Московский Энергетический Институт (Технический Университет)

Лабораторная работа №1

Выполнил: студент группы

А-13-05

Фролов Евгений

05.05.09

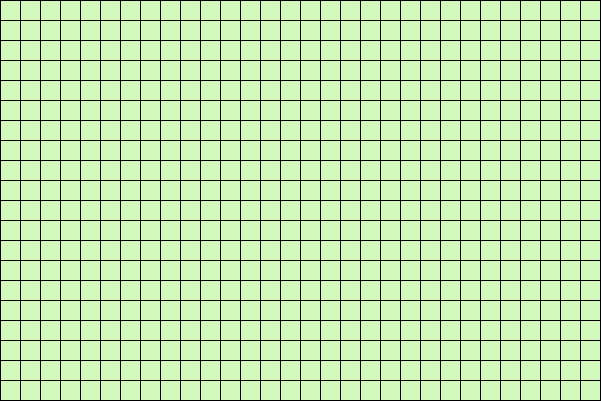
Постановка задачи

Создать игру, используя СИМПР. Игра заключается в моделировании поведения «классической» змейки с поведением «ракеты», пытающейся сбить змейку.

Описание предметной области

В разработанной игре присутствуют два динамических объекта: змейка и ракета. Движение змейки не зависит от перемещений ракеты, в то время как ракета учитывает позицию змейки. Поэтому сначала опишем поведение змейки. Для этого определим следующие объёкты: поле, стена, фрукт.

Поле – матрица 30х20. Описывает двумерное пространство, в котором происходит действие игры. Каждый элемент (стена, фрукт, ракета, секция змеи) занимает ровно одну клетку поля. Изображается:



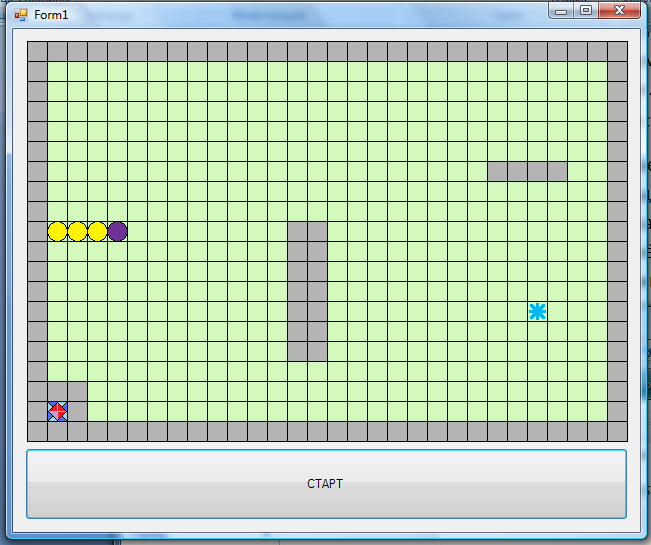
Стена – клетка поля, недоступная для перемещения змеи. Изображается: C:\Users\ASUS\Documents\МЭИ\ТИ\симпр\стена.bmp

Фрукт – объёкт, при совпадении координат которого с головой змейки (в этом случае змейка «съедает» фрукт) пропадает и появляется в произвольном месте. Также длина змейки увеличивается на единицу. Изображается: C:\Users\ASUS\Documents\МЭИ\ТИ\симпр\фрукт.bmp

На поле всегда находится ровно один фрукт, изначально заданное расположение стен, одна ракета и одна змейка. Цель змейки – съесть как можно больше фруктов. После «съедания» фрукта змейка обязательно направляется к следующему. При этом змейка не должна занимать клетки поля, уже занятые стенами. Также змея может передвигаться только по вертикали и горизонтали. Змея не может передвигаться по себе. Змея «делится» на голову и хвост. Голова: C:\Users\ASUS\Documents\МЭИ\ТИ\симпр\голова_змеи.bmp Хвост: C:\Users\ASUS\Documents\МЭИ\ТИ\симпр\хвост_змеи.bmp

