



5. Physics Libraries



박종화
suakii@gmail.com

Physics Libraries

- Learned about concepts from the world of physics — What is a vector? What is a force? What is a wave? etc.
- Understood the math and algorithms behind such concepts.
- Implemented the algorithms in Processing with an object-oriented approach.

Object

여러 운동들을 시뮬레이션

직접 설계한 월드의 물리 법칙을
창의적으로 정의



물리 라이브러리

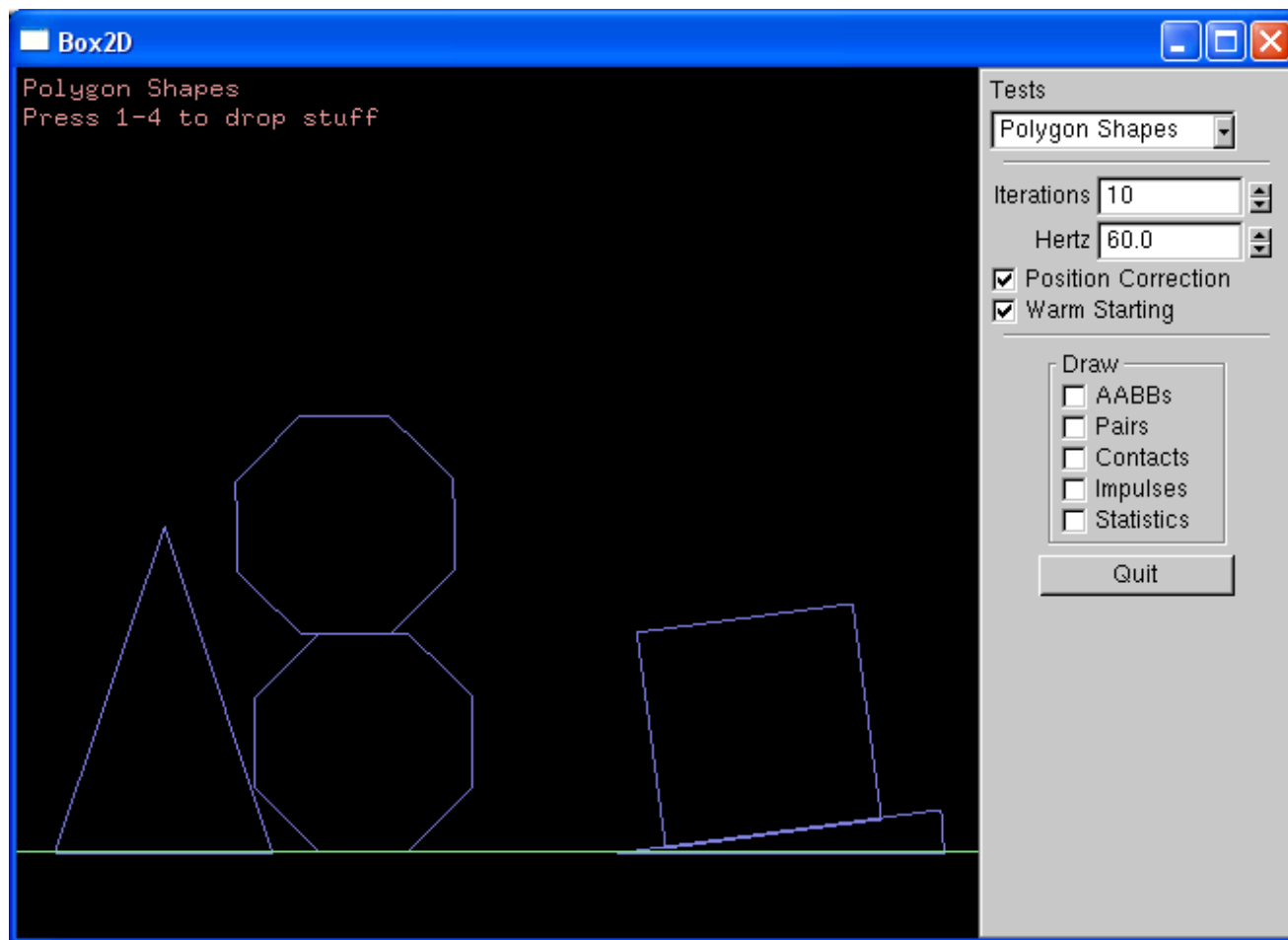
What Is Box2D?

