УДК 004.42

ПРОЕКТИРОВАНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ 2D-ИГРЫ «COLONY-13» В ЖАНРЕ SURVIVAL TOWER DEFENSE

Никулин С.А.

студент,

ФГБОУ ВО Уфимский университет науки и технологий,

Уфа, Россия

Шехтман Л.И.

к.ф.-м.н., доцент,

ФГБОУ ВО Уфимский университет науки и технологий,

Уфа, Россия

Аннотация

В статье рассматривается процесс проектирования и программной реализации 2D-игры «Colony-13» в жанре survival tower defense. Проект разрабатывается в рамках практического цикла обучения на технической специальности в вузе. Применялся игровой движок Unity и язык программирования C#. Основной игровой цикл построен на чередовании дневной фазы подготовки и ночной фазы обороны базы. Игрок управляет инженером аварийной колонии, собирает ресурсы, строит стены и турели, уничтожает врагов и восстанавливает передатчик связи. В работе описаны реализованные игровые механики: система движения и здоровья игрока, сбор металла и биоядер, строительство оборонительных объектов, поведение врагов, дневной и ночной цикл, миникарта, случайное размещение ресурсов и математическая модель усиления волн противников. Особое внимание уделяется связи между ресурсной системой, прогрессом ремонта передатчика и увеличением сложности ночных атак.

Ключевые слова: компьютерная игра, Unity, C#, survival tower defense, 2D-игра, игровая механика, ресурсы, волны врагов, процедурное размещение объектов.

DESIGN AND SOFTWARE IMPLEMENTATION OF THE 2D GAME "COLONY-13" IN THE SURVIVAL TOWER DEFENSE GENRE

Nikulin S.A.

student,

Ufa University of Science and Technology,

Ufa, Russia

Shehtman L.I.

PhD in Physics and Mathematics, Associate Professor,

Ufa University of Science and Technology,

Ufa, Russia

Abstract. The article examines the design and software implementation of the 2D game "Colony-13" in the survival tower defense genre. The project is being developed as part of a practical training cycle for a technical degree program at a university. The Unity game engine and the C# programming language were used. The core gameplay loop is based on the alternation of a daytime preparation phase and a nighttime base defense phase. The player controls an engineer of an emergency colony, collects resources, builds walls and turrets, destroys enemies, and repairs a communication transmitter. The paper describes the implemented game mechanics: player movement and health system, metal and biocore collection, defensive construction, enemy behavior, day and night cycle, minimap, random placement of resources, and a mathematical model for increasing enemy waves. Special attention is paid to the relationship between the resource system, transmitter repair progress, and the growing difficulty of nighttime attacks.

Keywords: computer game, Unity, C#, survival tower defense, 2D game, game mechanics, resources, enemy waves, procedural object placement.

В настоящее время компьютерные игры активно внедряются в образовательный процесс. Одним из возможных направлений является создание игровых проектов силами самих обучающихся, например [2; 4]. Такая деятельность способствует повышению интереса к учебе. Кроме того, выбор игровой разработки имеет и другие преимущества:

- студенты легко могут получить опыт в использовании готовых программных продуктов такого типа (и большинство имеют такой опыт еще до начала обучения в вузе), так что им понятно, элементы каких программных продуктов они учатся создавать, какие к ним требования и к чему надо стремиться;

- тестирование игровых программ очень наглядно и легко заметны недостатки.

В рамках практического цикла обучения на технической специальности в вузе перед студентами была поставлена задача разработать собственную компьютерную игру.

Современная разработка компьютерных игр представляет собой комплексный процесс, включающий проектирование игровой идеи, создание визуальных ресурсов, реализацию программной логики, настройку интерфейса, балансировку и тестирование. Даже небольшой учебный игровой проект требует согласованной работы нескольких систем: управления персонажем, обработки столкновений, поведения врагов, пользовательского интерфейса, расчета ресурсов и условий завершения игры [1; 5].

