

Міністерство освіти і науки України  
Харківський національний автомобільно-дорожній університет

Дорожньо-будівельний факультет

Кафедра будівництва та експлуатації автомобільних доріг ім. О. К. Біруля

## ДИПЛОМНА РОБОТА

магістра

# КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ВИЗНАЧЕННЯ ФІЗИЧНИХ ВЛАСТИВОСТЕЙ ҐРУНТІВ

Завідувач кафедри канд. техн. наук, доцент

Роман СМОЛЯНЮК

Нормоконтролер канд. техн. наук, доцент

Олександр ВОЛОВИК

Керівник канд. техн. наук, доцент

Андрій СЕДОВ

Студент гр. Д-63-22

Віталій ОНУФРІЄВ



## 6. Консультанти розділів роботи

Розділ	Консультант	Підпис, дата	
		завдання видав	завдання прийняв

## 7. Дата видачі завдання 1 вересня 2023 року

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Дистанційна освіта	2.09-9.09	
2	Віртуальні лабораторії	10.09-20.09	
3	3D редактори	21.09-1.10	
4	Створення 3D моделей	2.10-12.10	
5	Система навігації у Blender 3D	13.10-18.10	
6	Текстурування 3D моделей	19.10-23.10	
7	UV mapping (UV-розгортка)	24.10-3.11	
8	Оформлення роботи	4.11-5.12	

Студент \_\_\_\_\_ Онуфрієв В. Є.

Керівник проекту (роботи) \_\_\_\_\_ Седов А.В.

## РЕФЕРАТ

Дипломна робота: 82 с., 20 рис., 2 додатка, 8 джерел.

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ, ҐРУНТИ 3D МОДЕЛЮВАННЯ  
ПРОГРАМУВАННЯ, ВІРТУАЛЬНА ЛАБОРАТОРІЯ, ДИСТАНЦІЙНА  
ОСВІТА

Об'єкт дослідження – Інноваційні програмні забезпечення (ПЗ);  
принципи та методології реалізації концепцій створення сучасного ПЗ.

Мета роботи – впровадженням інноваційних технологій в процес  
навчання.

Метод дослідження – аналіз та систематизація даних світового досвіду  
в галузі створення сучасного ПЗ.

В процесі виконання дипломної роботи були детально розглянуті:  
світова практика застосування дистанційного навчання; світова практика  
впровадження дистанційного навчання; світова практика використання  
апаратно-програмних комплексів для дистанційного навчання; сучасні  
методології створення програмного забезпечення для дистанційного навчання.

## ЗМІСТ

Вступ.....	5
1 Дистанційна освіта.....	6
2 Віртуальні лабораторії.....	10
3. Програмне забезпечення для 3D моделювання.....	18
3.1 3D редактори, що розповсюджується на комерційній основі.....	18
3.2 3D-редактори, що поставляються на основі вільних ліцензій.....	21
4.Порівняння ігрових рушіїв.....	25
5 Створення 3D моделей.....	29
5.1 Основні терміни.....	29
5.2 Система навігації у Blender 3D.....	30
5.3 Виділення в Blender 3d.....	31
5.4 Основні маніпуляції в Blender 3D.....	31
5.5 Методика створення.....	31
6.1 Карта текстури.....	34
6.2 Типи карт.....	35
6.3 Uv-розгортка.....	37
7 Імпорт до 3D рушія.....	40
8 Створення скриптів.....	43
Висновки.....	44
Перелік посилань.....	45
Додаток А Необхідні скрипти для забезпечення процесу взаємодії об'єктів.....	46
Додаток Б Ілюстративний матеріал.....	67

## ВСТУП

Інформатизація освіти вплинула на традиційні підходи у навчанні. Особлива увага стала приділятися питанням використання спеціалізованих програмних засобів для проведення, відтворення та демонстрації різних явищ. З'явилися такі терміни як «віртуальна лабораторія» і «віртуальний лабораторний практикум», які в тому числі характеризують залучення інформаційних технологій у сферу навчальної діяльності.

Є великий потенціал у застосуванні віртуальних лабораторій для підтримки викладання та навчання в науковій галузі. 3D віртуальне середовище забезпечує занурення в навчальний процес і взаємодію у віртуальному світі, які керується встановленими науковими принципами.

Метою даної роботи є розробка та впровадження віртуальних 3D лабораторій. Дослідження включає в себе проектування програмного забезпечення, що поєднує в собі розширену візуалізацію, інтерактивне управління та інтелектуальні компоненти.

## 1 ДИСТАНЦІЙНА ОСВІТА

Дистанційна освіта – це термін, який використовується для позначення широкого спектру освітніх програм та курсів, починаючи від курсів підвищення кваліфікації, і закінчуючи акредитованими програмами вищої освіти, які реалізують можливість тісного спілкування між студентами та викладачами і однокурсниками, як у випадку з очним навчанням.

Сучасне дистанційне навчання є одним з динамічних секторів вищої освіти, що швидко розвиваються. Стрімкий розвиток інформаційних технологій дозволяє дистанційному навчанню конкурувати із традиційною моделлю навчання. Слід зазначити, що в західних країнах дистанційне навчання вже давно застосовується для різних форм навчання та має широке поширення.

Аналізуючи переваги дистанційного навчання серед плюсів, можна відзначити наступне (рис. 1.1).



Рисунок 1.1 – Можливості дистанційного навчання

– студент, який навчається дистанційно, може вирішити, коли і скільки часу присвятити протягом семестру вивченню матеріалу. Він становить індивідуальний графік навчання собі.

– студентам не потрібно турбуватися, що вони відстануть від своїх однокурсників. Ви завжди можете повернутися до вивчення складних питань, переглянути відеолекції кілька разів, прочитати листування з учителем та пропустити вже вивчені теми. Головне – успішно скласти проміжні та підсумкові іспити.

– студенти можуть навчатися, не виходячи з дому чи офісу, у будь-якій точці світу. Щоб розпочати навчання, у вас має бути лише комп'ютер із доступом до Інтернету. У світі широко застосовуються електронні навчально-методичні ресурси, інформаційно-довідкові системи, системи відео-зв'язку, вебінари, тестові матеріали, відеолекції, здійснюється технічна підтримка під час роботи з комп'ютерним устаткуванням. Відсутність необхідності відвідувати навчальний заклад щодня є певною перевагою для людей з обмеженими можливостями, для тих, хто живе у важкодоступних місцях та для батьків із маленькими дітьми.

– дослідження американських вчених показують, що результати дистанційного навчання не поступаються і навіть перевершують традиційні форми навчання. Студенти дистанційного навчання вивчають більшу частину навчального матеріалу самостійно. Це покращує запам'ятовування та розуміння вивчених тем. А вміння відразу застосовувати отримані знання практично допомагає закріпити їх. Крім того, використання новітніх технологій у навчальному процесі робить його більш цікавим та живим.

– спілкування з викладачами здійснюється по-різному як онлайн, так і офлайн. Консультації з викладачем електронною поштою іноді ефективніше і швидше, ніж планування особистої зустрічі.

– доступ до всієї необхідної літератури надається студентам після реєстрації на сайті, або вони отримують навчальні матеріали поштою.

Однак дистанційне навчання на даному етапі не може повністю замінити традиційні форми навчання. До недоліків можна віднести:

- складність з адаптацією до онлайн-занять;
- відсутність реального спілкування на занятті, недоступність прямого очного спілкування з викладачем;
- обов'язкова наявність технічного обладнання для доступу до джерелам інформації (комп'ютер, доступ до Інтернету та ін.);
- складність мотивації учнів;
- висока вартість побудови системи дистанційного навчання;
- висока трудомісткість розробки курсів дистанційного навчання;
- низька комп'ютерна грамотність;
- нераціональне використання часу.
- технічні перешкоди. Поганий інтернет, відключення світла, збій платформи і заняття закінчується раніше, ніж планували.

– складніше з організацією практичних та лабораторних робіт в онлайн-форматі. Мало здобути нові знання. Потрібно навчитися застосовувати їх у житті. Дистанційному навчанню доводиться бути винахідливим, щоб не просто давати знання, а й допомагати закріплювати їх на практиці.

Зважаючи на надзвичайно високу вартість такого обладнання та витратних матеріалів до нього, а також складність експлуатації (що вимагає спеціальних навичок для його використання, можливості виходу з ладу через неправильне використання), лабораторні роботи на регулярній основі вкрай затруднені. Віртуальна лабораторна робота може вирішити проблему. Віртуальне обладнання не ламається, не зношується та не вимагає матеріалів для роботи.

У віртуальній лабораторії студенти можуть спробувати всі види експериментів без ризику пошкодження обладнання чи травм. Враховуючи зручність використання та різні практикуми, можна використовувати віртуальні лабораторії для віддаленого навчання. Віртуальний практикум може виконуватися студентами як під керівництвом викладача, і у рамках

самостійної роботи. Він також дозволяє використовувати його як лекційні демонстрації, без громіздкості та матеріальних витрат.

Таким чином, щоб допомогти студентам у навчанні, підвищити їх інтерес до професійної галузі, а також покращити розуміння матеріалу, необхідні віртуальні лабораторні комплекси, які не лише дозволяють наочно пояснити та показати роботу на обладнанні, але є безпечнішими та дешевшими. Звичайно, робота з реальним обладнанням все одно має бути головним у цьому питанні, але з урахуванням ситуації, що склалася в Україні, використання віртуальних лабораторних робіт актуальне і досить перспективне.

## 2 ВІРТУАЛЬНІ ЛАБОРАТОРІЇ

Віртуальна лабораторія - це програмно-апаратний комплекс, що дозволяє проводити експерименти без безпосереднього контакту з реальною установкою або при повній її відсутності. У першому випадку ми маємо справу з так званою лабораторною установкою з віддаленим доступом, яка включає в себе реальну лабораторію, програмно-апаратні засоби для управління заводом і оцифрування отриманих даних, а також засоби зв'язку. У другому випадку всі процеси моделюються за допомогою комп'ютера.

Отже, під віртуальними лабораторіями маються на увазі два типи апаратно-програмних комплексів:

- лабораторна установка з віддаленим доступом – назвемо такі комплекси віддалені лабораторії;

- Програмне забезпечення, що дозволяє моделювати лабораторні експерименти – віртуальні лабораторії (у вузькому сенсі).

Основними перевагами віртуальних лабораторій є:

- немає необхідності купувати дороге обладнання та реагенти. У зв'язку з недостатнім фінансуванням у багатьох лабораторіях встановлено старе обладнання, яке може спотворювати результати експериментів і служити потенційним джерелом небезпеки для студентів. Крім того, в таких сферах, як, наприклад, хімія, крім обладнання, потрібні ще й витратні матеріали (реагенти), вартість яких досить висока. Звичайно, комп'ютерна техніка і програмне забезпечення теж дорогі, але універсальність комп'ютерної техніки і її поширеність компенсує цей недолік;

- можливість моделювати процеси, які принципово неможливо виконати в лабораторії. Візуальна візуалізація на екрані комп'ютера. Сучасні комп'ютерні технології дозволяють спостерігати за процесами, які складно розрізнити в реальних умовах без використання додаткового обладнання, наприклад, через малий розмір спостережуваних частинок;

- можливість спостерігати за тим, що відбувається в різному часовому масштабі, що актуально для процесів, які відбуваються за частки секунди або, навпаки, тривають кілька років;
- безпека. Безпека є важливою перевагою використання віртуальних лабораторій у випадках, коли проводяться роботи, наприклад, з високою напругою або хімічними речовинами. Завдяки тому, що віртуальним процесом керує комп'ютер, з'являється можливість оперативно провести серію експериментів з різними значеннями вхідних параметрів, що часто необхідно для визначення залежностей вихідних параметрів від вхідних;
- заощаджує час та ресурси для внесення результатів в електронний формат. Деякі роботи вимагають подальшої обробки досить великих масивів цифрових даних, які виконуються на комп'ютері після ряду експериментів. Слабким місцем в такій послідовності дій при використанні реальної лабораторії є введення отриманої інформації в комп'ютер. У віртуальній лабораторії цей крок відсутній, оскільки дані можуть бути внесені в електронну таблицю результатів безпосередньо під час експерименту або експериментатора. Автоматично. Таким чином економиться час і значно знижується відсоток можливих помилок;
- можливість використання віртуальної лабораторії в умовах дистанційного навчання, коли немає можливості працювати в лабораторіях університету.

Віртуальна лабораторія STAR (Software Tools for Academics and Researchers) – програма Массачусетського технологічного інституту (MIT) з розробки віртуальних лабораторій для досліджень і викладання. Напрямами діяльності програми є розробка освітніх та дослідницьких додатків у загальній біології, біохімії, генетиці, гідрології та в галузі розподілених обчислень. Більшість додатків реалізовано на java або html. Офіційний сайт Програми: <http://star.mit.edu>.

Віртуальна лабораторія StarBiochem – це 3D-візуалізатор білкових молекул. Він має гнучку та детальну настройку. Офіційний сайт Програми <http://star.mit.edu/biochem/index.html>.

StarGenetics. - дозволяє моделювати процеси схрещування, вивчати закономірності успадкування моногенних ознак (так звані закони Менделя).

Офіційний сайт Програми <http://star.mit.edu/genetics/index.html>.

Віртуальна лабораторія StarORF. – дозволяє навчитися ідентифікувати так звані відкриті рамки зчитування (ORF - Open Reading Frame) - одиниці в ланцюзі ДНК або РНК, які здатні кодувати білок.

URL: <http://star.mit.edu/orf/index.html>.

Віртуальна лабораторія StarMolSim – це серія інструментів, які моделюють процеси молекулярної динаміки. Кожен з інструментів має широкий діапазон вхідних значень і, так само, широкий діапазон виходів для аналізу та дослідження. (<http://star.mit.edu/molsim/index.html>).

StarBiogene – Інструмент по Генетики. (<http://star.mit.edu/biogene/index.html>).

StarCluster – Набір інструментів для створення, налаштування та управління кластерами віртуальних машин на хмарному веб-сервісі Amazon EC2 (<http://star.mit.edu/cluster/index.html>).

Оцінити ці віртуальні лабораторії можуть лише профільні фахівці, але можна з певною часткою впевненості сказати, що вони є фундаментальними, охоплюють широке коло завдань у певній галузі знань, мають багатий інструментарій.

Віртуальна лабораторія VirtualLab – це проєкт з розробки віртуальних лабораторних робіт для студентів з фізики, хімії, біології та екології. Віртуальна лабораторна робота реалізована за допомогою технології Flash. Їх відрізняє вузька спеціалізація, в більшості випадків лінійність досвіду (вся послідовність дій і результати експерименту задаються заздалегідь). Продукти VirtualLab мають пізнавальну цінність і вирішують проблему проведення

лабораторних робіт за відсутності необхідного обладнання (<http://www.virtulab.net/>).

