**1字型隊伍**

MakeMonster("A1", new MONSTER1(new Vector2(100, 0)), 17);

**一字型隊伍**

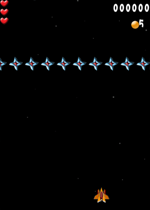
MakeMonster("A2", new MONSTER1(new Vector2(100, 0)), 17);

**斜一字型隊伍(左低右高)**

MakeMonster("A3", new MONSTER1(new Vector2(0, 0)), 17);

**斜一字型隊伍(左高右低)**

MakeMonster("A4", new MONSTER1(new Vector2(480, 0)), 17);



**螺旋型隊伍(直)**

MakeMonster("B2", new MONSTER1(new Vector2(240, 0)), 17);

**螺旋型隊伍(橫)**

MakeMonster("B3", new MONSTER1(new Vector2(240, 400)), 17);

**波浪型隊伍(橫)**

MakeMonster("B4", new MONSTER1(new Vector2(0, 0)), 17);

**波浪型隊伍(直)**

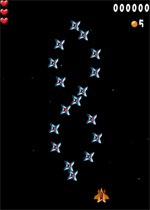
MakeMonster("B5", new MONSTER1(new Vector2(240, 0)), 17);

**(相反)波浪型隊伍(橫)**

MakeMonster("B4", new MONSTER1(new Vector2(0, 0)), 17);

**(相反)波浪型隊伍(直)**

MakeMonster("B5", new MONSTER1(new Vector2(240, 0)), 17)



**亂型排列**

MakeMonster("C1", new MONSTER1(new Vector2(240, 400)), 17);

**三角形隊伍**

MakeMonster("B1", new MONSTER1(new Vector2(0, 0)), 17);

MakeMonster("B1", new MONSTER1(new Vector2(-15, 20)), 11);

MakeMonster("B1", new MONSTER1(new Vector2(200, 200)),3);

//正方形走法

for (int i = 0; i < 20;i++ )

{

MONSTER11 MMM = new MONSTER11(new Vector2(100, -60 \* i));

MMM.N2 = 120;

MMM.ADD\_X2 = 2.5f;

MMM.ADD\_Y2 = 0;

MMM.N3 = 85;

MMM.ADD\_X3 = 0;

MMM.ADD\_Y3 = -2.5f;

MMM.N4 = 85;

MMM.ADD\_X4 = -2.5f;

MMM.ADD\_Y4 = 0;

MMM.N5 = 500;

MMM.ADD\_X5 = 0;

MMM.ADD\_Y5 = 2.5f;

BOX.Add(MMM);

}

for (int i = 0; i < 20;i++ )

{

MONSTER11 MMM = new MONSTER11(new Vector2(380, -60 \* i));

MMM.N2 = 120;

MMM.ADD\_X2 = -2.5f;

MMM.ADD\_Y2 = 0;

MMM.N3 = 85;

MMM.ADD\_X3 = 0;

MMM.ADD\_Y3 = -2.5f;

MMM.N4 = 85;

MMM.ADD\_X4 = 2.5f;

MMM.ADD\_Y4 = 0;

MMM.N5 = 500;

MMM.ADD\_X5 = 0;

MMM.ADD\_Y5 = 2.5f;

BOX.Add(MMM);

}

閃電跑法

for (int i = 0; i < 20; i++)

{

MONSTER11 MMM = new MONSTER11(new Vector2(380, -60 \* i));

MMM.Yn = 400;

MMM.N2 = 150;

MMM.ADD\_X2 = -1.5f;

MMM.ADD\_Y2 = -2f;

MMM.N3 = 500;

MMM.ADD\_X3 = 0f;

MMM.ADD\_Y3 = 2f;

BOX.Add(MMM);

}

for (int i = 0; i < 20; i++)

{

MONSTER11 MMM = new MONSTER11(new Vector2(100, -60 \* i));

MMM.Yn = 400;

MMM.N2 = 150;

MMM.ADD\_X2 = 1.5f;

MMM.ADD\_Y2 = -2f;

MMM.N3 = 500;

MMM.ADD\_X3 = 0f;

MMM.ADD\_Y3 = 2f;

BOX.Add(MMM);

}

二次斜線

for (int i = 0; i < 20; i++)

{

MONSTER11 MMM = new MONSTER11(new Vector2(430, -60 \* i));

MMM.Yn = 250;

MMM.N2 = 120;

MMM.ADD\_X2 = -1f;

MMM.ADD\_Y2 = 2f;

MMM.N3 = 500;

MMM.ADD\_X3 = 0f;

MMM.ADD\_Y3 = 2f;

BOX.Add(MMM);

}

for (int i = 0; i < 20; i++)

{

MONSTER11 MMM = new MONSTER11(new Vector2(50, -60 \* i));

MMM.Yn = 250;

MMM.N2 = 120;

MMM.ADD\_X2 = 1f;

MMM.ADD\_Y2 = 2f;

MMM.N3 = 500;

MMM.ADD\_X3 = 0f;

MMM.ADD\_Y3 = 2f;

BOX.Add(MMM);

}

菱形跑法

for (int i = 0; i < 20; i++)

{

MONSTER11 MMM = new MONSTER11(new Vector2(240, -60 \* i));

MMM.Yn = 100;

MMM.N2 = 150;

MMM.ADD\_X2 = -1.5f;

MMM.ADD\_Y2 = 2f;

MMM.N3 = 500;

MMM.ADD\_X3 = 1.5f;

MMM.ADD\_Y3 = 2f;

BOX.Add(MMM);

}

for (int i = 0; i < 20; i++)

{

MONSTER11 MMM = new MONSTER11(new Vector2(240, -60 \* i));

