

Создание визуальной новеллы в стиле аниме на Unity

Малик Александр Игоревич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье рассматривается процесс создания визуальной новеллы в Unity, включая разработку двух основных слайдов. Статья описывает основные этапы создания визуальной новеллы, от проектирования до реализации в Unity, и предоставляет практические советы и рекомендации для начинающих разработчиков. Особое внимание уделяется выбору инструментов и методов, которые обеспечивают создание качественного и увлекательного контента.

Ключевые слова: Визуальная новелла, Unity, Разработка игр, Проектирование интерфейса, Сценарий, Интерактивное повествование.

Creating an anime-style visual novel on Unity

Malik Aleksandr Igorevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses the process of creating a visual novel in Unity, including the development of two main slides. The article describes the main stages of creating a visual novel, from design to implementation in Unity, and provides practical tips and recommendations for novice developers. Special attention is paid to the choice of tools and methods that ensure the creation of high-quality and exciting content.

Keywords: Visual novel, Unity, Game development, Interface design, Scenario, Interactive storytelling

1 Введение

1.1 Актуальность

С развитием технологий и растущим интересом к интерактивным медиа, жанр визуальных новелл набирает популярность среди разработчиков и пользователей. Unity, благодаря своей гибкости и широким возможностям, стал одним из наиболее предпочтительных инструментов для создания визуальных новелл. Тем не менее, существует нехватка детализированных руководств и практических пособий, которые могли бы помочь начинающим разработчикам освоить процесс создания визуальных новелл в Unity. Данная статья заполняет этот пробел, предоставляя пошаговое руководство по

разработке визуальной новеллы, включая проектирование и реализацию двух основных слайдов.

1.2 Обзор исследований

На 15-й Международной конференции по основам цифровых игр (FDG '20) Д. Каминге, Э. Ф. Мелцер и Э. Карстенсдоттир представили исследование, посвященное стратегиям обучения в визуальных новеллах. Они анализируют, как интерактивные нарративы способствуют обучению и каким образом дизайн таких игр поддерживает учебный процесс, делая его более эффективным и увлекательным [1]. Цезарь Д. Галинган и Р. Б. Кагадас из Национального университета-Дасмаринас разработали систему диалогов между игроком и неигровым персонажем (NPC) в визуальной новелле с использованием модели BERT (Bidirectional Encoder Representations from Transformers) для распознавания намерений. Их исследование фокусируется на улучшении интерактивности и реалистичности взаимодействий в игре, что способствует более глубокой вовлеченности игроков [2]. Ф. Дженни и А. Сурядибрата создали мобильную визуальную новеллу "7-Day Math", направленную на обучение математике, в частности, интегралам. Их работа нацелена на преодоление негативного восприятия математики в обществе, которое часто возникает из-за менее интерактивных методов преподавания. Инновационный подход в их игре помогает сделать обучение математике более доступным и интересным [3]. М. Б. Гарсиа из Института технологий Филиппинского университета FEU исследует использование цифровых образовательных игр для улучшения знаний в дошкольном образовании. Его проект "Kinder Learns", основанный на учебной программе K-12 для детских садов, разработанной Департаментом образования, представляет собой образовательную визуальную новеллу, которая способствует раннему обучению детей, делая его веселым и познавательным [4]. М. Д. Гриффин II в своей диссертации "The Visual Novel: Fictional Space and Print After 1900" проводит глубокий анализ того, как роман адаптировался к визуальным медиа, включая фильмы, искусство и цифровые технологии, в ответ на распространение массовых медиа. Гриффин исследует, как визуальные новеллы интегрируют визуальные элементы в литературную форму, изменяя традиционные представления о романе и печатной продукции, что открывает новые горизонты для восприятия литературных произведений в цифровую эпоху [5].

1.3 Цель исследования

Цель данной статьи – описать процесс создания визуальной новеллы с двумя слайдами в Unity, предоставляя подробные инструкции и советы для разработчиков. Исследование нацелено на демонстрацию ключевых этапов разработки, начиная от концептуализации и заканчивая финальной реализацией, а также на освещение использования различных инструментов и методов, которые помогают создать качественную и захватывающую визуальную новеллу.