Теперь опишем поведение ракеты. Цель ракеты – попасть в голову змейке, не подозревающей о существовании ракеты. По этой причине скорость змейки в два раза выше, чем скорость ракеты. Если ракета попадает не в голову змеи, а в её хвост и счетчик оставшихся ракет равен нулю, то ракета проигрывает. Если же ракета попадает в голову, то выигрывает ракета. Ракета может летать над стенами. Ракета: C:\Users\ASUS\Documents\МЭИ\ТИ\симпр\ракета.bmp

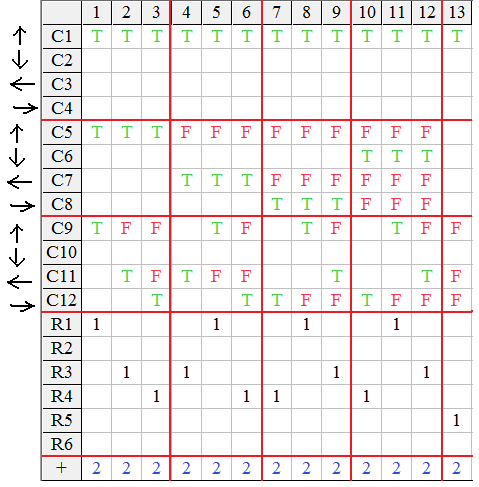
Начальное положение объектов и внешний вид приложения:



Описание табличной модели

Модель включает в себя две таблицы, отвечающие за логику поведения змеи и ракеты.

Таблица 1, регулирующая поведение змеи, состоит из четырёх частей идентичной структуры. Первая часть разделена красными линиями на логически схожие области:



Змея движется

Фрукт находится

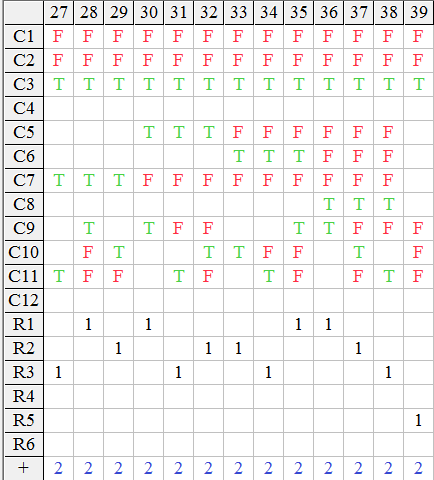
Свободная клетка



Пойти



Следующие части таблицы 1:



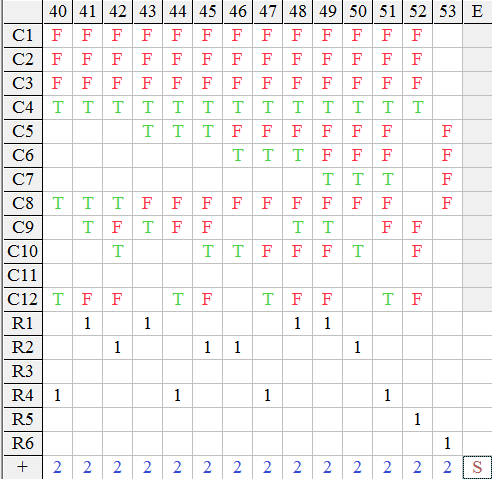
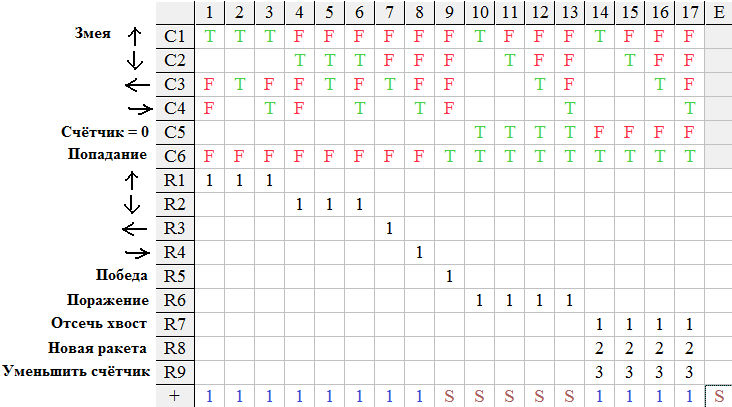
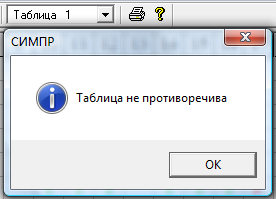
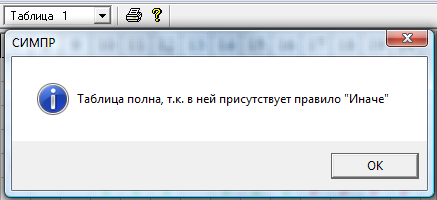
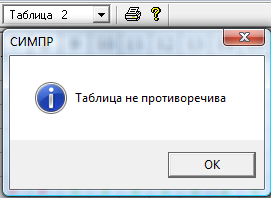
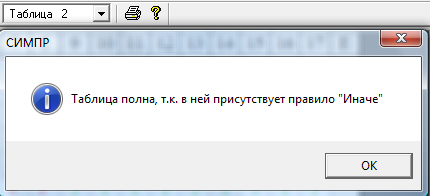


