Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Алтайский государственный университет»

Школа развития цифровых компетенций «Digital Up» (цифровая кафедра)

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

**на итоговый проект «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»**

**по ДПП ПП «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»**

| п/п  № | Задание | Исполнитель | Рабочий график (план) выполнения |
| --- | --- | --- | --- |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |

Руководитель проекта   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / «\_\_\_ » \_\_\_\_\_\_\_\_\_ 2025 г.

(подпись)

Соруководитель проекта *(при наличии)*   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / «\_\_\_ » \_\_\_\_\_\_\_\_\_ 2025 г.

(подпись)

СОГЛАСОВАНО:

Руководитель ДПП ПП

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_2024 г.

(подпись)

Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Алтайский государственный университет»

Школа развития цифровых компетенций «Digital Up» (цифровая кафедра)

Отчет о выполнении группового итогового проекта по ДПП ПП

«Основы Gamedev и VR-разработки»

**«Scalable Metroidvania Core»**

Исполнители:

Ненашев В.И.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«27» Мая 2025 г.

Руководитель проекта

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_ 2025 г.

г. Барнаул, 2025

# 1. Цель проекта

Разработка расширяемого технологического фундамента и игрового прототипа в жанре платформера с элементами метроидвании на Unity, демонстрирующего ключевые игровые механики и спроектированного для последующей эволюции в полноценную нелинейную игру.

# 2. Задачи проекта

1. **Проектирование и реализация модульной и расширяемой архитектуры игровых систем:** Обеспечение гибкости, поддержки и простоты добавления нового функционала в будущем (система состояний, способности, здоровье/урон).
2. **Разработка отзывчивого и функционального контроллера персонажа:** Реализация базового передвижения (ходьба, бег, прыжок) и ключевых способностей (рывок, скольжение, прыжок со стены).
3. **Создание усовершенствованной системы состояний персонажа:** Обеспечение четкого управления логикой поведения, анимациями и переходами между заданными наборами действий (передвижение, направление взгляда и т.д.).
4. **Построение системы способностей:** Создание основы для легкого добавления и управления активными навыками персонажа (рывок, прыжок, скольжение) с фокусом на расширяемость.
5. **Реализация кастомной системы управления анимациями:** Интеграция анимаций персонажа и эффектов с логикой состояний и способностей, обеспечивающая плавность и визуальную отзывчивость.
6. **Разработка системы здоровья:** Создание модульных компонентов (здоровье, нанесение и получение урона) для управления жизнями игрока и врагов, обработки столкновений и эффектов.
7. **Проектирование и создание демонстрационного игрового уровня:** Разработка компактного уровня, наглядно представляющего все реализованные механики, включая платформинг, преодоление препятствий, ловушки и зацепки для будущего расширения.
8. **Обеспечение базовой оптимизации и управляемости кода:** Следование принципам SOLID/KISS, использование паттернов проектирования, организация структуры проекта для поддержки и дальнейшего развития.

# 3. Актуальность и востребованность проекта

Данный проект обладает значительной актуальностью и востребованностью по нескольким ключевым аспектам:

1. **Образовательная ценность:** Проект является практической реализацией сложных концепций геймдев-разработки на Unity, включая проектирование архитектуры, реализацию физики персонажа, создание систем состояний и способностей, управление анимациями и проектирование игровых уровней. Он наглядно демонстрирует владение инструментарием и понимание процессов создания игр.
2. **Практическая значимость для разработчика:** Проект служит портфолио-проектом, подтверждающим навыки в:

* Создании чистого, расширяемого и хорошо документированного кода на C#.
* Применении принципов ООП и паттернов проектирования (States, ScriptableObject и др.) в игровом контексте.
* Работе с ключевыми подсистемами Unity (Rigidbody2D, Colliders, Animator, Tilemap).
* Понимании основ геймдизайна и создания играбельного прототипа.

1. **Востребованность подхода:** Разработка игр, особенно в жанрах метроидвании и сложных платформеров, часто требует итеративного подхода и создания прочного фундамента на ранних этапах. Данный проект демонстрирует эффективную стратегию создания ядра и расширения, которая:

* Позволяет сфокусироваться на качестве ключевых систем в ограниченные сроки.
* Обеспечивает предсказуемость и снижает риски при дальнейшей масштабной разработке.
* Является стандартной практикой как в инди-, так и в AAA-разработке для прототипирования и проверки концепций.

