

**ШИФР: «engine\_real»**

Міжнародний конкурс студентських наукових робіт

**КОНКУРСНА НАУКОВО-ДОСЛІДНА РОБОТА СТУДЕНТА**

Спеціальність 015 – «Професійна освіта»

**на тему: «РОЗРОБКА СИМУЛЯТОРА ДЛЯ ПРОФЕСІЙНОЇ ПІДГОТОВКИ  
ПІЛОТІВ ДРОНІВ»**

**Анотація.** Розвиток сучасних цифрових технологій та програмного забезпечення для віртуальної та доповненої реальності дозволив використовувати обчислювальні потужності комп'ютерів для створення освітніх систем різного призначення. Завдяки цьому стало можливим демонструвати процеси, які важко або небезпечно відтворювати в умовах навчальних аудиторій. Особливо важливим напрямком стало створення навчальних тренажерів для підготовки цивільних і військових пілотів, водіїв автотранспорту, медичного персоналу, військовослужбовців тощо. Відповідне програмне забезпечення забезпечувало якісну підготовку фахівців при значно менших витратах. З появою та поширенням нового напрямку сучасної промисловості, пов'язаного з розробкою, виробництвом та використанням безпілотних літальних апаратів, постало питання якісної підготовки пілотів відповідного класу. Представлена робота представляє один з можливих підходів до вирішення питання створення доступного тренажера для підготовки пілотів безпілотників. Розробка необхідного програмного забезпечення проходила в наступні етапи: визначення концепції проекту, створення тривимірного контенту та програмування навчального додатку. Основними засобами розробки запропонованого авторами проекту стали вільно розповсюджені програмні пакети. Зокрема, для створення тривимірного контенту використовувався 3D-редактор Blender, а середовищем, в якому створювався сам навчальний тренажер, став рушій Unreal Engine 5. Серед подальших напрямків дослідження та вдосконалення створеного програмного продукту автори вбачають розробку системи тестування якості представленого тренажера, вдосконалення його функцій, а також адаптацію технології до можливості використання з системою віртуальної реальності.

**Ключові слова:** тренажер, підготовка пілотів безпілотників, проектування навчальних комп'ютерних тренажерів, 3D-графіка, 3D-редактор Blender, ігровий рушій, Unreal Engine

## **ЗМІСТ**

ВСТУП.....	4
РОЗДІЛ 1. АНАЛІЗ ВИМОГ ПРОЕКТУ ТА ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, НЕОБХІДНОГО ДЛЯ РОЗРОБКИ. ....	7
1.1. Основні вимоги до проекту .....	7
1.2. Огляд основних інструментів розробки .....	8
ГЛАВА 2. ПРОЦЕС СТВОРЕННЯ ТРЕНАЖЕРА-СИМУЛЯТОРА.....	11
2.1. Створення графічного контенту для тренажера симулятора. ....	11
2.2. Огляд функціональних можливостей UE5 .....	13
2.3. Моделювання польоту дрона та оцінка результатів .....	14
ВИСНОВОК .....	17
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	18

## ВСТУП

**Актуальність.** У наш час тренажери для навчання стали активно використовуються лікарями, військовими, рятувальниками, водіями та багатьма іншими фахівцями із різних сфер життєдіяльності сучасного суспільства. Завдяки симуляторам-тренажерам (спеціалізованим моделям, які імітують роботу існуючих або створюваних систем) люди, які тренуються, отримують дані для прийняття рішень. Це дозволяє тестувати різні сценарії взаємодії користувача із системою.

Загалом комп'ютерні симуляції можна використовувати для налаштування продуктивності і оптимізації процесів, підвищення безпеки, теоретичного тестування, навчання персоналу тощо. Також такі системи дозволяють фахівцям отримати уявлення про вплив різних факторів при відпрацюванні певних дій.

Симуляції також можуть бути використані, коли фактична система недоступна або занадто небезпечна для оцінки, або коли система все ще знаходиться на стадії проектування чи теорії.

Ключем до будь-якої симуляції є інформація, яка використовується для побудови імітаційної моделі. Вона може удосконалюватися на основі нових даних, які виявляються при перегляді навчальних протоколів або вивчення досягнень наукової думки.

Досить поширеними є такі види симуляції.

### *1. Моделювання дискретної події*

Моделювання системи в міру її просування в часі: заводські операції (токарна обробка, штампування, фрезерування); аналіз трафіку (черги, дороги, мережі).

### *2. Динамічне моделювання*

Моделювання системи під час її руху в просторі: кінематика машини; ергономіка людини; аеродинамічні випробування; віртуальне прототипування.

### *3. Моделювання процесу*

Моделювання фізичних взаємодій між двома або більше системами: моделювання продукту в процесі експлуатації; внутрішньовиробниче моделювання виробів; прогноз погоди.

Активно використовуються тренажери-симулятори і при навчанні пілотів, зокрема і пілотів дронів.

**Аналіз останніх досліджень і публікацій.** Використання програмного забезпечення в навчальних цілях фактично відбувалося паралельно з розвитком обчислювальної техніки та прогресом у сфері комп'ютерної графіки. Значний поштовх у створенні комп'ютерних тренажерів отримав завдяки створенню мікропроцесорів. На їх основі активно створювалися тренажери для підготовки пілотів літаків. Так, у 1986 р. Сміт П. аналізує сучасні теорії, дослідження та практику використання навчальних тренажерів. [8]. Серед військових особливої популярності набули програми-симулятори, зокрема група авторів Джонс Р. М., Лерд Дж. Е., Нільсен П. Е., Култер К. Дж., Кенні П. та Косс Ф. В. описують використання інтелектуального комплексу TACAIR-SOAR [4]. Також в авіації для підготовки пілотів літальних апаратів активно застосовували спеціалізоване програмне забезпечення та спеціально створені навчальні комплекси. Зокрема, Gu, H., Wu, D., Liu, H. розкривають особливості створення недорогого авіасимулятора для задоволення потреб авіакомпаній у навчанні пілотів. [3], у той час як Шашидхара, Б. П., Чандрасекаран, Р., Бхатія, Й., Магеш, Г., Бінешкумар, К., Кумар, Х. В. зосередилися на описі концепції симулятора бойових місії (FMS) для пілотів винищувачів [7].

Відповідне програмне забезпечення також є важливим для підготовки професійних водіїв автомобільного транспорту. Зокрема, цій темі присвятили свої праці такі вчені: Влаквельд В.П. [9], Де Вінтер, Дж., Ван Левен, П. М., Хаппі, Р. [2], Сетрен, Г. Б., Біркеланд, Т. Ф., Педерсен, П. А., Ліндгейм, К., Скогстад, М. Р. [6].

Сьогодні значної популярності набули літаючі дрони різних форм-факторів (від літаків, побудованих за класичними аеродинамічними схемами, до коптерів різного формату). В результаті постала проблема максимально ефективної підготовки пілотів даного типу літальних апаратів.

Зокрема, Albeaino, G., Eiris, R., Gheisari, M., Issa, R.R. досліджують можливості симулятора DronSim, створеного на основі технологій віртуальної реальності. [1]. Група дослідників Рібейро, Р., Рамос, Дж., Сафадіньо, Д., Рейс, А., Рабадао, К.,

Баррозу, Дж., Перейра, А. запропонували багатоплатформне хмарне рішення Web AR для навчання пілотів безпілота [5].

Таким чином, констатуємо, що короткий аналіз наукових досліджень, пов'язаних зі створенням та використанням навчальних тренажерів у різних галузях, свідчить про актуальність обраної теми дослідження.

