Департамент образования администрации г. Перми
МБОУ «Лицей №1» г. Перми

Информатика

Учебно-исследовательская работа
Программирование видеоигры на языке Pascal

Выполнил:
Москоков И.А.
11 класс, 209 группа.
Научный руководитель:
Петушина Р.Р.

Пермь, 2021

АННОТАЦИЯ

Разработана видеоигра на языке программирования Pascal.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В конце прошлого века была создана первая видеоигра. В настоящий момент видеоигры развиваются очень стремительно. Они могут быть как развивающие и обучающие, так, и сделаны, чтобы просто скоротать время. Создание игры рассчитывается по ролям: продюсер, команда разработки и другие – все эти роли может принять на себя как один человек, так и большая фирма. В своей работе я приму на себя все эти роли, чтобы продемонстрировать пример возможностей одного человека при создании игры на языке программирования Pascal. Для реализации графики в паскале использовать модуль GraphABC, он представляет собой простую графическую библиотеку и предназначен для создания не событийных графических и анимационных программ в процедурном и частично в объектном стиле [1]. Реализация геймплея будет происходить через чтение символических констант клавиш используемых семейством операционных систем Microsoft Windows [2].

Объект – видеоигра.

Предмет – программирование на языке Pascal.

Цель работы – изучение языка программирования Pascal на примере создания в его помощью видеоигры

Задачи:

- Изучить графический модуль, работу с файлами, чтение нажатия клавиш с клавиатуры.

- Рассмотреть процесс создания видеоигр.

- Создать видеоигру на основе разработанной идеи.

ГЛАВА 1. ОСНОВНЫЕ ЭТАПЫ СОЗДАНИЯ ВИДЕОИГР.

Большая часть людей даже понятия не имеют, насколько создание видеоигр трудоёмкий и долгий по времени процесс. Именно по этой причине создание игры делится на роли. Роли используются поэтапно. Я выделил три основных этапа создания видеоигры.

На первом этапе концепция идеи. Продюсер разрабатывает идею игры.

На втором – разработка алгоритма. Команда разработки пишет алгоритмы по реализации идеи.

Третьим и последним этапом является реализация алгоритмов в компьютерную программу. Команда разработки переписывает алгоритмы в код и исправляет все ошибки путём отладки.

ГЛАВА 2. КОНЦЕПЦИЯ ИДЕИ.

Создание видеоигры в начале производства требует написания концепции идеи. Концепция идеи – это расписанный план того о чём будет наша игра, что в ней будет происходить.

Я решил написать развивающую видеоигру для пользователей от 6 лет. Задача видеоигры заключается в том, чтобы пройти все уровни сложности с 3 до 16. Идея игры состоит в том, что у нас есть несколько сосудов, визуально делящихся на 4 равные ячейки, в которых содержатся жидкости имеющие свойство не перемешиваться. Особенность этих жидкостей в том, что они состоят из антиматерии, материи, которая существует под действием обратно пропорциональным законам физики. Если обычная жидкость при выливании, к примеру, из стакана льётся вниз, то жидкость из антиматерии наоборот, вверх. Переливание жидкостей происходит путём переливания из портала в портал, а также при использовании особых оборудований, способствующих стабильному переливанию. Всего два вида порталов. Синий и оранжевый. Синий – принимает жидкость, а оранжевый – выдаёт. Для прохождения уровня необходимо разлить все жидкости по цветам. С каждым следующим уровнем повышается количество цветов и количество сосудов на 1.

Моя видеоигра будет содержать несколько сцен и простой геймплей. Геймплей – это компонент игры отвечающий за взаимодействие игры и игрока.

Сцены:

0. Начальный экран (выход на 1 сцену).

Начальный экран будет первой открывшейся сценой при запуске игры. Эта сцена будет содержать крупную надпись “Нажми Enter чтобы начать”, после нажатия соответствующей клавиши происходит перемещению в меню, на сцену 1.

1. Меню игры (выход на 2, 3, 4 сцену).

Меню игры содержит в себе три строчки, одна из которых подсвечивается красным и показывает позицию выбора. Каждая строчка содержит в себе

соответствующую надпись: в первой строчке “ИГРАТЬ” (перемещает после выбора на сцену 2), во второй “УРОВЕНЬ СЛОЖНОСТИ” (перемещает после выбора на сцену 3), в третьей “ВЫХОД” (после выбора перемещает на сцену 1). Перемещение между строчками выбираются клавишами стрелка вверх и вниз. Для выбора необходимо нажать клавишу Enter, чтобы совершить перемещения в нужную сцену.