1. **Потенциал для развития:** Созданный фундамент имеет явную практическую востребованность для будущего, так как представляет собой готовую базу для превращения в коммерческий или пет-проект в популярном жанре метроидвании, обладающем стабильной аудиторией. Расширяемость систем заложена изначально.

# 4. Общие сведения о проделанной работе

## Система состояний персонажа:

* Используется не классический подход *FSM* (*Finite State Machine*), а более гибкий, который позволяет разделять состояния по классам и управлять каждым из них вручную или автоматически. Разделение достигается за счёт использования перечислений состояний и управления ими через главный менеджер (*PlayerStates*).
* Все переходы между состояниями строго инициируются и регулируются (например смена направления персонажа только при смене направления ввода). В целях оптимизации минимизировано использование *Update()*.
* Пример кода перечисления состояний направления персонажа:

| public **enum** **Direction** {  Left = -1,  Right = 1, } |
| --- |

## Система способностей:

* Все способности наследуются от базового класса *Ability*, который содержит кулдаун, приоритет способности, стандартную анимацию, а также реализует методы активации/деактивации способности и сопутствующих проверок и действий.
* Способности управляются с помощью *PlayerStates*. Они могут прерываться в связи с несоответствием условий или другими способностями с большим приоритетом.

## Механики передвижения:

* Все основные механики (бег, прыжок, рывок, скольжение) реализуются как отдельные способности, унаследованные от базового класса *Ability*. Каждая из них содержит свою логику активации, проверок и взаимодействия с другими способностями.
* Добавлены необходимые платформинговые элементы (время койота, буфер прыжка).
* Контроллер ввода персонажа реализован с использованием новой *Unity Input System*, которая поддерживает гибкое переназначение ввода. Действия игрока из контроллера передаются напрямую способностям.

## Система анимаций:

* Была создана на базе текущей системы состояний. Исходная система *FSM Animator* не используется из-за недостаточной гибкости и плохой оптимизации при большом количестве состояний и условий.

## Система здоровья:

* Главный компонент *BaseHealth*. Содержит здоровье персонажа и методы его изменения.
* *DamageDealer* и *HurtBox*. Используются для нанесения и получения урона соответственно.

## Общая структура и оптимизация:

* Уделено большое внимание структуре, логике и гибкости кода. Заложены возможности для расширения. Все механики и системы (кроме эксклюзивных для игрока, таких как *InputSystem*) поддерживают использование другими персонажами — как в исходной форме, так и в модифицированном виде.
* Особое внимание уделено оптимизации. Везде где это возможно тяжелые методы (такие как *Update*, *GetComponent*, различные поиски) были заменены на более оптимизированные альтернативы. Используются события, кастомные обработчики, кэширование данных. Снижена нагрузка за счет продуманной архитектуры и минимизации избыточных вычислений.

## Создание демонстрационного уровня:

* Для демонстрации существующей кодовой базы были использованы **временные** бесплатные ассеты, включая спрайты и анимации. Основные используемые ресурсы: [Kenney.nl](http://kenney.nl), [opengameart.org](http://opengameart.org), Unity ассет [DOTween](https://assetstore.unity.com/packages/tools/animation/dotween-hotween-v2-27676?srsltid=AfmBOoo7v2tmbgPQw6w3W_14xP_0VazccC5mYCyD2k3vQKIpJOVSjRn_). Демонстрационный уровень несет в себе цель презентовать все существующие механики и системы.

# Выбранные инструменты

Не считая использования временных бесплатных ассетов, разработка осуществлялась исключительно на стандартном инструментарии Unity 6 и языке программирования C#.

# 5. Результаты проекта

* Цели и задачи проекта полностью достигнуты. Разработка выполнена согласно основным поставленным требованием, а именно: продуманная гибкая модульная архитектура, выполненная в соответствии с паттернами разработки; возможность расширения (как горизонтального, так и вертикального); высокая оптимизация.
* Создан компактный уровень, с использованием временных бесплатных ассетов, наглядно демонстрирующий все основные механики и игровые системы.
* Успешно создано прочное ядро системы, обеспечивающее:
  + стабильную основу для дальнейшего развития проекта
  + возможность трансформации в полноценную метроидванию.

# Приложение 1

https://github.com/brizrak/Fall-of-the-Sunset-Submission.git

# Приложение 2

https://drive.google.com/drive/folders/1cpsTP7Kv1lC1rWpJP\_Yq-0hkfK6KSffJ?usp=sharing