平成 29 年度卒業研究
初心者のためのゲームプログラミング用
教材の開発

函館工業高等専門学校
生産システム工学科　情報コース　5 年
25 番　細見　政央
指導教員　東海林智也

目次

第１章　英文アブストラクト

The purpose of this research is to develop a library that can create a simple 2D game as a teaching material for game programming for beginners. There are many competing libraries such as Greenfoot [1] and Unity[2], but it is difficult for beginners to deal with them. Therefore, we have developed a simple library using C# with that beginners can study game programming easily.

Key words：C#, game programming, teaching material

第 2 章　研究目的

本研究の目的はオブジェクト指向プログラミングの知識をある程度持っている中級者から初心者までを対象としたゲームプログラミング教育用の教材として、2D ゲーム制作ライブラリの開発をすることである。

第 3 章　研究背景

現在、数多くの競合ライブラリ[1][2]が存在するが、機能が多く複雑なため扱いが困難であるなど学習者にとって教材として向かないものが多かった。

第 4 章　開発環境

使用 OS :　　Windows10 home

使用 PC のスペック：Intel Core i7-3687U　RAM 8.00GB

使用ソフト Microsoft Visual Studio 2017

使用言語 C#［3］

第 5 章　開発した 2D ゲーム制作ライブラリの概要

本研究では学習者が 2D ゲームを制作するために必要なクラスをライブラリとして実装した。

実装したクラスとその機能は以下のとおりである。

Player クラス

ゲーム制作に必要不可欠な自キャラの以下の機能を実装したクラスである(ソース 1)。

・自キャラのスーパークラス

・自キャラ座標、高さ、幅などの情報を保持

・自キャラの描画処理

自キャラの描画処理内容はサブクラスである MyPlayer の MyDraw メソッドに記述する。

```
using System.Drawing;

namespace stg
{
    abstract public class Player
    {

        public int x;
        public int y;

        private Bitmap Img;

        public int w,h;

        public Player(int x1,int y1, string a )
        {
            Img = new Bitmap(a);

            x =x1;
            y =y1;
            w = Img.Width;
            h = Img.Height;


        }

        abstract public void MyDraw(Graphics g);

        abstract public void Hittest(Enemy enemy);

        public void Draw(Graphics g)
        {

            MyDraw(g);

            g.DrawImage(Img, x, y, Img.Width, Img.Height);//自機描画

            Img.MakeTransparent(Color.White);//透過職指定

        }

    }
}
```

ソース１　Player

MyPlayer クラス

Player のサブクラスであり学習者が制作したいゲームに合わせ、自キャラの以下の機能を拡張するクラスである(ソース 2)。

・自キャラ画像の参照

・自キャラの十字キーによる上下左右の操作

当たり判定時の処理として自キャラと当たった敵キャラを非表示にする。

```
public class MyPlayer : Player {
        public MyPlayer(int x1, int y1) : base(x1,y1, "playerA.png"){}
        public override void MyDraw(Graphics g){
        if ((Keyboard.GetKeyStates(Key.Right) & KeyStates.Down) > 0) x+=14;
        if ((Keyboard.GetKeyStates(Key.Left) & KeyStates.Down) > 0) x-=14;
        if ((Keyboard.GetKeyStates(Key.Up) & KeyStates.Down) > 0) y-=14;
        if ((Keyboard.GetKeyStates(Key.Down) & KeyStates.Down) > 0) y+=14;
        }
        public override void Hittest(Enemy enemy) { enemy.show = false; }
}
```

ソース 2 MyPlayer サンプル

Enemy クラス

ゲーム制作に必要不可欠な敵キャラの以下の機能を実装したクラスである(ソース 3)。

・敵キャラのスーパークラス

・敵キャラの座標、高さ、幅などの情報を保持

・敵キャラの描画処理

敵キャラの描画処理内容はサブクラスである Enemy1 の MyDraw メソッドに記述する。

```
using System.Drawing;

namespace stg
{
    public abstract class Enemy
    {
        public bool show;
        public int x;
        public int y;

        public int w, h;
        private Bitmap Img;

        public Enemy(int x1, int y1, string a)
        {
            Img = new Bitmap(a);

            x = x1;
            y = y1;
            w = Img.Width;
            h = Img.Height;
            show = true;
        }

        abstract public void MyDraw(Graphics g);

        public void Draw(Graphics g)
        {
            if (show == false) return;

            MyDraw(g);

            g.DrawImage(Img, x, y, Img.Width, Img.Height);


        }



    }
}
```

ソース 3　Enemy

Enemy1,Enemy2 クラス

Enemy のサブクラスであり、学習者が制作したいゲームに合わせ、敵キャラの以下の機能を拡張していくクラスである(ソース 4)。

・敵キャラ画像の参照

・敵キャラの移動処理

```
public class Enemy1 : Enemy
{
        private bool f = true;
        public Enemy1(int x1, int y1) : base(x1, y1, "enemyA.png") { }
        public override void MyDraw(Graphics g) {
                if (x == 300) f = true;
                if (x == 0) f = false;
                if (f == false) x += 10;
                if (f == true) x -= 10;
        }
}}
```

ソース 4　Enemy1 サンプル

Form1 クラス

各クラスのインスタンス化と以下の機能を実装したクラスである（ソース 5，6）

・背景画像の描画処理

・ゲームクリアの判定処理

・キャラクターの当たり判定処理

```
    private Player player;
    private Enemy[] enemy = new Enemy[2];
public   Form1() {
    InitializeComponent();
    player = new MyPlayer(380, 200);
    enemy[0]= new Enemy1(0, 100);
    enemy[1]= new Enemy2(200, 300);}
```

ソース 5　インスタンス化

```
for (int i = 0; i < 2; i++)
		{
			enemy[i].Draw(g);

			if (enemy[i].show == true)
			{
				f_clear = false;
			}

			if (enemy[i].x < player.x + player.w && enemy[i].x + enemy[i].w
> player.x && enemy[i].y < player.y + player.h && enemy[i].y + enemy[i].h >
player.y)
			{
				player.Hittest(enemy[i]);

			}
		}
```

ソース 6 当たり判定処理

第 6 章　ライブラリの使用方法

学習者は主に Form1、MyPlayer、Enemy1、Enemy2 の 4 つのクラスを使用する。

Form1 では背景の差し替え、メッセージ表示、敵キャラのインタスタンスの追加が可能である。

Myplayer では学習者が自キャラの操作や当たり判定処理のプログラムを組んでいく。

同様に Enemy1、Enemy2 では学習者が敵の動きのプログラムを組んでいく。

第 7 章　まとめと今後の課題

ゲームプログラミング用教材として簡易的な 2D ゲーム制作ライブラリを開発した。しかし、以下のような課題が残る。今後はこれらの課題の解決を目指す。

・未実装の機能

自キャラや敵キャラが発射する弾のクラスを実装し、制作できるゲームの幅を広げる。

・サンプルゲームの制作

動作確認程度のプログラムしか組んでいないため完成度の高いサンプルゲームを制作する。

・学習者からの評価

実際に学習者に使用してもらい、アンケート形式でゲームプログラミングを学習する教材として適切かどうか、評価してもらう必要がある

参考文献

[1] Greenfoot

https://www.greenfoot.org/door

[2]Unity

https://unity3d.com

[3] C#解説

https://dobon.net/