**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。

背景を描画(すでに実装済み)

## **2.プレイヤープログラムの仕様詳細**

落ち物のプログラムを実行(すでに実装済み)。

スコアを表示(すでに実装済み)

プレイヤーのプログラムを実行(これが今回の課題)。

　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。 |