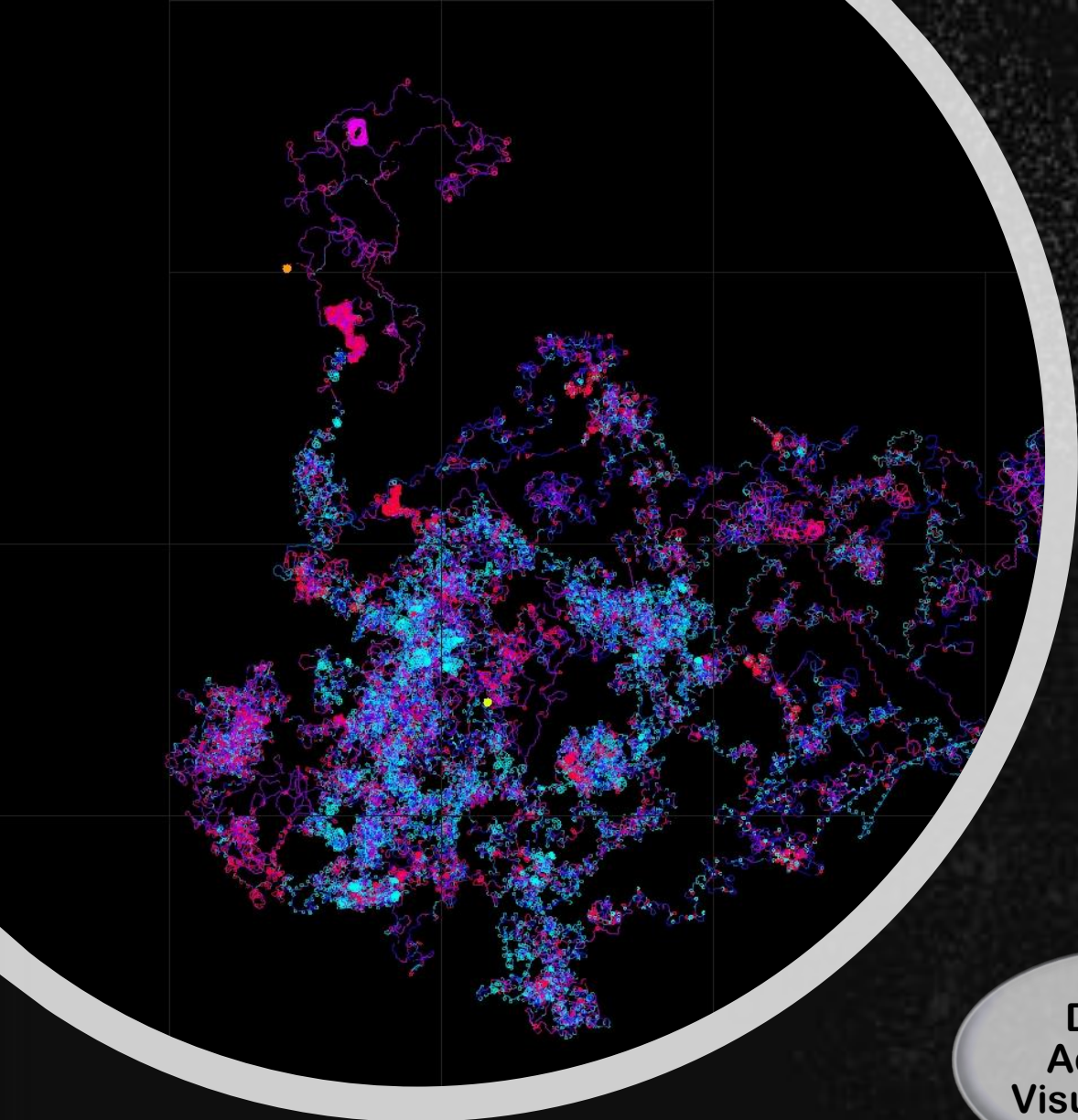


James Brooks and Chris Tacon

Processing Language



13



What is Processing

**Object
orientated**

**Complete
with IDE**

Java Based

Open source

**Develop
Advanced
Visualisations**

**Can run on
Windows,
Mac or Linux**

zoom

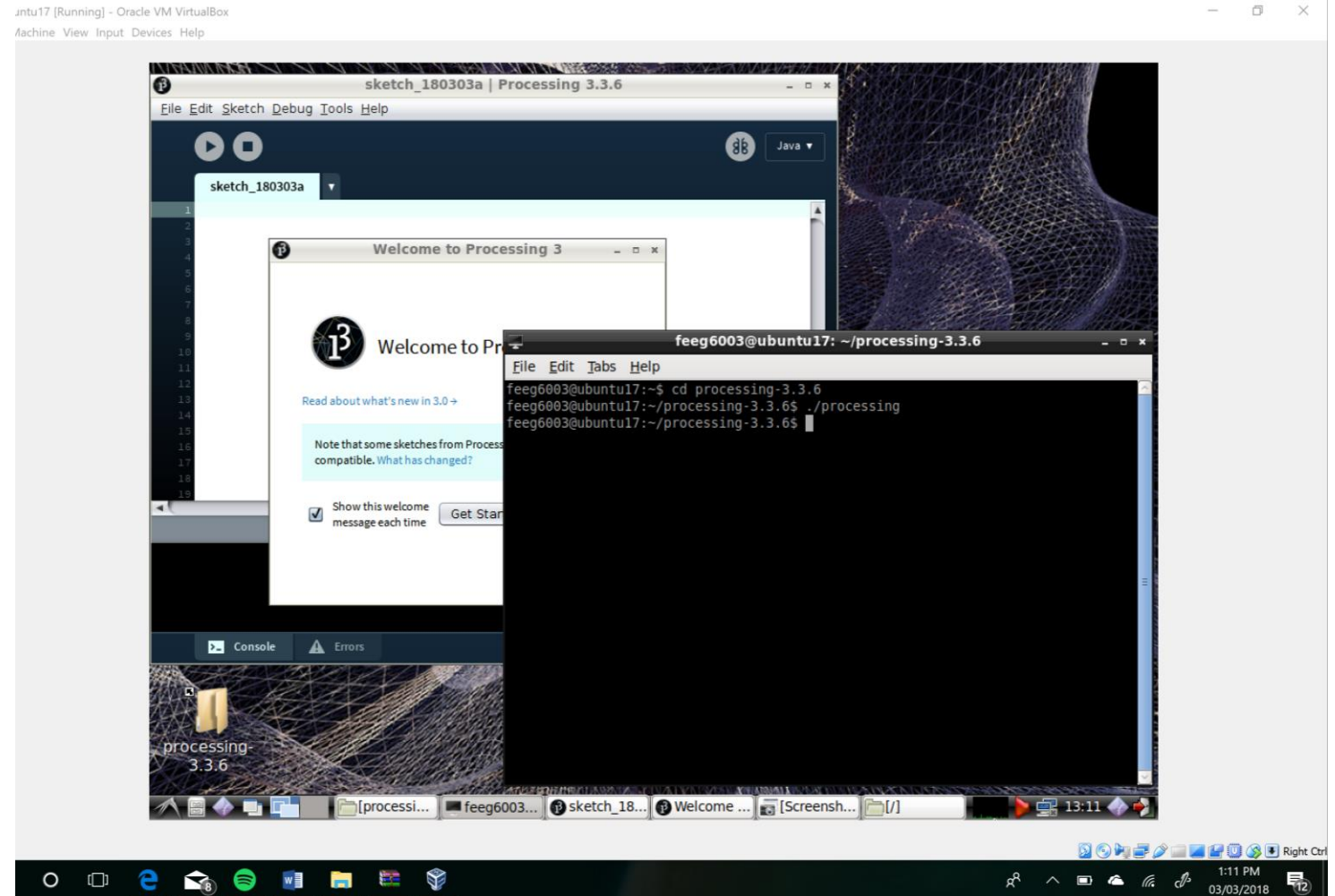
History

- Developed in 2001 by Casey Reas and Ben Fry at MIT Media Lab
- Inspired the Arduino IDE
- Further developed by the Processing Foundation along with Daniel Shiffman



How to Open Processing

1. Boot up VM (PW: feeg 6003)
2. Open Start -> System Tools -> LXTerminal
3. From terminal type:
 1. `cd processing-3.3.6`
 2. `./processing`



How to Find Processing Files

- Processing files need to be contained in folders
- Inside these folders is where auxiliary files are also stored (e.g. data files)
- The Processing file has the “.pde” extension
- Easiest to open them from within the Processing IDE

Handy Tools and Links

Interactive Online Tutorials:
<https://processing.org/tutorials/>

Example Codes:
<https://processing.org/examples/>

Download Processing on Your PC:
<https://processing.org/download/>

Handy
Frame Rate
Canvas Height and Width
Random Number

Main Variable Types
Null - Returns nothing
Int - 32,767 to -32,768
Float - Floating point
String - Array of Characters

Basics
Commenting
First Loop
Continuous Loop
For Loop Example
If Example

Text
Text Font Size
Text Font Example
Write Text

Shapes
Point
Line
Rectangle
Triangle
Ellipse
Arc
Bezier
Sphere

Object Oriented Programming
Class Structure
Class Classname
Constructor
Class Functions
Declare Objects
Initialize Objects
Call Object Functions

IO
Loading in Text File
Loading in Image File
Writing to Text File
Print to Console

Coordinates and Canvas
Set Canvas Size in Pixels
Coordinates
Align Text
Align Ellipse
Align Rectangle
Logic Statements

Interactivity
Keyboard
Special Keys
Mouse Position X and Y
Mouse Positions at previous frame
Button Mouse Click

Colours
Background Colour
Fill Colour
Remove Fill (Fully Transparent)
Border/Line Colour
Remove Borders

