

Олексюк Вадим, ст. магістратури факультету кібернетики; науковий керівник – д.т.н. професор Власюк А. П., (Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука, м. Рівне)

СТВОРЕННЯ АЛГОРИТМІВ ВЗАЄМОДІЇ ОБ'ЄКТІВ У КОМП'ЮТЕРНІЙ ГРІ НА ПРИКЛАДІ РУШІЯ UNITY 3D

***Анотація.** У статті досліджено особливості ігрової індустрії та процес створення конкурентно спроможної гри, яка створена у вигляді 3D Survival horror з допомогою ігрового рушія Unity 3D. Охарактеризовано найпопулярніші комп'ютерні ігри останнього року. Проаналізовано основні технології, які використовуються при створенні ігор, і представлено результати їх порівняльного аналізу.*

***Ключові слова:** комп'ютерна гра, 3D гра, гра - Survival horror, Unity 3D.*

***Аннотация.** В статье исследованы особенности игровой индустрии и процесс создания конкурентно способной игры, которая создана в виде 3D Survival horror с помощью игрового движка Unity 3D. Охарактеризованы популярные компьютерные игры последнего года. Проанализированы основные технологии, используемые при создании игр, и представлены результаты их сравнительного анализа.*

***Ключевые слова:** компьютерная игра, 3D игра, игра - Survival horror, Unity 3D.*

***Annotation.** This article describes the features of the gaming industry and the creation process of competitive game, created as a 3D Survival horror using Unity 3D game engine. The article discusses the most popular computer games of the last year. The basic technology used in the creation of games, and the results of their comparative is analyzed.*

***Keywords:** computer game, 3D game, the game - Survival horror, Unity 3D.*

В останні роки особливої набрало створення нових комп'ютерних ігор. Сьогодні створюються все нові і нові моделі та вдосконалюються уже наявні інструментарії для створення ігор. З допомогою цих програм навіть людина без досконального знання програмування може випробувати свої сили в цій цікавій сфері, не затрачуючи колосальних зусиль та фінансових ресурсів.

Останнім часом все частіше можна почути про збільшеннях масштабів розробки програмного забезпечення розважального характеру. До числа таких продуктів відносяться відеоігри, а особливо 3D шутери та бродилки. На протязі останнього десятку років індустрія відеоігор зайняла фіксоване

місце на ринку поряд з іншими розвагами сфери мультимедіа, такими як кіно, мультиплікація, музика, та впевнено витісняє їх.

За витісненням з ринку інших розваг стоять багатомільярдні компанії, що займаються розробкою комп'ютерних ігор та додатків на смартфони, які є в кожного з нас. Ці всі проекти складаються з мільйонів рядків програмного коду та файлів мультимедіа, над створенням яких трудяться команди розробників – програмістів, художників, дизайнерів рівнів, керівників підрозділів. Такі команди складаються від декількох чоловік у маленьких компаніях до декількох тисяч чоловік у великих корпораціях.

Грати в комп'ютерні ігри – заняття цікаве, однак набагато цікавіше, самому їх створювати і надавати можливість іншим людям грати в свій ігровий продукт. Створення складних ігор – заняття досить непросто. Ігри, які Ви купуєте, створюються від одного до трьох років і розробляються командами від 10 до 200 (а то й більше) осіб. Нерідко дохід від таких ігор сягає кількох мільйонів доларів. Люди, що працюють над такими проектами, професіонали своєї справи: програмісти, дизайнери, звукорежисери і т.д.

І що ж з цього випливає, що створити гру для нас стає неможливим? На щастя це не так. Звичайно, не слід розраховувати, що Ви зможете створити свій світ подібно Quake або Age of Empires за кілька тижнів. Але чи так це важливо? Існує маса простих ігор, таких як: Tetris, Pacman, Space Invaders і т.д., в які не менш цікаво грати і при цьому самому просто зробити. На жаль, у звичайних умовах для створення такої гри потрібні непогані навички програмування, уміння обробляти графіку, звук, інструменти управління та інше [1].

Але після появи Unity3D, який розроблений спеціально для того, щоб надати можливості створити повноцінну 3D гру, розробка таких ігор стала набагато простіша. Також одним з переваг цього движка є можливість створення продуктів як на ПК так і на мобільні пристрої з різними ОС (Операційними Системами).

Ситуація на ринку ігор у цьому році характеризується надзвичайною популярністю таких 3D ігор як:

- GTA 5;
- Battlefield Hardline;
- Far Cry 4;
- Bioshock Infinite;
- Assassin's Creed;
- DayZ.

Всі ці ігри основані на тривимірному просторі (3D) та керуванні від першої особи (First Person Shooter). Однією з причин їхнього успіху є використання популярних на даний час підходів конструювання світу (3D) та схеми керування (FPS) згаданих вище. Майже всі проекти, які розглядаються, створювались величезними компаніями з багатомільйонним фінансуванням

та величезними командами розробників, але гра DayZ (точніше перша її версія) була створена однією людиною як доповнення до вже існуючої гри Arma 2 [2].

