

A wireframe 3D scene with a black background. In the lower-left foreground, there is a hand rendered as a mesh of red lines. In the center, a white wireframe rectangular box is visible. The text is overlaid on the scene.

CGT 215

Computer Graphics Programming I

Introduction

CGT 215

- Programming fundamentals
- Logic
- Problem solving
- Provides the basis for developing object-oriented applications
- How to write, compile, build, and debug GRAPHICS applications.

Course Philosophy

- Teach programming through graphics
- Design & Visualization
- Interactivity
- Experimentation
- Learn by doing 5 projects



Processing IDE

Family Tree

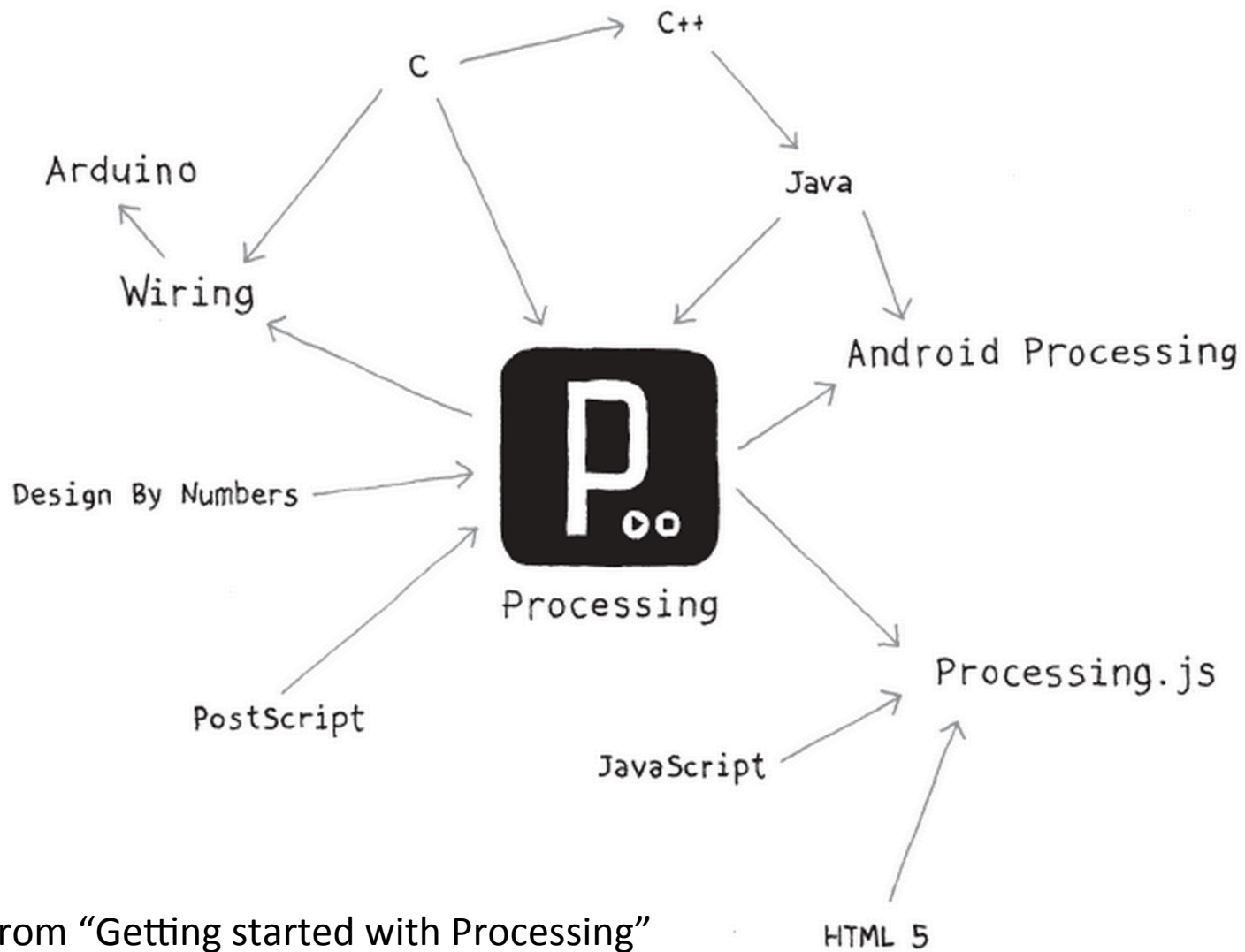


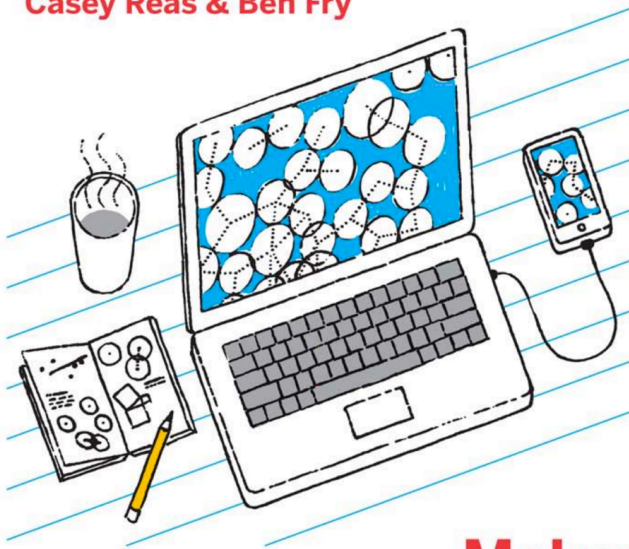
Image from "Getting started with Processing"

Make: PROJECTS

A HANDS-ON
INTRODUCTION
TO MAKING
INTERACTIVE
GRAPHICS

Getting Started with Processing

Casey Reas & Ben Fry



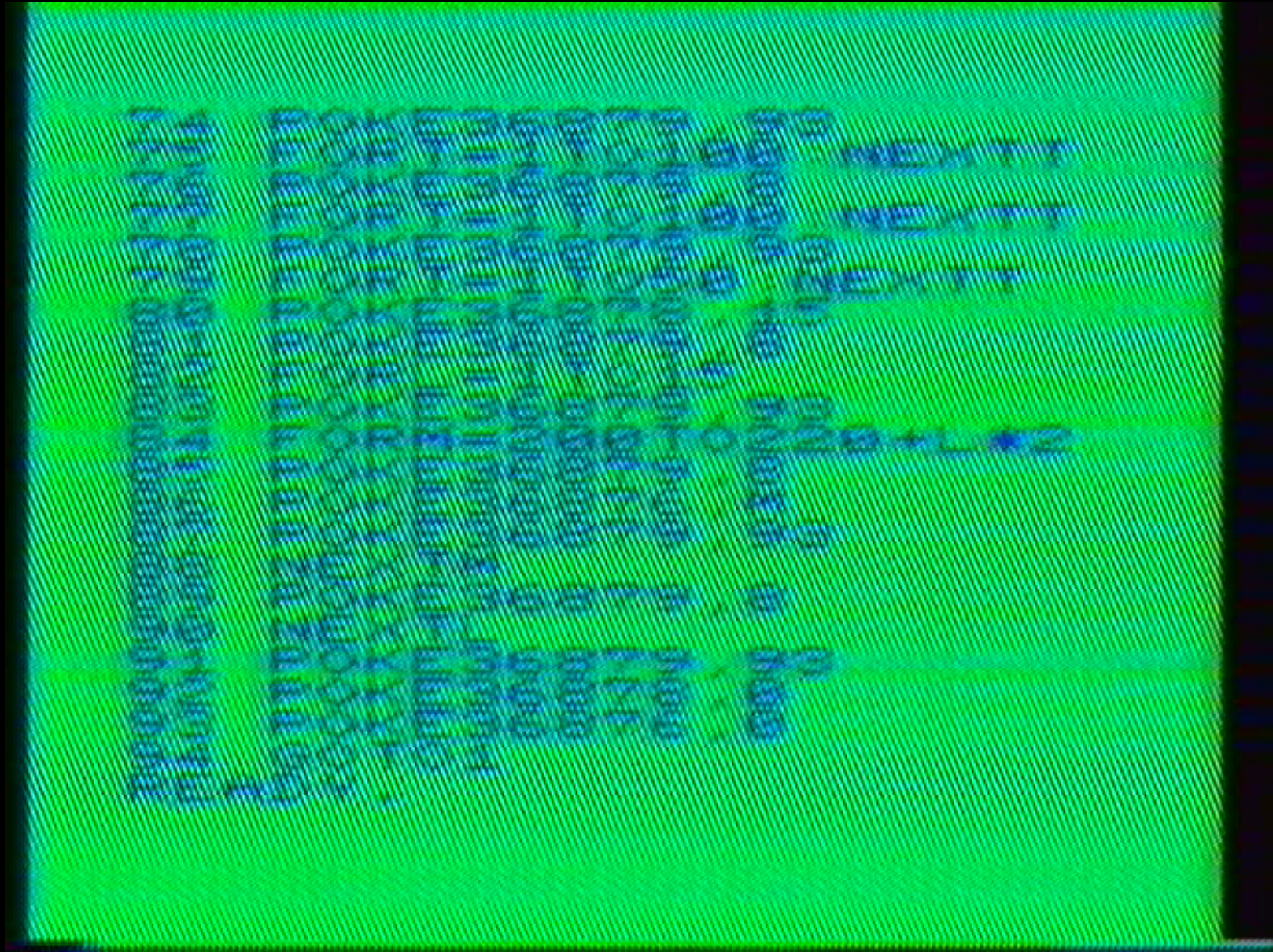
O'REILLY

Make:
makezine.com

← Course Textbook



Why Programming?



Basic Heart. 1999



In and Out. Collaboration with Max Carlson

Why Programming?

Assignment 1

Why do you want to learn computer graphics programming?

- Type up in one or two paragraphs.
- Submit on blackboard

?





Joshua Davis

<https://www.behance.net/joshuadavis>



Marius Watz

<https://www.flickr.com/photos/watz>



Rafael Rozendaal

<http://www.newrafael.com/>

Why Programming

Cybernetics

Big Data

We are data
(Data trails)

<http://vimeo.com/91054550>

Digital Fabrication

Core competency

Simulation

Parametric Design

CGT 215

Geometric Primitives

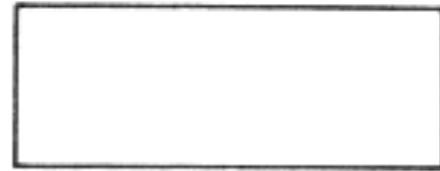
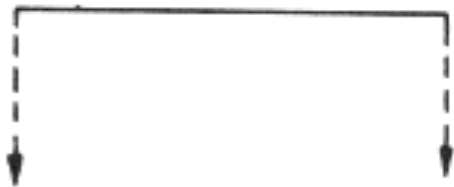
a



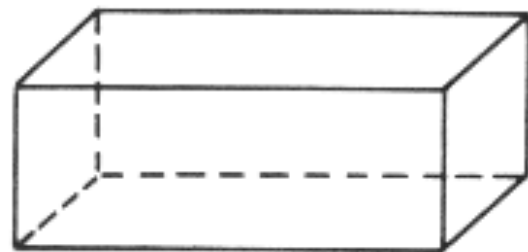
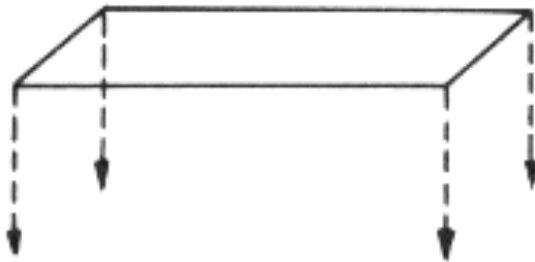
b