2 Материалы и методы

2.1. Инструменты

В качестве игрового движка для разработки был взят Unity, так как он прост в освоении с интуитивно понятным интерфейсом и возможностью написания скриптов на языке C#, также вместе с этим будет использоваться Adobe Photoshop для подготовки и базового редактирования ассетов. Для редактирования скриптов можно использовать любой текстовый редактор – в работе используется notepad++.

2.2. Подготовка

Первым делом необходимо создать 2D (Built-In Render Pipeline) проект в предустановленном Unity и импортировать все ассеты, если они подготовлены заранее. В данном случае были заранее подготовлены: шрифт, задний фон главного меню, кнопки смены сцен, панель для диалогов и два пустых файла для C#-скриптов.

2.3. Создание основы сцены

В панели иерархии проекта создадим пустое полотно (UI => Canvas) и настроим главную камеру(Main Camera), фокусируя ее на Canvas, для этого установим параметр компонента Canvas - Render Mode в **Screen Space – Camera** и перенесем объект камеры на параметр компонента Render Camera. В компоненте Canvas Scaler установим эталонное разрешение – 1920x1080 (Рис. 1).

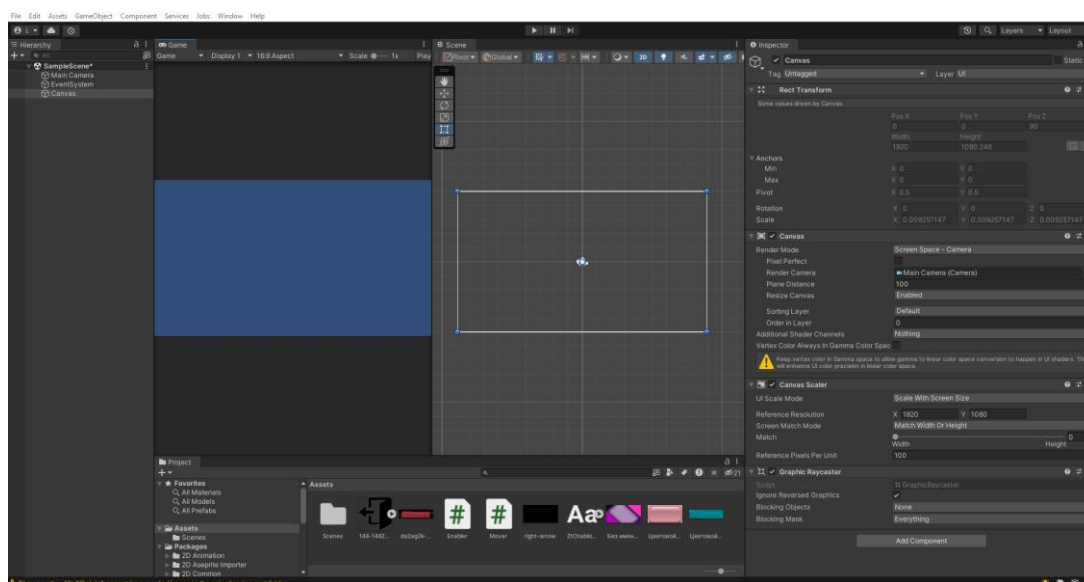


Рисунок 1 – Общий вид на проект и параметры объекта Canvas

2.4 Создание главного меню

Добавим главное меню, для этого создадим объект MainMenu, а внутри него 3 объекта: Raw Image и два Button. Первый объект будет служить задним фоном главного меню, а кнопки для запуска игры и выхода из приложения (Рис. 2).

