

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

Розробка 3d гри з використанням машинного навчання

Текстова частина до курсової роботи
за спеціальністю «Інженерія програмного забезпечення» 121

Керівник курсової роботи
доцент
Ющенко Ю.О

(підпис)

“ ____ ” _____ 2020 р.

Виконав студент 4-го курсу
Стахурський Д.В.

“ ____ ” _____ 2020 р.

Київ 2020

Зміст

<u>МЕТА ТА ПЛАН РОБОТИ.....</u>	<u>3</u>
<u>ІГРИ В СУЧАСНІЙ КУЛЬТУРІ.....</u>	<u>4</u>
<u>ЗАРОДЖЕННЯ ТА ЕВОЛЮЦІЯ ВІДЕОІГОР</u>	<u>6</u>
<u>КЛЮЧОВІ РОЛІ ПРИ РОЗРОБЦІ ІГОР</u>	<u>11</u>
<u>РОЗРОБКА СЮЖЕТУ ТА КОНЦЕПЦІЇ ГРИ.....</u>	<u>13</u>
<u>ПОРІВНЯЛЬНИЙ АНАЛІЗ ТА ВИБІР ІНСТРУМЕНТІВ РОЗРОБКИ.....</u>	<u>15</u>
<u>СТВОРЕННЯ КОНЦЕПЦІЇ ТА МОДЕЛЮВАННЯ КАРТИ СВІТУ ГРИ.....</u>	<u>17</u>
<u>РЕАЛІЗАЦІЯ ГОЛОВНОГО ПЕРСОНАЖА.....</u>	<u>21</u>
<u>РЕАЛІЗАЦІЯ ВОРОЖИХ ЮНІТІВ З ВИКОРИСТАННЯМ ЕЛЕМЕНТІВ МАШИННОГО НАВЧАННЯ.....</u>	<u>25</u>
<u>ВИСНОВОК.....</u>	<u>28</u>
<u>ДЖЕРЕЛА</u>	<u>28</u>

Мета та план роботи

Метою даної роботи є створення прототипу 3d гри, використовуючи при цьому сучасні інструменти для розробки, та механіки актуальні на сьогоднішній день. Прототип гри також передбачає реалізацію ворожих персонажів з використанням елементів машинного навчання та симуляції штучного інтелекту. В процесі розробки гри, крім того заплановано ознайомитися та використати архітектурний патерн Finite State Machine. У тому числі заплановано ознайомитися з роботою анімаційних та графічних компонентів, і роботи ігрових двигунів в цілому. Також передбачається використання зовнішніх сервісів, для зберігання даних про ігровий процес користувача.

Проаналізувавши принципи та методологію розробки 3d ігор, було прийнято рішення розділити роботу на наступні етапи, а саме:

- Розробка концепції та сюжету гри
- Порівняльний аналіз та вибір інструментів розробки
- Створення концепції та моделювання карти світу гри
- Реалізація головного персонажа
- Реалізація ворожих юнітів з використанням елементів машинного навчання
- Розробка ворожих юнітів з використанням скінченних автоматів

Ігри в сучасній культурі

На сьогоднішній день ігри можна вважати новою медіа-субкультурою. За останні десятиліття популярність відео ігор експоненціально зросла, все більше впливаючи на сучасну культуру. Культурно відео ігор паралельно розвивалася та еволюціонувала, пліч о пліч з інтернет культурою, стаючи дедалі популярнішою. Люди котрі грають у відеоігри, ідентифікуються як геймери, що може означати що завгодно - від людей які просто цікавлять іграми, як хоббі і проводять у іграх свій вільний час до людей, котрі займаються цим професійно. Таких людей можна класифікувати як стримери, або ж власне кіберспортсмени. В сучасних реаліях бути кіберспортсменом є доволі престижно, що формує велику конкуренцію між гравцями, і стимулює все більший інтерес з боку молоді.

Впродовж останніх п'яти років, середній вік гравців по всьому світі приблизно становить 35 років. Це число повільно зростає оскільки, люди котрі в дитячому віці грали в перші аркадні, комп'ютерні та консольні ігри, продовжують грати і сьогодні, користуючись сучасними рішеннями. Також варто зазначити, що зростає кількість гравців старшого віку, котрі зацікавилися іграми, за допомогою засобів масової інформації, або своїх молодших членів сім'ї. Однією з важливих причин збільшення гравців можна пояснити, зростаючою кількістю жанрів, котрі потребують меншої кількості аудиторії. В приклад можна навести ігри, котрі передбачають велику кількість рухових дій користувача, такі як симуляції гри в теніс і тд.

Стрімкий розвиток ігрової субкультури теж доволі сильно стимулювала кіноіндустрія та аніме індустрія. Впродовж останнього часу було створено неймовірно велику кількість фільмів, котрі пропагандують субкультуру геймерів та заохочують вникнути у віртуальний світ. Також з'являється все більше і більше фільмів, сюжет яких взятий з відеоігор. Одним з найновіших популярних фільмів, котрий базувався на однойменній серії ігор був Warcraft-Початок.

Протягом останніх десятиліть не затихають дискусії стосовно впливу відеоігор на підростаючі покоління. До негативних наслідків вчені відносять можливість розвитку депресії, або ж зростання агресивної

поведінки. Однак провівши більш поглиблені дослідження і розглянувши дану тему більш різносторонньо і збалансовано, можна побачити наступну картину. Незважаючи на те що деякі науковці стверджують, що відеоігри є інтелектуально лінивими і спричиняють деградацію, ігри насправді можуть зміцнити і покращити цілий ряд когнітивних навичок, таких як навігація у просторі, реакція, швидкість прийняття рішень, пам'ять. Одними з таких ігор є шутери, котрі доволі часто критикують за їхню жорстокість. Одне з досліджень показало, що гра у відео-ігри шутери значно покращила навички та здатність гравця працювати в команді, а також продумувати і реалізувати більш складні та доволі часто нестандартні моделі поведінки в процесі гри. Також гра у відеоігри стимулює розвиток критичного мислення у дітей.

