

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ
 Фізико-математичний факультет
 Кафедра інформатики та прикладної математики

«Допущено до захисту»

В.о завідувача кафедри

_____ Семеріков С.О.
 _____ 2025 р.

Реєстраційний № _____
 «___» _____ 2025 р.

**ТРЕНУВАЛЬНЕ СЕРЕДОВИЩЕ ДЛЯ ПІДГОТОВКИ ДО ЗМАГАНЬ ІЗ
 КІБЕРСПОРТУ**

Кваліфікаційна робота студента
 групи І-21

ступінь вищої освіти «бакалавр»
 спеціальності

014 Середня освіта (Інформатика)

Єрофєєва Олексія Олександровича

Керівник:

доцент кафедри інформатики та
 прикладної математики

Мерзликін Павло Володимирович

Оцінка:

Національна шкала _____

Шкала ECTS _____ кількість балів _____

Члени комісії

ЗАПЕВНЕННЯ

Я, Єрофеев Олексій Олександрович, розумію і підтримую політику Криворізького державного педагогічного університету з академічної доброчесності. Запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Я не надавав і не одержував недозволену допомогу під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають покликання на відповідне джерело.

Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Криворізького державного педагогічного університету ознайомлений. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.



ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1	
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Аналіз проблеми тренування точності й реакції в кіберспорті.....	7
1.2 Огляд наявних програмних рішень для тренування точності й реакції.....	9
1.3 Постановка задачі.....	25
Висновки до розділу 1.....	29
РОЗДІЛ 2.....	31
РОЗРОБКА ТРЕНАЖЕРА ТОЧНОСТІ НА UNITY.....	31
2.1. Реалізація тренажера влучності.....	31
2.2. Тестування, налагодження та результати.....	41
Висновки до розділу 2.....	43
ВИСНОВКИ.....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	47
ДОДАТКИ.....	49

ВСТУП

Актуальність теми зумовлена стрімким розвитком кіберспорту, зокрема змагань у жанрі **First-Person Shooter (FPS)** [5]. Сучасні професійні гравці витрачають сотні годин на відточування базових навичок, серед яких ключове місце займають швидкість реакції та точність прицільної стрільби. Саме ці показники часто вирішують результат дуелі у шутерах: навіть частки секунди затримки в реакції чи невлучний постріл можуть стати фатальними на найвищому рівні змагань. Середній час реакції людини становить близько 250 мс, тоді як елітні кіберспортсмени демонструють реакцію на рівні 100–120 мс [6]. Таким чином, існує об'єктивна потреба в цілеспрямованому тренуванні, що дає змогу гравцям наблизитися до меж власних можливостей. Крім того, на відміну від традиційних видів спорту, в кіберспорті є можливість використати спеціалізовані програмні засоби для тренування окремих навичок у віртуальному середовищі. Багато професійних гравців активно застосовують програмні тренажери для прицільної стрільби (*aim trainers*), які пропонують спеціальні сценарії для розвитку реакції та точності. Популярність таких інструментів надзвичайно висока – наприклад, платформу *Aim Labs* використовують понад 40 мільйонів гравців різного рівня [1]. Це підтверджує актуальність розробки ефективних тренувальних середовищ для підготовки до кіберспортивних змагань.

Мета дослідження: створити власне програмне тренувальне середовище на ігровому рушії Unity[10] для підготовки гравців FPS до змагань, орієнтоване на розвиток навичок швидкості реакції та точності стрільби, з інтегрованими функціями збору статистичних даних і аналітики результатів тренувань.

Об'єкт дослідження: процес тренування точності прицілювання та швидкості реакції у гравців FPS-ігор.

Предмет дослідження: методи і програмні засоби підвищення ефективності тренувань стрілецьких навичок (точності й реакції) у кіберспорті, зокрема спеціалізовані програмні середовища для їх розвитку.

Для досягнення поставленої мети необхідно вирішити такі **завдання дослідження:**

1. Проаналізувати вплив точності прицілювання та швидкості реакції на результати змагань в кіберспорті.
2. Оглянути й порівняти наявні програми для тренування цих навичок.
3. Визначити функціональні вимоги до власного тренувального середовища та сформулювати задачу розробки.
4. Спроектувати та реалізувати середовище на рушії Unity з інтерфейсом користувача та системою збору статистики.
5. Провести тестування та оцінити ефективність розробленого рішення в практичних тренуваннях.

Методи дослідження. Для виконання поставлених завдань використовувалися методи аналізу і синтезу науково-технічної літератури та інтернет-джерел з тематики кіберспортивних тренувань; порівняльний аналіз наявних програмно-апаратних рішень для тренування точності й реакції; методи програмної інженерії для проєктування архітектури тренувального середовища; безпосередньо методи об'єктно-орієнтованого програмування (C# в середовищі Unity) для реалізації програмного продукту; емпіричне тестування та аналіз зібраних статистичних даних для оцінки результатів роботи середовища.

Структура роботи. Робота складається зі вступу, двох розділів, висновків, списку використаних джерел та додатків. У вступі обґрунтовано актуальність теми, сформульовано мету, об'єкт, предмет і методи дослідження, наведено структуру роботи. У першому виконано аналіз предметної області:

розглянуто наявне програмне забезпечення для тренування точності та проведено порівняння ігрових рушіїв з метою вибору оптимального інструментарію для реалізації прототипу. Другий розділ присвячений проєктуванню власного тренувального середовища: визначено вимоги, архітектуру та алгоритми роботи прототипу. Також у цьому розділі описано реалізацію прототипу на рушії Unity, а також результати експериментальної перевірки його функціональності та ефективності. В кінці сформульовано висновки, що підбивають підсумки виконаної роботи та окреслюють напрямки подальшого вдосконалення системи. Робота містить 17 рисунків та 1 таблицю. Загальний обсяг роботи – 60 сторінок.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз проблеми тренування точності й реакції в кіберспорті

Кіберспорт високого рівня пред'являє жорсткі вимоги до індивідуальної майстерності гравців. У жанрі FPS двома фундаментальними показниками результативності є **швидкість реакції** гравця та **точність прицільної стрільби** [6]. Швидкість реакції визначає, наскільки швидко гравець здатен помітити раптову появу цілі та розпочати дію (постріл або ухилення). Від неї залежить успіх у ситуаціях «першого пострілу», коли виграє той, хто раніше зреагує на противника. Як зазначалося вище, середньостатистична реакція людини становить близько 0,25 с, тоді як провідні кіберспортсмени можуть реагувати за 0,1–0,15 с. Хоча базові межі реакції обумовлені фізіологією, тренування здатне незначно її поліпшити, а головне – навчити гравця швидше розпізнавати ціль і приймати рішення в ігровій ситуації. Регулярне виконання спеціальних вправ сприяє зміцненню нейронних зв'язків, що відповідають за швидкі рухові реакції, формує так звану «м'язову пам'ять» і автоматизм дій [6].

Точність прицільної стрільби – це вміння наводити приціл на ціль швидко й з мінімальною похибкою. В контексті FPS ігор точність вимірюється відсотком влучень у ціль та здатністю вражати **ключові зони** (наприклад, область голови для максимального урону). Висока точність особливо важлива в змагальних матчах, де рахунок іде на частки секунди: якщо гравець влучає в ціль швидше за опонента, це дає команді перевагу. Точність також тісно пов'язана з контролем віддачі зброї, умінням стріляти чергами, швидко наводитися на нові цілі (перекладати приціл). Всі ці піднавички потребують окремого тренування [6].

У реальних матчах гравець не завжди має можливість постійно практикувати стрільбу – значна частина ігрового часу йде на пересування мапою, пошук ресурсів, стратегічні дії тощо. Отже, без спеціальних методик розвиток реакції та точності відбувається стихійно і може займати більше часу. Аналогічно до традиційного спорту, де спортсмени виконують ізольовані вправи для розвитку конкретних навичок (наприклад, спринти для реакції старту, стрільба по мішенях для біатлоністів), кіберспортсмени запровадили у свою практику спеціалізовані тренування поза рамками змагальних матчів. Для тренування точності та швидкості реакції використовуються як спеціальні режими всередині самих ігор (тренувальні полігони, «deathmatch» для розминки, користувацькі карти з ботами), так і **сторонні програмні засоби [7]**, створені саме для відпрацювання стрілецьких навичок. Останні, як правило, надають широкі можливості для імітації різноманітних ситуацій та збору статистики, недоступні у стандартному ігровому клієнті.

Ефективне тренування точності й реакції зазвичай включає декілька типів вправ. По-перше, це вправи на **швидкі рефлекси**: наприклад, раптова поява мішені в випадковому місці екрану, що змушує гравця миттєво зреагувати та навести приціл. Подібні завдання розвивають час реакції та точність першого пострілу. По-друге, **тренування точності наведення та флік-шотів** – вправи, де гравець мусить швидко переводити приціл між кількома цілями, що з'являються послідовно або одночасно. Це розвиває координацію та швидкість **«фліків»** – різких прицільних рухів на ціль, і покращує влучність при швидкому наведенні. По-третє, **тренування трекінгу** (слідкування за рухомою ціллю) – завдання, де мішені плавно рухаються, а гравець повинен утримувати приціл на них, стріляючи безперервно. Такі вправи потрібні для ігор, де важливо завдавати шкоди по рухомих опонентах, наприклад, у дуелях на ближній відстані. Нарешті, деякі тренування поєднують **рух гравця і стрільбу**: від гравця вимагається ухилятися від

ворожих пострілів чи пересуватися смугою перешкод, одночасно вражаючи мішені (імітація бойової ситуації). У спеціалізованих тренажерах для стрільби зазвичай доступні різні режими, що охоплюють згадані типи вправ [13].

Наприклад, платформа Aimtastic [1] пропонує декілька тренувальних "кімнат", серед яких "Reflex" (реакція та точність), "Tracking" (відстеження цілей), "Assault course" (комплекс на рух і стрільбу) та інші. Для удосконалення окремих елементів стрільби, ефективним підходом є використання структурованих тренувань з повторюваними ситуаціями, що сприяє поступовому підвищенню точності та швидкості.

В цілому, проблема тренування точності та реакції в кіберспорті полягає у необхідності розробити ефективні методики, спрямовані на ізоляцію та розвиток цих навичок. Враховуючи ключову роль цих навичок у досягненні успіху в змаганнях, професійні гравці потребують інструментів, що дозволяють їм цілеспрямовано та ефективно вдосконалювати прицільну стрільбу поза звичайних матчів. Це призвело до появи різних програмних рішень – тренувальних середовищ для FPS, спрямованих на удосконалення реакції та точності. Далі ми розглянемо найпопулярніші з них, дослідимо їх функціонал, переваги та недоліки.

1.2 Огляд наявних програмних рішень для тренування точності й реакції

У сучасних умовах кіберспортсмени мають доступ до широкого спектру програм, спеціально розроблених для тренування навичок прицільної стрільби. Серед найпопулярніших рішень варто відзначити такі як Aim Labs [1], KovaaK's FPS Aim Trainer [2], Aimtastic [3], 3D Aim Trainer [4] та інші. Кожен з цих продуктів пропонує індивідуальний підхід до підвищення точності та швидкості

реакції гравців, має унікальні функції та підтримує різні платформи. Розглянемо та порівняємо ці програми за такими важливими критеріями: функціональність, підтримувані платформи, можливості аналізу даних, налаштування та сумісність з ігровими рушіями (або конкретними іграми).