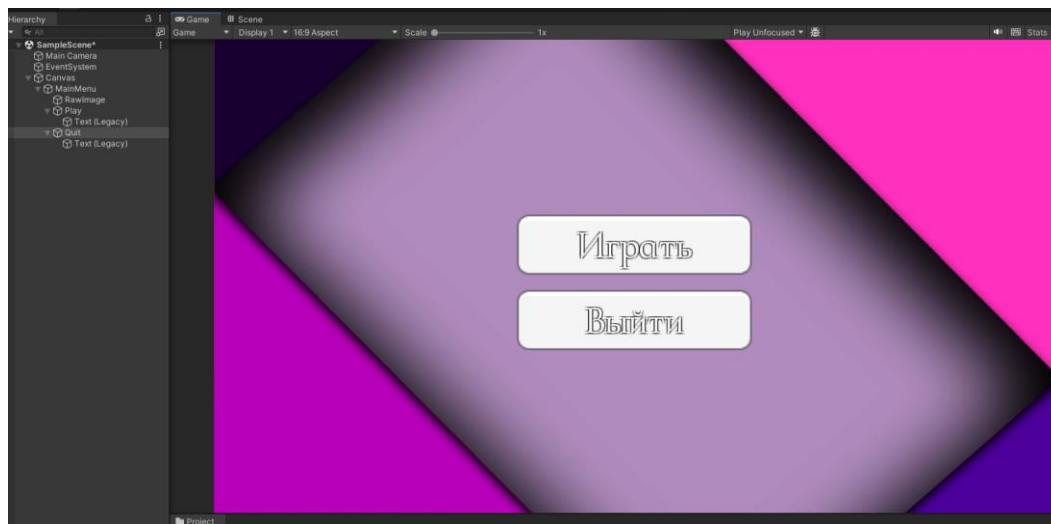


Рисунок 2 – Главное меню игры

Добавим функциональность кнопкам, для этого напомним код в ранее созданный файл Controller.cs (Рис. 3).

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Controller : MonoBehaviour
{
    public GameObject[] Enable;
    public GameObject[] Disable;

    public void ChangeScene()
    {
        foreach(GameObject x in Enable)
            x.SetActive(true);

