

Administrivia

CMSC110: Introduction to Computing

Fall 2016

Course Website: <http://cs.brynmawr.edu/cs110-01/>

Instructor:

Deepak Kumar, (dkumar@cs.brynmawr.edu)

Lectures

TuTh 12:55p to 2:15p in Park 338

TA-Support

>20 hrs/week in Park 231

Labs – Register and attend one of these

- Mondays 2:30p.m. to 3:30 p.m. (led by Prof. Eisenberg)
- Tuesdays 2:15 p.m. to 3:15 p.m. (led by Prof. Kumar)
- Wednesdays 2:30 p.m. to 3:30 p.m. (led by Prof. Eisenberg)
- Thursdays 2:15 p.m. to 3:15 p.m. (led by Prof. Kumar)

Office Hours

Available by appointment. Walk-ins are welcome!

Grading

• ~7 Assignments	45%
• Citizenship	10%
• Exam 1	20%
• Exam 2	25%
Total	100%

Administrivia

Software

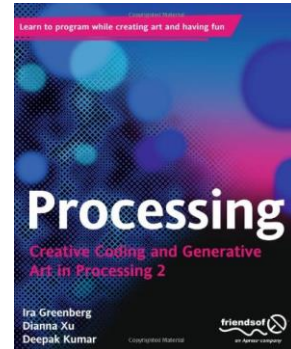
Processing 2.X

- Already installed in the CS Lab
- Also available for your own computer @ www.processing.org
- Processing == Java



Book

Creative Coding & Generative Art in Processing 2 by Ira Greenberg, Dianna Xu, Deepak Kumar, friendsofEd/APress, 2013. Available at the Campus Bookstore or amazon.com or other vendors.



Class Lottery

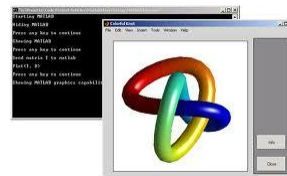
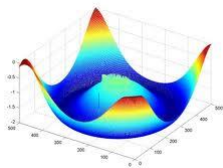
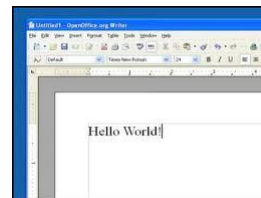
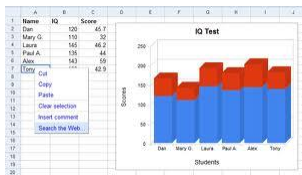
- Make sure to sign-in your name.
- If you are not on the class list, sign on the attached sheet. We will contact you by e-mail as soon as we have confirmation from other students.

What is Computing?

CMSC110 - Fall 2016

5

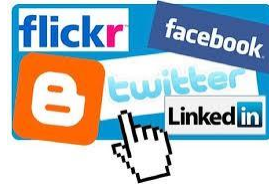
Computing: Your Parent's View



CMSC110 - Fall 2016

6

Computing: internet, e-mail, network...



CMSC110 - Fall 2016

7

Computing: Digital Photography



<http://www.alanzeyes.com/2009/02/hdr-photography.html>

8

Computing: Entertainment...