Рис. 1.1. Aim Labs

Aim Labs [1]. *Aim Labs* – один із найпопулярніших на сьогодні безкоштовних тренажерів для стрільби від першої особи. Він здобув широку аудиторію користувачів від новачків до професіоналів (понад 40 млн гравців) завдяки поєднанню доступності та потужних аналітичних функцій. Aim Lab надає гравцеві **необмежену кількість тренувальних сценаріїв** різних типів. Станом на 2024 рік доступно близько 12 тисяч вправ і задач, що охоплюють усі аспекти стрільби – реакція, швидкість наведення, трекінг рухомих цілей, точність тощо. Сценарії в Aim Labs поділені за категоріями, і інтерфейс

програми допомагає підбирати вправи під конкретні потреби гравця (напрямок тренування). Важливою перевагою Aim Labs є **офіційна інтеграція з низкою популярних ігор**. Розробники співпрацювали з такими компаніями, як Riot Games та Ubisoft, щоб створити тренувальні режими, що імітують карти та ситуації з *Valorant* і *Rainbow Six Siege*. Наприклад, у Aim Labs можна тренувати встановлення прицілу на кутах карти, подібних до карт Valorant, або відпрацьовувати сценарії, натхненні конкретними ігровими моментами турнірів. Хоча надто специфічні сценарії мають обмежений вплив на загальні навички, така інтеграція підтверджує якість тренажера і робить його привабливим для фанатів певних ігор.

Aim Labs також вирізняється розвинутою системою **збору статистики та аналітики**. Після виконання кожної вправи користувач отримує детальний зворотний зв'язок: відображаються показники точності, швидкості знищення цілей, час реакції, стабільність трекінгу тощо. Програма будує багатокомпонентний графік (шестикутник), який наочно показує сильні та слабкі сторони стрільби гравця по ключових параметрах. Передбачено навіть розділ **Insights** із порадами: на основі зібраної статистики Aim Lab може вказати, що, скажімо, гравець гірше влучає по цілях у нижній частині екрана, і запропонує перевірити положення миші чи вдосконалити коврик для миші. Такий рівень персоналізованої аналітики став можливим завдяки використанню елементів штучного інтелекту та великої бази даних результатів інших гравців. Отримані дані зберігаються в профілі користувача, доступні історичні графіки покращення навичок, а також глобальні рейтинги. Таким чином, Aim Labs не лише дозволяє тренуватися, а й виступає як інструмент **діагностики навичок** кіберспортсмена, допомагаючи цілеспрямовано працювати над слабкими аспектами стрільби.

З точки зору **платформ**, Aim Labs спочатку випущений для ПК (Windows, через Steam), причому він має повноцінну підтримку як клавіатури-мишки, так і

геймпадів (для тренування консольних гравців). Наразі Aim Lab є безкоштовним і знаходиться у стадії активного розвитку (станом на 2023 р. проєкт вийшов із раннього доступу). Окрім версії для Windows, існують мобільні додатки Aim Labs для платформ iOS та Android, що дозволяють тренувати базові навички на смартфоні. Планується розширення на консолі наступного покоління, про що свідчить підтримка геймпадів і партнерство з консольними кіберспортивними дисциплінами. Що стосується **гнучкості налаштувань**, Aim Labs надає широкі можливості підлаштувати тренування під себе: гравець може обирати рівень складності та швидкість появи цілей (деякі завдання мають адаптивне налаштування, що підлаштовується під поточний рівень умінь), змінювати розмір та колір мішеней, налаштовувати приціл (форма, колір прицільної точки) тощо. Важливо, що Aim Labs підтримує налаштування чутливості миші та поля зору (FOV) під різні ігри – тобто, гравець може ввести параметри зі своєї улюбленої гри (CS:GO, Valorant, Apex Legends тощо) і тренуватися з тими ж відчуттями сенси, які він має в реальних матчах. Така **сумісність із ігровими рушіями та налаштуваннями ігор** є критичною, адже дозволяє перенести здобуті навички з тренажера безпосередньо в гру – і Aim Labs успішно реалізує цю функцію (вбудовані конвертери чутливості, спеціальні профілі для різних ігор). Підсумовуючи, Aim Labs – комплексне рішення, що поєднує багатий вибір вправ, дружній інтерфейс для новачків та розширену аналітику для просунутих гравців. Не дивно, що його рекомендують як перший вибір для тренувань прицільної стрільби.



Рис. 1.2. Kovaak's

Kovaak's FPS Aim Trainer [2]. *Kovaak's* – ще один провідний продукт у цій галузі, добре відомий серед шанувальників FPS. На відміну від Aim Lab, Kovaak's є комерційним (платним) програмним забезпеченням, але за ціну близько \$10 користувач отримує надзвичайно гнучкий і багатофункціональний тренажер. Головна особливість Kovaak's – це **величезна бібліотека користувацьких сценаріїв та потужні засоби їхнього редагування**. Програма фактично слугує «пісочницею» для створення будь-яких уявних тренувань: користувачі можуть налаштовувати поведінку ботів, траєкторії руху цілей, параметри зброї, швидкість появи мішеней тощо. Завдяки активній спільноті гравців, яка постійно ділиться створеними сценаріями, в Kovaak's зібрано понад **15 500 різноманітних вправ** для тренування стрільби. Цей показник перевершує кількість вправ у Aim Lab, що робить Kovaak's надзвичайно привабливим для досвідчених користувачів, які шукають нестандартні або вузькоспеціалізовані тренування. За наявності великої кількості сценаріїв у відкритому доступі гравець майже напевно знайде потрібний режим під свою задачу, «підганяючи» тренування під конкретну гру чи ситуацію. Наприклад, існують сценарії, що імітують дуелі в стилі **Overwatch**: окремі користувачі

запрограмували рух ботів та їхню модель поведінки так, щоб вони копіювали певних героїв Overwatch (Tracer, Genji тощо). Є тренування, що повторюють типові бойові ситуації з *Apex Legends*, *Halo* та інших ігор із високою вертикальністю геймплею, яку складно відтворити в інших тренажерах. Фактично, Kovaak's надає інструментарій, а спільнота наповнює його контентом – за рахунок цього програма швидко адаптується під нові ігри і мета-гру, коли змінюються вимоги до стрільби.

Щодо **збору статистики**, Kovaak's пропонує базові, але необхідні метрики. Після завершення вправи гравцеві показується підсумковий рахунок (score), точність влучань (accuracy), завданий шквал (damage) та інші показники, залежно від сценарію. Присутні таблиці лідерів (leaderboards), де можна порівняти свій результат із результатами інших гравців онлайн. Хоча глибокого аналізу, як в Aim Lab, тут немає, Kovaak's робить акцент на **повторюваності та змагальності**: гравець мотивований покращувати свій попередній результат і підніматися в рейтингу. Для тих, хто серйозно ставиться до тренувань, у спільноті розроблені комплексні програми – існують керівництва (напр. гайд Aimer7) з порадами, які послідовності вправ виконувати новачкам, а які – «просунутим стрільцям». Інтерфейс Kovaak's менш інтуїтивний для початківців, оскільки через велику кількість сценаріїв доводиться самотійно шукати потрібні режими та налаштовувати їх. Однак досвідчені гравці цінують цю **гнучкість налаштувань**: фактично, Kovaak's дозволяє змінити практично будь-який параметр тренування. Програма має **редактор сценаріїв** і підтримує *Steam Workshop* для обміну контентом, що спрощує встановлення нових вправ. Налаштування чутливості та FOV під конкретну гру теж присутні: користувач може обрати зі списку потрібний ігровий профіль або ввести вручну коефіцієнти – Kovaak's потурбується, щоб рух миші в тренажері 1:1 відповідав руху в цільовій грі. Таким чином, **сумісність із рушіями/іграми** реалізована через налаштування профілів під ігри (CS:GO, Valorant, Overwatch, Fortnite та

десятки інших). Цей тренажер розроблено на рушії Unreal Engine, тому він добре підходить для емуляції фізики тих ігор, що теж працюють на Unreal (наприклад, Fortnite, Valorant) – користувацькі сценарії можуть використовувати схожі моделі переміщення, гравітації тощо.

У плані підтримки платформ Kovaak's поступається своїм конкурентам, оскільки доступний лише на Windows, через Steam. Незважаючи на це, програма має низькі системні вимоги, що дозволяє їй працювати навіть на старих ПК, забезпечуючи високу частоту оновлення кадрів для ефективного тренування. Відсутність версій для macOS та мобільних пристроїв свідчить про те, що Kovaak's зосереджується на серйозних геймерах ПК-платформи. Незважаючи на таку обмеженість, Kovaak's здобув популярність як професійний інструмент для тих, хто бажає приділити час та енергію досконалому налаштуванню своїх тренувань. Численні кіберспортсмени інтегрують Kovaak's у свої щоденні режими для розігріву та розвитку моторної пам'яті, особливо в жанрі шутерів, де необхідна безупречна точність. Для новачків Aim Lab стане відмінним вибором завдяки своїй простоті та безкоштовній доступності, тоді як Kovaak's частіше обирають досвідчені гравці, яким необхідні спеціалізовані сценарії та максимальна можливість контролювати процес тренування.



Рис. 1.3. Aimtastic

Aimtastic [3]. Aimtastic – це відносно простий та зручний у використанні тренажер для покращення навичок стрільби, доступний безкоштовно на платформі Steam. З'явившись у 2018 році, цей тренажер став одним із перших і спеціалізується на розминці та базовому підвищенні влучності. Aimtastic пропонує гравцям різноманітні тренувальні кімнати з готовими до виконання вправами. Серед пропонованих режимів, таких як Reflex, Tracking, Assault Course та інші, кожен з яких спеціалізується на окремому аспекті стрільби, загалом доступно більше десяти варіантів, що забезпечує різноманітність тренувань. Хоча функціональність Aimtastic поступається Aim Lab і Kovaak's за обсягом, вона має свої переваги. Зокрема, програма не вимагає великих ресурсів і легко освоюється – інтерфейс простий та інтуїтивно зрозумілий, а всі режими готові до використання без складних налаштувань. Це робить Aimtastic оптимальним вибором для швидкого налаштування перед поєдинком, щоб підготувати приціл до бою.

Aimtastic забезпечує необхідну гнучкість налаштувань для реалістичного досвіду тренувань. Зокрема, програма включає конвертери чутливості та поля

зору для більшості відомих ігор. Початківці можуть скористатися інструктором, щоб швидко перенести налаштування миші з улюбленої гри в тренажер, забезпечуючи точну відповідність відчуттів від руху прицілу до гри, наприклад, Valorant або CS:GO, навіть якщо Aimtastic має власний простий візуальний стиль. Програма також пропонує налаштування під індивідуальні потреби: можна змінити приціл (форма, колір та інші деталі), а також переключитися між режимами руху гравця (для тренування стрільби зі статичної позиції або в русі, пересуваючись по кімнаті за допомогою активованого режиму руху персонажа).

По-друге, Aimtastic підтримує необхідну **гнучкість налаштувань** для реалістичності тренувань. Зокрема, вбудовано **конвертери чутливості та поля зору** для більшості популярних ігор. Новачок може скористатися покроковим майстром (*wizard*), щоб швидко перенести свої налаштування миші з улюбленої гри в тренажер. Таким чином, навіть якщо Aimtastic має власний простий графічний стиль, відчуття від руху прицілу можуть бути точно підігнані під, скажімо, Valorant чи CS:GO. Програма також дозволяє налаштовувати ряд параметрів під себе: тут є опції зміни прицілу (форма, колір та інші **просунуті налаштування прицілу**), перемикання режимів руху гравця (можна тренувати стрільбу як зі стаціонарної позиції, так і в русі – переміщуючись по кімнаті завдяки вмиканню режиму руху персонажа).

Aimtastic має базові можливості **збору статистики**: після вправ користувач бачить свою точність, час проходження (для режимів на швидкість) тощо, а також ведеться запис цих показників для відстеження прогресу. Крім того, заявлена підтримка *Steam Stats* і *Leaderboards*, отже гравець може змагатися з друзями чи власними попередніми досягненнями. Хоча глибокого аналізу (як AI-підказок) тут немає, з точки зору тренування **результативність можна оцінювати самотійно** за збереженими даними про влучність. Ці дані

допомагають зрозуміти, чи є прогрес: Aimtastic, по суті, показує гравцеві, наскільки він покращив результати в тій чи іншій вправі за певний період.