Таблица 2, регулирующая поведение ракеты:



Проверка полноты и непротиворечивости





Приложение

protected override void WndProc(ref Message m) // в эту функцию приходят все сообщения от СИМПРА

{

int wparamhi, wparamlo, wparam;

int lParam = Convert.ToInt32("" + m.LParam);

if (m.Msg == simpr)

{

wparam = Convert.ToInt32("" + m.WParam);

wparamhi = wparam / 65536;

wparamlo = wparam - wparamhi \* 65536;

if (wparamhi == 0)//условия

{

if (wparamlo == 1)// таблица 1

{

if (lParam == 1)// таблица 1 условие 1

{

myGame.fldPoint pntHead = gmGame.snkSnake.GiveSnake()[0];

myGame.fldPoint pntTail = gmGame.snkSnake.GiveSnake()[1];

if ((pntHead.x == pntTail.x) && (pntHead.y < pntTail.y))

m.Result = new IntPtr(1);

else

m.Result = new IntPtr(0);//0

}

else if (lParam == 2)// таблица 1 условие 2

{

myGame.fldPoint pntHead = gmGame.snkSnake.GiveSnake()[0];

myGame.fldPoint pntTail = gmGame.snkSnake.GiveSnake()[1];

if ((pntHead.x == pntTail.x) && (pntHead.y > pntTail.y))

m.Result = new IntPtr(1);

else

m.Result = new IntPtr(0);

}

else if (lParam == 3)// таблица 1 условие 3

{

myGame.fldPoint pntHead = gmGame.snkSnake.GiveSnake()[0];

myGame.fldPoint pntTail = gmGame.snkSnake.GiveSnake()[1];

if ((pntHead.y == pntTail.y) && (pntHead.x < pntTail.x))

m.Result = new IntPtr(1);

else

m.Result = new IntPtr(0);

}

else if (lParam == 4)// таблица 1 условие 4

{

myGame.fldPoint pntHead = gmGame.snkSnake.GiveSnake()[0];

myGame.fldPoint pntTail = gmGame.snkSnake.GiveSnake()[1];

if ((pntHead.y == pntTail.y) && (pntHead.x > pntTail.x))

m.Result = new IntPtr(1);

else

m.Result = new IntPtr(0);//0

}

else if (lParam == 5)// таблица 1 условие 5

{

myGame.fldPoint pntHead = gmGame.snkSnake.GiveSnake()[0];

myGame.fldPoint pntFruit = gmGame.frtFruit.GiveFruit();

if (pntHead.y > pntFruit.y)

m.Result = new IntPtr(1);

else

m.Result = new IntPtr(0);//0

}

else if (lParam == 6)// таблица 1 условие 6

{

myGame.fldPoint pntHead = gmGame.snkSnake.GiveSnake()[0];

myGame.fldPoint pntFruit = gmGame.frtFruit.GiveFruit();

if (pntHead.y < pntFruit.y)

m.Result = new IntPtr(1);

else

m.Result = new IntPtr(0);//0

}

else if (lParam == 7)// таблица 1 условие 7

{

myGame.fldPoint pntHead = gmGame.snkSnake.GiveSnake()[0];

myGame.fldPoint pntFruit = gmGame.frtFruit.GiveFruit();

if ((pntHead.y == pntFruit.y) && (pntHead.x > pntFruit.x))

m.Result = new IntPtr(1);

else

m.Result = new IntPtr(0);