2. Игра (выход на 1 сцену).

Сцена «игра» визуально делится на три части. Первая треть содержит в себе две составляющие: в левом верхнем углу иконку с подписью «menu (esc)», означающая, что для выхода в меню необходимо нажать клавишу Esc, в центре расположена надпись, уведомляющая об уровне сложности. Вторая и третья часть сцены содержат сосуды, из которых будут перелиты жидкости.

Чтобы совершить переливание, необходимо выбрать сосуд, откуда сливаем, и сосуд, куда переливаем, а также выполнены два условия, что сосуд, в который сливаем, сверху имеет не заполненную ячейку, и что нижние ячейки имеют один и тот же цвет. Уровень считается законченным, когда остаются свободными две колбы, а все занятые содержат только один цвет. При окончании уровня выводится окно означающее, что уровень повышен и открывается следующий уровень сложности.

3. Настройка уровня сложности (выход на 1 сцену).

Сцена уровня сложности содержит в верхней части надпись “УРОВЕНЬ СЛОЖНОСТИ”, в центре номер уровня сложности, справа и слева от номера находятся стрелки. При нажатии на клавиши стрелки влево или вправо на клавиатуре, одна из стрелок загорается красным на долю секунды. Диапазон выбора уровня сложности начинается с 3 и до максимально пройденного уровня или пока не дойдёт до 16 уровня (максимального).

Игра считается полностью пройденной, когда открываются все уровни сложности.

ГЛАВА 3. НАПИСАНИЕ АЛГОРИТМОВ.

Следующим этапом после разработки идеи в создании видеоигры является написание алгоритма, по которому будет работать видеоигра. В основе моего алгоритма лежит простое перемещение ячеек двумерного массива с размерностью равной количеству сосудов с уровня на количество ячеек равным объёму сосуда, 4 единицы. Вместо цвета в массиве используется номер характерный соответствующему цвету. Колбы нумеруются слева направо, начиная с верхнего ряда.

Алгоритм процедуры переводящей нажатие клавиши в число:

В операторе выбора присваивается переменной определённое значение, зависящее от виртуального кода клавиши:

“VK_ESCAPE” - 27;

“VK_Enter” - 13;

“VK_Up” - 1;

“VK_Down” - 2;

“VK_RIGHT” - 3;

“VK_LEFT” - 4;

Алгоритм процедуры размешивания жидкости по сосудам:

Изначально все сосуды массива, отвечающие за колбы, разлиты по цветам. Затем происходит смешивание. Смешивание происходит путём получения двух разных рандомных значений от 1 и до количества сосудов. Одно число – номер сосуда, откуда сливают, а другое - куда. После полученных значений проверяется можно ли их перелить, то есть тот сосуд, куда переливаем должен содержать верхнюю ячейку пустой, а также оба сосуда должны иметь жидкости одного цвета в нижних частях. И только после этого происходит переливание между сосудами. Для усложнения прохождения уровня эти действия переливания зациклены и повторяются 700 раз.

Алгоритм процедуры, рассчитывающий положение сосудов:

Положение колб рассчитывается вначале уровня. Сначала

рассчитывается расстояние между колбами по формуле: k=(1900-(n+1-z)*120)div((n+1-z)+1), где k – искомое расстояние, n – номер последней колбы в ряду, z – номер первой колбы в ряду, 120 – ширина колбы. Далее в массив содержащий координаты верхней левой вершины сосуда, записывается значения по “x” формуле x=10+k*(i-z+1)+(i-z)*120), где I-номер колбы, 10 – отступ от края экрана, а по “y” номеру ряду, если первый ряд, то 350, иначе 700.

Алгоритм функции определяющей цвет по номеру:

Функции присваивается цвет по RGB кодировке, оператором выбора в зависимости от значения:

0.- RGB(233,150,122);

1. - RGB(0,204,255);

2. - RGB(51,0,255);

3. - RGB(204,51,0);

4. - RGB(153,255,0);

5. - RGB(153,153,0);

6. - RGB(153,102,51);

7. - RGB(153,204,51);

8. - RGB(204,0,51);

9. - RGB(153,255,51);

10. - RGB(204,51,102);

11. - RGB(153,102,153);

12. - RGB(153,204,153);

13. - RGB(204,0,153);

14. - RGB(153,153,204);

15. - RGB(153,255,204);

16. - RGB(204,51,153);

Алгоритм процедуры рисования сосуда с жидкостью с рамкой:

