



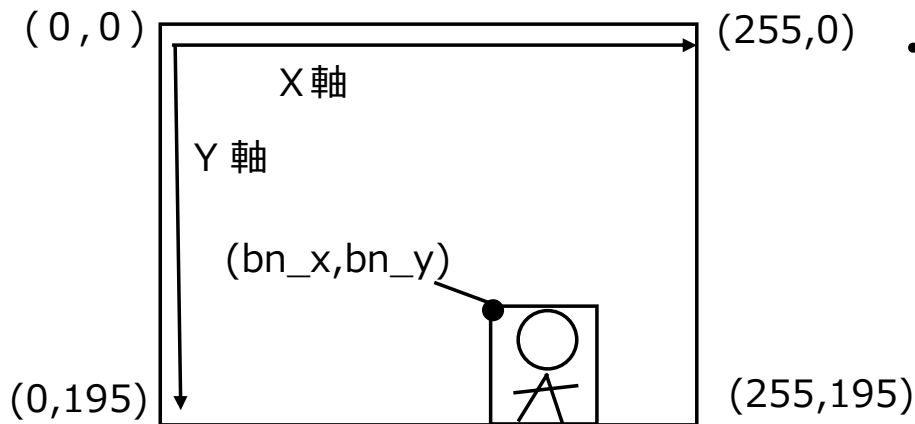
# GAME-MASTER MISSION\_3

アニメでゲーム作りを学ぼう(3/3)

## ホワイトボード

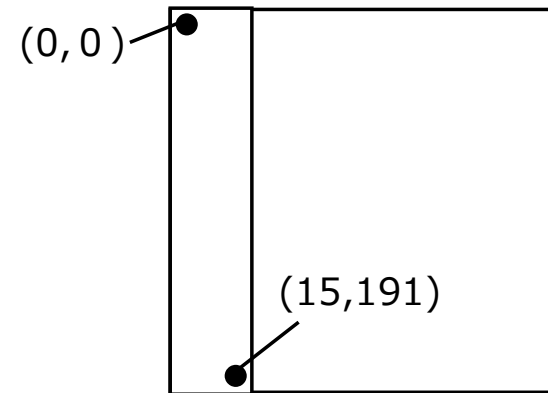
### ゲーム開発マスター MISSION 3

#### ●ピクセルマン・アニメの座標



### ピクセルマン the アニメを作る(3/3)

・ビット絵の座標(バンクは1)



#### ●背景の作り方(背景のクラス)

```
class Haikei :  
    def __init__(self) :  
        ~  
    def update(self) :  
        ~  
    def draw(self) :  
        ~
```

←クラス(設計図)「Haikei」の定義がはじまる  
←ここに初期処理を書く  
←ここに背景を張り付ける座標の計算を書く  
←ここに背景の描画を書く

このクラス(設計図)は、Animeの中で実体化、描画し、Pixelmanで座標を計算する！

## ■ 時間割

### MISSION\_2

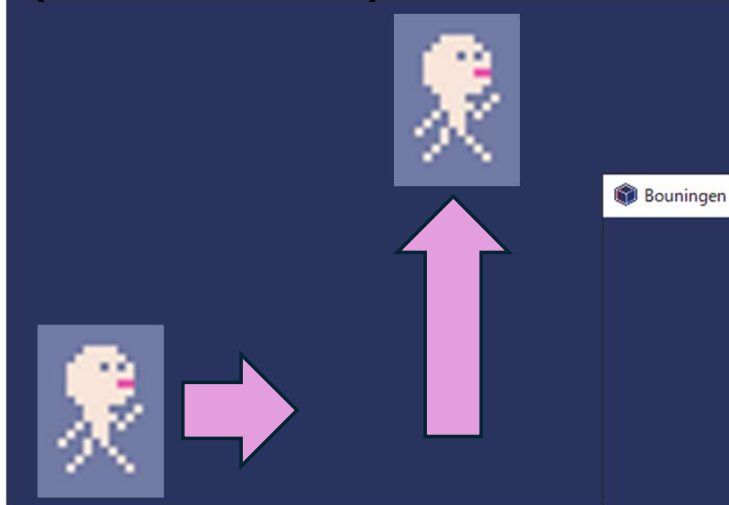
寿司打	5分
動く背景の書き方説明	90分
オリジナル改造	80分
寿司打	5分

