

たのしいコンピュータ プログラミング



君にもできる
シューティングゲーム



木更津高専 情報工学科
1日体験入学

プログラミングを体験してみよう

Step1 まずは動かしてみよう.

```
rect(10, 20, 30, 40);
```

Step2 ウィンドウサイズを大きくしてみよう. (太字部分を追加する)

```
size(400, 600);  
rect(60, 20, 200, 250);
```

Step3 表示モードをセンターにしてみよう.

```
rectMode(CENTER);  
size(400, 600);  
rect(200, 300, 150, 150);
```

Step4 いろいろ表示してみよう.

```
rectMode(CENTER);  
size(400, 600);  
rect(200, 470, 45, 45);  
rect(70,70,60,85);  
rect(190,200,60,85);  
rect(310,150,60,85);
```

Step5 変数を使ってみよう.

```
size(400, 600);  
rectMode(CENTER);  
rect(200, 470, 45, 45);  
int x, y;  
x = 60;  
y = 85;  
rect(70,70, x,y);  
rect(190,200, x,y);  
rect(310,150, x,y);
```

Step6 画像を表示してみよう. practice/step2_7を使います.
以下の行を変更して戦闘機の位置を変更してみましょう.

```
image(shipImg, 200, 450);
```

Step7 画像を動かしてみよう (practice/step3_1を使います)

```
// 1秒間に60回画面を更新する
void draw() {
  background(0); // 背景を黒くする.
  shipX = shipX + 3;
  image(shipImg, shipX, shipY);
}
```

Step8 画像を左に動かしてみよう.

```
// 1秒間に60回画面を更新する
void draw() {
  background(0);
  shipX = shipX - 3; // -3に変更して左に動かす.
  image(shipImg, shipX, shipY);
}
```

Step9 画像を止めてみよう.

```
// 1秒間に60回画面を更新する
void draw() {
  background(0);
  if (shipY > 100) { // もし, shipYが100よりも大きいなら,
    shipY = shipY - 3;
  }
  image(shipImg, shipX, shipY);
}
```

プログラムを改造してみよう

Step1 ゲームを起動する.

デスクトップ → 体験入学 → Edatos → Edatos.pdeをダブルクリック.

Step2 自機のHPを大きくする.

Shipのタブを選んで, 21行目を以下のようにHPの値を修正.

21行目	maxHp = hp = 3000;
------	--------------------

Step3 自機を上下にも移動できるようにする.

39行目と42行目の先頭の//を削除する.

39行目	y = y - speedY;
42行目	y = y + speedY;

Step4 自機の移動速度を速くする.

24, 25行目を以下のようにspeedの値を修正.

24行目	speedX = 20;
25行目	speedX = 20;

Step5 レーザーの発射間隔を速くする.

26行目を以下のように発射の間隔時間30から10に短くする.

26行目	rappid = 10;
------	--------------

Step6 敵の倒したときの得点を大きくする.

Enemyのタブを選んで, 43行目を以下のようにscoreに与える値を修正.

43行目	score = score + 1250;
------	-----------------------

Step7 敵の出現数を多くする.

Edatosのタブを選んで, 37行目を以下のように修正. これにより敵の出現確率が30%, 敵が最大10体出現するようになる.

37行目	if (random(1.0) < 0.3 && enemies.size()<10) {
------	---


Step8 敵が発射する弾の数を多くする.

51行目を以下のように修正して, 発射する確率を50%にする.

51行目	if (random(1) < 0.5) {
------	------------------------

Processingのインストール

1. <https://processing.org/> にアクセスして, Download Processing をクリック.



Processing p5.js Processing.py Processing for Android Processing for Pi Processing Foundation

Processing

Cover

Download
Donate

Reference
Libraries
Tools
Environment

Tutorials
Examples
Books

» Forum
» GitHub
» Issues
» Wiki
» FAQ

» **Download Processing**
» Browse Tutorials
» Visit the Reference

Processing is a flexible software sketchbook and a language for learning how to code within the context of the visual arts. Since