Сосуды с жидкостью создаются на экране путём повторения 4 раза

цикла из 2 процедур. Выбирается цвет жидкости функцией определяющей цвет по номеру, затем чертится прямоугольник по верхней левой и нижней правой вершине. Координаты вершины вычисляются по следующим формулам:

X1=xy[i,1]+10

Y1=xy[I,2]+20+(i-1)*60

X2=xy[i,1]+110

Y2 =xy[I,2]+80+(i-1)*60,

Где “x1”- координата первой вершины по “x”, а “y1” – координата первой вершины по “y”, соответственно “x2” – координата второй вершины по “x”, а “y2” - координата второй вершин по “y”, i – номер уровня сосуда, xy – массив с вершинами верхнего левого угла сосудов.

Алгоритм процедуры, рисования сосуда с жидкостью без рамки:

Сосуды с жидкостью создаются на экране путём повторения 4 раза цикла из 2 процедур. Выбирается цвет жидкости функцией определяющей цвет по номеру, затем чертится прямоугольник по верхней левой и нижней правой вершине. Координаты вершины вычисляются по следующим формулам:

X1=x+10

Y1=y+20+(i-1)*60

X2=x+110

Y2 =y+80+(i-1)*60,

Где “x1”- координата первой вершины по “x”, а “y1” – координата первой вершины по “y”, соответственно “x2” – координата второй вершины по “x”, а “y2” - координата второй вершин по “y”, i – номер уровня сосуда, “x” и “y” координаты вершины левого верхнего угла сосуда, полученные из массива содержащего верхние левые вершины сосуда.

Алгоритм процедуры построения и прохождения уровня, то есть самой игры:

Первым делом рисуется фон, затем идёт процедура расчёта положения

колб. Затем построения в первой трети, рисуются подсказки: выхода в меню и номер выбранного уровня сложности. Далее происходит запись верхней левой вершины для рисования сосудов процедурой рассчитывающей их положение. После этого идёт процедура, которая записывает и размешивает жидкость по сосудам. И только после этого начинается сама игра.

Перед каждой игрой происходит проверка, все ли сосуды разлиты по цветам, при положительном ответе игра заканчивается, выводится об этом надпись и, через пару секунд, переносит пользователя на сцену меню. В файл с сохранением записывает значение на единицу большее, чем было записано до прохождения уровня, если только само значение в файле равно пройденному уровню. А если проверка даёт отрицательный ответ об окончании уровня, то игра продолжается и идёт ряд процедур связанных с геймплеем. Если игра только началась, то под первым сосудом строится синий портал выхода, иначе там, где он стоял. Если переменная означающая номер сосуда, под которым стоит оранжевый портал, не равна нулю, то на том месте строится оранжевый портал приёма.

После происходит закрашивание синего портала и оранжевого, если он был построен. Затем начинается цикл, не прекращающийся до тех пор, пока процедурой переводящей нажатие клавиши в число, не будет выведено одно из нужных чисел. Оператор выбора, отталкиваясь от этого числа, задаёт следующие подпрограммы (для упрощения объяснения представим, что рассматриваем не положение порталов, а сосуд, под которым он находится):

1,2.-Перемещение вверх или вниз – Происходит расчёт нахождения положения сосуда относительно ряда и по результату определяется новое положение. Если номер сосуда больше, чем ((n div 2) + (n mod 2)), где n-количество всех колб:

- То значит, что находилась на верхнем ряду и переходит на нижний по формуле номер новой колбы = старый номер колбы + (n div 2).
- Иначе значит, что находилась на нижнем ряду и переходит

на верхний по формуле номер новой колбы =старый номер колбы - (n div 2).

3.-Перемещение вправо – Происходит определение ряда, на котором находится сосуд, путём продемонстрированным выше. Действия при значениях:

- Истинное – проверяется, является ли рассматриваемая колба, крайней в верхнем ряду, если так и есть, то номер становится равным 1, иначе увеличивается на единицу.
- Ложное - проверяется, является ли рассматриваемая колба, крайней в нижнем ряду, если так и есть, то номер становится равным ((n div 2) + (n mod 2) + 1), иначе увеличивается на единицу.

4.-Перемещение влево – происходит подобно перемещению влево. Разница лишь в том, что если колба не является крайней, 1 вычитается, а если оказывается крайней, то для верхнего ряда номер меняется на ((n div 2) + (n mod 2)), а для нижнего ряда на n.

