

Архитектура и боль

Механики игры и чемодан “инструментов”

В этом отделе мы рассмотрим подробно механики и их работу. Порядок механик будет идти в порядке разработки начиная от базового рендера, заканчивая логикой игры.

Список механик

- ❖ Рендер
- ❖ Базовый update и fullscreen
- ❖ Существо
- ❖ Игрок
- ❖ Боевая система
- ❖ Репозитории атак
- ❖ Торговец
- ❖ Этаж
- ❖ Логика игры
- ❖ Артефакты и одноразовые предметы

Рендер

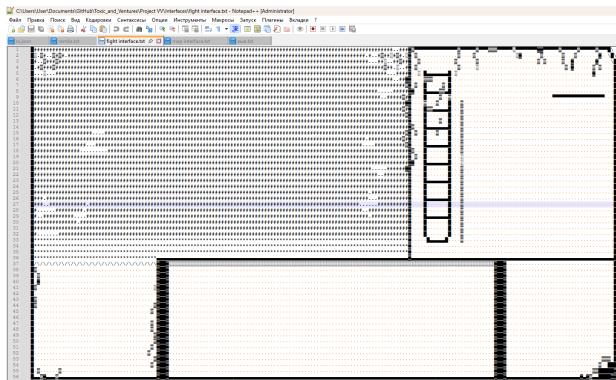
Библиотека BearLibTerminal предоставляет нам функционал для работы с окном, который воспринимает формат символов, а не пикселей и поддерживает многослойность

*Что крайне удобно для ASCII графики)

Сам рендер из себя будет представлять набор методов для удобного и упрощенного взаимодействия с этой библиотекой и внешних файлов проекта отвечающих за элементы интерфейса и текст.

Пример файла интерфейса:

fight interface.txt



*Для любого взаимодействия с интерфейсами советуется скачать дополнительное ПО, такое как Notepad++, где пробелы одинакового размера с обычными символами

Файл текста - представляет из себя json файл с тегами фраз и самими фразами.

Методы

- **Start** или начальная конфигурация в main

`terminal_open();` - для открытия терминала

`terminal_set();` - для установки параметров

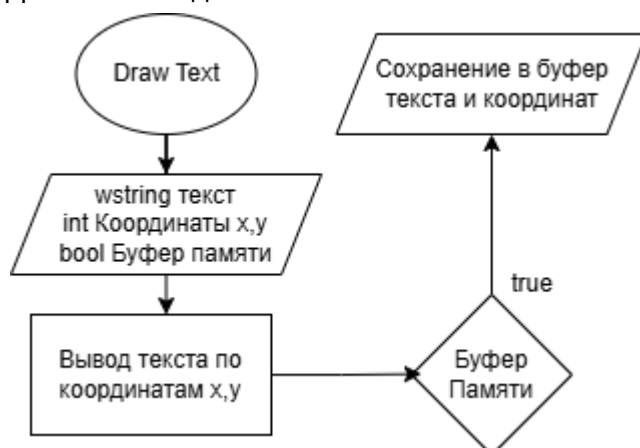
Пример начальных параметров -

"font: fonts/UbuntuMono-Regular.ttf, size=9, codepage=437"

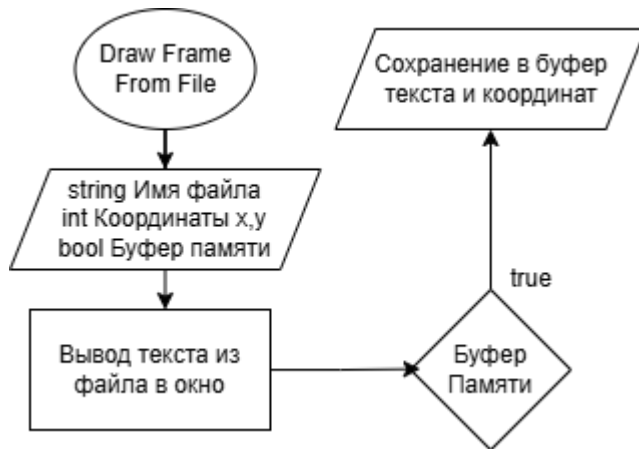
Более подробно о параметрах можно прочитать на [сайте документации BearLibTerminal](#)

