

# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

\_\_\_\_\_ Ігор ШЕЛЕХОВ  
(підпис)

\_\_\_\_\_ червня 2023 р.

## КВАЛІФІКАЦІЙНА РОБОТА на здобуття освітнього ступеня бакалавр

зі спеціальності 122 - Комп'ютерних наук,  
освітньо-професійної програми «Інформатика»  
на тему: Інформаційне і програмне забезпечення ігрової системи “Тривимірні  
шашки”  
здобувача групи ІН - 92 Гієнка Михайла Юрійовича

Кваліфікаційна робота містить результати власних досліджень.  
Використання ідей, результатів і текстів інших авторів мають посилання на  
відповідне джерело.

\_\_\_\_\_ Гієнко М.Ю.  
(підпис)

Керівник,  
кандидат технічних наук

Коробов А.Г.

\_\_\_\_\_ (підпис)

Суми – 2023

**Сумський державний університет**  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

\_\_\_\_\_ (підпис)

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

### на здобуття освітнього ступеня бакалавра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»  
здобувача групи ІН-92 Гієнка Михайла Юрійовича

1. Тема роботи: Інформаційне і програмне забезпечення ігрової системи "Тривимірні шахи" затверджую наказом по СумДУ від «01» червня 2023 р. № 0475-VI
2. Термін здачі здобувачем кваліфікаційної роботи до 09 червня 2023 року
3. Вхідні дані до кваліфікаційної роботи \_\_\_\_\_
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)  
1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.  
2) Огляд ігрового двигуна UE4, що використовуються для створення гри. 3) Розробка та дизайн гри в шахи. 4) Перевірка та наліз результатів.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

Завдання прийняв до виконання \_\_\_\_\_ (підпис)

Керівник \_\_\_\_\_ (підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>		
2	<i>Огляд ігрового двигуна UE4, що використовуються для створення гри</i>		
3	<i>Розробка та дизайн гри</i>		
4	<i>Перевірка та наліз результатів</i>		
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти \_\_\_\_\_ (підпис)

Керівник \_\_\_\_\_ (підпис)

## АНОТАЦІЯ

**Записка:** 37 стр., 48 рис., 0 додаток, 10 використаних джерел.

**Обґрунтування актуальності теми роботи** – Актуальність теми "шашки на UE4" полягає в тому, що розробка комп'ютерних ігор продовжує займати одне з провідних місць у галузі інформаційних технологій. Шашки - це класична настільна гра, яка вже багато років є популярною серед любителів логічних ігор. Створення віртуальної версії шашок на платформі Unreal Engine 4 (UE4) може мати кілька ключових переваг і наступні обґрунтування підкреслюють актуальність цієї теми:

**Навчальна цінність:** Розробка віртуальних ігор є потужним засобом для вивчення програмування та графічного дизайну. Робота над проектом "шашки на UE4" дозволить вам розширити свої знання та навички у розробці ігор, включаючи програмування штучного інтелекту, обробку фізики, візуалізацію та реалістичну анімацію.

**Популярність та потенційна аудиторія:** Шашки є широко відомою та улюбленою грою в багатьох країнах. Розробка віртуальної версії шашок на UE4 може зацікавити велику аудиторію гравців, що відкриє широкі можливості для розповсюдження гри та її комерційного успіху.

**Використання передових технологій:** UE4 є потужним та популярним інструментарієм для розробки відеоігор. Використання UE4 у проекті "шашки" дозволить вам ефективно використовувати передові можливості двигуна, такі як візуалізація високої якості, фізична симуляція, звукові ефекти та багато іншого.

**Ігрові можливості та інновації:** Розробка віртуальних шашок на UE4 відкриває широкий простір для вдосконалення та додавання нових ігрових елементів. Ви можете експериментувати з різними варіантами правил гри, режимами гри, мережевою взаємодією та іншими функціями, що розширять геймплей та забезпечать унікальний досвід гравця.

Враховуючи перелічені обґрунтування, тема "шашки на UE4" є актуальною та цікавою для дослідження та реалізації у вигляді віртуальної гри.

**Мета роботи** — Мета роботи полягає в розробці і створенні віртуальної гри в шашки на платформі Unreal Engine 4 (UE4). Основні завдання та цілі проекту можуть включати:

Розробка геймплею та механіки гри: Створення правильного функціоналу для гри в шашки, включаючи коректні правила гри, логіку руху фігур, спеціальні ходи та інші аспекти, які характеризують цю настільну гру.

Реалізація імітації штучного інтелекту: Розробка алгоритму для комп'ютерного противника, які володіють найпростішим рівнем інтелектуальності. Особливо підходить для дітей, або початківців в цій грі.

Графічне оформлення та візуалізація: Створення візуально привабливого інтерфейсу гри, включаючи відображення дошки, шашок, анімацію руху фігур та інші візуальні ефекти, які покращують геймплей та іммерсивність гри.

Тестування та оптимізація: Проведення тестування гри для виявлення помилок, удосконалення та виправлення їх. Оптимізація продукту для забезпечення плавної та ефективної роботи на різних платформах.

Мета роботи "створення гри в шашки на UE4" полягає в успішній реалізації всіх вищезгаданих завдань, щоб створити захоплюючу та задоволення віртуальну гру в шашки для гравців.

**Методи дослідження** — Аналіз існуючих ігор в шашки: Проведення дослідження та аналізу існуючих віртуальних і настільних ігор в шашки. Вивчення їх геймплею, правил, механік та візуального оформлення. Це дозволить отримати інсайти та ідеї для створення власної гри в шашки.

Літературний огляд: Проведення літературного огляду на тему гри в шашки, включаючи дослідження теорії гри, стратегій, відомих варіантів шашок та інших аспектів. Це дозволить отримати глибше розуміння гри та розробити ефективніші механіки та алгоритми для комп'ютерного противника.

## ЗМІСТ

ВСТУП.....	6
1 ПРОТОТИПУВАННЯ .....	9
1.1 Інструменти розробки.....	10
1.2 Постановка задачі .....	12
2 МЕТОДИКА ВИРІШЕННЯ ПОСТАВЛЕНИХ ЗАДАЧ.....	13
2.1 Проектування Checker(шашка).....	13
2.2 Проектування Desk(діщечка).....	16
2.3 Загальні взаємовідносини між об'єктами .....	17
3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	19
3.1 Створення необхідних об'єктів та їхній функціонал.....	19
3.2 Створення рівнів та віджетів .....	33
ВИСНОВКИ.....	36
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	37

## ВСТУП

**Актуальність** - створення гри в шашки на Unreal Engine 4 (UE4) може бути обґрунтована наступними чинниками:

**Популярність шашок:** Шашки є однією з найпопулярніших настільних ігор у світі, з багатою історією та громадською базою гравців. Створення віртуальної гри в шашки дасть можливість гравцям насолоджуватися цією класичною грою у вигляді ігрового досвіду, доступного на комп'ютерах та інших платформах.

**Економічний потенціал:** Віртуальні настільні ігри є популярним розважальним контентом, що привертає широку аудиторію. Створення гри в шашки на UE4 може мати комерційний потенціал, якщо буде вдало реалізовано та привабливо представлено на ринку ігрової індустрії[3].

**Розвиток навичок розробки на UE4:** Розробка гри в шашки на UE4 може бути чудовим способом вдосконалення навичок розробки ігор. UE4 надає потужні інструменти та ресурси для створення ігрового контенту, і робота над проектом шашок дозволить розширити розуміння різних аспектів розробки ігор, включаючи геймплей, анімацію, штучний інтелект та візуалізацію.

**Потенціал для інновацій:** Розробка гри в шашки на UE4 може бути площиною для інновацій та новаторських рішень. Ви можете впроваджувати нові механіки гри, альтернативні варіанти правил або надати особливий візуальний стиль гри. Це дозволяє експериментувати зі звичайними правилами та створювати унікальний геймплей, який приверне увагу гравців.

**Спільнота гравців та конкурентне середовище:** Гра в шашки має велику громадську базу гравців, які активно займаються цією грою на різних платформах[10]. Створення якісної гри в шашки на UE4 дозволить приєднатися до спільноти гравців та створити конкурентну гру, яка задовольнить їх потреби та вимоги.

Отже, створення гри в шашки на UE4 є актуальним завданням, яке поєднує популярність класичної гри, економічний потенціал, розвиток навичок розробки, потенціал для інновацій та можливість приєднатися до вже існуючої громади гравців.

### **Об'єкт дослідження -**

#### 1.Геймплей та механіка гри:

Дослідження різних варіантів правил гри в шашки: звичайні шашки, міжнародні шашки тощо. Розгляд можливості реалізації додаткових правил, які збагатять геймплей[9].

Аналіз та розробка алгоритмів для обробки ходів та перевірки правильності руху фігур.

Розробка штучного імітації комп'ютерного противника: дослідження та реалізація алгоритмів прийняття рішень для опонента.

#### 2.Графіка та візуалізація:

Дослідження та вибір різних стилів графічного оформлення шашок та дошки, включаючи класичні та сучасні варіанти.

Впровадження анімації руху шашок, збиття фігур, а також інших ефектів, які покращують реалізм та іммерсивність гри.

Реалізація візуальних ефектів для підсвічування шашок, попередження про неправильні рухи або збиття, використання звукових ефектів, та звукового супроводу[7].

### **Предмет дослідження**

Геймплей та механіка гри: Дослідження правил гри в шашки, включаючи рух фігур, правила захоплення фігур супротивника, варіанти ходів та стратегії. Аналіз та реалізація правильних механік гри дозволить забезпечити аутентичний геймплей шашок.

Інтерфейс користувача: Дослідження та розробка інтерфейсу, який забезпечує зручну та інтуїтивно зрозумілу взаємодію гравця з грою. Це включає

розміщення інформації на екрані, кнопки та елементи керування, а також звукові та візуальні ефекти.

**Графіка та візуалізація:** Дослідження та створення візуально привабливих моделей фігур, текстур, освітлення сцени та ефектів. Важливо досліджувати різні стилі візуалізації та забезпечувати естетично задовільний вигляд гри.

**Тестування та оптимізація:** Дослідження та проведення тестування гри для виявлення помилок, проблем з геймплеєм та недоліків. Оптимізація гри для забезпечення стабільної продуктивності та плавної геймплейної динаміки також є важливим аспектом дослідження.

**Гіпотеза** - розробка ігрової гри в шашки на Unreal Engine 4 (UE4) з використанням реалістичної 3D-графіки та захоплюючого геймплею забезпечить зацікавленість гравців, підвищить рівень їх залученості та створить відмінну ігрову інтеракцію.

Ця гіпотеза передбачає, що розробка гри в шашки на UE4 з використанням сучасних можливостей графічного рендерингу та створення захоплюючого геймплею сприятиме підвищенню зацікавленості гравців у цьому класичному настільному жанрі. Вона передбачає, що реалістична 3D-графіка та добре продумана геймплейна механіка привернуть увагу гравців і зроблять гру в шашки на UE4 вигідною інвестицією для розвитку геймінгової аудиторії.

**Структура.** Дане робота складається зі вступу, аналітичного огляду, постановки задачі, вибір методу розв'язання поставленої задачі, опису програмного забезпечення інформаційної системи, висновків, списку використаних джерел та додатків.



## 1 ПРОТОТИПУВАННЯ

Спочатку потрібно продумати, які класи потрібно створювати, та для чого, які будуть між ними співвідносини, як вони будуть підпорядковуватися правилам це4, та взаємодіяти між собою. Потрібно створити всі можливі класи, та приблизно їхні функції та змінні, які вони будуть використовувати. В даній роботі в нас не так багато об'єктів та не занадто глибока архітектура проєкту, але всерівно потрібно постаратися продумати все наперед, тому що здавалося, в такому легкому проєкті складно помилитися, але не все так просто.

Існує діщечка, на якій відбуваються дії, можна сказати це наш аналог ігрових правил( вона буде відповідати за функції розстановки шашок, зберігати про них інформацію, та оброблювати її, а також оброблювати дії користувача, коли він буде натискати на неї), а також шашка, об'єкт який містить функціонал шашки, її координати, перевірка чи клікнули на неї, тощо.

Створимо основні класи які нам знадобляться для роботи, а саме: BP\_Checker, BP\_Desk, BP\_GameManager, BP\_GameMode, BP\_Player, BP\_PlayerController, BP\_GameInstance, BP\_MainMenuMode

а також класи віджетів WBP\_EndGame, WBP\_HUD, WBP\_Menu, WBP\_Option, WBP\_Warning

Деякі з них існують як класи обгортки, наприклад (BP\_GameManager, BP\_GameMode, BP\_Player, BP\_PlayerController) і в них майже немає функціоналу, або взагалі не існує. А інші класи наповнені багатьма функціями для роботи.

Також заздалегіть потрібно загрузити необхідні файли:

Текстури - для віджетів, а саме різні кнопки взаємодії, текстури для покраски шашок та ігрового поля, а також сплайн трави.