Одним из распространенных направлений в инди-разработке являются игры, сочетающие элементы выживания и защиты базы (survival tower defense). В таких проектах игрок не только отражает атаки противников, но и заранее готовится к ним: собирает ресурсы, строит оборонительные сооружения, ремонтирует поврежденные объекты и принимает решения о развитии базы. Подобный игровой цикл удобен для учебной разработки, поскольку позволяет

реализовать несколько взаимосвязанных механик и показать работу компонентной архитектуры игрового движка.

В рамках данной работы разрабатывается 2D-игра «Colony-13» в жанре survival tower defense. Действие игры происходит на планете Терра-9, где аварийная колония оказывается под угрозой нападения враждебных существ и старых боевых дронов. Игрок управляет инженером колонии, который должен защитить командный модуль и восстановить передатчик связи. Победа достигается не простым выживанием в течение заданного количества дней, а полным ремонтом передатчика, что делает ресурсную систему более значимой.

Выбор игрового движка является важной задачей [3]. В данном случае игра реализуется в Unity. Данный движок предоставляет средства для разработки 2D-проектов, работы со спрайтами, физикой, сценами, префабами и пользовательским интерфейсом. Программная логика проекта создается на языке C#, который используется для написания пользовательских скриптов, обработки ввода, управления игровыми объектами и реализации взаимодействия между системами.

На рисунке 1 представлена общая схема игрового цикла. Игровой процесс строится на чередовании дневной и ночной фаз. Днем игрок исследует карту, собирает металл и биоядра, строит стены и турели, улучшает защитные объекты и ремонтирует передатчик. Ночью база подвергается атаке, а игрок должен защитить командный модуль от разрушения.

В жанре survival tower defense (выживание с элементами башенной защиты) слово «турель» (от англ. turret – башенная установка) означает стационарную автоматическую или полуавтоматическую оборонительную установку, которую игрок строит и размещает для защиты своей базы, ресурсов или персонажа от волн врагов.

Основные функции и особенности турелей:

– Автоматическая защита: турель сама сканирует территорию, захватывает цель и открывает огонь по врагам, когда они входят в ее радиус действия; это

позволяет игроку заниматься другими делами (собирать ресурсы, чинить турели) или пережидать сложные ночные волны в безопасности.

– Крафт и постройка: турели в играх редко даются бесплатно, в данной игре игроку приходится ради этого добывать металл.

