

**Московский Энергетический Институт
(Технический Университет)**

ТЕОРИЯ ИГР И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Отчет по лабораторной работе №1

РЕАЛИЗАЦИЯ СИСТЕМЫ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ СИСТЕМЫ СИМПР

Выполнила	Лаврушко В.В. гр. А-5-01
-----------	-----------------------------

2005 г.

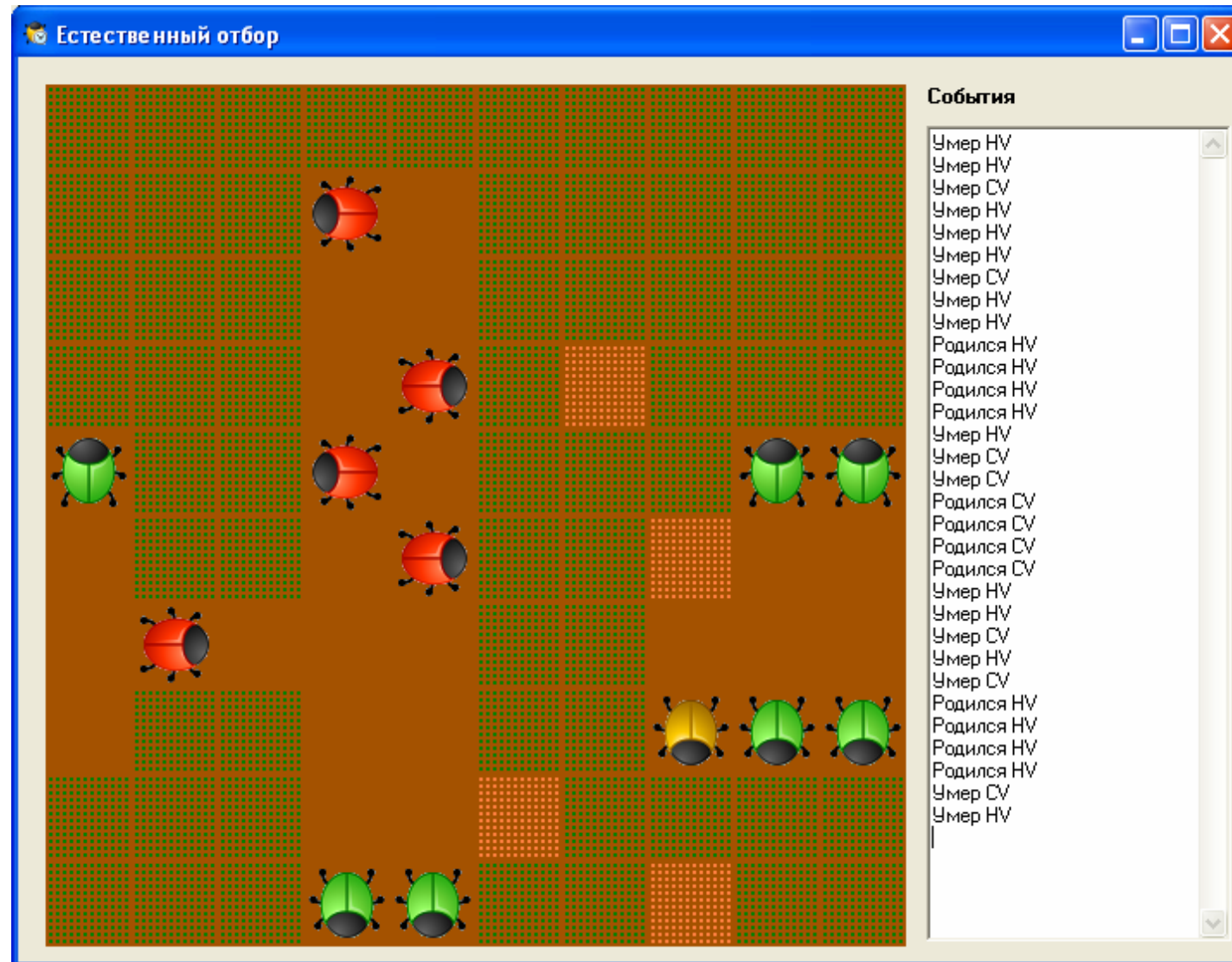
Постановка задачи

Сформировать модель принятия решений в виде совокупности таблиц решений и написать модуль-интерпретатор для данной модели. Объект моделирования – природное сообщество, состоящее из травоядных и хищных особей.