Spotify®



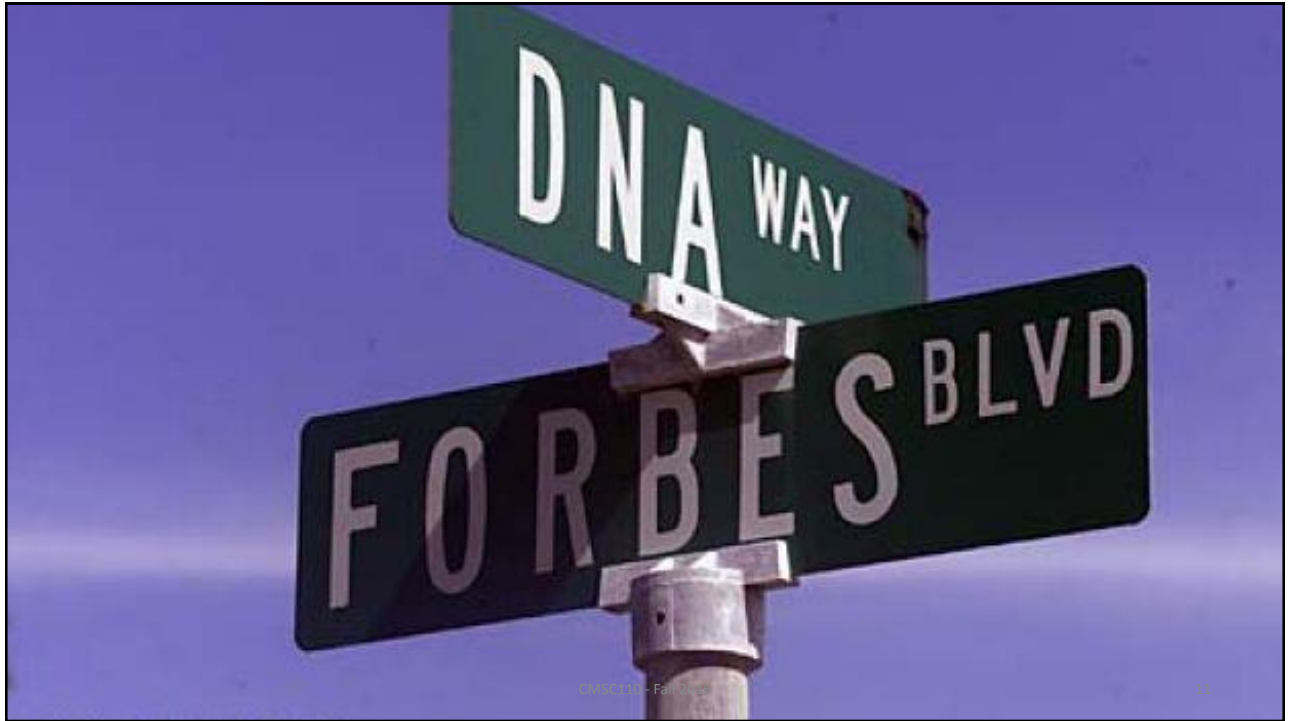
CMSC110 - Fall 2016

9

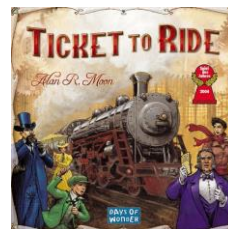
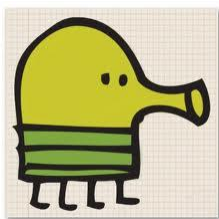


CMSC110 - Fall 2016

10

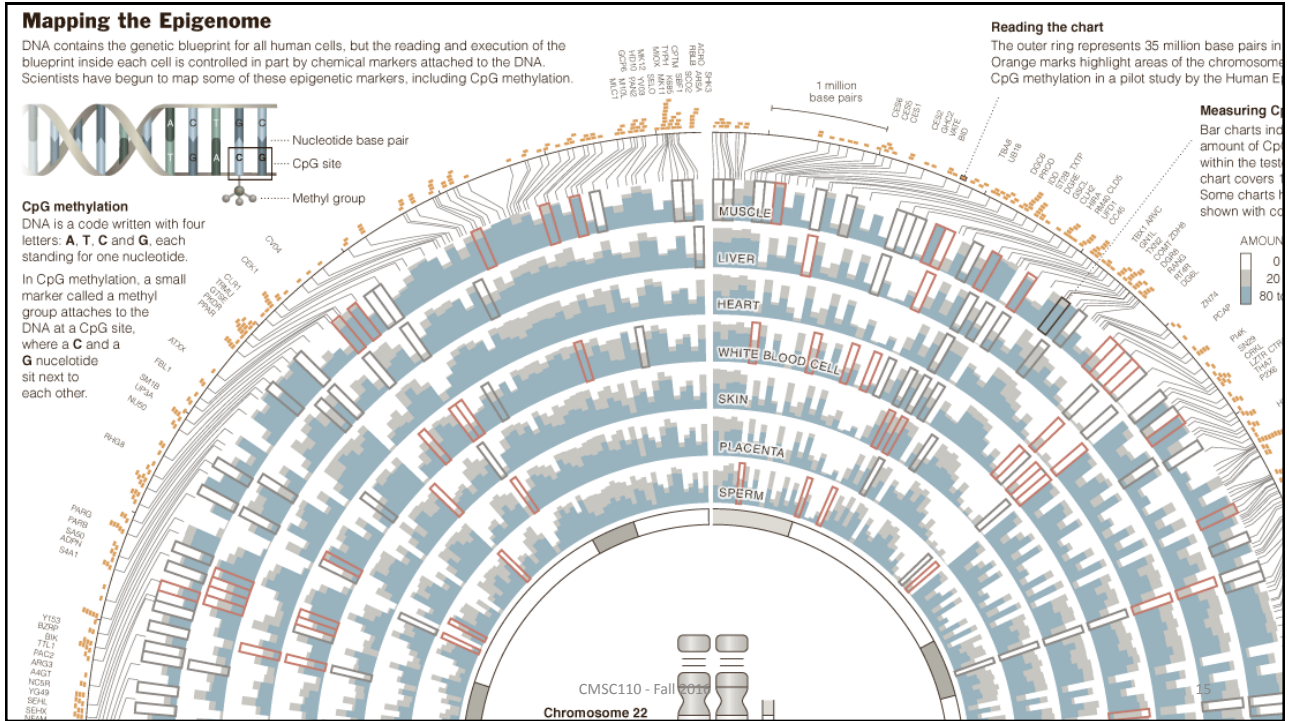


Computing: Entertainment...