- Box2D is open source C++ engine for simulating rigid bodies in 2D.
- Box2D is developed by Erin Catto and has the [zlib](#) license. While the zlib license does not require acknowledgement, we encourage you to give credit to Box2D in your product.

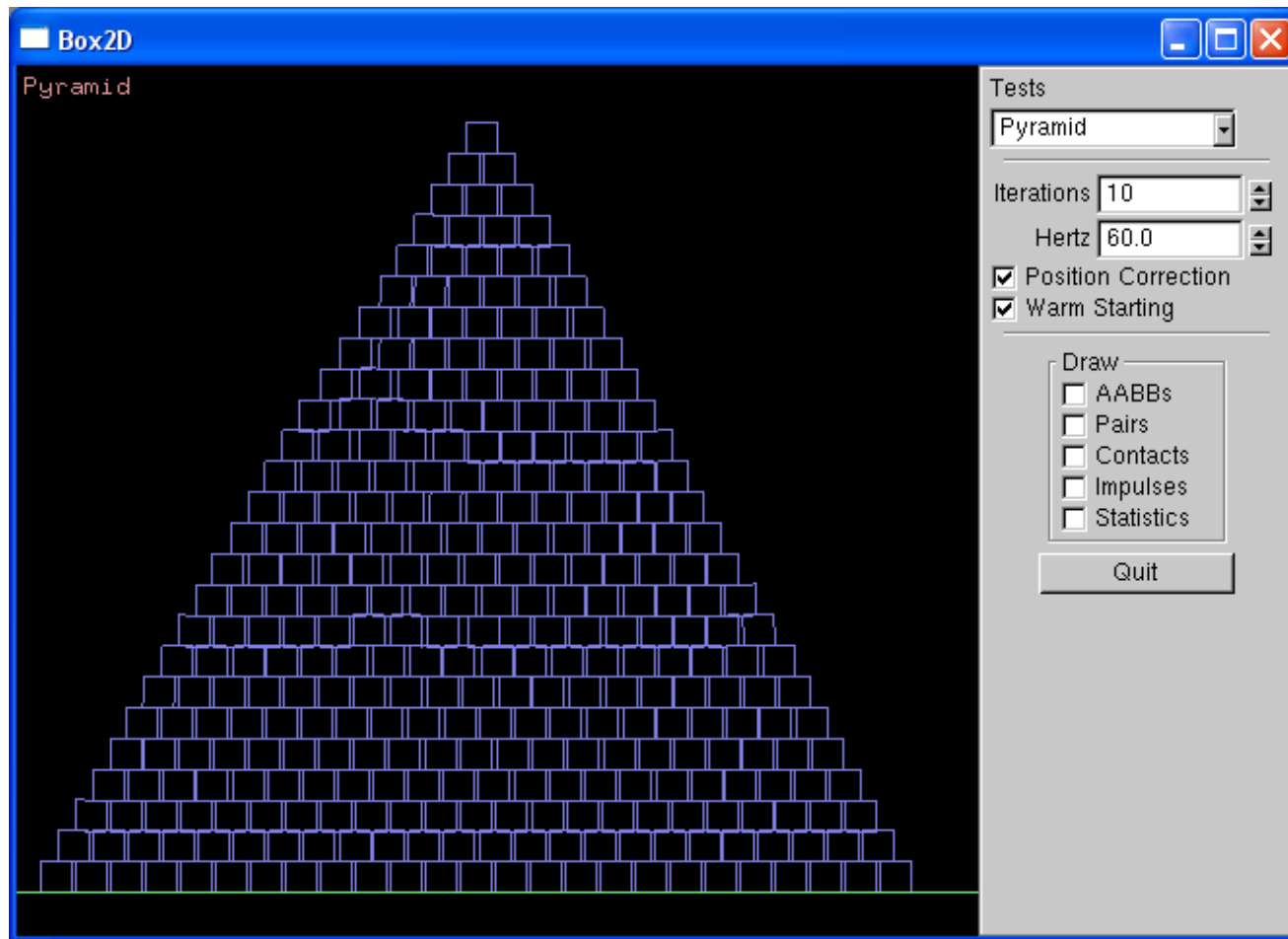
Box2D

- www.box2d.org
- Box2D의 단위 체계
 - Meters, kilograms, seconds, etc.