Приклади лабораторних робіт:

- вивчення взаємодії частинок та ядерних реакцій
- порівняння молярних теплоємностей металів
- знайомство зі зразками металів і сплавів
- ідентифікація неорганічних сполук
- вивчення зовнішньої будови і різноманіття членистоногих. Зовнішня будова річкового рака. Зовнішня будова комахи.
- дослідження змін екосистем на біологічних моделях (акваріумі)
- модель «хижак-жертва» Tray-Volterra

Віртуальна лабораторія Algodoo – програма, призначена для 2D-фізичного моделювання. Має дуже багатий інструментарій для створення різних об'єктів, механізмів і систем з метою моделювання їх фізичної взаємодії та властивостей. Програма здатна моделювати не тільки механічні процеси, а й оптичні, а можливість програмування за допомогою скриптової мови Thyme дозволяє створювати об'єкти з оригінальними фізичними властивостями, різними функціями, ефектами і явищами. Також можливе завантаження креслень: малюнок стає об'єктом моделювання і йому можна задати будь-які фізичні властивості. Програма безкоштовна. Існує репозиторій [algobox](http://www.algodoo.com/), де користувачі можуть ділитися своїми моделями. URL:: <http://www.algodoo.com/>

Віртуальна лабораторія PhET – це проект, розроблений Університетом Колорадо. Проект включає велику кількість віртуальних лабораторій, що демонструють різні явища в галузі фізики, біології, хімії, математики та наук про Землю.

Експерименти мають високу пізнавальну цінність і в той же час дуже захоплюючі. Приклади:

- Колірний зір URL: <http://phet.colorado.edu/en/simulation/color-vision>
- Балансуючий акт URL: <http://phet.colorado.edu/en/simulation>

### [/balancing-act](#)

- Джон Травольт URL: <http://phet.colorado.edu/en/simulation/travoltage>
- Звук <http://phet.colorado.edu/en/simulation/sound>
- Радіоактивна гра для побачень
- URL: <http://phet.colorado.edu/en/simulation/radioactive-dating-game>
- Побудуйте атом URL: <http://phet.colorado.edu/en/simulation/build-an-atom>
- Комплект для побудови схем (AC+DC)
- <http://phet.colorado.edu/en/simulation/circuit-construction-kit-ac-virtual-lab>

### [lab](#)

- Моя Сонячна система
- URL: <http://phet.colorado.edu/en/simulation/my-solar-system>
- Фотоефект URL: <http://phet.colorado.edu/en/simulation/photoelectric>
- Радіохвилі та електромагнітні поля
- URL: <http://phet.colorado.edu/en/simulation/radio-waves>
- Льодовики URL: <http://phet.colorado.edu/en/simulation/glaciers>

Демонстраційний проект Метою проекту Wolfram Demonstrations Project є демонстрація концепцій сучасної науки і техніки. Wolfram претендує на звання єдиної платформи для створення єдиного каталогу інтерактивних онлайн-лабораторій. Це, на думку його розробників, дозволить користувачам уникнути проблем, пов'язаних з використанням різномірних навчальних ресурсів і платформ розвитку. Є основні секції фізики, хімії та математики, а також ті, що присвячені техніці та суспільним наукам.

Наприклад віртуальна лабораторія ChemCollective присвячена вивченню хімії (<http://www.chemcollective.org/vlab/vlab.php>). Відмінною особливістю лабораторії є те, що тут немає завдань для користувача.

Іншими продуктами проекту є лабораторні проекти, присвячені певним темам і стосуються таких галузей хімії, як стехіометрія, термохімія, теорія кислот і основ, аналітична хімія та ін:

- Визначення невідомої рідини за її густиною  
<http://chemcollective.org/activities/vlab/69>
- Прогнозування концентрації ДНК <http://chemcollective.org/vlab/81>
- Проблема з кавою <http://chemcollective.org/vlab/91>
- Хлорид кобальту і принцип Ле-Шатльє  
<http://chemcollective.org/vlab/85>
- Стандартизація NaOH розчином КНР: кислотно-основне титрування  
<http://chemcollective.org/vlab/101>
- Створення буферного рішення <http://chemcollective.org/vlab/104>

Віртуальна лабораторія Late Nite Labs Це серія платних віртуальних лабораторій з хімії, біології та мікробіології. Вартість доступу становить близько \$50 за студента за семестр. Віртуальні лабораторії оформлені у вигляді 3D-світу. Адреса: <https://latenitelabs.com>

Віртуальна лабораторія Labshare – це консорціум, ініційований Міністерством освіти, зайнятості та трудових відносин Австралії, до складу якого входять 5 технічних університетів Австралії. Метою проекту є створення національної мережі віддалених лабораторій.

Приклади інсталяцій:

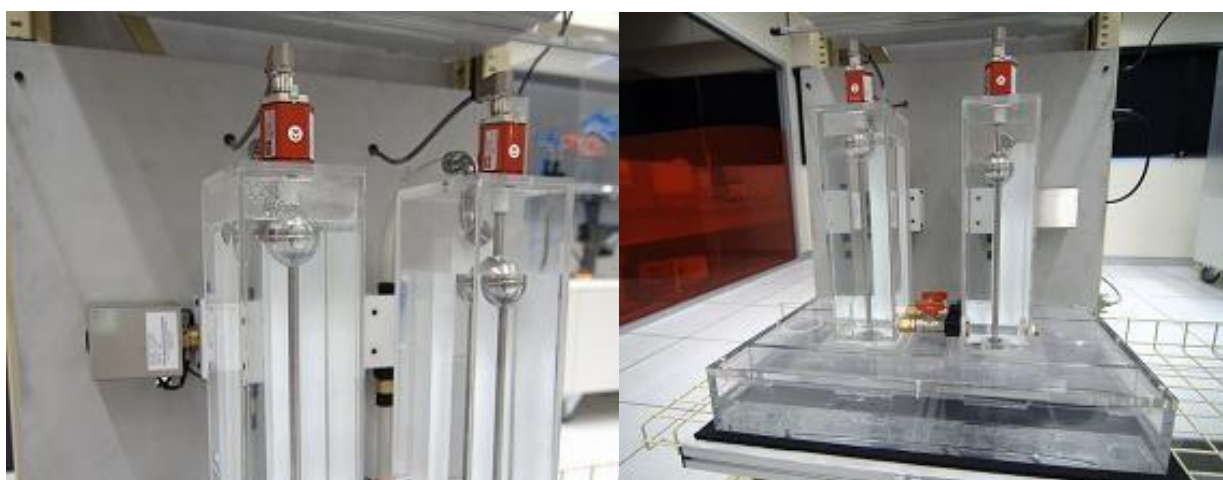


Рисунок 2.1 – Спарені танки покоління II

(<http://www.labshare.edu.au/catalogue/rigtypedetail/?id=1&version=2>)



Рисунок 2.2 – Інженерна геологія

(<http://www.labshare.edu.au/catalogue/rigtypedetail/?id=20&version=1>)

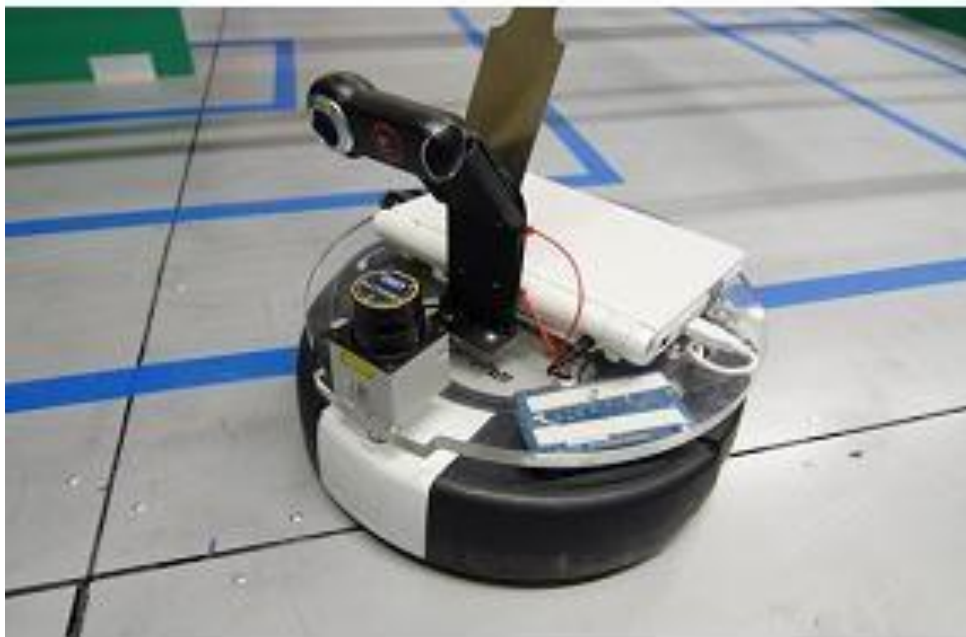


Рисунок 2.3 – iRobot

(<http://www.labshare.edu.au/catalogue/rigtypedetail/?id=15&version=1.1>)

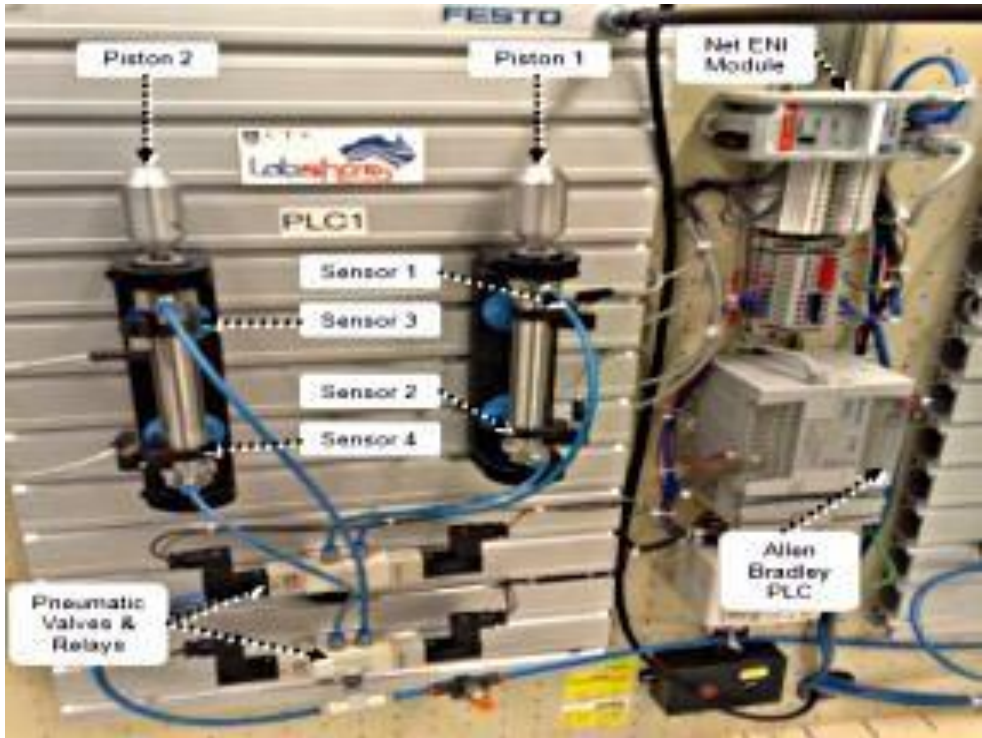


Рисунок 2.4 – ПКМ

### 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ 3D МОДЕЛЮВАННЯ

Сучасна тривимірна графіка активно використовується не тільки для створення зображень на аркуші поліграфічної продукції або на екрані комп'ютера, але і для розробки цілісних елементів, до яких відносяться, наприклад, деталі машин і окремі механізми, або цілі будівлі і споруди. 3D-моделювання дозволяє візуалізувати об'єкт, який існує тільки на кресленнях, наприклад, копію майбутньої будівлі. Застосування такої технології можна зустріти у різних сферах професійної діяльності

- кінематограф;
- відеоігри;
- анімації;
- віртуальна реальність;
- реклама;
- медицина (Створення модулів внутрішніх органів і протезів);
- архітектура;
- віртуальна археологія
- машинобудування та ін.

#### 3.1 3D редактори, що розповсюджується на комерційній основі

Maya вважається галузевим стандартом комп'ютерної графіки та може похвалитися широким набором інструментів і функцій (рис. 3.1). Maya чудово справляється з моделюванням, текстуруванням, освітленням і рендерингом – її широкий набір функцій включає частинки, волосся, фізику твердого тіла, тканину, моделювання рідини, анімацію персонажів тощо. Програма також дозволяє імітувати потужні вибухи, реалістичний рух одягу або волосся, поверхню води з невеликими хвилями або політ кулі – для цього існують спеціальні інструменти і модулі.

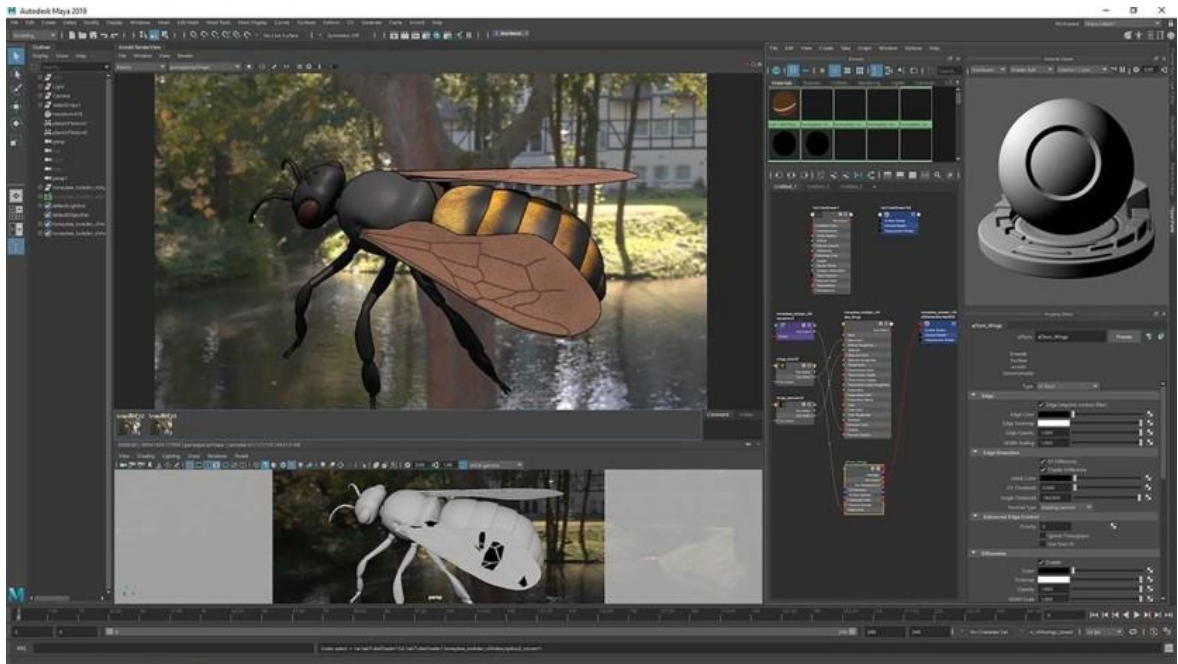


Рисунок 3.1 – Мауа

#### Функції:

- створення якісних героїв та персонажів;
- величезний вибір візуальних спецефектів;
- правдоподібні зміни форм різних об'єктів;
- можливість створення повітряної та водної імітації поверхонь;
- підтримка плагінів, що розширюють функціональність програми.
- ціна: \$215 на місяць.