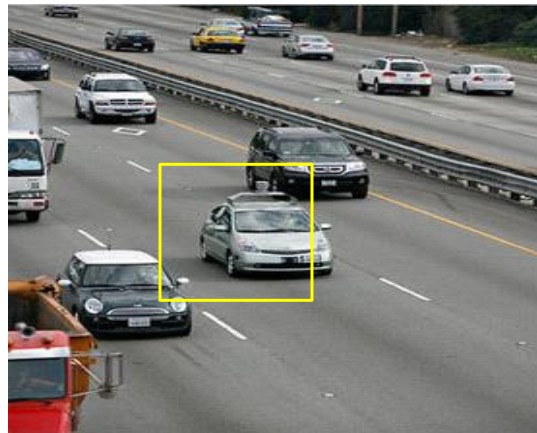




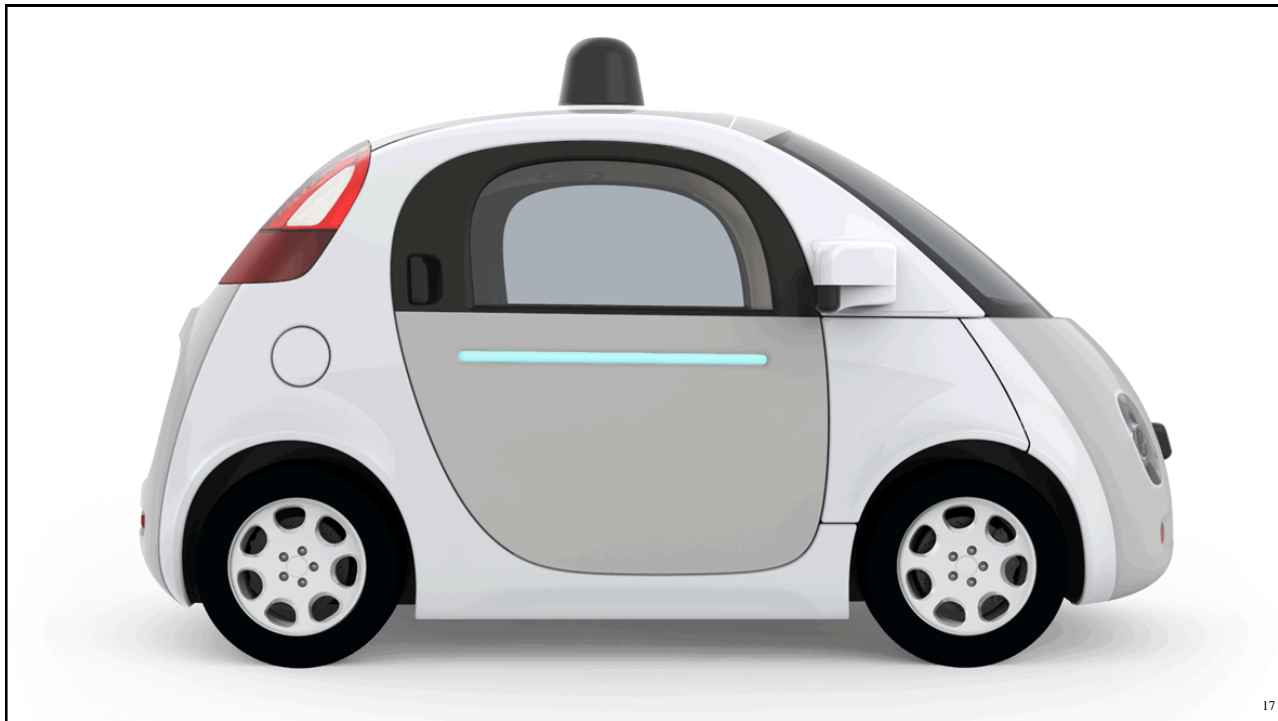
Cutting Edge Computer Science



Self-driving (Autonomous) Cars

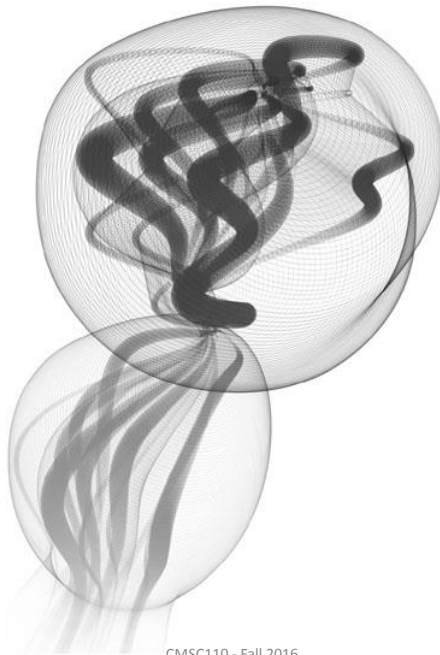


- Nevada made it legal for autonomous cars to drive on roads on March 1, 2012
- California, Florida, and Michigan as well by 12/2013





ART



CMSC110 - Fall 2016

Protobytes
By Ira Greenberg

20

Some Areas in Computer Science



Artificial
Intelligence



Robotics



Human-Computer
Interaction



Computer
Graphics



Computer
Vision



Operating
Systems



Computer
Networking



Databases



Computer
Security



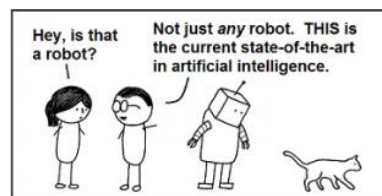
Ubiquitous
Computing

CMSC110 - Fall 2016

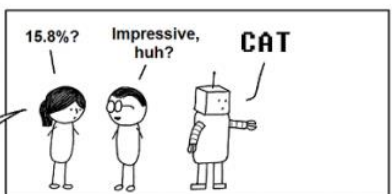
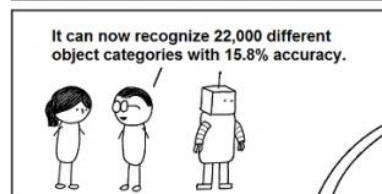
21

More trendy...

