

Создание счётчика захватов объекта в Unreal Engine

Андриенко Иван Сергеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Пащенко Екатерина Александровна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье представлен процесс разработки системы подсчёта количества захватов объекта в виртуальной реальности (VR) с использованием игровых инструментов Unreal Engine. Описан порядок добавления текстового компонента, создание переменной для хранения количества взаимодействий, а также настройка логики подсчёта на основе событий GrabComponent. В работе рассматриваются этапы конфигурации Blueprint-графа, включая инициализацию отображаемого значения и обновление счётчика при каждом взаимодействии пользователя с объектом. Приведён пример практической реализации и тестирования механики в VR-среде.

Ключевые слова: виртуальная реальность, Unreal Engine, Blueprints, VR-взаимодействие, счётчик, захват объекта, GrabComponent.

Creating an object capture counter in the Unreal Engine

Andrienko Ivan Sergeevich

Sholom-Aleichem Priamursky State University

Student

Paschenko Ekaterina Alexandrovna

Sholom-Aleichem Priamursky State University

Student

Abstract

This article presents the process of developing a system for calculating the number of object captures in virtual reality (VR) using the game tools Unreal Engine. The procedure for adding a text component, creating a variable to store the number of interactions, and configuring the counting logic based on GrabComponent events is described. The paper discusses the configuration stages of the Blueprint graph, including initialization of the displayed value and updating the counter with each user interaction with the object. An example of practical implementation and testing of mechanics in a VR environment is given.

Keywords: virtual reality, Unreal Engine, Blueprints, VR interaction, counter, object capture, GrabComponent.

1 Введение

1.1 Актуальность

Одним из ключевых аспектов разработки VR-приложений является обеспечение интерактивности объектов, позволяющей пользователю взаимодействовать с виртуальной средой максимально естественно. Подсчёт количества захватов объекта является важным инструментом для анализа поведения пользователей, отладки механик взаимодействия и создания игровых систем, реагирующих на действия игрока. Реализация подобного счётчика позволяет значительно расширить функциональные возможности VR-проекта и облегчает проведение тестирования интерфейсов и интерактивных элементов.

1.2 Обзор исследований

Разработка приложений виртуальной реальности (VR) с использованием современных игровых движков активно исследуется в научной литературе. В работе Д.А. Шеверева и И.Н. Козловой «Разработка приложения "VR.Lab" на основе игрового движка Unreal Engine 4 и технологий виртуальной реальности» рассматривается практический опыт создания VR-приложения, включая особенности работы с Unreal Engine 4 и интеграцию VR-технологий [1]. А.С. Фокин в своей статье проводит сравнительный анализ двух популярных движков, оценивая их возможности, производительность и удобство разработки VR-приложений [2]. В исследовании Д.А. Мирошникова «Создание 3D-моделей и перенос в виртуальную реальность в приложениях Blender и Unreal Engine при разработке VR-деловой игры по направлению "Строительство"» рассматривается процесс создания контента для VR с использованием Blender и его последующей интеграции в Unreal Engine [3]. А.А. Смирнов, А.А. Бабкин и В.В. Галеев в работе «Unity или Unreal Engine: что лучше для VR-разработки?» анализируют преимущества и недостатки обоих движков, уделяя внимание их применению в образовательных и коммерческих VR-проектах [4]. Е.Ю. Кузьмина, М.В. Лавлинский и А.И. Баландин описывают опыт разработки образовательного VR-приложения с использованием современных инструментов, включая Unreal Engine [5].

1.3 Цель исследования

Целью настоящей работы является создание системы подсчёта количества захватов объекта, основанной на механизме GrabComponent в Unreal Engine. Исследование направлено на разработку компактной и удобной в использовании Blueprint-логики, обеспечивающей отображение текущего количества взаимодействий в режиме реального времени.

2 Материалы и методы

Для реализации поставленной задачи использовался стандартный VR-шаблон Unreal Engine, включающий необходимые компоненты для работы с виртуальной реальностью. В каталоге VRTemplates, в папке Blueprints, был выбран объект Grabbable_SmallCube.uasset в качестве основы для разработки.