У процесі аналізу наукової літератури та відкритих джерел інформації, що стосуються використання тренажерів, визначено **мету** наукового поректу: створення зручного тренажера з високим рівнем деталізації для підготовки пілотів безпілотних літальних апаратів.

Серед основних завдань варто виділити такі:

1. Сформулювати базові вимоги до проекту.
2. Описати програмне забезпечення, необхідне для реалізації навчального симулятора.
3. Розкрити основні етапи розробки тренажера для професійної підготовки пілотів дронів.

Таким чином, **об'єктом** дослідження є програмне забезпечення для моделювання польоту літальних апаратів, а **предметом** – навчальний тренажер для підготовки пілотів безпілота.

Для реалізації поставленої мети були використані **загальнонаукові та теоретичні методи**: аналіз інформації, отриманої з відкритих наукових джерел та літератури щодо створення та використання навчальних програмних тренажерів; порівняння результатів та узагальнення досвіду фахівців з даної теми; синтез; моделювання навчальних ситуацій.

# РОЗДІЛ 1. АНАЛІЗ ВИМОГ ПРОЕКТУ ТА ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, НЕОБХІДНОГО ДЛЯ РОЗРОБКИ.

## 1.1. Основні вимоги до проекту

Навчальні тренажери-симулятори є складними системами, які вимагають вивірених підходів до проектування та розробки необхідного програмного забезпечення. При розробці таких рішень важливо чітко вирішити ряд питань:

- Визначитися із завданнями, які повинен вирішувати тренажер. Це може бути суто навчання (передача знань, умінь, навичок), а може бути контроль над досвідом та удосконалення навичок фахівця.

- Спроекувати архітектуру застосунку симулятора.

- Дослідити уміння та навички, які формуватиме створюваний навчальний симулятор.

- Розробити дизайн інтерфейсу. Інтерфейс повинен бути не грайливим, а наближеним до реального середовища.

- Алгоритмізувати всі процеси: і технологічні, і інтелектуальні, і навіть емоційні.

- Забезпечте незалежність від платформи та працюйте на різних пристроях.

Спираючись на ці пункти опишемо коротко вимоги до створюваного програмного продукту. Оскільки створення літаючих дронів (літаючі апарати активно використовуються при ліквідації надзвичайних ситуацій, військовій справі, транспортуванні тощо) є одним із основних трендів сучасності, то професійна підготовка пілотів таких апаратів є важливим напрямом професійної освіти.

Тренажер має забезпечити формування навичок пілотування певного виду дрону. На початковому етапі проекту таке навчання стосуватиметься найпопулярнішої моделі безпілотного літального апарата – квадрокоптера.

Щоб гарантувати, що пілот набуде необхідні знання необхідно завчасно сформувати набір завдань, які виконувати людини при роботі на тренажері. До таких вправ варто віднести такі: зліт безпілотника; переміщення дрона вгору-вниз; переміщення дрона вбік; переміщення дрона вперед і назад; переміщення по серії кіл;

цільова практика; посадка безпілотної. Керування дроном має відбуватися через контролер. Розкриємо вправи детальніше.

*Зліт безпілотної.* Це перший крок, який повинен навчитися будь-який майбутній пілот безпілотної. За допомогою контролера пілот повинен підняти дрон над землею.

*Переміщення дрона вгору і вниз.* Використовуючи контролер, пілот має навчитися контролювати рівень дрона над віссю Z.

*Переміщення дрона вліво-вправо.* Використовуючи контролер, пілот повинен навчитися переміщувати дрон по горизонталі *вліво-вправо*.

*Рух вперед і назад.* Використовуючи контролер, пілот повинен навчитися переміщувати дрон по горизонталі вперед і назад.

*Переміщення по серії кіл.* Використовуючи всі попередні прийоми, пілот повинен змусити безпілотної слідувати за певним маршрутом, уникаючи перешкоди. Маршрут задається серією кіл.

Виконання майбутнім пілотом усіх завдань гарантує швидкий і легкий перехід від керування дроном у тренажері симуляторі до керування дроном у реальній ситуації.

## **1.2. Огляд основних інструментів розробки**

Загалом створення навчальних тренажерів – це багатогранний і складний процес, який складається з багатьох етапів. Кожен з них вимагає детального опису. Однак найважливішими етапами є створення візуального контенту та програмування логіки навчального тренажера. Тому ми зосередимося саме на цих етапах процесу розробки.

Спочатку опишемо програмне забезпечення, яке використовувалося для розробки. Основним редактором 3D-графіки для створення просторових віртуальних 3D-моделей був Blender.

Blender 3D був розроблений як робочий інструмент голландською анімаційною студією NeoGeo. У червні 1998 року творець Blender, Тон Розендал, заснував Not a Number для подальшого розвитку та підтримки Blender. 18 липня 2002 року розпочалася програма збору спонсорських пожертвувань для покриття необхідної



суми. Вже 7 вересня 2002 року було оголошено про те, що необхідну суму зібрано, і найближчим часом планується перевести вихідний код і сам Blender під ліцензію GPL. 13 жовтня 2002 року фонд Blender Foundation випустив продукт під ліцензією GNU GPL. Наразі Blender є проектом з відкритим вихідним кодом і розробляється за активної підтримки Blender Foundation. Виходячи з дати створення перших файлів вихідного коду, 2 січня 1994 року вважається днем народження Blender. У 2023 році Блендеру виповнилося 29 років.

Blender є безкоштовним програмним забезпеченням, на відміну від інших 3D-середовищ. Ця програма має можливість створювати тривимірні моделі та програмувати скрипти на мові програмування Python, і, при цьому, отримувати дійсно якісні результати. Blender – це програма для створення та редагування тривимірної графіки, візуалізації, анімації, створення комп'ютерних ігор і навіть скульптингу.

Основні функції програми Blender:

*3D моделювання.* Blender містить всі існуючі засоби для створення і редагування тривимірних моделей (створення об'єктів на основі примітивів, багатокутників, кривих NURBS, кривих Без'є, метасфер, виконання логічних операцій на просторовими об'єктами тощо). Програма пропонує велику кількість різних модифікаторів для удосконалення моделей.

*Анімація.* У розпорядженні користувача присутні такі інструменти, як такелаж (скелетна анімація), інверсна кінематика, деформація сітки, обмеження, анімація ключових кадрів, редагування ваг вершин тощо. Реалізована динаміка твердих і м'яких тіл, а також анімація частинок.

*Набори текстур і шейдерів.* Програма дозволяє застосовувати декілька текстур до одного об'єкта та оснащена багатьма інструментами текстурування, включаючи інструменти для створення UV-карт. Blender містить якісні інструменти настроюваних шейдерів, що додають гнучкості в роботі з матеріалами.

Таким чином Blender є доступним і водночас професійним програмним засобом для створення та анімацій тривимірного графічного контенту для тренажера-симулятора навчання пілотів дронів.

Розробка навчального симулятора проводилася з використанням ігрового рушія Unreal Engine 5, який також є програмним забезпеченням, що вільно поширюється. Воно дозволяє інтегрувати весь графічний контент, створений у Blender, запрограмувати взаємодії між об'єктами у віртуальному світі, створювати високореалістичне та динамічне візуальне представлення та програмувати логіку ігрового симулятора.

Використовуючи базові інструменти та модифікатори Blender та Unreal Engine 5 було створено основну сцену проекту. Рельєф розроблено за допомогою інструменту «Пейзаж» за допомогою Unreal Engine 5, а інші об'єкти, такі як дрон, паркан, фонтан, трава та будинок, були створені за допомогою полігонального моделювання в Blender.

