Оглавление

[Введение 2](#_Toc52523575)

[Базы данных 2](#_Toc52523576)

[Окружение 2](#_Toc52523577)

[Предметы 3](#_Toc52523578)

[Персонажи 4](#_Toc52523579)

[Заклинания 5](#_Toc52523580)

[Актёры 6](#_Toc52523581)

[Объекты окружения 6](#_Toc52523582)

[Активаторы 6](#_Toc52523583)

[Анимированные тайлы 7](#_Toc52523584)

[Источники звука окружения 7](#_Toc52523585)

[Контейнеры 8](#_Toc52523586)

[Двери 8](#_Toc52523587)

[Структуры 9](#_Toc52523588)

[Предметы 9](#_Toc52523589)

[Игровые объекты 10](#_Toc52523590)

[Управляющий игрой 10](#_Toc52523591)

[Тайлы 12](#_Toc52523592)

[Снаряды 13](#_Toc52523593)

[Карта 13](#_Toc52523594)

[Магическая система 14](#_Toc52523595)

[Школы магии 14](#_Toc52523596)

[Базовые понятия магической системы 14](#_Toc52523597)

[Магические эффекты 15](#_Toc52523598)

[Магические снаряды 15](#_Toc52523599)

[Заклинания 15](#_Toc52523600)

[Персонаж 16](#_Toc52523601)

[Ролевая система 16](#_Toc52523602)

[Автоматический контроль персонажей игрока 17](#_Toc52523603)

[Неигровые персонажи 17](#_Toc52523604)

[Перемещение 18](#_Toc52523605)

[Поиск маршрута 18](#_Toc52523606)

[Боевая система 22](#_Toc52523607)

[Колдовство 22](#_Toc52523608)

[Рисование персонажей 23](#_Toc52523609)

[Игровые окна 23](#_Toc52523610)

[Инвентарь 23](#_Toc52523611)

[Окно развития персонажа 25](#_Toc52523612)

[Окно книги заклинаний 26](#_Toc52523613)

[Игровое окно 27](#_Toc52523614)

# Введение

В данном документе описан алгоритм работы моей игры, которую я случайно нашёл на закоулках своего гугл диска. Мною был прочитан код, понят принцип его работы и передан в простой (насколько возможно) язык. Текст разбит на главы и предоставлен в том же порядке, в котором написан исходный код игры.

# Базы данных

В базах данных сохраняется информация и параметры всех добавленных в игру объектов. Все базы данных находятся в папке res/dat. Каждый объект занимает одну строку в файле, и в каждой строке через разделитель описаны параметры объекта. При запуске игры все объекты записываются в оперативную память: при загрузке объектов программа проходит по всем строкам в базе и записывает параметры объекта в зависимости от типа параметра (переводит строки в числовой формат, если это требуется). Следующая операция при работе с базой – сохранение объекта, в качестве параметра операция принимает требуемые для сохранения переменные объекта. После чего записывает в конец базы данных новую строку для сохраняемого объекта. Отдельная операция – удаление объекта, при которой удаляется строка, в которой описан объект. Главная операция – чтение данных объекта – получает на вход ID требуемого объекта и возвращает требуемые параметры для создания копии объекта.

## Окружение

Информация об объектах окружения хранится в файле res/dat/Env.res. Объекты окружения используются для наполнения игрового мира. В игре реализованы следующие типы объектов окружения: анимированные тайлы, активаторы, двери, контейнеры, источники звука и структуры. Ниже в таблице приведены параметры, хранящиеся в базе данных, для каждого типа объекта окружения.

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Пояснение |
| Общие параметры | | |
| ID | String | Имя объекта, по которому к нему обращаются в базе данных |
| texture | String | Имя файла с текстурой объекта |
| typeName | String | Тип объекта |
| Анимированные тайлы | | |
| frames | Integer | Количество кадров анимации тайла |
| animSpeed | Double | Скорость анимации |
| Активаторы | | |
| partX, partY | Integer | Часть текстуры, которую занимает активатор |
| stateAnim | Double | Показывает, был ли использован активатор или нет (1 – готов к использованию, 0 – был использован) |
| lockLevel | Integer | Определяет требуемый уровень навыка «Ловкость рук» для использования данного активатора |
| trap | Integer | Определяет вид активатора, если значение переменной равно 0, тогда с активатором нужно взаимодействовать руками, 1 – на активатор достаточно наступить |
| Двери | | |
| partX, partY | Integer | Часть текстуры, которую занимает дверь |
| stateAnim | Double | Показывает, открыта дверь или нет |
| lockLevel | Integer | Определяет требуемый уровень навыка «Ловкость рук» для открытия данной двери |
| lockOffX, lockOffY | Integer | Если дверь закрыта, эти параметры определяют расположение замка на двери |
| Контейнеры | | |
| partX, partY | Integer | Часть текстуры, которую занимает дверь |
| stateAnim | Double | Показывает, открыта дверь или нет |
| lockLevel | Integer | Определяет требуемый уровень навыка «Ловкость рук» для открытия данной двери |
| lockOffX, lockOffY | Integer | Если контейнер закрыт, эти параметры определяют расположение замка на контейнере |
| Источники звука | | |
| source | String | Наименование файла, из которого проигрывается звук |
| rast | Double | Расстояние, на котором слышен данный источник звука |
|  |  |  |
| Структуры | | |
| sizeX, sizeY | Integer | Размер структуры в тайлах |
| sizeYSolid | Integer | Размер «твёрдой» части структуры |
| partLeftSolid | Double | Обрезка текстуры структуры с левой стороны |
| partRightSolid | Double | Обрезка текстуры структуры с правой стороны |
| partTopSolid | Double | Обрезка текстуры структуры с верхней стороны |
| partBotSolid | Double | Обрезка текстуры структуры с нижней стороны – граница «твёрдой» части |
| toptransperent | Double | Обрезка текстуры структуры с верхней стороны – граница «прозрачной» части |
| bottransperent | Double | Обрезка текстуры структуры с нижней стороны |

## Предметы

Информация о предметах хранится в файле res/dat/Items.res. Предметы могут лежать как на пространстве игрового мира, так и находится в обозе игрока, откуда предмет можно надеть на персонажей, подконтрольных игроку. На данный момент в игру введены следующие типы предметов: оружие ближнего боя, оружие дальнего боя, броня, щиты, расходники.

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Пояснение |
| Общие параметры | | |
| ID | String | Имя предмета, по которому к нему обращаются в базе данных |
| texture | String | Имя файла с текстурой предмета |
| typeName | String | Тип объекта в текстовом формате |
| type | Integer | Тип объекта в числовом формате |
| filter | Integer | Код, по которому в окне инвентаря фильтруются предметы |
| partX, partY | Integer | Координаты ячейки, в которой расположена картинка предмета на текстуре (на одной текстуре по столбцам и строкам находится по восемь предметов, а всего их - 64) |
| name | String | Внутриигровое название предмета |
| weight | Double | Вес предмета |
| cost | Integer | Цена предмета |
| Оружие | | |
| weaponClass | Integer | Цифровое обозначение типа оружия |
| weaponCLassName | String | Строковое обозначение типа оружия (кинжал, меч, топор, дробящее, копьё, лук, арбалет ) |
| damage | Integer | Базовый урон, наносимый оружием за удар |
| damageType | Integer | Цифровое обозначение типа урона оружия |
| damageTypeName | String | Строковое обозначение типа урона оружия (колющий, рубящий, дробящий) |
| distant | Double | Расстояние, с которого владелец оружия может вести атаку |
| attackSpeed | Double | Скорость атак с применением данного оружия |
| projectileID | Integer | Номер снаряда, который использует данное оружие (дополнительно создана база данных снарядов, они хранят: текстуру применения оружия, размеры текстуры, количество кадров в анимации атаки и активный кадр, в котором рассчитывается урон) |
| fires | Integer | Боеприпас оружия (применяется для оружия дальнего боя) |
| Щиты | | |
| armorClass | Integer | Цифровое обозначение типа щита |
| armorClassName | String | Строковое описание типа щита (Лёгкий или Тяжёлый) |
| armorPierce | Integer | Защита щита от колющего урона |
| armorSlash | Integer | Защита щита от рубящего урона |
| armorStrike | Integer | Защита щита от дробящего урона |
| stability | Double | Блокирование атак щитом тратит запас сил персонажа, а не здоровье. Показатель стабильности показывает какой процент урона переходит с здоровье на запас сил. |
| Броня | | |
| armorClass | Integer | Цифровое обозначение типа брони |
| armorClassName | String | Строковое описание типа брони (Голова, тело, руки, ноги) |
| armorPierce | Integer | Защита брони от колющего урона |
| armorSlash | Integer | Защита брони от рубящего урона |
| armorStrike | Integer | Защита брони от дробящего урона |
| armorFire | Integer | Защита брони от огненного урона |
| armorFreeze | Integer | Защита брони от морозного урона |
| armorElectricity | Integer | Защита брони от электрического урона |
| armorAcid | Integer | Защита брони от кислотного урона |
| armorMental | Integer | Защита брони от ментального урона |
| armorWizardry | Integer | Защита брони от волшебного урона |
| armorSacred | Integer | Защита брони от сакрального урона |
| Расходники | | |
| usesClassName | String | Строковое обозначение типа расходника (зелье, свиток, бомба, порошок) |
| spellIn | String | ID номер заклинания, которое заложено внутрь расходника |

## Персонажи

Информация о персонажах хранится в файле res/dat/NPC.res. Персонажи перемещаются по игровому миру, взаимодействуют с ним и друг с другом.

