

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВВГУ»)

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

## ТЕХНИЧЕСКОЕ ЗАДАНИЕ ТЕКСТОВАЯ RPG-ИГРА

По дисциплине

«Информатика и программирование»

Студент

гр. БИН-25-2

Ассистент

преподавателя

Д.В. Тимкин

М.В. Водяницкий

## Задание

Вы работаете программистом в небольшой японской компании на заре игровой индустрии. Компания разрабатывает свою первую экспериментальную игру - текстовую RPG, которая должна запускаться прямо в консоли и погружать игрока в атмосферу подземелий, опасностей и развития персонажа

Ваша задача - реализовать прототип игры, который демонстрирует основные игровые механики: характеристики персонажа, бои, прокачку, инвентарь и случайные события

### 1. Общая идея программы

Программа представляет собой консольную текстовую RPG, в которой игрок:

- 1) создает персонажа (выбор расы)
- 2) получает случайные характеристики в рамках выбранной расы
- 3) исследует подземелье, состоящее из случайных комнат
- 4) сражается с врагами, находит предметы и улучшает персонажа
- 5) повышает уровень и распределяет очки характеристик
- 6) принимает решения, влияющие на дальнейший путь

Игра работает в пошаговом режиме и управляется вводом команд с клавиатуры

### 2. Создание персонажа

#### 2.1 Выбор расы

В начале игры пользователь выбирает расу персонажа (например):

- 1) Человек
- 2) Эльф
- 3) Дворф

Каждая раса задает диапазоны генерации характеристик

#### 2.2 Характеристики персонажа

Характеристики генерируются случайным образом при создании персонажа, но в допустимых пределах для выбранной расы

Пример набора характеристик (можно расширять):

- 1) HP - здоровье
- 2) Attack - сила атаки
- 3) Defense - защита
- 4) Agility - ловкость (влияет на уклонение)

5) Height - рост

6) Weight - вес

Допускается, что некоторые характеристики влияют друг на друга (например, рост и вес влияют на уклонение или скорость)

### 3. Опыт и уровни

1) Персонаж получает опыт за победу над врагами

2) При накоплении нужного количества опыта повышается уровень

3) Каждый новый уровень дает очки прокачки

#### 3.1 Прокачка характеристик

Игрок может распределять очки вручную между характеристиками

Пример:

1) +1 к атаке

2) +2 к HP

3) +1 к ловкости

Распределение очков выполняется в комнатах отдыха

### 4. Инвентарь и экипировка

#### 4.1 Инвентарь

Инвентарь хранит предметы:

1) зелья (лечение и др.)

2) монеты

3) оружие 4) прочие предметы

Игрок может:

1) просматривать инвентарь

2) использовать предметы

3) выбрасывать любые предметы

#### 4.2 Экипировка

В инвентаре должны быть отдельные слоты:

1) оружие

2) броня

Экипированные предметы влияют на характеристики персонажа

### 5. Подземелье и комнаты

### 5.1 Структура подземелья

- 1) Игра начинается в подземелье
- 2) Подземелье состоит из комнат
- 3) После каждой комнаты игрок выбирает путь:
- 4) налево
- 5) направо

Развилка есть после каждой комнаты

### 5.2 Типы комнат

Комнаты генерируются случайно:

- 1) Боевая комната - бой с врагом
- 2) Комната отдыха - без событий
- 3) Комната с сундуком - предметы или золото

Возможны комбинации:

- 1) слева враг, справа сундук
- 2) оба врага
- 3) обе комнаты отдыха

### 5.3 Видимость комнат

Перед выбором направления игрок:

- 1) иногда знает, что находится дальше
- 2) иногда не знает (темно, неизвестно)

Информация о видимости определяется случайно.