Матеріали - після загрузки текстур створити необхідні матеріали з відповідних сплайнів.

Аудіофайли - музичні файли, та аудіоефекти різних дій, створити клас Master та Master SoundMix.

### 1.1 Інструменти розробки

Ми будемо використовувати інструменти двигуна ue4 та його функції, а саме:

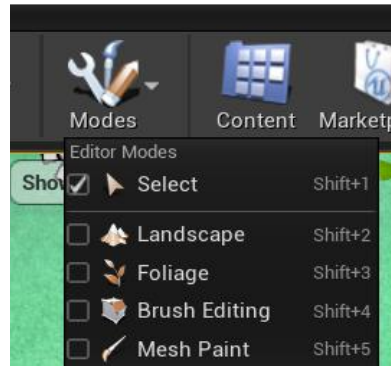


Рисунок 1.1 – Панель управління

Включає в себе багато механізмів налаштування проекту. Ми будемо використовувати деякі з них. Modes для редактировки локацій, для створення фону трави.

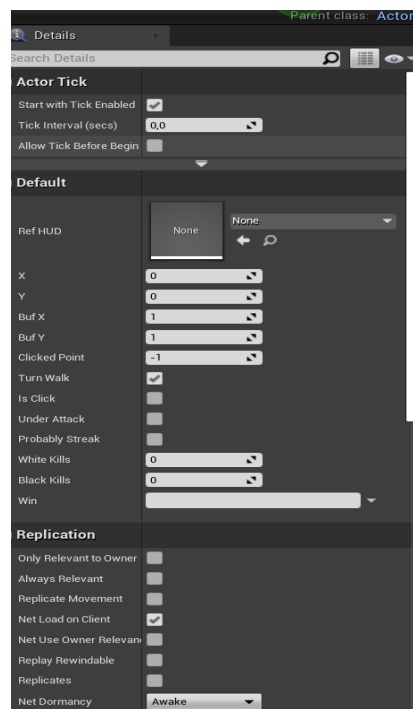


Рисунок 1.2 – Панель деталів об'єкту

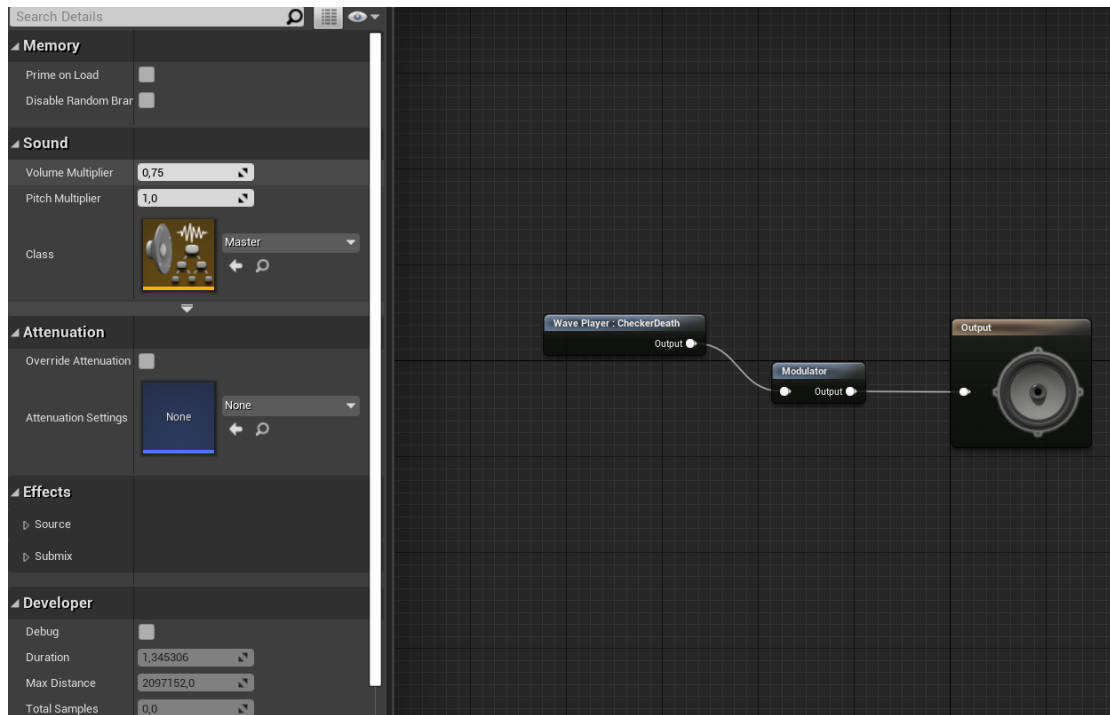


Рисунок 1.3 – Панель редагування музичних та аудіо файлів

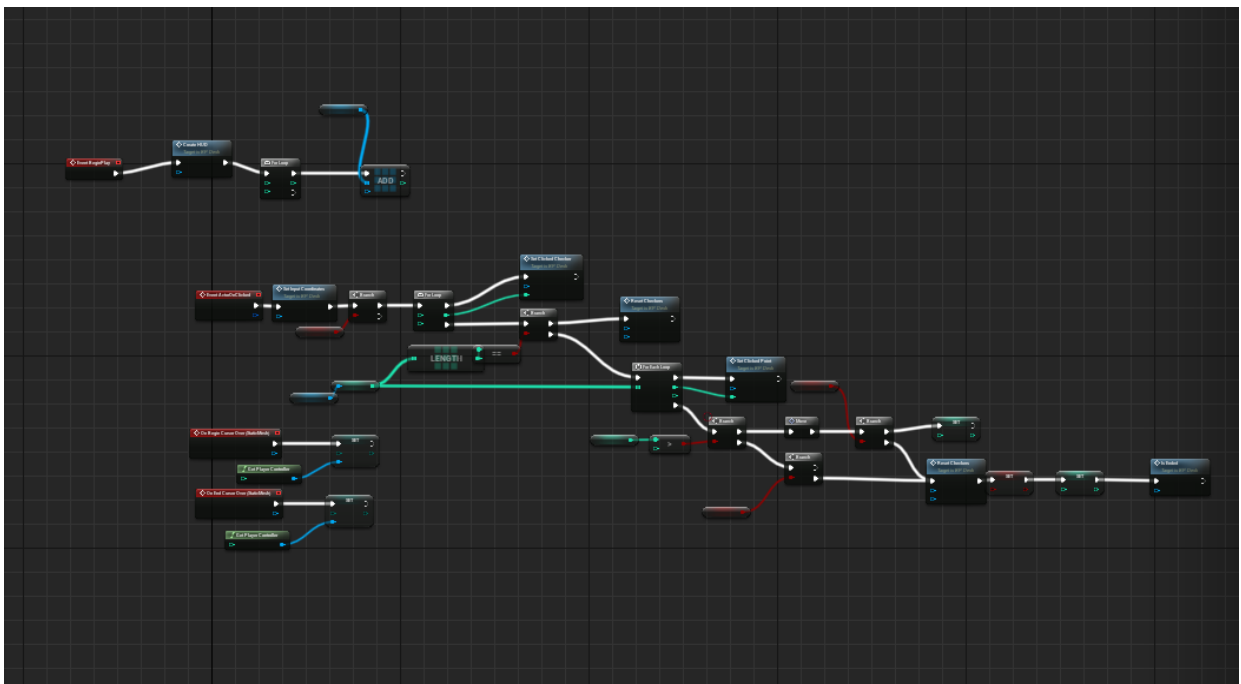


Рисунок 1.4 – Blueprint граф

А також графи br для створення логіки та взаємодії між об'єктами, ігрові правила, обмеження.

## 1.2 Постановка задачі

На початку роботи було ретельно продумано всі аспекти проекту і визначився з послідовністю їх виконання. Особливу увагу приділено чіткому визначенню правил гри, щоб уникнути недорозумінь і плутанини в подальшій розробці. Основні завдання роботи:

1. Детально визначення правил гри та її аспектів, включаючи правила руху фігур, умови перемоги або нічийї, обмеження та можливості руху шашок, а також правила перетворення шашок у дамки.
2. Створення деталей шашок та дощечки та їх кольорових моделей.
3. Створення заднього фону ігрового поля.
4. Алгоритмічна і програмна реалізація правил гри, реалізація механіки взяття шашок, а також перетворення шашок у дамки.
5. Розробка елементів, що реалізують функцію імітації супротивника.
6. Програмна реалізація віджетів початку та закінчення гри, натискання на клавіші, створення рівнів, та зв'язків між ними.
7. Тестування ігрового додатку

## 2 МЕТОДИКА ВИРІШЕННЯ ПОСТАВЛЕНИХ ЗАДАЧ

Незрівнянна, унікальна і захоплююча ідея починає свій шлях з обговорення в команді, зазвичай спочатку нею займається окремо створена група дизайнерів. Після прийняття концепції проекту — сюди входить приблизний сюжет, жанр, цільова аудиторія гравців — команда береться за роботу над найскладнішим — технічною і арт складовими. Чому найскладнішим? Тому що дизайн буде проходити правки і обкатуватися протягом всієї розробки аж до виходу гри. У той час, як технічна частина і візуальна складова дуже складно піддаються повній переробці. В основному ігри створюють на мовах програмування, перевагами яких є швидкість роботи, близькість роботи з процесором[6].

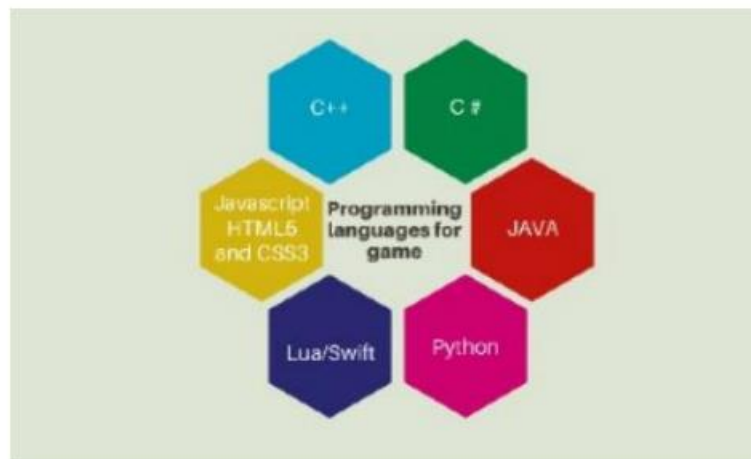


Рисунок 2.1 – Архітектура мов для створення ігор

### 2.1 Проектування Checker(шашка)

Пішак — це актор, який може бути «агентом» у світі. Пішаками може володіти Контролер, вони налаштовані на те, щоб легко приймати вхідні дані, і вони можуть робити різноманітні інші дії, схожі на гравця. Зауважте, що пішак не вважається гуманоїдом. Наш основний об'єкт роботи шашка буде створенна в вигляді Pawn[1].