c



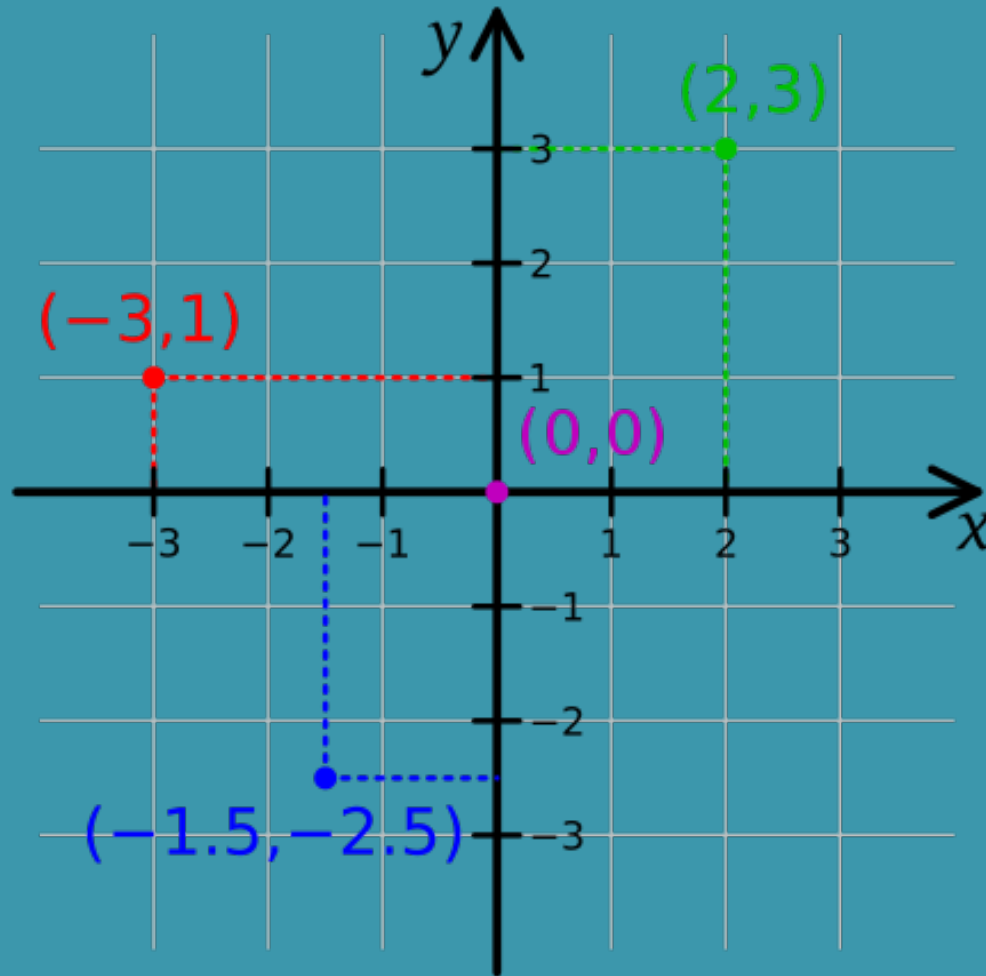
d



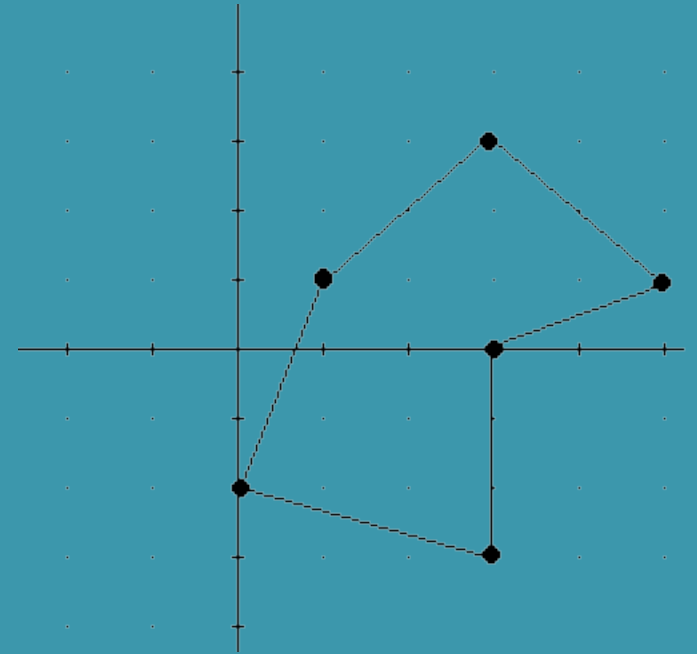
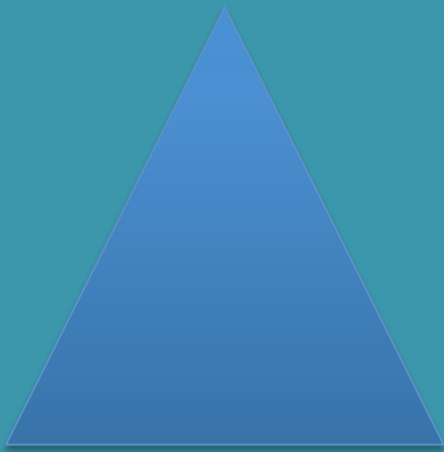
Geometric primitives

- Point
- Line
- Shape
- Curve
- Polygon
- Plane

Cartesian coordinate system



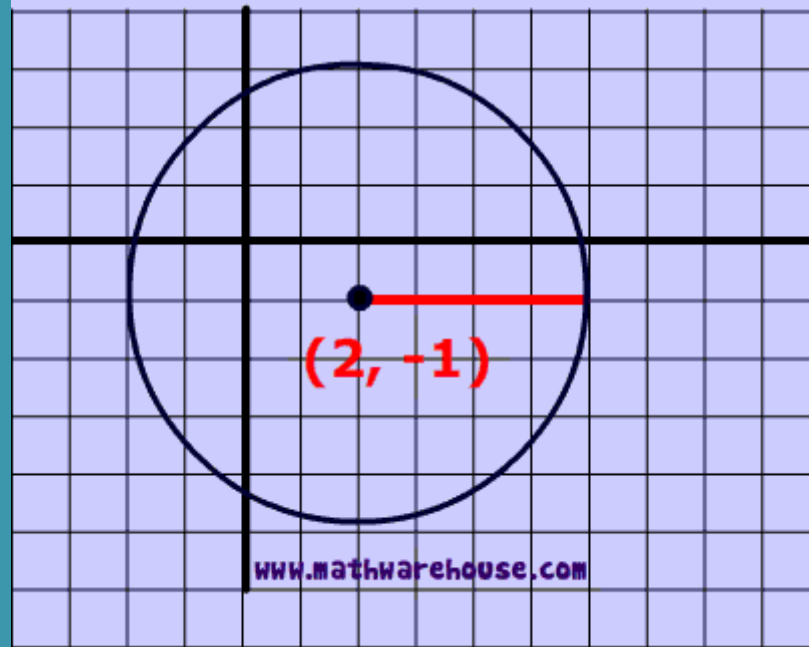
Polygons



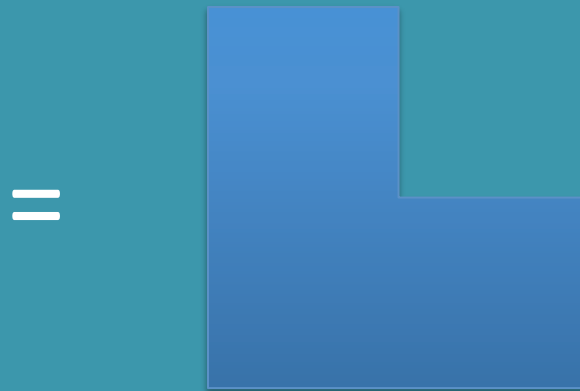
Circle

- $r^2 = x^2 + y^2$

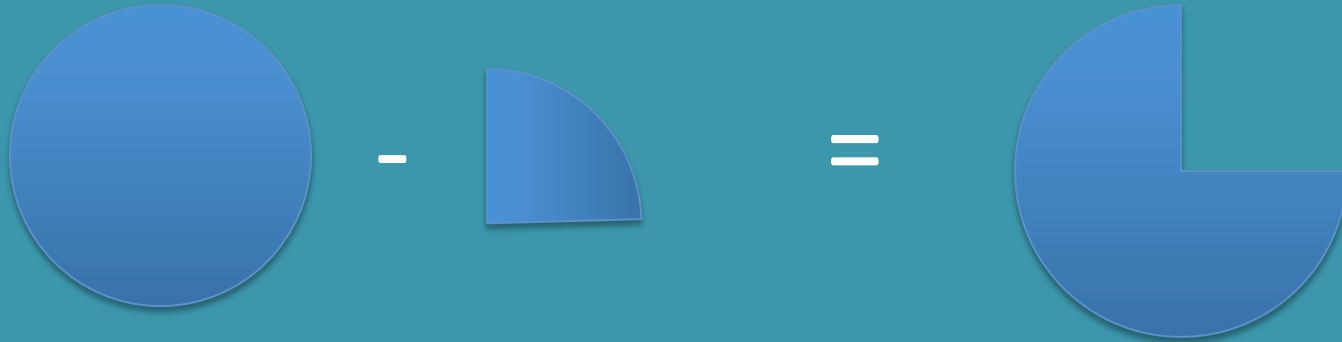
$$(x - h)^2 + (y - k)^2 = r^2$$
$$(x - 2)^2 + (y - -1)^2 = 4^2$$
$$(x - 2)^2 + (y + 1)^2 = 4^2$$



Composite shapes



Composite shapes



Vector Graphics

Vector graphics is the use of geometrical primitives such as points, lines, curves, and shapes or polygon(s), which are all based on mathematical expressions, to represent images in computer graphics.

Project 1: Vector primitives

- Design an abstract composition using the geometric primitives (See geometric abstraction examples).
- Use points, lines, shapes, curves and at least 4 values to produce an aesthetically pleasing image in the **Processing** programming environment.
- Use the functions explained in Chapter 3 (Reas & Fry, 2010)

