**АВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Научный руководитель,  Научный сотрудник МЛ ИССА  факультета компьютерных наук, канд.техн.наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.В. Максименкова  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия», канд.техн.наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **КОМПЬЮТЕРНАЯ ИГРА: СКАЗАНИЯ О МОНСТРЕ**  **Пояснительная записка**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.04.01-01 81 01-1-ЛУ** | | |
|  |  | |
| Исполнитель:  студент группы БПИ192  \_\_\_\_\_\_\_\_\_\_\_\_ / Глущенко З. С./  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. | |
|  | | |
|  | |  |

**2020**

|  |  |  |  |
| --- | --- | --- | --- |
| УТВЕРЖДЕНО  RU.17701729.04.01-01 81 01-1-ЛУ | |  |  |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **КОМПЬЮТЕРНАЯ ИГРА: СКАЗАНИЯ О МОНСТРЕ**  **Пояснительная записка**  **RU.17701729.04.01-01 81 01-1**  **Листов 50** | | | | |
|  |  | | | |
|  | | | |
| **2020** | | | | |

|  |  |
| --- | --- |
|  |  |
|  |  |

СОДЕРЖАНИЕ

[**1.** **ВВЕДЕНИЕ** 6](#_Toc40464829)

[**1.1.** **Наименование программы** 6](#_Toc40464830)

[**1.2.** **Документы, на основании которых ведется разработка** 6](#_Toc40464831)

[**2.** **НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ** 7](#_Toc40464832)

[**2.1.** **Назначение программы** 7](#_Toc40464833)

[**2.1.1.** **Функциональное назначение** 7](#_Toc40464834)

[**2.1.2.** **Эксплуатационное назначение** 7](#_Toc40464835)

[**2.2.** **Краткая характеристика области применения** 7](#_Toc40464836)

[**3.** **ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ** 8](#_Toc40464837)

[**3.1.** **Постановка задачи на разработку программы** 8](#_Toc40464838)

[**3.2.** **Описание алгоритма и функционирования программы** 8](#_Toc40464839)

[**3.2.1.** **Описание игрового процесса** 8](#_Toc40464840)

[**3.2.2.** **Описание функционирования различных меню приложения** 8](#_Toc40464841)

[**3.2.3.** **Описание функционирования персонажа** 12](#_Toc40464842)

[**3.2.4.** **Описание функционирования противников и алгоритм их поведения** 17](#_Toc40464843)

[**3.3.** **Обоснование выбора алгоритма решения задачи** 18](#_Toc40464844)

[**3.4.** **Описание и обоснование выбора метода организации входных и выходных данных** 18](#_Toc40464845)

[**3.4.1.** **Описание метода организации входных данных** 18](#_Toc40464846)

[**3.4.2.** **Описание метода организации выходных данных** 18](#_Toc40464847)

[**3.4.3.** **Обоснование выбора метода организации входных и выходных данных** 19](#_Toc40464848)

[**3.5.** **Описание и обоснование выбора состава технических и программных средств** 19](#_Toc40464849)

[**3.5.1.** **Состав технических и программных средств** 19](#_Toc40464850)

[**3.5.2.** **Обоснование выбора технических и программных средств** 19](#_Toc40464851)

[**4.** **ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ** 21](#_Toc40464852)

[**4.1.** **Предполагаемая потребность** 21](#_Toc40464853)

[**4.2.** **Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами** 21](#_Toc40464854)

[**5.** **ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ** 22](#_Toc40464855)

[**ПРИЛОЖЕНИЕ 1** 23](#_Toc40464856)

[**ПРИЛОЖЕНИЕ 2** 24](#_Toc40464857)

[**ПРИЛОЖЕНИЕ 3** 25](#_Toc40464858)

[**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ** 50](#_Toc40464859)

**АННОТАЦИЯ**

В данном программном документе приведена пояснительная записка к программе «Компьютерная игра: сказания о монстре».

В разделе «Введение» указано наименование программы, краткое наименование программы и документы, на основании которых ведется разработка.

В разделе «Назначение и область применения» указано функциональное назначение программы, эксплуатационное назначение программы и краткая характеристика области применения программы.

В разделе «Технические характеристики» содержатся следующие подразделы: - постановка задачи на разработку программы; - описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи и возможные взаимодействия программы с другими программами; - описание и обоснование выбора метода организации входных и выходных данных; - описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Настоящий документ разработан в соответствии с требованиями:

1) ГОСТ 19.101-77 Виды программ и программных документов [1];

2) ГОСТ 19.102-77 Стадии разработки [2];

3) ГОСТ 19.103-77 Обозначения программ и программных документов [3];

4) ГОСТ 19.104-78 Основные надписи [4];

5) ГОСТ 19.105-78 Общие требования к программным документам [5];

6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];

7) ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [7].

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

Перед прочтением данного документа рекомендуется ознакомиться с терминологией, приведенной в Приложении 1 настоящей пояснительной записки.

1. **ВВЕДЕНИЕ**
   1. **Наименование программы**

**Наименование программы:** «Компьютерная игра: сказания о монстре».

**Краткое наименование программы:** «Сказания о монстре».

**Условное обозначение темы разработки:** «Computer Game: Tales About The Monster».

* 1. **Документы, на основании которых ведется разработка**

Программа выполнена в рамках темы курсовой работы «Компьютерная игра: сказания о монстре» в соответствии с учебным планом подготовки бакалавров (НИУ ВШЭ, факультет компьютерных наук) по направлению «Программная инженерия».

Разработка ведется на основании приказа Национального исследовательского университета "Высшая школа экономики" № 2.3-02/1112-04 от 11.12.2019

1. **НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ**
   1. **Назначение программы**
      1. **Функциональное назначение**

Программа является игрой и несёт в себе развлекательную и досуговую функции. Игра является Action/RPG, поэтому её задача – это развитие внимания, скорости реакции и сообразительности.

* + 1. **Эксплуатационное назначение**

Программа может эксплуатироваться при наличии ПК на базе OC Windows. Конечные потребители – пользователи, без специальной подготовки, любители компьютерных игр.

* 1. **Краткая характеристика области применения**

«Компьютерная игра: сказания о монстре» предназначена для использования в области развлечений. Для отдыха, развития реакции и умения принимать быстрые решения.