        foreach(GameObject x in Disable)
            x.SetActive(false);
    }

    public void QuitGame()
    {
        Application.Quit();
    }
}
```

Рисунок 3 – Функции выхода из приложения и переключения активностей объектов

Далее, необходимо повесить этот скрипт на событие нажатия **OnClick()** у каждой кнопки. Для этого вешаем скрипт на каждую кнопку перетаскивая его на объекты кнопок, следом за этим переносим объекты кнопок на собственные события **OnClick()** и выбираем нужные функции, прописанные ранее в скрипте для каждой кнопки. Для кнопки «Играть» ставим функцию **Controller.ChangeScene()**, а для кнопки «Выйти» - **Controller.QuitGame()** (Рис. 4).

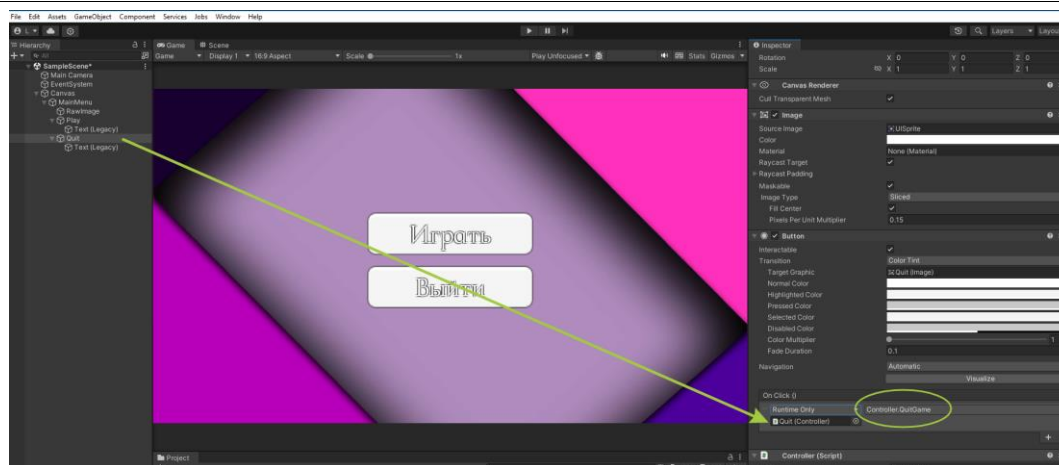


Рисунок 4 – Добавление скрипта на триггер нажатия кнопки

Теперь кнопки функционируют и полностью готовы, за исключением того, что кнопка «Играть» – визуально ничего не делает. Все потому, помимо главного меню нужно добавить больше сцен, чтобы можно было переключаться между ними.