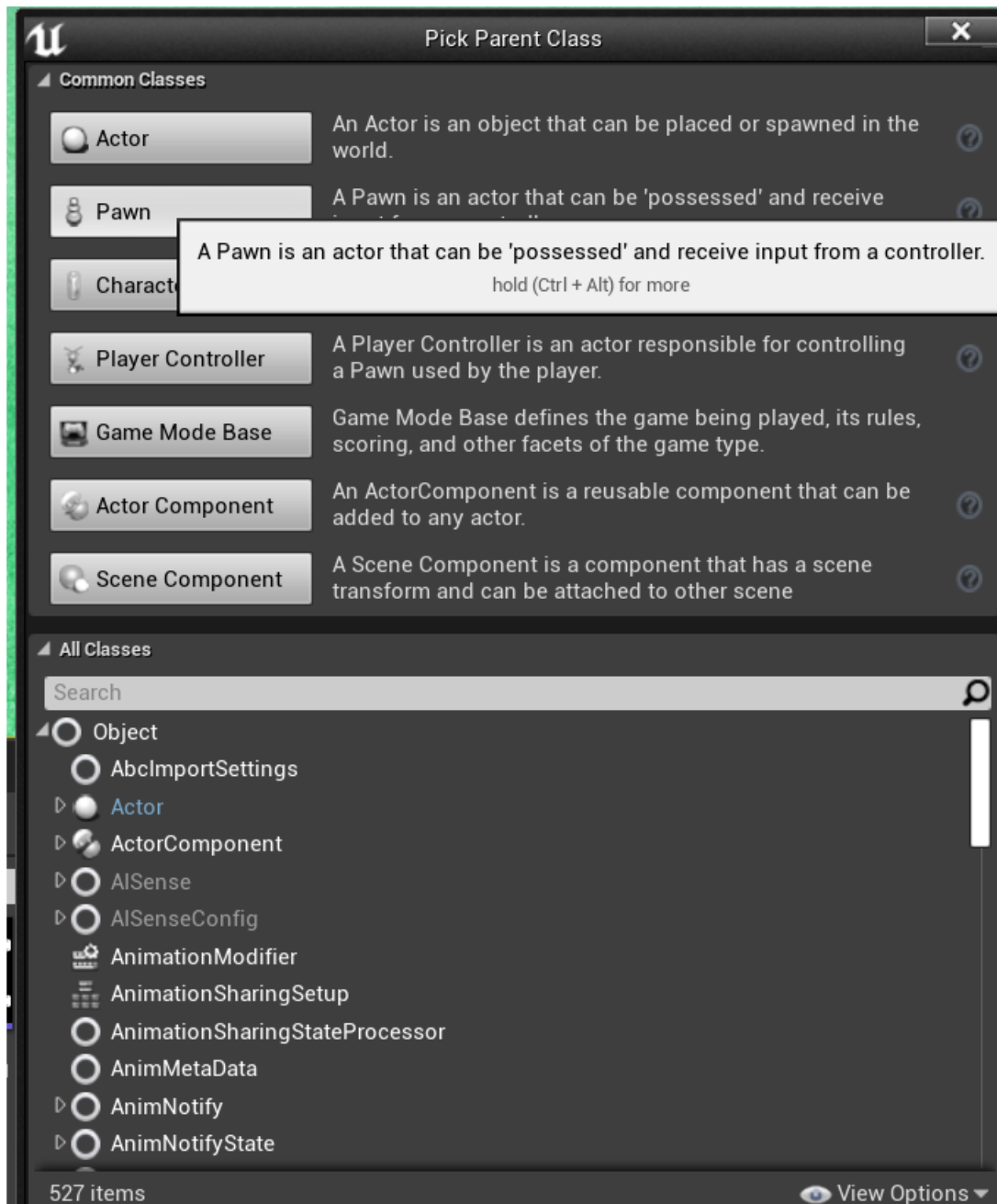


Рисунок 2.2 – Створення класу

Його основні функції та характеристики

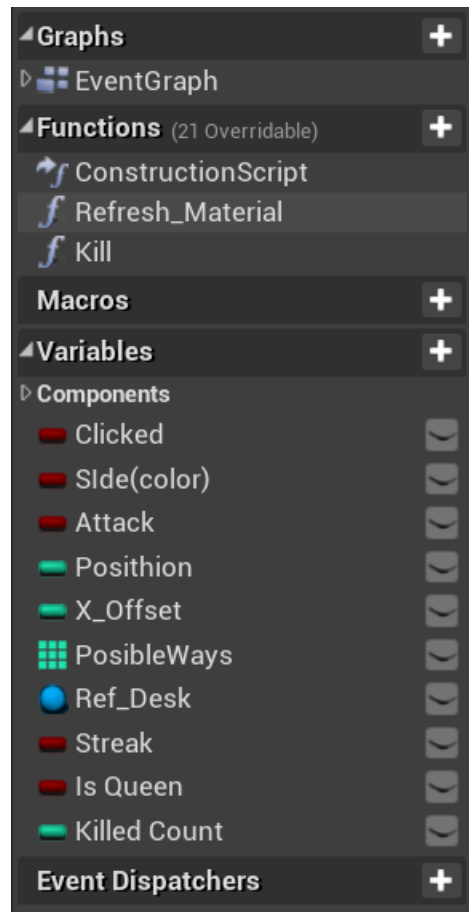


Рисунок 2.3 – Змінні класу Checker

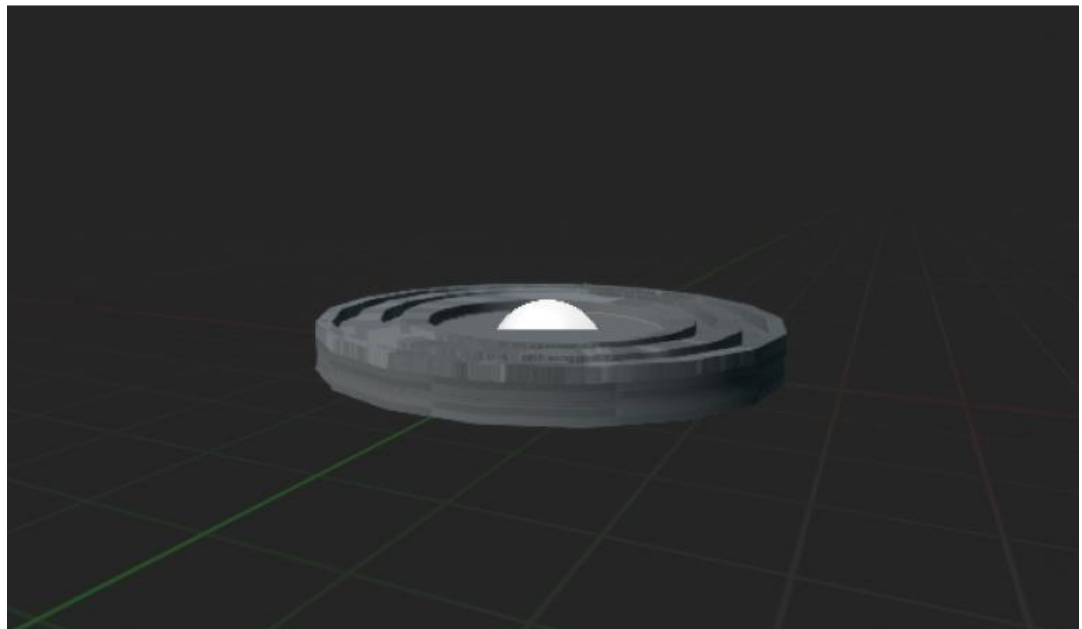


Рисунок 2.4 – Вигляд в 3D

## 2.2 Проектування Desk(дідечка)

Створена у нас також типом Pawn, як і шашка.

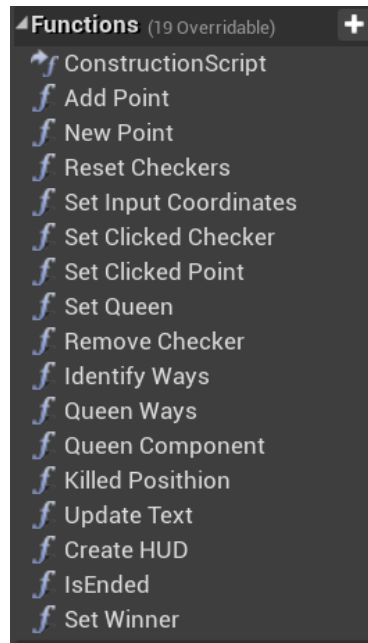


Рисунок 2.5 – Функції класу

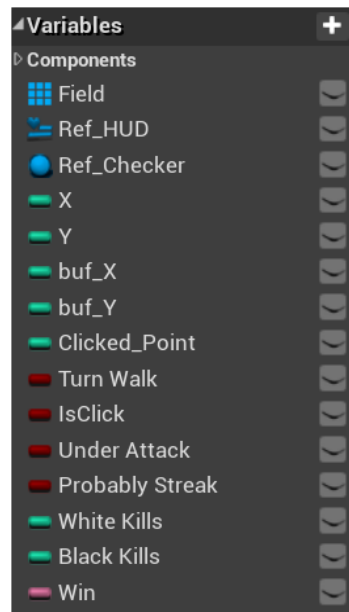


Рисунок 2.6 – Поля класу



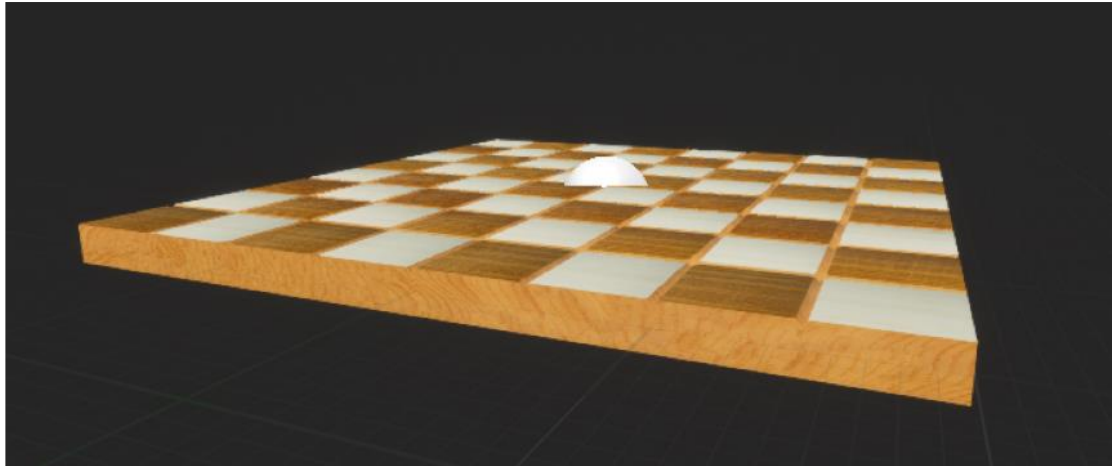


Рисунок 2.7 – Вигляд в 3D

Наш клас буде відповідати деякій аналогії, щось наче глобальних ігрових правил, буде виконувати різні функції зв'язані з переміщенням фігур, натисканням на них, опрацюванням можливих шляхів, перетворення на дамку, ведення рахунку, та кінець гри[5].

### 2.3 Загальні взаємовідносини між об'єктами

Наші актори будуть підпорядковуватися правилам інших класів, таких як BP\_GameMode та BP\_Controller

GameMode - Хоча певні фундаментальні параметри, як-от кількість гравців, необхідних для гри, або спосіб, за допомогою якого ці гравці приєднуються до гри, є загальними для багатьох типів ігор, можливі необмежені варіації правил залежно від конкретної гри, яку ви розробляєте. Незалежно від того, які ці правила, ігрові режими призначені для їх визначення та реалізації. Наразі існує два базові класи, які зазвичай використовуються для режимів гри[1].



Рисунок 2.8 – Створені класи для подальшої розробки

Controller — це нефізичні актори, які можуть володіти пішаком (або похідним від пішака класом, як-от персонаж), щоб контролювати його дії. PlayerController використовується гравцями для керування пішаками, а AIController реалізує штучний інтелект для пішаків, якими вони керують. Контролери беруть під контроль пішака за допомогою функції «Заволодіти» та відмовляються від контролю над пішаком за допомогою функції «Зняти володіння».

І вся наша програма буде огортатися в деякі віджети.

Сам компонент Widget є тривимірним екземпляром Widget Blueprint, з яким ви можете взаємодіяти у своєму ігровому світі.

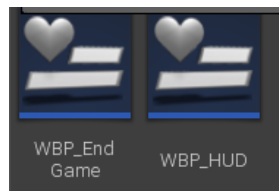


Рисунок 2.9 – Класи віджетів

## 3 ПРОГРАМНА РЕАЛІЗАЦІЯ

### 3.1 Створення необхідних об'єктів та їхній функціонал

Наступні скриншоти описуватимуть повну реалізацію класів створених в рамки цього проекту.

Створюємо основні функції за допомогою bp графів.

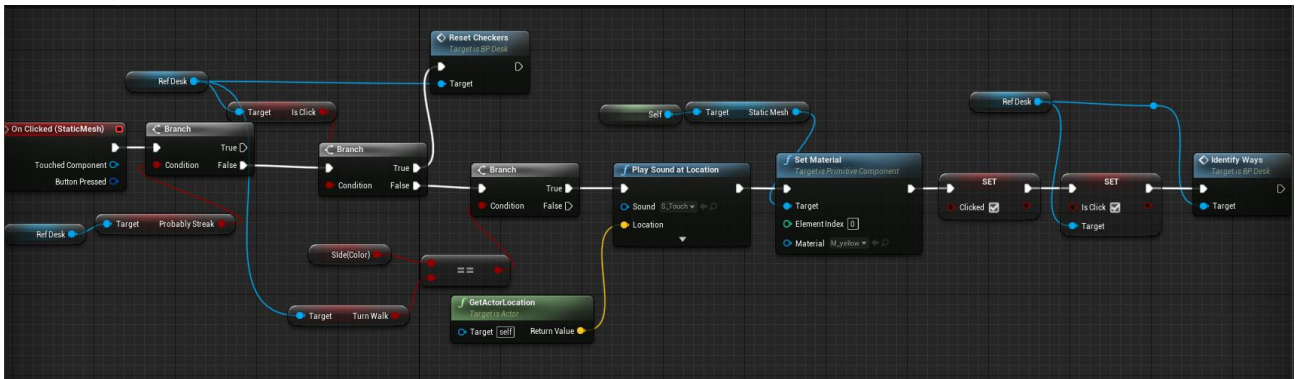


Рисунок 3.1 – Граф подій класу Checker

Що тут відбувається?

