2017SW 교육 전문연수

20171028-29

Jonghwa Park
suakii@gmail.com

GYEONGGI SCIENCE HIGH SCHOOL

Simulation – 2015 정보과학 성격

- …….. 또한 이러한 역량을 바탕으로 실세계나 타 학문 분야의 융합 문제들을 컴퓨팅 기반의 시뮬레이션이나…….

Simulation – 2015 정보과학 컴퓨팅 시스템 목표

- … 따라서 이 영역에서는 근사, 난수, 시각화 등의 시뮬레이션 프로그램을 개발하고……

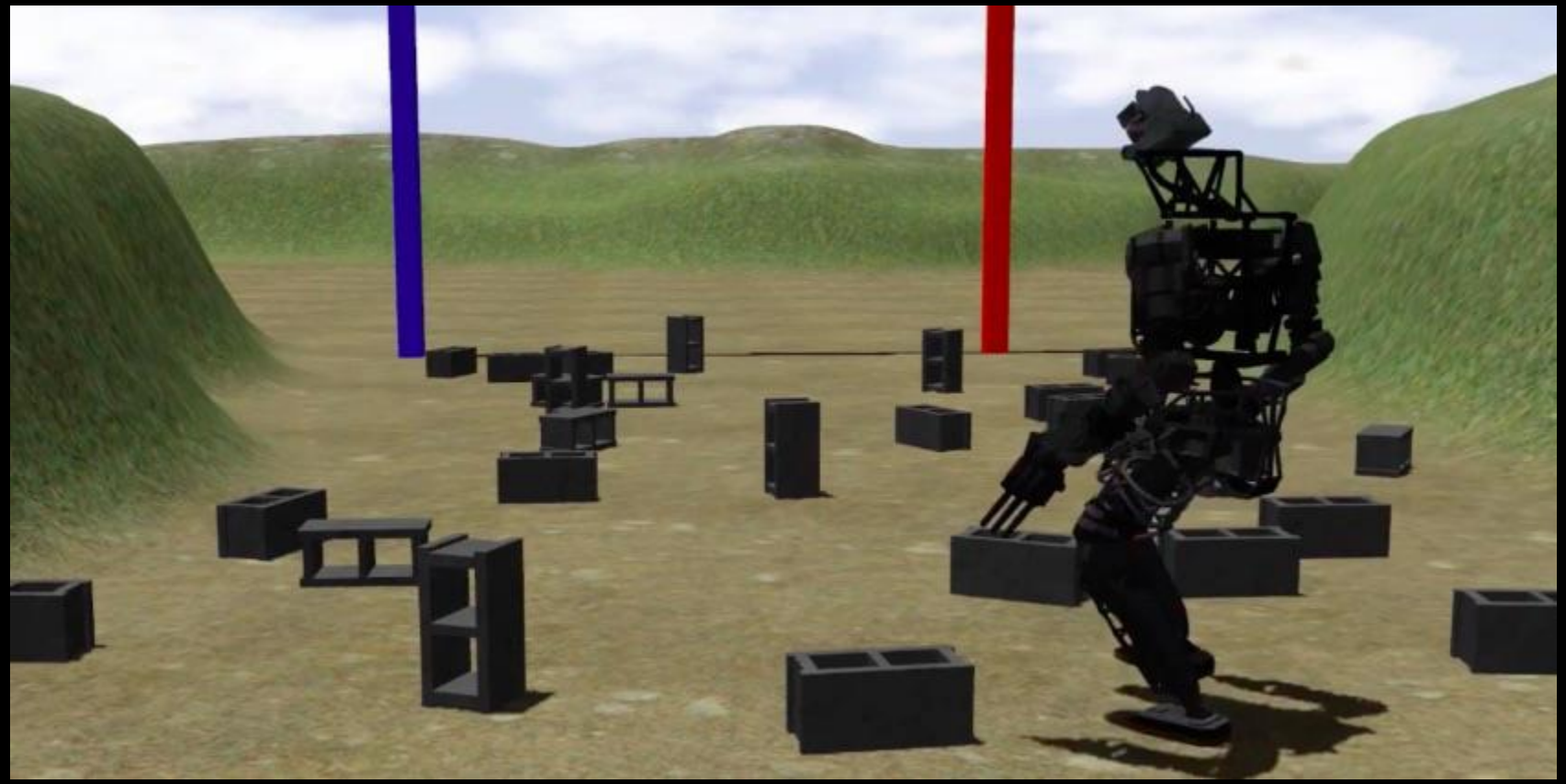
Simulation – 2015 정보과학 성취기준

- 실세계 및 다양한 학문 분야의 복잡한 문제들을 해결하기 위해서는 컴퓨팅 시스템의 계산 능력에 기반한 시뮬레이션 방법을 이해할 수 있어야 한다. 이를 위해 근사, 난수, 시각화의 개념을 이해하고 이를 활용하여 시뮬레이션 알고리즘을 설계할 수 있어야 한다.

Simulation – 2015 정보과학 교수학습 유의 사항

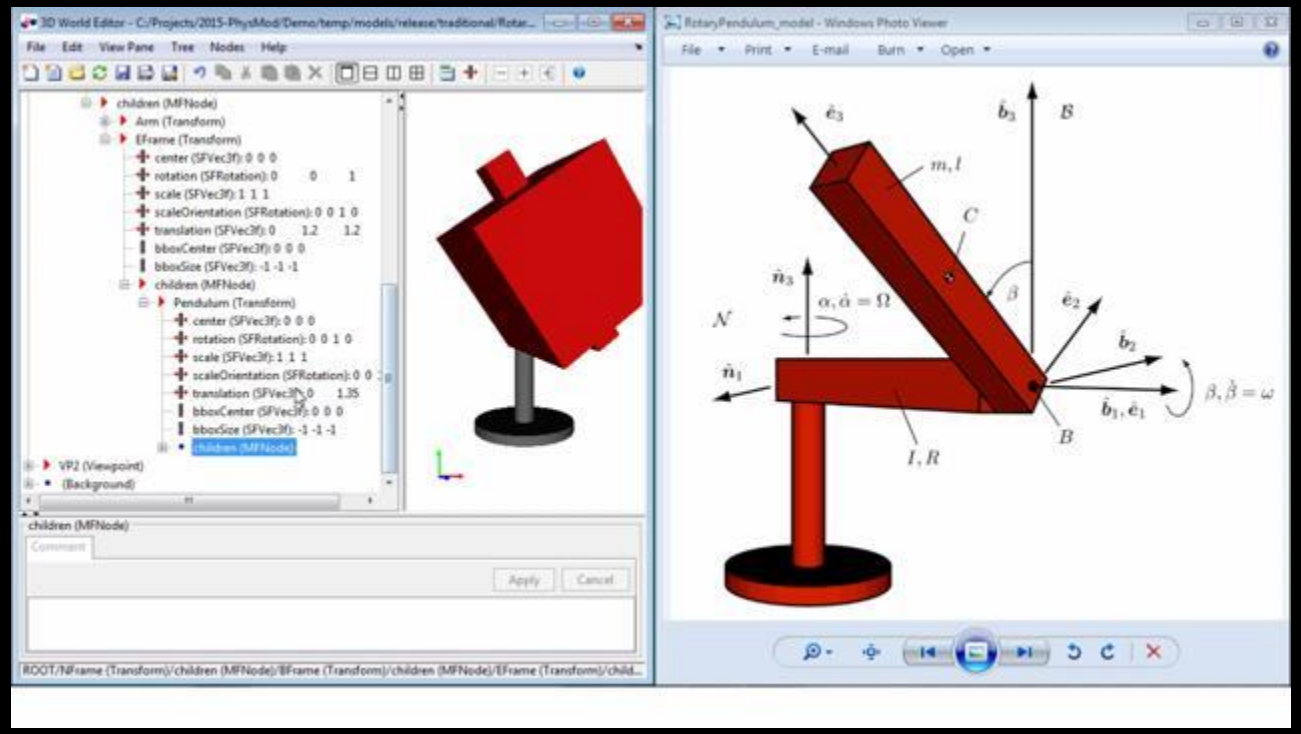
- 시뮬레이션 영역의 교수 학습을 전개할 때는 시각화의 구현이 용이한 텍스트 기반의 프로그래밍 언어를 활용한다.
- 다양한 수치해석 기법 등을 활용하여 근사, 난수 알고리즘에 대한 교수 학습을 전개한다.

