

# 情報学部 プログラミング体験教室 (中級編)

長俣也・丸山一貴  
12:30 ~ 14:00

1

## ファイルを開こう

- ・デスクトップのフォルダoc2017を開く
- ・main2.js を開く(エディタが起動)

```
1 //これをコピーして使しましょう!
2 class Template extends Actor {
3     *main() {
4     }
5 }
6
7 //ここまで!
8
9
10
11 setMain(Start)::
12 [EOF]
```

2

## Startクラスを作ろう!

- ・クラス = プログラムの部品
- ・まずはゲーム全体をコントロールする「Start」クラスを作成
- ・class Template { .... } をコピー

```
1 //これをコピーして使しましょう!
2 class Template extends Actor {
3     *main() {
4     }
5 }
6
7 //ここまで!
8
9
10
11 setMain(Start)::
12 [EOF]
```

選択  
Shift + 矢印キー

コピー  
Ctrl + C

3

## Startクラスを作ろう(2)

- ・class Template { .... } の下に貼り付け
- ・Template をStart に書き換え

```
//これをコピーして使しましょう!
class Template extends Actor {
    *main() {
    }
}
//ここまで!
class Start extends Actor {
    *main() {
    }
}
setMain(Start)::
[EOF]
```

貼り付け  
Ctrl + V

4

## プレイヤーを表示させよう!(1)

- ・次に「Player」クラスを作きましょう
- ・class Template { .... } をコピー&貼り付け
- ・Template をPlayer に書き換え

```
2 class Template extends Actor {
3     *main() {
4     }
5 }
6
7 //ここまで!
8
9
10
11 setMain(Start)::
12 [EOF]
```

```
13
14
15 class Player extends Actor {
16     *main() {
17     }
18 }
19
20
21
22 setMain(Start)::
```

5

## プレイヤーを表示させよう!(2)

- ・「Start」にプログラムを追加して、「Player」を画面に出しましょう
- ・\*main() { ... } の中にプログラムを記述

```
class Start extends Actor {
    *main() {
        new Player({ x:30, y:100, radius:10});
    }
}
```

- ・radius:10 → あたり判定の半径

間違っ  
てPlayer  
に書か  
ないよ  
うにし  
ましょ  
う!

- ・フォルダからindex2.htmlを開くとブラウザが起動し、「Player」が画面に出てくる

6

## プレイヤーに重力をつけよう！

- 「Player」にプログラムを追加して「Player」に重力をつけましょう
- 「Player」をクリックしてプログラムを変更

```
class Player extends BodyActor {
  *main() {
    }
}
```

間違ってもStartに書かないようにしましょう！

- BodyActor → 物理運動をするキャラクター
- ブラウザを再読み込みすると「Player」が画面下に落ちていく

7

## プレイヤーが落ちないようにしよう！(1)

- プレイヤーが途中で止まるよう「Block」クラスを作りましょう
- class Template { ... } をコピー&貼り付け
- TemplateをBlockに書き換え
- extends Actorをextends BodyActorに書き換え

```
class Block extends BodyActor {
  *main() {
    }
}
```

8

## プレイヤーが落ちないようにしよう！(2)

- 「Start」クラスにプログラムを追加して「Block」をゲームに登場させる

```
class Start extends Actor {
  *main() {
    new Player({ x:30, y:100, radius:10});
    new Block({ x:20, y:300, scaleX:5, scaleY:1, p:2});
  }
}
```

9

## プレイヤーが落ちないようにしよう！(3)

- 「Block」にプログラムを追加して「Block」を固定された物理オブジェクトにしましょう

```
class Block extends BodyActor {
  *main() {
    this.isStatic=true;
  }
}
```

- this.isStatic=true → 物体が固定されて運動しなくなる
- ブラウザを再読み込みすると「Block」上に「Player」が乗る

10

## プレイヤーを動かそう！

- 「Player」にプログラムを追加して「Player」を操作できるようにしましょう

- ブラウザを再読み込みすると、「Player」がキー操作で左右に動けるようになる

```
class Player extends BodyActor {
  *main() {
    while(true) {
      if(this.getkey("right")>0) {
        this.vx+=0.6;
      }
      if(this.getkey("left")>0) {
        this.vx-=0.6;
      }
      yield;
    }
  }
}
```