Велика кількість підлітків запевняють, що гра в стратегії та інші рольові ігри, дозволяє їм краще вдосконалюватися у вирішенні проблем, та покращує їхню успішність загалом. Також варто зазначити, що деякі ігри загалом сприяють покращенню настрою гравців, знімають стрес та напругу. Ще одним стереотипом стосовно ігор, є те що сучасні геймери є соціально ізольованими людьми. Проте сучасні дослідження стверджують, що більш як 80 відсотків гравців, грають з другом в команді. Також приблизно 40 відсотків таким чином соціалізуються і знаходять друзів сам в іграх. Багатокористувацькі ігри стають цілими спільнотами, і доволі часто потрібно приймати рішення про те, кому довіряти, а кому не слід. Також в процесі цього розвивають лідерські якості, оскільки потрібно вміло керувати групою. Деякі ігри спеціально пропагандують соціальну взаємодію між гравцями та спонукають їх об'єднуватися в команди, розробляючи сюжет та квести чином, що гравець просто фізично не може впоратися самотужки.

Зародження та еволюція відеоігор

Зародження ігрової культури можна датувати 50-ми роками минулого століття, проте в той період в них грала обмежена кількість людей. Засновниками ігор, в цілому можна вважати студентів, котрі вели дослідження в комп'ютерних лабораторіях великих університетів, таких як МІТ, або ж працівників військових об'єктів таких як Бруксейвенська національна лабораторія. Перші ігри такі як Космічна війна випущена в 1962, або ж Таємнича печера створена в 1976, були доволі простими і не містили, а ні сюжету, а ні графіки. Дані ігри відображалися на примітивних екранах з мінімальною роздільною здатністю. Після гри SpaceWar, в науковій лабораторії Університету Юти, в 1971 році було створено першу аркадну відеогру Computer Space. Після цього наприкінці 1970-х років, в США активно почали з'являтися перші аркадні автомати.

Одними з найпопулярніших аркадних ігор можна вважати Астероїди та Зоряний Замок. Дані ігри були реалізовані за допомогою векторної графіки. Векторну графіку можна охарактеризувати як точки на декартовій площині, котрі з'єднані лініями та кривими, що дає змогу створювати багатокутники та інші фігури. Важливою перевагою векторної графіки над растровою, є те що всі точки та криві можуть змінювати свій розмір для будь -якої роздільної здатності. Дані точки також визначають напрямок векторного шляху. Кожен відрізок може мати унікальні властивості, а саме: колір, кривизну, товщину заливки та форму. Наприкінці 80-х, в більшості іграх почали використовувати піксельну графіку. Найменшим і ключовим елементом в піксельній графіці є піксель. Піксель це елемент, котрому присвоюється спеціальне значення кольору. Основною характеристикою, що використовується для характеристики піксельної графіки, є кількість використаних пікселів для відображення картинки, та глибину кольору цих пікселів. Глибина кольору визначає яскравість для кожного елемента зображення. Завдяки цим властивостям, за використання растрової графіки можна реалізовувати дуже деталізовані зображення. Враховуючи, те що піксельна графіка складається з фіксованої кількості елементів, тому як якість так і обсяг пам'яті необхідний для зберігання, залежить від кількості пікселів. Через взаємозалежність між якістю зображення та кількістю пікселів, піксельну графіку не можна масштабувати без втрат якості.

На початку 80-х років значної популярності набуло три типи аркадних автоматів, а саме “Arcade table”, “Cocktail table” та “Arcade Cockerit”. У середині 1980-х років ігрова індустрія почала захоплювати ринок. Ігрові жанри та механіки стали більш різноманітними. Аркадні шафи також не відставали і були модернізовані, більш складними та реалістичними контролерами. В 1990-тих роки велика кількість аркадних ігор почала нагадувати міні атракціони. Було реалізовано можливість битися з віртуальними ворогами, використовуючи фактичні удари ногами та руками. Однією з найвидатніших аркад того часу був “Віртуальний Світ”. Попри свій успіх, аркадні ігри займали дуже багато місця і були дорогими в обслуговуванні. Наприкінці 1990-х років домашні системи почали конкурувати між собою та з часом обігнали по графіці та обчислювальній здатності аркадні ігри. На цьому і завершилася золота ера автоматів з іграми. Після цього аркадні автомати почали виходити із вжитку, будучи заміненими більш сучасними іграми, котрі були більше орієнтовані на навчання гравця. Після ліквідації дані автомати активно потрапляли в руки приватних колекціонерів. На зміну ігорим автоматам прийшли консолі - ігрові платформи, котрі можна використовувати вдома. Використовуючи мікропроцесор, відбувалася передача сигналу на дисплей користувача. Перевагою консолі було те, що достатня кількість кнопок, тригерів та аналогових елементів керування дозволяє зручно грати в різні ігри. Також на відміну від виділених материнських плат в ранніх аркадних автоматах, в яких могла вміститися лише одна гра, консольні приставки використовували CD та DVD носії, що дозволяло грати безліч різних ігор на одній консолі. Одним з прямих нащадків аркадних ігор можна вважати перші кишенькові приставки. Дані приставки мали вбудований дисплей, процесор і контролер, проте будучи достатньо малими щоб розміститися в руках гравця. Значним недоліком перших таких приставок було те, що вони могли відображати лише одну гру.

