

情報学部 プログラミング体験教室

1

プログラミングと言えば…



出典: <https://www.lego.com/ja-jp/themes/boost>



出典: <https://scratch-howto.com/>

2

より「本格的な」プログラミング

```
struct sockaddr_in server_addr;

void setup_server_addr(char *server_host, int server_port){
    struct hostent *hp;

    hp = gethostbyname(server_host);
    endhostent();

    if(hp == NULL){
        fprintf(stderr, "host %s unknown\n", server_host);
        exit(1);
    }

    bzero((char *) &server_addr, sizeof(server_addr));
    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(server_port);
    bcopy(hp->h_addr_list[0], (char *) &server_addr.sin_addr, 4);
}

void print_err_msg(char *s){
    fprintf(stderr, "Usage: %s [-v] -f <playername.txt> -c <server:port>\n", s);
    exit(1);
}
```

```
private boolean sendGreetingMsg(int id) throws ComException, ComTimeoutException {
    players[id].sendMsg("hello!shipserver protocol version "+ Constants.PROTOCOLVERSION + "\n");

    String[] msgs = players[id].readMsgs();
    players[id].setName(msgs[0]);
    logWriter.println("id:" + id, msgs[0]);
    String[] shipStr = new String[msgs.length-1];
    for(int i = 1; i < msgs.length; i++){
        shipStr[i-1] = msgs[i];
    }
    ShipsValidator validator = new ShipsValidator(Constants.BOARDSIZE);

    try{
        ShipGenerator gen = new ShipGenerator(shipStr);
        Ship[] ships = gen.generate();

        for(Ship s : ships){
            if(validator.assignShip(s)){ return false; }
            logWriter.println("Ship", s.toString());
        } catch(NumberFormatException e){
            e.printStackTrace();
            return false;
        }
        assert validator.isValid();
        board.sendMsg(validator.getValidBoard());
        return true;
    }
```

3

大学で学ぶプログラミングを「体験」

- ・初級編(午前の部)
 - ・大部分は作成済み
 - ・キーボード入力をはなるべく減らして
 - ・プログラムを少し変更
 - 見た目や動きが変わる！を実感
- ・中級編(午後の部)
 - ・より「ゼロから」に近い作り方で
 - ・キーボード入力もして、より本格的に
 - ・プログラムの内容の理解にも踏み込む
 - 「見よう見まね」の一步先へ



4

サンプルゲームを動かそう！

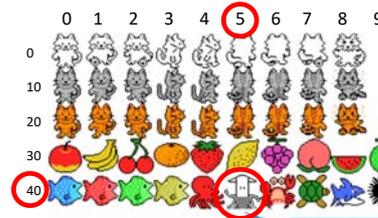
- ・まずは実行してみましょう
 - ・デスクトップの「オープンキャンパス」を開きます
 - ・「oc2019」を開きます
 - ・index.htmlを開きます
 - ・ブラウザが起動します
 - ・キャラクターはマウスポインタについてきます
 - ・たまに弾を撃ちます
- ・実行をやり直す場合は、再読み込みボタンを押します



5

プレイヤーの画像を変えよう！(1)

- ・neko.pngを開く → 表示したい絵を選ぶ
- ・絵の番号を調べる



絵の番号:45番

6

プレイヤーの画像を変えよう！(2)

- main1.jsを開く(テキストエディタが起動)
- class Player という行の場所まで移動する
- 次の行の先頭の // を削除する。

削除 //this.p=\$pat_neko+45;
• 番号を選んだ絵の番号に変更する;

```
17 |  
18 |  
19 |  
20 | *main() {  
21 |   //this.p=$pat_neko+45;  
22 |   this.count=0;  
23 |   while (true) {  
24 |     this.x=$mouseX;  
25 |     this.y=$mouseY;  
26 |     if (this.count==30) {  
27 |       new PlayerBullet({x:this  
28 |         this.count=0;  
29 |     }  
30 |     this.count+=1;  
31 |   }  
32 | }
```

- Ctrl+Sで保存しよう
- ブラウザを再読み込みして確認してみよう

7

弾の画像を変えよう！(1)

- 同じように一枚絵を選ぶ
- 先ほど追加した行を選択し、コピーする

Shift + 矢印キーで選択

```
18 | class Player extends Actor {  
19 |   *main() {  
20 |     this.p=$pat_neko+45;  
21 |     this.count=0;  
22 |     while (true) {  
23 |       this.x=$mouseX;  
24 |       this.y=$mouseY;  
25 |       if (this.count==30) {  
26 |         new PlayerBullet({x:this  
27 |           this.count=0;  
28 |       }  
29 |       this.count+=1;  
30 |     }  
31 |   }  
32 | }
```

Ctrl+C でコピー

8

弾の画像を変えよう(2)

- class PlayerBullet という行の場所まで移動する
- Ctrl+Vで貼り付け、番号を選んだ絵の番号に変更する;

```
38 | class PlayerBullet extends Actor {  
39 |   *main() {  
40 |     this.p=$pat_neko+45;  
41 |     this.vx=10;  
42 |     while (this.x<400) {  
43 |       this.x+=this.vx;  
44 |       var c=this.crashTo(Enemy);  
45 |     }  
46 | }
```

Ctrl+V で貼り付け

実行(再読み込み)して確認してみよう!