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Пояснение |
| ID | String | Имя персонажа, по которому к нему обращаются в базе данных |
| maxHeath | Integer | Максимальное здоровье персонажа |
| health | Integer | Текущее здоровье персонажа |
| maxStamina | Integer | Максимальный запас сил персонажа |
| stamina | Integer | Текущий запас сил персонажа |
| skillRestoration | Integer | Скорость восстановления запаса сил персонажа |
| speedWalk | Double | Скорость персонажа при ходьбе |
| speedRun | Double | Скорость персонажа при беге |
| name | String | Внутриигровое имя персонажа |
| ingameTexture | String | Внутриигровая текстура персонажа |
| portret | String | Текстура портрета персонажа |
| portretX, portretY | Integer | Координаты портрета на текстуре |
| stepsTexture | String | Текстура для обозначения шагов персонажа |
| bloodTexture | String | Текстура для обозначения крови персонажа |
| vision | Double | Расстояние, на котором персонаж замечает других персонажей |
| visionType | Integer | Тип зрения персонажа (как у человека: конусное и не видит сквозь препятствия; персонаж игрока: не видит сквозь препятствия, но смотрит на 360 градусов; призрак: видит на 360 градусов и игнорирует препятствия) |
| distant | Integer | Расстояние, на котором персонаж может атаковать |
| attackSpeed | Double | Скорость атак персонажа |
| damage | Integer | Урон, наносимый персонажем |
| damageType | Integer | Тип наносимого урона |
| armor | Integer | Показатели защиты персонажа от различного типа урона |
| loot | String | Названия предметов, которые падают, после смерти персонажа |
| lootGoldMin, lootGoldMax | Integer | Границы количества золота, которое падает с персонажа, после его смерти |
| exp | Integer | Количество опыта, которым обладает персонаж |

## Заклинания

Информация о заклинаниях хранится в файле res/dat/Spells.res. Заклинания накладывают особые эффекты на персонажей, предметы и объекты окружения. Их можно использовать для создания различных игровых ситуаций. Также сами персонажи могут использовать заклинания на других персонажей и окружение.

|  |  |  |
| --- | --- | --- |
| Параметр | Тип | Пояснение |
| ID | String | Имя заклинания, по которому к нему обращаются в базе данных |
| texture | String | Имя файла с текстурой заклинания |
| partX, partY | Integer | Координаты ячейки, в которой расположена картинка заклинания на текстуре (на одной текстуре по столбцам и строкам находится по восемь картинок, а всего их - 64) |
| typeName | String | Тип объекта в текстовом формате |
| type | Integer | Тип объекта в числовом формате |
| name | String | Внутриигровое название заклинания |
| filter | Integer | Код, по которому в окне книги заклинаний фильтруются заклинания |
| spellClass | Integer | Числовое обозначение школы, к которой относится заклинание |
| spellClassName | String | Текстовое обозначение школы заклинания |
| effect | Integer | Номер эффекта заклинания, например, 0 – укрепление здоровья, 1 – урон огнём |
| power | Double | Сила эффекта, например, для укрепления здоровья – величина, на которую увеличивается здоровье цели заклинания |
| time | Integer | Время действия эффекта |
| difficult | Integer | Сложность заклинания, если заклинание слишком сложно для персонажа, он не может его использовать |
| mana | Integer | Количество очков запаса сил, который нужно потратить на заклинание |
| distant | Integer | Расстояние до цели, с которого можно читать заклинание |
| targets | Integer | Количество целей для заклинания |
| targetType | Integer | Тип цели для заклинания (персонаж, область или бесцельное) |
| castAnim | String | Текстура анимации произнесения заклинания |
| attackSpeed | Double | Базовая скорость произнесения заклинания |
| projectileTexture | String | Текстура снаряда заклинания |
| projectileSizeX, projectileSizeY | Integer | Размеры снаряда заклинания |
| projectileSpeed | Double | Скорость снаряда заклинания |
| projectileFrames | Integer | Количество кадров в анимации снаряда заклинания |
| projectileExplode | String | Текстура взрыва снаряда заклинания |
| projectileExplodeFrames | Integer | Количество кадров в анимации взрыва заклинания |
| projectileExplodeRadius | Double | Радиус взрыва заклинания |

# Актёры

Всё, что используется игрой, называется актёрами. Каждый актёр обладает своей игровой логикой, методом рисования и списком параметров, которые отображаются на экране, когда на актёра наводится мышь. Большинство актёров занимают один тайл – это квадрат стороной в 32 пикселя. При наведении мыши на актёра появляется небольшое окно, привязанное к курсору мыши, в котором отображается информация об объекте под курсором мыши. Текст информации определяется для каждого актёра отдельно. Например, для двери это её ID и сложность замка, для звука окружения – проигрываемый файл и расстояние, на котором слышен звук.

# Объекты окружения

## Активаторы

Активаторы – это или рычаги, с которыми может взаимодействовать персонаж, или нажимные пластины, которые срабатывают, когда по ним персонаж проходит. Для создания активатора требуется указать его ID в базе данных и его позицию на игровом пространстве. Также для отслеживания анимации активации каждому активатору присваивается переменная openingClosing, которая показывает, был ли активатор использован (0-не был, 1 - был).

Использование активатора. Если игрок указывает в качестве цели перемещения активатор (вида рычага), персонаж идёт к нему. Когда персонаж может дотянуться до активатора, сравнивается его умение «ловкость рук» и сложность самого активатора. Если умение игрока выше или равно сложности активатора, запускается анимация использования.

Логика активатора. Пока с активатором не взаимодействовали, с ним ничего не происходит. Но как только его успешно использовали, начинает проигрываться анимация активации. Если активатор представляет собой нажимную пластину, тогда игра каждую секунду проверяет, не наступил ли на него подконтрольный игроку персонаж. Как только активатор использован, выполняется особое событие, с ним связанное. Например, активируется ловушка или открывается дверь.

Рисование активатора. Ниже представлен рисунок с текстурой активаторов. На одном рисунке размещается до четырёх активаторов, и каждый активатор имеет четыре кадра анимации использования (верхний кадр – активатор не использовался, нижний – анимация активации полностью проиграна).



Для отображения активатора на экране используется один кадр из 16-ти возможных. Нужный кадр по оси X определяется через параметры самого активатора, а по оси Y – через текущий кадр анимации использования активатора.

## Анимированные тайлы

В отличие от обычных тайлов, анимированные проигрывают циклично заданную им анимацию. Таким тайлом может являться вода, анимация используется для отображения волн на поверхности воды. На игровое пространство тайл загружается как и активатор.

Использование анимированного тайла. Его нельзя его использовать.

Логика анимированного тайла. Такие тайлы постоянно в цикле прокручивают свою анимацию.

Рисование анимированного тайла. Так как количество кадров в анимации каждого тайла может быть разным, для каждого отдельного тайла выделяется свой файл с текстурой анимации. Из этого файла вырезается текущий кадр анимации и отображается на требуемой картой месте. На рисунке ниже изображена четырёхкадровая анимация для тайла с водой.



## Источники звука окружения

К звукам окружения относятся: шум воды, стрекотание сверчков, вой волков и другие звуки природы. Чтобы указать игре место, где нужно проигрывать такие звуки, используются источники звука, которые загружаются на карту как и остальные объекты.

Использование звука окружения. Персонажи не могут взаимодействовать со звуками окружения.

Логика звука окружения. Звук окружения может находится в двух состояниях, задаваемых переменной playing: звук проигрывается или его воспроизведение приостановлено; в последнем случае источник звука ничего не делает. Если звук проигрывается, то игра определяет громкость проигрываемого звука, которая зависит от расстояния между источником и камерой игрока, а также расстоянием, на котором звук может быть услышан. Если расстояние между источником и камерой больше этого расстояния, звук не проигрывается, иначе чем ближе камера к источнику звука, тем громче звук.