Після натискання на шашку відбувається декілька перевірок чи це є хід комп'ютера, чи ця дія відбувається під час ходу декількох взятъ, чи відповідає колір шашки, які повинні зараз ходити. Далі якщо перевірки всі пройдені, то відбувається спавн звуку, в точці шашки, та зміна її матеріалу, змінюються деякі змінні, які відповідають за логіку, чи натиснута саме ця шашка чи ні. Далі запускається функція підрахунку всіх можливих ходів для сторони, яка зараз ходить, якщо таких ходів немає, то гра закінчується в залежності від того, чи незалешилось фігур, або немає ніяких шляхів для ходу[4].

Ось так вигляде граф подій для шашки.

Повна реалізація класу Checker:

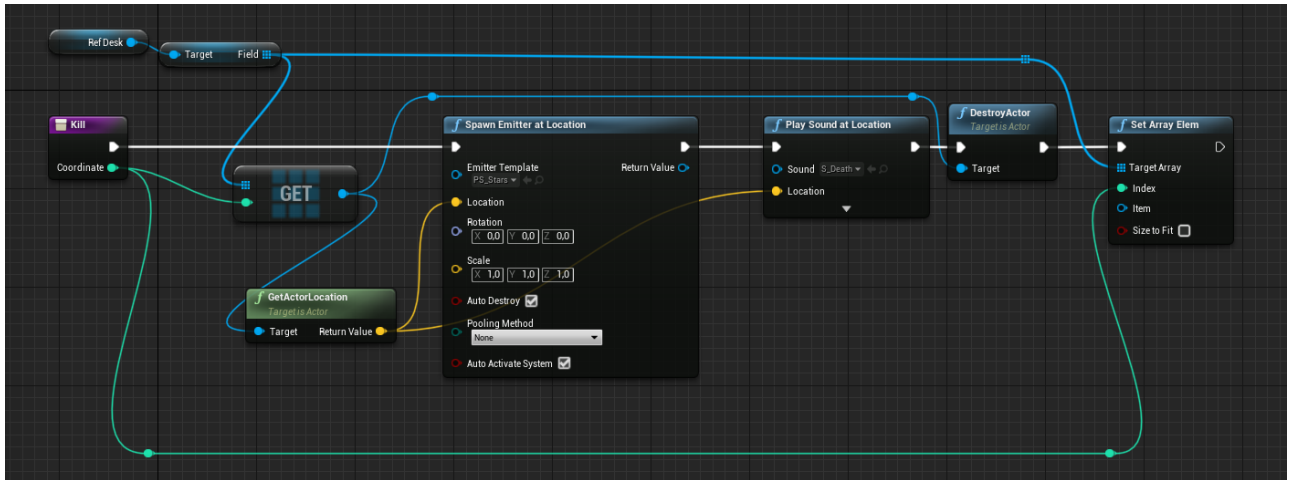


Рисунок 3.2 – Функція Kill

Дана функція виконує одне взяття ворожої фігури, та спавнить в місці шашки, яку побили звук та еміттер(спецеффект), потім знищує актора, та оновлює масив шашок.

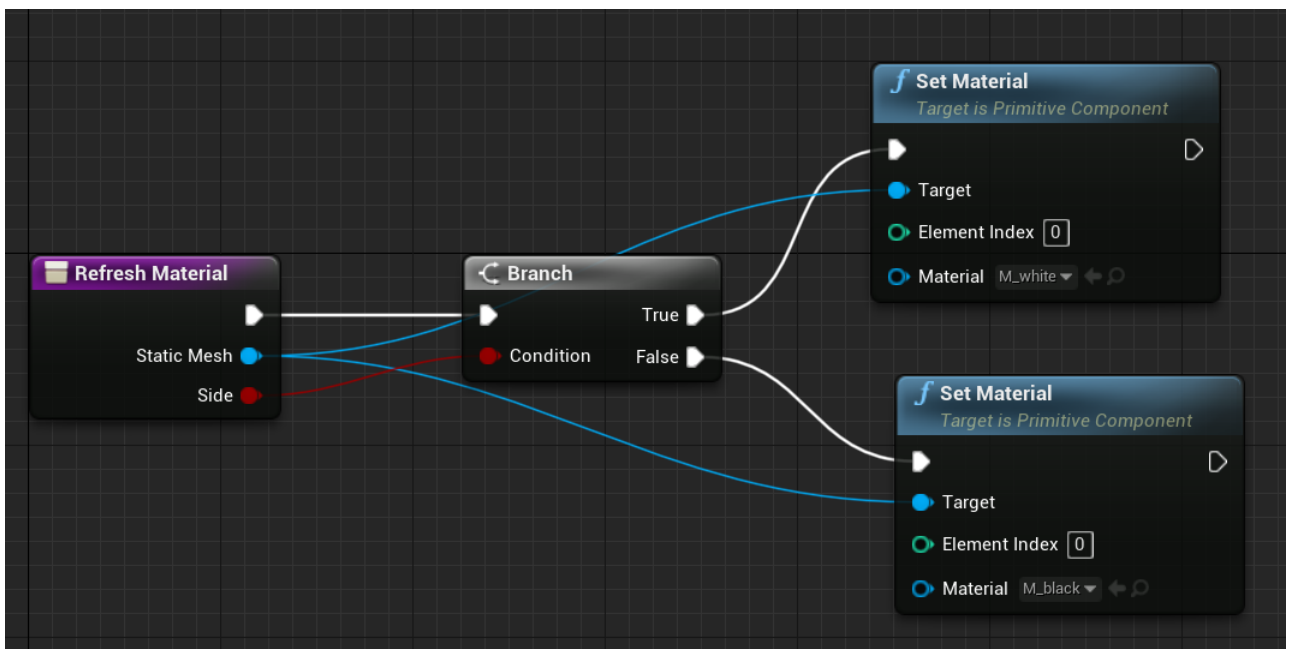


Рисунок 3.3 – Функція Refresh Material

Відновлює матеріал на шашці після ходу, натисканні на другу шашку, або в любую іншу точку, яка не являється правильним ходом.

Ось так виглядає вже граф подій дідечки. Він більш складний та оброблює більше різноманітної інформації

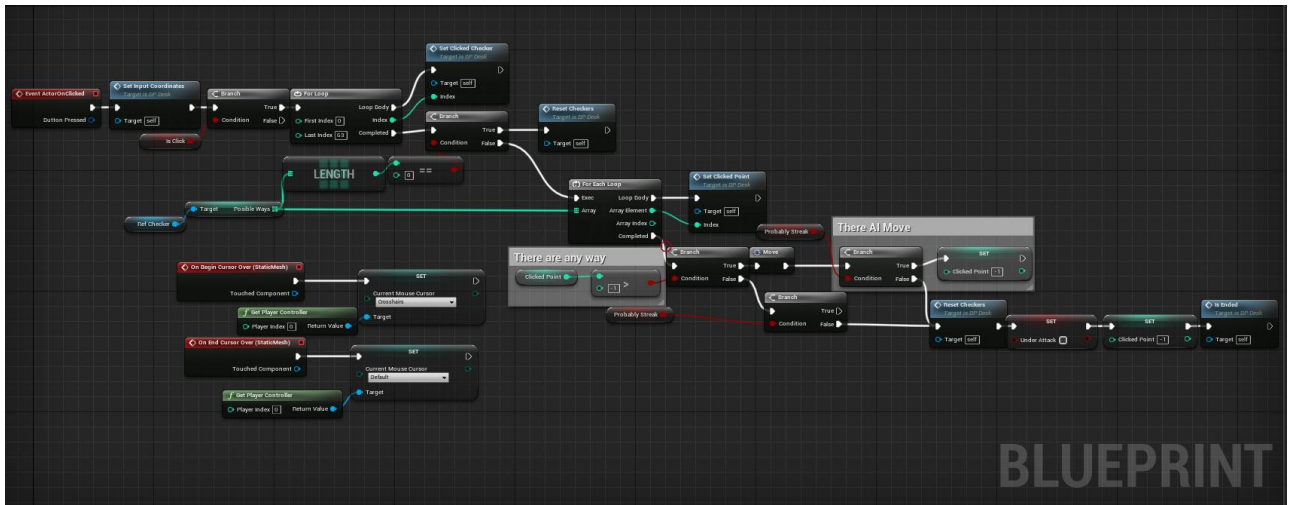


Рисунок 3.4 – Граф подій класу Desk

Його логіка полягає в тому, після натиску на шашку ми отримуємо деякі точки(шляхи) для ходу, а потім натискаємо на місце на дідечці, ось тут і починається її робота, перевірка клітинок, ходів, та інших змінних.

Спочатку граф перетворює вхідний клік на координати дідечки, далі в своєму масиві шашок знаходить саме натиснуту шашку, та зберігає на неї ссилку, наступний крок - якщо шляхів для шашки не існує, дошка оновлює всі дії до стартових(до кліка по шашці), якщо існують, то перетворює всі існуючі шляхи в клітинки на полі, та перевіряє чи натиснув користувач на одну із них, якщо ні, то поле знову оновлюється, а в іншому випадку шашка рухається в дану клітинку з урахунком зміни всіх характеристик, з урахуванням ходу взяття, декількох взяття, перетворення в дамки, та ходів дамк, після чого поле знову оновлює всі дані для продовження гри далі[8].

Повна реалізація класу Desk:

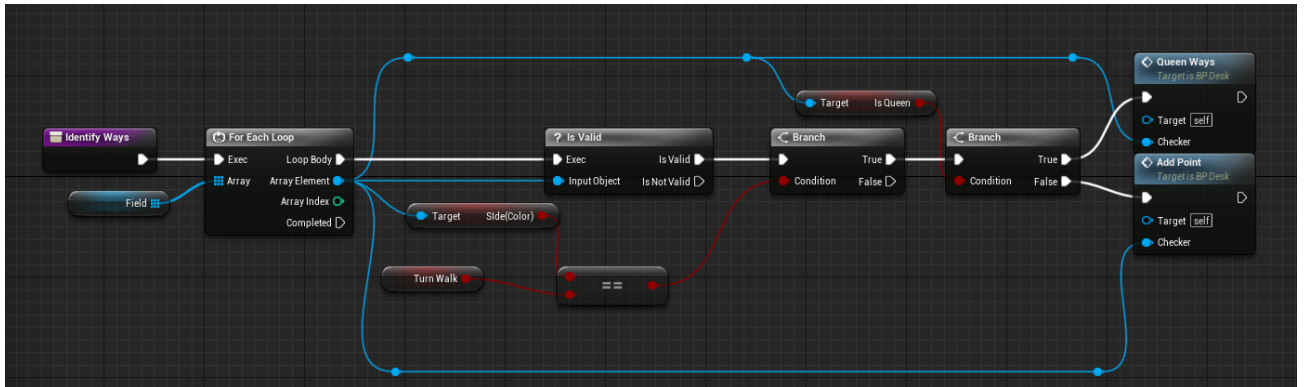


Рисунок 3.5 – Функція Identify Ways

Функція вираховує всі можливі шляхи для шашок даного ходу. Спочатку відбувається перевірка чи існує такий об'єкт (нода Is Valid) далі перевірка чий зараз хід, а далі використовується 2 функції Add Point та Queen Ways, перша вираховує шляхи для звичайних шашок, а інша для королев.

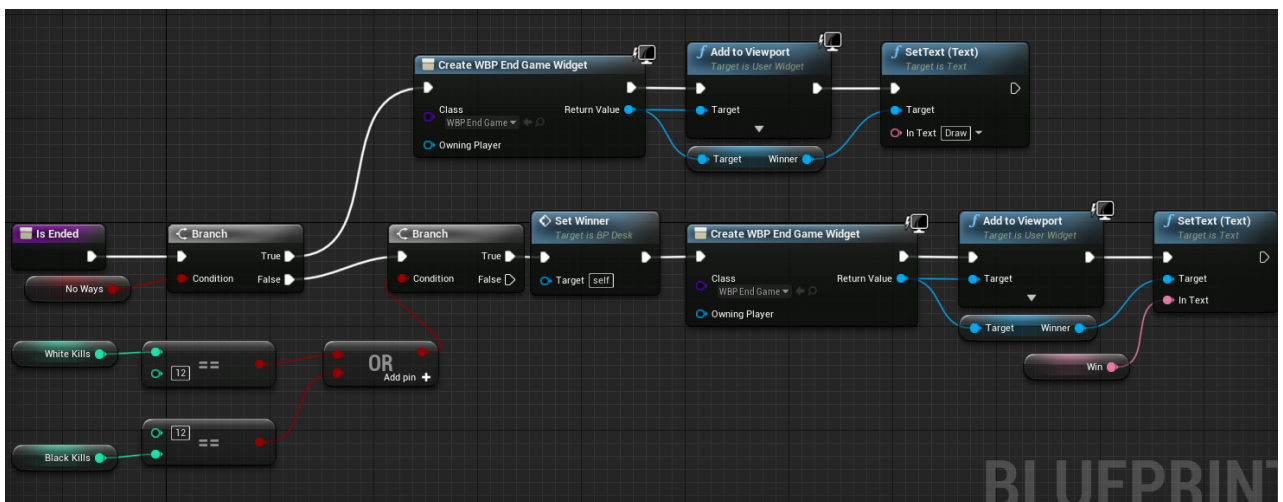


Рисунок 3.6 – Функція Is Ended

Перевіряє чи закінчилася гра в момент вивозу функції. Якщо змінна No ways істина, то закінчує гру нічиєю, в іншому випадку, закінчує перемогою

однієї із команд, у якої кількість взятих дорівнює 12. Функція створює віджет End Game.