Цікавим аспектом Aimtastic є **відкритість до модифікацій**. Розробники заявили про підтримку модів та поступове розширення функціоналу за відгуками спільноти. На практиці це реалізовано через майстерню Steam: користувачі можуть створювати власні мішені, змінювати параметри деяких сценаріїв і ділитися модифікаціями. Хоча екосистема модів Aimtastic не така розвинена, як у Kovaak's, сама наявність цієї функції приваблює ентузіастів, які бажають експериментувати.

Платформна підтримка Aimtastic обмежена ПК (Windows). На інших ОС програма офіційно не підтримується, проте завдяки невибагливості її можна запускати на різних апаратних конфігураціях, включно з дуже слабкими машинами. Aimtastic – це, скоріше, **компактний тренажер** для особистого користування, який не претендує на роль всеосяжної платформи змагань або глибокого аналізу. Він добре підходить для гравців, яким потрібна проста утиліта для регулярних тренувань влучності, або для тих, хто тільки починає знайомство з aim trainers і хоче спробувати щось інше, окрім Aim Lab. За відгуками користувачів, Aimtastic ефективний як інструмент розминки перед матчами або як доповнення до інших тренажерів у щоденній практиці.



Рис. 1.4. 3D Aim Trainer

3D Aim Trainer [4]. *3D Aim Trainer* відрізняється від розглянутих вище рішень тим, що з самого початку розроблявся як **кросплатформна онлайн-платформа** для тренувань. Він доступний у вигляді веб-додатку (в браузері) та мобільних застосунків, а з 2021 р. має і десктопну версію на Steam. 3D Aim Trainer став популярним завдяки тому, що не вимагає встановлення – кожен охочий може зайти на сайт і негайно почати серію вправ. Це суттєво розширило аудиторію: за даними розробників, сервісом скористалися понад 12 мільйонів гравців FPS. Основна концепція 3D Aim Trainer – **синхронізація з конкретною грою користувача та планомірне покращення навичок через систему рівнів**. Після запуску програми або веб-версії користувачу пропонується вибрати гру, під яку він хоче тренуватися (наприклад, Call of Duty, CS:GO, Apex тощо), і ввести її налаштування управління (чутливість миші, FOV) – тренажер адаптується під ці параметри. Далі рекомендується пройти оцінювальний тест (assessment), щоб визначити поточний рівень вмінь. На основі результатів 3D Aim Trainer **формує індивідуальні рекомендації** щодо тренувань.

Програма містить декілька режимів. Перший – **Quick Play**: це швидкий доступ до окремих коротких вправ (наприклад, стрільба по мішенях, що з'являються, або трекінг цілі), яких наразі понад 15 видів. Другий ключовий режим – **Fundamental Aim Academy**, структурована навчальна програма, що складається з понад 130 рівнів, розділених на 5 категорій навичок. Ці рівні поступово ускладнюються, вводячи нові елементи – таким чином гравець “прокачує” свій приціл, переходячи від базових вправ до просунутих. Розробники залучили професійних тренерів з точності, щоб створити докладні **покрокові туторіали** до окремих завдань та цілих тренувальних програм. Фактично, 3D Aim Trainer прагне бути не просто набором вправ, а **віртуальним тренером**, який веде гравця від початківця до експерта. Це відображається і в елементах гейміфікації: введені **ачівменти, медалі, рейтинги**, що стимулюють регулярно тренуватися і досягати прогресу.

Збір і візуалізація статистики – одна з сильних сторін 3D Aim Trainer. Платформа відстежує численні метрики: **точність, середній час реакції, швидкість наведення, стабільність трекінгу, недонаведення/перенаведення (undershoot/overshoot)** тощо. Після кожного рівня користувач бачить не тільки свій результат, а й як він виглядає відносно середнього по спільноті або відносно власних попередніх показників. На сторінці статистики доступні графіки прогресу по кожному вмінню, що дозволяє легко ідентифікувати, в чому саме відбулися покращення, а над чим ще слід працювати. Система також може давати узагальнені висновки – наприклад, вказати, що точність у користувача на високому рівні, а от час реакції відстає. Слід зазначити, що **візуалізація даних** тут одна з найкращих серед аналогів: враховуючи веб-орієнтованість, розробники реалізували зручні інтерактивні графіки і діаграми. Гравець може експортувати свій профіль або поділитися ним – фактично, 3D Aim Trainer створює “*паспорт стрілка*”, яким можна похвалитися чи проаналізувати спільно з тренером.

Платформна незалежність 3D Aim Trainer – значна перевага. **Підтримка платформ:** Windows (власний клієнт та Steam), Mac (через веб-версію або додаток SteelSeries GG), браузері (Chrome, Firefox тощо без плагінів), а також мобільні пристрої (є застосунки для Android та iOS). Це дозволяє тренувати реакцію навіть на смартфоні, наприклад, у дорозі, хоча, звісно, точність керування на сенсорному екрані інша. Проте для базових тренувань на швидкість реакції мобільний додаток може бути корисним. 3D Aim Trainer інтегровано в екосистему **SteelSeries GG** – програмне забезпечення від виробника ігрової периферії SteelSeries. Це означає, що програма отримує підтримку від відомого бренду, а також може використовуватися спільно з їхніми пристроями (приміром, відстеження рухів миші з високою точністю). **Гнучкість налаштувань** у 3D Aim Trainer присутня в тій мірі, щоб підлаштувати середовище під конкретну гру: як згадано, це синхронізація сенси, FOV, вибір моделі зброї з певної гри (є підтримка різних видів озброєння з їх унікальною віддачею для популярних шутерів). Однак користувацьких сценаріїв, як у Kovaak's, тут створювати не можна – всі режими задані розробниками. В цьому сенсі 3D Aim Trainer менш гнучкий для експериментів, зате забезпечує **стандартизовану тренувальну програму**.

3D Aim Trainer, в цілому, розрахований на масового користувача, який прагне підвищити свою точність стрільби без детального вивчення налаштувань. Програма пропонує готовий тренувальний курс з поступовим збільшенням складності, що особливо зручно для початківців або гравців середнього рівня. Професіонали можуть використовувати цей інструмент для відстеження своєї форми та участі в рейтингах точності на міжнародному рівні.

Порівняльна характеристика розглянутих тренувальних програм наведена в табл. 1.1. В таблиці узагальнено основні можливості кожного рішення за п'ятьма обраними параметрами.

Таблиця 1.1 – Порівняння програм для тренування точності й реакції в FPS

Програма	Функціональність (режими та можливості)	Підтримувані платформи	Можливості статистики	Гнучкість налаштувань	Сумісність із рушіями/іграми
Aim Labs	~12 тис. сценаріїв; офіційні тренування під Valorant, R6; адаптивні вправи; AI-аналітика навичок	PC (Windows); мобільні (Android/iOS); планується на консолях	Детальний аналіз: графік навичок, показники реакції, точності, поради AI	Висока: налаштування складності, мішеней, прицілу; створення власних сценаріїв (Creator Studio)	Профілі під різні ігри (чутливість, FOV); 1:1 карти Valorant; підтримка геймпада
Kovaak's	>15 тис. користувацьких сценаріїв; редагування та створення вправ; реалістичні боти та фізика руху	PC (Windows)	Базова: точність, очки, шкода, лідерборди	Дуже висока: повний редактор сценаріїв, підтримка модів (Workshop); тонке налаштування	Профілі під >20 ігор; емуляція фізики різних ігрових рушіїв (Unreal, Source)

				ння кожного параметра	тощо) через налаштува ння
Aimtastic	~10+ вбудованих режимів (реакція, трекінг, рух та стрільба тощо); базові вправи для розминки	PC (Windows)	Основна: відсоток влучань, час реакції; збережен ня прогресу	Помірна: налаштува ння чутливості, FOV через майстер; вибір прицілу; підтримка руху гравця; модифікаці ї	Конвертер и налаштува нь під популярні ігри; відчуття прицілу як у заданій грі
3D Aim Trainer	130+ рівнів Aim Academy; Quick Play режими; тренувальні плани; навчальні туторіали; гейміфікація (ачівменти)	PC (Windows, Mac); Web (браузер); мобільні (Android/iO S)	Розширен а: точність, час реакції, графіки прогресу, глобальні рейтинги	Обмежена: налаштува ння під конкретну гру (сенса, FOV, зброя); решта сценаріїв фіксовані	Профілі для ігор (CS:GO, Valorant та ін.); симуляція віддачі зброї тієї чи іншої гри

Примітка: усі тренажери дозволяють налаштувати чутливість миші відповідно до вибраної гри, що забезпечує коректну **трансляцію навичок** з тренувального середовища до змагальної гри. Aim Lab та 3D Aim Trainer є безкоштовними продуктами, Kovaak's – платний (комерційний), Aimgistic – безкоштовний. Кожне рішення регулярно оновлюється розробниками та спільнотами користувачів, отримуючи нові сценарії та функції.

Проаналізувавши таблицю, можна зробити ряд висновків. По-перше, **функціональність** тренажерів певною мірою відображає їхню цільову аудиторію. Aim Lab та 3D Aim Trainer націлені на широку аудиторію гравців: вони пропонують сотні готових вправ і навіть повноцінні програми навчання, що зручно для новачків і гравців середнього рівня. Натомість Kovaak's орієнтований на ентузіастів та професіоналів, які бажають самостійно формувати свій тренувальний процес – звідси акцент на користувацькому контенті і безпрецедентній гнучкості налаштувань. Aimgistic займає проміжне положення: він забезпечує базові режими для відпрацювання ключових навичок без зайвих складнощів, слугуючи скоріше доповненням або стартовим майданчиком.

По-друге, **можливості збору статистики й аналітики** найкраще реалізовані в Aim Lab та 3D Aim Trainer. Aim Lab пропонує унікальний AI-підхід з детальним аналізом помилок і рекомендаціями, а 3D Aim Trainer демонструє прогрес гравця по численних метриках і дозволяє порівнювати себе з іншими в глобальному масштабі. У Kovaak's і Aimgistic аналітика більш спрощена: вони зосереджені на основних показниках (точність, рахунок), надаючи гравцеві більше самостійності в інтерпретації результатів. Варто відзначити, що всі розглянуті програми зберігають історію результатів, отже гравець може відстежувати динаміку своїх навичок у часі.

По-третє, відмінності у **підтримці платформ** можуть бути важливими з практичної точки зору. 3D Aim Trainer має найбільше охоплення платформ, включаючи браузер і смартфони, що робить його найдоступнішим – не потребує потужного ПК чи взагалі будь-якої інсталяції. Aim Lab, хоча офіційно тільки на ПК, також розширюється на мобільні пристрої, а в майбутньому і на консолі. Для гравців, які грають на різних платформах (наприклад, і на ПК, і на консолі), це може мати значення – можливість тренуватися в одному середовищі з використанням різних пристроїв. Kovaak's та Aimgame обмежені Windows-ПК, що, втім, покриває потреби основної аудиторії FPS-кіберспортсменів.

Нарешті, **сумісність із іграми та рушіями** забезпечена у всіх тренажерах через систему профілів налаштувань, однак Aim Lab пішов далі, надавши офіційні карти та сценарії від розробників ігор. Це може покращити перенесення навичок, адже тренування відбувається в дуже наближених до реальних умовах конкретної гри (наприклад, та сама геометрія рівня і точки появи ворогів, що й на турнірній карті). З іншого боку, Kovaak's завдяки гнучкості дозволяє спільноті швидко адаптуватися до будь-якої нової гри: як тільки виходить новий шутер, користувачі можуть створити під нього модуль тренування, налаштувавши потрібну фізику та мішені. Таким чином, усі рішення мають високий ступінь сумісності з популярними FPS, відрізняючись лише підходом до її реалізації.