- **Draw Text** для отрисовки текста

Здесь поясню, что в библиотеке BearLibTerminal, при изменении экрана на фуллскрин ломается часть изображения, из за чего мы храним имена выводимых файлов и координаты для последующей перерисовки изображения. Далее сама идея



- **Draw Frame From File** для отрисовки фреймов



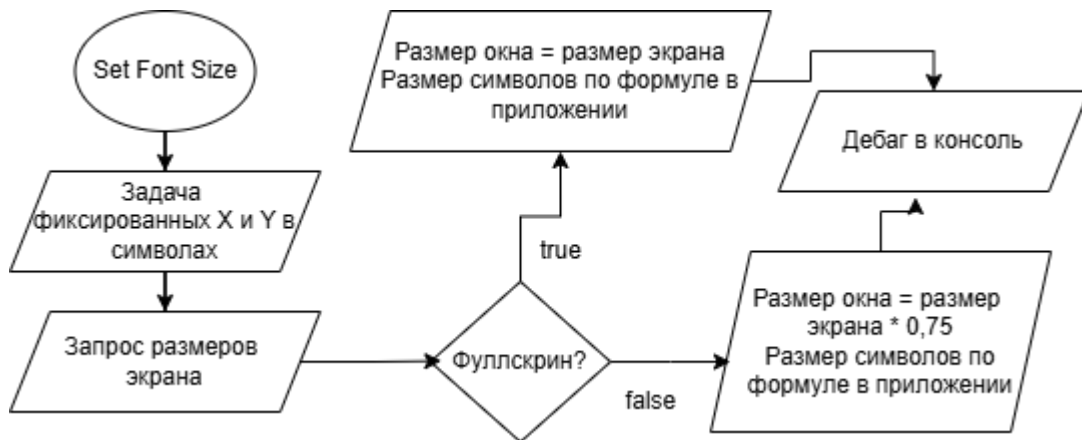
- **Set Font Size** для автоматической настройки размера шрифта при изменении окна

*Формулы

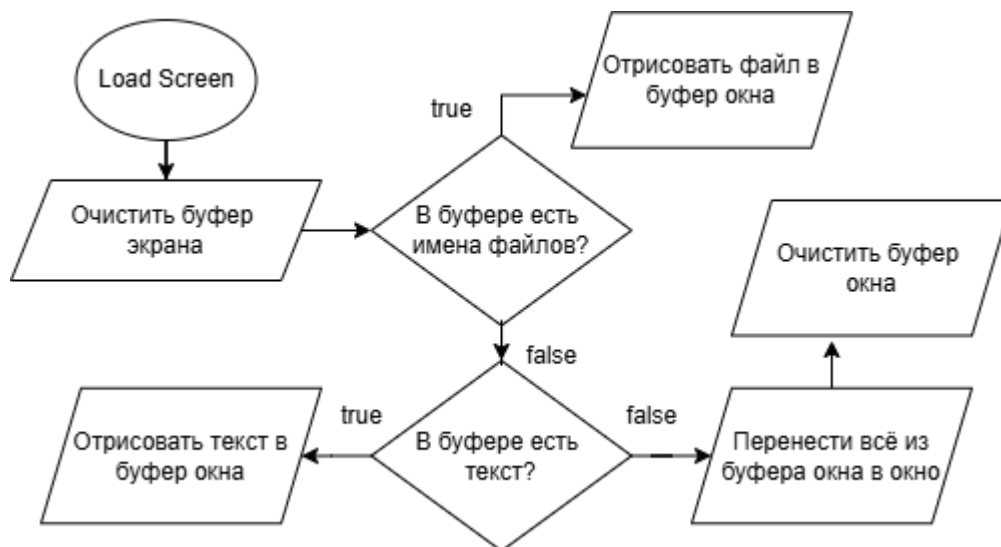
$C_Weight = \max(5, windowWidth / cellCountX);$