3 Результаты и обсуждения

В каталоге VRTemplates в папке BluePrints находится куб Grabbable_SmallCube.uasset. Редактируем и переходим в окно Components. В окне компонентов к объекту был добавлен компонент TextRender, предназначенный для отображения текста над кубом. Далее в разделе Variables была создана переменная Integer под названием TouchCounter, которая служила для хранения количества захватов объекта пользователем (рис. 1). Для визуализации числового значения счётчика в окно Components был добавлен компонент TextRender. Данный компонент предназначен для отображения текстовых данных непосредственно в трёхмерной сцене. В отличие от виджетов, TextRender является частью 3D-мира и способен сохранять читаемость под любым углом, что делает его удобным средством отображения служебной или игровой информации. После добавления компонента была создана переменная TouchCounter типа Integer, которая предназначена для хранения количества зафиксированных захватов.

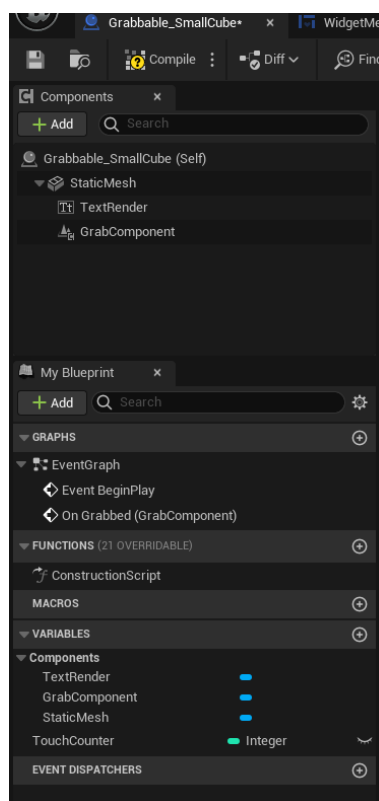


Рис. 1. Создание переменной

Следующим этапом стало создание начальной логики отображения данных. В Event Graph был использован нод Event BeginPlay, который

вызывается сразу после появления объекта на сцене. Через нод Set Text компоненту TextRender задавалось начальное значение текста. Для преобразования числового значения переменной в строковый формат использовался нод ToText (Integer). Подобное преобразование необходимо, поскольку большинство визуальных компонентов Unreal Engine работает со строками, а не с числовыми типами данных. Итоговая схема обеспечивала появление над кубом отображения «0» при запуске VR-проекта (рис. 2).

