**Введение**

На курсовое проектирование была поставлена задача, разработать игровой программный продукт «Warriors of the willage».

Цель курсового проекта заключается в создании игрового приложения для развлечения и отдыха. Главной задачей при разработке ПО является проработка пошагового боя и исключение патовых ситуаций во время игры.

Создаваемая программа будет рассчитана на любого рода пользователя. Она рассчитана на однопользовательское прохождение всех уровней, требующих от игрока использования тактики боя и правильного использования ресурсов.

Разрабатываемая игра является представителем жанра JRPG и примером при разработке некоторых моментов являлись игры этого жанра, такие как Final Fantasy, Dragoт Quest.

Далее приведем краткое описание разделов пояснительной записки.

Первый раздел носит название «Анализ задачи». В нем вы сможете ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, определение ее организационно-экономической функции. В подразделе «Инструменты разработки» будет рассмотрена среда, в которой разрабатывается данный курсовой проект. Здесь также будут установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающих правильное функционирование поставленной задачи.

В разделе «Проектирование задачи» будут рассмотрены основные аспекты разработки программного продукта. Здесь можно будет узнать об организации данных в контексте среды разработки. В данном разделе будет описан пользовательский интерфейс, составлены алгоритмы процесса обработки информации, описана разработка системы справочной информации.

В разделе «Реализация задачи» описываются элементы и объекты, которые будут использоваться при реализации данного программного продукта. Тут будут описаны функции пользователя и их структура.

Следующий раздел – «Тестирование». В нем будет описано полное и функциональное тестирование данной программы. Будут смоделированы все возможные действия пользователя при работе с программой.

В разделе «Применение» будет описано назначение, область применения, среда функционирования программы.

«Заключение» будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В «Литературе» будет приведен список используемых при разработке источников.

В приложении к пояснительной записке будет приведен листинг программы с необходимыми комментариями.

Схема работы системы будет представлена в графической части.

**1 Анализ задачи**

* 1. **Постановка задачи**

Темой курсового проекта является «Разработка игрового приложения в жанре RPG «Warrios of the willage»».

Данный программный продукт разрабатывается для людей любого возраста с целью отдыха и развлечения, а также развития стратегического мышления.

Программа будет использоваться по усмотрению пользователя. В игре предусмотрено сохранение данных и при выходе из игры все достижения останутся.

Аналогами данной игры являются другие представители жанра RPG, такие как Final Fantasy и Dragon Quest.

Целью игры является пройти по 10 уровней на всех 3 локациях. На каждом уровне будет 3 раунда, на каждый из раундов рандомно генерируются от 1 до 6 врагов. Весь бой проходит в пошаговом режиме. Сначала ходят все персонажи поочередно, каждый из них использует умение для атаки одного врага или использует бутылек восстанавливающий HP или MP. Умения тратят MP, поэтому лучше следит за их количеством. После того, как все персонажи сделают ход, идет очередь атаки врагов. Они при атаке уменьшают количество HP персонажа и если оно спустится до 0, то персонаж не сможет больше ходить на этом уровне. Если персонажи уничтожат всех врагов на уровне, то уровень считается пройденным.

Разрабатываемый программный продукт должен позволять выполнять следующие действия:

* просмотр правил игры;
* выбор локации;
* прохождение уровня (пошаговый бой);
* сохранение прогресса.

В разрабатываемой программе будут использоваться три вида данных: к входной информации можно отнести выбор пользователем локации и действия персонажей. К выходной – последствия действий, выбранных пользователем. Постоянной информацией в проекте будут являться картинки, музыка и данные о персонажах и монстрах.

Программный продукт предоставляет функционал только для одного пользователя (игрока).

* 1. **Инструменты разработки**

Для разработки данного проекта будет выбрана среда **Delphi** (RAD Studio 10.4) так как это одна из мощнейших сред для разработки приложений такого уровня. Delphi— это среда объектно-ориентированного программирования, относящаяся к классу RAD – (Rapid Application Development – «Средство быстрой разработки приложений»), основанная на языке Object Pascal. Используется для разработки и поддержки программного обеспечения. В отличии от своего прародителя, языка Pascal, многофункционален и удобен тем, что имеет визуальный редактор приложения, который позволяет создавать внешний облик будущей программы. Приложения Windows для создания которых требовалось большое количество человеческих усилий теперь могут быть написаны одним человеком, использующим Delphi.

Для разработки интерфейса я использовал программы Photoshop и Gale. Для создания Setup буду использовать Smart Install Maker. Для создания справки буду использовать html help worker.

Игровое приложение разрабатывается на компьютере с характеристиками:

* процессор AMD RYZEN 5;
* объем ОЗУ 8Gb;
* объем места на SSD 500Gb;
* видеоподсистема 1920/1080;
* ОС – Windows 10.
  1. **Требования к приложению**

**Требования к интерфейсу:** данное игровое приложение делается с оглядкой на 1990-2000 г. и игры того времени, поэтому основным стилем является PixelArt. Таким образом, шрифт, картинки и музыка подбирается в соответствии с играми того времени, но не забывая про удобство и простоту интерфейса.

**Требования к надежности:** специальных требований к надежности не предъявляется.

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы программы вследствие некорректных действий пользователя при взаимодействии с программой через графический интерфейс не должны влиять на конечный результат.

**2 Проектирование задачи**

**2.1 Организация данных**

Проектирование задачи – это очень важный этап в разработке любого приложения.

Необходимым условием при разработке данного приложения является описание организации данных, т.е. логическая и физическая структура данных в контексте среды разработки. В разрабатываемой программе будут использоваться три вида данных.

Первым видом являются данные, которые будут введены разработчиком на этапе реализации задачи. Сюда можно отнести изображения (иконки), описание, исходные коды сортировок. Также этими данными являются характеристики персонажей и их навыки. Характеристики прописаны в отдельном типизированном файле, а навыки прописаны в коде боя. Их количество HP и MP заносятся из типизированного файла в два массива и используются в коде боя.

Вторым видом данных, используемых в программе, является вводимая пользователем информация. Входной информацией будет являться выбор пользователем локации, умения, цели атаки. Все это он будет выбирать при помощи мышки. При выборе пользователем определенного действия, данные автоматически будут заноситься в переменные.

Третьим видом данных является результат программы – загрузка локации, убийство врага, прохождение уровня. Его также относят к отдельному виду, так как ни пользователь, ни разработчик его не вводят, а программа сама получает его в результате выполнения определенных действий.

**2.2 Процессы**

При входе в игру пользователь может начать новую игру, это удалит все сохранения, если они были, или он может продолжить игру, тогда загрузится его прогресс и он сможет продолжить прохождение игры.

Далее он переходит на форму, где он может выбрать локацию, на которую он хочет отправиться. На выбор будет дано 3 локации.

После выбора локации, пользователь переходит на форму «боя», на которой ему необходимо выбирать действия персонажей. Процесс боя проходит пошагово.

Пользователю необходимо выбрать умение персонажа и цель, которую нужно атаковать, после этого он нажимает на кнопку, отвечающую за ход. Алгоритм хода описан ниже в блок-схеме ниже (рисунок 1):

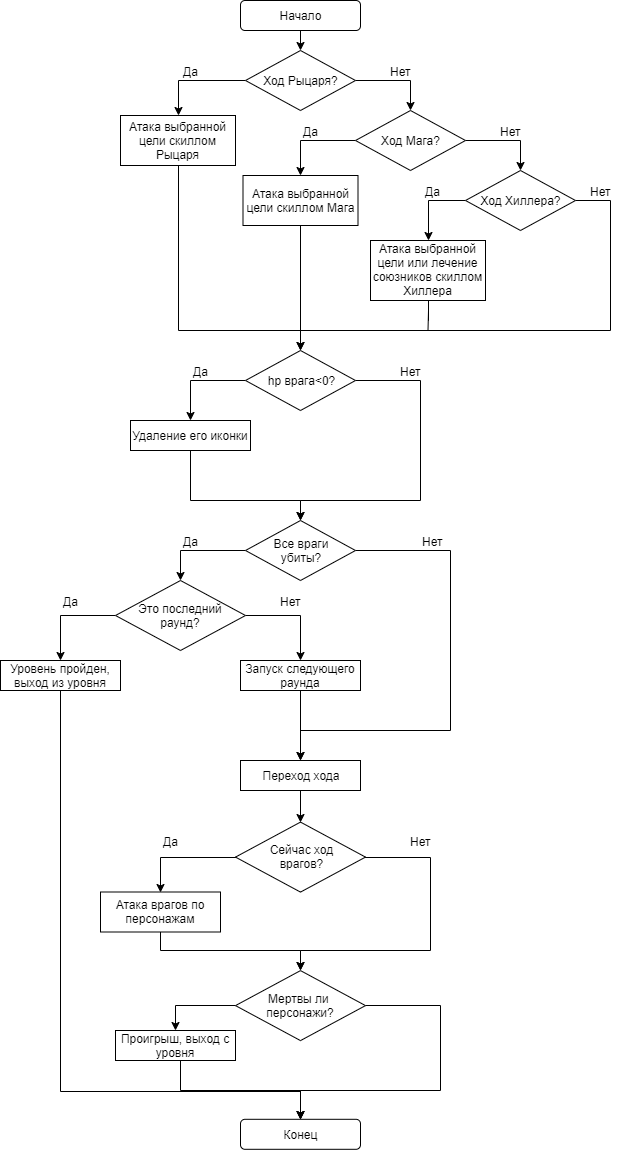


Рисунок 1 – Основной алгоритм боя

**2.3 Описание внешнего пользовательского интерфейса**

Для удобного пользования игровым приложением, интерфейс будет довольно простым и понятным. Его особенностью будет стилизация под графику 1990 г. Он не должен вызвать у пользователя затруднения при пользовании приложением.

Под управление игрока будут отданы 3 персонажа, которые имеют свои особенности и навыки, Рыцарь (Knight), Волшебник (Wizzard), Хиллер (Hiller). Они различаются количеством HP и MP, а также используют различные навыки при бое.

Ниже представлены примеры пользовательского интерфейса:

1. Главное меню (рисунок 2)
2. Меню выбора локации (рисунок 3)
3. Окно боя (рисунок 4)