$C_Height = \max(10, windowHeight / cellCountY);$

$fontSize = \min(C_Height, C_Width * 1.7);$



- **Load Screen** для перерисовки отображения в случае изменения размера экрана



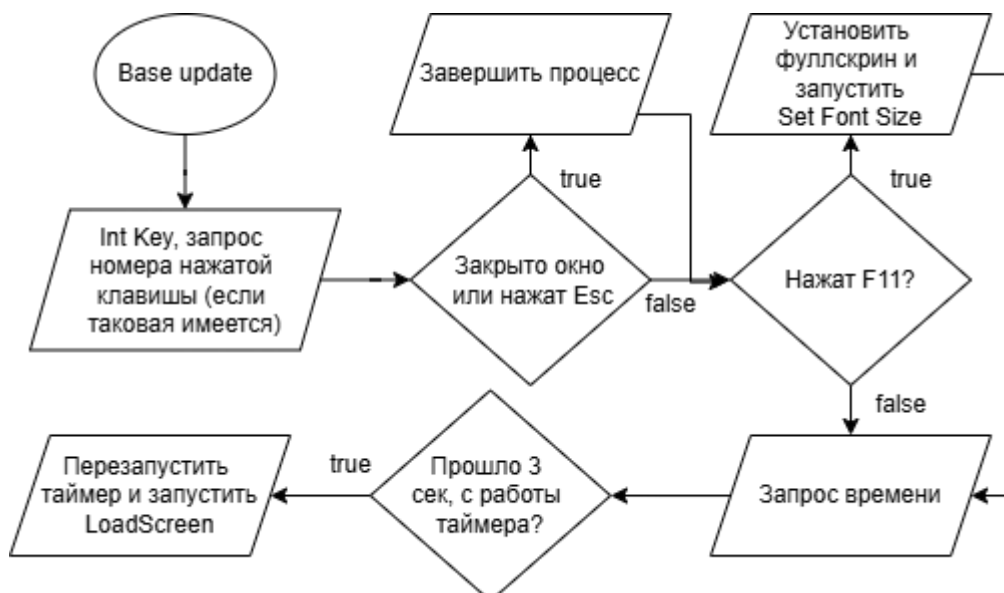
- **Clear Terminal** для очистки экрана и буфера

Очищение буфера и окна

Базовый update

Метод для упрощения дальнейших циклов, который будет принимать key(номер нажатой клавиши) и проверять состояние окна, нажатие кнопки F11 для изменения окна и таймер в 3 секунды, который перерисовывает картинку из буфера.

*Перерисовка нужна для взаимодействия с интерфейсами в прямом времени для разработчика и исправления непредвиденных ситуаций с ломанием окна, для игрока