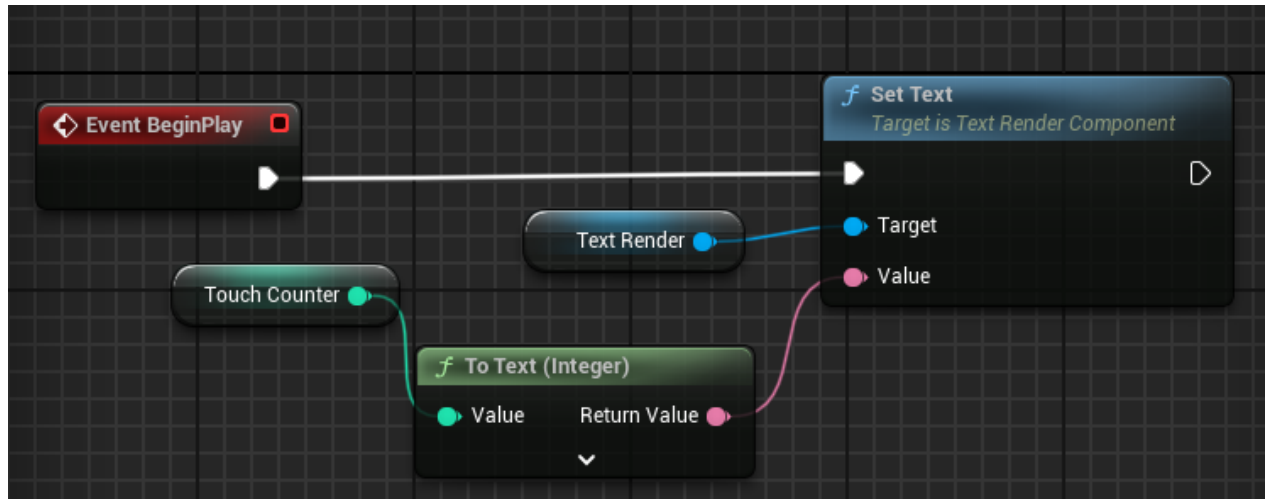


Рис. 2. Настройка нода Event BeginPlay

После настройки начального состояния была реализована основная логика подсчёта. В компоненте GrabComponent было найдено событие On Grabbed, вызываемое каждый раз, когда пользователь берёт объект в VR-контроллер. Событийная модель UE основана на принципе реактивного программирования: вместо постоянной проверки состояния объекта система генерирует события, которые автоматически вызывают связанные узлы логики. Это снижает нагрузку на процессор и обеспечивает точность фиксации взаимодействий.

В Event Graph событие On Grabbed инициировало цепочку действий, начинающуюся с нода Print Text, который использовался для отладки и подтверждения факта активации события. Далее происходило автоматическое увеличение TouchCounter через нод Increment TouchCounter, увеличивающий значение переменной на единицу без необходимости дополнительных вычислений. После обновления переменной текстовое значение снова задавалось компоненту TextRender, что обеспечивало мгновенное отражение изменения счётчика над объектом (рис. 3).

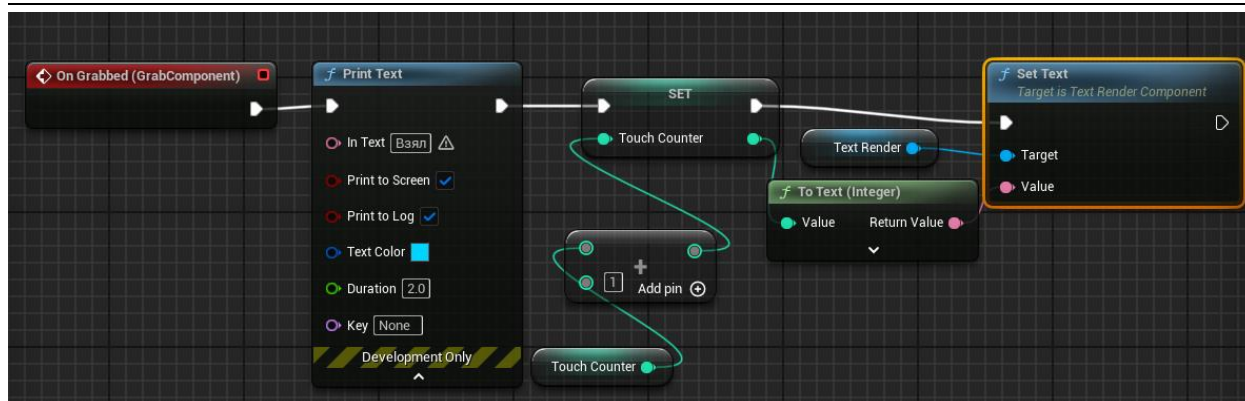


Рис. 3. Итоговая настройка узлов

На заключительном этапе была проведена проверка работоспособности созданной системы. В VR-среде объект корректно реагировал на каждое взаимодействие, а значение счётчика обновлялось при каждом захвате. Текст оставался визуально стабильным, хорошо читаемым и не зависел от ориентации пользователя в пространстве, что подтверждает целесообразность применения компонента TextRender для подобных задач. На изображении представлены результаты тестирования (рис. 4).

