

プログラム勉強会

- 移動について
- 入力について

まず、移動について説明します。

ゲームに於いて画像が最低限必要なのはおわかりですね？

その画像を描画するのは別項で説明されるので、その通りです。

しかし、ただ画像を描画するだけではゲームにはなりません、ちゃんと動かないといけません。

ではどうやったら画像を動かせるのでしょうか？

考え方は単純です、画像を描画する座標を出発点から目的地まで少しずつずらしていただくだけです。

具体的には画像の座標を(0,0)→(0,1)→(0,2)→(0,3)→……→(0,60)

のように変化させていきます

ちなみに、画面上では一番左上を(0,0)とし右に行けば X 座標が増えて、下に行けば Y 座標が増えていきます。

パラパラ漫画だって少しずつ変化していく絵を連続で見ているだけなのに動いているように見えますよね？それと同じことです。

ではプログラムでどうやれば動かせるか実際に見てみましょう。

まずは自動で動かしてみます

```
for(y = 0; y < 100; y++){  
  (y 座標を y とする描画)  
}
```

こうすると画像が上から下に降りてくるように見えます。

もちろんループの条件はそれぞれ都合良く変えて問題ありません。

無限ループにすればいつまでも下まで降りていきます。

降りる速さを速くしたいときは

y++(y+=1 と同義)のところを **y+=2** と変えると倍の速さになります。

このとき、例えば画面サイズの Y 座標の上限が 480 としたとき。

画像の座標がそれを上回ってしまったとしたら、画面内に画像は描画されませんがプログラム内では 480 を上回った座標に画像を描画し続けています。

これで画像を自動で動かすことが出来ました

しかしこれだけでもまだゲームとは呼べません

やっぱりゲームだったら自分で動かさなきゃいけませんよね

そこで**入力**が必要になってくるわけです。

如何なるゲームにおいても、入力がなければゲームとして成立しません(例外は認めません)

入力はキーボード、マウスやコントローラーによって行われます。

平たく言えば**ボタンを押す**、ということになります。

入力することによって移動、攻撃、ジャンプや防御、様々な行動がゲーム内で起こります。

また入力はただ押しただけではなく、押されている状態、押した瞬間、離された瞬間など、色々判定の種類もあります。

入力を検出するためには、入力機器から入力信号を受け取る必要があります。

さて、ではどうやってプログラムで入力して移動させるかを見てみましょう。
基本的に `if` 文を使って判断します。

```
if(右方向が入力されたら){
```

```
  x++;
```

```
}
```

```
  描画
```

こうすれば描画される座標が **x** に設定されている画像は右に移動します。

もちろん右だけじゃ物足りない、四方向全部作りましょう。

これを無限ループ内で記述することによって、ループでこの処理のところまで回ってくれば現在の状態がどちらかを判断して内容を実行します。

移動だけではなく

```
if(入力されていたら){
```

```
  描画
```

```
}
```

という風にすればボタンを押している間だけ画像が描画されます。

その他の細かい行動、攻撃やジャンプ等を使いたい時もこの `if` 文の中に記述します。