## ГЛАВА 2. ПРОЦЕС СТВОРЕННЯ ТРЕНАЖЕРА-СИМУЛЯТОРА.

### 2.1. Створення графічного контенту для тренажера симулятора.

Створення моделі варто розпочати з ескізування основних компонентів тривимірної сцени. Розберемо важливі кроки на прикладі тривимірної моделі дрона-квадрокоптера

Розмістивши куб у потрібному місці, завантажуюмо файли двовимірних ескізів дрона в Blender для того розуміти етапи моделювання.

#### *1. Створення базової форми моделі*

За допомогою базових форм формуємо ключові для сприйняття елементи цифрової моделі дрону. Спочатку використовуємо низький рівень деталізації. Далі поступово збільшували деталізацію. Тут доречно застосовувати різні модифікатори, які дозволяють автоматизувати та прискорити формування зовнішнього вигляду дрону. Зокрема таким модифікатором є Subdivision Surface, який дозволяє збільшити кількість полігонів і не спотворювати саму полігональну сітку.

#### *2. Створення матеріалів і текстурування моделі*

Реалістичну модель дрона не можна створити, спираючись суто на цифрову полігональну структуру. Для якісної візуалізації необхідно розробити максимально наближені до реальності матеріали та текстури. Це дозволяє передавати всі складні елементи поверхні без створення додаткової геометрії на, власне, самій поверхні.

Для якісного накладання текстурованих зображень важливим є також створення UV – розгортки. Такий підхід забезпечує точне позиціонування власне самих текстур на поверхні, а також спеціалізованих карт нерівностей, нормалей, бліків тощо.

#### *3. Анімація персонажа*

Анімація об'єкта не є обов'язковим кроком для статичних моделей і сцен, але для ігор і фільмів це важливо. У нашому випадку анімація була створена за допомогою Blueprints в Unreal Engine 5.

*4. Експорт моделей у формати, зручні для використання у рушії Unreal Engine 5.*

Експорт є дуже важливим кроком у процесі створення 3D-моделі. Зазвичай 3D-художники використовують величезну кількість програмного забезпечення, що спеціалізується на одній конкретній речі. Модель потрібно об'єднати, зберегти анімацію та вибрати правильний формат експорту. Це, фактично, основні кроки для експорту моделей у будь-яке інше візуальне програмне забезпечення.

На рисунку 1 представлені візуалізації основних тривимірних моделей для тренажера-симулятора.

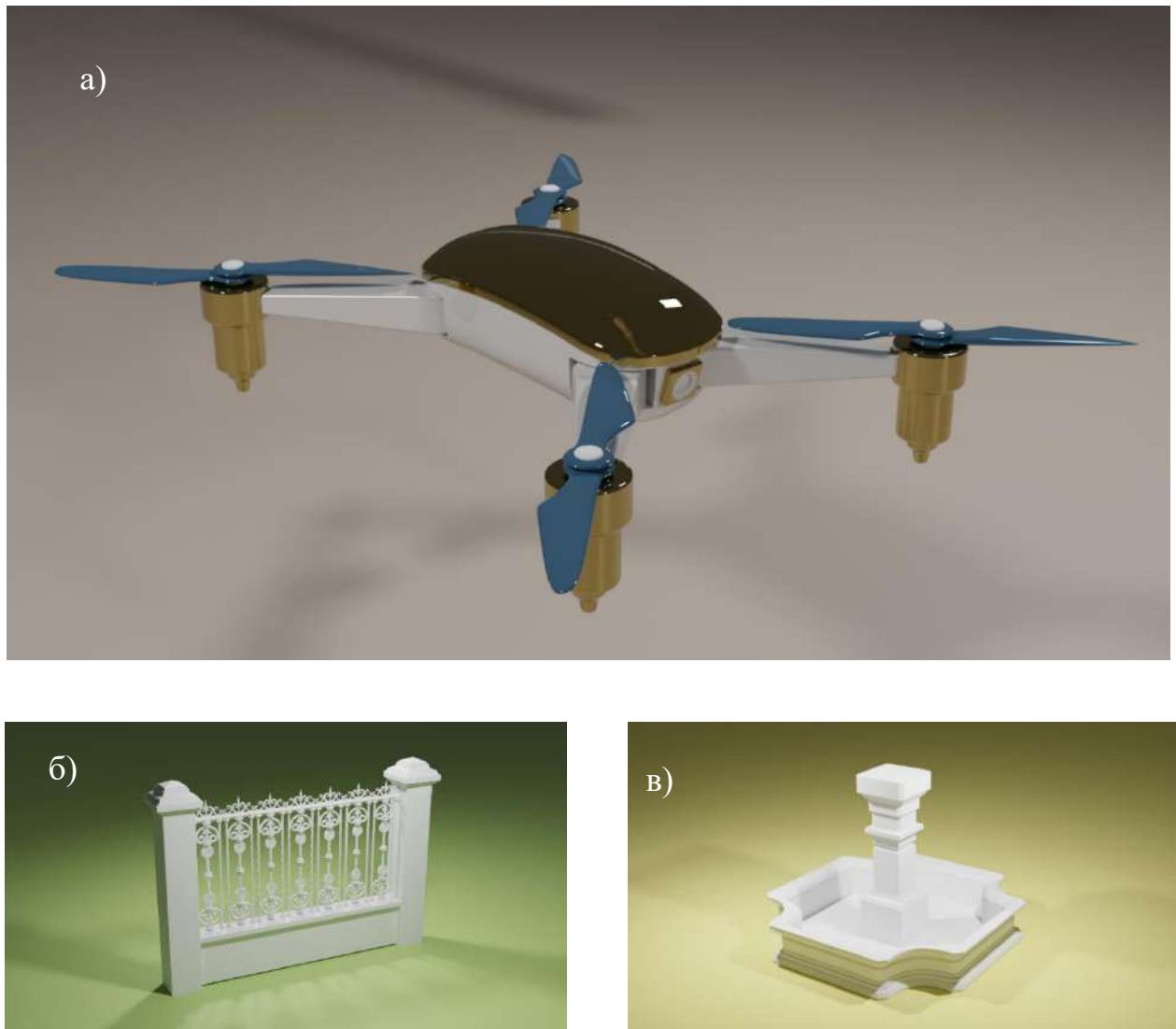


Рис. 1. Оптимізована ігрова модель 3D моделі  
а – Модель дрона, б – Модель паркану, в – Модель фонтану

## 2.2. Огляд функціональних можливостей UE5

Навчальний симулятор був створений з використанням Unreal Engine 5, C++ використовувався як мови програмування для програмування всіх функцій симулятора загалом, а Blueprints Unreal Engine використовувалися для реалізації керування самим дроном.

Моделювання керування дроном стало можливим завдяки використанню системи сутності компонентів Unreal Engine. В Unreal Engine існує 4 основних класи (рис. 2): «об'єкт», «актор», «пішак», «персонаж». Одне успадковує інше. Умовний «персонаж» володіє «пішаком», який також є «актором», який у той же час є «об'єктом». Умовне «керування» — це процес, коли персонаж бере контроль над «пішаком». У симуляторі присутні три основних класи, режим гри – «симуляція», персонаж – «пілот» і пішак – «дрон».