Chris Tacon and James Brooks
Processing 3 Cheat Sheet

Cheat Sheet – Found on VM Desktop

This PowerPoint is found on VM Desktop
(Notes are also included)

The Slides Ahead

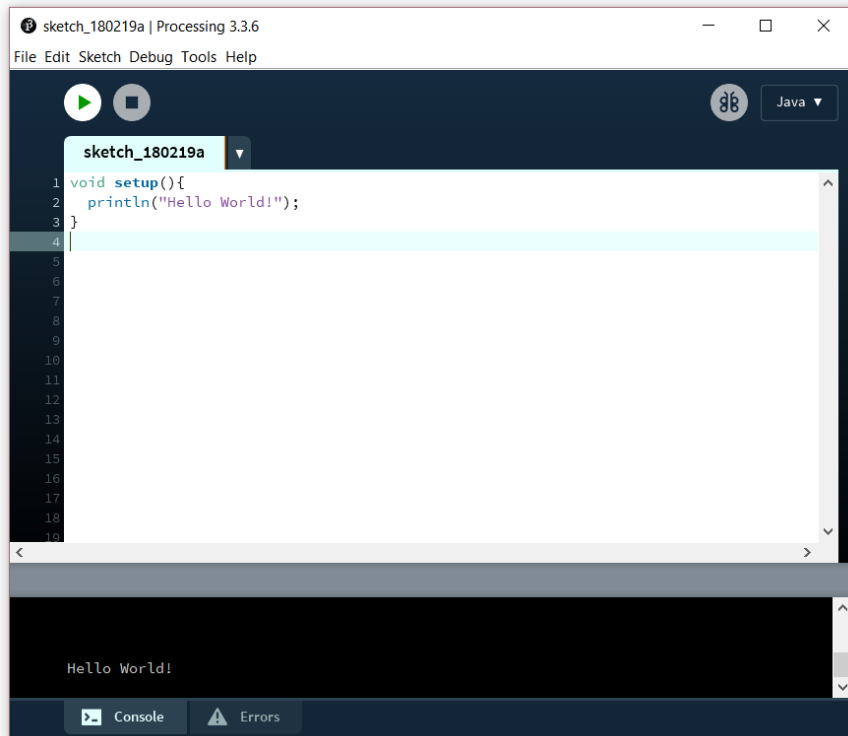
Try to follow along with the examples

Change and modify things to see what happens

If you break the code don't worry!

Folders for code given in bottom right of slides

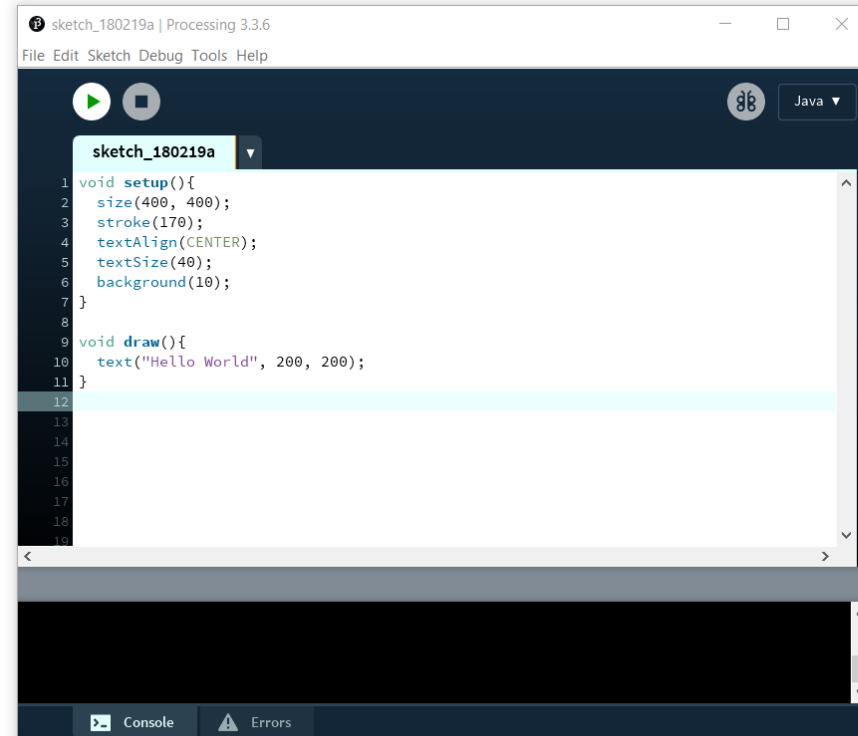
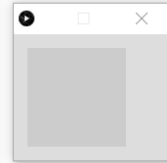
Hello World!



```
sketch_180219a | Processing 3.3.6
File Edit Sketch Debug Tools Help

sketch_180219a
1 void setup(){
2   println("Hello World!");
3 }
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

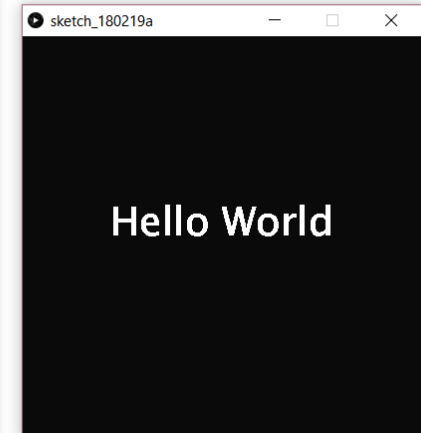
Hello World!
Console Errors
```



```
sketch_180219a | Processing 3.3.6
File Edit Sketch Debug Tools Help

sketch_180219a
1 void setup(){
2   size(400, 400);
3   stroke(170);
4   textAlign(CENTER);
5   textSize(40);
6   background(10);
7 }
8
9 void draw(){
10  text("Hello World", 200, 200);
11 }
12
13
14
15
16
17
18
19

Console Errors
```