Simulation Software



http://robohub.org/

Simulation Software



https://kr.mathworks.com

Processing

Vector

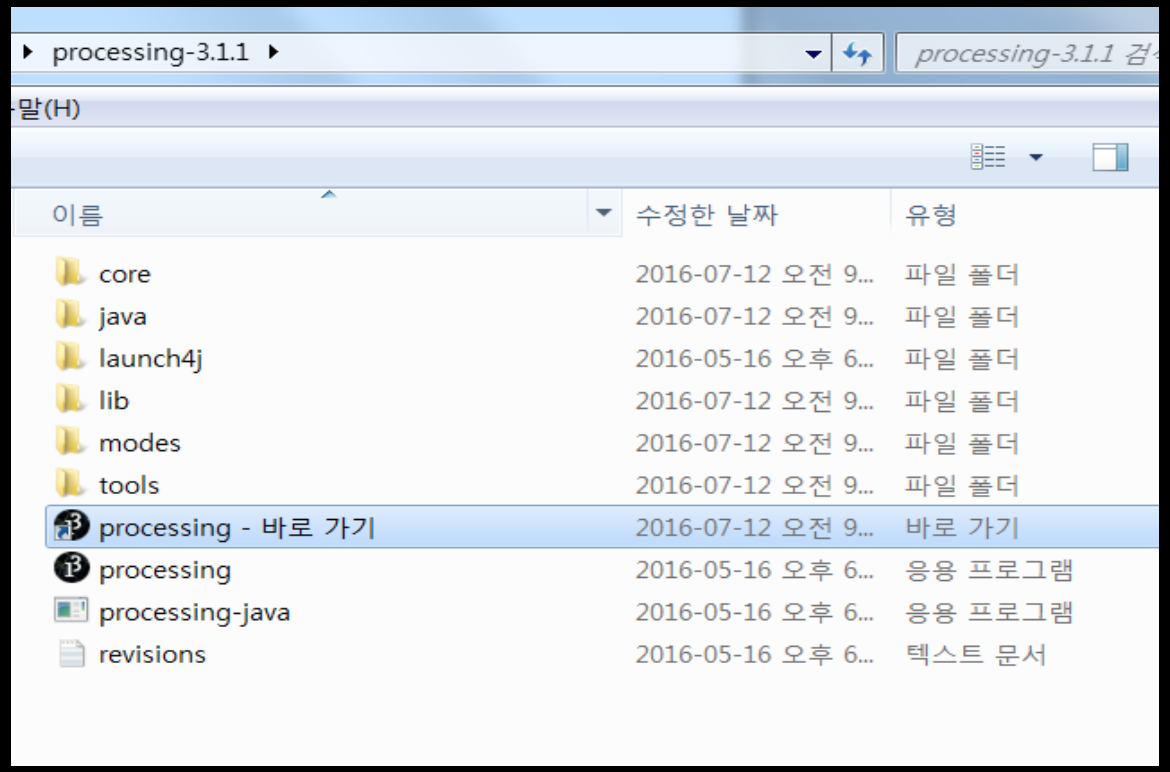
Processing Install

설치 순서 1: 다운로드

다운로드를 받고 압축을 품

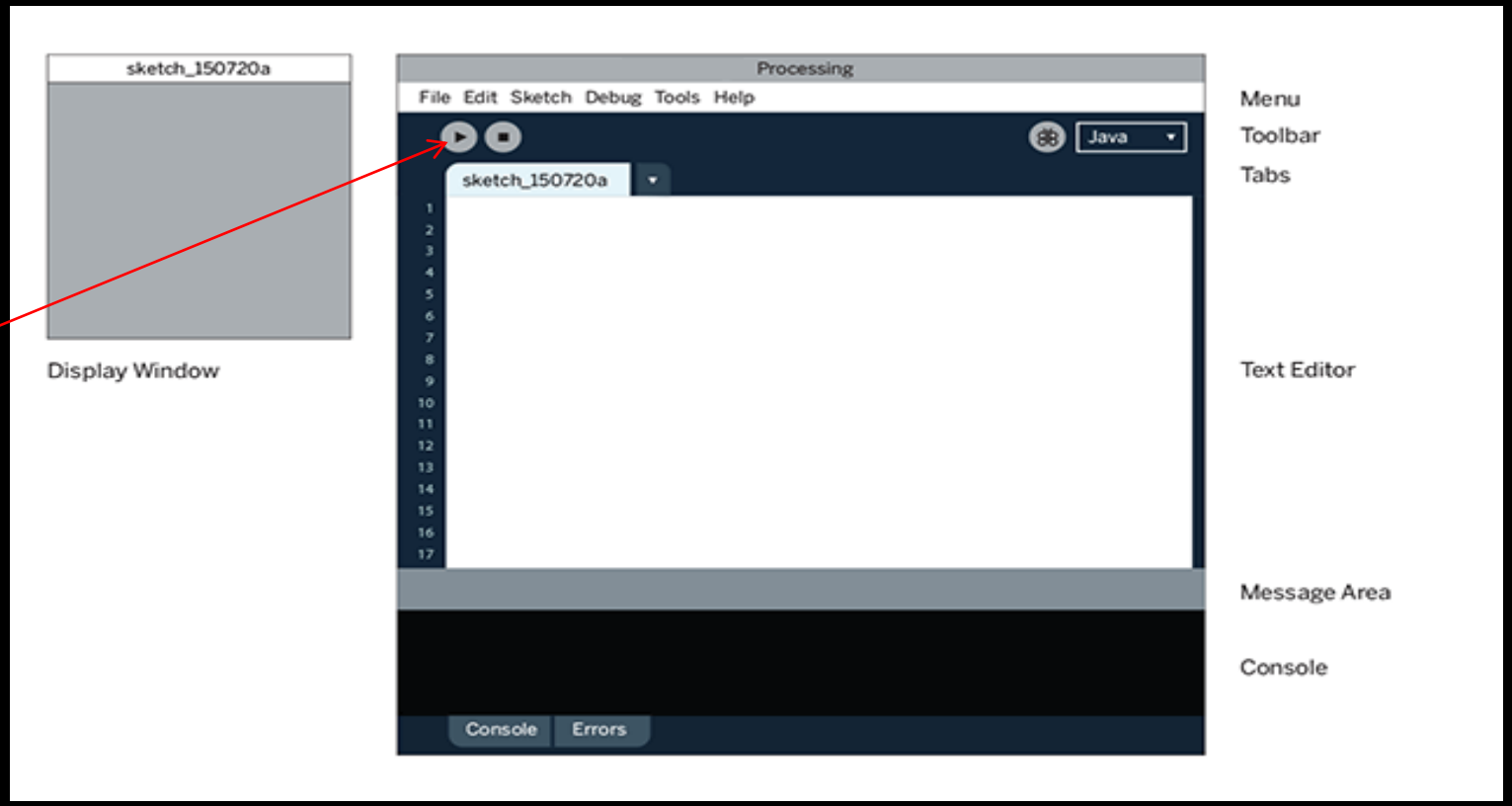
설치 순서 2: 디렉토리 복사

다운로드를 받고 압축을 풀린것을 C 디렉토리에 복사



설치 순서 3: 메뉴 구성

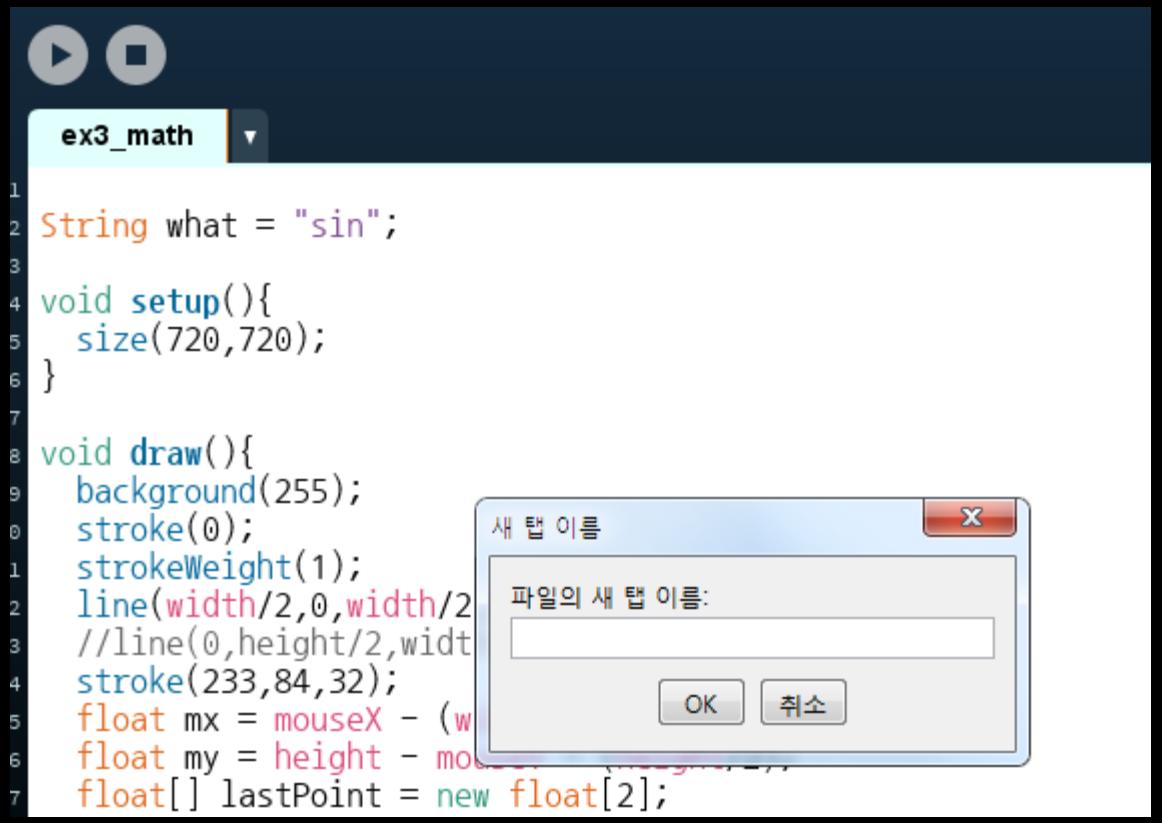
Processing을 생행하면 processing 화면이 실행됨



탭 이용하기

tab 기능

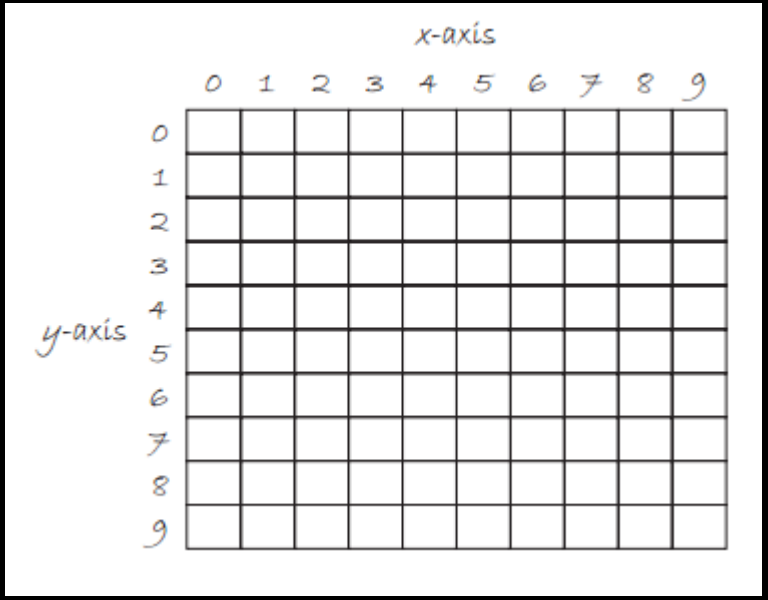
Tab은 코드를 분산 또는 클래스 작업

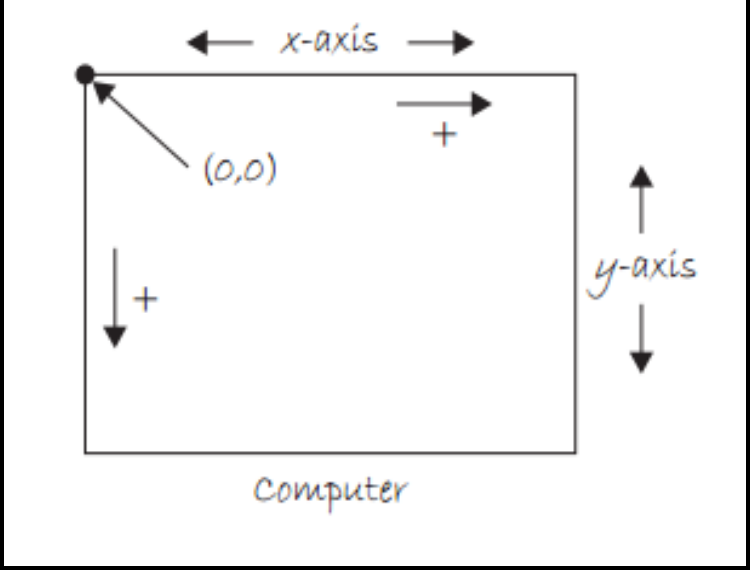


픽셀 이해하기

pixel 이해하기(2차원)

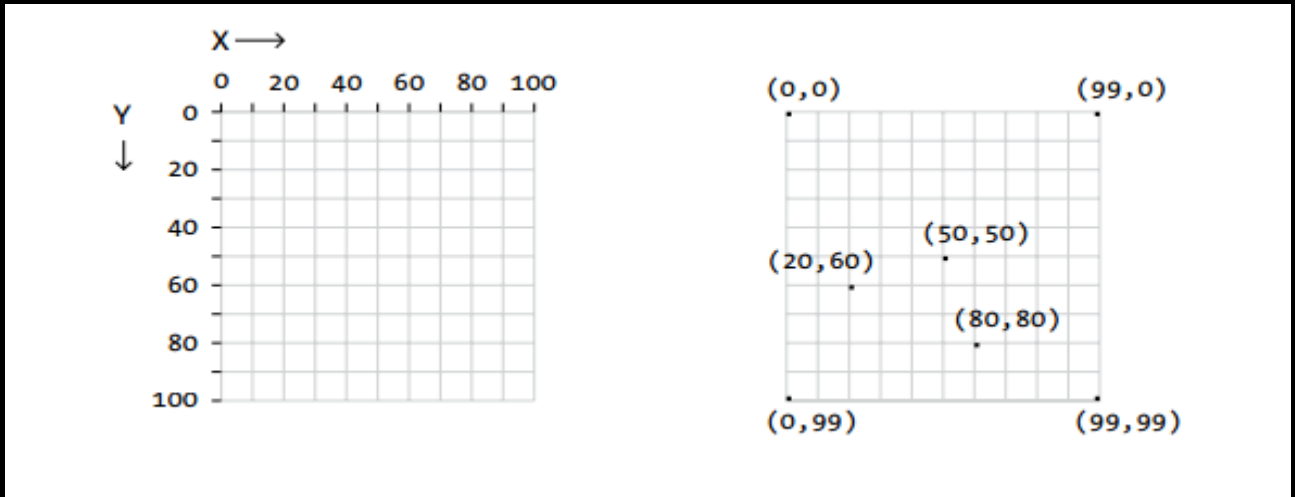
화면은 x 축과 y축을 기준으로 구성됨





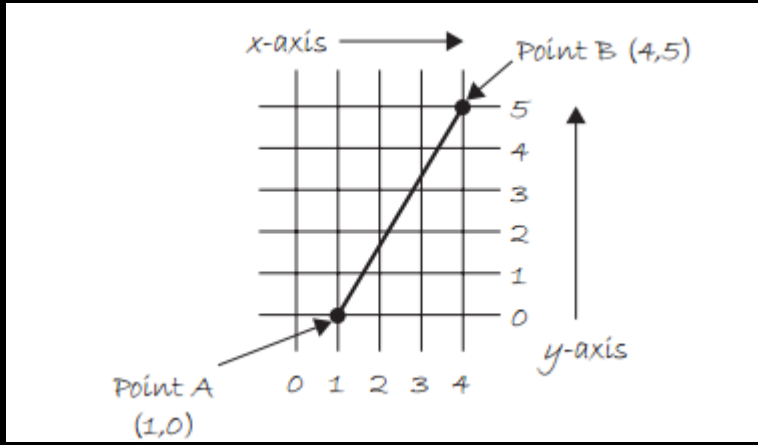
pixel 이해하기(2차원) 점 예시

화면에서 점을 확인하기



pixel 이해하기(2차원) 선 예시

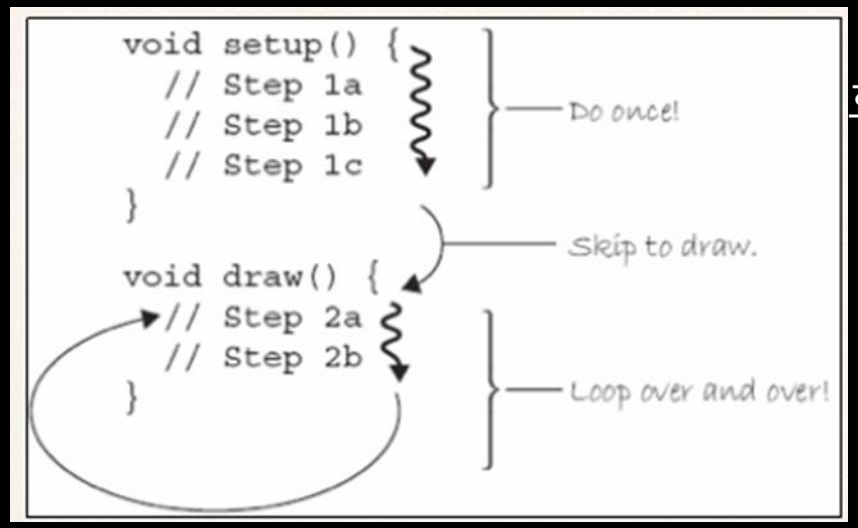
화면은 두 점 A, B를 만나는 선을 그리기



프로그램 구조 이해하기

Processing 작동원리

프로그램을 호출하면 setup()함수를 한번 호출한 후에 draw()함수를 지속적으로 호출 처리



호출

1초에 60번 호출

도형 그리기

rect 함수 : 사각형 하나 그리기

사각형 그리기(rect(x,y,width,height))

ellipse 함수 : 원 그리기

원 그리기(ellipse(x,y,width,height))

한번 실행 하기

한번만 실행하기 : noLoop

setup함수 내에 noLoop()를 실행하면 draw함수가 한번만 실행됨

Processing 특징

언어의 특징 : 기호

JVM상에서 작동하므로 Java 언어의 특징을 그대로 사용.

대부분의 문법은 C/C++과 동일 / 배열의 사용이 조금 다름.

문장 끝

복합 문장

Class

class

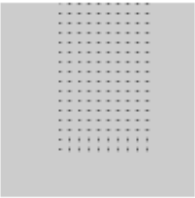
```
// Declare and construct two objects (h1, h2) from the class HLine
HLine h1 = new HLine(20, 2.0);
HLine h2 = new HLine(50, 2.5);

void setup()
{
  size(200, 200);
  frameRate(30);
}

void draw() {
  background(204);
  h1.update();
  h2.update();
}

class HLine {
  float ypos, speed;
  HLine (float y, float s) {
    ypos = y;
    speed = s;
  }
  void update() {
    ypos += speed;
    if (ypos > height) {
      ypos = 0;
    }
    line(0, ypos, width, ypos);
  }
}
```

for

```
for (int i = 0; i < 40; i = i+1) {
  line(30, i, 80, i);
}
```

```
for (int i = 0; i < 80; i = i+5) {
  line(30, i, 80, i);
}
```

```
for (int i = 40; i < 80; i = i+5) {
  line(30, i, 80, i);
}
```

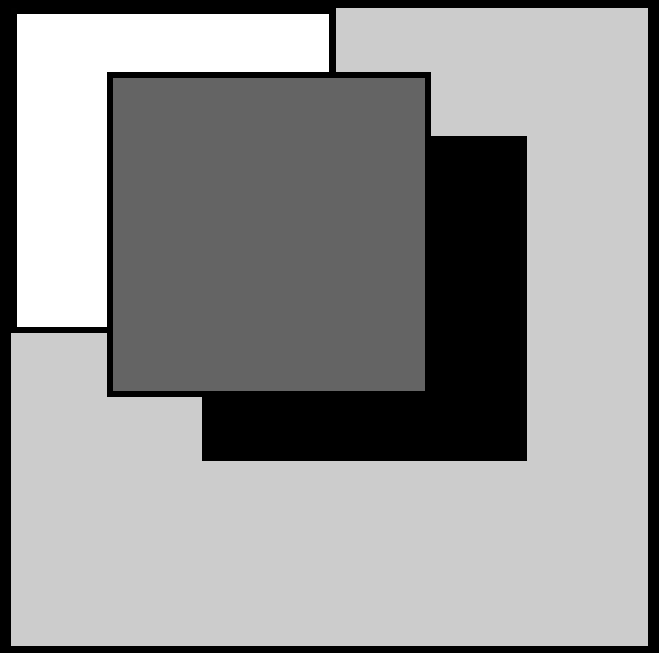


```
// Nested for() loops can be used to
// generate two-dimensional patterns
for (int i = 30; i < 80; i = i+5) {
  for (int j = 0; j < 80; j = j+5) {
    point(i, j);
  }
}
```

Array

```
int[] numbers = new int[3];
numbers[0] = 90;  // Assign value to first element in the array
numbers[1] = 150; // Assign value to second element in the array
numbers[2] = 30;  // Assign value to third element in the array
int a = numbers[0] + numbers[1]; // Sets variable 'a' to 240
int b = numbers[1] + numbers[2]; // Sets variable 'b' to 180
```

pushMatrix() ~ popMatrix()
레이어의 개념

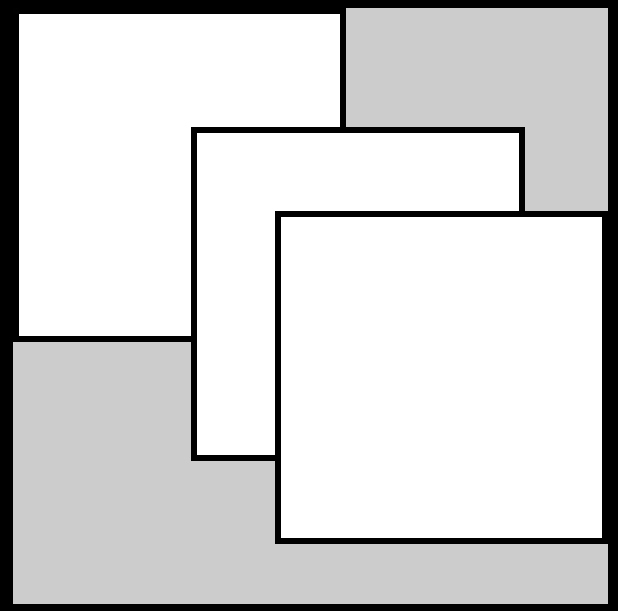


example pic

```
fill(255);
rect(0, 0, 50, 50);  // White rectangle

pushMatrix();
translate(30, 20);
fill(0);
rect(0, 0, 50, 50);  // Black rectangle
popMatrix();

fill(100);
rect(15, 10, 50, 50);  // Gray rectangle
```

translate()