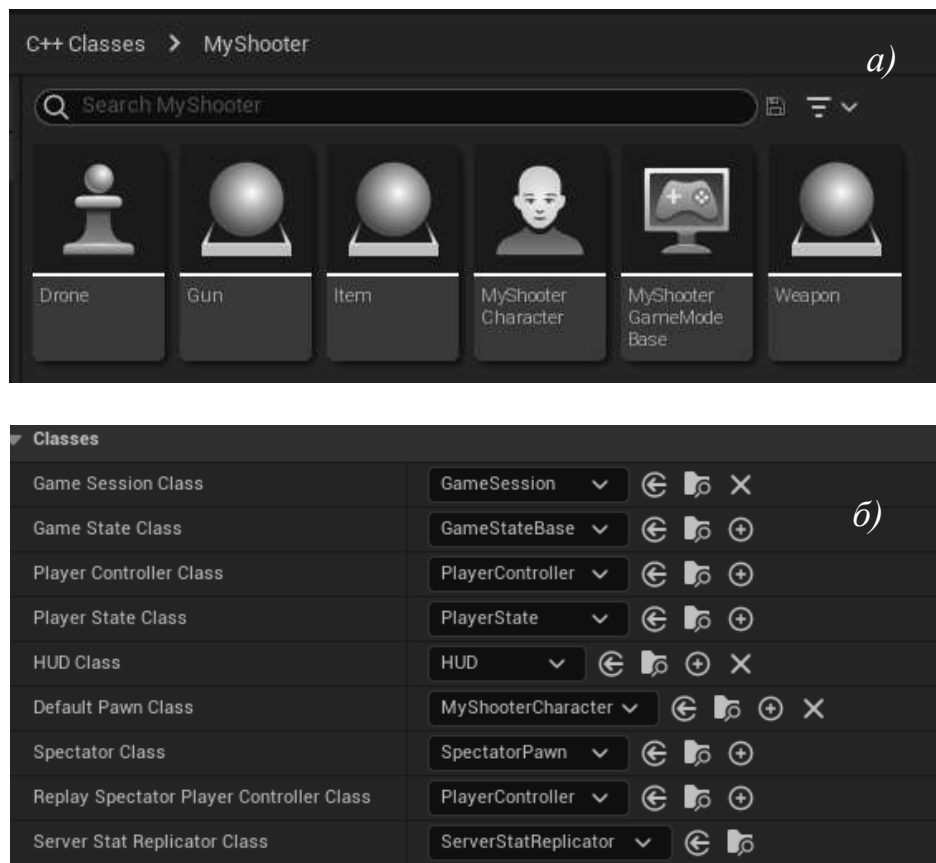


Рис 2. Система класів Unreal Engine 5

а – C++ класи, б – Blueprint класи

Опишемо базові класи для створюваного тренажера-симулятора. HUD Class – інтерфейс, який буде переглядатися під час симуляції. Клас «пішака» за

замовчуванням – персонаж, який буде породжений у віртуальному світі, у випадку симуляції це пілот. Клас контролера гравця – визначає, який клас контролера використовуватиметься для симуляції введення в грі.

Клас персонажа — Пілот, цей клас має функції для пересування, використання предметів та керування дроном. Клас заголовка визначає всі компоненти та функції, які має клас символів. Компонентами класу символів є: `UCharacterController`, `UMeshComponent`, `USpringArmComponent`, `UCameraComponent`. Вони використовуються для переміщення персонажа, бачення персонажа та наявності камери, яка стежить за персонажем. Функції класу персонажів: `MoveForward`, `MoveRight`, `TurnAtRate`, `LookUpAtRate`, `TraceUnderCrosshairs`, `FireWeapon`, `ToggleDrone`. Вони контролюють рух персонажа, використання предметів і зброї, володіння дроном. Властивості класу символів: `AGun`, `ADrone`, `isOnDrone`. Вони дозволяють персонажу посилатися на предмети та дрони, ця архітектура дозволяє персонажу посилатися на будь-який клас, який є дроном, що означає, що він не прив'язаний до одного конкретного класу.

Клас «пішак» містить функції для пересування, перемикання камери та захоплення предметів. Клас заголовка визначає всі компоненти та функції, які має клас пішака. Компоненти класу пішака: `UFloatingPawnMovement`, `UMeshComponent`, `USceneCameraComponent`. Вони використовуються для імітації руху дрона, малювання тіні дрона та відображення двох можливих видів дрона: один із задньої частини дрона, а інший діє як імітація камери дрона. Функції `Drone Class`: `MoveRight`, `MoveForward`, `MoveUp`, `SwitchView`, `DisableDrone` використовуються для того, щоб безпілотник міг рухатися в усіх напрямках, міг переключати огляд з першої камери на другу, а також щоб вимкнути дрон, щоб повернутися до керування пілотом.

### **2.3. Моделювання польоту дрона та оцінка результатів**

Симуляція польоту дрона виконується за допомогою компонента `UFloatingPawnMovement`, який є вбудованим компонентом `Unreal Engine 5`, він має функції для руху та змінні (табл. 1) для керування симуляцією руху.

Клас `C++`, який є батьківським для дрона, використовує в основному цей компонент для імітації польоту. Він також змінює базові змінні

UFloatingPawnMovement наступним чином: Acceleration встановлено на 2000.0f, Deceleration встановлено на 500.0f, Turning boost на 1.0f, MaxSpeed на 300.0f. Це дозволяє відтворити те, як дрон у реальному житті поводить себе в повітрі. У розробленому класі мовою C++ також є змінні для: імітації швидкості вітру, що може сповільнити чи збільшити швидкість керування дроном; маси дрона, яка вказує на вагу дрона, що також змінює всі вбудовані змінні UFloatingPawnMovement відповідно до того, наскільки важким чи легким є дрон. Термін служби батареї, додає моделювання роботи батареї в безпілотник, коли розряджається, симуляція повідомить майбутньому пілоту про невдачу, і йому доведеться спробувати ще раз.

Table 2.1. Змінник

Маркування	Вид	Наіменування	Опис
$V_a$	float	Acceleration	Прискорення, що подається на вхід (швидкість зміни швидкості)
$V_c$	<u>uint32: 1</u>	<u>bPositionCorrected</u>	Встановлюється в true, коли застосовується корекція положення
$V_d$	float	Deceleration	Уповільнення, що застосовується за відсутності входу (швидкість зміни швидкості)
$V_m$	float	MaxSpeed	Максимальне значення швидкості, дозволене для контрольованого пішака.
$V_t$	float	<u>TurningBoost</u>	Налаштування впливає на додаткове зусилля, що застосовується при зміні напрямку руху, завдяки чому

Маркування	Вид	Найменування	Опис
			при виконанні поворотів зменшується занос і підвищується чутливість.

Створена симуляція дрона дозволяє «відчути» поведінку дрона в реальному світі, успішне виконання завдань симуляції в межах часу роботи батареї дрона, готує майбутніх пілотів до пілотування справжніх дронів. Симуляція враховує швидкість вітру та масу дрона, щоб краще відтворити поведінку дрона в реальному житті. У майбутньому до симуляції можна буде додати можливість враховувати дощ, сніг та інші можливі погодні умови.



## ВИСНОВОК

Сучасний розвиток цифрових технологій дає можливість створювати освітні програмні засоби, які сприяють якісному оновленню освітнього процесу та значно поліпшують підготовку фахівців різних спеціальностей. Аналіз наукової літератури та відкритих джерел інформації підтвердив актуальність обраної теми дослідження.

Представлений тренажер-симулятор спрямований на покращення процесу підготовки пілотів дронів. Він вирішує низку важливих питань:

- використання симулятора наближає учасників навчального процесу до реальних умов використання безпілотного літального апарату;
- використання зручного та доступного програмного забезпечення зроблює навчання масовим та знижує його вартість;
- варіювання завдань та умов виконання дозволяє вивчити значну кількість типів завдань та умов їх виконання, що лише сприяє свідомості та готовності пілота;
- продумана система оцінювання дозволяє відібрати кращих, які будуть продовжувати вивчати пілотування на реальних пристроях.

