Comp Graphics

I. Lecture 2:
   a. What is in the Elegoo Kit
      i. Mega 2560 board has a lot of pins
      ii. Blue extension cord
         1. Has 2 ports one goes into the computer and the other goes into the board
         2. You can also connect to a power source and we will do some examples in the future
   b. What we will do today
      i. We will only be going over the board today in detail so we can clear a few myths.
   c. Arduino
      i. Go to the Arduino website (www.arduino.cc)
      ii. Click learn Arduino
      iii. Install the Arduino IDE that applies to your computer system
      iv. You will get a zip file that you will have to extract. The instructions will be provided for you in photos present on the Arduino web page.
      v. The IDE page
         1. www.arduino.cc/em/Guide/Linux (page Professor was on in class)
         2. Make sure that scroll and do the **please read** section
      vi. ls –l /dev/tty
   d. Once Installed
      i. Open it up
      ii. It will display a framework that presents two functions that you will have to write. As long as you understand c you should be able to write the code
         1. The setup function will allocate any memory that you want
         2. The loop function runs for all points in time the same function repeatedly
      iii. The board is configured that you will run the same function over and over again
      iv. Take note of the verify button (check mark) which is the compiler for the Arduino board
         1. If you have problems, this button should help with finding errors with your code
      v. Nothing is bullet proof your Arduino board can burn so don't burn your board.
         1. This is caused if you put too much current into your board.
      vi. The arrow button next to the check mark is what you click to upload the sketch onto the board. Once uploaded the sketch will keep running until the board runs out of power
      vii. File (menu option)
         1. Examples

a. There are built in examples that will give you examples that you can play around with to get to know your board.
b. The sketches are easy
    i. LED_BUILTIN
        1. Refers to the built in LED
    ii. OUTPUT
        1. Means some information will be displayed

e. On the board
- i. You will see different pins that for example show 5 volts make sure that you are paying attention to these. There is a max current for the board.
- ii. Pass a resistor to damp out the current that goes to your board.
- iii. We will do calculations that will give us the amount of current that we should be passing through the board.
- iv. Do not touch wires without knowing what you are touching
- v. There are 3 LEDs on board
    1. TX
    2. RX
        a. Both will glow when you upload a sketch
            i. Any code you upload to the board is called a **sketch**
    3. L
        a. This is something that you can control
        b. It will glow when there is power

II. To get started on your own

a. Plug in the board to the laptop via the USB port
- i. The LED blinks once connected
    1. May be preconfigured with the *blink* sketch
- ii. There is also an ON light which lets you know that the board has power

b. You will have to configure your board before you can upload sketches to it

c. File – board – click mega 2560

d. You can see in tools drop down the new board name and port

e. Configuration
- i. ls –l /dev/tty
- ii. ttyACM0 is what the port should be configured to

f. Log out and then log back in to your computer to view the changes

g. Once logged back in you will see the empty methods but the port should be changed

h. You should see done compiling and sketch uses 0% of program storage space at the bottom of the window

III. Uploading sketch on the board

a. If you upload a blank sketch the blinking will stop and the LED (L) will constantly be on.
- i. You can upload the blink sketch to start the blinking again.

b. You will see that the TX and RX will blink quickly to show that the sketch is now on the board

c. If you lose power and then add power back to the board the resulting behavior is unexpected, you might need to upload the sketch again.

d. When you build your circuit, make sure that there is no dangling. We are trying to avoid all cases that will fry the circuit.

IV. In class example

a. Setup ()

i. pinMode (LED_BUILTIN, OUTPUT)

b. Loop ()

i. digitalWrite (LED_BUILTIN, LOW)

1. LOW here is the same as off

ii. delay (1000)

c. Make sure to compile to make sure there are no errors then save the sketch before you upload it!!!

d. The Arduino will try to keep a refresh rate there is a built in refresh rate you can speed things up but if it is too fast, the Arduino will try to stick to the built in rate.

NOTES ON PAGE BELOW

Introduction to Basic Graphics Programming

I. File Formats
   a. Obj File
      i. Used for storing meshes
      ii. Comes with a list of note coordinates for your model
      iii. If you keep scrolling, you will see texture coordinates that go on top of the model
      iv. When you see the f this means face, these are cords
         1. First index
            a. Vertex
         2. Second index
            a. Texture
         3. Third index
            a. Normal

II. Basic Display Capabilities
   a. Opengl
      i. Written by the professor from scratch
         1. Lamborghini visualization
            a. A window that opens up and displays the object.
            b. There are different displays
               i. The ability to zoom in and zoom out
               ii. Move the object
               iii. Do an orbital spin
                  1. Must be coded
      ii. We will learn how to show this kind of thing and interact with it

III. Hello World example
   a. Opengl
      i. Graphical framework that lets you create your own GUIs to display your graphics
      ii. There is a big difference between Opengl 2.0 and 3.0 and beyond
         1. 3.0 is supported on most systems.
   b. Explanation of code (Code will be present on the class website)
      i. Setup
      ii. Check to see if memory is fine
      iii. Make it so that when you press exit the window will close
      iv. While loop is when the magic happens
         1. It opens the window that will display until you tell it to go away.

IV. Tools you may use for term project and beyond
   a. Blender www.blender.org
      i. Getting started is simple all you need to download and you can start playing with it.
      ii. This software is versatile
      iii. You can do a lot of modeling and rendering on this site

iv. It comes with suites that come with the software download
    1. Blender render
    2. Cycles render engine
v. Does have some support for simulation.
    1. Particle based fluids
    2. Scalable fluid based simulations are not available
vi. Better for rendering

b. In graphics get ready to get your hands dirty
c. There is nothing that works all the time or in every situation If you are trying to push something out there is some possibility that something will go wrong.
d. Graphic designers are always dealing with innovative things
e. Autodesk Maya (www.autodesk.com)
    i. Similar to blender
    ii. Heavily used for graphics
    iii. Free for students
    iv. Will be used on ilab machines
f. Houdini (www.sidefx.com)
    i. Can do a lot of fluid simulation
g. Renderman (https://renderman.pixar.com)
    i. This is open source
        1. BUT you can only install it once on one laptop for education license
    ii. There is a plug in specifically for Maya

V. Term Project
a. How is the data displayed?
b. How is the data created?
    i. We will get some exposure to physics simulation but not that deep
    ii. Check out some examples on Professor's cs site.