2.5 Добавление интерактивных сцен

Данный этап содержит основу всей игры и будет зависеть только от воображения автора. Для примера в работе будут созданы две сцены. В общем виде одна сцена содержит в себе объекты:

- Кнопка выхода в главное меню
- Кнопка смены сцены
- Задний фон, связанный с текстом героя
- Панель текста героя
 - Имя героя
 - Текст героя

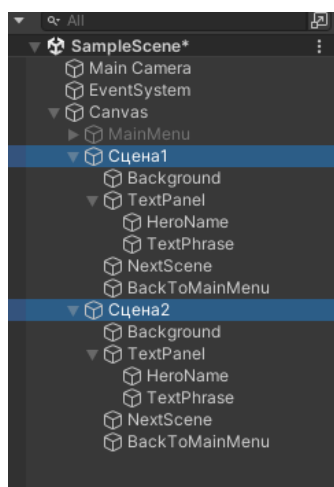


Рисунок 5 – Иерархия объектов проекта

Расставим все объекты в нужные места для понимания общей картины (Рис. 6).

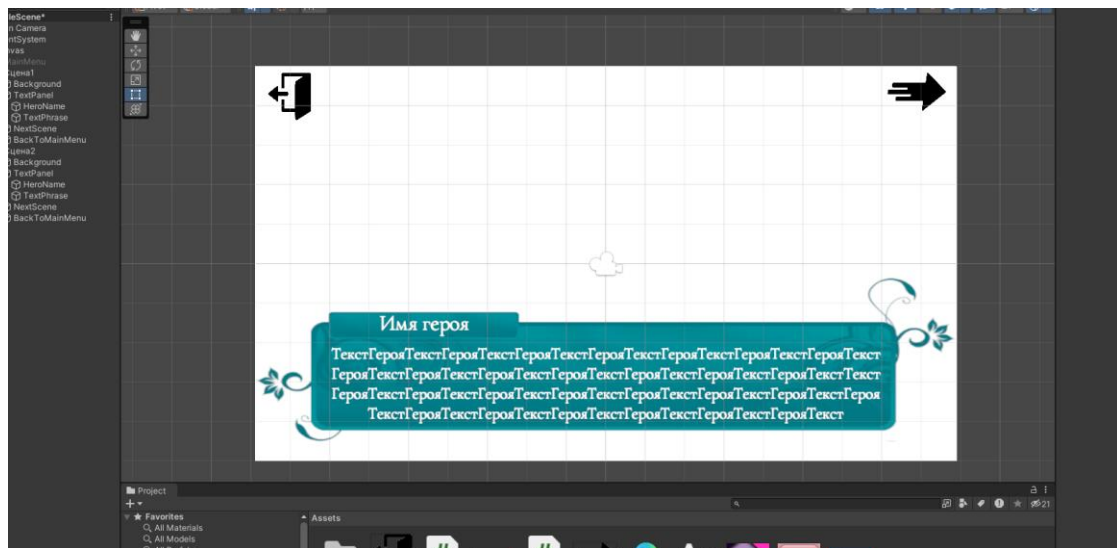


Рисунок 6 – Оформленная сцена

Следующим шагом - сгенерируем задний фон сцен из обычных фотографий, для этого воспользуемся случайным бесплатным сервисом для преобразования фотографий в аниме-стиль – Drawever.

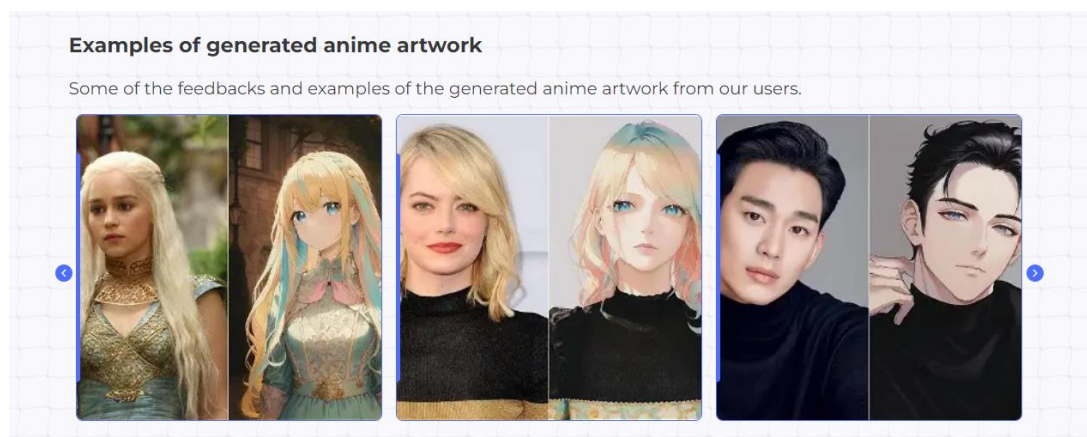
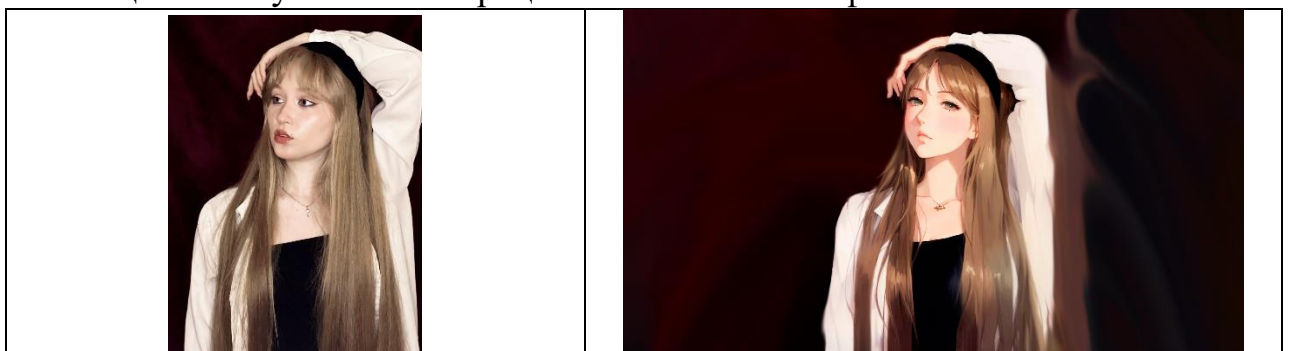