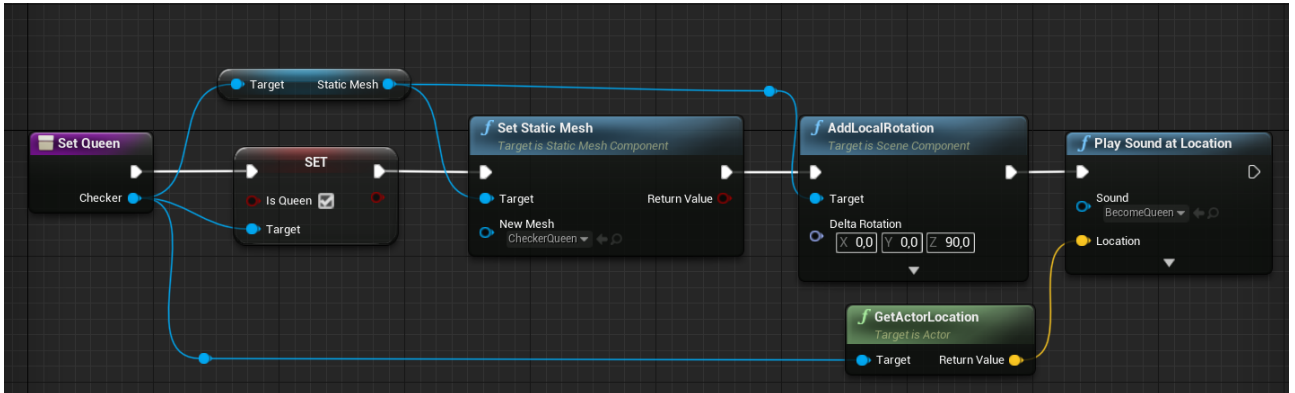


Рисунок 3.7 – Функція Set Queen

Ця функція перетворює звичайну шапку на королівську, програв характерний звук, змінює її модель, та булеву змінну.

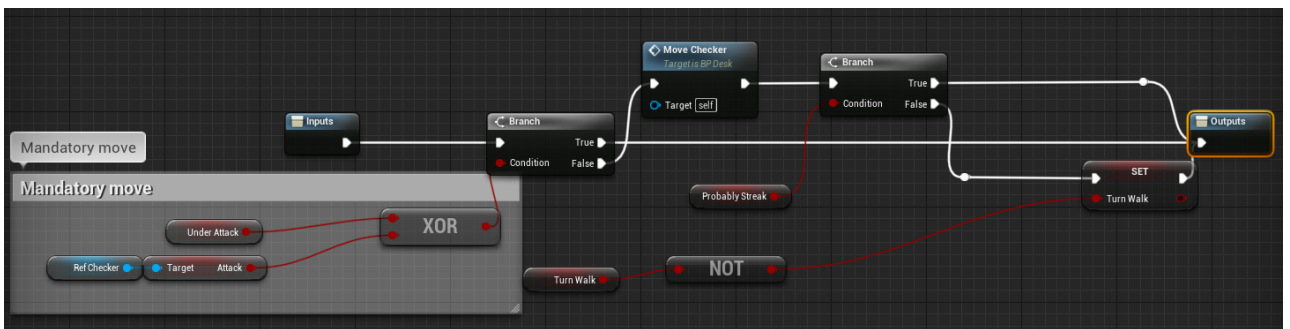


Рисунок 3.8 – Макрос Move

Виконує хід, забезпечує переміщення шапки на нову клітинку та оброблює всі данні зв'язані з цими діями.

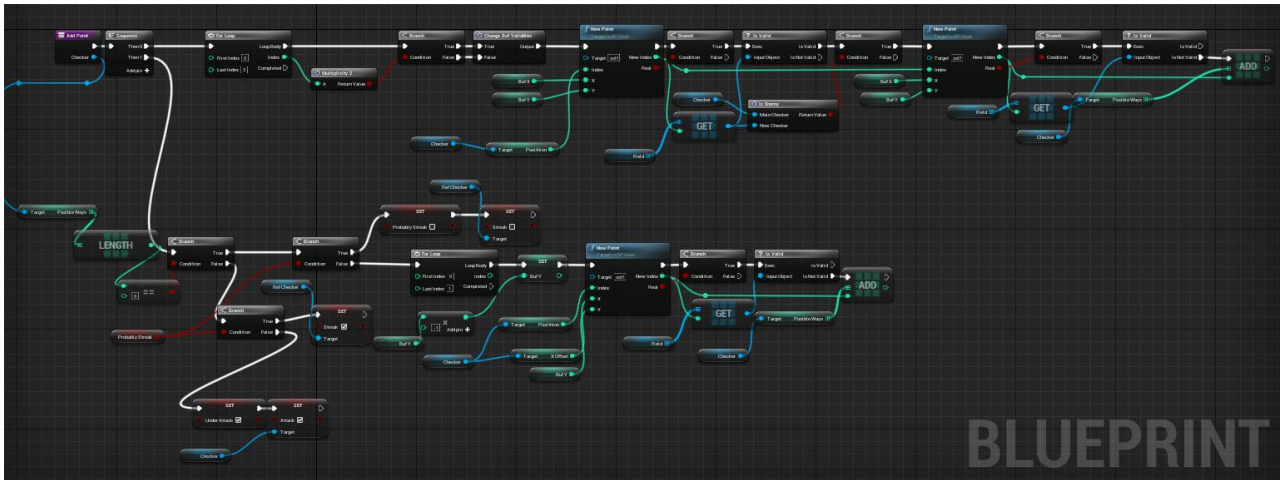


Рисунок 3.9 – Функція Add Point

Шукає можливі клітинки для ходів, для конкретної шашки. Функція перетворює значення клітинок та визначає сторону, в яку ходить шашка, принцип роботи, що програма перевіряє всі можливі ходи вперед та назад, далі за допомогою функції New Point визначає чи існує така клітинка чи ні, потім перевіряє чи є в цій клітинці шашка, якщо так, то чи ворожа вона чи ні, якщо так, далі йде перевірка чи пуста клітинка по вектору від першої шашки до другої, тобто чи можна її побити чи ні, якщо можна, то ця клітинка додається до масиву(Possible Ways).

Після цього йде ще одна перевірка, чи може шашка зробити декілька взятъ, таким самим чином програма вираховує чи існує щонайменше 1 клітинка для ходу. Змінюються змінні які вз'язані з можливим взяттям декількох фігур, якщо таких не існує, то програма оновлює змінні до звичайного стану[7].

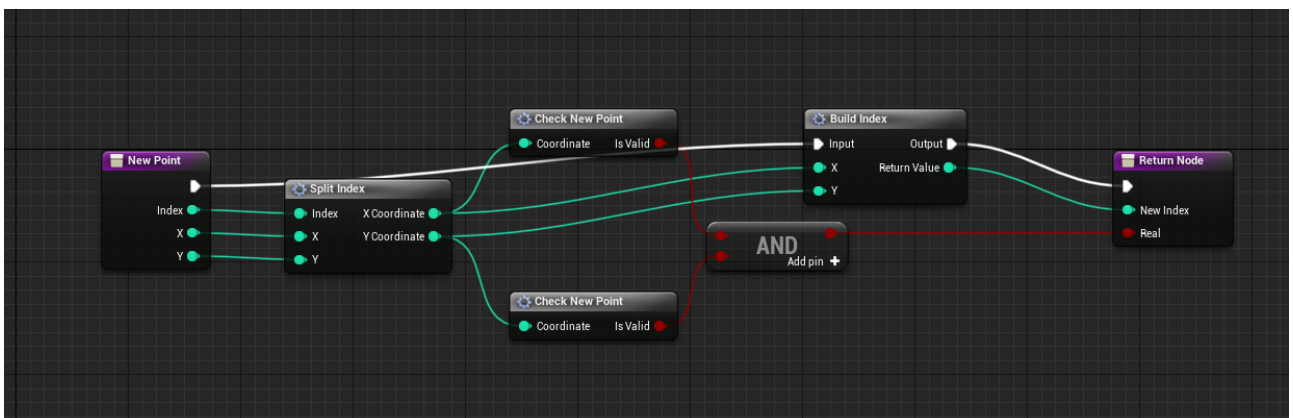


Рисунок 3.10 – Функція New Point



Створює нову точку, та перевіряє чи входить вона в клітинки поля, на вхід приймається індекс X і Y напрямки вектору, куди б'є шашка, на вихід отримуємо індекс та булеву змінну чи правильний цей індекс.

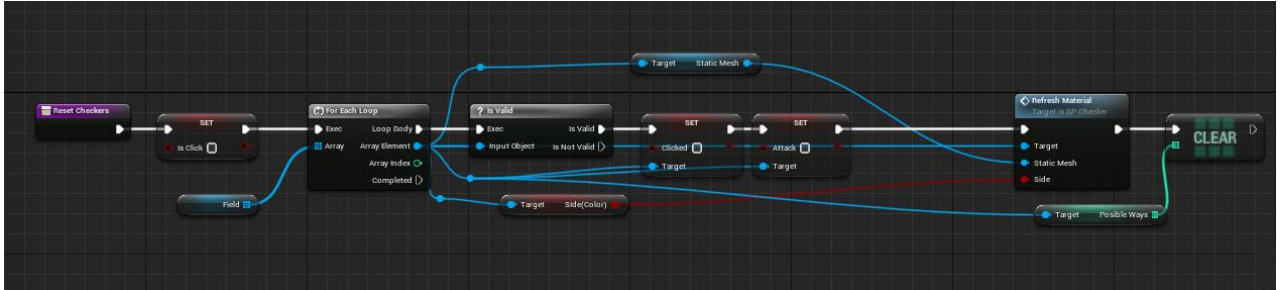


Рисунок 3.11 – Функція Reset Checkers

Оновляє стан шашок на полі. Зкидує значення змінних Attack та Clicked до дефолтного та оновлює матеріал.

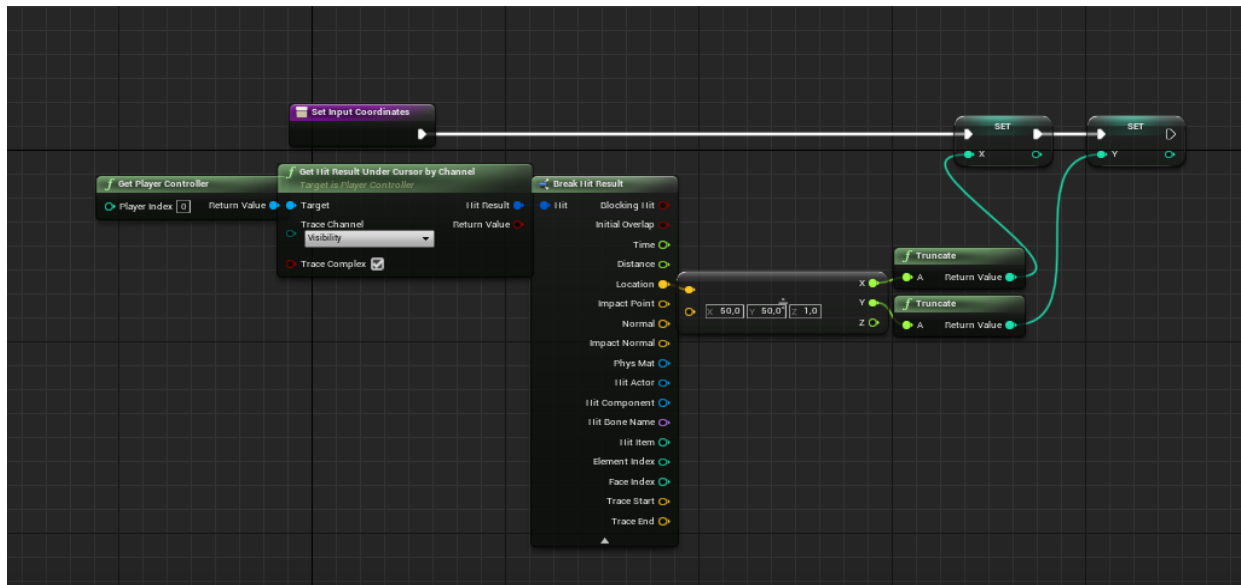


Рисунок 3.12 – Функція Set Input

Перетворює вхідні параметри кліку на значення змінних X і Y.