1.3 Постановка задачі

Аналіз предметної області показав, що наявні програмні рішення пропонують широкий спектр можливостей для тренування стрілецьких навичок. Водночас розробка власного тренувального середовища для підготовки до змагань із кіберспорту є виправданою з кількох причин. По-перше, це дозволяє інтегрувати необхідні функції саме під конкретні

потреби команди чи окремого гравця. Комерційні продукти, хоча й потужні, залишаються закритими платформами – немає змоги повністю переробити або розширити їх функціонал під індивідуальні вимоги. Власне рішення на рушії Unity дасть можливість реалізувати будь-які сценарії тренувань «з нуля», контролюючи кожен аспект: від поведінки цілей до інтерфейсу відображення статистики. По-друге, розробка такого середовища має навчально-дослідницьку цінність. Створюючи аналогічний інструмент, можна випробувати нові підходи або поєднати найкращі риси різних існуючих програм. Наприклад, можна сфокусуватися на **тренуванні реакції** шляхом розробки спеціальних динамічних мішеней, які будуть з'являтися непередбачувано, і поєднати це з **системою адаптивної складності** (як в Aim Lab) для підтримання оптимального рівня виклику для гравця. Інший приклад – можна реалізувати **модуль аналітики**, що фіксуватиме не лише кінцеві показники (точність, час), а й, скажімо, траєкторію руху прицілу, промахи відносно цілі, затримку перед реакцією тощо, з подальшою візуалізацією цих даних. Таке глибоке логування дій гравця може піти далі, ніж існуючі рішення, і надати тренеру або самому гравцю нові метрики для вдосконалення. По-третє, власний проєкт дозволить врахувати **технологічні аспекти** інтеграції. Оскільки розробка ведеться на Unity, теоретично можливо передбачити сумісність із іншими системами – наприклад, експортувати зібрану статистику в зовнішню базу даних або поєднати тренажер з навчальним веб-порталом для гравців. Unity як платформа також відкриває шлях до **кросплатформенності**: після створення основного функціоналу на ПК з мінімальними зусиллями можна згенерувати версії для Android чи WebGL (браузера), якщо виникне така потреба.

З урахуванням огляду аналогів, сформулюємо постановку задачі для даної кваліфікаційної роботи. Необхідно спроектувати і реалізувати **тренувальне середовище для FPS**, яке забезпечує ефективне тренування

навичок швидкості реакції та точності стрільби. Середовище повинно містити декілька режимів тренувань, зокрема:

- **Режим тренування реакції** – швидкоплинні цілі з непередбачуваною появою, що вимагають миттєвого реагування (наприклад, цілі, що з'являються на екрані на дуже короткий час). Метрика успіху: час реакції на появу цілі, процент вражених цілей.
- **Режим тренування точності (флік-шоти)** – серії стаціонарних або рухомих мішеней, які потрібно швидко вражати по черзі. Мета – мінімізувати промахи і час наведення між цілями.
- **Режим трекінгу** – одна або кілька цілей постійно рухаються по складній траєкторії, гравець мусить утримувати приціл на них, завдаючи безперервну шкоду. Оцінюється середня точність трекінгу та стабільність наведення.
- **Комплексний режим** – поєднання стрільби і руху гравця. Наприклад, реалізувати тренувальний рівень, де користувач пересувається лабіринтом або ареною і реагує на випадкові мішені (імітуючи ситуацію реального бою). Це дозволить відточувати навички стрільби в русі та просторі.

Кожен із режимів слід забезпечити **системою налаштувань**. Зокрема, необхідно передбачити: вибір рівня складності (швидкість цілей, їхній розмір, кількість одночасних цілей), налаштування параметрів управління (чутливість миші, інверсія осей, FOV – кут огляду камери). Останнє особливо важливо для **сумісності з іграми**: користувач повинен мати змогу ввести свої налаштування з улюбленої гри, щоб тренування відбувалося в умовах, максимально наближених до реальних. Таким чином, одним із завдань є реалізація модуля

конвертації чутливості – наприклад, надання заздалегідь заданих профілів для популярних ігор (CS:GO, Valorant, Overwatch, PUBG тощо) або можливість вручну ввести сантиметри на 360° повороту камери.

Наступним критично важливим компонентом є **система збору статистики та аналізу**. Тренувальне середовище повинно автоматично збирати дані про результати гравця в кожній сесії: процент влучень, кількість вибитих цілей, середній час реакції, кращий час реакції, коливання точності протягом вправи, кількість промахів, коефіцієнт успішності тощо. Зібрані дані мають відображатися користувачу у зручній формі по завершенні вправи – у вигляді таблиць або графіку. Бажано передбачити **збереження історії** результатів та можливість переглянути прогрес (наприклад, графік зміни точності за останні 10 сесій тренування). Для реалізації такого модуля аналітики в Unity слід розробити структури даних для зберігання показників і алгоритми розрахунку похідних метрик (середніх, перцентилів тощо). За можливості, можна інтегрувати прості підказки на основі правил – наприклад, якщо система бачить, що точність суттєво впала при збільшенні швидкості мішеней, вона може рекомендувати перейти на простіший рівень або зосередитися на окремому режимі.

Таким чином, постановку задачі можна підсумувати так: необхідно створити багатофункціональний програмний комплекс для тренування стрільби в FPS, який включає декілька тренувальних режимів (реакція, точність, трекінг, комбінований), підтримує гнучке налаштування під потреби гравця та різні ігри, а також здійснює збір і візуалізацію статистики для аналізу результатів і прогресу. Розробка здійснюватиметься на рушії Unity, що забезпечить необхідний рівень графічної та фізичної реалістичності, а також полегшить подальше розширення і підтримку програми. В результаті виконання даної роботи має бути отримано працюючий прототип тренувального середовища,

який можна застосувати для підготовки кіберспортсменів до змагань або для індивідуального вдосконалення навичок стрільби.

Вимоги до системи для розроблюваного середовища збігаються зі стандартними для ігор на Unity: цільова платформа – ПК (Windows) з можливістю запуску на середніх апаратних конфігураціях при 60+ FPS (для точності тренування бажаний високий і стабільний фреймрейт). Інтерфейс користувача має бути інтуїтивно зрозумілим, щоб гравець міг швидко обирати режими та переглядати статистику.

Отже, сформульовано концепцію та вимоги до тренувального середовища. Наступні розділи роботи будуть присвячені проєктуванню архітектури програми згідно з цими вимогами (розробці структури сцен Unity, скриптів керування логікою тренувань, схеми збереження даних тощо), безпосередній реалізації прототипу та його тестуванню.

Висновки до розділу 1

Проведений аналіз предметної області підтвердив високу значущість навичок швидкості реакції та точності стрільби для успішної гри в FPS на змагальному рівні. Розглянуто сучасні підходи до тренування цих навичок та проаналізовано найбільш популярні програмні рішення (Aim Lab, KovaaK's, Aimtastic, 3D Aim Trainer). Порівняння показало, що існуючі тренажери надають широкий спектр функцій: від тисяч користувацьких сценаріїв до AI-аналітики помилок гравця. Всі вони мають свої переваги і обмеження – жодне з рішень повністю не домінує за всіма критеріями, що обґрунтовує доцільність розробки нового тренувального середовища, яке б поєднало найкращі практики. На основі аналізу сформульовано чітку постановку задачі розробки: визначено перелік режимів тренувань, необхідні функції

налаштування та збору статистики, вимоги сумісності з іграми. Таким чином, в першому розділі закладено теоретичну основу для подальшої роботи – проєктування і реалізації власного програмного засобу для тренування кіберспортсменів. Отримані результати огляду будуть використані при розробці архітектури системи та виборі технологічних рішень у наступних розділах.

РОЗДІЛ 2

РОЗРОБКА ТРЕНАЖЕРА ТОЧНОСТІ НА UNITY

2.1. Реалізація тренажера влучності

У цьому підрозділі описано процес створення тренажера точності, розробленого на ігровому рушії Unity[9]. Метою цього тренажера є тренування швидкості реакції та влучності гравця у віртуальному середовищі.

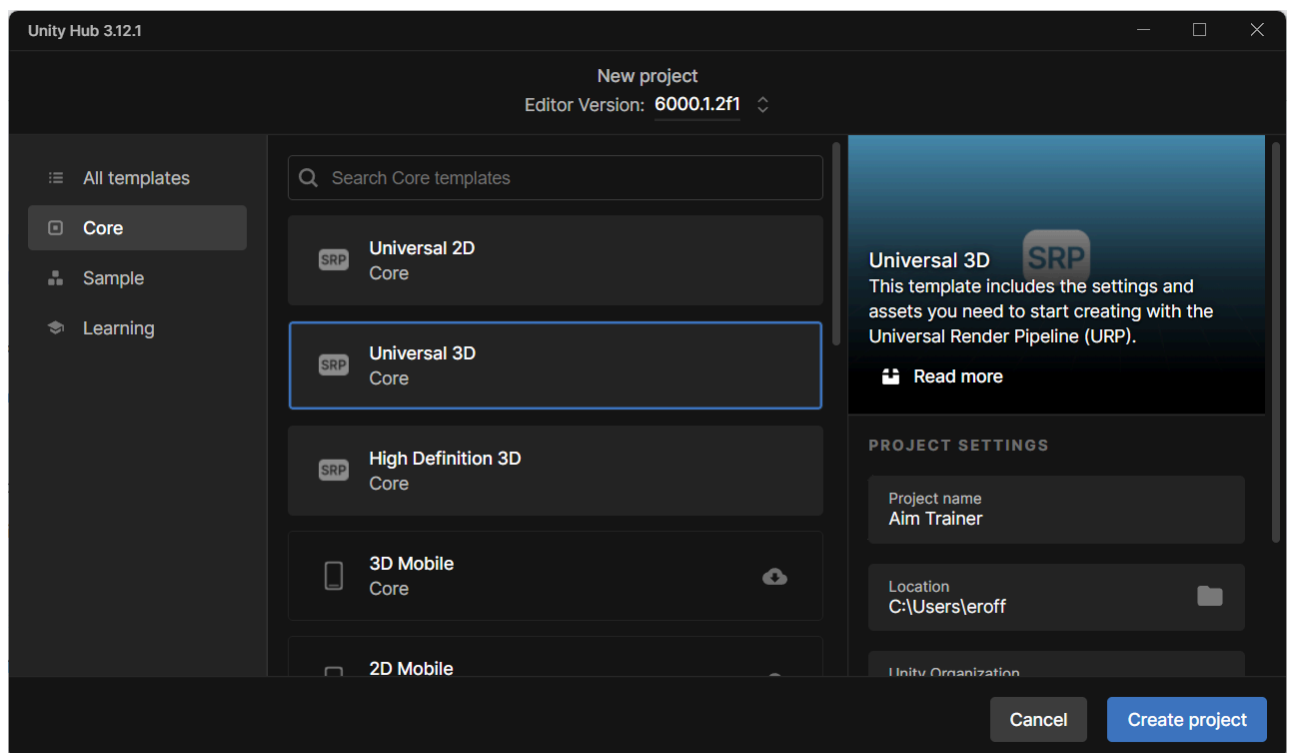


Рис 2.1. Створення проєкту

Для початку було створено у програмі Unity Hub[12] новий проєкт, беручи за основу Universal 3D Core (Рис. 2.1).