За рахунок того, що наприкінці 1970-х років, персональні комп'ютери стали стрімко поширюватися серед звичайного населення, програмування та відтворення ігор на комп'ютері стало стрімко поширюватися. Це породило ціле покоління розробників ігор, котрі починали свій шлях, намагаючись наслідувати та модернізувати ігри, котрі використовувались в аркадних автоматах. Перші такі ігри зберігалися на дискетах, котрими активно обмінювалися. Також варто зазначити, що більш

гнучкий функціонал клавіатури дозволив користувачам вводити більше інформації, що сприяло створенню унікальних жанрів ігор. Прикладом цього може бути текстова пригодницька гра, де сюжет розгортався від вибору користувача. Дані нововведення передбачали собою тривалий період часу проведений за комп'ютером і відповідно більшу кількість часу проведений в іграх і відповідно більший ігровий досвід. В той період часу значної популярності набули стратегії. У міру того як розвивалося апаратне забезпечення комп'ютера, комп'ютерні ігри також еволюціонували, ставши більш деталізованими та складнішими. За рахунок миші, як важливого елемента взаємодії з комп'ютером, зародився і набув популярності такий жанр ігор, як шутер від першого лиця.

Протягом багатьох років, відбулося розділення ігор на безліч різноманітних жанрів і піджанрів. Власне термін ігровий жанр використовується для опису стилю ігрового процесу.

До найпопулярніших ігрових жанрів можна віднести:

- Action - це гра де гравець знаходиться в центрі подій, і безпосередньо керує своїм персонажем, виконуючи різноманітні дії та задачі
- Platformer - даний жанр ігор отримав свою назву завдяки тому, що персонаж в процесі гри взаємодіє з платформами. Зазвичай це відбувається за рахунок бігу, стрибків, і падінь, збираючи в процесі монети, що використовується як внутрішньо ігрова валюта і в деяких іграх вбиваючи ворожих опонентів. Однією з найпопулярніших ігор цього жанру є "Маріо".
- Shooters - надають змогу гравцям використовувати різні види зброї, для участі в боротьбі. Здебільшого мета таких ігор є знищення ворожих мобів, або гравців. Шутери класифікують за відображенням персонажа гравця. Шутери від першої особи дозволяють, умовно кажучи дивитися на світ очима персонажа. У шутерах від третьої особи користувач повністю бачить модель свого персонажа і керує ним переважно зі спини.
- Fightings - фокусують свою увагу на битвах переважно один на один. Здебільшого дані битви проходять з використанням рукопашного

бою. В більшості файтингів є велика кількість збалансованих і різносторонніх персонажів, кожен з яких має свої унікальні здібності та стиль бою. У більшості ігор, гравці пробиваються до вершини, долаючи дедалі сильніших противників по мірі прогресу.

- Beat-em-up - ігри в яких гравцю необхідно подолати хвилі ворогів, котрі намагаються захопити локацію персонажа.
- Stealth - в цьому жанрі гравцю необхідно продемонструвати свою хитрість і кмітливість в вирішенні ігрових завдань. Стелс-ігри зазвичай спонукають гравців приховано брати участь у дії.
- Survival - жанр, котрий набуває активно популярності. Події в таких іграх переважно відбуваються в середовищах з відкритим світом де гравцю необхідно самотужки зібрати необхідні ресурси, або зброю, для виживання. Переважно в такій категорії ігор переможець тільки один, і на шляху до перемоги гравцю необхідно знешкодити усіх ворожих опонентів.
- Action-adventure - це жанр ігор де здебільшого використовують дві ігрові механіки, а саме довгострокові квести, або перешкоди, котрі необхідно подолати збираючи різноманітні інструменти, або артефакти, а також містять елементи сюжету, де ці предмети потрібно застосувати. В цих іграх гравець змушений більше зосередитися на дослідженні, вирішенні головоломок та пошуку правильного шляху, тоді як базовий бій це більше допоміжна діяльність для розвитку сюжету.
- Action RPG - приймає ігрові елементи як екшн-ігор, так і екшн-пригодницьких ігор. Визначальною характеристикою, є те що бій відбувається в режимі реального часу і значною мірою залежить від персональних навичок та ігрового досвіду гравця
- Strategy - в цьому ігровому жанрі, геймплей заснований на традиційних настільних іграх. В стратегіях гравець отримує богоподібний доступ до світу та його ресурсів. Проте це змушує гравця ретельно підходити до створення власної стратегії та тактики для подолання зовнішніх перешкод.

- Multiplayer online battle arena - можна класифікувати як піджанр стратегічних ігор, де гравці переважно не створюють власні ресурси, такі як бази, або бойові одиниці. Замість цього гравці контролюють одного персонажа, котрий є членом однієї з двох команд. Працюючи разом команда намагається подавити іншу команду, обмежити її в розвитку, і в її кінцевою метою є знищення бази противників.
- Sports games - імітують різноманітні види спорту, такі як футбол, баскетбол, бейсбол і тд. Супротивники в такому жанрі ігор здебільшого реалізовані алгоритмічно, або з використанням штучного інтелекту.

