ПРОГРАММНЫЙ КОМПЛЕКС
НАЗВАНИЕ КОМПЛЕКСА
КОМПОНЕНТ

Пояснительная записка

(Электронный носитель данных (веб-модуль))

Листов 10

2021

Изм. №1 от 13.06.2021

Литера «О»

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

АННОТАЦИЯ

Настоящий документ является пояснительной запиской к программному продукту «Runes of Oblivion». Документ содержит описание постановки задачи, детальное описание архитектуры и принципов функционирования программы, обоснование выбора алгоритмических и технических решений, а также требования к программным и техническим средствам. Проект разработан в качестве учебного по дисциплине «Генеративный Искусственный Интеллект» в НИЯУ МИФИ.

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ

Настоящая пояснительная записка подготовлена в соответствии с требованиями ГОСТ 19.404-79 «Единая система программной документации. Пояснительная записка» и ГОСТ 19.106-78 «Требования к программным документам, выполненным печатным способом».

Программа «Runes of Oblivion» представляет собой браузерное приложение, реализующее тактическую пошаговую игру с карточной механикой и процедурной генерацией.

2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

2.1. Назначение модуля

Программный модуль «Runes of Oblivion» предназначен для демонстрации прототипа пошаговой тактической игры, сочетающей механику roguelike (с процедурной генерацией и пермадентью), карточной системой и пошаговыми боями на клетчатой сетке. Модуль реализует полный игровой цикл: генерацию карты забега, управление боевыми системами (поиск пути, AI противников, карточные эффекты) и визуализацию игрового процесса.

Основные функции модуля:

- Процедурная генерация ветвящейся карты забега с узлами различного типа (FIGHT, EVENT, SHOP, BOSS).
- Реализация пошагового боевого цикла на клетчатой сетке с учётом препятствий.
- Управление колодой карт (розыгрыш, сброс, перетасовка) и применение эффектов карт.
- Реализация искусственного интеллекта для противников (поиск пути, тактические решения).
- Визуализация игрового состояния в реальном времени с помощью Canvas API и DOM-элементов.
- Управление ресурсами: здоровье (HP), мана (MP), очки действия (AP) и награды.

2.2. Область применения

Модуль применяется в качестве учебного проекта для демонстрации навыков архитектурного проектирования, реализации игровых алгоритмов, веб-разработки и организации кода. Практическое применение включает:

Обучение архитектурным паттернам (MVC, Event Emitter, Singleton).

Демонстрацию алгоритмов поиска пути (BFS) и процедурной генерации.

Демонстрацию работы с Canvas API и современными веб-технологиями (JavaScript ES6, модули, Vite).

Анализ игровых систем: баланс карт, настройка AI, процедурная генерация.

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи на разработку программы

Требуется разработать браузерное приложение со следующими требованиями:

Реализовать процедурную генерацию ветвящейся карты забега (8–10 этажей с 2–3 узлами на этаже).

Реализовать систему пошагового боя на сетке 10×8 клеток с препятствиями.

Реализовать карточную систему с розыгрышем, сбросом и перетасовкой колоды.

Реализовать AI противников с поиском пути и принятием тактических решений.

Реализовать визуализацию сетки, юнитов, карт и состояния игры.

Обеспечить кроссплатформенность (работа в браузере на Windows/Linux/macOS).

Предоставить чистую архитектуру с отделением логики от отображения.

3.2. Описание алгоритма и функционирования программы

Игровой цикл:

Инициализация: загрузка ресурсов, инициализация GameState.

Экран карты забега: игрок выбирает узел (FIGHT, BOSS).

Загрузка сцены: в зависимости от типа узла инициализируется бой или событие.

Боевой цикл: ход игрока → ход противников → проверка условий завершения.

Награда: выдача карт, восстановление здоровья; возврат на карту забега.

Окончание: смерть игрока (GAME OVER) или победа над боссом (WIN).

Боевая система:

Сетка 10×8 клеток; юниты занимают по одной клетке.

Ход игрока: выбор карты из руки, выбор цели (клетка или юнит), применение эффекта.

Проверка применения карты: достаточно ли маны, цель в диапазоне, цель валидна.

Ход противников: поиск пути к игроку (алгоритм BFS), атака при достижении дистанции.

Проверка победы: нет живых врагов → победа; HP игрока 0 → поражение.

Карточная система:

Колода состоит из карт типов: Удар, Защита, Лечение, Рывок, Призыв.

Розыгрыш: карта из стопки добора → рука игрока (до 5 карт).

Применение: карта из руки → сброс.

Перетасовка: когда стопка добора пуста, сброс перетасовывается → новая стопка добора.

AI и поиск пути:

Враги выбирают действие по иерархии приоритетов:

Если игрок в дистанции атаки → атака.

Если видна турель игрока → атака турели.

Если враг слишком близко (для дальнобойных) → отступление.

Иначе → поиск пути к игроку.

Поиск пути реализован алгоритмом BFS с учётом препятствий и занятых клеток.

Процедурная генерация:

Карта забега генерируется как ориентированный граф (DAG).

Каждый этаж содержит 2–3 узла; каждый узел имеет тип: FIGHT, BOSS.

Враги и препятствия варьируются по этажам (сложность растёт).

Использование детерминированной генерации (LCG seed) для воспроизводимости.

3.3. Архитектура и структура проекта

Программа реализована как модульное веб-приложение на JavaScript (ES6) с чистой архитектурой, разделяющей логику, состояние и отображение.

Основные модули:

GameManager (main controller): управление переходами между сценами, координация всех модулей.

GameState (singleton): хранение полного состояния игры; подписка на события.

