

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних технологій

Кваліфікаційна робота бакалавра

на тему: «Розробка ігрового застосунку «Лабіринт»
за допомогою 3DMax та рушія Unity»

Виконав студент групи К-41
Спеціальності 122 Комп'ютерні науки,
Саркісян Карен Георгійович

Керівник доктор філософії, асистент
Бучинська Ірина Вікторівна

Рецензент д.ф.-м.н., професор
Ковальчук Володимир Володимирович

Одеса 2021

ЗМІСТ

ВСТУП	8
1 ПОНЯТТЯ КОМП'ЮТЕРНОЇ ГРИ	10
1.1. Класифікація за жанрами	10
1.2. Класифікація за кількістю гравців	11
1.3. Актуальність гри «Лабіринт»	11
2 ОПИС НАЙБІЛЬШ ПОПУЛЯТНИХ ІГРОВИХ ДВИГУНІВ	13
2.1 Короткий опис двигунів	13
2.1.1 Характеристика Unity3D	13
2.1.2 Характеристика Unreal Engine	14
2.1.3 Характеристика CryEngine 3	15
2.1.4 Характеристика HeroEngine	15
2.1.5 Характеристика Rage Engine	16
2.1.6 Характеристика Project Anarchy	16
2.1.7 Характеристика GameSalad	17
2.1.8 Характеристика GameMaker: Studio	18
2.1.9 Характеристика App Game Kit	18
2.1.10 Характеристика Cocos2D	19
2.2 Програмне забезпечення для 3D-моделювання для Unity	20
3 ОСНОВНІ ХАРАКТЕРИСТИКИ ДИЗАЙНУ ГРИ	23
3.1 Процес створення 3D-моделі для Unity	24
3.2 Основні характеристики моделювання	24
3.3 Техніка 3D-моделювання в іграх	25
3.3.1 Характеристика ретопології	26
3.3.1 Характеристика УФ-відображення	26
3.3.3 Основні характеристика такелажу	27
3.3.4 Опис шкірування	28
3.3.5 Опис текстурування	29
3.4 Робота з моделями в Unity	30

3.5 Імпорт об'єктів із програмного забезпечення для тривимірного моделювання.....	30
3.6 Вбудована підтримка Unity 3ds Max.....	31
3.7 Вбудована підтримка з БД.....	32
4 ОПИС ПРОГРАМНОЇ ЧАСТИНИ.....	36
ВИСНОВКИ.....	48
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	50

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

NURBS – Non-uniform rational B-spline – нерівномірний
раціональний базисний сплайн

GPU – graphics processing unit – високопродуктивні графічні
процесори

MMORPG – Massively multiplayer online role-playing game,

ВСТУП

Після багатьох років еволюції електронні розваги стали потужним і зрілим середовищем. Часи, коли ігри робили в гаражах за допомогою купки людей, давно минули. Сучасні якісні ігри вимагають величезного та кваліфікованого персоналу та величезних фінансових витрат, що обчислюються мільйонами доларів.

Індустрія комп'ютерних ігор зросла стрибками, перетворившись на основний сектор розробки програмного забезпечення та заробляючи мільярди доларів щороку. Ігрова індустрія також була однією з основних рушійних сил розвитку сучасного сучасного обладнання, такого як багатоядерні, гіперпоточні процесори, високопродуктивні графічні процесори (GPU), вдосконалені пристрої обробки звуку та комп'ютерні інтерфейсні пристрої, такі як шоломи віртуальної реальності та кришки датчиків мозку. Комп'ютерні ігри схожі на книгу, фільм чи музей.

Процес створення віртуальних ігор надзвичайно складний. В роботі описані різні етапи розвитку, зосереджуючись на графіці та анімації ігор.

Багато компаній займаються темою проектування програмного забезпечення для створення 3D-графіки. На перший план виходить Autodesk зі своїми програмами: 3DS Max, Maya, MotionBuilder, Mudbox. Це набір інструментів, визнаний у всьому світі провідними розробниками ігор. Найбільші у світі виробництва були створені за допомогою програмного забезпечення Autodesk.

Метою даної роботи є створення гри на основі ігрового механізму Unity. Використавши візуальний, більш прийнятний засіб для зображення світу.

Для виконання мети необхідно виконати наступні завдання:

- проаналізувати програмне середовище;
- реалізувати проект, зручний у використанні.

Головні можливості, які гра повинна надавати гравцеві: можливість вибору рівнів і можливість навчання в процесі гри. Гра повинна надавати інформацію в коректному і зрозумілому вигляді, а ігровий процес повинен бути інтуїтивно зрозумілий і повинен сприяти кращому засвоєнню та закріпленню отриманої інформації. Рішення повинно володіти функціоналом, що дозволяє впроваджувати в ігровий процес створений фахівцями ігровий контент.

Даний проект повинен розроблятися на движку Unity. Вибір інструмента розробки обумовлений його гнучкістю, що дозволяє охопити такі платформи як Windows, IOS, Android і інші; доступністю. В якості мови написання скриптів слід вибрати C #. Для створення ігрових спрайтів, моделей і анімації слід використовувати такі засоби, як 3D Max і Adobe Photoshop.

Робота складається з вступу, розділів, висновків, джерел, рисунків, 1 таблиці.

1 ПОНЯТТЯ КОМП'ЮТЕРНОЇ ГРИ

Комп'ютерна гра – це комп'ютерна програма, що служить для організації ігрового процесу, створення і підтримання зв'язків між гравцями, або сама виступає в якості партнера. Як правило, ігровий процес в іграх відтворюється на екрані дисплея або телевізора, а управління ігровим процесом відбувається за допомогою будь-якого контролера.

1.1. Класифікація за жанрами

«Action» – жанр комп'ютерних ігор, в яких успіх гравця у великій мірі залежить від його швидкості реакції і здатності швидко приймати тактичні рішення. Дія таких ігор розвивається дуже динамічно і вимагає високу концентрацію уваги і швидкої реакції на події в грі події. [1]

«Симулятор» – імітатор (зазвичай механічний або комп'ютерний), завдання якого полягає в імітації управління будь-яким процесом, апаратом або транспортним засобом. Найчастіше зараз слово «симулятор» використовується стосовно до комп'ютерних програм (зазвичай ігор). [2]

«Стратегія» – популярний жанр комп'ютерних ігор, в якому запорукою досягнення перемоги є планування і стратегічне мислення. [3]

«Рольова гра» – жанр комп'ютерних ігор, заснований на елементах ігрового процесу традиційних настільних рольових ігор. У рольовій грі гравець управляє одним або декількома персонажами, кожен з яких описаний набором чисельних характеристик, списком здібностей і вмінь; прикладами таких характеристик можуть бути хіт-пойнти (англ. hit points, HP), показники сили, спритності, захисту, ухилення, рівень розвитку того чи іншого навичку і т.п. В ході гри вони можуть змінюватися. Одним з характерних елементів ігрового процесу є підвищення можливостей персонажів за рахунок поліпшення їх параметрів і вивчення нових здібностей. [4]

1.2. Класифікація за кількістю гравців

«Поодинокі» – вид комп'ютерної гри, в якій участь приймає одна людина. В якості опонента в таких іграх виступає комп'ютер або ігрова платформа.

«На багато користувачів гра» – тип комп'ютерних ігор, при якому одночасно грає кілька людей. Гра може проходити на одному комп'ютері або на декількох (онлайн-гра); гравці можуть грати одночасно або по черзі.

«Масова розрахована на багато користувачів ролева онлайн гра» – також називається ММОПГ (англ. Massively multiplayer online role-playing game, MMORPG). Це об'єднання таких жанрів ігор, як «багато користувачів гра» і «рольова гра». Відмінною особливістю цих уявлень є можливість одночасного перебування в грі великої кількості людей. Гравець приймає на себе роль ігрового персонажа і, що характерно для рольових ігор, займається його розвитком. Ігровий світ продовжує існувати постійно, підтримується силами видавця гри і не залежить від виходу з нього будь-якого гравця.[5]

1.3. Актуальність гри «Лабіринт»

В даний час актуальні стали шари-лабіринти або 3-D головоломки. Звичайні лабіринти, які цікаві в усі часи. Їх варіативність дуже різноманітна, настільно-друкований варіант, лабіринти-ниткові, і ін.

Розвиток інтелектуальних здібностей дітей дошкільного віку - одна з актуальних проблем сучасності. Дошкільнята з розвиненим інтелектом швидше запам'ятовують матеріал, більш впевнені в своїх силах, легше адаптуються в новій обстановці, краще підготовлені до школи.

Навчання краще здійснювати в природному, самому привабливому вигляді діяльності – грі. В процесі гри розвиваються – планування, вміння аналізувати результати, уяву та ін. Безсумнівним достоїнством ігри являЄ і внутрішній характер мотивації. Діти грають тому, що їм подобається сам ігровий процес. Гра призначена для дитячого віку.

Мета: сприяє розвитку логічного і просторового мислення, уваги, спостережливості і посидючості, а також розвивається особлива якість як цілеспрямованість при самостійному використанні.

Дана гра вимагає спокою і концентрації уваги – це тренує посидючість у дитини.

Цю гру можна використовувати для індивідуальної роботи з дітьми або діти використовують її в самостійній ігровій діяльності.

2 ОПИС НАЙБІЛЬШ ПОПУЛЯТНИХ ІГРОВИХ ДВИГУНІВ

Ринок консолей і мобільних ігор росте, а тому знайти правильний движок – не така проста задача, як може здатися. Технологічні платформи, які вже довели свою ефективність «в польових умовах», стануть очевидними фаворитами багатьох розробників. Проте, ринок мінливий, а тому питання вибору движка своєї актуальності найближчим часом не втратить. Отже, представляю вашій увазі десятку кращих.

2.1 Короткий опис двигунів

2.1.1 Характеристика Unity3D

Unity 3D – відмінний доступний движок. У нього є ряд незаперечних переваг перед конкурентами і, мабуть, ключове з них в тому, що за ліцензію ви платите всього один раз. Неважливо, наскільки популярною стає гра – якщо ви вибрали Unity, вам не доведеться розщедрюватися ще раз [5]. З фінансової точки зору – це вдале рішення, особливо для стартапів і початківців розробників (рис. 1).



Рисунок 1 – Логотип Unity

Плюси:

- вигідна ліцензійна політика;
- легкість у використанні;

- сумісність з будь-якою платформою;
- відмінне ком'юніті;
- низький поріг входу;
- популярний серед розробників (це означає, що помилки швидко знаходять і виправляють).

Мінуси:

- обмежений набір інструментів (вам, швидше за все, доведеться розробити деякі з них самим);
- процес виготовлення гри забирає багато часу.

