

Міністерство освіти і науки України
Одеський національний технологічний університет
Кафедра комп'ютерної інженерії



ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ

Багатокритерійна оптимізація ігрового процесу відеогри
на тему (назва кваліфікаційної роботи згідно наказу ОНТУ)
симулятора кулінарії

Здобувача Бабія М. О.
(прізвище, ініціали)

2 курсу 566А групи

Керівники: доцент Ненов О. Л.
(посада, прізвище та ініціали)

ст. викл. Слушина Н.В.

(посада, прізвище та ініціали)

Консультанти: проф. Басюркіна Н.Й.
(посада, прізвище та ініціали)

(посада, прізвище та ініціали)

Кваліфікаційна робота допускається до захисту

Рішення кафедри від 30.11 2022 р., протокол № 3

Завідувач кафедри комп. інженерії Сергій АРТЕМЕНКО

(назва кафедри)

(підпис)

(Ім'я ПРІЗВИЩЕ)

Одеса - 2023 рік

6. Консультанти по роботі, із зазначенням розділів роботи, що стосуються їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
<i>Економіка</i>	<i>Спаский І.Д., к. е. н., проф.</i>		
<i>Охорона праці</i>	<i>Нєнов О.Л., к. т. н., доц.</i>		
<i>Нормоконтроль</i>	<i>Нєнов О.Л., к. т. н., доц.</i>		

7. Дата видачі завдання 15.12.2022

Керівники _____ *Олексій НЄНОВ*
_____ *Наталя СЛУШНА*

Завдання прийняв до виконання _____ *Максим БАБІЙ*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	<i>Передпроектний аналіз предметної області.</i>	<i>18.03.2023</i>	
2.	<i>Постановка завдання на розробку.</i>	<i>15.04.2023</i>	
3.	<i>Оптимізаційне дослідження</i>	<i>06.05.2023</i>	
4.	<i>Проектування архітектури проекту гри.</i>	<i>28.05.2023</i>	
5.	<i>Проектування компонентів проекту гри.</i>	<i>11.06.2023</i>	
6.	<i>Проектування алгоритмів гри.</i>	<i>27.06.2023</i>	
7.	<i>Програмна реалізація гри.</i>	<i>24.07.2023</i>	
8.	<i>Тестування гри.</i>	<i>28.08.2023</i>	
9.	<i>Техніко-економічний аналіз проекту.</i>	<i>02.09.2023</i>	
10.	<i>Опрацювання питань охорони праці.</i>	<i>03.10.2023</i>	
11.	<i>Оформлення пояснювальної записки і презентації.</i>	<i>01.11.2023</i>	

Керівники роботи _____ *Олексій НЄНОВ*
_____ *Наталя СЛУШНА*

Несу відповідальність за ідентичність електронного та друкованого варіантів кваліфікаційної роботи, даю згоду на обробку персональних даних та не заперечую проти розміщення кваліфікаційної роботи на офіційних web-ресурсах ОНТУ.

Підтверджую, що в кваліфікаційній роботі відсутні порушення норм академічної доброчесності.

Здобувач - дипломник _____ *Максим БАБІЙ*

АННОТАЦІЯ

Робота присвячена проектуванню та розробці гри жанру «*Simulator*». Досліджено предметну область, розглянуто існуючі сучасні аналоги та популярні представники жанру. Показано їх переваги та недоліки. Розглянуті методи багатокритерійної оптимізації. Проведена оптимізація ігрового процесу за запропонованими критеріями. Розроблено концептуальний та дизайнерський документ. Створено графічну складову гри. Розроблено демонстраційну версію гри та проведено її тестування.

Пояснювальна записка складається з 108 аркушів основної текстової частини і 14 аркушів додатків

Ключові слова: ігри жанру «*Simulator*», ігровий рушій, *Unity*, дизайнерський документ.

ABSTRACT

The paper is devoted to the design and development of a game of the "Simulator" genre. The subject area is studied, existing modern analogs and popular representatives of the genre are considered. Their advantages and disadvantages are shown. The methods of multicriteria optimization are considered. The game process is optimized according to the proposed criteria. A conceptual and design document is developed. The graphic component of the game is created. A demo version of the game was developed and tested.

The explanatory note consists of 108 pages of the main text and 14 pages of appendices.

Keywords: games of the "Simulator" genre, game engine, Unity, design document.

РОЗДІЛ 3 РОЗРОБКА ДЕМОНСТРАЦІЙНОЇ ВЕРСІЇ ГРИ	45
3.1 Вибір засобів розробки	45
3.2 Початок розробки	47
3.2.1 Створення головної сцени	47
3.2.2 Створення ігрового інтерфейсу	49
3.3 Створення префабів	51
3.3.1 Префаби клієнтів	52
3.3.2 Префаби інгредієнтів та страв	58
3.3.3 Префаби інструментів	60
3.4 Створення головного персонажу	69
3.5 Створення штучного інтелекту співробітника	70
3.6 Створення штучного інтелекту клієнта	73
3.7 Багатокритерійна оптимізація геймплею	75
Висновки до третього розділу	78
РОЗДІЛ 4 ЕКОНОМІЧНІ РОЗРАХУНКИ	79
4.1 Організаційно-економічне й маркетингове обґрунтування проекту	79
4.1.1 Порівняльний техніко-економічний аналіз проекту	79
4.1.2 Організаційне обґрунтування	80
4.1.3 Маркетингові обґрунтування проекту	83
4.2 Економічні розрахунки	84
4.2.1 Визначення трудомісткості розробки ПП	84
4.2.2 Визначення ціни ПП	85
4.2.3 Визначення показника економічної ефективності	90
4.3 Бізнес план стартапу проекту	93
Висновки до четвертого розділу	94
РОЗДІЛ 5 ОХОРОНА ПРАЦІ	95
5.1 Класифікація виробництв за ступенем пожежної небезпеки.	95
5.2 Рішення з розміщенням проектованого обладнання	95
5.3 Електробезпека	96
5.4 Пожежна профілактика	99
5.5 Виробнича санітарія	100
ЗАГАЛЬНІ ВИСНОВКИ	106
СПИСОК ВИКОРИСТАНИХ СКОРОЧЕНЬ І ТЕРМІНІВ	107

					<i>КРМ.КІ.1.884-03.1.1</i>	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	108
ДОДАТКИ.....	109
ДОДАТОК А ВИБРАНІ СЛАЙДИ ПРЕЗЕНТАЦІЇ.....	109
ДОДАТОК Б ВИХІДНИЙ КОД.....	115

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Сучасний розвиток інформаційних технологій і комп'ютерних ігор сприяє постійному зростанню популярності відеогроманії серед різних вікових категорій користувачів. Ігрові процеси стають все складнішими і реалістичнішими завдяки застосуванню передових технологій у галузі комп'ютерної графіки, штучного інтелекту та симуляції.

Одним із найпопулярніших жанрів відеоігор є симулятори, що дозволяють користувачам зануритися в унікальний імерсивний світ, де вони можуть експериментувати, розвивати вміння та здібності, відчувати себе частиною іншого реального світу. Один із захоплюючих жанрів симуляторів – симулятори кулінарії, де користувачі відчують себе шеф-кухарями і можуть готувати різноманітні страви, вдосконалювати свої навички та конкурувати з іншими гравцями.

Однак, ігровий процес у симуляторах кулінарії може стикатися з рядом складних викликів, пов'язаних з багатокритерійною оптимізацією. Завдання гри полягає не лише в досягненні максимального рівня задоволення користувача, але й у врахуванні інших критеріїв, таких як ефективність готування, використання ресурсів, збалансованість страв та багато інших.

Таким чином, ця кваліфікаційна робота спрямована на дослідження та розробку методів багатокритерійної оптимізації ігрового процесу відеоігри – симулятора кулінарії.

Метою дослідження є створення імерсивного та захоплюючого ігрового досвіду для користувачів, який враховує їхні індивідуальні переваги та вміння, сприяючи оптимальному балансу між різними ігровими критеріями.

У цьому дослідженні будуть розглянуті основні проблеми і виклики, пов'язані з багатокритерійною оптимізацією, а також вплив цих методів на реалізацію ігрового процесу симулятора кулінарії.

					<i>КРМ.КІ.1.884-03.1.1</i>	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

Крім того, розроблені та апробовані нові підходи до оптимізації, що можуть поліпшити ігровий досвід для користувачів та зробити гру більш цікавою та захоплюючою.

Результати цього дослідження можуть знайти застосування у розробці сучасних відеоігор та допомогти вдосконалити існуючі підходи до багатокритерійної оптимізації ігрових процесів. Висновки та рекомендації, отримані під час дослідження, дозволять створити більш ефективні та забавні симулятори кулінарії, що забезпечать задоволення користувачів і привернуть ще більше уваги до цього захоплюючого жанру відеоігор.

Метою кваліфікаційної роботи є дослідження методів багатокритерійної оптимізації ігрового процесу відеогри – симулятора кулінарії. Головною метою практичної складовою є створення імерсивного та захоплюючого ігрового досвіду для користувачів, який враховує їхні індивідуальні переваги та вміння, сприяючи оптимальному балансу між різними ігровими критеріями. Це включає досягнення якісних та кількісних показників успішності гри, таких як ефективність готування страв, задоволення клієнтів, використання ресурсів, рівень складності та збалансованість геймплею.

Об'єктом дослідження є процес багатокритерійної оптимізації ігрового процесу відеоіго-симуляторів.

Предметом дослідження є методи багатокритерійної оптимізації, їхні можливості та обмеження при застосуванні у відеогрі-симуляторі кулінарії, а також критерії оптимізації, які можуть бути використані для оцінки ефективності та якості ігрового процесу.

При багатокритерійній оптимізації визначаються критерії оптимізації, які можуть бути використані для оцінки ігрового процесу. Серед них можуть бути час готування страв, рівень задоволення клієнтів, кількість використаних ресурсів (інгредієнтів, енергії тощо).

Впровадження багатокритерійної оптимізації може значно збільшити складність ігрового процесу, тому також звернено увагу на питання визначення

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

оптимального компромісу між кількістю критеріїв та швидкістю обчислення оптимального рішення.

Для дослідження реалізован прототип симулятора кулінарії, що базується на популярних ігрових платформах. Цей прототип дозволяє експериментувати з різними методами оптимізації та вирішувати реальні задачі готування страв у відеоігровому середовищі.

Дослідження багатокритерійної оптимізації ігрового процесу відеогри – симулятора кулінарії має великий потенціал у покращенні ігрового досвіду користувачів та збагаченні геймплею цього захоплюючого жанру. Результати цього дослідження можуть сприяти не тільки розвитку ігрової індустрії, але й застосуванню багатокритерійної оптимізації в інших сферах, де важлива рівновага між різними критеріями.

Зацікавлення у розв'язанні цих проблем та пошук інноваційних підходів до оптимізації ігрового процесу є ключовим фактором для створення найкращих ігрових досвідів, які зможуть зачарувати гравців та забезпечити популярність гри у широкому колі користувачів.

Наукова новизна кваліфікаційній роботи:

1. Дослідження інноваційних алгоритмів багатокритерійної оптимізації: кваліфікаційна робота досліджує алгоритми оптимізації, які дозволяють враховувати одночасно декілька критеріїв у геймплеї симулятора кулінарії. Ці алгоритми базуються на передових методах штучного інтелекту, еволюційних алгоритмах та гібридних підходах.

2. Кваліфікаційна робота впроваджує поєднання елементів таких як:

- аркадний стиль геймплею;
- система реалістичного приготування страв;
- розширена економічна система;
- впровадження системи штучного інтелекту.

3. Впровадження експериментальної системи штучного інтелекту повністю автоматизованого функціонування.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ДОСЛІДЖЕННЯ ІСНУЮЧИХ АНАЛОГІВ

1.1 Аналіз предметної області

Багатокритерійна оптимізація ігрового процесу – це методологія або підхід до розробки та налаштування відеоігор, яка спрямована на одночасне покращення декількох аспектів гри. У відеоіграх існують різноманітні параметри та характеристики, які можуть впливати на якість геймплею та задоволення від гри. Багатокритерійна оптимізація прагне знайти компроміс між цими різноманітними аспектами з метою досягнення оптимального варіанту для гравців.

Предметна область аркадних ігор жанру симулятор кулінарії зазнала помітного розвитку протягом останніх років. Цей жанр став популярним завдяки своїй привабливості для різних груп гравців: від досвідчених геймерів до новачків. Ось деякі аспекти предметної області, які варто розглянути:

1. Інтерактивність та швидкість: багато з цих ігор базуються на швидкості та реакції гравця. Гравці повинні оперативно готувати страви та обслуговувати клієнтів, щоб виконувати різноманітні завдання. Це додає динаміки і адреналіну гри.

2. Співпраця та мультиплеєр: багато ігор жанру симулятор кулінарії підтримують режим співпраці, де гравці можуть об'єднати свої зусилля для досягнення спільної мети. Це робить ігри ще цікавішими для ігрових груп або сімей.

3. Креативність та варіативність: багато ігор цього жанру пропонують широкий спектр страв та інгредієнтів, що дозволяє гравцям відчувати себе справжніми кухарями і творити різноманітні страви.

4. Менеджмент ресторану: деякі ігри цього жанру також включають елементи менеджменту ресторану, де гравці повинні вдосконалювати свій

					КРМ.КІ.1.884-03.1.1	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

заклад, приваблювати клієнтів і збільшувати дохід.

5. Розвиток персонажа: деякі ігри дозволяють гравцям розвивати свого персонажа, покращувати навички готування і отримувати нові можливості під час гри.

6. Красива графіка та анімації: багато ігор жанру симулятор кулінарії мають яскраву та привабливу графіку, що додає ще більше насолоди від гри.

7. Оновлення та доповнення: багато розробників підтримують свої ігри, додаючи нові рівні, страви, персонажів та інші вдосконалення, що збільшують тривалість гри та зацікавленість гравців.

Цей жанр є веселим та розважальним вибором для багатьох гравців, і завдяки сталому розвитку гейм-індустрії можна очікувати появу ще більш інноваційних і захоплюючих ігор у жанрі симулятор кулінарії.

Розглянемо основні аспекти предметної області, що підлягають аналізу.

1. Визначення ігрового процесу відеогри – симулятора кулінарії: з'ясування того, що включає в себе ігровий процес відеогри – симулятора кулінарії. Це може включати в себе різноманітні етапи, такі як планування меню, готування страв, обслуговування клієнтів, управління рестораном тощо.

2. Аналіз функцій та особливостей симулятора кулінарії: вивчення основних ігрових механік, елементів геймплею, системи керування, можливостей взаємодії з оточенням, а також різноманітних режимів гри. Важливо також розглянути, чи має гра елементи багатокритерійної оптимізації та як вони взаємодіють з головною механікою.

3. Багатокритерійна оптимізація в ігровому процесі: детальний аналіз концепції багатокритерійної оптимізації в контексті відеогри – симулятора кулінарії. Визначення критеріїв оптимізації, які можуть використовуватись у грі, таких як якість приготовлених страв, задоволеність клієнтів, прибутковість ресторану, складність гри, кількість рівнів тощо. Також важливо розглянути можливі взаємозв'язки та конфлікти між цими критеріями.

4. Попередні дослідження в сфері багатокритерійної оптимізації в іграх:

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

вивчення вже наявних наукових публікацій та досліджень, що стосуються багатокритерійної оптимізації в ігрових симуляторах кулінарії або схожих відеоіграх. Це дозволить з'ясувати, які підходи та методи використовувались раніше, які результати були отримані та чим можна інспіруватись для власного дослідження.

5. Обґрунтування актуальності теми: виокремлення причин, що роблять дану тему актуальною та важливою для дослідження. Можливі проблеми та виклики, які виникають у галузі розробки симуляторів кулінарії, які можна вирішити з допомогою багатокритерійної оптимізації. Також можна розглянути популярність та зростання інтересу до відеоігор про кулінарію серед гравців.

6. Можливість застосування результатів дослідження: розгляд можливостей практичного застосування отриманих результатів. Це можуть бути рекомендації для розробників гри з метою покращення геймплею, розробка нових алгоритмів багатокритерійної оптимізації для використання у відеоіграх, або можливість використання дослідження в інших сферах, де також існує потреба у багатокритерійній оптимізації.

Вибір показника оцінки за кожним критерієм у багатокритерійній оптимізації в ігрових симуляторах кулінарії є важливим завданням для досягнення бажаних результатів та геймплейного досвіду. Кожен показник повинен відображати важливі аспекти гри та кулінарного процесу, які визначають якість та задоволення гравців.

Ось деякі можливі критерії та показники для оцінки у багатокритерійній оптимізації в ігрових симуляторах кулінарії.

1. Складність готування: цей критерій може включати показники, які оцінюють складність рецептів та кулінарних процесів. Можна вимірювати кількість кроків, різноманітність інгредієнтів, час приготування тощо.

2. Смакові характеристики: Оцінка смакових якостей страв може включати показники, що визначають популярність, задоволення гравців від смаку, відповідність оригінальним рецептам.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Візуальний вигляд та презентація: показники для оцінки вигляду страв можуть включати деталізацію графіки, рівень деталізації інгредієнтів та готових страв, кольорову гаму тощо.

4. Задоволення гравця: Оцінка геймплейного досвіду та задоволення гравця може включати показники, які відображають, наскільки гравці відчують радість та виклик під час готування та взаємодії з грою.

5. Час приготування: Вимірювання часу, необхідного для приготування страв, може бути важливим аспектом геймплею.

6. Ефективність використання інгредієнтів: Показники ефективності можуть включати відсоток використаних інгредієнтів, мінімізацію втрат, раціональне використання ресурсів.

7. Різноманітність страв: Оцінка кількості та різноманітності доступних рецептів та страв.

8. Взаємодія з персонажами або гравцями: Якщо в грі передбачена можливість спільного готування або змагання між гравцями, то можуть використовуватись показники, пов'язані з результатами спільних зусиль.

1.2 Існуючі аналоги

1. "Cooking Simulator". Ця гра розроблена студією Big Cheese Studio та видана компанією PlayWay S.A. В "Cooking Simulator" гравці приймають на себе роль кухаря віртуальної кухні. Вони можуть готувати різноманітні страви, використовуючи реалістичні інгредієнти, кухонне обладнання та приладдя.

Гра пропонує різноманітні режими, включаючи безліч рецептів, а також випробування власної кулінарної майстерності від імпровізації до справжніх кулінарних випробувань.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 1.1 – Скріншот ігрового процесу гри «Cooking Simulator»

1. "Overcooked!" та "Overcooked! 2": Розроблена студією Ghost Town Games та видана компанією Team17, "Overcooked!" та його сиквел "Overcooked! 2" — це кооперативні аркадні головоломки про кулінарію. Гравці працюють разом, щоб готувати страви, обслуговувати клієнтів та керувати ресторанами. Гра акцентується на співпраці та координації між гравцями, створюючи захоплюючий та веселий ігровий досвід.



Рис. 1.2 – Скріншот ігрового процесу гри «Overcooked!»