- Machine Learning
(Deep Learning)
- Data Science
(Big Data)



We trained a 9-layered locally connected sparse autoencoder with pooling and local contrast normalization on a dataset of 10 million images. It was trained for 3 days on a cluster of 1000 machines comprising 16,000 cores.



CMSC110 - Fall 2016

22

What is Computer Science?

Computer science is the study of solving problems using computation

- Computers are part of it, but the emphasis is on the problem solving aspect



Computer scientists work across disciplines:

Mathematics

Biology (bioinformatics)

Chemistry

Physics

Geology

Geoscience

Archaeology

Psychology

Sociology

Cognitive Science

Medicine/Surgery

Engineering

Linguistics

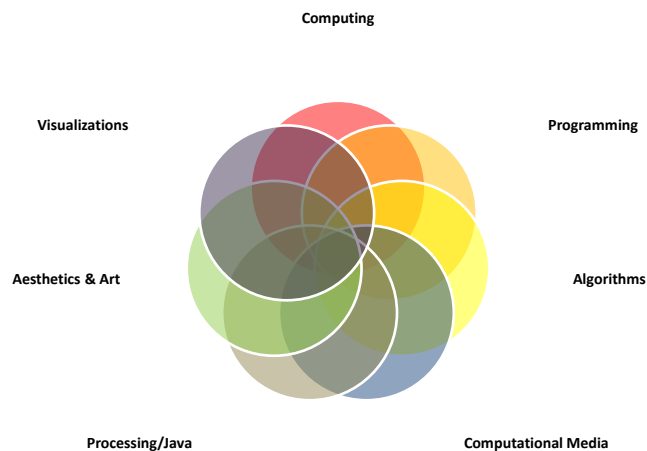
Art

...

CMSC110 - Fall 2016

23

Creative Introduction to ^ Computing



CMSC110 - Fall 2016

24

Algorithms

An **algorithm** is an effective method for solving a problem expressed as a finite sequence of instructions. For example,

Put on shoes

left sock
right sock
left shoe
right shoe



CMSC110 - Fall 2016

25

Programming = Writing Apps

Programming is the process of designing, writing, testing, debugging / troubleshooting, and maintaining the source code of computer programs.

This source code is written in a **programming language**.

CMSC110 - Fall 2016

26

A program

```
int areaOfCircle(int radius){
    return PI*radius*radius;
}

r = 10;
area = areaOfCircle(r);
```

Programming Languages

Processing	Python	Lisp
<pre>int areaOfCircle(int radius){ return PI*radius*radius; } r = 10; area = areaOfCircle(r);</pre>	<pre>def areaOfCircle(radius): return PI*radius*radius; r = 10 area = areaOfCircle(r)</pre>	<pre>(defun areaOfCircle (radius) (return (* PI radius radius))) (setq r 10) (setq area (areaOfCircle r))</pre>

Programming Languages

Processing	Python	Lisp
<pre>int areaOfCircle(int radius){ return PI*radius*radius; } r = 10; area = areaOfCircle(r);</pre>	<pre>def areaOfCircle(radius): return PI*radius*radius; r = 10 area = areaOfCircle(r)</pre>	<pre>(defun areaOfCircle (radius) (return (* PI radius radius))) (setq r 10) (setq area (areaOfCircle r))</pre>

FORTRAN, BASIC, Pascal, C, Ada, C++, C#, Java, Javascript, Perl, Ruby, Swift, R...

There are over 3000 of them!

A more interesting program...

```
Eye e1, e2, e3, e4, e5;

void setup()
{
  size(200, 200);
  smooth();
  noStroke();
  e1 = new Eye( 50, 16, 80);
  e2 = new Eye( 64, 85, 40);
  e3 = new Eye( 90, 200, 120);
  e4 = new Eye(150, 44, 40);
  e5 = new Eye(175, 120, 80);
} // setup()

void draw()
{
  background(102);

  e1.update(mouseX, mouseY);
  e2.update(mouseX, mouseY);
  e3.update(mouseX, mouseY);
  e4.update(mouseX, mouseY);
  e5.update(mouseX, mouseY);

  e1.display();
  e2.display();
  e3.display();
  e4.display();
  e5.display();
} // draw()
```

```
class Eye
{
  int ex, ey;
  int size;
  float angle = 0.0;

  Eye(int x, int y, int s) {
    ex = x;
    ey = y;
    size = s;
  } // Eye()

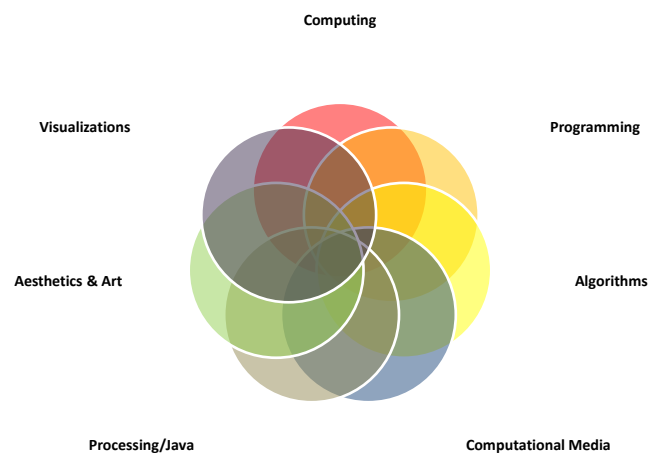
  void update(int mx, int my) {
    angle = atan2(my-ey, mx-ex);
  } // update()

  void display() {
    pushMatrix();
    translate(ex, ey);
    fill(255);
    ellipse(0, 0, size, size);
    rotate(angle);
    fill(153);
    ellipse(size/4, 0, size/2, size/2);
    popMatrix();
  } // display()
} // class Eye
```