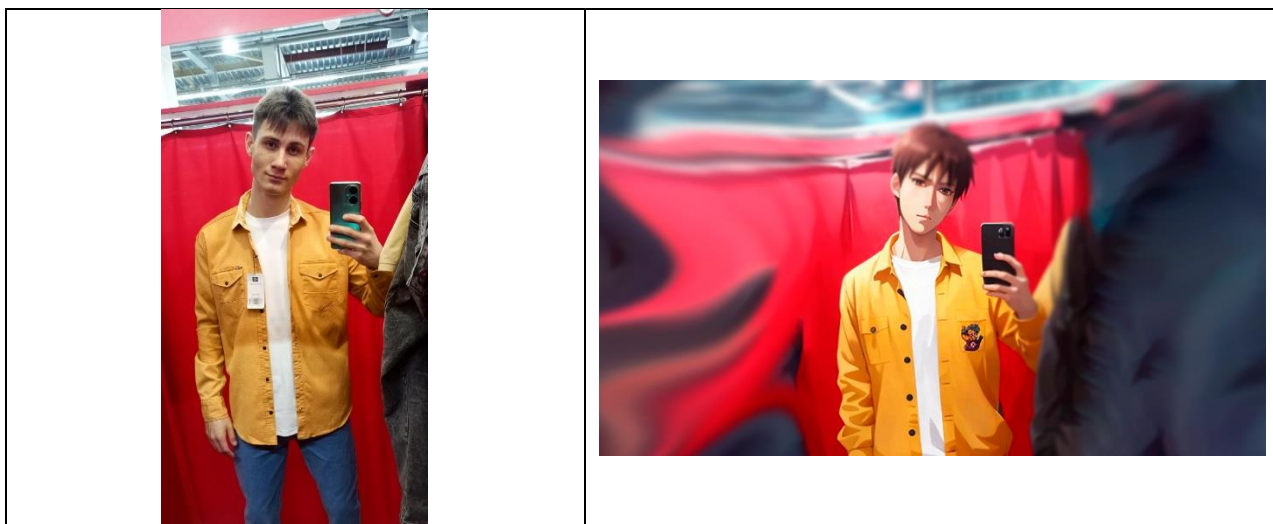


Рисунок 7 – Пример генераций с главной страницы сайта

После генерации необходимо также обработать изображение для дальнейшего использования. Результат генерации можно увидеть в таблице 1.

Таблица 1 – Результаты генераций из собственных фото





Следом, импортируем полученные изображения в проект и устанавливаем в качестве текстуры к нужным сценам, улучшаем позиционирование объектов и меняем текст (Рис. 8-9).

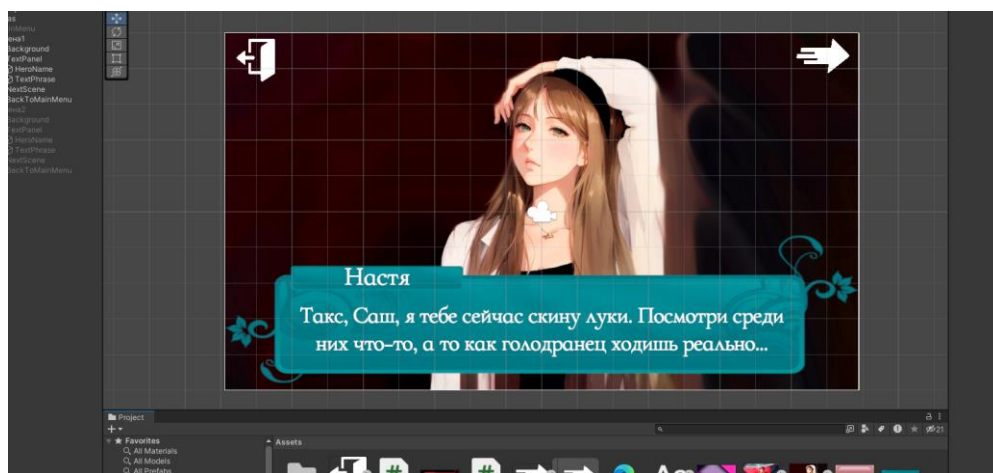


Рисунок 8 – Первая сцена

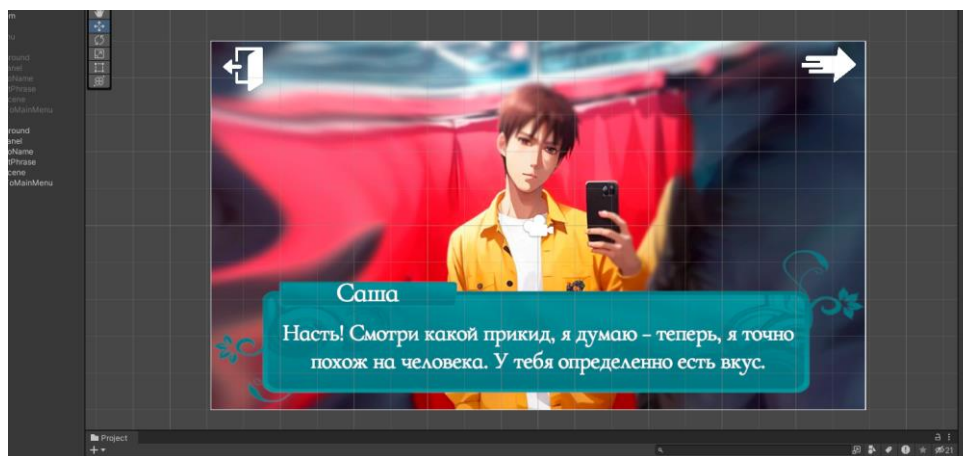


Рисунок 9 – Вторая сцена

Теперь, займемся добавлением функциональности для кнопок сцен и кнопки «Играть». Поскольку теперь присутствуют сцены, с которыми можно

оперировать – перенесем первую сцену в скрипт контроллера кнопки «Играть» в параметр **Enable**, а главное меню в параметр **Disable** для имитации перехода между сценами, после этого можно осуществлять переход в первую сцену из главного меню (Рис. 10).