2.1.2 Характеристика Unreal Engine

Unreal Engine – один з найбільш популярних движків для розробки AAA-ігор. Gears of War, Batman: Arkham Asylum, Mass Effect – всі ці хіти були зроблені саме на ньому. [6]

Плюси:

- оскільки безліч розробників його використовує, то у Unreal Engine, мабуть, краще ком'юніті серед конкурентів. Кілька годин відео-тutorіали тому підтвердження;
- відмінна техпідтримка механізм апдейта;
- нові інструменти виходять з кожним оновленням
- широкий асортимент інструментів для різних цілей (деякі настільки прості у використанні, що ними може керувати навіть школяр)
- сумісний з різними платформами (iOS, Android, Linux, Mac, Windows і більшість інших)
- нова ліцензійна політика включає підписку вартістю \$ 19 в місяць і 5% роялті, якщо гра заробить більше \$ 5,000, що робить движок куди більш привабливим для розробників, ніж раніше.

Мінуси:

- суб'єктивні. Деякі розробники скаржаться, що до певних інструментів складно звикнути .

2.1.3 Характеристика CryEngine 3

Якщо зовнішня складова гри – ваш пунктик, то вам потрібен саме CryEngine 3. [7]

Плюси:

- функція Flowgraph допоможе прикрасити гру відмінною графікою;
- набір функцій Fmod для створення потужного звукового супроводу;
- найпростіший процес створення AI в ф

Мінуси:

- щодо недбала техпідтримка безкоштовної версії;
- оскільки движок в індустрії порівняно недавно, йому ще тільки належить створити міцне ком'юніті;
- відносно високий поріг входження.

2.1.4 Характеристика HeroEngine

Цей движок добре зарекомендував себе в створенні мультиплеєрних ігор – взяти хоча б Star Wars: The Old Republic. Ліцензія досить дорога і навряд чи підійде початківцям розробникам, але якщо ваш проект амбітний, то я б радив розглянути цей варіант. [8]

Плюси:

- в наявності кілька карт для створення відкритого світу. Є можливість їх «безшовного» з'єднання;
- казково могутній AI;
- ручний набір інструментів для моделювання карт;
- підходить для створення комплексних місій, крафтинга і збирання ресурсів;
- техпідтримка здійснюється за допомогою сервісу HeroCloud, що вельми зручно.

Мінуси:

- скриптова движок потужний, але незручний в управлінні;

- HeroEngine разом з сервісом підтримки клієнтів HeroCloud занадто дорого коштує і навряд чи буде доступний початківцям розробникам;
- високий поріг входження.

2.1.5 Характеристика Rage Engine

Мало хто може конкурувати з широким спектром можливостей, які надає Rage Engine. Grand Theft Auto V, Red Dead Redemption і багато інших прославлених проекти зроблені за допомогою цього движка. [9]

Плюси:

- широкі можливості для створення великих світів і погодних ефектів;
- потужний AI;
- безліч стилів геймплея на вибір;
- швидкий мережевий код.

Мінуси:

- інтерфейс движка порівняно незручний;
- управління погано оптимізовано під клавіатуру і мишку.

2.1.6 Характеристика Project Anarchy

Цей потужний ігровий движок (рис. 2) подобається багатьом розробникам за наочну і зрозумілу документацію. Проте, і у нього є свої недоліки. [10]

Плюси:

- якщо ви плануєте розробляти гри на платформах iOS, Android і Tizen, то ліцензія – безкоштовна;
- потужні інструменти для пошуку і усунення багів;
- сильне ком'юніті;
- видавець надає чітку, зрозумілу документацію і зразки;
- Fmod для аудіо-супроводу;
- потужний Navok AI.



Рисунок 2 – Логотип Project Anarchy

Мінуси:

- відсутня можливість розробляти гру на Mac і Linux;
- немає вступного керівництва для початківців розробників;
- якщо гра для ПК, то ліцензія влетить вам у копійчку.

2.1.7 Характеристика GameSalad

Творці цього популярного ігрового движка обіцяють, що розробнику не доведеться написати ні рядка коду. В цілому, це дійсно так. Однак за все хороше доводиться платити: у движка є ряд істотних недоліків [11]. Якщо ви зібралися розробити гру на iPhone поодинці, то це ваш вибір.

Плюси:

- безкоштовна ліцензія (гроші з вас зажадають тільки за PRO-версію);
- активне ком'юніті;
- відмінний движок для швидкого створення прототипу;
- сумісність з популярними мобільними платформами такими, як Corona і Moai.

Мінуси:

- обмежений набір інструментів розробки;
- немає доступу до більшості можливостей платформи iOS.

2.1.8 Характеристика GameMaker: Studio

Якщо ви початківець розробник і вам потрібен простий і зрозумілий движок, то GameMaker: Studio повністю відповідає вашим цілям [12].

Плюси:

- • просте і інтуїтивно зрозуміле управління;
- • власну мову програмування Game Maker Language (GML);
- інтеграція з Steam;
- кроссплатформенність.

Мінуси:

- щодо складно усувати неполадки в грі;
- щоб експортувати свою гру на популярні платформи, доведеться доплатити круглу суму.

2.1.9 Характеристика App Game Kit

App Game Kit – багатоплатформовий софт для розробників (рис. 3). Цінується за універсальність і легкість в управлінні. [13]



Рисунок 3 – Логотип App Game Kit

Плюси:

- дозволяє писати коди для основних платформ: Android iOS, Windows, Mac і Linux;
- поставляється в комплекті з IDE, що дозволяє тестувати гри на будь-якому пристрої;
- без додаткової установки вже включає в себе IAP, AdMob і Push;

– є потужні скрипти для 2D графіки, фізики та мережевої взаємодії.

Мінуси:

– оскільки мало хто працює з цим движком, то недоліки програми довго не усуваються (відносно слабка технічна);

– безліч багів (що органічно впливає з попереднього пункту).

2.1.10 Характеристика Cocos2D

Багато дизайнерів вважають Cocos2D одним з небагатьох движків, що мають високу адаптивність і одночасно прекрасно пристосованих для початківців розробників (рис. 4) [14].



Рисунок 4 – Логотип Cocos2D

Плюси:

- відмінно інтегрований в платформу iOS;
- безкоштовний і з відкритим вихідним кодом;
- широкий вибір інструментів розробки;
- сильна підтримка ком'юніті.

Мінуси:

- більш складний в застосуванні, ніж більшість аналогів;
- високий поріг входження;
- «заточений» саме під Mac або iOS, відсутня кроссплатформеність.

На закінчення: люди грають в ігри на консолях, мобільних пристроях, ПК і навіть на ТБ – так що у розробника завжди є можливість знайти движок, який буде відповідати його цілям. Сподіваюся, моя стаття допоможе вам зробити свій вибір і з успіхом влитися в постійно зростаючу гейм-індустрію [6].

2.2 Програмне забезпечення для 3D-моделювання для Unity

Ігровий движок Unity був представлений компанією Unity Technologies у 2005 році і з тих пір став однією з найпопулярніших платформ для розробки 2D та 3D ігор. Її прийняли як малі сторонні розробники, так і великі комерційні розробки ігор компанії. Функцій, які підтримує Unity3D, дуже багато. Unity3D створює додатки на основі JavaScript та / або C#. Вони використовуються для призначення анімації або переходу в реальному часі ігрових об'єктів, визначених у програмі.

GUI Unity3D допомагає новому розробнику легко підійти до сценарію і запрограмувати перехід гри-об'єкта. Unity – це інтегроване середовище, яке поєднує в собі низку складних компонентів, таких як фізичний движок PhysX, анімаційна система Mechanim, автономний редактор місцевості та багато іншого. Він також легко інтегрується з редактором коду Monodevelop, так що будь-які зміни, внесені в Monodevelop, прозора компілюються компіляторами Unity C# або Javascript та вставляються в гру. Помилки компіляції відображаються у вікні консолі Unity.

Візуалізація Unity підтримує художні активи та формати файлів із 3ds Max, Maya, Softimage, Blender, Modo, ZBrush, Cinema 4D, Cheetah 3D, Adobe Photoshop, Adobe Fireworks та Algorithmic Substance.

Ці активи можна додавати до ігрового проекту та керувати ними через графічний інтерфейс користувача Unity. Unity також має вбудовану

підтримку фізичного двигуна PhysX від Nvidia (колишнього Ageia) (станом на Unity 3.0) з доданою підтримкою для моделювання тканини в режимі реального часу на довільних та шкірних сітках, товстих променях та шарах колізій. 3D-об'єкти, зроблені в 3dsMax, можна зберігати як файли .max безпосередньо в проект або експортуйте їх в Unity за допомогою Autodesk FBX або інших загальних форматів.

Unity3D Game Engine – це інтегрований інструмент розробки використовується для розробки інтерактивного вмісту, такого як відеоігри, архітектурна візуалізація та 3D-анімація в режимі реального часу. Його редактор працює на платформах Windows і Mac OS X. хоча він працює лише на двох платформах, він має можливість розробляти додатки для декількох платформ, які згадуються наступним чином - Windows, Windows Phone, Mac OS X, iOS, Linux, Android, Web Player тощо. Все, що нам для цього потрібно розробити додаток на відповідних платформах – це комплект для розробки програмного забезпечення (SDK) для нього.

Щоб використовувати весь функціонал Unity, потрібен об'єкт, 3D-модель. Однак для побудови 3D-моделі потрібно використовувати відповідне програмне забезпечення.

На ринку є 3 основні інструменти, з яких ви можете вибрати:

- Blender
- 3Ds Max
- Maya

Усі три можна використовувати для створення 3D-моделей для Unity. Основна різниця між цими трьома полягає в тому, що Blender – це безкоштовний двигун для розробки ігор, 3Ds Max – це програмне забезпечення, що використовується виключно для 3D-моделювання, а Maya має сильну анімацію.

Спільнота каже, що Blender – це складний двигун, якщо ви тільки починаєте, однак він безкоштовний, тож ви можете розглянути його.

3D-програми Max і Maya можуть бути занадто дорогими для початківців або якщо ви фрілансер. Ці інструменти в основному використовуються агентствами або відомими художниками 3D.

3 ОСНОВНІ ХАРАКТЕРИСТИКИ ДИЗАЙНУ ГРИ

3DS MAX включає багатий набір гнучких інструментів для створення першокласних дизайнів з повним художнім контролем:

- створює великі ігрові світи;
- надає високоякісні архітектурні візуалізації;
- детально моделює інтер'єри та предмети;
- є можливість оживити персонажів та елементи пейзажу за допомогою анімації та візуальних ефектів.

Отримавши необхідні посилання та рекомендації, можна перейти до створення моделей. Моделі також можна створювати в 3DS MAX. Ця програма має повний набір інструментів, необхідних для створення чудових 3D-моделей. Інша справа – привести створені моделі та об'єкти в рух. Для цього перед командою графічних дизайнерів ще багато роботи.

Першим етапом є побудова структури, на якій буде базуватися анімація. Іншим завданням є такелаж і зважування. Ці кроки необхідні для отримання натуральної та реалістичної анімації. Також стало стандартом нав'язувати реальні рухи акторів на вибраних моделях за допомогою технології Motion Capture, яка дає реальне відображення рухів у віртуальному світі. Motion Builder, підтримуваний 3ds Max, є ідеальним інструментом для цього завдання.

