

# C#によるゲーム開発第1回

Amusement Creators

# そもそも個人でゲーム開発って可能？

- 可能
- 個人で開発して何万本も売って生活している人もいます。
- グループで作るのもお勧め

- 例 シューティングゲーム  
東方風神録

# プログラム言語の種類

- C、Java、Perl、Haskell、Prolog、ひまわり etc
- ひとつの開発に複数使うことも珍しくありません。
- それぞれに得意不得意があります。
- C、C++言語は難しいですがハードウェアの機能（画像描画等）を直接扱うことができます。
- Java、C#は簡単ですがC言語で開発されたプログラムを経由しないと画像描画等ができません。
- 工学システム学類ではC言語とJavaを学びます。
- このサークルでは初心者にはC#を教えています。

# C#って？

- Microsoftが開発した言語
- 簡単！
- 実はMacOSX、Linuxでも動きます。
- 標準の機能ではまともな描画を行えないのでゲームライブラリ、ゲームエンジンなどと呼ばれるものを使用しています。

# ゲームライブラリ、エンジンって？

- 描画、音楽再生等、ゲームを開発するために必要な機能をまとめた部品のようなもの
- 今回は教育用に開発した手軽で低機能なものを使用します。

# 開発開始！

- あなたの前には既に  
「Visual C# 2008」  
というソフトが起動しているはずです。

# ウィンドウの表示

```
class Program
{
    [System.STAThread]
    static void Main(string[] args)
    {
        Game.Initialize();

        while (Game.DoEvent() == true)
        {

        }

        Game.FinalizeGame();
    }
}
```

- F6を押す。  
もしくはメニューバーのデバッグ→デバッグの開始を押す。
- 黒い画面が表示されるはずです。

# 少し解説

- 解説されていない部分は魔法の呪文です。
- いずれわかるようになります。
- `Game.Initialize();`
  - ウィンドウを表示する命令
- `while (Game.DoEvent()==true)`
  - {}で囲まれた中身を×ボタンが押されるまで、何度も実行します。今回は何もしていないので画面は黒いままで。
- `Game.FinalizeGame();`
  - ウィンドウを終了する命令。必ず最後に実行しましょう。



- ウィンドウの表示
  - ×ボタンが押されるまで何もしない。
  - ウィンドウの終了
- 
- という流れになっています。

# ●の表示

- 例えば、座標 $X=100$  座標 $Y=200$  に半径50の赤い●を描画するとします。
- `Graphics.DrawCircle(100,200,50,255,0,0);`
- これを{ }の中で使用すると●が描画されます。
- それぞれの数字はX座標、Y座標、半径、赤色の強さ、緑色の強さ、青色の強さを表しています。
- コンピュータ上では色は0～255で表現されます。

# ●を方向キーで動かせるようにする

- これでは、ただ●を表示しただけです。
- キーボードで移動させれるようにしましょう。
- シューティングなら主人公機になります。

# 変数

- 今までは描画する位置を直接数字で記入。
- 移動させたい場合、値を変更する必要あり。
- 値を一時的に保存する必要がある。
- そのための変数。
- 値には種類があり、intという整数、floatという少数、boolというYesNoなどがある。
- 今回はint と bool しか使用しません。

- `int x = 0;`
- 整数を保存できるxという領域に0を入れて生成。
- `x = 1;`
- xに1を代入します。数学と違い=は代入。
- `x = y;` としてyに保存されている値をxに代入することもできます。
- `Graphics.DrawCircle(x,200,50,255,0,0);`
- このように使用することができます。

# 注意

- 変数は記述する位置が重要です。
- 変数を使用する前に、宣言してください。
- {}の中で宣言すると、実行するたびに初期化されてしまいます。
- 少しずつxが増えていくプログラムは以下のようになります。

```
int x = 0;
while (Game.DoEvent()==true)
{
    x = x + 1;
}
```

# 実際にやってみましょう。

- ●が左から右に動くプログラムを書いてみましょう

# if文

- ボタンを押したときだけ、数値を増やしたい。
- そのときだけ命令を実行したいときに使う。

```
if(条件文)
{
    条件文が正のとき実行される内容
}
else
{
    条件文が否のとき実行される内容(省略可能)
}
```