}

else if (lParam == 8)// таблица 1 условие 8

{

myGame.fldPoint pntHead = gmGame.snkSnake.GiveSnake()[0];

myGame.fldPoint pntFruit = gmGame.frtFruit.GiveFruit();

if ((pntHead.y == pntFruit.y) && (pntHead.x < pntFruit.x))

m.Result = new IntPtr(1);

else

m.Result = new IntPtr(0);

}

else if (lParam == 9)// таблица 1 условие 9

{

//свободность

myGame.fldPoint pntPoint = gmGame.snkSnake.GiveSnake()[0];

pntPoint.y--;

myGame.fldPoint[] aWalls;

aWalls = new myGame.fldPoint[gmGame.lstWalls.Count];

gmGame.lstWalls.CopyTo(aWalls);

myGame.fldPoint[] aSnake;

aSnake = new myGame.fldPoint[600];

aSnake = gmGame.snkSnake.GiveSnake();

if ((gmGame.InWall(pntPoint, aWalls)) ||

(gmGame.InWall(pntPoint, aSnake)))

m.Result = new IntPtr(0);//0

else m.Result = new IntPtr(1);

}

else if (lParam == 10)// таблица 1 условие 10

{

myGame.fldPoint pntPoint = gmGame.snkSnake.GiveSnake()[0];

pntPoint.y++;

myGame.fldPoint[] aWalls;

aWalls = new myGame.fldPoint[gmGame.lstWalls.Count];

gmGame.lstWalls.CopyTo(aWalls);

myGame.fldPoint[] aSnake;

aSnake = new myGame.fldPoint[600];

aSnake = gmGame.snkSnake.GiveSnake();

if ((gmGame.InWall(pntPoint, aWalls)) ||

(gmGame.InWall(pntPoint, aSnake)))

m.Result = new IntPtr(0);

else m.Result = new IntPtr(1);

}

else if (lParam == 11)// таблица 1 условие 11

{

myGame.fldPoint pntPoint = gmGame.snkSnake.GiveSnake()[0];

pntPoint.x--;

myGame.fldPoint[] aWalls;

aWalls = new myGame.fldPoint[gmGame.lstWalls.Count];

gmGame.lstWalls.CopyTo(aWalls);

myGame.fldPoint[] aSnake;

aSnake = new myGame.fldPoint[600];

aSnake = gmGame.snkSnake.GiveSnake();

if ((gmGame.InWall(pntPoint, aWalls)) ||

(gmGame.InWall(pntPoint, aSnake)))

m.Result = new IntPtr(0);

else m.Result = new IntPtr(1);

}

else if (lParam == 12)// таблица 1 условие 12

{

myGame.fldPoint pntPoint = gmGame.snkSnake.GiveSnake()[0];

pntPoint.x++;

myGame.fldPoint[] aWalls;

aWalls = new myGame.fldPoint[gmGame.lstWalls.Count];

gmGame.lstWalls.CopyTo(aWalls);

myGame.fldPoint[] aSnake;

aSnake = new myGame.fldPoint[600];

aSnake = gmGame.snkSnake.GiveSnake();

if ((gmGame.InWall(pntPoint, aWalls)) ||

(gmGame.InWall(pntPoint, aSnake)))

m.Result = new IntPtr(0);

else m.Result = new IntPtr(1);

}

}

else if (wparamlo == 2)// таблица 2

{

if (lParam == 1)

{

myGame.fldPoint pntHead = gmGame.snkSnake.GiveSnake()[0];

myGame.fldPoint pntRocket = gmGame.rctRocket.GiveRocket();

if (pntHead.y < pntRocket.y)

m.Result = new IntPtr(1);

else

m.Result = new IntPtr(0);

}

else if (lParam == 2)

{

myGame.fldPoint pntHead = gmGame.snkSnake.GiveSnake()[0];

myGame.fldPoint pntRocket = gmGame.rctRocket.GiveRocket();

if (pntHead.y > pntRocket.y)

m.Result = new IntPtr(1);

else

m.Result = new IntPtr(0);