Останнім етапом є експорт у графічний движок та модифікація моделей у ньому. Виробники ігор мають власні графічні двигуни, але є також багато компаній, які пропонують свої двигуни як продукт для використання.

Готовий продукт може бути створений із самого графічного механізму. Однак це ще не кінець шляху розвитку гри. Випускаючи продукт, армія тестувальників стискається, щоб виявити якомога більше помилок перед прем'єрою. Також після прем'єри виявлені помилки систематично виправляються для покращення ігрового процесу.

Це, звичайно, скорочений опис розвитку гри. Цей процес насправді набагато складніший і складніший, що часто займає роки і вимагає величезної роботи.

3.1 Процес створення 3D-моделі для Unity

Не має значення, для якого двигуна ви моделюєте; процес створення 3D-моделей однаковий. Однак спосіб роботи з моделями в різних двигунах різниться.

У будь-якому випадку, коли ви моделюєте для Unity, вам доведеться пройти стандартний конвеєр 3D-моделювання (рис.5).

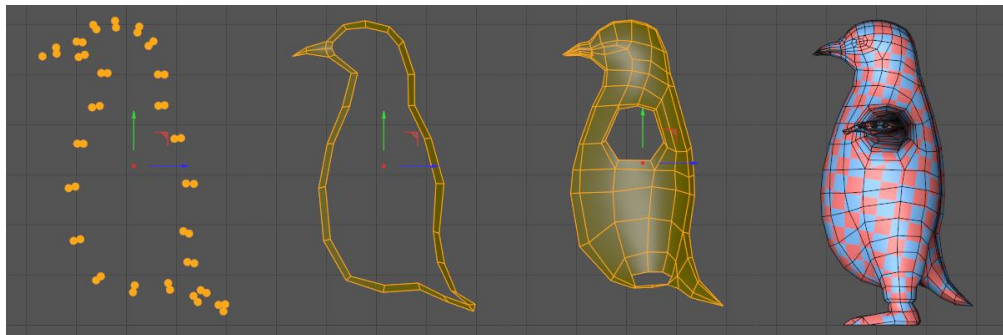


Рисунок 5 – Приклад 3D-моделювання в Unity

3.2 Основні характеристики моделювання

3D-моделювання – це процес створення візуального та цифрового подання реального чи вигаданого об’єкта. 3D-моделювання використовується скрізь у кінематографії, іграх, розробці додатків, архітектурі, автомобілебудуванні, охороні здоров’я, освіті, конфігурації продуктів, продажах та презентаціях.

Для побудови 3D-моделі використовується один із таких інструментів, як Blender, 3Ds Max або Maya. Ви можете створити 3D-модель, використовуючи різні методи:

- моделювання коробки – це техніка тривимірного моделювання, коли ви берете просту фігуру, наприклад куб, циліндр або кулю, як вихідну точку і працюєте в процесі моделювання, поки модель не буде завершена; використання простих фігур як будівельних блоків – це хороший спосіб зробити вашу роботу більш ефективною та зрозумілою.
- моделювання NURBS (нерівномірний раціональний базисний сплайн) – це математичний метод моделювання, який генерує та представляє криві та поверхні; методика моделювання NURBS допомагає художникам 3D використовувати так зване моделювання кривих; 3D-художник може малювати шорсткі лінії, а програмне забезпечення автоматично створює плавні фігури; наступний метод може бути використаний для швидкого створення прототипів моделей шляхом створення основних форм об'єкта.

3.3 Техніка 3D-моделювання в іграх

Техніка 3D-моделювання розподіляється на кілька видів:

- цифрове ліплення – це техніка моделювання, коли художник 3D використовує цифровий пензель для маніпулювання об'єктом 3D (процес схожий на використання глини в реальному житті, але в цьому випадку художник використовує редактор програмного забезпечення для роботи з цифровим об'єктом); метод дозволяє створювати моделі надзвичайно швидко, однак він також робить моделі дуже багатограними та важкими, так що потрібна подальша оптимізація;
- процедурне моделювання – це метод моделювання, коли 3D-художник використовує математичні формули, алгоритми та набори правил для побудови 3D-моделей або середовищ; процедурне моделювання використовується з випадковими деталями об'єктів, такими як дерева, пейзажі, гори, вогонь, вода та інші речовини;

– моделювання на основі зображень або фотограмметрія – це новий спосіб моделювання, який базується на візуалізації моделей із зображень (експерт робить 360° фотографії об'єкта, а потім завантажує їх у програмне забезпечення, яке генерує модель, програмне забезпечення створює 3D-модель на основі зображень, чим чистіші знімки, тим чистішою є модель, часто модель вимагає подальших маніпуляцій та ретопології, щоб зробити її візуально привабливою, гладкою та зручною в роботі).

3.3.1 Характеристика ретопології

Ретопологія – це процес перетворення високополігональних моделей у низькополігональні. Цей крок є невід'ємною частиною процесу постмоделювання, що допомагає спростити роботу з моделями в Unity.

Ретопологія допомагає зменшити кількість багатокутників, щоб оптимізувати модель для мобільних ігор, додатків, ігор для ПК або AR/VR додатків.

Художник може вручну створити нову сітку над існуючою моделлю або використовувати автоматичні або напівавтоматичні інструменти ретопології, такі як Maya, ZBrush або 3D-Coat.

3.3.1 Характеристика УФ-відображення

УФ-відображення – це процес проектування 2D-зображення на поверхню 3D-моделі для текстурування (рис. 6). Іншими словами, художник бере 3D-модель, розгортає її і створює 2D-зображення, на якому легше малювати текстури.

УФ-розгортання виконується до того, як ви застосуєте текстуру зображення до моделі, щоб повідомити програмне забезпечення положення, обертання та масштаб текстури. УФ-відображення може бути сформовано вручну або автоматично залежно від складності моделі, а також від навичок художника.

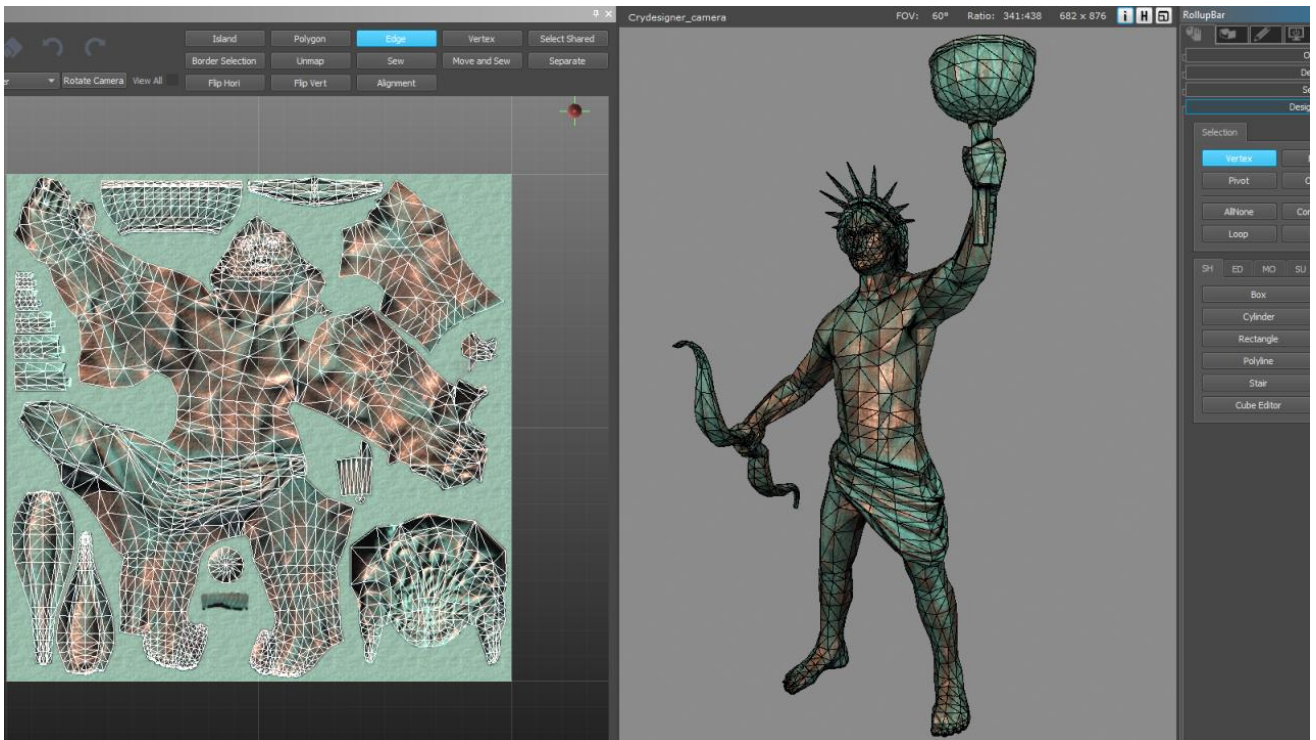


Рисунок 6– Приклад використання УФ-розгортання

Найпопулярнішими програмними засобами для УФ-відображення є Roadkill, Ultimate Unwrap 3D та вбудована функція УФ-картографування в Blender або Maya.

3.3.3 Основні характеристика такелажу

Такелаж – це метод 3D-моделювання додавання кісток до 3D-персонажа або об’єкта для створення гнучких та рухомих частин для анімації (рис. 7).

Метод такелажу використовується скрізь, від анімації обличчя персонажа до відкривання дверей автомобіля. Створюючи цифрову структуру скелета об’єкта, можна легко ним керувати для створення пози пострілу, створення анімаційного фільму або використання його для створення анімованого персонажа в грі.



Рисунок 7– Такелаж

Однак цифровий скелет не рухатиметься з “м’ясом”, якщо не прив’язати його до створеної сітки. Для цього потрібен метод, який називається зняття шкіри.

3.3.4 Опис шкірування

Очищення пов’язує 3D-сітку з кістками (такелаж), щоб можна було переміщати вершини та використовувати свою модель на повну потужність анімації.

Правильна обробка шкіри дуже важлива, оскільки вона допомагає створити плавну анімацію і гарантує, що ваші ноги не затягуються, коли необхідно маніпулювати очима персонажа (рис. 8).

Шкіра забезпечує правильну деформацію моделі. Шкіру можна зробити вручну або автоматично, використовуючи різні функції, такі як Smooth Bind у Maya.