화면의 원점을 지정한 곳으로 이동

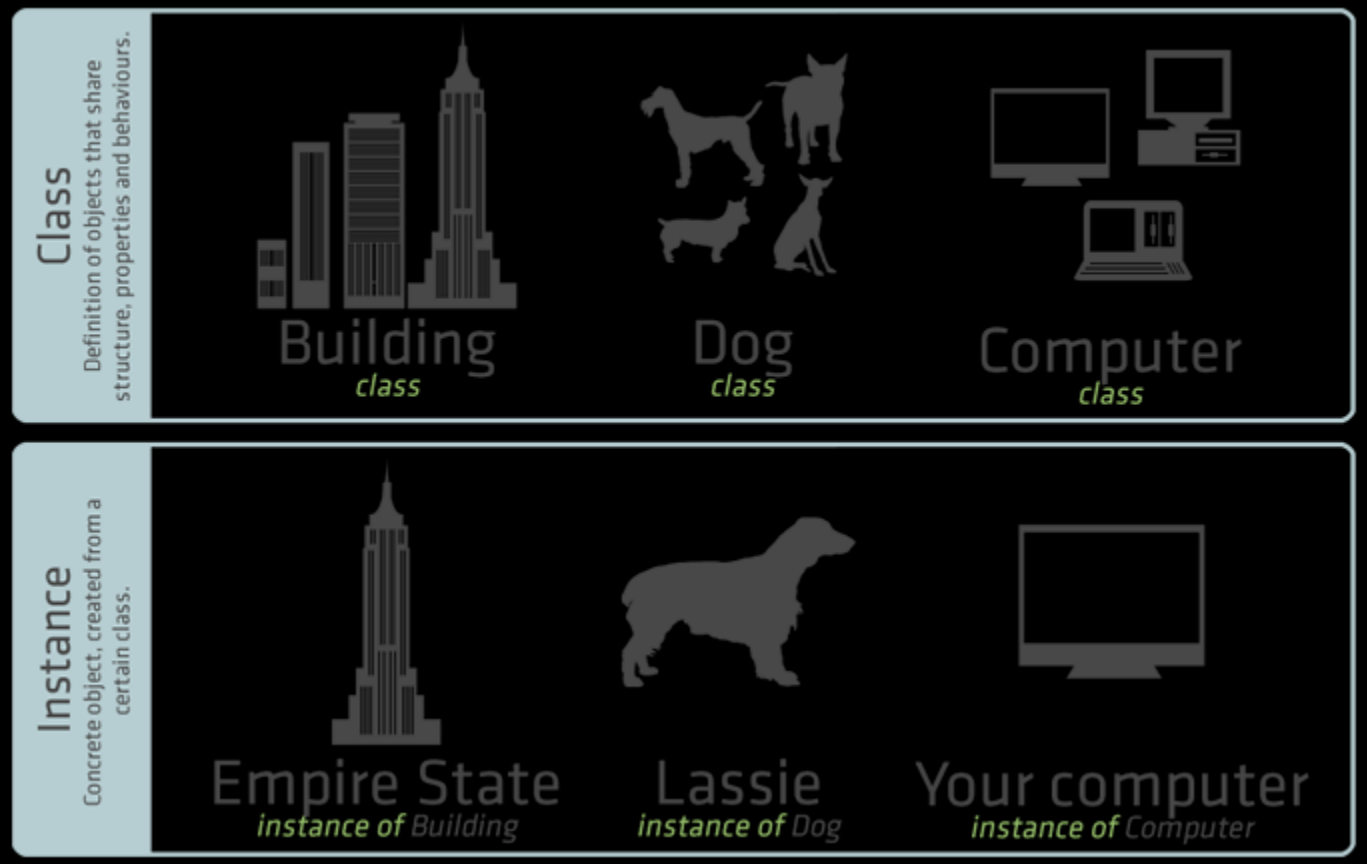


```
rect(0, 0, 55, 55);  // Draw rect at original 0,0
translate(30, 20);
rect(0, 0, 55, 55);  // Draw rect at new 0,0
translate(14, 14);
rect(0, 0, 55, 55);  // Draw rect at new 0,0
```

OOP

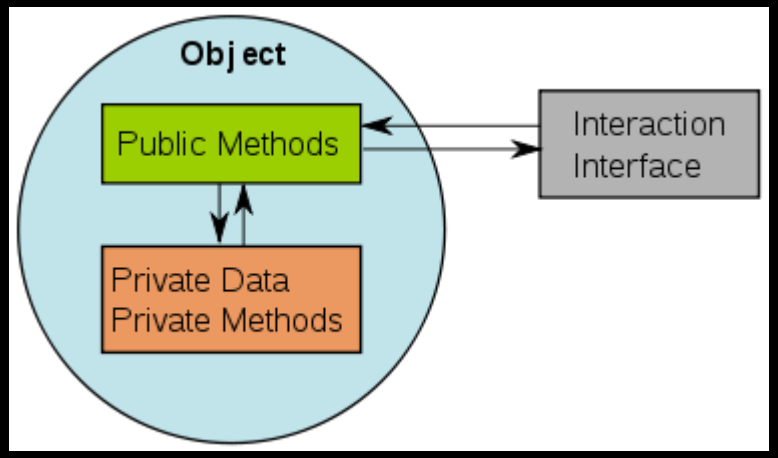
Object Oriented Programming

OOP



https://docs.sencha.com

OOP

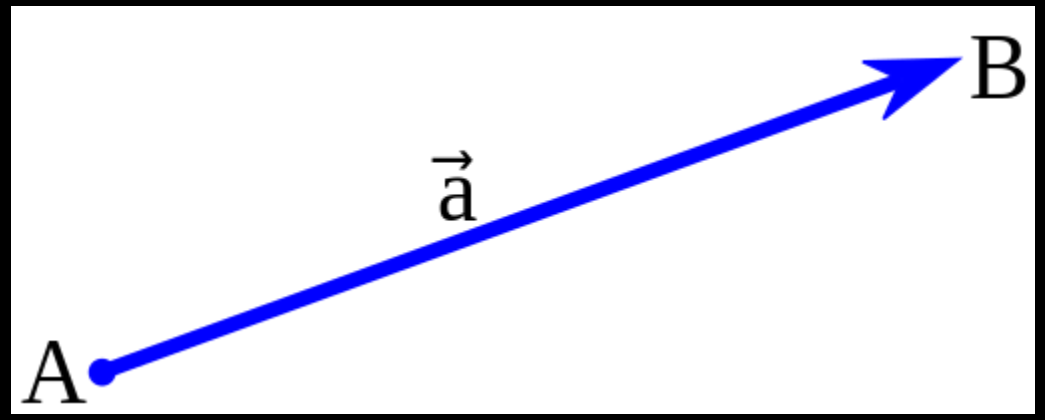


https://commons.Wikimedia.org

Vector

Vector

Vector



https://en.wikipedia.org

PVector = Processing Vector Class

- A class to describe a two or three dimensional vector, specifically a Euclidean (also known as geometric) vector.
- A vector is an entity that has both magnitude and direction. The datatype, however, stores the components of the vector (x,y for 2D, and x,y,z for 3D).
- The magnitude and direction can be accessed via the methods mag() and heading().

Example – no PVector

Example – no PVector

```
float x = 100;
float y = 100;
float xspeed = 2.5;
float yspeed = 2;
void setup() {
size(800, 200);
smooth();
}
void draw() {
background(255);
x = x + xspeed;
y = y + yspeed;
```

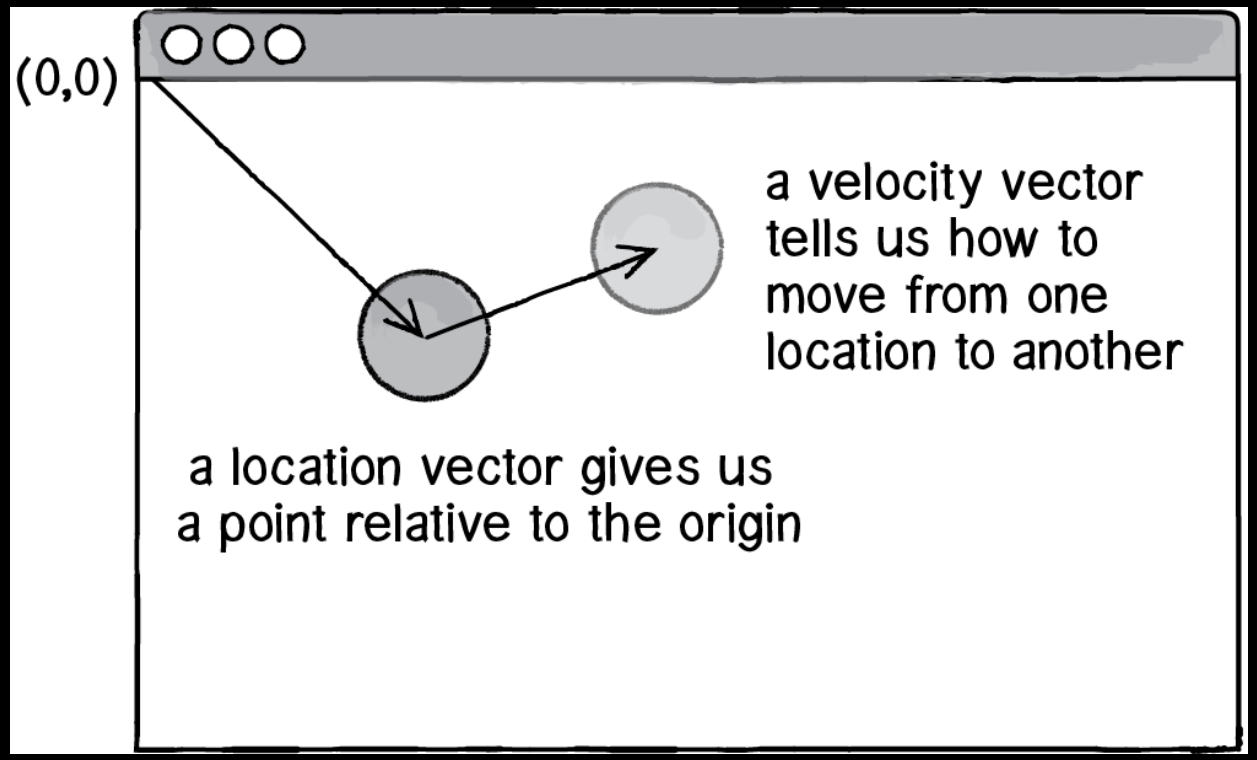
Example – no PVector

```
if ((x > width) || (x < 0)) {
xspeed = xspeed * -1;
}
if ((y > height) || (y < 0)) {
yspeed = yspeed * -1;
}
stroke(0);
strokeWeight(2);
fill(127);
ellipse(x, y, 48, 48);
}
```

Example – no PVector

- https://github.com/suakii/2017SWProfessional/blob/master/Chapter1_OOP/C1_bouncingball_novectors/C1_bouncingball_novectors.pde

Example: Using PVector



http://natureofcode.com

Example: Using PVector

```
PVector position;
PVector velocity;
void setup() {
size(800,200);
smooth();
position = new PVector(100,100);
velocity = new PVector(2.5,2);
}
void draw() {
background(255);
position.add(velocity);
```

Example: Using PVector

```
if ((position.x > width) || (position.x < 0)) {
velocity.x = velocity.x * -1;
}
if ((position.y > height) || (position.y < 0)) {
velocity.y = velocity.y * -1;
}
stroke(0);
strokeWeight(2);
fill(127);
ellipse(position.x,position.y,48,48);
}
```

Example: Using PVector

- https://github.com/suakii/2017SWProfessional/blob/master/Chapter1_OOP/C2_bouncingball_vectors/C2_bouncingball_vectors.pde

OOP vs Non OOP

Non OOP Car

Non OOP Car

```
color c;
float xpos, ypos, xspeed;

void setup() {
  size(640,360);

  c = color(175);
  xpos = width/2;
  ypos = height/2;
  xspeed = 1;

  rectMode(CENTER);
  stroke(0);
```

Non OOP Car

```
  fill(c);

}

void draw() {
  background(255);

  rect(xpos, ypos, 20, 10);

  xpos = xpos + xspeed;

  if (xpos > width)
    xpos = 0;

}
```

OOP Car

```
class Car {
  // Variables.
  color c;
  float xpos;
  float ypos;
  float xspeed;
  // A constructor.
  Car() {
    c = color(175);
    xpos = width/2;
    ypos = height/2;
    xspeed = 1;
  }
  // Function.
```

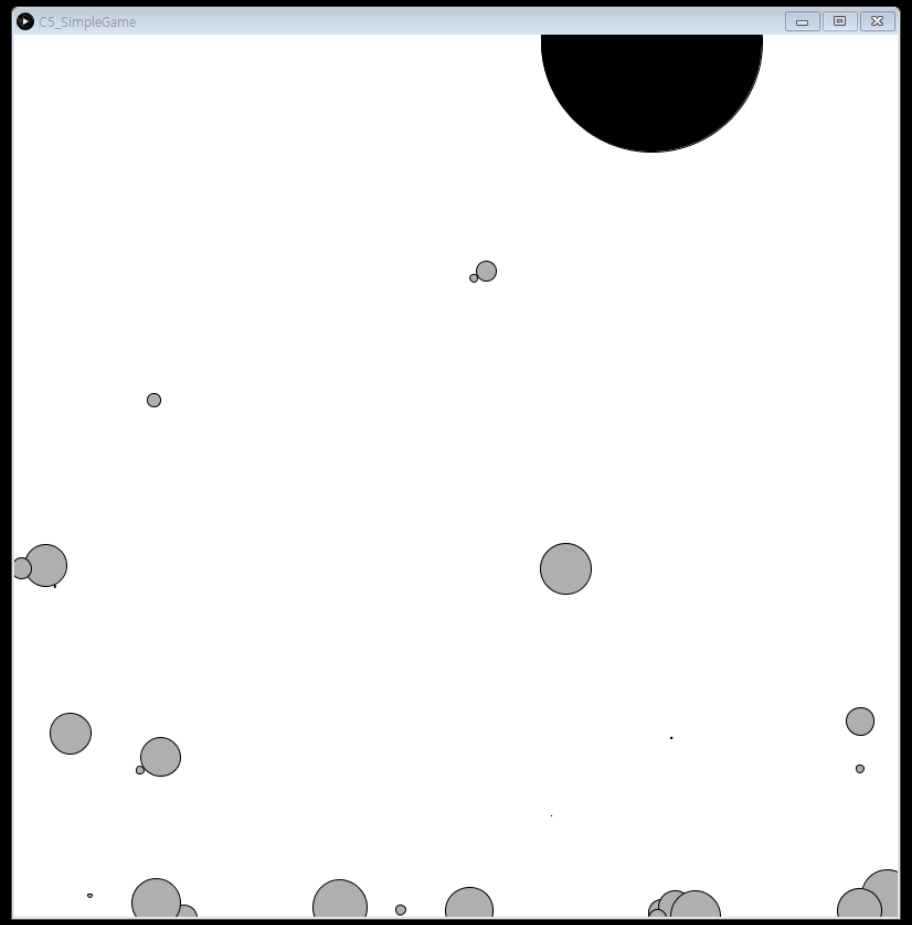
OOP Car

```
  void display() {
    // The car is just a square
    rectMode(CENTER);
    stroke(0);
    fill(c);
    rect(xpos, ypos, 20, 10);
  }
  // Function.
  void move() {
    xpos = xpos + xspeed;
    if (xpos > width) {
      xpos = 0;
    }
  }
}
```

OOP Car

```
Car myCar; // Declare car object as a globle variable.
void setup() {
size(480, 270);
// Initialize car object in setup() by calling constructor.
myCar = new Car();
}void draw() {
background(255);
// Operate Car object in draw() by calling
// object methods using the dot syntax.
myCar.move();
myCar.display();
}
```