Cinema 4D – користується підвищеним попитом серед професіоналів завдяки такому великому набору інструментів, як моделювання, ліплення, малювання, композиція, трекінг і анімація, якісний рендеринг, а також можливості реалізації незвичайних тривимірних ефектів. Cinema 4D має величезну спільноту з онлайн-бібліотекою навчальних посібників та інструкцій. Існує також навчальний сайт Cineversity (рис. 3.2).

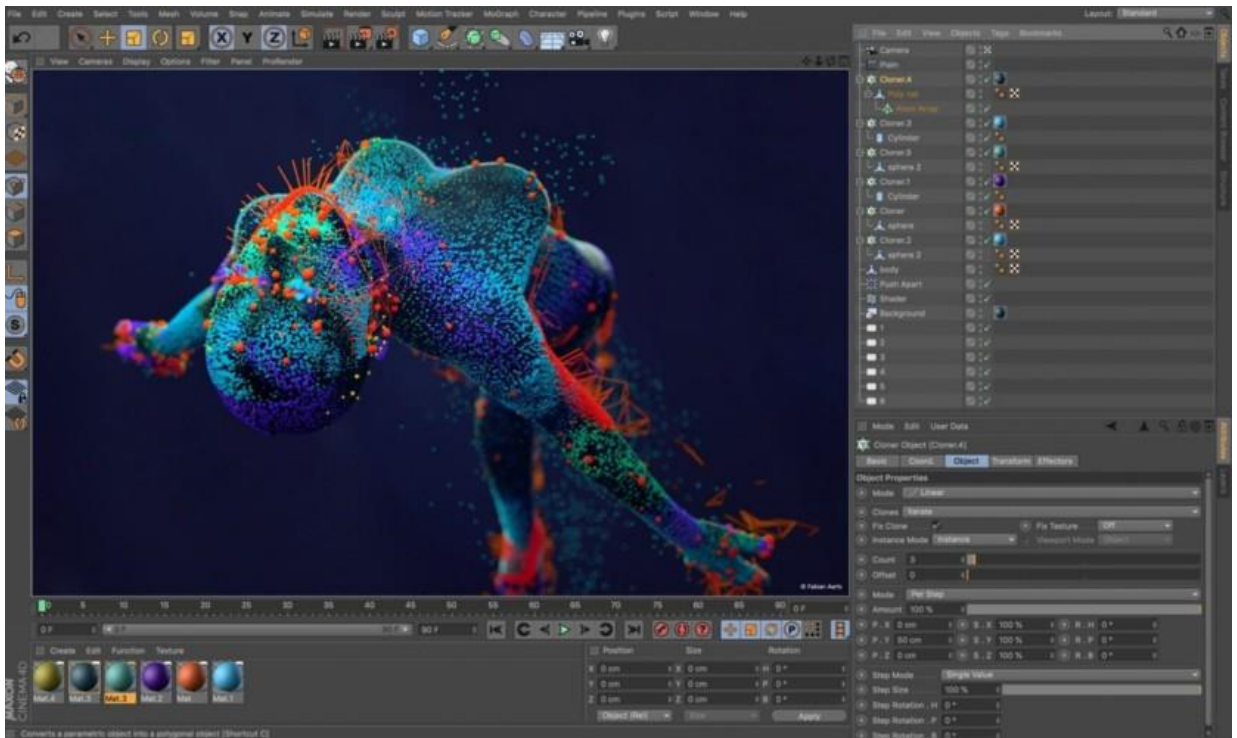


Рисунок 3.2 – Cinema 4D

### Функції:

- можна встановити сторонні візуалізатори
- великий вибір складних шейдерів та файлів зображень, анімацій;
- клонування об'єктів дозволить створювати складні анімації для кількох об'єктів у пару кліків;
- за допомогою модинаміки можна створювати фотореалістичні симуляції різних фізичних ефектів, таких як тертя або зіткнення;
- вартість: 999\$ на рік. Відсутня вільна ліцензія для здобувачів освіти.

3ds Max – програма для комп'ютерної графіки Autodesk 3D, призначена для ПК і використовується для виробництва телевізійних і художніх фільмів, а також візуалізації архітектури і продуктів., 3ds Max має дуже широкий набір інструментів 3D-моделювання, імітацію рідини, волосся та хутра. 3ds Max використовує методи прямого маніпулювання та процедурного моделювання,

а величезна бібліотека різних модифікаторів спрощує процес моделювання для початківців та досвідчених 3D-художників.

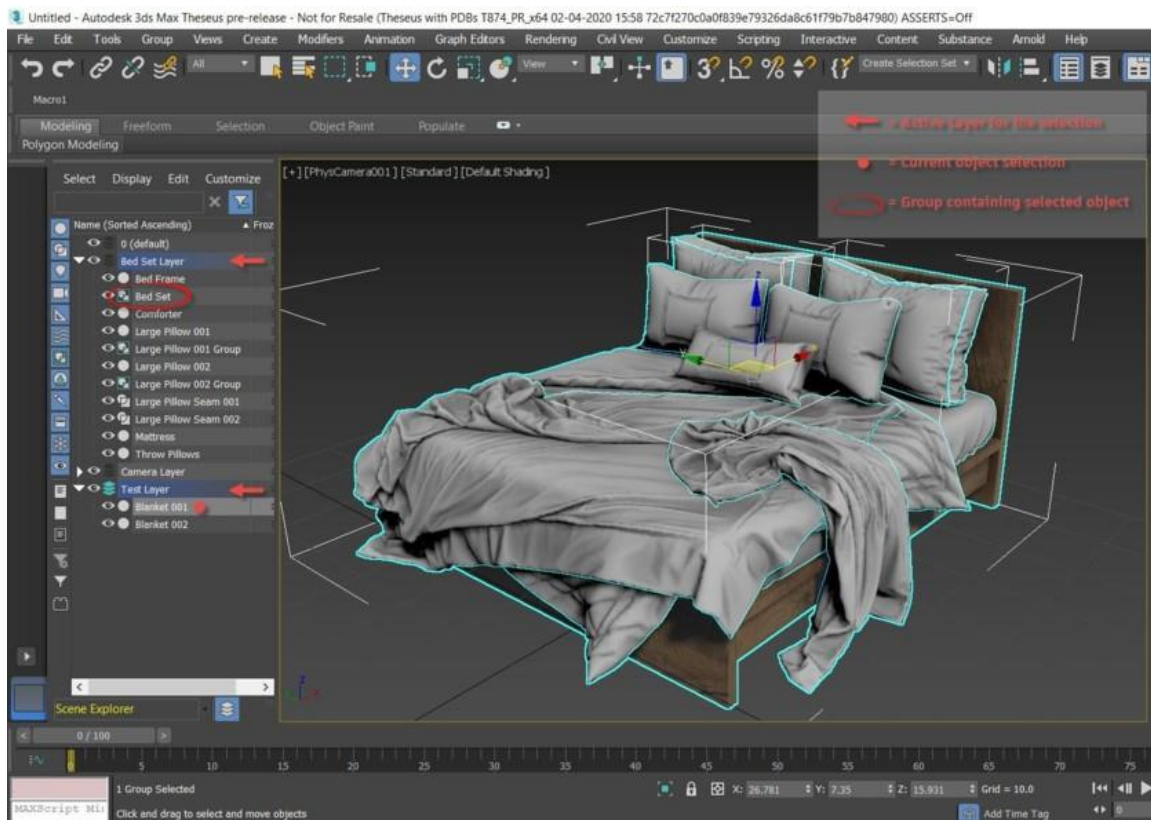


Рисунок 3.3 – 3ds Max

Функції:

- текстурування та UV-картування;
- за допомогою різних модифікаторів можна додати реалістичності персонажам та імітувати рідкі ефекти;
- є інструменти анімації та оснащення персонажів;
- ціна: \$215 на місяць.

### 3.2 3D-редактори, що поставляються на основі вільних ліцензій

Раніше на ринку професійного програмного забезпечення для 3D домінувало комерційне програмне забезпечення для 3D-моделювання. Однак зараз ситуація почала змінюватися на користь 3D-редакторів, що

поставляються на основі вільних ліцензій. Аналіз ринку ПЗ для 3D моделювання показав, що деякі програмні продукти що поширюються за вільною ліцензією за своїми функціональними можливостями не поступаються, а в деяких аспектах навіть випереджують ПЗ, що розповсюджується на комерційній основі.

В результаті дослідження ринку 3D-редакторів були обрані найбільш популярні безкоштовні програми для 3D-моделювання:

- Google SketchUp,
- Blender 3D
- Wings 3D
- SweetHome3D
- K-3D,
- Free CAD
- K3DSurf.

Google SketchUp – це програма, яка має зрозумілий і простий у використанні інтерфейс, повний набір інструментів. Програма також дозволяє імпортувати та експортувати різні формати двовимірної растрової та тривимірної графіки, зокрема (3ds, dwg, ddf, jpg, png, bmp, psd), є можливість використовувати плагіни, встановлювати тіні відповідно до заданої широти, довготи, часу доби та року. У програмі передбачена функція завантаження і використання численних готових скриптів, створених іншими користувачами, є поповнюються бібліотеки компонентів, матеріалів і стилів. Google SketchUp легко інтегрується з Google Планета Земля, який можна використовувати для моделювання будівель [1].

Blender наразі має статус найпотужнішого та найпопулярнішого програмного забезпечення. Програма є 3D-графічним редактором з відкритим вихідним кодом, володіє найширшим набором інструментів, надаючи своїм користувачам повну свободу від створення моделей для 3D-друку до створення фотореалістичних відеороликів, ігрової графіки, анімаційних фільмів,

візуальних ефектів. Особливістю моделювання в цьому додатку є підтримка всіляких комбінацій гарячих клавіш, які прискорюють процес створення моделі. Blender дозволяє вам зберігати результати симуляції у форматі .stl для 3D-друку, а також має масу плагінів для легкого моделювання. Програма має вбудовану підтримку російської мови [2].

Wings 3D – потужний 3D-редактор, призначений для моделювання тривимірних фігур, має інтуїтивно зрозумілий інтерфейс, має повний набір стандартних і додаткових інструментів [3].

SweetHome 3D має зручний та інтуїтивно зрозумілий інтерфейс і призначений для дизайнерів та архітекторів, використовується для моделювання дизайну інтер'єру. Програма має відкритий вихідний код і, відповідно, повністю безкоштовна, є можливість працювати онлайн в браузері [4].

K-3D – це потужне програмне забезпечення для 3D-моделювання, анімації та рендерингу, яке дозволяє створювати анімовані 3D-моделі та виконувати повний спектр операцій з 3D-об'єктами в режимі реального часу. В цей час у програмі вже є вбудовані геометричні примітиви і різні моделі і шаблони [5].

Free CAD з функцією параметричного моделювання ідеально підходить для інженерів або просунутих дизайнерів, які хочуть створювати складні, функціональні 3D-об'єкти для друку. На відміну від стандартного моделювання, параметричне (процедурне) моделювання є більш технологічним методом, що дозволяє швидко і якісно редагувати об'єкти, використовуючи історію створення моделі і зміни її параметрів. Великий набір професійних безкоштовних інструментів САПР надає користувачам практично необмежену свободу проектування [6].

K3DSurf – професійна програма, призначена для візуалізації та редагування математичних моделей у трьох, чотирьох, п'яти та шести вимірах; Він підтримує параметричні рівняння і роботу з ізоповерхнями [7].

Структура ринку показана на схемі (рис. 3.4).

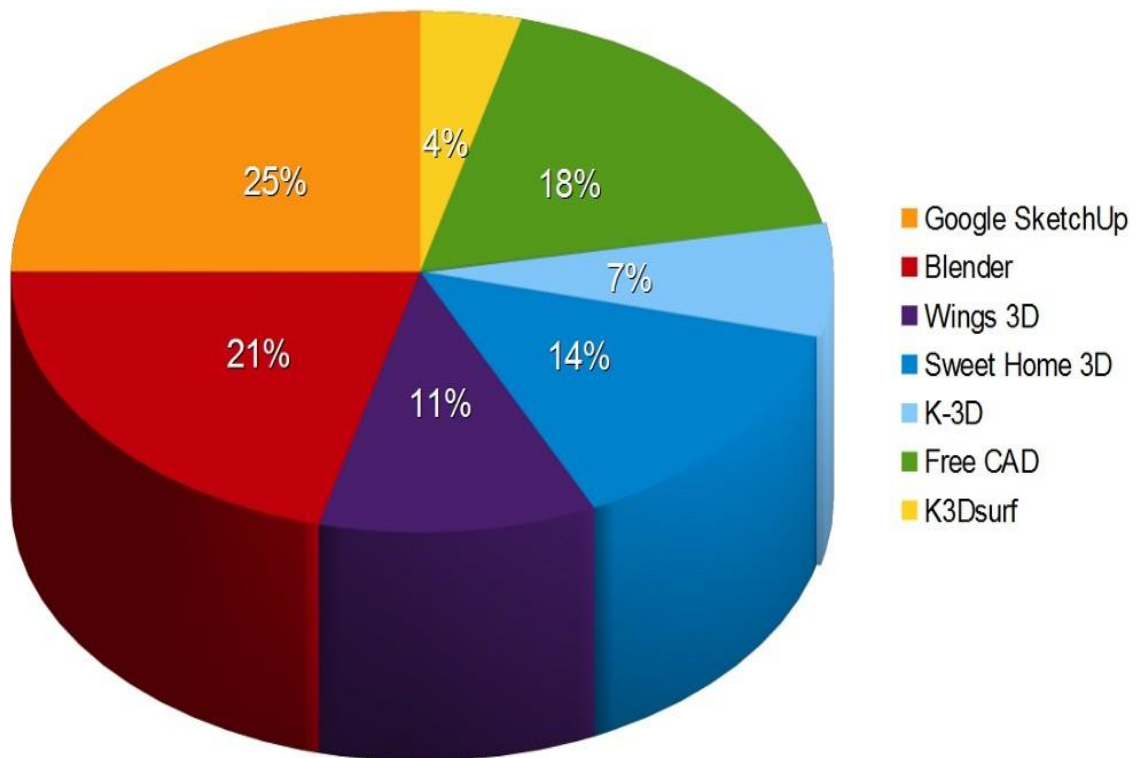


Рисунок 3.4 – Структура ринку

Виходячи з наявної мети та зважаючи на функціональні можливості, повноту опису цих можливостей та розповсюдження за вільною ліцензією - для використання було обране ПЗ Blender 3D

#### 4 ПОРІВНЯННЯ ІГРОВИХ РУШІЇВ

Ігровий рушій (IP) — це програмне забезпечення, яке надає розробникам інструменти та засоби для створення ігор. Він відповідає за основні функції гри, такі як рендеринг графіки, контроль фізики та керування звуком. Його вибір є важливим етапом у розробці ігор, і те, наскільки він правильний, може зробити процес розробки набагато простішим і швидшим, а також підвищити якість кінцевого продукту.

