

ПРОГРАМНА РЕАЛІЗАЦІЯ СЦЕНАРІЮ ТА ЛОГІКИ ІГРОВОГО АНТАГОНІСТИЧНОГО 2D-ДОДАТКУ В ЖАНРІ «АРЕНА»

Дмитрієва О.А., д.т.н., проф., olga.dmytriyeva@donntu.edu.ua

Кролевець В.І., бакалавр, vadim.krolevets.knt@donntu.edu.ua

Донецький національний технічний університет, Покровськ, Україна

На сьогоднішній день відеоігрова індустрія є однією з найпопулярніших і найприбутковіших сфер в ІТ світі [1]. Окрім безпосередньої розробки самих ігор завдяки ігровим рушіям можна відновляти зображення при проведенні медичних досліджень, відтворювати спеціальні ефекти в кіно, реалізовувати доповнену реальність. Вклад рушіїв в наукову сферу, а саме в космічні дослідження (імітація гравітації для об'єкту, побудова ландшафту планети та інш.), в тренажерну симуляцію підготовки фахівців (медичного персоналу, персоналу, що забезпечує функціонування небезпечних об'єктів, пілотів, військових, дизайнерів і т.п.) було б неможливим без сучасних технологій обробки фізичних закономірностей, графіки і моделювання з використанням ігрових рушіїв.

Сучасні технології розробки відеоігор суттєво відрізняються від тих, що спостерігалися ще декілька років тому [1], сучасна ігрова індустрія має велику кількість безкоштовних і потужних рушіїв, які орієнтовані на різні платформи, жанри, віртуальні середовища.

В даній роботі використовувався безкоштовний ігровий рушій Unreal Engine 4 [2], який написаний на мові програмування C++ та підтримує створення ігрових додатків, 3D-моделей, тренажерів, спеціальних ефектів для фільмів (зокрема «Зоряні війни», «Мандалоріанець», «Аватар» та інш.). Можливості для створення ігор в Unreal Engine дозволяють розробникам займатися дизайном та логікою гри, не перетинаючись між собою, це досягається завдяки можливостям внутрішньої мови програмування Blueprint. У складі технології Unreal Engine також наявні фізичний рушій, штучний інтелект, управління системними та мережевими файлами, багатофункціональний редактор UnrealEd, що забезпечує редагування рівнів і ресурсів в реальному часі на основі моделювання поведінки об'єктів з використанням конструктивної блочної стереометрії.

Сюжет гри, що розроблялася, розповідає про двох безіменних магів з ворогуючих кланів, які зустрілись в свій останній час, щоб дізнатися, хто з них найсильніший маг. Гравець бере на себе управління одним з магів і починає змагатися з опонентом, поки не програє або переможе.

Гра повинна бути динамічною, та легкою у використуванні [3]. Гравець повинен стріляти в різних напрямках, поєднуючи все з бігом і стрибками. Персонаж має обмежену кількість життєвих показників. Коли персонаж втрачає останній показник життя, він програє, а інший перемагає. На ігровому полі весь час знаходяться обидва персонажі та платформи, по яким

герої можуть пересуватися. Кожен з гравців може наносити пошкодження своєму супротивнику, за певних умов блокувати його здібності, втрачати та відновлювати сили.

Проектування програмної системи та опис логіки здійснювався на основі побудови UML-діаграм. На рис. 1 зображена UML-діаграма програмного додатку, що описує процеси взаємодії персонажів. Обидва персонажа змагаються між собою, Hrbar показує динаміку зміни здібностей у кожного з гравців. За результатами взаємодії гравців формуються вікна поразки (перемоги).