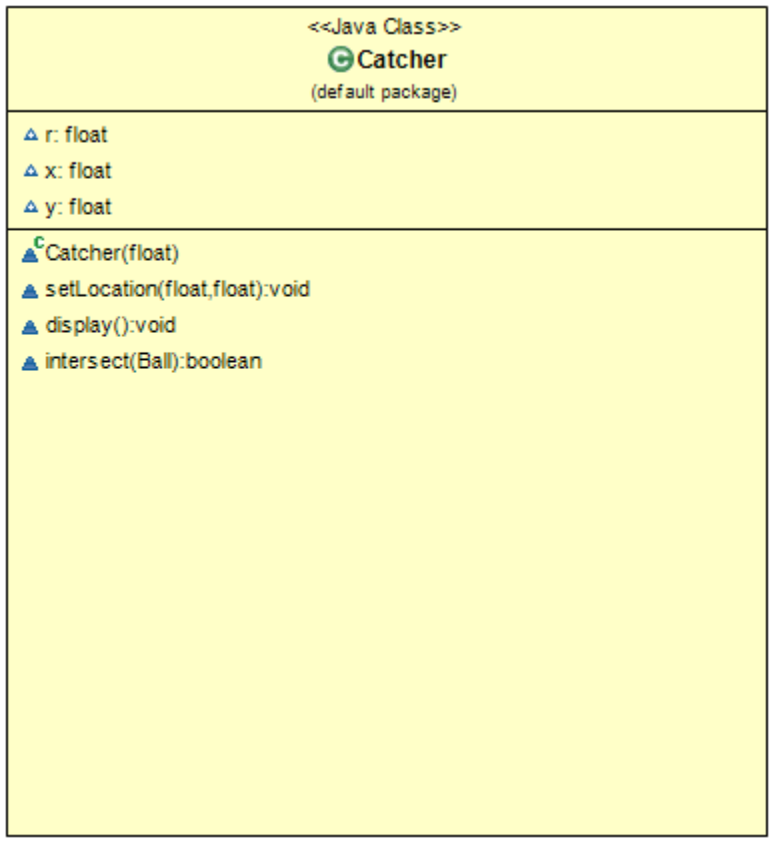
Simple OOP Game



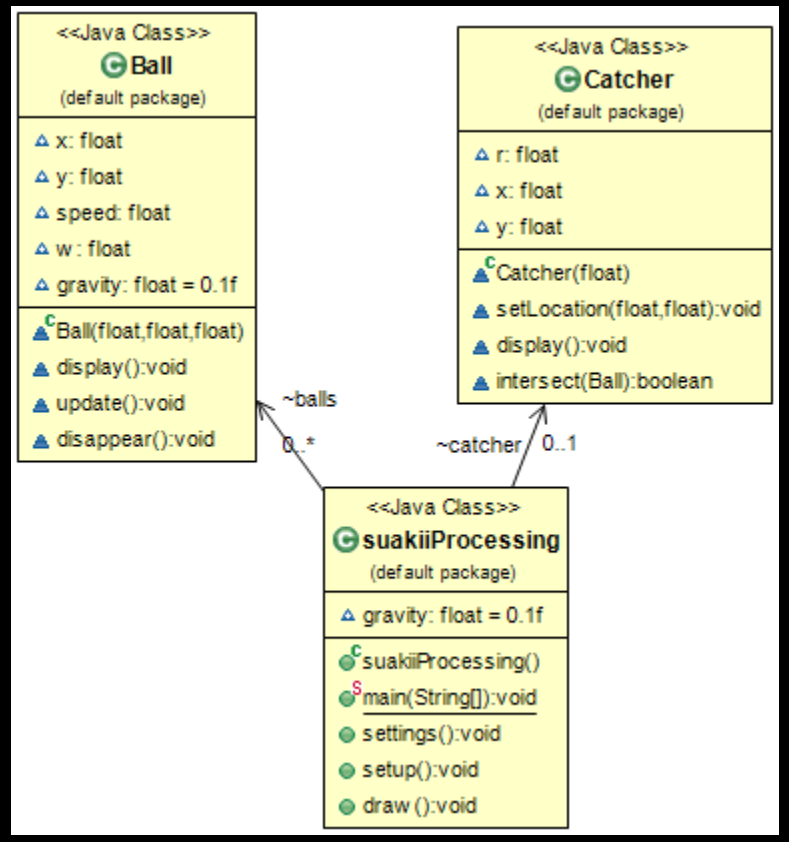
Simple OOP Game

- Design Ball Class

Simple OOP Game – Catcher Class



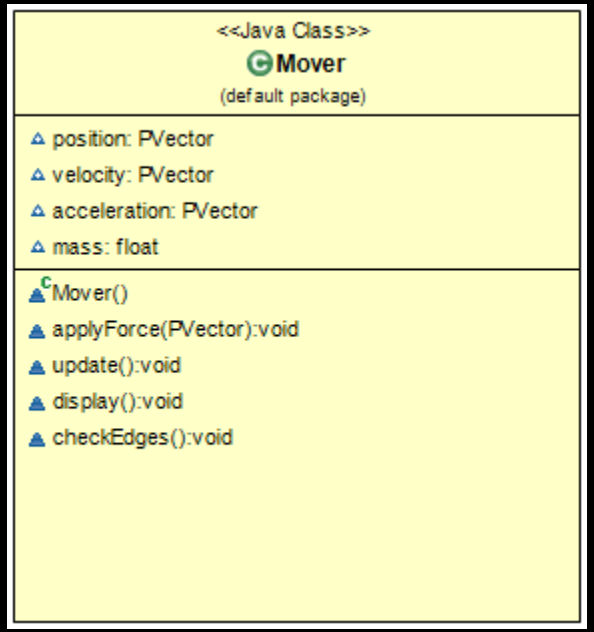
Simple OOP Game : All



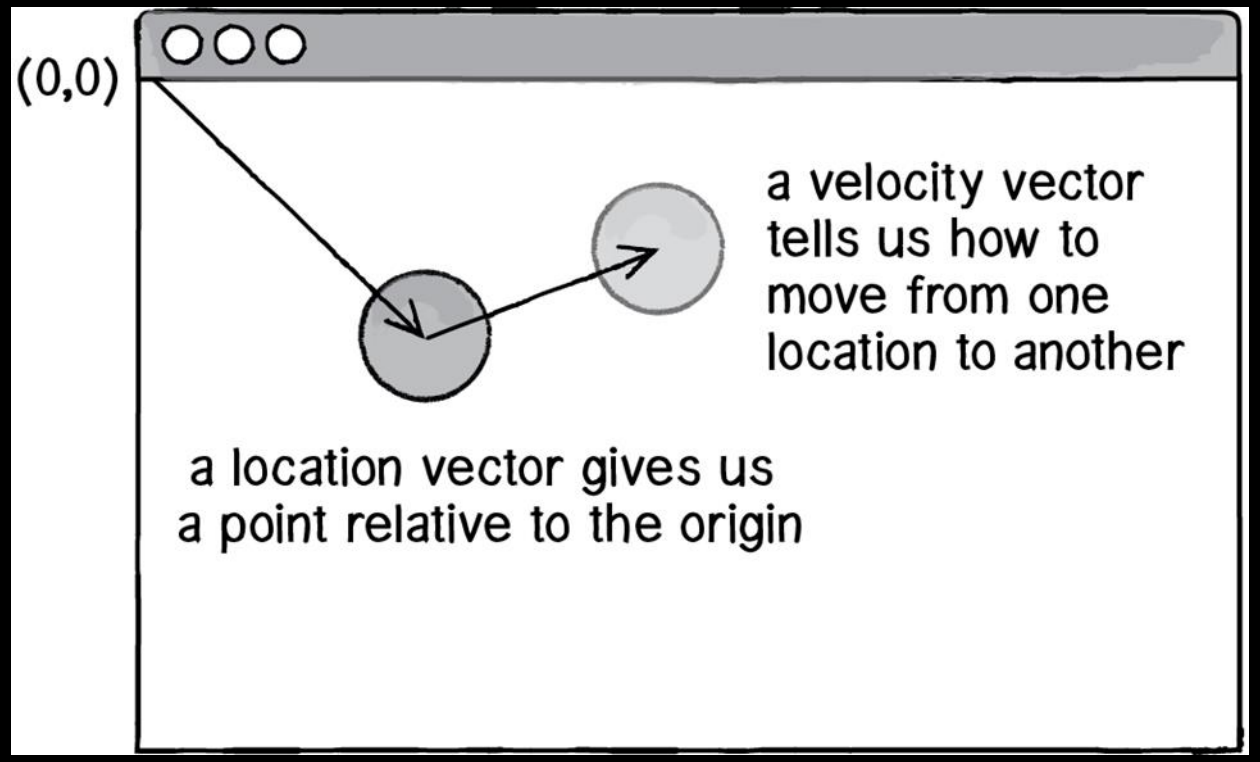
Simple OOP Game

- https://github.com/suakii/2017SWProfessional/tree/master/Chapter1_OOP/C5_SimpleGame

Forces with OOP and Pvector - Mover



Forces with OOP and Pvector - Mover



Forces with OOP and Pvector - Mover

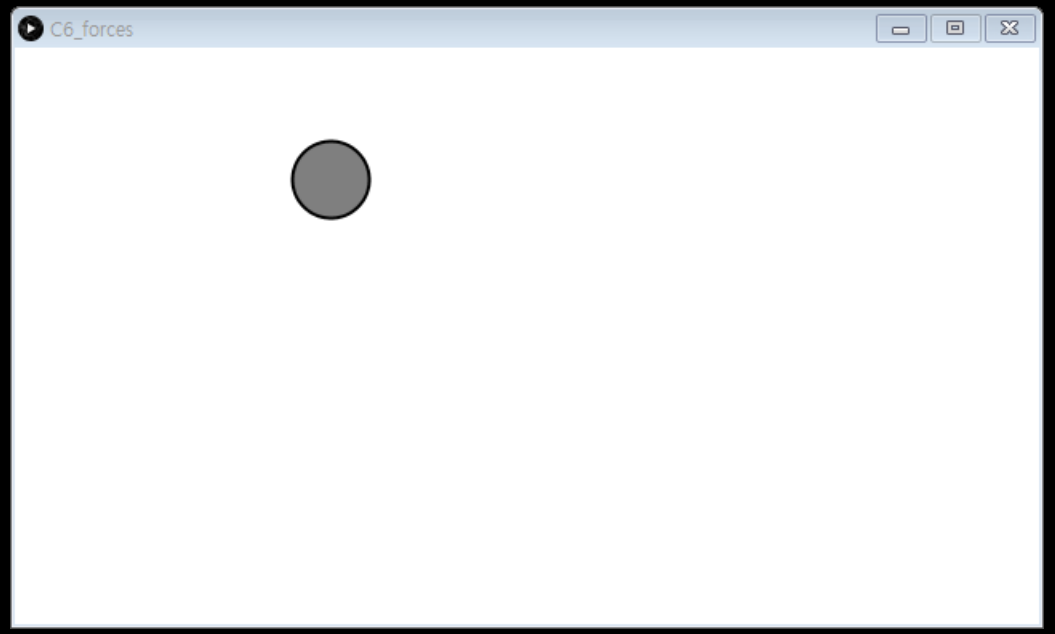
```
Mover m;

void setup() {
  size(640,360);
  m = new Mover();
}

void draw() {
  background(255);
```

Forces with OOP and Pvector - Mover

```
  PVector wind = new PVector(0.01,0);
  PVector gravity = new PVector(0,0.1);
  m.applyForce(wind);
  m.applyForce(gravity);


  m.update();
  m.display();
  m.checkEdges();

}
```

Forces with OOP and PVector - Mover

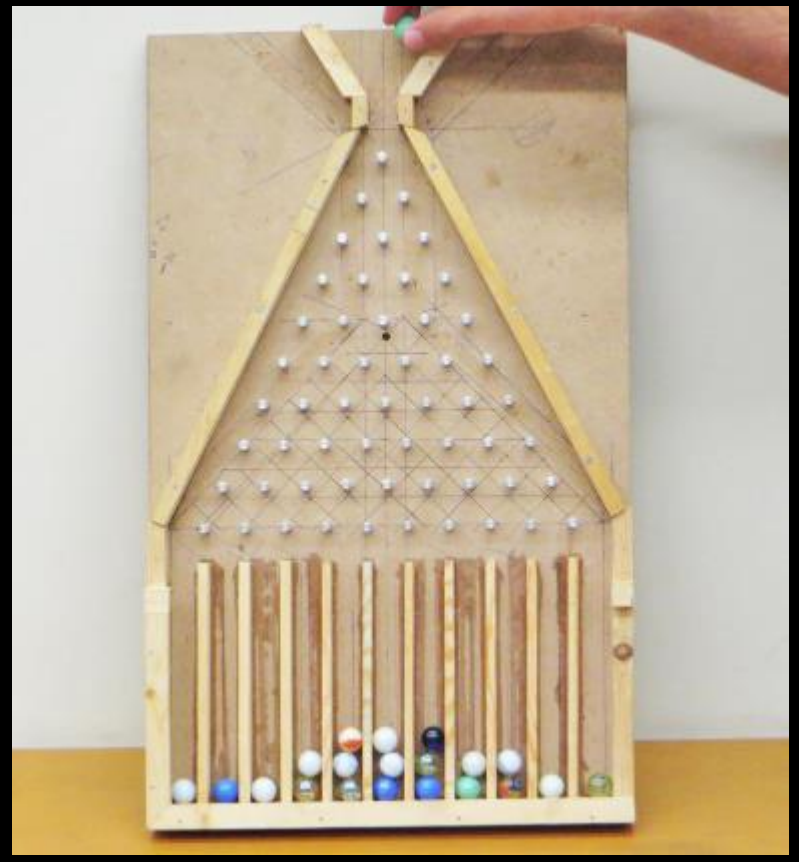
- https://github.com/suakii/2017SWProfessional/tree/master/Chapter1_OOP/C6_forces



Real Simulation

Math

Math



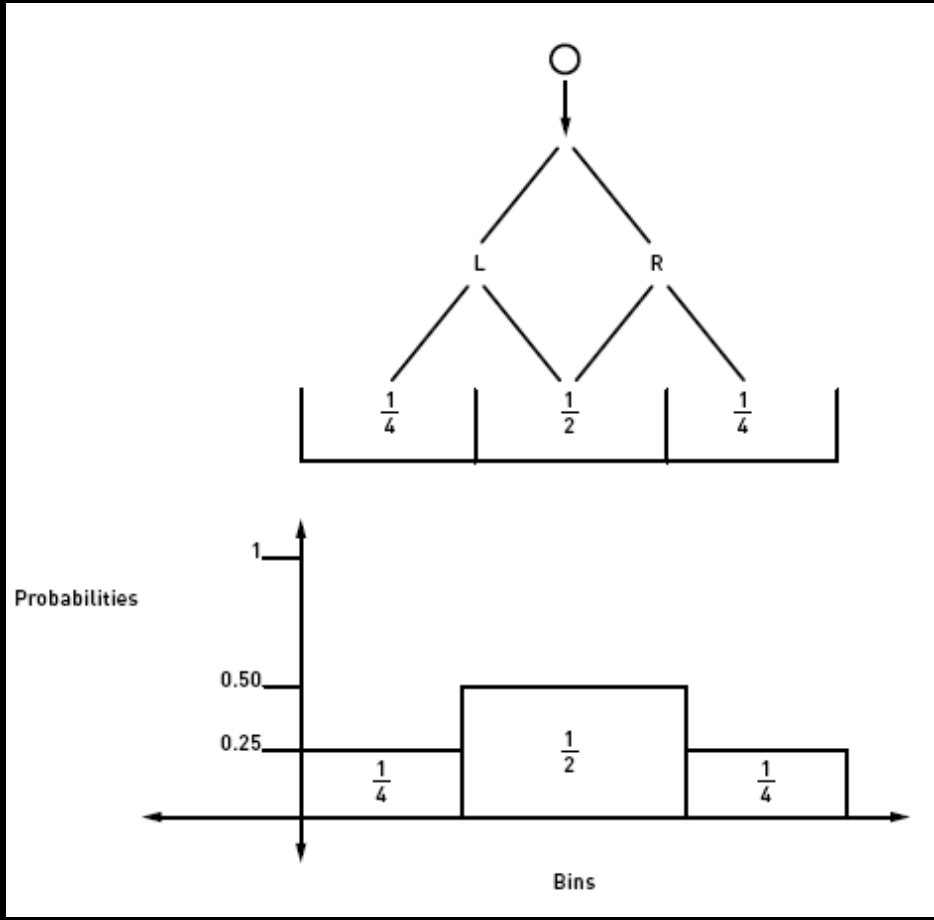
http://physlab.org

Binomial Distribution

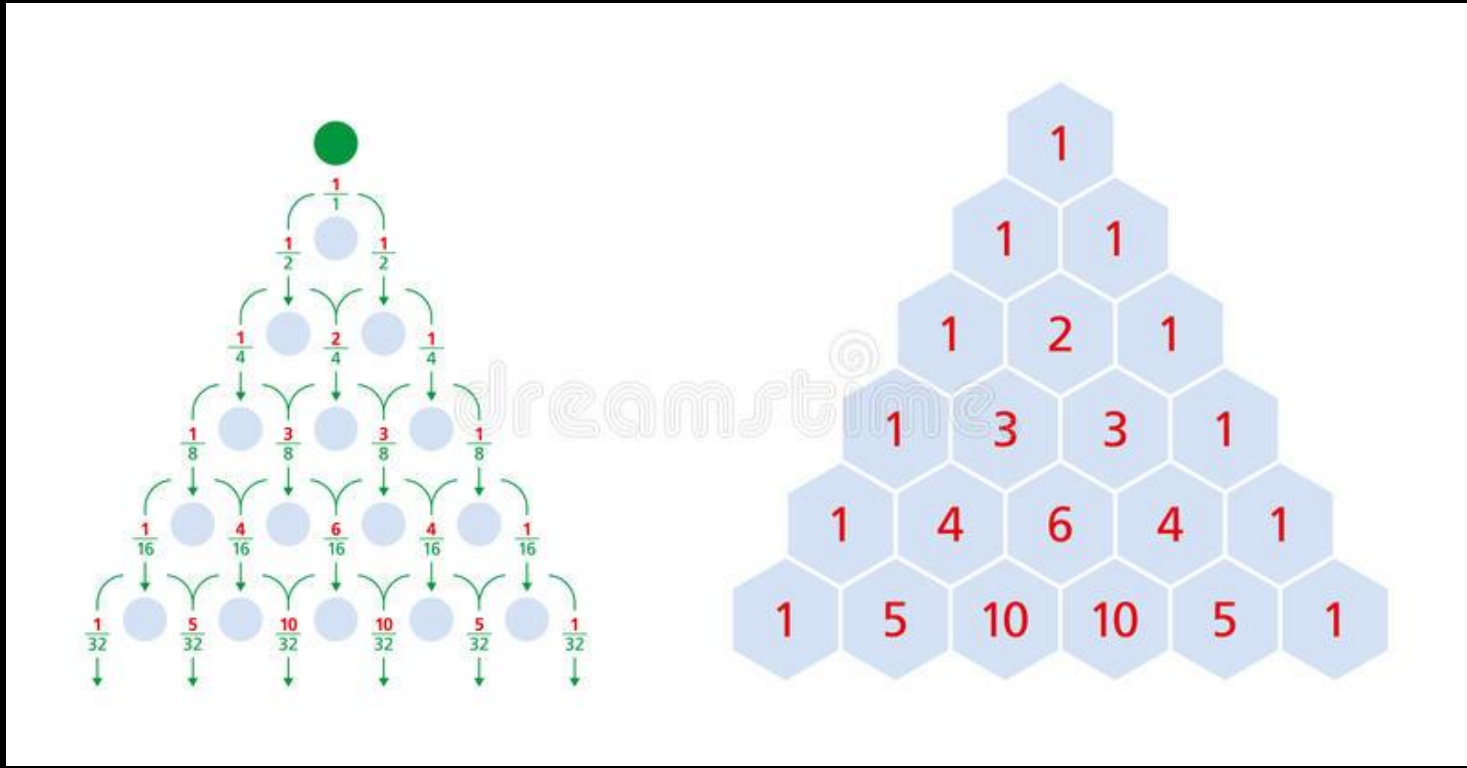
이항 분포는 1회 시행에서 사건 A 가 일어날 확률을 p, 여사건의 확률을 q 라고 하고 n 회의 독립 시행에서 A가 일어날 횟수를 X 라 할 때 다음과 같이 나타나는 것을 뜻한다.