2. "Cook, Serve, Delicious!". Це серія ігор розроблених студією Vertigo Gaming. У серії ігор "Cook, Serve, Delicious!" гравці перевтілюються в власника та

					<i>КРМ.КІ.1.884-03.1.1</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

Ці ігри представляють лише деякі приклади симуляторів кулінарії, які існують на ринку. Жанр симуляторів кулінарії є досить популярним серед гравців, тому його представники можуть здобути успіх, пропонуючи різноманітні та захоплюючі ігрові досвіди з кулінарною тематикою. За час, який минув після моїх останніх знань, можливо, з'явилися нові ігри, які варто розглянути при аналізі предметної області.

1.3 Порівняння елементів гри жанру «Симулятор кулінарії» в існуючих аналогах

Перелік ігрових елементів представлено у таблиці 1.1.

Таблиця 1.1

Порівняння ігрових компонентів в існуючих аналогах

Назва гри	Cooking Simulator	Overcooked!	Cook, Serve, Delicious!	Chef: A Restaurant Tycoon Game
Розробник / Видавець	Big Cheese Studio / PlayWay S.A.	Ghost Town Games / Team17	Vertigo Gaming / Vertigo Gaming	Inner Void / Digital Tribe
Головний акцент	Приготування страв	Кооперативна аркада	Управління рестораном	Управління рестораном
Геймплей	Реалістична фізика, інтеракція зі складниками	Готування страв та обслуговування клієнтів у команді	Приготування страв та управління рестораном	Створення та управління власним рестораном

Висновки до першого розділу

1. У результаті аналізу предметної області було виявлено основні риси ігор жанру «Симулятор кулінарії» та представлені їх основні різновиди.
2. Аналіз показав, що сучасні ігри в жанрі симуляторів кулінарії часто недостатньо враховують багатокритерійний підхід до оптимізації ігрового процесу. Більшість існуючих аналогів фокусуються на реалістичності рецептів та готування, але не дотримуються оптимального балансу між реалізмом та задоволенням гравця. Проаналізовано елементи гри жанру «Симулятор кулінарії» в існуючих аналогах.
3. Здійснено постановку задачі з зазначенням бажаного результату, що є необхідним для розробки проєктної документації та демонстраційної версії гри.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2

ПРОЕКТНА ДОКУМЕНТАЦІЯ

2.1 Вступ

Ігровий процес проходить у підготовленій локації (ресторан). Весь геймплей проходить на локації “кухня”, головний персонаж взаємодіє з різним обладнанням для приготування страв, щоб обслужити клієнтів (штучний інтелект).

Однією з особливостей гри є так звана система «взаємодії». Гравець має можливість взаємодіяти з великою кількістю інструментів та інгредієнтами.

Основна ціль гри – як можна швидше виконувати замовлення зроблений штучним інтелектом (клієнтом), для покращення геймплею є функціонал найняття співробітника (штучний інтелект) який допомагає у приготуванні замовлення.

За допомогою різних фільтрів передається загальне напруження. Уся графіка представлена у воксельному стилі, використовуються оригінальні моделі та текстури.

2.2 Жанр та аудиторія

Гра «*Los Pollos Hermanos*» належать до ігор жанру «*Симулятор кулінарії*» від першої особи. Цей жанр включає в себе імітацію приготування їжі та управління рестораном чи кулінарною кар'єрою.

Гра орієнтована на підліткову та дорослу аудиторію, містить обмежуючий контент, мінімальний вік гравця – 12 років. Додаткову привабливість гра має для власників не найсучаснішої конфігурації ПК та людей, які шукають альтернативи в даному жанрі, на платформі.

Для підліткової частини цієї категорії характерна потреба у динамічних іграх.

					КРМ.КІ.1.884-03.1.1	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

Гра сподобається фанатам класичних симуляторів кулінарії від третьої особи. Прикладами ігор цього жанру можна назвати такі серії ігор, як «Overcooked», «Cook, Serve, Delicious», «Cooking Simulator». Гра має інтуїтивно зрозуміле управління, динамічний ігровий процес.

2.3 Основні особливості гри

Основними особливостями гри (*USP – Unique Selling Points*) є:

1. Реалістичне приготування їжі: Гра надає гравцям можливість приготувати страви з врахуванням рецептів і реалістичних кулінарних процесів. Це може включати різні етапи готування, від різання і смаження до приготування соусів.

2. Управління рестораном: Окрім готування їжі, гравці можуть керувати власним рестораном або кулінарною кар'єрою. Це включає в себе відкриття ресторану, найм персоналу, управління запасами і обслуговування клієнтів.

3. Система розвитку персонажа: Гравці можуть розвивати свого персонажа, вдосконалюючи його навички в готуванні, отримуючи нові рецепти і піднімаючи рівень в кулінарній ієрархії.

4. Система взаємодії з будь-яким об'єктом гри.

2.4 Опис гри

2.4.1 Хід гри

Гра має механіку класичних «симуляторів кулінарії», але з певними геймплейними відмінностями.

Основне завдання гравця – як можна швидше створити замовлення.

Ігровий процес полягає в використанні гравцем різних типів інструментів та спорядження задля найбільш ефективного методу створення замовлення.

Інструменти для приготування страв:

– каса;

					КРМ.КІ.1.884-03.1.1	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

- заказ ігрідентів;
- гриль;
- сода машина;
- нарізна дошка;
- фритюр;
- стіл для створення блюда;
- співробітник.

В таблиці 2.1 приведено повний розбір кожного інструменту

Таблиця 2.1

Інструменти для приготування страв

Назва	Опис
Каса	Важливий елемент гейплею. Інструмент для прийняття та підготовки замовлення, зроблений клієнтом.
Заказ інгредієнтів	Економічна система гри. Інструмент для замовлення інгредієнтів для створення страв. Кожен інгредієнт має свою ціну.
Гриль	Обов'язковий елемент геймплею. Інструмент для смаження м'ясний та інших став.
Сода машина	Інструмент для створення напоїв.
Нарізна дошка	Інструмент для нарізання овочів.
Фритюр	Інструмент для смаження картоплі.
Стіл для створення блюда	Інструмент для збирання всіх інгредієнтів та створення страви.
Співробітник	Елемент поліпшення геймплею. Штучний інтелект котрий виконує завдання дане гравцем.

2.4.2 Ігрове оточення

Оскільки представлена гра є грою жанру «Симулятор кулінарії» з елементами аркади, її локації є прототип ресторану типу «fast food».

Вся графічна складова виконана у низькополігональному, «воксельному» стилі. У демонстраційній версії гри представлена одна ігрова локація – ресторан під назвою «Los Pollos Hermanos». Зміна локацій, а саме ресторанів з’являються з прогресом у грі. Різноманітні інгредієнти також з’являються з прогресом у грі.

Візуальна основа для зовнішнього виду гри є просто графіка, в аркадному стилі.

Використовуючи додаткові ефекти освітлення та згладжування, загальна картинка посилює враження від ігрового процесу. Ігрові елементи використовують схожу палітру фарб, набори текстур, котрі логічно доповнюють один одного та виправдовують деякі механіки. Демонстраційний рівень зображено на рисунку 2.1



Рис. 2.1 – Зображення дизайну демонстраційної версії гри

Елементи оточення мають контрастну палітру кольорів. Це зроблено таким чином, щоб передати гравцю загальний ігровий настрій, що сприятливо позначається на процесі геймплею.

					КРМ.КІ.1.884-03.1.1	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

2.4.3 Штучний інтелект

Для будь-якого жанру гри штучний інтелект є невід'ємною частиною геймплею.

Ігровий штучний інтелект (AI) – це область штучного інтелекту, яка спеціалізується на розробці і використанні технологій та алгоритмів для створення імітації людського мислення і поведінки у відеоіграх і ігрових середовищах. Ігровий штучний інтелект використовується для покращення геймплею, створення більш реалістичних персонажів і опонентів, а також для оптимізації графіки та фізики в іграх [1].

Основні області застосування ігрового штучного інтелекту:

1. Розробка і вдосконалення алгоритмів прийняття рішень для комп'ютерних персонажів. Це включає в себе розумне ведення персонажів, прийняття рішень щодо їх дій у грі і взаємодію з іншими об'єктами в ігровому світі.

2. Створення систем штучного інтелекту для створення враження гравця про реалістичність поведінки ігрових персонажів. Це включає в себе реалістичну емоційну реакцію персонажів на події в грі.

3. Розробка алгоритмів для оптимізації графіки та фізики в реальному часі для покращення візуального і фізичного реалізму ігрового світу.

4. Використання машинного навчання і нейронних мереж для покращення ігрового досвіду, зокрема для підвищення ігрової складності, покращення адаптивності гри до стилю гравця та інших аспектів.

Ігровий штучний інтелект є важливою частиною розвитку сучасних відеоігор і сприяє створенню більш захоплюючих та реалістичних ігрових досвідів.

Загальний алгоритм дій ШІ клієнта представлено на рисунку 2.4

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 2.4 – Загальний алгоритм дій клієнта

Загальний алгоритм дій ШІ співробітника представлено на рисунку 2.5

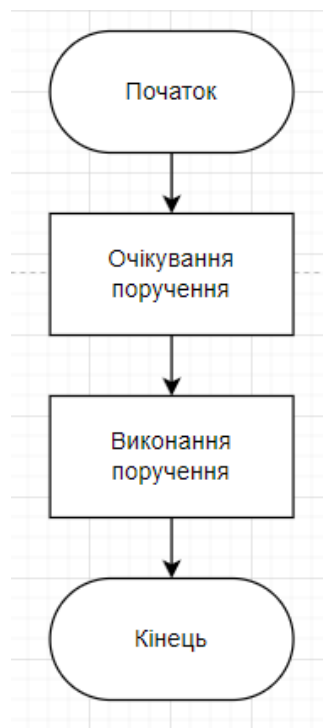


Рис. 2.5 – Загальний алгоритм дій співробітника

2.4.4 Елементи ігрового спорядження

У грі представлено декілька видів об'єктів приготування.

Гравець має створювати різноманітні блюда.

Таблиця 2.1

Види об'єктів приготування

Назва	Рецепт
 <p>Рис 2.7 – Чізбургер</p>	<p>Біфштекс із натуральної яловичини, шматочок сиру “Чеддер”, шматочок маринованого огірка та цибуля, заправлені гірчицею і кетчупом, у булочці з пшеничного борошна.</p> <p>Ігровий рецепт: котлета – 1 шт. булочка – 2 шт. сир – 1 шт.</p>
 <p>Рис 2.8 – Дабл Чізбургер</p>	<p>Два біфштекси з натуральної яловичини, два шматочки сиру «Чедер», два мариновані огірки, цибуля, гірчиця, кетчуп, булочка.</p> <p>Ігровий рецепт: котлета – 2 шт. булочка – 2 шт. сир – 2 шт. бекон – 2 шт.</p>
 <p>Рис 2.9 – Біг Мак</p>	<p>Два біфштекси з натуральної яловичини, цибуля, маринований огірок, сир “Чеддер”, свіжий салат, заправлені спеціальним соусом, у булочці з насінням сезаму.</p> <p>Ігровий рецепт: котлета – 2 шт. булочка – 2 шт. сир – 1 шт. лист салату – 1 шт. шматочок помідора – 1 шт. шматочок цибулі – 1 шт.</p>

Змн.	Арк.	№ докум.	Підпис	Дата

KPM.KI.1.884-03.1.1

Арк.

28

Назва	Рецепт
 <p data-bbox="252 600 718 689">Рис 2.13 – Сэндвіч <i>Mediterranean Veggie</i></p>	<p data-bbox="753 250 1487 340">Це смачний і корисний бутерброд, який можна приготувати зі свіжих інгредієнтів.</p> <p data-bbox="753 353 1008 398">Ігровий рецепт:</p> <ul style="list-style-type: none"> <li data-bbox="753 412 1161 448">шматочок шинки – 2 шт. <li data-bbox="753 465 944 501">хліб – 2 шт. <li data-bbox="753 519 932 555">сир – 1 шт. <li data-bbox="753 573 1059 609">лист салату – 1 шт. <li data-bbox="753 627 1114 663">смажене яйце – 1 шт. <li data-bbox="753 680 1193 716">шматочок помідору – 2 шт. <li data-bbox="753 734 1152 770">шматочок цибулі – 1 шт. <li data-bbox="753 788 1145 824">шматочок огірця – 2 шт.

2.4.5 Ігрові бонуси

У грі представлено ігрова валюта, яка потрібна для покращення геймплею. Гравець накопичує ігрову валюту, виконуючи замовлення клієнтів. Кількість ігрової валюти, отриманої за виконане замовлення залежить від складності замовлення.

За допомогою ігрової валюти можна покращити характеристики гравця, наприклад, швидкість пересування, швидкість приготування, зменшення закупівельної ціни та інше.

В якості інтерфейсу для використання валюти був створений *NPC*.



Рис 2.14 – *NPC* для використання валюти

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

2.4.6 Платформа

Гра розробляється на ПК для операційної системи *Windows*. У майбутньому можливо здійснити експорт на інші операційні системи. Системні вимоги для гри приведені у таблиці 2.2.

Таблиця 2.2

Системні вимоги гри

Параметр	Мінімальні вимоги	Рекомендовані вимоги
ОС	<i>Windows 10/11</i>	<i>Windows 10/11</i>
Процесор	<i>Intel Core i3-10100F</i> <i>3.6GHz</i>	<i>Intel Core i5-12100F</i> <i>4.0 GHz</i>
Оперативна пам'ять	<i>8GB DDR4</i>	<i>16GB DDR4</i>
Відеокарта	<i>NVIDIA GTX 1050ti</i>	<i>NVIDIA GTX 1070</i>
Місце на диску	<i>7 GB HDD</i>	<i>7 GB SSD</i>

2.4.7 Модель гри

Запускаючи гру, гравець у першу чергу бачить меню гри. На фоні зображено ресторан. На першому плані зображено кнопки управління головним меню.

Меню має три кнопки: «нова гра», «налаштування», та «вихід». Натиснувши на першу кнопку, гравець має можливість вибрати: чи почати гру з навчанням.

У розділі «налаштування» головного меню, гравець має змогу змінити роздільність екрана, якість графіки. Кнопка «вихід» дає змогу покинути гру.

Меню зображено на рисунку 2.15, меню налаштувань на рисунку 2.16, вибір важкості на рисунку 2.17.



Рис 2.15 – Меню

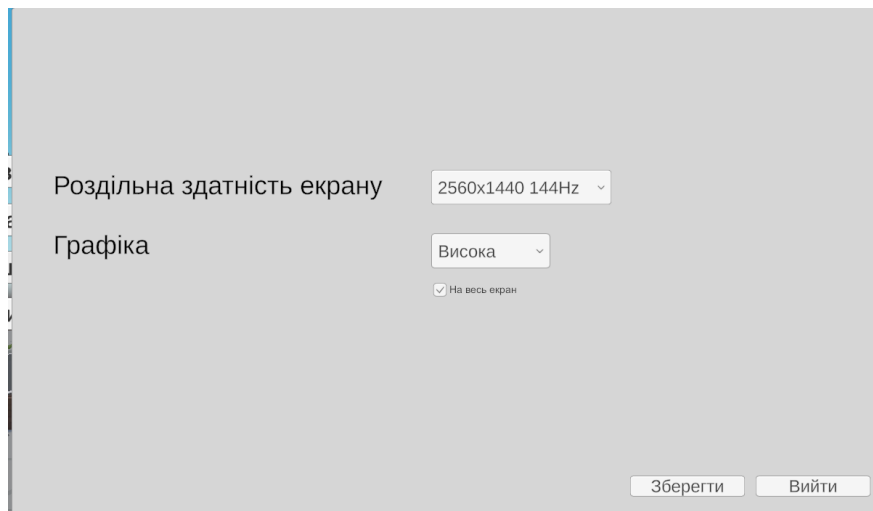


Рис 2.18 – Меню налаштувань

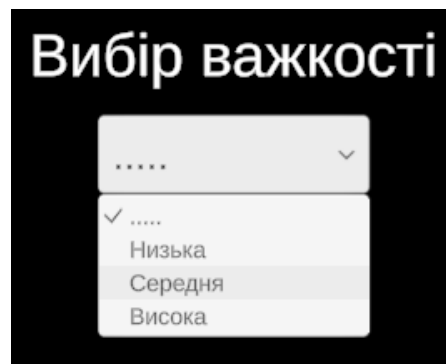


Рис 2.17 – Вибір важкості нової гри

2.4.8 Формули

Розроблена гра має деяку кількість важливих формул, розрахунок котрих необхідний для коректної роботи гри.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

В Unity, компонент *RaycastHit* використовується для виявлення перетину променя з поверхнею в 3D-просторі. Формула, яка використовується для цього в Unity, подібна до формули для лінії променя та формули площини.

Математично це може бути представлено так: Нехай P_0 – точка початку променя (*origin*), d – вектор напрямку променя (*direction*), t – параметр, який визначає точку перетину, P – точка перетину.

Формула променя може бути визначена як:

$$P = P_0 + t \cdot d.$$

Якщо ми маємо поверхню в формі площини, визначену її нормаллю N та точкою P_{plane} , можемо використати рівняння площини:

$$N \cdot (P - P_{plane}) = 0.$$

Підставимо формулу променя у це рівняння:

$$N \cdot (P_0 + t \cdot d - P_{plane}) = 0.$$

З цього рівняння можна визначити значення параметра t , яке вказує на відстань вздовж напрямку променя до точки перетину.

У Unity функція переміщення гравця може бути представлена у вигляді лінійного руху з використанням векторів. Найпростіший спосіб визначити цю функцію – це використати лінійну інтерполяцію (LERP) між початковою і кінцевою позиціями гравця.

Математично це виглядає так: P_0 – початкова позиція гравця, P_1 – кінцева позиція гравця, t – параметр інтерполяції (зазвичай від 0 до 1).

Формула LERP:

$$P(t) = (1 - t) \cdot P_0 + t \cdot P_1.$$

Ця формула обчислює нову позицію гравця ($P(t)$) в залежності від параметра t . Якщо $t=0$, то $P(t)$ дорівнює P_0 , якщо $t=1$, то $P(t)$ дорівнює P_1 , а при $0 < t < 1$ отримуємо проміжні значення між P_0 і P_1 .

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

Формула часу очікування замовлення залежно від вибраної важкості:

$$T = (n \cdot t) \cdot d,$$

де: n – кількість позицій замовлення;

t – константа часу;

d – рівень важкості.

Формула перерахунку вартості інгредієнтів залежно від вибраної важкості:

$$P = \frac{p}{d}$$

де: p – початкова ціна інгредієнту;

d – рівень важкості.