Our Goal

- Use computing to realize works of art
- Explore new metaphors from computing: images, animation, interactivity, visualizations
- Learn the basics of computing
- Have fun doing all of the above!

Introduction to ^{Creative} Computing



Examples

CMSC110 - Fall 2016

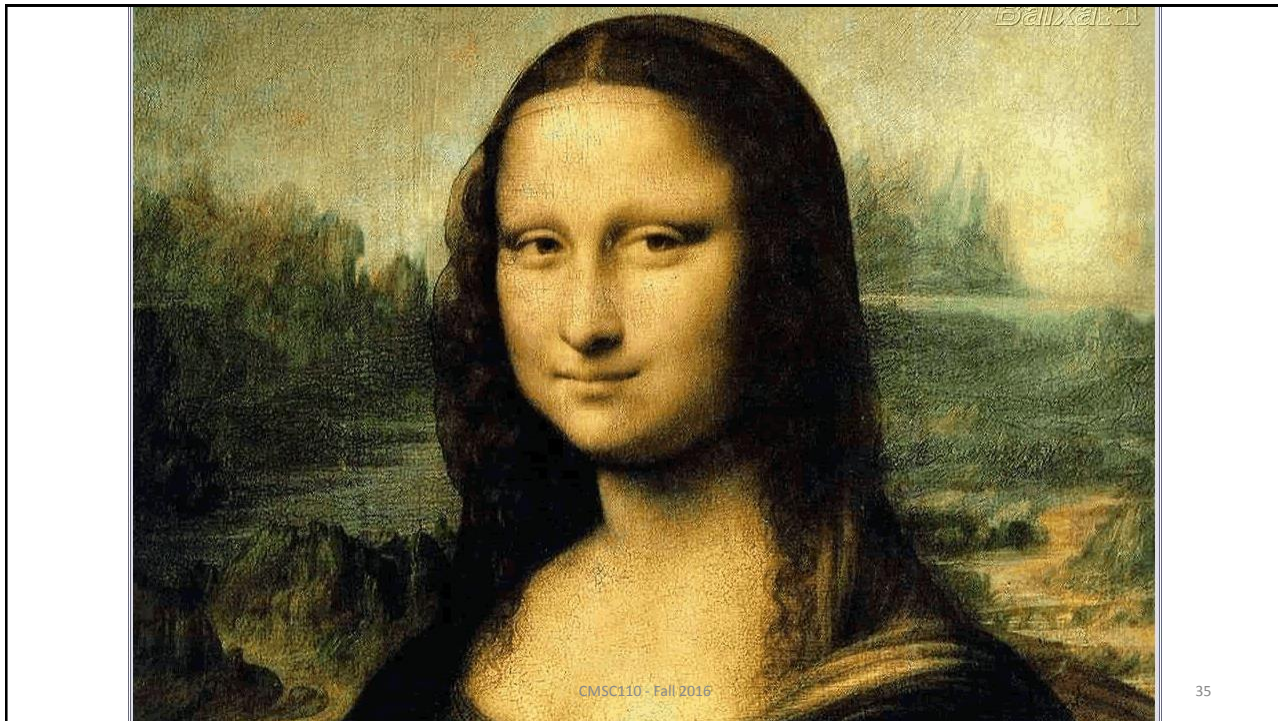
33

Shepard Fairey



CMSC110 - Fall 2016

34





CMSC110 - Fall 2016

37

Summertime

Summertime,
And the livin' is easy
Fish are jumpin'
And the cotton is high

Your daddy's rich
And your mamma's good lookin'
So hush little baby
Don't you cry

One of these mornings
You're going to rise up singing
Then you'll spread your wings
And you'll take to the sky

But till that morning
There's a'nothing can harm you
With daddy and mamma standing by

Summertime,
And the livin' is easy
Fish are jumpin'
And the cotton is high

Your daddy's rich
And your mamma's good lookin'
So hush little baby
Don't you cry

Word Cloud

Created using: wordle.net

Lyrics by George Gershwin@MSC110 - Fall 2016

38

World Cloud



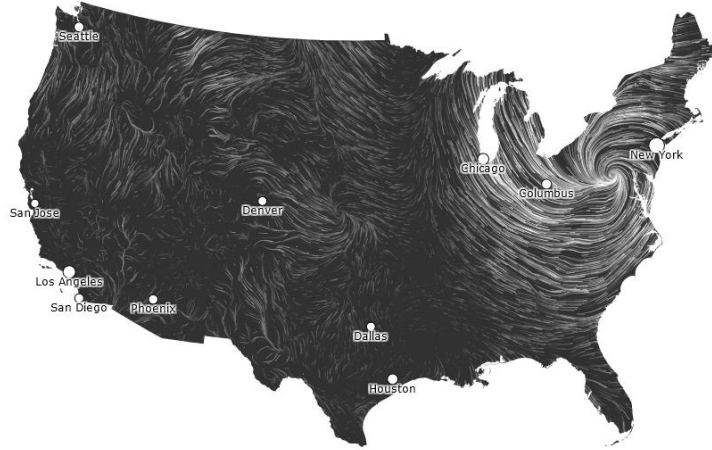
President's Inaugural Addresses



Map-based

October 30, 2012
6:59 am EST
(time of forecast download)