13.- Выбор сосуда – Происходит проверка, выбрана ли колба, откуда будет происходить слив жидкости, если такая колба не выбрана, то тогда соответствующей переменной приравнивают значение номера колбы, куда будет происходить слив. Если это значение не равно нулю, то колба слива выбрана и происходит проверка, не выбрали ли одинаковые колбы, если номера колб равны, то номер колбы, откуда сливается, приравнивается к нулю, а если разные, то происходит проверка, перед тем как перелить. Проверяется возможность перелива, то есть пуста ли верхняя ячейка, куда переливаем, а также схожесть нижних цветов или отсутствия одного из них.

После всех операций, в целях проверки, проводится профилактическая процедура, которая проверяет, каждую колбу на наличие в нижней части пустого пространства и при нахождении такой сдвигает ячейки с жидкостью той самой колбы вниз на 1.

ГЛАВА 4. НАПИСАНИЕ ПРОГРАММЫ И ОТЛАДКА.

Последним этапом в создании видеоигры является написанием кода программы и отладка. Для написания кода выбирается язык программирования, который оптимальнее остальных подходит для выполнения поставленных задач. В моём случае подойдёт любой язык, в связи с тем, что требуемые задачи не требуют каких либо предпочтений в выполнении задачь. Для написания программы был выбран язык Pascal.

§1. Работа графического модуля.

Паскаль содержит в себе графический модуль, что позволяет сделать не просто игру, а видеоигру, игру с изображением картинки. Для работы с модулем Graph необходимо в разделе библиотек написать:

Uses GraphABC;

Для вызова окна графического модуля в самой программе прописываются две команды. Одна отвечает за область работы, другая за размер окна:

Определяет область изображения - Rectangle(10,10,1910,1000);

Открывает окно графического модуля в полный экран - Maximizewindow;

Графический модуль паскаля содержит все цвета имеющиеся в RGB кодировке. Поэтому можно считать, что Pascal не ограничен в выборе цвета. Цвет выбирается следующей процедурой:

RGB(R,G,B); {в ней вместо букв R,G и B стоят значения от 0 и до 255}

Процедура для очистки окна – clearwindow;

Процедура для закрытия окна - closewindow;

Основные процедуры для работы с графическим окном:

Цвет текущей кисти – SetBrushColor(c:Color);

Толщина текущего пера –SetPenWidth(d:DashStyle);

Цвет текущего пера – SetPenColor(c:Color);

Рисует заполненный прямоугольник по верхней левой и нижней правой вершине – Rectangle(x1,y1,x2,y2:Integer);

Заполняет внутренность прямоугольника по верхней левой и нижней правой вершине – FillRect(x1,y1,x2,y2:Integer);

Рисует заполненный эллипс ограниченный прямоугольником с заданными вершинами– Ellipse(x1,y1,x2,y2:Integer);

Рисует границу прямоугольника по заданным верхней левой и нижней правой вершине – DrawRectangle(x1,y1,x2,y2:Integer);

Выводит текст начиная с заданной верхней левой вершины – Textout(x1,y1:Integer; s:String);

Устанавливает цвет текущего шрифта – SetFontColor(c:Color);

Устанавливает размер текущего шрифта – Setfontsize(n:Size);

Заполняет внутренность прямоугольника с заданными вершинами верхнего левого и нижнему правому – FillRectangle(x1,y1,x2,y2:Integer);

Заливает область одного цвета, начиная с заданной точки – FloodFill(x1,y1:Integer; c:Color);

§2. Чтение нажатия клавиш с клавиатуры.

Чтение нажатия клавиш клавиатуры осуществляется путём записи виртуального кода в переменную. После чего оператор выбора присваивается соответствующее значение переменной, и в зависимости от этого значения далее осуществляется ряд команд, которые соответствует нажатой клавише.

§3. Работа с файлами.

Данное изучение поможет мне использовать сохранение уровней, чтобы сделать игру интереснее. Файлы в Pascal делятся на три типа: текстовые файлы, типизированные файлы, не типизированные файлы – все сразу типы использовать не к чему, подойдёт только текстовый файл. Поэтому на изучение берётся только он.

Работа с файлами происходит путём связи файла с переменной, после чего через эту переменную происходят нужные действия.

Файловая переменная описывается с помощью типа file of integer;

Пример описания:

```
Var
F:file of integer;
```

Перед действием с файлом необходимо связать его с переменной, которая будет отвечать за связь файла с Pascal. Осуществляется это процедурой assign(f:File of integer; Filename);

Открыть файл можно для двух целей и для каждой своя процедура. Для чтения, процедурой reset(f:Fileinteger), а для записи, процедурой rewrite(f:File of integer).