DayZ – багатокористувачка комп'ютерна гра зі схемою керуванні від першої особи в жанрі survival horror (завданням гри є виживання у суворому світі з мінімальними ресурсами) з відкритим світом. Гра є переробкою однойменної модифікації до гри Arma 2, створеної геймдизайнером Діном Холлом.

Ця гра є яскравим прикладом того, що одна людина без великих фінансових ресурсів та команди може придумати та реалізувати ідею, хоч спочатку і не повністю, яку потім підхоплять спонсори та нададуть їй все для того, щоб закінчити проект та зробити його справді якісним.

Потрібно створити гру у форматі 3D з характерним для First Person Shooter типом керування від першої особи в жанрі Horror (гра в жаху) – такий жанр надзвичайно популярний на даний час. Вона буде створена з усіма описаними вище особливостями та особливою атмосферою.

Від того, на скільки гра цікава для користувачів залежить, успішність проекту в цілому, тому потрібно вивчити та дотримуватись усіх правил створення цікавої комп'ютерної гри. Під актуальністю розуміється напрям розробки ігор в даний час. За останній рік великої популярності досягли так звані survival horror's (завданням гри є виживання у суворому світі з мінімальними ресурсами) [3].

В якості движка для створення гри був обраний Unity, який є дуже популярним в даний час через можливість розробки проектів як на ПК та і на мобільні пристрої. Редактор Unity має простий Drag&Drop інтерфейс, з інтуїтивним інтерфейсом його налаштування, який складається з різних вікон, завдяки чому можна налагоджувати гру прямо в редакторі. Движок підтримує три сценарних мови: C#, JavaScript (модифікація), Boo (діалект Python). Редактор підтримує DirectX 11 і HDR. Розрахунки фізики робить фізичний движок PhysX.

Проект в Unity ділиться на сцени (рівні) – окремі файли, що містять свої ігрові світи зі своїм набором об'єктів, сценаріїв та налаштувань. Сцени можуть містити в собі як, власне, об'єкти (моделі), так і порожні ігрові об'єкти — об'єкти, які не мають моделі («пустушки»). Об'єкти, в свою чергу містять набори компонентів, з якими і взаємодіють скрипти. Також у об'єктів є назва (у Unity допускається наявність двох і більше об'єктів з однаковими назвами), може бути тег (мітка) і шар, на якому він повинен відображатися. Так, у будь-якого об'єкта на сцені обов'язково присутній компонент Transform – він зберігає в собі координати місця розташування, повороту, і розмірів об'єкта по всіх трьох осях. У об'єктів з видимою геометрією також за замовчуванням присутній компонент Mesh Renderer, що робить модель об'єкта видимою [4; 5].

До об'єктів можна застосовувати колізії (в Unity вони називаються Колайдери - collider). Існує кілька типів колайдерів:

- Character controller – вид фізичної моделі, створений спеціально під використання його для ігрових персонажів;

- Box collider (фізична модель утворює куб, в який потрапляє вся модель об'єкта);

- Sphere collider (фізична модель утворює сферу, в яку потрапляє вся модель об'єкта);

- Capsule collider (фізична модель утворює капсулу, в яку потрапляє модель об'єкта. На відміну від попереднього типу розміри можна міняти і по одній, і по трьох осях відразу);

- Mesh collider (фізична модель повністю повторює реальну геометрію об'єкта);

- Wheel collider (фізична модель колеса);

- Terrain collider – тип фізичної моделі, створений спеціально для використання на об'єкті типу Terrain — земля, що генерується редактором Unity з можливостями створення та фарбування місцевості.

Також Unity підтримує фізику твердих тіл і тканини, а також фізику типу Ragdoll (ганчіркова лялька). У редакторі є система успадкування об'єктів; дочірні об'єкти будуть повторювати всі зміни позиції, повороту і масштабу батьківського об'єкта. Скрипти в редакторі прикріплюються до об'єктів у вигляді окремих компонентів.

Редактор Unity підтримує написання та редагування шейдерів. Редактор Unity має компонент для створення анімації, але також анімацію можна створити попередньо в 3D-редакторі та імпортувати разом з моделлю, а потім розбити на файли.

Крім порожнього ігрового об'єкту і моделей, на сцену можна додавати ще такі об'єкти типу GameObject:

- Система частинок;

- Камера;

- GUI текст;

- GUI текстура;

- 3D текст;

- Точковий світло;

- Направлене світло;

- Освітлення території;

- Джерело світла, що імітує сонце;

- Стандартні примітиви;

- Дерева;

- Terrain (земля).

Unity 3D підтримує систему Level Of Detail (скор. LOD), суть якої полягає в тому, що на далекій відстані від гравця високо деталізовані моделі

замінюються на менш деталізовані, і навпаки, а також систему Occlusion culling, суть якої в тому, що у об'єктів, які не потрапляють у поле зору камери не візуалізується геометрія і колізія, що знижує навантаження на центральний процесор і дозволяє оптимізувати проект. При компіляції проекту створюється виконуваний (.exe) файл гри (для Windows), а в окремій папці – дані гри (включаючи всі ігрові рівні і спільні бібліотеки) [6; 7].