$$P(X=r) = {}_nC_r p^r q^{n-r} \ (q=1-p, r=0, 1, 2, \cdots, n)$$

Binomial Distribution

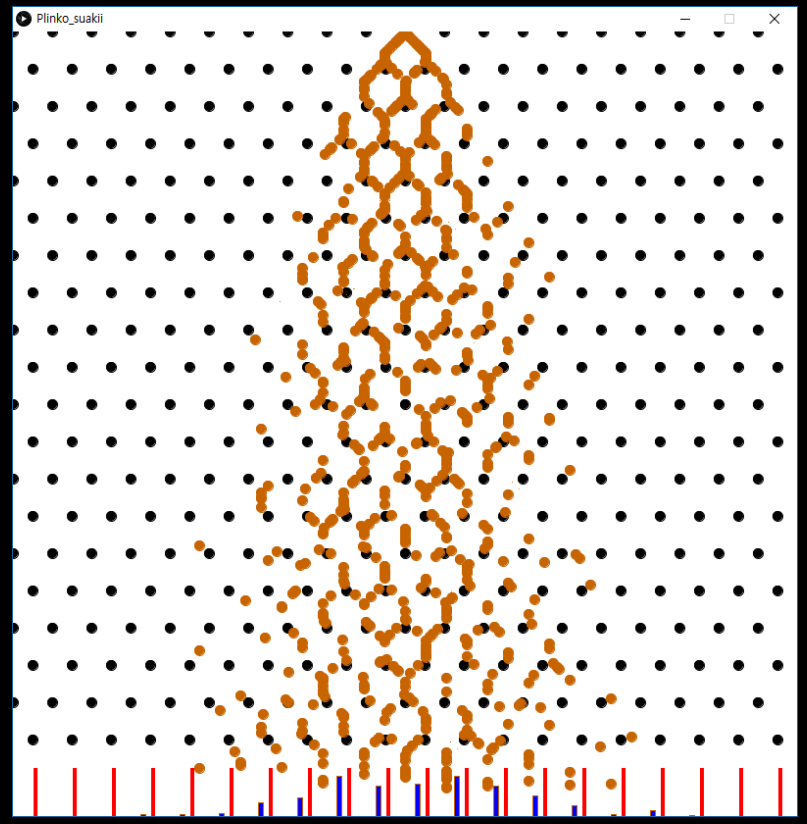


Binomial Distribution

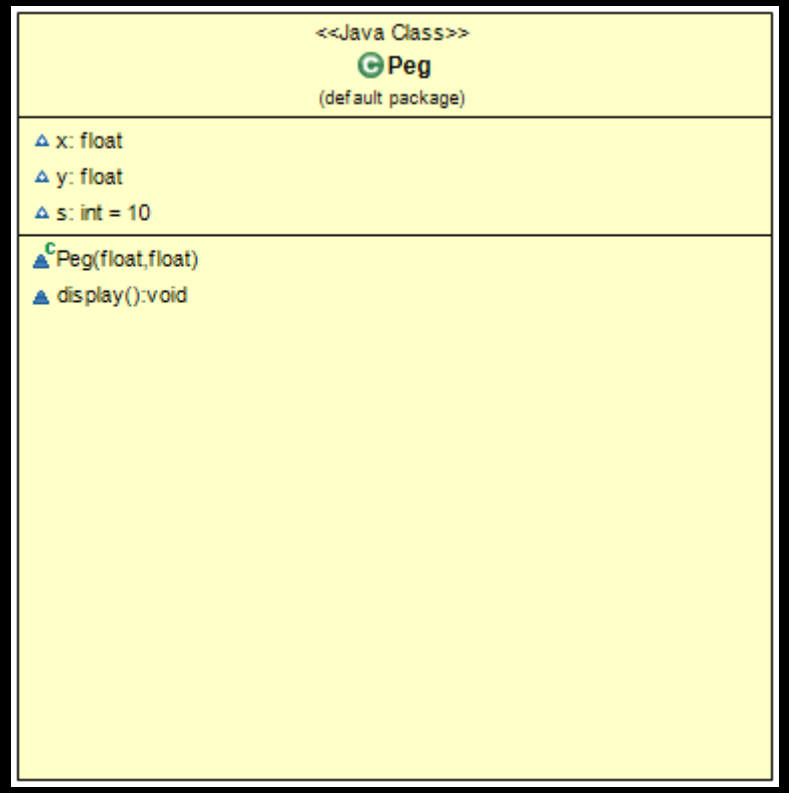


Simulation Binomial Distribution

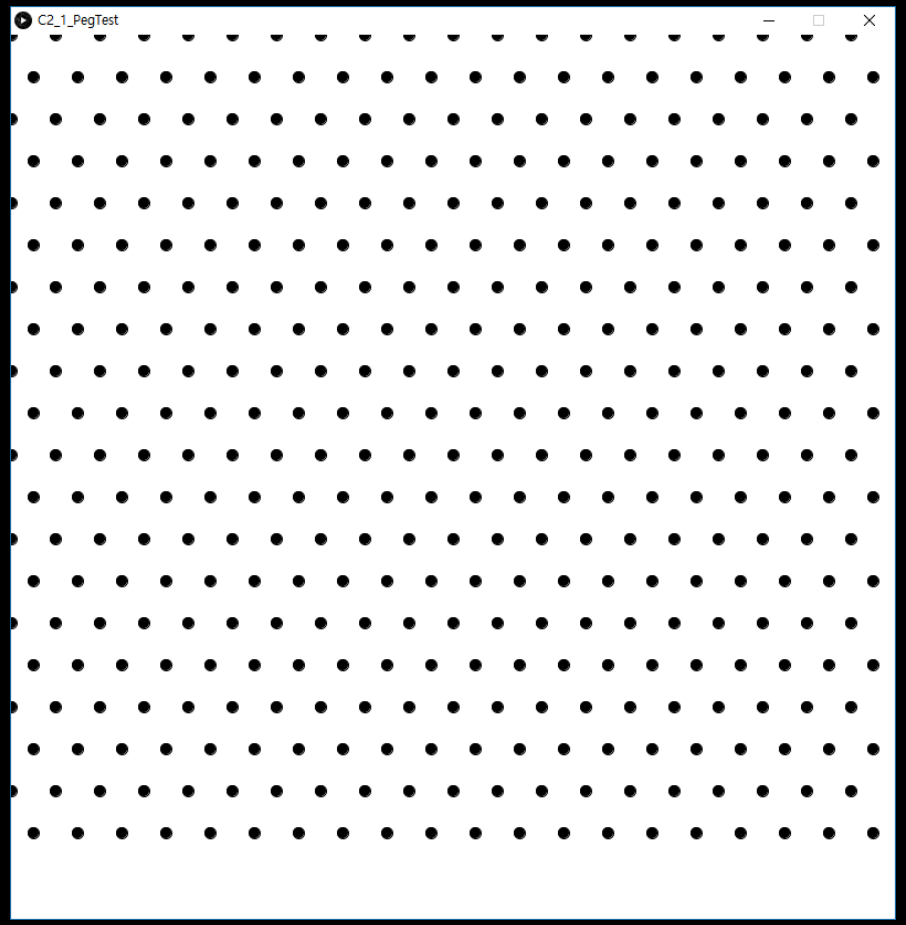
- We want to…



Math: Peg class



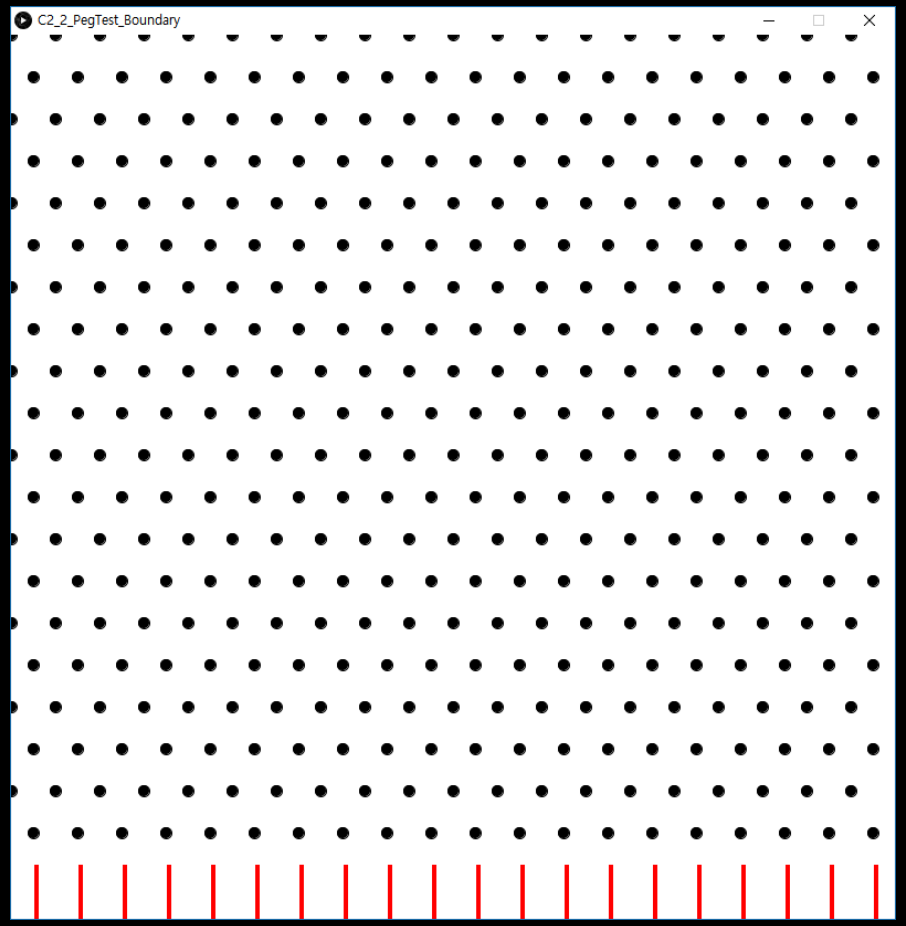
Peg Test



Peg Test

- https://github.com/suakii/2017SWProfessional/tree/master/Chapter2_Math/C2_1_PegTest

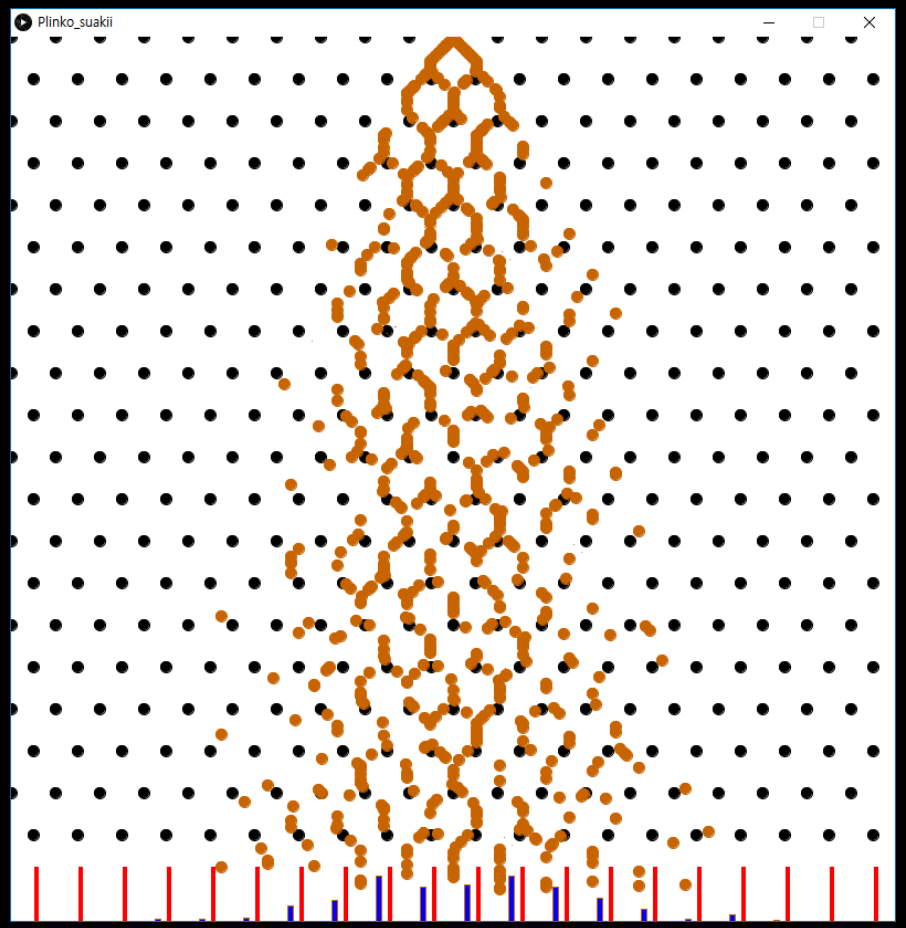
Peg with bound



Peg with Bound

- https://github.com/suakii/2017SWProfessional/tree/master/Chapter2_Math/C2_2_PegTest_Boundary

All Together



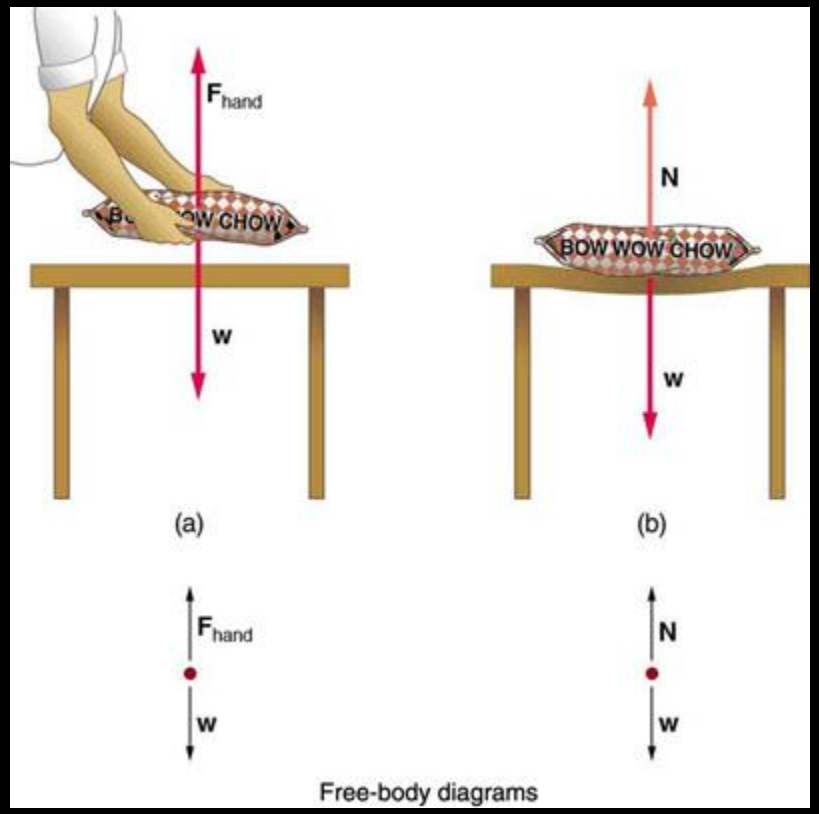
All Together

- https://github.com/suakii/2017SWProfessional/tree/master/Chapter2_Math/C2_3_PegTest_Boundary_Ball