BUT! This is really not what Processing is made for!

Hello_World_Text
Hello_World_Graphical

Taking a Closer Look at Our First Programme

This is the size of the canvas
size(*width in pixels, height in pixels*)

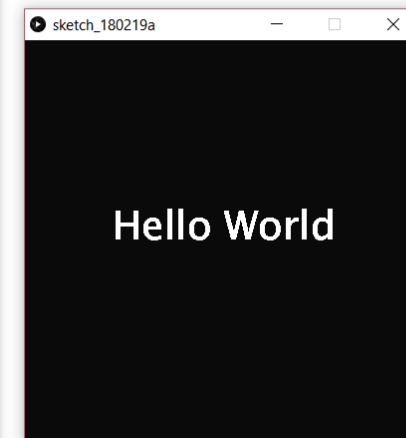
This is the background colour
background(*0 to 255 grey scale*) or
background(R, G, B)
Where R, G, B are red, green and blue
and vary between 0 and 255

```
sketch_180219a | Processing 3.3.6
File Edit Sketch Debug Tools Help

sketch_180219a
1 void setup(){
2   size(400, 400);
3   stroke(170);
4   textAlign(CENTER);
5   textSize(40);
6   background(10);
7 }
8
9 void draw(){
10  text("Hello World", 200, 200);
11 }
12
13
14
15
16
17
18
19
```

This is the setup function,
it runs once

This is the draw function,
it runs in a loop



This is the canvas

This is the IDE

Some Shapes!

Sets coordinate system of rectangle

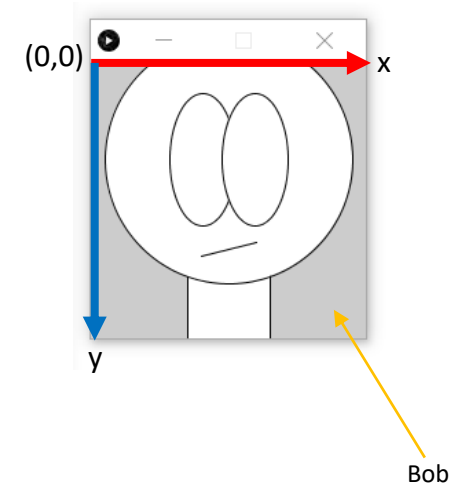
Rectangle shape
`rect(x, y, width, height)`

Ellipse shape
`ellipse(x, y, width, height)`

Draw a line
`line(x1, y1, x2, y2)`

```
sketch_180219a | Processing 3.3.6
File Edit Sketch Debug Tools Help

sketch_180219a
1 void setup(){
2   size(200,200);
3   rectMode(CENTER);
4   rect(100,100,60,300);
5   ellipse(100,70,180,180);
6   ellipse(81,70,48,96);
7   ellipse(119,70,48,96);
8   line(80,140,120,130);
9 }
10
11 void draw(){
12 }
13
14
15
16
17
18
19
20
```



Add Some Colour!