Рисование звука окружения. Звук окружения не рисуется на экране во время игры. В редакторе карт звук окружения обозначается специальным символом, а вокруг него рисуется зона, в которой звук может быть услышан (как показано на рисунке ниже).



## Контейнеры

Контейнеры – это различные сундуки, ящики, бочки – различные объекты, в которых могут лежать вещи. Если персонаж игрока сможет открыть контейнер, то он сможет забрать всё его содержимое. Для загрузки сундука требуется указать его ID, позицию на карте, а также его содержимое (ID предмета и его количество в сундуке).

Использование контейнеров. Если игрок указывает в качестве цели перемещения контейнер персонаж идёт к точке открывания контейнера (обычно она на тайл ниже самого сундука). Когда персонаж дойдёт до этой точки, сравнивается его умение «ловкость рук» и сложность замка на контейнере. Если умение игрока выше или равно сложности замка, запускается анимация открывания сундука, и в инвентарь игрока добавляется содержимое контейнера. Уже открытый контейнер нельзя повторно использовать.

Логика контейнеров. Пока контейнер не открывали, он просто стоит. Но как только персонаж его использовал, начинает проигрываться анимация открытия.

Рисование контейнера. Рисование контейнера реализовано аналогично рисованию активаторов, ниже представлен рисунок с контейнерами.



Если контейнер заперт, тогда на нём дополнительно рисуется замок: .

## Двери

Двери – это объекты, которые преграждают дорогу персонажам в коридорах замков или подземелий. Если навыки персонажа достаточно развиты, он сможет открыть запертую дверь, но пока не определено, давать ли возможность игроку закрывать двери.

Использование дверей. Дверь используется как и контейнеры, но при её открывании по тайлу, который она занимает, можно будет пройти.

Логика дверей. Пока дверь не открывали, она просто стоит. Но как только персонаж её использовал, начинает проигрываться анимация открытия(!).

Рисование дверей. Такое же как и рисование контейнеров.

## Структуры

Структуры – это особые объекты, которые дополняют поверхность карты. К ним относятся, например, камни, фонари, мебель и др. Каждая структура состоит из двух частей: прозрачной и твёрдой.

Использование структур. Игрок со структурами не взаимодействует напрямую.

Логика структур. Игра проверяет, находится ли на одном тайле структура или персонаж группы игрока. Если находится, то прозрачная часть структуры не отображается на экране.

Рисование структур. Ниже представлена текстура, на которой находится несколько структур. С помощью параметров структур из текстуры вырезается нужная часть картинки, которая затем отображается на экране.



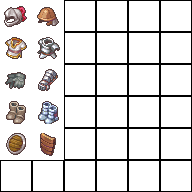
Также ниже представлено, как рисуются структуры, когда персонаж игрока стоит на структуре и если не стоит. Следует отметить, что прозрачная и твёрдая части рисуются отдельно.



## Предметы

Предметы так же можно причислить к объектам окружения, когда они лежат на игровом пространстве. Если предмет становится целью движения персонажа, то он (когда персонаж дошёл до предмета) удаляется с карты и перемещается в инвентарь групп игрока. Но у предметов не прописана логика, так как они просто лежат на карте и у них нет анимации или другой специфической логики.

Рисование предметов. На рисунке ниже изображена текстура с предметами. Для отображения на игровом пространстве из текстуры вырезается требуемая картинка предмета, и она рисуется на экране.

# Игровые объекты

Данные вид объектов используется как связующее звено между персонажами и объектами окружения, реализуя взаимодействия между ними. К таким объектам относятся: карта, тайлы, снаряды и управляющий игрой.

## Управляющий игрой

Как понятно из названия, данный вид объекта контролирует все игровые процессы на базовом уровне. Ниже перечислены все реализованные на данный момент процессы.

Управление временем.

В управление времени входит две задачи: управление суточным циклом в игровом мире, то есть реализация цикла смены дня и ночи и управление скоростью игры, то есть скоростью игровых процессов.

За управление суточным циклом отвечают две переменные time – текущий момент времени - и timeMax – верхняя граница переменной time, как только она достигается, начинается новый день и переменная time сбрасывается на 0. В дальнейшем смена дня и ночи будет влиять на освещение игрового пространства. Ниже представлена таблица соответствия значения переменной time, реальному времени и виртуальному времени в игровом мире.

|  |  |  |
| --- | --- | --- |
| time | Виртуальное время | Реальное время |
| 0 | 00:00 | 00:00:00 |
| 100 | 00:01 | 00:00:01 |
| 1000 | 00:10 | 00:00:10 |
| 3000 | 00:30 | 00:00:30 |
| 6000 | 01:00 | 00:01:00 |
| 12000 | 02:00 | 00:02:00 |
| 36000 | 06:00 | 00:06:00 |
| 72000 | 12:00 | 00:12:00 |
| 144000 | 24:00 | 00:24:00 |

Как видно по таблице, за одну секунду реального времени проходит одна минута виртуального, а весь виртуальный суточный цикл проходит за 24 минуты реального времени. Значение переменной time увеличивается на единицу за каждый игровой цикл, в секунду происходит 100 таких циклов. Для отображения текущего времени на экране используется следующая картинка.



Игрок видит лишь часть изображения, которая показывает текущее время дня, он называется индикатором времени..



Игрок может менять скорость игровых процессов. Существует три скорости игры: пауза (все игровые процессы приостановлены, однако игрок может отдавать команды своим персонажам, после снятия игры с паузы, они выполнят полученные команды), обычная скорость и двойная скорость (все игровые процессы ускорены в два раза, например, суточный цикл пройдёт за 12 минут). На рисунке ниже красными кругами выделены индикаторы скорости игры. Они крутятся вокруг индикатора времени, если игра не поставлена на паузу.



!важно, при установке игры на паузу запоминается предыдущая скорость игры, при снятии игры с паузы игра возвращается на ту скорость. То есть, если игрок поставил игру на паузу при ускоренной скорости, то при снятии игры с паузы, она перейдёт обратно на ускоренную скорость, а не на обычную.

Управление содержимым карты.

Всё содержимое текущей карты (структуры, предметы, персонажи и пр) помещается в управляющий игрой объект. Каждому объекту присваивается локальный ID, по которому можно обратиться к конкретному объекту на карте. Каждый тип содержимого помещаются в соответствующий список, и с помощью управления содержимым карты, в них можно добавлять и удалять объекты. Сюда же относится метод, который очищает карту от всего содержимого, чтобы загрузить другую карту, и заполнить списки объектов её содержимым. Здесь же реализованы загрузка и сохранение карт, но об этом в соответствующем разделе.

Управление группой игрока.

Игрок может управлять несколькими персонажами, которые находятся в группе игрока. Они помещаются в отдельный список, а специальный маркер показывает, какой конкретный игрок в данный момент находится в фокусе управления. В этот же блок входит инвентарь игрока, то есть список всех предметов, которыми владеет игрок и книга заклинаний, список всех заклинаний в распоряжении игрока. Здесь же реализовано процессы изменения всех трёх списков.

Управление музыкой.