Процедуры, которые относятся к файлу открытому для чтения:

Read(f:File of integer; n:Integer); - записывает строчку файла f в переменную n.

Процедуры, которые относятся к файлу открытому для записи:

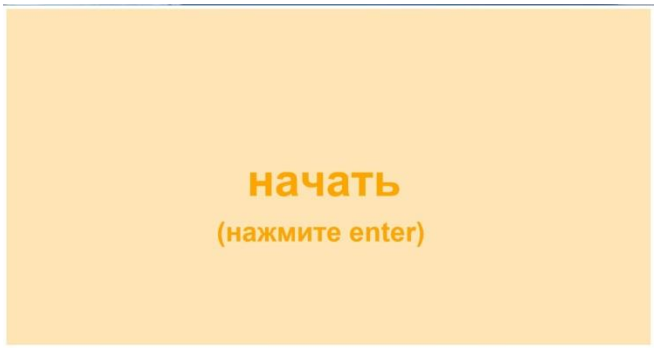
Write(f:File of integer; n:Integer) – записывает в файл f строчку n;

После окончания работы необходимо закрыть файл процедурой Close(f:Integer);

После написания всей программы необходимо проводить отладку, которая будет показывать, где и что не функционирует или функционирует не так как задумано. Отладку необходимо проводить пока не перестанут появляться ошибки, баги. В тот момент, когда программа видеоигры начинает полностью функционировать и безошибочно работать, можно считать, что создание видеоигры завершилось.

ГЛАВА 5. РЕЗУЛЬТАТ.

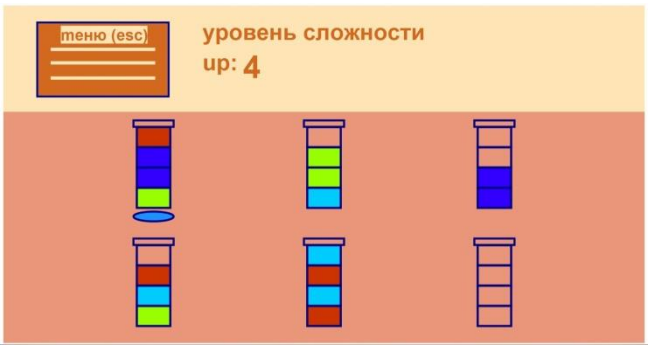
В ходе работы успешно написана программа, которая представляет собой полноценную видеоигру. Она по своему типу является развивающей и обучающей. Созданная мной видеоигра обучает тому, что все действия стоит рассчитывать, так как после каждого действия следуют свои последствия, например если в игре неправильно перелить жидкости, то возможно такое, что игру придётся начинать заново по причине отсутствия ходов. Играя в неё, происходит развитие таких навыков как тактика, критическое мышление, решение проблем, творчество, анализ, оценка и другие. Несмотря на свою простую идею и несложные алгоритмы, в неё можно играть достаточно долгое время, преодолевая уровень за уровнем, а время, затраченное на создание всей игры, не превышает трёх дней, включая в себя начало разработки идеи и заканчивая тестовым запуском. Ниже представлены снимки экранов из моей видеоигры.



Сцена 0, начальный экран.

Сцена 1, меню.

Сцена 3, уровень сложности.



Сцена 2, уровень(4).

ЗАКЛЮЧЕНИЕ.

Видеоигры один из самых увлекательных способов развития и проведения свободного времени. А само создание видеоигр ещё увлекательнее. Каждый человек, которые работает над созданием игры творческий и способен мыслить не стандартно, и чем эти навыки лучше, тем интереснее получится результат.

В процессе создания я не только научился реализовывать геймплей, работать с графикой и файлом, но и познакомился и изучил весь путь по её созданию, узнал множество нюансов и ощутил на себе роль каждой профессии, что нужна при создании видеоигры.

Моё желание пойти в IT сферу после этого проекта только усилилось, этот бесценный опыт мне пригодится при выборе конкретной профессии. Я буду дальше совершенствовать свои навыки в подобных проектах и возможно, что когда-нибудь смогу создать что-то такое, от чего мной будут восхищаться так же, как и я сейчас восхищаюсь проектами других людей и компаний.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Web-сайт справочник по PascalABC.Net - http://pascalabc.net/downloads/pabcnethelp/index.htm?page=PABCUnits/GraphABC/gr_FontClass.html
2. Страница с Web-сайта “Encyclopedia for Developers” с виртуальными кодами клавиш - https://api.farmanager.com/ru/winapi/virtualkeycodes.html

ПРИЛОЖЕНИЕ.