Ключові ролі при розробці ігор

Подібно до того, як існує багато жанрів і різновидностей ігор, існує багато позицій розробників, котрі займаються їхньою реалізацією. На початку зародження відеоігор, ігри створювалися конкретними людьми. Прикладом цього може послужити перша частина гри “Принц Персії”, котра була створена одним розробником, котрий самостійно займався написанням коду, розробку анімацій та навіть музичним супроводом. Згодом з’явилася необхідність формувати команду з декількох розробників, оскільки комерційна розробка ігор стала більш важкою і трудоємкою. З ростом попиту та більше акцентування уваги на графічній складовій, до команд розробників також доєдналися художники. Спочатку ідеї та сюжет гри створювався учасниками команди, проте згодом коли зросла цінність ігрового сюжету та з’явилася необхідність формувати неочікувані та цікаві сюжетні повороти, з’явилася така посада як гейм-дизайнер. Попри те, що члени сучасних команд, по розробці ігор можуть одночасно займатися різними елементами процесу створення гри, конкретна спеціалізація стає все більш необхідною, оскільки ігри стають більшими, складнішими та потребують більше ресурсів для реалізації. В стандартній команді по розробці гри можна виділити наступних учасників:

- **Programmer** - використовуючи мови програмування програміст займається розробкою коду, котрий відображає текст та графіку гри. Також до його обов’язків входить розробка системи управління персонажем, що дозволяє гравцю взаємодіяти з грою, створює фізичну систему, що підпорядковує собі гравця та ігровий світ. Також до обов’язків програміста входить розробка систем камер та їхнього налаштування, для візуальної взаємодії. Програміст розробляє системи AI для управління ворожими юнітами, та їхньої взаємодії як зі світом гри так і з персонажем.
- **Artist** - з появою потужних комп’ютерів, в іграх почала з’являтися тривимірна графіка, що спонукало залучення професіоналів, котрі можуть реалізовувати складні графічні елементи. Художники-концепціоналісти використовують як традиційні інструменти так і комп’ютерне забезпечення для малювання персонажів, ворогів та ігрового світу в цілому. Після цього інші художники займаються моделюванням та створенням 3d моделей, використовуючи такі засоби

розробки як Maya та 3D Studio Max. Художники, котрі займаються розробкою текстур буквально малюють на 3d моделях, роблячи текстури та моделі більш реалістичними. Також до обов'язків художників можна віднести створення візуальних ефектів, використовуючи поєднання 2d та 3d технологій. Аніматори роблять персонажа гравця живим і створюють заставки, подібно до анімацій у високобюджетних анімаційних фільмах.

- **Designer** - робота геймдизайнера полягає в розробці ідей, та концепції гри, а також розробкою та продумуванням сюжету. Одна з головних задач дизайнера відрізнити хорошу концепцію від поганої. До обов'язків дизайнера входить створення умовних карт світу, їхня початкова реалізація за допомогою спеціальних програм. Дизайнери рівнів відповідають за наповнення тої чи іншої локації, починаючи від ворогів, і босів закінчуючи секретними кімнатами і скарбами. Системні дизайнери відповідають за розробку і модернізацію балансу в грі. Також до їхніх обов'язків входить реалізація різноманітних дерев навичок персонажа, та реалізація внутрішньої економіки в грі. Бойові дизайнери спеціалізуються на впровадженні коректного балансу в процесі битв персонажа, роблячи їх більш цікавими та видовищними.
- **Producer** - в обов'язки продюсера входить здебільшого організація процесу розробки гри. В його обов'язки входять такі дії як формування та створення команд по розробці, укладення необхідних контрактів, формування та збалансування бюджету гри, вирішення суперечок в процесі створення гри, а також представлення гри та команди серед вищого керівництва. Продюсера можна вважати, як першим так і останнім членом команди, котрий фактично є публічним обличчям створюваної гри.
- **Tester** - дана позиція є дуже важливою і потребує великої витримки та хорошого розуміння ігрових механік. Робота тестера має вирішальне значення в процесі розробки гри оскільки від цього залежить як успіх гри серед користувачів так і підтримка іміджу компанії розробника. До його обов'язків входить усе можлива перевірка всіх наявних механік в грі, починаючи від успішної авторизації для користувача і закінчуючи тестуванням битви між гравцем та фінальним антагоністом.

Розробка сюжету та концепції гри

Вибір жанру при створенні гри передбачає собою один з найважливіших етапів, на стадії створення концепції, оскільки це дуже сильно впливає на майбутній успіх гри починаючи, з актуальності гри на сьогоднішній день та чисельності зацікавленої аудиторії і закінчуючи, складністю реалізації та необхідною кількістю ресурсів. Проаналізувавши сучасні тенденції було прийнято рішення розробити гру в жанрі AAAA, котра буде називатися “Magic Maze”.

Сюжет даної гри полягає у тому, що головний персонаж потрапляє у зачароване підземелля, кожен рівень якого уособлює собою унікальний лабіринт наповнений монстрами та різноманітними скарбами. Перед користувачем постає задача, подолати кожен з поверхів підземелля, розвиваючи свого персонажа, шляхом збільшення його рівня в грі. З підвищенням рівня персонаж буде отримувати нові навички та додаткові характеристики.

Враховуючи, те що знайти вихід з лабіринту може бути доволі складно, передбачено наявність декількох магічних порталів, котрі можуть перемістити персонажа в безпечну локацію, таким чином гарантувавши проходження рівня. Проте не все так легко, оскільки самі портали будуть оберігатися великою кількістю монстрів, і щоб їх пройти необхідно буде скористатися силою і подолати усіх ворогів. Також для урізноманітнення сюжету передбачено декілька типів порталів:

- зелений портал, на страх і ризик гравця зможуть перемістити його в невідому частину лабіринту. Даний лабіринт, корисний у випадку коли за гравцем женеться велика кількість монстрів.
- синій портал, як зазначалося вище буде телепортувати гравця в безпечну локацію, і передбачатиме кінець рівня.
- червоний портал, буде всього один на поверх і зможе перемістити гравця в таємну локацію. Дана локація буде одночасно наповнена як скарбами і цінними предметами, так і дуже сильними монстрами, котрих персонажу доведеться перемогти

Також на кожному рівні передбачений свій унікальний бос, котрий буде генеруватися в випадковій локації лабіринту, безпосередньо перед гравцем, або ж буде патрулювати лабіринт. Поведінка боса буде варіюватися відповідно до складності поверху. На деяких поверхах, гавцю буде необхідно знайти певний предмет для того, щоб вбити боса і отримати додаткові бонуси.

