

## Создание визуального эффекта на игровом движке Godot (часть 2)

*Черкашин Александр Михайлович*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### Аннотация

В данной статье описан процесс создания визуального эффекта (VFX) на основе системы частиц на игровом движке Godot. В работе использовалась система частиц и текстура написанная на языке шейдера в конструкторе игр на движке Godot. В результате работы был создан визуальный эффект системы частиц на игровом движке Godot.

**Ключевые слова:** Godot, VFX, Particles system.

## Creating a visual effect using the Godot game engine (part 2)

*Cherkashin Alexander Mihailovich*

*Sholom-Aleichem Priamursky State University*

*student*

### Abstract

This article describes the process of creating a visual effect (VFX) based on a particle system using the Godot game engine. The work used a particle system and texture written in the shader language in the game designer on the Godot engine. As a result of the work, a visual effect was created - a particle system using the Godot game engine.

**Keywords:** Godot, VFX, Particles system.

## 1 Введение

### 1.1. Актуальность исследования

Актуальность исследования заключается в том что по созданию визуальных эффектов (VFX) в Godot Engine весьма актуальны для разработчиков игр и энтузиастов, которые заинтересованы в использовании возможностей движка для визуального улучшения своих игр. Godot Engine — это бесплатный игровой движок с открытым исходным кодом, который предоставляет широкий спектр инструментов и функций для разработки 2D- и 3D-игр. Godot Engine предлагает различные функции и ресурсы для создания визуальных эффектов. К ним относятся GDScript и C++, Godot Assistant, Godot Documentation, Godot Assets Marketplace и Godot Visual Effects. Исследования по созданию визуальных эффектов в Godot Engine важны по нескольким причинам. К ним относятся улучшение визуальных эффектов игры, улучшение пользовательского опыта, поддержание конкурентоспособности и сотрудничество с сообществом. В заключение отметим, что исследования по

созданию визуальных эффектов в Godot Engine весьма актуальны для разработчиков игр и энтузиастов, которые хотят использовать возможности движка для визуального улучшения своих игр. Godot Engine предоставляет различные ресурсы, включая документацию, ресурсы и поддержку сообщества, чтобы помочь разработчикам создавать впечатляющие визуальные эффекты для своих игр.

### *1.2. Цель исследования*

Целью работы создания визуальный эффект на основе система частиц на игровом движок Godot.

### *1.3. Обзор исследований*

Г. К. Ульманн и др. провели предварительное исследование по восстановлению архитектуры игрового движка в своей статье под названием «Исследовательский подход к восстановлению архитектуры игрового движка». Исследование сосредоточено на восстановлении архитектурного проектирования игровых движков с целью дать представление о структуре и организации этих сложных программных систем.

Авторы начинают с подчеркивания важности понимания архитектуры игрового движка для различных целей, таких как разработка игр, обслуживание программного обеспечения и оптимизация. Они подчеркивают, что процесс восстановления архитектуры может помочь разработчикам глубже понять основные принципы проектирования, что облегчит будущие улучшения и модификации.

Для достижения цели исследования Г. К. Ульмана. предложить исследовательский подход, сочетающий методы обратного проектирования с качественным анализом. Они применяют свой подход к нескольким игровым движкам с открытым исходным кодом, анализируя их исходный код, документацию и связанные с ними артефакты.

В исследовании представлены результаты процесса восстановления архитектуры, включая определение ключевых архитектурных компонентов, их взаимоотношений и их обязанностей в игровых движках. Авторы описывают восстановленную архитектуру, используя визуальные представления, такие как диаграммы и графики, чтобы обеспечить четкое представление о выявленных компонентах и их взаимодействии.

Кроме того, Ullmann G.C. et al. обсудить проблемы, возникающие в процессе восстановления архитектуры, такие как отсутствие полной документации и присущая игровым движкам сложность. Они дают представление о потенциальных стратегиях преодоления этих проблем и повышения эффективности и результативности будущих усилий по восстановлению архитектуры.

Статья завершается обсуждением последствий результатов исследования и их потенциального влияния на разработку и обслуживание игровых движков. Авторы подчеркивают ценность своего исследовательского подхода в раскрытии базовой архитектуры игровых движков, что позволяет

разработчикам принимать обоснованные решения и оптимизировать свои программные системы.

Таким образом, исследовательская статья Г. К. Ульмана и др. «Исследовательский подход к восстановлению архитектуры игрового движка» представляет собой углубленное исследование восстановления архитектуры игрового движка. Благодаря своему исследовательскому подходу авторы предоставляют ценную информацию о структуре и организации игровых движков, облегчая будущие усилия по разработке, обслуживанию и оптимизации в области разработки игр [1].