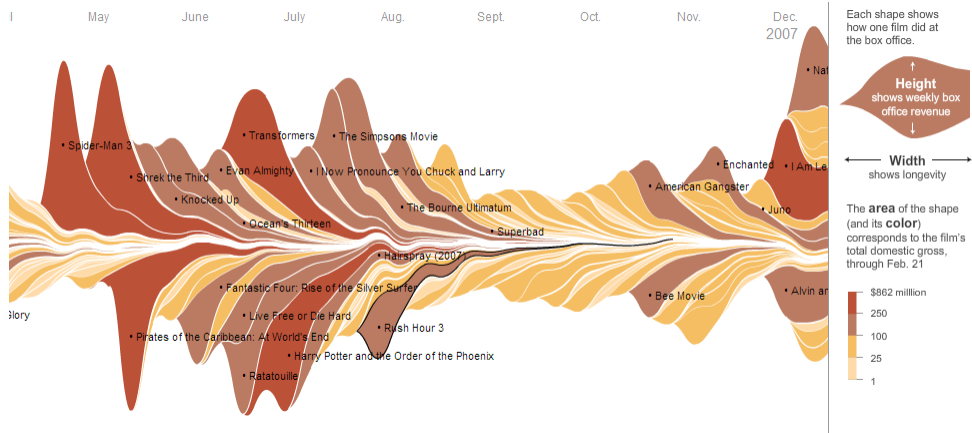
top speed: 39.7 mph
average: 8.4 mph



CMSC110 - Fall 2016

41

Box Office Earnings



From: *The Ebb and Flow of Movies: Box Office Receipts 1986 — 2008*
nytimes.com
February 23, 2008

CMSC110 - Fall 2016

42

Our Goal

- Use computing to realize works of art
- Explore new metaphors from computing:
images, animation, interactivity, visualizations
- Learn the basics of computing
- Have fun doing all of the above!

Let's get started...

Administrivia

Software

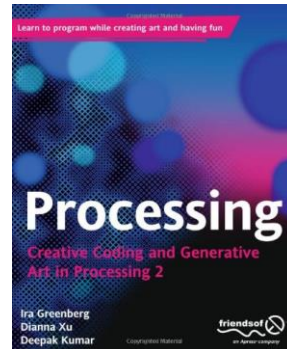
Processing 2.X

- Already installed in the CS Lab
- Also available for your own computer @ www.processing.org
- Processing == Java



Book

Creative Coding & Generative Art in Processing 2 by Ira Greenberg, Dianna Xu, Deepak Kumar, friendsofEd/APress, 2013. Available at the Campus Bookstore or amazon.com or other vendors.



CMSC110 - Fall 2016

45

Homework

- Go the CS Computer Lab (Room 231 PSB)
- Log in
- Start the Processing application (Make sure it is Version 2.x)
- In a web browser, go to the Tutorials section of processing.org

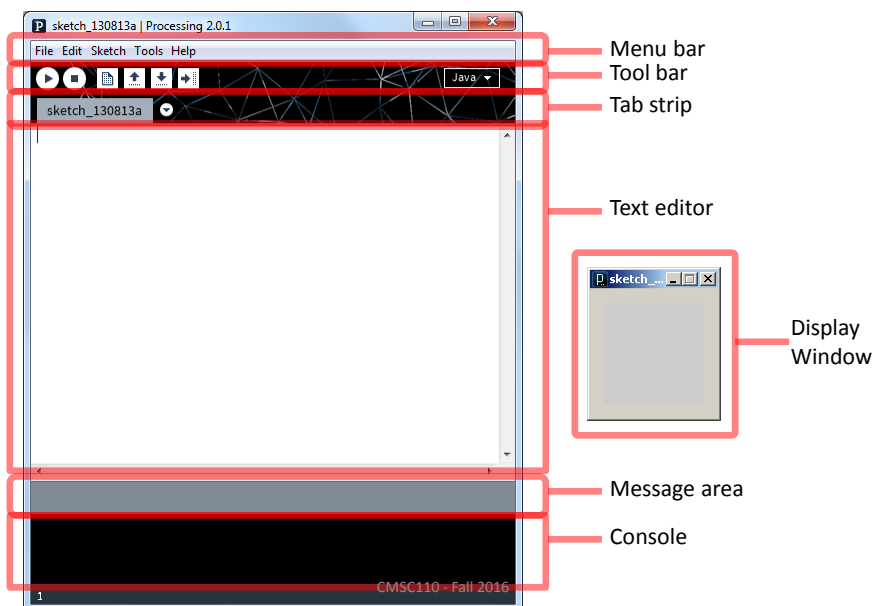
<http://www.processing.org/tutorials/gettingstarted/>