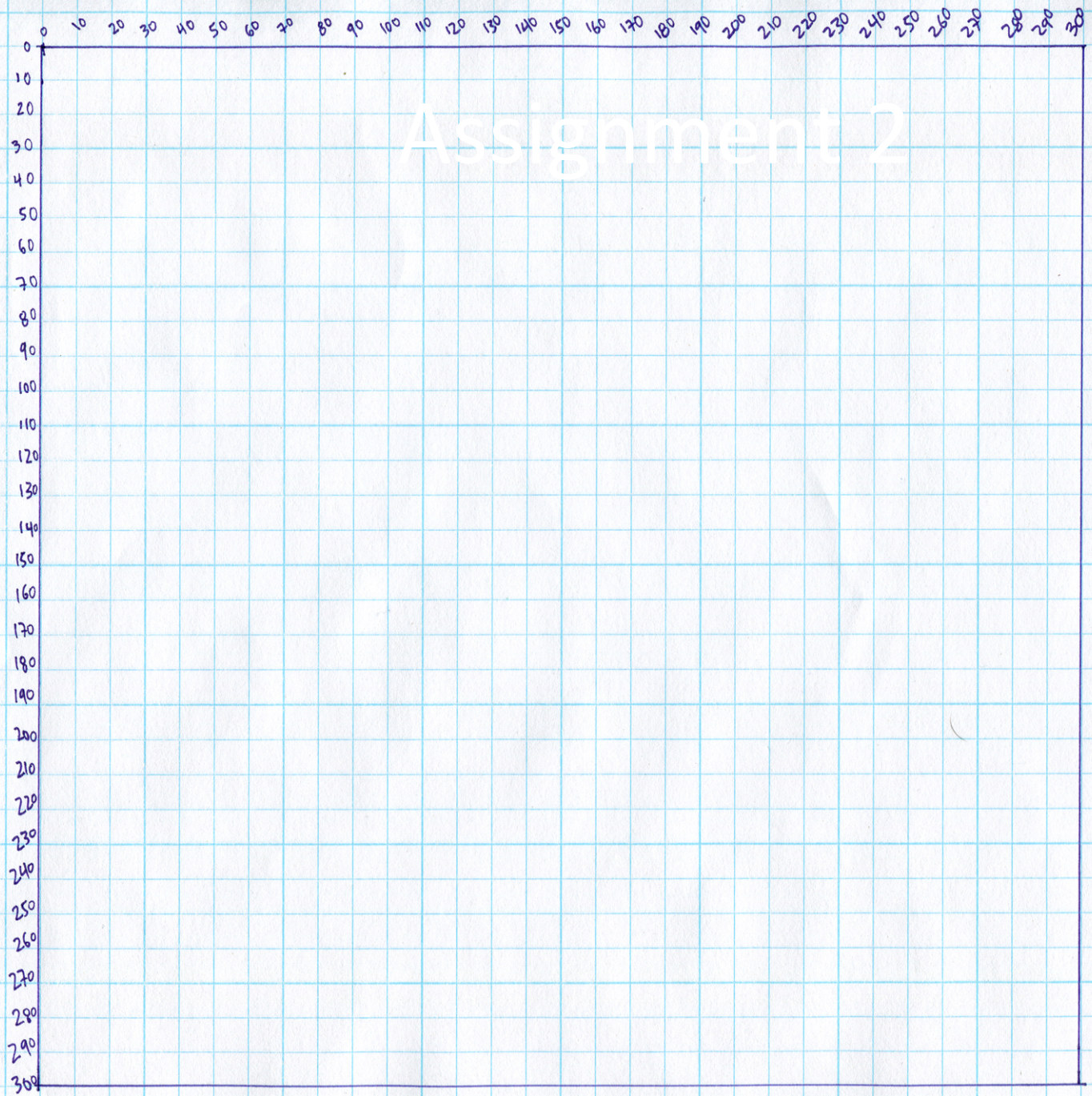
Assignment 2

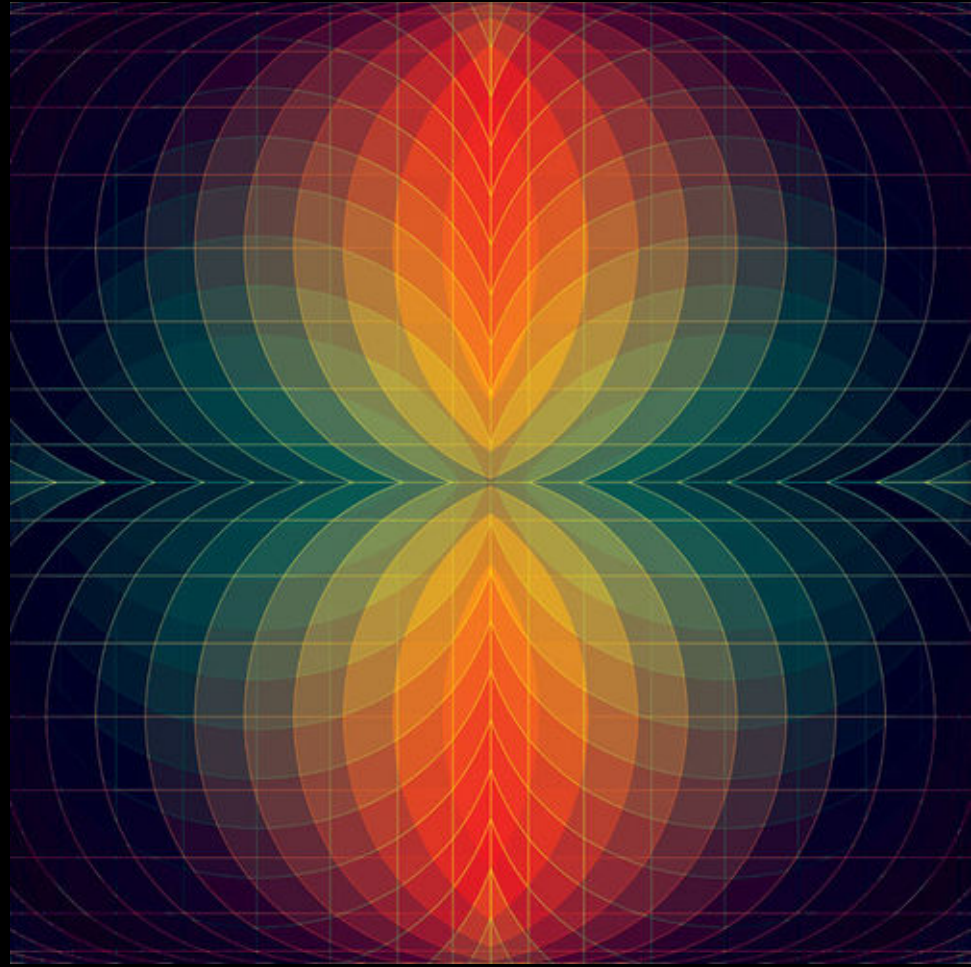
CGT 215

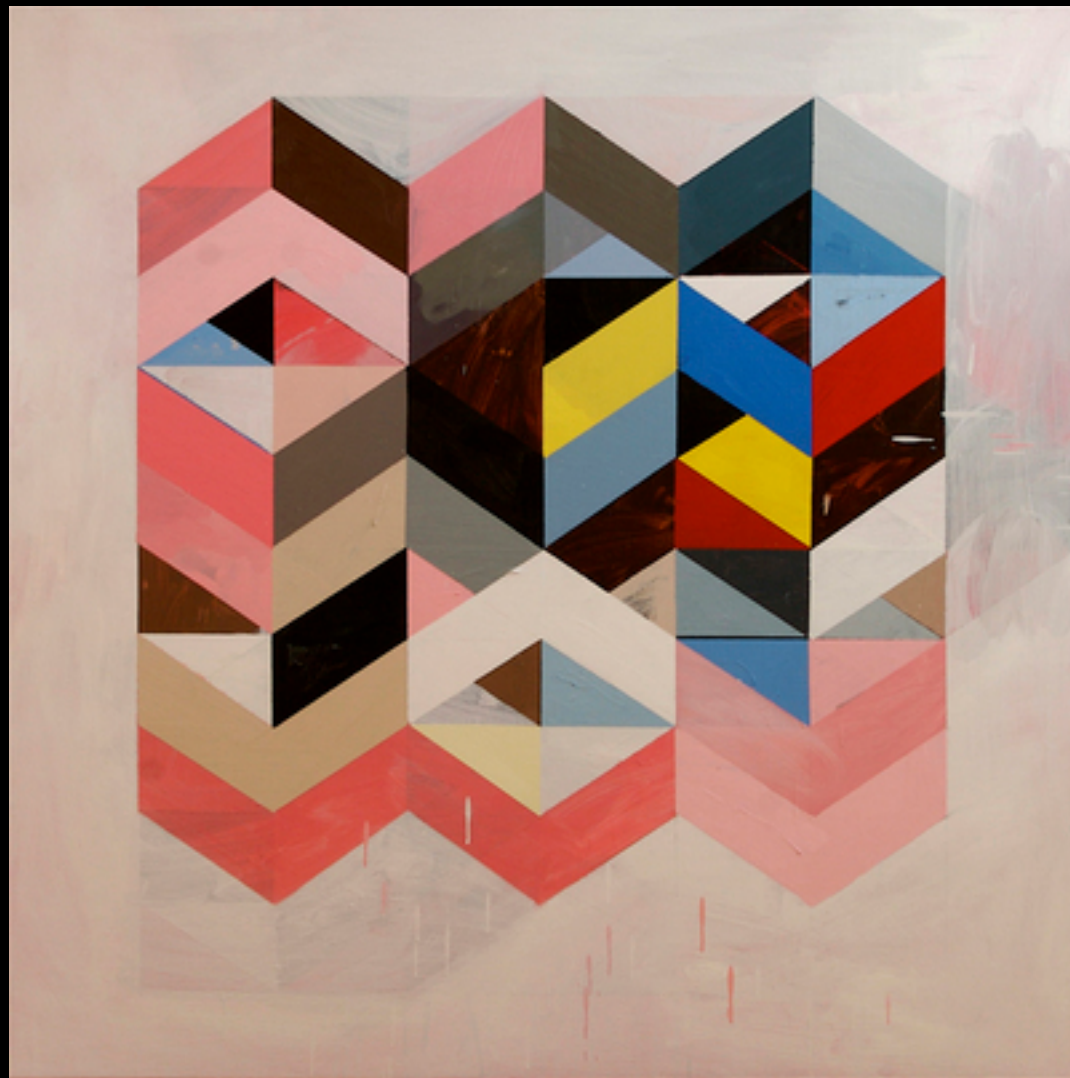
VECTOR
PRIMITIVES
GRID

300 x 300 UNITS

THUMBNAIL IDEAS

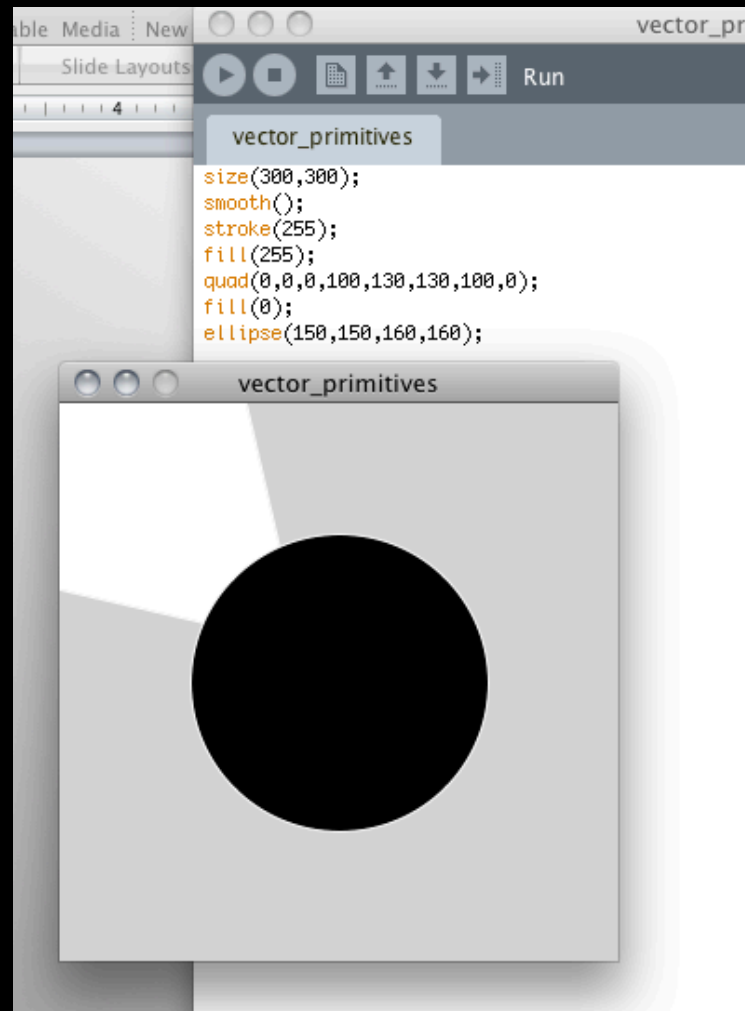






Project 1

- Vector primitives





CGT 215
Computer Graphics Programming I

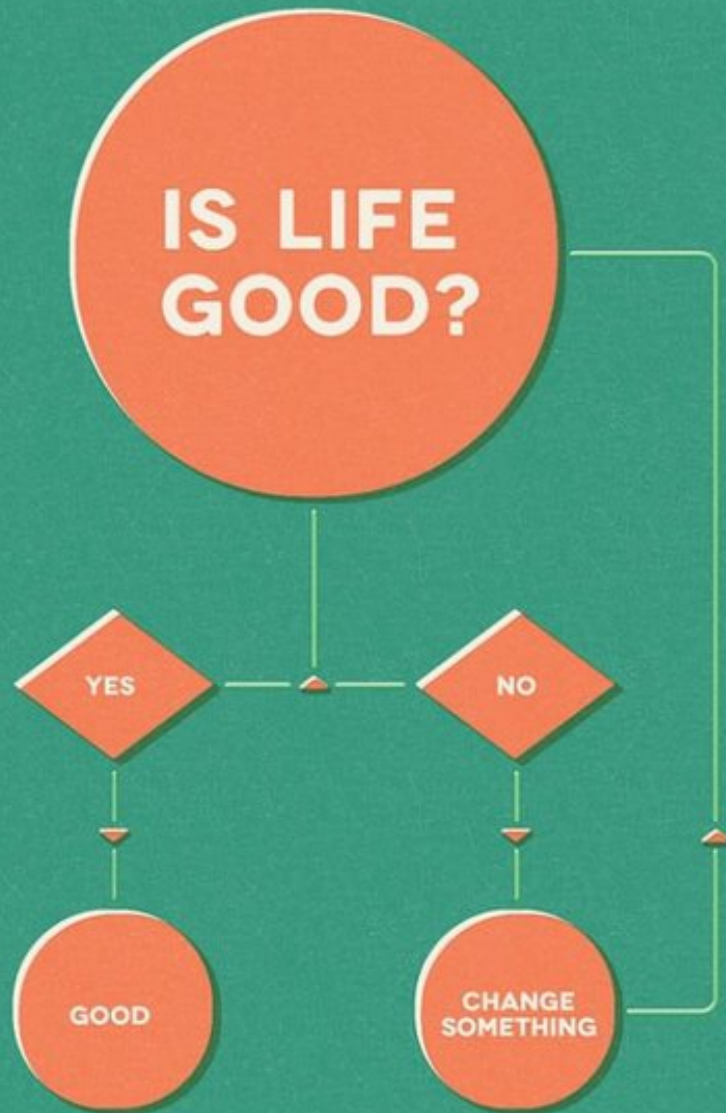
Variables

Variables

Data type	Meaning	Value range
Int	Whole numbers	-2,147 Million to 2,147 Million
float	Floating point (decimals)	3.40282347E+38 to -3.40282347E+38
boolean	T F statements	True or False
string	Store words or sentences	“for example...”
color	Store colors	~ 17 Million

Assignment 3

On a white piece of paper, sketch a flow diagram of your morning routine. Be as detailed as possible.





CGT 215
Computer Graphics Programming I

If... else

The background is a dark blue wireframe grid. In the bottom-left corner, there is a 3D wireframe model of a hand, rendered in a light blue color. The text is centered in the upper half of the image.

CGT 215
Computer Graphics Programming I

And - Or

Relational Operators

> Greater than

< Less than

>= Greater than or equal to

<= Less than or equal to

== Equality

!= Inequality

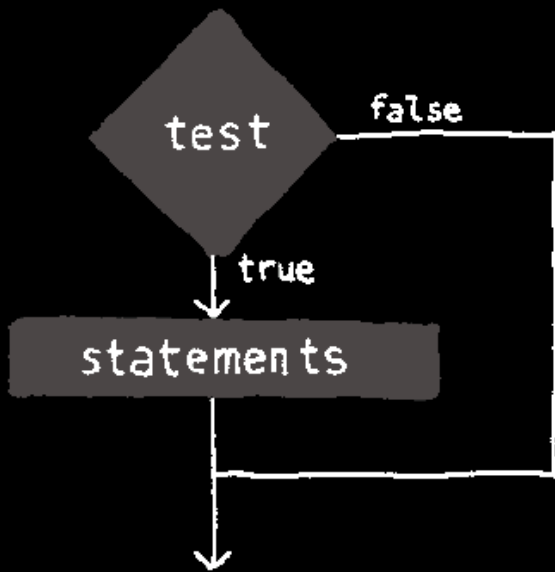
Logical Operators

|| Logical OR

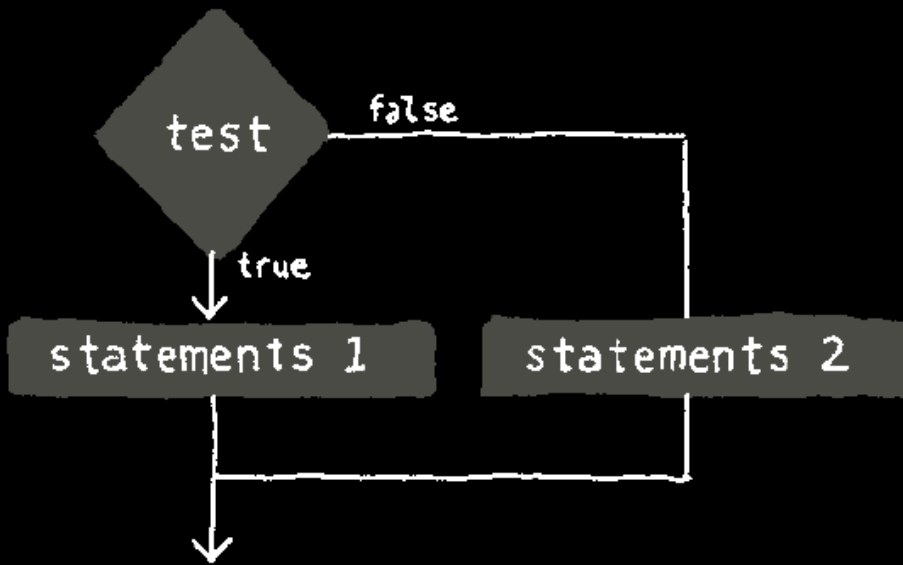
&& Logical AND

! Logical NOT

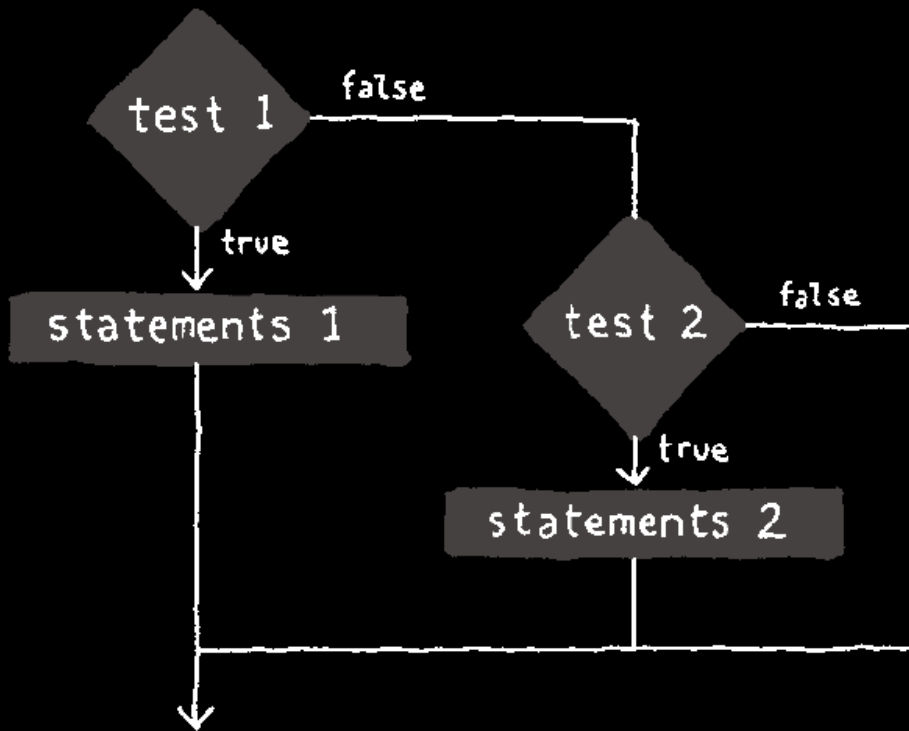
Expression (test)	Result
(5 == 4)	False
(5 == 5)	True
!(23 < 50)	False
(23 > 3) && (23 < 8)	False
(23 > 3) (23 < 8)	True



```
if (test) {  
    statements  
}
```



```
if (test) {  
    statements 1  
} else {  
    statements 2  
}
```