## 6. Враги и сложность

- 1) Враги генерируются случайно
- 2) У врагов есть характеристики (НР, атака, защита и т.д.)
- 3) С каждым этажом подземелья сложность возрастает
- 4) Каждые 'N' комнат или действий происходит переход на новый этаж

## 7. Боевая система

Бой происходит в пошаговом режиме:

Пример действий игрока:

- 1) атаковать
- 2) использовать предмет
- 3) попытаться уклониться

Учитываются:

- 1) характеристики игрока
- 2) экипировка
- 3) случайные факторы (уклонение, критический удар)

## 8. Предметы и добыча

- 1) Враги и сундуки могут давать:
- 2) зелья
- 3) оружие
- 4) другие предметы
- 5) Полученные предметы добавляются в инвентарь
- 6) При нехватке места игрок решает, что выбросить

## 9. Хранение данных

Допускается (но не обязательно):

- 1) сохранение состояния игры в файл
- 2) использование формата JSON для хранения:
- 3) характеристик персонажа
- 4) инвентаря
- 5) текущего этажа

10. Пример работы программы (фрагмент) показан на рисунке 1.

```
1 Выберите
2   расу:
3   1 - Человек
4   2 - Эльф
5   3 - Дворф
6 > 2Ваш
7
8 персонаж создан!
9 HP: 85
10 ATK: 12
11 DEF: 6
12 AFI: 14Вы входите в
13 подземелье...Перед вами
14 развилка.
15 (1) Слева: ???
16 (2) Справа: Комната отдыхаКуда
17 пойти?
```

Рисунок 1 – Листинг программы примера работы игры

## 11. Ограничения и требования

- 1) Программа консольная
- 2) Управление через текстовое меню и ввод команд
- 3) Язык программирования - не ограничен <s>(в том числе можно Python)</s>
- 4) Код должен быть читаемым и логически структурированным, можно делить на разные файлы

## Содержание

Введение .....	8
1 Создание персонажа .....	8
1.1 Выбор расы .....	8
1.2 Характеристики персонажа .....	9
2 Опыт и уровни .....	9
2.1 Прокачка характеристик .....	10
3 Инвентарь и экипировка .....	10
3.1 Инвентарь .....	10
3.2 Экипировка .....	10
4 Подземелье и комнаты .....	10
4.1 Структура подземелья .....	10
4.2 Типы комнат .....	11
4.3 Видимость комнат .....	11
5 Враги и сложность .....	11
6 Боевая система .....	11
7 Предметы и добыча .....	12
8 Хранение данных .....	12
9 Ограничения и требования .....	12
10 Классы .....	12
10.1 Класс Item .....	12
10.2 Класс Weapon .....	13
10.3 Класс Armor .....	13
10.4 Класс Potion .....	13
10.5 Класс Character .....	13
10.6 Класс Enemy .....	14
11 Основные механики боя .....	15
11.1 Атака .....	15
11.2 Использование зелья .....	15
11.3 Уклонение .....	15
12 Битва .....	15
13 Механики вне боя .....	15
13.1 Прокачка уровней .....	15
13.2 Комнаты отдыха .....	16
13.3 Инвентарь и экипировка .....	16

## Введение

Цель проекта — разработка текстовой RPG-игры. Проект позволил изучить принципы объектно-ориентированного программирования, генерацию случайных событий, пошаговые бои и работу с консольным вводом/выводом. Игра управляется через консоль, игрок создаёт персонажа, исследует подземелье, сражается с врагами и собирает предметы.