1. **ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ**
   1. **Постановка задачи на разработку программы**

Разрабатываемая программа должна:

1. Возможность регулирования громкости воспроизводимой музыки
2. Управление героем
3. Отображение текущего состояния героя в HUD (Head-Up Display)
4. Обработка состояний противника и их действий
5. Возможность приостановить игру на паузу, сделать сохранение игры, загрузить сохранение, открыть меню настроек, выйти в главное меню
6. Переход в другую локацию при выполнении условий
   1. **Описание алгоритма и функционирования программы**
      1. **Описание игрового процесса**

Смысл игры заключается в зачистке локаций от монстров. Для продвижения в следующие локации пользователю необходимо зачистить текущую. При выполнении необходимых условий, в специальном месте появится подсказка для смены локации. Дойдя до последней локации и убив босса, игра считается пройденной. В процессе игры с повышением уровня будут открываться способности персонажа, которые разнообразят прохождение.

* + 1. **Описание функционирования различных меню приложения**

В начале игры перед пользователем появится начальное меню (рис.1). В нем можно:

1. Начать новую игру, с помощью кнопки Start
2. Загрузить существующее сохранение, нажав кнопку Load и открыв меню сохранений (рис.2)
3. Изменить громкость музыки, нажав кнопку settings открыв меню настроек (рис.3)
4. Выйти из игры, нажав Exit.

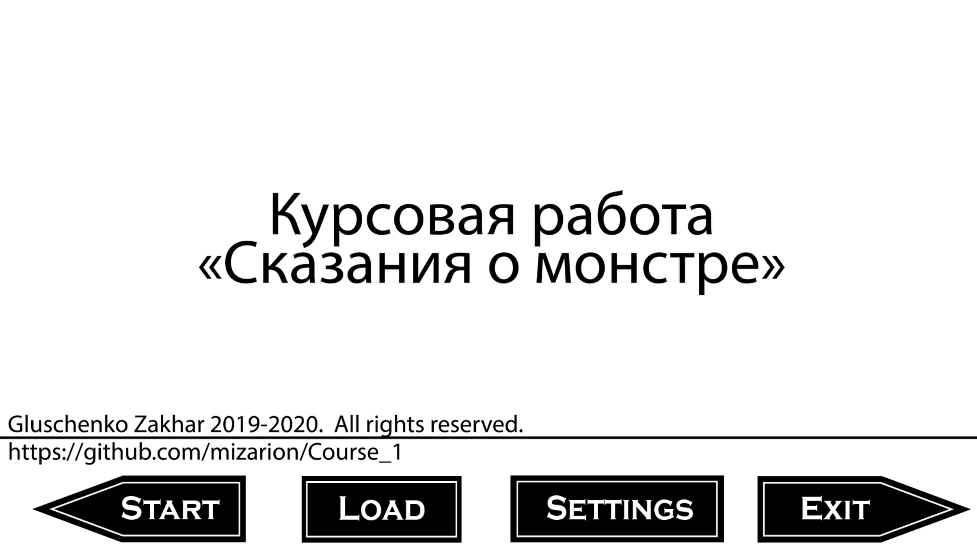


Рисунок 1. Начальное меню

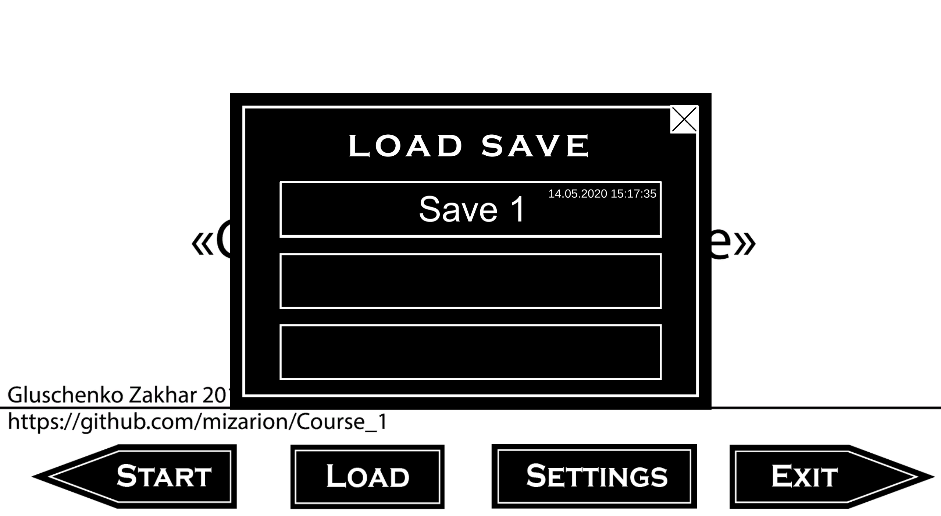


Рисунок 2. Окно загрузки сохранений, вызванное из главного меню

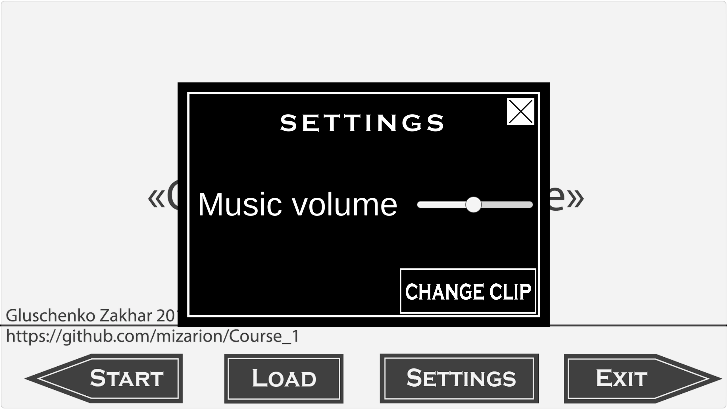


Рисунок 3. Меню настроек

Меню загрузки сохранений и настроек можно закрыть, нажав на белый крестик в правом верхнем углу.