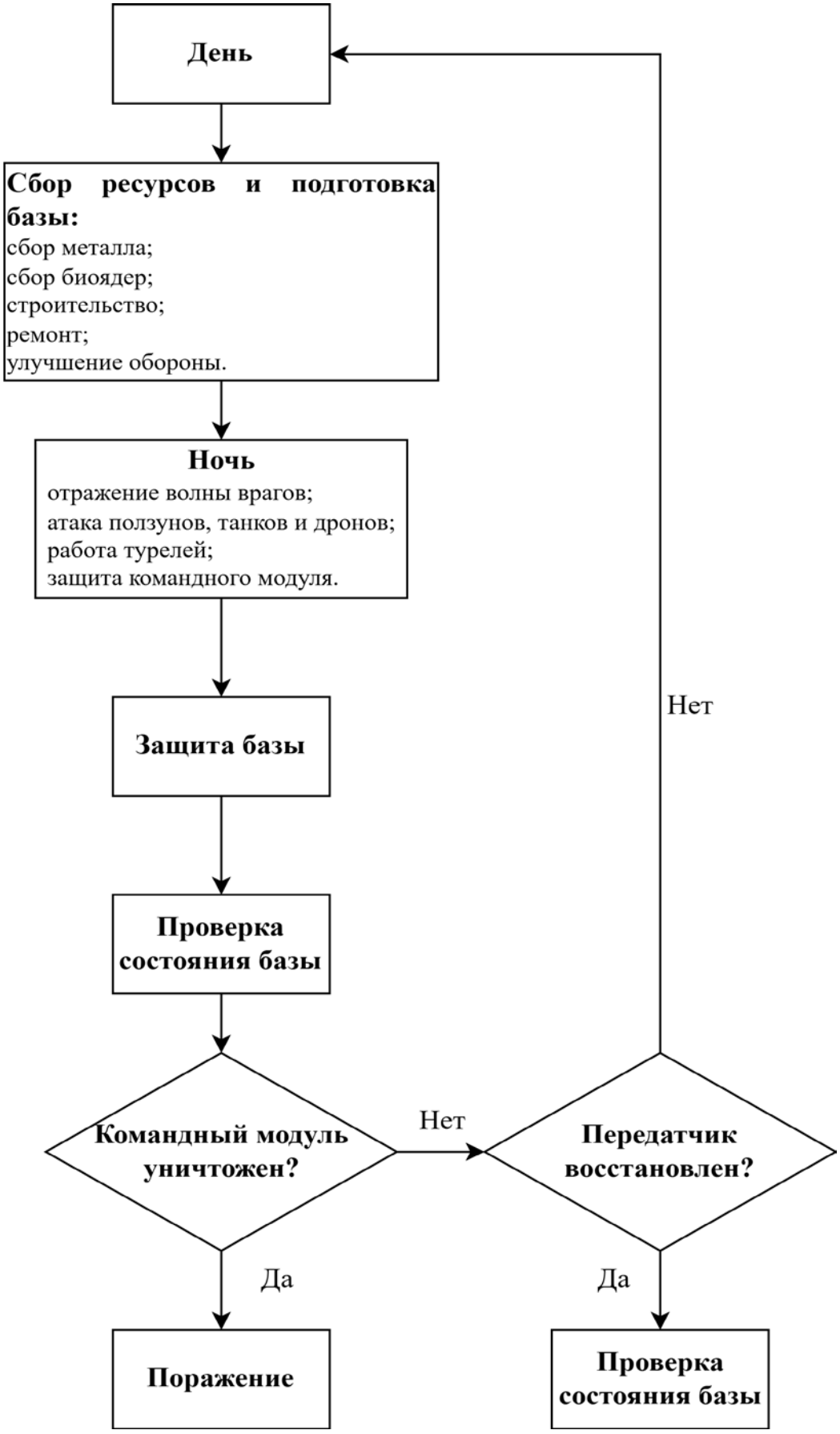

Рис. 1 – Схема основного игрового цикла «Colony-13» (авторская разработка)

– Уязвимость: турели имеют запас прочности; умные враги могут целенаправленно уничтожать их, поэтому игроки часто строят вокруг турелей стены, укрытия или размещают их за спинами друг друга для перекрестного огня.

– Улучшение: по мере прохождения или развития базы турели можно улучшать, увеличивая наносимый ими урон, скорострельность, радиус действия.

Реализованные механики перечислены в таблице 1. Они связаны между собой. Сбор металла и биоядер обеспечивает развитие базы, строительство стен и турелей повышает устойчивость обороны, а ремонт передатчика формирует конечную цель прохождения. При этом командный модуль остается главным защищаемым объектом: его разрушение приводит к поражению.

Таблица 1 – Основные реализованные механики игры «Colony-13»

Механика	Назначение	Статус
Управление игроком	Перемещение инженера по карте, взаимодействие с объектами	Реализовано
Здоровье игрока	Определяет возможность продолжения прохождения	Реализовано
Металл	Основной ресурс для строительства и ремонта	Реализовано
Биоядра	Редкий ресурс для ремонта поздних этапов передатчика и улучшений	Реализовано
Командный модуль	Главный объект базы; при разрушении наступает поражение	Реализовано
Стены и турели	Оборонительные объекты для защиты базы	Реализовано
Мясные наросты	Опасные объекты карты с охранниками и ценными ресурсами	Реализовано
Передатчик	Объект, восстановление которого является условием победы	Реализовано
Миникарта	Показывает базу, игрока, врагов и важные объекты	Реализовано

Главное меню выступает начальной точкой взаимодействия пользователя с проектом. Через него игрок переходит к игровой сцене. Пример главного меню показан на рисунке 2.

На рисунке 3 представлена основная игровая сцена. В центре игрового процесса находится командный модуль. Вокруг него игрок размещает стены и турели, собирает ресурсы и отражает атаки врагов.