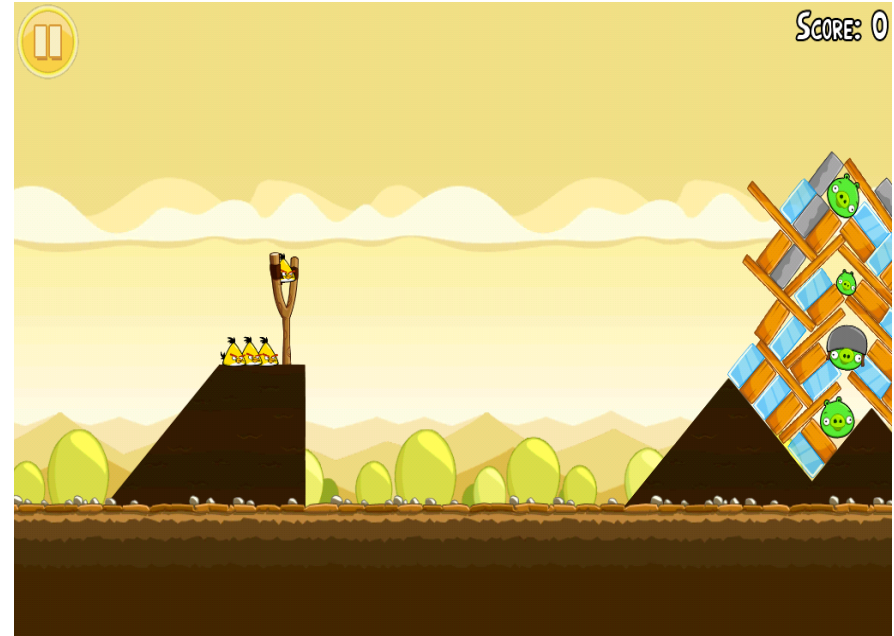
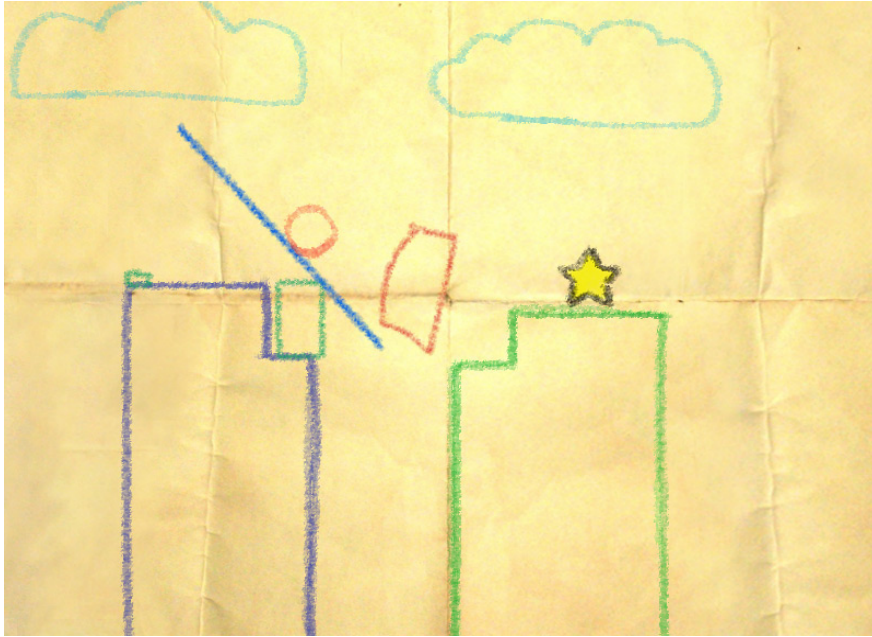
Box2D