В процессе игры пользователь может приостановить игру нажав клавишу Escape на клавиатуре. После этого откроется меню паузы (рис.4), в котором можно:

1. Resume – возобновить игру, альтернатива повторному нажатию клавиши Escape на клавиатуре
2. Save Game – Открывает меню сохранений с возможностью сохранить или перезаписать игру в выбранный слот (рис.5)
3. Load Game – Открывает меню сохранений с возможностью загрузить существующее сохранение (рис.6)
4. Settings – открывает меню настроек, как из главного меню (рис.7)
5. Exit – Кнопка выхода в главное меню.



Рисунок 4. Меню паузы



Рисунок 5. Меню сохранений, возможностью сохранить текущий прогресс



Рисунок 6. Меню сохранений, возможностью загрузить сохраненный прогресс



Рисунок 7. Меню настроек

Если в процессе игры у героя заканчивается здоровье, то он погибает и появляется меню смерти (рис.8). В котором пользователю предлагается выбор:

1. Main menu – выход в главное меню
2. Load Game – загрузить существующее сохранение



Рисунок 8. Меню смерти

Для прохождения игры пользователю нужно зачистить все локации и победить финального босса (рис.9). В случае победы появляется поздравительное окно предлагающее перейти в главное меню (рис.10). Если пользователь проиграет битву, то появится меню смерти.



Рисунок 9. Битва с финальным боссом.



Рисунок 10. Победа над финальным боссом. Конец игры.

* + 1. **Описание функционирования персонажа**

Герой состоит из следующих компонентов:

Animator отвечает за анимацию персонажа при перемещении по сцене и атаке противников (рис.11)

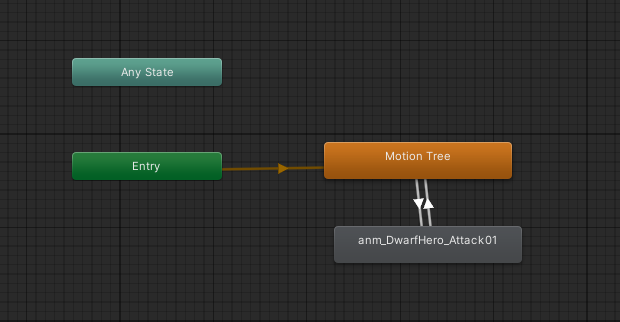


Рисунок 11. Аниматор героя

Nav Mesh Agent отвечает за построение оптимальных путей до цели и перемещение героя по этому пути. Для начала нужно прикрепить Nav Mesh Agent к движимому объекту (рис.12). Перемещение осуществляется по заранее настроенной сцене (по синей области) (рис.13).

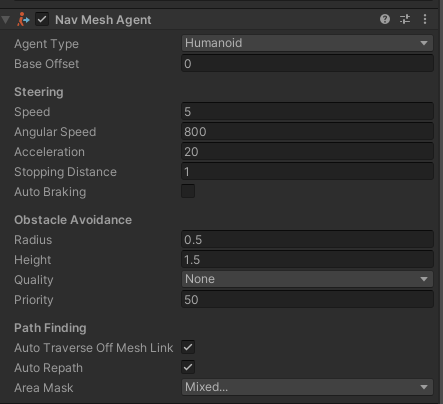


Рисунок 12. Nav Mesh Agent героя

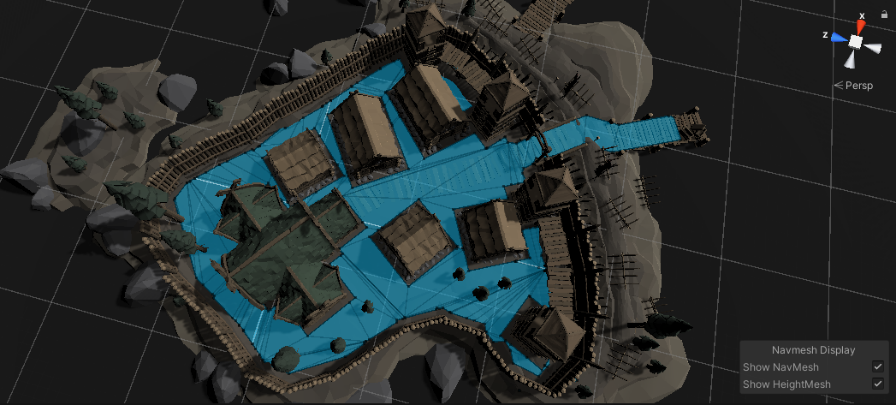


Рисунок 13. Пример локации, настроенной для работы Nav Mesh Agent

Capsule Collider отвечает за коллизии. Необходим для срабатывания триггеров при смене локации. (рис.14)

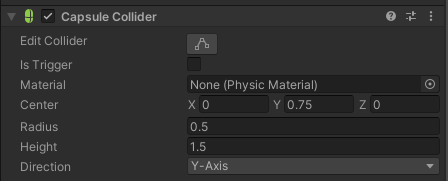


Рисунок 14. Компонент Capsule Collider

Скрипт Player отвечает за основные возможности героя (рис.15). Требует наличия Nav Mesh Agent.

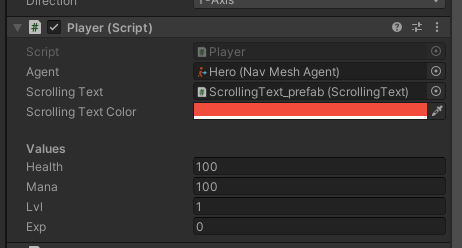


Рисунок 15. Скрипт Player

Скрипт SkillManager отвечает за работу способностей героя. В Editor’е можно редактировать значения в ограниченном диапазоне (рис.16)

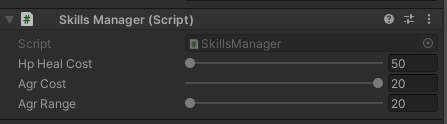


Рисунок 16. Скрипт SkillManager

Если герой получает урон, то его шкала здоровья в HUD уменьшается и рядом героем всплывает красное число, означающее количество полученного урона (рис.17).

Если герой атакует противника, то у противника изменяется полоска жизни над ним, и также всплывает желтое число, обозначающее количество полученного урона (рис.18,19).

Если герой придет в определенное место и выполнит необходимые условия, то он может перенестись в другую локацию (рис.20,21).



Рисунок 17. Получение урона героем.

Рисунок 19. Сравнения здоровых и побитых противников.

Рисунок 18. Получение противником урона.

