**Міністерство освіти і науки України**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

**«ДНІПРОВСЬКА ПОЛІТЕХНІКА»**

**ІНСТИТУТ ЕЛЕКТРОЕНЕРГЕТИКИ**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
Кафедра програмного забезпечення комп’ютерних систем**

**Звіт**

***з передатестаційної практики***

**Період проходження**

з « 19 » квітня 2021 р. по « 2 » травня 2021 р.

Виконавець:

ст. гр. 122-17-3

Бондар Андрій Павлович

Керівник практики від кафедри:

Бердник Михайло Геннадійович

Дніпро

2021

**ЗМІСТ**

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

* 1. Загальні відомості з предметної галузі…………………………………
  2. Призначення розробки та галузь застосування………………………..
  3. Підстава для розробки…………………………………………………..
  4. Постановка завдання……………………………………………………….
  5. Вимоги до програми або програмного виробу………………………….
     1. Вимоги до функціональних характеристик………………………………
     2. Вимоги до інформаційної безпеки…………………………………………
     3. Вимоги до складу та параметрів технічних засобів……………………….
     4. Вимоги до інформаційної та програмної сумісності ……………………

**Варіації назви теми:**

«Стратегія у реальному часі із використанням машинного навчання для адаптації до дій гравця».

«Розробка стратегії у реальному часі із використанням машинного навчання для адаптації до дій гравця».

Розробка ігрового додатку «Стратегія у реальному часі із використанням машинного навчання для адаптації до дій гравця».

* 1. **Загальні відомості з предметної галузі**

Комп`ютерні ігри міцно закріпили за собою звання відмінного засобу не тільки відпочинку, але і навчання. Вони здатні сформувати у гравця принципи мислення для вирішення поставлених завдань, які потім можливо застосувати для вирішення реальних проблем, і все це - в доступній, ігровій формі, за якої за виконання завдань гравець отримує задоволення, що закріплює отриманий досвід.

Один з жанрів комп’ютерних ігор, що дає творчу свободу та широкий набір інструментів для реалізації власних ідей – Sandbox, («пісочниця» у перекладі). Яскраві представники цього жанру:

* Minecraft (з англ. «шахтарське ремесло») – пісочниця з виглядом від першої/третьої особи. Гравець отримує у своє розпорядження трьохмірний відкритий, процедурно-генерований світ, що повністю складається з кубічних «блоків», які можна видобути та створювати власні структури у світі, або перетворити на блоки іншого типу. Можливість змінити гру за допомогою модифікацій розширюють її ще більше.



Рис. 1.1. Ігровий процес Minecraft

* Factorio – пісочниця, що виконана в жанрі ізометричного економічного симулятора з елементами стратегії. Основою ігрового процесу є створення великих виробничих ланцюжків і повсюдна автоматизація – персонаж гравця хоч і може виробляти необхідні компоненти та пристрої самостійно, але швидкість його виробництва не йде ні в яке порівняння з виробничими можливостями налагодженої фабрики. Забруднення, яке розповсюджується фабрикою, турбує мешканців планети – вигаданих членистоногих істот, та змушує збиратися в групи і нападати на джерело забруднення.



Рис. 1.2. Ігровий процес Factorio

Стратегія, як жанр ігор, існував задовго до створення перших комп’ютерів. Яскравий приклад – шахи, абстрактна стратегічна покрокова гра. Стратегії у реальному часі (RTS) відрізняються від покрокових тим, що події у них розвиваються одночасно, що вимагає від гравців швидкої реакції та стратегічного мислення.



Рис. 1.3. Ігровий процес StarCraft 2

RTS зазвичай відтворюють військові дії та розбудову військових баз для здобуття перемоги над противником, мають специфічний інтерфейс. Їм притаманні численні ігрові умовності, такі як врівноваження можливостей протиборчих сторін, нереалістично короткі терміни проведення боїв і будівництва, видобування абстрактних ресурсів що потребуються для створення бойових одиниць та будівель.



Рис. 1.4. Ігровий процес Supreme Commander

У більшості стратегій гра проходить як протистояння двох або більше сторін. Гравці-люди не завжди можуть скласти компанію в грі, через особливості механіки гри або з інших причин. Для подібних випадків передбачено заміщення реального гравця комп'ютерним, ШІ-суперником.

Ігровий ШІ в основному займається вибором дій в залежності від поточних умов. У традиційній літературі по ШІ це називається "інтелектуальними агентами". Агентом зазвичай є персонаж гри, але це може бути і машина, робот або навіть щось більш абстрактне - ціла група сутностей, країна або цивілізація. У будь-якому випадку це об'єкт, що стежить за своїм оточенням, що приймає на підставі нього рішення і діє відповідно до цих рішень.

Задачі ШІ реального світу, особливо ті що є актуальними сьогодні, зазвичай зосереджені на «сприйнятті». Наприклад, безпілотні автомобілі повинні отримувати зображення знаходиться перед ними дороги, комбінуючи їх з іншими даними (радара й лідара) та намагаючись інтерпретувати те, що вони бачать. Зазвичай це завдання вирішується машинним навчанням, яке особливо добре працює з великими масивами зашумлених даних реального світу (наприклад з фотографіями дороги перед автомобілем або декількома кадрами відео) і надає їм якесь значення, отримуючи семантичну інформацію, наприклад «через 10 метрів переді мною є ще одна машина».