Код моей видеоигры на языке программирования PascalABC.NET:

```
uses GraphABC;

type mas=array[1..18,1..4]of integer;
     krdt=array[1..18,1..2]of integer;

var q,u,k,k1:integer;
    a:mas;
    xy:krdt;
    o:boolean;
    f: file of integer;

procedure menu(n:integer); forward;
procedure start; forward;
procedure klava(key:integer); forward;
procedure uslo; forward;
procedure play(n,KR,KL:integer); forward;
procedure kaps(x,y:integer); forward;
procedure radklb(n,y,z:integer; var xy:krdt); forward;
function CLR(g:integer):color; forward;
procedure RZBRS(n,p,r:integer); forward;
procedure CVT(i:integer); forward;

procedure klava(key:integer);
 begin
  case key of
   VK_ESCAPE: q:=27;
   VK_Enter: q:=13;
   VK_Up: q:=1;
```

```
   VK_Down: q:=2;
   VK_RIGHT: q:=3;
   VK_LEFT: q:=4;
   VK_Q: q:=5;
  end;
 end;

procedure start;
 begin
  q:=0;
  SetFontColor(clorange);
  SetBrushColor(clMoccasin);
  FillRect(10,10,1910,1000);
  SetFontSize(100);
  SetFontStyle(fsbold);
  TextOut(720,440,'начать');
  SetFontSize(60);
  TextOut(630,620,'(нажмите enter)');
  repeat
   onkeydown:=klava;
   sleep(50);
  until (q=13)or(q=27);
  clearwindow;
  if q=13
   then menu(1);
  closewindow;
 end;

procedure menu(n:integer);
 begin
```

```
n:=1;
assign(f,'c:\Users\user\file.txt');
if not fileexists('c:\Users\user\file.txt')
 then begin
       assign (f, 'c:\Users\user\file.txt');
       rewrite (f);
       write(f,2);
       close(f);
      end;
SetFontSize(80);
SetBrushColor(clMoccasin);
FillRect(10,10,1910,1000);
SetFontColor(clorange);
TextOut(750,400,'Играть');
TextOut(415,520,'уровень сложности');
TextOut(750,640,'выход');
repeat
 q:=0;
 case n of
  1: begin
      SetFontColor(clcrimson);
      TextOut(750,400,'Играть');
     end;
  2: begin
      SetFontColor(clcrimson);
      TextOut(415,520,'уровень сложности');
     end;
  3: begin
      SetFontColor(clcrimson);
      TextOut(750,640,'выход');
```

```
    end;
 end;
repeat
 sleep(50);
 onkeydown:=klava;
until ((q=13)or(q=1)or(q=2));
if q=1
 then begin
       case n of
        1: begin
            SetFontColor(clorange);
            TextOut(750,400,'Играть');
           end;
        2: begin
            SetFontColor(clorange);
            TextOut(415,520,'уровень сложности');
           end;
        3: begin
            SetFontColor(clorange);
            TextOut(750,640,'выход');
           end;
        end;
       case n of
        1: n:=3;
        2: n:=1;
        3: n:=2;
        end
       end
 else begin
       case n of
```

```
          1: begin
              SetFontColor(clorange);
              TextOut(750,400,'Играть');
             end;
          2: begin
              SetFontColor(clorange);
              TextOut(415,520,'уровень сложности');
             end;
          3: begin
              SetFontColor(clorange);
              TextOut(750,640,'выход');
             end;
         end;
        if q=2
         then case n of
               1: n:=2;
               2: n:=3;
               3: n:=1;
               end;
        end;
  until ((q=13));
  clearwindow;
  case n of
   3: start;
   2: uslo;
   1: play(u+2,0,0);
   end;
 end;

procedure uslo;
```

```
begin
 repeat
  q:=0;
  SetFontColor(clorange);
  SetBrushColor(clMoccasin);
  FillRect(10,10,1910,1000);
  SetFontSize(170);
  TextOut(880-(60*(u div 10)),370,u);
  SetFontSize(80);