Серед подальших напрямів розвитку вказаного проекту визначаємо наступні: ретельне тестування вже створеної програми з метою виявлення потенційних проблемних ситуацій та їх усунення; створення більш детальних та об'ємних віртуальних карт світів, що дозволить пілоту вивчити різноманітні завдання; вдосконалення системи оцінювання досягнень тощо.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Albeaino, G., Eiris, R., Gheisari, M., & Issa, R. R. (2022). DroneSim: A VR-based flight training simulator for drone-mediated building inspections. *Construction Innovation*, 22(4), 831-848. DOI:<https://doi.org/10.1108/CI-03-2021-0049>.
2. De Winter, J., Van Leeuwen, P. M., and Happee, R. (2012, August). Advantages and disadvantages of driving simulators: A discussion. *Proceedings of measuring behavior*, 2012, pp. 28-31.
3. Gu, H., Wu, D., and Liu, H. (2009). Development of a novel low-cost flight simulator for pilot training. *International Journal of Computer and Systems Engineering*, 3(12), 1581-1585.
4. Jones, R. M., Laird, J. E., Nielsen, P. E., Coulter, K. J., Kenny, P., and Koss, F. V. (1999). Automated intelligent pilots for combat flight simulation. *AI magazine*, 20(1), 27-27. DOI: <https://doi.org/10.1609/aimag.v20i1.1438>.
5. Ribeiro, R., Ramos, J., Safadinho, D., Reis, A., Rabadão, C., Barroso, J., and Pereira, A. (2021). Web AR solution for UAV pilot training and usability testing. *Sensors*, 21(4), 1456. DOI: <https://doi.org/10.3390/s21041456>
6. Sætren, G. B., Birkeland, T. F., Pedersen, P. A., Lindheim, C., and Skogstad, M. R. (2019). Opportunities and limitations in use of simulators in driver training in Norway. A qualitative study. In *Proceedings of the 29th European Safety and Reliability Conference (ESREL). 22–26 September 2019 Hannover, Germany*. DOI: [https://doi.org/10.3850/978-981-11-2724-3\\_0216-cd](https://doi.org/10.3850/978-981-11-2724-3_0216-cd).
7. Shashidhara, B. P., Chandrasekaran, R., Bhatia, Y., Magesh, G., Bineshkumar, K., and Kumar, H. V. (2018). Development of a Full Mission Simulator for Pilot Training of Fighter Aircraft. *Defence Science Journal*, 68(5). DOI : <https://doi.org/10.14429/dsj.68.12235>.
8. Smith, P. (1986). Instructional simulation: Research, theory, and a case study.
9. Vlakveld, W. P. (2005). The use of simulators in basic driver training. In *Humanist TFG Workshop on the Application of New Technologies to Driver Training, Brno, Czech Republic*. Available at: [www.escope.info/download/research\\_and\\_development/HUMANISTA\\_13Use.pdf](http://www.escope.info/download/research_and_development/HUMANISTA_13Use.pdf).

**CODE:** «engine\_real»

International competition of student research papers

**STUDENTS RESEARCH COMPETITION PROJECT**

Specialty 015 – « Professional education »

**title: «DEVELOPMENT OF A SIMULATOR FOR PROFESSIONAL  
TRAINING DRONE PILOTS»**

**Abstract.** The development of modern digital technologies and software for virtual and augmented reality has made it possible to use the computing power of computers to create educational systems for various purposes. Thanks to this, it became possible to demonstrate processes that are difficult or dangerous to reproduce in the conditions of educational classrooms. The creation of training simulators for the training of civil and military pilots, motor transport drivers, medical personnel, military personnel, etc., became a particularly important direction. Appropriate software ensured high-quality training of specialists at significantly lower costs. With the emergence and spread of a new direction of modern industry related to the design, production, and use of unmanned aerial vehicles, the question of high-quality training of pilots of the appropriate class arose. The presented work represents one of the possible approaches to solving the issue of creating an affordable training simulator for training drone pilots. The development of the necessary software took place in the following stages: definition of the project concept, creation of three-dimensional content, and programming of the educational application. The main means of development of the project proposed by the authors have freely distributed software packages. In particular, the 3D editor Blender was used to create three-dimensional content, and the environment in which the training simulator itself was created was the Unreal Engine 5 engine. Among the further areas of research and improvement of the created software product, the authors see the development of a system for testing the quality of the presented training simulator, improvement of its functions, and adaptation technology to the possibility of use with a virtual reality system.

**Keywords:** training simulator, drone pilot training, design of training computer simulators, 3D graphics, Blender 3D editor, game engine, Unreal Engine

## **CONTENT**

INTRODUCTION .....	4
CHAPTER 1. PROJECT REQUIREMENTS ANALYSIS AND DESCRIPTION OF REQUIRED SOFTWARE FOR DEVELOPMENT.....	7
1.1. Basic project requirements.....	7
1.2. An overview of the main development tools.....	8
CHAPTER 2. CREATING A TRAINING SIMULATOR PROCESS. ....	10
2.1. Creating graphic content for simulator training.....	10
2.2. Functionality overview UE5 .....	12
2.3. The simulation of drone flight, and assessment of results.....	13
CONCLUSIONS .....	16
REFERENCES .....	17

## INTRODUCTION

**The relevance of the topic.** In our time, training simulators have become actively used by doctors, military personnel, rescuers, drivers, and many other professionals in various spheres of modern society. Thanks to training simulators (specialized models that simulate the operation of existing or created systems), people who train can obtain data for decision-making. This allows testing various scenarios of interaction between the user and the system.

Computer simulations can be used in general for tuning productivity and optimizing processes, increasing safety, theoretical testing, personnel training, and more. Such systems also allow experts to gain an understanding of the impact of various factors when working through certain actions.

Simulation can also be used when the actual system is not available or too dangerous to evaluate, or when the system is still at the design or theory stage.

The key to any simulation is the information that is used to build the simulation model. It can be improved based on new data that emerges during the review of training protocols or the study of scientific achievements.

The following types of simulation are quite common.

### *1. Discrete Event Simulation*

Modelling a system as it progresses through time: factory operations (turning, stamping, milling); traffic analysis (queues, roads, networks).

### *2. Dynamic Simulation*

Modelling a system as it progresses through space: machine kinematics; human ergonomics; aerodynamic testing; virtual prototyping.

### *3. Process Simulation*

Modelling physical interactions between two or more systems: in-service product modelling; in-manufacture product modelling; weather forecasting.

Training simulators are also actively used in the training of pilots, including drone pilots.

**Analysis of recent studies and publications.** The use of software for educational

purposes actually occurred parallel to the development of computing technology and progress in the field of computer graphics. A significant boost in the creation of computer simulators was given thanks to the creation of microprocessors. On their basis, training simulators for training aircraft pilots were actively created. Thus, in 1986, Smith P. analyzes contemporary theories, research and practice of using educational simulators [8]. Simulator programs have gained particular popularity among the military, in particular, the group of authors Jones, R. M., Laird, J. E., Nielsen, P. E., Coulter, K. J., Kenny, P. and Koss, F. V. describe the use of the TACAIR-SOAR intelligent system [4]. Also, specialized software and specially created training complexes were actively used in aviation to train aircraft pilots. In particular, Gu, H., Wu, D., Liu, H. reveal the features of creating a low-cost flight simulator to meet the pilot training needs of airlines [3], while Shashidhara, B. P., Chandrasekaran, R., Bhatia, Y., Magesh, G., Bineshkumar, K., Kumar, H. V. focused on describing the concept of a full mission simulator (FMS) for fighter pilots [7].