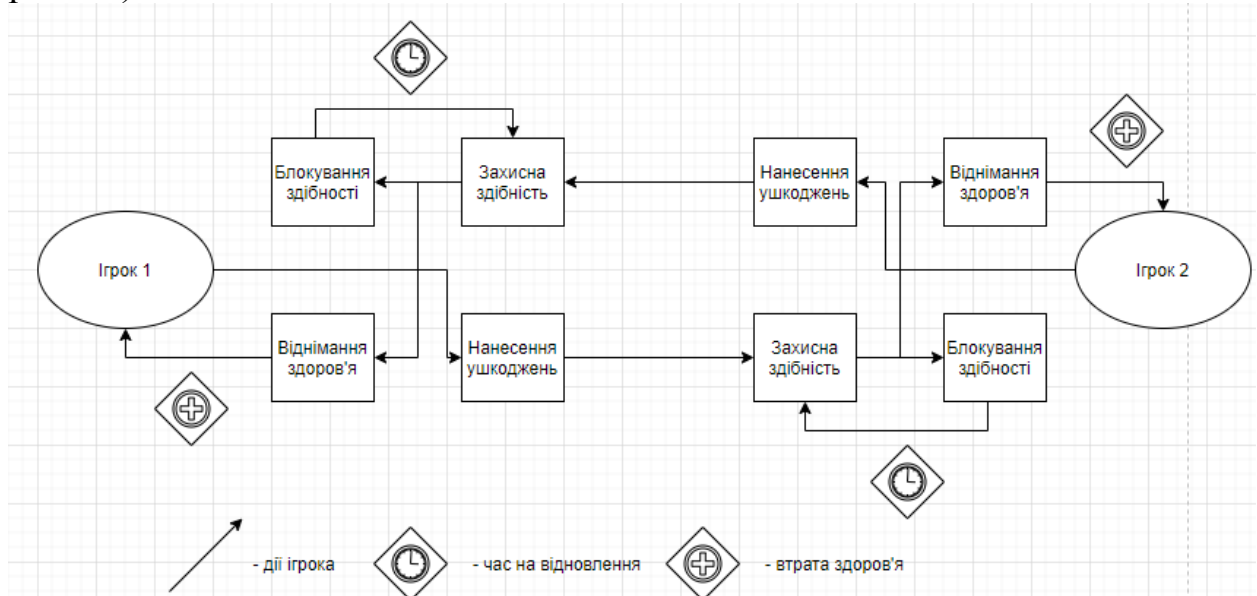


Рисунок 1. UML-діаграма взаємодії гравців

Також при проектуванні програмного додатку використовувалися UML-діаграми прецедентів, де кожен прецедент — це набір дозволених дій, які управляють системою за допомогою діалогу з актором. Крім зв'язку між акторами і прецедентами на діаграмах представлено і відносини між прецедентами. Відносини включення використовуються для копіювання описів, якщо в системі є фрагмент, який повторюється в декількох прецедентах. Відносини розширення використовується, якщо є прецедент, подібний до іншого, але набагато ширше його, крім того, при побудові моделі розширений прецедент може доповнювати поведінку базового, для цього в базовому передбачено наявність точок «розширення». Діаграма послідовностей в роботі використовувалася для опису взаємодії об'єкта в рамках одного прецеденту. Основним завданням такої діаграми є забезпечення впорядкованої за часом взаємодії об'єктів.

Для формування сцен гри використовувалися спеціальні мапи середовища Unreal Engine 4, за допомогою яких створювалися текстури, шейдери, освітлення, платформи, декоративні об'єкти.

Вигляд головного персонажу (рис. 2) є однією з самих важких частин

в розробці гри. Персонаж має бути несхожим на інших подібних персонажів з других ігор, мати тематичний одяг. Тому якщо команда розробників не має у своєму складі дизайнера, то пошук спрайтів, або зображень для своєї гри, становиться важкою, а в деяких випадках навіть неможливою справою. Тому було прийнято рішення, використовувати безкоштовний набір ассетів Unreal Engine 4 [2].