2.5 Графіка та анімація

Таблиця 2.3

Графічні елементи гри

Елемент	Опис
Клієнти	Жінки та Чоловіки
Страви	Бургери, сендвічі, картопля, торти, напої
Інгредієнти	М'ясо, овочі, яйця, соуси, булочки, солодощі
Елементи інтер'єру	Кухня, стільця, столи, ліхтарики, картини, меню
Елемент	Опис
Погодні умови	Сонце, хмари, небо, туманність
Елементи інтерфейсу	Кнопки, графічні зображення, спливаючі підказки

Анімації гри

Тип анімації	Опис
Анімація простою	Програється під час бездіяльності клієнта
Анімація спокійного крокування	Звичайний шаг
Анімація розмови	Програється під час замовлення клієнта
Анімація переміщення з замовленням	Програється під час переміщення до вільного столу зі створеним замовленням
Анімація процесу прийняття їжі	Програється під час прийняття їжі за столом

2.6 Звуки та музика

Музикальні ефекти гри

Елемент	Тривалість
Фонова музика головного меню	Поки головне меню активне
Фонова музика під час геймплею	Поки гравець не натисне на паузу
Звуки спокійного шага	Під час переміщення гравця, 1 секунда
Звуки взаємодії з інтерактивними елементами	Під час взаємодії з елементами, 1 секунда
Звук отримання ігрової валюти	Після успішно виконаного замовлення, 2 секунди
Звук роботи гриля	Під час роботи гриля

Елемент	Тривалість
Звук нарізання інгредієнтів	Під час взаємодії з нарізною дошкою, 3 секунди
Звук створення напою	Під час взаємодії з нарізною дошкою, 1 секунда

2.8 Критерії оптимізації та оцінки

Оцінка геймплею в симуляторі кулінарії може базуватися на різноманітних критеріях, які визначають задоволення від гри та реалістичність ігрового процесу.

1. Реалістичність ігрового процесу: оцінка реалістичності процесу приготування страв, включаючи етапи підготовки і обробки інгредієнтів, використання кухонного інвентарю і техніки.

2. Креативність та варіативність рецептів: наявність різноманітних рецептів, які можна готувати. Можливість експериментувати з інгредієнтами та створювати власні унікальні страви.

3. Розвиток кулінарних навичок: прогресія у розвитку вмінь гравця, можливість вивчати нові рецепти та отримувати нові навички.

4. Менеджмент часу та ресурсів:

– ефективно управління часом для успішного виконання замовлень або приготування страв для гостей;

– оптимальне використання інгредієнтів та кухонного обладнання.

5. Динаміка гри: події та швидкість їх надходження, які можуть впливати на геймплей, такі як несподівані відвідування гостей, конкуренція з іншими шеф-кухарями тощо.

6. Якість графіки та анімація: якість графіки та реалістичність анімацій приготування страв.

7. Якість звукове супроводження: свіжі та реалістичні звуки кухні, атмосферна музика, можливість комунікації з персоналом або гостями.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

8. Складність проходження:

- складність завдань та рецептів: Оцінка того, наскільки складними є завдання та рецепти, які гравець повинен виконати;
- прогресія важкості: Зміна важкості гри з часом або залежно від досягнутого рівня;
- можливості налаштувань важкості: Наявність опцій, які дозволяють гравцю вибрати рівень важкості, щоб гра відповідала його навичкам та вподобанням.

Було проведено опитування серед 5 геймерів за кожним критерієм та визначені середні оцінки.

Таблиця 2.6

Середні оцінки гри

Критерій	Середня оцінка
Реалістичність приготування страв	8
Креативність та варіативність рецептів	5
Розвиток кулінарних навичок	5
Менеджмент часу та ресурсів	7
Динаміка гри	6
Графіка та анімації	6
Звукове супроводження	6
Складність проходження	6

Для поліпшення середньої оцінки у майбутньому було проведено дослідження доступних рішень.

1. Реалістичність приготування страв:

- використання фізичного двигуна: Додавання фізичного двигуна для інгредієнтів, що дозволяє їм взаємодіяти з навколишнім середовищем реалістичніше;

– моделювання температурних ефектів: Введення системи температур, яка впливає на приготування та текстуру страв.

2. Креативність та варіативність рецептів:

– система кастомізації страв: Можливість змінювати інгредієнти та експериментувати з рецептами на різних етапах гри;

– збільшення кількості рецептів: Додавання постійних оновлень або можливості завантажувати нові рецепти.

3. Розвиток кулінарних навичок: глибокий прогресивний режим: Додавання більше етапів розвитку кулінарії, можливість отримувати навички у специфічних областях кулінарії.

4. Менеджмент часу та ресурсів:

– розширення системи ресурсів: Додавання нових ресурсів або розширення системи використання інгредієнтів;

– динамічний час: Зміна умов гри впливає на час приготування, наприклад, підвищення вимог до часу в залежності від складності страви.

5. Динаміка гри:

– випадкові події: Додавання випадкових подій, які можуть виникати під час гри та впливати на геймплей;

– сценарії та історії: Введення інтригуючих історій або сценаріїв, які розвиваються під час гри.

6. Графіка та анімації:

– вдосконалені анімації приготування страв: Деталізація анімацій, які дозволяють гравцеві більше взаємодіяти з процесом готування;

– реалістичні ефекти освітлення: Додавання вражаючих світлових ефектів, які підкреслюють атмосферу кухні.

7. Звукове супроводження:

– Ембієнтний звук кухні: Деталізація атмосферного звуку кухні, щоб гравець відчувався, ніби він дійсно готує страви;

– голосові коментарі: Введення голосових коментарів персоналу або гостей.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

Оптимізація важкості гри може бути досягнута за допомогою різноманітних варіацій та аспектів геймплею. Ось кілька конкретних підваріантів для оптимізації важкості гри:

1. Складність завдань та рецептів:

– градація завдань: Поступове збільшення складності завдань та рецептів з прогресом гравця;

– варіація рецептів: Додавання варіантів рецептів з різним рівнем складності, які гравець може вибирати.

2. Прогресія важкості:

– адаптація до гравця: Система, яка адаптує важкість гри в залежності від навичок та досвіду гравця;

– навчальний режим: Спрощений режим для новачків, який зменшує важкість завдань та рецептів.

3. Можливості налаштувань важкості:

– система вибору важкості: Опції, які дозволяють гравцеві вибирати рівень важкості в процесі гри;

– наступність вибору важкості: Забезпечення можливості змінити важкість перед кожним завданням або рівнем.

4. Баланс між часом та вимогами:

– гнучкі терміни: Збільшення або зменшення обмежень часу для завдань в залежності від вибраного рівня важкості;

– система порад чи допомоги: Надання гравцеві підказок або додаткового часу при необхідності.

5. Рандомізація:

– випадкові події: Додавання випадкових факторів, які можуть впливати на важкість гри, створюючи більше непередбачуваності;

– випадкові обставини: Наявність факторів, які можуть зробити завдання легше чи важче при кожному запуску.

6. Сценарії та історії:

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

– важкість у залежності від сюжету: Зв'язок важкості гри з ходом сюжету або інтересуючими сценаріями.

Ці варіації дозволяють розширити та вдосконалити кожен критерій, щоб зробити гру більш захопливою та задовільною для гравців.

2.9 Методи багатокритерійної оптимізації

Методи багатокритерійної оптимізації використовуються для вирішення задач, де існує більше одного критерію оптимізації, і ці критерії часто конфліктують між собою. В таких випадках ціль полягає в тому, щоб знайти набір рішень, які, утім важко чи неможливо поліпшити за одним критерієм, не погіршуючи при цьому інші.

1. Метод збалансованих оцінок (Weighted Sum Method):

– опис: кожен критерій має ваговий коефіцієнт, який відображає його важливість. Функція вартості обчислюється як сума критеріїв, помножених на їх вагові коефіцієнти.

– переваги: простий у реалізації і розумінні.

– недоліки: не враховує можливих взаємозв'язків між критеріями, і вагові коефіцієнти можуть бути суб'єктивними.

2. Метод гіперпараметричного програмування (Pareto-based Methods):

– опис: використовує принцип Парето-оптимальності. Рішення, які не можуть бути поліпшені за одним критерієм, не погіршуючи інших, утворюють множину Парето.

– переваги: враховує компроміси між критеріями та дозволяє знаходити ефективні рішення.

– недоліки: велика обчислювальна складність, особливо при зростанні кількості критеріїв.

3. Метод аналізу ієрархій (Analytic Hierarchy Process - AHP):

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

– опис: використовує ієрархічні структури для представлення критеріїв та їх взаємозв'язків. За допомогою парних порівнянь отримуються вагові коефіцієнти для кожного критерію.

– переваги: враховує ієрархію важливості критеріїв, допомагає уникнути суб'єктивності при визначенні ваг.

– недоліки: вимагає консенсусу щодо парних порівнянь, може бути чутливий до невизначеності.

4. Метод еволюційної оптимізації (Evolutionary Algorithms):

– опис: використовує генетичні алгоритми, еволюційні стратегії чи інші еволюційні підходи для пошуку оптимальних рішень в просторі багатовимірних об'єктів.

– переваги: добре справляється з нелінійністю та нестабільністю задач оптимізації.

– недоліки: великі витрати обчислювальних ресурсів, складно визначити параметри алгоритму.

Ці методи використовують різні підходи для вирішення проблем багатокритерійної оптимізації в залежності від конкретного контексту задачі і вимог дослідження.

2.1 Nav Mesh Agent

В якості інструменту реалізації *Pathfinding Algorithms* використано *Nav Mesh Agent*.

Nav Mesh Agent – це компонент у багатьох гральних рушіях та інших середовищах для розробки ігор, який використовується для навігації об'єктів (часто юнітів або інших гральних персонажів) в 3D-середовищі за допомогою навігаційної сітки (Nav Mesh).

Основні поняття, пов'язані з Nav Mesh Agent та юнітами:

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

1. nav Mesh (навігаційна сітка): це віртуальна сітка або мережа, яка репрезентує доступні для навігації області в 3D-середовищі. Ця сітка використовується для планування шляхів та керування рухом об'єктів;

2. nav Mesh Agent (агент навігаційної сітки): це компонент, який може бути прикріплений до грального об'єкта (юніта). Він відповідає за навігацію об'єкта вздовж навігаційної сітки. Nav Mesh Agent автоматично обчислює шляхи та керує рухом юніта, враховуючи перешкоди, терейн та інші обмеження в середовищі;

3. агенти навігаційної сітки: В середовищах ігор можуть бути кілька Nav Mesh Agents, які рухаються незалежно один від одного. Кожен агент може мати власні параметри руху, такі як швидкість, радіус об'єкта і так далі;

4. планування шляху: Nav Mesh Agent використовує алгоритми для обчислення оптимального шляху від поточного положення до цільової точки на навігаційній сітці;

5. оновлення шляху в реальному часі: nav Mesh Agent може оновлювати свій шлях, якщо з'являються перешкоди або цільова точка змінюється;

6. анімація руху: після обчислення шляху і керування рухом агента, відповідні анімації можуть бути відтворені для реалістичного відображення руху юніта;



Рис 2.22 - Навігаційна сітка

					КРМ.КІ.1.884-03.1.1	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

На рисунку 2.22 показана навігаційна сітка створена за допомогою Nav Mesh Agent.

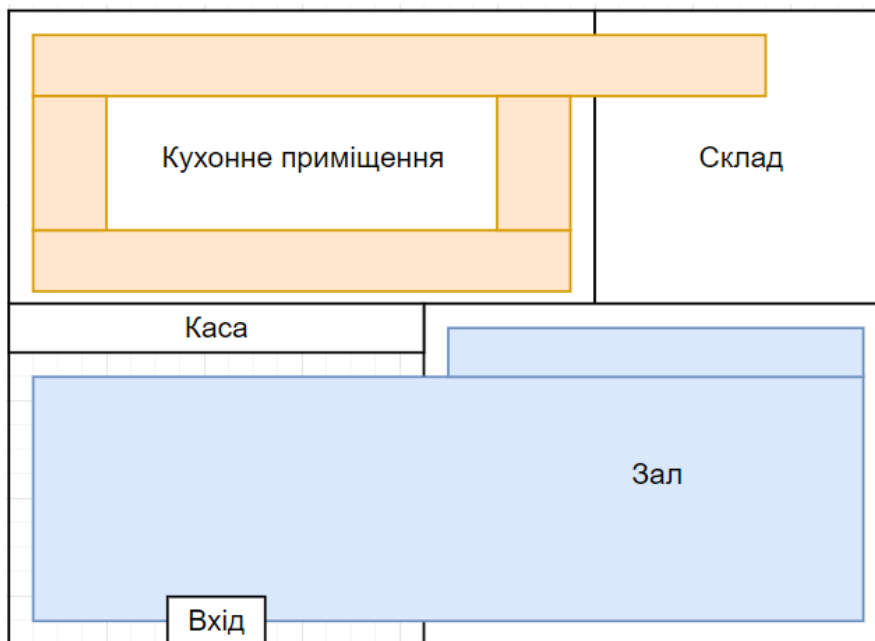


Рис. 2.23 – Структурне представлення голової сцени

За допомогою Nav Mesh Agent було створено дві зони переміщення ШІ, помаранчева зона переміщення співробітника, а синя – зона переміщення клієнтів.

Unity NavMeshAgent використовує алгоритми оптимізації для навігації штучного інтелекту в ігровому середовищі. Основні алгоритми, які використовуються під капотом, включають в себе:

1. navigation Mesh (NavMesh): unity використовує меш як основну структуру даних для навігації. NavMesh є 3D-мешем, який представляє поверхню, по якій можуть переміщатися об'єкти;

2. a (A-Star) Algorithm: * A* є одним з найпоширеніших алгоритмів пошуку шляху. Він використовує евристичні дані для ефективного визначення шляху між двома точками на NavMesh;

3. обрізка шляху (Path Clipping): деякі додаткові алгоритми використовуються для обрізки шляху і врахування обмежень, таких як стіни та інші перешкоди;

4. локальне переміщення (Local Avoidance): для уникнення зіткнень у реальному часі, NavMeshAgent використовує алгоритми локального уникнення, які дозволяють об'єктам уникати одне одного при русі;

5. dynamic Obstacle Avoidance: unity також підтримує уникнення динамічних перешкод. Якщо об'єкт, що рухається, стає перешкодою для іншого об'єкта, NavMeshAgent може автоматично реагувати і обирати новий шлях.

Ці алгоритми розроблені таким чином, щоб забезпечити швидку та ефективну навігацію для об'єктів у ігрових сценах. Unity автоматично керує цими алгоритмами під час використання NavMeshAgent, спрощуючи процес для розробників гри.

Висновки до другого розділу

1. У процесі проектування було розроблено концептуальний документ гри, необхідний для демонстрації інвестору або видавцю проекту.

2. Розроблено дизайнерський документ гри, який описує основні особливості розроблюваної гри..

3. У цьому розділі були ретельно розглянуті та визначені ключові критерії для оптимізації ігрового процесу симулятора кулінарії. Вони включають в себе аспекти, такі як реалізм готування, взаємодія з ігровим оточенням та задоволення користувача.

4. З урахуванням виявлених критеріїв були визначені потенційні напрямки оптимізації ігрового процесу симулятора кулінарії. Це включає в себе вдосконалення алгоритмів готування, реалістичніше моделювання інгредієнтів та розвиток нових ігрових функцій.

5. Для успішної реалізації запропонованих вдосконалень необхідно враховувати технічні обмеження, такі як обсяг пам'яті, обчислювальна потужність та можливості різних платформ.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3

РОЗРОБКА ДЕМОНСТРАЦІЙНОЇ ВЕРСІЇ ГРИ

3.1 Вибір засобів розробки

Для реалізації комп'ютерної гри був обраний ігровий рушій Unity 2021.3.12f1. Unity – це популярний і потужний ігровий двигун і розробкове середовище, яке використовується для створення ігор, віртуальної реальності, інтерактивних застосунків і симуляцій. Він має багато переваг і використовується для розробки ігор для різних платформ, включаючи ПК, мобільні пристрої, консолі та веб.

Основні переваги Unity включають:

- кросплатформеність: unity підтримує багато платформ, що дозволяє створювати ігри для різних операційних систем і пристроїв, включаючи Windows, macOS, iOS, Android, консолі та веб;
- мова програмування: unity використовує мову програмування C#, яка є досить популярною і має велику спільноту розробників;
- графічний інструментарій: Unity надає потужний інструментарій для роботи з 2D та 3D графікою, включаючи підтримку фізики, світла, тіней та спеціальних ефектів;
- гнучкість: unity дозволяє розробникам створювати ігри різних жанрів та типів, включаючи ігри віртуальної реальності та аугментованої реальності;
- велика спільнота: unity має активну спільноту розробників, що допомагає знайти підтримку, документацію, плагіни та ресурси;
- asset Store: unity Asset Store містить безліч безкоштовних та платних активів, які полегшують розробку ігор;

					<i>KPM.KI.1.884-03.1.1</i>	Арк. 45
Змн.	Арк.	№ докум.	Підпис	Дата		

– безкоштовна версія: unity надає безкоштовну версію для початківців і невеликих команд, що дозволяє вивчити і розробляти ігри без великих витрат.

У якості середовища розробки було обране середовище програмування Microsoft Visual Studio 2019, котре легко інтегрується у редактор Unity.

Для розробки було обрано мову програмування C#, підтримувану Unity.

C# є дуже популярною і могутньою мовою програмування для розробки ігор у середовищі Unity. Ось деякі переваги використання C# для розробки ігор:

– інтеграція з Unity: c# є однією з основних мов програмування для Unity, і у вас є прямий доступ до всіх функцій та можливостей двигуна;

– легка вивченість: c# вважається однією з найбільш доступних мов програмування для початківців. Вона має чіткий синтаксис, добре документована і має широкий вибір навчальних матеріалів та онлайн-ресурсів;

– підтримка від Microsoft: c# розробляється і підтримується Microsoft, що робить її стабільною та актуальною мовою програмування;

– велика спільнота: є велика спільнота розробників, яка готова надавати підтримку та допомогу вирішувати проблеми, пов'язані з розробкою ігор на C# у Unity;

– широкий вибір бібліотек і плагінів: існують багато бібліотек та плагінів, які допомагають розширити можливості C# в Unity, надаючи готові рішення для різних завдань;

– ефективність та продуктивність: c# є високорівневою мовою, яка дозволяє розробляти ігри швидше та ефективніше, зменшуючи кількість рутинних операцій і сприяючи зосередженню на креативному процесі розробки;

– кросплатформеність: гри, розроблені на C# у Unity, можуть бути легко виведені на різні платформи, включаючи ПК, мобільні пристрої, консолі та інші.