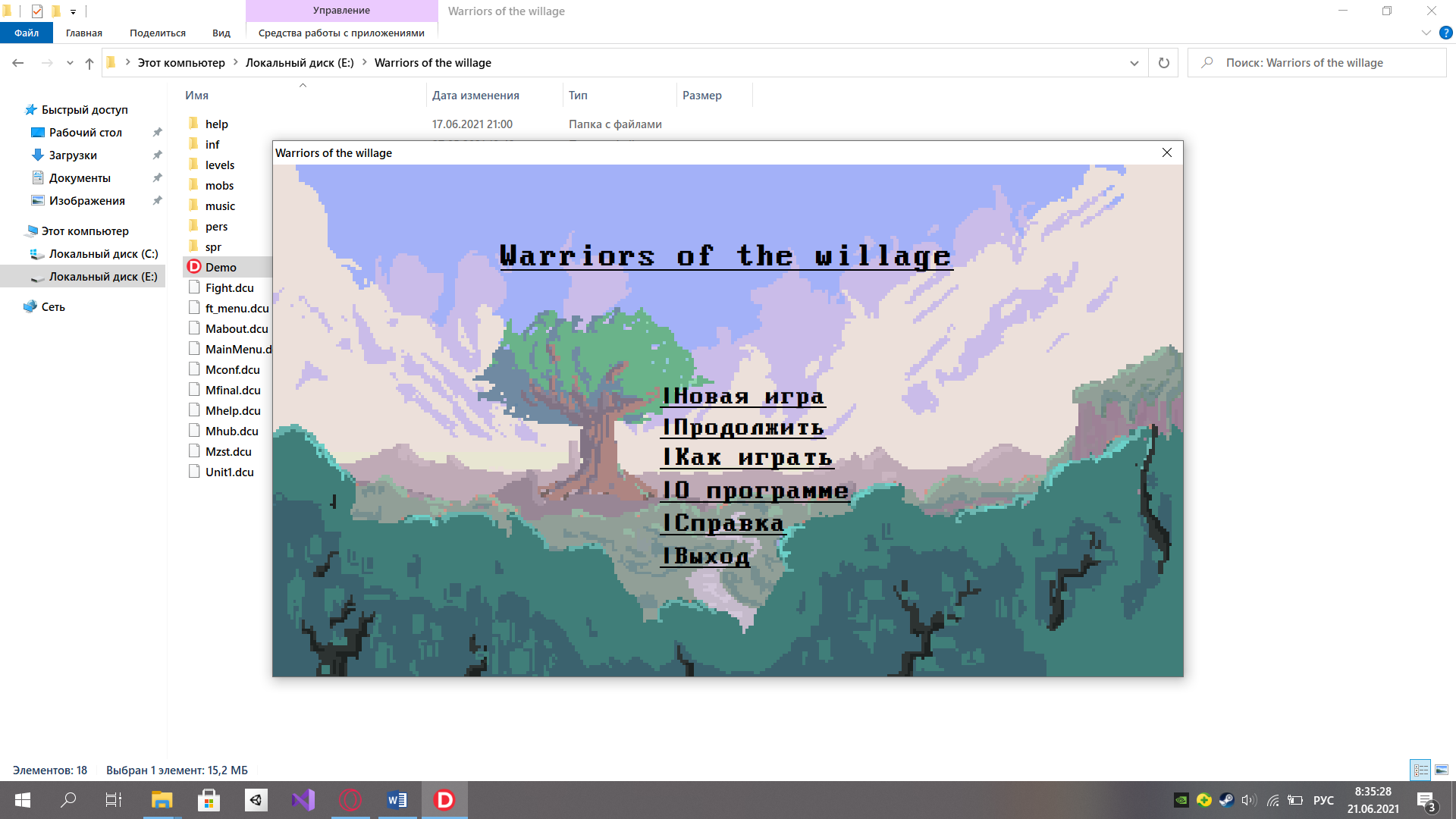


Рисунок 2 – Главное меню

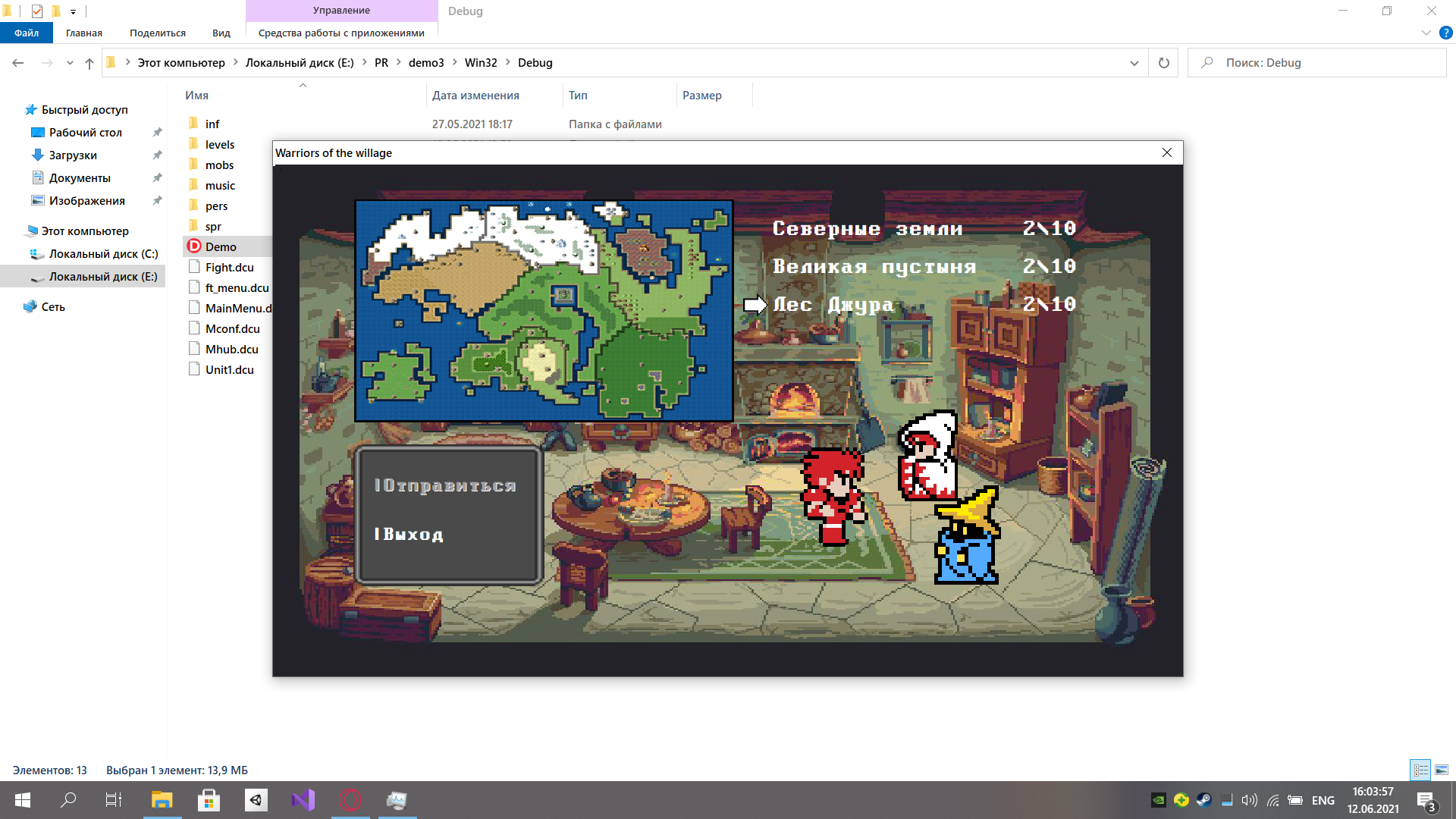


Рисунок 3 – Меню выбора локации

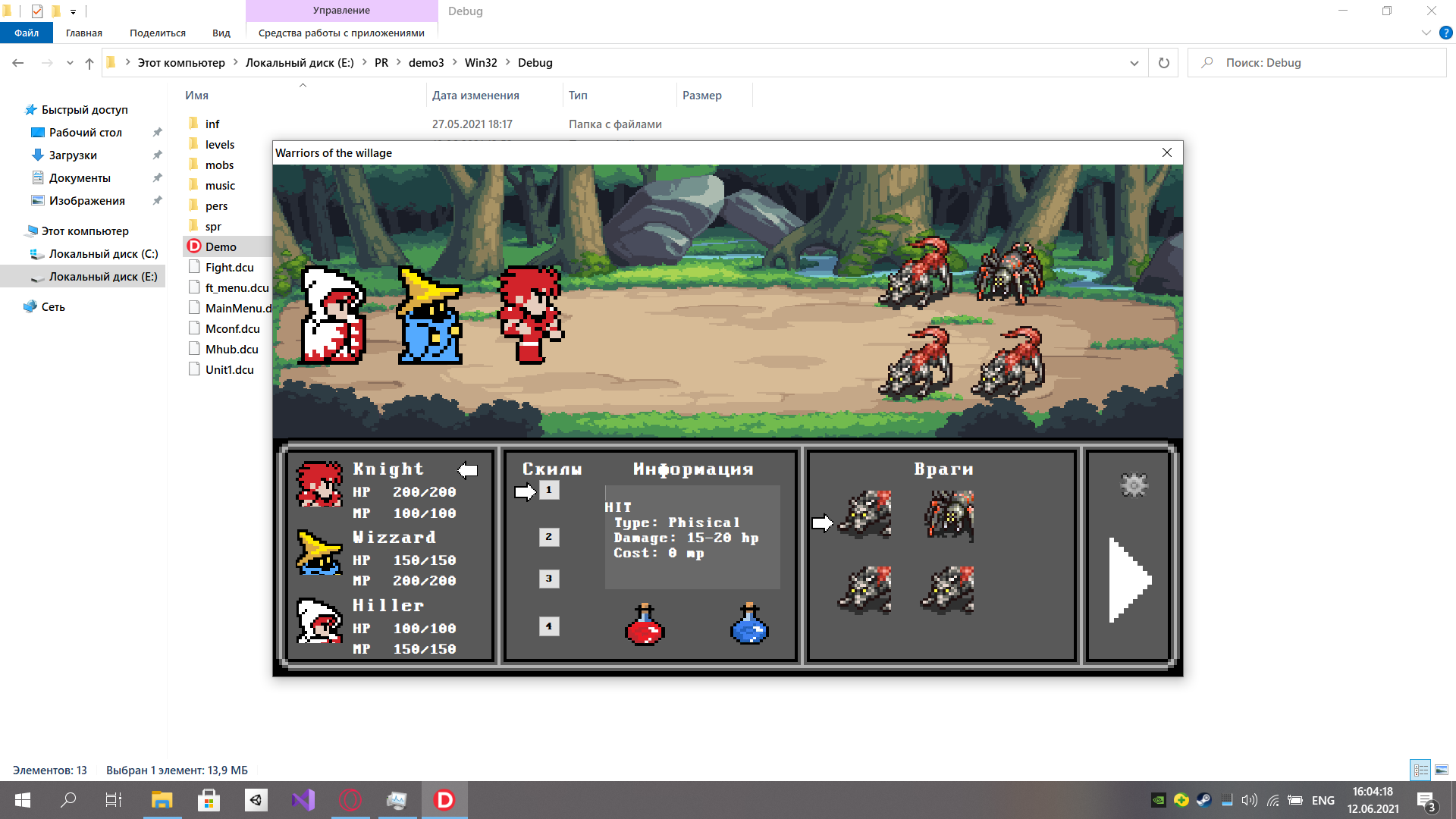


Рисунок 4 – Окно боя

**3 Реализация**

**3.1 Структура программы**

Данный курсовой проект состоит из 6 модулей. Далее будет описано назначение каждого из них.

Модуль MainMenu является модулем, содержащим главное меню программы.

Модуль Mconf является модулем, содержащим форму для подтверждения новой игры в главном меню.

Модуль Mhub является модулем, содержащим форму для выбора локации и отображения пройденных уровней.

Модуль Fight – модуль, содержащий форму боя.

ft\_menu – модуль, содержащий небольшое меню, которое можно активировать в предыдущей форме.

Mfinal – форма означающая финал игры.

Модуль Mhelp содержит окно «Правила игры». его запуск происходит с главного меню.

Mabout – форма содержащая пункт О программе.

Модуль Mzst является заставкой, которая запускается при запуске приложения.

**3.1.1 Структура и описание процедур и функций пользователя**

Описание процедур и функций приводится в таблице 1.

Таблица 1 – Процедуры и функции.

|  |  |  |  |
| --- | --- | --- | --- |
| Имя процедуры или функции | В каком модуле находится | За каким элементом управления закреплены | Назначение |
| procedure TMM.FormHide(Sender: TObject); | MainMenu, Mhub, Fight | MM | Отключение музыки |
| rocedure TMM.ngameClick(Sender: TObject); | MainMenu | ngame | Запуск подтверждения новой игры |
| procedure TMM.playClick(Sender: TObject); | MainMenu | play | Продолжения прохождения игры |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| Имя процедуры или функции | В каком модуле находится | За каким элементом управления закреплены | Назначение |
| procedure TMM.lhelpClick(Sender: TObject); | MainMenu | lhelp | Открывает форму с правилами игры |
| procedure Tconf.Label1Click(Sender: TObject); | Mconf | Label1 | Перезаписывает файл с сохранением для новой игры |
| procedure Thub.FormShow(Sender: TObject); | Mhub | hub | Вывод информации о пройденных уровнях и сохранение пройденных уровней |
| procedure Thub.lmap1Click(Sender: TObject); | Mhub | lmap1 | Выбор локации для прохождения уровня |
| procedure Thub.playClick(Sender: TObject); | Mhub | play | Запускает уровень на выбранной локации |
| procedure Thelp.menuClick(Sender: TObject); | Mhelp | menu | Показывает информацию о выбранной части игры |
| procedure heroes\_stats(index : integer); | Fight | ft | Выводит информацию о персонажах из файла |
| procedure Tft.en\_icon1Click(Sender: TObject); | Fight | en\_icon1 | Выбор цели атаки |
| procedure Tft.FormShow(Sender: TObject); | Fight | ft | Запуск музыки, загрузка фона локации, генерация мобов |
| procedure Tft.ftm\_imgClick(Sender: TObject); | Fight | ftm\_img | Открытие меню во время боя |
| procedure Tft.nextClick(Sender: TObject); | Fight | next | Код боя, описан в разделе 2 картинка 1 |
| procedure Tft.skill1Click(Sender: TObject); | Fight | skill1 | Выбор умения для атаки |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| Имя процедуры или функции | В каком модуле находится | За каким элементом управления закреплены | Назначение |
| procedure Tftm.contClick(Sender: TObject); | Ft\_menu | cont | Выход из боя, до его окончания |
| procedure Tfinalf.Label4Click(Sender: TObject); | Mfinal | Label4 | Выход из формы финала игры |

**3.1.2 Описание использованных компонентов**

Описание, использованных при разработке приложения, компонентов приводится в таблице 2.