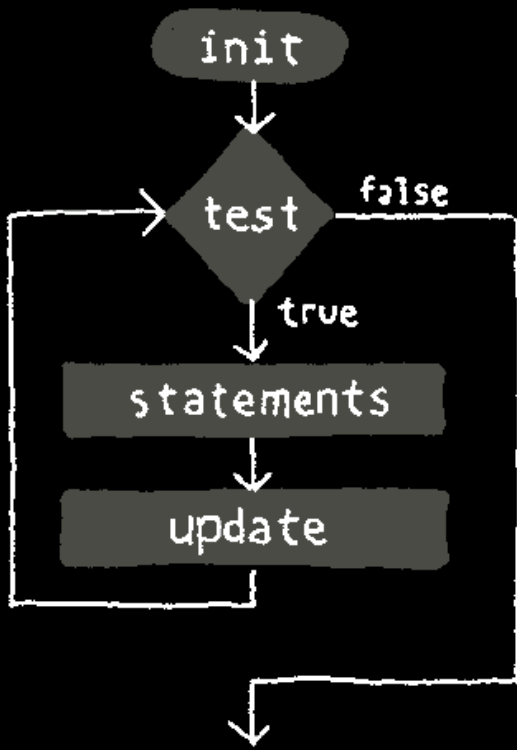


```
if (test 1) {  
    statements 1  
} else if (test 2) {  
    statements 2  
}
```



CGT 215
Computer Graphics Programming I

Loops



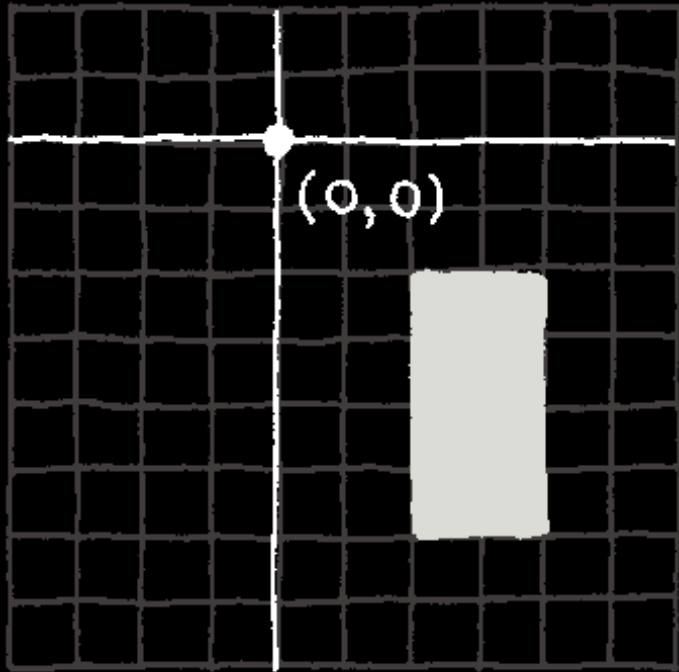
```
for (init; test; update) {  
  statements  
}
```

A wireframe mesh of a hand is visible in the lower-left corner, and a large wireframe cube is centered in the background. The background is black, and the wireframes are composed of thin white lines.

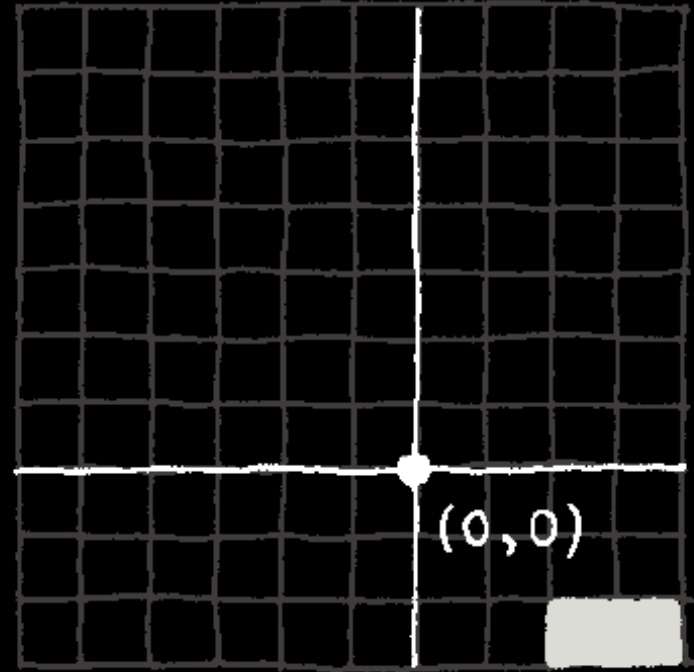
CGT 215
Computer Graphics Programming I

Geometric Transformations

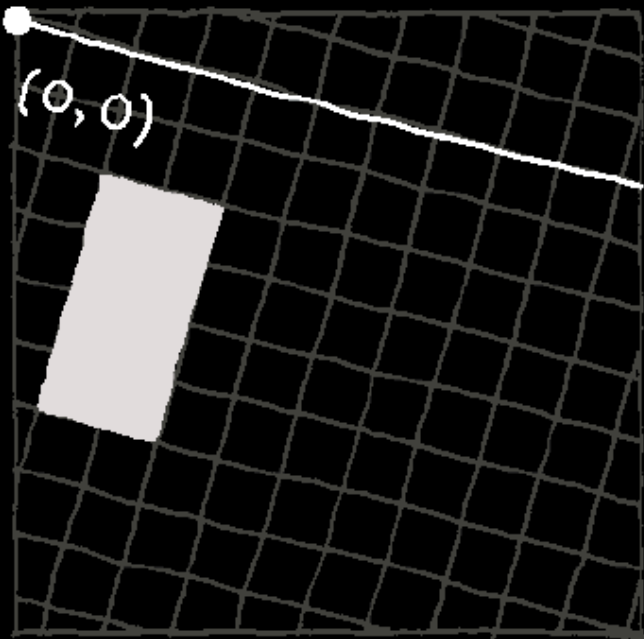
```
translate(40, 20);  
rect(20, 20, 20, 40);
```



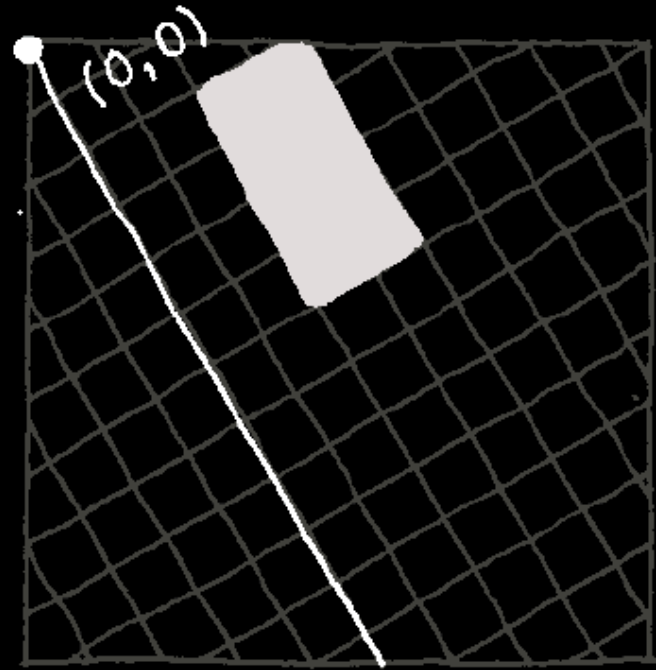
```
translate(60, 70);  
rect(20, 20, 20, 40);
```



```
rotate(PI/12);  
rect(20, 20, 20, 40);
```



```
rotate(-PI/3);  
rect(20, 20, 20, 40);
```



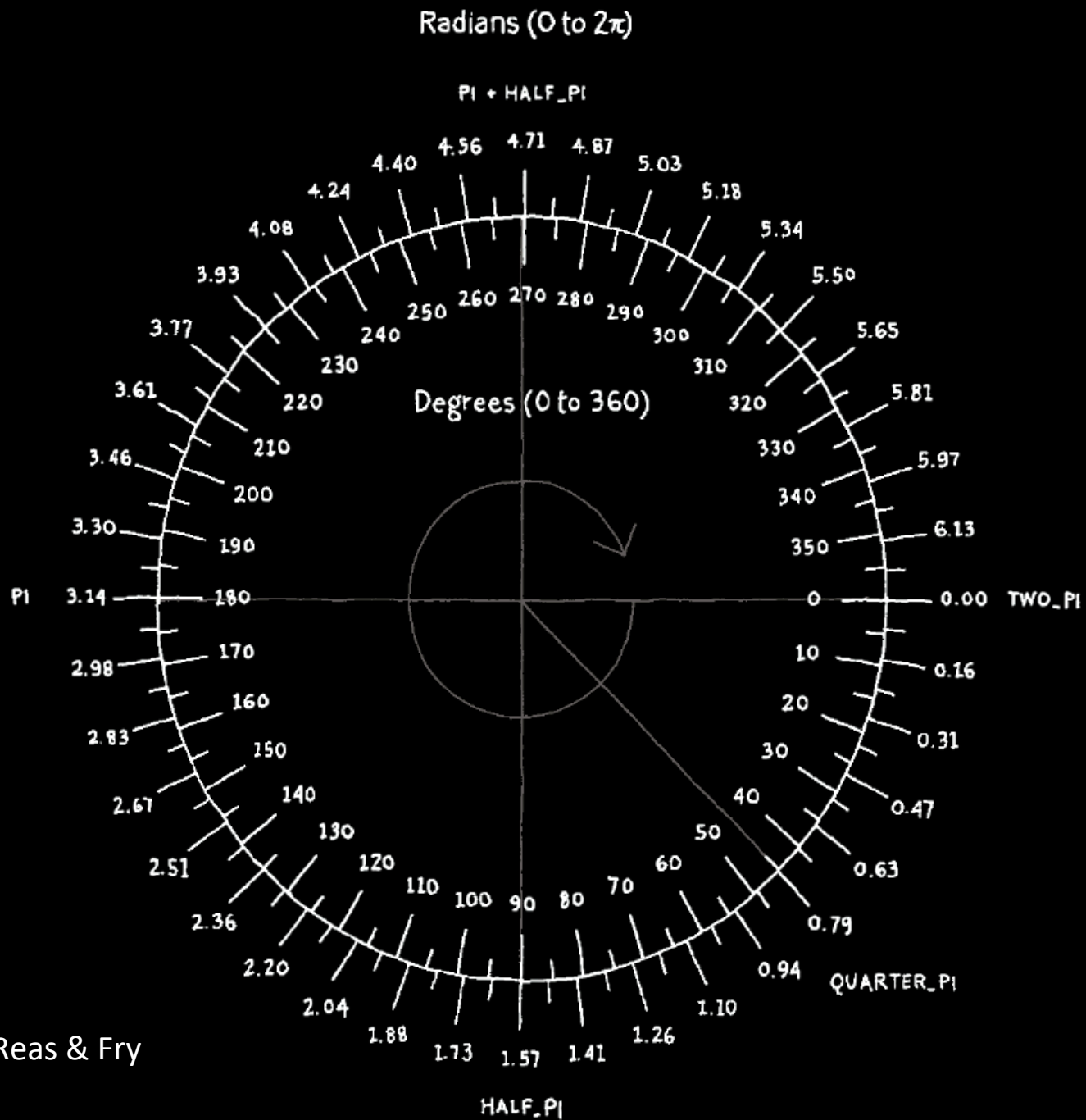
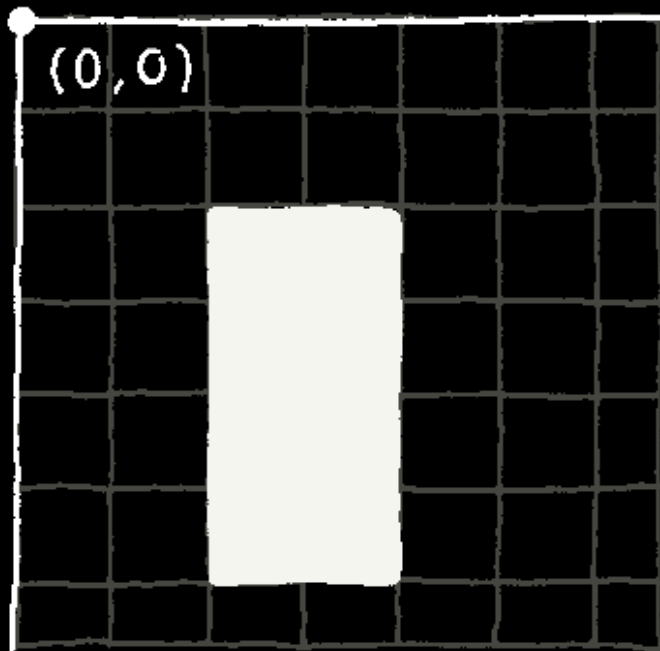
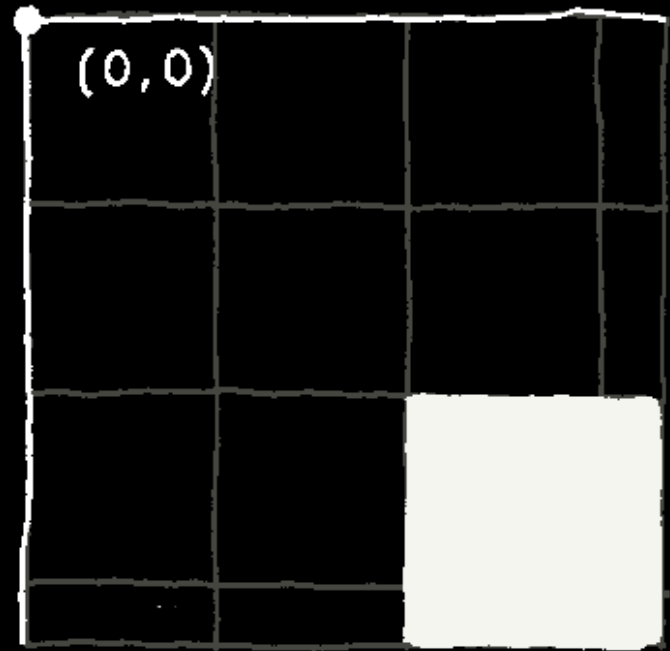