Box2D



Box2D



시초: Erin Catto가 C++로 작성한 물리 튜토리얼
'진짜' 물리엔진
그래픽, 픽셀 X
Input: 수 Output: 수

Box2D

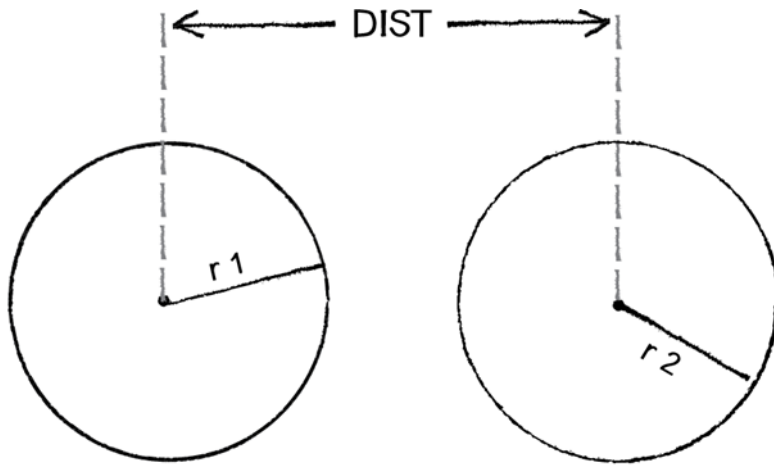
- Erin: "Hi Peter, could you tell me which physics engine Angry Birds uses?"
- Peter: "Box2D"
- Erin: "Great. Would you consider giving credit to Box2D in your game?"
- Peter: "Yes, of course"
- Erin: "Thank you! By the way, I am Erin Catto the creator of Box2D"
- Peter: "Great! I would like to talk to you after the session"

PBox2D

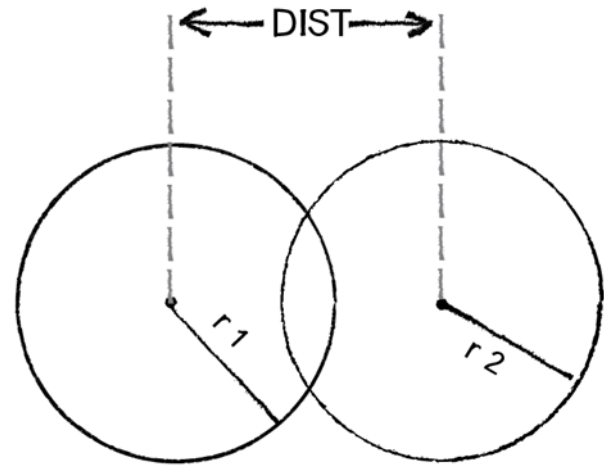
- C++로 쓰여진 Box2D를 자바로 JBox2D
- 프로세싱에서 반복해서 쓰는 기능만 모아보자
 - PBox2D ^^

Why Box2D?

- 충돌을 다룰 수 있다.



$DIST > (r_1 + r_2)$
not intersecting



$DIST < (r_1 + r_2)$
intersecting

Box2D Basics

- Setup
 - 객체 생성
- Draw
 - 힘을 계산
 - 객체들에 힘을 적용
 - 가속도와 속도를 기반으로 객체 위치 변경
 - 객체를 그린다.

Box2D Basics

- Setup
 - 객체를 생성
- Draw
 - 객체를 그린다.

Box2D

- Setup
 - 픽셀 단위로 모든 객체를 만들어 준다.
 - 픽셀 단위를 Box2D 세계의 단위로 변환
- Draw
 - 객체의 위치를 Box2D에게 물어봄
 - Box2D의 단위를 픽셀로 변환
 - 객체를 그린다.

Box2D



Box2D

- **World:**
 - 물리 시뮬레이션을 관리, 좌표공간을 관리하고 생성한 모든 객체를 저장
- **Body:**
 - Body는 좌표와 속도를 가진다.
- **Shape:**
 - 보디에 붙어서 충돌과 관련된 지오메트리 처리
- **Fixture:**
 - 셰이프를 보디에 붙이고 밀도, 마찰, 복원력 등의 요소를 적용
- **Joint:**
 - 2개의 보디(또는 1개의 보디와 월드)를 연결해서 상호 작용을 일으키게 함.
- **Vec2:**
 - Box2D에서 사용할 수 있는 벡터

Pvector VS Vec2

PVector	Vec2
<pre>PVector a = new PVector(1,-1); PVector b = new PVector(3,4); a.add(b);</pre>	<pre>Vec2 a = new Vec2(1,-1); Vec2 b = new Vec2(3,4); a.addLocal(b);</pre>
<pre>PVector a = new PVector(1,-1); PVector b = new PVector(3,4); PVector c = PVector.add(a,b);</pre>	<pre>Vec2 a = new Vec2(1,-1); Vec2 b = new Vec2(3,4); Vec2 c = a.add(b);</pre>