Порівняння IP проводилося за такими критеріями:

- функціональність та можливості: Різні IP можуть мати різні функції та можливості, такі як редактор рівнів, фізична система, система освітлення тощо. Деякі IP можуть бути більш придатними для створення певних типів продуктів або додатків.

- графіка та продуктивність: Продуктивність IP та його здатність створювати високоякісну графіку є важливими критеріями при виборі рушія.

- кросплатформність: деякі IP можуть бути більш придатними для розробки продукту для певних платформ, таких як мобільні пристрої, ПК або консолі. Якщо проекту потрібно працювати на різних платформах, то потрібно переконатися, що обраний движок підтримує всі необхідні платформи.

- вартість і доступність: Вартість і доступність IP можуть значно відрізнятись. Деякі можуть надавати безкоштовні ліцензії для некомерційних проектів, тоді як інші можуть вимагати оплати за певних умов.

- спільнота та підтримка: Підтримка спільноти та розробників є важливими критеріями при виборі IP. Наявність великої та активної спільноти може означати, що розробники можуть отримати швидку відповідь на свої запитання та проблеми.

Функціональність і особливості:

- Unreal Engine 5 пропонує безліч нових функцій і можливостей, включаючи технології Nanite і Lumen. Nanite дозволяє створювати

інструменти з мільярдами полігонів без шкоди для продуктивності, а Lumen пропонує нову систему освітлення без необхідності використання традиційних карт світла.

- Unity пропонує інтуїтивно зрозумілий і гнучкий інструментарій для створення ігор. Він надає широкі можливості для створення 2D і 3D ігор, підтримує широкий спектр платформ включаючи WebGL і включає систему Scriptable Render Pipeline (SRP), яка дозволяє налаштовувати рендеринг графіки відповідно до конкретних вимог проекту.

- CryEngine відомий своїми візуальними ефектами та реалістичною фізикою. Він пропонує потужні інструменти для створення ігрових світів, включаючи технологію тесселяції та вдосконалену систему освітлення. CryEngine також підтримує розробку VR-проектів.

- Source 2 — IP від Valve, призначений для створення ігор та додатків віртуальної реальності. Він надає різноманітні інструменти для роботи з анімацією, графікою та звуком. Source 2 також підтримує інтеграцію зі Steam Workshop та SteamVR для легкої розробки та розповсюдження продуктів.

#### Графіка та продуктивність:

- Unreal Engine 5 пропонує високоякісну графіку та вражаючу продуктивність завдяки новим технологіям, таким як Nanite та Lumen. Ці технології забезпечують реалістичне освітлення та деталізацію сцени без значної втрати продуктивності.

- Unity пропонує гнучкість для налаштування рендерингу графіки завдяки системі, яка була розглянута раніше: Scriptable Render Pipeline (SRP

- CryEngine відомий своєю видатною графікою та реалістичними ефектами. Однак це може призвести до підвищення системних вимог і труднощів з оптимізацією продуктивності, особливо на пристроях низького класу.

- Source 2 забезпечує якісну графіку і хорошу продуктивність,

особливо при роботі з віртуальною реальністю. Він оптимізований для роботи з різними пристроями та платформами, що дозволяє розробникам створювати візуально привабливі дизайни з хорошою продуктивністю.

#### Кросплатформність:

- Unreal Engine 5 є кросплатформним, що дозволяє розробникам створювати продукт для різних платформ, таких як Windows, macOS, Linux, iOS, Android, Xbox, PlayStation і навіть Switch.

- Unity пропонує одну з найширших платформних підтримки серед усіх IP, включаючи WebGL, всі мобільні, консольні та VR/AR пристрої. Це робить його привабливим вибором для розробників, які хочуть охопити максимальну кількість користувачів.

- CryEngine підтримує кросплатформність для пристроїв Windows, Xbox, PlayStation і VR. Однак його підтримка мобільних платформ і деяких інших операційних систем, таких як macOS і Linux, обмежена.

- Source 2 підтримує розробку ігор для Windows, macOS, Linux, SteamOS та віртуальної реальності. Підтримка мобільних платформ і консолей обмежена,

#### Ціни та доступність:

- Unreal Engine 5 є безкоштовним з деякими обмеженнями. Epic Games вимагає відрахування 5% від валового доходу після першого \$1 млн за кожен продукт. Вартість може бути нижчою для навчальних закладів та деяких інших випадків.

- Unity пропонує безкоштовну версію для особистого користування, має студентську безкоштовну ліцензію, але комерційні організації з річним доходом понад \$100 000 повинні платити за підписку на Unity Plus, Pro або Enterprise.

- CryEngine доступний для безкоштовного використання з умовою підписки на обов'язковий «роялті» у розмірі 5% від доходу після перших €5 000 за квартал. Опціонально доступні платні підписки з додатковими функціями.

– Source 2 доступний безкоштовно розробникам, які створюють продукт для платформи Steam.

Спільнота та підтримка:

– Unreal Engine 5 має величезну спільноту розробників, які активно діляться досвідом, ресурсами та підтримкою. Epic Games також надає документацію, навчальні посібники та форуми для підтримки розробників.

– Unity має одну з найбільших і найактивніших спільнот розробників, яка надає різноманітні ресурси, такі як документація, навчальні посібники, форуми та сховище активів. Також Unity Technologies надає офіційні послуги підтримки та консалтингу.

– CryEngine має невелику, але активну спільноту розробників, які надають ресурси та підтримку один одному. Crytek надає офіційну документацію, навчальні посібники та форуми для підтримки розробників.

– Source 2 має відносно невелику, але активну спільноту розробників, яка в основному зосереджена навколо розробки контенту для ігор Valve. Офіційна підтримка та ресурси доступні на форумах та у документації

Вибір IP залежить від багатьох факторів, таких як вимоги до графіки та продуктивності, кросплатформність, вартість та доступність, а також підтримка та спільнота. Unreal Engine 5 пропонує передові технології та потужні можливості, але інші рушії, такі як Unity, можуть бути більш придатними для деяких проєктів через їхню гнучкість та широку підтримку платформи.

Виходячи з наявної мети та зважаючи на функціональні можливості, повноту опису цих можливостей, розповсюдження за вільною для здобувачів освіти ліцензією та можливістю створення ПЗ для платформи WebGL -- для використання було обране ПЗ Unity 2022.3.4.

## 5 СТВОРЕННЯ 3D МОДЕЛЕЙ

### 5.1 Основні терміни

Рендеру комп'ютерній графіці — це процес побудови фінального зображення.

Вершина – точка у тривимірному просторі.

Ребро – це завжди пряма лінія, що з'єднує між собою 2 вершини.

Грань – це заповнений простір між замкнутою петлею ребер. Грані створюють поверхню об'єкта, які відображаються при рендері об'єкта. Грані можуть бути трикутними, чотирикутними та багатокутними. Трикутник завжди плоский, за рахунок цього трикутні грані обраховуються при рендері швидше. Саме тому при створенні ігор не використовуються меші, що складаються з багатокутників.

Меш – це полігональна сітка, або об'єкт, що складається з вершин, ребер та граней, які визначають форму багатогранного об'єкта у тривимірній комп'ютерній графіці та об'ємному моделюванні.

Нормаль – вектор перпендикулярні до площини, дотичної до поверхні в окремій точці. Він визначає напрямок та орієнтацію поверхні в кожній точці, що дозволяє отримати реалістичне відображення об'єктів. Нормалі використовуються у різних алгоритмах тривимірної графіки, таких як затінення, відображення світла, розрахунок тіней та багатьох інших.

Масштабування – пропорційна зміна габаритних розмірів.

Екструдкування – операція, що дозволяє змінювати геометрію об'єктів за рахунок створення копій ребер і граней шляхом свого роду видавлювання – переміщення та зміни розмірів.

Boolean – прості логічні операції. У комп'ютерній 3D-графіці це спосіб взаємодії двох об'єктів та створення на їх основі змішаного результату, який може являти собою різницю двох об'єктів, їх об'єднання або перетин.

Фаска – щоб досягти більш реалістичного вигляду, часто доводиться округляти деталі різних об'єктів. Для цих цілей у Blender існує інструмент Фаска (Bevel).

## 5.2 Система навігації у Blender 3D

Три основні види – вид зверху, вид спереду та вид справа збоку відповідають клавішам 7, 1, 3.

Цифра 2 на NumPad здійснює обертання вниз на задану кількість градусів; протилежний поточному. Комбінація клавіш за допомогою Numpad та Ctrl теж має функціональність: Ctrl+1 – вид ззаду, Ctrl+2 – переміщення вниз, Ctrl+3 – вид зліва, Ctrl+4 – переміщення вліво, Ctrl+6 – переміщення вправо, Ctrl+7 – вид знизу, Ctrl+8 – переміщення вгору. Shift + 1 – вирівнює вид виділеної грані щодо площини XZ (вид спереду, орієнтація трансформації Normal). Shift+Ctrl+1 – вирівнює вид виділеної грані щодо площини XZ (вид ззаду, орієнтація трансформації Normal). Shift+ 3 – вирівнює вид виділеної грані щодо площини YZ (вид справа, орієнтація трансформації Normal). Shift+Ctrl+3 - вирівнює вид виділеної грані щодо площині YZ (вид ліворуч, орієнтація трансформації Normal). Shift+4 – обертає вид вліво щодо перпендикулярного екрану осі, Shift + 6 – обертає вид вправо щодо перпендикулярного екрану осі, Shift + 7 – вирівнює вид виділеної грані щодо площини XY (вид зверху, орієнтація трансформації Normal). Shift+Ctrl+ 7 – вирівнює вид виділеної грані щодо площини XY (вид знизу, орієнтація трансформації Normal),

NumPad+ • (точка) – переміщує вигляд до поточного виділеного об'єкта.  
NumPad +/(слеш) - перемикає між локальним та глобальним видами.

### 5.3 Виділення в Blender 3D

У Blender існує безліч способів виділення об'єктів та їх частин. За замовчуванням виділення здійснюється за допомогою ЛКМ. Правий клік викликає контекстне меню, яке змінюється в залежності від місця кліка. Якщо виділити кілька об'єктів, останній виділений завжди буде активним об'єктом. Рамка виділення активного об'єкта також відрізнятиметься за кольором від інших об'єктів.

### 5.4 Основні маніпуляції в Blender 3D

Три основних інструмента – це переміщення (Grab), масштабування (Scale), обертання (Rotate). Відповідні кнопки: G, R, S. Крім того, для кожного інструменту є можливість застосування цих інструментів щодо об'єкта виключно по потрібній координатній осі: клавіша X – по осі X, Y – по Y, Z – Z.

### 5.5 Методика створення

Беручи за основу нагідний до задачі тривимірний примітив з наявного набору (куб, циліндр, шар, площина ікосфера) та застосовуючи вишевикладені маніпуляції до всього об'єкту чи до його окремих елементів в потрібній логічній послідовності отримаємо потрібні 3D моделі. Для створення неплоскої поверхні для імітації об'єму ґрунту на подносі до цієї поверхні було застосовано 2 модифікатори. Завдяки першому було підрозділено поверхню та отримано високополігональний меш. Застосування другого модифікатора дало нам змогу отримати зміщення поверхні (тобто вершини мешу перестали належати одній площині і ми отримали потрібну нам форму)