## 1 Создание персонажа

### 1.1 Выбор расы

В начале игры пользователь выбирает расу персонажа:

- 1) Человек
- 2) Эльф
- 3) Дворф

Каждая раса задаёт диапазоны генерации характеристик листинг программы показан на рисунке 2.

```
Добро пожаловать в текстовую RPG!  
Выберите расу:  
1 - Человек, 2 - Эльф, 3 - Дворф  
1  
  
Ваш персонаж создан!  
  
Нитан - Уровень 1  
HP: 85/85 ATK: 11 DEF: 7 AGI: 12  
Рост: 184 см Вес: 84 кг  
Оружие: Нет  
Броня: Нет  
  
Этаж 1, комната 1  
(1) Слева: chest  
(2) Справа: ???  
Куда пойти? 1-налево, 2-направо  
_
```

Рисунок 2 – Листинг программы примера работы игры



## 1.2 Характеристики персонажа

Характеристики генерируются случайным образом при создании персонажа:

- 1) HP – здоровье
- 2) Attack – сила атаки
- 3) Defense – защита
- 4) Agility – ловкость (влияет на уклонение)
- 5) Height – рост
- 6) Weight – вес

В коде характеристики генерируются случайно с помощью `private static Random rand = new Random()`.

## 2 Опыт и уровни

- 1) Персонаж получает опыт за победу над врагами.
- 2) При накоплении нужного количества опыта повышается уровень.
- 3) Каждый новый уровень даёт 5 очков характеристик для распределения.

На рисунке 3 показана механика игры и получения уровня и опыта.

```

Этаж 1, комната 1
(1) Слева: chest
(2) Справа: ???
Куда пойти? 1-налево, 2-направо
2
Вы встретили Тролля!

Ваш HP: 85/85 | Тролля HP: 32
1-Атаковать, 2-Использовать зелье, 3-Уклонение, 4-Инвентарь
1
Вы нанесли 5 урона .
Тролля атакует и наносит 2 урона!

Ваш HP: 83/85 | Тролля HP: 27
1-Атаковать, 2-Использовать зелье, 3-Уклонение, 4-Инвентарь
1
Вы нанесли 5 урона .
Тролля атакует и наносит 2 урона!

Ваш HP: 81/85 | Тролля HP: 22
1-Атаковать, 2-Использовать зелье, 3-Уклонение, 4-Инвентарь

```

Рисунок 3 – Листинг программы примера работы игры

## 2.1 Прокачка характеристик

Игрок может распределять очки вручную между характеристиками:

- 1) HP (+5 HP)
- 2) Attack (+1 ATK)
- 3) Defense (+1 DEF)
- 4) Agility (+1 AGI)

Распределение очков выполняется в комнатах отдыха или при достижении нового уровня.

## 3 Инвентарь и экипировка

### 3.1 Инвентарь

Инвентарь хранит предметы:

- 1) Зелья (лечение)
- 2) Монеты
- 3) Оружие
- 4) Броня

Игрок может:

- 1) Просматривать инвентарь
- 2) Использовать предметы
- 3) Экипировать оружие или броню
- 4) Выбрасывать предметы (кроме монет)

### 3.2 Экипировка

В инвентаре есть отдельные слоты:

Оружие

Броня

Экипированные предметы влияют на характеристики персонажа (увеличивают атаку или защиту).

## 4 Подземелье и комнаты

### 4.1 Структура подземелья

- 1) Игра начинается в подземелье.
- 2) Подземелье состоит из комнат.
- 3) После каждой комнаты игрок выбирает путь: налево или направо.
- 4) Каждые 5 комнат повышается этаж и сложность врагов.

## 4.2 Типы комнат

Комнаты генерируются случайно:

- 1) Боевая комната – бой с врагом
- 2) Комната отдыха – восстанавливает НР и позволяет распределить очки
- 3) Комната с сундуком – находка оружия, брони, зелья или монет

## 4.3 Видимость комнат

Перед выбором направления игрок:

Иногда видит, что находится дальше

Иногда неизвестно (???)

## 5 Враги и сложность

- 1) Враги генерируются случайно с характеристиками (НР, Attack, Defense).
- 2) Сложность возрастает с каждым этажом.
- 3) Враги получают бонусы к характеристикам в зависимости от уровня игрока и этажа.