Рис. 2 – Главное меню игры «Colony-13» (авторская разработка)

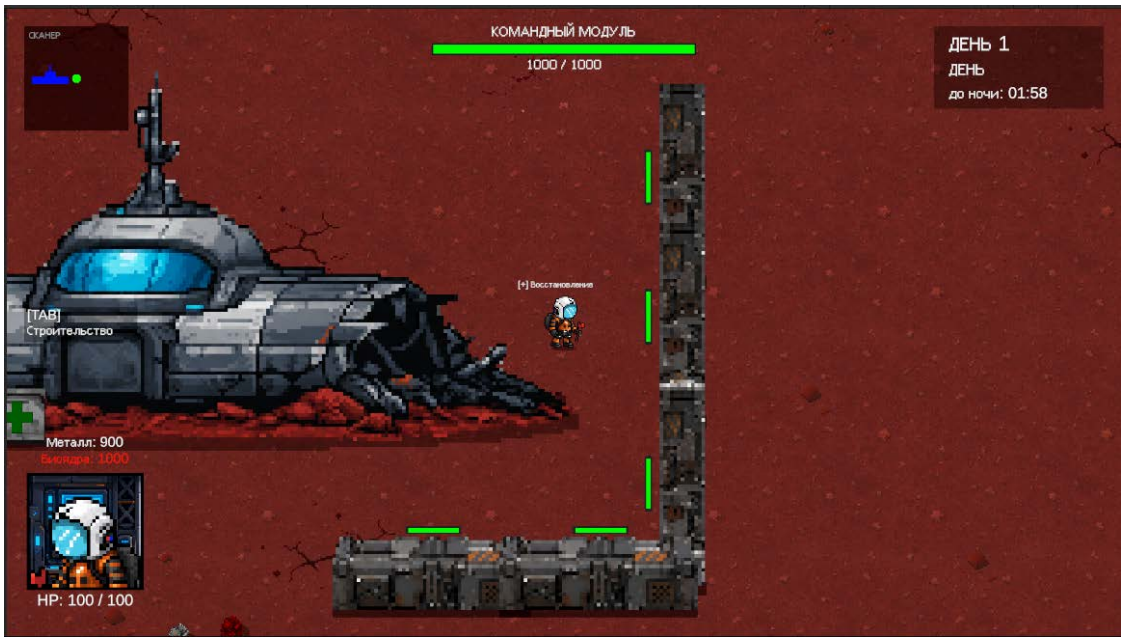

Рис. 3 – Игровая сцена с командным модулем, игроком и оборонительными объектами (авторская разработка)

Внутриигровая экономика основана на двух ресурсах: металле и биоядрах. Металл используется для строительства и ремонта, а биоядра применяются как редкий ресурс для поздних этапов восстановления передатчика и улучшения

защитных объектов. Такое разделение позволяет создать выбор между быстрым продвижением к победе и укреплением базы.

Для расчета количества ресурсов на карте используется формула:

$$Rкарты = (Rпобеда + Rоборона + Rулучшения) \times Kзапаса,$$

где Rкарты – общее количество ресурса на карте; Rпобеда – ресурс, необходимый для достижения победы; Rоборона – ресурс, необходимый для минимальной защиты базы; Rулучшения – ресурс, необходимый для развития оборонительных объектов; Kзапаса – коэффициент запаса. В проекте коэффициент запаса может составлять 1,2–1,3, что дает игроку возможность допустить ошибки в распределении ресурсов.

Пример расчета количества ресурсов приведен в таблице 2. При округлении итоговое количество металла на карте составляет около 700 единиц. Биоядра рассчитываются отдельно, поскольку часть из них должна быть гарантированно доступна через уничтожение мясных наростов, а дополнительные биоядра могут выпадать с редких врагов.