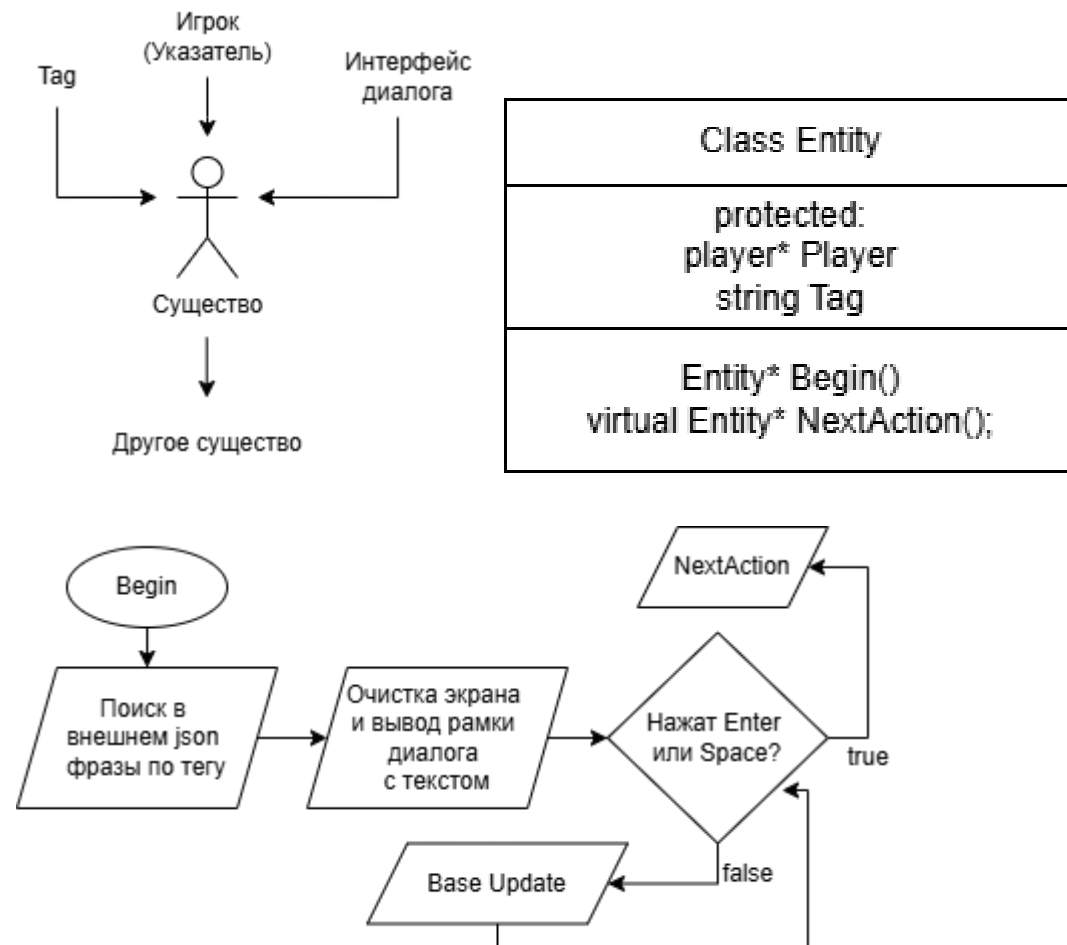


Существа

Всех существ в нашей игре объединяет 3 вещи: Tag (имя существа), Player (Указатель на класс игрока) и окно диалога, которое будет нас встречать при встрече с любым

существом (то есть перед входом на этаж, перед монстром и т.п.).

Также все существа после себя оставляют новое существо, которое может быть null.

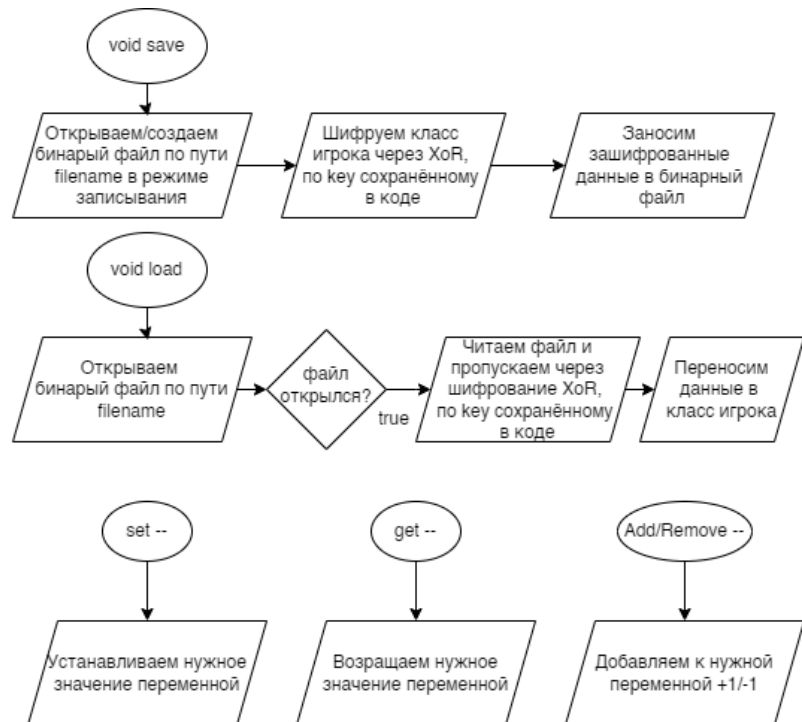
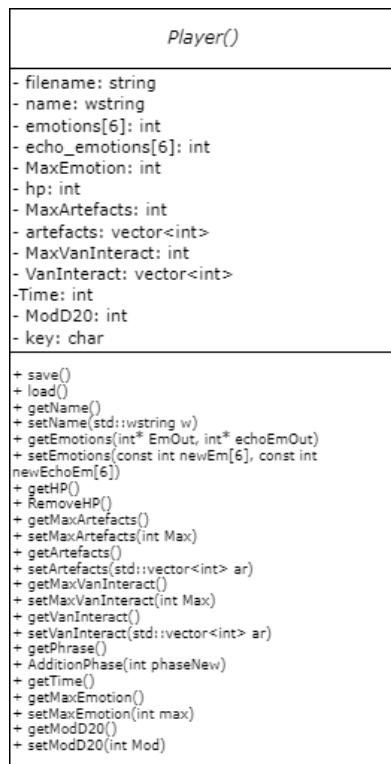