合計	180分
----	------

# アニメでゲーム作りを学ぶ

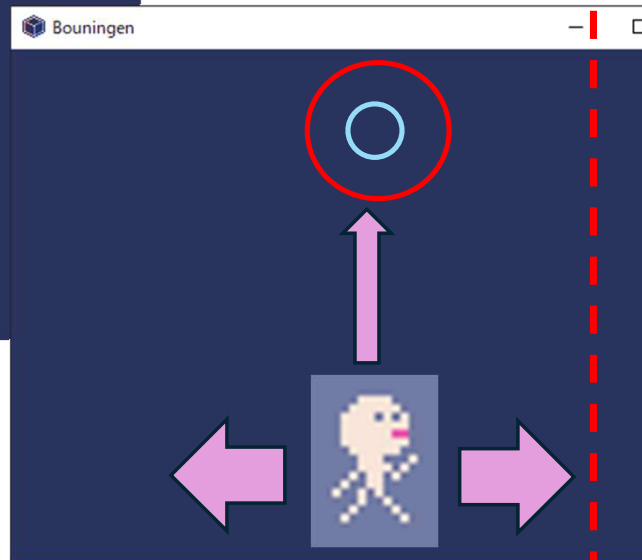
前々回

(MISSION\_1)



ピ°ケルマン、走る、ジャンプ

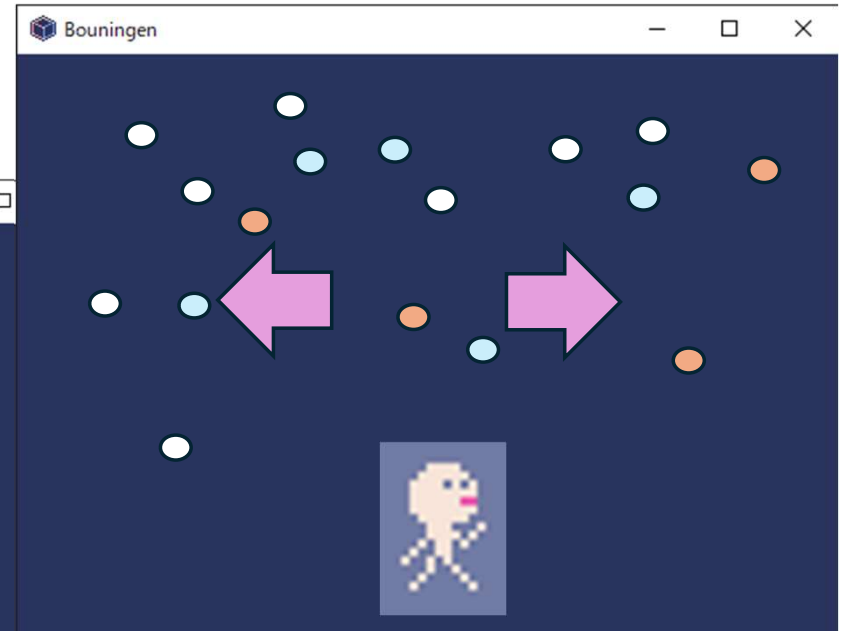
前回(MISSION\_2)



