Assignment #1
Computational Urban Science Workshop, Spring 2019

Due: 9:00am, Thursday, February 21th

Goals: Create a piece of animated, digital art that runs indefinitely and responds to user inputs. Briefly present it at next class!

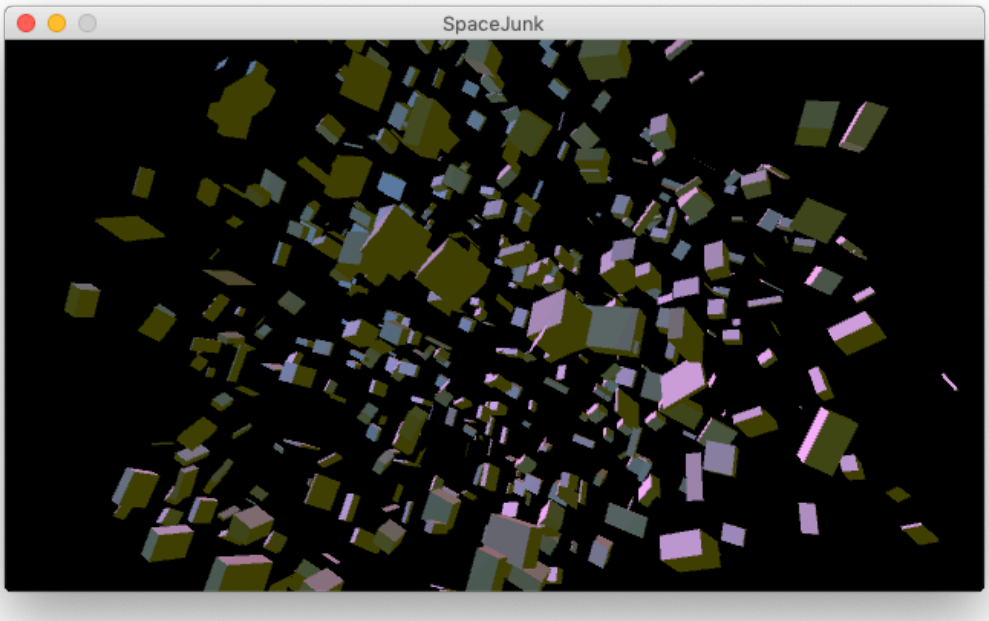

Figure 1. "Space Junk," an example provided by Processing

Background: The purpose of this assignment is to get you comfortable with some of the basics of Processing while saving and submitting your work to GitHub. You have a lot of discretion to create your own artistic piece, so have fun. Feel free to get ideas by browsing examples that are included with Processing: "*File > Examples …*" The processing.org reference library is also a great place to become acquainted with all of Processing's capabilities:

https://processing.org/reference/

Minimum Requirements: Your piece of art should utilize, at a minimum, the following elements:

- Screen Resolution: Use a screen resolution of 640 x 360 by using the "**size(640, 360)**" command in **"void setup() {**"

- Mouse Input: dynamically change your piece of art using the mouse in some way. For Example, you may use the '**mouseX**' and '**mouseY**' variables to recall the mouse position on the canvas at any point in time. Here are some other functions you might use:

  Mouse
  mouseButton
  mouseClicked()
  mouseDragged()
  mouseMoved()
  mousePressed()
  mousePressed
  mouseReleased()
  mouseWheel()
  mouseX
  mouseY

- Keyboard Input: dynamically change your art using at least one key command. This is often done by programing a key command to turn a Boolean variable (i.e. true/false) on and off. Other examples:

  Keyboard
  key
  keyCode
  keyPressed()
  keyPressed
  keyReleased()
  keyTyped()

- Output Statistics: On your canvas, represent dynamic quantitative statistics for at least 2 parameters of your art piece (e.g. number of particles, average velocity, etc …)

- Frame Rate: Maintain frame rate of 60 frames per second (FPS). Check your framerate with the '**frameRate**'. By default, processing tries to run at 60fps, but will slow down if your script is too 'heavy.'

Submission Directions: Locally In your GitHub repository folder (i.e. cusw-spr19-lastName), create a folder called "*Assignment_1*". Save your Processing script to this folder. For example, if I created a Processing script called **Apples** and I saved it to this folder, the folder structure would look like this:

Github/*cusw-spr19-winder/Assignment_1/Apples/Apples.pde*

To submit your code online, use the Github Desktop app:

(1) Navigate to your repository, you should see changes summarized
(2) **Commit** your changes
(3) **Sync** or **Push** your commits to github.com