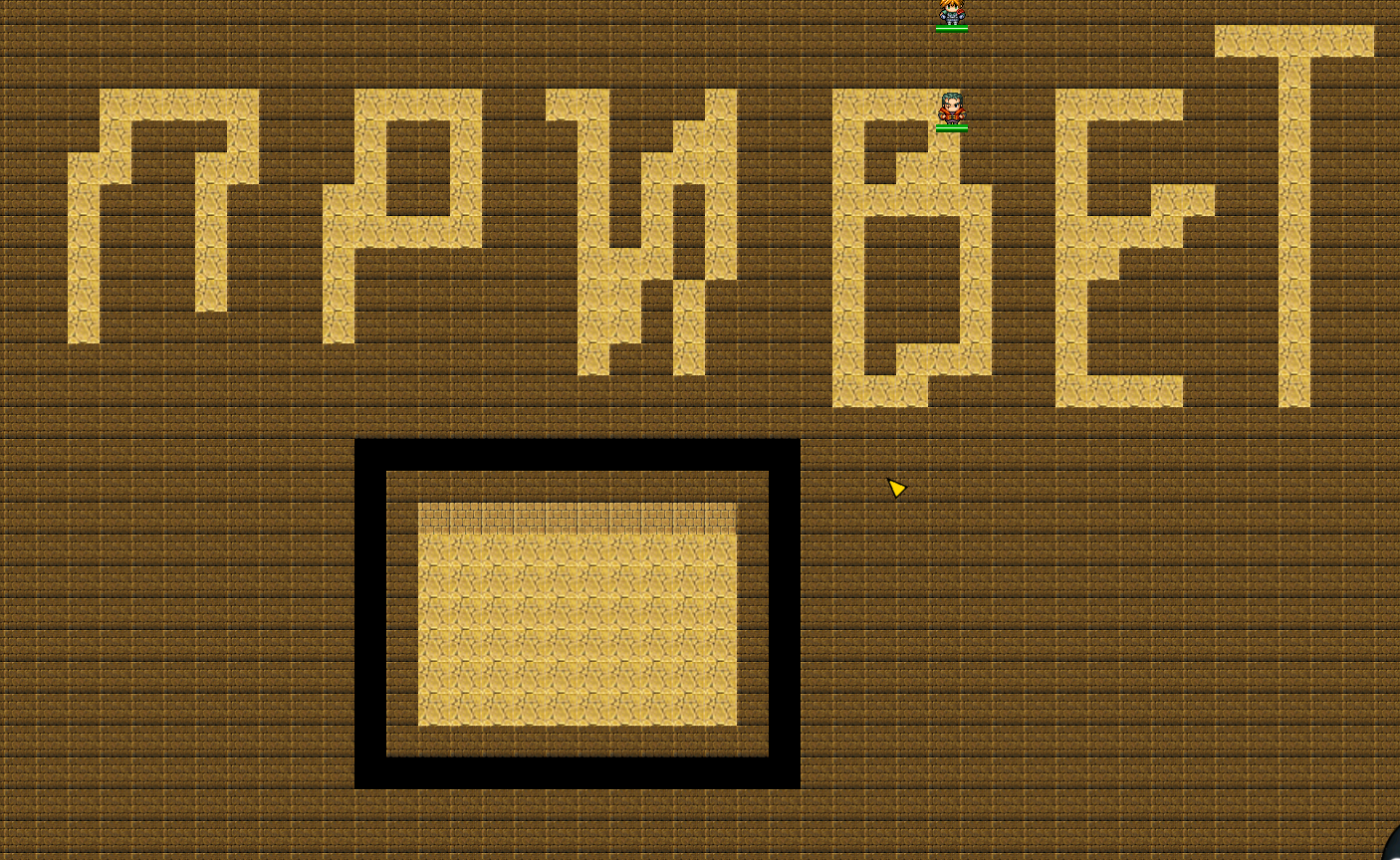
Во время игры игрока сопровождает музыка. Она делится на два типа: исследование и бой. Второй тип играет, когда персонажам игрока угрожает опасность. Воспроизводимая музыка выбирается из списков доступной музыки для конкретной карты, для каждого типа создаётся свой список музыки. Музыку следует менять в трёх случаях, когда происходит: начало боя, завершение боя, воспроизведение текущей музыки закончилось. При начале боя музыка исследования резко сменяется на боевую. При завершении боя боевая музыка начинает плавно затихать, пока её громкость не упадёт до нуля, тогда начинает играть музыка исследования. Если во время затихания снова начался бой, тогда боевая музыка обратно постепенно повышает громкость. Если воспроизведение музыки закончилось, через некоторое время из того же списка выбирается следующая композиция, если игрок не в бою, иначе музыка ставится на повторение.

## Тайлы

Тайл – это квадрат стороной 32 пикселя, единица измерения расстояния в игре и главная составляющая карт. Каждый тайл хранит текстуру и границы тайла на этой текстуре. Текстура с тайлами представлена на рисунке ниже.



Из таких тайлов собирается карта, например, как на рисунке ниже.



Также каждый тайл обладает уровнем проходимости. Если он равен -1, тогда через тайл нельзя видеть и ходить по нему. Если он равен 0, по тайлу нельзя ходить, но он не мешает визуальному обнаружению других персонажей. Если уровень проходимости больше нуля, тогда по тайлу можно ходить. Тогда уровень проходимости показывает, с какой скоростью по тайлу можно ходить, например, уровень проходимости 0.5 означает, что скорость персонажа на данном тайле упадёт в два раза.

## Снаряды

Используются как переносчики воздействия между персонажами и бывает одного вида: снаряд ближнего боя, который представляет собой анимацию атаки, а также хранит наносимый урон и тип урона.



Она проигрывается при каждой атаки оружием ближнего боя. Один из кадров помечен как кадр атаки. Когда анимация доходит до него, и враждебный персонаж пересекает снаряд – засчитывается попадание и рассчитывается урон. Как только снаряд проигрывает анимацию до конца, он удаляется из игрового мира.

## Карта

Место действия игровых событий. Каждая карта заполняется тайлами и различными игровыми объектами. Как обещал раньше, расскажу, как происходит сохранение и загрузка карт. Файл с сохранением карты состоит из нескольких строк.

На первой строке записывается общая информация о карте, её внутриигровое название, размеры, в дальнейшем планируется добавить списки музыки для карты, а также тип региона карты, который будет регулировать погодные явления и влияние суточного цикла на изменение освещения карты.

На второй строке через точку с запятой записаны тайлы, из которых состоит карта. Для каждого тайла через двоеточие записывается текстура тайла и его позиция на текстуре, а также проходимость тайла.

На третьей строке записываются предметы, лежащие на карте. Записывается их ID, положение на карте и количество предметов.

На четвёртой строке записываются двери: их позиция и ID.

На пятой строке – анимированные тайлы, также ID и позиция.

Шестая строка хранит данные об активаторах на карте, не трудно догадаться какие.

Седьмая строка хранит расположение структур (камней, полок и пр) на карте.

Восьмая строка состоит из данных обо всех источниках звуков на карте.

Девятая строка содержит информацию о персонажах на карте, но кроме стандартных данных, она содержит модель поведения персонажа, когда он не в бою. Это может быть прогулка в выделенной зоне или патрулирование по заданному маршруту.

Десятая строка содержит информацию о сундуках, в том числе и их содержимое.

При сохранении карты из объекта, управляющего игрой, в файл записываются все описанные выше данные. При загрузке соответственно наоборот, данные из файла загружаются на объект, управляющий игрой.

# Магическая система

## Школы магии

В игре существуют следующие школы магии: магия огня, воды, воздуха, земли, разума, волшебство и сакральная магия. В каждой школе от десяти до сотни магических эффектов, код их действия описан в отдельных классах для каждой школы. Эффекты бывают двух типов: единоразовые и постоянные. Первые действуют один момент времени, при наложении эффекта. Вторые в свою очередь делятся: периодические и непериодические. Периодические воздействуют на персонажа каждый момент времени, в течении которого действует заклинание. Непериодические висят на персонаже до тех пор, пока не кончится время их действия и они снимутся с персонажа. Приведу примеры каждого вида эффекта.

Единоразовый: огненный урон, снимает 20 единиц здоровья, при попадании.

Периодические постоянные: отравление, снимает по 50 единиц здоровья в течении 10 секунд.

Непериодические постоянные: укрепление здоровья, увеличивает максимальное здоровье персонажа на 100 единиц на минуту.

Действие каждого эффекта описывается отдельно, исключая эффекты прямого магического урона по здоровью.

## Базовые понятия магической системы

Во главе угла всей системы лежат коды магических эффектов. Для каждой школы зарезервировано по сотне магических эффектов, тогда всего доступно 700 магических эффектов. При наложении эффекта по его номеру определяется школа заклинаний, и программа переходит в нужный файл, в котором описано действие заклинаний определённой магической школы. В базовом виде магические эффекты временно меняют различные параметры персонажей. Частный случай – эффект уменьшающий здоровье. Сила действия таких эффектов уменьшается на защиту персонажа от данного вида урона. Если урон периодический, тогда полученное значение размазывается во времени. Вспоминая предыдущий пример (Периодические постоянные: отравление, снимает по 50 единиц здоровья в течении 10 секунд), допустим, что у персонажа защита от отравления равна 20. Тогда чистый урон по персонажу (50-20) = 30 единиц. Эффект действует 10 секунд, то есть за одну секунду персонаж теряет три единицы здоровья.