Таблица 2 – Пример расчета металла для прохождения игры

Статья расхода	Количество металла	Пояснение
Ремонт передатчика	250	Пять этапов восстановления
Минимальная оборона	200	Стены, турели и базовая защита
Улучшения	120	Часть улучшений стен и турелей
Итого без запаса	570	Сумма основных расходов
Итого с запасом 1,2	684	Округляется до 700 единиц металла

Важной частью проекта является система усиления ночных атак. В первоначальной версии победа могла быть связана только с выживанием в течение заданного количества дней. В текущей модели победа зависит от ремонта передатчика, а дни используются как фактор роста сложности. Количество врагов в ночной волне определяется по формуле:

$$N(d, s) = N0 + a \times (d - 1) + b \times s,$$

где N(d, s) – количество врагов в ночной атаке; N0 – базовое количество врагов; d – номер игрового дня; s – этап ремонта передатчика; a – прирост врагов за каждый день; b – прирост врагов за каждый завершенный этап ремонта передатчика. Такая модель связывает прогресс игрока с усилением угрозы: чем ближе игрок к победе, тем опаснее становятся ночные атаки.

В таблице 3 приведен пример расчета при N0 = 3, a = 2 и b = 2. Модель позволяет гибко изменять сложность через несколько параметров, не переписывая логику игры. Кроме того, она делает ремонт передатчика рискованным решением: быстрый ремонт приближает победу, но одновременно увеличивает силу атак.

Таблица 3 – Пример расчета количества врагов в ночной атаке

День	Этап ремонта	Формула	Количество врагов
1	0	3 + 2 × (1 - 1) + 2 × 0	3
2	1	3 + 2 × (2 - 1) + 2 × 1	7
3	2	3 + 2 × (3 - 1) + 2 × 2	11
4	4	3 + 2 × (4 - 1) + 2 × 4	17

Для повышения вариативности прохождения в игре реализовано случайное размещение металла и мясных наростов (процедурная генерация). При запуске игровой сцены генератор карты выбирает точки появления объектов в пределах заданных границ. При этом учитываются ограничения: ресурсы не должны появляться слишком близко к базе и игроку, а мясные наросты должны располагаться на удалении от безопасной зоны.

Мясные наросты выполняют роль опасных точек интереса (рис. 4). Рядом с ними появляются охранники, а после уничтожения нароста игрок получает доступ к ценным ресурсам. Такая структура создает риск и награду: безопасный металл можно собирать рядом с базой, а биоядра требуют выхода в более опасные зоны карты.

Программная структура проекта построена на компонентном подходе Unity. Каждый значимый объект сцены получает набор компонентов,

отвечающих за визуальное отображение, физическое взаимодействие и игровую логику. Например, игрок содержит компоненты движения, здоровья и инвентаря; враги используют скрипт поведения и компонент здоровья; постройки имеют собственные скрипты прочности и улучшения.

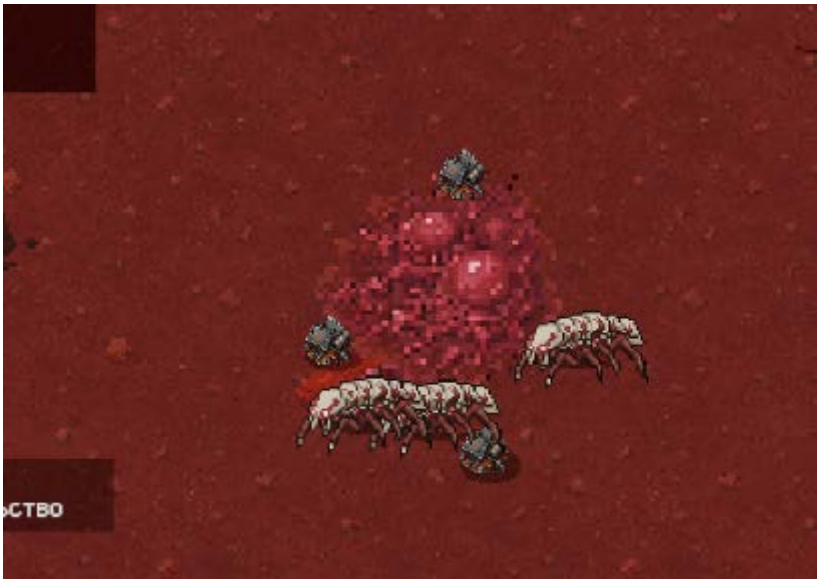

Рис. 4 – Мясной нарост с охранниками и ресурсами (авторская разработка)