  TextOut(415,220,'Уровень сложности');
  SetFontSize(120);
  TextOut(710,700,'=');
  TextOut(1100,700,'=');
  TextOut(660,700,'<');
  TextOut(1155,700,'>');
  repeat
   onkeydown:=klava;
   sleep(50);
  until (q=13)or(q=3)or(q=4);
 assign(f,'c:\Users\user\file.txt');
 reset(f);
 while not Eof(f) do
  read(f,k1);
 close(f);
  if q=3
   then begin
         if (u<16)and(u<=k1)
          then begin
                u:=u+1;
               end;
```

```
          SetFontColor(clcrimson);
          TextOut(1155,700,'>');
          TextOut(1100,700,'=');
          sleep(100);
         end
    else if q=4
         then begin
              if (u>3)
               then u:=u-1;
              SetFontColor(clcrimson);
              TextOut(660,700,'<');
              TextOut(710,700,'=');
              sleep(100);
             end;
   clearwindow;
  until (q=13);
  menu(0);
 end;

procedure kaps(x,y:integer);
 begin
  DrawRectangle(x,y,x+120,y+20);
  DrawRectangle(x+10,y+20,x+110,y+80);
  DrawRectangle(x+10,y+80,x+110,y+140);
  DrawRectangle(x+10,y+140,x+110,y+200);
  DrawRectangle(x+10,y+200,x+110,y+260);
 end;

procedure play(n,KR,KL:integer);
 begin
```

```
SetBrushColor(clMoccasin);
FillRect(10,10,1910,1000);
SetBrushColor(cldarksalmon);
FillRect(10,325,1910,1010);
SetBrushColor(clMoccasin);
SetPenWidth(6);
SetPenColor(clnavy);
radklb((n div 2)+(n mod 2),350,1,xy);
radklb(n,700,(n div 2)+(n mod 2)+1,xy);
DrawRectangle(110,60,500,280);
FloodFill(115,65,clchocolate);
FillRectangle(150,135,460,145);
FillRectangle(150,175,460,185);
FillRectangle(150,220,460,230);
SetFontSize(35);
SetFontColor(clchocolate);
TextOut(180,70,'меню (esc)');
setfontsize(50);
Textout(600,50,'уровень сложности');
Textout(600,140,'up:');
setfontsize(70);
Textout(720,140,u);
for var i:=1 to n do
 begin
  kaps(xy[i,1],xy[i,2]);
 end;
KR:=1;
KL:=0;
RZBRS(n,0,0);
repeat
```

```
o:=true;
for var i:=1 to n do
 begin
  if (a[i,1]=a[i,2])and(a[i,2]=a[i,3])and(a[i,3]=a[i,4])
   then
   else o:=false;
  CVT(i);
 end;
if o=true
 then begin
       setfontsize(130);
       SetFontColor(clchocolate);
       SetBrushColor(clMoccasin);
       DrawRectangle(200,100,1500,330);
       Textout(200,100,'Молодец(LVL. up)');
       assign(f,'c:\Users\user\file.txt');
       rewrite(f);
       write(f,k1);
       if u=k1
        then
         begin
         reset(f);
         while not Eof(f) do
          read(f,k1);
         k1:=k1+1;
         u:=u+1;
         close(f);
         assign (f, 'c:\Users\user\file.txt');
         rewrite (f);
         write(f,k1);
```

```
            end;
           close(f);
           DrawRectangle(200,100,1800,330);
           sleep(3600);
           menu(0);
          end;
        SetBrushColor(cldodgerblue);
        Ellipse(xy[KR,1],xy[KR,2]+270,xy[KR,1]+120,xy[KR,2]+300);
        if KL<>0
         then begin
           SetBrushColor(clorangered);
           Ellipse(xy[KL,1],xy[KL,2]+270,xy[KL,1]+120,xy[KL,2]+300);
          end;
        q:=0;
        repeat
         onkeydown:=klava;
         sleep(50);
        until (q=1)or(q=27)or(q=2)or(q=3)or(q=4)or(q=13);
        SetBrushColor(cldarksalmon);
        FillRect(xy[KR,1]-5,xy[KR,2]+270-5,xy[KR,1]+120+5,xy[KR,2]+300+5);
        if KL<>0
         then                                    fillRect(xy[KL,1]-5,xy[KL,2]+270-
5,xy[KL,1]+120+5,xy[KL,2]+300+5);
        case q of
        1: begin