Таблица 2 – Использованные компоненты

|  |  |  |
| --- | --- | --- |
| Компонент | На какой форме расположен | Назначение |
| TMainMenu | MM | Используется для создания главного меню |
| TImage | zst, MM, help, hub, ft, conf, ftm, finalf | Используется для размещения на форме изображений |
| TLabel | zst, MM, help, hub, ft, conf, ftm, finalf | Используется для отображения надписей на форме |
| TMemo | ft | Используется для отображения текста на форме |
| TTimer | MM,hub,ft,zst | Позволяется задавать интервалы времени |

**3.2 Спецификация программы**

Точное название проекта и его состав приводится в таблице 3.

Таблица 3 – Спецификация программы

|  |  |
| --- | --- |
| Имя файла | Назначение |
| Demo.exe | Исполняемый файл проекта, используется для запуска приложения |
| help | Папка и файлами, которые используются в форме help |
| inf | Папка с файлами, которые используются в форме ft |
| levels | Папка с файлами, которые используются для сохранения |

Продолжение таблицы 3

|  |  |
| --- | --- |
| Имя файла | Назначение |
| mobs | Папка с файлами, которые используется для генерации врагов |
| music | Папка с файлами, которые используется для воспроизведения фоновой музыке |
| pers | Палка с файлами, которые используются для занесения информации о персонажах |
| spr | Папка с файлами, хранящая изображения для форм |
| Fight.pas | Файл программного модуля для формы боя |
| ft\_menu.pas | Файл программного модуля для формы меню для боя |
| MainMenu.pas | Файл программного модуля для формы главного меню |
| Mconf.pas | Файл программного модуля для формы подтверждения новой игры |
| Mfinal.pas | Файл программного модуля для формы финала игры |
| Mhelp.pas | Файл программного модуля для формы с правилами игры |
| Mhub.pas | Файл программного модуля для формы выбора локации |
| Mzst.pas | Файл программного модуля для формы заставки |
| Mabout.pas | Файл программного модуля для формы «О программе» |
| Sеtup.exe | Установочный файл |
| About.chm | Файл справки |
| Fight.dfm | Форма боя |
| ft\_menu.dfm | Форма меню для боя |
| MainMenu.dfm | Форма главного меню |
| Mconf.dfm | Форма подтверждения новой игры |
| Mfinal.dfm | Форма финала игры |
| Mhelp.dfm | Форма с правилами игры |
| Mhub.dfm | Форма выбора локации |
| Mzst.dfm | Форма заставки |
| Mabout.dfm | Форма «О программе» |

**4 Тесты на использование**

В ходе разработки программы были составлены тесты, которые необходимо выполнить в дальнейшем. Тесты составлены таким образом, чтобы предусмотреть максимум возможных действий. Тесты описаны в таблице 4.

Таблица 4 – Тесты на использование

|  |  |  |
| --- | --- | --- |
| Модуль приложения | Суть теста | Ожидаемый результат |
| Главное меню | Проверка всех пунктов Главного меню | Открытие необходимых форм и их закрытие |
| Форма выбора локации | Проверка считывания файла сохранения | Если были пройдены уровни, то они будут отображены на форме |
| Форма выбора локации и форма боя | Проверка подгрузки фона на форме боя, в соответствии с выбранной локацией | Подгрузка фона соответствующей локации |
| Форма боя | Несколько раз перезапустить форму боя для проверки рандомного генерирования врагов | Изменение в количестве и видах врагов |
| Форма боя | Проверка перемещения указателей при выборе умения и врага | Перемещение указателей в нужные места, для указания выбранного умения или врага |
| Форма боя | Проверка вывода информации об умении | Вывод информации о каждом умении всех персонажей |
| Форма боя | Проверка нанесения урона врагам, в зависимости от выбранного умения | Нанесение врагам урона |
| Форма боя | Убийство врагов при уменьшении их HP до 0 | Убийство врагов после нанесения определенного количества урона |
| Форма боя | Нанесение урона персонажам | Ход врагов и нанесение урона персонажам в зависимости от количества врагов |
| Форма боя | Проверка смерти персонажей | Смерть персонажей при нанесении определенного количества урона |
| Форма боя | Проверка прохождение уровня | Выход с формы боя при убийстве всех врагов |
| Форма финала | Проверка запуска финальной формы при прохождении всех уровней | Запуск формы и обнуление результатов |

По итогам тестирования все проблемы были исправлены и все работает исправно. Программа может считаться полностью готовой к использованию.

**5 Руководство пользователя**

**5.1 Общие сведения о программном продукте**

Разработанная программа является игровым приложением для развлечения и отдыха «Warriors of the willage». Она предназначена для любителей пошаговых стратегий и RPG. Сделана в стиле игр 90-х годов.

**5.2 Инсталяция**

Для установки приложения необходимо запустить файл Setup.exe. Появиться окно установки приложения «Warriors of the willage» (рисунок 5).

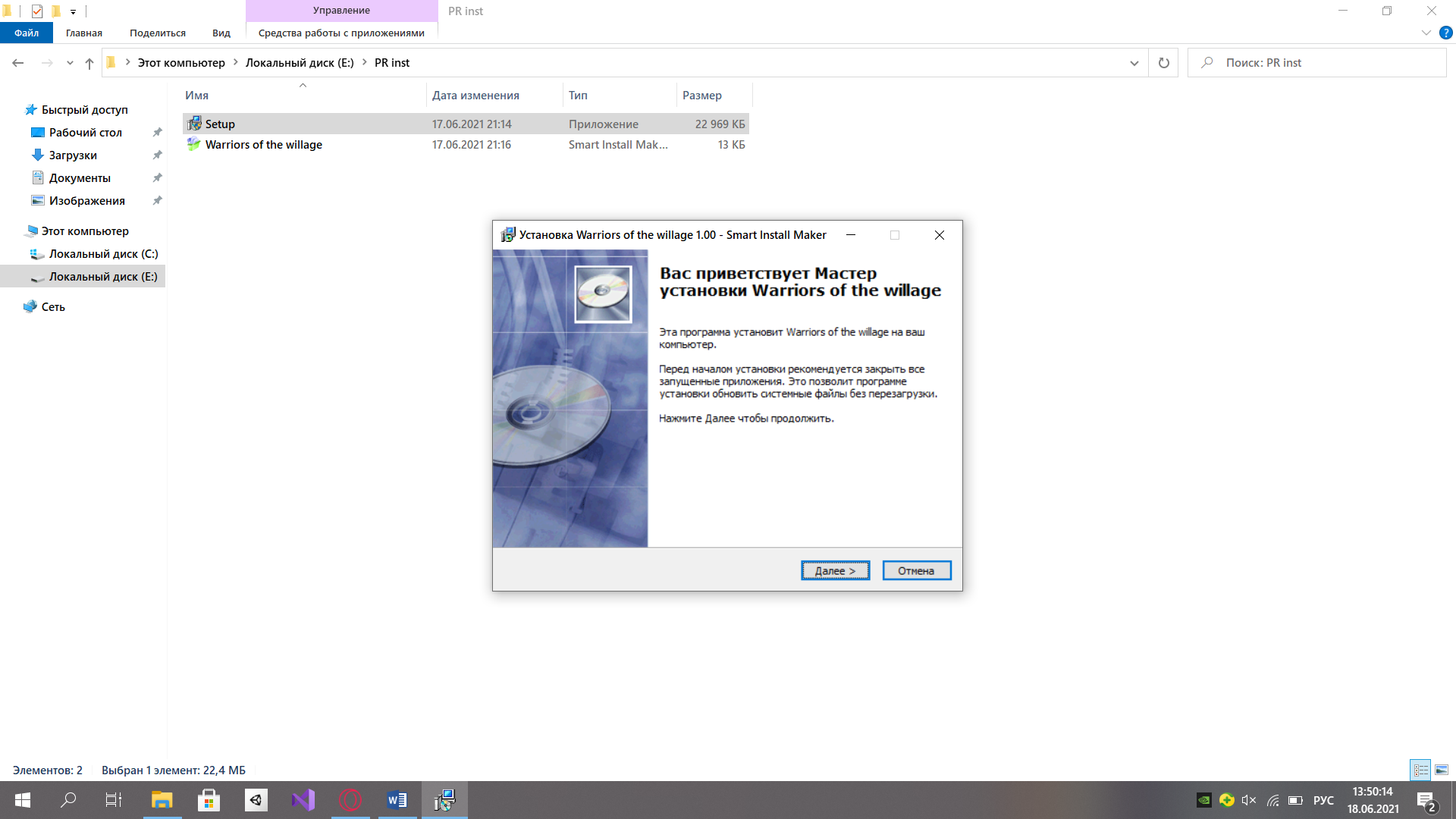


Рисунок 5 – Установка приложения «Warriors of the willage»

После нажатия на кнопку Далее, необходимо выбрать путь установки приложения (рисунок 6).