Г. К. Ульманн и др. провели исследование и описали на своей статье под названием «Визуализация шаблонов взаимодействия подсистем игрового движка» в этой статье авторы используют подход к восстановлению архитектуры для создания архитектурных моделей для 10 игровых движков с открытым исходным кодом. Г. К. Ульманн использует эти модели, чтобы ответить на следующие вопросы: Какие подсистемы чаще связаны друг с другом? Имеют ли игровые движки общие шаблоны взаимодействия подсистем? Г. К. Ульманн и др. заметили, что низкоуровневый рендерер, уровень независимости платформы и диспетчер ресурсов часто связаны с игровым движком Core. Выявляя наиболее часто встречающиеся шаблоны сопряжения, авторы работ описывает возникающую архитектуру игрового движка и обсуждаем, как ее могут использовать специалисты-практики для улучшения понимания системы и ее удобства обслуживания [2].

Ю. Чжан представил проект в этой диссертации направлен на разработку голографического интерфейса для устройства Rod. Процесс проектирования включает в себя анализ повествования фильма, исследование схожих поджанров научно-фантастических фильмов, технологию голографического отображения и реализацию визуальных эффектов интерфейса для устройства rod. В качестве основного инструмента реализации интерфейса использовались граф визуальных эффектов и система анимации в игровом движке Unity. Конечный продукт представляет собой цифровой прототип с интерактивным трехмерным интерфейсом модуля. Кроме того, дизайн прототипа является частью подготовки к фильму, призванной передать слова сценария в визуальные эффекты и взаимодействие. Описывая процесс проектирования, автор предлагает возможный метод организации предварительного исследования научно-фантастического фильма и использования игрового движка Unity в качестве платформы для создания визуальных эффектов к фильму [3].

## 2. Рабочий процесс

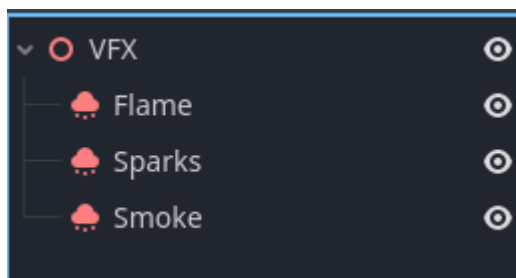


Рисунок 1. Дерево объектов визуального эффекта на сцене

Мы открыли проект «vfx\_godot» и открыли сцену «res://VFX/Main.tscn».

В «Flame», свойства Process Material созданный тип ParticleProcessMaterial. Мы задали следующие значение

Spawn → Position → Emission Shape задали в Sphere. Это форма источник испускание частиц, мы выбрали сфера.

Spawn → Position → Emission Sphere Radius задали 1. Это радиус сферы.

Spawn → Velocity → Spread задали в 180. Это начальное направление каждой частицы.

Spawn → Velocity → Initial Velocity Min и Initial Velocity Max задали 4 и 6. Это минимальная и максимальная начальная величина скорости для каждой частицы. Направление исходит из направления и распространения [4].

Accelerations → Gravity задали x: 0, y: 0, z: 0. Это ускоренное свободное падение для каждой частицы.

Display → Scale → Scale Min задали 0.25.

Display → Scale → Scale Max задали 8.

Это минимальный и максимальный начальный масштаб примененный к каждой частице.



Рисунок 2. Заданный Curve для свойства Scale Curve

Display → Scale → Scale Curve задали от 1 до 0 (рис 2). Это кривая шкала продолжительность. По вертикальный ось графике — значение Scale, а горизонтальное ось — расстояние частиц от центра испускание (рис 2).

Display → Color Curve → Color задали Raw (2, 1, 0, 1). Это цвет

Display → Color Curve → Color Ramp задали количество градиентов 5, цвета #000000, #ffd000, #ff8800, #f50000, #000000, а расположение 0, 0.1, 0.4, 0.6, 1. Это цвет будет меняться зависимости от времени жизни частиц (рис 3). А затем мы задали width (ширина) на 2048.