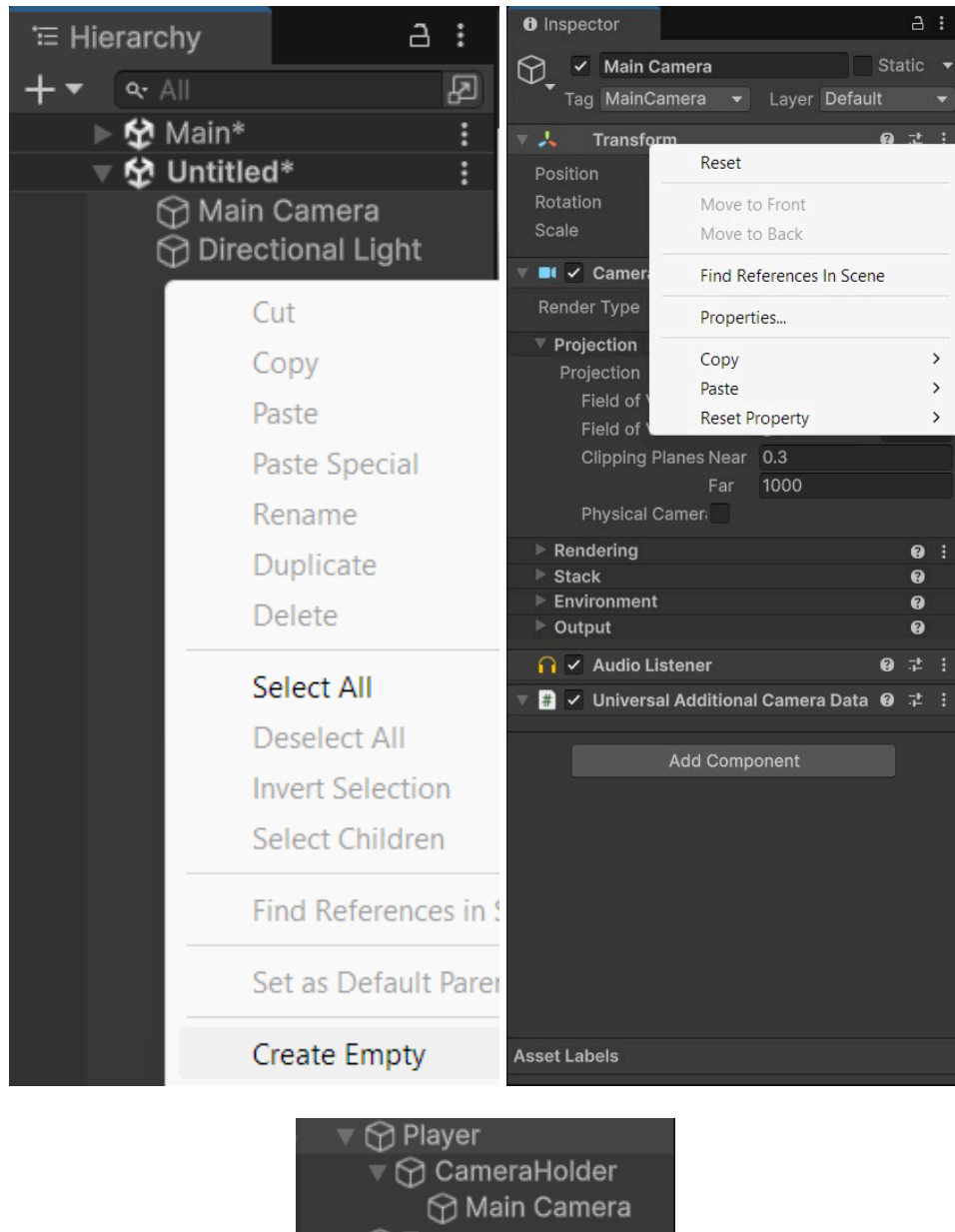


Рис. 2.2 Створення та перенос об'єктів

У полі Hierarchy у порожній сцені створюється порожній об'єкт “Create Empty” (Рис.2.2.1), якому надається ім'я Player. Наступним кроком у Player знову створюється порожній об'єкт CameraHolder та переноситься в нього головну камеру (Рис. 2.2.2), попередньо повернувши її в початкове положення (Рис.2.2.3).

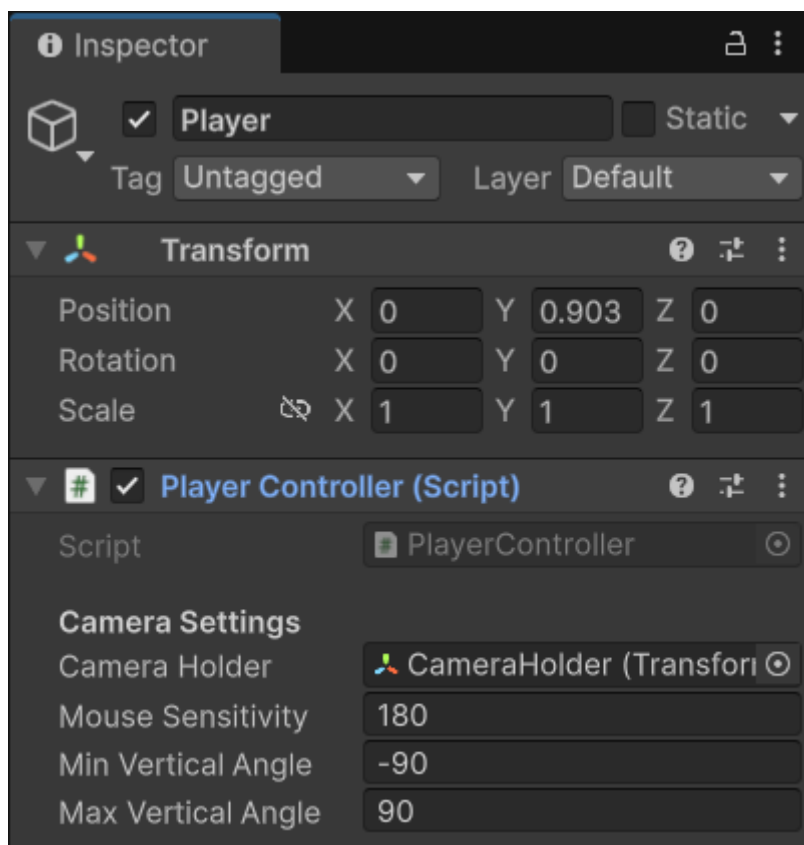


Рис. 2.3 Додавання компонента

Далі переходимо до кодування. Для зручності було створено у ассетах проекту папку Scripts, де в подальшому зберігатимуться всі файли коду. Для початку треба зробити так, щоб камера гравця рухалася. До папки Scripts було додано порожній файл PlayerController та прописано код (див. Додаток А). Після цього, відкривши вікно Inspector для об'єкту Player, було додано PlayerController у вигляді компонента. Обов'язково обирається у полі Camera Holder відповідний об'єкт, щоб прив'язати саме головну камеру до рухів миші (Рис. 2.3).

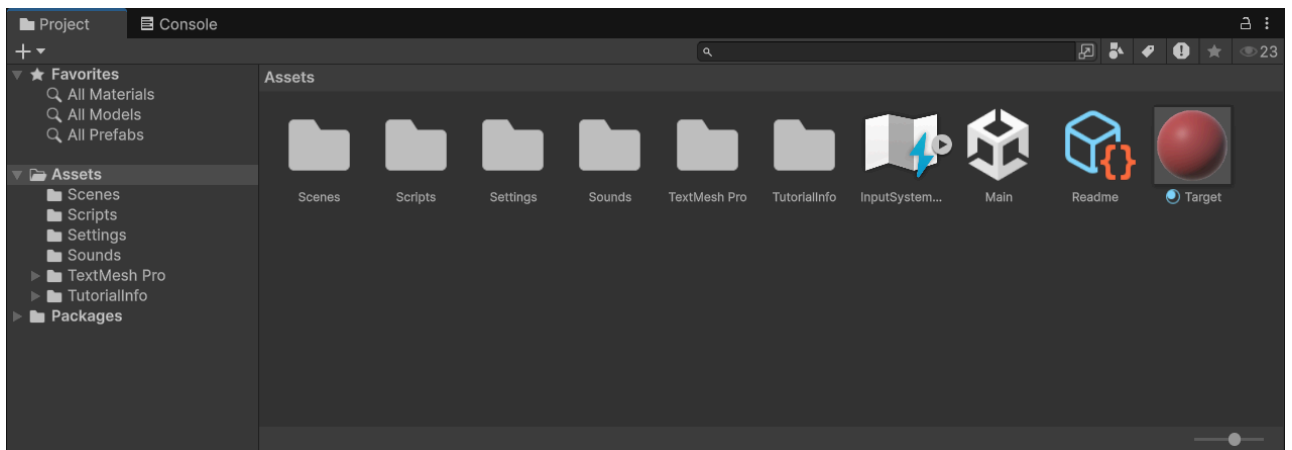


Рис. 2.4. Додавання матеріалу мішені

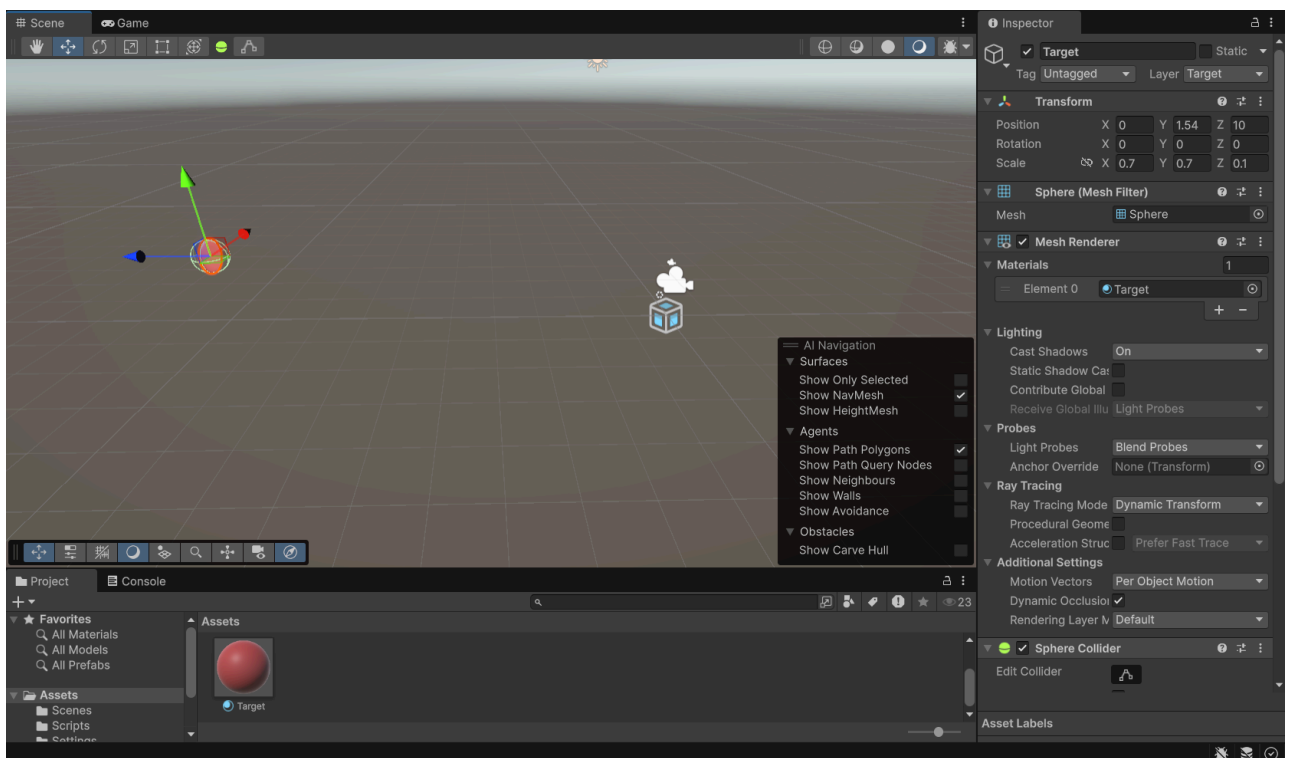


Рис. 2.5. Перенос та трансформація мішені

Наступний етап це створення мішені. До сцени було додано 3D об'єкт кулі, котрій надано ім'я Target. У вікні проекту було створено матеріал для Target та перенесено на відповідний об'єкт (Рис. 2.4). Залишається лише віддалити Target від гравця та змінити форму на більш подібну до мішені. (Рис. 2.5).