ピ°ケルマン、左右に走る、花火を上げる

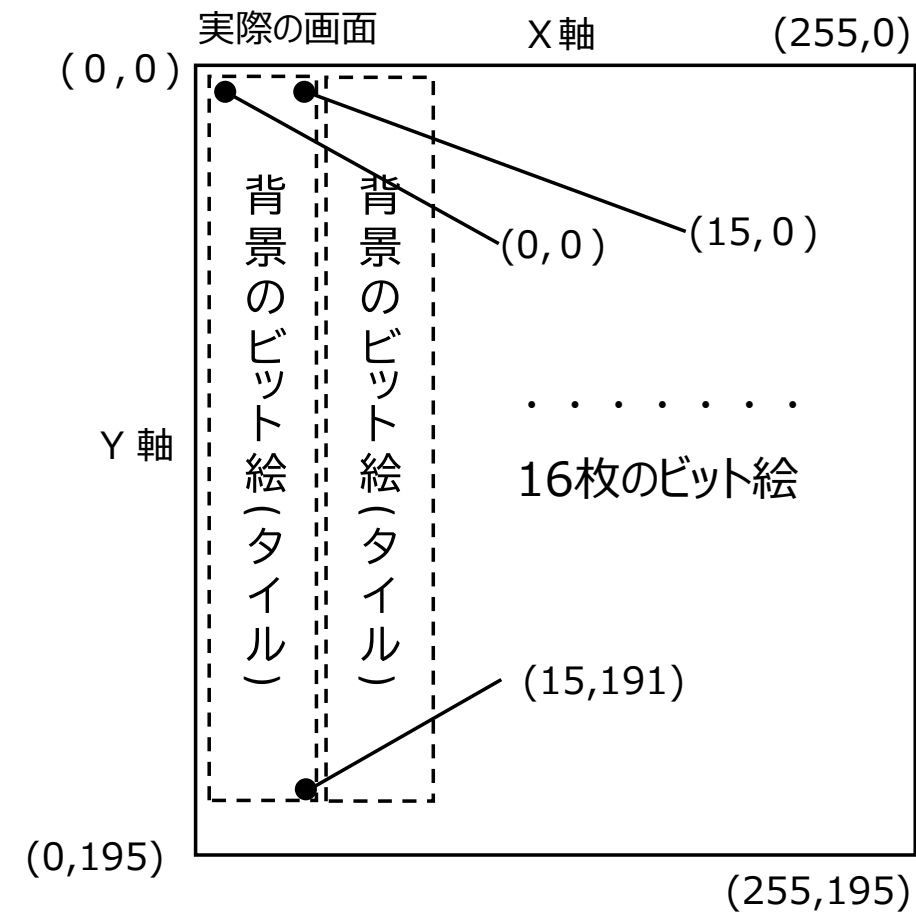
今回(MISSION\_3) (完成)

ピ°ケルマン、走って、ジャンプ、花火を上げ、背景を動かして完成させよう！



# ピクセルマン・アニメの動く背景を描く(1/5)

## ■ 動く背景の考え方 (1/2)



- 16×192ピクセルの背景のビット絵を作る (タイルと呼ぶ)

- タイルを16枚 (256÷16) 画面に貼る(描く)

それを動かすためには...