- Read the Getting Started tutorial (by Casey Reas & Ben Fry) and try out the two examples of simple Processing programs presented there
- If you'd like, install Processing 2.x on your own computer
- Read Chapter 1 (Read pages 1-12, skim 12-32)

CMSC110 - Fall 2016

46

Processing 2.0 IDE



Primitive 2D Shapes

- point
- line
- triangle
- rect (rectangle)
- quad (quadrilateral, four-sided polygon)
- ellipse
- arc (section of an ellipse)
- curve (Catmull-Rom spline)
- bezier (Bezier curve)

CMSC110 - Fall 2016

49

Processing p5.js Processing.py Processing for Android Processing Fo

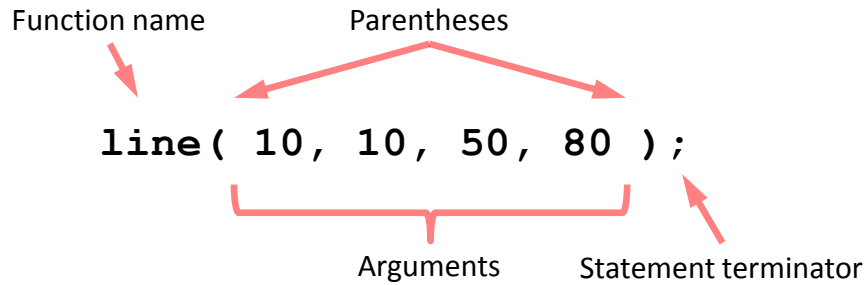
Processing

[Cover](#) [Download](#) [Exhibition](#) [Reference](#) [Libraries](#) [Tools](#) [Environment](#) [Tutorials](#) [Examples](#) [Books](#) [Handbook](#)

Reference. Processing was designed to be a flexible software sketchbook.

	Structure	Shape	Color
Reference	() (parentheses)	createShape()	Setting
Libraries	, (comma)	loadShape()	background()
Tools	. (dot)	PShape	clear()
Environment	/**/ (multiline comment)		colorMode()
	/**/ (doc comment)	2D Primitives	fill()
Tutorials	// (comment)	arc()	noFill()
Examples	;(semicolon)	ellipse()	noStroke()
Books	= (assign)	line()	stroke()
Handbook	[] (array access)	point()	
	{ } (curly braces)	quad()	Creating & Reading

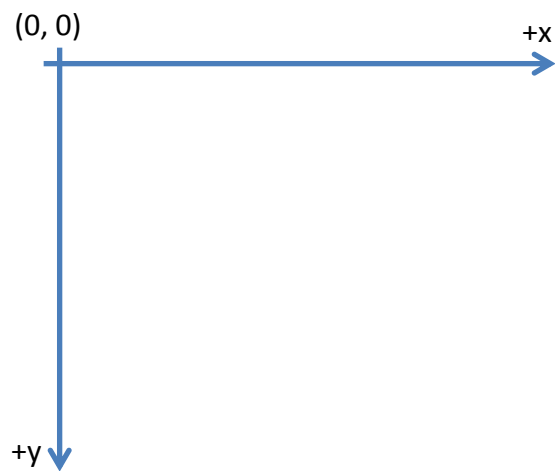
Anatomy of a Function Call



CMSC110 - Fall 2016

51

Coordinate System



CMSC110 - Fall 2016

52

Pixels



CMSC110 - Fall 2016

53

Processing Canvas

```
size( width, height );
```

Set the size of the canvas.

```
background( [0..255] );
```

Set the background grayscale color.

CMSC110 - Fall 2016

54

Drawing Primitives

```
point( x, y );
```

```
line( x1, y1, x2, y2 );
```

```
triangle( x1, y1, x2, y2, x3, y3 );
```

```
quad( x1, y1, x2, y2, x3, y3, x4, y4 );
```

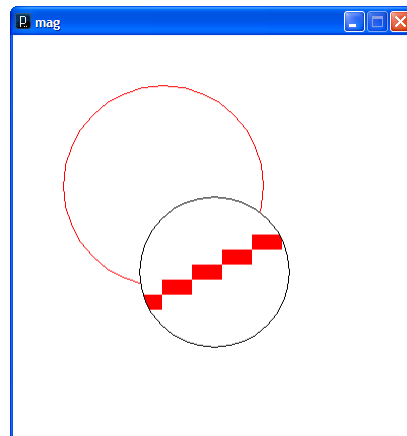
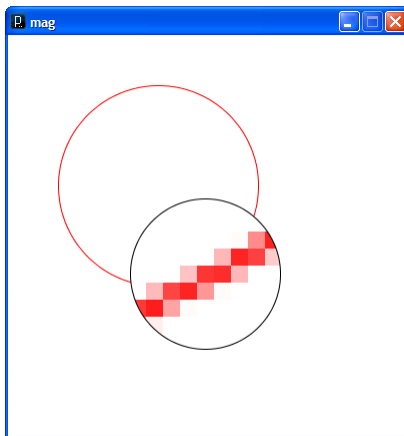
```
rect( x, y width, height );
```

```
ellipse( x, y, width, height );
```

CMSC110 - Fall 2016

55

smooth() vs. noSmooth()



CMSC110 - Fall 2016

56

Colors

Composed of four elements:

1. Red

2. Green

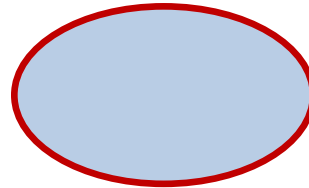
3. Blue

4. Alpha (Transparency)

Why 0 .. 255?