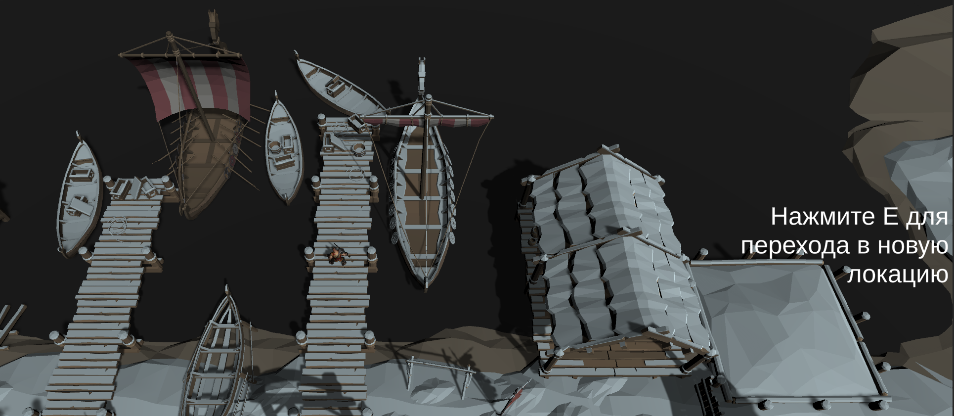


Рисунок 20. Переход в новую локацию. Отплытие на остров

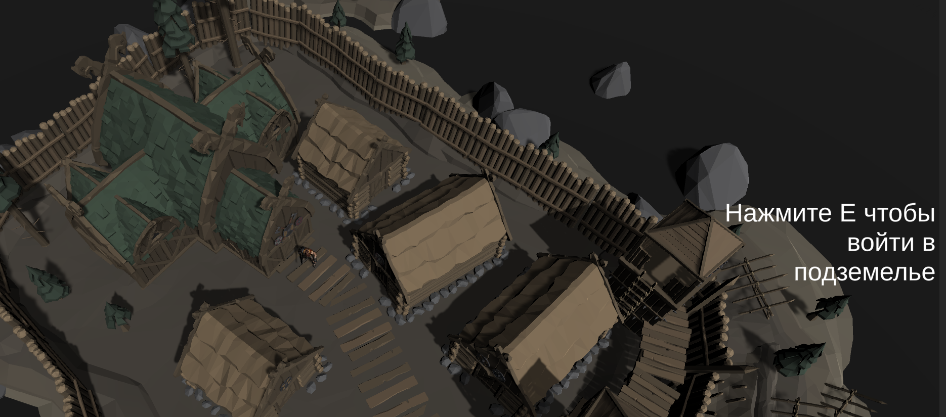


Рисунок 21. Переход в новую локацию. Спуск в подземелье

* + 1. **Описание функционирования противников и алгоритм их поведения**

Противники состоят из следующих компонентов:

Animator анимирует действия противника (рис.22).

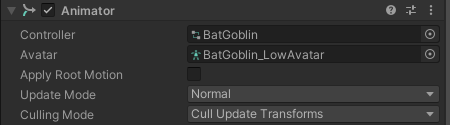


Рисунок 22. Animator противника

Nav Mesh Agent в данном случае служит для построение оптимального маршрута до героя и перемещения по нему (рис.23).

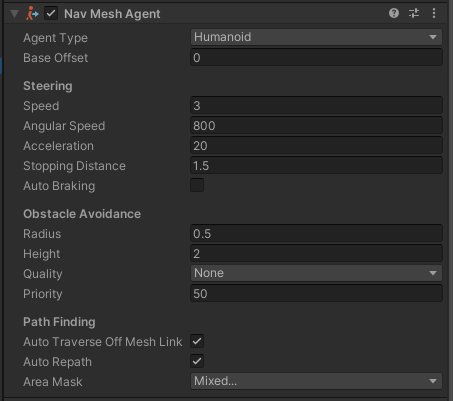


Рисунок 23. Nav Mesh Agent противника

Capsule Collider отвечает за коллизии.

Скрипт Goblin отвечает за основные возможности противника (рис.24). Требует наличия Nav Mesh Agent.

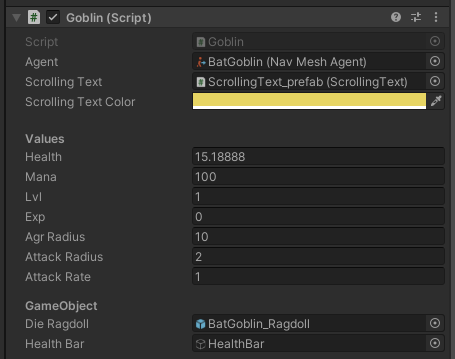


Рисунок 24. Скрипт Goblin

Если герой зашел в радиус провокации противника, то последний начинает преследовать его. Герой быстрее противника поэтому может уйти от него. Если герой покидает радиус провокации, то противник добегает до последнего нахождения героя в радиусе провокации.

Если противник догнал героя и последний попал в радиус атаки, тогда противник начинает атаковать героя, с постоянной частотой.

* 1. **Обоснование выбора алгоритма решения задачи**

Данный алгоритм решения был выбран, так как я считаю его наиболее эффективным и удобным с точки зрения реализации и дальнейшего развития.

* 1. **Описание и обоснование выбора метода организации входных и выходных данных**
     1. **Описание метода организации входных данных**

Пользовательский ввод обрабатывается в классе CanvasManager, в случае взаимодействия с UI, иначе в классе InputManager. Также если нужно программа может загрузить заранее сохраненные данные в json файлы из папки Saves/Save\*, где \* - номер сохранения.

* + 1. **Описание метода организации выходных данных**

Выходные данные, представленные в виде json файлов, расположенных в папке Saves/Save\*, где \* - номер сохранения. Если файлы уже существуют, то производится перезапись.

* + 1. **Обоснование выбора метода организации входных и выходных данных**

Методы организации входных и выходных данных, описанные в п. 3.4.1. и 3.4.2., выбраны так, как являются самыми оптимальными для работы с пользовательскими данными.