Описание модуля интерпретатора

Данная модель имитирует поведение жуков двух видов: herbivore (травоядные) и carnivore (хищники). Начальное состояние - на «поляне» находятся 8 хищников и 12 травоядных. Конечное состояние достигается в случае, когда погибают все жуки какого-либо вида. Программа была реализована в среде Delphi 6.

Основное окно:



Описание модели на языке таблиц решений

Модель включает в себя две таблицы, содержащие правила поведения для жуков каждого вида. Таблицы полны и непротиворечивы.

Таблица 1 (Жуки вида **herbivore**)

Жуки-вегетарианцы, питаются исключительно травкой (сокращенно НV).

Жизненный цикл

1. Рождение. В самом начале на поле живут 12 особей. Далее появляются каждые 30с (в результате деления).
2. Питание. Хотят питаться всегда. Смерть наступает в результате голода через 20с с момента последнего приема пищи.
3. Движение. Двигаются как каждые 3-6с и только вверх-вниз. Видят на расстоянии: 2 кл. по прямой, 1 кл. по диаг., не видят через к.-л. Важнее убежать, нежели поесть.
4. Размножение. Делятся (каждые 30с).
5. Болезни. Болеют после съедания травки оранжевого цвета (едят ее в самом крайнем случае). Симптомы: изменение цвета на оранжевый, хотят есть/двигаться каждые 7с, теряют способность к делению.
6. Смерть. Возможны варианты: жизнь >100с, голод >20с, съеден.

[illegible]

C12. З свободен?																					T	T	F	T		T	F	T	F					
C13. В свободен?																					T	T	T	F	T		T	F	F					
C14. В ближе к Ц?													T	F	T	F					T	F												
C15. С свободен?					T	T	T	T	F	T	F																							
C16. Ю свободен?					T	T	T	T	T	F	F																							
C17. HV(Ю)>HV(С)?					T	F	F	F																										
C18. HV(Ю)<HV(С)?						T	F	F																										
C19. Ю ближе к Ц?							F	T																										
Действия																																		
A1. Смерть			1	1										3	3	3			3												3		3	
A2. Шаг на З													1		1	1	1		1		1		1		1		1		1	1				
A3. Шаг на В												1		1				1	1	1		1		1		1		1			1	1		
A4. Поесть												2	2	2	2	2	2	2	2	2	2									2	2	2	2	
A5. Следующий					2	2	2	2	2	2	1	1	3	3				3	3		3	3	2	2	2	2	2	2	2	2	1	3		3
A6. Новый на С						1	1			1																								
A7. Новый на Ю					1			1	1																									
+	S	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Логические отношения:

- C10->!C11
- C17->!C18
- C6->!C13
- C7->!C12

Оптимизированная трансляция:

Таблица 1

Оптимизированная трансляция

C1 P1 C2 P2 C3 P3 C4 C7 P4 C8 C9 P29 C12 P23 C14 P26 C10 P7 C11 P10 C16 P15 P16 C5 C6 P37 P62 P40 P43 P48 P49 P56 P59 C17 C18 P36 P34 P35 C19 P30 C20 P31 C21 P32 P33 C15 P5 P17 P27 P38 P50 P60 P6 C13 P8 P11 P12 P19 P21 P22 P24 P41 P44 P45 P52 P54 P55 P57 P9 P13 P14 P18 P20 P25 P28 P39 P42 P46 P47 P51 P53 P58 P61

OK

Таблица 2

Оптимизированная трансляция

C1 P1 C2 P2 C3 P3 C4 C5 P11 C15 C16 P10 P8 P9 C17 P4 C18 P5 C19 P6 P7 C6 C7 C12 C13 P30 C10 C11 P24 P25 P26 P27 P28 P29 C14 P22 P23 C9 P31 P32 C8 P12 P13 P14 P15 P16 P17 P18 P19 P20 P21 P33 P34