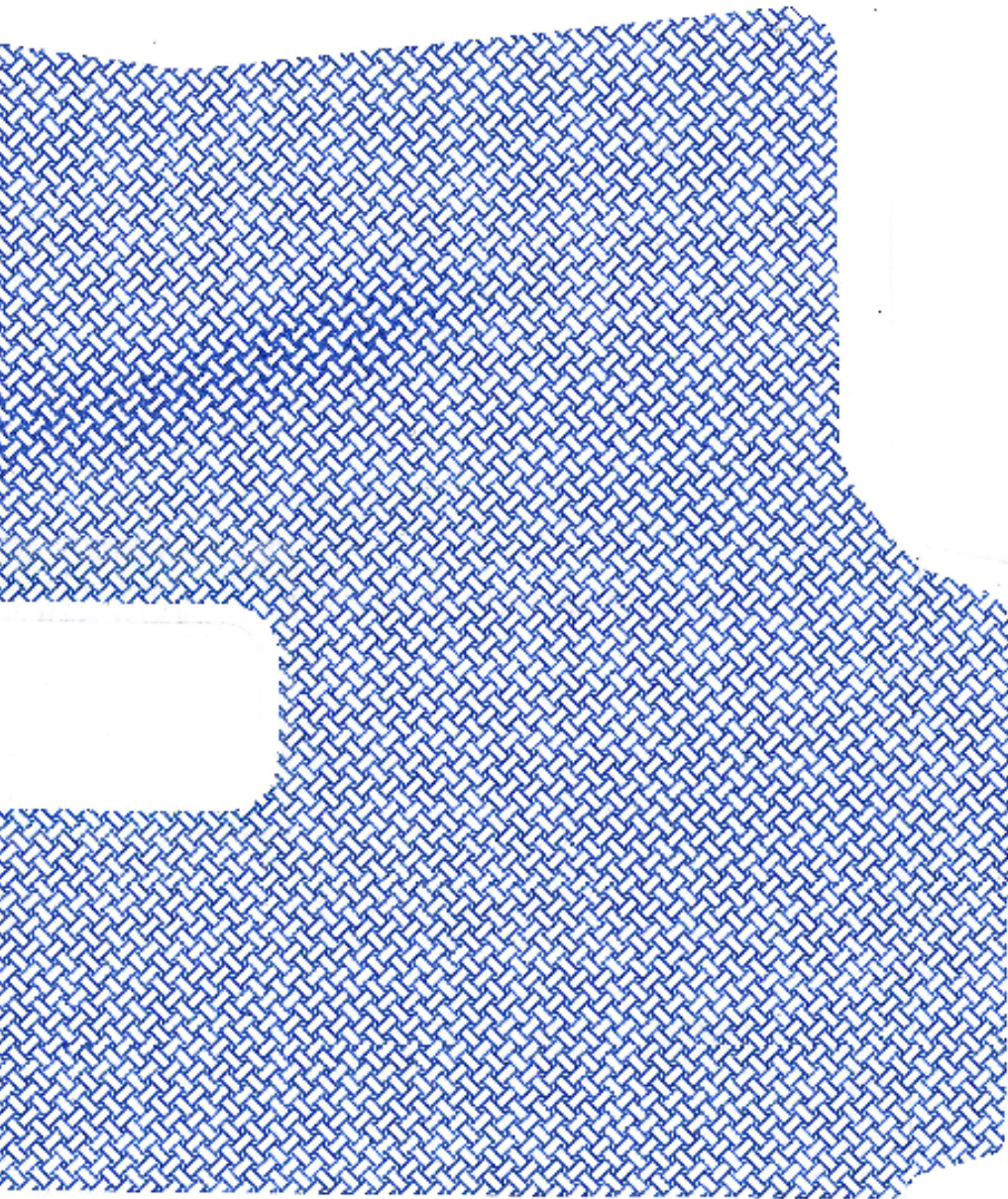
Image from Reas & Fry

```
scale(1.5);  
rect(20, 20, 20, 40);
```



```
scale(3);  
rect(20, 20, 20, 40);
```





Project 2

Recreate the one or more mail envelope patterns provided in the attached PDF. You may use **For loops**, **If statements** or **geometric transformations** to produce a visual output that resembles the original pattern.

A wireframe mesh of a hand is visible in the lower-left corner, and a large wireframe cube is centered in the background. The entire scene is rendered in a light blue color on a black background.

CGT 215
Computer Graphics Programming I

Digital Images

Raster Images

- Have a given resolution
- Consist of pixels
- Photorealistic

Examples:

JPG, GIF, TIF, BMP, PNG
(different types of compression)



Vector Images

- A database of 2D objects
- Have no fixed resolution
- Does not consist of pixels
- Can be scaled arbitrarily
- Graphics stored as algorithms
- Simplified as geometric elements-objects

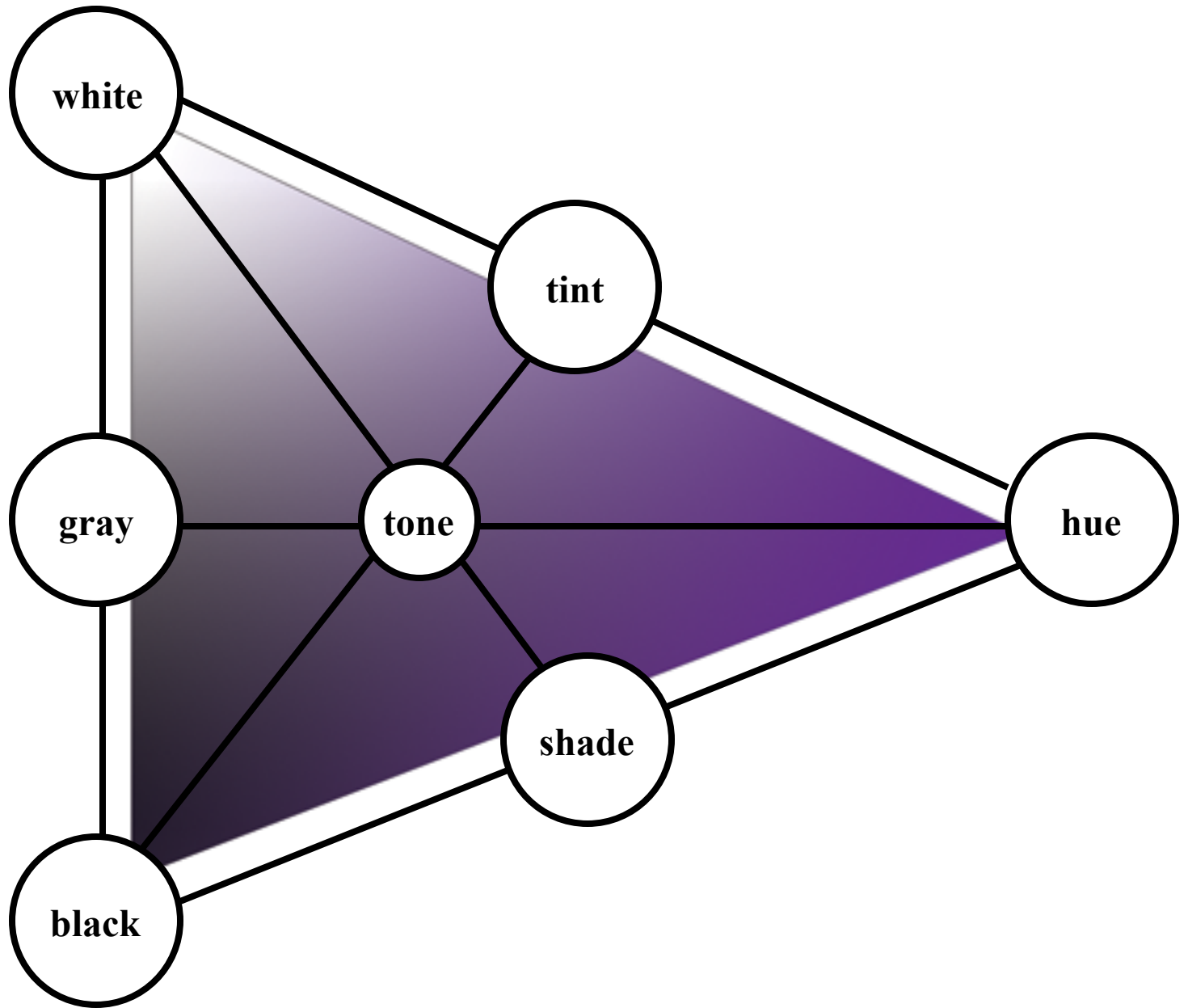
Examples: PDF, SWF, clip arts,
SVG, AI





CGT 215
Computer Graphics Programming I

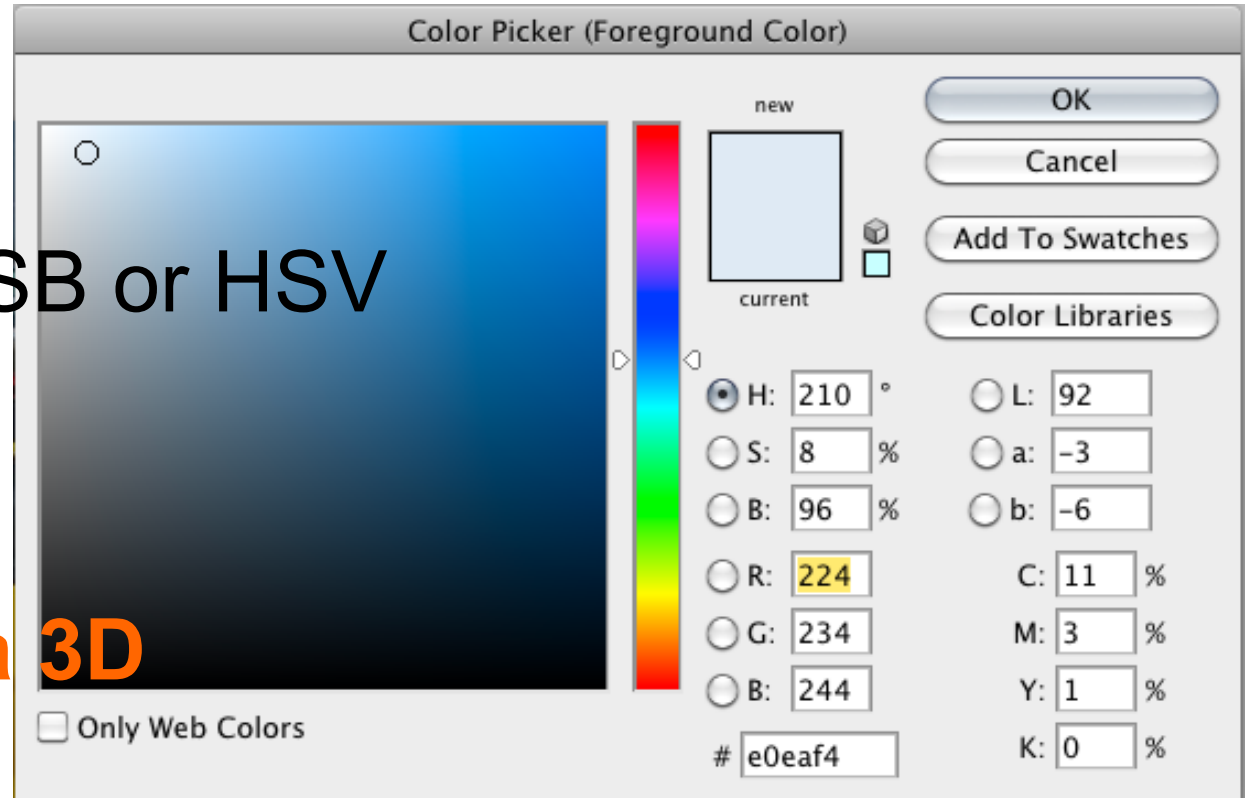
Color



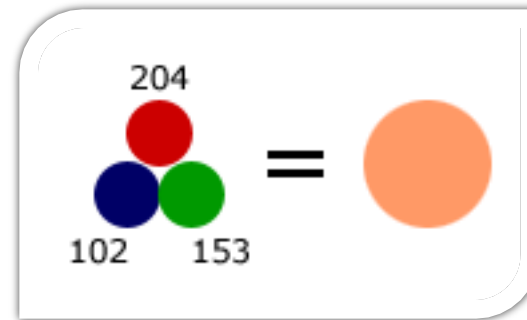
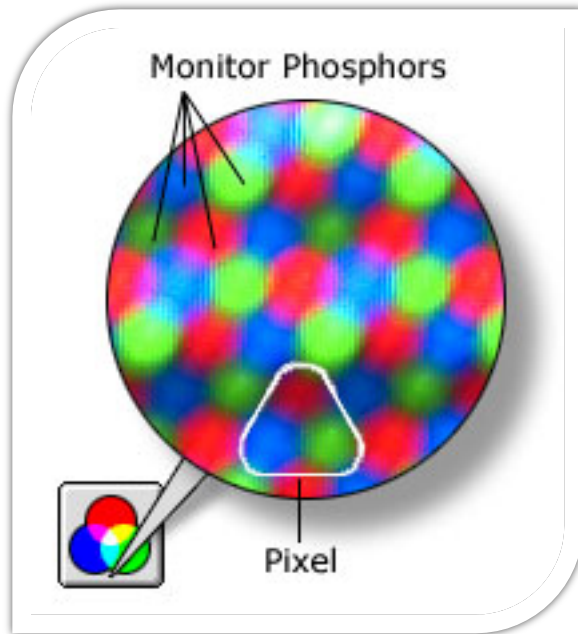
Color Systems in CG