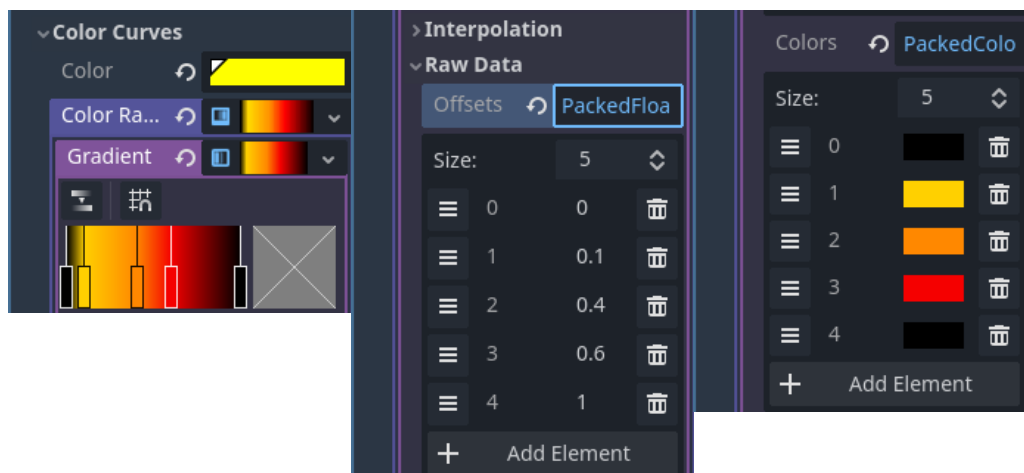


Рисунок 3. Заданный свойство Color Ramp

Display → Hue Variation → Hue Variation Max задали 0.1. Это максимальный значение по цветовой модели Hue.

В «Sparks», свойства Process Material созданный тип ParticleProcessMaterial. Мы скопировали из «Flame» и изменили следующие свойства.

Particle Flags → Align Y задали на On.

Spawn → Velocity → Initial Velocity Min и Initial Velocity Max задали на 10 и 15.

Accelerations → Damping → Damping Max задали 10. Это максимальная скорость, с которой частицы теряют скорость. Например, значение 10 означает, что скорость частицы изменится со 10 до 0 за 1 секунду.

Display → Scale → Scale Min и Scale Max задали 0.5 и 2.

Display → Scale → Scale Curve создали тип CurveXYZTexture.

По кривую осью X мы задали от 0.5 до 0. А осью Y задали от 5 до 0. Z задали от 5 до 5 (рис 4).

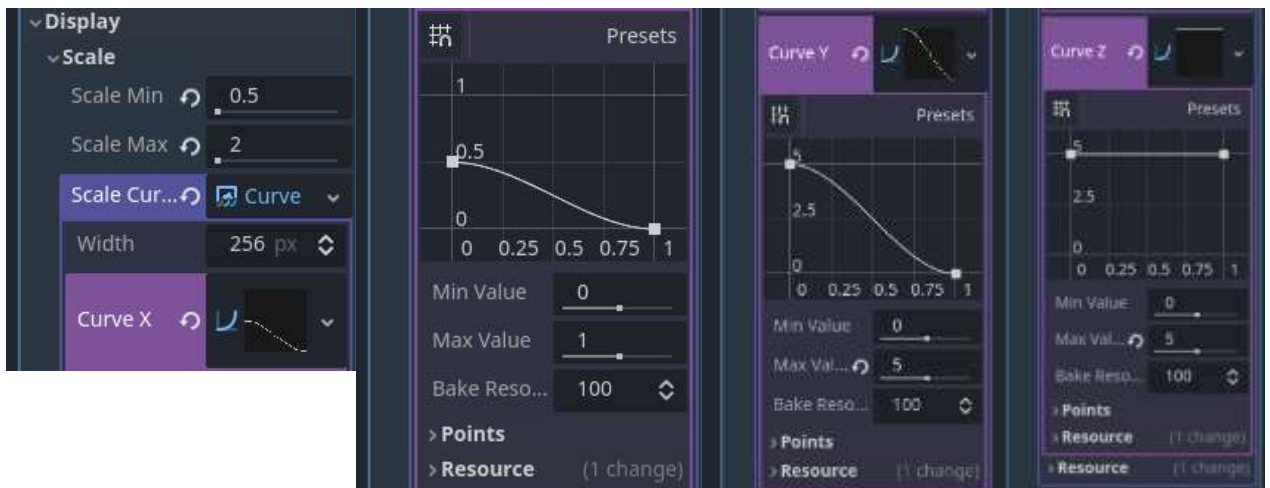


Рисунок 4. Заданное свойство Scale Curve

В «Smoke», свойства Process Material созданный тип ParticleProcessMaterial. Мы скопировали из «Flame» и изменили следующие свойства.

Spawn → Position → Emission Sphere Radius задали 0.2.

Spawn → Velocity → Initial Velocity Min и Initial Velocity Max задали 10 и 15.

Accelerations → Damping → Damping Max задали 10.

Display → Scale → Scale Min и Scale Max задали 6 и 8.

