see a library room cluttered with stacks and stacks of images and avian photographs. The only living creature you can see is a blue point Siamese cat curled up on the mantel. The disembodied voice continues:

“Greetings, young wizard. Were you the one who cast a spell that launched a cottage industry of magical owl racing in the nearby village?”

You shuffle your feet modestly. *“Yes, it was I. It didn't violate any magical gambling regulations did it?”*

“Well actually, there could be some trouble there. But I will put in a good word with the Wizard Council if you will do me a favor. I have a task for you. As you can see, I have a large collection of important bird photographs. But recently, I have become disenchanted with photorealism, and hunger for the beauty of impressionist art. Can you compose a spell that will create impressionist art out of each of my bird images?”

You bite your lower lip, and try to remember your art history so you don't look like an uncultured fool¹ – impressionism? Was that Monet or Manet? Or both? And wasn't there a weekend beach scene with a bunch of fancily dressed people, and a monkey? Seurat? Yeah, that sounds right. Using lots of little spots of color? Could I try, maybe – pointillism²?

“Well, I'm not so sure about Impressionism, but I think I could manage a Pointillism spell. Would that do?”

“Well, technically Pointillism would be Neo-Impressionism, but let's not quibble over details – it still sounds splendid. Can you start right away?”

Assignment Specification:

You will need to compose two spells (programs):

Program 1: ArtMaker.py

- This will provide a function to convert square GIF images into a specific text file format that describes the placement of many colored circles that is a “pointillist” version of that image.
- This will involve opening an image, repeatedly choosing random pixels from it, and **writing** the location and color of the pixel as one line of the file.

Program 2: ArtViewer.py

- This program will provide a function that opens a specified text file, and draws the pointillist art in a graphical window.
- This will involve reading from a text file that was created by ArtMaker.py, and using each line of the file to place a colored circle in the graphics window.

1 If there's one thing a wizard doesn't like to appear, it's an uncultured fool. Especially true of wizards that attended liberal arts colleges.

2 If you're feeling completely at sea here, you might want to read: <http://en.wikipedia.org/wiki/Pointillism> and http://en.wikipedia.org/wiki/A_Sunday_Afternoon_on_the_Island_of_La_Grande_Jatte. It will help prevent some embarrassment at cocktail parties later on in your life.

Here is the precise text file format that I would like your ArtMaker.py program to write, and your ArtViewer.py program to read.

<code><x1> <y1> <radius1> <red1> <green1> <blue1></code>	<i>← first circle's location, size, and color</i>
<code><x2> <y2> <radius2> <red2> <green2> <blue2></code>	<i>← second circle's location, size, and color</i>
<code><x3> <y3> <radius3> <red3> <green3> <blue3></code>	<i>← third circle's location, size, and color</i>
<code>...</code>	

Don't include the angle brackets! Just put 6 numbers on each line, separated by spaces.

I have included a few example data files, so you can test your ArtViewer.py before you write your ArtMaker.py, if you want. These text filenames should always end with the “.art” file extension.

You are also provided with a large number of source images of birds from the magic mansion gallery³, but you can also use Photoshop or the Gimp (a free Photoshop alternative – Google it) to convert any image you like into GIF format, and make artwork out of your own images!

ArtMaker.py should have a function, defined as follows:

```
def convert(imageFileName, numCircles):  
    #Fill in here
```

Then inside your main() function (in ArtMaker.py) you should be able to call:

```
convert("gallery/bird184.gif", 5000)
```

and it should create (or overwrite) a text file called “bird184.art” in the “gallery” subfolder, with 5000 lines of text, where each line specifies one random point from the image “bird184.gif”.

Note 1: Since all pixels are 1x1, to create better art, pick a random radius from 2 to 8 for each circle.

Note 2: You may assume that all input GIF images will be square, but not necessarily 200x200. You should scale your location output in “birds184.art” so that it uses the full 600x600 canvas that ArtViewer.py will be painting it on.

ArtViewer.py should have a function, defined as follows:

```
def display(artFileName):  
    #Fill in here
```

which inside your main() function (in ArtViewer.py) should be called like:

```
display("gallery/bird184.art")
```

and a 600x600 graphics window with a black background should appear, with the pointillist artwork (colored circles) displayed beautifully upon it.

³ Okay, actually these images are all from photographs of birds taken by my wife, Dr. Susa Stonedahl, during 2013. If you want to see the bigger more beautiful versions, go to: <http://susa.stonedahl.com/mybirds>. (I've taken a lot of bird pictures too, but I haven't had time to organize them properly!)

Useful Spell Ingredients:

Even though our textbook doesn't include information about it, the Zelle graphics library **does** allow us to read in GIF (but not JPG or PNG) images. It's described in the online documentation here:

<http://mcs.wartburg.edu/zelle/python/graphics/graphics/node12.html>

You'll want to look at the **getWidth()**, **getHeight()** and **getPixel()** methods.

You should NOT need to use the `setPixel()` or `save()` methods.

ArtMaker.py shouldn't need to display a graphical window at all. It just needs to create an image object, and get the appropriate data out of it, and write that data into a file.

ArtViewer.py will need to use the **color_rgb()** function.

Tip: It looks more Pointillist if the circles don't have black borders around them. How can you fix that?

General advice: Get small pieces of your code working one at a time. Test each piece as you go!

Advanced Spell Crafting

Bonus task 1: Necessary (but not guaranteed sufficient) condition for A grade work.

Provide more artistic rendering modes besides pointillism, for ArtViewer.py. (But don't change the .art file format.) You could play with shape, color, etc. The user should be able to choose between modes!

Bonus task 2: Not necessary, some possibility for extra credit. Or maybe just for the glory.

To impress me with your wizardly prowess, write a third program "**BatchArt.py**" that automatically goes through every ".gif" image in the subfolder called "gallery", and converts each one into a ".art" file, by repeatedly calling the `convert()` method inside **ArtMaker.py** (and you must not copy and paste the code from `convert()` into `BatchArt.py` – just call the function!)

Bonus task 3: Not necessary, some possibility for extra credit. Or maybe just for the glory.

To further impress me with your wizardly prowess, make another program, **ArtSlideShow.py**, that looks inside a subfolder called "gallery" for all the ".art" files, and displays them one-by-one in a randomly-ordered slideshow (by repeatedly **calling** the `display()` function from `ArtViewer.py`).

Note: I **haven't yet** given you all of the magical tools you would need to accomplish the latter two bonus tasks, so you would need to learn some new things on your own. Don't attempt these two unless you get everything else working. *(We will likely take a poke at these two bonus tasks as a class, after the assignment is turned in. But you're welcome to try it on your own first!)*

BEFORE YOU SUBMIT: Fill out the provided template header comment, similar to before!

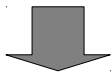
ALSO, submit all .py files, along with a couple of your favorite example art pieces.

Due: 8:30 A.M. Friday, Oct. 4

Grade value: Worth 20 points, like the last one.

Artistic value of a bunch of auto-generated impressionist paintings of birds: Priceless!

Illustrative example:



**.art
text file**

```
x y radius r g b  
x y radius r g b  
x y radius r g b
```