Основные системы проекта можно разделить на несколько групп (рис. 5). Система игрока отвечает за перемещение, здоровье и сбор ресурсов. Система врагов управляет выбором цели, движением, атакой и выпадением наград. Система строительства отвечает за размещение стен и турелей. Система игрового цикла переключает день и ночь, рассчитывает волну врагов и передает параметры в спавнер (класс, который создает новых врагов). Система интерфейса отображает здоровье, ресурсы, текущий день, фазу суток и миникарту (рис. 6).

Использование префабов (это шаблонов игровых объектов, сохранённых в виде отдельных файлов в проекте игры) позволяет повторно применять настроенные игровые объекты. Например, враги, снаряды, ресурсы, стены, турели и мясные наросты могут храниться как префабы, после чего создаваться в сцене через скрипты. Это упрощает балансировку и уменьшает количество ручной настройки объектов.

Интерфейс в «Colony-13» выполняет не декоративную, а функциональную роль. Игрок должен постоянно видеть количество металла и биоядер, здоровье персонажа, состояние командного модуля, текущий день, фазу суток и время до смены фазы. Миникарта помогает ориентироваться на карте и отслеживать положение важных объектов и врагов.

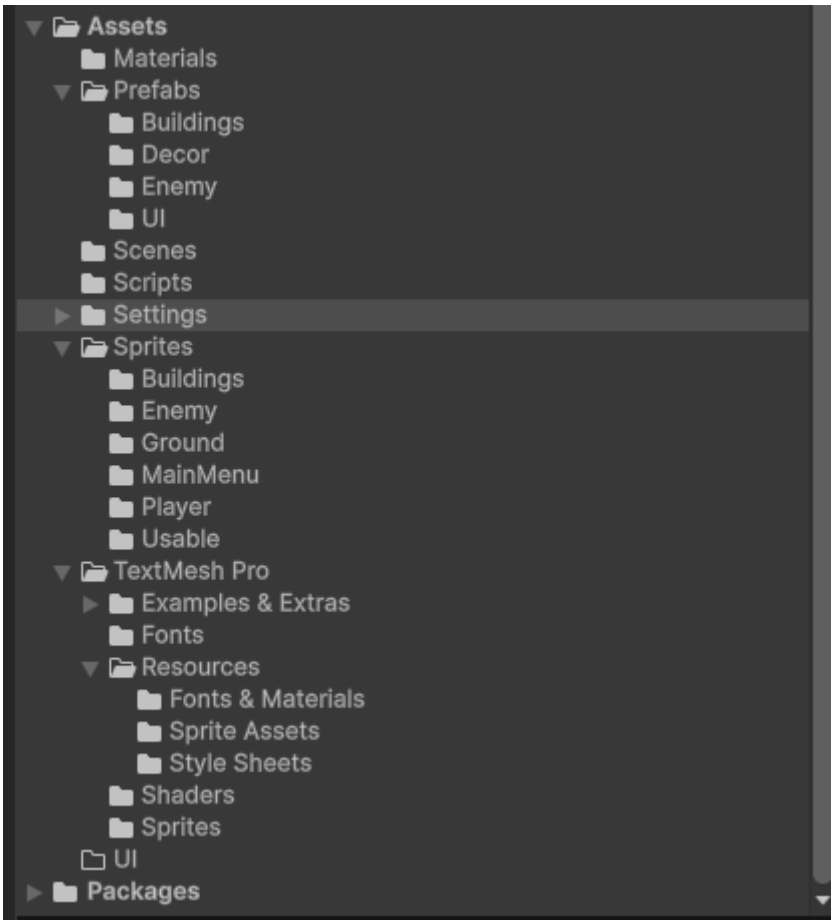

Рис. 5 – Пример структуры проекта в Unity (авторская разработка)