Гравець також може повторно повертатися на пройдені рівні, для того щоб знайти той чи інший бажаний предмет, або ж задля знищення монстрів, що дасть змогу підвищити свій рівень.

В грі також буде присутня внутрішньо ігрова валюта, котру гравець буде отримувати за вбивство монстрів та проходження лабіринту. Це надасть можливість користувачеві купувати необхідні предмети в магазині.

Порівняльний аналіз та вибір інструментів розробки

Потрапляючи в ігрову індустрію, новачки часто задаються питанням, який ігровий двигун кращий, і чому. З точки зору, універсальності, потужності, популярності та розміру користувачького ком'юніті, виділяють два такі двигуни як Unity та Unreal Engine.

Дехто вважає, що Unreal Engine кращий з тої причини, що велика кількість студій користуються ним для реалізації AAA - ігор. Інші стверджують, що Unity є кращим рішенням, оскільки є більш гнучким і використовується в різноманітних сферах, починаючи рендерингом фонів і різноманітних об'єктів, в процесі створення кінофільмів і закінчуючи машинним навчанням.

Варто зазначити, що обидва двигуни підтримують доволі велику кількість різноманітних ігор, проте можна виокремити деякі відмінності:

- 3D ігри - обидва двигуни мають широкий спектр можливостей в даній сфері, проте з точки зору графічної реалізації, Unreal Engine має перевагу.
- 2D ігри - обидва двигуни підтримують розробку, проте Unity підходить краще, завдяки великій кількості розширень і вбудованих інструментів.
- Віртуальна реальність - Unity домінує в даній галузі, оскільки містить більшу кількість плагінів, котрі легко інтегруються в загальну інфраструктуру AR.
- Розширена реальність - обидва двигуни користуються попитом, але завдяки довшому перебуванні на ринку і більш досконалим інструментам Unity домінує.
- Multiplayer - обидва двигуни надають змогу реалізовувати багатокористувачький режим, проте Unreal на відміну від Unity має інтегровану підтримку
- Мобільні ігри - на даному ринку, Unity вважається безпрецедентним лідером

В Unity розробка скриптів здійснюється, за допомогою мови C #, тоді як в Unreal використовують C++. Попри те що, C ++ вважається більш складною для вивчення мовою, вона є дуже гнучкою і надає можливість для ефективної оптимізації коду. Також Unreal має власний вбудований візуальний скрипт, який називається Blueprints. Даний скрипт дозволяє, без використання програмування, створити вузли, котрі можна з'єднувати та комбінувати, щоб об'єднати логіку та елементи розроблюваної гри.

Unity - можна вважати, найпопулярнішим двигуном для незалежних розробників та галузі мобільної розробки. За допомогою Unity також було розроблено ряд великомаштабних ігор, наприклад таких як Hearthstone та Rust. Проте, все ж таки, в AAA-галузі, Unreal користується більшим попитом, ніж Unity.

Ще одним важливим аспектом, є користувацька спільнота. На сьогоднішній день, спектр навчальних матеріалів, форумів та ресурсів набагато більший у Unity, що в свою чергу спонукає ще стрімкіший розвиток спільноти.

Зробивши порівняльний аналіз, Unity та Unreal Engine було прийнято рішення використовувати Unity як інструмент для розробки.

Для створення графічних елементів та 3D моделей, було обрано Blender. В якості інструменту для роботи з анімаціями, використовується графічний редактор Maya 3D.

Створення концепції та моделювання карти світу гри

В процесі розробки концепції було вирішено розробити рівні у вигляді лабіринтів для створення атмосферності і ефекту неочікуваності, оскільки гравець не буде знати, що його очікує за рогами.

У літературі можна зустріти різні типи алгоритмів генерації лабіринтів, більшість з них використовують деревоподібну структуру. Алгоритми генерації як правило спрямовані на створення ідеальних лабіринтів. Ідеальний лабіринт можна охарактеризувати, як лабіринт який не містить петель, або локацій куди не можна потрапити. Такі лабіринти мають лише одне правильне рішення.

Для реалізації лабіринту було обрано алгоритм "Hunt-and-Kill". Даний вибір можна пояснити, тим що в основі алгоритму лежить випадкове блукання. Це робить лабіринт неочевидним і доволі заплутаним. Випадкове блукання можна охарактеризувати як випадковий процес, який характеризує шлях, що складається з послідовності випадкових кроків.

Алгоритм вибирає будь-яку випадкову комірку лабіринту, після чого відбувається випадкове блукання, поки не закінчатся невідвідні комірки. Варто зазначити, що алгоритм не ітерується по комірках, котрі були вже попередньо відвідані. В момент коли усі сусідні комірки лабіринту вже є відвіданими, алгоритм переходить до наступного кроку. Запускається режим полювання, який починає ітеруватися по усіх комірках лабіринту, завжди починаючи з верхнього лівого краю. Коли знайдеться, хоча б одна невідвідана комірка, вона об'єднується з випадковою, вже відвіданою сусідньою коміркою. У випадку коли комірок було б, дві або більше, алгоритм би вибрав одну з них випадковим чином, після чого продовжив випадкове блукання. Цей процес триватиме до того моменту, поки фаза полювання, не зможе більше знайти невідвідані клітинки і це буде означати, що лабіринт завершив свою роботу.