Игрок

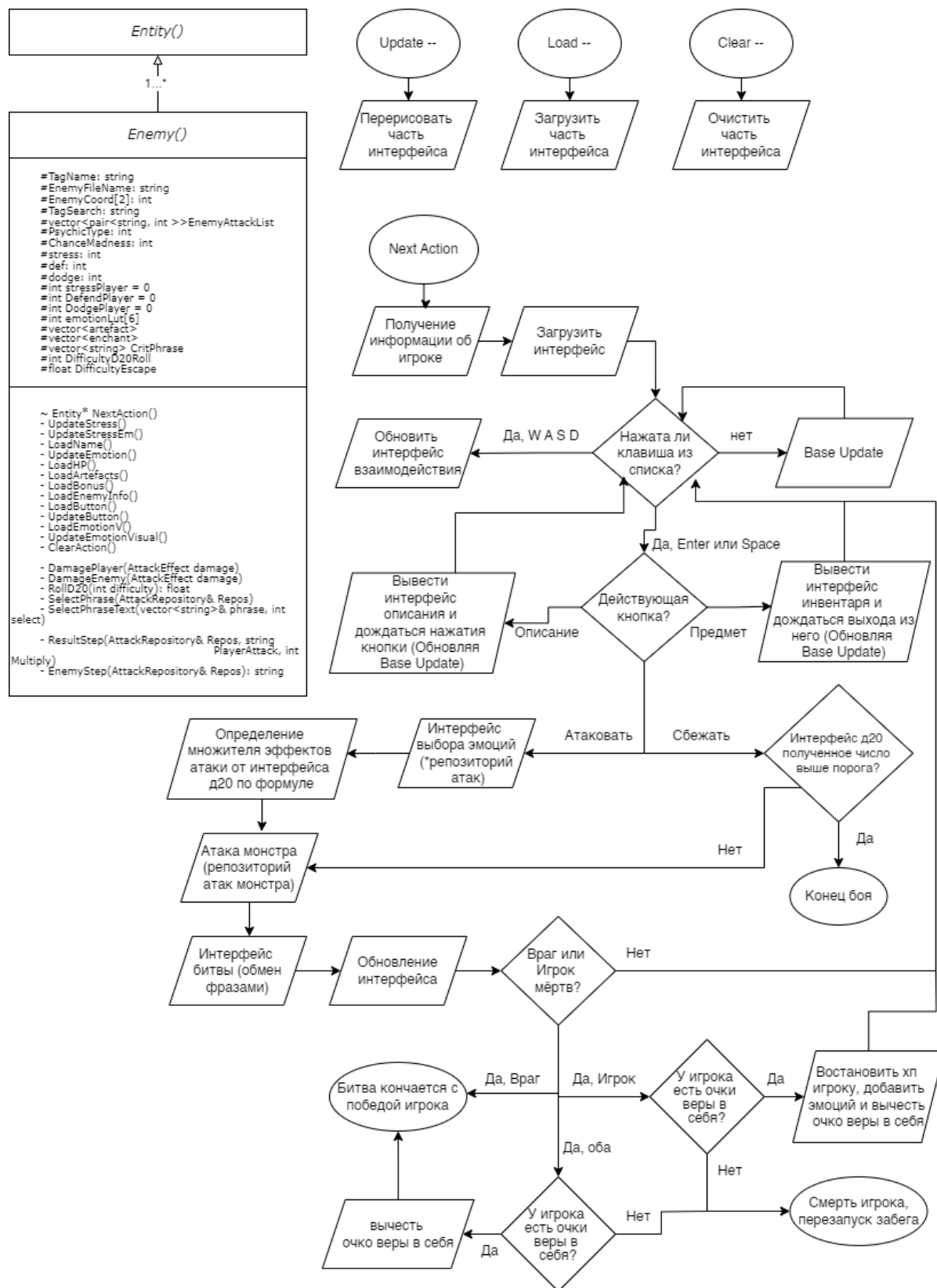
Класс игрока будет в себе иметь - Функции для работы с файлами сохранения и все данные в запрашиваемом виде.

