

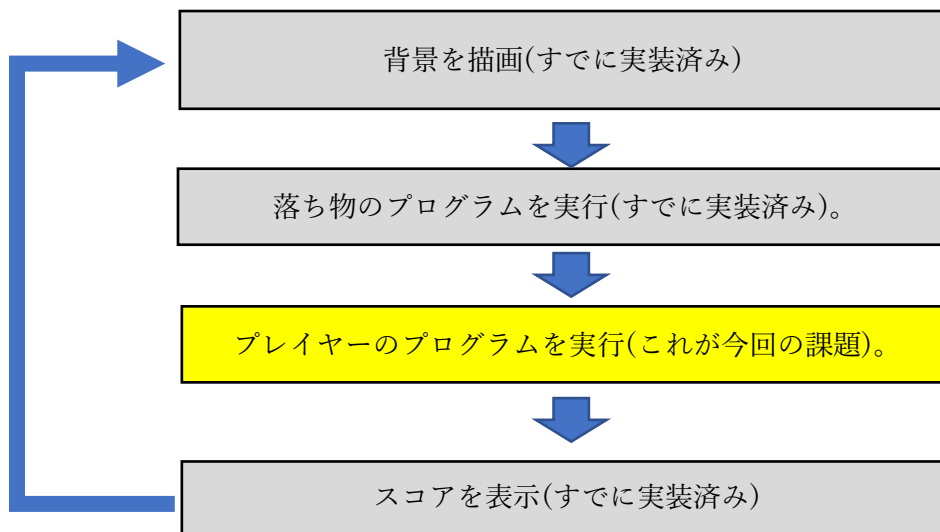
制作演習_課題 02

落ち物ゲームの作成。落ち物ゲーム.exe を起動してゲームをプレイして、動作を確認してください。今回の課題は、このゲームと同様の仕様のゲームを作成することです。

1.全体仕様の詳細

プログラムにはゲームループとなる、無限ループがあります。このループの中に落ち物のプログラムを記述してください。

ゲームのフローは下記のようになります。



2.プレイヤープログラムの仕様詳細

- ① キーボードの入力で、左右に移動できる。ただし、@マークは壁となるため、@マークにめり込んではいけない。

キーボードの入力判定は WindowsAPI の GetAsyncKeyState 関数を使用するように。

下記にサンプルコードを示す。

GetAsyncKeyState 関数の使用例

```
if (GetAsyncKeyState(VK_LEFT) != 0) {  
    //左のキーが押されたときの処理をここに記述する。  
}
```

- ② 落ち物とプレイヤーとの衝突判定を行い、落ち物をとれていたら、変数 `getCount` をインクリメントして、落ち物の死亡フラグを 1 に設定する。

落ち物は構造体 `Otimono` の可変長配列型の `otimonoArray` という変数を使うことでアクセスすることができる。

下記に `otimonoArray` を使用して、落ち物にアクセスするサンプルコードを記する。

落ち物へのアクセスのサンプルコード。

```
//otimonoArray は可変長配列として定義されている。otimonoArray.size() を使うことで、  
//可変長配列の要素数を取得することができる。  
for (int i = 0; i < otimonoArray.size(); i++) {  
    if (何かしらの条件文){  
        otimonoArray[i].dead = 1;    //これで死亡フラグを立てることができる。  
    }  
}
```

- ③ プレイヤーを描画する。プレイヤーは P という文字で描画してください。文字の描画には、`kbcDrawMoji` 関数を使用してください。`kbcDrawMoji` の使い方のサンプルコードを下記に示します。

`kbcDrawMoji` 関数の使用例

```
kbcDrawMoji(10, 5, 'P'); //x 座標が 10、y 座標が 5 の場所に P を表示。
```