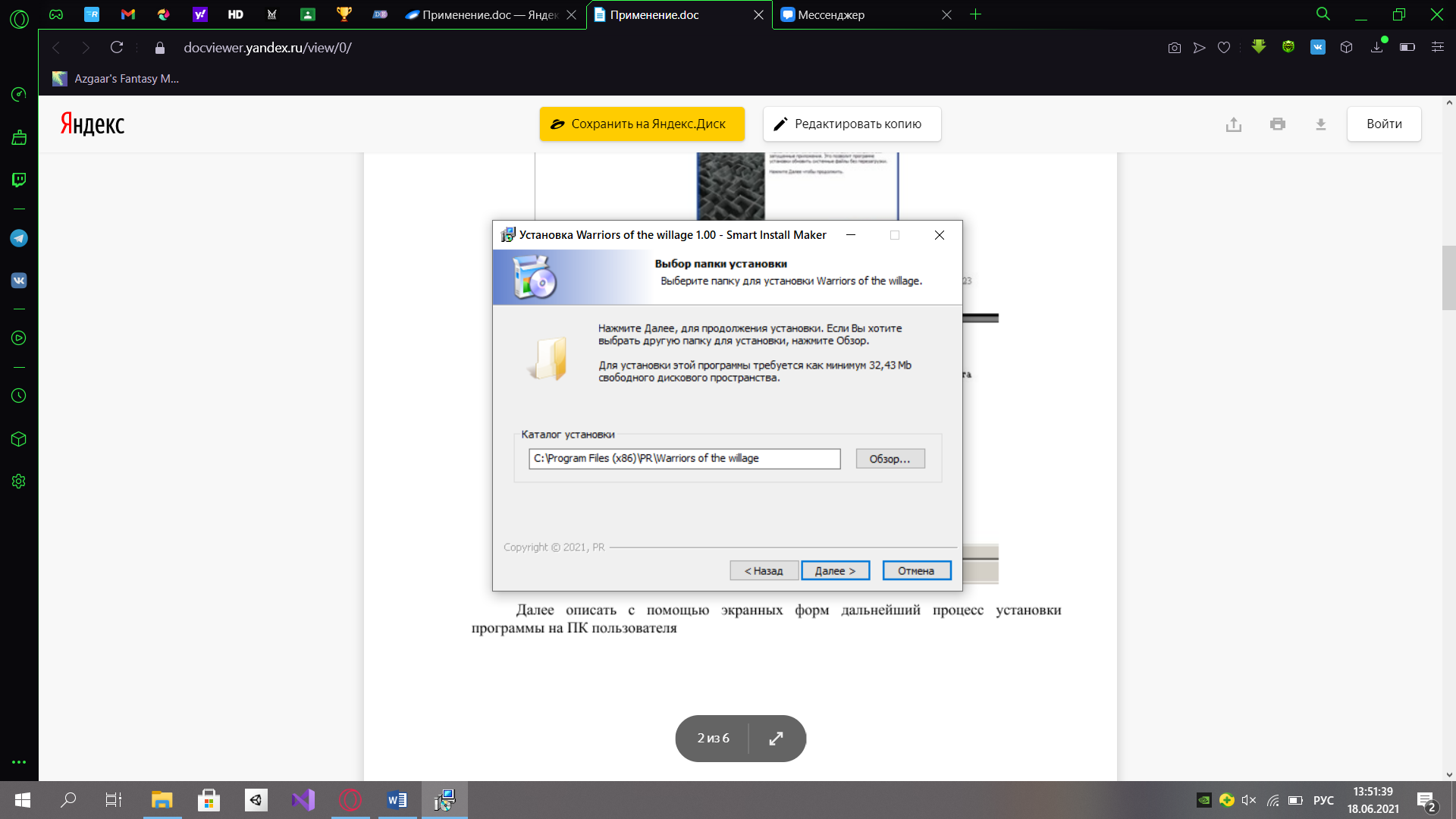


Рисунок 6 – Выбор пути установки

После этого необходимо подтвердить установку программы и дождаться завершения установки, после чего можно пользоваться игрой.

**5.3 Выполнения программы**

**5.3.1 Запуск программы**

Для запуска программы, необходимо перейти в папку с самой программой и запустить файл Demo.exe.

**5.3.2 Инструкция по работе с программой**

После запуска программы и проигрывания заставки, откроется Главное меню (рисунок 7).

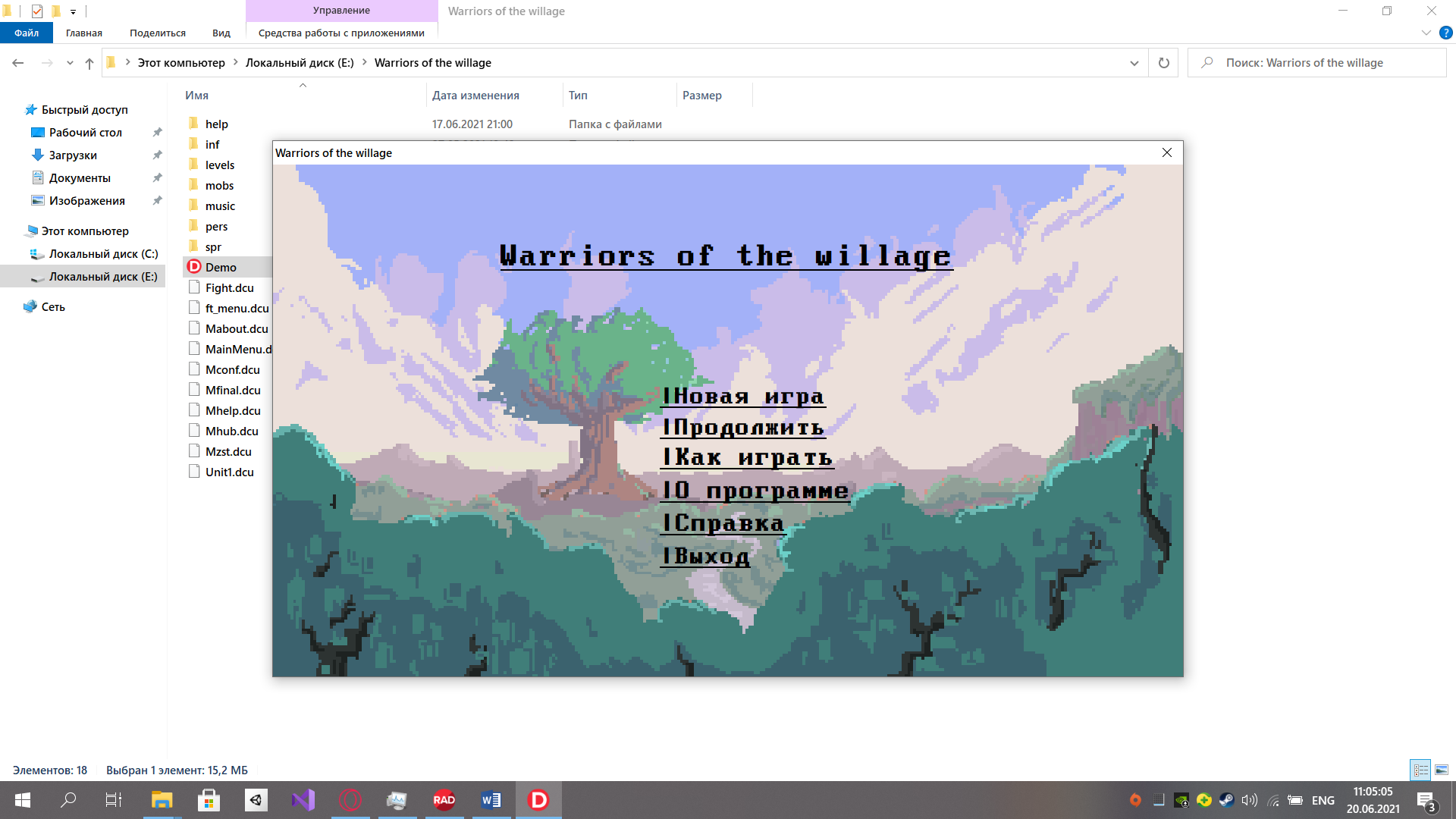


Рисунок 7 – Главное меню

Далее можно ознакомиться с правилами игры нажав на пункт Как играть. Откроется форма, на которой расписаны правила игры (рисунок 8). Нажимая на пункты внизу формы, вы ознакомитесь с основными формами игры и поймете, как в нее играть.

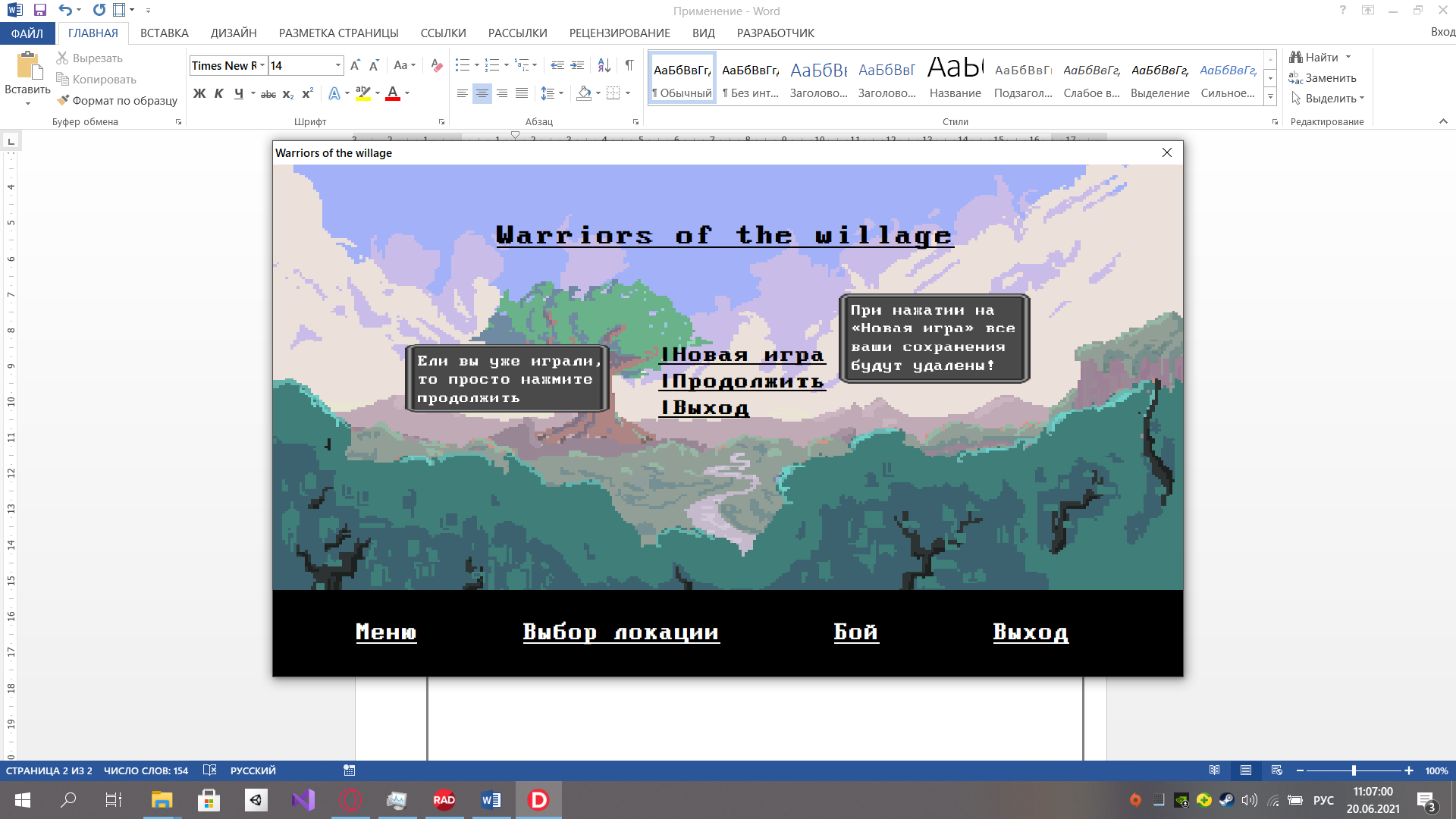


Рисунок 8 – Как играть

Чтобы начать играть, вам нужно нажать на пункт Новая игра или Продолжить в Главном меню. После этого вы перейдете на форму выбора локации (рисунок 9). Здесь вы можете выбрать локацию, на которой будете проходить уровень. Также здесь отображается количество пройденных уровней.