* Файлы сохранения будут представлять из себя бинарные файлы с XoR шифровкой по заданному ключу

* Про ресурсы хранимые в игроке можно узнать из раздела [Ресурсы](#) в дизайн документе



Боевая система



В этой таблице представлены характеристики для монстров, формулы для расчета характеристик игрока(STRESS, DEF, DODGE) после нанесения атаки монстром и наоборот.

```
AttackEffect DefaultEffectFormulaTarget(struct AttackEffect Attack, std::array<int, 6>& minEmotions, std::array<int, 6>& Emotions) {  
    Attack.Stress += 3 * (Emotions[0] - minEmotions[0]); //страх  
    Attack.Stress += 4 * (Emotions[1] - minEmotions[1]); //злость  
    Attack.Stress += 2 * (Emotions[2] - minEmotions[2]); //презрение  
    Attack.Dodge -= 5 * (Emotions[2] - minEmotions[2]);  
    Attack.Stress += 2 * (Emotions[5] - minEmotions[5]); //вдохновение  
    return Attack;  
}  
AttackEffect DefaultEffectFormulaMe(struct AttackEffect Attack, std::array<int, 6>& minEmotions, std::array<int, 6>& Emotions) {  
    Attack.Stress += 1 * (Emotions[0] - minEmotions[0]); //страх  
    Attack.Dodge += 2 * (Emotions[0] - minEmotions[0]);  
    Attack.Defense -= 2 * (Emotions[1] - minEmotions[1]); //злость  
    Attack.Dodge -= 2 * (Emotions[1] - minEmotions[1]);  
    Attack.Stress -= (Emotions[3] - minEmotions[3]); //радость  
    Attack.Defense += 2 * (Emotions[3] - minEmotions[3]);  
    Attack.Stress -= (Emotions[4] - minEmotions[4]); //умиротворение  
    Attack.Dodge += 2 * (Emotions[4] - minEmotions[4]);  
    Attack.Stress -= (Emotions[5] - minEmotions[5]); //вдохновение  
    return Attack;  
}
```

Реализация броска d20:

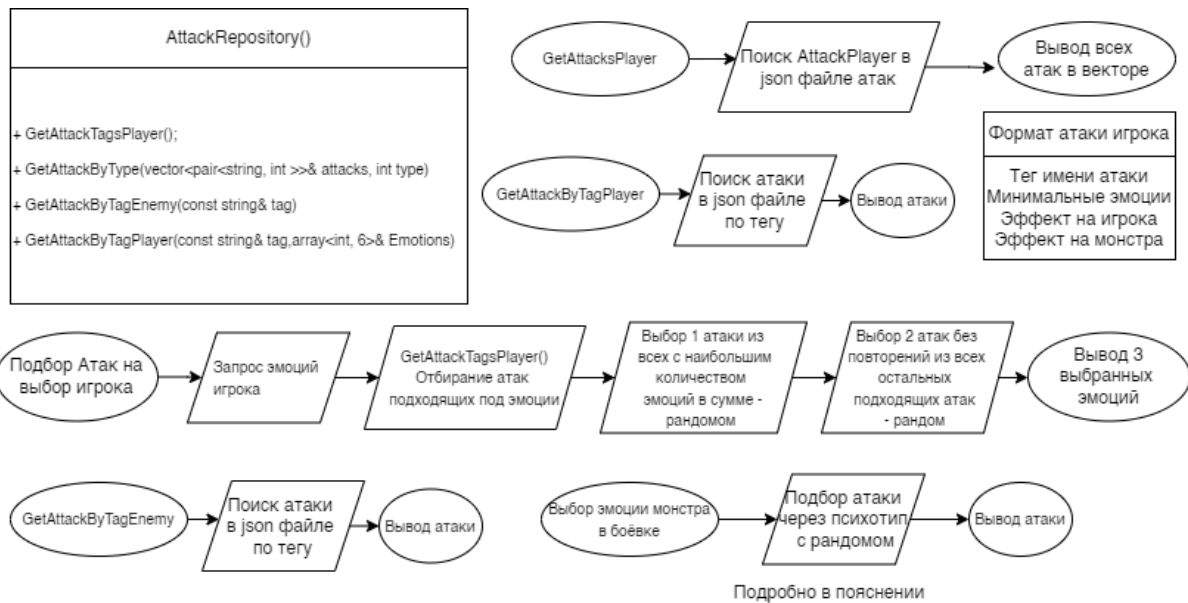
Перед игроком появляются цифры от 1 до 20 в случайном порядке(создается массив M из n чисел от 1 до 20), за это время ему нужно будет остановить дайс чтобы выпало > 15, в ином случае, когда пул чисел кончается, выпавшим числом считается M[-1]. Значение n зависит от режима данж: Обучающий - 3 повтора(т.е. n = 60), обычный - 2, бесконечный - 1.