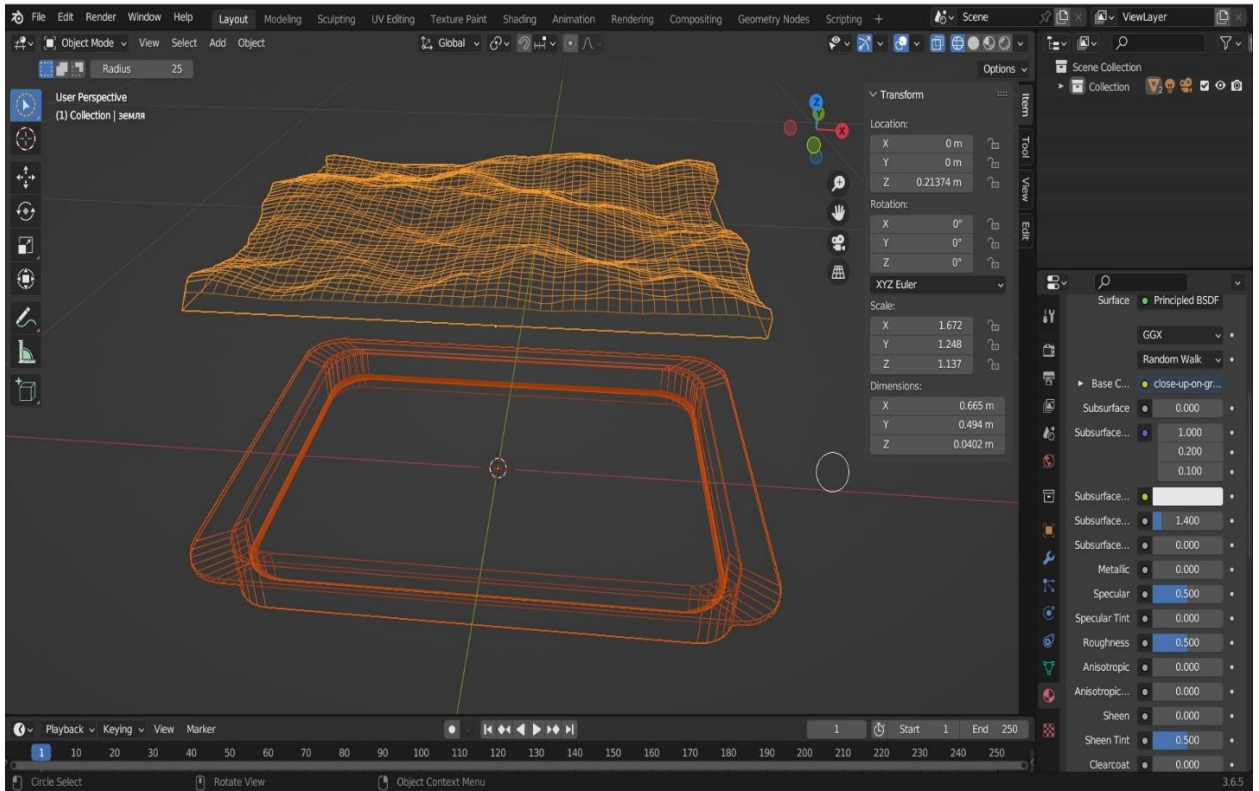


Рисунок 5.1 – 3D модель подносу з ґрунтом

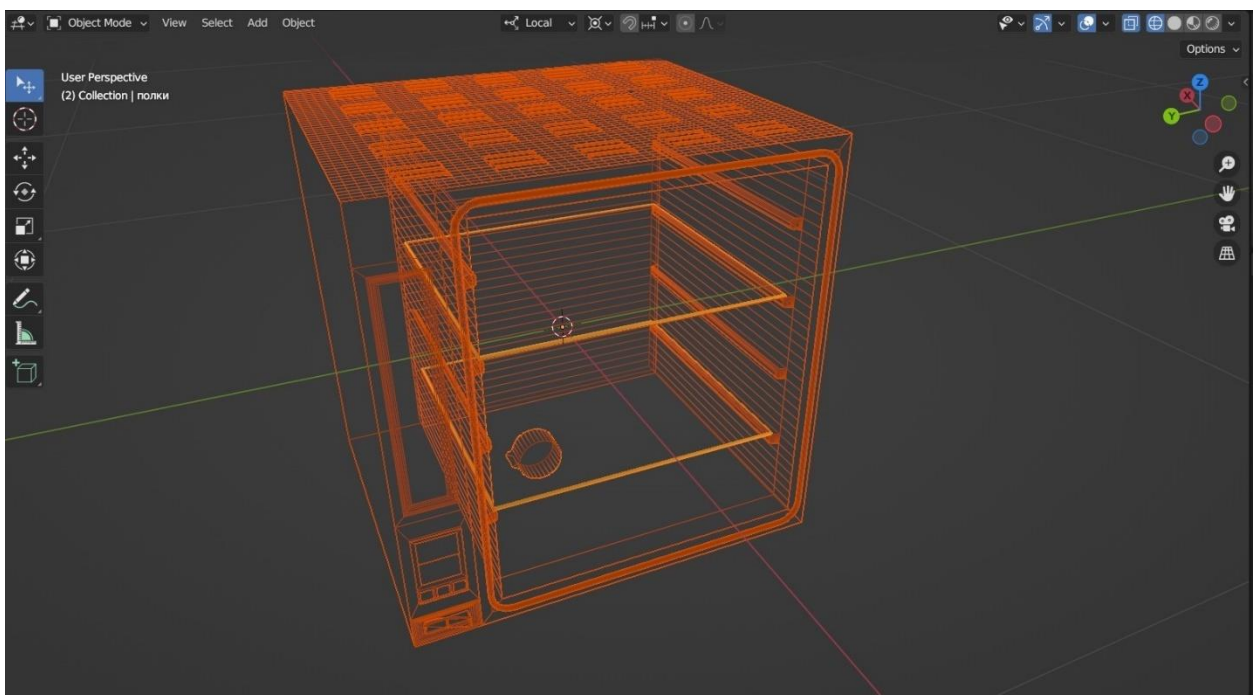


Рисунок 5.2 – 3D модель сушильної шафи

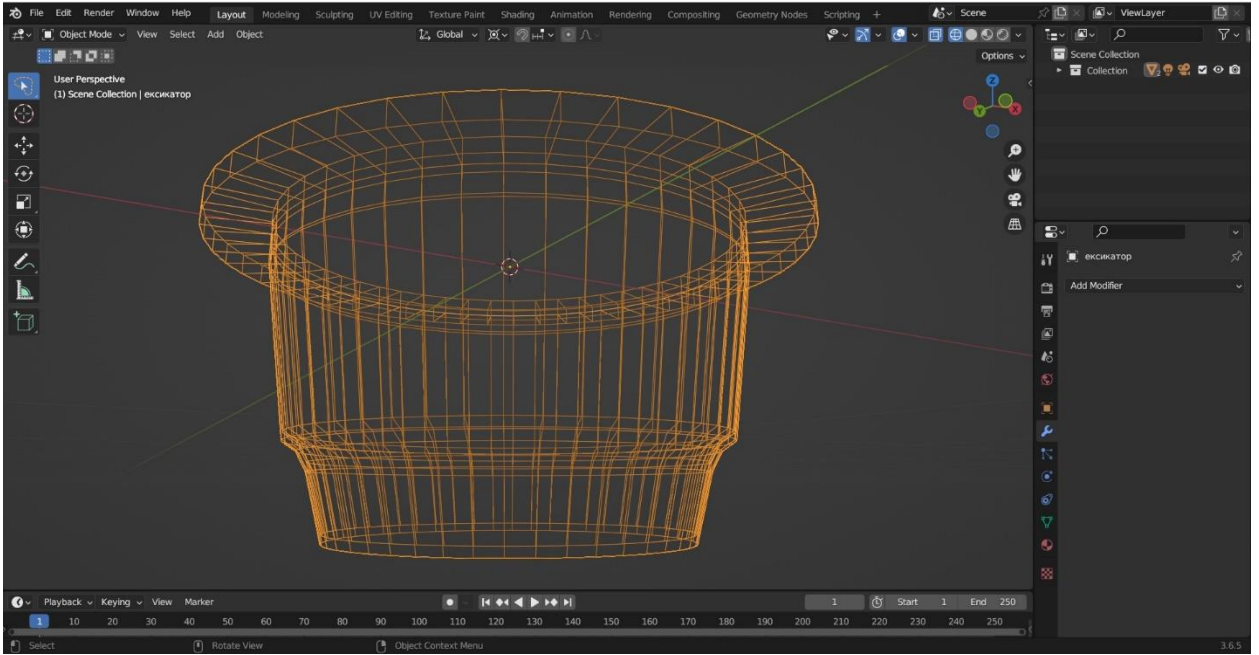


Рисунок 5.3 – 3D модель эксикатору

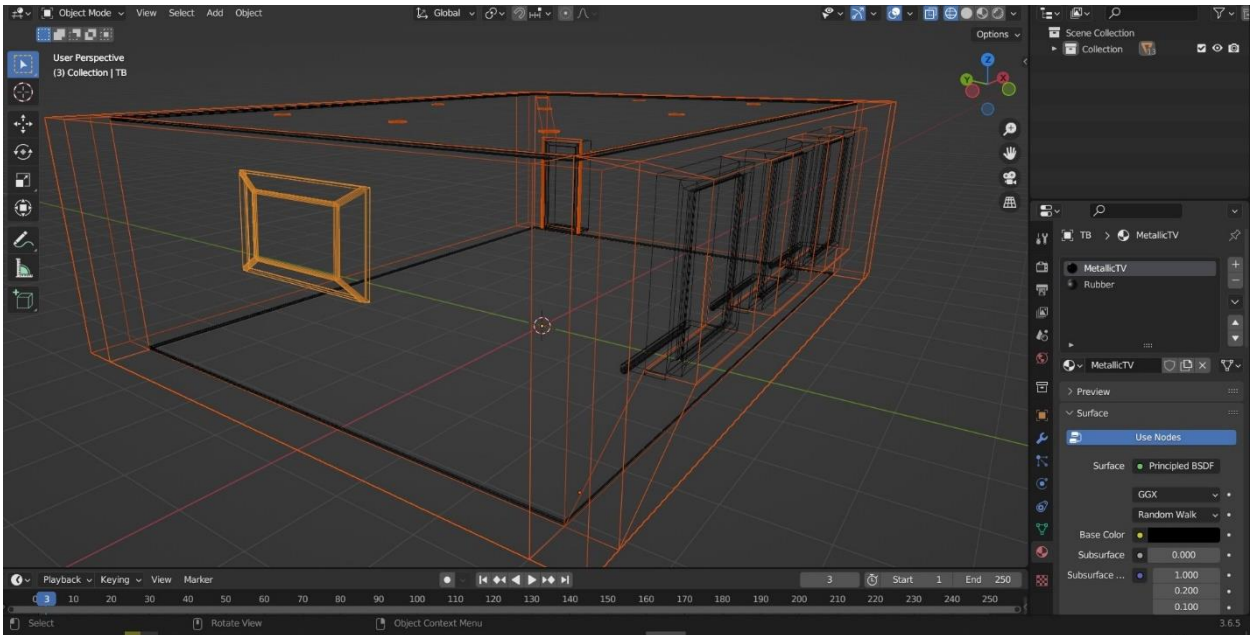


Рисунок 5.4 – 3D модель лабораторії

## 6 ТЕКСТУРУВАННЯ 3D МОДЕЛІ

Текстура (англ. Texture mapping) – це спосіб надання поверхні 3D деталі – полігону: кольору, фактури, блиску, матовості та інших фізичних властивостей (для імітації найчастіше якогось природного матеріалу, наприклад: паперу, дерева, каменю, металу тощо).

Поняття «текстура» є важливим елементом 3D-моделювання, оскільки дозволяє відтворити також малі об'єкти поверхні, створення яких полігонами виявилось б надмірно ресурсоємним. Наприклад, шрами на шкірі, складки на одязі, дрібні камені, предмети на поверхні стін і ґрунту та багато іншого. Отже, текстура використовується для заповнення поверхонь об'єктів і як шар для додання певного ефекту або зміни геометрії всьому зображенню або його частини.

Текстури бувають шовні та безшовні (патерни або візерунки). Головна відмінність останніх в тому, що при поєднанні одних і тих же фрагментів поверхня залишається цілісною.

Якість поверхні текстури визначається текселями – кількістю пікселів на мінімальну одиницю текстури. Оскільки сама по собі текстура є зображенням, роздільність текстури і її формат відіграють велику роль, яка згодом позначається на загальному враженні від якості графіки у 3D-додатку.

Створення параметрів матеріалу здійснюється за допомогою використання карт (зображення) розташованих у відповідних каналах, або за допомогою процедурних карт, а також змішаного методу.

### 6.1 Карта текстури

Карта текстури застосовується для утворення певного параметру візуального відображення на поверхні заданої форми. Цей процес нагадує застосування візерунчастого паперу на звичайній білій коробці. Кожній

вершині в 3D моделі присвоюється координати текстури (яка у разі 2D відома, як UV координата). Місця відбору проб зображення згодом інтерполюється по поверхні моделі з отриманням візуального результату.

Найчастіше це файл растрового (рідше векторного) зображення з розширенням JPEG, PNG або PAM.

## 6.2 Типи карт

– Diffuse map (карта базового кольору) — текстура, якою задається основний колір поверхні. Для того, щоб добре працювати з картою нормалей, дзеркальною та іншими картами дифузна карта не повинна мати ніякого спрямованого світла на своїй поверхні, вона повинна мати тільки загальні навколишні оклюзії, тобто поверхня стає темніша в глибоких тріщинах і навколо тиснення деталі;

– Specular map (карта відбиття) — карта, яка показує здатність відображення матеріалу. На відміну від reflection map, specular map не показує відображення сцени, в якій знаходиться об'єкт, а зображує відбиття світла падаючого на нього. Specular Map містить в собі пікселі в чорно-білій колірній гамі. Чим світліший піксель, тим більше здатність матеріалу відбивати світло і тим яскравіше на ньому відблиски від світла. Відповідно чим темніший піксель, тим більш матовим стає матеріал і втрачає свою властивість відбивати світло. Для матеріалів керамічної плитки і полірованого металу можуть використовуватися світлі тони, а для тканин і дерева темні;

– Bump map (карта рельєфу) — карта для імітації рельєфу поверхні. Створюється в градаціях сірого кольору. При рендерингу сірі області карти (мається на увазі сірий колір —  $R = 150, G = 150, B = 150$ ) відобразатиметься звичайним чином, чорні будуть замальовуватись, а білі — освітлюватись (чим область світліше, тим більше рівень яскравості) — ця ілюзія світла і тіні створює ефект рельєфу поверхні;

– Normal map – карта для імітації рельєфу поверхні. Дозволяє змінювати нормаль відображуваного пікселя ґрунтуючись на кольоровій карті нормалей, в якій ці відхилення зберігаються у вигляді текселя, колірні складові якого  $[r, g, b]$  інтерпретуються в осі вектора  $[x, y, z]$ , на основі якого обчислюється нормаль, використовувана для розрахунку освітленості пікселя. Завдяки тому, що в карті нормалей використовують 3 каналу текстури, цей метод дає більшу точність, ніж Bump mapping, в якому використовується тільки один канал і нормалі, по суті, всього лише інтерпретуються в залежності від «висоти». Карти нормалей зазвичай бувають двох типів: object-space – використовується для об'єктів, що не деформуються, таких як стіни, двері, зброя тощо; tangent-space – застосовується для можливості деформувати об'єкти, наприклад персонажів. Для створення карт нормалей зазвичай використовується високополігональні і низькополігональні моделі, за їхнім порівнянням встановлюють потрібне значення відхилення нормалей.

– Displacement map – карта зміщення, для показу рельєфу поверхні. Також представляється в градаціях сірого кольору. Принципова відмінність від попередніх двох методів, що цей спосіб автоматично посправжньому моделює за картою полігони. Природно він найякісніший і найбільш ресурсомісткий;

– Height map – двовимірний масив, кожен елемент якого інтерпретується як висота. Часто використовуються в програмах для створення ландшафтів (Terrain), щоб зберігати інформацію про висоту кожної точки місцевості. Використовуються також в технології бамп-маппінга (bump mapping);

- AmbientOcclusion (AO) map – карта затінення;
- Reflection map – карта відображення;
- Curvature (Concave, Convexity, Pits, Peaks, Worn Edge) map – карта, яка зберігає опуклості, вигини сітки;

– *Opacity map* (*opaque map*), (карта непрозорості) – формується на основі чорно-білої текстури, де повністю чорний колір буде відображатися прозорим (вирізатися з дифузної карти), білий повністю залишатиметься відображеним, а сірі відтінки відображатимуться напівпрозора з альфа інтенсивністю [9].

### 6.3 UV-розгортка

UV mapping (UV-розгортка) – відповідність між координатами на поверхні тривимірного об'єкту ( $X, Y, Z$ ) і координатами на текстурі ( $U, V$ ). Значення  $U$  і  $V$  зазвичай змінюються від 0 до 1. Розгортка може будуватися як вручну, так і автоматично.

Сучасне тривимірне апаратне забезпечення вважає, що UV-перетворення в межах одного трикутника є афінним – тому достатньо задати  $U$  і  $V$  для кожної вершини кожного з трикутників. Втім, як саме стикувати трикутників один з одним, вибирає 3D-моделер, і вміння будувати вдалу розгортку – один з показників його класу. Існує кілька показників якості розгортки, які можуть суперечити один одному:

– Максимально повне використання площі текстури. Втім, в залежності від розриву між «мінімальними» і «максимальними» системними вимогами, по краях розгорнення текстури потрібен певний «допуск» на генерацію текстур меншого розміру.

– Відсутність областей з недостатньою або надлишковою деталізацією текстури.

– Відсутність областей з зайвими геометричними спотвореннями.

– Подібність із стандартними ракурсами, з яких зазвичай малюється або фотографується об'єкт – спрощує роботу художника по текстурам.

– Вдало розташовані «шви» – лінії, відповідні одному ребру, але розташовані в різних місцях текстури. Шви бажані, якщо є природний

«розрив» поверхні (шви одягу, кромки, зчленування і т. д.), і небажані, якщо таких немає.

– Для частково симетричних об'єктів: вдале поєднання симетричних і асиметричних ділянок розгорнення. Симетрія підвищує деталізацію текстури і спрощує роботу художника по текстурам; асиметричні деталі «оживляють» об'єкт [9].