OK

Приложение

Цикл обработки сообщений:

```
procedure TLifeField.WndProc(var Message: TMessage);
var Res: boolean;
begin
  if message.Msg=msgSIMPR then begin//ПОЛУЧИЛИ СООБЩЕНИЕ!!!!!!!!!!
    Res:=false;
    if message.WParamHi=0 then begin//УСЛОВИЕ!!!!!!!!!!
      case message.WParamLo of
        //таблица
        1: begin
          case message.LParam of
            1: begin // условие 1: есть живые?
              Res:=isHVAlive;
            end;
            2: begin // условие 2: живет>100с?
              aHV:=lstHV.Items[idHV];
              Res:=(SecondsBetween(Time,aHV^.birthTime)>100);
            end;
            3: begin // условие 3: ел>20с назад?
              aHV:=lstHV.Items[idHV];
              Res:=(SecondsBetween(Time,aHV^.lastAte)>20);
            end;
            4: begin // условие 4: болеет?
              aHV:=lstHV.Items[idHV];
              Res:=not(aHV^.isHealthy);
            end;
            5: begin // условие 5: делился>30с назад?
              aHV:=lstHV.Items[idHV];
              Res:=(SecondsBetween(Time,aHV^.lastDiv)>30);
            end;
            6: begin // условие 6: пора двигаться?
              aHV:=lstHV.Items[idHV];
              Res:=(SecondsBetween(Time,aHV^.lastMove)>aHV^.nextMove);
            end;
            7: begin // условие 7: двигался>7с назад?
              aHV:=lstHV.Items[idHV];
              Res:=(SecondsBetween(Time,aHV^.lastMove)>7);
            end;
            8: begin // условие 8: Ю свободен?
              aHV:=lstHV.Items[idHV];
              Res:=((aHV^.IPos<n) and (fieldArr[aHV^.IPos+1,aHV^.JPos]<otCV));
            end;
            9: begin // условие 9: С свободен?
              aHV:=lstHV.Items[idHV];
              Res:=((aHV^.IPos>1) and (fieldArr[aHV^.IPos-1,aHV^.JPos]<otCV));
            end;
            10: begin // условие 10: вр(Ю)>вр(С)?
              Res:=(enemiesHV(atSouth,idHV)>enemiesHV(atNorth,idHV));
            end;
            11: begin // условие 11: вр(Ю)<вр(С)?
```

```

    Res:=(enemiesHV(atSouth,idHV)<enemiesHV(atNorth,idHV));
end;
12: begin // условие 12: есть травка на Ю?
    aHV:=lstHV.Items[idHV];
    Res:=((aHV^.IPos<n) and ((fieldArr[aHV^.IPos+1,aHV^.JPos]=otFGR) or (fieldArr[aHV^.IPos+1,aHV^.JPos]=otPGR)));
end;
13: begin // условие 13: травка на Ю хорошая?
    aHV:=lstHV.Items[idHV];
    Res:=((aHV^.IPos<n) and (fieldArr[aHV^.IPos+1,aHV^.JPos]=otFGR));
end;
14: begin // условие 14: есть травка на С?
    aHV:=lstHV.Items[idHV];
    Res:=((aHV^.IPos>1) and ((fieldArr[aHV^.IPos-1,aHV^.JPos]=otFGR) or (fieldArr[aHV^.IPos-1,aHV^.JPos]=otPGR)));
end;
15: begin // условие 15: травка на С хорошая?
    aHV:=lstHV.Items[idHV];
    Res:=((aHV^.IPos>1) and (fieldArr[aHV^.IPos-1,aHV^.JPos]=otFGR));
end;
16: begin // условие 16: Ю ближе к Ц?
    aHV:=lstHV.Items[idHV];
    Res:=(farFromCenter(atSouth,aHV^.IPos,aHV^.JPos)<farFromCenter(atNorth,aHV^.IPos,aHV^.JPos));
end;
17: begin // условие 17: З свободен?
    aHV:=lstHV.Items[idHV];
    Res:=((aHV^.JPos>1) and (fieldArr[aHV^.IPos,aHV^.JPos-1]<otCV));
end;
18: begin // условие 18: В свободен?
    aHV:=lstHV.Items[idHV];
    Res:=((aHV^.JPos<n) and (fieldArr[aHV^.IPos,aHV^.JPos+1]<otCV));
end;
19: begin // условие 19: вр(В)>вр(З)?
    Res:=(enemiesHV(atEast,idHV)>enemiesHV(atWest,idHV));
end;
20: begin // условие 20: вр(В)<вр(З)?
    Res:=(enemiesHV(atEast,idHV)<enemiesHV(atWest,idHV));
end;
21: begin // условие 21: В ближе к Ц?
    aHV:=lstHV.Items[idHV];
    Res:=(farFromCenter(atEast,aHV^.IPos,aHV^.JPos)<farFromCenter(atWest,aHV^.IPos,aHV^.JPos));
end;
end; //case
end;
//таблица2
2: begin
    case message.LParam of
        1: begin // условие 1: есть живые?
            Res:=isCVAlive;;
        end;
        2: begin // условие 2: живет>120с?
            aCV:=lstCV.Items[idCV];
            Res:=(SecondsBetween(Time,aCV^.birthTime)>120);
        end;
        3: begin // условие 3: ел>40с назад?