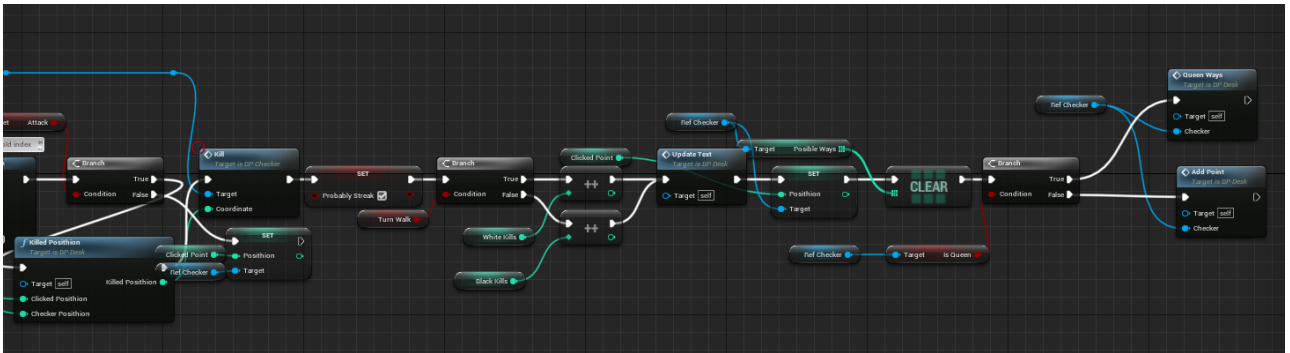


Рисунок 3.16 – Функція Move Checker(2)

Функція яка забезпечує зміну позиції шахи. Спочатку йде перевірка чи це хід комп'ютера, якщо так, то пересумання відбувається автоматично, якщо ні то пересувається в натиснуту клітинку, потім йде перевірка, чи є тепер дана шашка королевою, далі відбувається затирання старого індексу в полі, і зміна на новий(даної шахи), після цього якщо шашка атакувала іншу запускається декілька процедур з кінцевою Kill та знищує фігуру, далі оновлюються змінні рахунків, та запускається функція для підрахування нових можливих ходів[2].

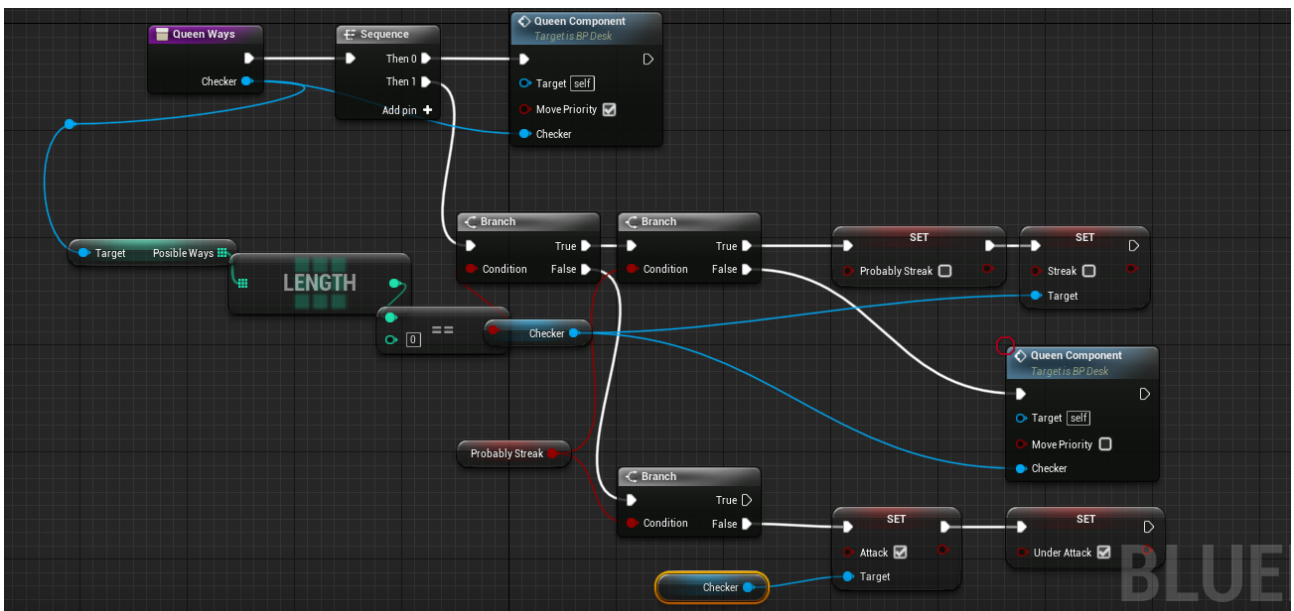


Рисунок 3.17 – Функція Queen Ways

Підраховує ходи, тільки для королевських шашок.

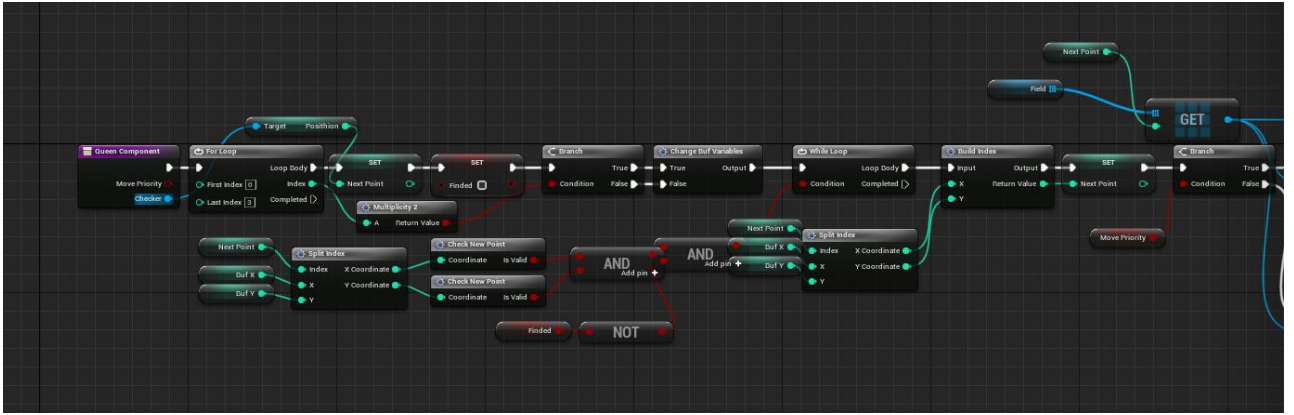


Рисунок 3.18 – Функція Queen Component(1)

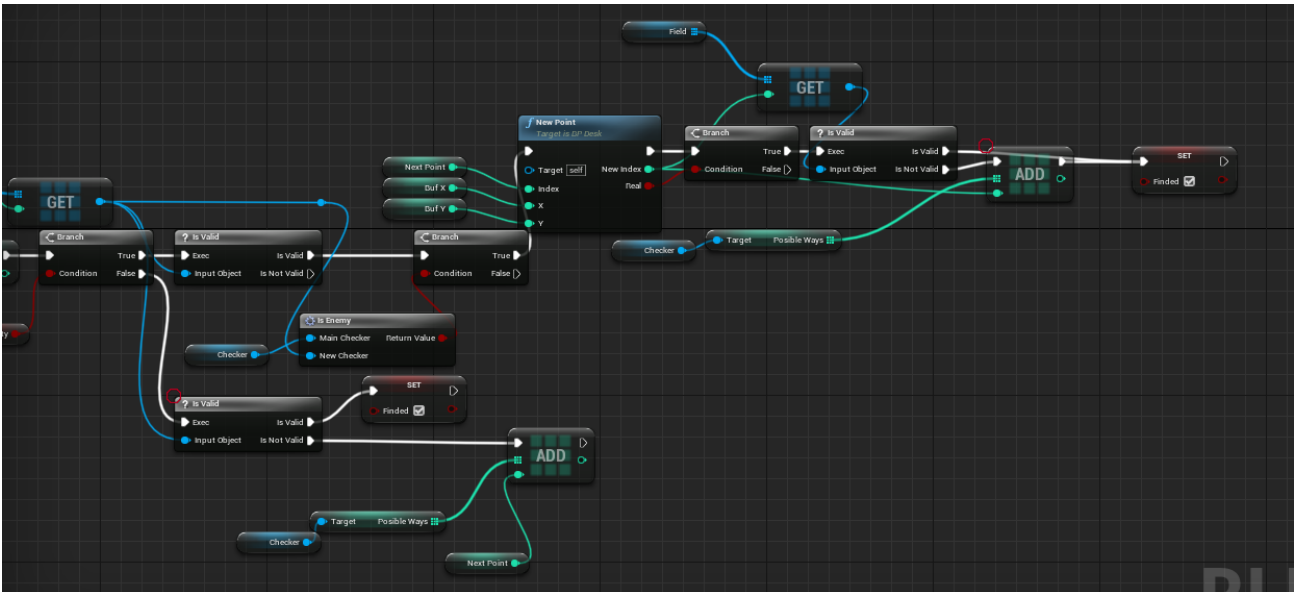


Рисунок 3.19 – Функція Queen Component(2)

Метод який використовується в Queen Ways, для підрахування клітинок, а саме для всіх можливих клітинок по діагоналям.

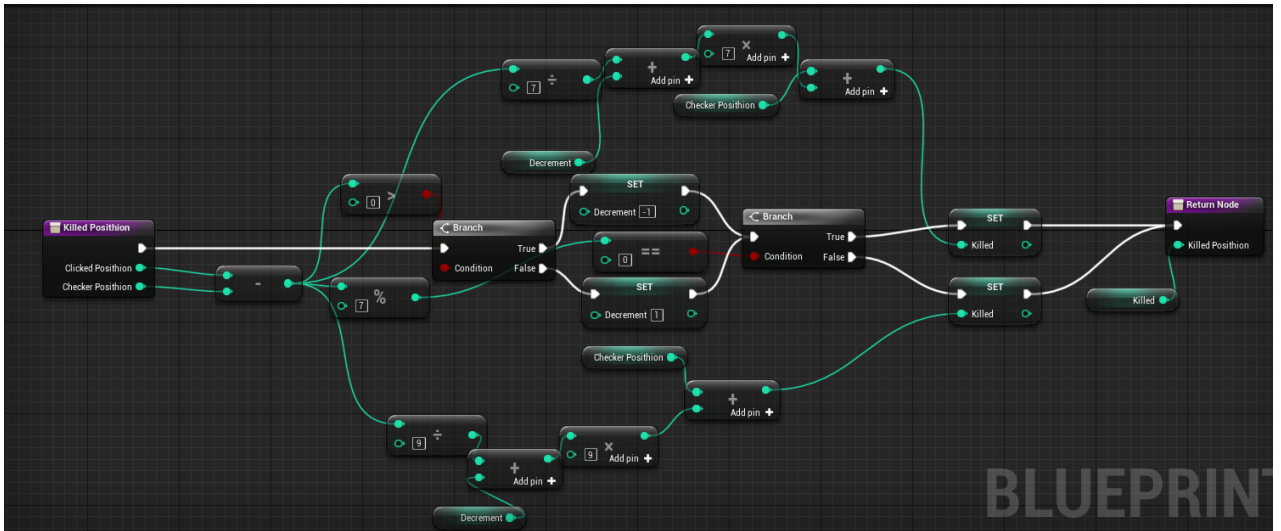


Рисунок 3.20 – Функція Killed Position

Метод підраховує в якому напрямі відбувається взяття, та видає клітинку Killed Position.

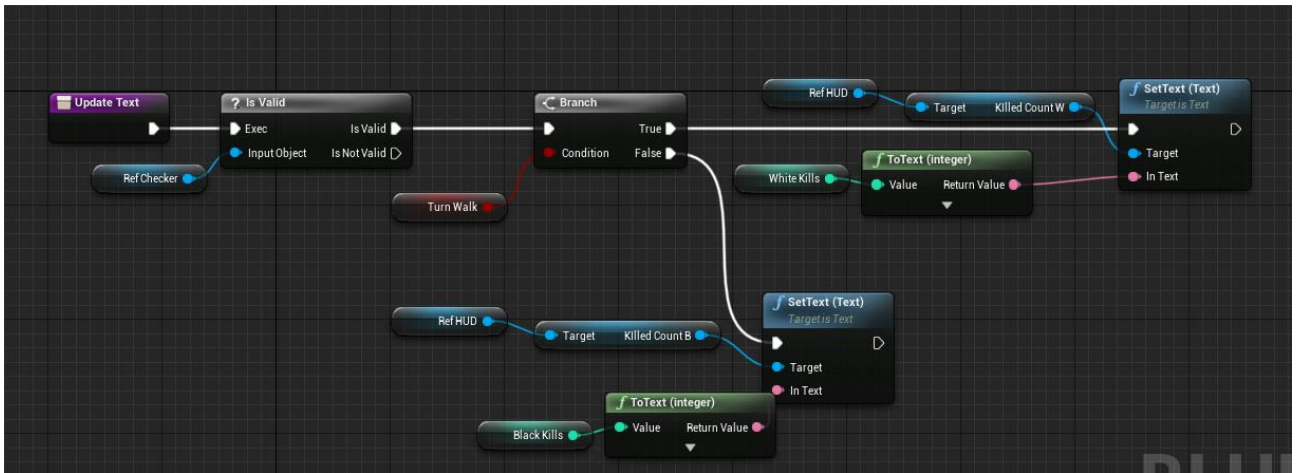


Рисунок 3.21 – Функція Update Text

Функція яка оновлює значення звань у віджеті.

