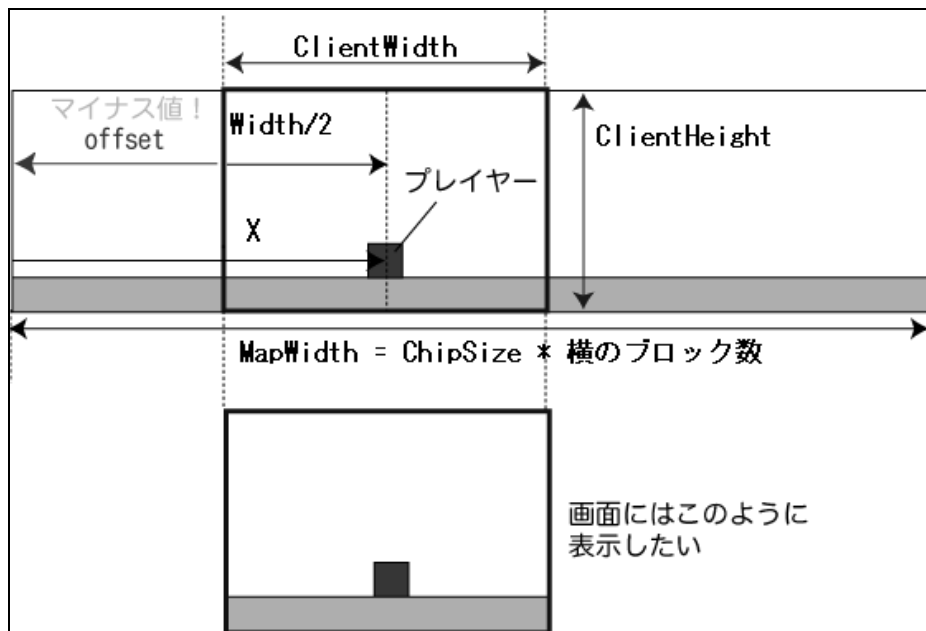


横スクロール

マップが大きくなったとき画面がスクロールするようにします。横スクロールも縦スクロールも基本は同じなので、ここでは横スクロールのやり方を紹介します。

オフセットを使ったスクロール



上の図がスクロールの基本です。

- プレイヤーの移動に伴って枠が移動する。
- プレイヤーは枠の真ん中に配置する。
- ただし、マップの端ではプレイヤーを真ん中に配置しない。

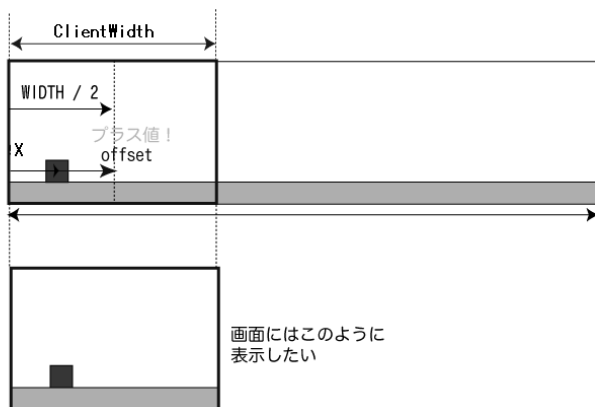
スクロールはオフセットを使うと簡単に実装できます。オフセットはマップ全体のうちスクリーンに表示する開始位置を表しています。オフセットを求める式はプレイヤーがマップ中央、左端、右端のどこにいるかに応じて使い分けます。

X方向のオフセットの計算

- マップ中央

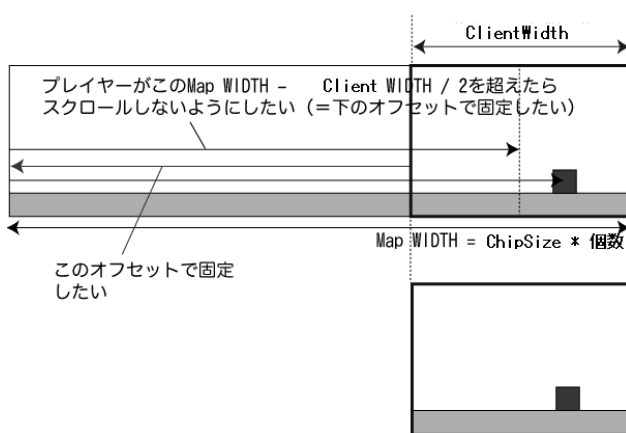
$$\text{offsetX} = X - \text{ClientWidth}/2;$$

- マップ左端



$$\text{offsetX} = 0;$$

- マップ右端



$$\text{offsetX} = \text{MapWidth} - \text{ClientWidth};$$

描画の最初と最後の位置の StartX と EndX はオフセットの値によって変化するのだから左端に行ったときは 0 に固定され、右端に行ったときは MapWidth - ClientWidth に固定されます。

今回は横スクロールしか解説してませんが、縦スクロールもほとんど同じです。Y 方向のオフセット値を同じように求めれば簡単にできます。

移動の処理の例

```
void __fastcall TForm1::Photo()
{
    ALL = Rect(0,0,ClientWidth,ClientHeight); //ウィンドウのサイズ

    Image1->Canvas->FillRect(ALL); //画面クリア
//マップデータ-----
    int MapDate[10][30] = {
        [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
        [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
        [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
        [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
        [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
        [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
        [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
        [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
        [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
        [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
        [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1],
    };
//移動処理-----
    //画面左端に来たとき
    if(X<ClientWidth/2){
        offsetX = 0;
        PRect = Bounds(X,Y,80,112);
    }
    //画面右端に来たとき
    else if(X>MapWidth-ClientWidth/2){
        offsetX = MapWidth - ClientWidth;
        PRect = Bounds(X-offsetX,Y,80,112);
    }
    //通常
    else {
        offsetX = X - ClientWidth/2;
        PRect = Bounds(ClientWidth/2,Y,80,112);
    }
//描画-----
    int StartX,EndX; //ウィンドウに描画する範囲の最初と最後
    StartX=offsetX/ChipSize-1;
    EndX=(offsetX+ClientWidth)/ChipSize+1;

    for(int i=StartX;i<EndX;i++){
        for(int j=0;j<10;j++){
            if(MapDate[j][i] == 1){
                BRect = Bounds(i*ChipSize-offsetX,j*ChipSize,ChipSize,ChipSize);
                Image1->Canvas->CopyMode = cmSrcCopy;
                Image1->Canvas->CopyRect(BRect,Image2->Canvas,Rect(0,0,ChipSize,ChipSize));
            }
        }
    }

    Image1->Canvas->CopyMode = cmSrcCopy;
    Image1->Canvas->CopyRect(PRect,Image3->Canvas,Rect(0,0,80,112));

    Form1->Canvas->CopyMode = cmSrcCopy;
    Form1->Canvas->CopyRect(ALL,Image1->Canvas,ALL);
}
```