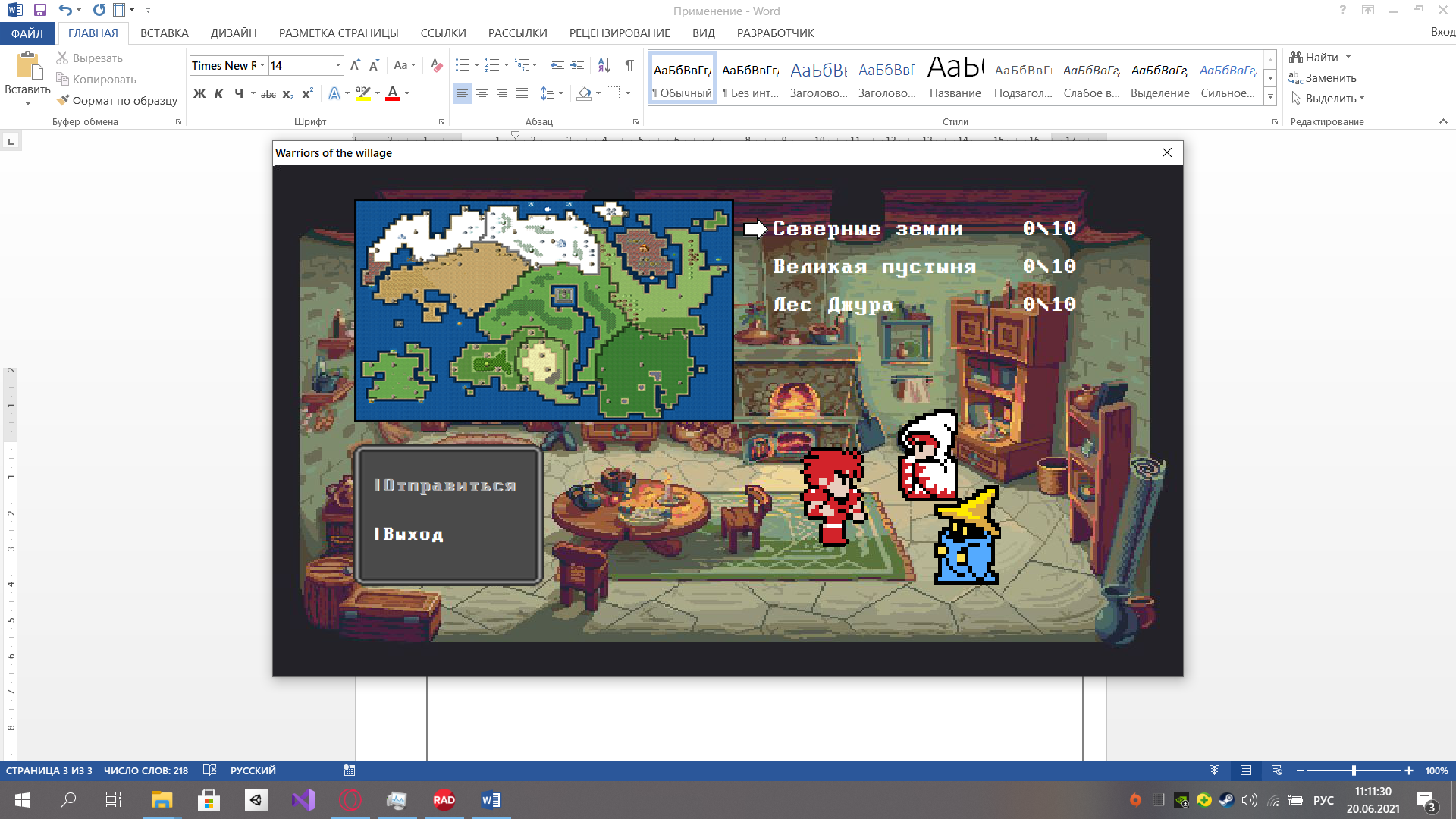


Рисунок 9 – Выбор локации

После выбора локации, нажмите на пункт Отправиться, вы перейдете в форму боя (рисунок 10). Здесь вам необходимо одолеть всех врагов для прохождения уровня.

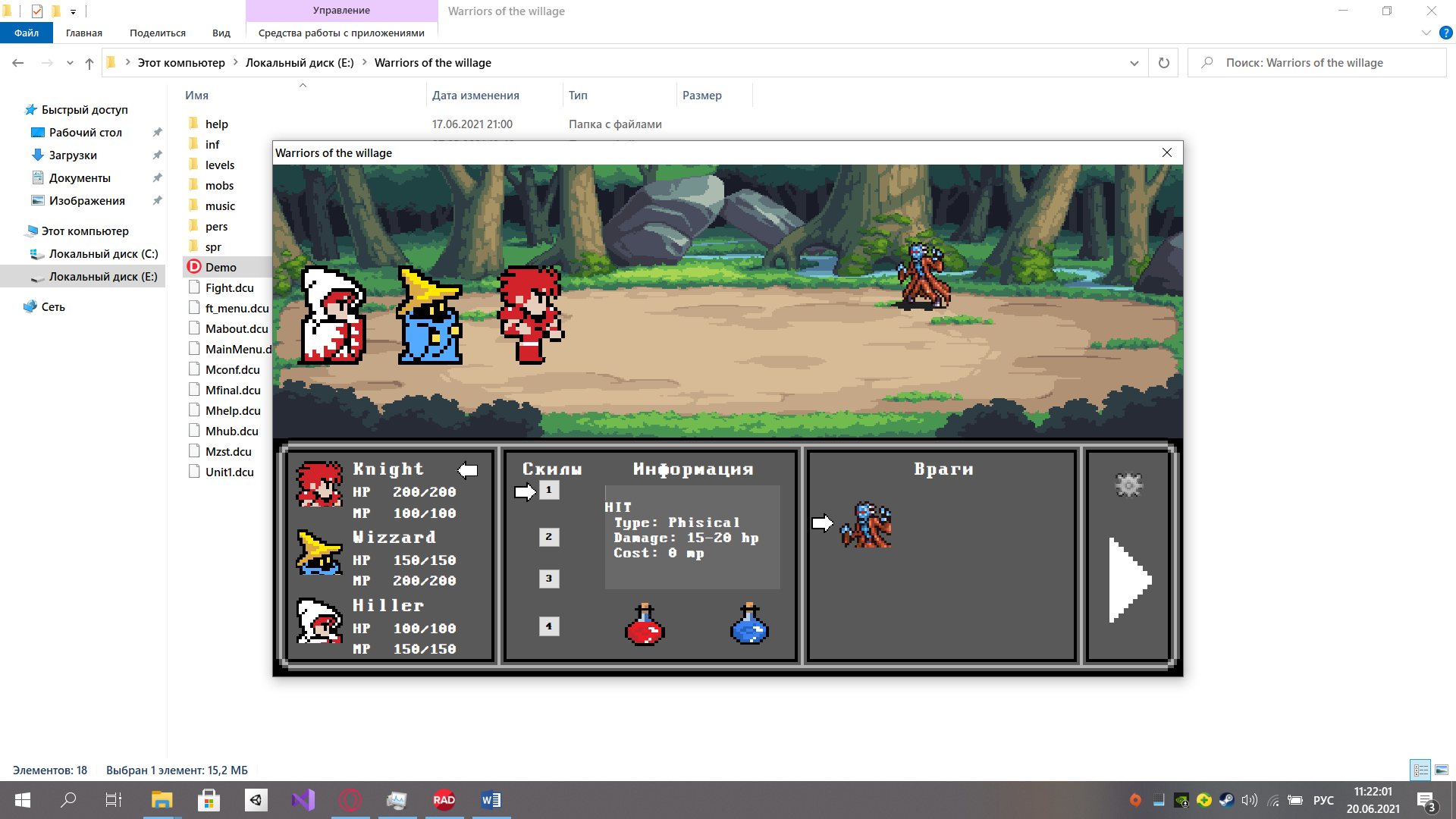


Рисунок 10 – Бой

Весь бой проходит пошагово. Для хода вам необходимо выбрать умение персонажа, при выборе умения в окошке Информация появиться описание этого умения. Далее вам необходимо выбрать врага, которого вы будете атаковать. Если умение не является атакующим вам все равно необходимо выбрать врага. После этого нужно нажать на треугольник в правой части экрана. Умения тратят MP, так что следите за их количеством.

После хода персонажа Hiller, идет очередь хода врагов, каждый из них наносит урон одному из персонажей и уменьшает количество HP. Если количество HP уменьшится до 0, то персонаж не сможет ходить до конца уровня. Бой будет выигран, если вы убьете всех врагов в трех раундах. Удачи!

Если вы пройдете по 10 уровней на каждой локации вы пройдете игру.

**5.3.3 Завершение работы с программой**

Для выхода из программы необходимо вернуться в Главное меню и нажать на пункт Выход или просто закрыть приложение в правом верхнем углу.

**5.4 Использование справочной системы**

Войти в справочную систему можно в Главном меню, нажав на пункт Справка. Откроется справочная система (Рисунок 11). Здесь вы можете выбрать нужный раздел и прочитать информацию, содержащуюся в нем.

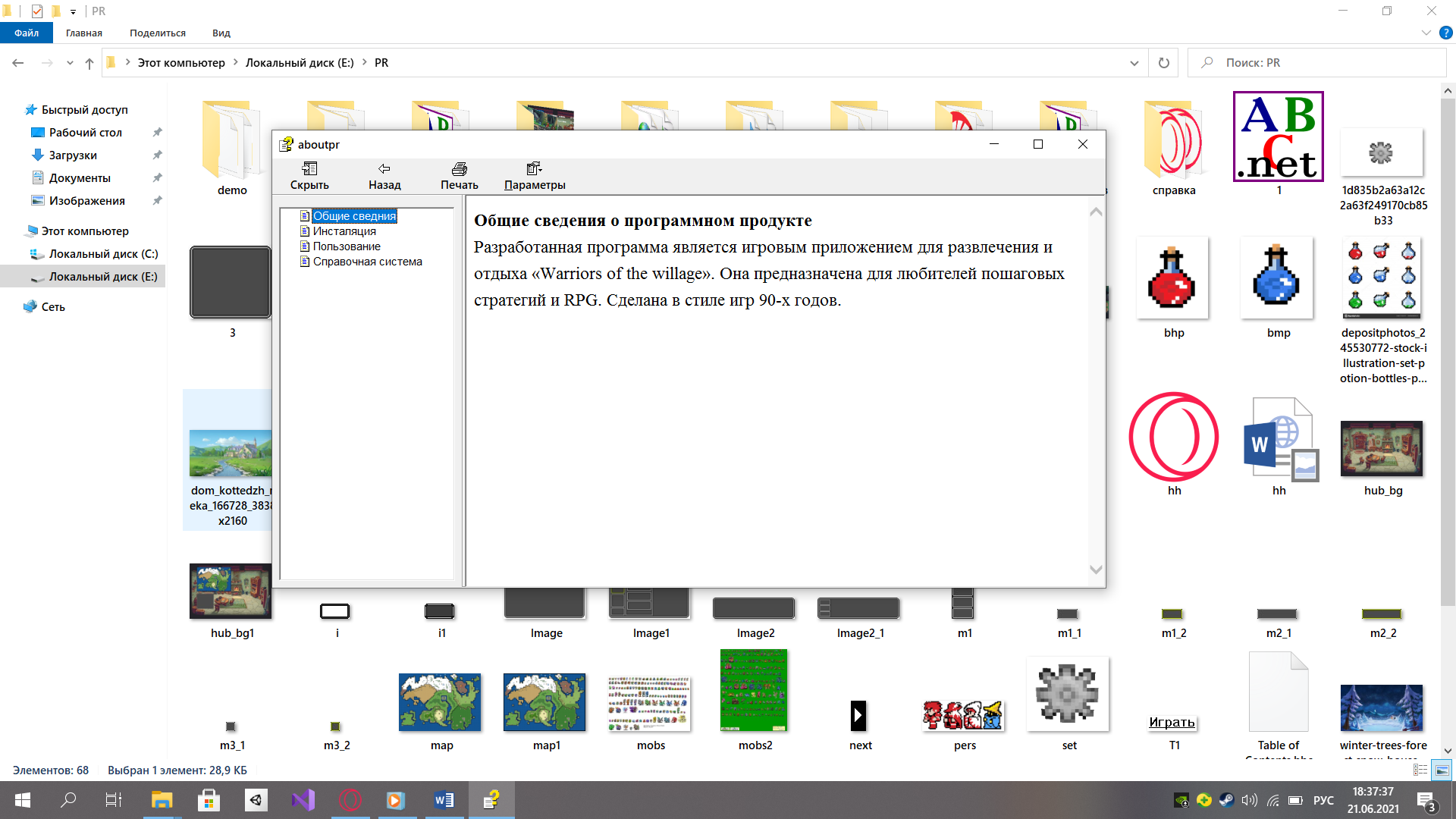


Рисунок 11 – Справка

**Заключение**

Целью данного курсового проекта было разработать игровое приложение в жанре RPG «Warriors of the willage».

В поставленной задаче был реализован пошаговый бой с выбором локации уровня.

В ходе тестирования все исключительные ситуации были обработаны. Проект работает без сбоев и производит вычисления во время боя без ошибок.

Основные механики были реализованы достаточно неплохо. Интерфейс довольно простой и понятный, но без прочтения правил игры пошаговый бой может вызвать небольшие трудности.

Исходя из этого, можно сделать вывод, что программа реализована успешно.

**Список использованных источников**

1. Киберфорум [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.cyberforum.ru>.
2. Джейсон Шрейер «Кровь, пот и пиксели. Обратная сторона индустрии видеоигр» - М.: Изд-во «Эскимо», 2018.

