

ゲーム制作勉強会

移動、入力について

目次

移動

1. 画像の動かし方
2. 数学とプログラム上での座標の違い
3. 画像を描画するときの注意点

入力

1. 自分で画像を動かす
2. 攻撃、ジャンプなどの動きのやり方

移動について

ゲームを作る上で画像は必要不可欠なものです。

しかし、画像を描画するだけではゲームとして成り立ちません。

画像が動かないとゲームとは言えません。

では、どのようにすれば画像を動かせるのでしょうか？

動かし方はいたって単純です。

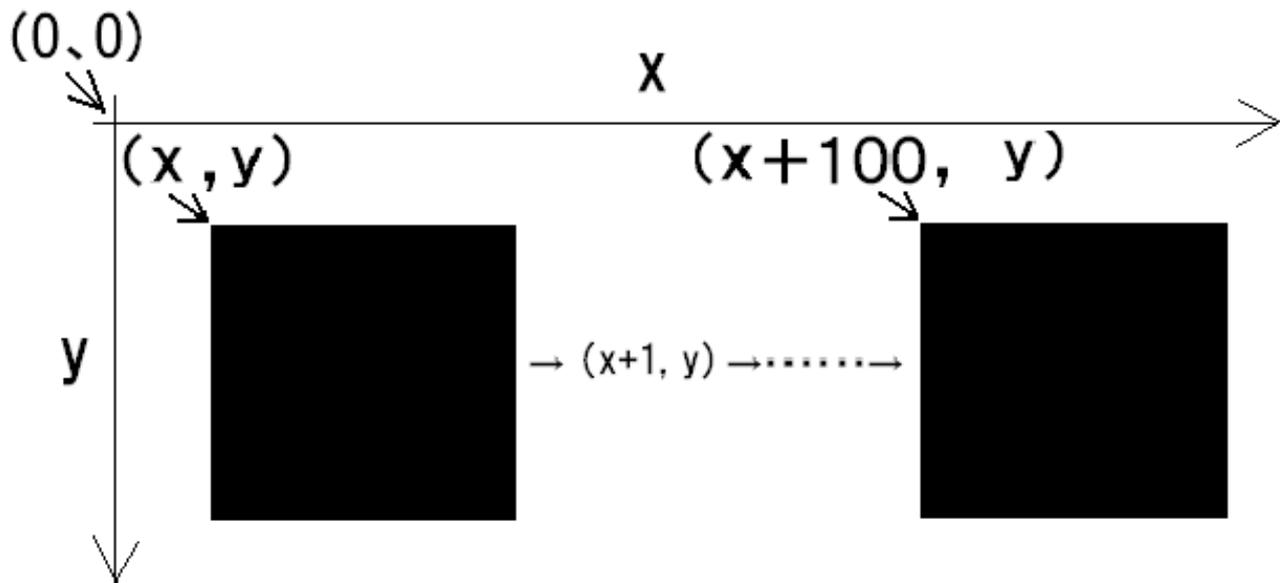
$(0, 0) \rightarrow (1, 0) \rightarrow (2, 0) \rightarrow \dots \rightarrow (100, 0)$

上記のように画像の x 座標をずらしていきだけで簡単に画像を動かすことができます。

これをプログラムで表すと次のようになります。

```
for(i=0; i < 100; i++) //100回ループする。
{
    x = x + 1;
    //(x, y)の位置に描画する処理
}
```

では実際に見ていきましょう。



画像の左上の (x, y) 座標の位置が描画するときの基準点になります。

この状態で x を増やすと画像は右に動き、 y を増やすと画像は下に動きます。

逆に画像を左に行かせたい場合は x を減らせばいいわけです。

(" $x = x+1$ " のところを " $x = x-1$ " にする)

また、移動する速さを変えたいときには、ずらす量を大きくすれば移動が速くなります。

描画の注意点

当然のことですが画面外に画像を描画しても画面には何も映りません。

しかし実際にプログラム上では画像を描画する処理は行われてしまい、そういった描画する処理をたくさん行くと処理落ちが起こってしまいます。

なので、画像が画面外に描画されてしまうときは描画されないようにプログラムを作るようにしましょう。

これをプログラムで表すと次のようになります。

```
if(もし画面の中に画像があるとき)
{
    //( x , y)の位置に描画する処理
}
```

入力について

入力はキーボード、マウス、コントローラーなどで行います。

簡単に言えばボタンを押したかどうかで判断します。

ボタンが押された場合、そのボタンに応じて攻撃、防御、走る、歩くなどの動作がゲーム内で可能となってきます。

プログラムでは主に “if” 文を使います。

では、どのようにプログラムを書くのか実際に書いてみましょう。

```
if(もし右が押されたら)
{
    x = x + 5
    //( x, y) の位置に描画をする処理
}
```

このようにプログラムを書けば右を押せば画像が右に動きます。もちろん、右だけではゲームとして成り立たないので4方向作りましょう。

そして、これをゲーム中の無限ループに記述することによって、この処理の位置に来た時に入力の有無を判断して動作させることができます。

もちろん、移動だけでなく攻撃などの処理も行います。

```
if(もし特定のボタンが押されたら)
{
    //弾を出す処理
}
```

という風を書いてこれを無限ループに入れるとボタンを押している間画像が描画されます。攻撃やジャンプなどの動作も if の中に記述します。

移動、入力のまとめ

1. 画像は座標をずらすと動いているように見える
2. プログラム上の座標は数学と違っている
3. 画面左上が (0, 0)
4. 画面外には描画しない
5. 入力は if 文を用いることによって色々なことができる