* 1. **Описание и обоснование выбора состава технических и программных средств**
     1. **Состав технических и программных средств**

Для работы программы необходим следующий состав программных средств:

1. Операционная система Microsoft Windows 10 (версии 1809 или позднее);
2. Установленный Microsoft .NET Framework 4.7.2;

Для работы программы необходим следующий состав технических средств:

1. Не менее 2 ГБ свободного места на жёстком диске;
2. Монитор с соотношением сторон 16:9 и разрешением 1920x1080 или выше;
3. Мышь и Клавиатура;
4. Оперативная память 8 ГБ или больше.
5. Видеокарта AMR Radeon RX 5700 XT
6. Процессор AMD Ryzen 7 3700X
   * 1. **Обоснование выбора технических и программных средств**

Данные технические средства пользователя были выбраны для оптимальной работы программы. 2 ГБ свободного места на жестком диске были выбраны для хранения файлов приложения и файлов, которые оно может создать. Монитор с соотношением сторон 16:9 и разрешением 1920x1080 или выше был выбран для корректного отображения UI. Клавиатура и мышь необходимы для ввода входных данных и работы с программой. Оперативная память объемом 8 ГБ или более требуется для корректной работы операционной системы Windows 10 и приложения. Звуковая плата, наушники или динамики рекомендуется использовать для прослушивания музыки. Операционная система Windows 10 была выбрана, в связи с распространённостью и простотой работы.

1. **ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**

В рамках данной работы расчет экономической эффективности не предусмотрен.

* 1. **Предполагаемая потребность**

«Компьютерная игра: сказания о монстре», как и другие игры, предназначена для развлечения пользователя.

* 1. **Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами**

Главным преимуществом приложения является низкие затраты на разработку.

1. **ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ**
2. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
11. ГОСТ 15150-69 Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды. – М.: Изд-во стандартов, 1997.
12. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
13. ГОСТ 19.602-78 Правила дублирования, учета и хранения программных документов, выполненных печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
14. Unity User Manual. URL: https://docs.unity3d.com/Manual/index.html (Дата обращения: 2020-04-16).
15. Unity Learn. URL: https://learn.unity.com/tutorials (Дата обращения: 2020-04-16).
16. Руководство по программированию на C#. URL: https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/. (Дата обращения: 2020-04-16).

# **ПРИЛОЖЕНИЕ 1**

**ТЕРМИНОЛОГИЯ**

|  |
| --- |
| Таблица 1 |

|  |  |
| --- | --- |
| **Термин** | **Определение** |
| **HUD** | Head-Up Display. Часть визуального интерфейса игрока, отображающаяся на фоне виртуального игрового пространства. Содержит индикаторы здоровья, маны, полученного опыта, уровня и тд. |
| **NPC** | Non-Player Character. Персонаж, который находится не под контролем пользователя. |
| **UI** | User interface. Пользовательский интерфейс, обеспечивающий передачу информации между пользователем-человеком и программно-аппаратными компонентами компьютерной системы. |

# **ПРИЛОЖЕНИЕ 2**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ, ПЕРЕЧИСЛЕНИЙ И ИНТЕРФЕЙСОВ**

Описание и функциональное назначение классов, перечислений и интерфейсов.

Таблица 2

|  |  |
| --- | --- |
| Класс | Назначение |
| AudioManager | Класс, реализующий воспроизведение музыки. |
| CanvasManager | Класс, реализующий работу UI. |
| EnemyManager | Класс, реализующий работу с врагами. |
| GameManager | Класс, реализующий главную логику игры. |
| InputManager | Класс, реализующий обработку пользовательского ввода. |
| SkillsManager | Класс, реализующий способности героя. |
| DataManager.Stats.Player | Статический класс, хранящий характеристики героя. |
| DataManager.Stats.Goblin | Статический класс, хранящий характеристики гоблина (противника). |
| AbstractCharacter | Абстрактный класс, отвечающий за базовый функционал существа. |
| AbstractEnemy | Абстрактный класс, дополняющий AbstractCharacter, реализуя базовый функционал врага. |
| RagdollScript | Класс, анимирующий смерть противника. |
| SaveSlotButton | Класс, реализующий работу кнопки сохранений |
| ScrollingText | Класс, реализующий визуализацию информации об уроне. |
| Singleton<T> | Класс, реализующий паттерн Singleton. |
| SmoothFollowTarget | Класс, реализующий плавное движение камеры за игроком |
| Player | Класс, отвечающий за действия игрока. |
| PlayerData | Класс, хранящий данные необходимые для сохранения. |
| Goblin | Класс, реализующий противника гоблина. |
| GoblinBoss | Класс, реализующий противника короля гоблинов. |
| GoblinData | Класс, содержащий данные о гоблинах для сохранения. |
| Cutscene\_dock | Класс, отвечающий за переход между локациями. Переход с причала на остров. |
| Cutscene\_dungeon | Класс, отвечающий за переход между локациями. Переход с острова в подземелье |
| ICharacter | Интерфейс, отвечающий за основную механику персонажей. |