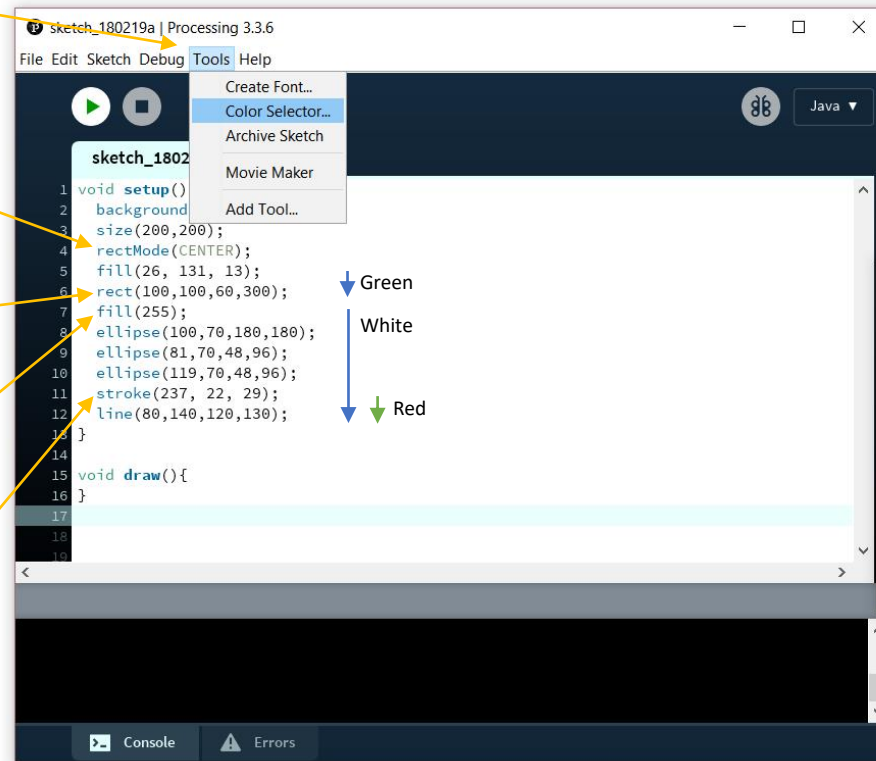
Open color selector from tools

Change fill to green

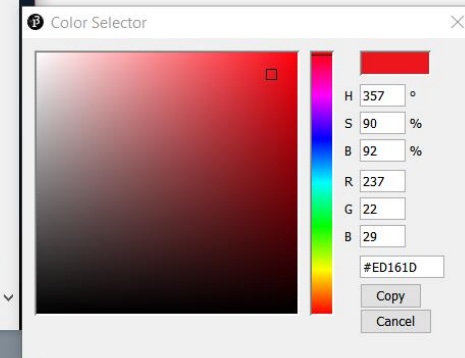
Fill this rectangle with green

Change fill colour to white

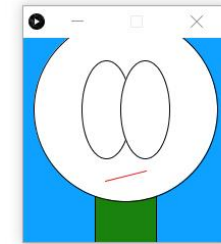
Stroke (line colour) is default black, change to Red



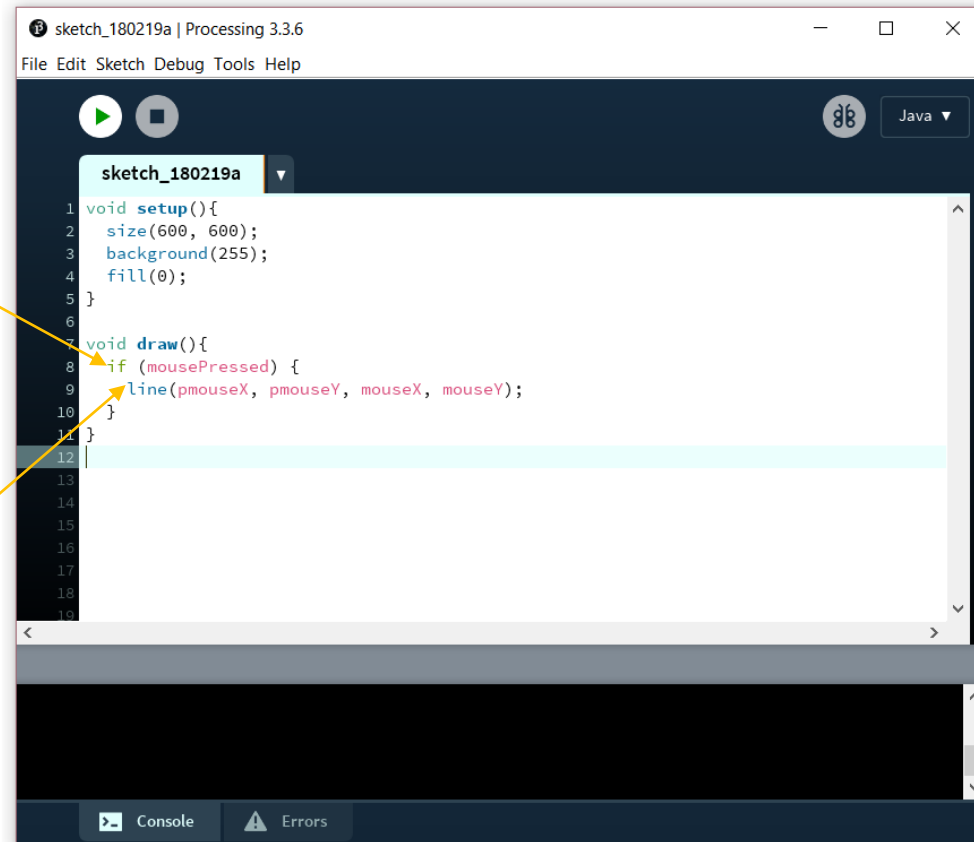
```
1 void setup()
2   background(255);
3   size(200,200);
4   rectMode(CENTER);
5   fill(26, 131, 13);
6   rect(100,100,60,300);
7   fill(255);
8   ellipse(100,70,180,180);
9   ellipse(81,70,48,96);
10  ellipse(119,70,48,96);
11  stroke(237, 22, 29);
12  line(80,140,120,130);
13 }
14
15 void draw(){
16 }
17
18
19
```