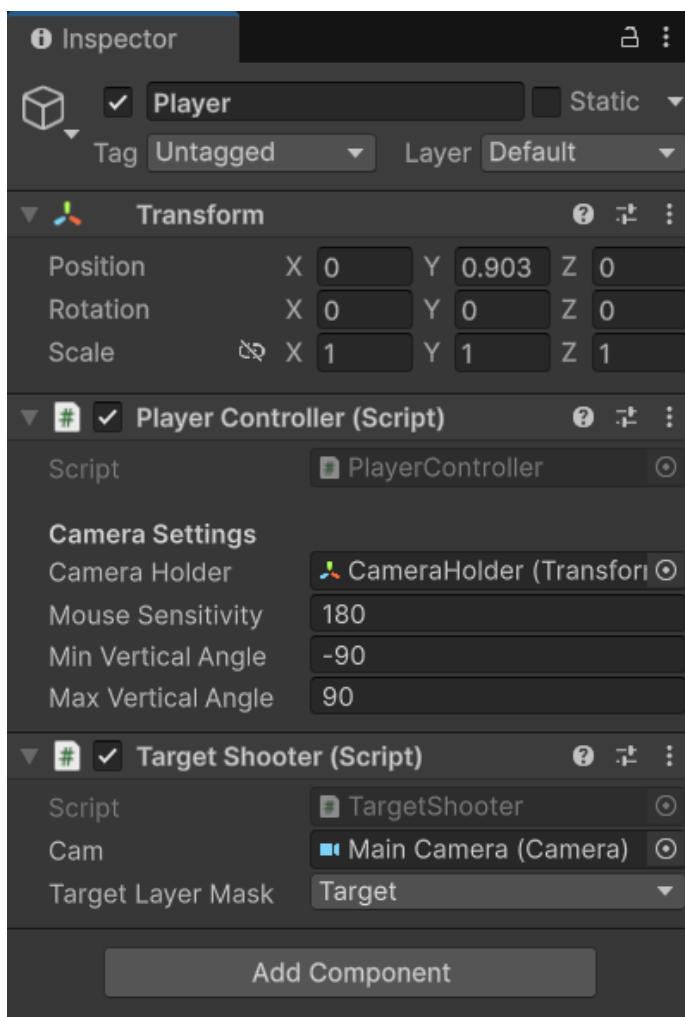


Рис. 2.6. Налаштування компоненту TargetShooter

Після створення мішені переходимо до програмування. Необхідно зробити так, щоб мішені після влучання зникали та з'являлися іншому місці в межах певної області. Для реалізації було створено наступні файли у Scripts: Target (див. Додаток Б), TargetShooter (див. Додаток В) та TargetBounds (див. Додаток Г). Код з файлу Target буде використовуватися для відповідного об'єкта, TargetShooter до гравця, а TargetBounds до області появи мішеней. Для TargetShooter необхідно вказати камеру а також шар цілі. Тому було додано шар Target та застосовано до мішені, після чого обрано відповідний пункт у полі TargetShooter.

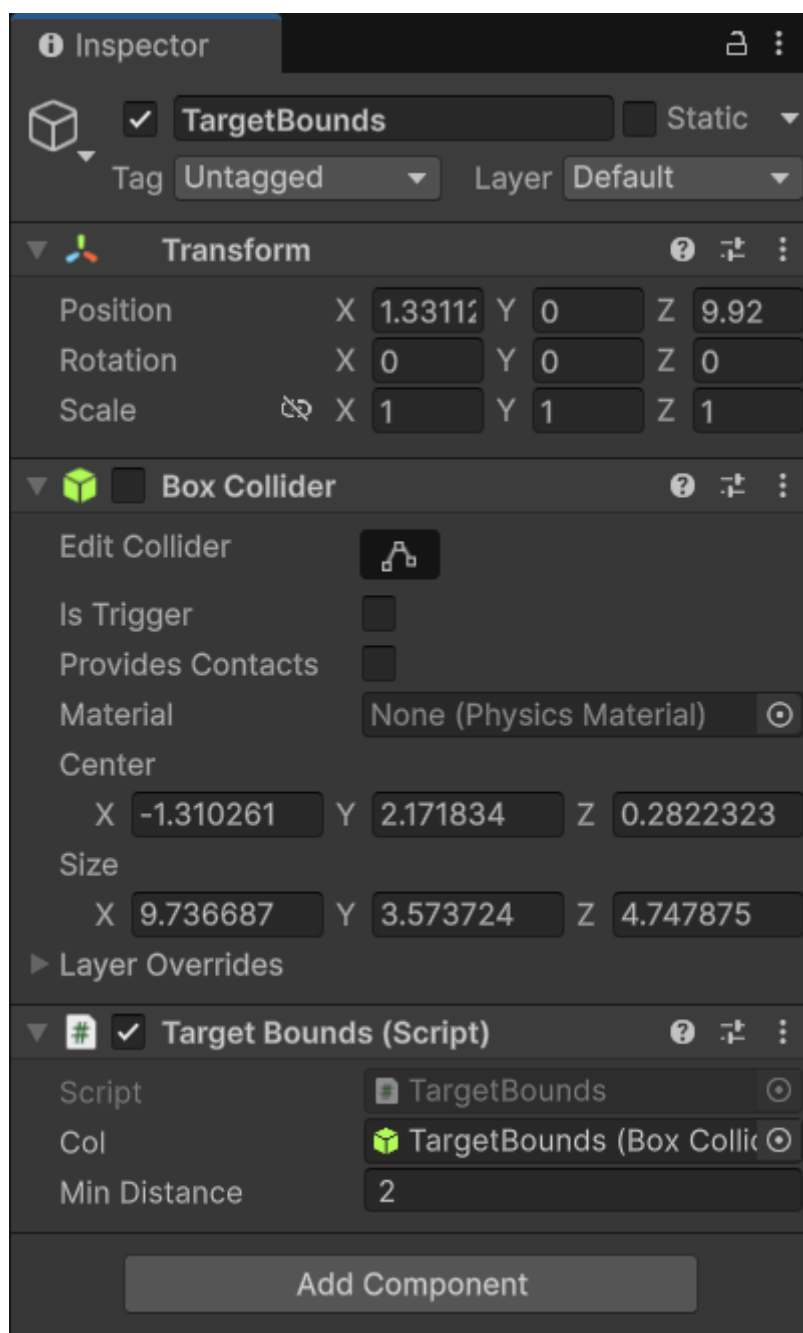


Рис. 2.7. Налаштування TargetBounds у Inspector

Перейдемо до TargetBounds. Знову було створено порожній об'єкт на ім'я TargetBounds та додано компонент Box Collider, який і буде областю появи мішеней. Додається компонент скрипта до TargetBounds та налаштовується. Обов'язково прибирається флажок біля компоненту Box Collider, так як він потрібен лише як область (Рис. 2.7.).