# **ПРИЛОЖЕНИЕ 3**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ КЛАССОВ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса AudioManager  Таблица 3.1 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| Clips | private | AudioClip[] | Массив, содержащий музыку, для воспроизведения | |
| CurrentClip | private | MyAudioClips | Проигрываемый клип. | |
| Свойства | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| AudioSource | public | AudioSource | Ссылка, на компонент AudioSource, используемый для работы с аудио. | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| FixedUpdate | private | void | - | Встроенный метод, вызываемый каждый Time.fixedDeltaTime. Используется для смены песни, после ее завершения. |
| PlayNextClip | public | void | - | Воспроизводит следующий клип. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса CanvasManager  Таблица 3.2 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| \_startSceneUI | private | GameObject | Контейнер для объектов стартового меню | |
| \_settingsPanel | private | Image | Ссылка на меню настроек | |
| \_pauseImage | private | Image | Ссылка на меню паузы | |
| \_musicVolume | private | Slider | Ссылка на ползунок громкости в меню настроек | |
| \_loadMenu | private | Image | Ссылка на меню загрузки сохранений | |
| \_saveMenu | private | Image | Ссылка на меню создания сохранений | |
| loadOne | public | SaveSlotButton | Ссылка на класс, реализующий работу с кнопками сохранения | |
| loadTwo | public | SaveSlotButton | Ссылка на класс, реализующий работу с кнопками сохранения | |
| loadThree | public | SaveSlotButton | Ссылка на класс, реализующий работу с кнопками сохранения | |
| saveOne | public | SaveSlotButton | Ссылка на класс, реализующий работу с кнопками сохранения | |
| saveTwo | public | SaveSlotButton | Ссылка на класс, реализующий работу с кнопками сохранения | |
| saveThree | public | SaveSlotButton | Ссылка на класс, реализующий работу с кнопками сохранения | |
| HUD | private | GameObject | Контейнер для объектов пользовательского интерфейса | |
| \_healthImage | private | Image | Полоска здоровья героя | |
| \_manaImage | private | Image | Полоска маны героя | |
| \_expImage | private | Image | Полоска опыта героя | |
| \_lvlText | private | TMPro.TextMeshProUGUI | Ссылка на текст, отвечающий за показ уровня героя | |
| \_messageText | private | TMPro.TextMeshProUGUI | Ссылка на текст, для инструкций при смене локаций | |
| Skill1 | public | Button | Ссылка на способности героя | |
| Skill2 | public | Button | Ссылка на способности героя | |
| \_storyImage | private | Image | Ссылка на изображение с предысторией | |
| \_victoryPanel | private | Image | Ссылка на изображение, появляющиеся после победы над боссом | |
| \_DeathPanel | private | Image | Ссылка на меню смерти | |
| needLoad | public | bool | отвечающее за информирование о необходимости загрузки сохранения | |
| path | public | string | Путь до папки с сохранением | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| UpdateHUD | public | void | - | Обновляет значения здоровья, маны, опыта, уровня |
| UpdateMessage | public | void | string message | Включает / выключает элементы интерфейса. |
| ActivateUI | public | void | bool showHUD = false, bool showStartMenu = false, bool showPauseMenu = false, bool showDeathMenu = false | Включает / выключает элементы интерфейса. |
| StartHandler | public | void | - | Обработчик загрузки сцены |
| PauseHandler | public | void | - | Ставит игру на паузу или возобновляет и показывает/скрывает меню паузы. |
| MakeSave | public | void | SaveSlotButton button | Начинает игру с загруженным сохранением |
| LoadSave | public | void | string path | Загружает сохранение по определенному пути. |
| ActivateLoadMenu | public | void | - | Открывает / Закрывает меню загрузки |
| ActivateSaveMenu | public | void | - | Открывает / Закрывает меню сохранений |
| ActivateSettings | public | void | - | Открывает / Закрывает меню настроек |
| SettingsHandler | public | void | int action = 0 | Обрабатывает действия в меню настроек |
| ShowStory | public | void | - | Показывает / Скрывает меню с предысторией. |
| RestartHandler | public | void | - | Обработчик перезапуска игры / выхода в главное меню |
| DeathHandler | public | void | - | Обработчик меню смерти |
| Victory | public | void | - | Обработчик победы над главным злодеем |
| QuitHandler | public | void | - | Обработчик завершения игры |
| SpellOne | public | void | - | Обработчик способности 1 |
| SpellTwo | public | void | - | Обработчик способности 2 |
| UpdateSaves | public | void | - | Обновляет доступные сохранения |
| UpdateSkills | public | void | - | Обновляет отображение доступных навыков |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса EnemyManager  Таблица 3.3 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| enemySample | private | GameObject | Образец создаваемых врагов | |
| \_spawnPositions | private | Transform[] | Позиции для создания противников | |
| poolStartLength | private | int | Минимальное количество противников в массиве. Для Object Pooling | |
| \_spawnRadius | private | float | Минимальное расстояние между героем и созданным противником. | |
| \_abstractEnemiesPool | private | List<AbstractEnemy> | Список противников приведенных к AbstractEnemy | |
| ragdollPool | public | List<GameObject> | Список ragdoll’ов, которые нужно удалить при загрузке сохранения. | |
| \_pool | private | List<GameObject> | Список GameObject противников. | |
| Свойства | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| GetEnemy | public | GameObject | Свойство, находящее не используемого противника из пула, активирует и возвращает его. | |
| GetPool | public | List<GameObject> | Свойство, возвращающее ссылку на пул GameObject’ов. | |
| GetActiveAbstractEnemyPool | public | List<AbstractEnemy> | Свойство, находящее используемых противников из пула, добавляет их в список и возвращает его. | |
| GetAbstractEnemyPool | public |  | Свойство, возвращающее список противников приведенных к AbstractEnemy. | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Awake | protected | void | - | Встроенный метод. Используется для инициализации значений. |
| DestroyAllRagdolls | public | void | - | Удаляет все существующие ragdoll'ы со сцены |
| StartNewWawe | private | IEnumerator | int count, int enemyLevel = 4 | Корутина для создания новой "волны" атаки противников на героя |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса GameManager  Таблица 3.4 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| CurrentState | public | GameState | Текущее состояние игры | |
| CurrentScene | public | Scenes | Текущая сцена | |
| isSaveOneAvailable | public | bool | Доступность 1 сохранения | |
| isSaveTwoAvailable | public | bool | Доступность 2 сохранения | |
| isSaveThreeAvailable | public | bool | Доступность 3 сохранения | |
| \_deathCounter | private | int | Счетчик убийств | |
| canTeleport | public | bool | Возможность перемещения на остров | |
| canEnterInDungeon | public | bool | Возможность перемещения в подземелье | |
| avaibleFirstSkill | public | bool | Доступность 1 способности | |
| avaibleSecondSkill | public | bool | Доступность 2 способности | |
| Свойства | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| DeathCounter | public | int | Свойство, обрабатывающее количество смертей | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Start | private | void | - | Встроенный метод. Используется для задавания уничтожаемых объектов, инициализации полей и проверки существующих сохранений. |
| UpdateGameState | public | void | GameState state | Изменяет и обрабатывает изменение состояния игры |
| LimitFPS | public | void | int count | Обрабатывает ограничение fps |
| ResetGameManager | private | void | - | Сбрасывает данные GameManager |
| TogglePause | public | void | - | Возобновляет/приостанавливает игру |
| StartGame | public | void | - | Начинает игру, и, если нужна загрузка, загружает нужную сцену и данные к ней, иначе загружает стартовую сцену. |
| ChangeScene | public | void | Scenes newScene, bool changeLocation = false | Обрабатывает переход между игровыми сценами, с загрузкой необходимых данных |
| RestartGame | public | void | - | Перезапускает игру, выгружая текущую сцену и меняя режим интерфейса на стартовое меню. |
| CheckSaves | public | void | - | Проверяет существование сохранений и сообщает об их наличии |
| CheckSave | private | void | string path, SaveSlotButton save, SaveSlotButton load | Проверяет наличие сохранения, по переданному пути. |
| SaveGame | public | void | SaveSlotButton button | Сохраняет игру |
| LoadGame | public | void | string path | Загружает игру |
| CheckCurrentScene | public | void | string path | Проверяет текущую сцену и загружаемую, и решает, что дальше делать. |
| QuitGame | public | void | - | Завершает работу игры |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса InputManager  Таблица 3.5 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| clickableLayer | private | LayerMask | Используется для определения кликабельных слоев | |
| pointer | private | Texture2D | Стандартный курсор | |
| target | private | Texture2D | Курсор в виде прицела | |
| sword | private | Texture2D | Курсор в виде меча | |
| \_scrollSens | private | float | Чувствительность колесика мышки. | |
| OnClickEnviroment | public | EventVector3 | Событие для перемещения игрока с помощью клика мышки. | |
| onClickAttackable | public | EventGameObject | Событие для атаки врага. | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Update | private | void | - | Встроенный метод. Используется для обработки пользовательского ввода. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса SkillsManager  Таблица 3.6 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| hpHealCost | private | float | Стоимость восполнения жизни второй способностью | |
| agrCost | private | float | Стоимость использования первой способности | |
| agrRange | private | float | Радиус провокации противников | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| FirstSkill | public | void | - | Первая способность – провокация противников |
| SecondSkill | public | void | - | Вторая способность - восстановление хм за счет маны |