**Приложение А**

**Листинг программы**

**unit** Fight; //Форма боя

**interface**

**uses**

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Imaging.pngimage, Vcl.ExtCtrls,

Vcl.StdCtrls, Vcl.Buttons, Vcl.MPlayer;

**type**

Tft = **class**(TForm)

bg: TImage;

console: TImage;

ftm\_img: TImage;

Knight\_l: TLabel;

Wizzard\_l: TLabel;

Hiller\_l: TLabel;

hp\_l1: TLabel;

hp\_l2: TLabel;

hp\_l3: TLabel;

mp\_l1: TLabel;

mp\_l2: TLabel;

mp\_l3: TLabel;

k\_icon: TImage;

w\_icon: TImage;

h\_icon: TImage;

k\_hp: TLabel;

k\_mp: TLabel;

w\_hp: TLabel;

w\_mp: TLabel;

h\_hp: TLabel;

h\_mp: TLabel;

inf\_l: TLabel;

inf: TMemo;

skill1: TSpeedButton;

skill2: TSpeedButton;

skill3: TSpeedButton;

skill4: TSpeedButton;

next: TImage;

en\_l: TLabel;

en\_icon1: TImage;

en\_icon3: TImage;

en\_icon5: TImage;

en\_icon2: TImage;

en\_icon4: TImage;

en\_icon6: TImage;

h\_img: TImage;

w\_img: TImage;

k\_img: TImage;

en\_img1: TImage;

en\_img2: TImage;

en\_img3: TImage;

en\_img4: TImage;

en\_img6: TImage;

en\_img5: TImage;

cursor1: TImage;

cursor2: TImage;

cursor3: TImage;

roundl: TLabel;

Timer1: TTimer;

Timer2: TTimer;

skills\_l: TLabel;

bhp: TImage;

bmp: TImage;

music: TMediaPlayer;

Timer3: TTimer;

**procedure** FormClose(Sender: TObject; **var** Action: TCloseAction);

**procedure** ftm\_imgClick(Sender: TObject);

**procedure** FormShow(Sender: TObject);

**procedure** en\_icon1Click(Sender: TObject);

**procedure** en\_icon2Click(Sender: TObject);

**procedure** en\_icon3Click(Sender: TObject);

**procedure** en\_icon5Click(Sender: TObject);

**procedure** en\_icon4Click(Sender: TObject);

**procedure** en\_icon6Click(Sender: TObject);

**procedure** skill1Click(Sender: TObject);

**procedure** skill2Click(Sender: TObject);

**procedure** skill3Click(Sender: TObject);

**procedure** skill4Click(Sender: TObject);

**procedure** nextClick(Sender: TObject);

**procedure** Timer1Timer(Sender: TObject);

**procedure** Timer2Timer(Sender: TObject);

**procedure** bhpClick(Sender: TObject);

**procedure** bmpClick(Sender: TObject);

**procedure** FormHide(Sender: TObject);

**procedure** Timer3Timer(Sender: TObject);

**private**

{ Private declarations }

**public**

{ Public declarations }

**end**;

**type** pers = **record** //запись героев

code : byte;

hp : integer;

mp : integer;

**end**;

**var**

ft: Tft;

col : byte; //переменные мобов

enhp : **array** [0..5] **of** integer;

enind : **array** [0..5] **of** byte;

mp, hp, en :integer;

en1:string[2];

main\_spr1 : string[11];

spr1 : string[13];

motion, hero\_attack, target, round, res1, res2, lev : byte;

pers\_file : **file of** pers;

hero : pers;

hpp, mhp, mpp, mmp: **array** [0..2] **of** integer;

load : **file of** byte;

**procedure** heroes\_stats(index : integer);

**implementation**

{$R \*.dfm}

**uses** MainMenu, ft\_menu, Mhub;

**procedure** heroes\_stats(index : integer);

**var** k:byte;

**begin**

AssignFile(pers\_file,'pers\pers\_demo.dat');

k := 0;

reset(pers\_file);

**while** (**not** EOF(pers\_file)) **do**

**begin**

seek(pers\_file,k);

read(pers\_file,hero);

**if** index = hero.code **then**

**begin**

hp := hero.hp;

mp := hero.mp;

**break**;

**end**;

k := k+1;

**end**;

close(pers\_file);

**end**;

**procedure** Tft.bhpClick(Sender: TObject); //выбор хилки hp

**begin**

**if** res1=0 **then**

**begin**

cursor2.Top := bhp.Top+25;

cursor2.left := bhp.left-35;

hero\_attack := 4;

inf.Lines.LoadFromFile('inf\bhp.txt');

**end**;

**end**;

**procedure** Tft.bmpClick(Sender: TObject);

**begin**

**if** res2=0 **then**

**begin**

cursor2.Top := bmp.Top+25;

cursor2.left := bmp.Left-35;

hero\_attack := 5;

inf.Lines.LoadFromFile('inf\bmp.txt');

**end**;

**end**;

**procedure** Tft.en\_icon1Click(Sender: TObject); //выбор цели 1

**begin**

**if** enhp[0] >0 **then**

**begin**

cursor3.left := en\_icon1.left-30;

cursor3.top := en\_icon1.top+30;

target := 0;

next.Enabled := true;

**end**;

**end**;

**procedure** Tft.en\_icon2Click(Sender: TObject); //2

**begin**

**if** enhp[1] >0 **then**

**begin**

cursor3.left := en\_icon2.left-30;

cursor3.top := en\_icon2.top+30;

target := 1;

next.Enabled := true;

**end**;

**end**;

**procedure** Tft.en\_icon3Click(Sender: TObject); //3

**begin**

**if** enhp[2] >0 **then**

**begin**

cursor3.left := en\_icon3.left-30;

cursor3.top := en\_icon3.top+30;

target := 2;

next.Enabled := true;

**end**;

**end**;

**procedure** Tft.en\_icon4Click(Sender: TObject); //4

**begin**

**if** enhp[3] >0 **then**

**begin**

cursor3.left := en\_icon4.left-30;

cursor3.top := en\_icon4.top+30;

target := 3;

next.Enabled := true;

**end**;

**end**;

**procedure** Tft.en\_icon5Click(Sender: TObject); //5

**begin**

**if** enhp[4] >0 **then**

**begin**

cursor3.left := en\_icon5.left-30;

cursor3.top := en\_icon5.top+30;

target := 4;

next.Enabled := true;

**end**;

**end**;

**procedure** Tft.en\_icon6Click(Sender: TObject); //6

**begin**

**if** enhp[5] >0 **then**

**begin**

cursor3.left := en\_icon6.left-30;

cursor3.top := en\_icon6.top+30;

target := 5;

next.Enabled := true;

**end**;

**end**;

**procedure** Tft.FormClose(Sender: TObject; **var** Action: TCloseAction);

**begin**

hub.show;

ft.hide;

**end**;

**procedure** Tft.FormHide(Sender: TObject);

**begin**

music.Stop;

Timer3.Enabled := false;

**end**;

**procedure** Tft.FormShow(Sender: TObject);

**var** mobhp: textfile;

mob\_main\_spr : textfile;

k : integer;

**begin**

motion := 0;

round := 1;

hero\_attack := 0;

cursor1.Top := Knight\_l.Top;

cursor2.Top := skill1.Top;

cursor2.Left := skill1.left-40;

cursor3.left := en\_icon1.left-30;

cursor3.top := en\_icon1.top+30;

next.Enabled := false;

Timer2.Enabled := false;

Timer3.Enabled := true;

music.FileName := 'music\Battle Scene.mp3';

music.Open;

music.Play;

AssignFile(load,'levels/load.dat');

reset(load);

read(load,lev);

Closefile(load);

bg.Picture.LoadFromFile('spr\ft\_bg'+IntToStr(lev)+'.png');

roundl.Caption := 'Раунд 1';

Timer1.Enabled := true;

**for var** i := 0 **to** 2 **do** //занесение данных персонажей и их статов

**begin**

heroes\_stats(i);

hpp[i] := hp;

mpp[i] := mp;

mhp[i] := hp;

mmp[i] := mp;

**end**;

k\_hp.Caption := IntToStr(hpp[0])+'/'+ IntToStr(mhp[0]);

w\_hp.Caption := IntToStr(hpp[1])+'/'+ IntToStr(mhp[1]);

h\_hp.Caption := IntToStr(hpp[2])+'/'+ IntToStr(mhp[2]);

k\_mp.Caption := IntToStr(mpp[0])+'/'+ IntToStr(mmp[0]);

w\_mp.Caption := IntToStr(mpp[1])+'/'+ IntToStr(mmp[1]);

h\_mp.Caption := IntToStr(mpp[2])+'/'+ IntToStr(mmp[2]);

col := random(6); //рандом мобов

**for var** i := 0 **to** col **do**

**begin**

k := random(15);

enind[i] := k;

AssignFile(mobhp,'mobs\enemies\_hp.txt');

reset(mobhp);

**for var** o := 0 **to** 14 **do**

**begin**

**if** o = k **then**

**begin**

readln(mobhp,en1);

**break**;

**end**;

readln(mobhp,en1);

**end**;

enhp[i] := StrToInt(en1);

closefile(mobhp);

AssignFile(mob\_main\_spr,'mobs\enemies\_main\_spr.txt');

reset(mob\_main\_spr);

**for var** o := 0 **to** 14 **do**

**begin**

**if** o = k **then**

**begin**

readln(mob\_main\_spr,main\_spr1);

**break**;

**end**;

readln(mob\_main\_spr,main\_spr1);

**end**;

TImage(FindComponent('en\_img'+IntToStr(i+1))).Picture.LoadFromFile(main\_spr1);

closefile(mob\_main\_spr);

AssignFile(mob\_main\_spr,'mobs\enemies\_spr.txt');

reset(mob\_main\_spr);

**for var** o := 0 **to** 14 **do**

**begin**

**if** o = k **then**

**begin**

readln(mob\_main\_spr,spr1);

**break**;

**end**;

readln(mob\_main\_spr,spr1);

**end**;

TImage(FindComponent('en\_icon'+IntToStr(i+1))).Picture.LoadFromFile(spr1);

closefile(mob\_main\_spr);

**end**;

**end**;

**procedure** Tft.ftm\_imgClick(Sender: TObject);

**begin**

ftm.showmodal;

**end**;

**procedure** Tft.nextClick(Sender: TObject); //атака персонажей

**var** mobhp: textfile;

mob\_main\_spr : textfile;

k : integer;

**begin**

cursor2.Top := skill1.Top;

cursor2.Left := skill1.left-40;

cursor3.left := en\_icon1.left-30;

cursor3.top := en\_icon1.top+30;

**if** res1>0 **then** //откат хилки hp

**begin**

res1 := res1-1;

**end**;

**if** res2>0 **then** //откат хилки mp

**begin**

res2 := res2-1;

**end**;

**if** (motion = 0) **and** (hpp[0]>0) **then**

**begin**

**if** hero\_attack = 0 **then**

**begin**

enhp[target] := enhp[target]-(15+random(5));

**end**

**else if** hero\_attack = 1 **then**

**begin**

enhp[target] := enhp[target]-(25+random(10));

mpp[0] := mpp[0]-20;

**end**

**else if** hero\_attack = 2 **then**

**begin**

enhp[target] := enhp[target]-(20+random(5));

mpp[0] := mpp[0]-10;

**end**

**else if** hero\_attack = 3 **then**

**begin**

enhp[target] := enhp[target]-(35+random(10));

mpp[0] := mpp[0]-30;

**end**

**else if** hero\_attack = 4 **then**

**begin**

hpp[0] := hpp[0]+50;

**if** hpp[0]>mhp[0] **then** hpp[0] := mhp[0];

k\_hp.Caption := IntToStr(hpp[0])+'/'+ IntToStr(mhp[0]);

res1 := 8;

**end**

**else if** hero\_attack = 5 **then**

**begin**

mpp[0] := mpp[0]+50;

**if** mpp[0]>mmp[0] **then** mpp[0] := mmp[0];

res2 := 8;