Выпало:

1 -> -120% к эффекту
2-4 -> -30% к эффекту
5-9 -> 0% к эффекту
10-14 -> +10% к эффекту
15-19 -> +30% к эффекту
20 -> +100% к эффекту

Репозитории атак

Подробнее про выбор атак монстров можно прочитать в [психике монстров](#)



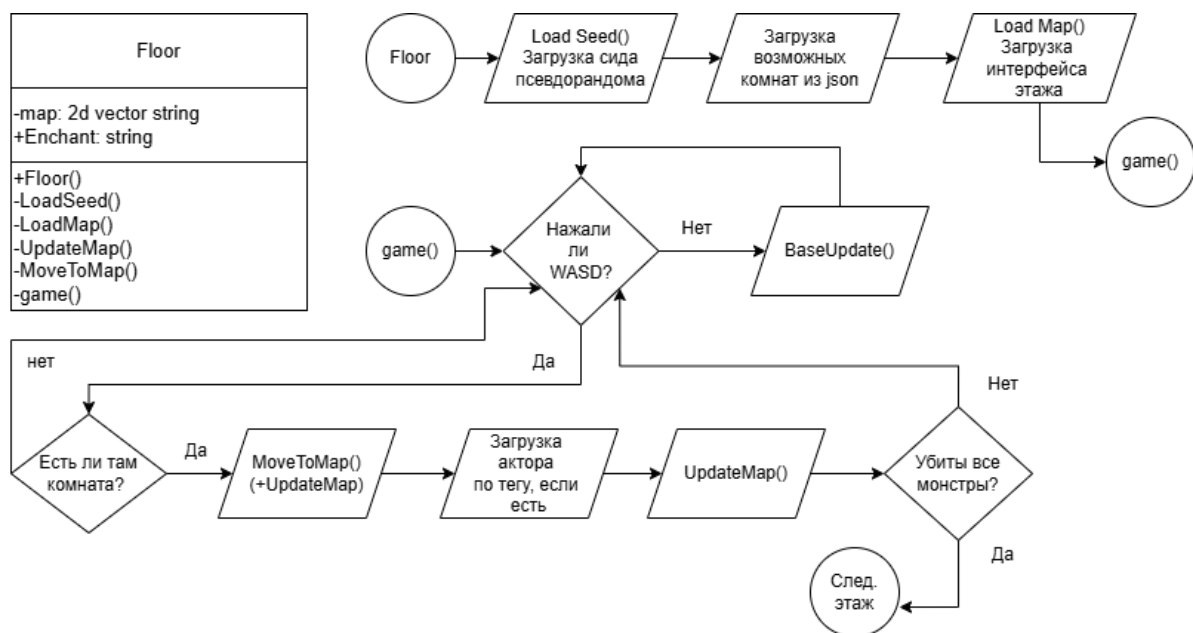
Торговец

Торговец - класс наследуемый от существа, берущий 3 уникальных товаров из json и продаёт их за эмоции. После покупки, купленный товар пропадает с прилавка.