|  |  |  |  |
| --- | --- | --- | --- |
| Описание полей методов и свойств класса DataManager.Stats.Player  Таблица 3.7 | | | |
| Поля | | | |
| Имя | Модификатор  доступа | Тип | Назначение |
| Health | public readonly | List<int> | Содержит значения здоровья для каждого уровня героя |
| Manapool | public readonly | List<int> | Содержит значения маны для каждого уровня героя |
| Experience | public readonly | List<int> | Содержит значения необходимого опыта для каждого уровня героя |
| Damage | public readonly | List<int> | Содержит значения урона для каждого уровня героя |

|  |  |  |  |
| --- | --- | --- | --- |
| Описание полей методов и свойств класса DataManager.Stats.Goblin  Таблица 3.8 | | | |
| Поля | | | |
| Имя | Модификатор  доступа | Тип | Назначение |
| Health | public readonly | List<int> | Содержит значения здоровья для каждого уровня гоблина |
| Manapool | public readonly | List<int> | Содержит значения маны для каждого уровня гоблина |
| Experience | public readonly | List<int> | Содержит значения необходимого опыта для каждого уровня гоблина |
| Damage | public readonly | List<int> | Содержит значения урона для каждого уровня гоблина |

Таблица 3.9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса AbstractCharacter | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| agent | protected | NavMeshAgent | Перемещение персонажа на сцене | |
| animator | protected | Animator | Анимирует действия персонажа | |
| \_scrollingText | private | ScrollingText | Ссылка на клонируемый объект | |
| \_scrollingTextColor | private | Color | Цвет вслывающего текста | |
| \_scrollingTextContainer | private | Transform | Контейнер, куда будут помещаться созданные объекты. | |
| \_health | private | float | Хранит текущее количество здоровья персонажа | |
| \_\_mana | private | float | Хранит текущее количество маны персонажа | |
| \_\_lvl | private | int | Хранит текущий уровень | |
| \_\_exp | private | float | Хранит текущее количество опыта | |
| \_listHp | protected | List<int> | Список возможных значений здоровья | |
| \_listMp | protected | List<int> | Список возможных значений маны | |
| \_listExp | protected | List<int> | Список возможных значений опыта | |
| \_listDmg | protected | List<int> | Список возможных значений урона | |
| Свойства | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| Name | public | string | Имя персонажа | |
| Health | public | float | Свойство, отвечающее за здоровье | |
| Manapool | public | float | Свойство, отвечающее за ману. | |
| Level | public | int | Свойство, отвечающее за уровень. | |
| Experience | public | float | Свойство, отвечающее за опыт. | |
| Damage | public | float | Свойство, отвечающее за урон | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| InitializeProperties | protected | void | List<int> health, List<int> mana, List<int> experience, List<int> damages, string name, int lvl = 1 | Метод для создания абстрактного персонажа. Производит инициализацию необходимых компонентов. |
| Attack | public | void | ICharacter character | Атакует выбранную цель, нанося урон в итервале [Damage/2, Damage] урона. |
| GetDamage | public | void | float damage | Обрабатывает получение урона в размере damage. |
| Recovery | public | IEnumerator | float healRecovery = 5, float manaRecovery = 5, float delay = 1 | Восполняет здоровье и ману персонажа через определенный интервал. |
| Die | public | void | - | Обрабатывает смерть персонажа |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса AbstractEnemy  Таблица 3.10 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| player | protected | Player | Ссылка на главного героя | |
| \_expMultiplier | protected | float | Множитель получаемого опыта за убийство этого противника | |
| \_agrRadius | protected | float | Радиус, с которого враг замечает героя | |
| \_attackRadius | protected | float | Радиус, с которого враг начинает атаковать героя | |
| \_attackRate | protected | float | Время необходимое для совершения одной атаки | |
| DieRagdoll | protected | GameObject | Ragdoll’ для анимации смерти. | |
| healthBar | protected | GameObject | Полоска здоровья | |
| timer | private | float | Отсчитывает время прошедшее с предыдущей атаки | |
| Свойства | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| Health | public | float | Свойство, отвечающее за здоровье персонажа | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| InitializeProperties | protected | void | List<int> health, List<int> mana, List<int> experience, List<int> damages, string name, int lvl = 1, float expMultiplier = 1 | Метод для создания абстрактного врага. Производит инициализацию необходимых компонентов. |
| Start | protected | void | - | Встроенный метод. Используется для инициализации полей. |
| FixedUpdate | protected | void | - | Встроенный метод, который отрабатывает, каждые Time.fixedDeltaTime. |
| Hit | public | void | - | Обработчик события |
| Move | public | void | Vector3 target | Реализует перемещение врага к игроку. |
| Die | public | void | - | Обрабатывает смерть врага. |
| GetDamage | public | void | float damage | Обрабатывает получение урона. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса RagdollScript  Таблица 3.11 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| rbody | private | Rigidbody | Ссылка на компонент Rigidbody | |
| forceToAdd | private | float | Множитель для силы отталкивания | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| StartDeath | public | void | Vector3 force, float liveTime | Анимирует смерть врага. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса SaveSlotButton  Таблица 3.12 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| path | public | string | Путь до папки с сохранением | |
| \_dateTMP | private | TMPro.TextMeshProUGUI | Ссылка компонент с текстом | |
| \_text | private | string | Хранит значение для сериализации | |
| isAvaible | public | bool | Доступность сохранения | |
| Свойства | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| DateTMP | public | string | Свойство для работы с отображаемым текстом на кнопке сохранения | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| UpdateButton | public | void | bool forceActivate = false | Обновляет содержимое кнопки и ее отображение |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса ScrollingText  Таблица 3.13 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| \_textMesh | private | Player | Ссылка на главного героя | |