Shape Formatting

1. Fill color
2. Line thickness
3. Line color



*These are properties of your paintbrush,
not of the object you are painting.*



CMSC110 - Fall 2016

59

Fill Color

```
fill(gray) ;  
fill(gray, alpha) ;  
fill(red, green, blue) ;  
fill(red, green, blue, alpha) ;  
  
noFill() ;
```



CMSC110 - Fall 2016

60

Stroke (Line) Color

```
stroke(gray);
stroke(gray, alpha);
stroke(red, green, blue);
stroke(red, green, blue, alpha);

noStroke();
```



CMSC110 - Fall 2016

61

strokeCap()



```
smooth();
strokeWeight(12.0);
strokeCap(ROUND);
line(20, 30, 80, 30);
strokeCap(SQUARE);
line(20, 50, 80, 50);
strokeCap(PROJECT);
line(20, 70, 80, 70);
```

strokeWeight()

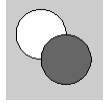


```
smooth();
strokeWeight(1); // Default
line(20, 20, 80, 20);
strokeWeight(4); // Thicker
line(20, 40, 80, 40);
strokeWeight(10); // Beastly
line(20, 70, 80, 70);
```

http://processing.org/reference/strokeCap_.html
http://processing.org/reference/strokeWeight_.html

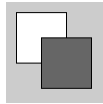
62

ellipseMode



```
ellipseMode(CENTER);
ellipse(35, 35, 50, 50);
ellipseMode(CORNER);
fill(102);
ellipse(35, 35, 50, 50);
```

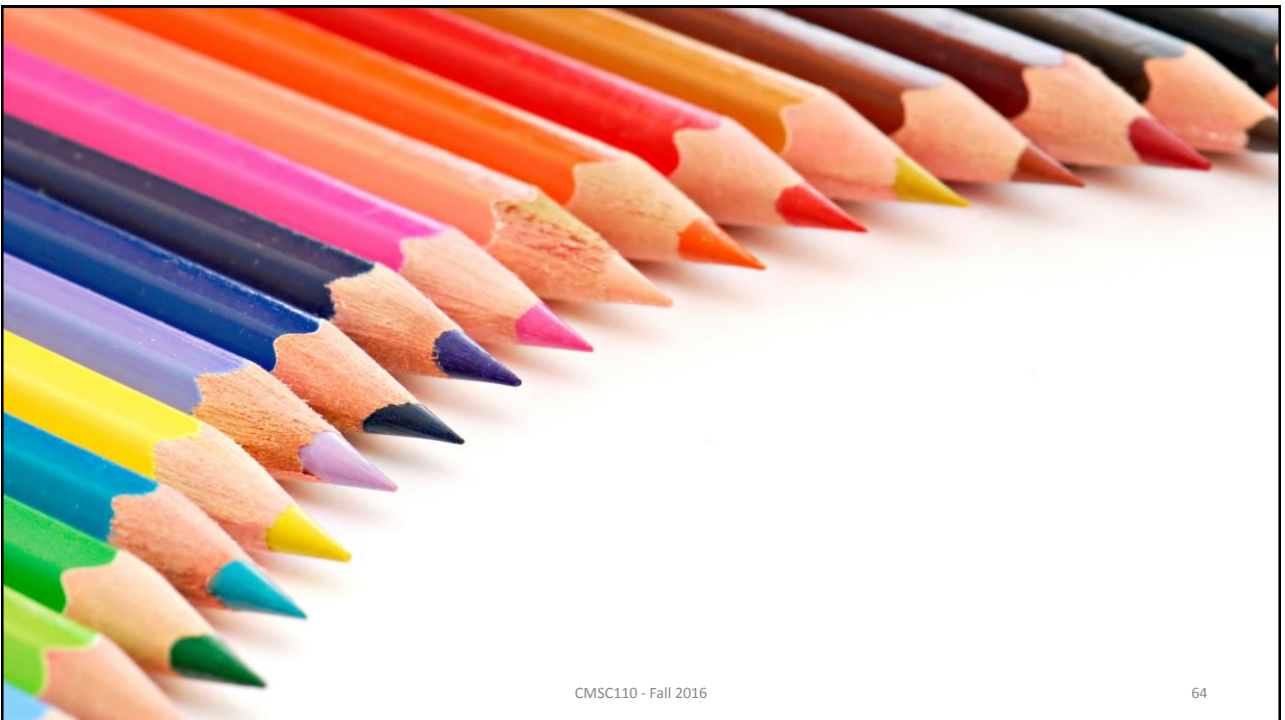
rectMode



```
rectMode(CENTER);
rect(35, 35, 50, 50);
rectMode(CORNER);
fill(102);
rect(35, 35, 50, 50);
```

CMSC110 - Fall 2016 http://processing.org/reference/ellipseMode_.html
http://processing.org/reference/rectMode_.html

63



CMSC110 - Fall 2016

64