Real Simulation

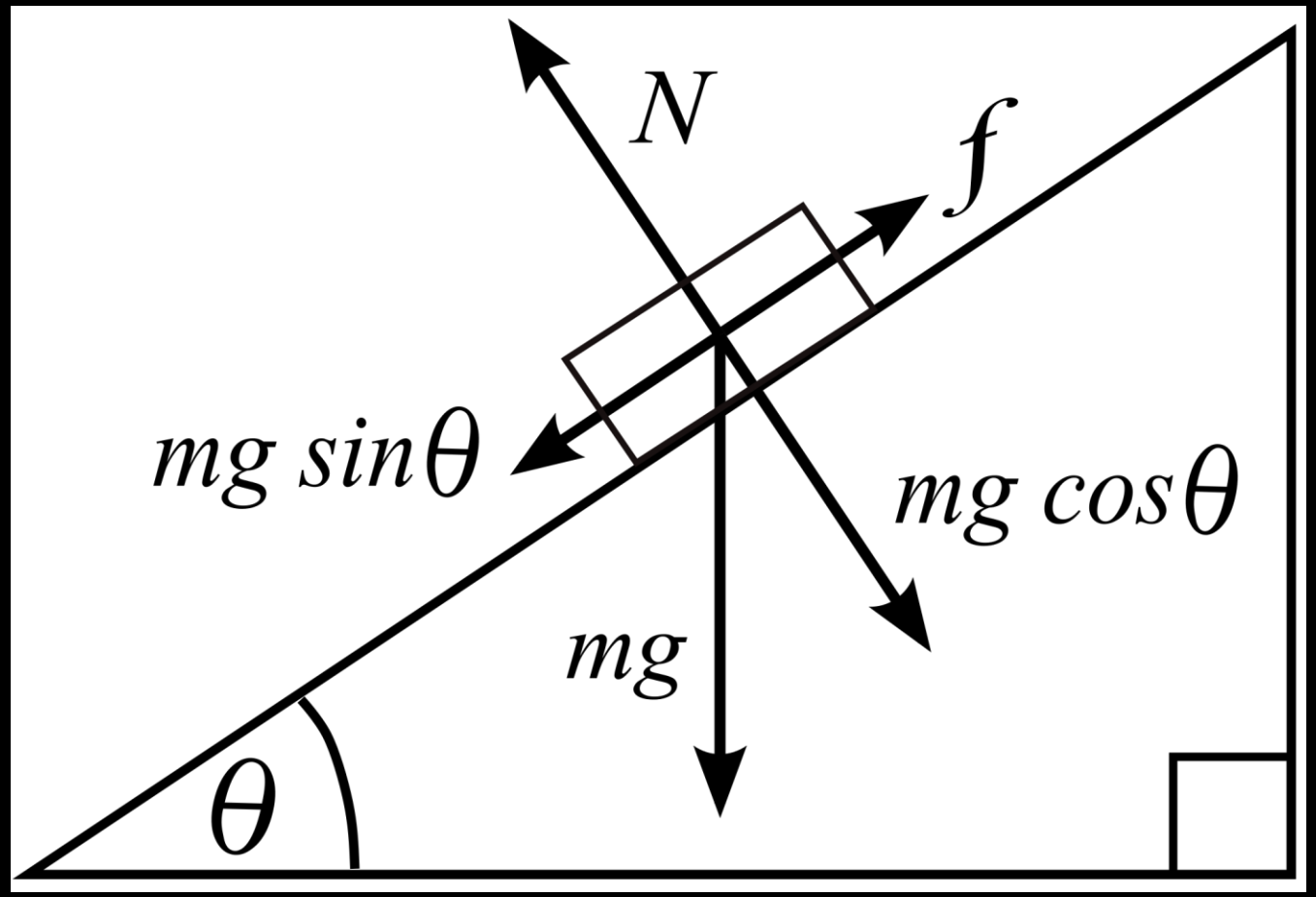
Physics

Physics – Normal Force



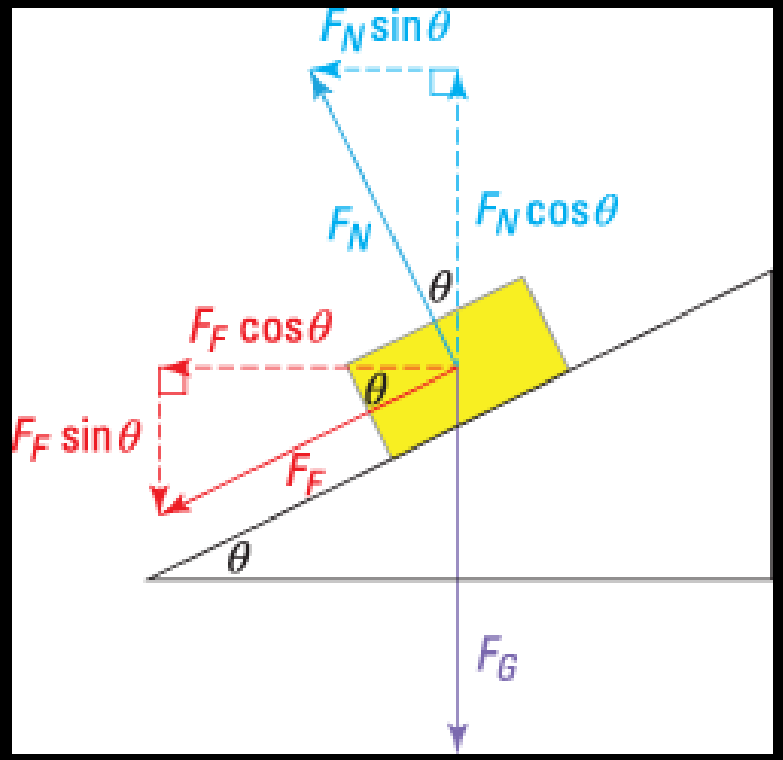
http://www.khanacademy.org

Physics – Normal Force



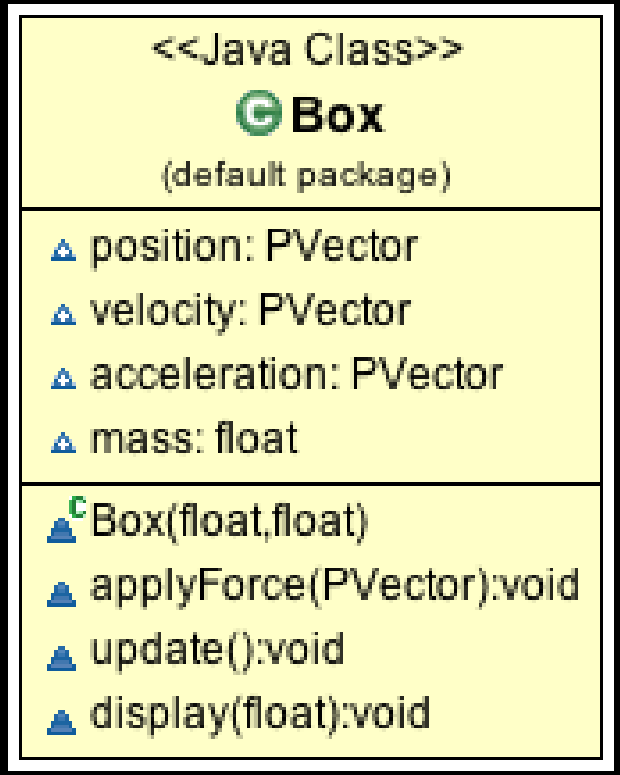
http://www.quora.com

Physics – Normal Force

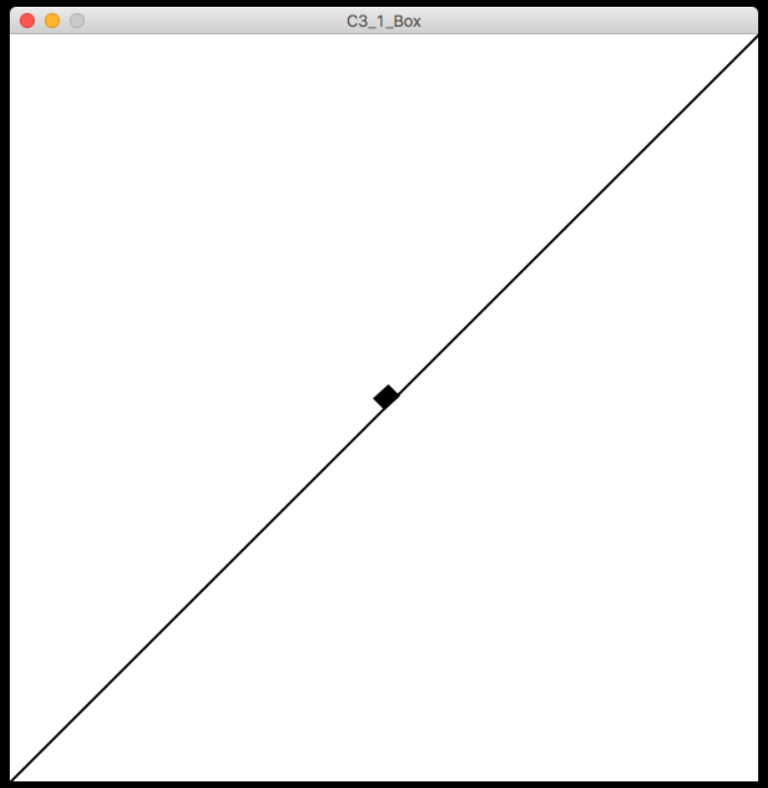


http://www.dummies.com/education/science/physics/normal-force-in-physics-problems/

Physics : Box


<<Java Class>>
Box
(default package)
position: PVector
velocity: PVector
acceleration: PVector
mass: float
Box(float,float)
applyForce(PVector):void
update():void
display(float):void

Physics: Box



Physics: Box

- https://github.com/suakii/2017SWProfessional/tree/master/Chapter3_Physics/C3_1_Box

Physics: Box

```
Box box;
float s = PI / 4;
float mu = 0.5;

void setup() {
  size(600,600);
  box = new Box(width, 0);
  noLoop();
}

void draw() {
```

Physics: Box

```
background(255);

stroke(0);
strokeWeight(2);
line(0,height,width,0);

fill(0);
//text("theta=45, normal=mgcos(theta)", width-200, height-20);
//text("friction=0.5*normal",width-200, height-8);

PVector g = new PVector(0,0.1);
```

Physics: Box

```
float Nsize = g.mag()*cos(s);
PVector N = new PVector(-sin(s), -cos(s));
N.mult(Nsize);
float Fsize = mu*N.mag();
PVector F = new PVector(cos(s),-sin(s));
F.mult(Fsize);

box.applyForce(g);
box.applyForce(N);
box.applyForce(F);
```

Physics: Box

```
  box.update();
  box.display(s);
  if (box.position.x < 0) {
    box.position.x = width;
    box.position.y = 0;
  }
}

void mousePressed() {
  loop();
}
```

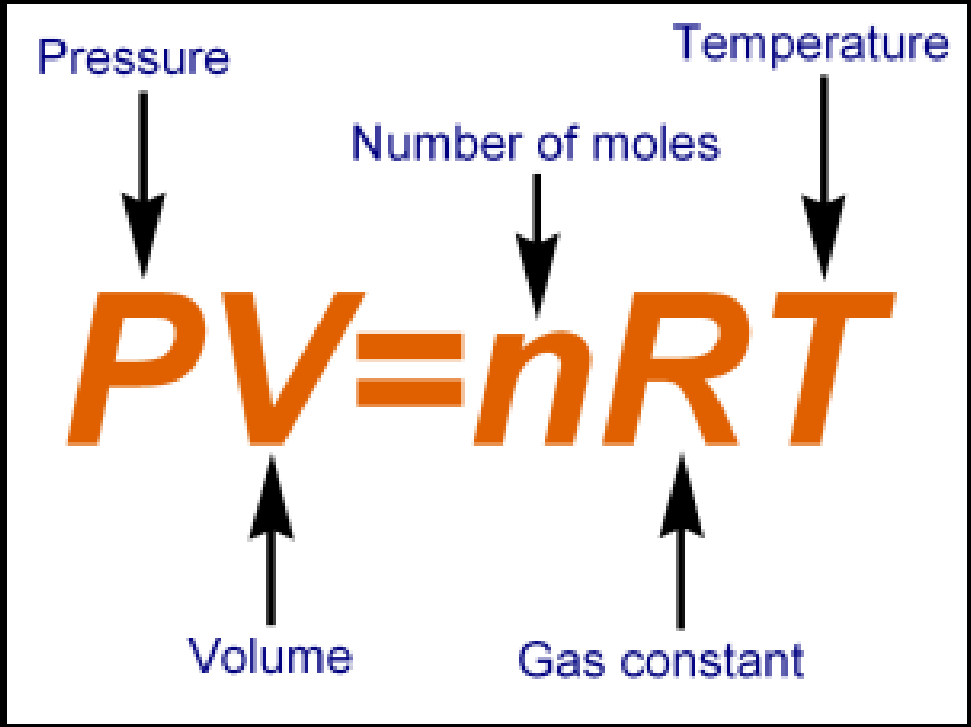
Physics: Move

- https://github.com/suakii/2017SWProfessional/tree/master/Chapter3_Physics/C3_2_BoxMove