- 条件文はbool型の変数も使用できます。



# 例

- `if( x == 1 ) { xが1のときに実行 }`
- `if( x < 1 ) { xが1より小さいときに実行 }`
- `if( x < 1 && x > -1 ){ xが-1～1のときに実行 }`
- `if( x > 1 || x < -1 ){ xが～-1もしくは1～のときに実行 }`
- 「&&」は「かつ」、「||」は「もしくは」
- `( x < 1 && x > -1 ) || x > 100`
- ()を使用で複雑な条件式も記述可能

# キーを押してることを判断するには？

- `if(Key.IsPressUp())`{↑ボタンを押している時}
- `if(Key.IsPressDown())`{↓ボタンを押している時}
- `if(Key.IsPressLeft())`{←ボタンを押している時}
- `if(Key.IsPressRight())`{→ボタンを押している時}
- `if(Key.IsPressZ())`{Zボタンを押している時}
  
- これらの方法でキーの状態によって分岐できます。
- `Key.IsPush～`で押した瞬間かどうかわかります

# 実際にやってみましょう。

- ●をキーボードで動かせるようにしましょう。
- ↑を押している時は $y$ を減らす、  
→を押している時は $x$ を増やす、  
というようにすればできるでしょう。

## ここまでで

- 労力を無視すれば変数とif文だけで作れない2Dゲームはほとんどありません。
- これまでの知識でインベーダーなら問題なくつくれるでしょう。
- 弾と敵との当たり判定は点と点の距離と半径を使用します。

# 弾や敵を沢山出すために

- 現在の知識では、敵や弾が増えることに変数が増えます。
- 2, 3体でしたら、変数の名前を考えるのも面倒ではないでしょうか？
- その問題を解消するための機能がいくつかあります。

# 配列

- 一度の宣言で大量の変数を用意できる。
- 例えば
- `int[] int_array = new int[10];`  
と宣言するとint型の変数を10個生成。
- `int_array[0] = 0; ... int_array[9] = 9;`  
などとして代入が可能。
- 10個生成した場合は0～9番。注意！

# for文

- 配列だけでは沢山宣言する必要があったのが1行になっただけ。
- 例えば、`int_array[0] = 0; ... int_array[9] = 9;`の代入を手軽に行いたい。
- そのためのfor文
- 以下のようにすると0～9までがそれぞれ代入。

```
for(int i = 0; i < 10; i++)  
{  
    int_array[i] = i;  
}
```

// i++ は `i = i + 1;` の省略

# さらにfor文

- for( 変数宣言; 条件文; 一周するごとに実行する内容 )  
  { 毎回実行する内容 }
- 条件を満たしている限り{}の中身を実行。
- for(int i = 0; i < 10; i++){ } は  
  i = 0で{}の中を実行、i=1で{}の中を実行、・・・i=9で{}の中を実行
- for(・・・){ for(・・・){ ・・・ } }  
  と何重にも重ねれる。



# while

- とにかく同じ内容を繰り返したい時もある。
- そのときのwhile。
- while(条件文){ 実行する内容 }
- 条件文が正しい限り{}の中を繰り返し実行。

# さらにwhile

- `while (Game.DoEvent()==true){}`
- `Game.DoEvent()`はゲームを終了しないかbool型で取得する関数
- `true`である限りゲームは終了しない。
- 注意！ `while( 1 == 1 ){} などとするとフリーズ`

# シューティング開発のヒント

- ここまでの知識でシューティング程度なら問題なく開発可能
- 敵の最大表示数の分、X座標、Y座標、使用中かどうか？の変数の配列を用意
- 弾も最大数の分、X座標、Y座標、使用中かどうか？の変数の配列を用意
- 使用中のときだけ描画
- 弾と敵が衝突したら使用中でなくす。
- 一定時間ごとに敵の座標を代入して敵を出現
- Zボタンを押すたびに弾の座標に数値を代入して発射
- 弾が使用中のときは上方向に移動
- さらに楽する方法、複雑なことをする方法は次回以降

今の知識で色々作ってみましょう

# 自習のために

- 開発環境(Visual C# 2008 Express Edition)
- <http://www.microsoft.com/japan/msdn/vstudio/express/default.aspx>
- 猫でもわかるプログラミング  
[http://homepage2.nifty.com/c\\_lang/](http://homepage2.nifty.com/c_lang/)
- C#によるプログラミング入門
- <http://ufcpp.net/study/csharp/>