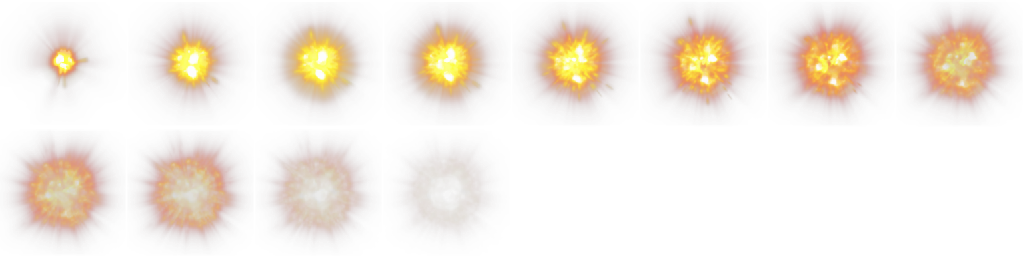
Магические эффекты также делятся на негативные и позитивные. Негативные эффекты заставляют персонажей, на которых наложен данный эффект, атаковать того, кто наложил данный эффект. Эффекты, наносящие урон здоровью, по умолчанию относятся к негативным.

## Магические эффекты

У каждого персонажа имеется список действующих на него эффектов. По типу перенесения эффекты бывают двух видов: проходящие и перманентные. Первые проходят в течении времени. Вторые действуют до тех пор, пока их вручную не снимут. Перманентные эффекты могут быть наложены, если на персонажа надеть особый предмет, например, кольцо регенерации, которое будет восстанавливать здоровье персонажа.

## Магические снаряды

В отличии от снарядов ближнего боя магические снаряды испускаются одним персонажем и летят в цель. Когда они долетят или врежутся в преграду они взрываются: эффект, лежащий в основе снаряда заклинания, накладывается на всех персонажей в радиусе взрыва. Во время полёта заклинания проигрывается анимация снаряда. При взрыве проигрывается анимация взрыва. Ниже представлена текстура анимации взрыва снаряда.



Большинство снарядов проигрывают анимацию взрыва один раз и один раз накладывают магические эффекты. Но некоторые снаряды имеют продолжительную анимацию взрыва. Например, заклинание огненного шторма оставляет на игровой зоне область, в которой бушует огонь в течении 10 секунд. Значит и анимация взрыва будет проигрываться циклически в течении 10 секунд, и эффекты также накладываются в течении проигрывания анимации взрыва. Урон от таких снарядов также считается периодическим но урон наносится только когда персонаж находится в радиусе взрыва. Если базовая скорость снаряда меньше нуля, тогда снаряд долетает до цели мгновенно и сразу же взрывается.

## Заклинания

Объекты, которые используют персонажи для генерации снарядов заклинаний, называются заклинаниями. Доступные игроку заклинания помещаются в общую книгу заклинаний, откуда их можно передать персонажам, находящимися под контролем игрока. Однако заклинания имеют требования к персонажу, чтобы их можно было использовать. Это требуемый запас сил на прочтение заклинания и сложность заклинания. Если у персонажа не хватает сил или у него недостаточные познания в школе магии данного заклинания, заклинание нельзя передать персонажу и, конечно, персонаж не сможет использовать заклинание.

Кроме требований к запасу сил и познаниям персонажа заклинание требует указать цель. Целью может быть персонаж или область на карте, некоторые заклинания автоматически нацеливаются на использующего заклинание. После указания цели проигрывается анимация произнесения заклинания, время произнесения заклинания зависит от скорости действий персонажа. Когда анимация проигрывается до конца, в цель запускается снаряд заклинания, а персонаж теряет запас сил ровно на столько, сколько требует заклинание.

# Персонаж

## Ролевая система

Ролевая система состоит из двух частей: личные параметры персонажей и надетых на персонажей предметы.

|  |  |
| --- | --- |
| Атрибуты | |
| Сила | Влияет на максимальное здоровье, нагрузку и урон в ближнем бою |
| Ловкость | Влияет на максимальный запас сил, скорость атак и урон в дальнем бою |
| Разум | Влияет на стоимость произнесения заклинаний и познания в видах магии |
| Основные параметры | |
| Здоровье | Количество урона, которое может перенести персонаж перед смертью |
| Запас сил | Количество очков, которые тратятся на бег и произнесение заклинаний |
| Нагрузка | Общий вес надетого на персонажа снаряжения, который негативно не повлияет на персонажа |
| Скорость атак | Скорость нанесения ударов и чтения заклинаний |
| Познания в видах магии | Максимальная сложность заклинаний, которые может читать игрок |
| Восстановление | Определяет скорость восстановления запаса сил |
| Взлом замков | Определяет насколько сложные замки может вскрывать игрок |

Отдельным параметром является опыт. Персонажи зарабатывают его за убийство врагов и выполнение заданий. Чем больше общий набранный опыт, тем выше здоровье и запас сил персонажа, также опыт можно потратить на повышение навыков и умений, которые описаны ниже.

|  |  |
| --- | --- |
| Живучесть | Увеличивает максимальное здоровье |
| Выносливость | Увеличивает максимальный запас сил |
| Восстановление | Увеличивает параметр восстановления |
| Физическая мощь | Увеличивает максимальную нагрузку |
| Подвижность | Увеличивает скорость атак |
| Навыки владения оружием | Увеличивает наносимый урон, при использовании определённого вида оружия |
| Навыки в магических школах | Увеличивает познание в конкретном виде магии |
| Ловкость рук | Повышает параметр взлом замков |

Базовое стоимость повышения навыка – 100 очков опыта. К этой стоимости добавляется 100 очков, за каждый уровень развития навыка и ещё 50 за каждый уровень во всех навыках.

Персонаж может носить два вида оружия и щита (основное и вторичное), чтобы менять их во время боя. Шлем, бронник, поножи и перчатки – броня, а также восемь расходников и заклинаний. Всё надетое снаряжение повышает нагрузку на персонажа. Оружие определяет урон, тип урона, скорость атак и их дистанцию для персонажа, а броня защиту от различных видов урона.

Как работает восстановление. Запас сил восстанавливается со временем порционно. То есть часть запаса сил восстанавливается через заданные интервалы, которые зависят от параметра Восстановление. Размер восстанавливаемой порции зависит от перегруза персонажа: чем выше перегруз, тем меньше восстанавливаемая порция. Интервал между восстановлением порций сбрасывается, когда персонаж двигается, атакует или читает заклинания.

Размер получаемого опыта за убитых врагов или за выполнение заданий зависит от уже набранного персонажем опыта. Если убить слабого врага, то опыта за него можно вовсе не получить. Формула расчёта получаемого опыта:

Текущий общий опыт персонажа (xp)

Опыт врага (xpE)

Полученный опыт (xpGain)

Если (xp – xpE) < 250 тогда xpGain = 0; Персонаж гораздо сильнее убитого врага

Иначе если (xp – xpE) > 0 тогда xpGain = (xp – xpE)/25; Персонаж не особо сильнее убитого врага

Иначе xpGain = (xpE – xp)/25+10; Персонаж слабее убитого врага

## Автоматический контроль персонажей игрока

Так как управлять несколькими персонажами сложно, особенно в пылу боя, в игру введена система простого автоконтроля. Персонаж может находиться о одном из четырёх состояний автоконтроля: ожидание, контратака, боевая готовность, защитная стойка. В первом состоянии персонаж свободен от автоматического контроля и подчиняется только командам игрока.

Состояние Боевая готовность. В этом режиме, если у персонажа нет другой команды, он будет атаковать всех враждебных персонажей, которые появятся в его поле зрения.

Состояние Контратака. В этом режиме, персонаж атакует врагов, в случае, если его или его сопартийца атаковали первым.

Состояние Защитная стойка. В этом режиме персонаж не атакует противников, а встаёт в защитную стойку и принимает все атаки в свою сторону на щит. Для этого он поворачивается в сторону атакующего противника.

## Неигровые персонажи

Всеми действиями неигровых персонажей управляет компьютер. Неигровой персонаж (НИП), может находится в следующих состояниях тревоги: спокоен, подозрителен, встревожен, враждебен.

В спокойном состоянии НИП действует согласно спокойному пакету поведения, который записан на файле карты для данного персонажа. Пакеты поведения делятся на два вида (пока что): патрулирование и прогулка. При патрулировании НИП следует по заранее проведённому для него маршруту. Для каждой точки маршрута указываются её координаты и время, которое НИП стоит на данной точке. Как только НИП дошёл до последней точки маршрута, его целью становится первая точка в маршруте. При прогулке указывается зона прогулки НИП и границы интервала до получения новой цели. Новую цель движения НИП выбирает случайным образом из зоны прогулки. Затем он выбирает время, по истечение которого персонаж выберет новую цель движения. (Цель движения – координаты на игровом пространстве, куда идёт персонаж).