- ピクセルマンの左右の動きに合わせて、  
(ピクセルマンの位置に応じて、)  
タイルを貼る位置を変えていく

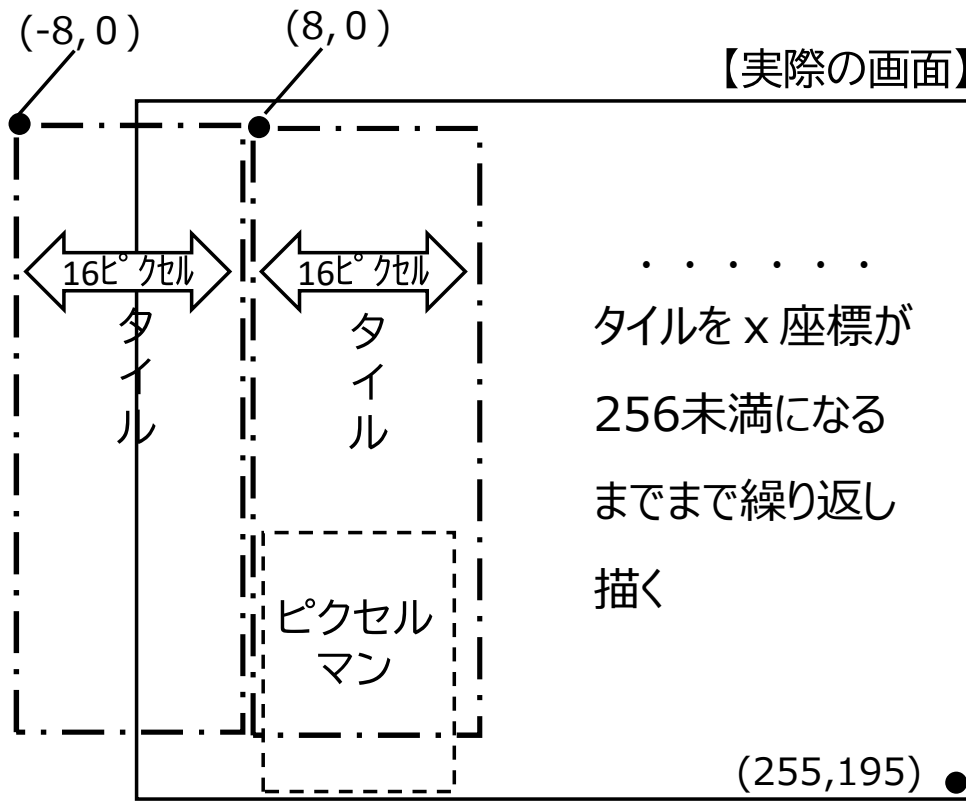
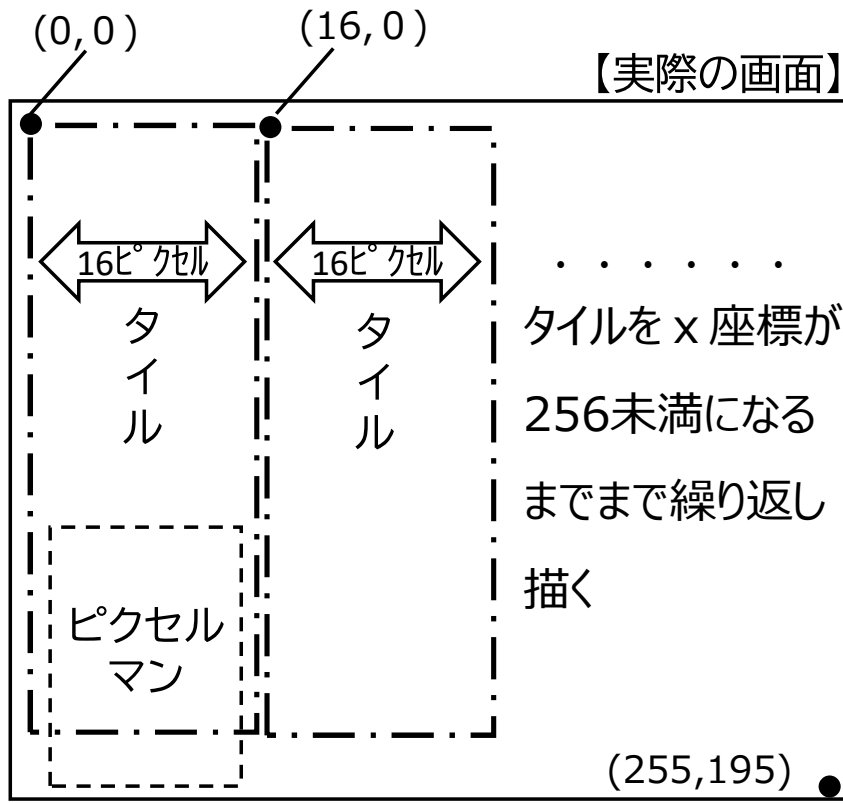
# ピクセルマン・アニメの動く背景を描く(2/7)