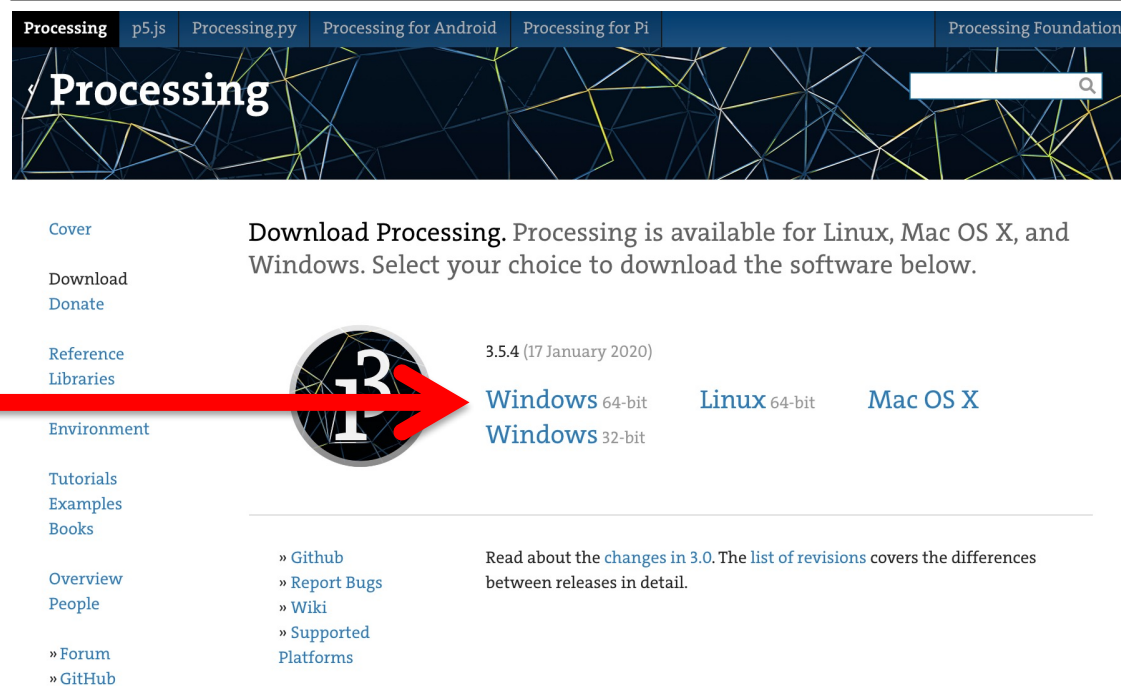
» Donate

Please support the [Processing Foundation](#). We need your help!

» PCD 2020

The Processing Community Day (PCD) initiative is evolving. For 2020, we will offer a mentorship program for PCD Worldwide Organizers who are interested in learning from past community organizers and mentors. The goal is to help a diverse group of organizers launch a PCD in their local communities. Check out the [PCD @ Worldwide](#) site to learn more about starting or attending an event in 2020!

2. Windows (64bit) をクリックして, zipファイルをダウンロード. 解凍して実行する.
(パソコンのOSに応じて選択する)



Processing p5.js Processing.py Processing for Android Processing for Pi Processing Foundation

Processing

Cover

Download
Donate

Reference
Libraries

Environment

Tutorials
Examples
Books

Overview
People

» Forum
» GitHub

» **Download Processing**
» Report Bugs
» Wiki
» Supported Platforms

Read about the [changes in 3.0](#). The [list of revisions](#) covers the differences between releases in detail.

Download Processing. Processing is available for Linux, Mac OS X, and Windows. Select your choice to download the software below.

3.5.4 (17 January 2020)

Windows 64-bit Linux 64-bit Mac OS X
Windows 32-bit

・ サンプルプログラムの紹介

1. 体験入学で使った教材

<https://tinyurl.com/kisarazu-taiken>

2. 情報工学科のホームページからでもリンクを辿れます。

<http://www.kisarazu.ac.jp/gakka/information/>



3. 体験入学で作成したシューティングゲームの実況プログラミング

<http://youtu.be/W8FQhGHLIC4>

・ 参考文献

1. “Processing アニメーションプログラミング入門”, 田中孝太郎, 技術評論社, 2011.

2. Processingのサイト, <https://processing.org>

3. フリー素材画像

(現在サイトがなくなっている. 2015年時点に下記サイトのフリー素材を使用した.)

<http://game.yu-nagi.com/index.htm>

<http://homepage2.nifty.com/hamcorossam/>