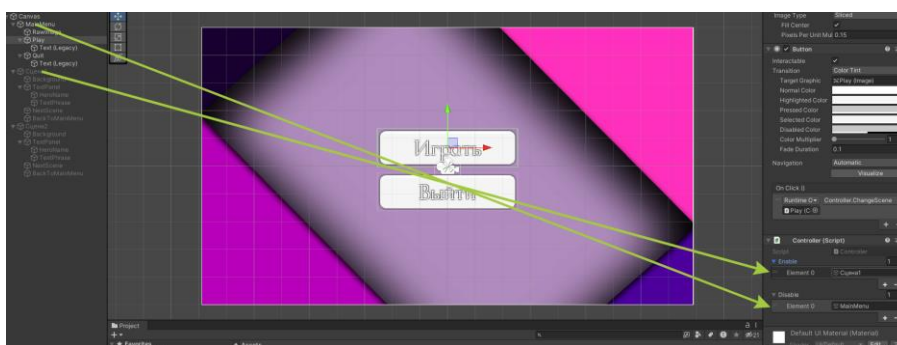


Рисунок 10 – Определение массивов для переключения активности объектов

Добавим к кнопкам выхода в меню и смены сцены, для этого заполним параметры кнопки выхода в меню соответствующими объектами, для выполнения действия выхода в главное меню, аналогично делаем с кнопкой перехода между сценами (Рис. 11-12).

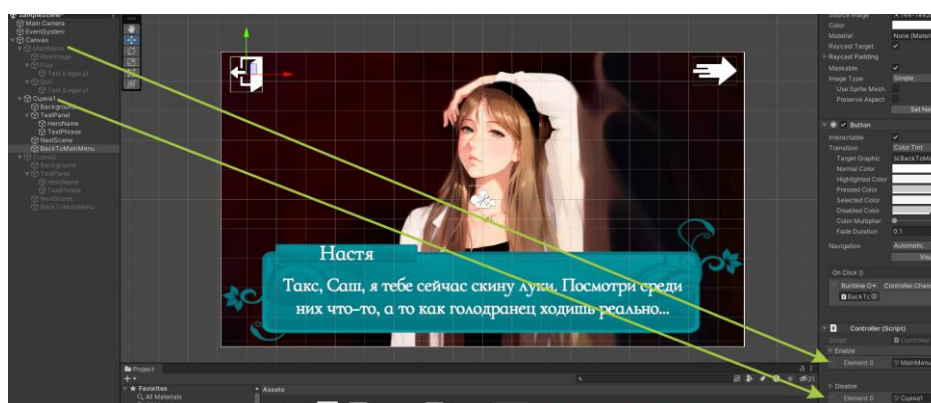


Рисунок 11 – Добавление интерактивности к кнопке выхода в меню

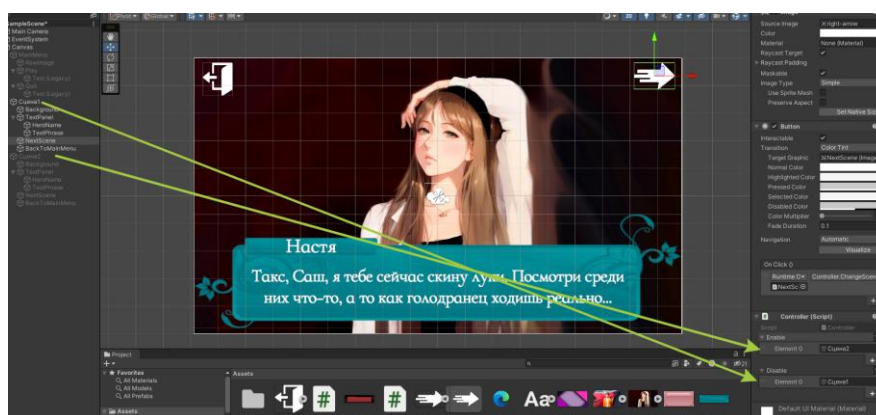


Рисунок 12 – Добавление интерактивности к кнопке смены сцен

Аналогичным образом добавляем функциональность к следующей и последующим сценам, но на последней переход выполняется в главное меню, так как эта сцена будет завершающей.

Завершающим этапом добавим плавную тряску картинки в случайную сторону, для этого добавим в файл Mover.cs скрипт, изображенный на рисунке 13.