## ■ 動く背景の考え方 (2/3)

【ピクセルマンの位置とタイルを描き始める位置】

〔はじめ〕

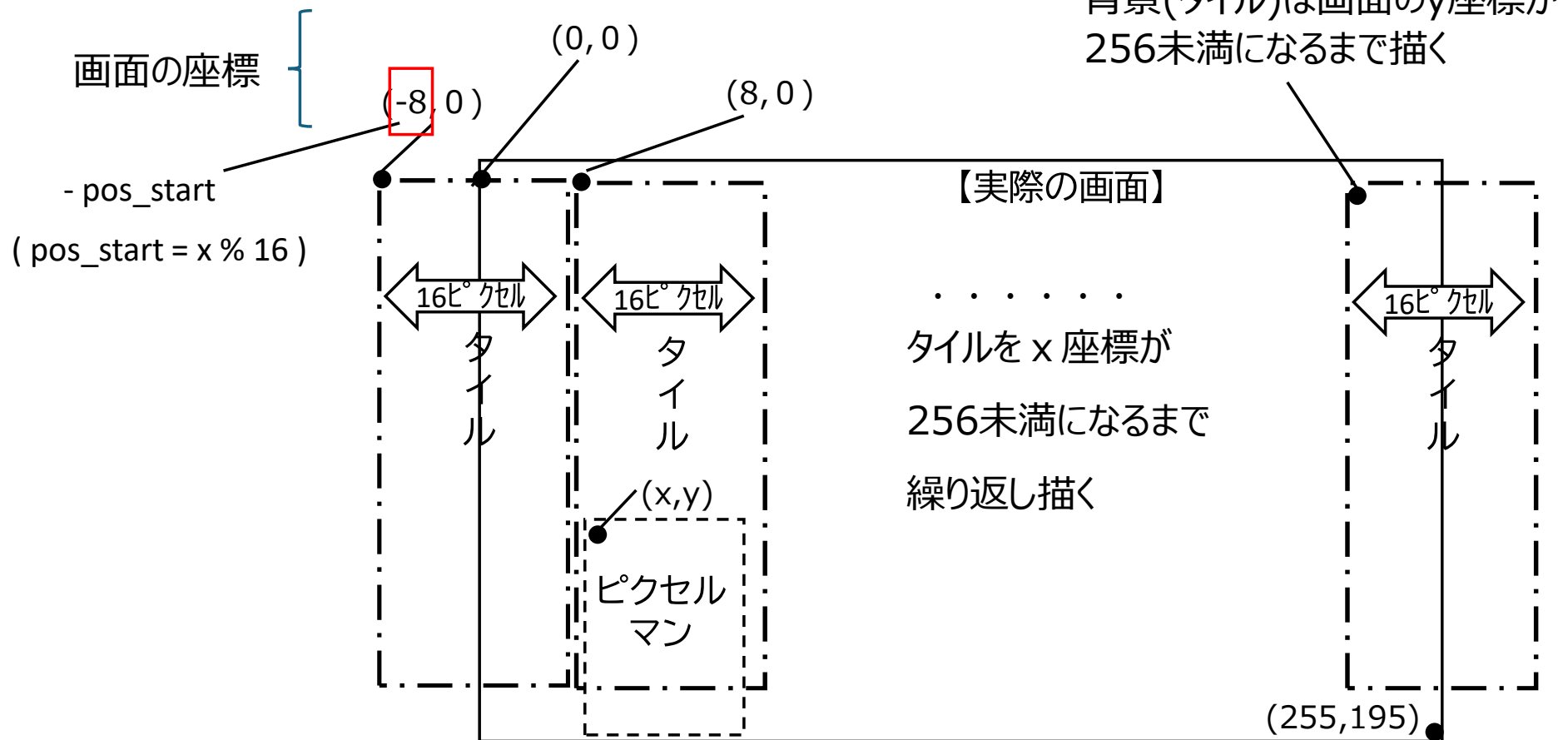
〔ピクセルマンが8ピクセル右に動くと背景は左に8ピクセル動かしたい〕