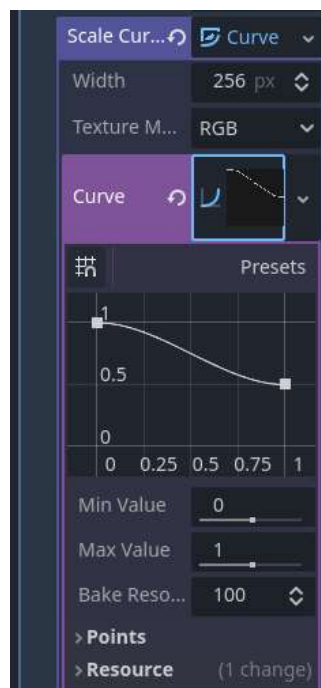


Рисунок 5. Заданный Curve для свойство Scale Curve

Display → Scale → Scale Curve задали от 1 до 0.5 (рис 5).

Display → Color Curves → Color задали #1a1a1a.

Display → Color Curves → Color Ramp задали количество градиентов 2, цвета #ffffff, #ffffff00, а расположение 0, 1 (рис 6).

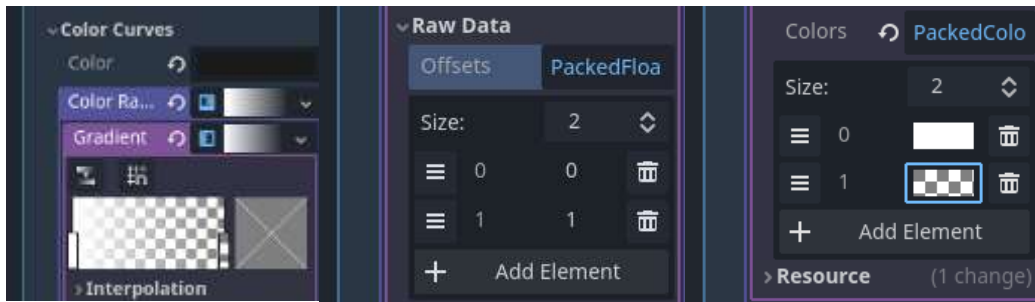


Рисунок 6. Заданный свойство Color Ramp.

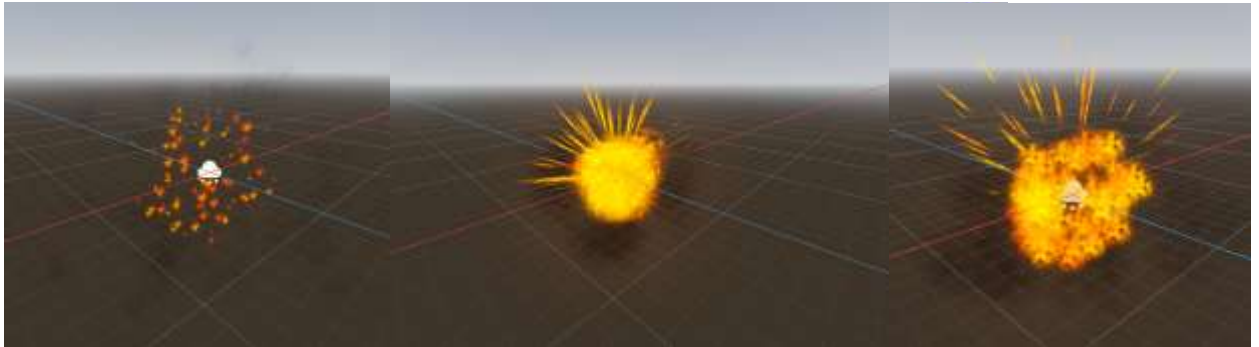


Рисунок 7. Готовый визуальный эффект

### 3 Выводы

В данной статье была создана визуальный эффект VFX. В результате работы было создано 3 объекта система частиц который испускают частицы и создают визуальный эффекты и так же прикреплен шейдеры для на рисование текстуры плоскости частиц.

### Библиографический список

1. Ullmann G. C. et al. An Exploratory Approach for Game Engine Architecture Recovery //arXiv preprint arXiv:2303.02429. 2023.
2. Ullmann G. C. et al. Visualising Game Engine Subsystem Coupling //arXiv preprint arXiv:2309.06329. 2023.
3. Zhang Y. et al. Using the Unity game engine as a platform for prototyping cinematic visual effects. 2020.
4. ParticleProcessMaterial &mdash; Godot Engine (stable) documentation in English [GitHub](https://docs.godotengine.org/en/stable/classes/class_particleprocessmaterial.html) URL: [https://docs.godotengine.org/en/stable/classes/class\\_particleprocessmaterial.htm](https://docs.godotengine.org/en/stable/classes/class_particleprocessmaterial.html) l (дата обращения: 2024-01-28).