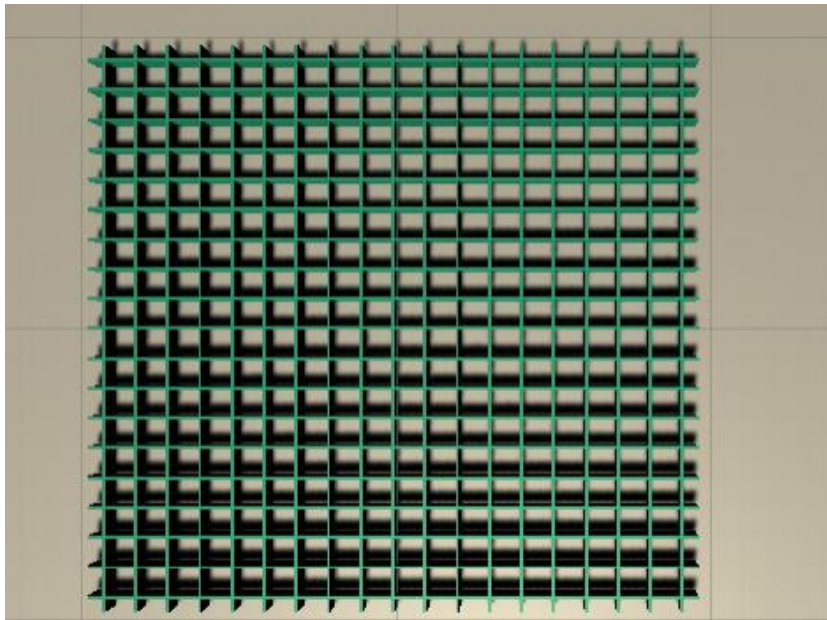
Розглядаючи формальне визначення алгоритму можна виділити наступні пункти:

1. Випадковий вибір стартової позиції
2. Реалізація випадкового блукання до моменту, поки в поточній комірці не буде вільних сусідів.
3. Активація режиму полювання, до моменту знаходження не відвіданої комірки
4. Повторення другого та третього кроку до моменту, поки не залишаться не відвіданих комірок

В реалізації гри, даний алгоритм представлений наступними класами:

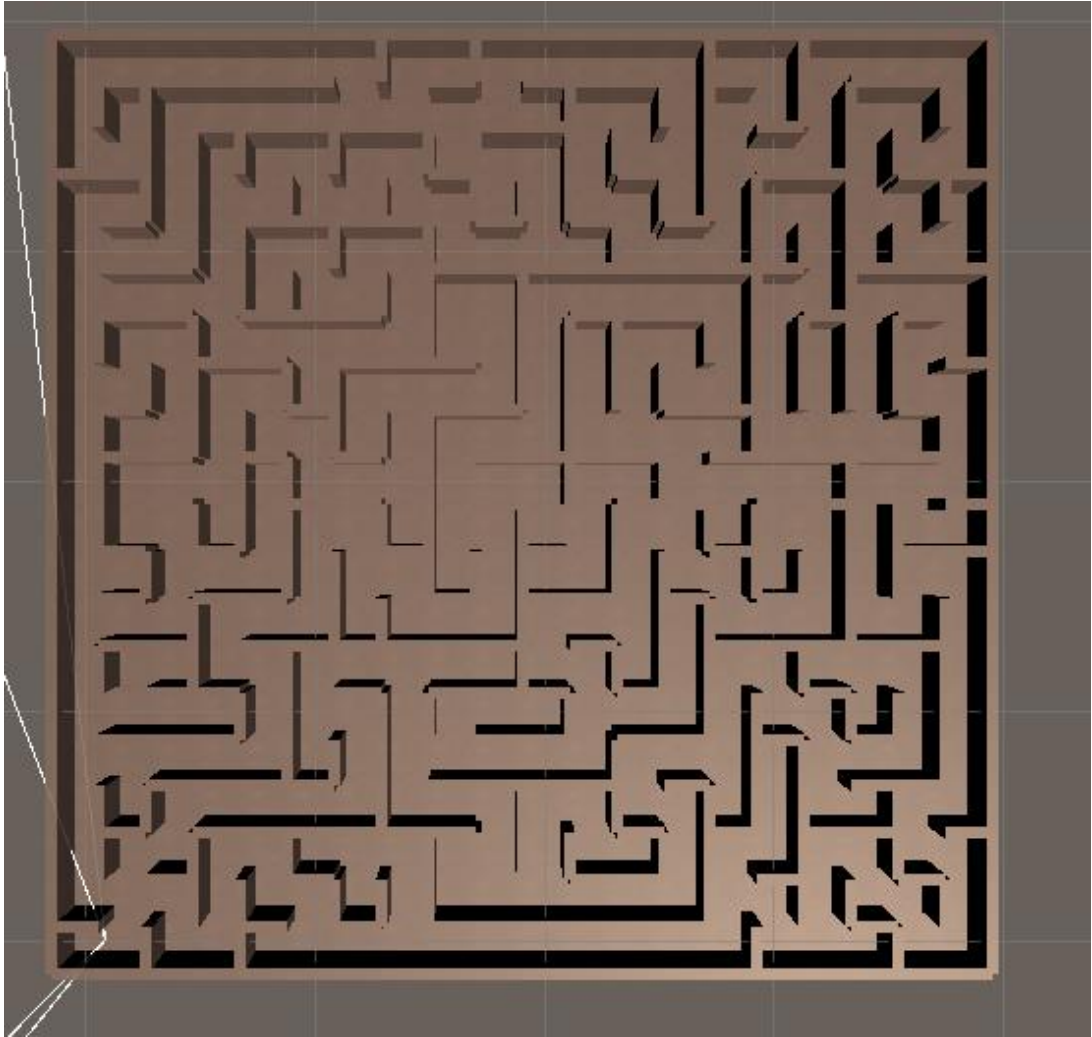
- Клас MazeCell містить булеве поле, яке вказує, була комірka відвідана чи ні. Також в даному класі розміщене поле об'єктів, котрі відповідають за ідентифікацію стінок, в комірці масиву.
- Абстрактний клас MazeAlgorithm - розроблений для реалізації меншої зв'язності коду, і в подальшому для наслідування від нього інших алгоритмів
- Клас HuntAndKillMazeAlgorithm відповідає за реалізацію алгоритму hunt and kill, і містить наступні методи:
 - HuntAndKill - відповідає за власне реалізацію алгоритму hunt and kill
 - Kill - реалізує випадкове блукання
 - Hunt - здійснює ітерацію по лабіринту, в пошуках не відвідуваних комірок.
 - RouteStillAvailable - перевіряє, чи можна здійснити випадкове блукання у випадково обраному напрямку