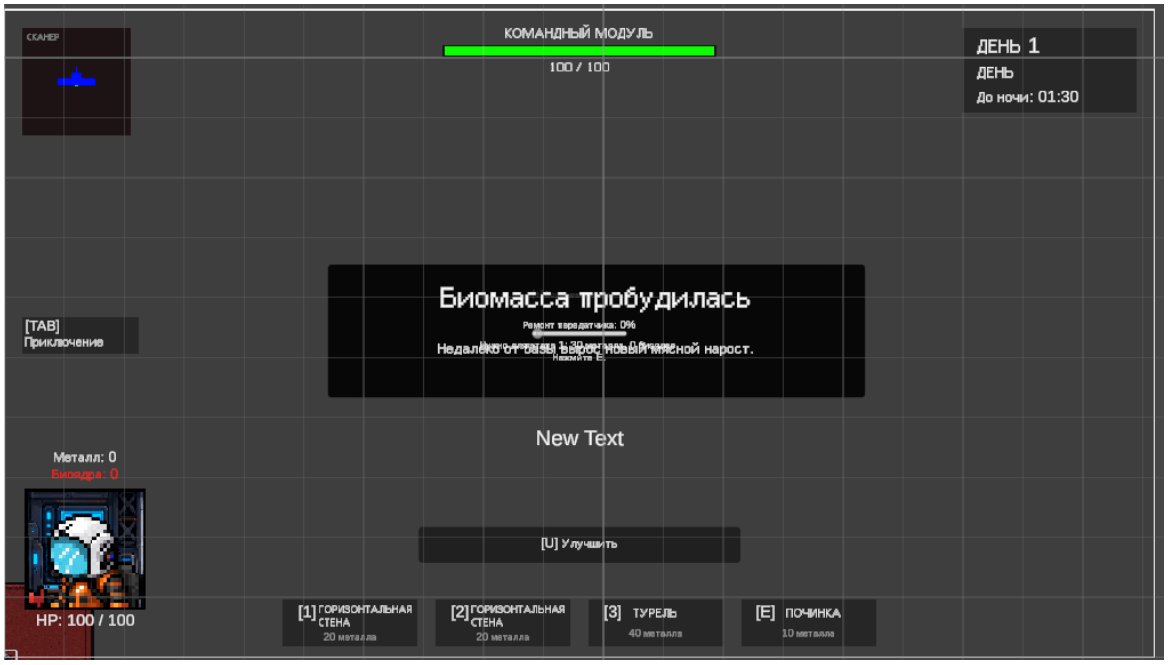

Рис. 6 – Интерфейс игровой сцены и миникарта (авторская разработка)

Отдельное значение имеют экраны победы и поражения (рис. 7). Победа достигается после полного восстановления передатчика связи, а поражение наступает при разрушении командного модуля или гибели игрока.

В результате разработки получен рабочий прототип 2D-игры «Colony-13», включающий основные элементы жанра survival tower defense. Реализованы управление игроком, здоровье, инвентарь, сбор металла и биоядер, командный модуль, стены, турели, улучшения защитных объектов, враги нескольких типов, мясные наросты, дневной и ночной цикл, миникарта, передатчик как условие победы, случайное размещение ресурсов и математическая модель усиления волн.

Разработанная игра показывает, что даже в рамках небольшого учебного проекта можно связать несколько систем в единый игровой цикл. Ресурсы используются не только для строительства, но и для достижения победы. Враги усиливаются не произвольно, а по формуле, учитывающей день и прогресс ремонта. Случайное размещение объектов повышает вариативность карты и делает каждое прохождение менее предсказуемым.