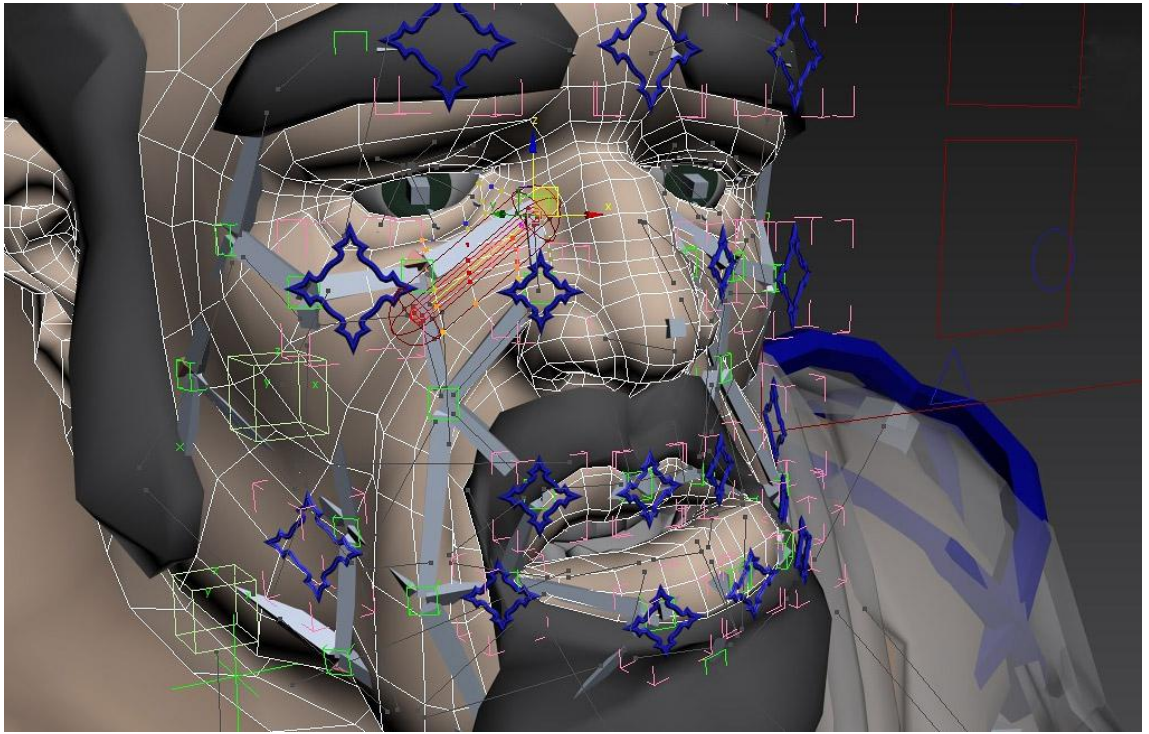


Рисунок 8– Приклад шкірування

3.3.5 Опис текстурування

Текстурування – це завершальний етап процесу моделювання (рис. 9). Для досягнення найкращих результатів текстурування з використанням матеріалів та фактур художник повинен переконатися, що модель була сфальсифікована, оброблена шкірою, ретопологізована та успішно нанесена на УФ.

Існують різні типи методи текстурування, такі як багатотекстурування, карти мап, відображення висоти, відображення нерівностей, нормальне відображення, відображення переміщень, відображення дзеркал та відображення оклюзії, які можна використовувати для створення приголомшливої фотореалістичної або мультиплікаційної графіки, яку можна побачити в додатках та іграх сьогодні.

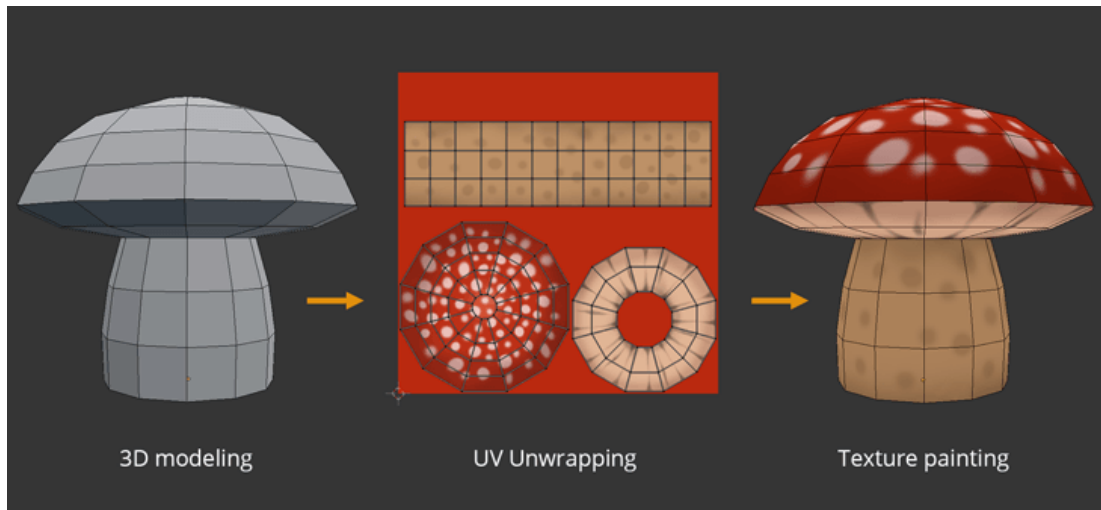


Рисунок 9 – Приклад текстурування

Текстурування зазвичай виконується в Substance Painter, BodyPaint 3D, Zbrush, Mudbox, 3D-Coat або іншому програмному забезпеченні, яке дозволяє художникам створювати реалістичні та класні текстури моделі.

3.4 Робота з моделями в Unity

Незважаючи на той факт, що є можливість тривимірно моделювати в Unity за допомогою ProBuilder, все-таки краще використовувати відповідне програмне забезпечення, яке забезпечує більшу гнучкість та складне моделювання. Це може бути не настільки необхідним на самому початку, але в міру просування дуже здорово вже бути знайомим з професійним програмним забезпеченням. Повний опис з роботи з моделями можна знайти в посібниках Unity.

3.5 Імпорт об'єктів із програмного забезпечення для тривимірного моделювання

Імпорт активів в Unity простіший, ніж здається. Просто треба створити папку в редакторі Unity і перетягнути туди необхідні ресурси. Після цього можете почати працювати з активами в Unity і робити такі речі:

- використовувати стиснення сітки, оптимізацію сітки, збереження квадратиків, формат індексу, зварювання вершин, щоб зменшити

- ресурси та заощадити пам'ять, що ідеально підходить для мобільних і настільних додатків;
- використовувати збереження ієрархій, для збереження анімаційної властивості моделі;
- імпорт матеріалів та текстур;
- робота з повноцінним 3D, ортографічними 3D-сценами або 2D з 3D графікою.

Unity дозволяє по-різному працювати з моделями та досягати чудових результатів, використовуючи вбудовану функціональність від досвідченого розробника Unity та 3D-художника.

3.6 Вбудована підтримка Unity 3ds Max.

Коли вперше працюєш з Unity, одним із перших моментів, що вражає, є твердження, що Unity може завантажувати власні файли 3ds Max. Це здається нездійсненою мрією для художників, і здебільшого здається, що це чудово працює.

Те, що робить Unity при імпорті файлу 3ds Max, – це експорт FBX з Max при завантаженні. Отже, кожного разу, коли необхідно оновити дерево активів у Unity, він робить динамічний виклик 3ds Max, щоб змусити його експортувати всі файли 3ds Max у FBX, а потім завантажує файли FBX у Unity. Зараз це все чудово і для багатьох людей (зокрема, для команд з одним розробником), працює як мрія.

Виконавець створює нову 3d-модель у 3ds Max, зберігає власний файл .max у каталозі Assets / Resources проекту Unity. Unity плавно завантажує його на робочу станцію художника, він проводить години часу, будуючи вигадливу сцену, заселяючи її, освітлюючи і т. д.

3D Max – це специфічне програмне забезпечення для 3D-моделювання, анімації, композиції та рендерингу, і саме ці потужні програми дизайнери пропонують вбудовані інструменти 3D-анімації, моделювання, композиції та рендерингу швидко примножують продуктивність як дизайнерів, так і

художників, і це те, що всі їх версії програми мають однакову технологію та інструменти, але різні версії цих програм орієнтовані на різних типів користувачів.

Деякі включають спеціальні інструменти для розробників відеоігор, графічних дизайнерів та фахівців з візуальних ефектів; тоді як інші версії орієнтовані на архітекторів, дизайнерів та інженерів та містять спеціалізовані функції для цих груп. Але в основному всі версії включають потужні інструменти дизайну, анімації та рендерингу, які виведуть різні твори на інший рівень.

3.7 Вбудована підтримка з БД.

Одна з найважливіших речей в грі - збереження призначених для користувача даних: налаштувань і ігрових результатів. Колись гри були короткими, і зберігати там було особливо нічого. У кращому випадку гра записувала найвищий бал для складання рейтингу, але технології стрімко розвиваються.

Зараз потрібно зберігати тонни даних, включаючи прогрес і статистику гравця. Ця потреба актуальна як для ігор з величезним відкритим світом, так і для простих лінійних пригод. Найчастіше потрібно кілька сесій, щоб завершити гру, значить, потрібен спосіб зберегтися і повернутися пізніше до того ж моменту. Навіть для простих коротких ігор може бути корисно записувати якусь інформацію.

SQLite – компактна вбудована реляційна СУБД, яка є досить таки популярною. Важливий плюс SQLite - це кроссплатформенність, з цього можна використовувати SQLite для різних платформ. SQLite можна використовувати коли потрібна швидкість і компактність, з цього, при виникненні проблеми зберігання даних вирішити її використанням даної СУБД.

Для створення і редагування БД є велика кількість безкоштовних утиліт і плагінів, особисто я використовував DB Browser (SQLite), т.к він

простотою у використанні. Використовуючи SQLite можна спокійно створити таблиці, зробити між ними зв'язку і заповнити їх даними не вдаючись до використання SQL. SQLite может напрямую работать из Unity.