- `CellIsAvailable` - перевіряє чи комірка лабіринту, була вже попередньо відвідана
 - `DestroyWallIfItExists` - реалізує знищення стіни комірки
 - `CellHasAnAdjacentVisitedCell` - перевіряє, чи комірки є ще не відвідані сусідні комірки
 - `DestroyAdjacentWall` - реалізує знищення зайвих стінок комірки, відповідно до напрямку руху випадкового блукання
- Клас `MazeLoader` відповідає за відображення лабіринту в сцені гри, та містить наступні методи:
 - `InitializeMaze` - реалізує ініціалізацію усіх комірок лабіринту, відповідно до заданих розмірів



Малюнок 1. Ініціалізація комірок лабіринту

- В процесі ініціалізації методу `Start`, відбувається кінцева генерація лабіринту, за допомогою перевизначення методу `CreateMaze`



Малюнок 2. Згенерований лабіринт

Реалізація головного персонажа

Реалізації головного героя була приділена особлива увага, тому що це одна з основних речей, якій приділяє увагу гравець.

Основне правило, на яке треба орієнтується, це правило трьох С:

1. Персонаж (Character)
2. Камера (Camera)
3. Управління (Control)

Персонаж

Для спрощення створення персонажа було розроблено багато патернів характеру, поведінки та його стилю. Три основних типи героїв: гумористичні, епічні та анти-герої.

- Гумористичний персонаж робить і каже смішні речі. Піднімає настрій будь-якими способами. Найпростіший в вигадуванні тип, тому що все, що потребується для образу, це гарний гумор. Є багато прикладів таких героїв, особливо в дитячих іграх. Одним з таких
- Епічний персонаж більш схожий на казкового персонажа, котрий робить гарні речі, такі як рятування принцеси, світу тощо. Також помітною ознакою такого персонажа є його особливі навички, котрі показують, що саме це головний герой гри, прикладом може послужити Сонік з його швидким бігом чи Лара Крофт з її навичками в розграблюванні гробниць. Однак не всі герої ідеальні, тому дуже часто їм додають якісь проблеми, аби додати реалістичності своєму персонажу. Це збільшує зацікавленість гравця, бо він більше асоціює себе з героєм. Прикладом такого можуть бути Індіана Джонс з його страхом змій, або мультиплеєрній грі “Чемпіони Онлайн” ви самі можете додати такі страхи чи недоліки, що додає різноманіття грі. Саме тому епічний герой більш складний в розробці, бо потребує сюжету, цілі та характеру, і щоб усе це було зрозумілим гравцю.

- Анти-герой, як зрозуміло з назви, не робить добрі речі, але такі персонажи часто знаходять визнаку серед гравців, бо які б погані речі не були створені, вони будуть заворожувати гравців, якщо все зроблено гарно. Це розслабляє та дає випустити пар, коли пару клавiш доведуть вашого опонента до того стану, коли він вже не встане. Гарними прикладами таких персонажей можуть бути майже всі герої гри “Смертельна битва”. Кожен з них виходить на дуелі не на життя, а на смерть, задля своїх цілей і не буде шкодувати свого опонента.

У старих іграх дуже часто зустрічаються ці типи, але зараз більшість героїв мають змішані типи, котрі ще можуть змінюватися протягом сюжету, показуючи вплив сюжету та/або ваших рішень на характер персонажу.

Для цієї гри був обран змішаний тип, котрий вкладає в себе деякі елементи епічного героя, котрий сражається з монстрами, та поганого, котрий робить все це не задля “миру на всьому світі”, а задля отримання багатств та слави.

Під час дизайну персонажу також треба приділяти багато уваги його вигляду. Дуже багато основних принципів створення персонажей графічно ігрові дизайнери взяли з мультфільмів та анімацій, але з удосконаленням технологій все більше персонажів створювалося з орієнтиром на реальний вигляд людини, що призвело до ефекту зловісної долини, під час якого гравець починав почувати неприязнь та навіть страх до персонажа, тому що через реалістичну графіку він відчувається як людина, а деякі недоліки в обробці емоцій чи поведінки персонажа призводили до відторгнення цього почуття.

Наразі більшість таких персонажів позбавлені цього ефекту через удосконалення технологій, але ці технології не доступні усім, а навіть якщо доступний, то не завжди він вписується в концепт та історію гри. Наприклад, другу частину гри “Командна фортеця”, котра з самого початку була доповненням до гри “Quake” та не претендувала на реалістичність через старі технології, хотіли зробити такою, але в один момент команда змінили свою думку та зробила більша мальованийий стиль. Більшість критиків та експертів кажуть, що саме це рішення призвело до

успіху сиквелу, котрий і через 14 років ще має активну аудиторію. Тому і по цей час ігровий дизайн поділяється на реалістичний (також його часто називають фотореалістичний) та стилізований.

В цій грі був обраний стилізований дизайн, задля запобігання проблеми зловіщої долини та зменшення навантаження на комп'ютер.



Камера

Поведінка камери - це та річ, котра не так сильно запам'ятовується, як персонаж, але може призвести до більшого розчарування гравця, якщо з нею будуть якісь проблеми. Тому існує багато інструментів, для спрощення роботи на камерою та зменшення вірогідності помилки.