PVector	Vec2
<pre>PVector a = new PVector(1,-1); float n = 5; a.mult(n);</pre>	<pre>Vec2 a = new Vec2(1,-1); float n = 5; a.mulLocal(n);</pre>
<pre>PVector a = new PVector(1,-1); float n = 5; PVector c = PVector.mult(a,n);</pre>	<pre>Vec2 a = new Vec2(1,-1); float n = 5; Vec2 c = a.mul(n);</pre>

PVector	Vec2
<pre>PVector a = new PVector(1,-1); float m = a.mag(); a.normalize();</pre>	<pre>Vec2 a = new Vec2(1,-1); float m = a.length(); a.normalize();</pre>

Living in a Box2D World

- World
 - 모든 것을 관리하는 객체
 - 모든 객체의 좌표를 관리
 - 시간 체크

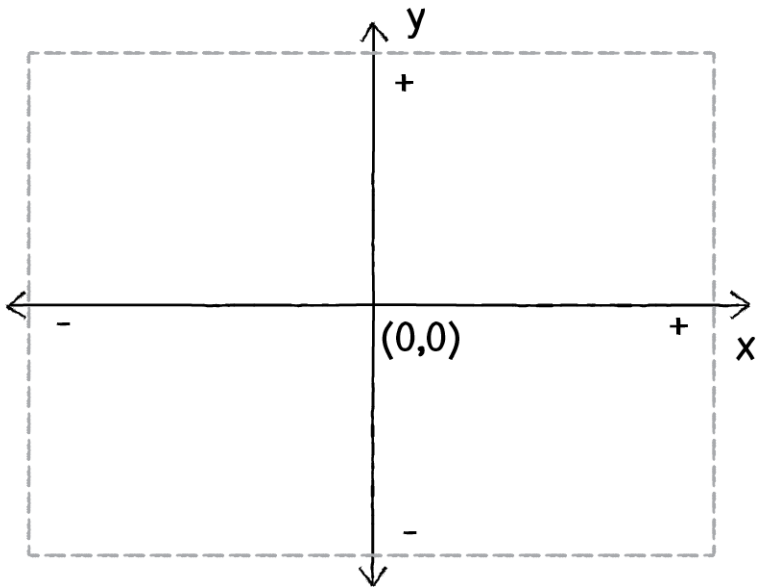
World

```
import shiffman.box2d.*;  
import org.jbox2d.collision.shapes.*;  
import org.jbox2d.common.*;  
import org.jbox2d.dynamics.*;
```

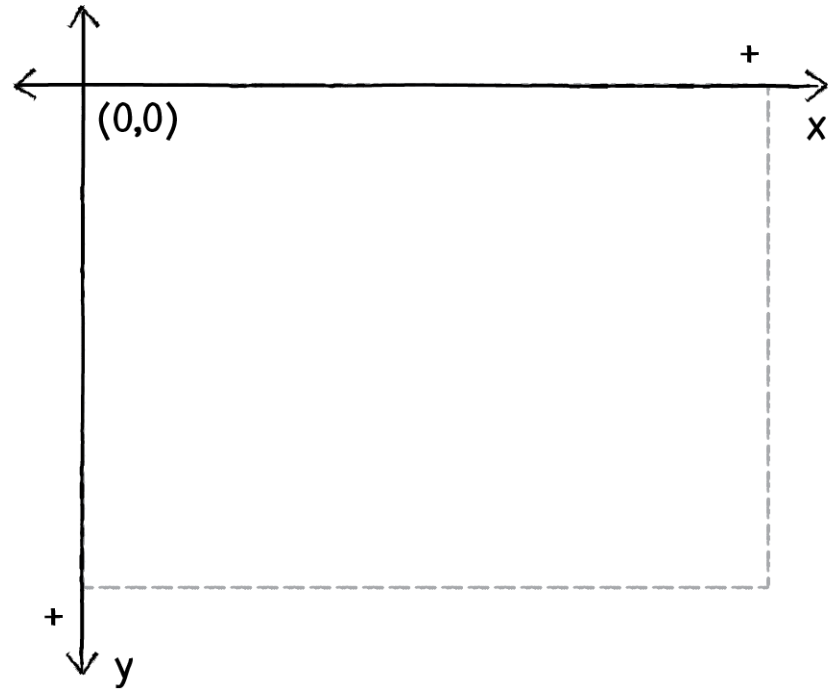
```
Box2DProcessing box2d;
```

```
void setup() {  
    box2d = new Box2DProcessing(this);  
    box2d.createWorld();  
}
```