Для створення бази даних в Assets / StreamingAssets проекту (db.bytes, так як Unity розуміє тільки *.bytes для баз даних необхідно використовувати саме це розширення). Для прикладу створена БД з наступними таблицями:

- таблиця «Name», яка описує сутність гравця:

```
CREATE TABLE "Name" (
    "Id" INTEGER NOT NULL,
    "Name" TEXT NOT NULL,
    PRIMARY KEY("Id")
);
```

- таблиця «Level», яка описує сутність рівня гри:

```
CREATE TABLE "Level" (
    "Id" INTEGER NOT NULL,
    "Level" TEXT NOT NULL,
    PRIMARY KEY("Id")
);
```

- таблиця «Time», яка описує сутність часу проходження гри:

```
CREATE TABLE "Time" (
    "Id" INTEGER NOT NULL,
    "Time" TEXT NOT NULL,
    PRIMARY KEY("Id")
);
```

- таблиця «Game», яка введена для підвищення рівня БД

```
CREATE TABLE " Game " (
    "id" INTEGER NOT NULL,
    PRIMARY KEY("id"),
    FOREIGN KEY("Id_Name") REFERENCES "Name"("Id")
    FOREIGN KEY("Id_Level") REFERENCES "Level"("Id")
    FOREIGN KEY("Id_Time") REFERENCES "Time"("Id")
);
```

Для роботи з БД необхідно отримати шлях до неї, потрібно зробити статичний конструктор, який якраз і буде отримувати шлях до БД

```

static Game()
{
    DBPath = GetDatabasePath(); } /
// <summary> Возвращает путь к БД. Если её нет в нужной папке на
// Андроиде, то копирует её с исходного apk файла. </summary>
private static string GetDatabasePath()

{
#if UNITY_EDITOR
    return Path.Combine(Application.streamingAssetsPath, fileName);
#endif
#if UNITY_STANDALONE
    string filePath = Path.Combine(Application.dataPath, fileName);
    if(!File.Exists(filePath)) UnpackDatabase(filePath);
    return filePath;
#endif
#if UNITY_ANDROID
    string filePath = Path.Combine(Application.persistentDataPath,
    fileName);
    if(!File.Exists(filePath)) UnpackDatabase(filePath);
    return filePath; #endif
}

/// <summary> Распаковывает базу данных в указанный путь. </summary>
/// <param name="toPath"> Путь в который нужно распаковать базу
/// данных. </param>

private static void UnpackDatabase(string toPath)

{
    string fromPath = Path.Combine(Application.streamingAssetsPath,
    fileName);

    WWW reader = new WWW(fromPath);
    while (!reader.isDone) { }

    File.WriteAllBytes(toPath, reader.bytes);
}

```

Збереження даних – важлива частина більшості ігор, за винятком лише деяких специфічних винятків (наприклад, ігри з дуже короткими ігровими сесіями або механікою перманентної смерті, в якій після кожного програшу вся гра скидається на початок).

Пам'ятайте про те, що PlayerPrefs – це простий спосіб зберегти настройки та переваги користувача, але його не слід використовувати для

запису важливих ігрових даних. Серіалізація і запис в файл – більш складний шлях, але він дає вам більшу гнучкість і безпеку.

Які саме дані зберігати і яким способом – залежить від особливостей проекту.

4 ОПИС ПРОГРАМНОЇ ЧАСТИНИ

В даній роботі передбачається розробка гри «ЛАБІРИНТ». В якості програми для розробки інтерфейсу вибрано програмний продукт 3D MAX.

Основна задача гри: найти вірний вихід для кульки.

В рамках гри передбачено два варіанти руху кульки:

– при невірному виборі шляху кулька потрапляє в діру та автоматично гра закінчується;

– при правильному виборі шляху кулька знаходить вірний шлях та на фініші змінює колір.

Розберемо поетапно створення інтерфейсу для нашої гри, але спочатку покажемо, що ми повинні отримати (рис. 10):

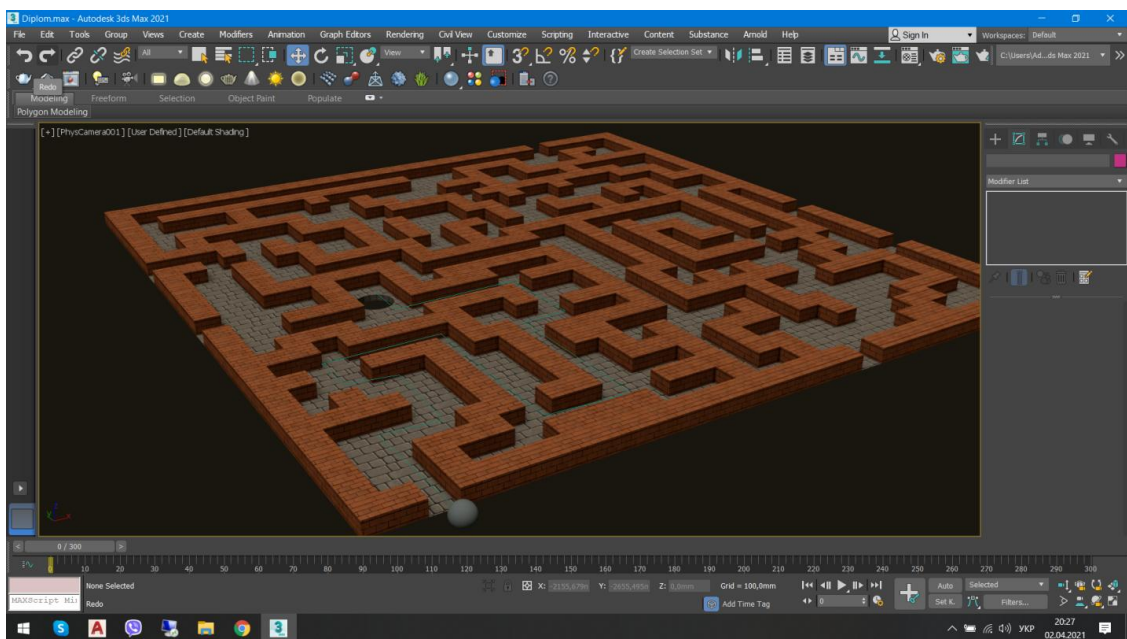


Рисунок 10– Результат гри

Спершу необхідно запустити програму 3D MAX та створити нову сцену (рис. 11)

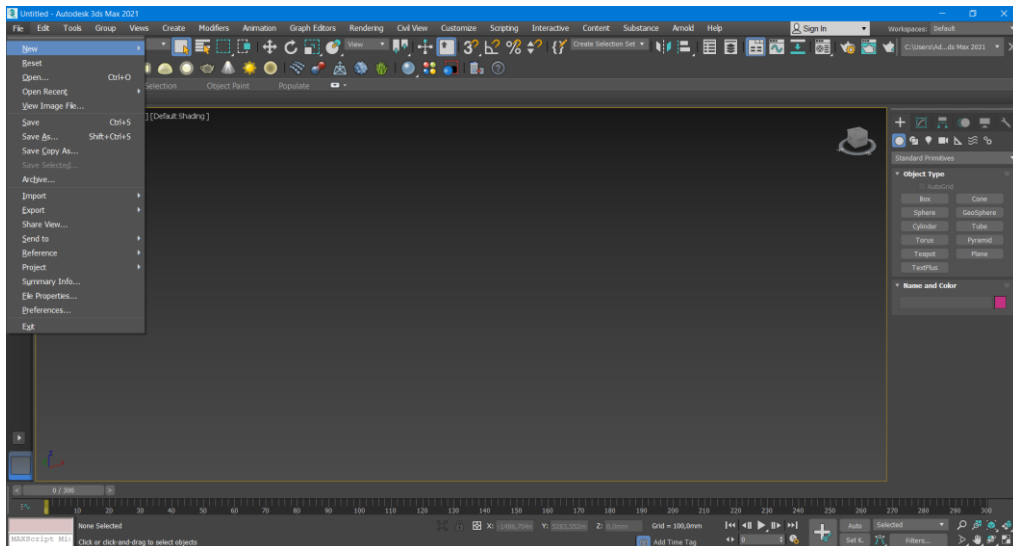


Рисунок 11– Приклад створення нової сцени

Після потрібно створити основні комплектуючі гри: основу, шлях, стіну та шар.

Всі ці елементи ми створюємо за допомогою інструментів (рис. 12):

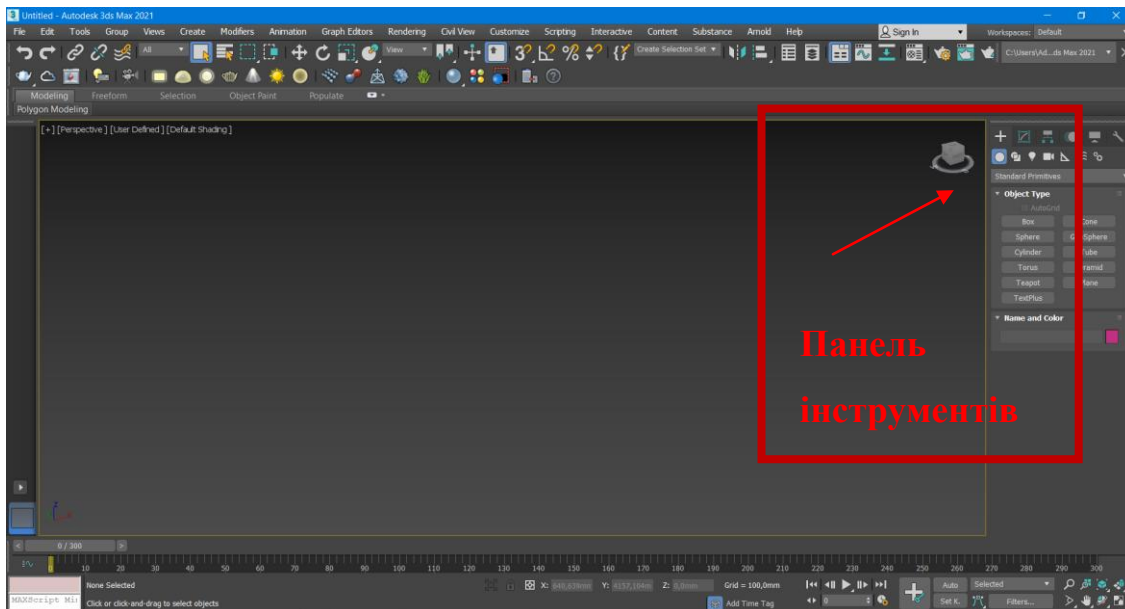


Рисунок 12 – Панель інструментів

На вкладці Geometry: основа – інструмент «Plane», шар – інструмент «Sphere».

На вкладці Shapes: шлях – інструмент «Line», стіна – інструмент «Plane».

Робоча зона програми розділена на 4 види: Top, Front, Left та Perspective. Створюємо елемент в одному із видів, для зручності виберемо Top.

Всі елементи ми створюємо в одному з 4 робочих вікон програми: Top.