У Unity Hub можна обрати потрібний проєкт та версію редактора, якою ми бажаємо скористатися.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

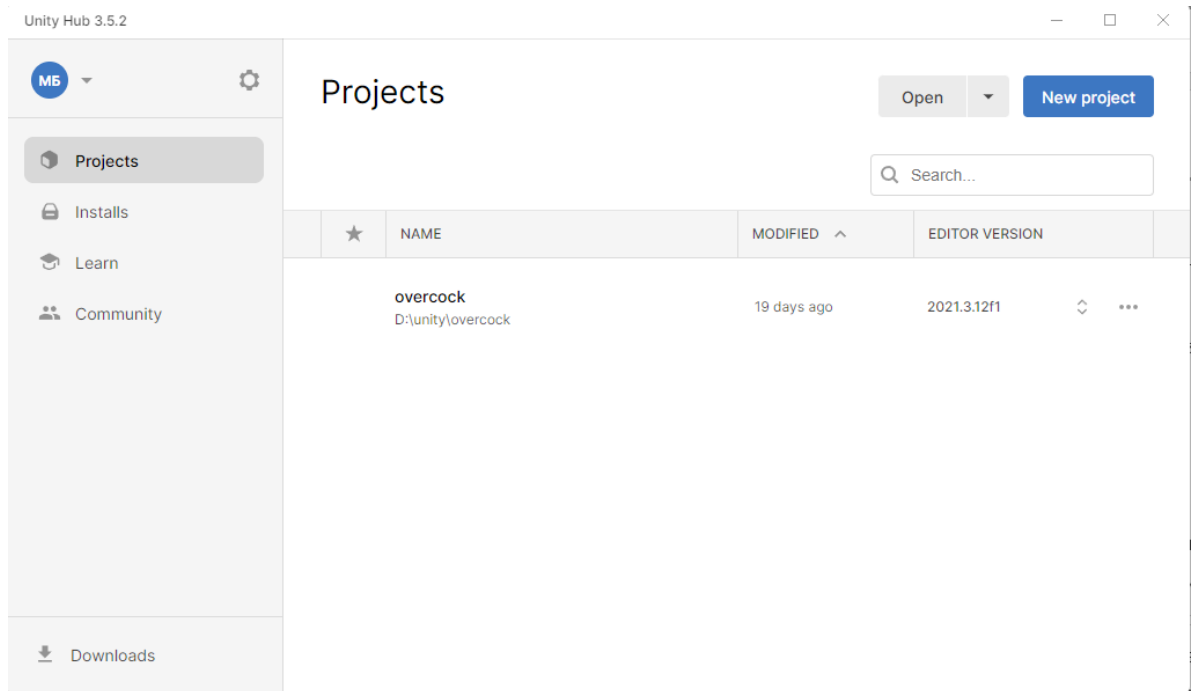


Рис. 3.1 – Вікно вибору проектів у Unity Hub

3.2 Початок розробки

3.2.1 Створення головної сцени

Інтерфейс Unity має досить характерну структуру, яка складається з різних панелей, вкладок та інструментів. Ось короткий огляд основних елементів інтерфейсу Unity:

- сцена (Scene): у цій області можна переглядати та редагувати сцену в режимі 2D або 3D. Можна переміщати, обертати та масштабувати об'єкти, а також взаємодіяти з ними.

- ієрархія (Hierarchy): ця панель показує ієрархічну структуру всіх об'єктів на сцені. Можна переглядати та управляти об'єктами, а також створювати батьківсько-дитячі відносини між ними.

- проєкт (Project): тут відображається вміст проєкту, включаючи всі ресурси, такі як моделі, текстури, звуки, скрипти тощо. Можна організувати файли у папках і шукати ресурси.

- інспектор (Inspector): ця панель надає інформацію та налаштування для вибраного об'єкта. Можливість змінювати параметри об'єкта, додавати компоненти та взаємодіяти з ним у цій панелі.
- гра (Game): ця область відображає гру в реальному часі, коли запускається у режимі перегляду. Можливість використовувати це вікно для тестування гри.
- аніматор (Animator): ця панель призначена для створення і редагування анімацій об'єктів в грі. Можливість встановлювати ключові кадри, створювати анімаційні переходи та управляти анімацією.
- мінікарта (Scene View): мінікарта надає загальний огляд сцени та дозволяє швидко навігуватися по ній.
- управління (Hierarchy / Project / Console): ці вкладки дозволяють перемикатися між різними панелями, такими як ієрархія, проєкт і консоль для виводу повідомлень та помилок.
- головне меню (Main Menu): в головному меню знаходиться доступ до основних опцій і команд Unity, таких як створення нових сцен, імпорт ресурсів, налаштування проєкту тощо.
- панель інструментів (Toolbar): панель інструментів містить корисні інструменти для редагування сцени, збереження змін, перегляду гри тощо;
- програвач (Play Button): кнопка "Play" дозволяє запускати і відтворювати гру в режимі перегляду.



Рис. 3.2 – Головна сцена гри

					<i>КРМ.КІ.1.884-03.1.1</i>	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

Всі об'єкти на сцені структуровані за допомогою `GameObject` у вікні ієрархії. Усі об'єкти розставлені вручну на головній сцені. Створені кореневі об'єкти, такі як *Player*, *Exterior*, *Props*, *UICanvas*, що містять префаби головного героя, моделі навколишнього середовища, інтерактивних елементів та елементи інтерфейсу гравця.

3.2.2 Створення ігрового інтерфейсу

Для реалізації інтерфейсу гравця створено об'єкт *UICanvas*, який служить для відображення ігрових елементів безпосередньо на екрані гравця.

Canvas в *Unity* – це інструмент для створення і редагування інтерфейсу користувача (*UI*) у грі. Він надає зручний інтерфейс для розміщення і налаштування різних елементів інтерфейсу, таких як тексти, кнопки, зображення, поля вводу та інші

Даний об'єкт містить наступні графічні елементи: відображення кількості внутрішньоігрової валюти, відображення поточного замовлення, таймер замовлення, модальні вікна інтерактивних елементів, вікно діалогу, вікно паузи.



Рис. 3.3 – Відображення кількості внутрішньоігрової валюти



Рис. 3.4 – Відображення поточного замовлення



Рис. 3.5 – Таймер замовлення

Добрий день, мені, будь ласка,
DoubleCheeseBurger, MacMuffinWithBacon, BigMac

Press E to next...

Рис. 3.6 – Вікно діалогу



Рис. 3.7 – Модальні вікна інтерактивних елементів



Рис. 3.8 – Вікно паузи

Для реалізації головного меню гри у каталогі ігрових префабів проекту була створена окрема сцена – *MainMenu*.

					КРМ.КІ.1.884-03.1.1	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

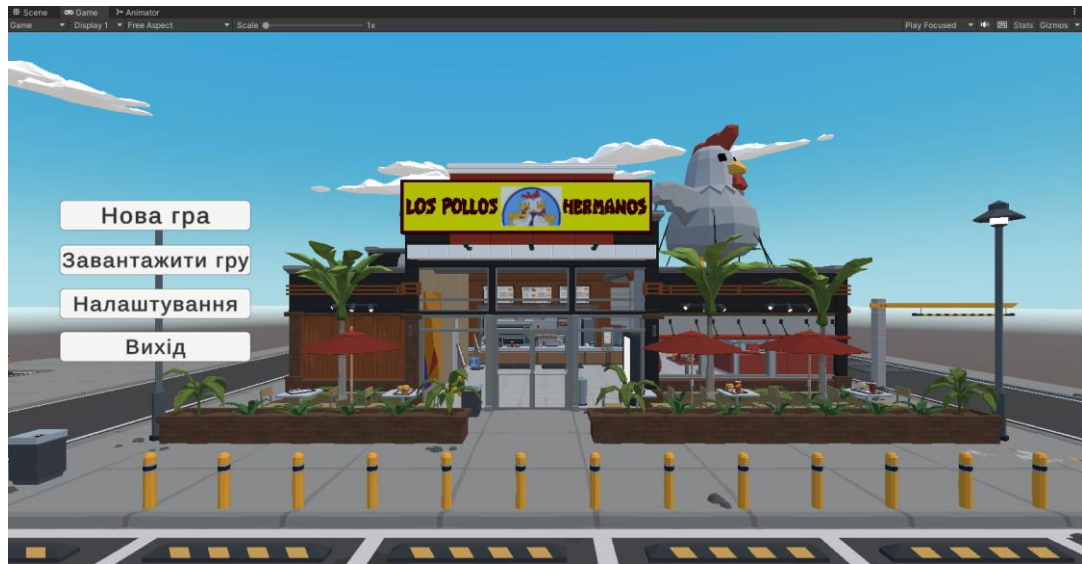


Рис. 3.9 – Головне меню гри

Вона має у своєму складі наступні елементи: *Canvas* – основний графічний об’єкт інтерфейсу, *MenuManager* – елемент, який включає до себе скрипт реалізації основних кнопок інтерфейсу

Canvas – має у своєму складі наступні елементи: *NewGame Button* – кнопка початку нової гри, *Load Button* – кнопка завантаження гри, *Setting Button* – кнопка налаштувань гри, включає налаштування графіки, роздільну здатність екрана, *Exit Button* – кнопка для виходу з гри.

При виборі нової гри, відкривається модальне вікно яке пропонує пройти навчання.

Кожна кнопка має у своєму складі елемент *Button*, зі станом *OnClick*, який активується при безпосередньому натисканні на ігровий об’єкт. До нього додано ігровий об’єкт зі скриптом, який викликає потрібний метод. Усі потрібні методи прописані у елементі *MenuManager*.

3.3 Створення префабів

Префаб – це шаблон об’єкта або групи об’єктів, який може бути створений та змінений в одному місці і використовуваний в багатьох місцях у гри. Це допомагає уникнути дублювання коду та ресурсів.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

У *Unity* префаби (*Prefab*) є потужним інструментом для ефективного управління об'єктами та їх ресурсами в грі. Префаби дозволяють створювати, зберігати і використовувати групи об'єктів як один об'єкт.

У грі реалізовані префаби різних типів:

- префаби клієнтів;
- префаби інгредієнтів;
- префаби готових страв;
- префаби інструментів;
- префаби інтер'єру.

3.3.1 Префаби клієнтів

У грі реалізовано 10 префабів клієнтів, кожен має свою анімацію.

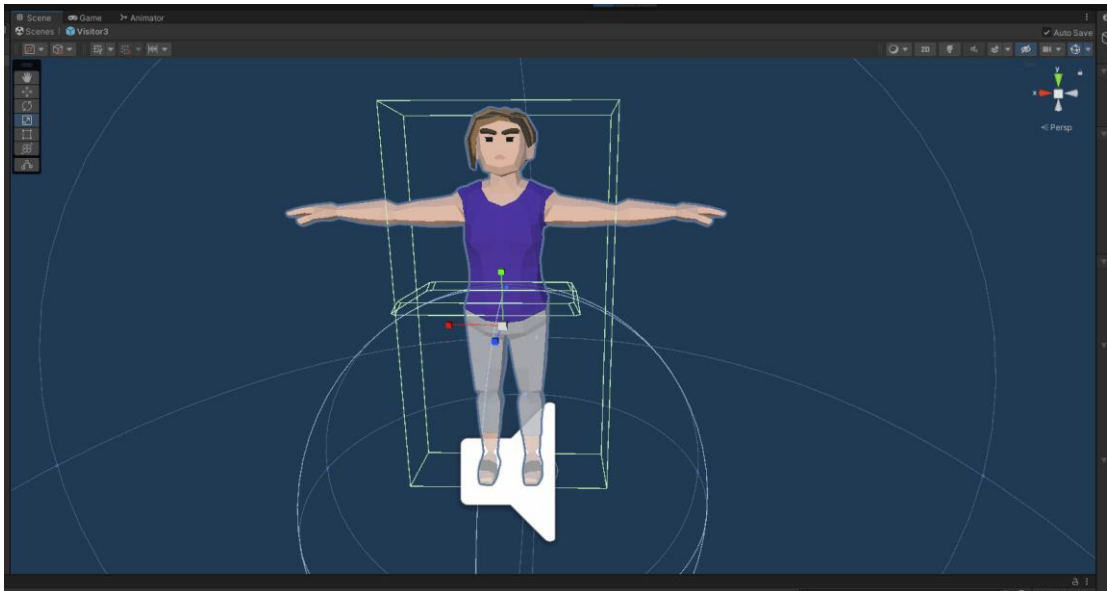


Рис. 3.10 – Префаб клієнта

В *Unity*, компонент *Animator* є ключовим для створення та управління анімаціями об'єктів у грі. *Animator* дозволяє вам визначити різні стани та переходи між ними, а також керувати анімаціями на основі параметрів і подій. Основні поняття та функції компонента *Animator*:

– стан (*State*): стан визначає конкретну анімацію або поведінку об'єкта в певний момент часу. Наприклад, це може бути стоячий стан, біг, стрибок, атака тощо. Кожен стан містить анімації, які пов'язані з цим станом;

					КРМ.КІ.1.884-03.1.1	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

– перехід (*Transition*): перехід – це зв'язок між станами, що вказує, як об'єкт переходить з одного стану в інший. Можливість визначити умови для переходів, такі як значення параметрів або виконання певних подій;

– параметри (*Parameters*): параметри – це значення, які використовуються для умов переходів між станами. Вони можуть бути числовими, булевими або типом "триггер". Параметри дозволяють динамічно змінювати поведінку об'єкта;

– аніматор контролер (*Animator Controller*): це скрипт, який керує всіма анімаціями і переходами для конкретного об'єкта. Можливість створити аніматор контролер у *Visual Animator Controller* або просто додавши компонент *Animator* на об'єкт;

– анімаційний *Clip (Animation Clip)*: це конкретна анімація, яка визначає рухи об'єкта. Анімаційні кліпи можуть бути додані до станів або переходів;

– події (*Events*): можливість встановлювати події в анімаційних кліпах або станах, щоб викликати код в грі під час відтворення анімації.

Параметри *Animator Controller*: можливість змінювати значення параметрів, щоб керувати переходами та анімаціями. Це дозволяє реагувати на дії гравця або події в грі.

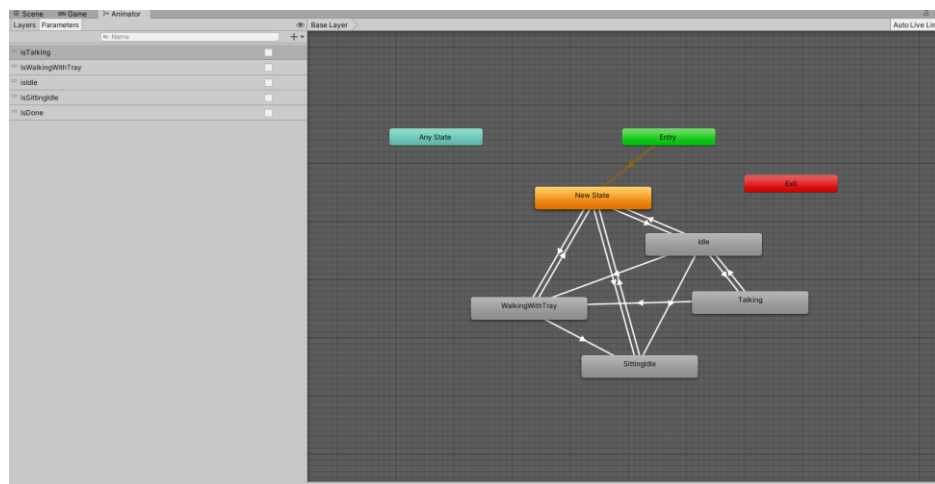


Рис. 3.11 – Компонент *Animator* клієнта

Кожен клієнт має скрипт *Visitor*, *Persistence Item*, компонент *Audio Source* та *Nav Mesh Agent*.

Компонент "*Audio Source*" в *Unity* – це компонент, який дозволяє відтворювати звукові ефекти та аудіо у грі. Він призначений для додавання аудіо до об'єктів та керування різними аспектами відтворення звуку. Ось основні функції та властивості компонента "*Audio Source*":

– *audio Clip*: ця властивість визначає аудіофайл, який буде відтворюватися компонентом "*Audio Source*". Можливість прикріпити до неї аудіофайл, який відтвориться;

– *volume*: властивість *Volume* визначає гучність аудіо. Можливість налаштувати гучність від 0 (тишина) до 1 (повна гучність);

– *pitch*: ця властивість дозволяє змінювати висоту тону аудіо. Значення 1 відповідає нормальному тону, і можливість зменшувати його для зміни висоти тону або збільшувати для підвищення висоти тону;

– *loop*: якщо ця властивість встановлена в значення "*true*", то аудіофайл буде відтворюватися постійно в циклі, створюючи постійну аудіофонотаску;

– *play On Awake*: якщо ця властивість встановлена в значення "*true*", то аудіофайл буде відтворюватися автоматично при запуску сцени;

– *spatial Blend*: ця властивість дозволяє визначити, як аудіо поширюється у просторі. Вона може бути налаштована від *2D* (ніде) до *3D* (повністю просторове);

– *doppler Level*: ця властивість контролює рівень *Doppler*-ефекту для аудіо. Вона впливає на зміну тону звуку при руху об'єкта;

– *component Spatial Blend*: ця властивість визначає, як звуковий ефект прив'язаний до об'єкта. Вона впливає на спосіб відтворення звуку на об'єкті;

– *max Distance* та *Min Distance*: ці властивості визначають максимальну та мінімальну відстані, на яких аудіо може бути почутим. Звук буде гучним на мінімальній відстані і ставатиме тише зі збільшенням відстані;

– *spatialize*: ця властивість вказує, чи спростовувати аудіо для кожного об'єкта в сцені.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		


```

isSodaNeeded = (UnityEngine.Random.value > 0.5f);
isSauceNeeded = (UnityEngine.Random.value > 0.5f);

if (isSodaNeeded) order += "Soda ";

if (!isSodaNeeded && !isSauceNeeded)
{
    countOfDishes = UnityEngine.Random.Range(1, 4);
}
else if(isSodaNeeded && !isSauceNeeded)
{
    countOfDishes = UnityEngine.Random.Range(1, 3);
}
for (int i = 0; i < countOfDishes; i++)
{
    var currentUnit = menu[UnityEngine.Random.Range(0,
menu.Length)];
    if (currentUnit == "Soda") isSodaNeeded = false;
    lines[0] += currentUnit + ",";
    order += currentUnit + " ";
}
lines[0] = lines[0].Substring(0, lines[0].Length - 1);
lines[1] = "Потрібна газована вода? ";

if(isSodaNeeded)
{
    lines[2] = "Так, будь ласка";
}
else {
    lines[2] = "Ні, дякую";
}
lines[3] = "Потрібен соус? ";

if (isSauceNeeded)
{
    lines[4] = "Так, будь ласка";
    lines[5] = "Вам який?";
    var currentUnit = sause[UnityEngine.Random.Range(0,
sause.Length)];
    lines[6] += currentUnit;
    order += currentUnit + " ";
}
else
{
    lines[4] = "Ні, дякую";
    Array.Resize(ref lines, 5);
}
var str = order.Trim();
textSplit = str.Split(" "[0]);
for (int i = 0; i < textSplit.Length; i++)
{
    if (!listOfOrder.Contains(textSplit[i]))
    {
        listOfOrder.Add(textSplit[i]);
    }
}
}

```

Метод *MakingOrder()* вирішує яке замовлення зробить клієнт і створює КОНТЕНТ для діалогу.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

Також метод *PlaceChanger()* знаходить вільне місце і направляю туди клієнта.

```

void PlaceChanger()
{
    places = FindObjectsOfType<Place>();
    for (int i = 0; i < places.Length; i++)
    {
        if (places[i].isRightEmpty)
        {
            GameObject[] array =
GameObject.FindGameObjectsWithTag(places[i].name);

            for (int y = 0; y < array.Length; y++)
            {
                if (array[y].name == "RightAnchor")
                {
                    listOfPlaces.Add(array[y].gameObject);
                }
            }
        }
        if(places[i].isLeftEmpty)
        {
            GameObject[] array =
GameObject.FindGameObjectsWithTag(places[i].name);

            for (int y = 0; y < array.Length; y++)
            {
                if (array[y].name == "LeftAnchor")
                {
                    listOfPlaces.Add(array[y].gameObject);
                }
            }
        }
        int index = UnityEngine.Random.Range(0, listOfPlaces.Count);
        currentPlacce = listOfPlaces[index];
        currentRotation = currentPlacce.transform.rotation;
        string str = currentPlacce.name + "Food";
        for (int q = 0; q < currentPlacce.transform.parent.childCount;
q++)
        {
            if (currentPlacce.transform.parent.GetChild(q).name == str)
            placeFood = currentPlacce.transform.parent.GetChild(q).gameObject;
            if (currentPlacce.transform.parent.GetChild(q).name ==
"Point")
            {
                placeAnchor =
currentPlacce.transform.parent.GetChild(q).gameObject;
                pos.transform.parent =
currentPlacce.transform.parent.GetChild(q).transform;
                pos.transform.position =
currentPlacce.transform.parent.GetChild(q).transform.position;
                agent.SetDestination(pos.transform.position);
            }
        }
    }
}

```

					<i>KPM.KI.1.884-03.1.1</i>	Арк. 57
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 3.14 – Префаби клієнтів

3.3.2 Префаби інгредієнтів та страв

Кожен префаб інгредієнту має типову структуру – компонент *RigidBody*, скрипт *Persistence Item* та визначений скрипт об'єкту.