**end**;

k\_mp.Caption := IntToStr(mpp[0])+'/'+ IntToStr(mmp[0]);

**end**

**else if** (motion = 1) **and** (hpp[1]>0) **then**

**begin**

**if** hero\_attack = 0 **then**

**begin**

enhp[target] := enhp[target]-(10+random(5));

**end**

**else if** hero\_attack = 1 **then**

**begin**

enhp[target] := enhp[target]-(25+random(30));

mpp[1] := mpp[1]-30;

**end**

**else if** hero\_attack = 2 **then**

**begin**

enhp[target] := enhp[target]-(35+random(5));

mpp[1] := mpp[1]-40;

**end**

**else if** hero\_attack = 3 **then**

**begin**

enhp[target] := enhp[target]-(40+random(10));

mpp[1] := mpp[1]-60;

**end**

**else if** hero\_attack = 4 **then**

**begin**

hpp[1] := hpp[1]+50;

**if** hpp[1]>mhp[1] **then** hpp[1] := mhp[1];

w\_hp.Caption := IntToStr(hpp[1])+'/'+ IntToStr(mhp[1]);

res1 := 8;

**end**

**else if** hero\_attack = 5 **then**

**begin**

mpp[1] := mpp[1]+50;

**if** mpp[1]>mmp[1] **then** mpp[1] := mmp[1];

res2 := 8;

**end**;

w\_mp.Caption := IntToStr(mpp[1])+'/'+ IntToStr(mmp[1]);

**end**

**else if** (motion = 2) **and** (hpp[2]>0) **then**

**begin**

**if** hero\_attack = 0 **then**

**begin**

enhp[target] := enhp[target]-(10+random(5));

**end**

**else if** hero\_attack = 1 **then**

**begin**

**for var** i := 0 **to** 2 **do**

**begin**

hpp[i] := hpp[i]+25;

**if** hpp[i]>mhp[i] **then** hpp[i] := mhp[i];

**end**;

mpp[2] := mpp[2]-30;

k\_hp.Caption := IntToStr(hpp[0])+'/'+ IntToStr(mhp[0]);

w\_hp.Caption := IntToStr(hpp[1])+'/'+ IntToStr(mhp[1]);

h\_hp.Caption := IntToStr(hpp[2])+'/'+ IntToStr(mhp[2]);

**end**

**else if** hero\_attack = 2 **then**

**begin**

**for var** i := 0 **to** 2 **do**

**begin**

mpp[i] := mpp[i]+25;

**if** mpp[i]>mmp[i] **then** mpp[i] := mmp[i];

**end**;

mpp[2] := mpp[2]-15;

k\_mp.Caption := IntToStr(mpp[0])+'/'+ IntToStr(mmp[0]);

w\_mp.Caption := IntToStr(mpp[1])+'/'+ IntToStr(mmp[1]);

**end**

**else if** hero\_attack = 3 **then**

**begin**

**for var** i := 0 **to** 2 **do**

**begin**

hpp[i] := hpp[i]+50;

**if** hpp[i]>mhp[i] **then** hpp[i] := mhp[i];

**end**;

mpp[2] := mpp[2]-45;

k\_hp.Caption := IntToStr(hpp[0])+'/'+ IntToStr(mhp[0]);

w\_hp.Caption := IntToStr(hpp[1])+'/'+ IntToStr(mhp[1]);

h\_hp.Caption := IntToStr(hpp[2])+'/'+ IntToStr(mhp[2]);

**end**

**else if** hero\_attack = 4 **then**

**begin**

hpp[2] := hpp[2]+50;

**if** hpp[2]>mhp[2] **then** hpp[2] := mhp[2];

h\_hp.Caption := IntToStr(hpp[2])+'/'+ IntToStr(mhp[2]);

res1 := 8;

**end**

**else if** hero\_attack = 5 **then**

**begin**

mpp[2] := mpp[2]+50;

**if** mpp[2]>mmp[2] **then** mpp[2] := mmp[2];

res2 := 8;

**end**;

h\_mp.Caption := IntToStr(mpp[2])+'/'+ IntToStr(mmp[2]);

**end**;

inf.Lines.Clear;

hero\_attack := 0;

**for var** i := 0 **to** 5 **do** //убийство мобов

**begin**

**if** enhp[i]<=0 **then**

**begin**

TImage(FindComponent('en\_icon'+IntToStr(i+1))).Picture := nil;

TImage(FindComponent('en\_img'+IntToStr(i+1))).Picture := nil;

**end**;

**end**;

**if** (enhp[0]<=0) **and** (enhp[1]<=0) **and** (enhp[2]<=0) **and** (enhp[3]<=0) **and** (enhp[4]<=0) **and** (enhp[5]<=0) **then**

**begin**

**if** round = 3 **then**

**begin**

AssignFile(load,'levels/load.dat'); // победа

rewrite(load);

lev := 4;

write(load,lev);

Closefile(load);

next.Enabled := false;

roundl.Caption := 'Ты выиграл!';

Timer2.Enabled := true;

**end**

**else** //смена раунда

**begin**

round := round+1;

col := random(6);

roundl.Caption := 'Раунд '+ IntToStr(round);

Timer1.Enabled := true;

//рандом мобов

**for var** i := 0 **to** col **do**

**begin**

k := random(15);

enind[i] := k;

AssignFile(mobhp,'mobs\enemies\_hp.txt');

reset(mobhp);

**for var** o := 0 **to** 14 **do**

**begin**

**if** o = k **then**

**begin**

readln(mobhp,en1);

**break**;

**end**;

readln(mobhp,en1);

**end**;

enhp[i] := StrToInt(en1);

closefile(mobhp);

AssignFile(mob\_main\_spr,'mobs\enemies\_main\_spr.txt');

reset(mob\_main\_spr);

**for var** o := 0 **to** 2 **do**

**begin**

**if** o = k **then**

**begin**

readln(mob\_main\_spr,main\_spr1);

**break**;

**end**;

readln(mob\_main\_spr,main\_spr1);

**end**;

TImage(FindComponent('en\_img'+IntToStr(i+1))).Picture.LoadFromFile(main\_spr1);

closefile(mob\_main\_spr);

AssignFile(mob\_main\_spr,'mobs\enemies\_spr.txt');

reset(mob\_main\_spr);

**for var** o := 0 **to** 2 **do**

**begin**

**if** o = k **then**

**begin**

readln(mob\_main\_spr,spr1);

**break**;

**end**;

readln(mob\_main\_spr,spr1);

**end**;

TImage(FindComponent('en\_icon'+IntToStr(i+1))).Picture.LoadFromFile(spr1);

closefile(mob\_main\_spr);

**end**;

**end**;

**end**;

motion := motion+1; //переход хода

**if** (motion = 0) **and** (hpp[motion] = 0) **then**

motion := motion+1;

**if** (motion = 1) **and** (hpp[motion] = 0) **then**

motion := motion+1;

**if** (motion = 2) **and** (hpp[motion] = 0) **then**

motion := motion+1;

**if** motion >= 3 **then**

**begin**

motion := 0 ;

**for var** i := 0 **to** 5 **do**

**begin**

**if** enhp[i]>0 **then**

**begin**

k := random(3);

hpp[k] := hpp[k]-(10+random(20));

**if** hpp[k]<0 **then** hpp[k] := 0

**end**;

**end**;

k\_hp.Caption := IntToStr(hpp[0])+'/'+ IntToStr(mhp[0]);

w\_hp.Caption := IntToStr(hpp[1])+'/'+ IntToStr(mhp[1]);

h\_hp.Caption := IntToStr(hpp[2])+'/'+ IntToStr(mhp[2]);

**if** hpp[0]=0 **then** k\_img.Picture := nil; //смерть персонажей

**if** hpp[1]=0 **then** w\_img.Picture := nil;

**if** hpp[2]=0 **then** h\_img.Picture := nil;

**if** (hpp[0]=0) **and** (hpp[1]=0) **and** (hpp[2]=0) **then**

**begin**

next.Enabled := false;

roundl.Caption := 'Ты проиграл!';

Timer2.Enabled := true;

**end**;

**end**;

**if** motion = 0 **then**

**begin**

cursor1.Top := Knight\_l.Top;

**end**

**else if** motion = 1 **then**

**begin**

cursor1.Top := Wizzard\_l.Top;

**end**

**else if** motion = 2 **then**

**begin**

cursor1.Top := Hiller\_l.Top;

**end**;

next.Enabled := false;

**end**;

**procedure** Tft.skill1Click(Sender: TObject); // выбор скила 1

**begin**

**if** motion = 0 **then**

**begin**

**if** mpp[0] >= 0 **then**

**begin**

cursor2.Top := skill1.Top;

cursor2.Left := skill1.left-40;

hero\_attack := 0;

inf.Lines.LoadFromFile('inf\k1.txt');

**end**;

**end**

**else if** motion = 1 **then**

**begin**

**if** mpp[1] >= 0 **then**

**begin**

cursor2.Top := skill1.Top;

cursor2.Left := skill1.left-40;

hero\_attack := 0;

inf.Lines.LoadFromFile('inf\w1.txt');

**end**;

**end**

**else if** motion = 2 **then**

**begin**

**if** mpp[2] >= 0 **then**

**begin**

cursor2.Top := skill1.Top;

cursor2.Left := skill1.left-40;

hero\_attack := 0;

inf.Lines.LoadFromFile('inf\h1.txt');

**end**;

**end**;

**end**;

**procedure** Tft.skill2Click(Sender: TObject); //2

**begin**

**if** motion = 0 **then**

**begin**

**if** mpp[0] >= 20 **then**

**begin**

cursor2.Top := skill2.Top;

cursor2.Left := skill2.left-40;

hero\_attack := 1;

inf.Lines.LoadFromFile('inf\k2.txt');

**end**;

**end**

**else if** motion = 1 **then**

**begin**

**if** mpp[1] >= 30 **then**

**begin**

cursor2.Top := skill2.Top;

cursor2.Left := skill2.left-40;

hero\_attack := 1;

inf.Lines.LoadFromFile('inf\w2.txt');

**end**;

**end**

**else if** motion = 2 **then**

**begin**

**if** mpp[2] >= 30 **then**

**begin**

cursor2.Top := skill2.Top;

cursor2.Left := skill2.left-40;

hero\_attack := 1;

inf.Lines.LoadFromFile('inf\h2.txt');

**end**;

**end**;

**end**;

**procedure** Tft.skill3Click(Sender: TObject); //3

**begin**

**if** motion = 0 **then**

**begin**

**if** mpp[0] >= 10 **then**

**begin**

cursor2.Top := skill3.Top;

cursor2.Left := skill3.left-40;

hero\_attack := 2;

inf.Lines.LoadFromFile('inf\k3.txt');

**end**;

**end**

**else if** motion = 1 **then**

**begin**

**if** mpp[1] >= 40 **then**

**begin**

cursor2.Top := skill3.Top;

cursor2.Left := skill3.left-40;

hero\_attack := 2;

inf.Lines.LoadFromFile('inf\w3.txt');

**end**;

**end**

**else if** motion = 2 **then**

**begin**

**if** mpp[2] >= 15 **then**

**begin**

cursor2.Top := skill3.Top;

cursor2.Left := skill3.left-40;

hero\_attack := 2;

inf.Lines.LoadFromFile('inf\h3.txt');

**end**;

**end**;

**end**;

**procedure** Tft.skill4Click(Sender: TObject); //4

**begin**

**if** motion = 0 **then**

**begin**

**if** mpp[0] >= 30 **then**

**begin**

cursor2.Top := skill4.Top;

cursor2.Left := skill4.left-40;

hero\_attack := 3;

inf.Lines.LoadFromFile('inf\k4.txt');

**end**;

**end**

**else if** motion = 1 **then**

**begin**

**if** mpp[1] >= 70 **then**

**begin**

cursor2.Top := skill4.Top;

cursor2.Left := skill4.left-40;

hero\_attack := 3;

inf.Lines.LoadFromFile('inf\w4.txt');