На рис. 13 представлений приклад створення основи Plane

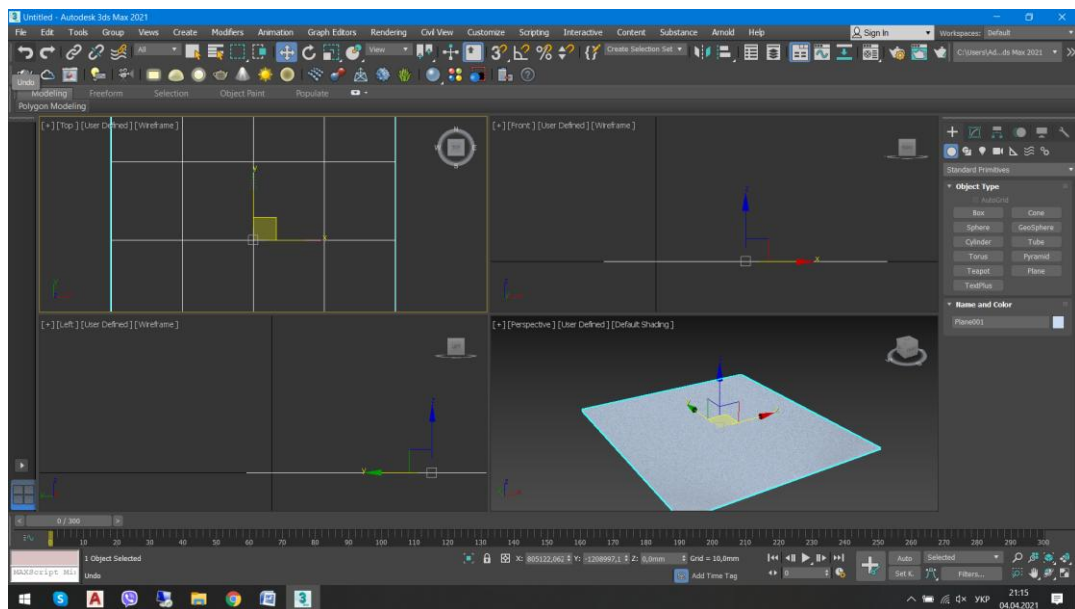


Рисунок 13 – Приклад створення основи Plane

Для створення стін (рис. 14) необхідно план розбити на зони. Для цього в параметрах інструмента Plane вказуємо кількість сегментів на які буде поділено нашу основу. Приймаємо 10x10.

Для створення шару (рис. 15) необхідно використовувати інструмент Sphere. За розмірами шар повинен бути не більше однієї клітини нашої основи.

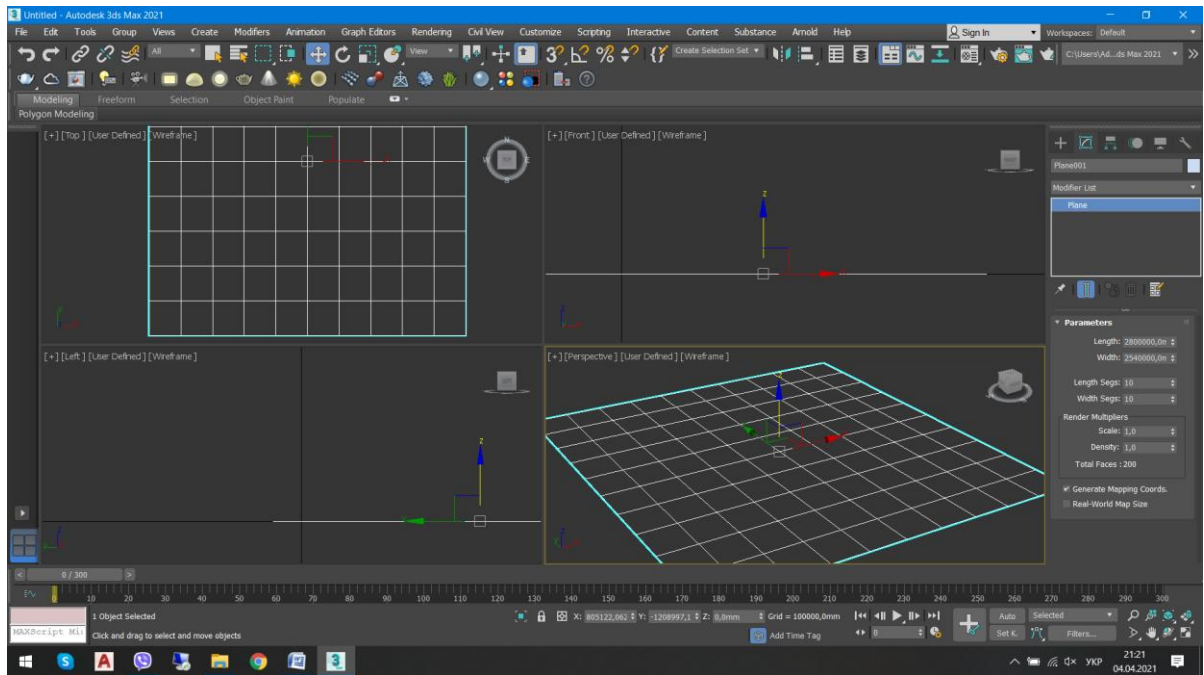


Рисунок 14 – Приклад створення стіні

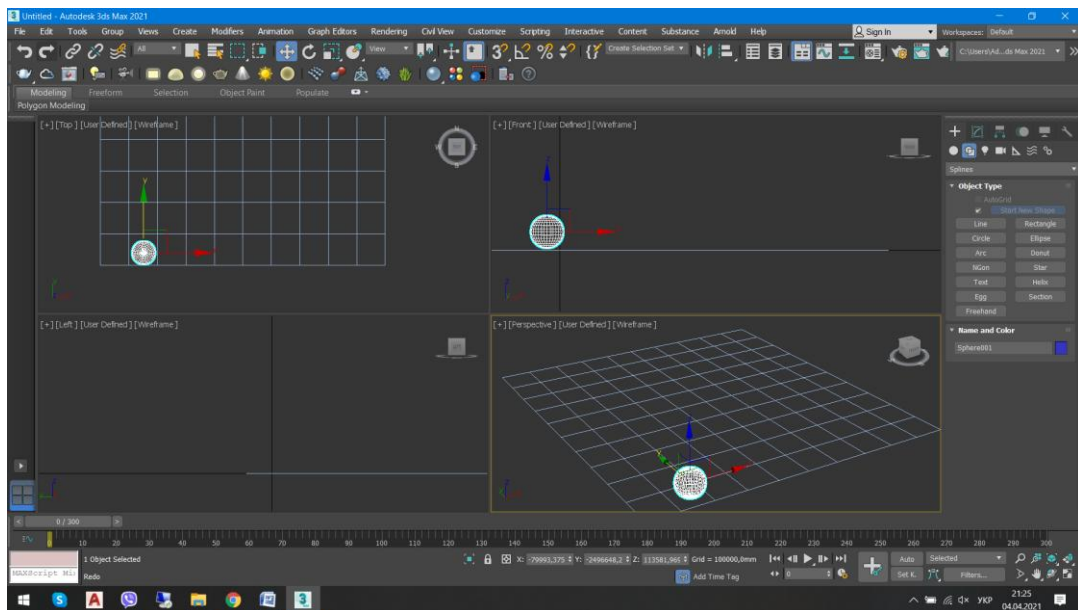


Рисунок 15 – Приклад створення шару

Для створення шляху (рис. 16) використовуємо інструмент Line, та проводимо через всю нашу основу. Прив'язуємо лінію умовно по центру кожної клітинки. В подальшому, по цій траєкторії буде пересуватись, створений раніше шар.

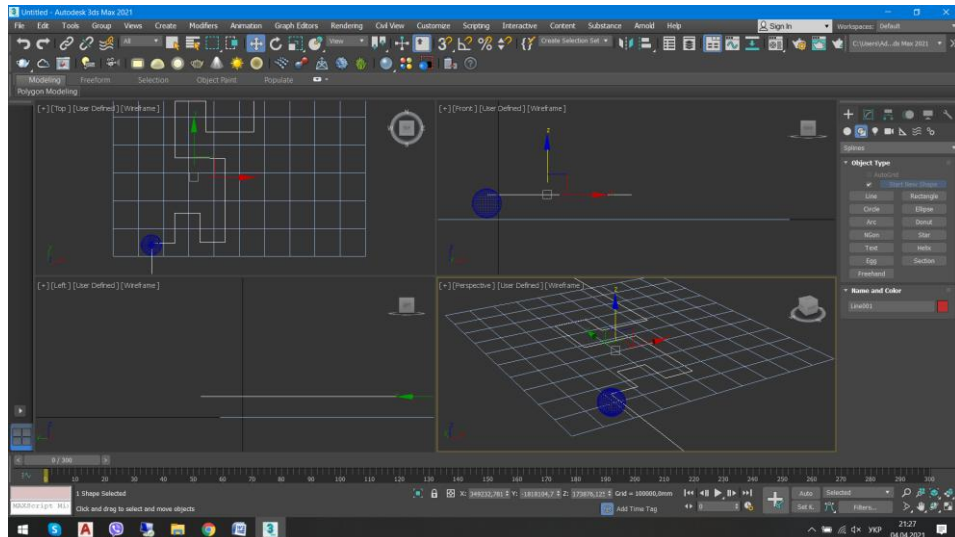


Рисунок 16– Приклад створення шляху

Для створення стіни використано інструмент Plane. Попередньо було розблено основу на сегменти, та вказано за допомогою інструмента Line траєкторію руху шара (рис. 17). Всі вільні від шляху клітинки необхідно буде підіймати та створювати стіну. Для цього необхідно використовувати елемент Plane зробити активним для редагування. Необхідно виділити Plane та натиснути правою кнопкою мишки та вибрати Convert to Editable Poly.

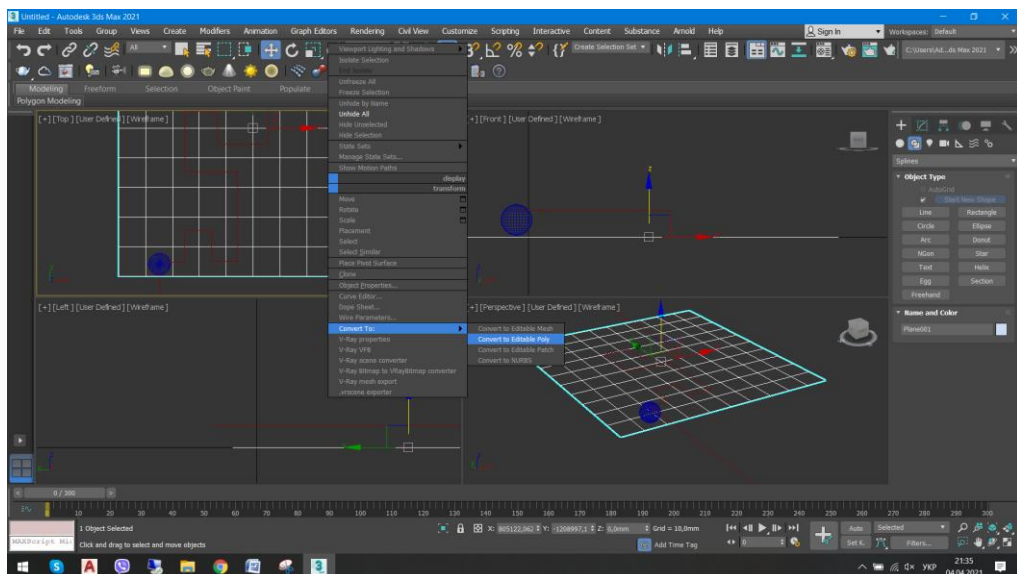


Рисунок 17– Приклад виконання траєкторії

Після на вкладці Modify зі списку Modifier Poly вибрано з Editable Poly – Polygon.