Appropriate software is also important for the training of professional road transport drivers. In particular, the following scientists devoted their works to this topic: Vlakveld W. P. [9], De Winter, J., Van Leeuwen, P. M., Happee, R. [2], Sætren, G. B., Birkeland, T. F., Pedersen, P. A., Lindheim, C., Skogstad, M. R. [6].

Today, flying drones of various form factors have gained considerable popularity (from aircraft built according to the classic aerodynamic schemes of airplanes to copters of various formats). As a result, the problem of training pilots of this type of aircraft as efficiently as possible has arisen.

In particular, Albeaino, G., Eiris, R., Gheisari, M., Issa, R.R. investigate the capabilities of the DronSim simulator, created on the basis of virtual reality technologies [1]. A group of researchers Ribeiro, R., Ramos, J., Safadinho, D., Reis, A., Rabadão, C., Barroso, J., Pereira, A. proposed a multi-platform cloud-based Web AR solution for drone pilot training [5].

Thus, we state that the brief analysis of scientific research related to the creation and use of educational simulators in various fields proves the relevance of the chosen research topic.

In the process of analyzing scientific literature and open sources of information related to the use of training simulators, the **goal** of scientific project was determined: the creation of a convenient training simulator with a high level of detail for training pilots of unmanned aerial vehicles.

Among the main tasks, the following should be highlighted:

1. Formulating basic requirements for the project.
2. Describing the necessary software for implementing a training simulator.
3. Expanding on the main stages of developing a simulator for professional training of drone pilots.

Thus, the **object** of research is software for simulating the flight of aircraft, and the **subject** is a training simulator for training drone pilots.

To realize the set goal, **general scientific and theoretical methods** were used: analysis of information obtained from open scientific sources and literature on the creation and use of educational software simulators; comparison of results and generalization of the experience of specialists in this subject; synthesis; simulation of educational situations.



# CHAPTER 1. PROJECT REQUIREMENTS ANALYSIS AND DESCRIPTION OF REQUIRED SOFTWARE FOR DEVELOPMENT.

## 1.1. Basic project requirements.

Educational simulators are complex systems that require well-thought-out approaches to design and development of the necessary software. When developing such solutions, it is important to clearly address a number of issues:

- To determine the tasks that the simulator should solve. This can be purely training (knowledge, skills, and abilities transfer), or it can be control over the experience and improvement of the skills of a specialist.

- To design the architecture of the simulator application.

- To investigate the skills and competencies that will be developed by the created educational simulator.

- To design the interface. The interface should not be playful, but close to the environment.

- Algorithmize all processes: technological, intellectual, and even emotional.

- Ensure platform independence and run on different devices.

Based on these points, let's briefly describe the requirements for the created software product. Since the creation of drones (flying vehicles that are actively used in emergency situations, military affairs, transportation, etc.) is one of the main trends of modern times, professional training of pilots for such vehicles is an important direction of professional education.

The simulator should provide the formation of skills for piloting a certain type of drone. At the initial stage of the project, such training will relate to the most popular model of unmanned aerial vehicle - quadcopter.

To ensure that the pilot acquires the necessary knowledge, it is necessary to form a set of tasks in advance that people will perform while working on the simulator. Such exercises should include the following: takeoff of the drone; moving the drone up and down; moving the drone sideways; moving the drone forward and backward; moving in a series of

circles; target practice; landing of the drone. Drone control should be performed through a controller. Let's discuss these exercises in more detail.

*Takeoff of the drone.* This is the first step that any future drone pilot should learn. Using the controller, the pilot must lift the drone off the ground.

*Moving the drone up and down.* Using the controller, the pilot must learn to control the level of the drone above the Z-axis.

*Moving the drone left and right.* Using the controller, the pilot must learn to move the drone horizontally from left to right.

*Moving forward and backward.* Using the controller, the pilot must learn to move the drone horizontally forward and backward.

Moving in a series of circles. Using all the previous techniques, the pilot must guide the drone to follow a specific path while avoiding obstacles. A series of circles define the way.

Completing all the tasks by the future pilot guarantees a fast and easy transition from controlling the drone in a simulator trainer to controlling it in a real-life situation.

## **1.2. An overview of the main development tools.**

In general, creating educational simulators is a multifaceted and complex process that consists of many stages. Each of them requires a detailed description. However, the most important stages are creating visual content and programming the logic of the educational simulator. That's why we will focus specifically on these steps of the development process.

First, let's describe the software that was used for development. The main 3D graphics editor used for creating spatial virtual 3D models was Blender.

Blender 3D was developed as a working tool by the Dutch animation studio NeoGeo. In June 1998, the creator of Blender, Ton Rosendahl, founded Not a Number to further develop and support Blender. The program was distributed free of charge. On July 18, 2002, a program began to collect sponsorship donations to cover the required amount. As early as September 7, 2002, it was announced that the required amount had been collected, and plans were being made to bring the source code and Blender itself under the GPL license in the near future. On October 13, 2002, the Blender Foundation released a product licensed under the GNU GPL. Blender is currently an open source project and is developed with the active

support of the Blender Foundation. Based on the date the first source code files were created, January 2, 1994 is considered Blender's birthday. In 2023, Blender turned 29 years old.

Blender is free software, unlike other 3D environments. This program has the ability to create three-dimensional models, and program scripts in the Python programming language, and, at the same time, get really high-quality results.

The main Blender functions:

*3D modeling.* Blender is represented by almost all existing methods of creating and editing with three-dimensional models (designing objects based on primitives, polygons, NURBS curves, Bezier curves, metaspheres, boolean operations, Subdivision Surface and basic sculpting tools is available). The program offers a wide range of different modifiers to enhance the models.

*Animation.* The user has access to tools such as rigging (skeletal animation), inverse kinematics, mesh deformation, constraints, keyframe animation, vertex weight editing, and more. Both rigid and soft body dynamics are implemented, as well as particle animation.

*Texturing and shader sets.* The program allows you to apply multiple textures to one object and is equipped with many texturing tools, including UV mapping and partial texture adjustment. Blender includes high-quality tools for customizable shaders that add flexibility to working with materials.

Thus, Blender is an affordable yet professional software tool for creating and animating three-dimensional graphics content for a drone pilot training simulator.

The development of the training simulator was carried out using the Unreal Engine 5 game engine, which is also freely distributed software. It allows for the integration of all the graphic content created in Blender, programming interactions between objects in the virtual world, creating high-fidelity and dynamic visual representations, and programming the logic of the game simulator.

Using basic tools and modifiers of Blender and Unreal Engine 5 we created the main scene of the project. Relief was created with a “Landscape tool” using Unreal Engine 5, and other objects like drone, fence, fountain, grass, and house were created by polygonal modeling in Blender.

## **CHAPTER 2. CREATING A TRAINING SIMULATOR PROCESS.**

### **2.1. Creating graphic content for simulator training**

Creating a model should start with sketching the main components of the 3D scene. Let's take the example of creating a quadcopter drone model. After placing a cube in the desired location, we load the 2D drone sketch files into Blender to understand the modeling stages.

1. Creating the base model shape Using basic shapes, we form key elements of the drone's digital model. We start with a low level of detail and gradually increase it. Various modifiers can be applied to automate and speed up the formation of the drone's appearance. One such modifier is the Subdivision Surface, which increases the number of polygons without distorting the polygonal mesh itself.

2. Creating materials and texturing the model A realistic drone model cannot be created solely based on the digital polygonal structure. To achieve high-quality visualization, it is necessary to create materials and textures that are as close to reality as possible. This allows for the transmission of all the complex surface elements without creating additional geometry on the surface itself. Creating UV unwraps is also important for accurate overlaying of textured images. This approach ensures precise positioning of textures on the surface, as well as specialized maps of roughness, normals, highlights, etc.