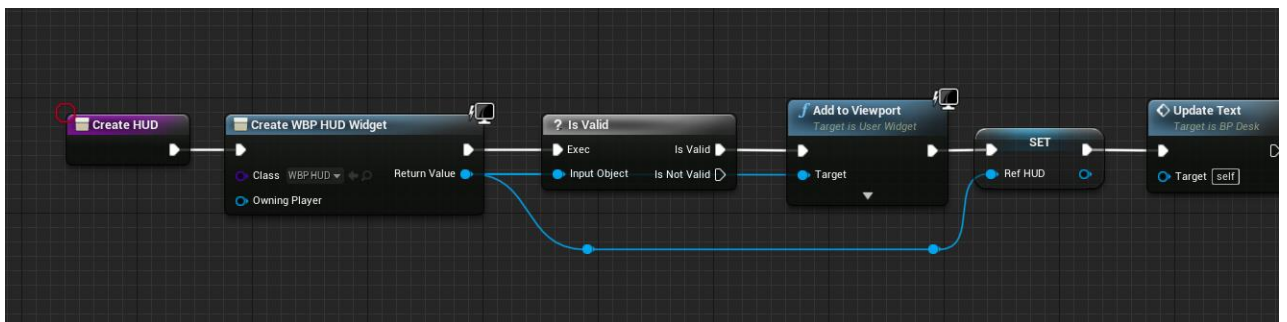


Рисунок 3.22 – Функція Create HUD

Створює віджет, та додає його на екран.

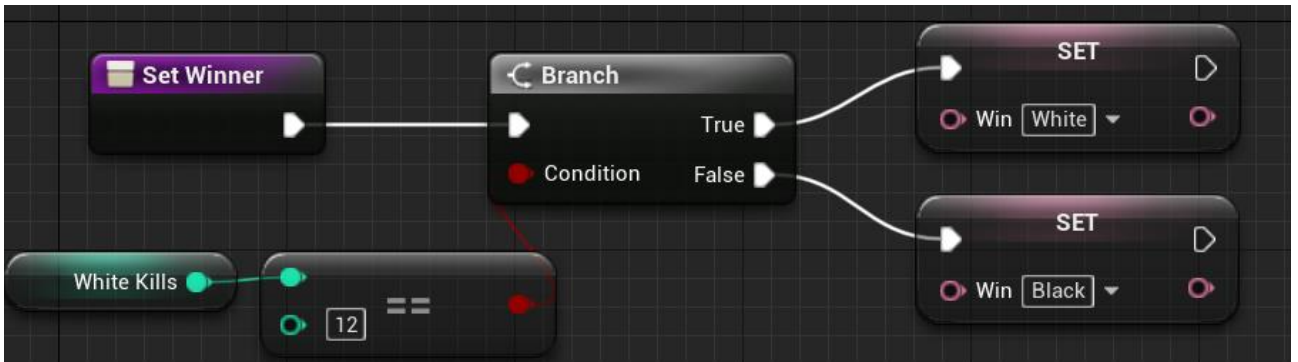


Рисунок 3.23 – Функція Set Winner

В залежності від того, хто першим зробив 12 взятъ, те і записується в змінну переможця, або білі або чорні.

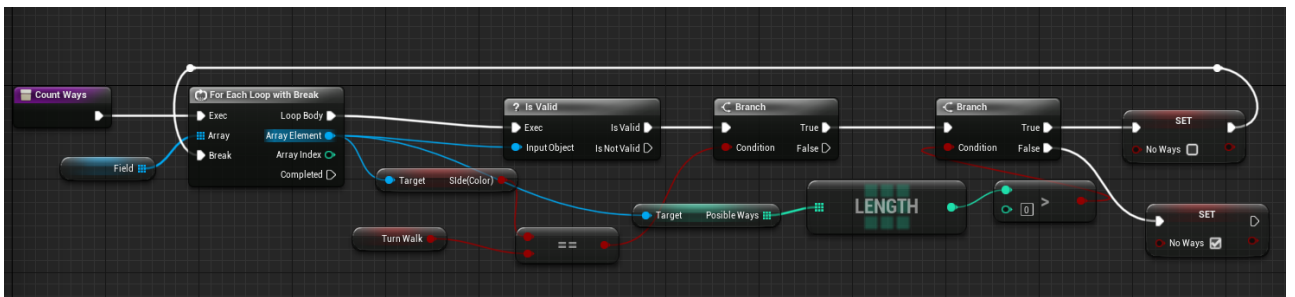


Рисунок 3.24 – Функція Count Ways

Підраховує кількість можливих ходів.

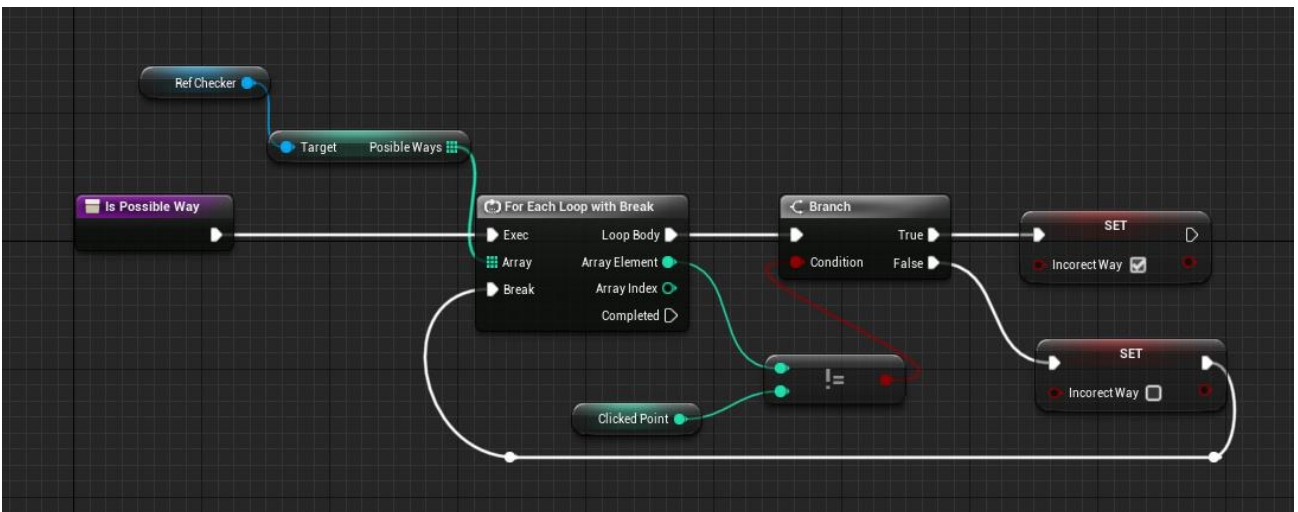


Рисунок 3.25 – Функція Is Possible Way

Підраховує чи натиснута клітинка є можливим шляхом?

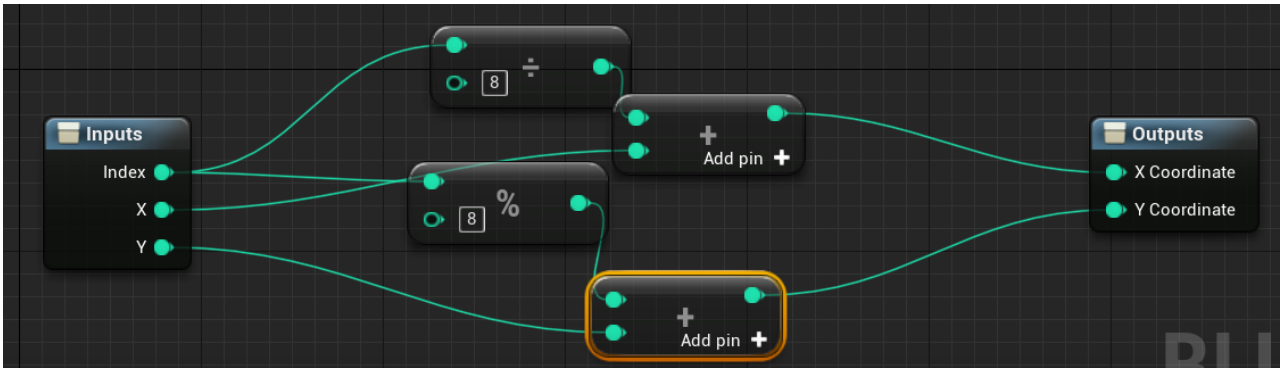


Рисунок 3.26 – Макрос Split index

Перетворює індекс на X та Y

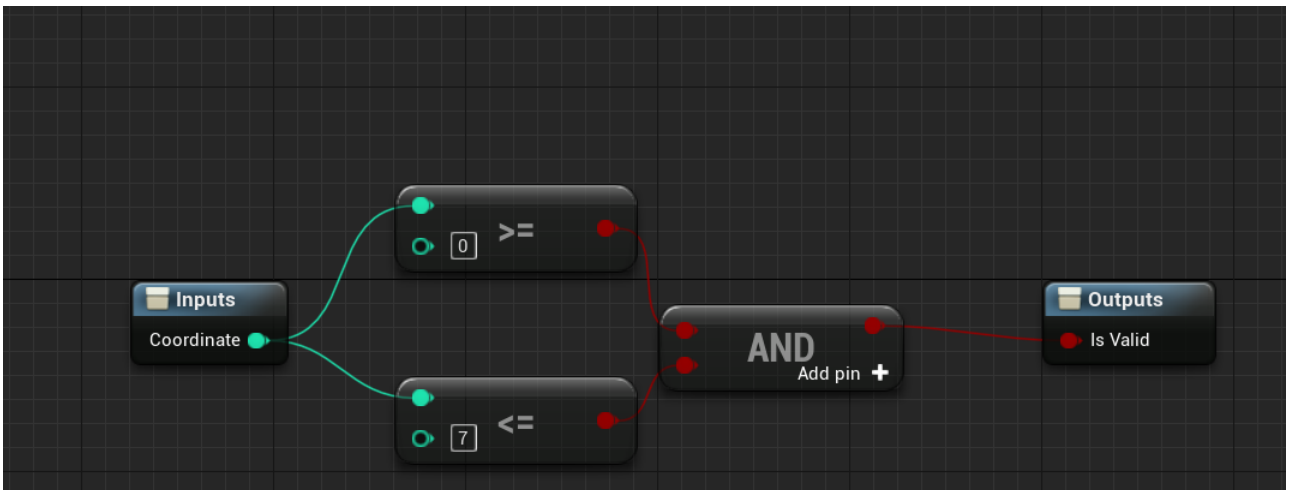


Рисунок 3.27 – Макрос Check New Point

Перевіряє чи це існуюча клітинка.

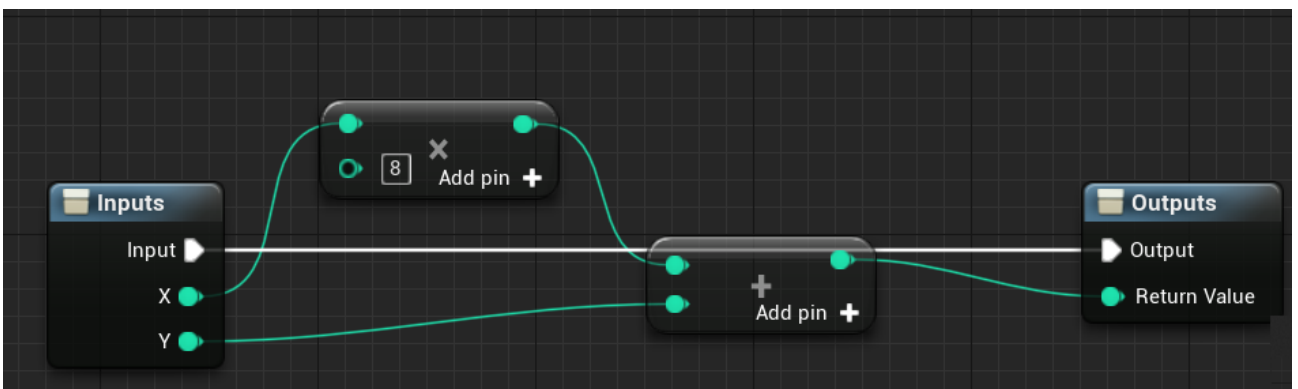


Рисунок 3.28 – Макрос Build Index

Створює з координат індекс.