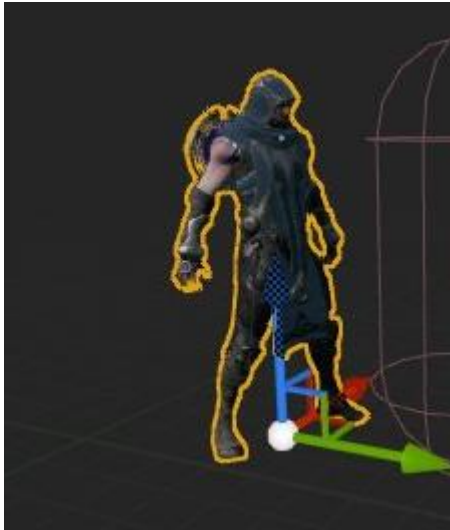


Рисунок 2. Модель головного персонажа



Рисунок 3. Колізія вогняної кулі при переміщенні

Тестування ігрового додатку проводилося за множиною напрямків, серед яких тестування функціональності, відповідності, навантаження, бета-тестування. Функціональні тести було спрямовано на пошук проблем всередині самого додатку (питання стійкості, пошук механічних проблем, цілісність ігрових активів, оцінювання часових затримок) та в інтерфейсі гри. Тестування відповідності було спрямовано на форматування стандартних повідомлень про помилки, обробку даних з карти пам'яті. Тестування навантаження визначало вплив кількості гравців на потужність ММО сервера, кількість активних спрайтів на екрані, число потоків, колізії об'єктів (рис. 3) та персонажів при пересуванні.

Остаточним результатом даної роботи є унікальна та динамічна гра, в якій можна змагатися з іншим гравцем (гравцями), використовуючи здібності персонажа.

Література

1. Популярність відеоігор в ІТ-сфері. [Електронний ресурс]. – Режим доступу: <https://www.epravda.com.ua/ukr/publications/2018/09/11/640413/>
2. Системні характеристики Unreal Engine 4.[Електронний ресурс]. – Режим доступу: <https://www.native-game.com/ue4-docs/get-started/tehnicheskie-harakteristiki/>
3. Диксит А. Теория игр. Искусство стратегического мышления в бизнесе и жизни/ А. Диксит, Б. Нейлбафф. – М.: «Манн, Иванов и Фербер». - 2015. – 464 с.

Анотація

В роботі досліджено та реалізовано процеси створення ігрового додатку на базі рушія Unreal Engine 4. Розроблено сценарій, логіку, основні методи і алгоритми антагоністичної 2D гри, спроектовано програмну систему засобами C++, Blueprint, UnrealEd. Методи, що використовувалися в роботі, базуються на основних принципах теорії ігор, теорії алгоритмів, структур даних, об'єктно-орієнтованого програмування, комп'ютерної графіки. Проектування програмного додатку здійснювалося з використанням UML-діаграм. Для формування сцен гри використовувалися спеціальні мапи середовища Unreal Engine 4, за допомогою яких створювалися текстури, шейдери, освітлення, платформи, декоративні об'єкти. Тестування ігрового додатку проводилося за множиною напрямків, серед яких тестування функціональності, відповідності, навантаження, бета-тестування. Практичним результатом є розроблений працездатний ігровий додаток в жанрі «арена».

Ключові слова: ігровий додаток, логіка, сценарій, рушій Unreal Engine, мова C++, Blueprint, UML-діаграма, тестування

Аннотация

В работе исследованы и реализованы процессы создания игрового приложения на базе движка Unreal Engine 4. Разработаны сценарий, логика, основные методы и алгоритмы антагонистической 2D игры, спроектирована программная система средствами C++, Blueprint, UnrealEd. Используемые в работе методы базируются на основных принципах теории игр, теории алгоритмов, структур данных, объектно-ориентированного программирования, компьютерной графики. Проектирование программного приложения осуществлялось с использованием UML-диаграмм. Для формирования сцен игры использовались специальные карты среды Unreal Engine 4, с помощью которых создавались текстуры, шейдеры, освещение, платформы, декоративные объекты. Тестирование игрового приложения проводилось по множеству направлений, среди которых тестирование функциональности, соответствия, нагруженности, бета-тестирование. Практическим результатом является разработанное работоспособное игровое приложение в жанре «арена».

Ключевые слова: игровое приложение, логика, сценарий, движок Unreal Engine, язык C++, Blueprint, UML-диаграмма, тестирование

Abstract

The work investigates and implements the processes of creating a game application based on the Unreal Engine 4. A scenario, logic, basic methods and algorithms for an antagonistic 2D game have been developed, a software system has been designed using C ++, Blueprint, UnrealEd. The methods used in this work are based on the basic principles of game theory, theory of algorithms, data structures, object-oriented programming, computer graphics. The design of the software application was carried out using UML diagrams. To form the scenes of the game, special maps of the Unreal Engine 4 environment were used, with the help of which textures, shaders, lighting, platforms, decorative objects were created. Testing of the gaming application was carried out in many areas, including testing of functionality, compliance, load, beta testing. The practical result is a developed workable game application in the arena genre.

Keywords: game application, logic, script, Unreal Engine, C ++, Blueprint, UML diagram, testing