Sarah Andrews
ASIM 1310 Final Project: Musical Sky

Because seemingly everyone is doing the Snake Game for their project, I decided to do something different and go with the Interactive Algorithmic Art project topic. Originally I wanted to do something with the Samchillian project topic that involved an aurora that changed size as you played a key, but I couldn't figure out how to make it work correctly without looking up another person's code on the web. So I decided to do the simpler option and do the Interactive Algorithmic Art.

For this project, I had to look back a lot on the demos we did involving classes, midi handler, and PImages. As you will see in the code, the majority of my visuals—except for the stars—are all PImages; each image I created using GIMP. In order to make sure all the images in the setup fit together, I created a duplicate of the sketch in GIMP and created each PImage on a different layer. Once each layer was complete, I copied and pasted each layer as its own image and then exported them as a PNG file (except for the background which is a JPEG). I actually went through a few slightly different design ideas before coming to the final product; the main difference being the foreground image in front of the mountains. At first, the foreground was going to be a guy sitting on a dock in front of his cabin fishing in a lake--but as you can probably image, this idea was a little too much for the amount of time I had to work on this project. Then I thought about a guy looking into his telescope (like a little video game sprite), but I didn't have the patience to make all of the frames, and then my idea was to have a glowing tent on the ledge. In the tent idea, the tent was supposed to have one image that had no glow, then have a second image on top with a light and the image was meant to fade in and out (similar to what I ended up doing for my image slideshow project). However, the tint function I used caused both images to fade away at the same time, and whenever I tried to fix it, either both would disappear, or not appear at all. So, in the end, I decided just to go with a sign that lit up whenever the mouse was over it.

Now because we didn't cover stars in class, I had to borrow some code from the Processing website in order to get the shape I wanted (a link to the page is located in the code's Star Class). Now as for the midi handler code and the MidiUtils.java file, I used the one that you provided for us in the making sounds demo a few months back; and all I did was copy and paste the files in there and left them as they are since you already put your name into them.

(Honestly, I was rather hesitant to use the Midi stuff because I knew the majority of our code had to be our own, so I tried to search the web for other sounds that I could use instead, but unfortunately I wasn't able to find anything that I liked for the sounds. So in the end, I decided to use the Midi instead.)

Instructions for the program:

- For the sound, set the SimpleSynth instrument to "Crystal"
- When running the program:
  - There should be a gradient background with a moon in the top-right corner in front of it, along with a mountain range covering the lower half of the sketch, and a ledge with a sign post reading "Fill the sky with stars!" in black lettering
    - If the user hovers the mouse over the image of the sign and the ledge, the lettering on the sign changes to a cyan color
    - (the moon and the mountains remain the same)
  - Also appearing between the ledge and the mountains are two clouds running at different speeds across the screen. Like the moon and the mountains, these clouds are non-interactive and will just go across the screen and reappear again on the other side and repeat.
- To fill the sky with stars, simply press any of the letter keys (any letter in the alphabet from a to z will work)

    - When pressed, two stars (one solid and one transparent) will appear on top of each other rotating in opposite directions. Each star is accompanied by a random pitch between 60 and 100.
    - After a while each star will slowly fade away, giving the user a chance to fill the sky all over again

    - *Note: the stars are placed behind the moon and the mountains in the code, so if the user pressed a letter key and doesn't show, chances are the star is probably behind either the moon or the tops of the mountains*

- As a bonus, if the user wishes to add some color to the sketch, all they have to do is press a number key (keys 1-9 work, not 0) and an aurora will appear on top of the stars, mountains, and moon
    - The auroras only appear in one set Y coordinate, so if the user presses a number key to add another aurora, the auroras will overlap each other to create different mixes of color

        - *Note: due to the width of the aurora's image file, the user should keep in mind not to have a lot of auroras in at once ("a lot" being roughly over 3 or 4 auroras) because it will slow the program down. So be careful!*

    - Now unlike the stars, the auroras do not make a sound nor will they fade over time. Instead, they will continue to roam across the screen just like the clouds
        - If the user wishes to remove the auroras, they can either rerun the program again or press the backspace key to remove one aurora at a time.

Have Fun!