9

敵を出してみよう!

- 今の状態では敵が出てこない…
- class Start という行の場所まで移動する
- 次の行の先頭の // を削除する。

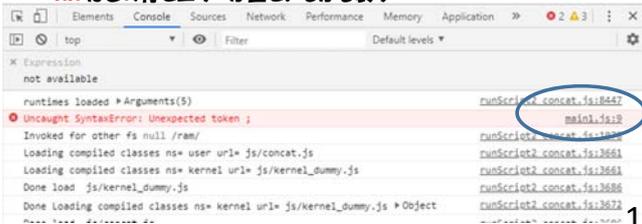
削除 //new Enemy({ x:300 , y:200 , vx:-3 , vy:1 });

```
6 | class Start extends Actor {  
7 |   *main() {  
8 |     $player=new Player;  
9 |     new Enemy({x:300, y:200, vx:-3, vy:1});  
10 |     while(true) {  
11 |       $t++;  
12 |       yield;  
13 |     }  
14 |   }  
15 | }
```

10

エラーが出たら…

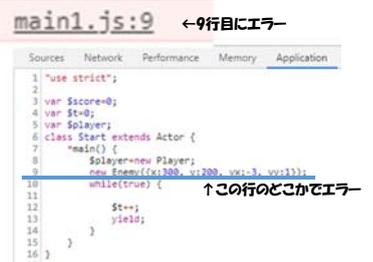
- ブラウザの画面をクリック
- Ctrl+Shift+Jを押して、「コンソール」を表示
- エラーメッセージの左端 main1.js:XX をクリック
 - XX はどの行でエラーが出ているかを表す



11

エラーが出たら…

- エラーの発生箇所を確認



- テキストエディタで直す

```
6 | class Start extends Actor {  
7 |   *main() {  
8 |     $player=new Player;  
9 |     new Enemy({x:300, y:200, vx:-3, vy:1});  
10 |     while(true) {  
11 |       $t++;  
12 |     }  
13 |   }  
14 | }
```

12

敵と敵の弾の画像を変えよう！(1)

- 一枚絵を選ぶ
- class Player という行の場所まで移動する
- this.p=\$pat_neko+XX; という行を選択し、コピーする
↑(xxには数字が入る)

```
18 class Player extends Actor {
19     *main() {
20         this.p=$pat_neko+45;
21         this.count=0;
22         while (true) {
23             this.x=$mouseX;
24             this.y=$mouseY;
25             t+=30;
26             Bullet({x:this.x, y:this.y});
27             this.count++;
28         }
29     }
30 }
```

Shift + 矢印キーで選択

Ctrl+C でコピー

13

敵と敵の弾の画像を変えよう！(2)

- class Enemy という行の場所まで移動して、Ctrl+V で貼り付ける
- class Enemy {
 *main () {
 this.p=\$pat_neko+選んだ絵の番号;
- 「EnemyBullet」も同様に行なう

実行して確認してみよう！

14

敵をたくさん出してみよう！(1)

- 敵が一体しか出てこない…
- class Startに行き、先ほど追加したプログラムをコピーする。
new Enemy ({ x:300, y:200, vx:-3, vy:1 });

```
8 class Start extends Actor {
9     *main() {
10         $player=new Player;
11         new Enemy({x:300, y:200, vx:-3, vy:1});
12         new Enemy({x:300, y:200, vx:-3, vy:1});
13         new Enemy({x:300, y:200, vx:-3, vy:1});
14         while(true) {
15             $t++;
16             yield;
17         }
18     }
19 }
```

コピー : Ctrl+C
貼り付け : Ctrl+V

実行して確認してみると…

15

敵をたくさん出してみよう！(2)

- やっぱり敵が一体しか出てこない？
- …実は3体重なっている
- それぞれのyの値を変えてみよう

```
new Enemy({x:300, y:200, vx:-3, vy:1});
new Enemy({x:300, y:100, vx:-3, vy:1});
new Enemy({x:300, y:150, vx:-3, vy:1});
```

実行して確認してみよう！

16

敵をたくさん出してみよう！(3)

- 敵の出現位置をランダムにしてみよう
- rnd(数値)→0から数値未満のランダムな値

```
new Enemy({ x:300, y:this.rnd(400), vx:-3, vy:1});
new Enemy({x:300, y:this.rnd(400), vx:-3, vy:1});
new Enemy({x:300, y:this.rnd(400), vx:-3, vy:1});
```