- RGB
- CMY+K
- HLS or HSB or HSV
- Lab

- **Color is a 3D system!**

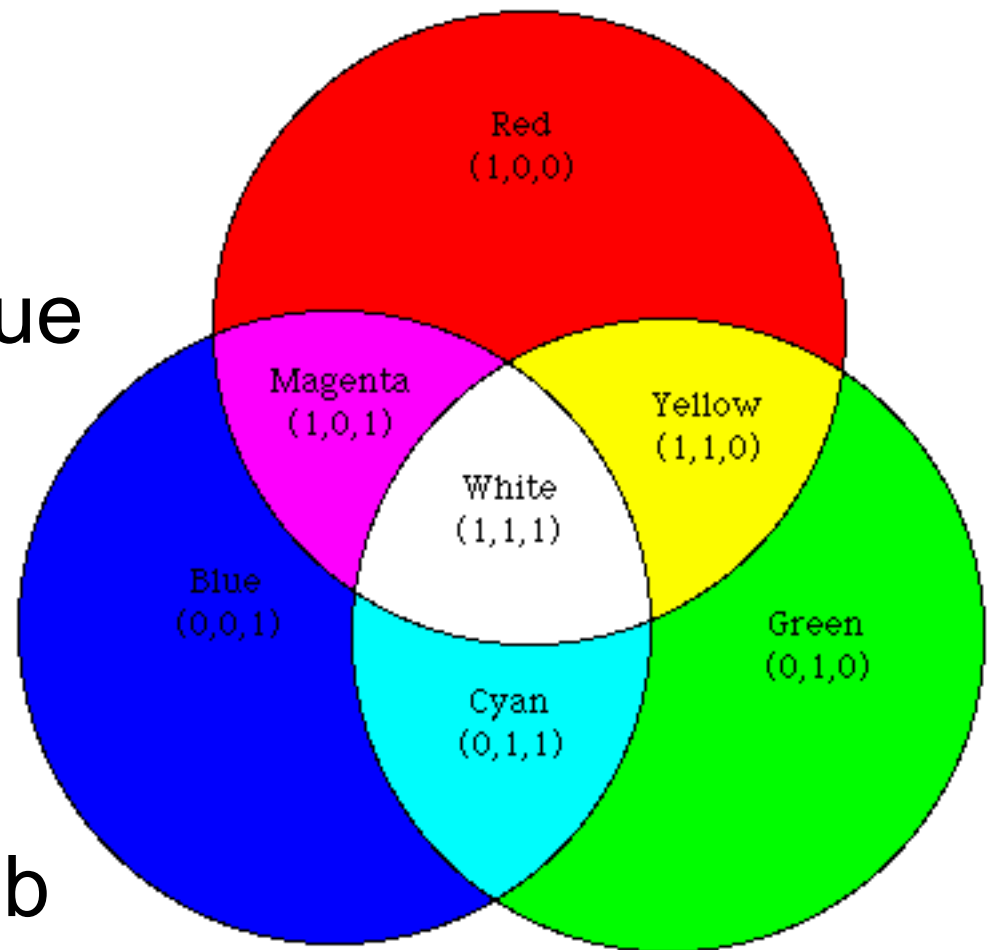


Pixels and values



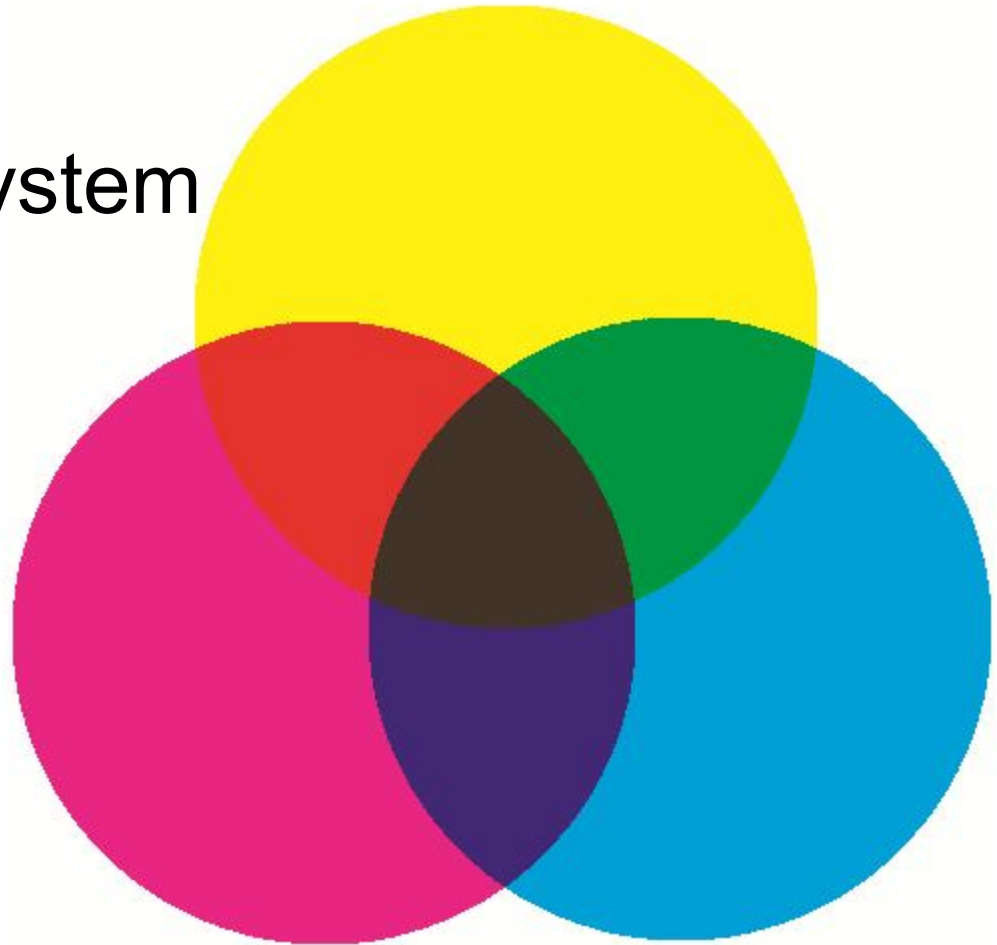
RGB

- Additive system
- Reference to value of Pixels
- 0 to 255
- Hex is subset of RGB: #0000FF
- Screen color, web

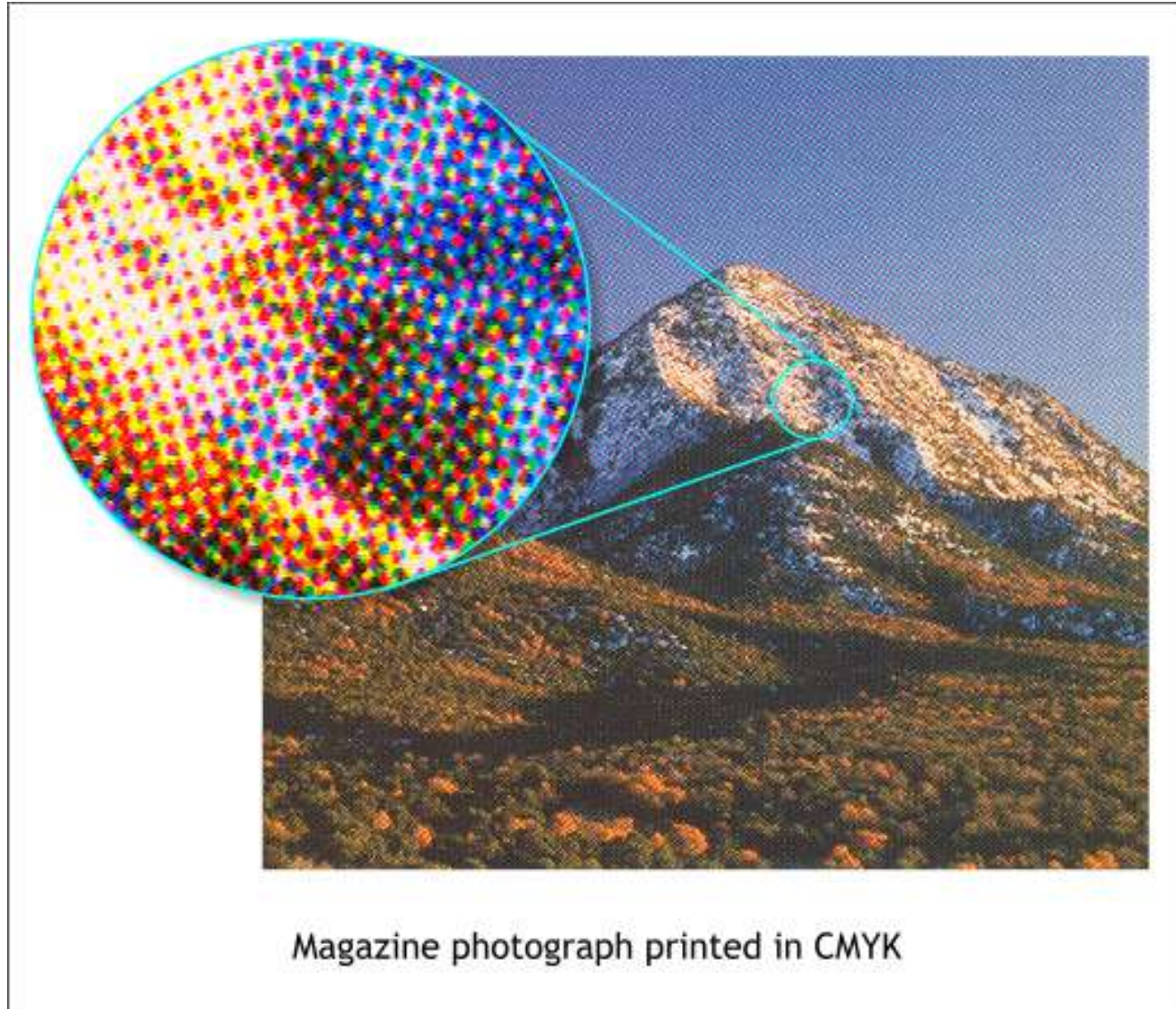


CMYK

- Subtractive system
- Reference to pigments
- %
- Print
- Add “K”