Після цього можна вибирати окремо клітинки на Plane. Необхідно обрати всі вільні від шляху клітинки та застосовуємо для них інструмент «Видавлювання»: Edit Polygons – Extrude та вказуємо на яку висоту ми «Видавлюємо» нашу клітинки. Значення повинне бути по висоті на $\frac{1}{2}$ менше ніж наш шар (рис. 18)

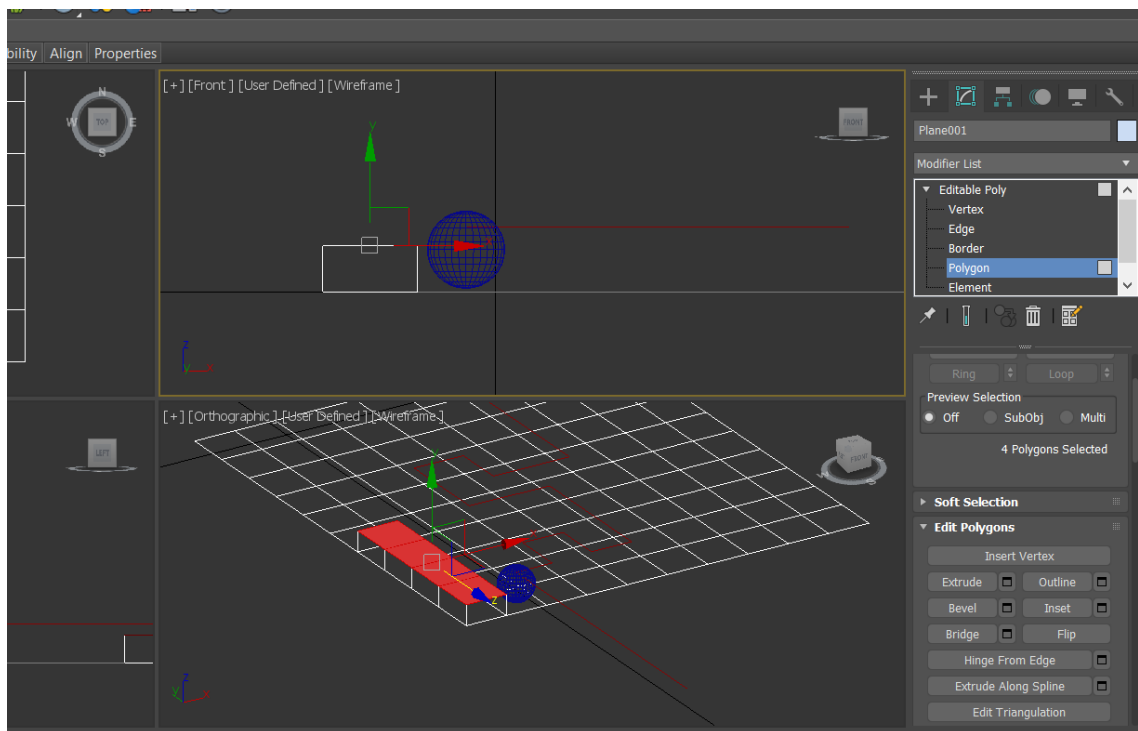


Рисунок 18 – Приклад створення стіни

Після необхідно на одному з невірних шляхів створити отвір, в який при невірному виборі шляху потрапить кулька. Для цього необхідно за допомогою інструмента Cylinder створити циліндр за діаметром кульки, та перенести його на Plane. Після використати інструменти: Compound Objects → Boolean (рис. 19).

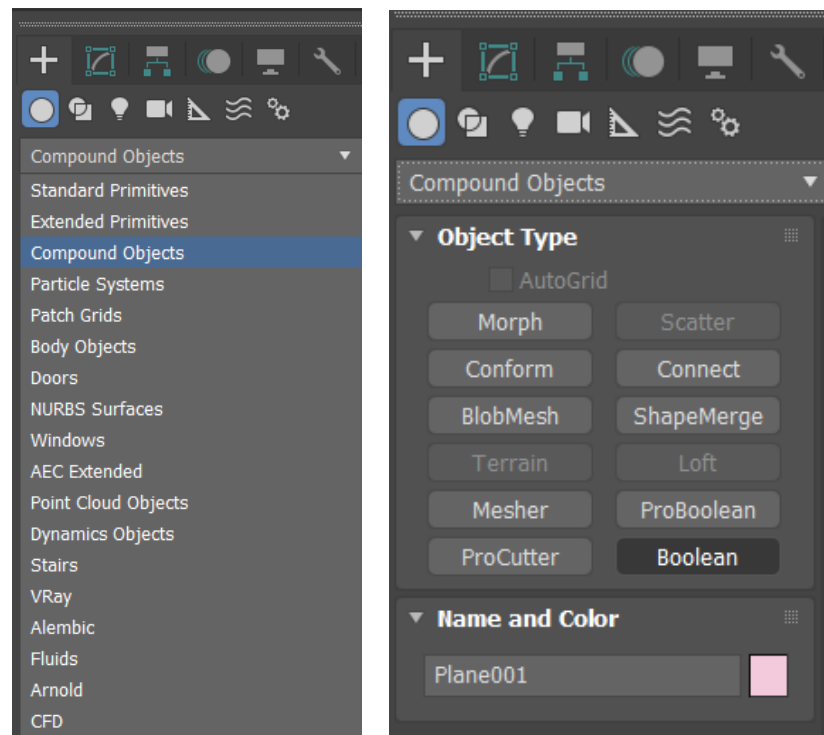

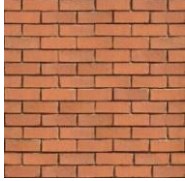
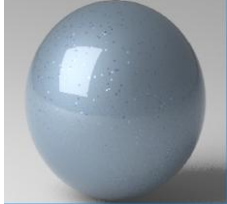


Рисунок 19 – Використання інструменту Boolean

Після створення усіх елементів, необхідно присвоїти їм матеріали (табл. 1).

Таблиця 1 – Присвоєння елементам

Елементи	Матеріал
Основа	
Стіни	
Шар	

Всі матеріали заносимо до редактора матеріалів Material Editor, який знаходиться на панелі інструментів. Після внесення матеріалів до бази, присвоюємо кожному елементу свій (рис. 20).

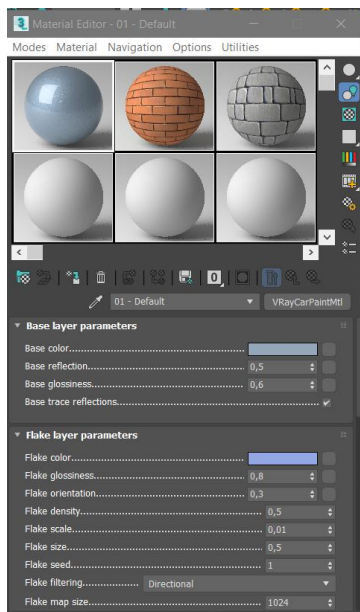


Рисунок 20 – Material Editor

Після присвоєння усіх матеріалів загальна картинка має вигляд (рис. 21).

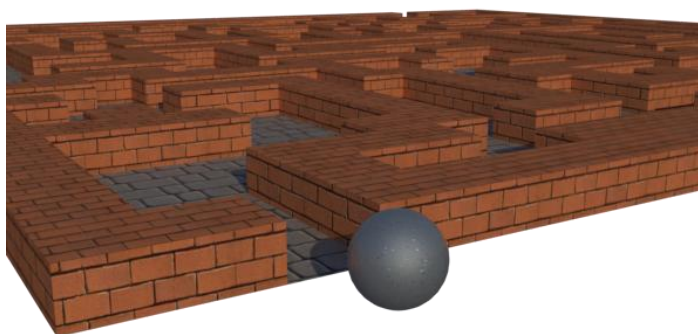


Рисунок 21– Приклад використання Material Editor

Останнім етапом є експорт у графічний движок та модифікація моделей у ньому.

Після цього в першу чергу необхідно просто експортувати вміст в FBX.

За допомогою GameCore можна просто зберегти файли кеш-пам'яті для кожного об'єкта в тому самому місці, що і вихідні об'єкти – «як текстури, так і об'єкти», щоб одразу знати, з якими файлами йшов файл, і, що найгірше, якщо потрібно очистити / видалити або зберегти їх.

Коли необхідно опублікувати остаточну гру, механізм просто пакує файли кеш-пам'яті там, де він є, і ігнорує вихідні файли для цього вмісту.

Зараз Unity має мільярди випадкових незрозумілих кеш-файлів, що зберігаються в папці бібліотек в тому числі я кешовані версії файлів FBX.

Тепер необхідно створити кнопки для запуску програми. Дані будуть створені за допомогою програмного продукту Adobe Photoshop.

Гра починається при натисканні на кнопку нова гра, але початком гри виходить вікно реєстрації ,де необхідно ввести ім'я гравця (рис. 22), для подальшої статистики гри.



Рисунок 22– Реєстрація ігрока

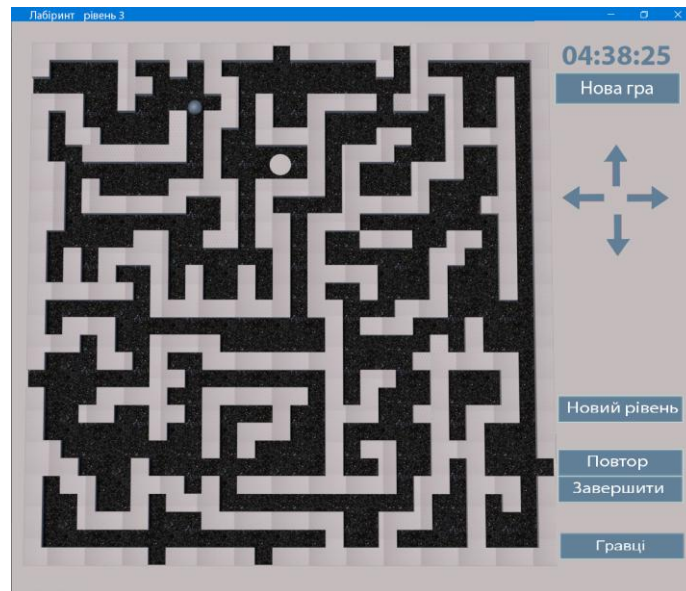
Далі на рис. 23 представлені різні рівні гри.



а)



б)



в)

Рисунок 23 – Приклад різних рівнів гри: а) перший; б) другий; в) третій

При закінченні гри, та виборі правильного маршруту шар змінює колір на червоний.