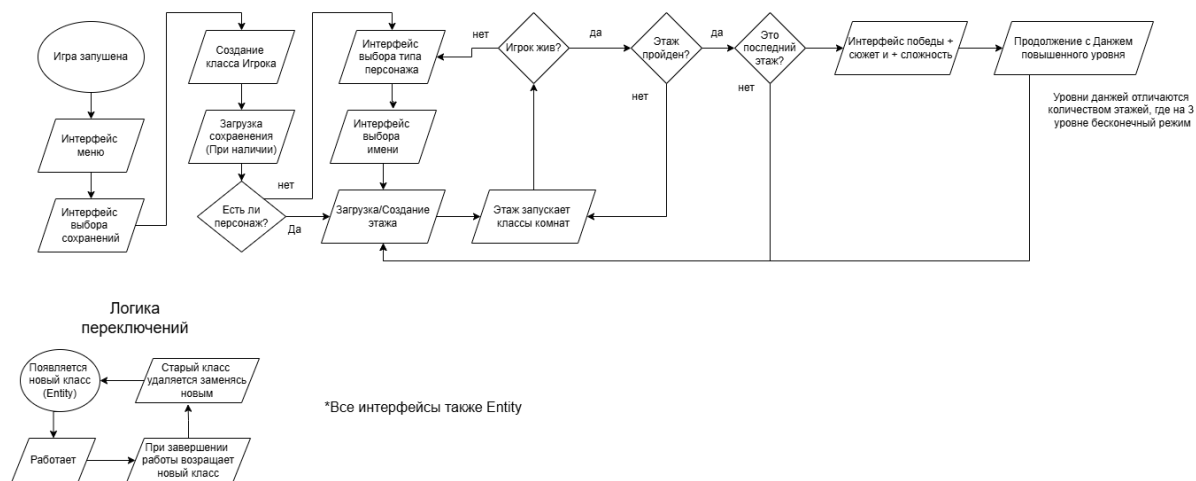
*Не вижу логики в схеме без новых функций.

Этаж

Подробнее про интерфейс этажа в [интерфейсах пользователя](#)



Логика игры



Артефакты и одноразовые предметы

Артефакты и одноразовые предметы представляют из себя структуры с методами.

Артефакты будут хранить в себе дополнительные формулы для атак врага или персонажа.

Дополнительные формулы будут применяться, после основных формул перед атакой.

Одноразовые вещи - содержат в себе методы, которые могут влиять на все ресурсы персонажа и на множитель атаки, с d20.

Примеры:

Одноразовая вещь **Вдохновение** - при использовании увеличивает количество очков веры в себя у игрока.

Артефакт **Шлем Варвара** - когда в инвентаре, урон увеличивается в 2 раза, но игроку недоступен хилл.

Внешние файлы по папкам 🧑

Каталогизация внешних файлов

- ❖ Шрифты
- ❖ Иконки
- ❖ Интерфейсы
- ❖ Сохранения
- ❖ Текст
- ❖ Баланс

Шрифты

Папка с шрифтами используемыми в игре

Формат ttf

Шрифты должны поддерживать изменение размеров

Иконки

Папка с иконками игры.

Формат ico

Размерность любая поддерживаемая форматом ico

Интерфейсы

Папка с игровыми интерфейсами

Формат txt

Символы поддерживаемые codepage 437

Сохранения

Папка с ячейками сохранений

Формат бинарные файлы

Редактирование вне кода не предусмотрено

Текст

Папка с json файлами разных языков

Формат - раздел phrases, в нём множество блоков

блок хранит в себе tag фразы на английском

и текст самой фразы

```
{
  "phrases": [
    {
      "name": "primer",
      "text": "Здесь сам текст - название primer, для тега при поиске из кода"
    }
  ]
}
```

Баланс

Файл json с атаками врагов, игрока и информацией о монстрах их характеристиках, а также об этажах.

Формат - разделы **PlayerAttack**, **EnemyAttack**, **Enemy** и **Floor**

PlayerAttack формат блока

Тег атаки - string

Минимальные эмоции (в int - Страх, злость, презрение, радость, умиротворение, вдохновения)

Эффект на игрока (в int - Стресс, защита, уклонение)

Эффект на монстра (в int - Стресс, защита, уклонение)

EnemyAttack формат блока

Тег атаки - string

Эффект на игрока (в int - Стресс, защита, уклонение)

Эффект на монстра (в int - Стресс, защита, уклонение)

Enemy формат блока

Тег монстра - string

Тип психики - int

Начальный стресс - int

Начальная защита - int

Начальное уклонение - int

Массив возможных атак - string

Массив средних количеств атак - int

Floor формат блока

Массив тегов возможных боссов - string

Массив тегов возможных монстров - string

Массив тегов возможных артефактов - string

Среднее количество монстров - int

Среднее количество хороших точек - int

Среднее количество пустых комнат - int

Распределение механик по папкам

Файл реднер

Рендер

Базовый update и fullscreen

Файл существ

Существо

Игрок

Боевая система

Торговец

Этаж

Файл предметов

Артефакты и одноразовые предметы

Файл репозитория атак

Репозитории атак

Файл main

Логика игры