           if KR>((n div 2)+(n mod 2))
            then KR:=KR-(n div 2)
            else KR:=KR+(n div 2);
          end;
        2: begin
```

```
    if KR>((n div 2)+(n mod 2))
     then KR:=KR-(n div 2)
     else KR:=KR+(n div 2);
   end;
3: begin
    if KR>((n div 2)+(n mod 2))
     then if KR<n
          then KR:=KR+1
          else KR:=((n div 2)+(n mod 2))+1
     else if KR<((n div 2)+(n mod 2))
          then KR:=KR+1
          else KR:=1;
   end;
4: begin
    if KR>((n div 2)+(n mod 2))
     then if KR>((n div 2)+(n mod 2))+1
          then KR:=KR-1
          else KR:=n
     else if KR>1
          then KR:=KR-1
          else KR:=((n div 2)+(n mod 2));
   end;
5: begin
    RZBRS(u+2,0,0);
   end;
13: begin
     if KL=0
      then KL:=KR
      else if KL=KR
           then KL:=0
```

```
            else begin
                 if ((a[KL,4]=a[KR,4])or(a[KR,4]=0))and(a[KR,1]=0)
                  then begin
                       a[KR,1]:=a[KR,2];
                       a[KR,2]:=a[KR,3];
                       a[KR,3]:=a[KR,4];
                       a[KR,4]:=a[KL,4];
                       a[KL,4]:=a[KL,3];
                       a[KL,3]:=a[KL,4];
                       a[KL,4]:=0;
                       KL:=0;
                      end;
                end;
      end;
   end;
   for var i:=1 to n do
    if a[i,4]=0
     then begin
          a[i,4]:=a[i,3];
          a[i,3]:=a[i,2];
          a[i,2]:=a[i,1];
          a[i,1]:=0;
         end;
  until q=27;
  if q=27
   then menu(0);
 end;

procedure radklb(n,y,z:integer; var xy:krdt);
 begin
```

```
  k:=(1900-(n+1-z)*120) div ((n+1-z)+1);
  for var i:=z to n do
   begin
    xy[i,1]:=10+k*(i-z+1)+(i-z)*120;
    xy[i,2]:=y;
   end;
 end;

procedure CVT(i:integer);
 begin
  SetBrushColor(CLR(a[i,1]));
  Rectangle(xy[i,1]+10,xy[i,2]+20,xy[i,1]+110,xy[i,2]+80);
  SetBrushColor(CLR(a[i,2]));
  Rectangle(xy[i,1]+10,xy[i,2]+80,xy[i,1]+110,xy[i,2]+140);
  SetBrushColor(CLR(a[i,3]));
  Rectangle(xy[i,1]+10,xy[i,2]+140,xy[i,1]+110,xy[i,2]+200);
  SetBrushColor(CLR(a[i,4]));
  Rectangle(xy[i,1]+10,xy[i,2]+200,xy[i,1]+110,xy[i,2]+260);
 end;

function CLR(g:integer):color;
 begin
  case g of
   0: CLR:=cldarksalmon;
   1: CLR:=RGB(0,204,255);
   2: CLR:=RGB(51,0,255);
   3: CLR:=RGB(204,51,0);
   4: CLR:=RGB(153,255,0);
   5: CLR:=RGB(153,153,0);
   6: CLR:=RGB(153,102,51);
```

```
   7: CLR:=RGB(153,204,51);
   8: CLR:=RGB(204,0,51);
   9: CLR:=RGB(153,255,51);
   10: CLR:=RGB(204,51,102);
   11: CLR:=RGB(153,102,153);
   12: CLR:=RGB(153,204,153);
   13: CLR:=RGB(204,0,153);
   14: CLR:=RGB(153,153,204);
   15: CLR:=RGB(153,255,204);
   16: CLR:=RGB(204,51,153);
  end;
 end;

procedure RZBRS(n,p,r:integer);
 begin
  for var i:=1 to n-2 do
   begin
    a[i,1]:=i;
    a[i,2]:=i;
    a[i,3]:=i;
    a[i,4]:=i;
   end;
  for var i:=n-1 to n do
   begin
    a[i,1]:=0;
    a[i,2]:=0;
    a[i,3]:=0;
    a[i,4]:=0;
   end;
  for var j:=1 to 7000 do
```

```
  begin
   p:=random(n-1)+1;
   r:=random(n-1)+1;
   if (a[p,1]=0)and(p<>r)
    then begin
          a[p,1]:=a[p,2];
          a[p,2]:=a[p,3];
          a[p,3]:=a[p,4];
          a[p,4]:=a[r,4];
          a[r,4]:=a[r,3];
          a[r,3]:=a[r,2];
          a[r,2]:=a[r,1];
          a[r,1]:=0;
         end;
  end;
 end;

begin
 Rectangle(10,10,1910,1000);
 MaximizeWindow;
 u:=3;
 start;
 closewindow;
end.
```