В спокойном состоянии НИП постоянно сканирует местность на подозрительные события. Если враждебный НИПу персонаж окажется в его поле зрения, НИП немедленно поднимет тревогу и начнёт атаку замеченного персонажа. Если враждебный НИПу персонаж пробежит у него за спиной, или подойдёт слишком близко со спины, у НИПа активируется таймер перехода в режим тревоги, по истечение которого он перейдёт в подозрительное состояние. Также в подозрительное состояние НИП переходит, когда слышит тревогу, в него попадает снаряд или снаряд пролетает рядом с ним. Поле зрения НИПа зависит от его параметров дальности зрения и типа зрения (обычное, круговое, рентгеновое).

В подозрительном состоянии НИП фиксирует точку, которая называется точкой тревоги, и бежит в её сторону, не переставая сканировать местность. Когда он добежит до точки тревоги, и какое-то время в ней осмотрится, и если не обнаружит врага, то он возвращается в спокойное состояние.

В враждебном состоянии НИП преследует обнаруженного врага и атакует его, когда тот оказывается в пределах досягаемости атаки. Если НИП теряет из виду всех врагов, то он переходит в встревоженное состояние.

В встревоженном состоянии НИП фиксирует точку потери врага и двигается в неё. После активируется таймер поиска, пока время поиска не истечёт НИП расследует область вокруг зафиксированной точки. Когда таймер истечёт, он перейдёт в спокойное состояние.

## Перемещение

Модуль перемещения решает две задачи: назначение цели движения и само движение к цели.

Целью движения может быть какой-либо объект или просто точка на карте. При нажатии на цель левой кнопкой мыши, игрок задаёт персонажу цель движения. Персонаж незамедлительно выполняет команду (если не проигрывает анимацию атаки, в этом случае он сначала доведёт удар до конца). Во время движения проигрывается анимация ходьбы, а сам персонаж поворачивается в сторону движения.

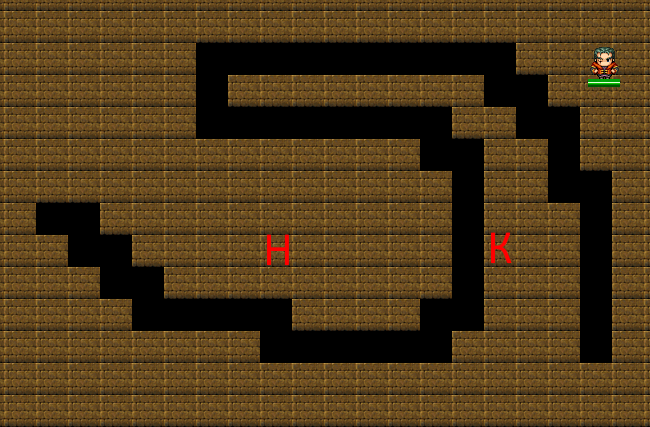
Игрок может задать скорость движения персонажа от скрытного перемещения до бега. При скрытном перемещении враги не услышат, что персонаж подошёл к ним сзади, а бег расходует запас сил (если он равен нулю, персонаж не может бегать).

Как только персонаж пришёл к цели, он останавливается и взаимодействует с целью движения (например, открывает дверь, дергает рычаг, подбирает предмет или атакует врага).

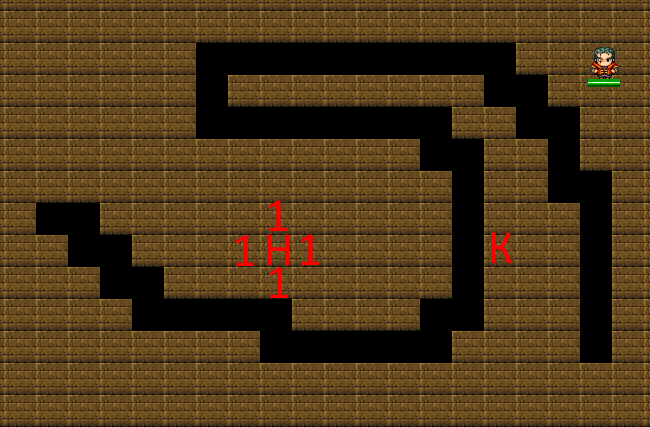
Если персонаж слишком сильно нагружен от теряет скорость, а при сверхвысокой нагрузке теряет возможность бегать.

## Поиск маршрута

Так как между персонажем и целью его движения не всегда может быть прямая дорога, то есть на пути встречаются препятствия, которые нужно обходить, в игру добавлен поиск маршрута. В результате его работы получается маршрут от начальной и конечной точкой движения персонажей в обход всех преград. В основе алгоритма поиска маршрута лежит поиск в ширину. Приведём пример поиска маршрута. На рисунке ниже обозначены буквами Н и К начальная и конечная цель маршрута.



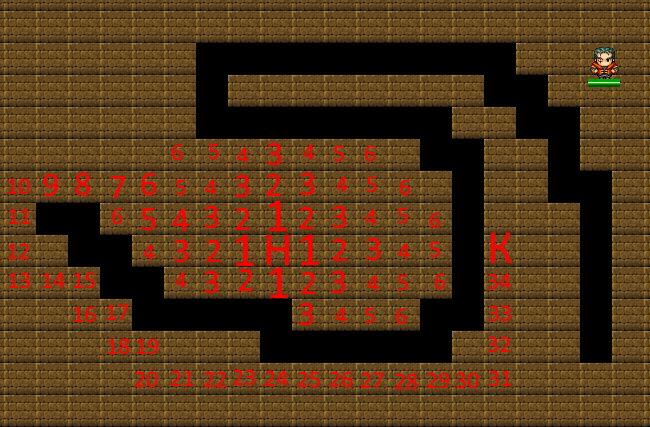
Из начальной точки алгоритм делает шаг на одну клетку во все четыре стороны. Данным клеткам присваивается номер 1.



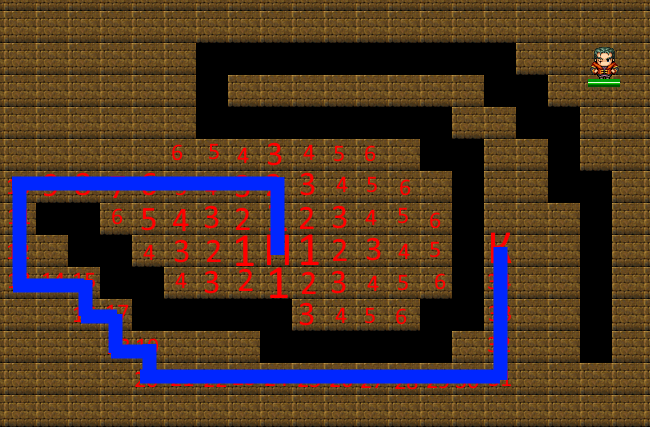
Из этих точек мы тоже делаем шаг во все стороны, но игнорируем клетки, в которые нельзя пройти, или клетки, которые уже помечены номером.



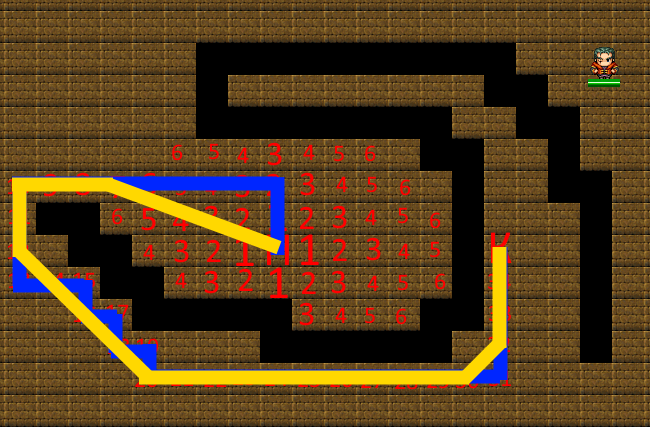
Клетки получают свой номер до тех пор, пока мы не попадём в конечную точку маршрута.



Когда достигнутая конечная точка, начинается восстановление маршрута. То есть алгоритм с конечной точки маршрута перемещается в клетку с номером на единицу меньше текущей клеткой.



На последнем этапе маршрут оптимизируется, чтобы сократить длину маршрута. Проще говоря, срезаются углы.



Полученный маршрут передаётся персонажу, и он проходит по нему к своей цели движения.

## Боевая система

Персонажи способны атаковать друг друга в ближнем бою, тут в игру вступает боевая система. Чтобы приказать персонажу атаковать другого персонажа, на цель атаки нужно навести мышь и нажать ЛКМ. Целью движения подконтрольного персонажа станет враг, и он начнёт преследование, пока не подойдёт на расстояние проведения атаки. Когда добежит, начнёт замахиваться. Во время замаха появляется снаряд и проигрывается его анимация, скорость которой зависит от скорости действия оружия в руках персонажа и его собственной скорости атак. В активный кадр атаки происходит расчёт нанесённого урона.