GridModel: модель сетки, хранение позиций юнитов и препятствий.

UnitModel: модель юнита (здоровье, мана, координаты, характеристики).

CardLibrary: библиотека карт и их параметров (стоимость, эффекты, дальность).

CardEffects: реализация эффектов карт (паттерн Strategy).

BattleLogic: логика боя (проверка применения, расчёт урона, применение эффектов).

AILogic: принятие решений врагов (выбор действия, поиск цели).

PathFinding: алгоритм BFS для поиска пути.

GridRenderer: отрисовка сетки и препятствий на Canvas.

UnitRenderer: отрисовка юнитов (спрайты, эффекты, индикаторы).

UIManager: управление DOM-элементами (рука карт, статистика, уведомления).

RunMapGenerator: генерация карты забега.

RunFlowController: управление переходами между узлами карты.

EventEmitter: система событий для синхронизации между модулями.

3.4. Описание и обоснование выбора метода организации входных и выходных данных

Входные данные:

Действия пользователя (через mouseclick, keydown):

Клик по карте на экране карты забега (выбор узла).

Клик по карте в руке (выбор карты для применения).

Клик по клетке сетки (выбор цели для карты).

Нажатие Enter для подтверждения выбора.

Конфигурационные параметры (констан ты в constants.js):

Размеры сетки (10×8), радиусы зрения врагов, стоимость карт.

Выходные данные:

Визуализация на Canvas: сетка, юниты, препятствия, анимация эффектов.

Информация в DOM: здоровье/мана игрока, колода (draw/discard), рука, логи действий.

Уведомления: ошибки («Not enough mana», «Target out of range»), результаты боёв.

Организация данных:

Централизованное состояние через GameState (паттерн Singleton).

Подписка визуального слоя (Renderer, UIManager) на события изменения состояния (Event Emitter).

Отвязка логики от представления: GameManager вызывает BattleLogic, BattleLogic генерирует события, Renderer реагирует на события и обновляет визуальное представление.

3.5. Описание и обоснование выбора состава технических и программных средств

Технические средства:

Персональный компьютер / ноутбук с ОС Windows, macOS или Linux.

Разрешение экрана не ниже 1366×768 пикселей.

Интернет-соединение (для доступа к публикуемому приложению).

Программные средства:

Браузер с поддержкой HTML5 Canvas и ES6: Chrome 90+, Firefox 88+, Safari 14+, Edge 90+.

Среда разработки: VS Code.

Средства сборки: Node.js 16+, npm, Vite.

Обоснование выбора:

JavaScript + Canvas: Полная кроссплатформенность; работа без установки; полный контроль над отрисовкой.

Vite: Быстрая сборка; поддержка модулей ES6; Hot Module Replacement для разработки.

Cloudflare Pages: Бесплатный хостинг; автоматический деплой из GitHub; высокая надёжность.

Ванильный JavaScript (без фреймворков): Демонстрирует глубокое понимание архитектуры; отсутствие зависимостей упрощает развёртывание.

4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Функциональные показатели:

Полная реализация игровых систем: карточная механика, боевая система, AI.

Поддержка до 5–6 одновременно активных враагов без заметного торможения.

Плавная отрисовка (60 FPS) при стандартном разрешении экрана.

Технико-экономические показатели:

Снижение затрат на развёртывание: приложение доступно по одной ссылке без установки ПО.

Упрощение демонстрации и тестирования: запуск на любом устройстве через браузер.

Кроссплатформенность: работает идентично на Windows, macOS, Linux.

Объём исходного кода: ~15–20 тысяч строк JavaScript, организованных в модульную архитектуру.

Размер финального bundle (gzip): ~116 кБ (исходный ~350 кБ).

Учебная ценность:

Демонстрация архитектурных паттернов (MVC, Event Emitter, Singleton, Strategy).

Реализация классических алгоритмов (BFS для поиска пути, процедурная генерация).

Практика работы с современным веб-стеком (JavaScript ES6, модули, Vite).

Анализ и балансировка игровых систем.

5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. Introduction to Algorithms. MIT Press, 2009.

2. Vite Documentation: https://vitejs.dev/

ПЕРЕЧЕНЬ ССЫЛОЧНЫХ ДОКУМЕНТОВ

1. ГОСТ 19.101-77. Единая система программной документации. Виды программ и программных документов [текст]. — Введ. 1980-01-01. — М.: Стандартинформ, 2010. — 4 с. — (Единая система программной документации).

2. ГОСТ 19.103-77. Единая система программной документации. Обозначения программ и программных документов [текст]. — Введ. 1980-01-01. — М.: Стандартинформ, 2010. — 3 с. — (Единая система программной документации).

3. ГОСТ 19.104-78. Единая система программной документации. Основные надписи [текст]. — Введ. 1980-01-01. — М.: Стандартинформ, 2010. — 7 с. — (Единая система программной документации).

4. ГОСТ 19.106-78. Единая система программной документации. Требования к программным документам, выполненным печатным способом [текст]. — Введ. 1980-01-01. — М.: Стандартинформ, 2010. — 11 с. — (Единая система программной документации).

5. ГОСТ 19.404-79. Единая система программной документации. Пояснительная записка. Требования к содержанию и оформлению [текст]. — Введ. 1981-01-01. — М.: Стандартинформ, 2010. — 3 с. — (Единая система программной документации).

6. ГОСТ 19781-90. Обеспечение систем обработки информации программное. Термины и определения [текст]. — Взамен ГОСТ 19781-83, ГОСТ 19.004-80 ; введ. 1992-01-01. — М.: Стандартинформ, 2010. — 14 с.