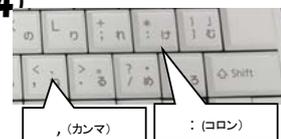
実行して確認してみよう！

17

敵をたくさん出してみよう！(4)

- 敵の移動速度をランダムにしてみよう
- rnd(5): 0から4の乱数
- rnd(5)-2 -2から2の乱数

```
new Enemy({ x:300,y:this.rnd(400),vx:-3,vy:this.rnd(5)-2 });
```



実行して確認してみよう！

18

敵をたくさん出してみよう！(5)

- ずっと敵が出てくるようにしたい！
- 繰り返し文(while文)を使う
- yield; → 処理を少し待つ

```
class Start extends Actor {
  *main() {
    $player=new Player;
    new Enemy({ x:300 , y:this.rnd(400),vx:-3, vy:this.rnd(5)-2 });
    while(true) {
      new Enemy({ x:300 , y:this.rnd(400),vx:-3, vy:this.rnd(5)-2 });
      $t++;
      yield;
    }
  }
}
```

Shift+矢印キーで選択
Ctrl+Xで切り取り

while(true)…の次の行にCtrl+Vで追加
Tabキーで字下げを忘れずに

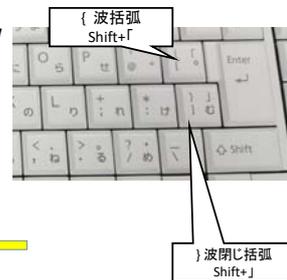
実行して確認してみよう！

19

敵をたくさん出してみよう！

- 敵が出すぎ！
- ランダムで出すようにする

```
class Start extends Actor {
  *main() {
    $player=new Player;
    while(true) {
      if (this.rnd(100)<10) {
        new Enemy({ x:300 , y:this.rnd(400) ,
          vx:-3 , vy:this.rnd(5)-2 });
      }
      $t++;
      yield;
    }
  }
}
```



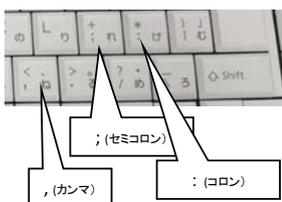
実行して確認してみよう！

20

スコアを表示してみよう！

- 画面にスコアを表示させる
- 「Start」の最初にプログラムを追加

```
class Start extends Actor {
  *main() {
    $player=new Player;
    new Score({x:200,y:300});
    while(true) {
      if (this.rnd(100)<10) {
        new Enemy({ x:300 , y:this.rnd(400) ,
          vx:-3 , vy:this.rnd(5)-2 });
      }
      $t++;
      yield;
    }
  }
}
```



自分の好きな位置に変更しよう

実行して確認してみよう！

21

パラメータの変更

- パラメータを変えるとゲームのバランスが調整できます
- プレイヤーが弾を撃つ頻度を変える
 - Playerのプログラムを一部変更

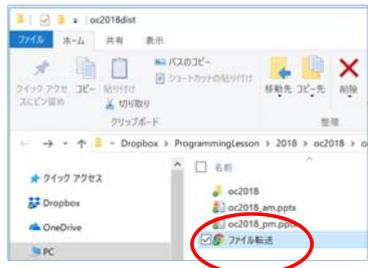
```
if (this.count==30){
  // 30の値を多くすると撃つ頻度が下がり、少なくすると頻度が上がる
  // 1の値を多くすると撃つ頻度が上がる
}
```

- 敵が弾を撃つ頻度を変える
 - Enemyのプログラムを一部変更

22

スマートフォンで遊んでみよう！(1)

- デスクトップの「オープンキャンパス」フォルダを開き、「ファイル転送」を開く

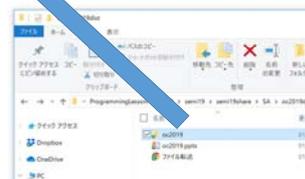


23

スマートフォンで遊んでみよう！(2)

- oc2019のフォルダをブラウザヘドラッグ&ドロップ

ここにoc2019フォルダをドラッグ&ドロップしてください。



24

スマートフォンで遊んでみよう！(3)

- ・「アップロード完了」が表示されたら、スマートフォンでQRコードを読み取ってみよう

アップロード完了

ゲームのURLとQRコード

<http://cho.is.meisei-u.ac.jp/pc/g/301>



25

最後に

- ・USBメモリにファイルを持ち帰ってください
 - ・USBメモリをPCに接続
 - ・デスクトップの「オープンキャンパス」を開く
 - ・全ファイルを選択し、USBメモリのフォルダにコピー
- ・今日作成したプログラムは、次のWebブラウザで動作します 必ず最新のバージョンをお使い下さい
 - ・Google Chrome
 - ・Firefox
- ・詳しい解説は次のページでも見られます
<http://cho.is.meisei-u.ac.jp/profile/>

26