На приклад префаб інгредієнту *Meat*, префаб має скрипт з таким самим ім'ям. Метод *Update* який контролює ступінь прожарювання

```
private void Update()
{
    complete = FindObjectOfType<CompleteFood>();
    if (timeRemaining > 0)
    {
        if(isFoodOnGrill && grill.isGrillToggle1 && !spoiled)
        {
            timeRemaining -= Time.deltaTime;

            if (timeRemaining < timeToGreatMake)
            {
                this.gameObject.GetComponent<MeshFilter>().mesh =
GreatPref;

                this.gameObject.name = nameOfCookedRes;
                isReady = true;
                Debug.Log("Ready");
            }
        }
    }
    else
    {
        this.gameObject.GetComponent<MeshFilter>().mesh =
spoiledPref;

        spoiled = true;
        Debug.Log("Spoiled");
    }
}
}
```

Кожен інгредієнт є інтерактивним об'єктом тому він наслідує методи з класу *Interactable*.

```
public abstract class Interactable : MonoBehaviour
{
    public enum InteractionType
    {
        Click,
        Hold
    }
    public InteractionType interactionType;
    float holdTime;
    public abstract string GetDescription();
}
```

```

public abstract void Interact();

public void IncreaseHoldTime()
{
    holdTime += Time.deltaTime;
}

public void ResetHoldTime() => holdTime = 0f;
public float GetHoldTime() => holdTime;
}

```

Також префаб *Meat* є об'єктом який взаємодіє з інструментом *Grill*. Префаб *Meat* має три стадії:

- НЕ ГОТОВО;
- ГОТОВО;
- зіпсовано.



Рис. 3.15 – Стадії префабу *Meat*

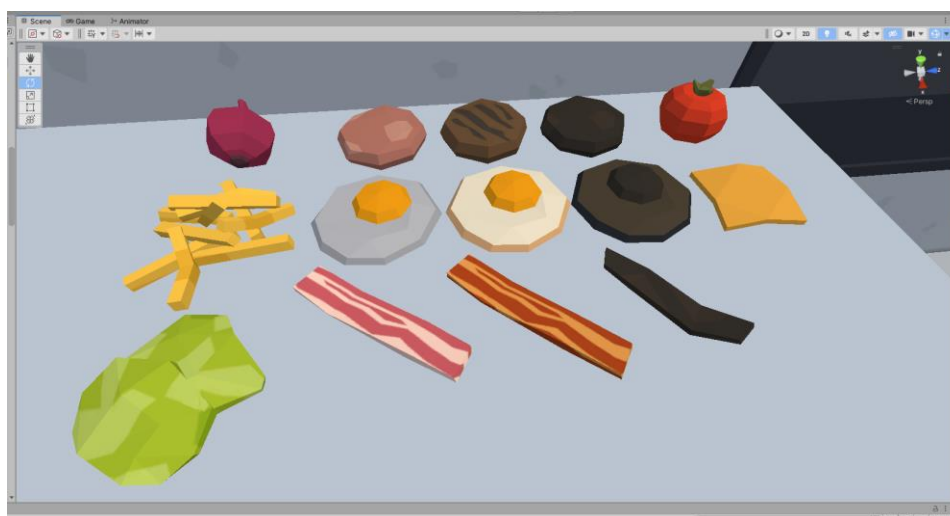


Рис. 3.16 – Префаби інгредієнтів

Для створення страви необхідно вибрати потрібну у об'єкті *FoodMaker*

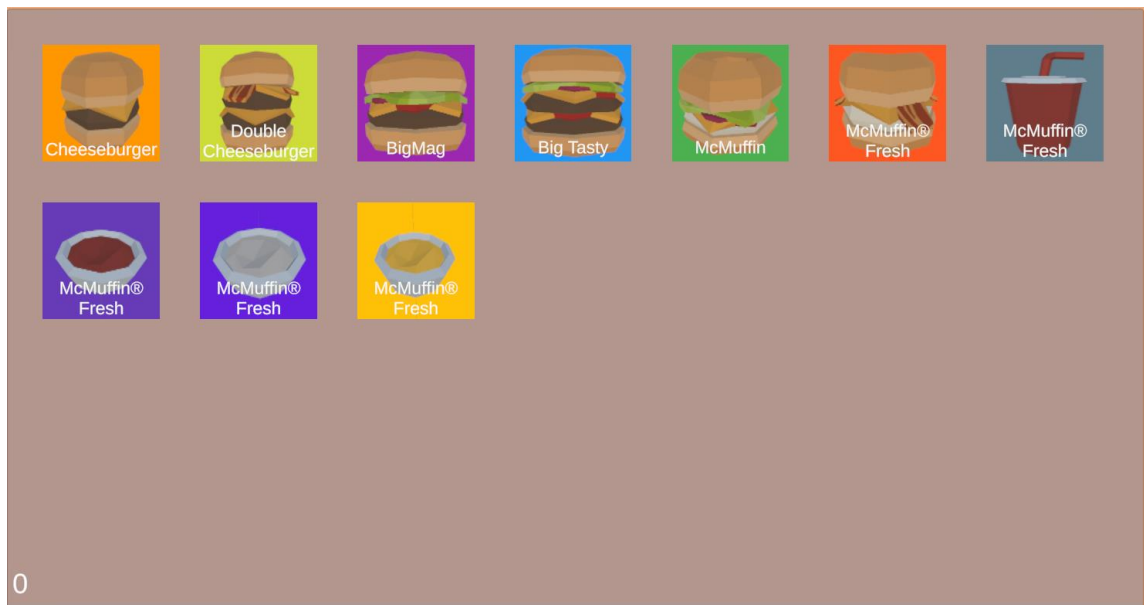


Рис. 3.17 – Модальне вікно FoodMaker

Після вибору потрібної страви треба зібрати всі потрібні інгредієнти.

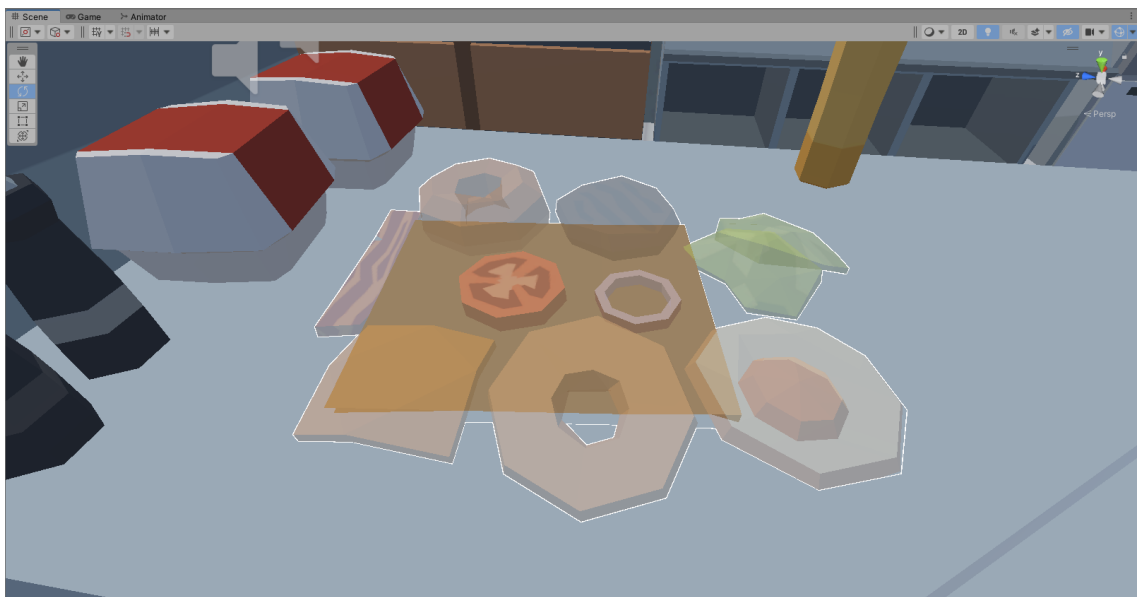


Рис. 3.18 – Створення страви

3.3.3 Префаби інструментів

У грі реалізовані такі інструменти:

- *grill*;
- *sodaGenerator*;

- *friesGenerator*;
- *terminal*;
- *casa*.

Grill інструмент для смаження певних інгредієнтів. Потрібно взяти інгредієнт, покласти його на решітку та включити сам об'єкт *Grill*. Після включення грилю буде чути типовий звук смаження. Кожен об'єкт під смаження має свій час для приготування. Реалізован скрипт для інструменту *Grill*.

```

private void OnCollisionStay(Collision collision)
{
    if (collision.gameObject.name != "Meat")
    {
        if (collision.gameObject.CompareTag("Grill"))
        {
            if (boss.stageOfProggres == 1) {

Methods.FindInActiveObjectByName("InteractableCanvasGrill").SetActive(false)
;

Methods.FindInActiveObjectByName("InteractableCanvasGrillToggle").SetActive(
true);
                }
                isFoodOnGrill = true;
                whatGrillUse
collision.gameObject.name.Substring(collision.gameObject.name.Length - 1);
            }
        }
    }

private void Update()
{
    complete = FindObjectOfType<CompleteFood>();
    if (timeRemaining > 0)
    {
        if(isFoodOnGrill && grill.isGrillToggle1 && !spoiled)
        {
            timeRemaining -= Time.deltaTime;

            if (timeRemaining < timeToGreatMake)
            {
                this.gameObject.GetComponent<MeshFilter>().mesh =
GreatPref;

                this.gameObject.name = nameOfCookedRes;
                isReady = true;
            }
        }
    }
    else
    {
        this.gameObject.GetComponent<MeshFilter>().mesh =
spoiledPref;

        spoiled = true;
        Debug.Log("Spoiled");
    }
}

```

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

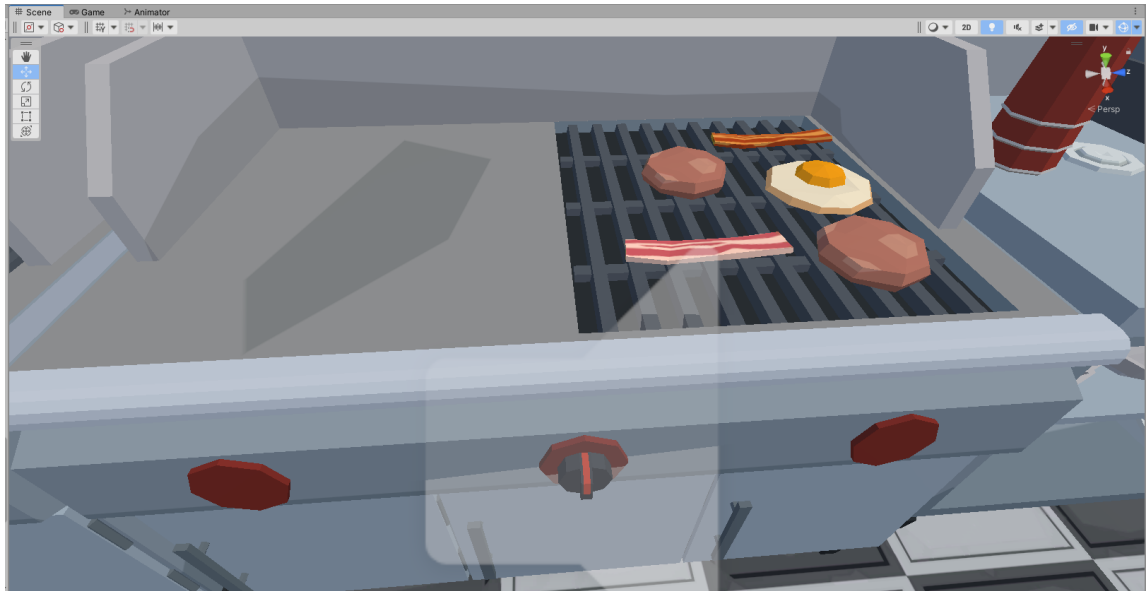


Рис. 3.19 – Інструмент *Grill*

SodaGenerator інструмент для приготування напоїв. Потрібно взяти порожній пластиковий стаканчик, підставити його під потрібний напой та натиснути на кнопку. *SodaGenerator* також має типовий звук. Після створення напою потрібно закрити стаканчик кришкою.

Скрипт *SodaGenerator*:

```
public class SodaGenerator : Interactable
{
    private void Update()
    {
        taken = GameObject.FindGameObjectWithTag("Taken");
        if(timerOn)
        {
            timeLeft -= Time.deltaTime;
            if (timeLeft < 0 && MyFunctionCalled == false)
            {
                FillTheCup();
                MyFunctionCalled = true;
            }
        }
    }
    IEnumerator isReadyCup()
    {
        yield return new WaitForSeconds(1f);
        isReady = true;
    }
    void FillTheCup()
    {
        timerOn = false;
        if (sodaAnchorFilled)
        {
            first.GetComponent<MeshRenderer>().enabled = false;
            Destroy(first);
            var newWish = Instantiate(cupprefab, new Vector3(0.2551f,
0.0977f, 0.395f), Quaternion.Euler(0f, 0f, 0f));
```



```

    }
    else if (this.gameObject.name ==
"SM_Prop_Kitchen_Deep_Fryer_Net_02")
    {
        var newItem = Instantiate(prefab2,
anchor2.transform.position, anchor2.transform.rotation);
        newItem.GetComponent<GUID>().GenerateGUID();
        newItem.GetComponent<UniqueId>().ID =
newItem.GetComponent<GUID>().guid;
        newItem.name = prefab2.name;

        if (boss.stageOfProgres == 3)
        {

Methods.FindInActiveObjectByName ("InteractableCanvasReadyFries") .SetActive(t
rue);

        }
    }
    else
    {
        var newItem = Instantiate(prefab3,
anchor3.transform.position, anchor3.transform.rotation);
        newItem.GetComponent<GUID>().GenerateGUID();
        newItem.GetComponent<UniqueId>().ID =
newItem.GetComponent<GUID>().guid;
        newItem.name = prefab3.name;
    }

    isReady = true;
}

```

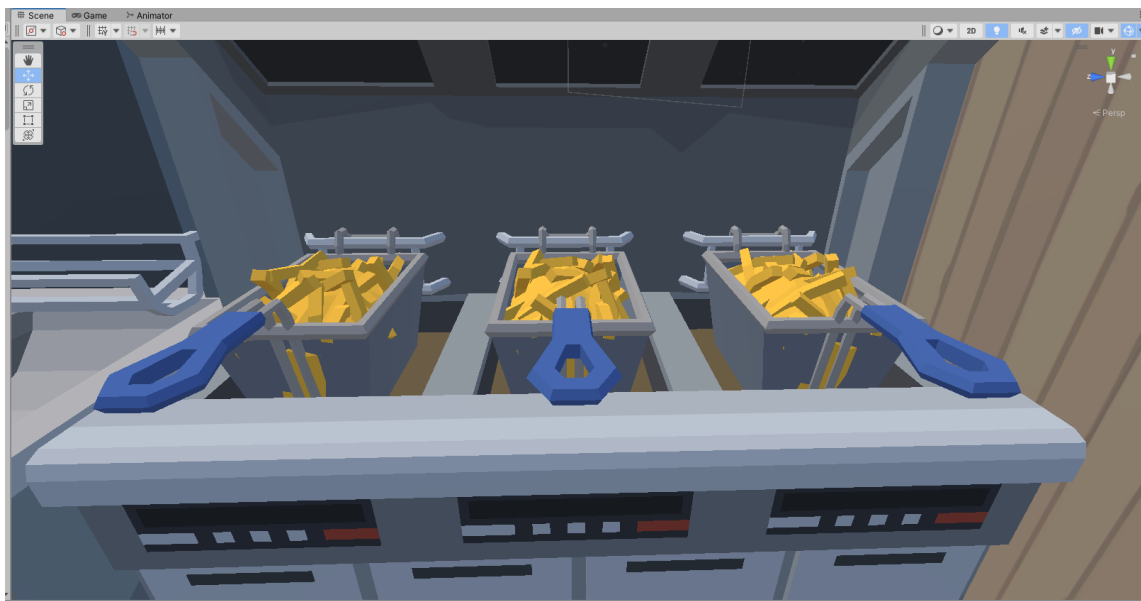


Рис. 3.21 – Інструмент *FriesGenerator*

Terminal інструмент для замовлення інгредієнтів, кожен інгредієнт має свою ціну.