# ピクセルマン・アニメの動く背景を描く(3/7)

## ■ 動く背景の考え方 (3/3)

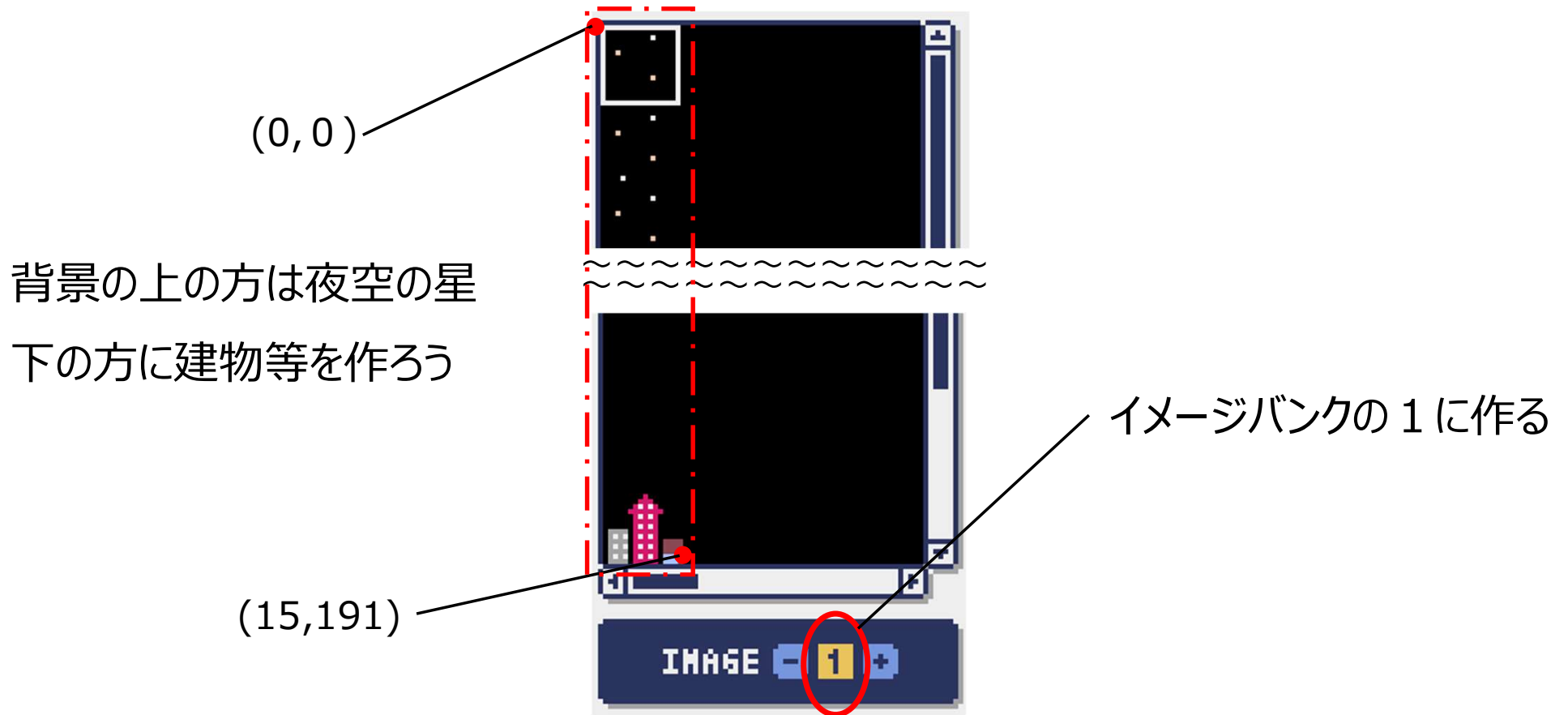
- 座標を式に変換すると (ピクセルマンは $x, y$ )