Color Selector, Helps with RGB selection



Make Your Own Paint Software!



```
sketch_180219a | Processing 3.3.6
File Edit Sketch Debug Tools Help

sketch_180219a
1 void setup(){
2   size(600, 600);
3   background(255);
4   fill(0);
5 }
6
7 void draw(){
8   if (mousePressed) {
9     line(pmouseX, pmouseY, mouseX, mouseY);
10  }
11 }
12
13
14
15
16
17
18
19
```



If ANY mouse button is held down

Then draw a line between last mouse location (pmouseX, pmouseY) and current one (mouseX, mouseY)

Genuinely my handwriting

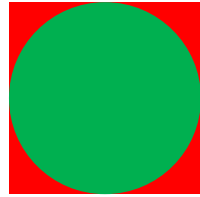
Remember, It's Not Python!

;

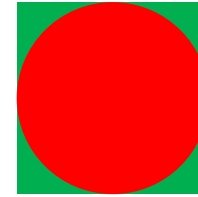
{ }

Remember, Order Matters!

```
fill(237, 22, 29);  
rect(100, 100, 50, 50);  
fill(26, 131, 13);  
ellipse(100, 100, 50, 50);
```



```
fill(26, 131, 13);  
rect(100, 100, 50, 50);  
fill(237, 22, 29 );  
ellipse(100, 100, 50, 50);
```



```
fill(237, 22, 29);  
ellipse(100, 100, 50, 50);  
fill(26, 131, 13);  
rect(100, 100, 50, 50);
```

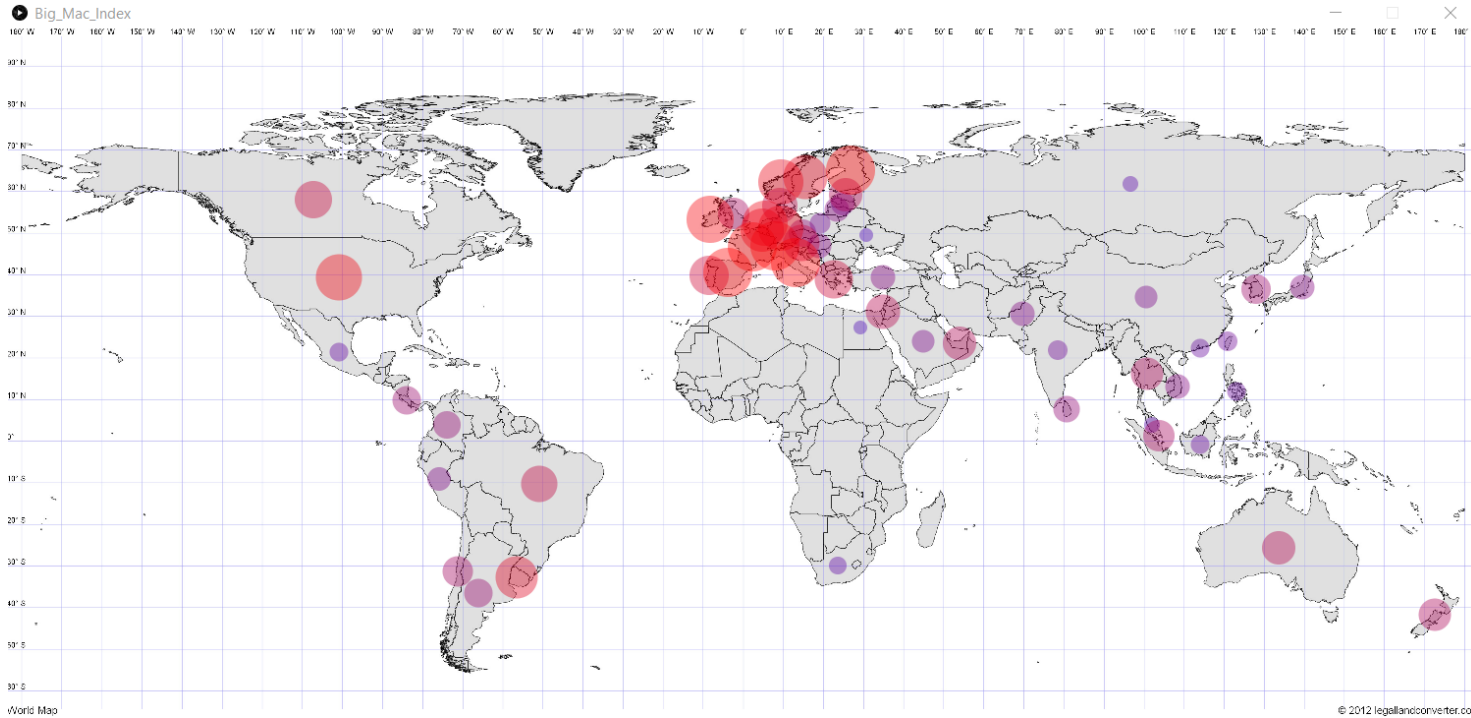


Think of fill (or stroke) as picking up a brush of that colour which is applied for every subsequent object.