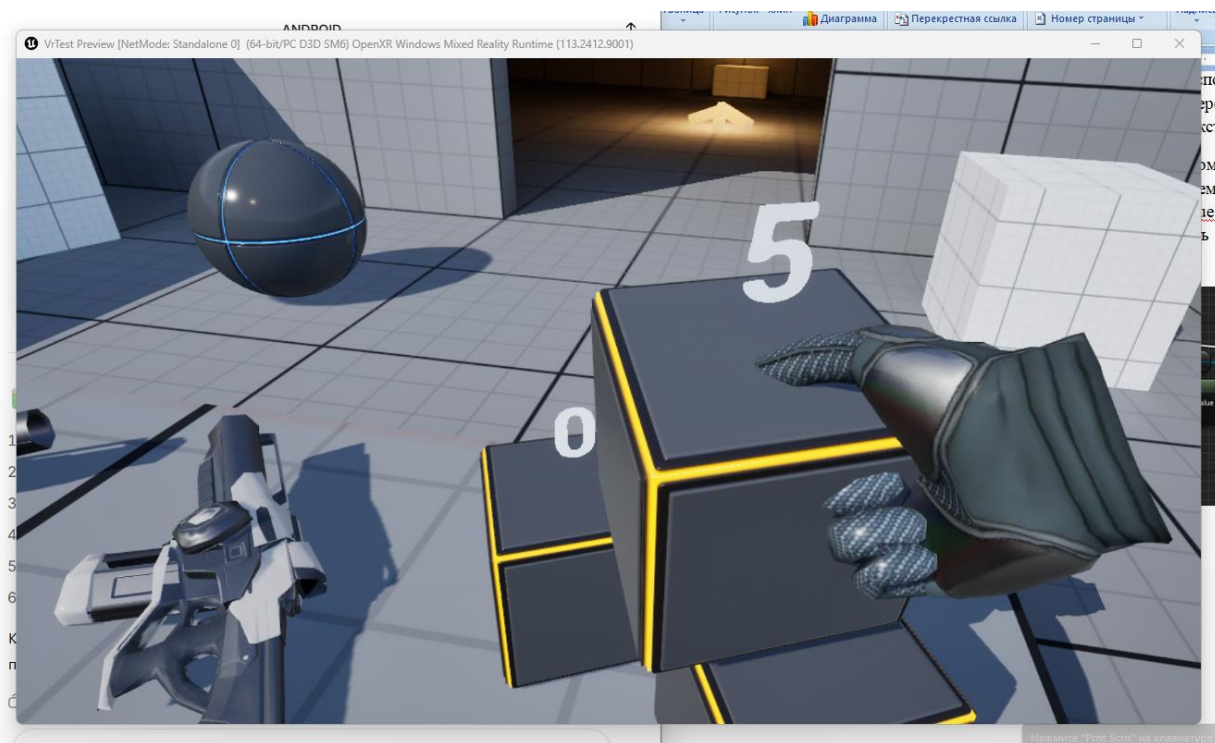


Рис. 4. Тестирование куба

4 Выводы

В ходе работы была разработана система подсчёта захватов объекта в VR-среде Unreal Engine. Настроенное взаимодействие через GrabComponent и обновление текстовой информации с помощью TextRender обеспечили стабильную фиксацию событий и корректное отображение результата в трёхмерном пространстве. Проведённое тестирование подтвердило

работоспособность механики: счётчик обновляется своевременно, а отображение остаётся точным и удобным для пользователя. Полученное решение может служить основой для расширенных аналитических и игровых систем в VR-приложениях.

Библиографический список

1. Шеверев Д.А., Козлова И.Н. Разработка приложения "VR.Lab" на основе игрового движка Unreal Engine 4 и технологий виртуальной реальности // В сборнике: Перспективные информационные технологии (ПИТ 2017). Труды Международной научно-технической конференции. 2017. С. 1123–1126.
2. Фокин А.С. Разработка новейших VR-приложений на Unity и Unreal Engine – что лучше? // Студенческий вестник. 2024. № 20-19 (306). С. 72–73.
3. Мирошников Д.А. Создание 3D-моделей и перенос в виртуальную реальность в приложениях Blender и Unreal Engine при разработке VR-деловой игры по направлению "Строительство" // В сборнике: IX Международный студенческий строительный форум – 2024. Сборник докладов форума. Белгород, 2024. С. 135–141.
4. Смирнов А.А., Бабкин А.А., Галеев В.В. Unity или Unreal Engine: что лучше для VR-разработки? // Научный Лидер. 2024. № 34 (184). С. 7–8.
5. Кузьмина Е.Ю., Лавлинский М.В., Баландин А.И. Создание VR-приложения "Класс информационно-коммуникационных технологий лицея ИГУ" // В сборнике: Актуальные задачи прикладной дискретной математики (Дискретный анализ и информатика). Сборник научных трудов. Иркутск, 2024. С. 54–60.