Скрипт *Terminal*

```

public class Terminal : Interactable
{
    public void addToOrder (GameObject prefabOfDish)

```

						KPM.KI.1.884-03.1.1	Арк. 64
Змн.	Арк.	№ докум.	Підпис	Дата			

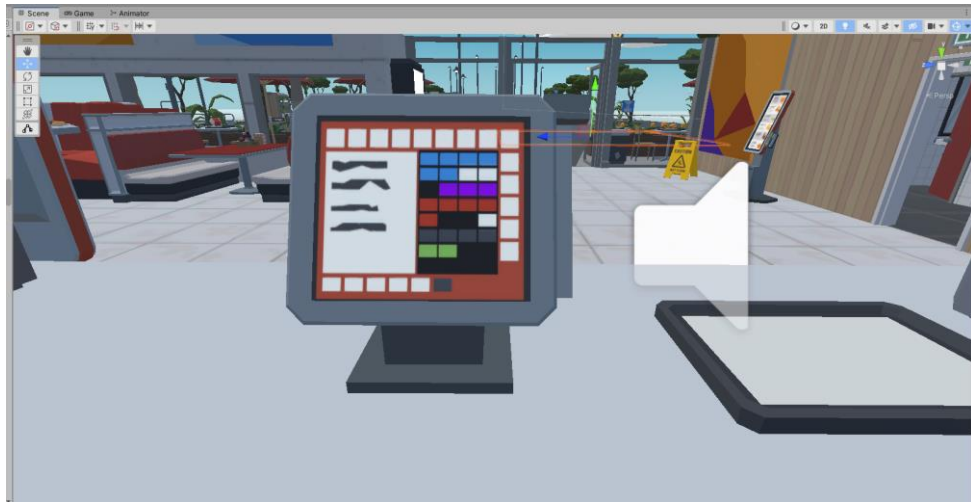


Рис. 3.24 – Модальне вікно інструменту *Casa*



Рис. 3.25 – Модальне вікно інструменту *Terminal*



Рис. 3.26 – Підготовлений заказ

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

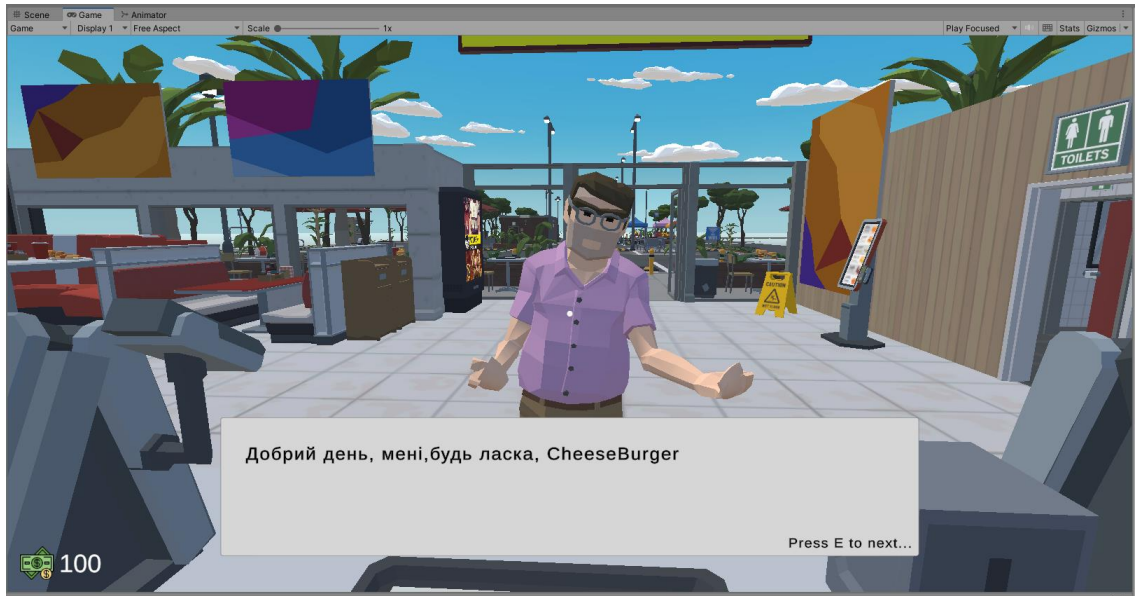


Рис. 3.29 – Взаємодія з клієнтом

3.5 Створення штучного інтелекту співробітника

У грі реалізовано функціонал найняття співробітника для поліпшення геймплею. Найняття співробітника потребує певну плату, але він допоможе швидше виконувати замовлення.



Рис. 3.30 – Зовнішній вигляд співробітника

Після найняття треба вибрати страву з модального вікна.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

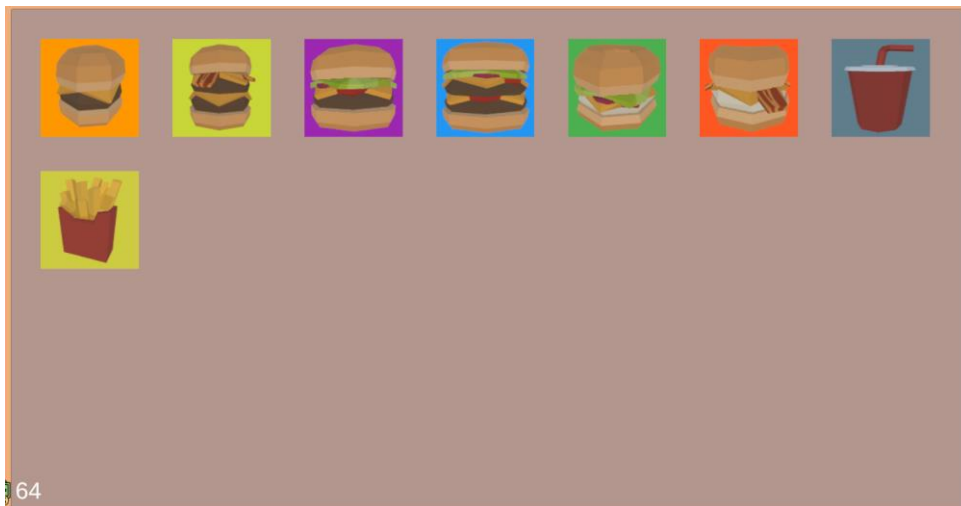


Рис. 3.31 – Модальне вікно співробітника

Після вибору страви ШІ аналізує склад страви та заказує потрібні інгредієнти, після отримання інгредієнтів ШІ бере інгредієнти для смаження та йде до грилю, чекає поки всі інгредієнти будуть готові, наступним кроком ШІ аналізую інгредієнти що залишилися і якщо є інгредієнти які потрібно нарізати бере їх та направляється до нарізної дошки, після підготовки всіх інгредієнтів ШІ збирає страву.

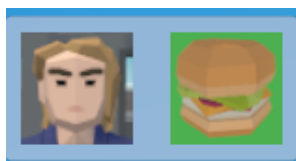


Рис. 3.32 – Індикатор поточної страви

Переміщення реалізовано за допомогою компоненту *NavMeshAgent*

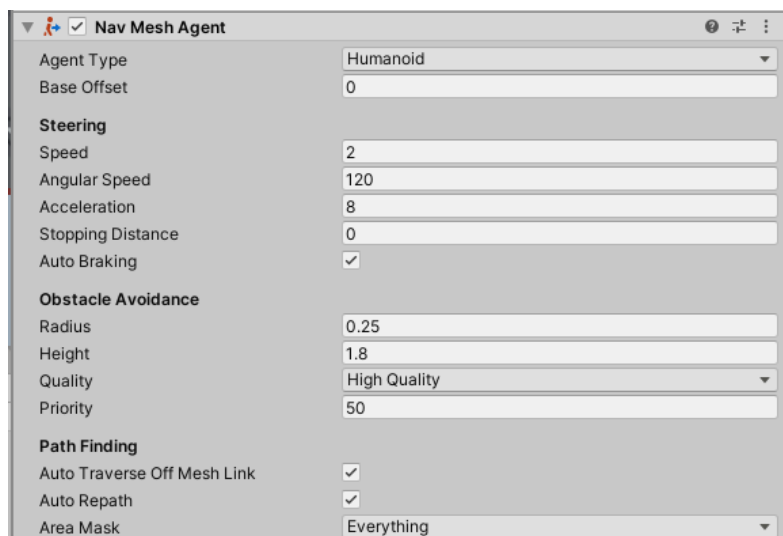


Рис. 3.33 – Компонент *NavMeshAgent*


```

pos.transform.parent = sodaMaker.transform;
anim.SetBool("IsWalking", true);
agent.SetDestination(pos.transform.position);
workerMenu.SetActive(true);
int children = workerMenu.transform.childCount;

    }
}
}

```

3.6 Створення штучного інтелекту клієнта

При створенні клієнту була поставлена ціль зробити його дії якомога реалістичніше. Життєвий цикл клієнту починається з появи його *GameObject* на сцені. Одразу він направляється до каси, ШІ аналізує стан каси вільна або є черга, якщо є черга - займає місце в черзі.



Рис. 3.35 – Клієнт займає місце у черзі

Наступною дією клієнт робить замовлення. ШІ аналізує меню та робить замовлення. Після створення замовлення клієнт очікує його готовність, час очікування залежить від складності заказу.



Рис. 3.36 – Поточний заказ та час очікування

Якщо заказ не було виконано за час очікування клієнт уходить з ресторану.


```

        GameObject[] array =
GameObject.FindGameObjectsWithTag(places[i].name);

        for (int y = 0; y < array.Length; y++)
        {
            if (array[y].name == "RightAnchor")
            {
                listOfPlaces.Add(array[y].gameObject);
            }
        }
        if (places[i].isLeftEmpty)
        {
            GameObject[] array =
GameObject.FindGameObjectsWithTag(places[i].name);

            for (int y = 0; y < array.Length; y++)
            {
                if (array[y].name == "LeftAnchor")
                {
                    listOfPlaces.Add(array[y].gameObject);
                }
            }
        }
        int index = UnityEngine.Random.Range(0, listOfPlaces.Count);
        currentPlacce = listOfPlaces[index];
        currentRotation = currentPlacce.transform.rotation;
        string str = currentPlacce.name + "Food";

        for (int q = 0; q < currentPlacce.transform.parent.childCount;
q++)
        {
            if (currentPlacce.transform.parent.GetChild(q).name == str)
placeFood = currentPlacce.transform.parent.GetChild(q).gameObject;
            if (currentPlacce.transform.parent.GetChild(q).name ==
"Point")
            {
                placeAnchor =
currentPlacce.transform.parent.GetChild(q).gameObject;
                pos =
currentPlacce.transform.parent.GetChild(q).transform.position;
                agent.SetDestination(pos);
            }
        }
    }
}

```

3.7 Багатокритерійна оптимізація геймплею

Для проведення багатокритерійної оптимізації за допомогою методу збалансованих оцінок, спочатку необхідно призначити ваги кожному критерію, щоб визначити їхню важливість в контексті оцінки гри. Ваги можна призначити експертно або провести опитування серед користувачів, щоб з'ясувати їх уподобання.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		

Вагові коефіцієнти визначають важливість кожного критерію в багатокритерійній оптимізації. Ось розподіл важливості для кожного критерію відповідно до їх вагових коефіцієнтів:

1. Реалістичність приготування страв: 0.15 (15%).
2. Креативність та варіативність рецептів: 0.10 (10%).
3. Розвиток кулінарних навичок: 0.15 (15%).
4. Менеджмент часу та ресурсів: 0.15 (15%).
5. Динаміка гри: 0.15 (15%).
6. Графіка та анімації: 0.12 (12%).
7. Звукове супроводження: 0.08 (8%).
8. Складність проходження: 0.10 (10%).

Далі, для кожного аспекту гри (наприклад, окремого рецепту), ми можемо визначити оцінки від 1 до 10 за кожним критерієм.

За допомогою призначених ваг, ми можемо обчислити збалансовану оцінку для кожного аспекту:

$$\text{Збалансована оцінка} = \sum_{i=1}^8 (\text{Оцінка за критерієм} \times \text{Вага критерію}) \quad (3.1)$$

$$\begin{aligned} \text{Збалансована оцінка} &= (8 \times 0.15) + (5 \times 0.10) + (5 \times 0.15) + (7 \times 0.15) + \\ &+ (6 \times 0.15) + (6 \times 0.12) + (6 \times 0.08) + (6 \times 0.10) = 5.1 \end{aligned}$$

Отже, збалансована оцінка багатокритерійної оптимізації для наданих вагових коефіцієнтів та середніх оцінок становить 5.1.

Таким чином, для кожного аспекту гри отримуємо збалансовану оцінку, яка враховує важливість різних критеріїв. Потім можна ранжувати аспекти гри за збалансованими оцінками для визначення їхньої відповідності вимогам користувачів.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до третього розділу

1. У ході проведеної роботи було розроблено програмну та візуальну структуру для реалізації демонстраційної версії гри.
2. Розроблено моделі навколишнього середовища, бонусів та інших елементів гри. Усі об'єкти було розташовані на сцені належним чином.
3. Проєкт був зкомпільований та у ході його тестування не було виявлено ніяких проблем та помилок.

					КРМ.КІ.1.884-03.1.1	Арк.
						78
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 4 ЕКОНОМІЧНІ РОЗРАХУНКИ

4.1 Організаційно-економічне й маркетингове обґрунтування проекту

4.1.1 Порівняльний техніко-економічний аналіз пропонованого проекту й обраного аналогу

Темою організаційно-економічної частини кваліфікаційної роботи є техніко-економічне обґрунтування (ТЕО) на тему: Проектування і розробка аркадної гри жанру «*Cooking simulator*». В кваліфікаційній роботі проводиться проектування й розробка прототипу комп'ютерної гри, яку можна продавати через інтернет магазини, самостійно або за допомогою видавництва.

В ТЕО необхідно розглянути наступні питання:

- планування розробки з побудовою мережевого графіку;
- розрахунок вартості розробки;
- економічну ефективність розробки;
- склад і призначення основних розділів господарського договору.

В організаційній частині ТЕО необхідно розрахувати термін розробки і побудувати мережевий графік робіт по створенню програмного забезпечення. В розрахунковій частині ТЕО необхідно розрахувати вартість розробки програмного забезпечення з моменту отримання першого варіанту технічного завдання і закінчуючи оформленням документації і здачі розробки. На завершення необхідно обґрунтувати економічну ефективність розробки.

Основною прикладною метою даної роботи є створення аркадної комп'ютерної гри жанру «*Cooking simulator*».

					КРМ.КІ.1.884-03.1.1	Арк. 79
Змн.	Арк.	№ докум.	Підпис	Дата		

4.1.2 Організаційне обґрунтування

1. Класифікаційна оцінка різновиду проекту. Цей проект є монопроектом змішаного типу. Вид проекту – учбовоосвітній, тривалість – короткостроковий. Проект є малим, простим за складністю. Рівень проекту – галузевий.

2. Визначення мети і результатів проекту. Метою проекту є розробка демонстраційної версії аркадної гри жанру «*Cooking simulator*». Результатом має бути розроблений програмний продукт і документація

Етапи виконання робіт проекту з орієнтованими термінами (табл. 4.1):

- збір інформації по предметній області (2 дні);
- постановка задачі (4 днів);
- детальне дослідження предметної області (7 днів);
- планування структури проекту (8 днів);
- розробка програмної документації (13 днів);
- розробка пояснювальної записки (6 днів);
- розробка геймдизайнерського документу (11 днів);
- повна реалізація програмного коду (20 днів);
- повна реалізація дизайну гри (12 днів);
- проведення системи випробувань (3 днів);
- доробка програмного коду (8 днів);
- доробка дизайну гри (8 днів);
- доробка програмної документації (6 днів);
- доробка пояснювальної записки (6 днів);
- здача проекту (6 днів).

3. Розробка проекту по життєвому циклу.

Перша фаза: збір інформації, аналіз й порівняння з існуючими аналогами. Затвердження основної компетенції.

Друга фаза: вибір програмних засобів, затвердження плану розробки.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

Третя фаза: детальне планування і проектування робіт. Контроль проведення реалізації проекту, вирішення проблем і задач, що виникли у стадії розробки. Закінчення робіт.

Четверта фаза: випробування остаточного продукту, завершення усіх робіт, підготовка до закриття проекту.

4. Навколишнє середовище та учасники проекту.

На виконання проекту впливає низка різноманітних зовнішніх чинників, які необхідно врахувати при розробці.

Учасниками проекту є:

- керівник проекту;
- розробник.

5. Календарний план робіт проекту і його мережева модель з розрахунком предметів і оптимізацією побудови моделі.

Таблиця 4.1

Склад робіт проекту і їх тривалість

Код роботи	Найменування робіт	Тривалість, дні
0-1	Збір даних та аналіз стану	8
1-2	Дослідження предметної області	6
2-3	Затвердження концепції проекту	3
3-4	Планування необхідних заходів забезпечення виконання працювати	6
4-5	Організація виконання робіт	1
2-6	Дизайн концепції проекту	10
6-7	Розробка програмної документації	10
5-8	Розробка пояснювальної записки	9
7-9	Повне впровадження програмного коду	13
9-10	Повна реалізація дизайну	13
10-11	Виконайте тестування системи	2

					<i>KPM.KI.1.884-03.1.1</i>	Арк. 81
Змн.	Арк.	№ докум.	Підпис	Дата		

Код роботи	Найменування робіт	Тривалість, дні
11-12	Доробка програмного коду	8
11-13	Доробка дизайну	8
12-14	Доробка програмної документації	4
8-12	Доробка пояснювальної записки	4
13-14	Проект зданий у виробництво	7
14-15	Оцінка та підсумки результатів проекту результат	2

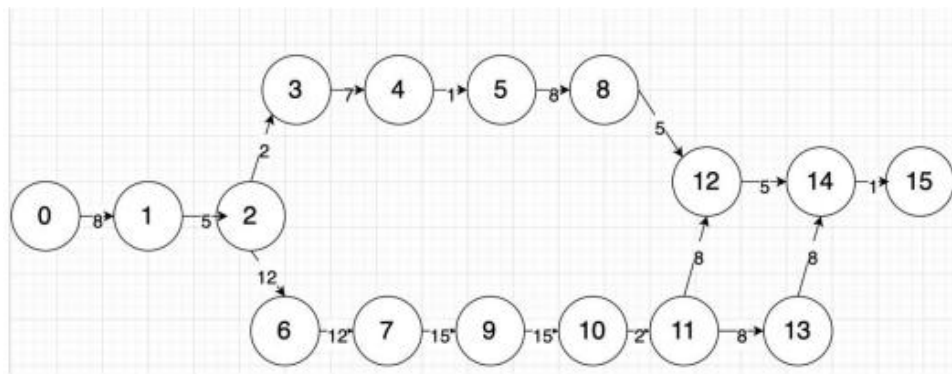


Рис 4.1 – Календарний план робіт

Таблиця 4.2

Розрахунок мережевого графіку

Подія i/j		Ранній термін здійснення події	Тривалість роботи	Ранній термін здійснення події j	Пізній період здійснення події i	Тривалість роботи	Пізній період здійснення події j	Резерв
i	j	$Tr[i]$	$T[i,j]$	$Tr[j]$	$Trp[i]$	$T[i,j]$	$Trp[i,j]$	$R[j]$
0	1	0	8	8	0	8	8	0
1	2	8	5	13	8	5	13	0
2	3	13	2	15	13	2	15	0

Подія i/j		Ранній термін здійсн ення події	Три ва ліст ь робо ти	Ранній термін здійснен ня події j	Пізній період здійсненн я події i	Трива лість робот и	Пізній період здійсненн я події j	Резе рв
2	6	13	12	25	13	12	25	0
3	4	15	7	22	15	7	22	0
4	5	22	1	23	22	1	23	0
5	8	23	8	31	23	8	31	0
6	7	25	12	37	25	12	37	0
7	9	37	15	52	37	15	52	0
8	14	31	5	36	80	5	85	49
9	10	52	15	67	52	15	67	0
10	11	67	2	69	67	2	69	0
11	12	69	8	77	69	8	77	0
11	13	69	8	77	69	8	77	0
12	14	77	5	82	80	5	82	0
13	14	77	8	85	77	8	85	0
14	15	85	1	86	85	1	86	0

В результаті розрахунку параметрів виявлено, що без оптимізації тривалість проекту складає 86 днів. Оптимізувати проект не має сенсу, так як його тривалість є невеликою.