Real Simulation

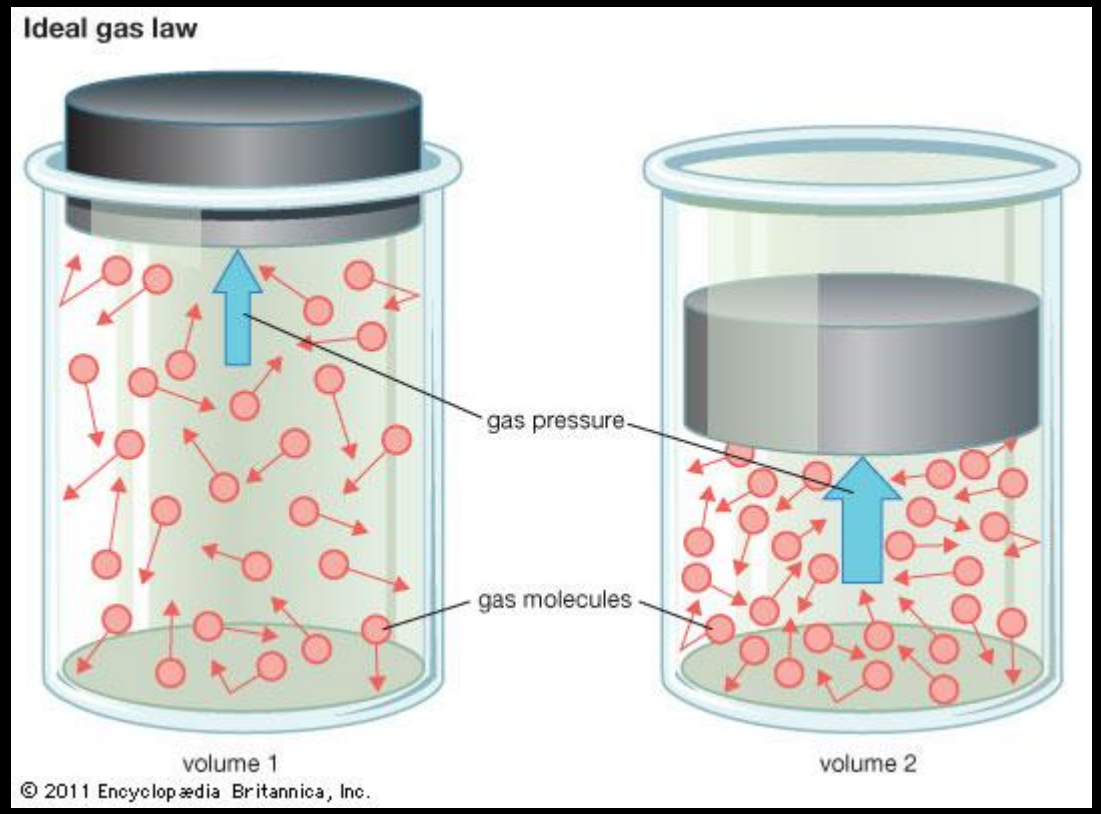
Chemistry

Chemistry



http://www.calctool.org/CALC/chem/c_thermo/ideal_gas

Chemistry

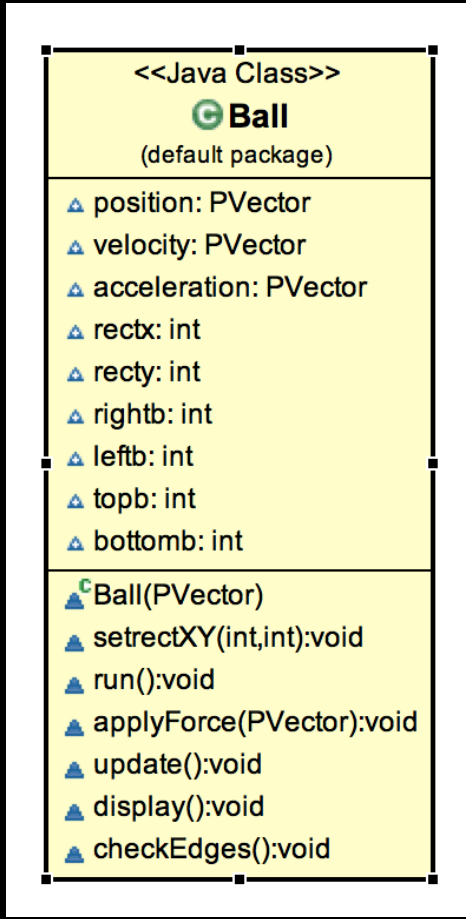


https://www.britannica.com/science/perfect-gas-law

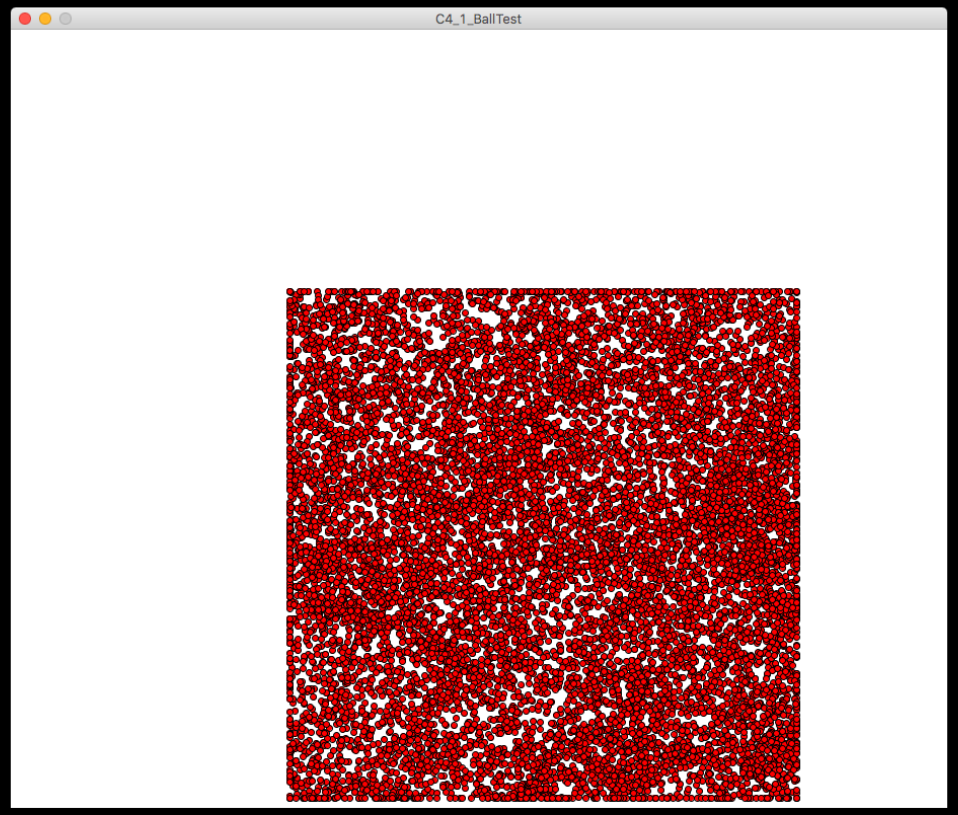
Chemistry

- According to the ideal gas law, when a gas is compressed into a smaller volume, the number and velocity of molecular collisions increase, raising the gas's temperature and pressure.

Chemistry - Ball



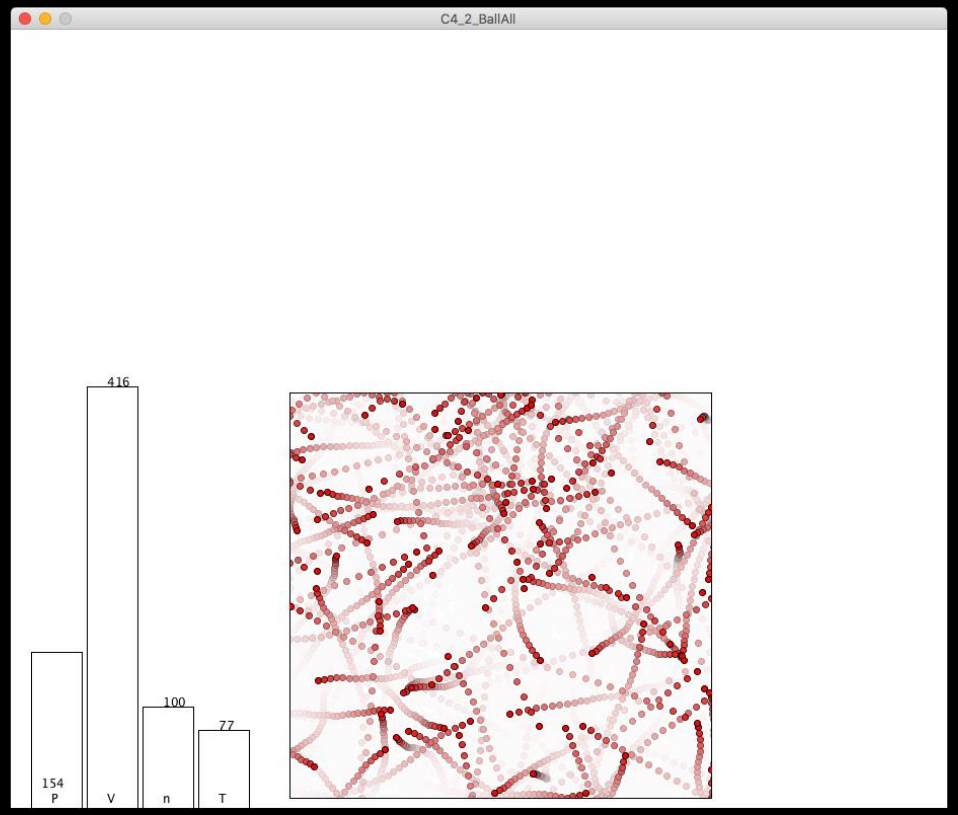
Chemistry - Ball



Chemistry - Ball

- https://github.com/suakii/2017SWProfessional/tree/master/Chapter4_Chemistry/C4_1_BallTest

Chemistry - All



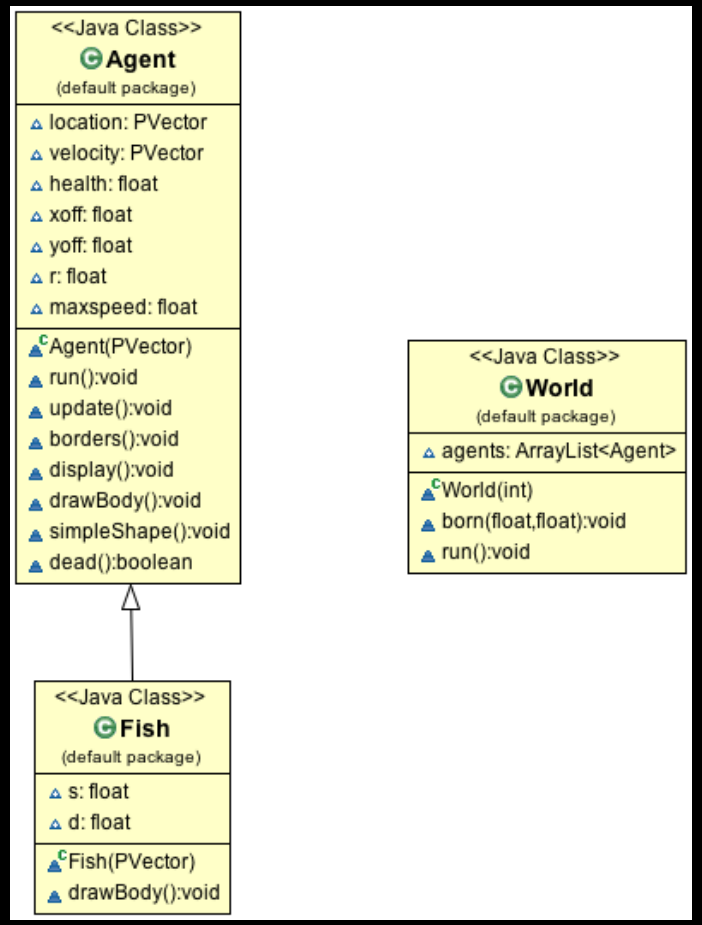
Chemistry - All

- https://github.com/suakii/2017SWProfessional/tree/master/Chapter4_Chemistry/C4_2_BallAll

Real Simulation

Biology

Biology


<<Java Class>>
Agent
(default package)
location: PVector
velocity: PVector
health: float
xoff: float
yoff: float
r: float
maxspeed: float
Agent(PVector)
run():void
update():void
borders():void
display():void
drawBody():void
simpleShape():void
dead():boolean
<<Java Class>>
World
(default package)
agents: ArrayList<Agent>
World(int)
born(float,float):void
run():void
<<Java Class>>
Fish
(default package)
s: float
d: float
Fish(PVector)
drawBody():void

Biology

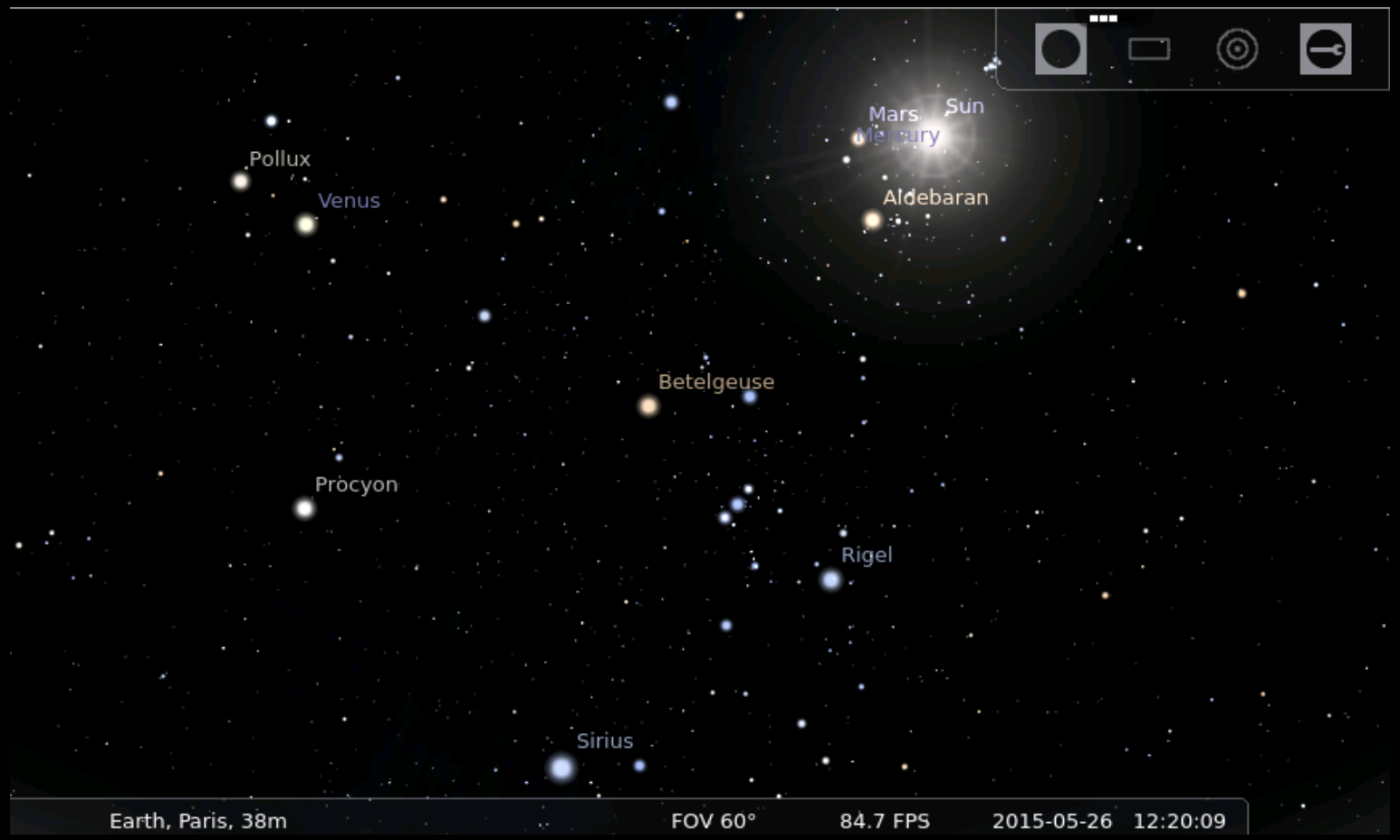
Biology

- https://github.com/suakii/2017SWProfessional/tree/master/Chapter5_Biology/MyEcoSystem

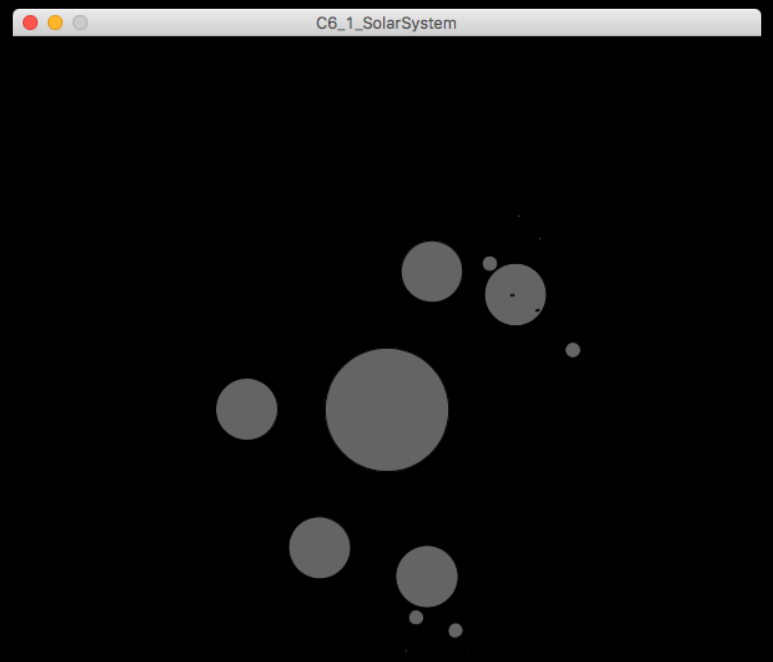
Real Simulation

Earth

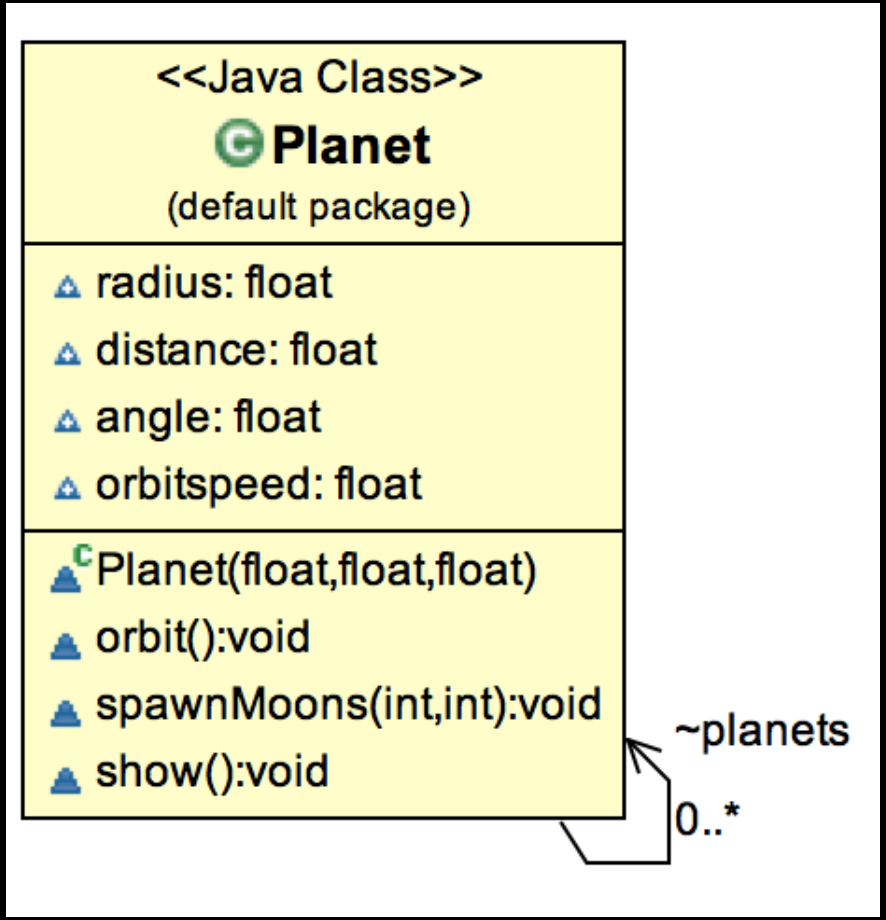
Earth – Too Difficult.....