## 6 Боевая система

Бой пошаговый:

- 1) Игрок выбирает действие:

Атаковать

Использовать зелье

Попытаться уклониться

Вызвать инвентарь

- 2) Атака учитывает:

Характеристики игрока

Экипированное оружие/броню

Случайные факторы (критический удар 10%)

- 3) Враг наносит урон игроку после хода игрока, если не погиб.

## 7 Предметы и добыча

- 1) Враги и сундуки могут давать оружие, броню, зелья, монеты.
- 2) Игрок может использовать или экипировать предметы сразу.
- 3) При нехватке места игрок решает, что выбросить.

## 8 Хранение данных

- 1) Состояние игры сохраняется в JSON (savegame.json)
- 2) Хранятся характеристики персонажа, инвентарь и уровень.

## 9 Ограничения и требования

- 1) Программа консольная
- 2) Управление через текстовое меню
- 3) Язык программирования – C#
- 4) Код читаемый, структурирован по классам

## 10 Классы

### 10.1 Класс Item

Базовый абстрактный класс для всех предметов.

Содержит только название предмета, родительский класс который в дальнейшем будет использоваться показан на рисунке 4.

Ссылка: 7

```
abstract class Item
{
    public string Name;
```

Рисунок 4 – Листинг класса Item

## 10.2 Класс Weapon

Наследуется от Item, содержит бонус к атаке показано на рисунке 5.

```
class Weapon : Item
{
    public int AttackBonus;
    Ссылка: 1
    public Weapon(string name, int bonus) { Name = name; AttackBonus = bonus; }
}
```

---

Рисунок 5 – Листинг класса Weapon

## 10.3 Класс Armor

Наследуется от Item, содержит бонус к защите показано на рисунке 6.

```
class Armor : Item
{
    public int DefenseBonus;
    Ссылка: 1
    public Armor(string name, int bonus) { Name = name; DefenseBonus = bonus; }
}
```

Рисунок 6 – Листинг класса Armor

## 10.4 Класс Potion

Наследуется от Item, восстанавливает здоровье показано на рисунке 7.

```
class Potion : Item
{
    public int HealAmount;
    Ссылка: 1
    public Potion(string name, int heal) { Name = name; HealAmount = heal; }
}
```

Рисунок 7 – Листинг класса Potion

## 10.5 Класс Character

Хранит характеристики игрока, инвентарь, экипировку, уровни и очки статусов как показано на рисунке 8.

Содержит методы:

- 1) GenerateStats() — генерация характеристик по расе
- 2) GainExp() — получение опыта
- 3) LevelUp() — повышение уровня и распределение очков
- 4) ShowStats() — отображение характеристик
- 5) DistributeStatPoints() — распределение очков
- 6) ShowInventory() — просмотр и использование предметов

```

class Character
{
    public string Name;
    public Race Race;
    public int Level = 1;
    public int Exp = 0;
    public int ExpToLevel = 100;
    public int MaxHP;
    public int HP;
    public int Attack;
    public int Defense;
    public int Agility;
    public int Height;
    public int Weight;
    public int StatPoints = 0;

    public Weapon EquippedWeapon = null;
    public Armor EquippedArmor = null;
    public List<Item> Inventory = new List<Item>();

    private static Random rand = new Random();
}

```

Рисунок 8 – Листинг класса Character

## 10.6 Класс Enemy

Генерируется случайно с характеристиками в зависимости от уровня. Примеры врагов: Гоблин, Скелет, Тролль, Разбойник. На рисунке 9 показано как создаются характеристики врагов.

```

class Enemy
{
    public string Name;
    public int HP;
    public int Attack;
    public int Defense;
    private static Random rand = new Random();

    Ссылка: 1
    public Enemy(int level)
    {
        string[] names = { "Гоблин", "Скелет", "Тролль", "Разбойник" };
        Name = names[rand.Next(names.Length)];
        HP = rand.Next(20, 31) + level * 5;
        Attack = rand.Next(5, 11) + level * 2;
        Defense = rand.Next(2, 6) + level;
    }
}