**end**;

**end**

**else if** motion = 2 **then**

**begin**

**if** mpp[2] >= 45 **then**

**begin**

cursor2.Top := skill4.Top;

cursor2.Left := skill4.left-40;

hero\_attack := 3;

inf.Lines.LoadFromFile('inf\h4.txt');

**end**;

**end**;

**end**;

**procedure** Tft.Timer1Timer(Sender: TObject);

**begin**

roundl.Caption := '';

**end**;

**procedure** Tft.Timer2Timer(Sender: TObject);

**begin**

ft.Close;

Timer2.Enabled := false;

**end**;

**procedure** Tft.Timer3Timer(Sender: TObject);

**begin**

music.Play;

**end**;

**end**.

**unit** ft\_menu; //менню боя

**interface**

**uses**

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ExtCtrls,

Vcl.Imaging.pngimage;

**type**

Tftm = **class**(TForm)

Image1: TImage;

cont: TLabel;

**exit**: TLabel;

**procedure** contClick(Sender: TObject);

**procedure** exitClick(Sender: TObject);

**private**

{ Private declarations }

**public**

{ Public declarations }

**end**;

**var**

ftm: Tftm;

**implementation**

{$R \*.dfm}

**uses** Fight;

**procedure** Tftm.contClick(Sender: TObject); //продолжить игру

**begin**

ftm.close;

**end**;

**procedure** Tftm.exitClick(Sender: TObject); //выйти с уровня

**begin**

ftm.close;

ft.close;

**end**;

**end**.

**unit** Mabout; //О программе

**interface**

**uses**

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls;

**type**

Tabout = **class**(TForm)

Memo1: TMemo;

**private**

{ Private declarations }

**public**

{ Public declarations }

**end**;

**var**

about: Tabout;

**implementation**

{$R \*.dfm}

**end**.

**unit** MainMenu; //Главное меню

**interface**

**uses**

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ExtCtrls, Vcl.StdCtrls,

Vcl.Imaging.pngimage, Vcl.MPlayer, ShellAPI;

**type**

TMM = **class**(TForm)

bg: TImage;

name: TLabel;

play: TLabel;

**exit**: TLabel;

ngame: TLabel;

music: TMediaPlayer;

lhelp: TLabel;

Timer: TTimer;

Label1: TLabel;

Label2: TLabel;

**procedure** exitClick(Sender: TObject);

**procedure** playClick(Sender: TObject);

**procedure** ngameClick(Sender: TObject);

**procedure** lhelpClick(Sender: TObject);

**procedure** FormShow(Sender: TObject);

**procedure** FormHide(Sender: TObject);

**procedure** TimerTimer(Sender: TObject);

**procedure** Label1Click(Sender: TObject);

**procedure** Label2Click(Sender: TObject);

**private**

{ Private declarations }

**public**

{ Public declarations }

**end**;

**var**

MM: TMM;

**implementation**

{$R \*.dfm}

**uses** Fight, Mhub, Mconf, Mhelp, Mabout;

**procedure** TMM.FormHide(Sender: TObject);//остановка музыки

**begin**

music.Stop;

Timer.Enabled := false;

**end**;

**procedure** TMM.FormShow(Sender: TObject); //включение музыки

**begin**

Timer.Enabled := true;

music.FileName := 'music\Opening.mp3';

music.Open;

music.Play;

**end**;

**procedure** TMM.Label1Click(Sender: TObject); //справка

**begin**

ShellExecute(0,PChar('Open'),PChar('help\about.chm'),nil,nil,SW\_SHOW);

**end**;

**procedure** TMM.Label2Click(Sender: TObject); //О программе

**begin**

about.showmodal;

**end**;

**procedure** TMM.exitClick(Sender: TObject); //выход

**begin**

MM.close;

**end**;

**procedure** TMM.ngameClick(Sender: TObject); //новая игра

**begin**

conf.show;

**end**;

**procedure** TMM.playClick(Sender: TObject); //продолжить

**begin**

hub.show;

MM.hide;

**end**;

**procedure** TMM.TimerTimer(Sender: TObject);

**begin**

music.Play;

**end**;

**procedure** TMM.lhelpClick(Sender: TObject); //как играть

**begin**

help.showmodal;

**end**;

**end**.

**unit** Mconf; //подтверждение новой игры

**interface**

**uses**

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Imaging.pngimage,

Vcl.ExtCtrls;

**type**

Tconf = **class**(TForm)

Image1: TImage;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

**procedure** Label2Click(Sender: TObject);

**procedure** Label1Click(Sender: TObject);

**private**

{ Private declarations }

**public**

{ Public declarations }

**end**;

**var**

conf: Tconf;

**implementation**

{$R \*.dfm}

**uses** Mhub, MainMenu;

**procedure** Tconf.Label1Click(Sender: TObject); //создание нового файла сохранения

**var** save : **file of** byte;

buf : byte;

**begin**

buf := 0;

AssignFile(save,'levels/save.dat');

rewrite(save);

**for var** i:=0 **to** 2 **do**

write(save,buf);

Closefile(save);

hub.show;

MM.hide;

conf.close;

**end**;

**procedure** Tconf.Label2Click(Sender: TObject);

**begin**

conf.close;

**end**;

**end**.

**unit** Mfinal; //форма финала игры

**interface**

**uses**

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Imaging.pngimage,

Vcl.ExtCtrls;

**type**

Tfinalf = **class**(TForm)

Image1: TImage;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

**procedure** Label4Click(Sender: TObject);

**procedure** FormClose(Sender: TObject; **var** Action: TCloseAction);

**private**

{ Private declarations }

**public**

{ Public declarations }

**end**;

**var**

finalf: Tfinalf;

**implementation**

{$R \*.dfm}

**uses** MainMenu, Mhub;

**procedure** Tfinalf.FormClose(Sender: TObject; **var** Action: TCloseAction);

**begin**

hub.close;

MM.show;

**end**;

**procedure** Tfinalf.Label4Click(Sender: TObject);

**begin**

finalf.close;

**end**;

**end**.

**unit** Mhelp; //как играть

**interface**

**uses**

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Imaging.pngimage,

Vcl.ExtCtrls;

**type**

Thelp = **class**(TForm)

Image1: TImage;

menu: TLabel;

Hub: TLabel;

ft: TLabel;

Label1: TLabel;

**procedure** menuClick(Sender: TObject);

**procedure** HubClick(Sender: TObject);

**procedure** ftClick(Sender: TObject);

**procedure** Label1Click(Sender: TObject);

**private**

{ Private declarations }

**public**

{ Public declarations }

**end**;

**var**

help: Thelp;

**implementation**

{$R \*.dfm}

**procedure** Thelp.ftClick(Sender: TObject);

**begin**

image1.Picture.LoadFromFile('help\HFT.png');

**end**;

**procedure** Thelp.HubClick(Sender: TObject);

**begin**

image1.Picture.LoadFromFile('help\HHub.png');

**end**;

**procedure** Thelp.Label1Click(Sender: TObject);

**begin**

help.close;

**end**;

**procedure** Thelp.menuClick(Sender: TObject);

**begin**

image1.Picture.LoadFromFile('help\HMM.png');

**end**;

**end**.

**unit** Mhub; //форма выбора локации

**interface**

**uses**

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.Imaging.jpeg, Vcl.ExtCtrls,

Vcl.Imaging.pngimage, Vcl.StdCtrls, Vcl.MPlayer;

**type**

Thub = **class**(TForm)

hub\_bg: TImage;

k\_img: TImage;

h\_img: TImage;

w\_img: TImage;

**exit**: TLabel;

play: TLabel;

lmap1: TLabel;

lmap2: TLabel;

lmap3: TLabel;

cursor: TImage;

levmap3: TLabel;

levmap2: TLabel;

levmap1: TLabel;

music: TMediaPlayer;

Timer: TTimer;

**procedure** FormClose(Sender: TObject; **var** Action: TCloseAction);

**procedure** exitClick(Sender: TObject);

**procedure** lmap3Click(Sender: TObject);

**procedure** lmap2Click(Sender: TObject);

**procedure** lmap1Click(Sender: TObject);

**procedure** FormShow(Sender: TObject);

**procedure** playClick(Sender: TObject);

**procedure** FormHide(Sender: TObject);

**procedure** TimerTimer(Sender: TObject);

**private**

{ Private declarations }

**public**

{ Public declarations }

**end**;

**var**

hub: Thub;

load : **file of** byte;

lev, did: byte;

maps : **array** [0..2] **of** byte;

**implementation**

{$R \*.dfm}

**uses** MainMenu, Fight, Mfinal;

**procedure** Thub.exitClick(Sender: TObject);

**begin**

MM.show;

hub.hide;

**end**;

**procedure** Thub.FormClose(Sender: TObject; **var** Action: TCloseAction);

**begin**

MM.close;

ft.close;

**end**;

**procedure** Thub.FormHide(Sender: TObject); //остановка музыки

**begin**

music.Stop;

Timer.Enabled := false

**end**;

**procedure** Thub.FormShow(Sender: TObject);

**var** save : **file of** byte;

k,buf : byte;

**begin**

play.Enabled := false;

Timer.Enabled := true; //включение музыки

music.FileName := 'music\OST.mp3';

music.Open;

music.Play;

AssignFile(load,'levels/load.dat'); //подтверждение пройденого уровня

reset(load);

read(load,did);

Closefile(load);

**if** did = 4 **then**

**begin**

maps[lev-1] := maps[lev-1]+1;

AssignFile(save,'levels/save.dat');

rewrite(save);

**for var** i:=0 **to** 2 **do**

write(save,maps[i]);

Closefile(save);

**end**;

AssignFile(load,'levels/load.dat');

rewrite(load);

lev := 0;

write(load,lev);

Closefile(load);

AssignFile(save,'levels\save.dat'); //отображение пройденых уровне

k :=0;

reset(save);

**while** (**not** EOF(save)) **do**

**begin**

seek(save,k);

read(save,buf);

**if** buf>10 **then** buf := 10;

maps[k] := buf;

TLabel(FindComponent('levmap'+IntToStr(k+1))).caption := IntToStr(maps[k])+'\10';

k := k+1;

**end**;

closefile(save);

**if** (maps[0] = 10) **and** (maps[1] = 10) **and** (maps[2] = 10) **then** //финал игры

**begin**

AssignFile(save,'levels/save.dat');

rewrite(save);

**for var** i:=0 **to** 2 **do**

write(save,buf);

Closefile(save);

finalf.showmodal;

**end**;

**end**;

**procedure** Thub.lmap1Click(Sender: TObject); //выбор локации 1

**begin**

cursor.top := lmap1.top;

play.Enabled := true;

lev := 1;

AssignFile(load,'levels/load.dat');

rewrite(load);

write(load,lev);

Closefile(load);

**end**;

**procedure** Thub.lmap2Click(Sender: TObject); //2

**begin**

cursor.top := lmap2.top;

play.Enabled := true;

lev := 2;

AssignFile(load,'levels/load.dat');

rewrite(load);

write(load,lev);

Closefile(load);

**end**;

**procedure** Thub.lmap3Click(Sender: TObject); //3

**begin**

cursor.top := lmap3.top;

play.Enabled := true;

lev := 3;

AssignFile(load,'levels/load.dat');

rewrite(load);

write(load,lev);

Closefile(load);

**end**;

**procedure** Thub.playClick(Sender: TObject); //запуск боя

**begin**

ft.Show;

hub.hide;

**end**;

**procedure** Thub.TimerTimer(Sender: TObject); // обновление музыки

**begin**

music.Play;

**end**;

**end**.

**unit** Mzst; //заставка

**interface**

**uses**

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Imaging.jpeg,

Vcl.ExtCtrls;

**type**

Tzst = **class**(TForm)

Image1: TImage;

Label1: TLabel;

Timer1: TTimer;

**procedure** Timer1Timer(Sender: TObject);

**private**

{ Private declarations }

**public**

{ Public declarations }

**end**;

**var**

zst: Tzst;

**implementation**

{$R \*.dfm}

**procedure** Tzst.Timer1Timer(Sender: TObject);

**begin**

Timer1.Enabled := false;

**end**;

**end**.