За основу берётся номинальный урон оружия, который увеличивается в зависимости от навыка персонажа владеть данным видом оружия. Также урон от оружия ближнего боя повышается или понижается в зависимости от Силы персонажа. Если у персонажа сильно повреждены руки, тогда урон от атак снижается. Полученное значение называется «сила атаки».

Со стороны получающего урон персонажа считается следующее: в какую часть тела был нанесён удар, был ли удар принят на щит, был ли удар внезапным. Начнём с последнего, если цель не была встревожена и получила удар, то он считается внезапным, и цель получает увеличенный урон. Если цель приняла удар на щит, тогда урон снижается на показатель защиты щита от данного оружия, а оставшейся урон также уменьшает запасом сил в зависимости от стабильности щита (чем выше стабильность, тем меньше запаса сил потратится на блок). Однако, если запас сил равен нулю, персонаж будет получать увеличенный урон.

После определяется место, куда пришёлся удар. Если удар попал в голову, цель получает увеличенный урон. При ранении руки цель наносит более слабые удары, ноги – начинает медленнее ходить. Кроме этого можно приказать персонажу наносить прицельные удары. В этом случае при прицельных ударах в голову появляется шанс нанести повышенный урон, но также появляется шанс не нанести никакого урона вовсе. Прицельные удары по рукам и ногам наносят меньше урона. Прицельные удары в тело наносят слегка увеличенный урон, но не позволяют нанести повышенный урон или повредить руки или ноги цели.

Когда вычислен нанесённый урон, цель теряет ровно столько же здоровья. Если здоровье упало до нуля, цель умирает. Убивший цель персонаж получает опыт. Мёртвого персонажа можно обобрать, получив золото и предметы, которыми он владел.

Если персонаж носил в руках оружие дальнего боя, тогда при атаке генерируется снаряд заклинания в виде стрелы, а боезапас оружия уменьшается на 1. Если боезапас упал до нуля, из оружия больше нельзя стрелять до пополнения боезапаса.

## Колдовство

Все основные механики применения заклинаний описаны в разделе Магическая система. Здесь стоит добавить пару моментов.

Первое. Требуемый запас сил на произнесение заклинаний зависит от Разума персонажа, чем выще разум, тем меньше сил требуется.

Второе. В игру введены многоцелевые заклинания. То есть помимо выбранного игроком персонажа, целью заклинания становятся и другие персонажи, стоящие рядом с основной целью.

Третье. Заклинание может быть заключено в расходник. В этом случае при его произнесении персонажу не требуются познания в школах магии, однако предмет после применения уничтожается.

## Рисование персонажей

Персонаж рисуется на экране, если кто-то из группы игрока видит его в своём поле зрения или персонаж входит в группу игрока. Также по умолчанию под персонажем отображается текущий уровень его здоровья. Ниже представлена текстура внутриигровая персонажа и отображение персонажей внутри игры (здоровье дружественного персонажа красится зелёной полоской, враждебного – красной).

Для дружественных персонажей на экране также отображается панель с портретом персонажа и индикаторами запаса сил и здоровья. Выбранный персонаж из группы игрока выделяется рамкой.



# Игровые окна

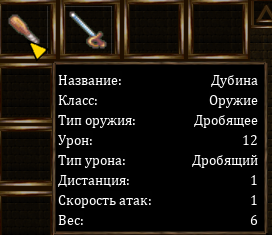
В предыдущих главах рассмотрены процессы, проходящие внутри игры, теперь опишем процессы взаимодействия игры и самого игрока. Игрок управляет игрой через специальные окна, которые называются игровыми. На данный момент реализованы: окно инвентаря, окно развития персонажей, окно книги заклинаний.

## Инвентарь

Окно инвентаря используется для просмотра вещей, которые находятся в распоряжении игрока, и передачи предметов подконтрольным персонажам. Окно показано на рисунке ниже. Под номером один – надетые на персонажа вещи, два – параметры персонажа, три – все вещи, которые есть у игрока (с прокруткой справа и фильтрами по типу предметов снизу). Каждый предмет занимает одну выделенную для него ячейку.



При наведении курсора на предмет, появляется окно с описанием параметров предмета, а сам предмет отмечается как выделенный.



Если нажать ЛКМ когда предмет выделен, персонаж наденет его (если это оружие, броня или расходник). Надетый предмет исчезает из инвентаря и появляется в снаряжении персонажа, вместе с чем меняет его параметры. Если выделить предмет, который надет на героя, и нажать по нему, тогда персонаж его снимет, и предмет вернётся в общий инвентарь. Если ячейка для надеваемого предмета уже занята, персонаж переодевает предмет – это справедливо для брони. Оружие, щиты и расходники занимают первую свободную для них ячейку, но если таких ячеек нет – передача персонажу предмета не происходит. Игрок может дать персонажу два вида оружия, однако если попробовать передать ему третье, ничего не произойдёт, аналогично с расходниками: персонаж может носить только 8 расходников.



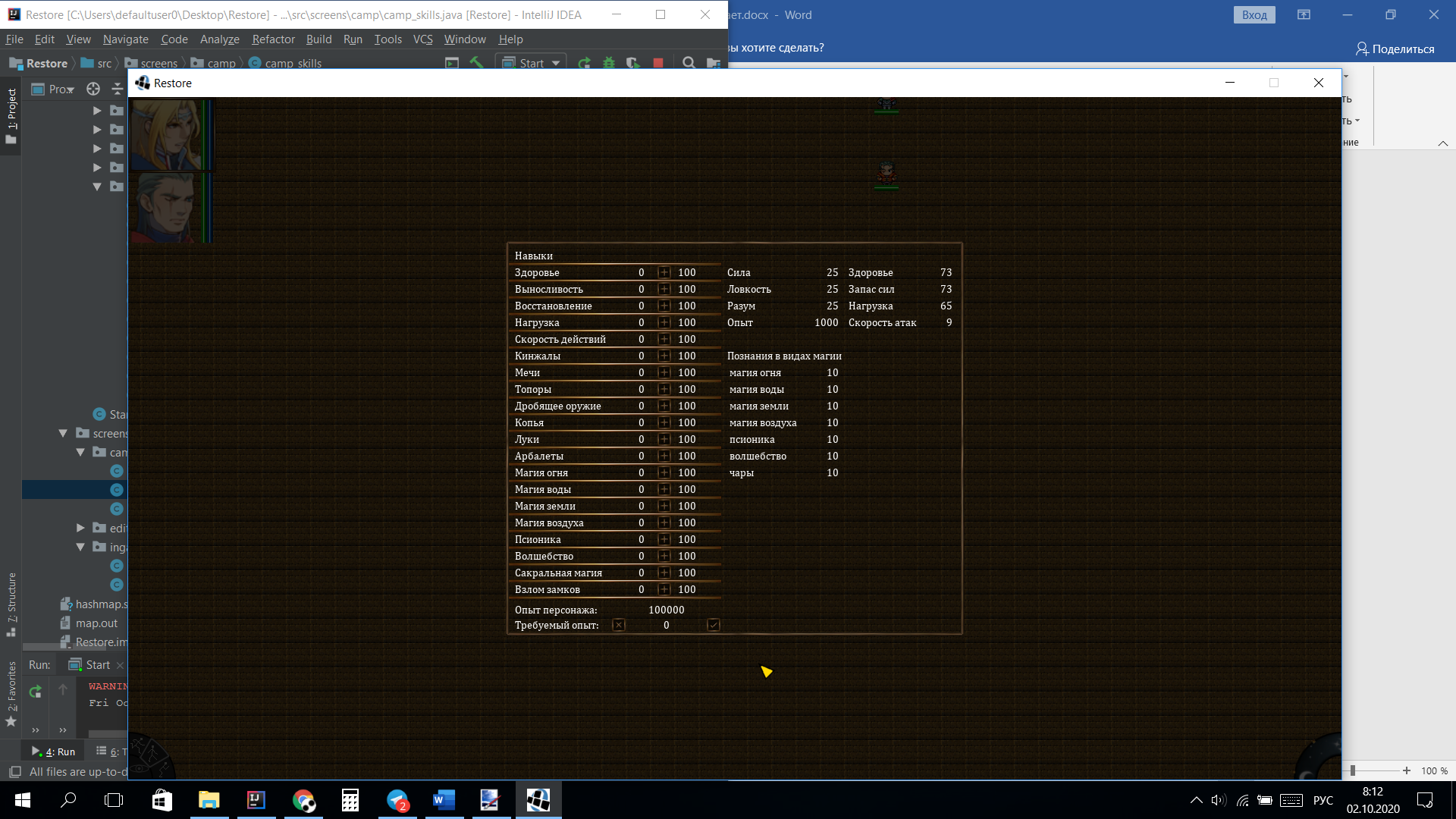
Каждый надетый предмет нагружает персонажа, отношение текущей нагрузки к максимальной называется процентом перегруза. Чем выше процент, тем медленнее персонаж восстанавливает силы, медленнее перемещается, а при полном перегрузе теряет возможность бегать.

