

第7回 画像のスクロール

長かったゼミBも今回で最後です。プログラムもゲームとしての骨格は大体出来上がってきたと思います。最後の仕上げに、様々なゲームで必須といえる、相対座標、絶対座標、スクロールについて説明します。

○スクロールの仕組み

画面のスクロールをする、ということは、ゲームの世界全体の中の、画面に表示する部分を動かす、ということです。たとえば、RPGのフィールドマップ等では画面内にはその一部しか表示されません。フィールドマップ全体を描画しては処理が膨大になってしまいますから、その中でも画面に表示される部分だけ描画していきます。

今回用いるスクロールの方法は、マップを1フレーム毎に少しずつずらしていくという手法です。まず、以下のプログラム文をご覧ください。

```
int Game( void ){
    int i , j ;
    // 絶対座標の初期化
    abs_pos[ 0 ] = 0 ;
    abs_pos[ 1 ] = 0 ;

                                中略

    /*****
        ここまで描画
    *****/
    abs_pos[ 1 ]++ ;
}
//エラー発生時はゲーム終了処理
return 0;
}
```

`abs_pos[1]`を1つずつ加算することで、表示位置は下へと動いていき、物体は上へと流れていきます。ただし、まだ描画部分を変更していないので、現在の状態だとマップの当たり判定だけが上へと流れていきます。

○物体のスクロール

物体には、相対座標と絶対座標があります。マップ上の地点 **a** が、絶対座標であり、これはマップ左上を **(0,0)** とする座標系です。これに対し、画面上の地点 **b** は相対座標で、画面左上を **(0,0)** とする座標系です。なお、マスごとに区切った座標系とは無関係ですので注意してください。

今まで、敵の座標、自機の座標、その他様々な座標には、相対座標を用いてきました。この相対座標を絶対座標に変換するならば、画面左上の絶対座標と、求めたい点の相対座標を足せばよいのです。今回のプログラムでは、画面のスクロールと共に、マップはスクロールします(相対座標が変化する)が、他はスクロールしません。マップの当たり判定についてはすでに計算済みですので、今回は **DrawMap** 関数を変更しながら、これについて学んでいきましょう。

今まで、**DrawMap** 関数は、画面左上の絶対座標、つまり `abs_pos` が **(0,0)** の場合の描画のみをしてきました。それが、**i**、**j**、**x**、**y** の初期値の意味です。これらに、きちんと画面左上の絶対座標を使用するようにプログラムを変更していきます。下記のように **DrawMap** 関数を書き直してください。

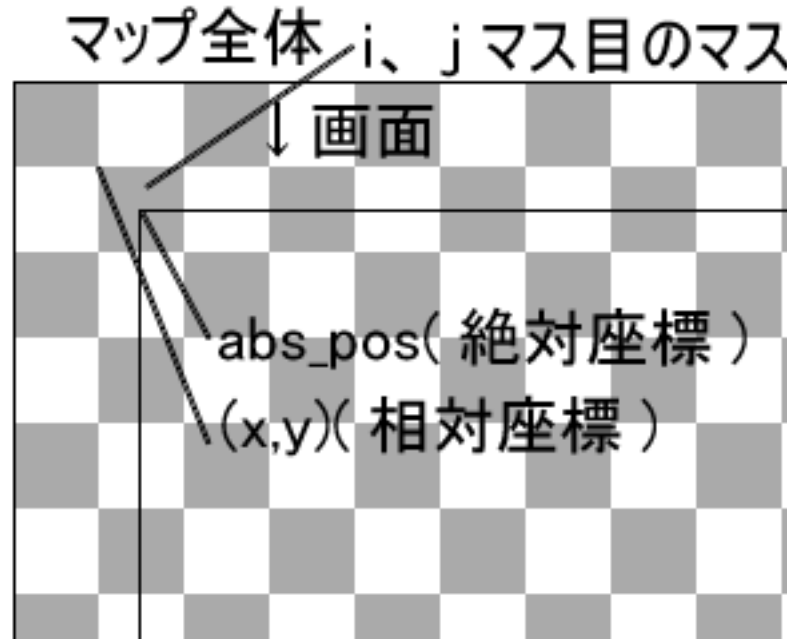
```
void DrawMap( void ){
    // i , j : マスのインデックス x , y : 画面上の座標
    int i , j , x , y ;
    for( i = abs_pos[ 0 ] / CHIP_WIDTH ,
        x = - ( abs_pos[ 0 ] % CHIP_WIDTH ) ;
        x < 640 && i < map.width ; i++ , x += CHIP_WIDTH ){
        for( j = abs_pos[ 1 ] / CHIP_HEIGHT ,
            y = - ( abs_pos[ 1 ] % CHIP_HEIGHT ) ;
            y < 480 && j < map.height ; j++ , y += CHIP_HEIGHT ){
            if( map.chipID[ i ][ j ] == 1 ){
```

```
// 任意のマス目の chipID が 1 の時描画
DrawGraph( x , y , img.chip , TRUE ) ;
    }
}
}
```

画面左上の絶対座標をチップの幅・高さでわれば、小数点切り捨てにより、画面左上のマップチップが左打から何マス目で上から何マス目かがわかります。そして、画面左上の絶対座標をチップの幅・高さで割ったものに、**-1**をかければ、そのチップの左上の相対座標がわかります。

あとは、描画の際に画面外にでないよう、**for** 文の脱出条件を設定してやれば OK です。DrawGraph 関数は、渡された座標を画像の左上として秒がするので、この座標が右画面外や下画面外にでたときに **for** 文を抜けてやればよいのです。これは、以前の DrawMap 関数でも同じですね。

以下は参考画像です。画面左上とマップ左上が近い場合の、左上の拡大図です。



画面内に入っている全てのマップチップを描画しなくてはならないので、描画範囲は少し画面外にはみ出ます。また、**i**、**j**、**x**、**y** はいずれも初期値です。

これでゼミ B に関する説明は全て終わりです。お疲れ様でした。これからは自分たちでゲームの製作をしていくことになります。判らないことも多々でてくるでしょうが、そういう時は迷わず私たちに質問してみたりしてください。出来る限り、力は貸すよう、私達も頑張ります。

○宿題

Programing Place Plus(http://www.geocities.jp/ky_webid/)の C 言語編の、言語学周辺を、第 0 章から順に読んでいってください。小規模なプログラムを組むのには、ゼミ A で使用した **BCC** コンパイラが有効です。**BCCDeveloper** というフリーソフトと併用してプログラムをしてみましょう。設定等は、検索すればすぐに表示されるかと思いますが、わからなければメールで聞くなり、掲示板で聞くなりしてください。

一日一章ずつ進め、38 章まで終了できればベストです。これは義務ではありませんが、プログラミングをする上でより自分のためになるはずです。もちろんこれもわからないことがあれば連絡をください(ただし、自ら調べ自ら考えることを怠らないこと)。