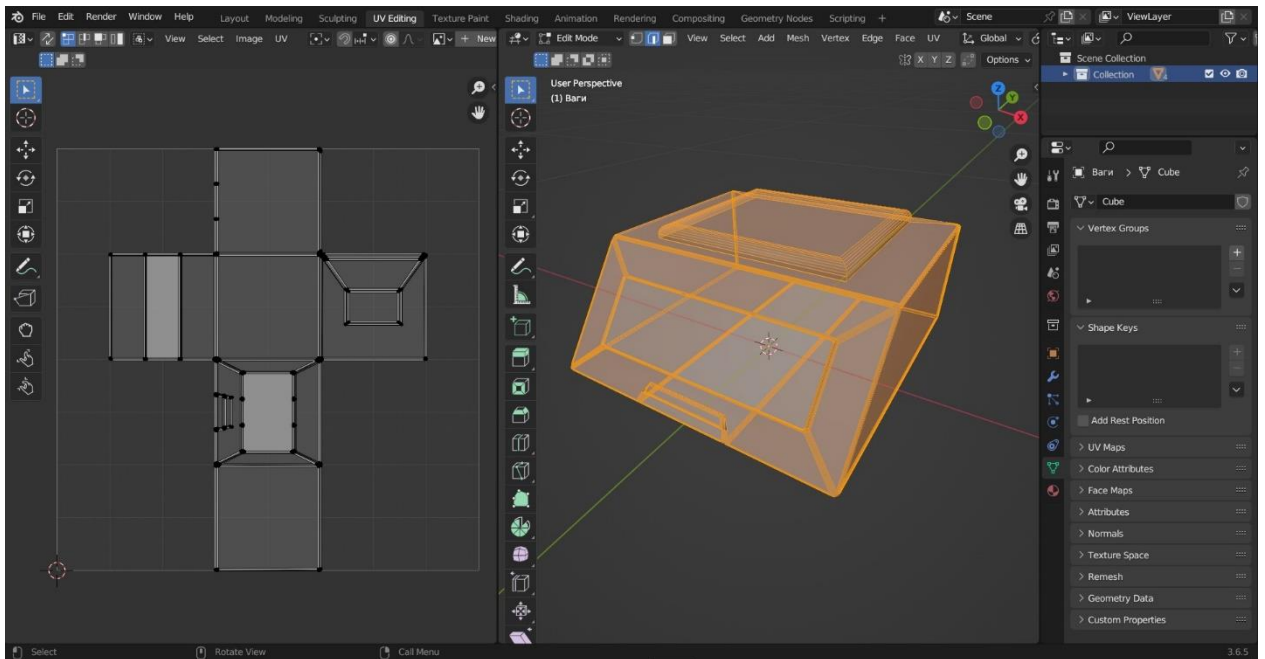


Рисунок 6.1 – 3D модель та її UVрозгортка

У Blender 3D текстура виступає як частина компоненту Матеріал. Налаштування властивостей матеріалу може бути розширено завдяки використанню спеціальних вузлів (Nodes). Використовуючи текстури за наявної UV розгортки в поєднанні з Nodes ми отримуємо дуже широкі можливості щодо вибору зовнішнього забарвлення 3D моделі.

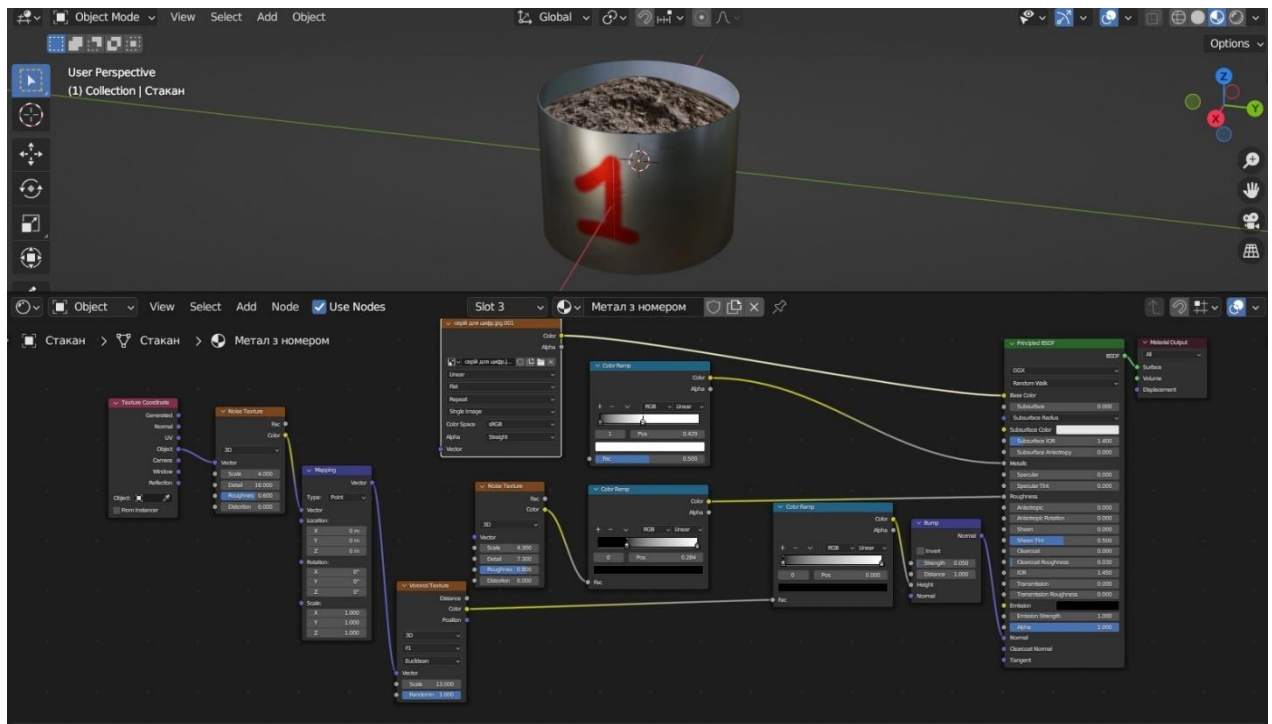


Рисунок 6.2 – 3D модель стакану та налаштування матеріалів

Однак слід враховувати, що отриманий матеріал в такому вигляді неможливо імпортувати до ІР. За для успішного імпорту текстурованої моделі потрібно провести процедуру запікання текстур.

Запікання – це процес взяття інформації з високополігональної моделі та створення на її основі текстурних карт, які перенесуть цю інформацію на низькополігональну модель за допомогою її UV-сканування. Це частина процесу оптимізації моделей для ігор, і це дуже важливо, оскільки це хороший спосіб зберегти зовнішній вигляд моделі, та зменшити навантаження на ІР під час рендерінгу.

Після проведення вищевказаних маніпуляції можна експортувати 3D модель до ІР.

## 7 ІМПОРТ ДО 3D РУШІЯ

3D модель імпортується у форматі fbx, який дозволяє імпортувати модель одразу з усіма належними їй ресурсами. Після імпорту модель набуває властивостей об'єкту, до якого можна застосовувати компоненти IP, що роблять можливою її взаємодію з іншими об'єктами на сцені. Вона розміщується на сцені у потрібному місті та положенні.

Колайдер – геометрична «оболонка» об'єкту, що відповідає в першу чергу за фізичну взаємодію з іншими об'єктами на сцені, що мають свої колайдери. До примітивних колайдерів відносяться колайдери у формі

- куба
- капсули
- сфери

В ідеальному випадку колайдер має повторювати форму 3D моделі. Кожен об'єкт на сцені може мати декілька колайдерів. На малюнку вони позначені зеленими лініями

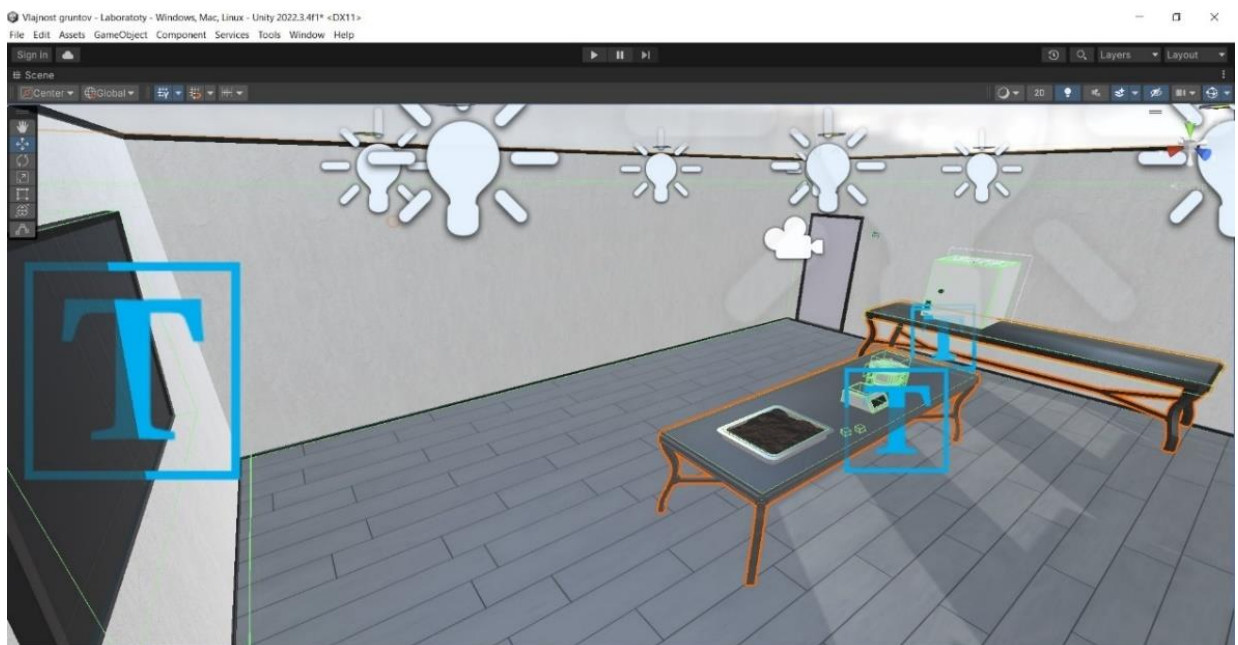


Рисунок 7.1 – Імпортовані 3D моделі до ігрового рушія з колайдерами

При розрахунку фізичної взаємодії на колайдери складної форми накладено обмеження: вони не можуть бути вгнутими. За для подолання цього обмеження потрібно створити необхідну кількість примітивних колайдерів, що можуть охопити потрібну нам зону.

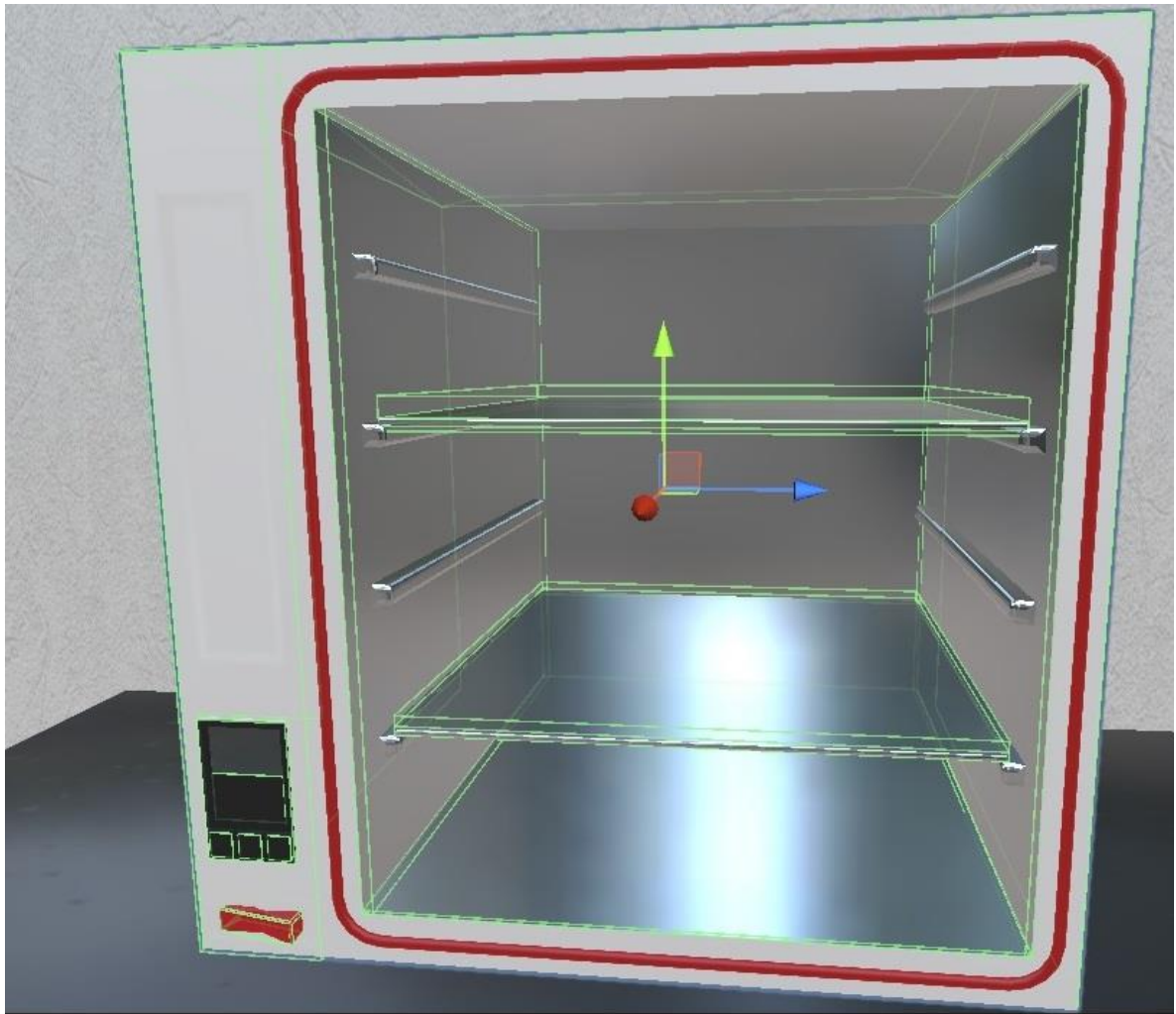


Рисунок 7.2– Створення колайдеру ігрової шафи

Документація Unity вказує на те, що через неодночасний прорахунок фізичної взаємодії об'єктів на сцені та їх рендерінгу існує вірогідність, некоректного прорахунку поведінки об'єкта в цілому. Іншими словами, об'єкт може пролетіти скрізь колайдер між двома ітераціями обчислення фізичної взаємодії. Для уникнення подібної ситуації розробники Unity пропонують збільшити частоту прорахунку фізичної взаємодії. Але це несе додаткове

навантаження на ресурси комп'ютера. Тому було прийнято рішення створити колайдери, габаритні розміри яких гарантуватимуть нам виклик ітерації обчислення фізичної взаємодії поки наш об'єкт ще знаходиться у середині колайдера іншого об'єкту. Це рішення було реалізоване для потенційно небезпечних зон, виявлених при ретельному аналізі наявної сцени.