| \_duration | private | float | Получаемый опыт за убийство этого врага | |
| \_speed | private | float | Радиус, с которого враг замечает героя | |
| \_timer | private | float | Время жизни объекта | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Awake | private | void | - | Встроенный метод. Используется для инициализации полей. |
| FixedUpdate | private | void | - | Встроенный метод, который отрабатывает, каждые Time.fixedDeltaTime. |
| SetTextAndColor | public | void | string damage, Color color | Устанавливает текст и цвет. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса Singleton  Таблица 3.14 | | | | |
| Свойства | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| Instance | public | T | Свойство, хранящее ссылку на экземпляр класса. | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Awake | protected | void | - | Встроенный метод. Используется для инициализации свойства. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса SmoothFollowTarget  Таблица 3.15 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| target | public | GameObject | Ссылка на главного героя | |
| Offset | public | Vector3 | Расстояние до цели | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| LateUpdate | private | void | - | Встроенный метод, который вызывается в конце каждого кадра. Используется для перемещения камеры за героем |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса Player  Таблица 3.16 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| instance | public | Player | Ссылка на героя | |
| isAttacking | private | bool | Состояние атаки героя. Начата ли атака | |
| attackTarget | private | GameObject | Ссылка на противника | |
| Свойства | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| Level | public | int | Свойство, отвечающее за работу с уровнем героя | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Awake | private | void | - | Встроенный метод. Используется для инициализации значений |
| FixedUpdate | private | void | - | Встроенный метод, вызываемый каждый Time.fixedDeltaTime. Используется для анимации перемещения персонажа |
| ApplyPlayerData | public | void | PlayerData data | Принимает и обрабатывает сохраненные данные |
| AttackTarget | private | IEnumerator | - | Сопрограмма, реализующая атаку персонажа |
| Hit | public | void | - | Обработчик для завершения сопрограммы. |
| AttackHandler | public | void | GameObject target | Обработчик события атаки |
| GetDamage | public | void | float damage | Обрабатывает получение урона в размере damage. |
| Recovery | public | IEnumerator | float healRecovery = 5, float manaRecovery = 5, float delay = 1 | Восполняет здоровье и ману персонажа. Обновляет HUD. |
| Die | public | void | - |  |
| Clone | public | object | - | Клонирует данные героя |
| ResetRotine | public | IEnumerator | - | Завершает все корутины и после возобновляет Recovery |

|  |  |  |  |
| --- | --- | --- | --- |
| Описание полей методов и свойств класса Player.PlayerData  Таблица 3.17 | | | |
| Поля | | | |
| Имя | Модификатор  доступа | Тип | Назначение |
| Health | public | float | Хранение данных о здоровье героя |
| Manapool | public | float | Хранение данных о мане героя |
| Level | public | int | Хранение данных об уровне героя |
| Experience | public | float | Хранение данных о полученном опыте героя |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса Goblin  Таблица 3.18 | | | | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Awake | private | void | - | Встроенный метод. Используется для инициализации значений |
| ApplyData | public | void | GoblinData goblinData | Принимает и обрабатывает сохраненные данные |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса GoblinBoss  Таблица 3.19 | | | | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| Awake | private | void | - | Встроенный метод. Используется для инициализации значений |
| Die | public | void | - | Обработчик смерти босса - победа в игре. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса GoblinData  Таблица 3.20 | | | | |
| Поля | | | | |
| Имя | Модификатор  доступа | Тип | Назначение | |
| Health | public | float | Хранение данных о здоровье героя | |
| Manapool | public | float | Хранение данных о мане героя | |
| Level | public | int | Хранение данных об уровне героя | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| GoblinData | public | GoblinData | Goblin goblin | Конструктор. Принимающий Goblin и получающий оттуда необходимые значения. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса Cutscene\_dock  Таблица 3.21 | | | | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| OnTriggerStay | private | void | Collider other | Встроенный метод. Используется для обработки нахождения в триггере |
| OnTriggerExit | private | void | Collider other | Встроенный метод. Используется для обработки выхода из триггера |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание полей методов и свойств класса Cutscene\_dungeon  Таблица 3.22 | | | | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| OnTriggerStay | private | void | Collider other | Встроенный метод. Используется для обработки нахождения в триггере |
| OnTriggerExit | private | void | Collider other | Встроенный метод. Используется для обработки выхода из триггера |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Описание методов и свойств интерфейса ICharacter  Таблица 3.23 | | | | |
| Методы | | | | |
| Имя | Модификатор  доступа | Тип | Аргументы | Назначение |
| GetDamage | public | void | float damage | Получиет урон в размере damage. |
| Attack | public | void | ICharacter character | Атакует выбранную цель. |
| Die | public | void | - | Обрабатывает смерть персонажа. |
| Recovery | public | IEnumerator | float healRecovery = 5, float manaRecovery = 5, float delay = 1 | Восполняет здоровье и ману персонажа. |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированх |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

# **ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**