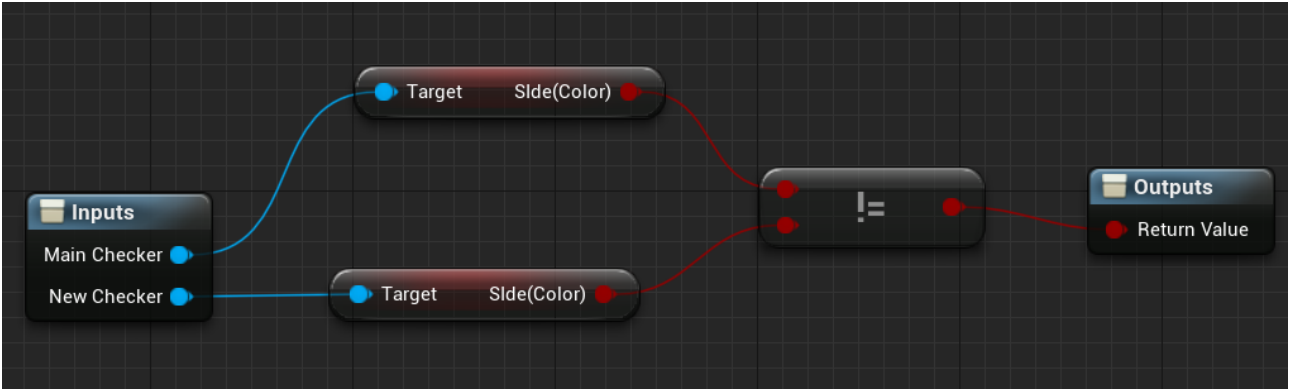


Рисунок 3.29– Макрос Is Enemy

Чи є шапка ворогом до поточної.

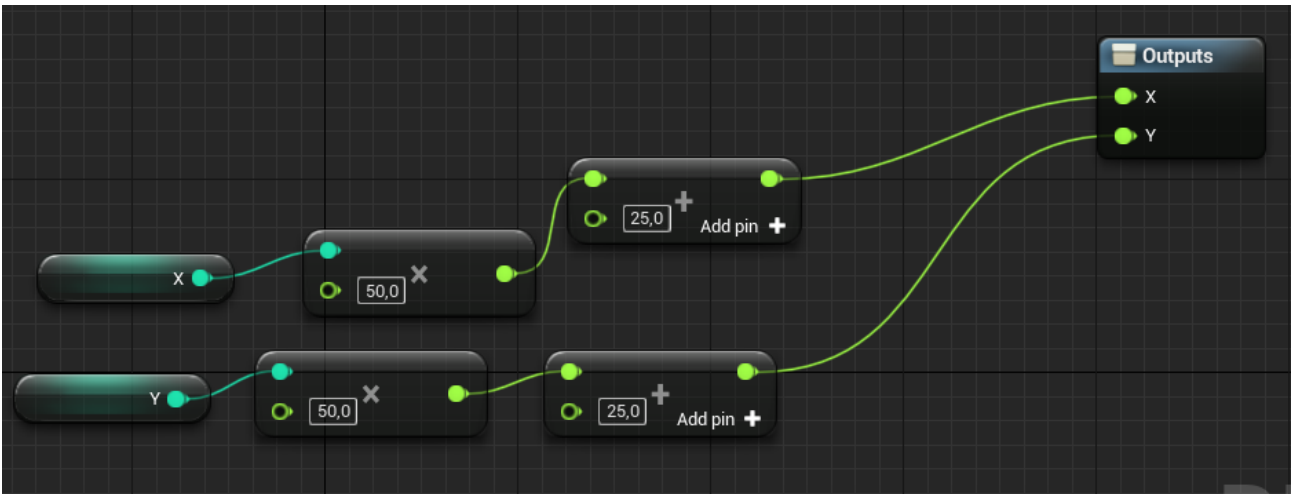
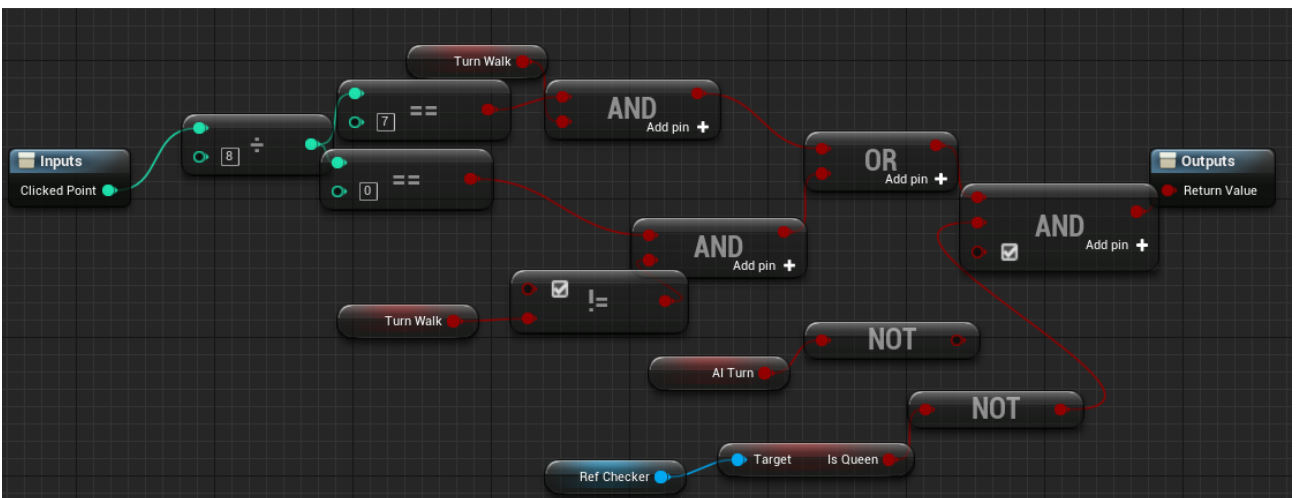


Рисунок 3.30– Макрос Checker Coordinate

Видає координати шапки в світі.





### Рисунок 3.31 – Макрос Become A Queen

Макрос який використовує багато перевірок, щоб видати кінцевий результат, чи стала шашка королевською чи ні?

### 3.2 Створення рівнів та віджетів

Також потрібно створити рівні гри, та графічні інтерфейси для взаємодії в грі.

Створимо два рівні: MainMenu, Game.

Додамо відповідну музику на рівні, логіку переходу на інші рівні, та на рівні Game створимо деяку сцену, де будуть відбуватися події[4]



Рисунок 3.32 – Сцена гри



Рисунок 3.33 – Віджет Main Menu

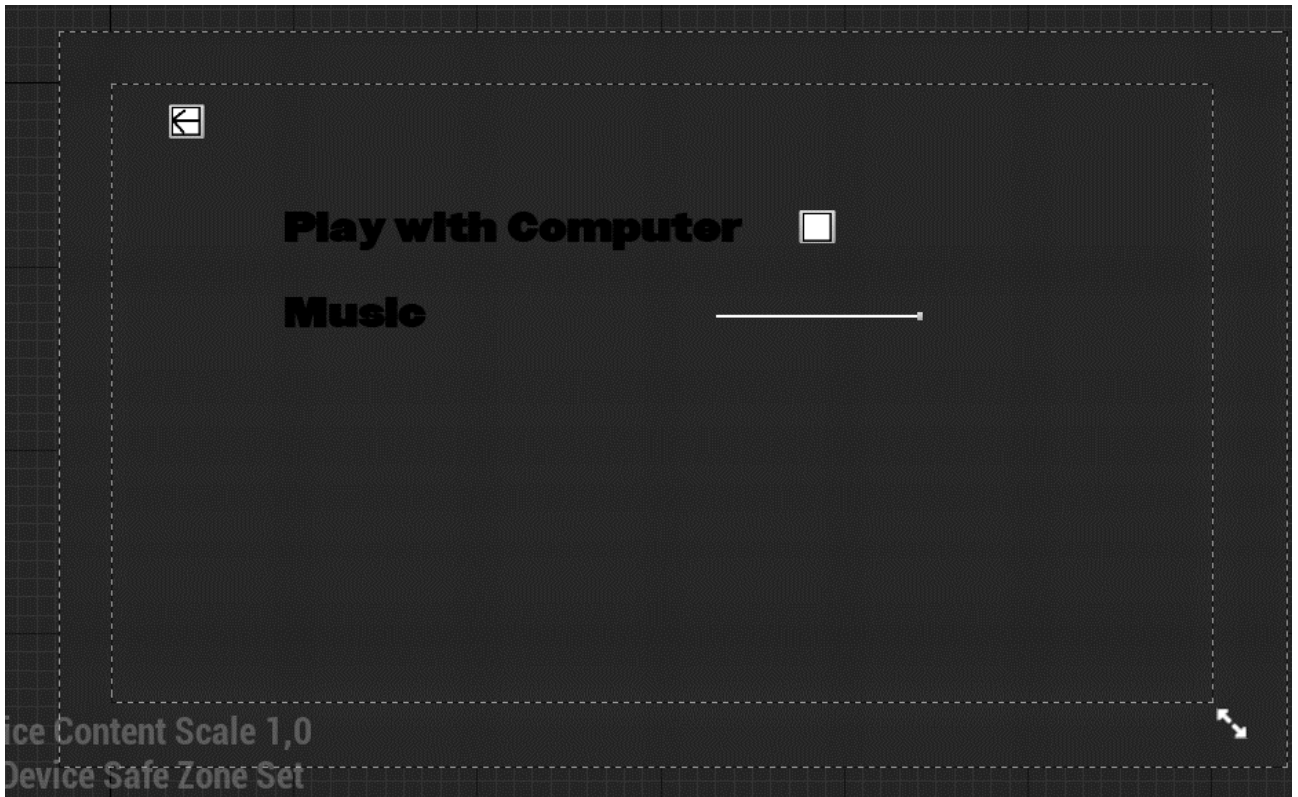


Рисунок 3.34 – Віджет Options



Рисунок 3.35 – Віджет End Game

## ВИСНОВКИ

У цій дипломній роботі була проведена розробка гри в шашки з використанням потужного ігрового двигуна Unreal Engine 4. В процесі реалізації проекту було вивчено функціональні можливості та інструменти, надані Unreal Engine 4, що дозволило створити гру з високоякісною графікою, реалістичною фізикою та захоплюючим геймплеєм.

Розробка гри в шашки на базі Unreal Engine 4 вимагала комплексного підходу та використання різних компонентів двигуна. Було створено моделі шашок та дошки, розроблено правила гри та їх втілення в програмному коді. Гра отримала додаткові функції, такі як взяття шашок, перетворення на дамку та можливість грати з комп'ютером.

Завдяки використанню Unreal Engine 4 в розробці гри в шашки, отримані результати були вражаючими. Гра надає користувачам захоплюючий іммерсивний досвід, забезпечуючи їх візуальними та геймплейними задоволеннями. Інтуїтивний інтерфейс, реалістична анімація та візуальні ефекти додають грі рівень полішеності та привабливості.

Завершуючи, розробка гри в шашки за допомогою Unreal Engine 4 була успішною і вдалою. Цей проект підтверджує потужні можливості та гнучкість Unreal Engine 4 у створенні високоякісних ігрових досвідів. Він слугує відмінним прикладом використання ігрових двигунів у розробці ігор та демонструє важливість їх використання для досягнення високих стандартів у галузі ігрового розроблення.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Unreal Engine Documentation [Електронний ресурс] // Режим доступу до ресурсу: <https://docs.unrealengine.com/4.27/en-US/>
2. Блум, М. Unreal Engine 4 Game Development Essentials. Packt Publishing, 2016. <https://www.packtpub.com/product/unreal-engine-4-game-development-essentials/9781784391966>
3. Хо, К. Mastering Unreal Engine 4.X. Packt Publishing, 2019. <https://www.packtpub.com/product/mastering-unreal-engine-4x/9781785883569>
4. Флемінг, Д. Unreal Engine 4 Scripting with C++ Cookbook. Packt Publishing, 2019. <https://www.packtpub.com/product/unreal-engine-4x-scripting-with-c-cookbook-second-edition/9781789809503>
5. Вакконе, Е. Unreal Engine 4.X By Example. Packt Publishing, 2019. <https://www.packtpub.com/product/unreal-engine-4x-by-example/9781785885532>
6. Веб-енциклопедія Вікіпедія, Unreal Engine [https://en.wikipedia.org/wiki/Unreal\\_Engine](https://en.wikipedia.org/wiki/Unreal_Engine)
7. YouTube канал Unreal Engine <https://www.youtube.com/user/UnrealDevelopmentKit>
8. YouTube канал Virtus Learning Hub <https://www.youtube.com/channel/UCzeYJAUGSE-mqzKtit7m9g>
9. Інформаційний форум, спільнота Epic Developer Community Forums <https://forums.unrealengine.com/categories?tag=unreal-engine>
10. Інформаційний форум, спільнота Reddit <https://www.reddit.com/r/unrealengine/>