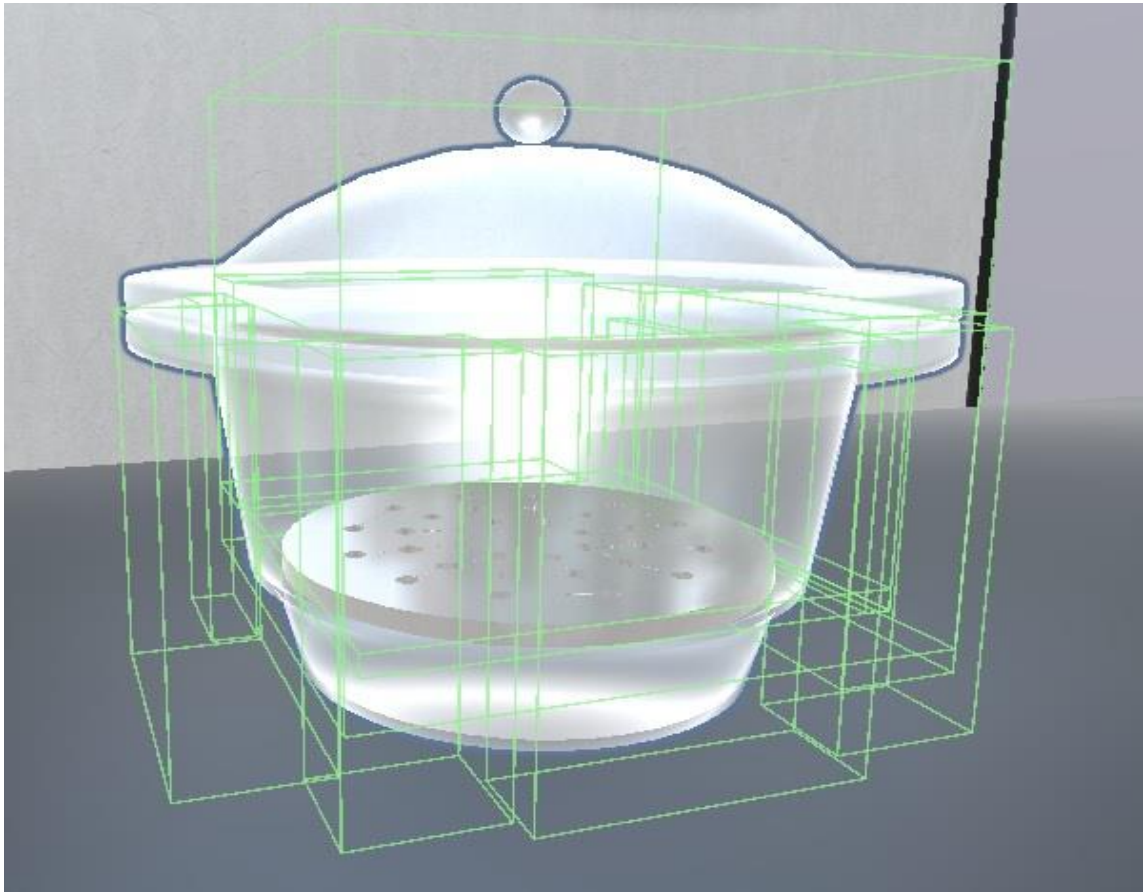


Рисунок 7.3 – Створення колайдеру ексикатора

## 8 СТВОРЕННЯ СКРИПТІВ

Поведінкою ігрових об'єктів керують Компоненти, що до них приєднані. Unity дозволяє створювати власні компоненти за допомогою скриптів. Вони дозволяють запускати внутрішньоігрові події, змінювати налаштування компонентів і реагувати на введення користувача будь-яким способом. Unity за замовчуванням підтримує три мови програмування: C# (вимовляється як Csharp), стандартна мова, схожа на Java або C++; UnityScript, мова, розроблена спеціально для використання в Unity, створена за зразком JavaScript;

Структура файлу скрипта:

```
using UnityEngine;
using System.Collections;
public class MainPlayer : MonoBehaviour
{
    // Use this for initialization
    void Start () { }
    // Update is called once per frame
    void Update () { }
}
```

Скрипт взаємодіє з внутрішніми механізмами Unity, створюючи клас, який успадковується від вбудованого класу під назвою MonoBehaviour. Щоразу, коли ви прикріплюєте компонент сценарію до ігрового об'єкта, створюється новий екземпляр цього сценарію. Ім'я класу береться з імені, яке ви вказали під час створення файлу. Ім'я класу та ім'я файлу повинні бути однаковими, щоб скриптовий компонент був прикріплений до об'єкта гри.

## ВИСНОВКИ

1. Особлива актуальність створення системи дистанційної освіти сьогодні обумовлена в цілому рядом факторів: військовий стан, епідеміологічна ситуація, посилення міграції населення, розвиток науково-технічних центрів у великих містах, формування нових потреб населення у розвитку.

2. Дистанційне навчання забезпечує ефективну взаємодію між студентами та викладачами. Водночас дистанційне навчання має ряд серйозних недоліків серед яких є питання організації практичних та лабораторних робіт в онлайн-форматі. Мало здобути нові знання. Потрібно навчитися застосовувати їх у житті.

3. Таким чином, щоб допомогти студентам у навчанні, підвищити їх інтерес до професійної галузі, а також покращити розуміння матеріалу, необхідні віртуальні лабораторні комплекси, які не лише дозволяють наочно пояснити та показати роботу на обладнанні, але є безпечнішими та дешевшими. Звичайно, робота з реальним обладнанням все одно має бути головним у цьому питанні, але з урахуванням ситуації, що склалася в Україні, використання віртуальних лабораторних робіт актуальне і досить перспективне.

4. Завдяки обраним програмним продуктам було створено програмне забезпечення, що дозволяє студентам вивчати фізичні властивості ґрунтів у випадку, коли відвідування лабораторії з тих чи інших причин неможливе.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Google SketchUp. URL: <http://www.sketchup.google.com/> (дата звернення: 04.11.2023).
2. Blender. URL: <http://www.blender.org/> (дата звернення: 04.11.2023).
3. Wings 3D. URL: <http://www.wings3d.com/> (дата звернення: 04.11.2023).
4. Sweet Home 3D. URL: <http://www.sweethome3d.com> (дата звернення: 05.11.2023).
5. K-3D. URL: <http://www.k-3d.org/> (дата звернення: 04.11.2023).
6. Free CAD. URL: <http://www.freecadweb.org/> (дата звернення: 05.11.2023).
7. K3DSurf URL: <http://www.k3dsurf.sourceforge.net/> (дата звернення: 04.11.2023).
8. Текстура (тривимірна графіка) URL: [https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D1%83%D1%80%D0%B0\\_\(%D1%82%D1%80%D0%B8%D0%B2%D0%B8%D0%BC%D1%96%D1%80%D0%BD%D0%B0\\_%D0%B3%D1%80%D0%B0%D1%84%D1%96%D0%BA%D0%B0\)](https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D1%83%D1%80%D0%B0_(%D1%82%D1%80%D0%B8%D0%B2%D0%B8%D0%BC%D1%96%D1%80%D0%BD%D0%B0_%D0%B3%D1%80%D0%B0%D1%84%D1%96%D0%BA%D0%B0)) (дата звернення: 04.11.2023).

ДОДАТОК А  
НЕОБХІДНІ СКРИПТИ ДЛЯ ЗАБЕЗПЕЧЕННЯ ПРОЦЕСУ ВЗАЄМОДІЇ  
ОБ'ЄКТІВ

Скрипт, що відповідає за збільшення температури у сушильній шафі

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using Unity.VisualScripting;
using UnityEngine.UIElements;

public class IncreaseTemperature : MonoBehaviour
{
    [SerializeField] TMP_Text Display;
    [SerializeField] int NeededTemp;
    bool start = false;
    void Inc_1C()
    {
        NeededTemp++;
        if (NeededTemp > 320)
        {
            NeededTemp = 320;
            Display.text = NeededTemp.ToString();
            return;
        }

        Display.text = NeededTemp.ToString();
    }
    void Inc_10C()
    {
        NeededTemp += 10;
        if (NeededTemp > 320)
        {
            NeededTemp = 320;
            Display.text = NeededTemp.ToString();
            return;
        }
        Display.text = NeededTemp.ToString();
    }
}
```

```

public void IncreaseTemp()
{
    int.TryParse(Display.text, out NeededTemp);
    if (NeededTemp == 320) return;
    if (!start)
        StartCoroutine(plusTemp());
}
IEnumerator plusTemp ()
{
    start = true;
    int counter = 0;
    while (Input.GetKey(KeyCode.Escape))
    {
        if (counter < 4) Inc_1C();
        else Inc_10C();
        yield return new WaitForSeconds(0.4f);
        counter++;
    }
    start = false;
    counter = 0;
}
}
}

```

Скрипт, що відповідає за стан стакану та за вивід інформації про нього

```

using System.Collections;
using System.Collections.Generic;
using System.Globalization;
using System.Text;
using TMPro;
using UnityEngine;

```

```

public class Stakan : MonoBehaviour

```

```

{

    public bool Dehydrated = false;
    public bool CompletelyDehydrated=false;
    public float m0;
    public double m1;
    public double m2;
    public double W;
    string InfoString;
    [SerializeField] private TextMeshProUGUI TV;

    void Awake()
    {
        System.Random rnd = new System.Random();
        GetComponent<Rigidbody>().mass = 0.019f+(float)rnd.Next(-5,5)/10000;
        transform.GetChild(0).GetComponent<Rigidbody>().mass =
0.0000000000000001f;

    }

    public void GenerateStringInfo()
    {
        m0 = GetComponent<Rigidbody>().mass * 1000;
        m2 = System.Math.Round((GetComponent<Rigidbody>().mass +
transform.GetChild(0).GetComponent<Rigidbody>().mass),5)*1000;
        W = System.Math.Round(((m1 - m2) / (m2 - m0)) ,4)*100;
        InfoString = "\n" + " " + this.name.ToString() + "  m0 = " + m0.ToString() +
"  m1 = " + m1.ToString() + "  m2 = " + m2.ToString() + "  W = " +W.ToString()
+"%";
        UpdateTv();

    }

    void UpdateTv()
    {
        StringBuilder sb = new StringBuilder();
        sb.Append(TV.text);
    }
}

```

```

        sb.Append(InfoString);
        TV.text = sb.ToString();
    }
}

```

Скрипт, що емулює процес сушіння

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Dehydrogenating : MonoBehaviour

{
    public static List<GameObject> StayInside = new List<GameObject>();

    void OnTriggerEnter(Collider other)
    {
        StayInside.Add(other.gameObject);
    }

    void OnTriggerExit(Collider other)
    {
        StayInside.Remove(other.gameObject);
    }

    public void Dehydrate()
    {
        foreach (GameObject go in StayInside)
        {
            Stakan _stakan = go.GetComponent<Stakan>();
            if (_stakan &&
go.transform.GetChild(0).GetComponent<MeshRenderer>().enabled==true)
            {
                if (_stakan.Dehydrated == true && _stakan.CompletelyDehydrated == false)
                    dehydrateIteration_2(go);
            }
        }
    }
}

```

```

        if (_stakan.Dehydrated == false && _stakan.CompletelyDehydrated ==
false)
            dehydrateIteration_1(go);
        }
    }
}
private void dehydrateIteration_1 (GameObject go)
{
decimal tmp;
tmp=(decimal)(go.transform.GetChild(0).GetComponent<Rigidbody>().mass/100
*9.2);
go.transform.GetChild(0).GetComponent<Rigidbody>().mass -= (float)tmp;
go.GetComponent<Stakan>().Dehydrated = true;
go.transform.GetChild(0).GetComponent<MeshRenderer>().material.color =
Color.yellow;
}
private void dehydrateIteration_2(GameObject go)
{
    System.Random rnd = new System.Random();
go.transform.GetChild(0).GetComponent<Rigidbody>().mass -= (float)rnd.Next(0,
40) / 1000000;
    go.GetComponent<Stakan>().CompletelyDehydrated = true;
    go.GetComponent<Stakan>().GenerateStringInfo();
}
}
}

```

Скрипт, що відповідає за взяття предмета до рук

```

using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using Unity.VisualScripting;
using UnityEngine;
using UnityEngine.XR;

public class DragableObject : MonoBehaviour
{

```

```

public GameObject handTake;
    public void Drag(Collider _selected)
    {

        if (handTake.GetComponent<HandOfStudent>().isFull == false)
        {

            _selected.gameObject.GetComponent<Rigidbody>().isKinematic = true;
            _selected.gameObject.GetComponent<Rigidbody>().useGravity = false;
            // _selected.gameObject.GetComponent<BoxCollider>().enabled = false;
            _selected.gameObject.GetComponent<DragableObject>().transform.parent
= handTake.transform;

            _selected.gameObject.GetComponent<DragableObject>().transform.position =
handTake.GetComponent<Transform>().position;

            _selected.gameObject.GetComponent<DragableObject>().transform.rotation =
handTake.GetComponent<Transform>().rotation;
            handTake.GetComponent<HandOfStudent>().isFull = true;

        }
    }
}

```

Скрипт, що відповідає за керування дисплеєм

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;

public class DisplayOnOff : MonoBehaviour
{
    [SerializeField] TMP_Text Display;
    bool DisplayON = false;

    public void OnOff()

```

```

    {
        DisplayON = !DisplayON;
        Display.enabled = DisplayON;
    }
}

```

Скрипт, що відповідає за випускання предмету з рук та постановку його на потрібне місце

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DropableObject : MonoBehaviour
{
    public GameObject handTake;

    // Start is called before the first frame update

    public void Drop(Vector3 _selectedDropPosition, Collider NewParent)
    {
        if (handTake.GetComponent<HandOfStudent>().isFull == true)
        {
            Vector3 _pointToSet = _selectedDropPosition;

            handTake.transform.GetChild(0).transform.position = _selectedDropPosition;
            handTake.transform.GetChild(0).transform.rotation = new
            Quaternion(this.transform.rotation.x,this.transform.rotation.y,0,this.transform.rotation.w);
            handTake.transform.GetChild(0).GetComponent<Rigidbody>().isKinematic =
            false;
            handTake.transform.GetChild(0).GetComponent<Rigidbody>().useGravity =
            true;
            handTake.transform.GetChild(0).GetComponent<BoxCollider>().enabled =
            true;

            handTake.GetComponent<HandOfStudent>().isFull = false;
            handTake.transform.GetChild(0).parent = null;

```

```

    }
  }
}

```

Скрипт, що відповідає за наповнення стакана ґрунтом

```

using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class FillableObject : MonoBehaviour
{
    [SerializeField] GameObject handTake;
    [SerializeField] float _massOfGrunt;
    // Start is called before the first frame update

    public void Fill()
    {
        if (handTake.GetComponent<HandOfStudent>().isFull == true)
        {
            if
            (handTake.transform.GetChild(0).transform.GetChild(0).GetComponent<MeshRe
            nderer>().enabled == false)
            {
                System.Random rnd = new System.Random();
                _massOfGrunt = 0.015f + (float)rnd.Next(-10, 10)/10000- (float)1e-07;

                handTake.GetComponent<HandOfStudent>().transform.GetChild(0).transform.Ge
                tChild(0).GetComponent<Rigidbody>().mass += _massOfGrunt;

                handTake.transform.GetChild(0).transform.GetChild(0).GetComponent<MeshRen
                derer>().enabled = true;
            }
        }
    }
}

```

```

        handTake.transform.GetChild(0).GetComponent<Stakan>().m1 =
        Math.Round((double)_massOfGrunt +
        (double)handTake.transform.GetChild(0).GetComponent<Rigidbody>().mass, 5) *
        1000;
    }
}
}
}