3. Object animation of the object is not a mandatory step for static models and scenes, but it is crucial for games and movies. In our case, animation was created using Blueprints in Unreal Engine 5.

4. Exporting models in formats convenient for use in the Unreal Engine 5 engine.

Export is a really important step in a process of creating 3D model. Usually 3D artists use a huge number of software specializing in one thing particular. The model needs to be merged, the animation saved and the correct export format selected, those are the main steps for exporting your models (Fig. 1) in any other visual software.

The visualizations of the main 3D models for the simulator trainer are presented in Figure 1.

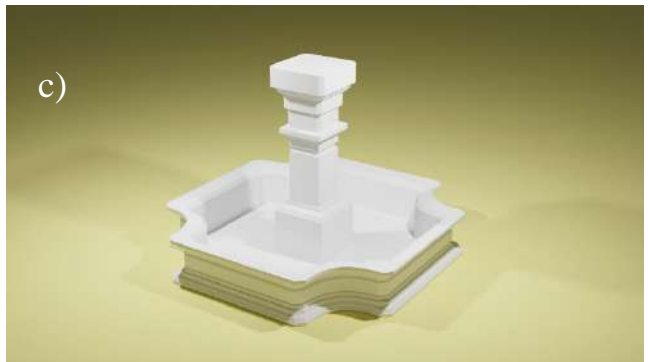
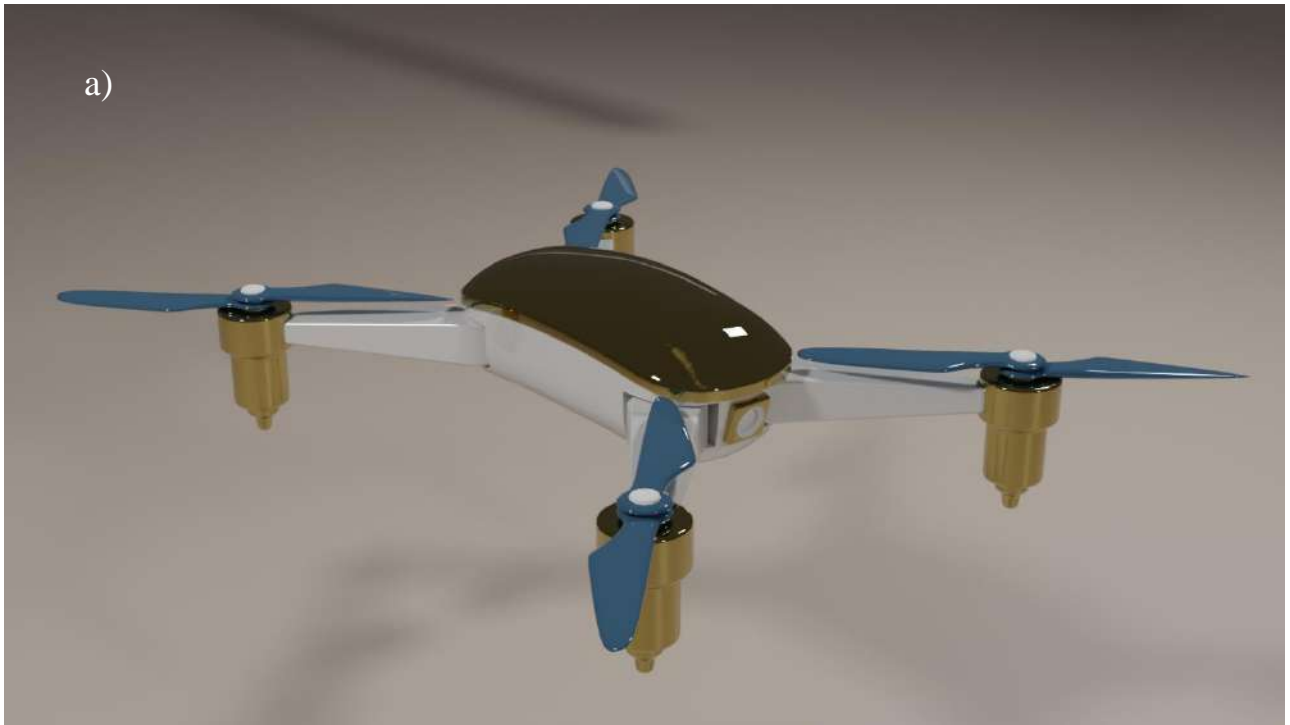


Fig. 2.1. Game ready 3D models

a – drone model, b – fence model, c – fountain model

## 2.2. Functionality overview UE5

The training simulator was created using Unreal Engine 5, with C++ being used as the programming language for programming all simulator functions overall, and Unreal Engine Blueprints were used to implement the drone control itself.

The drone control modeling was made possible by using Unreal Engine's entity component system. There are 4 main classes (Fig. 2) in Unreal Engine, object, actor, pawn, character. One inherits the other. The character possesses the pawn, which is also an actor, which is also an object. The possession is a process of a character taking control over a pawn. There are three main classes in the simulator, the game mode – simulation, the character – the pilot, and the pawn – the drone.

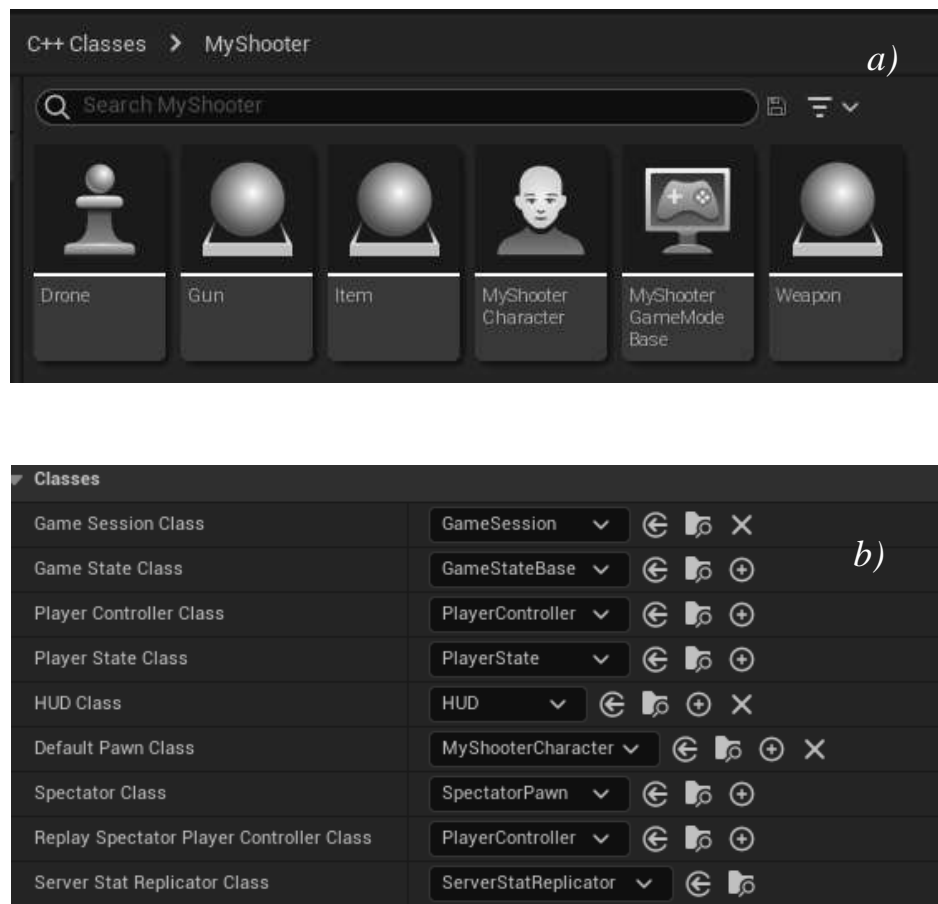


Fig. 2. Unreal Engine 5 class system

a – C++ classes, b – class blueprints

It defines such functionality as, the world that the simulation will start in, also known as the game level. The HUD Class – the heads-up display, the interface that that will be

viewed in the simulation. The Default Pawn Class – the character that will be spawned in the world, in the case of the simulation, it's the pilot. The Player Controller Class – defines which controller class will be used for simulating input in the game.