MMM.Yn = 100;

MMM.N2 = 150;

MMM.ADD\_X2 = 1.5f;

MMM.ADD\_Y2 = 2f;

MMM.N3 = 500;

MMM.ADD\_X3 = -1.5f;

MMM.ADD\_Y3 = 2f;

BOX.Add(MMM);

}

僅能選擇一邊攻擊法

for (int i = 0; i < 20; i++)

{

MONSTER11 MMM = new MONSTER11(new Vector2(20, -60 \* i));

MMM.Yn = 650;

MMM.N2 = 150;

MMM.ADD\_X2 = 2f;

MMM.ADD\_Y2 = 0f;

MMM.N3 = 500;

MMM.ADD\_X3 = 0f;

MMM.ADD\_Y3 = 2f;

BOX.Add(MMM);

}

for (int i = 0; i < 20; i++)

{

MONSTER11 MMM = new MONSTER11(new Vector2(440, -60 \* i));

MMM.Yn = 650;

MMM.N2 = 150;

MMM.ADD\_X2 = -2f;

MMM.ADD\_Y2 = 0f;

MMM.N3 = 500;

MMM.ADD\_X3 = 0f;

MMM.ADD\_Y3 = 2f;

BOX.Add(MMM);

}

單條後分開

for (int i = 0; i < 20; i++)

{

MONSTER11 MMM = new MONSTER11(new Vector2(240, -40 \* i));

MMM.Yn = 100;

if (i % 2 == 0)

{

MMM.N2 = 150;

MMM.ADD\_X2 = -1f;

MMM.ADD\_Y2 = 2f;

MMM.N3 = 500;

MMM.ADD\_X3 = 1f;

MMM.ADD\_Y3 = 2f;

}

else

{

MMM.N2 = 150;

MMM.ADD\_X2 = 1f;

MMM.ADD\_Y2 = 2f;

MMM.N3 = 500;

MMM.ADD\_X3 = -1f;

MMM.ADD\_Y3 = 2f;

}

BOX.Add(MMM);

}

天裡散花 散開型

//中間

for (int i = 0; i < 100; i++)

{

MONSTER11 MMM = new MONSTER11(new Vector2(240, -40 \* i));

MMM.Yn = 100;

MMM.N2 = GamePage.randObj.Next(50, 200);

MMM.ADD\_X2 = (float)(GamePage.randObj.Next(-150,150)\*0.01);

MMM.ADD\_Y2 = 2f;

MMM.N3 = 500;

MMM.ADD\_X3 = 0f;

MMM.ADD\_Y3 = 2f;

BOX.Add(MMM);

}

//右邊

for (int i = 0; i < 100; i++)

{

MONSTER11 MMM = new MONSTER11(new Vector2(450, -40 \* i));

MMM.Yn = GamePage.randObj.Next(0, 150);

MMM.N2 = GamePage.randObj.Next(80, 320);

MMM.ADD\_X2 = (float)(GamePage.randObj.Next(-150,0)\*0.01);

MMM.ADD\_Y2 = 2f;

MMM.N3 = 500;

MMM.ADD\_X3 = 0f;

MMM.ADD\_Y3 = 2f;

BOX.Add(MMM);

}

//左邊

for (int i = 0; i < 100; i++)

{

MONSTER11 MMM = new MONSTER11(new Vector2(50, -40 \* i));

MMM.Yn = GamePage.randObj.Next(0, 150);

MMM.N2 = GamePage.randObj.Next(80, 320);

MMM.ADD\_X2 = (float)(GamePage.randObj.Next(0, 150) \* 0.01);

MMM.ADD\_Y2 = 2f;

MMM.N3 = 500;

MMM.ADD\_X3 = 0f;

MMM.ADD\_Y3 = 2f;

BOX.Add(MMM);

}

//後面攻擊

for (int i = 0; i < 3; i++)

{

MONSTER11 MMM = new MONSTER11(new Vector2(40 \* i+480-50, 750));

MMM.Yn = 0;

MMM.N2 = 250;

MMM.ADD\_X2 = -1f;

MMM.ADD\_Y2 = -2f;

MMM.N3 = 500;

MMM.ADD\_X3 = -1f;

MMM.ADD\_Y3 = 2f;

BOX.Add(MMM);

}

for (int i = 0; i < 3; i++)

{

MONSTER11 MMM = new MONSTER11(new Vector2(-40 \* i+50, 750));

MMM.Yn = 0;

MMM.N2 = 250;

MMM.ADD\_X2 = 1f;

MMM.ADD\_Y2 = -2f;

MMM.N3 = 500;

MMM.ADD\_X3 = 1f;

MMM.ADD\_Y3 = 2f;

BOX.Add(MMM);

}

//中間攻擊法

for (int i = 0; i < 3; i++)

{

MONSTER11 MMM = new MONSTER11(new Vector2(40 \* i+480-50, 500));

MMM.Yn = 500;

MMM.N2 = GamePage.randObj.Next(50,120);

MMM.ADD\_X2 = -1f;

MMM.ADD\_Y2 = -2f;

MMM.N3 = 500;

MMM.ADD\_X3 = -1f;

MMM.ADD\_Y3 = 1f;

BOX.Add(MMM);

}

for (int i = 0; i < 3; i++)

{

MONSTER11 MMM = new MONSTER11(new Vector2(-40 \* i+50, 500));

MMM.Yn = 500;

MMM.N2 = GamePage.randObj.Next(50, 120);

MMM.ADD\_X2 = 1f;

MMM.ADD\_Y2 = -2f;

MMM.N3 = 500;

MMM.ADD\_X3 = 1f;

MMM.ADD\_Y3 = 1f;

BOX.Add(MMM);

}