4.1.3 Маркетингові обґрунтування проекту

Можливі сфери використання цього проекту: індустрія розваг, інформаційні технології.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						83
Змн.	Арк.	№ докум.	Підпис	Дата		

Потенційне коло споживачів: широка аудиторія, така як ентузіасти ігор різного віку, видавничі компанії чи розробники комп'ютерних ігор.

Конкурентні переваги проекту: цікавий ігровий всесвіт, популярний ігровий жанр, нестандартні ігрові елементи жанру.

4.2 Економічні розрахунки

4.2.1 Визначення трудомісткості розробки ПП

Розроблювальному ПП відповідає аналог – ПЗ організації обчислювального процесу з $V0 = 5000$ умовних машинних команд із трудомісткістю $TP H = 299$ люд/г.

Трудомісткість розроблювального ПП визначається на кожному етапі окремо на підставі трудомісткості аналога з урахуванням складності розробки, ступеня новизни й ступеня використання в розробці стандартних модулів на підставі формул 4.1 - 4.4.

$$TT3 = TP * L1 * KH. \quad (4.1)$$

$$TTP = TP * L2 * KH. \quad (4.2)$$

$$TRP = TP * L3 * KH * KT. \quad (4.3)$$

$$TBH = TP * L4 * KH. \quad (4.4)$$

Tp – укрупнена норма часу на розробку аналога ПС, люд/год, що коректується поправочним коефіцієнтом, що враховує умови розробки ПС, тобто в умовах комп'ютера, $Kk = (0,7 + 0,8)$, тобто

$$TP = 299 * 0,8 = 239 \text{ люд/год.}$$

Даний проект можна віднести до ступеня новизни: Б

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						84
Змн.	Арк.	№ докум.	Підпис	Дата		

L_j – питома вага j -го етапу розробки (залежно від ступеня новизни й відповідних стадій):

$$L1 = 0,12;$$

$$L2 = 0,15;$$

$$L3 = 0,61;$$

$$L4 = 0,15.$$

KN – поправочний коефіцієнт, що враховує ступінь новизни, у цьому випадку $KN = 0,7$; KT – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм $KT = 0,6$.

Тоді:

$$TT3 = 239 * 0,12 * 0,7 = 20(\text{дні}).$$

$$ТПП = 239 * 0,15 * 0,7 = 25(\text{дні}).$$

$$ТРП = 239 * 0,61 * 0,7 * 0,6 = 61 (\text{дні}).$$

$$ТВН = 239 * 0,15 * 0,7 = 25 (\text{дні}).$$

Тривалість розробки ПП у роках визначається за формулою 4.5

$$ТПП = TT3 + ТПП + ТРП + ТВН. \quad (4.5)$$

де $ТПП$ – сумарна тривалість розробки.

$$ТПП = 20 + 25 + 61 + 25 = 131(\text{дні}) = 0,358(\text{р}).$$

4.2.2 Визначення ціни ПП

Оскільки ПП розглядається й створюється як продукція виробничотехнічного призначення, що допускає багаторазове тиражування й відчуження від безпосередніх розроблювачів, значить:

$$Ц = C * K + Пр. \quad (4.6)$$

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						85
Змн.	Арк.	№ докум.	Підпис	Дата		

де C – витрати на розробку програмної продукції (кошторисна собівартість);

K – коефіцієнт обліку витрат на виготовлення досвідченого зразка ПП як продукції виробничо-технічного призначення.

PP – нормативний прибуток, що розраховується по формулі:

$$PP = (C - CM) * \frac{P_H}{100} \quad (4.7)$$

де P_H – норматив рентабельності, 25%;

CM – матеріальні витрати, грн./вироб.

Витрати на розробку програмної продукції можуть бути представлені у вигляді кошторису витрат, що включає в себе наступні статті:

1. Матеріали. Витрати на матеріали визначаються по формулі 4.8:

$$CM = KMP * \sum Ci * Vi. \quad (4.8)$$

де KMP – коефіцієнт транспортно-заготівельних видатків;

Ci – ціна одиниці i -го матеріалу, грн.;

Vi – придбана кількість i -го матеріалу.

В таблиці 4.3 представлено витрати на матеріали.

Таблиця 4.3

Витрати на матеріали

Найменування товару	Опис матеріалу	Кількість	Ціна за одиницю, грн.	Сума, грн.
Папір	Папір офісний	1	100	100
Флеш накопичувач	Флеш пам'ять USB Goodram UUN2 Unity 32GB	1	160	160

Найменування товару	Опис матеріалу	Кількість	Ціна за одиницю, грн.	Сума, грн.
Тонер для принтеру	WWM для Pantum M6500/M6607	1	60	60
Усього				320
КМР = 0,1				32.0
Разом:				320

$CM = 320$ грн.

2. Спеціальні устаткування.

Витрати, які пов'язані з використанням обчислювальної техніки, визначаються по формулі:

$$C_{EOM} = t^{EOM} * K_I^{EOM} * C^{EOM} * K_E^{EOM} * K_{БД}^{EOM}. \quad (4.9)$$

де t^{EOM} – час використання ЕОМ для розробки даного ПО, год (185);

K_I^{EOM} – поправочний коефіцієнт обліку часу використання ЕОМ (1,08);

C^{EOM} – ціна 1-ої години роботи ЕОМ, грн. (5);

$K_E^{EOM} = 1,0$;

$K_{БД}^{EOM} = 1,0$ (не використовується).

Тоді: $C_{EOM} = 320 * 1,08 * 5 = 1728$ грн.

3. Основна заробітна плата

У статтю включається основна заробітна плата двох виконавців, безпосередньо зайнятих розробкою даного ПП (керівник, нормо контроль), з обліком їхнього посадового окладу (8500 та 14000 грн. відповідно) і часу участі в розробці. Розрахунок ведеться по формулі 4.10:

$$P_{30} = \sum \frac{z_i * K_0 * \tau_i}{100} \quad (4.10)$$

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						87
Змн.	Арк.	№ докум.	Підпис	Дата		

де KH – коефіцієнт накладних видатків (50%).

Тоді: $C_H = 0,5 * 35199 = 17599$ грн.

Результати розрахунку кошторисної вартості ПП можна звести в табл. 4.4

Таблиця 4.4.

Результати розрахунку кошторисної вартості

Найменування статті	Кошторисна собівартість, грн.	Питома вага, %	Примітка
Матеріали	320	1	–
Найменування статті	Кошторисна собівартість, грн.	Питома вага, %	Примітка
Спеціальне устаткування	1728	3	–
Основна заробітна плата	35199	52	–
Додаткова заробітна плата	3519	5	–
Відрахування на соціальне страхування	8617	13	–
Накладні витрати	17599	26	–
Разом	66982	100	–

Таким чином:

$$C = C_M + C_{EOM} + C_{ZO} + C_{ЗД} + C_{CC} + C_H = 66982 \text{ грн.}$$

$$P_P = (66982 - 320) * 0,25 = 16665 \text{ грн.}$$

$$Ц = 1,1 * 66982 + 16665 = 90\ 345 \text{ грн.}$$

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						89
Змн.	Арк.	№ докум.	Підпис	Дата		

4.2.3 Визначення показника економічної ефективності

Очікуваний економічний ефект визначається за формулою:

$$E_0 = E_T - E_H * K_{П}. \quad (4.14)$$

де E_T – річна економія на поточних витратах (грн.);

$K_{П}$ – одноразові витрати на проект (грн.). У цьому випадку: вартість комп'ютера – 20 000 грн та ПП – 90 345 грн;

E_H – нормативний коефіцієнт ефективності одноразових витрат (рекомендований $E_H = 0,25$; може бути також заданий господарюючим суб'єктом, або приймається на рівні процентної ставки по депозитних рахунках банку).

Річна економія складається з поточних витрат до і після впровадження ПП, у такий спосіб:

$$E_T = (C_1 - C_2) + \Delta П. \quad (4.15)$$

де C_1 , C_2 – відповідно поточні витрати, відповідно до й після впровадження проекту (грн.);

$\Delta П$ – приріст прибутку господарюючого суб'єкта або його структурного підрозділу при впровадженні проекту (грн.) визначається експертним шляхом. В цьому випадку вона складе 0 грн.

$$C_1 = (C_{30} + C_{зд} + C_{СС}) * K_{Р}. \quad (4.16)$$

Під час проведення аналізу предметної області за базовий варіант було обрано діяльність компанії «Сарсом» без використання спеціалізованої програми. Весь процес контролю виконують три співробітника, заробітна плата кожного складає 10000 грн. Розрахунок річного фонду основної і додаткової оплати праці персоналу з нарахуванням.

$$C_{30} = 10000 * 12 = 120\,000 \text{ грн.}$$

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						90
Змн.	Арк.	№ докум.	Підпис	Дата		

Додаткова заробітна плата становить:

$$C_{зд} = 120\,000 * 0,1 = 12\,000 \text{ грн.}$$

Відрахування на соціальне страхування складає:

$$C_{СС} = 0,22 * (120\,000 + 12\,000) = 29\,040 \text{ грн.}$$

Таким чином, поточні витрати до розробки гри ПП, становлять:

$$C_1 = (120\,000 + 12\,000 + 29\,040) * 3 = 483\,120 \text{ грн.}$$

Одноразові витрати на проект складають:

$$K_{П} = 20\,000 + 90\,345 = 110\,345 \text{ грн.}$$

$$C_2 = C_{ЕОМ} + C_A + (C_{зо} + C_{зд} + C_{СС}) * K_P + C_P + C_{ВСП}. \quad (4.17)$$

Витрати, пов'язані з використанням обчислювальної техніки, становлять:

$$C_{ЕОМ} = t^{ЕОМ} * K_{И}^{ЕОМ} * Ц^{ЕОМ}. \quad (4.18)$$

Де $t^{ЕОМ}$ – річний фонд часу роботи ЕОМ, який визначається виходячи з кількості робочих днів в році, тривалості робочого дня і з урахуванням часу на профілактичні огляди за рік:

$$t^{ЕОМ} = 8 * 365 = 2\,920 \text{ (год).}$$

$K_{И}^{ЕОМ}$ – поправочний коефіцієнт обліку часу використання ЕОМ (1,08);

$Ц^{ЕОМ}$ – ціна за 1 кВт світла, грн. (3,25);

$$C_{ЕОМ} = 2\,920 * 1,08 * 3,25 = 10\,249 \text{ грн.}$$

C_A – сума річних амортизаційних відрахувань від вартості основного й допоміжного устаткування ІС (КМ) (25% від вартості устаткування);

$$C_A = 0,25 * 15\,000 = 3\,750 \text{ грн.}$$

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						91
Змн.	Арк.	№ докум.	Підпис	Дата		

C_p – вартість річного ремонту основного й допоміжного устаткування (6% $K_{ко}$);

$C_{всп}$ – річна вартість допоміжних матеріалів, пов'язаних з експлуатацією ІС (КМ) (2% $K_{ко}$);

Таким чином, поточні витрати після впровадження ПП, становлять:

$$C_2 = 5298 + 3\,750 + (120\,000 + 12\,000 + 29\,040) + 20\,000 * (0,06 + 0,02) \\ = 171\,688 \text{ грн.}$$

$$E_r = (483\,120 - 114\,923) + 0 = 368\,197 \text{ грн.}$$

$$E_o = 199\,077 - 0,25 * 110\,345 = 171\,490 \text{ грн.}$$

Потім розраховується коефіцієнт ефективності одноразових витрат за формулою:

$$E = \frac{E_r}{K_n} \quad (4.19)$$

Якщо $E > E_n$, то проект ефективний.

$$E = \frac{368197}{20000+90345} = 3.33.$$

Розраховується строк окупності одноразових витрат проекту, років:

$$T = \frac{1}{E}, \quad (4.20)$$

$$T = \frac{1}{3.33} = 0.3.$$

Основні економічні показники проекту надані в таблиці 4.5.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						92
Змн.	Арк.	№ докум.	Підпис	Дата		

Економічні показники проекту

№	Показники	Числове значення	Одиниці виміру
1	Тривалість розробки	144	Дні
2	Ціна ПП	90 345	Грн
3	Капітальні затрати	110 345	Грн
4	Річна економія на поточних витратах після введення ПП	368 197	Грн
5	Економічний ефект	171 490	Грн
6	Коефіцієнт ефективності	3.33	–
7	Термін окупності проекту	0.3	Рік

4.3 Бізнес план стартапу проекту

Міні-бізнес план стартап-проекту для гри жанру «Cooking Simulator»

Міні-бізнес план стартап-проекту

Основна ціль проекту	Основною ціллю проекту стартапу розробки гри жанру "Cooking Simulator" є досягнення успіху в гейміндустрії шляхом створення високоякісної, цікавої та популярної гри, яка буде привертати гравців та приносити прибуток.
Аналіз ринку	Група цільової аудиторії включає геймерів віком від 14 років та цінителів кулінарії. Зростаючий інтерес до кулінарних симуляторів у гейміндустрії. Можливість монетизувати гру через продаж гри, мікротранзакції для отримання внутрішньої валюти, продаж додаткового контенту або рекламу.
Розробка продукту (геймплей, графіка)	Реалістичність готування: Важливо створити реалістичний процес готування, де гравці можуть відчувати себе справжніми кухарями. Додавання реалістичних фізичних властивостей до продуктів, взаємодія з кухонними приладами та технічні аспекти готування.

										Арк.
										93
Змн.	Арк.	№ докум.	Підпис	Дата	КРМ.КІ.1.884-03.1.1					

Маркетинг та реклама	<p>Розроблення веб-сайту для гри, де гравці можуть знайти інформацію про гру, оновлення та можливості спілкування з іншими гравцями. Також створення сторінки у популярних соціальних медіа для активного спілкування з аудиторією.</p> <p>Розроблення трейлеру та геймплей-відео для гри та розмістити їх на платформах, таких як YouTube і Twitch, для привертання уваги геймерів.</p>
Фінансові доходи	<p>Основним джерелом доходів для більшості ігрових проєктів є продаж самої гри. Гравці можуть купувати гру через ігрові платформи, такі як Steam, PlayStation Store, Xbox Live тощо.</p> <p>Можливість додання внутрішню валюту або предмети, які гравці можуть купувати за допомогою мікротранзакцій. Наприклад, це може бути купівля нових рецептів, приладів або декору для кухні.</p>
Висновки	<p>Проєкт має високий потенціал завдяки популярності симуляторів і зростаючому інтересу до гейміндустрії. Кулінарний симулятор може привернути широку аудиторію, особливо серед геймерів, які цікавляться кулінарією.</p>

Висновки до четвертого розділу

1. Розробка даного проєкту є вигідною, бо має швидку окупність і перспективи додаткового прибутку у майбутньому. Розробка ігор є прибутковим бізнесом, попит на який з кожним роком усе зростає.
2. Даний проєкт має великі перспективи за рахунок популярного жанру, знайомих широкому користувачу механік й цікавому сеттингу. Також проєкт може бути легко адаптований для усіх платформ й має необмежену перспективу росту.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						94
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 5 ОХОРОНА ПРАЦІ

Охорона праці – це система законодавчих актів, соціально-економічних, організаційних, технічних, гігієнічних і лікувально-профілактичних заходів і засобів, що забезпечують безпеку, збереження здоров'я й працездатності людини в процесі праці.

5.1 Класифікація виробництв за ступенем вибухової, вибухопожежної і пожежної небезпеки.

Усі підприємства поділяють за вибуховою, вибухо-пожежною і пожежною небезпекою на категорії: А, Б, В, Г, Д. Категорія виробництва по пожежній небезпеці визначає вимоги до конструкцій і плануванню будинку, організації пожежної охорони і її технічної оснащеності, до режиму й експлуатації. Розглянуте в даній роботі приміщення належить до категорії Д – це виробництва, у яких застосовуються негорючі речовини й матеріали в холодному стані.

5.2 Рішення з розміщенням проектного обладнання

Розміри заданого приміщення наступні: довжина – $A = 5$ м; ширина – $B = 6$ м; висота – $H = 3$ м.

Площа приміщень, у яких розташовують відео-термінали, визначають відповідно діючим нормативним документам з розрахунку на одне робоче місце: площа – не менше 6,0 кв. м, обсяг – не менше 20,0 куб. м, з урахуванням максимальної кількості осіб, що одночасно працюють в приміщенні. У нашому випадку площа – 30 кв.м, а об'єм – 90 куб.м. Отже найбільша кількість людей, що одночасно працюють у приміщенні, дорівнює 4 людини.

					<i>KPM.KI.1.884-03.1.1</i>	Арк. 95
Змн.	Арк.	№ докум.	Підпис	Дата		

5.3 Електробезпека

Електробезпека – це система організаційних, технічних заходів і засобів, що забезпечують захист людей від шкідливого і небезпечного впливу електричного струму, електричної дуги, електромагнітного поля і статичної електрики. Електробезпека на підприємстві забезпечується завдяки дотриманню вимог, викладених у таких актах законодавства:

Правила безпечної експлуатації електроустановок споживачів (наказ Держнаглядохоронпраці від 09.01.1998 № 4); Правила безпечної експлуатації електроустановок (наказ Держнаглядохоронпраці України від 06.10.1997 № 257), дія яких поширюються на працівників, що виконують роботи в електроустановках Міністерства енергетики України;

Правила технічної експлуатації електроустановок споживачів, затверджені наказом Мінпаливенерго України від 25.07.2006 № 258 (у редакції наказу від 13.02.2012 №91), якими унормовано організаційні й технічні вимоги щодо експлуатації електроустановок споживачів;

Правила експлуатації електро-захисних засобів, затверджені наказом Міністерства праці та соціальної політики України від 05.06.2001 № 253, в яких визначено перелік засобів захисту, вимоги до них, норми випробувань, порядок їх застосування, зберігання, а також норми комплектування електроустановок;

Правила улаштування електроустановок визначають вимоги до електроустановок, принципи будови електроустановок, особливі вимоги до окремих вузлів і комунікацій (наказ Міністерства енергетики та вугільної промисловості України від 24.07.2017 № 476); ДСТУ 2843-94 «Електротехніка. Основні поняття.

Терміни та визначення», яким установлені терміни і визначення основних понять з електробезпеки. Одним з основних засобів захисту від напруги є правильно виконане заземлення. Далі проводяться його розрахунки. Визначення розрахункового значення питомого опору ґрунту:

$$\rho_{\text{розр}} = \rho_{\text{гр}}\psi, (\text{Ом}) \quad (5.1)$$

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						96
Змн.	Арк.	№ докум.	Підпис	Дата		

$$R_0 = \frac{54}{2 \cdot 3,14 \cdot 9} \left[\ln \frac{2 \cdot 9}{0,00175} + \frac{1}{2} \ln \frac{4 \cdot 4,5 + 9}{4 \cdot 4,5 - 9} \right] = 2,77$$

де $\rho_{гр} = 30$ Ом – питомий опір ґрунту (чорнозем).