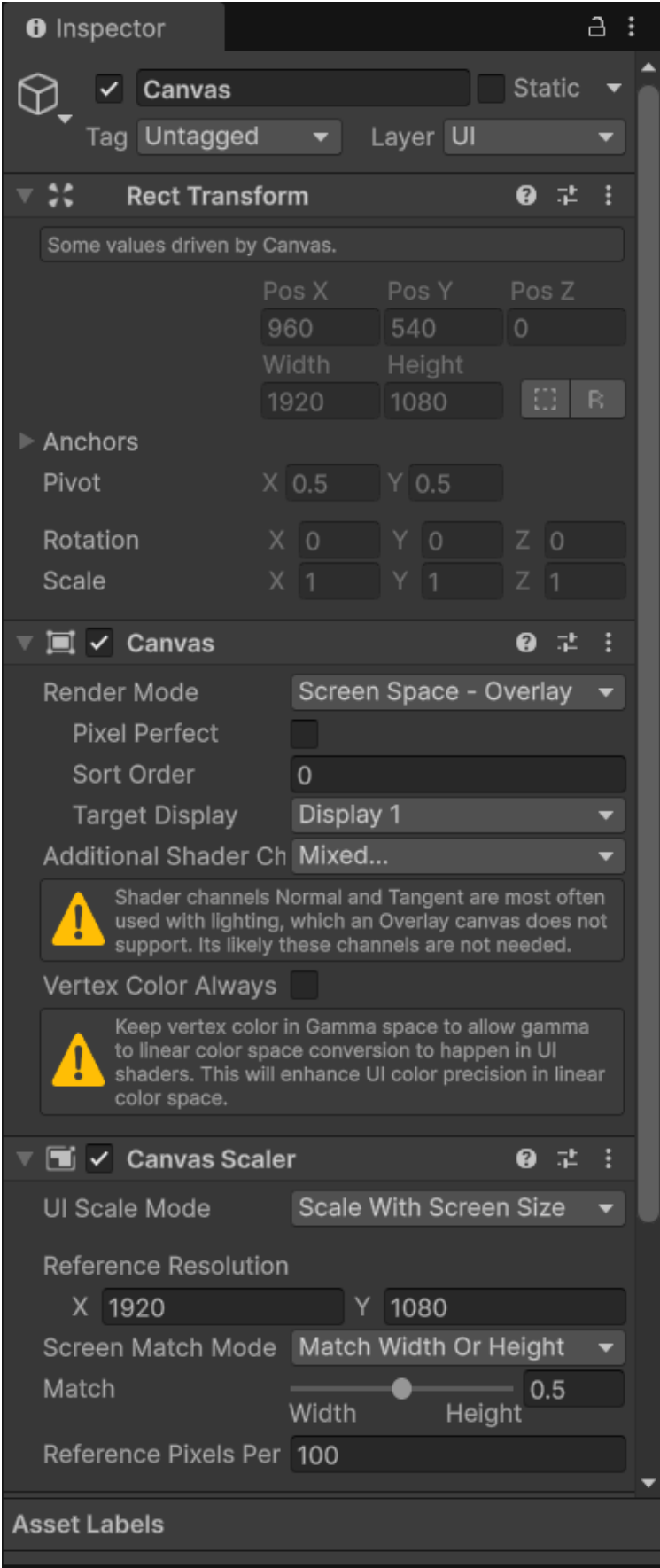


Рис. 2.8. Налаштування полотна

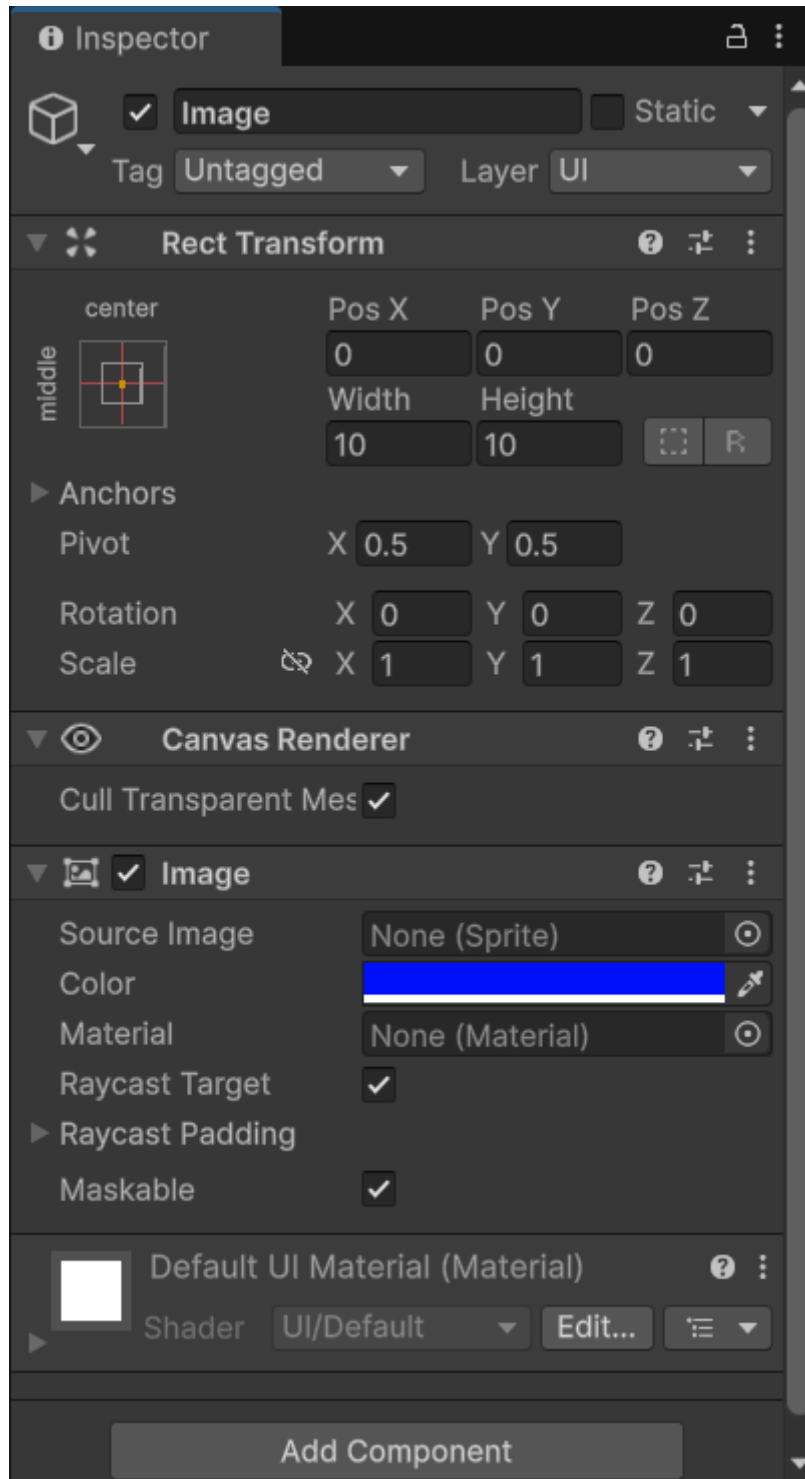


Рис. 2.9. Налаштування перехрестя

На цей момент основа тренажера вже створена. Переходимо до інтерфейсу. В першу чергу слід створити полотно. В ньому в подальшому буде знаходитися весь інтерфейс. Вказавши розміри полотна (рис. 2.8), у нього

додається елемент зображення - це буде перехрестя. Було змінено його розміри та колір для більшої зручності під час тренування (рис. 2.9).

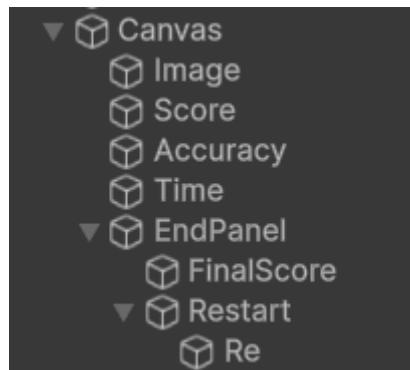


Рис. 2.10. Вміст полотна

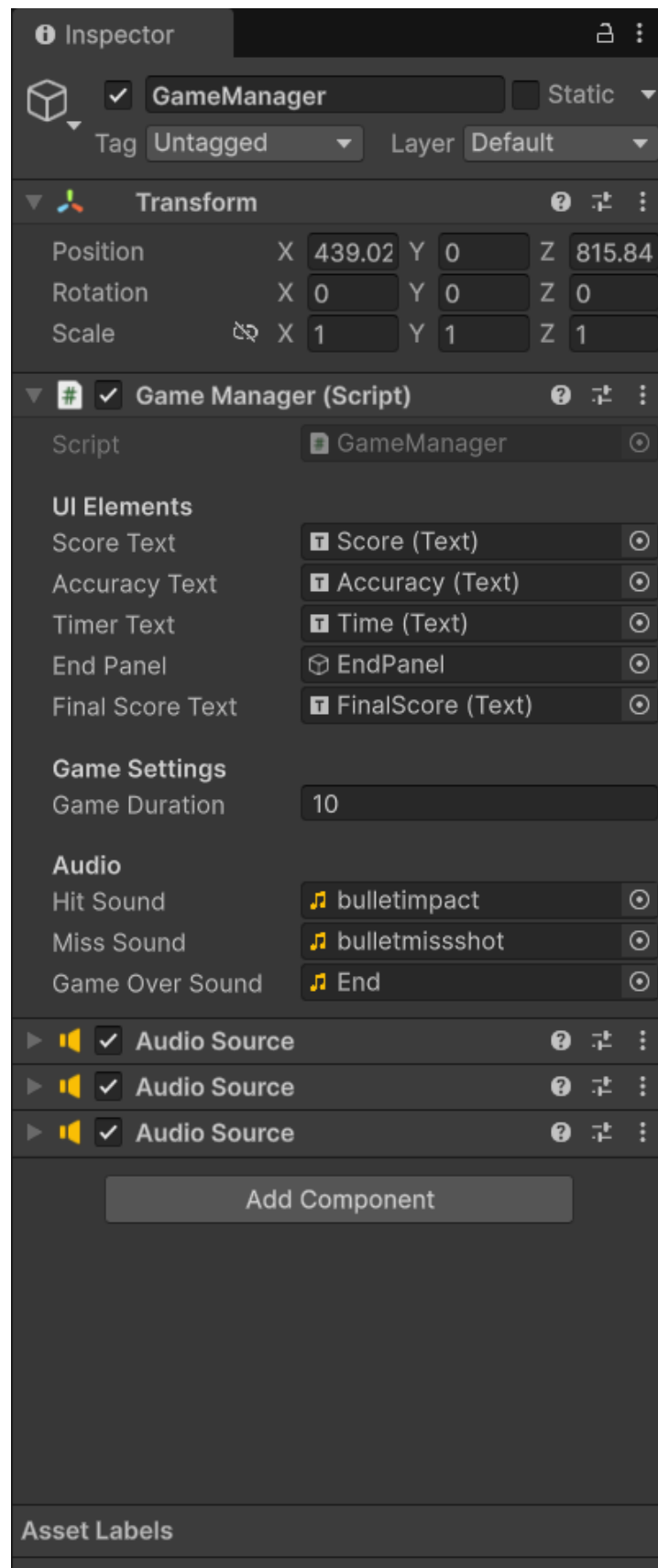


Рис. 2.11. Налаштування GameManager у Inspector

Наступним кроком було додано статистичні дані до інтерфейсу, а саме таймер, показник влучності та лічильник влучань у вигляді текстових полей. Також необхідно зробити меню зі статистикою та можливістю почати тренування ще раз. Щоб все працювало, створюємо новий скрипт GameManager (див. Додаток Д). Для зручності було створено порожній об'єкт під назвою Game Manager на сцені та додаю до нього скрипт.

Для створення меню до полотна було додано панель EndPanel. Вона повинна бути нижче інших частин полотна, щоб мати найвищий пріоритет. До панелі було додано текстове поле з загальним підсумком влучань а також кнопку для перезапуску тренажера.

Також було прийнято рішення додати звуковий супровід на пострілів, промахів та появи меню[8]. У ассетах проєкту було створено папку Sounds, до якої було додано туди три аудіодоріжки. Щоб GameManager бачив аудіодоріжки, додаємо три компонента Audio Source. Коли все вищеперелічене створено, переходимо до налаштування GameManager (рис. 2.11).

Останній крок це додавання стін та підлоги. Створється порожній об'єкт Room. В нього додається підлога та стіни, після чого підключається матеріал. Тепер гравець буде більш сконцентрований у межах простору тренажера.

2.2. Тестування, налагодження та результати:

Для перевірки правильності роботи гри було проведено всебічне тестування, що включало модульну перевірку окремих компонентів, інтеграційне тестування системи в цілому, а також пілотне тестування із залученням кількох користувачів. Спочатку перевірялися окремі модулі гри – коректність роботи алгоритмів руху, генерації цілей, підрахунку очок тощо. Надалі виконувалося інтеграційне тестування: гра запускалася у повному режимі, щоб виявити можливі проблеми взаємодії компонентів та перевірити стабільність роботи програми під навантаженням. Після внутрішнього

тестування гри було передано на пробне тестування незалежним гравцям, які спробували тренажер та надали свої відгуки.

Процес налагодження дозволив виявити та усунути низку помилок. Зокрема, було виявлено логічну помилку в алгоритмі обчислення влучності: відсоток влучень обчислювався неправильно через некоректне урахування частини тестових пострілів. Це виправлено шляхом скидання лічильників на початку кожного раунду та фільтрації незарахованих пострілів. Також виникала проблема з оновленням інтерфейсу: після повторного запуску раунду лічильник очок не обнулявся, що могло ввести гравця в оману. Цю помилку вирішено додаванням процедури скидання значень інтерфейсних елементів при старті нової гри. Крім того, виявилось, що при значному збільшенні кількості об'єктів-цілей на екрані помітно знижувалася частота кадрів (гра починала пригальмовувати). Для усунення цього оптимізовано процеси рендерингу та встановлено обмеження на кількість активних цілей одночасно. Усі виявлені баги були успішно виправлені, і програма демонструє стабільну роботу.

У ході тестування фіксувалися та аналізувалися основні метрики результатів гравця. Серед них ключовими є влучність стрільби та набраний рахунок. Влучність обчислюється як відсоток влучних пострілів від загальної кількості здійснених пострілів. Цей показник оновлюється в реальному часі й відображається в інтерфейсі, даючи гравцю можливість стежити за своїм прогресом. Рахунок нараховується за кожне влучення з урахуванням складності цілі та швидкості її знищення і відображається у кутку екрана під час гри. Після завершення тренувального раунду гравцю показується зведення результатів: підсумковий рахунок, середня влучність та інші статистичні показники (наприклад, кількість влучень та затрачений час).

Загалом працююча версія гри є повністю функціональною та відповідає початковим вимогам. Інтерфейс інтуїтивний і містить усі необхідні елементи:

приціл для наведення, лічильник очок та індикатор влучності, які оновлюються в реальному часі. Наприклад, у верхньому лівому куті екрана гравець бачить свій поточний рахунок, поруч із ним – відсоток влучності, а посередині вгорі розташовано таймер зворотного відліку часу. На ігровому полі мішені з’являються у випадкових позиціях, що підтримує інтерес та належний рівень складності тренування. Після завершення раунду відображається екран із підсумками та кнопками для повторного запуску або виходу в меню. Таким чином, протягом етапів тестування і налагодження гра була доведена до робочого стану, а її точність та стабільність суттєво покращено.

Висновки до розділу 2

Таким чином, у другому розділі було повністю реалізовано тренувальне середовище згідно з вимогами проєкту. Усі поставлені завдання виконано в повному обсязі. Розроблено всі ключові компоненти тренажера: ігрова логіка (генерація і знищення цілей, нарахування очок), система оцінки та відображення результатів (рахунку, влучності тощо), а також графічний інтерфейс користувача. У результаті отримано повністю працездатний зразок тренувального застосунку, що відповідає заданим критеріям.

Для практичної реалізації використовувалися сучасні інструменти та технології. Зокрема, гру було створено за допомогою ігрового рушія Unity 3D мовою C#, що забезпечило високий темп розробки та якісну реалізацію ігрових механік. У ході роботи також використовувалися графічні редактори для створення ресурсів (текстур, іконок тощо) та система контролю версій Git.

На момент завершення цього розділу проєкт можна вважати готовим до експлуатації. Розроблене тренувальне середовище пройшло контрольне тестування і підтвердило свою надійність та функціональність. Отже, мету даного розділу – створення і практична перевірка працездатності

тренувального середовища – успішно досягнуто, а створений продукт готовий до застосування.