```

```

    aCV:=lstCV.Items[idCV];
    Res:=(SecondsBetween(Time,aCV^.lastAte)>40);
end;
4: begin // условие 4: делился>45с назад?
    aCV:=lstCV.Items[idCV];
    Res:=(SecondsBetween(Time,aCV^.lastDiv)>45);
end;
5: begin // условие 5: пора двигаться?
    aCV:=lstCV.Items[idCV];
    Res:=(SecondsBetween(Time,aCV^.lastMove)>aCV^.nextMove);
end;
6: begin // условие 6: есть травоядный на В?
    aCV:=lstCV.Items[idCV];
    Res:=((aCV^.JPos<n) and (fieldArr[aCV^.IPos,aCV^.JPos+1]>otCV));
end;
7: begin // условие 7: есть травоядный на З?
    aCV:=lstCV.Items[idCV];
    Res:=((aCV^.JPos>1) and (fieldArr[aCV^.IPos,aCV^.JPos-1]>otCV));
end;
8: begin // условие 8: тр(В) здоров?
    aCV:=lstCV.Items[idCV];
    Res:=((aCV^.JPos<n) and (fieldArr[aCV^.IPos,aCV^.JPos+1]=otHV));
end;
9: begin // условие 9: тр(З) здоров?
    aCV:=lstCV.Items[idCV];
    Res:=((aCV^.JPos>1) and (fieldArr[aCV^.IPos,aCV^.JPos-1]=otHV));
end;
10: begin // условие 10: тр(В)>тр(З)?
    Res:=(foodCV(atEast,idCV)>foodCV(atWest,idCV));
end;
11: begin // условие 11: тр(В)<тр(З)?
    Res:=(foodCV(atEast,idCV)<foodCV(atWest,idCV));
end;
12: begin // условие 12: З свободен?
    aCV:=lstCV.Items[idCV];
    Res:=((aCV^.JPos>1) and (fieldArr[aCV^.IPos,aCV^.JPos-1]<otCV));
end;
13: begin // условие 13: В свободен?
    aCV:=lstCV.Items[idCV];
    Res:=((aCV^.JPos<n) and (fieldArr[aCV^.IPos,aCV^.JPos+1]<otCV));
end;
14: begin // условие 14: В ближе к Ц?
    aCV:=lstCV.Items[idCV];
    Res:=(farFromCenter(atEast,aCV^.IPos,aCV^.JPos)<farFromCenter(atWest,aCV^.IPos,aCV^.JPos));
end;
15: begin // условие 15: С свободен?
    aCV:=lstCV.Items[idCV];
    Res:=((aCV^.IPos>1) and (fieldArr[aCV^.IPos-1,aCV^.JPos]<otCV));
end;
16: begin // условие 16: Ю свободен?
    aCV:=lstCV.Items[idCV];
    Res:=((aCV^.IPos<n) and (fieldArr[aCV^.IPos+1,aCV^.JPos]<otCV));
end;

