ESライブラリ＆＆
ゲームプログラミング

○　荷物勇者編 - 第１１回　目標地点の設置　○

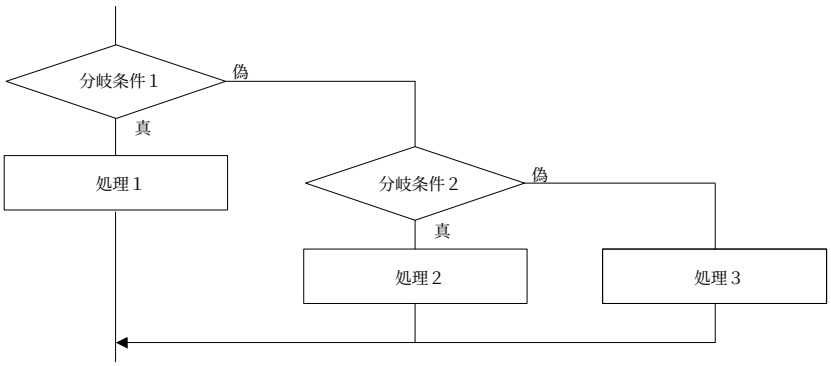
if-else-ifによる多分岐

if-else文を連ねて条件分岐を多重に行う場合、else-if文を使用することができます。たとえば、

```
if(条件式１) {
    条件式１を満たしたときに実行する処理
} else {
    if(条件式２) {
        条件式２を満たしたときに実行する処理
    } else {
        if(条件式３) {
            条件式３を満たしたときに実行する処理
        } else {
            条件式３を満たさないときに実行する処理
        }
    }
}
```

というようなif-else文は、以下のようにelse-if文で記述することができます。

```
if(条件式１) {
    条件式１を満たしたときに実行する処理
} else if(条件式２) {
    条件式２を満たしたときに実行する処理
} else if(条件式３) {
    条件式３を満たしたときに実行する処理
} else {
    条件式３を満たさないときに実行する処理
}
```



▲if-else-if

switch-caseによる多分岐

switch-case文は、変数の値によって実行する処理を分岐させることができます。switch-case文では、

変数の値が、
値１なら処理１を実行し、
値２なら処理２を実行し、
値３なら処理３を実行し、
値４なら処理４を実行し、
上記のどれでもなければ処理５を実行する

という分岐をさせることができます。switch-case文の書式は次のようになっています。

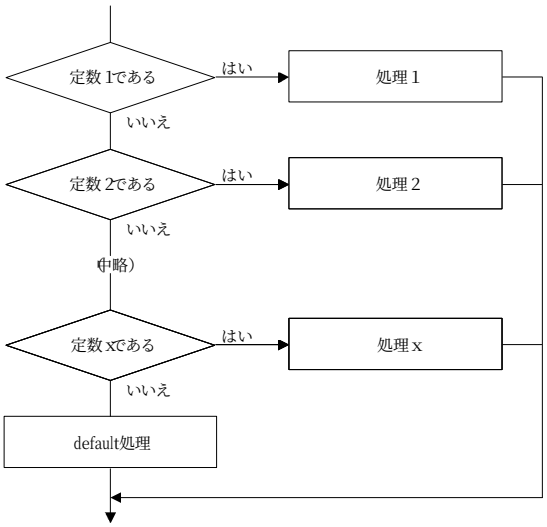
```
switch(変数) {
case 値１または定数式１:
     処理１
     break;

case 値２または定数式２:
     処理２
     break;

  (中略)

case 定数式ｘ:
     処理ｘ
     break;

default:
     すべてのcaseを満たさなかったときの処理
     break;
}
```



case文とdefault文の最後は":"（コロン）で終了します。case文の分岐条件に記述できるのは整数型の定数式のみです。default文は必要のないときは省略することができます。また、case文の最後にあるbreakは、switch文を抜けるために必要です。breakがないと、breakが見つかるまで、またはswitch文の最後まで処理が実行されます。

```
switch(value) {
case 1:
case 2:
case 3:
     処理１
     break;

case 4:
     処理２
     break;

default:
     処理３
     break;
}
```

このような場合は、変数valueの値が1、2、3の場合は処理１が、4の場合は処理２が、それ以外の場合は処理３が実行されます。

switch-case文では、整数型の値による分岐しかできません。小数型や文字列による分岐、複雑な条件で分岐先が多い場合は、if-else文を連ねて分岐させます。

課　　題

アイテムを運ぶ目標地点を設置しましょう。地形データは'G'とします。

(１)目標地点の画像は32×32ピクセルとします。以下のような画像を用意しましょう。
(１パターンでかまいません)

(２)目標地点のスプライトを管理するための変数の宣言をします。以下のプログラムを適切な場所に追加しましょう。

```
SPRITE    goalSpr;
```

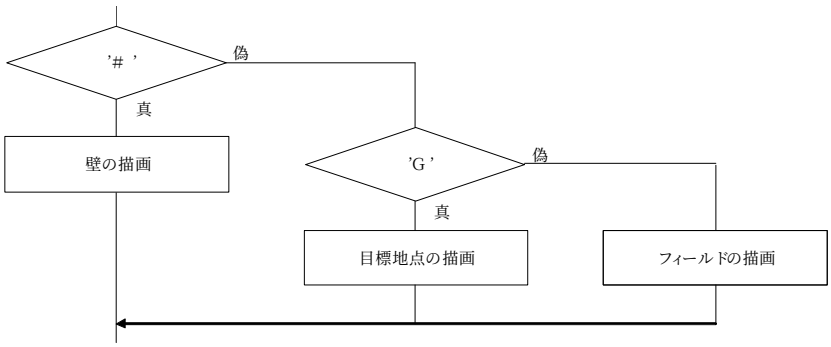
(３)目標地点の画像を読み込むプログラムを適切な場所に追加しましょう。

(４)以下のように地形データに目標地点を追加しましょう。

```
// マップデータ
mapData[0] = TEXT( "##########" );
mapData[1] = TEXT( "#p       #" );
mapData[2] = TEXT( "# ###### #" );
mapData[3] = TEXT( "#     G# #" );
mapData[4] = TEXT( "# #  # # #" );
mapData[5] = TEXT( "# #    # #" );
mapData[6] = TEXT( "# #G  G# #" );
mapData[7] = TEXT( "# ###### #" );
mapData[8] = TEXT( "#        #" );
mapData[9] = TEXT( "##########" );
```

(５)地形データが'G'だった場合、画面の適切な場所に目標地点の画像を表示するプログラムを作成しましょう(目標地点は通りぬけられるものとします)。

ヒント：



※壁以外の地形の場合は、フィールドを描画し、さらにその上にゴールなどの地形を「重ね合わせて」描画する方法もあります