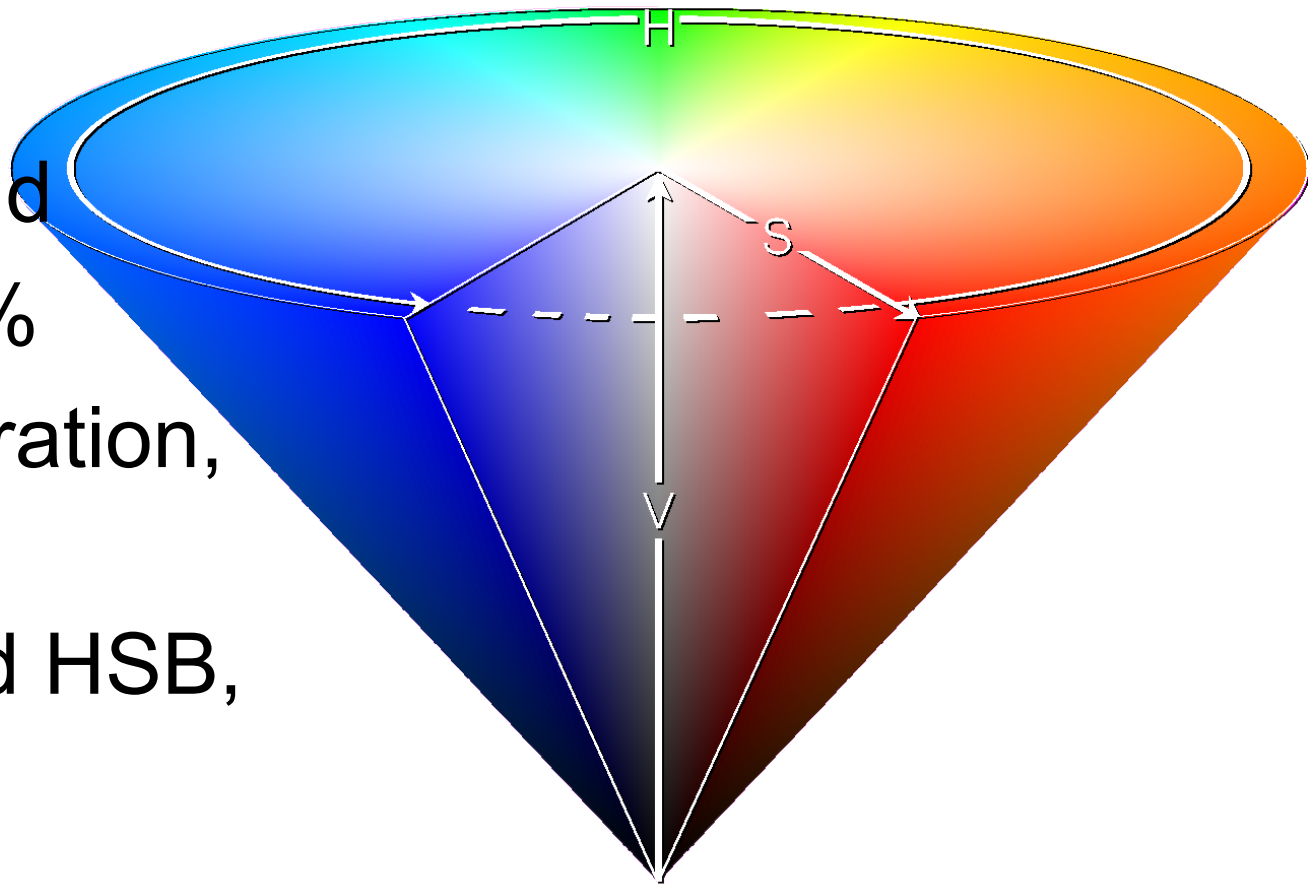
CMYK dot



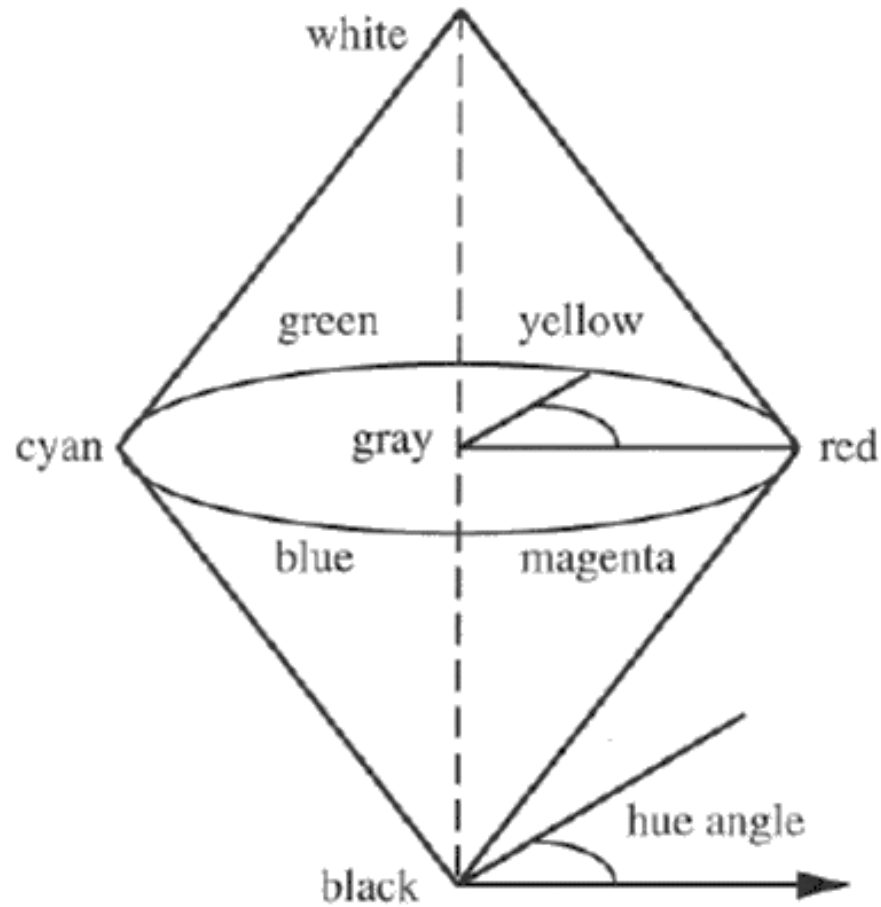
Magazine photograph printed in CMYK

HSV

- Intuitive
- Easy to find
- degrees, %
- Hue, Saturation, Value
- Also called HSB, HLS

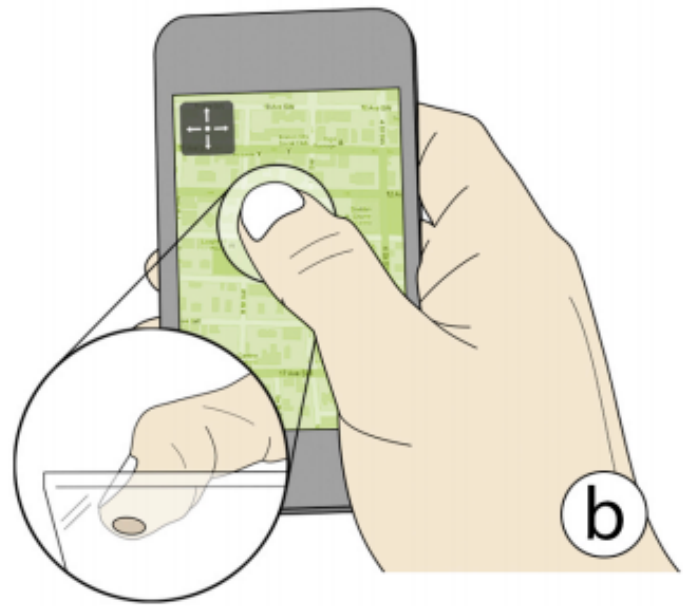
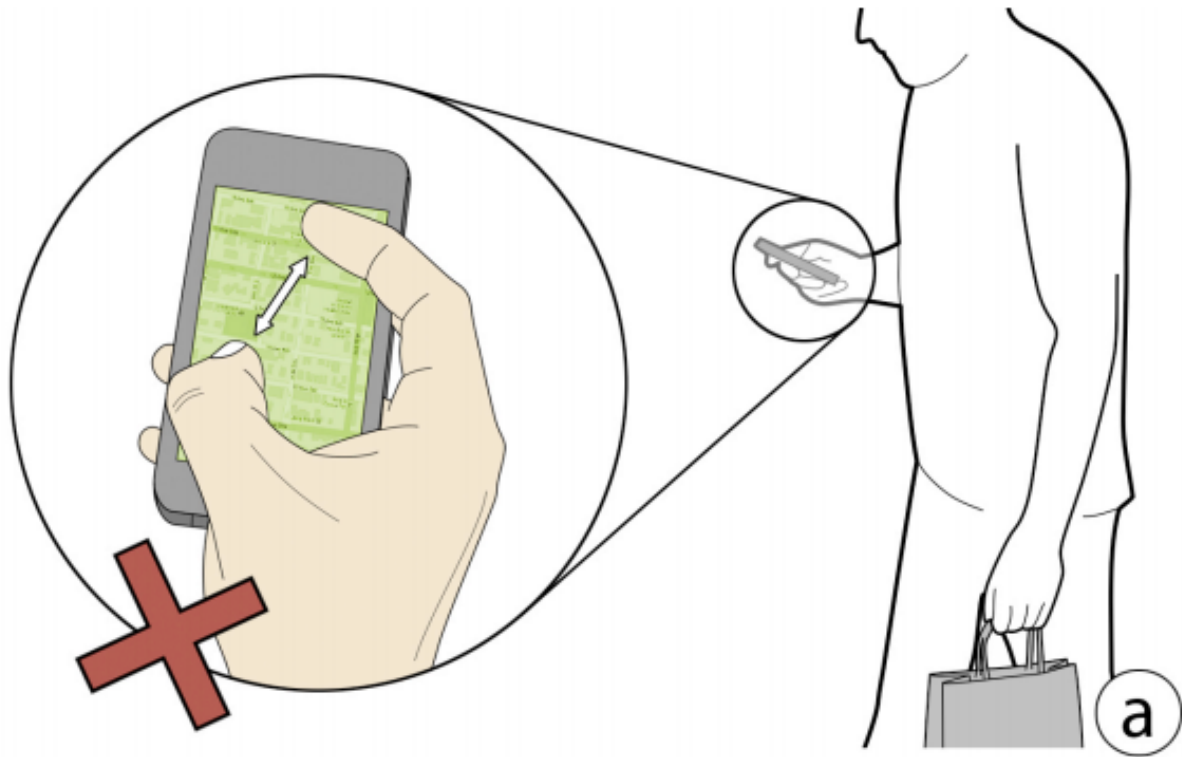


RGB , CMYK & HSV





CGT 215
Computer Graphics Programming I
Mouse & Keyboard
Inputs



Mouse

mouseButton

mouseClicked()

mouseDragged()

mouseMoved()

mousePressed()

mousePressed

mouseReleased()

mouseWheel()

mouseX

mouseY

pmouseX

pmouseY

Keyboard

key

keyCode

keyPressed()

keyPressed

keyReleased()

keyTyped()



CGT 215
Computer Graphics Programming I

Functions



Project 3: “Carnival Ride”

Create a program that simulates a carnival ride that is controlled by the user.

Your program must use the following elements:

- Geometric Transformations (Rotate, Scale)
- on/off states, backwards
- Import Raster/Vector images

Assignment 4

Concept brief:

- What are you going to do for project 3?
- What kind of programming elements will you be using?
- How will you create your images?
- Interactivity: What will trigger the different behaviors?



The background is a solid green color with a white wireframe pattern. The wireframe consists of various geometric shapes, including a large, irregular polygon in the center and a hand-like shape in the bottom-left corner. The hand is composed of many small, interconnected polygons, giving it a mesh-like appearance. The overall effect is a technical, computer-graphics aesthetic.

CGT 215
Computer Graphics Programming I

Functions

Functions

- Also called Methods, Subroutines, Subprograms
- A program inside a program
- User-defined or Language defined
- Can return values but it doesn't have to

Functions

General Form

```
Return-type method-name ( argument-type argument-name, ...){  
    statements;  
    return value  
}
```

Functions

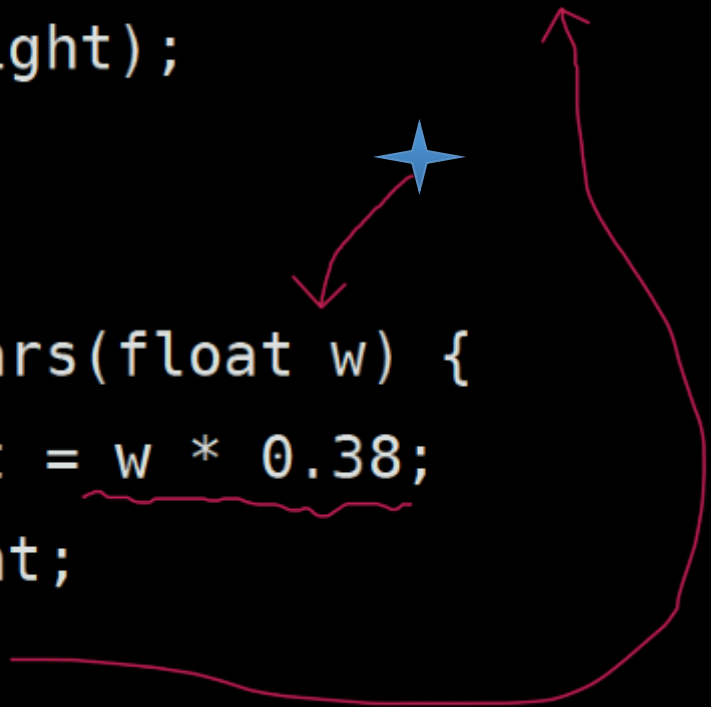
General Form

```
void plant( int x,int y){  
    translate(x,y);  
    //No return value  
}
```

```
float mars ( float w ){  
    float newWeight= w*0.38;  
    return newWeight;  
    //this value can be used later on the program  
}
```

```
void setup() {  
    float yourWeight = 132;  
    float marsWeight = calculateMars(yourWeight);  
    println(marsWeight);  
}
```

```
float calculateMars(float w) {  
    float newWeight = w * 0.38;  
    return newWeight;  
}
```

A blue star is positioned between the two code blocks. A red arrow points from the star down to the parameter 'w' in the function definition. Another red arrow points from the star up to the function call 'calculateMars(yourWeight)' in the setup function. A large red bracket on the right side of the image spans from the function definition up to the function call, indicating the scope of the function call.

Return Values

```
float d = random(numSides);
```

The background is a dark purple color with a white wireframe pattern. The wireframe consists of various geometric shapes, including a large cube-like structure in the center and a hand-like shape in the bottom-left corner. The text is centered in the upper half of the image.