ВИСНОВКИ

У першому розділі було проведено аналіз існуючих тренувальних програм і засобів для підготовки кіберспортсменів, вивчено сучасні підходи до тренування гравців та сформульовано мету і завдання для розробки власного тренажера.

Другий розділ присвячено практичній реалізації цього задуму. У ньому створено, детально описано і протестовано програмний продукт – тренувальне середовище для кіберспорту. Усі заплановані функції реалізовано, а тестування підтвердило коректність і стабільність роботи системи. У підсумку було створено функціональний і надійний тренажер, який дає можливість кіберспортсменам підвищувати свій рівень.

Розроблений продукт є важливим для підготовки до кіберспорту, оскільки надає гравцям можливість практикуватися в контрольованому середовищі, спрямованому на вдосконалення певних навичок. Такий тренажер допомагає систематизувати тренувальний процес, кількісно вимірювати прогрес та підвищувати ефективність підготовки до змагань. Його впровадження у тренувальну діяльність кіберспортсменів може стати невід’ємною частиною підготовки, покращуючи їхні результати під час тренувальних спарингів та офіційних матчів.

Проаналізовано вплив точності прицілювання та швидкості реакції на результати кіберспортивних змагань. Встановлено, що гравці з високою точністю стрільби та швидкою реакцією отримують суттєву перевагу в динамічних ігрових ситуаціях, що безпосередньо відображається на підсумкових результатах матчів. Оглянуто та порівняно існуючі програмні засоби, призначені для тренування навичок точності прицілювання та швидкості реакції. Аналіз їхніх функцій і можливостей продемонстрував, що хоча такі програми забезпечують базове відпрацювання навичок, жодна з них

не охоплює повний спектр вимог, необхідних для всебічної підготовки кіберспортсменів. Таким чином, підтверджено актуальність розробки власного тренувального середовища, яке враховує виявлені недоліки існуючих рішень. На основі проведеного аналізу сформульовано функціональні вимоги до розроблюваного тренувального середовища та конкретизовано задачу його розробки. Ключові вимоги передбачають наявність різноманітних тренувальних режимів, інтегрованої системи збору й аналізу статистики результатів, зручного інтерфейсу користувача та можливості адаптації складності вправ під рівень гравця. Визначення цих вимог дозволило чітко окреслити цілі розробки системи та спрямувати проєктування на задоволення виявлених потреб. Відповідно до сформульованих вимог, спроектовано та реалізовано тренувальне середовище із використанням ігрового рушія Unity. Розроблене середовище має інтуїтивно зрозумілий інтерфейс користувача та містить інтегровану систему збору статистики, яка автоматично фіксує ключові показники точності й швидкості під час виконання вправ.

Перспективи подальшого розвитку проєкту включають розширення його функціоналу і впровадження нових можливостей. Зокрема, можна додати різноманітні режими та завдання для тренування інших навичок і адаптації тренажера до різних кіберспортивних дисциплін, впровадити багатокористувацький режим для колективних тренувань та онлайн-змагань, а також інтегрувати систему аналізу і моніторингу прогресу гравця. Окремим напрямом є застосування штучного інтелекту для адаптивного налаштування рівня складності тренувань під конкретного користувача. Таким чином, створений тренажер має великий потенціал для вдосконалення і може бути застосований у ширшому масштабі в майбутньому.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. State Space Labs, Inc. Aimlabs. State Space Labs, Inc., 2023. URL: <https://store.steampowered.com/app/714010/Aimlabs/> (дата звернення: 13.05.2025)
2. Kovaak Games. Kovaak's. Kovaak Games, 2018. URL: <https://store.steampowered.com/app/824270/Kovaaks/> (дата звернення: 13.05.2025)
3. Pixel Pointer Studios. Aimtastic. 2018. URL: <https://store.steampowered.com/app/757240/Aimtastic/> (дата звернення: 13.05.2025)
4. 3D Aim Trainer. 3D Aim Trainer. 3D Aim Trainer, 2021. URL: https://store.steampowered.com/app/1600530/3D_Aim_Trainer/ (дата звернення: 13.05.2025)
5. Учасники проєктів Вікімедіа. Шутер від першої особи – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Шутер_від_першої_особи. (дата звернення: 13.05.2025)
6. What is Reaction Time and Does It Make You Better at CS2 or VALORANT? URL: <https://insider-gaming.com/what-is-reaction-time-and-does-it-make-you-better-at-cs2-or-valorant/> (дата звернення: 13.05.2025)
7. Human Benchmark. *Human Benchmark*. URL: <https://humanbenchmark.com/tests/reactiontime>. (дата звернення: 13.05.2025)
8. Download 4K and HD Royalty-Free Stock Video | Pond5. *Pond5*. URL: <https://www.pond5.com> (дата звернення: 20.05.2025)
9. Unity Asset Store. *The Best Assets for Game Making | Unity Asset Store*. URL: <https://assetstore.unity.com/?srsltid=AfmBOoqxRlF1D6VEqyIzQDszV7od-Cs sweDkEmZd2gZ-vDPOEn8vA4mr> (дата звернення: 16.05.2025)

10. Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine. *Unity*. URL: <https://unity.com> (дата звернення: 16.05.2025)
11. Unity Learn. *Unity Learn*. URL: <https://learn.unity.com> (дата звернення: 20.05.2025)
12. Unity Hub – Manage Editor versions and collaborate with other creators. *Unity*. URL: <https://unity.com/unity-hub> (дата звернення: 16.05.2025)
13. Aim Trainer - Boost your skills in FPS games. *Aim Trainer - Boost your skills in FPS games*. URL: <https://aimtrainer.io> (дата звернення: 16.05.2025)

ДОДАТКИ

Додаток А

```
using UnityEngine;

public class PlayerController : MonoBehaviour
{
    [Header("Camera Settings")]
    [SerializeField] private Transform cameraHolder;
    [SerializeField] private float mouseSensitivity = 100f;
    [SerializeField] private float minVerticalAngle = -90f;
    [SerializeField] private float maxVerticalAngle = 90f;

    private float verticalLookRotation;

    void Start()
    {
        Cursor.lockState = CursorLockMode.Locked;
        Cursor.visible = false;
    }

    void Update()
    {
        // Отримуємо введення миші з урахуванням чутливості та часу кадру
```

```
float mouseX = Input.GetAxis("Mouse X") * mouseSensitivity *  
Time.deltaTime;  
float mouseY = Input.GetAxis("Mouse Y") * mouseSensitivity *  
Time.deltaTime;  
  
transform.Rotate(Vector3.up * mouseX);  
  
verticalLookRotation -= mouseY;  
verticalLookRotation = Mathf.Clamp(verticalLookRotation, minVerticalAngle,  
maxVerticalAngle);  
cameraHolder.localEulerAngles = new Vector3(verticalLookRotation, 0, 0);  
}  
}
```

Додаток Б

```
using UnityEngine;

public class Target : MonoBehaviour
{
    private void Start()
    {
        MoveToNewPosition();
    }

    public void Hit()
    {
        MoveToNewPosition();
    }

    private void MoveToNewPosition()
    {
        transform.position = TargetBounds.Instance.GetRandomPosition();
    }
}
```

Додаток В

```
using UnityEngine;
```

```
public class TargetShooter : MonoBehaviour
```

```
{
```

```
    [SerializeField] private Camera cam;
```

```
    [SerializeField] private LayerMask targetLayerMask;
```

```
    private void Update()
```

```
    {
```

```
        if (Input.GetMouseButtonDown(0))
```

```
        {
```

```
            ShootRay();
```

```
        }
```

```
    }
```

```
    private void ShootRay()
```

```
    {
```

```
        Ray ray = cam.ViewportPointToRay(new Vector3(0.5f, 0.5f));
```

```
        if (Physics.Raycast(ray, out RaycastHit hit, Mathf.Infinity, targetLayerMask))
```

```
        {
```

```
            if (hit.collider.TryGetComponent<Target>(out Target target))
```

```
            {
```

```
                target.Hit();
```

```
                GameManager.Instance.RegisterShot(true);
```

```
    }  
    else  
    {  
        GameManager.Instance.RegisterShot(false);  
    }  
}  
else  
{  
    GameManager.Instance.RegisterShot(false);  
}  
}  
}
```

Додаток Г

```
using UnityEngine;

public class TargetBounds : MonoBehaviour
{
    public static TargetBounds Instance { get; private set; }

    [SerializeField] private BoxCollider col;
    [SerializeField] private float minDistance = 2.0f;

    private Vector3 lastTargetPosition;
    private bool hasLastPosition;

    private void Awake()
    {
        if (Instance != null && Instance != this)
        {
            Destroy(gameObject); // Запобігаємо дублю
        }
        else
        {
            Instance = this;
        }
    }
}
```

```

public Vector3 GetRandomPosition()
{
    Vector3 center = col.center + transform.position;
    Vector3 size = col.size;

    Vector3 randomPosition;
    int maxAttempts = 10;
    int attempts = 0;

    do
    {
        float x = Random.Range(center.x - size.x / 2f, center.x + size.x / 2f);
        float y = Random.Range(center.y - size.y / 2f, center.y + size.y /
2f);
        float z = Random.Range(center.z - size.z / 2f, center.z + size.z /
2f);

        randomPosition = new Vector3(x, y, z);
        attempts++;

    } while (hasLastPosition && Vector3.Distance(randomPosition,
lastTargetPosition) < minDistance && attempts < maxAttempts);

    lastTargetPosition = randomPosition;
    hasLastPosition = true;

    return randomPosition;
}
private void OnDrawGizmosSelected()

```

```
{  
    if (col == null) return;  
    Gizmos.color = Color.green;  
    Gizmos.matrix = transform.localToWorldMatrix;  
    Gizmos.DrawWireCube(col.center, col.size);  
}  
}
```


Додаток Д

```
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class GameManager : MonoBehaviour
{
    public static GameManager Instance { get; private set; }

    [Header("UI Elements")]
    [SerializeField] private Text scoreText;
    [SerializeField] private Text accuracyText;
    [SerializeField] private Text timerText;
    [SerializeField] private GameObject endPanel;
    [SerializeField] private Text finalScoreText;

    [Header("Game Settings")]
    [SerializeField] private float gameDuration = 60f;

    [Header("Audio")]
    [SerializeField] private AudioClip hitSound;
    [SerializeField] private AudioClip missSound;
    [SerializeField] private AudioClip gameOverSound;
    private AudioSource audioSource;
```

```
private int score = 0;
private int shots = 0;
private float timer;
private bool gameActive = true;
```

```
private void Awake()
{
    if (Instance == null) Instance = this;
    else Destroy(gameObject);
}
```

```
private void Start()
{
    timer = gameDuration;
    audioSource = GetComponent();
    UpdateUI();
    endPanel.SetActive(false);
}
```

```
private void Update()
{
    if (!gameActive) return;
    timer -= Time.deltaTime;
    if (timer <= 0f)
    {
        EndGame();
    }
    UpdateTimerUI();
}
```

```
public void RegisterShot(bool hit)
{
    shots++;
    if (hit)
    {
        score++;
        audioSource.PlayOneShot(hitSound);
    }
    else
    {
        audioSource.PlayOneShot(missSound);
    }
    UpdateUI();
}

private void UpdateUI()
{
    scoreText.text = $"Score: {score}";
    float accuracy = shots > 0 ? (score / (float)shots) * 100f : 100f;
    accuracyText.text = $"Accuracy: {accuracy:F1}%";
}

private void UpdateTimerUI()
{
    int seconds = Mathf.CeilToInt(timer);
    timerText.text = $"Time: {seconds}s";
}
```

```
private void EndGame()
{
    gameActive = false;
    timerText.text = "Time: 0s";
    finalScoreText.text = $"Final Score: {score}";
    endPanel.SetActive(true);
    audioSource.PlayOneShot(gameOverSound);
}

public void RestartGame()
{
    SceneManager.LoadScene(SceneManager.GetActiveScene().name);
}
}
```