Результатом роботи є гра у форматі 3D з типом керування від першої особи (First Person) в жанрі Horror — такий жанр надзвичайно популярний на даний момент. Була створена динамічна гра з елементами хорору в яку цікаво грати. Ігровий проект реалізований по таким головними вимогами:

- цікавість для гравця;
- актуальність сучасним ігровим тенденціям;
- зручність інтерфейсу;
- відсутність зайвих елементів;
- максимальна оптимізація під сучасну техніку.

Цей проект був реалізований на обраному движку Unity, який надав необхідні можливості для створення якісного продукту без великих грошових та людських ресурсів. Так як цей движок надає можливість розробки проектів як на ПК та і на мобільні пристрої, то у майбутньому можна буде портувати дану гру на різні пристрої, таким чином охоплюючи більшу аудиторію користувачів.

Взаємодія між користувачем та комп'ютером зводиться до керування руху персонажем стрілками-вказівниками на клавіатурі та камерою через рух миші. Така схема керування є традиційною для ігор такого жанру, тому всі вимоги для такої механіки були виконані. Також додатковими клавішами керування руху були створені контролери на клавішах WASD, які є стандартними майже для усіх ігор які виходять у світ.

Також під час створення даного проекту були здобуті дорогоцінні знання та навички які знадобляться у майбутньому. Основним здобутком вважається удосконалення знань таких мов як C# та JavaScript, а головне їхню коректну взаємодію [8].

Ця стаття присвячена розробці алгоритмів взаємодії між об'єктами, які організуються у повноцінний ігровий проект, який потім можна буде рекламувати та шукати спонсорів для доопрацювання та продажу, і з допомогою його залишитись у ігровій індустрії.

У останні роки все частіше можна почути про збільшення масштабів розробки програмного забезпечення розважального характеру. До числа таких продуктів відносяться відеоігри. На протязі останнього десятиліття індустрія відеоігор зайняла фіксоване місце на ринку поряд з іншими розвагами сфери мультимедіа, таких як кіно, мультиплікація, музика. Це багатомільярдні компанії, що займаються розробкою розважальних

продуктів та програмних забезпечень, що складаються з мільйонів рядків програмного коду і файлів мультимедіа.[9]

Навіть при великій кількості багатофункціональних інструментів, розробник має затрачати дуже велику кількість часу та зусиль, для створення дійсно цікавого продукту. Найголовнішим плюсом написання гри на вже готовому інструменті (движку), є набагато більша швидкість написання коду механіки. Раніше програміст був змушений писати сотні, і навіть тисячі, рядків коду, щоб надати об'єкту елементарну фізику, то при використанні движка все це викликається лише однією функцією.

З проведеного дослідження можна зробити висновок, що ігри стали охоплювати величезну аудиторію по всьому світу, з'являючись на різних ігрових пристроях. Вони стали складніші і масштабніші: покращилася обробка візуального простору, обробка фізики об'єктів та імітація інтелекту для юнітів якими керує комп'ютер. Але напрямна методологія розробки не змінилася. У віртуальному просторі може знаходитися величезна кількість різноманітних об'єктів, але їх кількість як і раніше залежить від технічних можливостей пристроїв, на які розрахована відеогра. Інтерактивність в такій програмі зводиться до взаємодії між об'єктами. Вони можуть пересилати інформацію один одному, навколишньому простору і користувачеві в тому числі. Основною проблемою є завантаженість коду об'єктів взаємодії, що виконують, закріплений за ними, певний сценарій.

1. Липков А. Всюду деньги, деньги, деньги // А. Липков. Ящик Пандоры: феномен компьютерных игр в мире. — М., 2011. — 420 с. 2. Архангельский А. Я. Разработка прикладных программ для Windows / А. Я. Архангельский. — М. : БИНОМ, 2011. — 256 с. 3. Братищенко В. В. Проектирование информационных систем. — Иркутск : Изд-во БГУЭП, 2012. — 384 с. 4. Дашко Ю. В. «Основы разработки компьютерных игр» / Ю. В. Дашко, А. А. Заика. — М. : «Форум» 2012. — 350 с. 5. Роллингз, Эндрю. «Проектирование и архитектура игр» : пер. с англ. / Э. Роллингз, Д. Моррис. — М. : Вильямс, 2011. — 1040 с. 6. Силбигер М. Press Play to Grow! Designing Video Games as «Trojan Horses» to Catalyze Human Development through the Conveyor Belt of Growth // John F. Kennedy University, Pleasant Hill, CA, 2013 — 218 с. 7. Деникин А. А. Могут ли видеоигры быть искусством? / А. А. Деникин // Международный журнал исследований культуры, № 2(11), 2013. — М. : Эйдос, 2013. — 125 с. 8. Роллингз Э., Моррис Д. Проектирование и архитектура игр. Вильямс, 2011. — 1035 с. 9. Д. Конгер. Физика для разработчиков компьютерных игр. Бином, 2012. — 520 с.