Earth – like this…



Earth – Planet Class



Earth – Planet Main

```
Planet sun;

void setup() {
  size(600, 600);
  sun = new Planet(50, 0, 0);
  sun.spawnMoons(5, 1);
}

void draw() {
  background(0);
  translate(width/2, height/2);
  sun.show();
  sun.orbit();
}
```

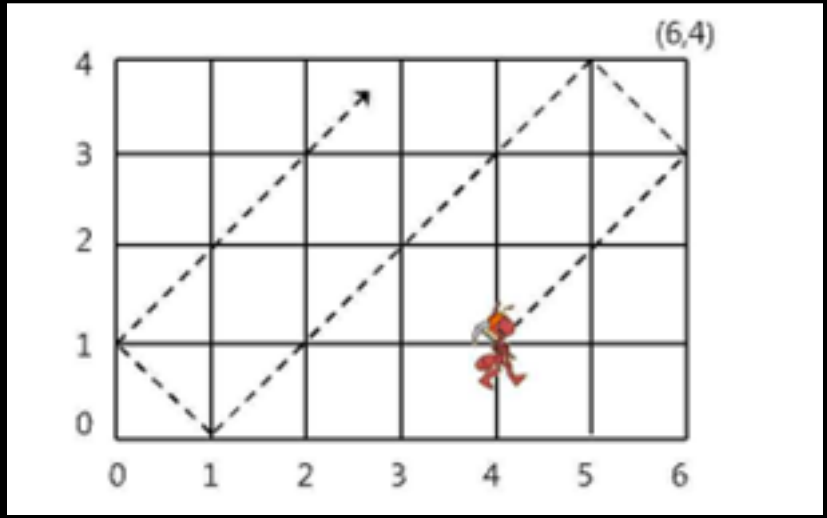
Earth

- https://github.com/suakii/2017SWProfessional/tree/master/Chapter6_Earth/C6_1_SolarSystem

Real Simulation

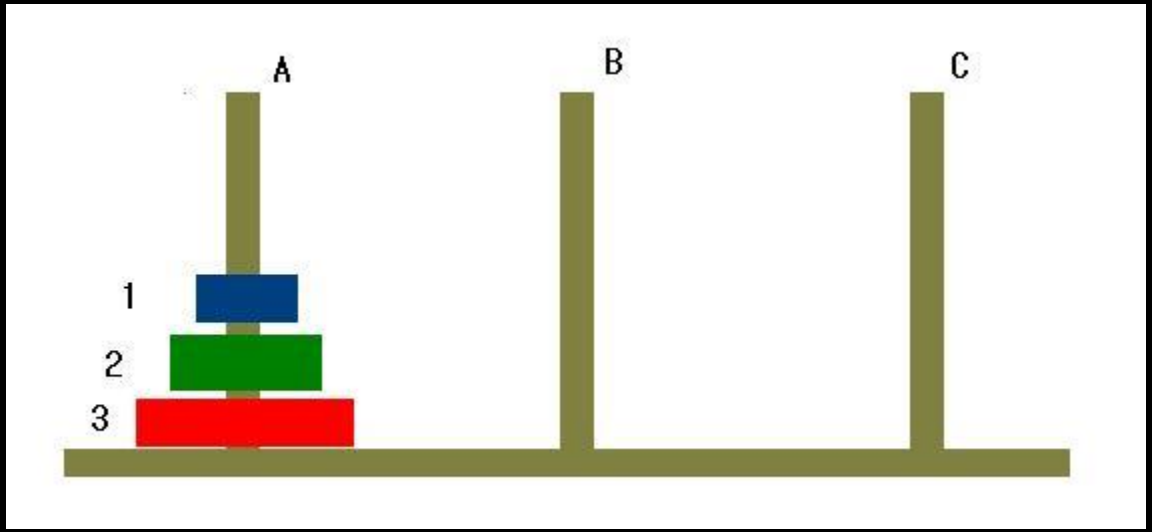
Informatics

Informatics



http://koistudy.net/?mid=prob_page&NO=1043&SEARCH=0

Informatics



Tower Test

```
class Tower {

  float x;
  float y;
  float w;
  float h;
  String name;
  Stack disks;

  Tower(float x_, float y_, float w_, float h_, String name_) {
    x = x_;
    y = y_;
```

Tower Test

```
    w = w_;
    h = h_;
    disks = new Stack();
    this.name = name_;
  }

  void display() {
    stroke(255);
    fill(0);
    rect(x, y, w, h);
  }

}
```

Tower Main

```
import java.util.Stack;


int numTower = 3;
Tower[] towers = new Tower[numTower];
```

Tower Main

```
float towerheight = 100;
float towerWidth = 10;
float towerspacing = 150;

int i;

void setup() {
  size(800, 400);
```

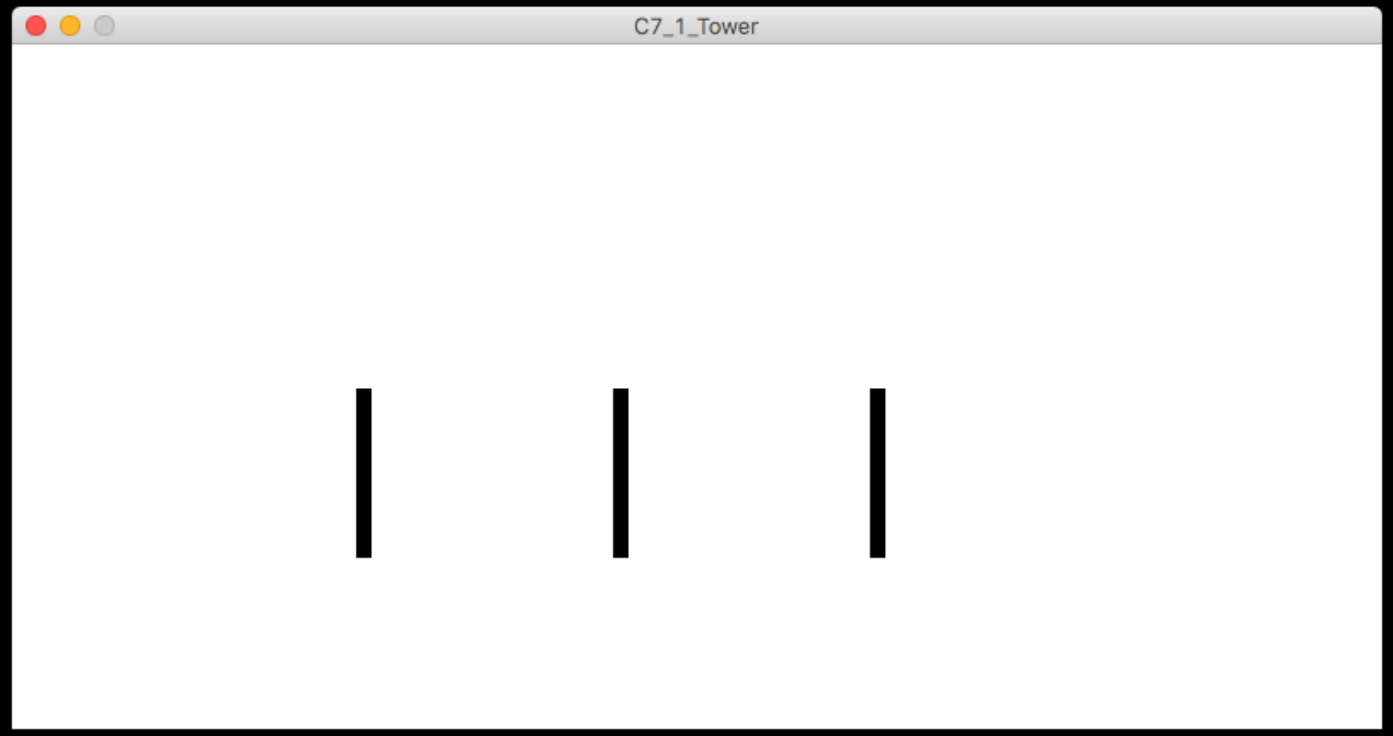
Tower Main

```
  for (int i = 0; i < towers.length; i++) {
    towers[i] = new Tower(width/4 + i*towerspacing,
height/2,towerWidth,towerheight, str(char('A'+i)));
  }

}
void draw() {
  background(255);
```

Tower Main

```
  for (int i = 0; i < towers.length; i++) {
    towers[i].display();
    //println(towers[i].name);
  }

}
```

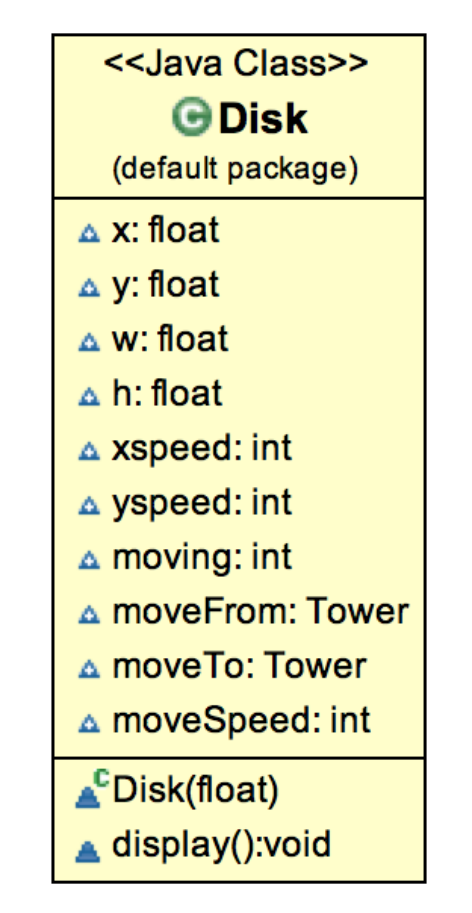
Tower Test



Tower Test

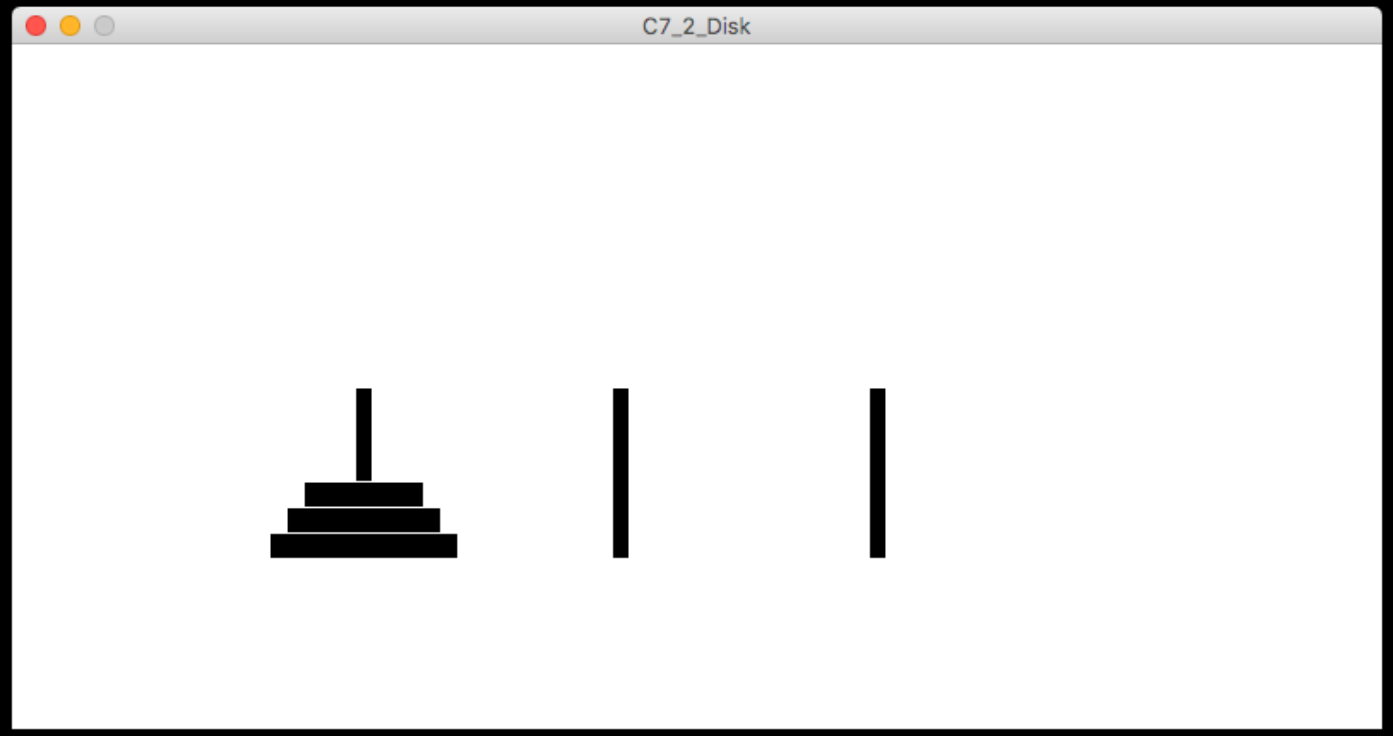
- https://github.com/suakii/2017SWProfessional/tree/master/Chapter7_Informatics/C7_1_Tower

Disk Test


<<Java Class>>
Disk
(default package)
x: float
y: float
w: float
h: float
xspeed: int
yspeed: int
moving: int
moveFrom: Tower
moveTo: Tower
moveSpeed: int
Disk(float)
display():void

Disk Test

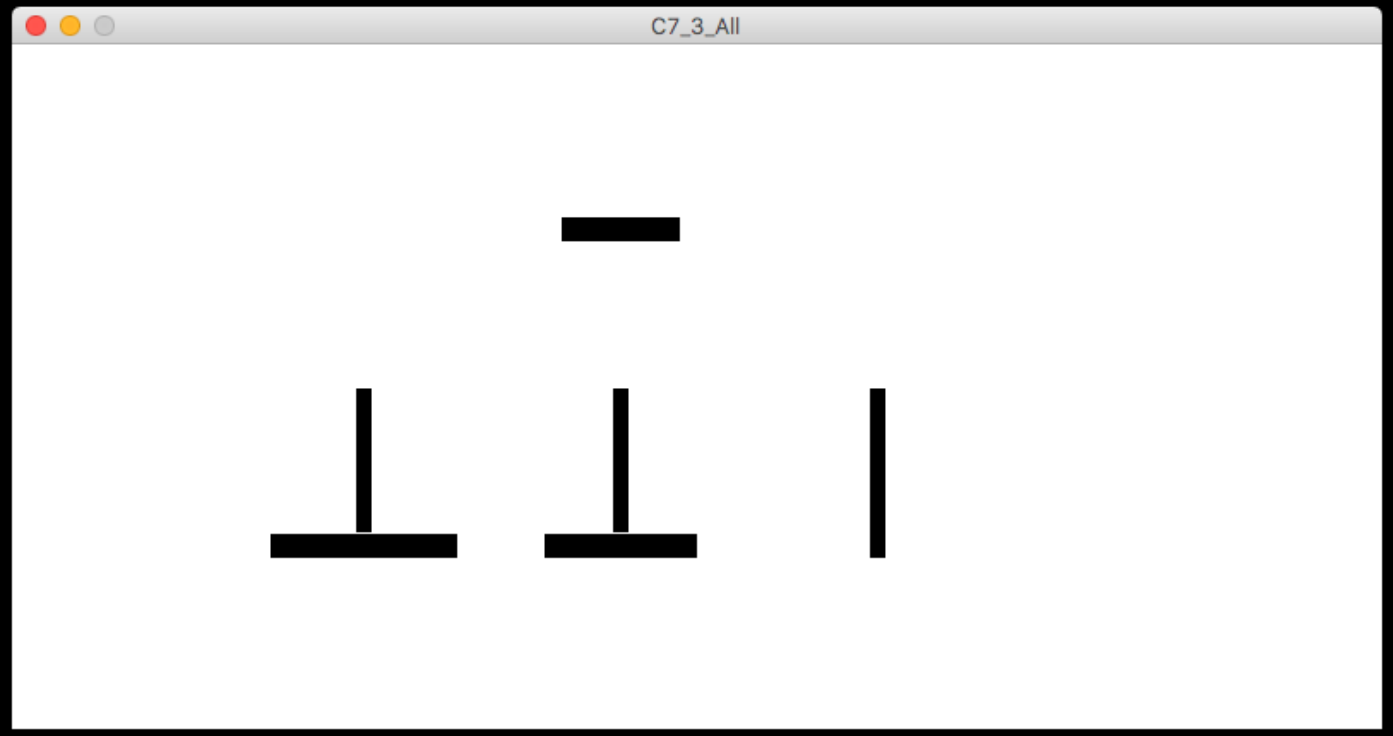
- https://github.com/suakii/2017SWProfessional/tree/master/Chapter7_Informatics/C7_2_Disk


C7_2_Disk

Move Class

```
class Move {

  Tower from;
  Tower to;
  int n;

  Move(int n_, Tower from_, Tower to_) {
    n = n_;
    from = from_;
    to = to_;
  }

}
```

All



All

- https://github.com/suakii/2017SWProfessional/tree/master/Chapter7_Informatics/C7_3_All

감사합니다

suakii@gmail.com

경기과학고등학교 교사
박종화

과학기술정보통신부
교육부
한국과학창의재단
Korea Foundation for the Advancement of Science & Creativity