The character class is the Pilot, this class has functionality for movement, using items, using weapons, and controlling the drone. The header class defines all the components and functions, that the Character Class has. The components of character class are: UCharacterController, UMeshComponent, USpringArmComponent, UCameraComponent. These are used for moving the character, seeing the character, and having a camera that follows the character. The functions of the character class are: MoveForward, MoveRight, TurnAtRate, LookUpAtRate, TraceUnderCrosshairs, FireWeapon, ToggleDrone. These control the movement of the character, the use of items and weapons, the possession of the drone. The properties of the Character Class are: AWeapon, ADrone, isOnDrone. These let the character reference items, and drones, this architecture allows the character to reference any class that is a drone, which means it's not tied to one specific class.

The pawn class is the drone, this class has functionality for movement, camera switching and getting possessed. The header class defines all the components and functions, that the pawn class has. The components of pawn class are: UFloatingPawnMovement, UMeshComponent, USceneCameraComponent. These are used for simulating drone movement, drawing a shadow of the drone, and displaying two possible views of the drone, one from the back of the drone, one acting as simulation of the drone's camera. The functions of the Drone Class are: MoveRight, MoveForward, MoveUp, SwitchView, DisableDrone. These are used for having the drone being able to move in all directions, being able to switch the view from first camera to second, and to disable the drone, to return to controlling the Pilot.

### **2.3. The simulation of drone flight, and assessment of results**

The simulation of drone flight is done via the use of the UFloatingPawnMovement component, which is a built-in component in Unreal Engine 5, it has functions for movement, and variables (Tab. 1) for controlling how the simulation of movement works.

C++ class that is parent to the drone uses primarily this component for simulating drone flight. It also modifies the base variables of the UFloatingPawnMovement as follows:

Acceleration is set to 2000.0f, Deceleration is set to 500.0f, Turning boost to 1.0f, MaxSpeed to 300.0f. This achieves replication of how a drone in real life behaves in air. The C++ class also has variables for: wind speed, for simulating how fast the wind moves, which can slow down, or increase the speed of drone's handling; drone mass, which indicates on how much the drone weights, which also modifies all of the built in variables of the UFloatingPawnMovement accordingly to how heavy or light the drone is. Battery life, adds a battery life simulation to the drone, when depleted, the simulation will inform the future pilot of failure, and they will have to try again.

Table 2.1. Variables

Mark	Type	Name	Description
$V_a$	float	<b>Acceleration</b>	Acceleration applied by input (rate of change of velocity)
$V_b$	<u>uint32: 1</u>	<u>bPositionCorrected</u>	Set to true when a position correction is applied.
$V_d$	float	<b>Deceleration</b>	Deceleration applied when there is no input (rate of change of velocity)
$V_m$	float	<b>MaxSpeed</b>	Maximum velocity magnitude allowed for the controlled Pawn.
$V_t$	float	<u>TurningBoost</u>	Setting affecting extra force applied when changing direction, making turns have less drift and become more responsive.



The created drone simulation achieves the “feel” of how a real-world drone behaves, successful completion of simulation tasks within the time limit of the drone’s battery, and prepares future pilots for piloting real drones. The simulation takes into account the speed at which the wind is traveling, and the mass of the drone, to better replicate a real-life drone’s behavior. What could be added in the future to the simulation is the ability to take into account rain, snow, and other possible weather conditions.

## CONCLUSIONS

The modern development of digital technologies makes it possible to create educational software tools that contribute to the qualitative renewal of the educational process and significantly improve the training of specialists in various specialties. The analysis of scientific literature and open sources of information confirmed the relevance of the selected research topic.

The presented training simulator is aimed at improving the process of training drone pilots. It addresses a number of important points:

- the use of the simulator brings the participants of the training process as close as possible to the real conditions of using an unmanned aerial vehicle;
- the use of convenient and affordable software makes training mass and lowers its cost;
- the variation of tasks and performance conditions allows you to work out a significant number of types of tasks and their performance conditions, which only contributes to the awareness and preparedness of the pilot;
- a well-thought-out evaluation system allows you to select the best, who will continue to master piloting on already real devices.

Among the further directions of development of the indicated project, we define the following: thorough testing of the already created application, with the aim of identifying potential problem situations and their elimination; creation of more detailed and larger virtual maps of the worlds, which will allow the pilot to work out various tasks; improve the achievement evaluation system, etc.

## REFERENCES

1. Albeaino, G., Eiris, R., Gheisari, M., & Issa, R. R. (2022). DroneSim: A VR-based flight training simulator for drone-mediated building inspections. *Construction Innovation*, 22(4), 831-848. DOI:<https://doi.org/10.1108/CI-03-2021-0049>.
2. De Winter, J., Van Leeuwen, P. M., and Happee, R. (2012, August). Advantages and disadvantages of driving simulators: A discussion. *Proceedings of measuring behavior*, 2012, pp. 28-31.
3. Gu, H., Wu, D., and Liu, H. (2009). Development of a novel low-cost flight simulator for pilot training. *International Journal of Computer and Systems Engineering*, 3(12), 1581-1585.
4. Jones, R. M., Laird, J. E., Nielsen, P. E., Coulter, K. J., Kenny, P., and Koss, F. V. (1999). Automated intelligent pilots for combat flight simulation. *AI magazine*, 20(1), 27-27. DOI: <https://doi.org/10.1609/aimag.v20i1.1438>.
5. Ribeiro, R., Ramos, J., Safadinho, D., Reis, A., Rabadão, C., Barroso, J., and Pereira, A. (2021). Web AR solution for UAV pilot training and usability testing. *Sensors*, 21(4), 1456. DOI: <https://doi.org/10.3390/s21041456>
6. Sætren, G. B., Birkeland, T. F., Pedersen, P. A., Lindheim, C., and Skogstad, M. R. (2019). Opportunities and limitations in use of simulators in driver training in Norway. A qualitative study. In *Proceedings of the 29th European Safety and Reliability Conference (ESREL). 22–26 September 2019 Hannover, Germany*. DOI: [https://doi.org/10.3850/978-981-11-2724-3\\_0216-cd](https://doi.org/10.3850/978-981-11-2724-3_0216-cd).
7. Shashidhara, B. P., Chandrasekaran, R., Bhatia, Y., Magesh, G., Bineshkumar, K., and Kumar, H. V. (2018). Development of a Full Mission Simulator for Pilot Training of Fighter Aircraft. *Defence Science Journal*, 68(5). DOI : <https://doi.org/10.14429/dsj.68.12235>.
8. Smith, P. (1986). Instructional simulation: Research, theory, and a case study.
9. Vlakveld, W. P. (2005). The use of simulators in basic driver training. In *Humanist TFG Workshop on the Application of New Technologies to Driver Training, Brno, Czech*

*Republic. Available at: www. escope.  
info/download/research\_and\_development/HUMANISTA\_13Use.pdf.*