# ピクセルマン・アニメの動く背景を描く(4/7)

## ■ 背景のビット絵（タイル）を作ろう

- pyxel edit でイメージバンクの 1 の(0,0)~(15,191)に背景のタイルを作ろう





# ピクセルマン・アニメの動く背景を描く(5/7)

## • Haikeiのクラスを作ろう

プログラムを打ち込んで、動かしてみよう！

```
class Haikei :  
    def __init__(self, anime) :  
        self.anime = anime  
        self.anime.haikei = self  
        self.pos_start = 0  
    def update(self, anime, x, y) :  
        self.pos_start = x % 16  
    def draw(self) :  
        for i in range(-self.pos_start, Anime.SCREEN_WIDTH, 16) :  
            pixel.blit(i, 0, 1, 0, 15, 191, 0)  
            # ...実際の画面に背景(タイル)を描く座標  
            # ...イメージバンクは1を使う  
            # ...参照するビット絵の左上の座標  
            # ...参照するビット絵の右下の座標  
            # ...黒は透明色とする
```

...HaikeiクラスはAnimeクラスで実体化する

...背景(タイル)を描く位置を初期化

...背景(タイル)を描く位置をPixelmanの座標から計算する

# ピクセルマン・アニメの動く背景を描く(6/7)

- Haikeiのクラスを組み込もう (Animeクラス)

プログラムを打ち込んで、動かしてみよう！

```
class Anime :
```

```
    ~~~
```

```
    def __init__(self) :
```

```
        ~~~
```

```
        self.hanabis=[]
```

```
        Pixelman(self)
```

```
        Haikei(self)    ...Haikeiクラスを実体化する
```

```
        ~~~
```

```
    ~~~
```

```
def draw(self) :
```

```
    pixel.cls(1)
```

```
    self.haikei.draw()    ...Haikeiクラスのdrawメソッドを呼んで描く
```

```
    self.pixelman.draw()
```

```
    for hanabi in self.hanabis.copy() :
```

```
        hanabi.draw()
```

# ピクセルマン・アニメの動く背景を描く(7/7)

- Haikeiのクラスを組み込もう (Pixelmanクラス)

プログラムを打ち込んで、動かしてみよう！

```
class Pixelman :
```

```
    ~~~
```

```
    def update(self, anime) :
```