Застосований один з таких інструментів для Unity, Cinemachine. В ньому можна знайти багато пресетів. Найкращий варіант для 3Д лабіринту була стежуча камера з використанням FOV, котрий замилює края камери при повороті.

Управління

Управління - це друга річ, котра не запам'ятовується, але дуже впливає на досвід гравця. Існує багато варіантів налаштування кнопок, але є декілька таких варіантів, котрі повторюються усюди, що допомагає гравцю скоріше звикнути до управління персонажем (Наприклад кнопки WASD завжди відповідають за переміщення по осях x та z, пробіл за стрибка, С або Ctrl за присідання і тд)

Тому для цієї гри була взята та трохи змінена основна розкладка клавіатури, котра була застосована в більшості ігор. А саме:

WASD - Переміщення по осях x та z

Tab -

CapsLock

Q - переكات вперед

E - переكات назад

R - перший магічний скіл героїні

F - другий магічний скіл героїні

C - третій магічний скіл героїні

ЛКМ - перший фіз-удар

ПКМ - другий фіз-удар

СКМ - третій фіз-удар

Для реалізації управління персонажа був використаний патерн State, котрий виключає баги, зав'язані на одночасному натисканні на кнопки контролю персонажу.

Патерн State оснований на машині кінцевих автоматів і працює таким чином, що записує послідовність натискання клавіш і в залежності від них змінює стан контрольованого персонажа. Таким чином він запобігає створенню ситуацій, де персонаж знаходиться в декількох станах (Наприклад в стані бігу та атаки водночас).

Реалізація ворожих юнітів з використанням елементів машинного навчання

Навчання з підкріпленням - це машинного навчання, яке вчить агента, як вибрати дію зі свого простору дій, у певному середовищі, щоб максимізувати винагороду з часом.

Підкріплення навчання складається з чотирьох важливих елементів:

- **Агент.** Програма, яку ви тренуєте, з метою виконання заданої вами роботи.
- **Середовище.** Світ, реальний чи віртуальний, в якому агент виконує дії.
- **Дія.** Крок, здійснений агентом, який спричиняє зміну статусу в середовищі.
- **Нагороди.** Оцінка дії, яка може бути позитивною чи негативною.

Розглянемо кілька прикладів:

1. Керування ходячим роботом

Агент: програма, що керує ходячим роботом.

Навколишнє середовище: Реальний світ.

Дія: Один з чотирьох ходів (1) вперед; (2) назад; (3) лівий; та (4) правильно.

Нагорода: Позитивна, коли вона наближається до цільового пункту призначення; негатив, коли він марнує час, йде в неправильному напрямку або падає вниз.

У цьому останньому прикладі робот може навчитися рухатися ефективніше, адаптуючи свою політику на основі отриманих винагород.

2. Створення персоналізованої системи навчання

Агент: Програма, яка вирішує, що показувати далі в онлайн-навчальному каталозі.

Середовище: система навчання.

Дія: Відтворення відео нового класу та реклами.

Нагорода: Позитивна, якщо користувач вирішить натиснути на представлене відео класу; більша позитивна винагорода, якщо користувач вирішить натиснути рекламу; негативний, якщо користувач піде.

Ця програма може зробити персоналізовану систему класів більш цінною. Користувач може отримати вигоду з більш ефективного навчання, а система може отримати вигоду завдяки більш ефективній рекламі.

Агенти Unity ML - це плагін Unity, який дозволяє підключити ваших ігрових агентів (наприклад, ворожий ШІ) до сервера підкріплене навчання, який відповідає за 3 речі.

1. Надсилання необхідних даних RL: стану, дії, винагороди
2. Навчання моделі RL
3. Введення дій у гру з моделі

Підкріплення навчання - це декларативний прийом для гри AI

Декларативний, оскільки вам не потрібно розуміти, як працює модель RL. Насправді потрібно зробити лише 2 справи, а агенти Unity ML подбають про все інше.

1. Кодувати дії та стани
2. Встановити функцію винагороди

Приклади програм на базі RL:

1. Робототехніка - RL може бути використана для проблем з керуванням великих розмірів та для різних промислових застосувань.
2. Видобуток тексту - RL, разом із моделлю генерації тексту, може бути використаний для розробки системи, яка здатна створювати зручні для читання зведення довгих текстів.
3. Виконання торгівлі - Основні компанії у фінансовій галузі використовують алгоритми RL для вдосконалення своєї торгової стратегії.
4. Охорона здоров'я - RL корисний для дозування ліків та для оптимізації лікування для людей, які страждають на хронічні хвороби тощо.
5. Ігри - RL відомі тим, що є основним алгоритмом, що використовується для вирішення різних ігор та досягнення надлюдських результатів.

Висновок

В процесі написання курсової роботи було розроблено прототип 3D гри, користуючись сучасними інструментами та методологіями розробки. Прототип гри було реалізовано, за допомогою ігрового двигуна Unity. В процесі розробки механік головного персонажа було реалізовано такий патерн програмування як State. Новизна і актуальність роботи полягає у використанні машинного навчання з підкріпленням, котре було задіяне для навчання ворожих NPC. Було досліджено та проаналізовано процес самого навчання, що в кінцевому результаті дозволило вдосконалити кінцеві моделі.

Джерела

1. Python 3.8.3rc1 documentation[Електронний ресурс] – режим доступу:
<https://docs.python.org/3/>
2. ML-agents library <https://github.com/Unity-Technologies/ml-agents>
3. Jamis Buck, Book: Mazes for programers
4. Scott Rogers, Book: Level UP
5. <https://towardsdatascience.com/diving-deeper-into-unity-ml-agents-e1667f869dc3>