Ігри відрізняються тим, що для вилучення інформації їм не потрібна окрема система, оскільки вона є невід'ємною частиною симуляції. Немає необхідності виконувати алгоритми розпізнавання зображень, щоб виявити ворога перед собою: гра знає, що там є супротивник і може передати цю інформацію безпосередньо процесу прийняття рішень.

Зазвичай передбачається, що гра повинна розважати і кидати гравцеві виклик, а не бути «оптимальною» – тому навіть якщо і можна навчити агентів протистояти гравцям найкращим чином, то найчастіше гейм дизайнерам потрібно від них зовсім іншого. Яскравим прикладом цього принципу є ігри жанру шутер, де рефлекси навіть професійних геймерів-людей ледве зможуть протистояти ботам-«шахраям», що заздалегідь знають де знаходиться опонент та ціляться з точністю, недосяжною для людей.

Розробники мають великий арсенал інструментів, щоб зробити дії ІШІ більш схожими на людські, без необхідності повністю копіювати людську поведінку. Для жанру стратегій це можуть бути спеціальні маркери (AI markers) на ігровій мапі, за допомогою яких ШІ може направляти свої юніти, жорстко прописані скрипти опису дій, що необхідно виконати, або ж трохи «шахрайства», як-то видача додаткових ігрових ресурсів або відсутність туману війни (область що приховує юніти супротивника та потребує розвідки) для «зору» ШІ.



Рис. 1.5. Графи маршрутів юнітів (AI markers) для мапи гри Supreme Commander

**1.2. Призначення розробки та галузь застосування.**

Розробка ігрового додатку «Стратегія у реальному часі з використанням машинного навчання для адаптації до дій гравця».

Основні терміни та ключові слова:

Машинне навчання – підгалузь штучного інтелекту в галузі інформатики, яка застосовується для надання комп'ютерам здатності «навчатися», покращуючи свою продуктивність у певній задачі з поточних даних, замість того, щоби бути програмованими явно.

Штучний інтелект – здатність системи здобувати, обробляти та застосовувати знання та вміння подібно до того, як це виконує людина.

**1.3. Підстави для розробки.**

**1.4. Постановка завдання.**

Мета завдання – розробити стратегію у реальному часі, що розрахована на одного користувача, та буде адаптуватися під дії гравця. Для цього програма має використовуючи алгоритми машинного навчання, такі як штучні нейронні мережі та пошук оптимального шляху з врахуванням вагів.

Ландшафт ігрової мапи має створюватися процедурною генерацією на основі шуму Перліна.

Пошук шляху юнітів ШІ-супротивника має базуватися на цільовому векторному полю, що генерується з врахуванням типу ландшафту та співвідношенням отриманої та нанесеної шкоди у обчислювальних тайлах.

Економічна складова стратегії має залежати від кількості тайлів (клітин мапи ігрового поля), що захоплено гравцем та його супротивником. Захоплення тайлів має відбуватися поступово, шляхом створення допоміжних юнітів-будівель.

Області мапи, що ще не розвідані юнітами, мають бути покриті «туманом війни» та підлягают розвідці як зі сторони гравця, так і зі сторони ШІ-супротивника.

Розподілення видобутих ресурсів супротивника на створення нових юнітів має здійснюватися ШІ-агентом на основі штучних нейронних мереж, що мусить самостійно навчатись згідно з поточними обставинами та результуючою ефективністю своїх юнітів.

Розроблений ігровий додаток спрямований на розвиток стратегічного мислення у гравця, шляхом пошуку оптимальної стратегій та адаптації до мінливої ситуації.

**1.5. Вимоги до програми або програмного виробу**

**1.5.1. Вимоги до функціональних характеристик.**

Процес введення даних та команд до програми проводиться за допомогою клавіатури та миші. Результуючі значення, що розраховані на основі введених користувачем даних, мають бути приведені до діапазону можливих значень: як-то, користувач не може створити юніта за межами ігрового поля.

Цикл обчислення положень та атрибутів об’єктів також має враховувати ці діапазони. Наприклад, юніти не можуть вийти за межі ігрового поля, а більшість атрибутів не можуть дорівнювати від’ємному значенню.

Виведення даних проводиться у графічному вигляді за допомогою монітору комп’ютера.

**1.5.2. Вимоги до інформаційної безпеки.**

Програма у фінальному вигляді скомпільована та не потребує зв’язку з мережею Інтернет, тому додаткових вимог до безпеки не потребує. Супутні файли (DLL-бібліотеки, файли графічних ресурсів тощо) мають зберігати своє положення відносно виконуваного файлу програми задля уникнення можливих несправностей програми.

**1.5.3. Вимоги до складу та параметрів технічних засобів**

Для користувача є важливим мати систему з відеокартою, що сумісна з OpenGL версії 3.3 або вище. Операційною системою має бути Windows 10 версії 64 bit. Мінімально можлива комплектація ЕОМ:

* CPU: сімейства Intel Celeron;
* GPU: 3D адаптер Nvidia, Intel HD або AMD/ATI;
* RAM: 2048 МБ.
* VRAM (відеопам’ять): 64 МБ;
* Вільне місце на накопичувачі: 128 МБ;

**1.5.4. Вимоги до інформаційної та програмної сумісності.**

Програма має бути реалізована на мові C++ за допомогою IDE Visual Studio 2019. Допоміжні бібліотеки, використані для розробки:

* RayLib v3.5, безкоштовна бібліотека на мові C з вільним вихідним кодом, що спрямована на допомогу у створені відеоігр.
* OpenMP 2.0, відкритий стандарт та бібліотека для створення застосунків з використанням паралельних обчислень.