CGT 215
Computer Graphics Programming I

Arrays []

Arrays

- A list of variables
- Shared name
- Allows to work with lots of variables



Arrays

- Each item in the list is an array element
- We access each element by calling its index number [6]
- Elements must be of the same data type

Arrays

- Each item in the list is an array element
- We access each element by calling its index number [6]
- Elements must be of the same data type:

```
String [] animals={ cow, cat, dog }; ✨
```

Arrays

```
int[] years = { 1920, 1972, 1980, 1996, 2010 };
```

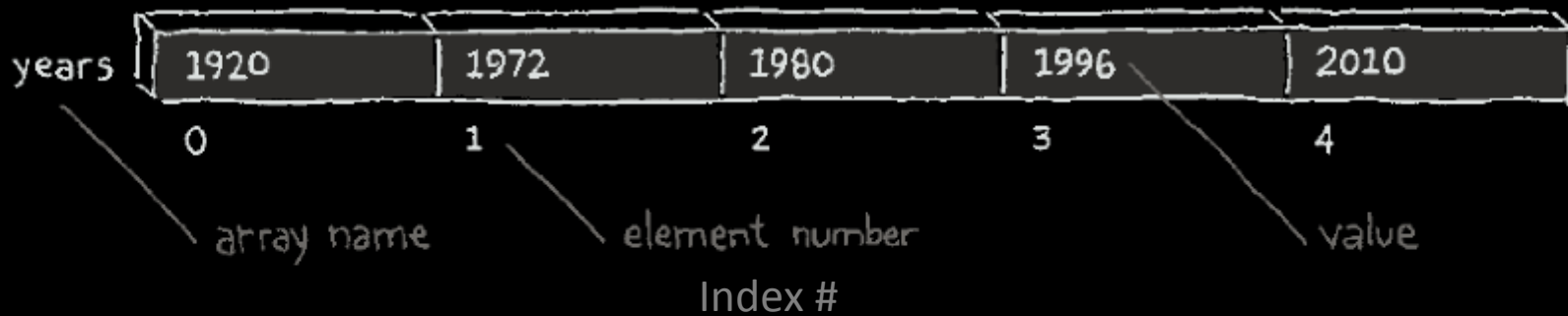


Image from Reas & Fry: "Getting started with Processing"

Arrays

Compact array assignment:

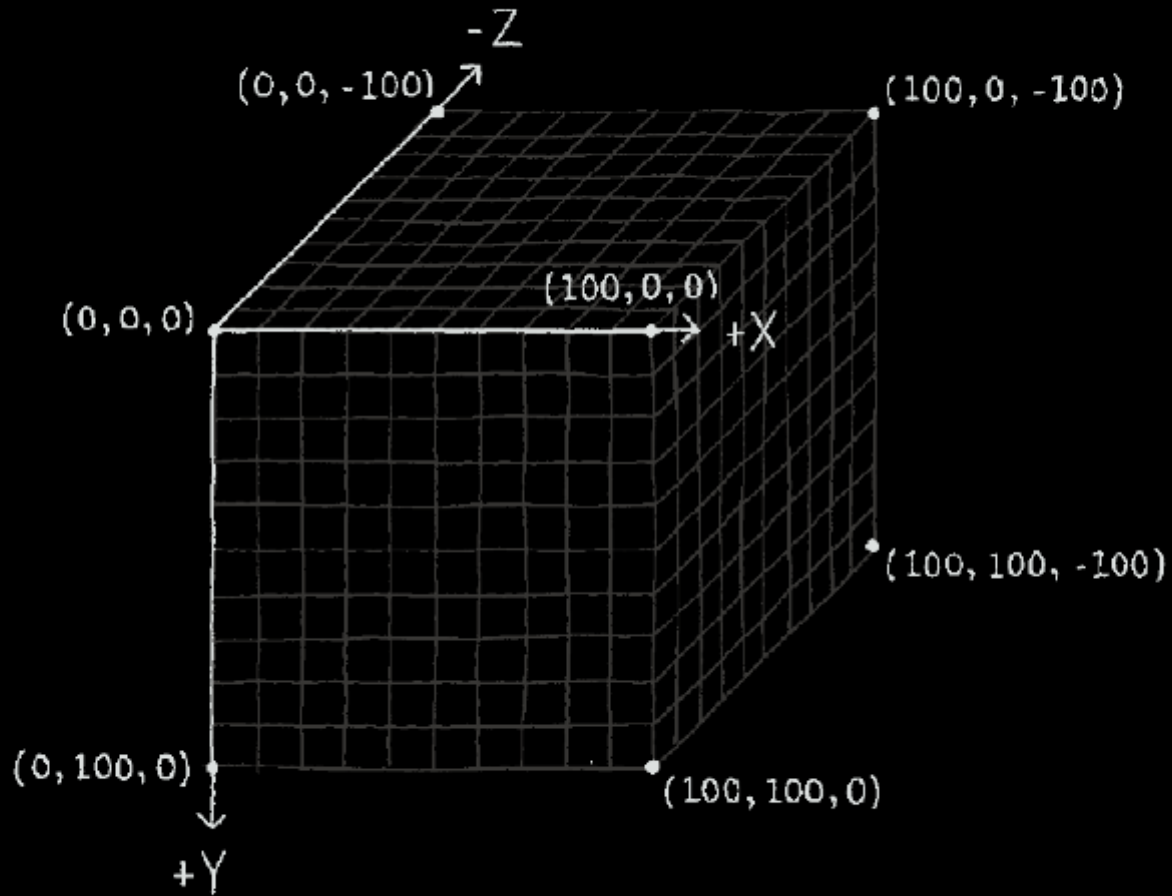
```
int [] x = new int [3000];
```



CGT 215
Computer Graphics Programming I

3D primitives

3D



3D

box()

sphere()

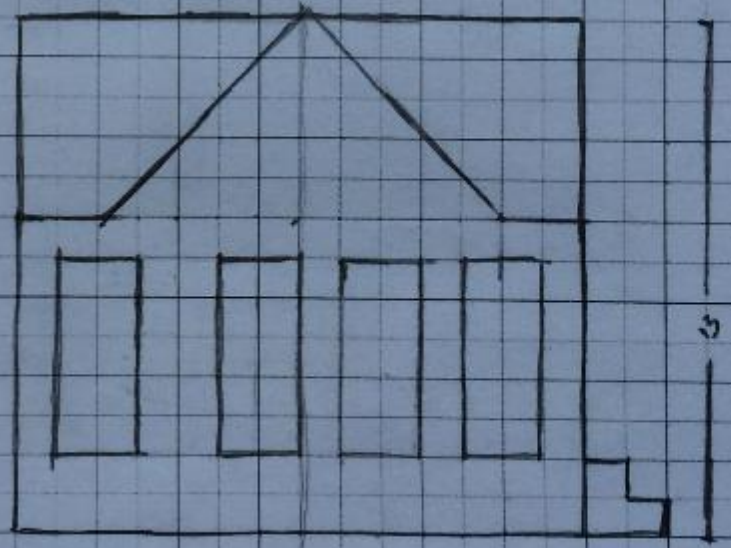
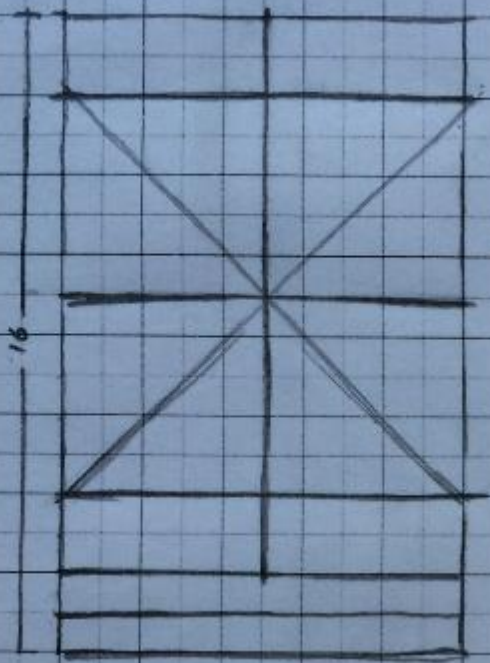
beginShape()

endShape()

vertex()

line()

push/popMatrix...



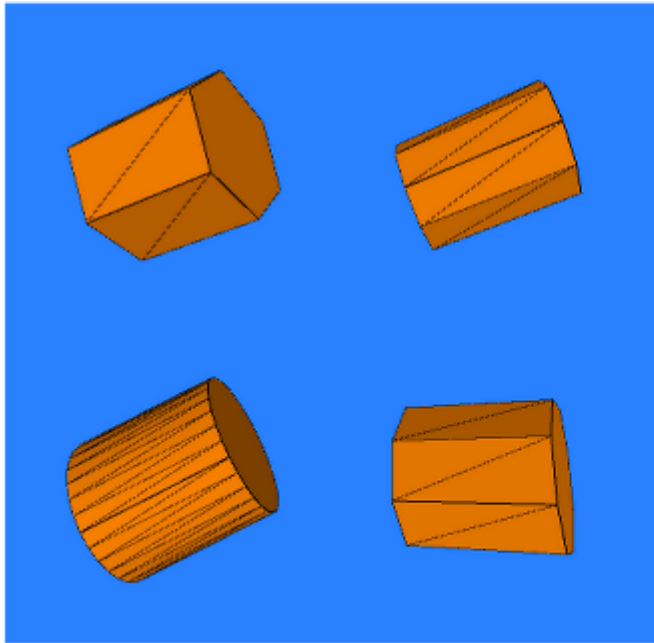
Assignment 5

On a quad piece of paper, sketch the basic geometry of your house in a orthographic projection. Keep the design simple using planes, boxes or spheres only.



Assignment 5

```
// draw body
beginShape(TRIANGLE_STRIP);
for (int i = 0; i < sides + 1; i++) {
  float x = cos( radians( i * angle ) ) * r;
  float y = sin( radians( i * angle ) ) * r;
  vertex( x, y, halfHeight);
  vertex( x, y, -halfHeight);
}
endShape(CLOSE);
```



<http://vormplus.be/blog/article/drawing-a-cylinder-with-processing>

The background is a solid green color with a white wireframe pattern. The wireframe consists of various geometric shapes, including a large cube-like structure in the center and a hand-like shape in the bottom left corner. The text is centered in the upper half of the image.

CGT 215
Computer Graphics Programming I

Objects

Object-oriented programming (OOP)

- Like “real” objects
- Have common characteristics
- Have common functions



Object-oriented programming (OOP)

- A collection of related variables and functions
- A variable is a “Field” and behaviors or functions called “methods”

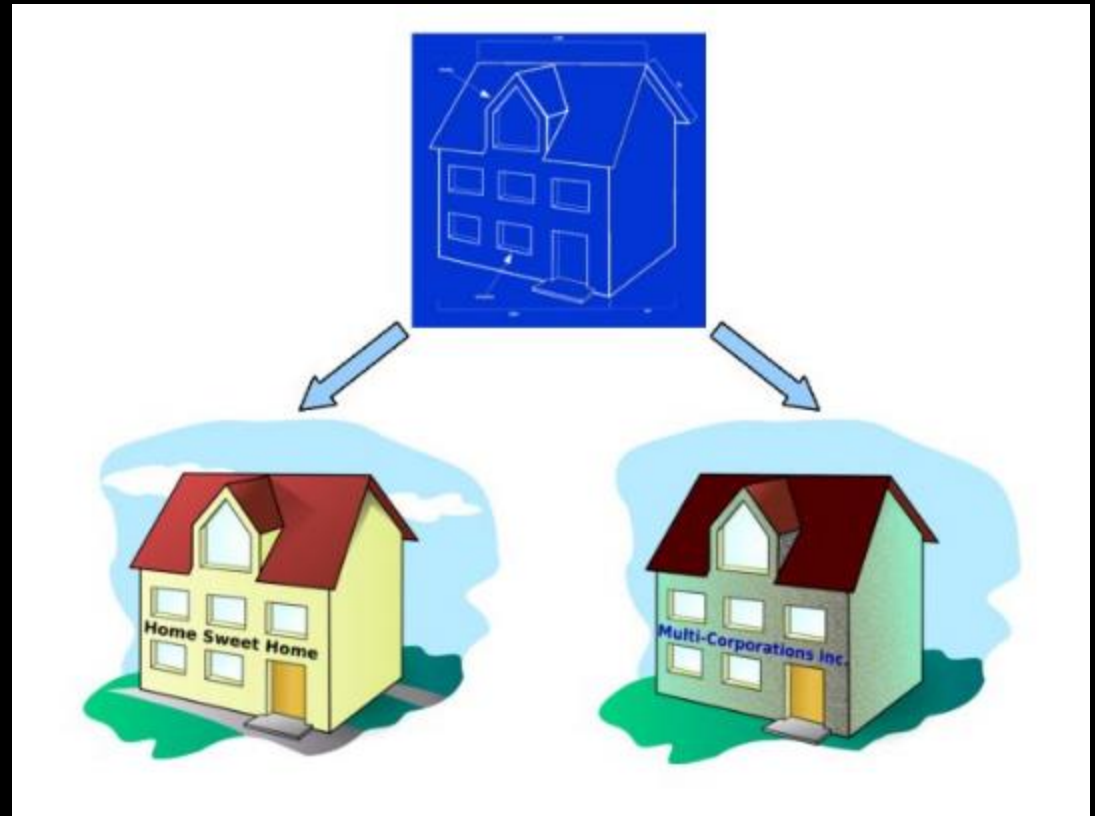



image from www.processing.lyndondaniels.com/

```
class JitterBug {  
  
    float x;  
    float y;  
    int diameter;  
    float speed = 2.5;  
  
    JitterBug(float tempX, float tempY, int tempDiameter) {  
        x = tempX;  
        y = tempY;  
        diameter = tempDiameter;  
    }  
}
```



constructor

```
Train red, blue;
```

```
void setup() {  
  size(400, 400);  
  red = new Train("Red Line", 90);  
  blue = new Train("Blue Line", 120);  
}
```

```
class Train {  
  String name;  
  int distance;  
  Train (String tempName, int tempDistance) {  
    name = tempName;  
    distance = tempDistance;  
  }  
}
```

constructor

Assign "Red Line" to the "name" variable for the "red" object

Assign "90" to the "distance" variable for the "red" object

The background is a solid green color with a white wireframe pattern. The wireframe consists of various geometric shapes, including a large cube-like structure in the center and a hand-like shape in the bottom-left corner. The text is centered in the upper half of the image.

CGT 215
Computer Graphics Programming I

Objects

Final project

The theme for our final project is free and therefore, you must create a concept, design a program and implement it. Take into consideration the different topics covered during this course and apply them in a new and creative way. There are no limitations what to do, except due dates. This is your opportunity to do what you want! Imagine it could be a game, an app or anything from interactive to generative...

Assignment 6

Create a brief that outlines the concept of your final project. Provide sketches and flow diagrams to complement your idea.