Как заполняется инвентарь. В инвентаре можно указать тип предметов, которые хочет видеть игрок – это называется фильтрацией. Возможные фильтры слева-направо: все предметы, оружие ближнего боя, броня, щиты, оружие дальнего боя, расходники, чертежи, материалы, руны, трофеи, ключевые предметы. Если выбрать один из фильтров, в инвентаре будут отображаться только предметы выбранного типа. (Чертежи, материалы, руны, трофеи и ключевые предметы пока в игру не введены).

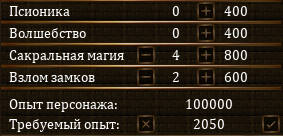


## Окно развития персонажа

В окне развития персонажа, игрок может потратить свободный опыт на повышение навыков.



Справа выведены основные параметры персонажа, а слева перечислены все навыки персонажа и их уровень развития, а внизу количество свободного опыта и требуемого опыта, чтобы поднять навыки.



На примере выше показан результат повышения Сакральной магии на 4 и Взлома замков на 2. Цена повышения Псионики и Волшебства увеличилась с 100 до 400 (плюс 50 очков опыта за каждый уровень развития каждого навыка), для сакральной магии стоимость развития поднялась до 800 (плюс 50 очков опыта за каждый уровень развития каждого навыка и плюс 100 очков за каждый уровень развития самого навыка Сакральная магия), аналогично для Взлома замков:

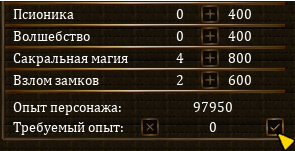
100 + 50 \* 6 + 100\*2 = 100 + 300 + 200 = 600.

Требуемый опыт это сумма очков опыта, которые требуются для поднятия навыков. В нашем случае это 2050 очков. Как получилась это число показано в таблице ниже.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Уровень навыка «Сакральная магия» | Уровень навыка «Взлом замков» | Цена развития навыка «Сакральная магия» | Цена развития навыка «Взлом замков» | Требуемый опыт |
| 0 | 0 | 100 | 100 | 0 |
| 1 | 0 | 250 | 150 | 100 |
| 1 | 1 | 300 | 300 | 250 |
| 1 | 2 | 350 | 450 | 550 |
| 2 | 2 | 500 | 500 | 900 |
| 3 | 2 | 650 | 550 | 1400 |
| 4 | 2 | 800 | 600 | 2050 |

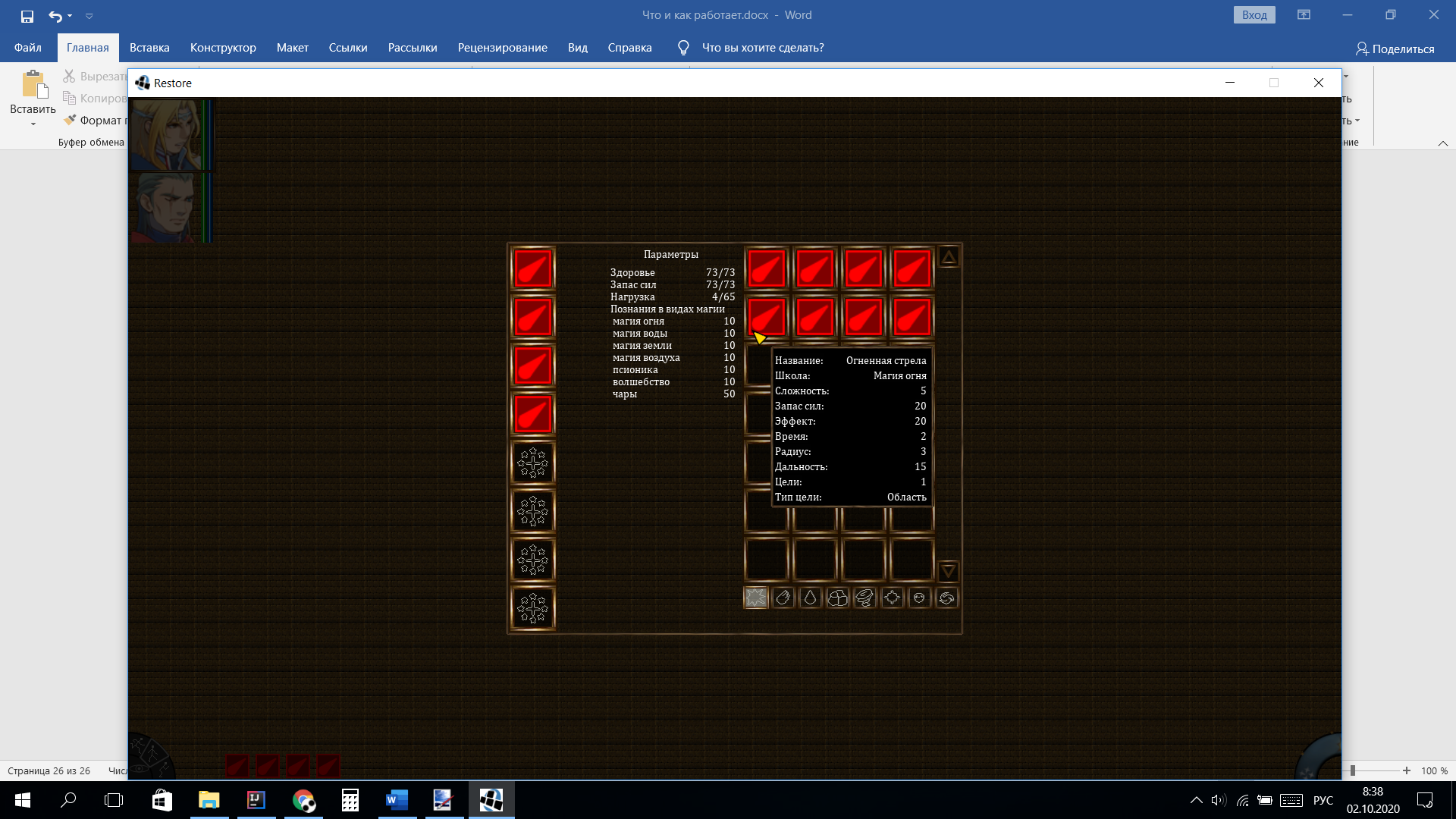
Желтым цветов выделен повышаемый навык, а оранжевым – цена его повышения.

По левую и правую сторону от требуемого опыта находятся кнопки принятия повышения навыков и сброс. Нажав на сброс, игрок вернёт все параметры к начальному виду до повышения, а подтверждение – повышенные навыки зафиксируют новые значения, а требуемый опыт вычитается из свободного опыта персонажа. Стоит отметить, что после подтверждения понизить уровень навыка будет нельзя, как и вернуть потраченный опыт.



## Окно книги заклинаний

Данное окно используется для передачи заклинаний персонажам. В целом взаимодействие с ним аналогично взаимодействию с окном инвентаря, но на нём отображаются заклинания.



Передача заклинаний работает как и передача персонажу расходников, больше воьми заклинаний персонаж взять не может. Также каждое заклинание увеличивает нагрузку на персонажа на единицу. Перед передачей заклинания проверяется возможность персонажа прочитать заклинание (сравнивается максимальный запас сил персонажа и запас сил, требуемый на прочтение заклинания, познания в школе магии заклинания и сложность заклинания). Фильтры в книге заклинаний распределены по школам магии.

## Игровое окно

На игровом окне отображаются все игровые объекты и именно в нём и проходит игра. Так же на нм отображается панель управления персонажем.



В левом нижнем углу находятся панель выбора скорости движения (в данный момент выбрана скорость движения бегом) и вида автоконтроля персонажа (в данный момент выбранный вид – боевая готовность). Правее отображается текущее и запасное оружие и щит (текущее находится выше). Если кликнуть по этим картинкам, персонаж поменяет текущее оружие или щит на запасное и наоборот основное оружие/щит станет запасным. Еще правее отображаются взятые расходники и заклинания. При клике на их картинки расходник/заклинание становятся активными, и игрок может указать цель их применения.