ψ – кліматичний коефіцієнт, що враховує сезонні коливання вологості ґрунту, приймають від 1.2 до 2.0. Для даного випадку приймається $\psi = 1.8$.

Довжина вертикального заземлювача приймається з умови $\frac{l}{l} = 1; 2; 3(3)$,

де l – відстань між заземлювачами, м;

l – довжина заземлювачів, м.

$$t = \frac{l}{2} + t_0, \text{ (м)} \quad (5.2)$$

де $t_0 \geq 0,5$ (м)

t_0 – глибина, на яку закопали заземлювач, $t_0 = 1,5$.

Довжина заземлювачів приймається рівною 9 м, а відстань між заземлювачами – 3 м. Тоді $\frac{l}{l} = 2.1$. Розраховується t :

$$t = \frac{9}{2} + 1,5 = 4,5 \text{ (м)}$$

Вибирається система розподілу заземлювачів по контуру. Далі відбувається розгляд опору одного вертикального заземлювача:

$$R_0 = \frac{54}{2 \cdot 3,14 \cdot 9} \left[\ln \frac{2 \cdot 9}{0,00175} + \frac{1}{2} \ln \frac{4 + 4,5 + 9}{4 \cdot 4,5 - 9} \right] \quad (5.3)$$

Для штучного заземлення в якості електродів приймають сталеві труби $d = 0.035 \times 0.05$ (м);

Визначається кількість вертикальних заземлювачів. При $U = 1000$ В необхідний опір $R_m \leq 4$ (Ом) для мережі змінного току $R_m = 2$.

					КРМ.КІ.1.884-03.1.1	Арк.
						97
Змн.	Арк.	№ докум.	Підпис	Дата		

$$x = \frac{R_0}{R_m} \quad (5.4)$$

де $n = \frac{2,77}{2} = 1,375$, округляється до найближчого стандарту й стає рівним

Далі визначається опір системи вертикальних заземлювачів:

$$R_{cv} = \frac{R_0}{n \cdot \eta_b} \text{ (Ом)} \quad (5.5)$$

Де n – округлене число вертикальних заземлювачів до найближчого стандарту, $n = n$ (шт).

η_b – коефіцієнт використання вертикальних заземлювачів (0.78).

$$x = \frac{2,77}{2 \cdot 0,78} = 1,78 \text{ (Ом)}.$$

Потім розраховується опір сполучної смуги при розміщенні по контуру:

$$L = n \cdot l, \text{ (м)} \quad (5.6)$$

$$L = 2 \cdot 3 = 6 \text{ (м)}$$

$$R_n = \frac{\rho_{рвсч}}{2\pi L \eta_r} \cdot l n = \frac{l^2}{dt_0}, \text{ (Ом)} \quad (5.7)$$

де η_r – коефіцієнт використання горизонтальних заземлювачів (0.55).

$$R_n = \frac{54}{2 \cdot 3,14 \cdot 6 \cdot 0,55} \cdot l n = \frac{36}{0,00175 \cdot 1,5}, \text{ (Ом)}$$

Визначається загальний опір системи:

$$R_c = \frac{R_n \cdot R_{cv}}{R_n + R_{cv}}, \text{ (Ом)} \quad (5.8)$$

Якщо $R_{cv} > R_t$, то треба збільшити n і провести повторний розрахунок.

$$R_c = \frac{5,784 \cdot 4,96}{5,784 + 4,96} = 2,67 \text{ (Ом)}$$

Отримали заземлення менше чотирьох Ом, що задовольняє умові.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						98
Змн.	Арк.	№ докум.	Підпис	Дата		

Лінія електромережі для живлення ПК виконується шляхом прокладання фазового, нульового робочого й нульового захисного провідників, як окрема групова трьох-провідна мережа.

ПК підключається до електричної мережі за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення. Тому що ПК буде розташований біля стіни, електромережа штепсельних розеток для живлення ПК прокладається по підлозі поруч зі стінами приміщення в заземлених металевих трубах і гнучких металевих рукавах.

Якщо в приміщенні обслуговуються більше 5 комп'ютерів, то необхідно на видному і доступному місці встановити аварійний резервний вимикач, який може відключити електричне живлення приміщення крім освітлення. Підлога всієї зони обслуговування, ремонту й налагодження ПК повинна бути покрита діелектричними килимками, або викладена ізолюючими підстилками (шириною не менше 0.75-0.8 м) для ніг. Заземлені конструкції, що знаходяться у приміщенні повинні бути надійно захищені діелектричними щитками або сітками.

5.4 Пожежна профілактика

При визначенні видів та кількості первинних засобів пожежогасіння слід враховувати фізико-хімічні та пожежонебезпечні властивості горючих речовин, їх ставлення до вогнегасною речовин, а також площа виробничих приміщень, відкритих майданчиків і установок.

Азбестові полотна, грубошерсті тканини і повсть розміром не менше 1 x 1 м призначені для гасіння невеликих вогнищ пожеж при займанні речовин, горіння яких не може відбуватися без доступу повітря. Ємності для піску, що входять в конструкцію пожежного стенду, повинні бути місткістю не менше 0,1 куб. м. Конструкція ящика повинна забезпечувати зручність діставання піску та виключати попадання опадів.

Комплектування технологічного обладнання вогнегасниками здійснюється згідно з вимогами технічних умов (паспортів) на це обладнання або відповідними «Правилами пожежної безпеки». Вибір типу і розрахунок

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						99
Змн.	Арк.	№ докум.	Підпис	Дата		

необхідної кількості вогнегасників слід проводити в залежності від їх вогнегасної здатності, граничної площі, класу пожежі горючих речовин і матеріалів в закритому приміщенні або на об'єкті згідно ІСО N 3941 – 77. В замкнутих приміщеннях об'ємом не більше 50 куб. м для гасіння пожеж замість переносних вогнегасників, або додатково до них можуть бути використані вогнегасники порошкові.

Вибираючи вогнегасник з відповідним температурним межею використання, необхідно враховувати кліматичні умови експлуатації будівель та споруд. Вогнегасники, відправлені з підприємства на перезарядку, повинні замінюватися відповідною кількістю заряджених вогнегасників. При захисті приміщень ПК слід враховувати специфіку взаємодії вогнегасних речовин з захищеними обладнанням, виробами, матеріалами і т. п. Дані приміщення слід обладнати вуглекислотними вогнегасниками з урахуванням гранично допустимої концентрації вогнегасної речовини.

Приміщення, обладнані автоматичними стаціонарними установками пожежогасіння, забезпечуються вогнегасниками на 50%, виходячи з їх розрахункової кількості. Відстань від можливого вогнища пожежі до місця розміщення вогнегасник не повинна перевищувати 20 м для громадських будівель і споруд; 30 м для приміщень категорій А, Б і в; 40 м для приміщень категорій Г; 70 м для приміщень категорії Д.

Використання первинних засобів пожежогасіння для господарських та інших потреб, не пов'язаних з гасінням пожежі, не допускається. Тип даної будівлі – громадське приміщення, а можливий клас пожежі – (Е) (оскільки у приміщенні багато комп'ютерів). Використовуючи ці дані, визначається необхідна кількість вогнегасників – один порошковий вогнегасник місткістю 5 літрів.

5.5 Виробнича санітарія

Умови праці визначаються характером і важкістю роботи, яка виконується, а також параметрами навколишнього виробничого середовища.

					<i>КРМ.КІ.1.884-03.1.1</i>	Арк.
						00
Змн.	Арк.	№ докум.	Підпис	Дата		

Плануючи робоче місце, виходять з того, щоб робітник протягом зміни не знижував встановленої продуктивності праці, витрачаючи мінімум фізичних зусиль, був захищений від впливу небезпечних і шкідливих виробничих факторів.

Важливе значення мають також робоча поза, темп і ритм виконуваної роботи. Виключаються статичні м'язові напруги і, як наслідок, профзахворювання (викривлення хребта, розширення вен, плоскостопість) при робочій позі, яка забезпечує розташування центру ваги тіла в межах площі опори.

Загальні ергономічні вимоги до робочих місць викладено в міждержавних стандартах ГОСТ 12.2.032-78 (під час виконання робіт сидячи) та ГОСТ 12.2.033-78 (під час виконання робіт стоячи). Робочим місцем вважають місце постійного або тимчасового перебування працівника в процесі трудової діяльності. У разі вільного ритму роботи темп її визначається самим працівником відповідно до його індивідуальних можливостей.

В процесі одноманітних, короткочасних робочих операцій (наприклад, розфасовці та упакувці товарів) можливе відчуття монотонності праці. Чітко виражена монотонність виконуваної роботи призводить не тільки до передчасної втоми, але й до швидкого нервового виснаження.

Втомлюваність можна знизити періодичними змінами робочої пози і ритму виконуваних операцій, зміною робочих місць, естетизацією праці, використанням раціональної організації режиму праці та відпочинку. У роботі, пов'язаній з великим м'язовим навантаженням, перерви повинні бути тривалими (до 10 хв.), але не частими. Причому, їх тривалість і частота повинні змінюватися протягом робочої зміни. На основі загальних енерговитрат організму людини розрізняють (згідно ГОСТ 1.005-88) фізичні роботи :

- легкі (категорія I);
- середньої важкості (категорія II);
- важкі (категорія III).

Легкі фізичні роботи :

- категорії I-а з енерговитратами до 139 Вт

					<i>КРМ.КІ.1.884-03.1.1</i>	Арк.
						01
Змн.	Арк.	№ докум.	Підпис	Дата		

– роботи, які виконуються сидячи і супроводжуються незначною фізичною напругою;

– I-б – з енерговитратами до 140 ... 174 Вт – роботи, які виконуються сидячи, стоячи або пов'язані з ходінням та супроводжуються деяким фізичним напруженням; Фізичні роботи середньої важкості охоплюють види діяльності із затратами енергії 175 ... 232 Вт – категорія II-а і 233 ... 290 Вт – категорія II-б. – до категорії II-а відносяться роботи, пов'язані з постійним ходінням і переміщенням дрібних (до 1 кг) виробів чи предметів в положенні сидячи та вимагають деякої фізичної напруги; – до категорії II-б – роботи, пов'язані з ходінням, переміщенням і перенесенням вантажів до 10 кг і супроводжуються помірним фізичним напруженням.

Важкі фізичні роботи (категорії III) з енерговитратами понад 290 Вт пов'язані з постійним переміщенням і перенесенням значних (більше 10 кг) вантажів та вимагають великих фізичних зусиль. На ефективність виконання робіт, самопочуття і стан здоров'я працівника значно впливає навколишнє виробниче середовище.

Виробниче середовище характеризують мікрокліматичні умови, склад і ступінь запиленості повітря, рівні шуму та вібрації, вид і якість освітлення, наявність та інтенсивність теплових, електромагнітних випромінювань і деяких інших чинників. Його вважають оптимальним, якщо несприятливі впливи на робітника відсутні, і нормальним, якщо вони знаходяться в межах допустимих рівнів.

Важливе значення у формуванні виробничого середовища відіграє технічна естетика, яка дозволяє попередити нервово-психічні перевантаження робітників. Це досягається використанням раціональних архітектурнопланувальних рішень, високохудожніх інтер'єрів, а також науково обґрунтованого кольорового фарбування стін, стель та устаткування. Кольори для фарбування та устаткування вибирають залежно від їх функціонального призначення, умов зорової роботи, особливостей клімату, орієнтації вікон відносно сторін світу.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						02
Змн.	Арк.	№ докум.	Підпис	Дата		

Так, в приміщеннях з великим тепловиділенням огорожувальні конструкції та поверхні устаткування фарбують в кольори холодних тонів; в приміщеннях з природним освітленням – в кольори теплих тонів. Використання з метою охорони праці різноманітних кольорів та знаків безпеки регламентовано ГОСТ 12.4.026-76.

Для забезпечення нормальних умов праці санітарні норми встановлюють на одного працівника: об'єм виробничих приміщень не менше 15 м³; площа приміщення, огороженого стінами чи глухими перегородками – не менше 4,5 м²; висота – 3,2 м. На підприємствах передбачаються кабінети психофізичного розвантаження (емоційно-вольового тренування), приміщення для виконання фізичних вправ, санітарно-побутові приміщення.

Санітарно-гігієнічні умови праці на підприємствах визначаються в основному рівнем наявних на робочих місцях шкідливих виробничих чинників. Перевірка відповідності умов праці вимогам санітарного законодавства здійснюється шляхом паспортизації цехів, дільниць, робочих місць (карти умов праці).

Вимірювання параметрів виробничого середовища виконують за договором працівники санепідемстанцій. За результатами цих вимірювань заповнюють відповідні паспорти. У випадку невідповідності значень виміряних параметрів чинним нормам на підприємстві розробляються і здійснюються комплексні заходи щодо досягнення встановлених нормативів безпеки, гігієни праці та виробничої санітарії.

Завдання попередження гострих і хронічних професійних захворювань та отруєнь вирішує виробнича санітарія – система організаційних заходів і технічних засобів, які запобігають впливу на працівників шкідливих виробничих чинників чи зменшують його. Проводиться розрахунок світлового потоку.

$$\Phi = \frac{K * E_H * Z * S * 100}{n * \eta}, \text{ (Лк)} \quad (5.9)$$

де S – площа приміщення. K – коефіцієнт запасу.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						03
Змн.	Арк.	№ докум.	Підпис	Дата		

Оскільки в цьому випадку передбачається робота із ПК, то приймається розмір, як для диспетчерських, пультів операторів, контрольно-вимірювальних приладів. $E_n = 200$ лк. – нормована мінімальна освітленість.

Тип світильника – ЛСП, отже коефіцієнт нерівномірності освітлення $Z = 1.1$.

Для того щоб порахувати світловий потік необхідно обчислити висоту над робочою поверхнею, відстань між центрами світильників (L), кількість світильників (n), індекс приміщення (i) й коефіцієнт використання світлового потоку (η).

$$H_p = H - h_p - 1, \text{ (м)} \quad (5.10)$$

де $H_p = 2$.

$$\frac{L}{H_p} = x, \text{ (м)} \quad (5.11)$$

$$n = \frac{S}{L^2}, \text{ (шт)} \quad (5.12)$$

$$i = \frac{S}{H \cdot (A+B)}, \text{ (шт)} \quad (5.13)$$

де $x = 1.4$ для світильників денного світла ПВЛП.

$A = 5$ м і $B = 6$ м, довжина і ширина приміщення.

$H = 3$ м – висота приміщення.

$h_p = 0$ м – відстань від світильника до стелі

$$H_p = 3 - 0 - 1 = 2 \text{ (м)}$$

$$L = H_p \cdot 1.4 = 2 \cdot 1.4 = 2.8 \text{ (м)}$$

$$n = \frac{30}{7.84} = 3.83,$$

проводиться округлення до більшого найближчого цілого. Результат – 4 світильника.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						04
Змн.	Арк.	№ докум.	Підпис	Дата		

$$i = \frac{30}{2*(5+6)} = 1,36.$$

Визначаємо коефіцієнт використання світлового потоку, при $P_n = 50\%$ (коефіцієнт відображення стіни), $P_c = 30\%$ (коефіцієнт відображення), $\eta = 36\%$.

$$\Phi = \frac{1.5*200*1.1*30*100}{3,83 * 0.42} = 6,155 \text{ (лм)}.$$

$$\Phi = 6,155$$

Висновки до п'ятого розділу

Було розглянуто потрібний тип приміщення і виробництва та визначені їх належні умови безпеки. Представлені необхідні норми пожежної профілактики та виробничої санітарії. Був здійснений розрахунок електричного опору системи та визначення потрібної кількості освітлювачів, потрібних для безпечної роботи.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						05
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗАГАЛЬНІ ВИСНОВКИ

1. У результаті аналізу предметної області було виявлено основні риси ігор жанру «Симулятор».

2. Проаналізовано сучасні аналоги аркадних ігор жанру «Симулятор кулінарії». Представлено основні необхідні елементи та режими гри даного жанру.

3. Здійснено постановку задачі з зазначенням бажаного результату, що є необхідним для розробки проектної документації та демонстраційної версії гри.

4. У процесі проектування було розроблено концептуальний документ гри, необхідний для демонстрації інвестору або видавцю проекту.

5. Розроблено дизайнерський документ гри, який описує основні особливості розроблюваної гри.

6. У ході проведеної роботи було розроблено програмну та візуальну структуру для реалізації демонстраційної версії гри.

7. Розроблено моделі навколишнього середовища, бонусів та інших елементів гри. Усі об'єкти було розташовані на сцені належним чином.

8. Проект був зкомпільований та у ході його тестування не було виявлено ніяких проблем та помилок.

9. Розробка даного проекту є вигідною, бо має швидку окупність і перспективи додаткового прибутку у майбутньому. Розробка ігор є прибутковим бізнесом, попит на який з кожним роком усе зростає.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						06
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ СКОРОЧЕНЬ І ТЕРМІНІВ

NPC – персонаж в комп'ютерних і настільних рольових іграх, керований програмою або майстром, в останньому випадку іноді може називатися майстровим персонажем.

ШІ – властивість штучних інтелектуальних систем виконувати творчі функції, які традиційно вважаються прерогативою людини.

USP – це частина конкурентної переваги, на основі якої клієнт вибирає компанію або товар (виходячи з властивостей товару або послуги).

ПК – персональний комп'ютер.

ПП – програмний продукт.

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						07
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Unity Manual* [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/Manual/index.html>
2. Багато критеріальна оптимізація [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Multi-objective_optimization
3. Performance of cooking simulator games [Електронний ресурс]. – Режим доступу: <https://www.data.ai/en/apps/ios/app/anime-cooking-simulator-games/>
4. Методи багатокритерійної оптимізації [Електронний ресурс]. – Режим доступу: <https://foundry.kpi.ua/wp-content/uploads/2020/05/teslyuk-vm-metody-bagatokryterialnoyi-optymizacziyi.pdf>
5. *Cooking Simulator* – Вікіпедія [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Cooking_Simulator
6. *Overcooked 2* – Вікіпедія [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Overcooked_2
7. *Cook, Serve, Delicious!* – Вікіпедія [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Cook,_Serve,_Delicious!_3
8. *NavMeshAgent* – Вікіпедія [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/ru/530/Manual/class-NavMeshAgent.html>
9. *Mixamo* – Вікіпедія [Електронний ресурс]. – Режим доступу: <https://www.mixamo.com/#/>
10. Francesco Sapia. *Unity UI Cookbook*. — Packt Publishing: 2015. – 284 с.: Іл.
11. Alan Thorm. *Unity Animation Essentials*. — Packt Publishing: 2015. – 200 с.: Іл.
12. Басюркіна Н.Й., Свистун Т.В. "Оцінка науково-технічної ефективності: Методичні вказівки". Одеса: ОНТУ, 2023 р. 18 с

					<i>KPM.KI.1.884-03.1.1</i>	Арк.
						08
Змн.	Арк.	№ докум.	Підпис	Дата		