```



```

17: begin // условие 17: тр(Ю)>тр(С)?
    Res:=(foodCV(atSouth,idCV)>foodCV(atNorth,idCV));
end;
18: begin // условие 18: тр(Ю)<тр(С)?
    Res:=(foodCV(atSouth,idCV)<foodCV(atNorth,idCV));
end;
19: begin // условие 19: Ю ближе к Ц?
    aCV:=lstCV.Items[idCV];
    Res:=(farFromCenter(atSouth,aCV^.IPos,aCV^.JPos)<farFromCenter(atNorth,aCV^.IPos,aCV^.JPos));
end;
end; //case
end;
end; //case
end;

if message.WParamHi=1 then begin//ДЕЙСТВИЕ!!!!!!!!!!
case message.WParamLo of
    //таблица
    1: begin
        case message.LParam of
            1: begin // действие 1: смерть!
                killHV(idHV,true);
            end;
            2: begin // действие 2: шаг на север
                moveHV(atNorth,idHV);
            end;
            3: begin // действие 3: шаг на юг
                moveHV(atSouth,idHV);
            end;
            4: begin // действие 4: поесть
                aHV:=lstHV.Items[idHV];
                aHV^.lastAte:=Time;
            end;
            5: begin // действие 5: заболеть
                aHV:=lstHV.Items[idHV];
                aHV^.isHealthy:=false;
                //покрасить в желтый
                if (aHV^.IPos<(n div 2)+1) then imgMain.Canvas.Draw(koef*(aHV^.JPos-1),koef*(aHV^.IPos-1),bmpHVIS)
                else imgMain.Canvas.Draw(koef*(aHV^.JPos-1),koef*(aHV^.IPos-1),bmpHVIN);
                fieldArr[aHV^.IPos,aHV^.JPos]:=otHVI;
            end;
            6: begin // действие 6: следующий жучок
                if (lstHV.Count>0) then begin
                    if (idHV=(lstHV.Count-1)) then idHV:=0
                    else idHV:=idHV+1;
                end else idHV:=1;
            end;
            7: begin // действие 7: новый на 3
                aHV:=lstHV.Items[idHV];
                aHV^.lastDiv:=Time;
                createHV(aHV^.IPos,aHV^.JPos-1);
                Memol.Lines.Add('Родился HV');
            end;
        end;
    end;
end;

```

```

8: begin // действие 8: новый на В
  aHV:=lstHV.Items[idHV];
  aHV^.lastDiv:=Time;
  createHV(aHV^.IPos,aHV^.JPos+1);
  Memol.Lines.Add('Родился HV');
end;
end;//case
end;
//таблица2
2: begin
  case message.LParam of
    1: begin // действие 1: смерть!
      killCV(idCV);
    end;
    2: begin // действие 2: шаг на запад
      moveCV(atWest,idCV);
    end;
    3: begin // действие 3: шаг на восток
      moveCV(atEast,idCV);
    end;
    4: begin // действие 4: поесть
      //найти жучка HV для еды
      aCV:=lstCV.Items[idCV];
      aCV^.lastAte:=Time;
      killHV(getHVIdByIJ(aCV^.IPos,aCV^.JPos),false);
    end;
    5: begin // действие 5: следующий жучок
      if (lstCV.Count>0) then begin
        if (idCV=(lstCV.Count-1)) then idCV:=0
        else idCV:=idCV+1;
        end else idCV:=1;
      end;
    6: begin // действие 6: новый на С
      aCV:=lstCV.Items[idCV];
      aCV^.lastDiv:=Time;
      createCV(aCV^.IPos-1,aCV^.JPos);
      Memol.Lines.Add('Родился CV');
    end;
    7: begin // действие 7: новый на Ю
      aCV:=lstCV.Items[idCV];
      aCV^.lastDiv:=Time;
      createCV(aCV^.IPos+1,aCV^.JPos);
      Memol.Lines.Add('Родился CV');
    end;
  end;//case
end;
end;//case
Res:=true;
end;
//ВЫДАЛИ РЕЗУЛЬТАТ!!!!!!!!!!
if Res then message.Result:=Yes else message.Result:=No;
end else inherited;
end;

```