World



Box 2D world



Processing World

Coordinate Convert

Task	Function
Convert location from World to Pixels	<code>Vec2 coordWorldToPixels(Vec2 world)</code>
Convert location from World to Pixels	<code>Vec2 coordWorldToPixels(float worldX, float worldY)</code>
Convert location from Pixels to World	<code>Vec2 coordPixelsToWorld(Vec2 screen)</code>
Convert location from Pixels to World	<code>Vec2 coordPixelsToWorld(float pixelX, float pixelY)</code>
Scale a dimension (such as height, width, or radius) from Pixels to World	<code>float scalarPixelsToWorld(float val)</code>
Scale a dimension from World to Pixels	<code>float scalarWorldToPixels(float val)</code>

Building a Box2D Body

- Mover 객체처럼 공간에 자신의 위치를 가지고 힘을 받게 된다.
- It's important to note, however, that a body has no geometry; it isn't anything physical.
 - 물리적인 반응을 하지 않는다.
- Shape를 붙여서 물리적인 반응 가능

Building a Box2D Body

- Step 1: Define a body
 - BodyDef 객체를 생성
 - 바디의 속성을 정의한다.
- Make a body definition before making a Body
 - `BodyDef bd = new BodyDef();`

Building a Box2D Body

- Step 2: Configure the body definition
 - 바디의 특성 또는 속성을 정의

```
Vec2 center = box2d.coordPixelsToWorld(width/2,height/2));
```

```
bd.position.set(center);
```


Building a Box2D Body

- Body's type
 - Dynamic
 - 모든 물리 시뮬레이션이 적용되며, 제일 많이 사용된다. 동적 보디는 월드를 움직이며 다른 보디와 상호 작용 가능. 외부 힘에도 반응
 - Static
 - 움직이지 않음. 틀과 같은 환경을 만들 때.
 - Kinematic
 - 속도를 직접 설정해서 수동으로 움직일 수 있다. 사용자가 움직이게 만들 수 있는 요소를 월드 안에 넣고 싶을 때 사용. 동적 보디와만 충돌을 일으킴

Building a Box2D Body

- `bd.fixedRotation = true;`
- `bd.linearDamping = 0.8;` //선형 감쇠
- `bd.angularDamping = 0.9;` // 회전 감소
- `bd.bullet = true;`
-

Building a Box2D Body

- Step 3: Create Body
 - `Body body = box2d.createBody(bd);`
- Step 4: set other..
 - `body.setLinearVelocity(new Vec2(0, 3));`
 - `body.setAngularVelocity(1.2);`

Bodies & Shapes & Fixtures

- 보디 자체는 물리적인 반응을 하지 못한다.
 - $\pi\pi$
- 보디에 질량을 부여하려면 셰이프를 만들고 픽스처로 셰이프를 보디에 부착한다.

Bodies & Shapes & Fixtures

- Step 1 : Define a shape
 - PolygonShape ps = new PolygonShape();

```
float box2Dw = box2d.scalarPixelsToWorld(150);  
float box2Dh = box2d.scalarPixelsToWorld(100);
```

```
ps.setAsBox(box2Dw, box2Dh);
```

Bodies & Shapes & Fixtures

- Step 2: Create a fixture
 - 생성한 셰이프를 보디에 붙이려면 픽스처를 사용
 - FixtureDef
 - FixtureDef fd = new FixtureDef();
 - fd.shape = ps;
 - The fixture is assigned the PolygonShape we just made

```
fd.friction = 0.3;
```

마찰계수

```
fd.restitution = 0.5;
```

탄성계수

```
fd.density = 1.0;
```

밀도

Bodies & Shapes & Fixtures

- Step 3: Attach the shape to the body with the fixture
 - `body.createFixture(fd);`

Summary

- Define a body using a BodyDef object (set any properties, such as location).
- Create the Body object from the body definition.
- Define a Shape object using PolygonShape, CircleShape, or any other shape class.
- Define a fixture using FixtureDef and assign the fixture a shape (set any properties, such as friction, density, and restitution).
- Attach the shape to the body.

Box2D and Processing

- Box2D는 물리와 관련된 연산만 책임
- 결국 화면에 그리는 것은 우리의 책임

