

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
13:00 ~ 14:30

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いましょう↓
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いましょう↓
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

```
1 //これをコピーして使いましょう↓
2 class Template extends Actor {↓
3     *main() {↓
4         ↓
5     }↓
6 }↓
7 //ここまで↓
8 ↓
9 class Start extends Actor {↓
10     *main() {↓
11         ↓
12     }↓
13 }↓
14 ↓
15 ↓
16 setMain(Start);↓
17 [EOF]
```

貼り付け
Ctrl + V

4

プレイヤーを表示させよう!(1)

- ・次に「Player」クラスを作らしよう
- ・class Template { } をコピー&貼り付け
- ・TemplateをPlayerに書き換え

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

```
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
- フ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 Player { ... } をコピー&貼り付け
- PlayerをBlock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,width:100,height:20,fillStyle:"white"});
    }
}
```

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
- フ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！(1)

- ボタンをゲームに登場させる
- Startに追加

```
class Start extends Actor {
    *main() {
        window.LB=new Button({left:10,top:400,width:80,text:"L"});
        window.RB=new Button({left:100,top:400,width:80,text:"R"});
    }
}
```

(これより下は省略)

- 実行すると画面にボタンが出てくる
- ボタンを押してみると…?

11

プレイヤーを動かそう！(2)

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

```
class Player extends BodyActor {
    *main() {
        while(true){
            if(window.RB.clicked>0) {
                this.vx+=0.6;
            }
            if(window.LB.clicked>0) {
                this.vx-=0.6;
            }
            yield;
        }
    }
}
```

- フラウザを再読み込みすると、「Player」がボタン操作で左右に動けるようになる

12

プレイヤーを動かそう！(3)

- PCではキーボードでも操作できたほうが便利
- 先ほど作ったボタンにkey: "キーの名前" という属性を追加

```
window.LB=new Button({left:10,top:400,width:80,text:"L",key:"left"});
```

右のボタン (RB)も同様に追加しよう

主なキーの名前
"Left" "Right" "Up" "Down" : 矢印キー
"Space": スペースキー
"shift": シフトキー
"enter": エンターキー
"A" から "Z" 英字キー
"0" から "9" 数字キー

13

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() {  
    //中略  
    new Block({x:20,y:300,width:100,height:20,fillStyle:"white"});  
    new Block({x:300,y:200,width:100,height:20,fillStyle:"white"});  
  }  
}
```

- ブラウザを再読み込みすると「Block」が一枚増える

14

プレイヤーをジャンプさせよう！(1)

- 「Start」にプログラムを追加して「Player」がジャンプするためのボタンを作しましょう

```
class Start extends Actor {  
  *main() {  
    window.LB=new Button({left:10,top:400,width:80,text:"L"});  
    window.RB=new Button({left:100,top:400,width:80,text:"R"});  
    window.JB=new Button({left:200,top:400,width:80,text:"Jump",key:"space"});  
  }  
}
```

(これより下は省略)

15

プレイヤーをジャンプさせよう！(2)

- 「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(window.JB.clicked==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

16

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,  
    width:100,height:20,  
    fillStyle:"white", 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？

17

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(window.RB.clicked>0) {  
    this.vx+=0.6;  
  }  
  if(window.LB.clicked>0) {  
    this.vx-=0.6;  
  }  
  if(window.JB.clicked==1 && this.contactTo(Block)) {  
    this.vy=-10;  
  }  
  $Screen.scrollTo(this.x,this.y);  
  yield;  
}
```

- ブラウザを再読み込みすると「Player」が画面の中心になる

18

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、そこに書きする
- ・スマートフォンで、午前に作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/oc2018/d/0/index2.html>

19

タイトルを作ろう!

- ・「Title」クラスを作りましょう
- ・一番下の setMain(Start); を setMain(Title);に

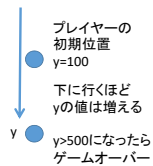
```
class Title extends Actor {
  *main() {
    new Label({text:"ジャンプゲーム",x:230,y:230,size:50});
    var b=new Button({top:300,text:"Start"});
    while(true) {
      if (b.clicked==1) this.loadPage(Start);
      yield;
    }
  }
}
//一番下
setMain(Title);
```

20

ゲームオーバーを作ろう!

- ・Playerの処理に追加

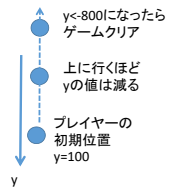
```
while(true){
  if(window_RB.clicked=0) {
    this.vx=0.6;
  }
  if(window_LB.clicked=0) {
    this.vx=-0.6;
  }
  if(window_SB.clicked==1 && this.contactTo(Block)) {
    this.vy=-10;
  }
  $Screen.scrollTo(this.x,this.y);
  if (this.y>500) {
    new Label({text:"ゲームオーバー",x:230,y:230,size:50});
    var b=new Button({top:300,text:"Title"});
    while(true) {
      if (b.clicked==1) this.loadPage(Title);
      yield;
    }
  }
  yield;
}
```



21

クリアを作ろう!

- ・「ゲームオーバーを作ろう!」と同様の文をPlayerの処理に追加
- ・クリアの判定条件はy<-800 に
- ・表示されるメッセージを変更



22

敵を作ろう!

- ・Enemyクラスを追加

```
class Enemy extends BodyActor {
  *main() {
    while(true){
      if(this.y>500) this.die();
      yield;
    }
  }
}
```

- ・Playerの処理に敵を出現させる処理を追加

```
while(true){
  //中略
  if (this.rnd(100)<5) {
    new Enemy({x:this.rnd(400),y:this.y-300,p:10});
  }
  yield;
}
```

23

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/>

24