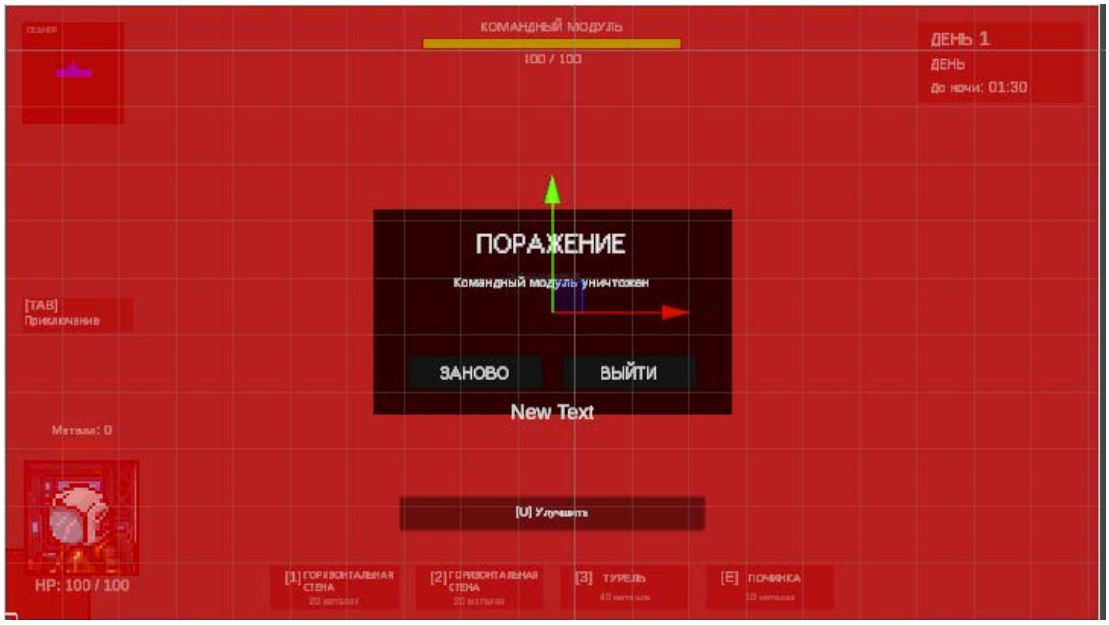

Рис. 7 – Экран поражения (авторская разработка)

Перспективами дальнейшей работы являются переработка системы строительства в режим планирования, добавление экрана итогов ночной атаки, ворот базы, системы сохранения, звукового оформления, улучшений персонажа и дополнительных зон карты. Эти элементы позволят сделать игру более завершенной и расширить стратегическую составляющую.

Заключение

В статье рассмотрены проектирование и программная реализация 2D-игры «Colony-13» в жанре survival tower defense. Основной результат работы заключается в создании взаимосвязанной системы игровых механик, включающей сбор ресурсов, строительство обороны, дневной и ночной цикл, поведение врагов, ремонт передатчика и математическую модель усиления волн. Проект демонстрирует возможности Unity и C# для реализации учебной 2D-игры с элементами выживания, защиты базы и контролируемой процедурной генерации.

Библиографический список:

1. Костер Р. Разработка игр и теория развлечений / Р. Костер. – Москва : ДМК Пресс, 2018. – 289 с.

2. Пименов, Д. С. Разработка компьютерной игры «Иван Царевич - оборотень в погонах» / Д. С. Пименов, А. Ф. Бактыбаев, Э. Р. Латыпов // Мавлютовские чтения : Материалы XIX Всероссийской молодёжной научной конференции. В 8-ми томах, Уфа, 24–28 ноября 2025 года. – Уфа: Уфимский университет науки и технологий, 2025. – С. 1267-1271. – EDN GEISUA.

3. Попов В. Л., Шехтман Л. И., Лапин А. Н. Аналитический обзор функциональных инструментов современных игровых движков в контексте разработки интерактивных приложений // Информационные технологии. Проблемы и решения. – 2025. – № 1 (30). – С. 90–96. – EDN IILQWZ.

4. Сахипов, А. Э. Разработка игры «Симулятор огородника» в жанре Tower Defense на платформе Unity / А. Э. Сахипов // Мавлютовские чтения : Материалы XIX Всероссийской молодёжной научной конференции. В 8-ми томах, Уфа, 24–28 ноября 2025 года. – Уфа: Уфимский университет науки и технологий, 2025. – С. 1283-1287. – EDN HCCQUQ.

5. Тайнан С. Геймдизайн. Рецепты успеха лучших компьютерных игр от Super Mario и Doom до Assassin’s Creed и дальше / С. Тайнан. – Санкт-Петербург : Питер, 2020. – 448 с.