The later coded objects will appear on top of the earlier coded objects

Let's Do Some Data Visualisation!

The Economist

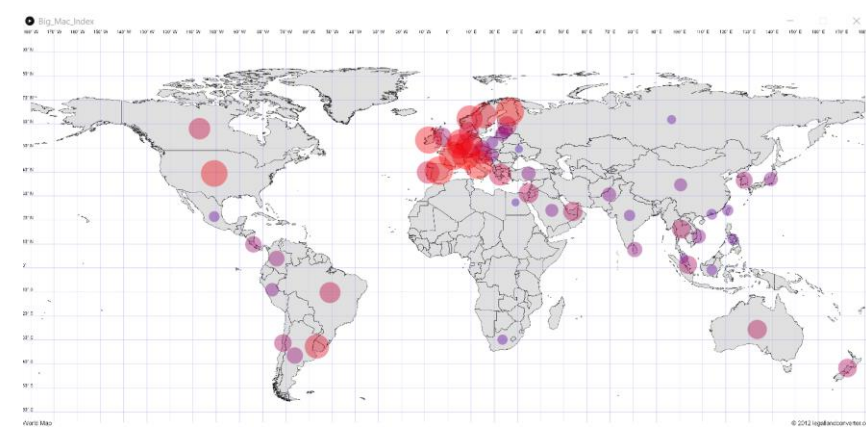


The Big Mac Index

THE ECONOMIST BIG MAC INDEX, MARCH 2016

© 2016 The Economist Group
First Published in The Economist, 12 March 2016

Big Mac Code



Declare a variable `img` of the built in image type

Set a scale variable to allow adjustment of circle max radius

No outlines for shapes

Load in background Earth image

Position image, `image(img, x, y, width, height)`

Load in text data file, each new line being a new item in the array

Iterate over lines and extract data from lines by splitting with tab

Notice now an extra channel in the fill function, this is the transparency (between 0 and 255)

Since we are not doing anything in a loop and the figure is only drawn once we do not need the draw function

```
Big_Mac_Index | Processing 3.3.6
File Edit Sketch Debug Tools Help

Big_Mac_Index
1 PImage img; // Declare variable "a" of type PImage
2 int scale = 50;
3
4
5 void setup() {
6   size(1500, 700);
7   noStroke();
8
9   img = loadImage("world.jpg"); // Load the image into the program
10  image(img, 0, 0, img.width/2.6, img.height/2.6);
11
12  String lines[] = loadStrings("data.txt");
13  for(int i = 0; i < lines.length; i++){
14    String pieces[] = split(lines[i], '\t');
15    int Size = int(scale * float(pieces[2]));
16    int Red = int(255*float(pieces[2]));
17    int Blue = int(255*(1-float(pieces[2])));
18    fill(Red, 0, Blue, 100);
19    ellipse(int(pieces[3]), int(pieces[4]), Size, Size);
20  }
21
22
23 void draw() {
24
25
```

Styling parameters, note that casting is important!

Brief Introduction to Classes

Classes are used to describe a 'type' of object. This allows us to easily create new instances of an object without having to write much code.

You declare an object as you would a variable:

```
Classname object1;
```

You can then initialise this object by:

```
object1 = new Classname(Temp Values);
```

This creates a new object of the type *Classname* with specific attributes given by *Temp Values*. E.g

```
red_ball = new Ball([255, 0, 0]);
```

These values are then passed to the constructor within the class which constructs the object with these variables

Class Structure

```
Class Classname {  
    //Class Variables  
    Var_type Var_name;  
    //Constructor  
    Classname(Temp Variables) {  
        /*Assign value to variable  
        From temporary variable*/  
        Var_name = Temp_Var;  
    }  
    //Class Functions  
    Return_type func1(/*External Inputs*/) {  
    }  
}
```

The Class variables should include those that will be passed into the object and any variables that are defined internally. E.g

```
Class Ball {  
    int[] colour;  
    float radius = 5.0;  
}
```

You can also define functions for the class, these functions are called for an object of this class by the syntax: *object.function(variables)*

E.g *red_ball.display();*

Where you have defined an internal function display which for example could be:

```
void display {  
    fill(colour);  
    ellipse(0, 0, radius, radius)  
}
```

Class Structure

```
Class Classname {  
    //Class Variables  
    Var_type Var_name;  
    //Constructor  
    Classname(Temp Variables) {  
        /*Assign value to variable  
        From temporary variable*/  
        Var_name = Temp_Var;  
    }  
    //Class Functions  
    Return_type func1(/*External Inputs*/) {  
    }  
}
```

You Decide What to do Next!

See The Upcoming Slides for Details

Pong

The aim of this task is to recreate the classic game Pong.

This version should be single player with one bat and ball where the ball interacts with the bat and walls of the canvas.

The game should also have a method to reset once you miss the ball.