```
using UnityEngine;

public class RandomMovement : MonoBehaviour
{
    public float speed = 0.1f;
    public float radius = 0.3f;

    private Vector3 center;
    private Vector3 targetPosition;

    void Start()
    {
        center = transform.position;
        SetRandomTargetPosition();
    }

    void Update()
    {
        transform.position = Vector3.MoveTowards(transform.position, targetPosition, speed * Time.deltaTime);

        if (Vector3.Distance(transform.position, targetPosition) < 0.1f)
        {
            SetRandomTargetPosition();
        }
    }

    void SetRandomTargetPosition()
    {
        float angle = Random.Range(0, 2 * Mathf.PI);
        float x = center.x + radius * Mathf.Cos(angle);
        float y = center.y + radius * Mathf.Sin(angle);
        targetPosition = new Vector3(x, y, transform.position.z);
    }
}
```

Рисунок 13 – Код перемещения изображения в случайную сторону относительно центра изображения

Выводы

В заключение, создание визуальной новеллы в Unity, даже с использованием всего двух слайдов, представляет собой комплексный и многогранный процесс, который охватывает как творческие, так и технические аспекты разработки игр. Эта статья подробно рассмотрела ключевые этапы разработки, включая концептуализацию, проектирование, реализацию и тестирование. Подход, описанный в статье, демонстрирует, как эффективно использовать инструменты и возможности Unity для создания интерактивного контента, который может привлечь внимание и удержать интерес пользователей.

Процесс разработки визуальной новеллы в Unity требует внимательного подхода к созданию сюжетных линий, персонажей и интерфейсов, а также тщательной настройки и оптимизации технических аспектов. Обозначенные в статье методы и рекомендации могут служить полезным ресурсом для начинающих разработчиков, помогая им избежать распространенных ошибок и эффективно использовать имеющиеся ресурсы.

Статья подчеркивает важность как творческого, так и технического аспектов разработки, а также способствует лучшему пониманию процессов, связанных с созданием визуальных новелл. Учитывая растущий интерес к этому жанру и популярность Unity, как инструмента разработки, полученные знания будут полезны не только для индивидуальных разработчиков, но и

для образовательных учреждений и студий, занимающихся созданием интерактивного контента.

Библиографический список

1. Camingue J., Melcer E. F., Carstensdottir E. A (Visual) novel route to learning: a taxonomy of teaching strategies in visual novels // Proceedings of the 15th International Conference on the Foundations of Digital Games. 2020. С. 1-13.
2. Galingana C. D., Cagadas R. B. Enticing Player-NPC Dialogue System in a Visual Novel Game Using Intent Recognition with BERT // East Asian Journal of Multidisciplinary Research. 2023. Т. 2. №. 7. С. 2999-3014.
3. Florensia J., Suryadibrata A. 7-Day Math: A Mobile Visual Novel Game for Mathematics Education // International Journal of Interactive Mobile Technologies. 2023. Т. 17. №. 6. С. 197.
4. Garcia M. B. Kinder Learns: An Educational Visual Novel Game as Knowledge Enhancement Tool for Early Childhood Education // FEU Institute of Technology. 2020. Т. 27. №. 1. С.1-25.
5. Griffin II M. J. The Visual Novel: Fictional Space and Print After 1900. Louisiana State University, 2014.