Computer Graphics: LAB ONE

Why computer graphics?

- Rendering 2D and 3D computer graphics
- Medical Imaging
- Games
- VR
- CAD
- Flight Simulator
- Surgical simulators
- Paint programs

It’s not usually something we think about, but computer graphics play a big part in most of our lives. When you’re browsing the internet, playing a game on your phone or enjoying a movie on the television, you’re staring right into the soul of computer graphics.

Computer graphics is an art of drawing pictures, lines, charts, etc using computers with the help of programming. Computer graphics are made up of a number of pixels. Pixel is the smallest graphical picture or unit represented on the computer screen. Basically there are two types of computer graphics based on the interaction, namely .

Interactive Computer Graphics: Interactive Computer Graphics involves a two way communication between computer and user. Here the observer is given some control over the image by providing him with an input device for example the video game controller of the ping pong game. This helps him to signal his request to the computer.

The computer on receiving signals from the input device can modify the displayed picture appropriately. To the user it appears that the picture is changing instantaneously in response to his commands. He can give a series of commands, each one generating a graphical response from the computer. In this way he maintains a conversation, or dialogue, with the computer.

Interactive computer graphics affect our lives in a number of indirect ways. For example, it helps to train the pilots of our airplanes. We can create a flight simulator which may help the pilots to get trained not in a real aircraft but on the grounds at the control of the flight simulator. The flight simulator is a mock up of an aircraft flight deck, containing all the usual controls and surrounded by screens on which we have the projected computer generated views of the terrain visible on take off and landing.

Flight simulators have many advantages over the real aircrafts for training purposes, including fuel savings, safety, and the ability to familiarize the trainee with a large number of the world’s airports.

Non Interactive Computer Graphics: In non interactive computer graphics otherwise known as passive computer graphics. It is computer graphics in which the user does not have any kind of control over the image. Image is merely the product of a static stored program and will work according to the instructions given in the program linearly. The image is totally under the control of program instructions not under the user. Example: screensavers.

We can also divide computer graphics into two based on techniques.

Raster graphics

It’s a safe bet that most people will be more familiar with raster graphics. Also known as bitmaps, raster graphics are composed of individual coloured pixels, each of which contributes to the whole image. If you’ve ever opened an image and zoomed all the way to find it becomes a mess of blurry squares, you’re looking at a raster image.

Raster images are capable of rendering complex images which use a wide range of colors and gradients. This allows the user to create a detailed image with the possibility of maybe even making them look lifelike. The quality of the image will depend on the amount of pixels-per-inch or ppi. If an image has more pixels in an inch, more separate colors can be contained, making the gradient of the overall image appear smoother. Say for example you have an image that is 8ppi, then you’d be able to fit in a more gradual change of colors than an image that is instead 4ppi.

How many pixels are in an inch also affects the quality of an image when scaled up. If you were to zoom in on an image or scale it up to print out larger, an image with a low ppi will appear as a blurry image made up of clearly visible squares or pixels. In turn, an image with a higher ppi can be zoomed in on more before it begins to sacrifice quality.

Vector graphics

Whereas raster graphics depend on pixels, vector graphics use paths made from a mathematical formula. This formula, also known as vectors, tells the path its shape, color and if there is one, its border.

The most noticeable difference between vector graphics when compared to raster is its lack of detail. As vectors are composed of shapes, with each shape having its own color, they cannot work with gradients, shadows or any complex detail. Because of this, vector images are best used for simple designs that have solid colors.

On the plus side, since vectors are mathematical, they can be scaled infinitely without sacrificing quality. This is because as a vector image is scaled up, the mathematical equation recalculates, presenting the image as a crisp clear shot, but bigger. Because of this, vector images are great for things such as logos, simple product artwork and illustrations, but not for detailed painting or photo editing.

What are we gonna do then?

In this course we will try and see everything. From the easy steps of setting up a window to doing shades,3D objects and maybe even a little video game.

First What tools are we using?

- Download pycharm
    - https://www.jetbrains.com/pycharm/download/download-thanks.html?platform=mac&code=PCC
- Once downloaded install the following packages in the pycharm terminal
    - pip install PyOpenGL PyOpenGL_accelerate
    - pip install glfw
        - GLFW is an Open Source, multi-platform library for OpenGL, OpenGL ES and Vulkan development on the desktop. It provides a simple API for creating windows, contexts and surfaces, receiving input and events.
    - pip install numpy
    - pip install pyrr
    - pip install pillow

Create Window

```
#import the glfw
import glfw

#Run Exception for initialization
if not glfw.init():
    raise Exception("glfw can't be initiated")


#Create window (width, height,Title, monitor, share )
window = glfw.create_window(1280, 720, "My OpenGL Window",None,None)
```

```
#Run exception if window not created properly
if not window:
    glfw.terminate()
    raise Exception("glfw window can't be created")

#set window position
glfw.set_window_pos(window,400,200)

#Make context current
#Read about context here: https://open.gl/context
glfw.make_context_current(window)

#The main application loop
while not glfw.window_should_close(window):
    glfw.poll_events()

    glfw.swap_buffers(window)


glfw.terminate()
```

Wanna get excited about Computer Graphics?

Assignment (You don’t have to do it if you don’t want too, But if you do 😉)

Download an app named Blender from (https://www.blender.org/) and create the planet Earth.

Hint: Just import a sphere and play with the texture.

Due Date: Next class

NOTE: Form a group of two and research project idea