Altered Images

Day 1: Colour

Hi. I'm Joel.

j.lewis@csm.arts.ac.uk
http://old.hellicarandlewis.com/altered-images/
@hellicarlewis

You are going to write code, but more importantly think about code first.

Think first. Then do. I find this Very Hard.

Everyone is going to talk about their code and art.

Including me, including you, after every code exercise someone is going to take us through their solution. Bonus marks available for volunteering and creative and actual honesty. It's not a bug it's a feature!

We are in beta.

Fasten seatbelts.
No stupid questions, only stupid answers.

Program 1: First Drawing Program

- What is the most basic drawing program you can think of?

- How many colours would it have?

- How many pixels would it have?

- What is a pixel really?

- How do we check what value the pixel is?

a boolean: something which is either true or false

- http://processing.org/reference/boolean.html

- Only two values

- true (lower case!)

- false (lower case!)

- boolean isPixelBlack = false;

- use "if" to check

- Aside: http://en.wikipedia.org/wiki/George_Boole

an int : a whole number

- http://processing.org/reference/int.html

- Negative and Positive

- int theWidthOfTheScreen = 320;

- ints can make booleans by comparing:
  2 == 2 is true, 2 ==3 is false. Fun to do not equal too: 2 != 3 is true.

- Aside: Do you know what a bit is?

- Aside: A byte?

- Aside: What is 8 in binary? What is 101 in decimal?

a float : a decimal number

- http://processing.org/reference/float.html

- Negative and Positive, decimal place

- float framesPerSecond = 29.97;

- floats can make booleans like ints. You can also try this
  with either:
  21.f < 20.f is false. 21.f > 20.f is true. Bonus, add = to
  add equality, i.e. 18.f <= 18.f is true.

- Aside: http://en.wikipedia.org/wiki/Frame_rate

Check that you have Processing

http://processing.org/download/
Version 2.1.2 (15 April 2014) is what I'm using

Program 1: First Drawing Program

- Will have to remember if the screen is black or white. What kind of variable should we use? What should it be at the start?

- A pixel in this case is the whole canvas - how big do you want to make it? How can you change it?

- Make it any size you like.

- How do you draw a rectangle of the correct size?

- How do you detect a click?

- http://processing.org/reference/ is your friend.

Go!

(Processing Reference is your friend!)

Program 2: Grid Drawing Program

- Last time we did 1 pixel. What is the next lowest resolution we could do?

- How do we remember more than one value?

- How do we tell where the mouse is?

- We need more than if, we need else.

Go!

(Processing Reference is your friend!)

Program 3: Bigger Grid Drawing Program

- Last time we did 4 pixels. What is the highest resolution we could do?

- How do we remember all those values? How do we deal with them? Can we automate?

- Aside: Programmers are essentially lazy

- Aside: "Computers are bicycles for the mind" Steve Jobs

- We need a better way. To remember and batch process too.

an array is a way of storing lots of the things in a convenient group

- http://processing.org/reference/Array.html

- Decide how big they are at the start, like that forever

- Must be all the same type - all ints or all booleans or all floats

- boolean[] allThePixels = new boolean[1024*768];

a for loop lets you automate for as many loops as you like

- http://processing.org/reference/for.html

- You need a counter - convention is you call it i.

- Careful on your edge cases!

- How do we deal with rows and columns? What is the letter after i?

How to go between a grid and a list and vice versa

- A list is one dimensional - what does that look like?

- A grid is two dimensional - what does that look like?

- The key is to thing of a list as lots of strips laid end to end, or a grid as a stack of lists.

- How do you work out where to go from a grid to a list of the same data ? Or vice versa?

- Draw both a list and a grid of the same data! (From 0, don't forget!)

- Look at Processing Tutorials online: Two Dimensional Arrays

Go!

(Processing Reference is your friend!)