При не правильному виборі маршруту гра закінчується, та на екрані з'являється напис «Гра закінчена» (рис. 24)

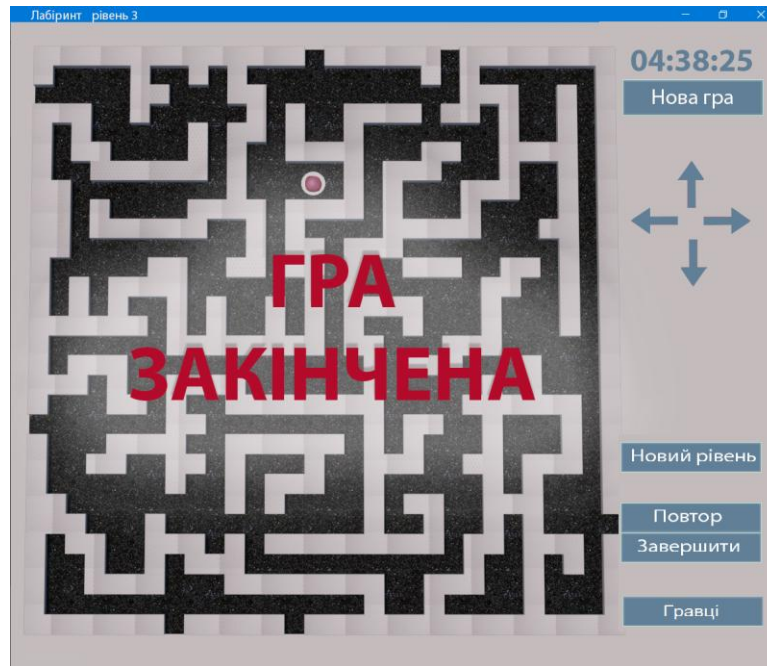


Рисунок 24 – Приклад закінчення гри

Гравці генерують безліч даних, які необхідно обробляти і зберігати, і до яких необхідно отримувати доступ в режимі реального часу – будь то таблиці лідерів, віртуальні товари. Тому для гри потрібно мати БД, де зберігаються всі дані,.

Належним чином структурована база даних:

- допомагає заощадити дисковий простір за рахунок виключення зайвих даних;
- підтримує точність і цілісність даних;
- забезпечує зручний доступ до даних.

Основні етапи розробки бази даних:

- аналіз вимог або визначення мети бази даних;
- організація даних в таблицях;
- вказівка первинних ключів і аналіз зв'язків;
- нормалізація таблиць.

На рис. 25 представлена структура БД для виконання цілосновності даних

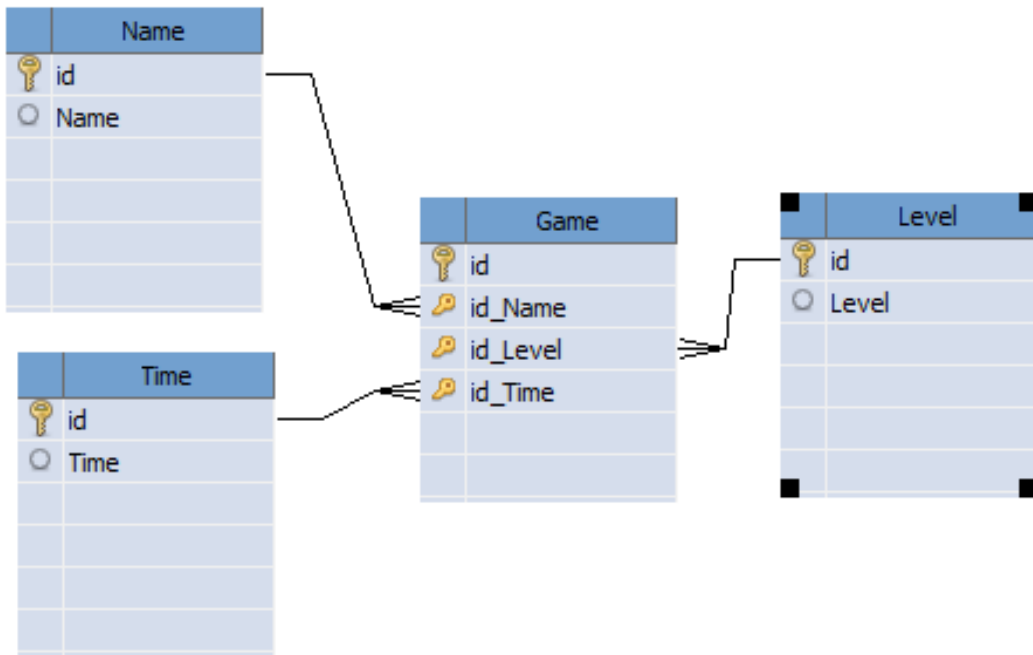


Рисунок 25– Структура БД гри

Дана база зберігає всі дані, та результат гри можна подивитися у вигляді статистики гравців, при натисканні на кнопку «ГРАВЦІ» (рис. 25)

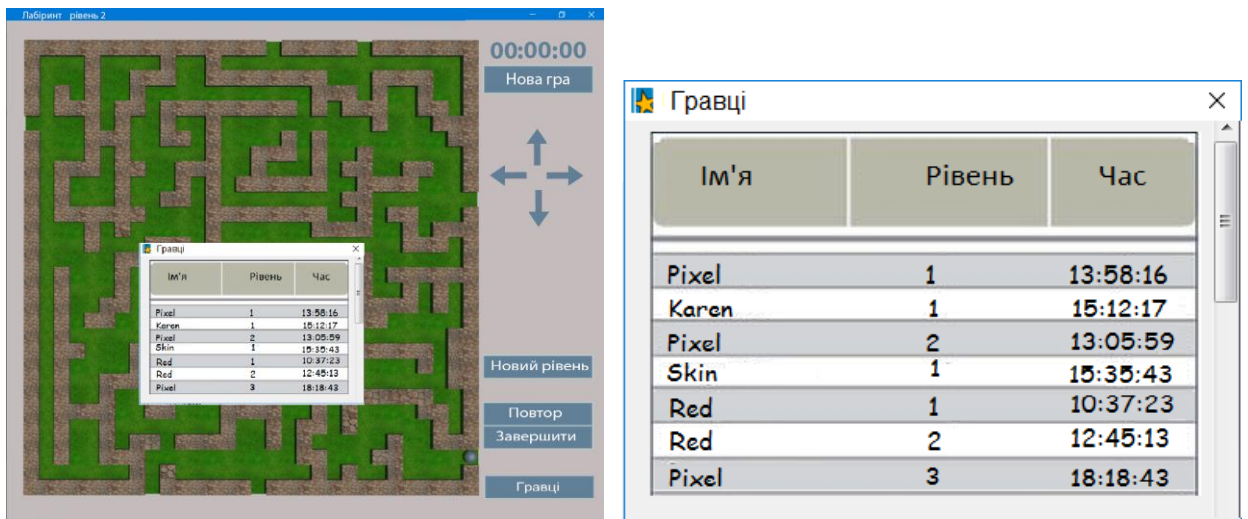


Рисунок 25 – Статистика гравців

Результатом є таблиця, де показано який гравець, за який рівень і за скільки часу пройшов.

ВИСНОВКИ

Я вважаю, що інтерфейс 3Ds Max з більшою кількістю кнопок більше підходить для створення ігор, оскільки ці кнопки можуть змусити людей технічно працювати з швидшими швидкими клавішами та можуть мати дешевшу роботу в невеликій інді-грі з кращою метою. Таким чином, гра може продублювати більш дешевий вміст і заощадити більше грошей.

Метою даної роботи була розробка гри «Лабіринт» ігрової системи для закріплення розвитку мислення.

Зроблено аналіз інструментів розробки і вибрано найбільш актуальне і відповідне для реалізації даного проекту програмне забезпечення:

- Unity 3D – багатоплатформовий ігровий движок;
- Adobe Photoshop – графічний редактор;
- Autodesk 3DS Max – система для створення 3D моделей і анімації моделей;
- SQLite – система управління базами даних.

В ході виконаної роботи були вирішені наступні завдання:

- проведено аналіз існуючих аналогічних рішень;
- сформульовані вимоги до розроблюваної системи;
- спроектовані архітектура і інтерфейс системи;
- спроектована база даних системи;
- реалізовано додаток, що відповідає вимогам;
- реалізований інтерфейс ігрової програми;
- реалізована система квестів ігрової програми;

Розробляється система має наступні перспективи розвитку:

- розробка робочих модулів, таких як система тестування гравців і подальшого їх взаємодії, система кастомізації персонажа;
- розробка нових ігрових рівнів;
- розробка ігрових моделей, їх оптимізація та анімація, і
- впровадження в ігровий процес;

- створення контенту кваліфікованими фахівцями та впровадження їх в ігровий процес;

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1 Жанр компьютерных игр – экшн – Официальный сайт Википедии – свободной энциклопедии. URL: <https://ru.wikipedia.org/wiki/Action>. (Дата звернення 04.03.2021).

2 Жанр компьютерных игр – симулятор – Официальный сайт Википедии – свободной энциклопедии. URL: <https://ru.wikipedia.org/wiki/Симулятор>. (Дата звернення 04.03.2021).

3 Жанр компьютерных игр – стратегия – Официальный сайт Википедии – свободной энциклопедии: <https://ru.wikipedia.org/wiki/Стратегия>. (Дата звернення 05.03.2021).

4 Жанр компьютерных игр – ролевая игра – Официальный сайт Википедии – свободной энциклопедии: URL: https://ru.wikipedia.org/wiki/Ролевая_игра. (Дата звернення 04.03.2021).

5 Игровой движок Unity – Официальный сайт Википедии – свободной энциклопедии. URL: <https://ru.wikipedia.org/wiki/Unity>. (Дата звернення 06.03.2021).

6 Топ-10 игровых движков: выбери свой. URL: https://app2top.ru/game_development/top-10-igrovyy-h-dvizhkov-vy-beri-svoj-45170.html. (Дата звернення 04.03.2021).

7 CryEngine 3 – это CryEngine 2 на средних настройках. URL: <https://3dnews.ru>. (Дата звернення 04.03.2021).

8 Ответы на большинство общих вопросов о HeroEngine и HeroCloud. URL: https://gamedev.ru/community/heroengine/articles/General_questions. (Дата звернення 04.03.2021).

9 Лучшие игровые движки – Движок Unreal Engine. URL: <https://cubiq.ru>. (Дата звернення 04.03.2021).

10 Project Anarchy – обзор, плюсы и минусы программы. URL: <https://8d9.ru/program/project-anarchy>. (Дата звернення 14.03.2021).

11 GameSalad скачать на Windows бесплатно. URL: <https://freesoft.ru>.
(Дата звернения 14.03.2021).

12 GameMaker Studio 2 Mobile в Steam. URL:
<https://store.steampowered.com>. (Дата звернения 14.03.2021).

13 AppGameKit Classic – обзор, плюсы и минусы программы. URL:
<https://8d9.ru> › program › appgamekit-classic. (Дата звернения 14.03.2021).

14 Спрайт Кит против Cocos2D URL: CoderLessons.com:
<https://coderlessons.com>. (Дата звернения 14.03.2021).

14 Unity – всё по этой теме для программистов. URL: <https://tproger.ru> ›
tag › unity. (Дата звернения 14.03.2021).