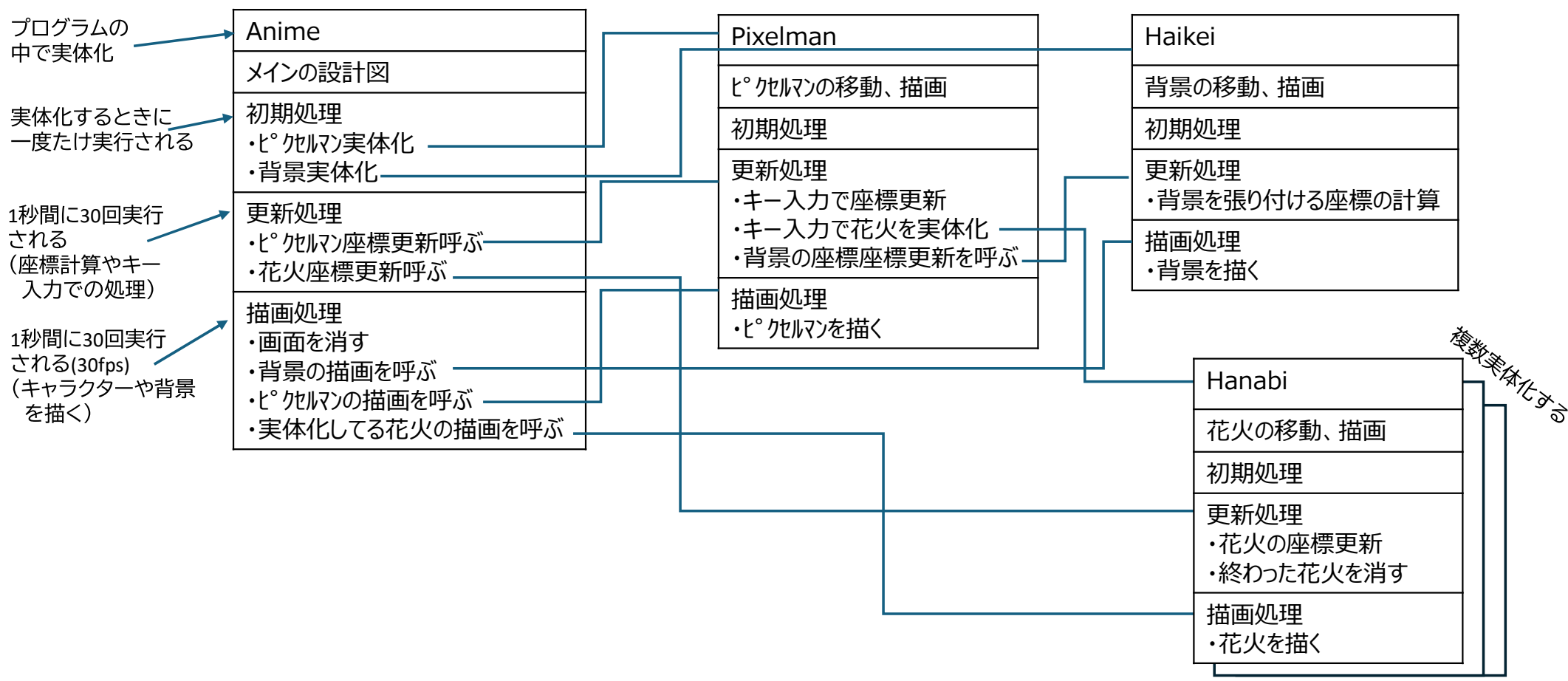
```
        ~~~
```

```
        self.anime.haikei.update(self.anime, self.bn_x, self.bn_y)
```

- ・・・ピクセルマンの現在の座標を引数にして、背景を描く座標を計算する  
Haikeiクラスのupdateメソッドを呼ぶ

# ピクセルマン・アニメのまとめ

・今回作ったアニメのプログラムにおける、クラス（設計図）の相関図



# ピクセルマン・アニメを改造してみよう(1/3)

---

■ 次のような改造を行ってみよう！ どこを改造すればいいか、わかるかな？

- ピクセルマンの移動速度を変えてみよう

- ヒント！ クラス:Pixelmanの初期処理をみてみよう！

- ピクセルマンのジャンプする高さを変えてみよう

- ヒント！ クラス:Pixelmanの初期処理をみてみよう！

- ピクセルマンが進みながらジャンプするように変えてみよう

- ヒント！ クラス:Pixelmanの更新処理をみてみよう！

ピクセルマンがジャンプする直前の移動方向を変数で管理

# ピクセルマン・アニメを改造してみよう(2/3)

---

- 花火が咲く高さを変えてみよう
  - ヒント！ クラス:Hanabiの更新処理をみてみよう！  
変数hana\_timやhana\_cntがフレーム数だよ
- 花火が咲くときの形を変えてみよう
  - ヒント！ クラス:Hanabiの描画処理だね、パターンで書きか方を変えてるだけだよ！

# ピクセルマン・アニメを改造してみよう(3/3)

---

- 背景が動く速さを変えてみよう

- ヒント！ クラス:Haikaiの更新処理をみてみよう！

- ピクセルマンの位置（変数bn\_x）と張り付ける位置（変数pos\_start）が鍵だよ

- 遠くはゆっくり、手前は早く動くように改造しよう

- ヒント！ クラス:Haikaiの更新処理で背景のタイルを張り付ける位置を2つ管理して、張り付ける背景を遠くと手前で変えてみるのが簡単かな

- その他、やってみたい改造があれば、先生に聞いて挑戦してみよう！