}

else if (lParam == 3)

{

myGame.fldPoint pntHead = gmGame.snkSnake.GiveSnake()[0];

myGame.fldPoint pntRocket = gmGame.rctRocket.GiveRocket();

if (pntHead.x < pntRocket.x)

m.Result = new IntPtr(1);

else

m.Result = new IntPtr(0);

}

else if (lParam == 4)

{

myGame.fldPoint pntHead = gmGame.snkSnake.GiveSnake()[0];

myGame.fldPoint pntRocket = gmGame.rctRocket.GiveRocket();

if (pntHead.x > pntRocket.x)

m.Result = new IntPtr(1);

else

m.Result = new IntPtr(0);

}

else if (lParam == 5)

{

if (gmGame.rctRocket.GiveCount() <= 0)

m.Result = new IntPtr(1);

else

m.Result = new IntPtr(0);

}

else if (lParam == 6)

{

myGame.fldPoint pntRocket = gmGame.rctRocket.GiveRocket();

myGame.fldPoint[] aSnake;

aSnake = new myGame.fldPoint[600];

aSnake = gmGame.snkSnake.GiveSnake();

if (gmGame.InWall(pntRocket, aSnake))

m.Result = new IntPtr(1);//0

else m.Result = new IntPtr(0); //1

}

}

}

else if (wparamhi == 1)//действия

{

if (wparamlo == 1)// таблица 1

{

if (lParam == 1)// таблица 1 действие 1

{

// действие

gmGame.snkSnake.Step((myGame.NSWE)1);

}

else if (lParam == 2)// таблица 1 действие 2

{

// действие

gmGame.snkSnake.Step((myGame.NSWE)2);

}

else if (lParam == 3)// таблица 1 действие 3

{

// действие

gmGame.snkSnake.Step((myGame.NSWE)3);

}

else if (lParam == 4)// таблица 1 действие 4

{

// действие

gmGame.snkSnake.Step((myGame.NSWE)4);

}

else if (lParam == 5)// таблица 1 действие 5

{

// действие

//стоим)

}

else if (lParam == 6)// таблица 1 действие 6

{

// действие

gmGame.snkSnake.Grow();

myGame.fldPoint[] aWalls;

aWalls = new myGame.fldPoint[gmGame.lstWalls.Count];

gmGame.lstWalls.CopyTo(aWalls);

gmGame.frtFruit.NewFruit(aWalls);

}

}

else if (wparamlo == 2)// таблица 2

{

if (lParam == 1)// таблица 2 действие 1

{

// действие

if (gmGame.rctRocket.bStep)

gmGame.rctRocket.Step((myGame.NSWE)1);

}

else if (lParam == 2)// таблица 2 действие 2

{

if (gmGame.rctRocket.bStep)

gmGame.rctRocket.Step((myGame.NSWE)2);

}

else if (lParam == 3)

{

if (gmGame.rctRocket.bStep)

gmGame.rctRocket.Step((myGame.NSWE)3);

}

else if (lParam == 4)

{

if (gmGame.rctRocket.bStep)

gmGame.rctRocket.Step((myGame.NSWE)4);

}

else if (lParam == 5)

{

//myGame.bWin = true;

gmGame.bWin1 = true;

}

else if (lParam == 6)

{

gmGame.bLose = true;

}

else if (lParam == 7)

{

gmGame.snkSnake.Butch(gmGame.rctRocket.GiveRocket());

}

else if (lParam == 8)

{

gmGame.rctRocket.New();

}

else if (lParam == 9)

{

gmGame.rctRocket.DecCount();

}

if (gmGame.rctRocket.bStep) gmGame.rctRocket.bStep = false;

else gmGame.rctRocket.bStep = true;

}

Application.DoEvents();

Thread.Sleep(200); // если у нас есть визуальное отображение, то задержку можно установить здесь

m.Result = new IntPtr(1); // ответом на запрос действия со стороны СИМПР должна быть единица

}

}

else

{

base.WndProc(ref m); // для всех действий не связанных с СИМПР возвращаем управление программе

}

}