```

Рисунок 9 – Листинг класса Enemy

## 11 Основные механики боя

### 11.1 Атака

- 1) Физическая атака учитывает базовую атаку персонажа + бонус оружия
- 2) Есть 10% шанс критического удара
- 3) Урон уменьшается на броню противника

### 11.2 Использование зелья

Восстанавливает HP персонажа

Зелье исчезает из инвентаря после использования

### 11.3 Уклонение

Шанс уклонения =  $\text{Agility} * 2 - \text{Attack врага}$

Генерация случайного числа для успеха или провала

## 12 Битва

- 1) Игрок выбирает действие
- 2) Атака или использование зелья
- 3) Враг наносит урон, если жив
- 4) Цикл продолжается пока не победит один из участников
- 5) Игрок получает опыт после победы

На рисунке 10 показано условие при котором игра даст выбор что сделать с врагом если у него не меньше 0 HP.

```
while (player.HP > 0 && enemy.HP > 0)
{
    Console.WriteLine($"Ваш HP: {player.HP}/{player.MaxHP} | {enemy.Name} HP: {enemy.HP}");
    Console.WriteLine("1-Атаковать, 2-Использовать зелье, 3-Уклонение, 4-Инвентарь");
    string action = Console.ReadLine();
    switch (action)
    {
        .
    }
}
```

Рисунок 10 – Фрагмент цикла битвы в методе Battle

## 13 Механики вне боя

### 13.1 Прокачка уровней

Каждые 100 опыта повышается уровень

При повышении уровня начисляются 5 очков характеристик

## 13.2 Комнаты отдыха

Восстанавливают HP

Позволяют распределять очки характеристик

## 13.3 Инвентарь и экипировка

Позволяет использовать зелья

Экипировать оружие и броню

Просматривать количество монет

Ограничений по использованию нет, кроме монет

## Заключение

Проект позволил реализовать полноценную текстовую RPG-игру с широким спектром возможностей и игровых механик. В ходе разработки была создана система создания персонажа, позволяющая игроку настраивать основные параметры и характеристики, что напрямую влияет на дальнейший игровой процесс. По мере прохождения игры игрок сталкивается с различными противниками, сложность которых постепенно возрастает, благодаря чему сохраняется баланс между развитием персонажа и уровнем испытаний. Пропорциональный рост силы врагов делает игровой процесс более динамичным и мотивирует игрока продумывать стратегию своих действий.

Особое внимание в проекте было уделено логике взаимодействия игровых элементов, обработке пользовательского ввода и корректной работе игровых состояний. Это позволило сделать игру более стабильной, понятной и удобной для пользователя. В процессе разработки пришлось продумывать структуру программы, разбивать код на логические блоки и оптимизировать его, что значительно повысило качество итогового результата. Работа с большим объёмом кода помогла развить навыки анализа, поиска и исправления ошибок, а также научила более ответственному подходу к написанию программ.

Кроме технической части, данный проект способствовал развитию творческого мышления, так как необходимо было продумывать сюжетные элементы, описание событий и атмосферу игрового мира. Это позволило лучше понять, как сочетаются программирование и геймдизайн, и насколько важно учитывать интересы конечного пользователя. Реализация текстовой RPG показала, что даже без графического



интерфейса можно создать увлекательный и интересный игровой процесс, основанный на логике, выборе и воображении игрока.

В целом проект стал ценным учебным опытом, который позволил закрепить теоретические знания на практике, получить представление о полном цикле разработки программного продукта и повысить уверенность в собственных навыках. Полученный опыт будет полезен при выполнении более сложных заданий в будущем, а также станет хорошей основой для дальнейшего изучения разработки игр и программирования в целом. Я с интересом и энтузиазмом буду ждать новых схожих проектов, которые позволят продолжить развитие и углубление полученных навыков.