11

## ブロックを増やそう！

- 「Start」にプログラムを追加して「Block」を増やしましょう

```
class Start extends Actor {
  *main() {
    new Player({ x:30, y:100, radius:10});
    new Block({ x:20, y:300, scaleX:5, scaleY:1, p:2});
    new Block({ x:300, y:200, scaleX:5, scaleY:1, p:2});
  }
}
```

- ブラウザを再読み込みすると「Block」が一枚増える

12

## プレイヤーをジャンプさせよう！

- 「Player」にプログラムを追加して「Player」がジャンプできるようにしましょう

```
class Player extends BodyActor {
  *main() {
    while(true){
      if(this.getKey("right")>0 {
        this.vx+=0.6;
      }
      if(this.getKey("left")>0 {
        this.vx-=0.6;
      }
      if(this.getKey("space")==1 && this.contactTo(Block)){
        this.vy=-10;
      }
    }
    yield;
  }
}
```

- ブラウザを再読み込みするとスペースキーで「Player」がジャンプ

13

## ブロックをもっと増やそう！

- 「Start」にプログラムを追加して「Block」を増やしましょう

```
// これ以前のプログラムは省略
this.y=200;
for (var i=0 ; i<10 ; i++) {
  new Block({x:this.rnd(500) , y:this.y,
    scaleX:5, scaleY:1, p:2,
    rotation:this.rnd(20)-10});
  this.y-=100;
}
```

- ブラウザを再読み込みすると「Block」が増える？

14

## 画面をスクロールさせよう！

- 「Player」にプログラムを追加して画面が「Player」についてくるようにしましょう

```
while(true){
  if(this.getKey("right")>0 || window.$rightB.clicked) {
    this.vx+=0.6;
  }
  if(this.getKey("left")>0 || window.$leftB.clicked) {
    this.vx-=0.6;
  }
  if( (this.getKey("space")==1 || window.$jumpB.clicked) && this.contactTo(Block)){
    this.vy=-10;
  }
  $Screen.scrollTo(this.x-$screenWidth/2,
    this.y-$screenHeight/2);
  yield;
}
```

- ブラウザを再読み込みすると「Player」が画面の中心になる

15

## スマホで動かせるようにしよう！(1)

- 今のままではスマホで動かすことができない…
- ボタンをゲームに登場させる
  - Startに追加

```
class Start extends Actor {
  *main() {
    window.$leftB=new Button({left:10, top:400, width:80, text:"Left"});
    window.$rightB=new Button({left:100, top:400, width:80, text:"Right"});
    window.$jumpB=new Button({left:300, top:400, width:80, text:"Jump"});
  }
}
```

(これより下は省略)

- 実行すると画面にボタンが出てくる

16

## スマホで動かせるようにしよう！(2)

- 今のままではプレイヤーがボタンに反応しない
  - Playerのコードを一部追加

```
while(true){
  if(this.getKey("right")>0 || window.$rightB.clicked) {
    this.vx+=0.6;
  }
  if(this.getKey("left")>0 || window.$leftB.clicked) {
    this.vx-=0.6;
  }
  if( (this.getKey("space")==1 || window.$jumpB.clicked) && this.contactTo(Block)){
    this.vy=-10;
  }
  $Screen.scrollTo(this.x-$screenWidth/2, this.y-$screenHeight/2);
  yield;
}
```

- 実行するとボタンでも操作できるようになる

17

## スマートフォンで動作させてみよう！

- 午前(初級編)と同じように行なう
- すでに自分のフォルダを作ってある場合は、そこに上書きする
- スマートフォンで、午前に行ったゲームにアクセスする
- アドレスバーの最後に index2.html を追加

<http://cho.is.meisei-u.ac.jp/oc/2017/cho/index2.html>

18