As an additional task you should try to add a scoring method during each game and also a method that keeps track of your highscore.



Your Ultimate Challenge



Yes I'm terrible....

```
15. float spe = 7;
16. float the = random(3*PI/4, 5*PI/4);
17. ball = new Ball(x, y, dia, spe, the);
18.
19. float w = width/100;
20. float h = 3*dia;
21. float v = 9;
22. bat = new Bat(w/2, height/2, w, h, v);
23.
24. f = createFont("Arial", 32, true);
25.}
26.
27.void draw() {
28.  background(0);
29.  bat.move();
30.  bat.display();
31.  ball.move();
32.  ball.display();
33.  textFont(f,32);
34.  textAlign(LEFT);
35.  fill(255);
36.  text("Score: "+score, 10, 30);
37.  if (ball.move() == 1) {
38.    if (score > Highscore) {
39.      Highscore = score;
40.    }
41.    score = 0;
42.    textFont(f,32);
43.    fill(0, 255, 0);
44.    textAlign(RIGHT);
45.    text("Highscore: "+Highscore, width-10, 30);
46.    textAlign(CENTER);
47.    text("Press CONTROL to reset", width/2,height/2);
48.    if (key == CODED) {
49.      if (keyCode == CONTROL) {
50.        if (keyPressed == true) {
51.          reset();
52.        }
53.      }
54.    }
55.  }
```

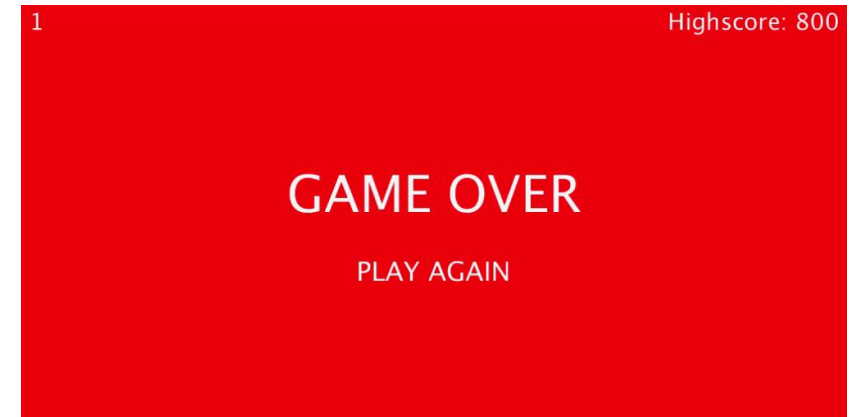
Too Hard or Short on Time?

Don't worry, there is another version of Pong we created that is slightly easier to code, has fewer lines and requires no classes.



You Decide!

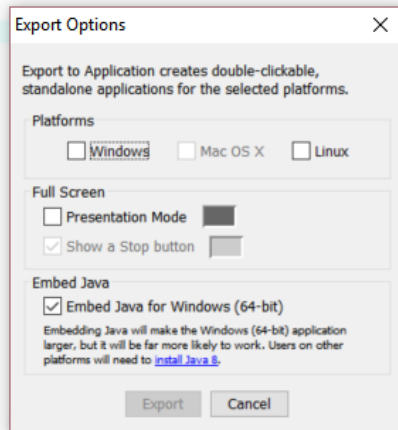
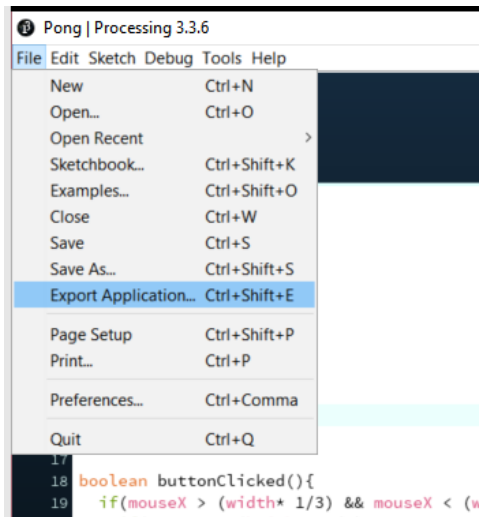
- **Expert:** Take the skeleton version of Pong *Pong_Skeleton* and try to recreate *Pong_Complete*
- **Hard:** Take *Easy_Pong_Without_High_Score_Skeleton* and make it function like *Easy_Pong_Without_High_Score*
- **Medium:** Take *Easy_Pong_Without_High_Score* and use file IO to save the high score like in *Easy_Pong_With_High_Score*
- **Easy:** Create a new sketch and create a rectangle which tracks the x-axis movement of the mouse



Use the cheat sheet provided

Anyone able to complete the Expert level *unaided* gets a reward!

Exporting as Standalone Software



- Processing can export programs including Java or without Java
- With Java included, Java does not need to be installed on host machine
- Either way, Processing does not need to be installed on host
- Mac OS X can only be exported on Macs

Thank You For Coming!