```

Скрипт, що відповідає за поточну точку спостереження

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class TargetFinder : MonoBehaviour
{
    public Transform CameraTransform;
    public float Size = 0.011f;

    void LateUpdate()
    {
        float scale = Vector3.Distance(transform.position,
        CameraTransform.position);
        transform.localScale = Vector3.one * scale * Size;
    }
}

```

Скрипт, що відповідає за передстартові перевірки в процесі роботи сушильної шафи

```

using System.Collections;
using System.Collections.Generic;
using System.Linq;
using TMPro;
using Unity.VisualScripting;

```

```

using UnityEngine;
using UnityEngine.Assertions.Must;

public class StartDehydrating : MonoBehaviour
{
    [SerializeField] GameObject Polka;
    [SerializeField] TextMeshProUGUI Display;
    [SerializeField] GameObject Door;
    float ClosedDoorRotation;
    void Start()
    {
        ClosedDoorRotation = Door.transform.rotation.y;
    }

    public void StartDehydrate()
    {
        float CurrentDoorRotation = Door.transform.rotation.y;
        TextMeshProUGUI DisplayText =
Display.GetComponentInChildren<TextMeshProUGUI>();

        if (CurrentDoorRotation == ClosedDoorRotation)
        {
            if (DisplayText.enabled == true && DisplayText.text == "105")
            { Polka.GetComponent<Dehydrogenating>().Dehydrate(); }
        }
    }
}

```

Скрипт, що відповідає за відкривання та закривання

```

using System.Collections;
using System.Collections.Generic;
using Unity.VisualScripting;
using UnityEngine;

public class OpenableClosableObject : MonoBehaviour

```

```

{
    bool Open = false;

    public void OpenClose(Collider _selected)
    {

        Open = !Open;
        GetComponentInParent<Animator>().SetBool ("Open", Open);
        GetComponentInParent<Animator>().SetBool ("Close", !Open);
    }
}

```

Скрипт, що відповідає за процес зважування

```

using System;
using System.Collections;
using System.Collections.Generic;
using TMPro;
using Unity.VisualScripting;
using UnityEngine;

public class GetWeightableObject : MonoBehaviour
{
    [SerializeField] TMP_Text Display;
    [SerializeField] double mass = 0;
    // Start is called before the first frame update

    void OnTriggerEnter(Collider other)
    {
        mass += other.gameObject.GetComponent<Rigidbody>().mass +
other.gameObject.transform.GetChild(0).GetComponent<Rigidbody>().mass;
        Display.text= (Math.Round(mass, 5) * 1000).ToString() + " g";
    }

    void OnTriggerExit(Collider other)
    {
        mass -= other.gameObject.GetComponent<Rigidbody>().mass +
other.gameObject.transform.GetChild(0).GetComponent<Rigidbody>().mass;
        Display.text = (Math.Round(mass, 5) * 1000).ToString() + " g";
    }
}

```

```
}
```

```
}
```

Скрипт, що відповідає за керування освітленням

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Experimental.GlobalIllumination;

public class LightOnOff : MonoBehaviour
{
    [SerializeField] GameObject ContainerOfRoomLights;
    [SerializeField] GameObject Svetilnik;
    Component[] Lampochki;
    bool LightON = false;

    public void OnOff()
    {
        LightON = !LightON;
        Lampochki = ContainerOfRoomLights.GetComponentsInChildren<Light>();
        foreach (Light light in Lampochki) { light.enabled = LightON; }
        Svetilnik.GetComponentInChildren<MeshRenderer>().enabled = !LightON;
    }
}
```

Скрипт, що відповідає за обертання голови персонажу

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[AddComponentMenu("Camera-Control/StudentEyesView")]
public class MouseLook : MonoBehaviour
{
    public enum RotationAxes { MouseXandY = 0, MouseX = 1, MouseY = 2 };
    public RotationAxes axes = RotationAxes.MouseXandY;
    public float sensitivityX = 2f;
    public float sensitivityY = 2f;
    public float minimumX = -360f;
    public float maximumX = 360f;

    public float minimumY = -360f;
    public float maximumY = 360f;
    float rotationX = 0f;
    float rotationY = 0f;
    Quaternion originalRotation;

    void Start()
    {
        if (GetComponent<Rigidbody>())
        {
            GetComponent<Rigidbody>().freezeRotation = true;
            originalRotation = transform.localRotation;
        }
    }
    public static float ClampAngle(float angle, float min, float max)
    {
        if (angle < -360f) angle += 360f;
        if (angle > 360f) angle -= 360f;
        return Mathf.Clamp(angle, min, max);
    }
    void Update()
    {
        if (axes == RotationAxes.MouseXandY)
        {
            rotationX += Input.GetAxis("Mouse X") * sensitivityX;

```

```

rotationY += Input.GetAxis("Mouse Y") * sensitivityY;
rotationX = ClampAngle(rotationX, minimuX, maximumX);
rotationY = ClampAngle(rotationY, minimuY, maximumY);
Quaternion xQuaternion = Quaternion.AngleAxis(rotationX, Vector3.up);
Quaternion yQuaternion = Quaternion.AngleAxis(rotationY, -
Vector3.right);
transform.localRotation = originalRotation * xQuaternion * yQuaternion;
}
else if (axes == RotationAxes.MouseX)
{
rotationX += Input.GetAxis("Mouse X") * sensitivityX;
rotationX = ClampAngle(rotationX, minimuX, maximumX);
Quaternion xQuaternion = Quaternion.AngleAxis(rotationX, Vector3.up);
transform.localRotation = originalRotation * xQuaternion;
}
else if (axes == RotationAxes.MouseY)
{
rotationY += Input.GetAxis("Mouse Y") * sensitivityY;
rotationY = ClampAngle(rotationY, minimuY, maximumY);
Quaternion yQuaternion = Quaternion.AngleAxis(-rotationY, Vector3.right);
transform.localRotation = originalRotation * yQuaternion;
}
}
}
}

```

Скрипт, що відповідає за рух персонажа у просторі

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class characterControl : MonoBehaviour
{
public float StudetnSpeed = 4.0f;
public float StudentJumpSpeed = 8.0f;
public float StudentGravity = 10.0f;
public Vector3 StudentMovingDirection = Vector3.zero;

```

```

private CharacterController Controller;

void Start()
{
    Controller = GetComponent<CharacterController>();
}

void Update()
{
    if (Controller.isGrounded)
    {
        StudentMovingDirection = new Vector3(Input.GetAxis("Horizontal"), 0,
Input.GetAxis("Vertical"));
        StudentMovingDirection =
transform.TransformDirection(StudentMovingDirection);
        StudentMovingDirection *= StudentSpeed;
    }

    if (Input.GetKey(KeyCode.Space) && Controller.isGrounded)
    {
        StudentMovingDirection.y = StudentJumpSpeed;
    }
    StudentMovingDirection.y -= StudentGravity * Time.deltaTime;
    Controller.Move(StudentMovingDirection * Time.deltaTime);
}
}

```

Скрипт, що відповідає за механіку симуляційного процесу

```

using System.Collections;
using System.Collections.Generic;
using System.Reflection;
using Unity.VisualScripting;
using UnityEngine;
using UnityEngine.Assertions.Must;

```

```

public class StudentRay : MonoBehaviour
{
    public Transform Pointer;
    public Selectable CurrentSelectable;
    public RaycastHit hit;

    void LateUpdate()

    {

        Vector3 Ray_start_position = new Vector3(Screen.width / 2, Screen.height / 2,
0);
        Ray ray = Camera.main.ScreenPointToRay(Ray_start_position);
        Debug.DrawRay(transform.position, transform.forward*100, Color.yellow);

        if (Physics.Raycast(ray, out hit))
        {
            Pointer.position = hit.point;

Selectable selectable = hit.collider.gameObject.GetComponent<Selectable>();
DragableObject draggable =
hit.collider.gameObject.GetComponent<DragableObject>();
DropableObject dropable =
hit.collider.gameObject.GetComponent<DropableObject>();
OpenableClosableObject openableClosable =
hit.collider.gameObject.GetComponent<OpenableClosableObject>();
FillableObject fillAble =
hit.collider.gameObject.GetComponent<FillableObject>();
StartDehydrating startSushka =
hit.collider.gameObject.GetComponent<StartDehydrating>();
LightOnOff SwitcherLight =
hit.collider.gameObject.GetComponent<LightOnOff>();
DisplayOnOff SwitcherDisplay =
hit.collider.gameObject.GetComponent<DisplayOnOff>();

```

```

DecreaseTemperature MinusTemp =
hit.collider.gameObject.GetComponent<DecreaseTemperature>();
IncreaseTemperature PlusTemp =
hit.collider.gameObject.GetComponent<IncreaseTemperature>();

    if (selectable)
    {
        if (CurrentSelectable && CurrentSelectable != selectable)
        {
            CurrentSelectable.Deselect();
        }
        CurrentSelectable = selectable;
        if (hit.distance < 2)
        {

            selectable.Select();

        }
    }
}
else
{
    if (CurrentSelectable != null)
    {

```

```

        CurrentSelectable.Deselect();
        CurrentSelectable = null;
    }
}
else
{
    if (CurrentSelectable != null)
    {
        CurrentSelectable.Deselect();
        CurrentSelectable = null;
    }
}
}
}

```

Скрипт, що відповідає за приближення

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Zoom : MonoBehaviour
{
    public Camera cam;
    public float defaultFov = 90;

    void Update()
    {
        if (Input.GetMouseButton(1))
        {
            cam.fieldOfView = (defaultFov / 2.5f);
        }
        else
        {
            cam.fieldOfView = (defaultFov);
        }
    }
}

```

Скрипт, що відповідає за зменшення температури у сушильній шафі

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using Unity.VisualScripting;
using UnityEngine.UIElements;

public class DecreaseTemperature : MonoBehaviour
{
    [SerializeField] TMP_Text Display;
    [SerializeField] int NeededTemp;
    bool start = false;

    void Dec_1C()
    {
        NeededTemp--;
        if (NeededTemp < 0)
        {
            NeededTemp = 0;
            Display.text = NeededTemp.ToString();
            return;
        }
        Display.text = NeededTemp.ToString();
    }

    void Dec_10C()
    {
        NeededTemp -= 10;
        if (NeededTemp < 0 )
        {
            NeededTemp = 0;
            Display.text = NeededTemp.ToString();
            return;
        }
        Display.text = NeededTemp.ToString();
    }
}

```

```
}  
public void DecreaseTemp()  
{  
    int.TryParse(Display.text, out NeededTemp);  
    if (NeededTemp == 0) return;  
    if (!start)  
        StartCoroutine(plusTemp());  
  
}
```

```
IEnumerator plusTemp()  
{  
    start = true;  
    int counter = 0;  
    while (Input.GetKey(KeyCode.Escape))  
    {  
        if (counter < 4) Dec_1C();  
        else Dec_10C();  
        yield return new WaitForSeconds(0.4f);  
        counter++;  
    }  
    start = false;  
    counter = 0;  
}  
  
}
```

ДОДАТОК Б  
ІЛЮСТРАТИВНИЙ МАТЕРІАЛ

Міністерство освіти і науки України

Харківський національний автомобільно-дорожній університет

Дорожньо-будівельний факультет

Кафедра будівництва та експлуатації автомобільних доріг ім. О. К. Біруля

ІЛЮСТРАТИВНИЙ МАТЕРІАЛ ДО ДИПЛОМНОЇ РОБОТИ

магістра

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ВИЗНАЧЕННЯ ФІЗИЧНИХ ВЛАСТИВОСТЕЙ  
ҐРУНТІВ

Завідувач кафедри канд. техн. наук, доцент

Роман СМОЛЯНЮК

Нормоконтролер канд. техн. наук, доцент

Олександр ВОЛОВИК

Керівник канд. техн. наук, доцент

Андрій СЄДОВ

Студент гр. Д-63-22

Віталій ОНУФРІЄВ

Харків 2023

# Зміст

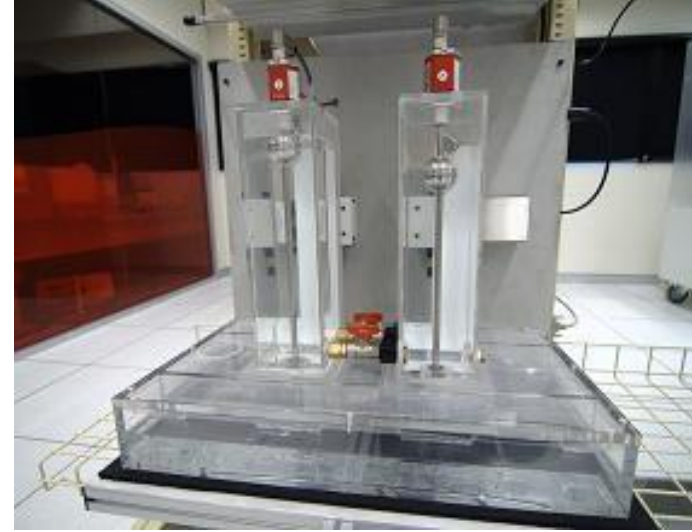
1. Можливості дистанційного навчання
2. Приклади віртуальних лабораторій
3. 3D редактор
4. 3D engine
5. 3D модель лабораторії
6. 3D модель сушильної шафи
7. 3D модель ексикатора
8. UV розгортка
9. Текстурована 3d модель стакану з ґрунтом
10. Імпорт створених моделей у unity
11. Створення колайдерів
12. Створення скриптів для взаємодії об'єктів
13. Висновки

# МОЖЛИВОСТІ ДИСТАНЦІЙНОГО НАВЧАННЯ



# ПРИКЛАДИ ВІРТУАЛЬНИХ ЛАБОРАТОРІЙ

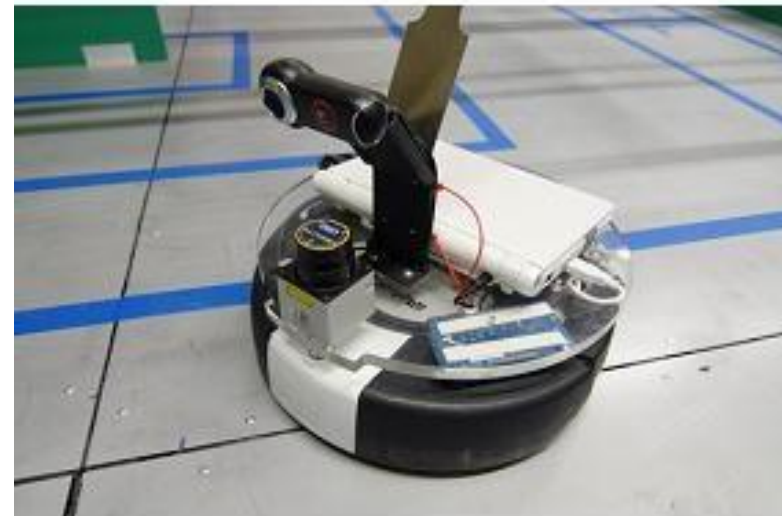
Спарені танки покоління II



Інженерна геологія



iRobot



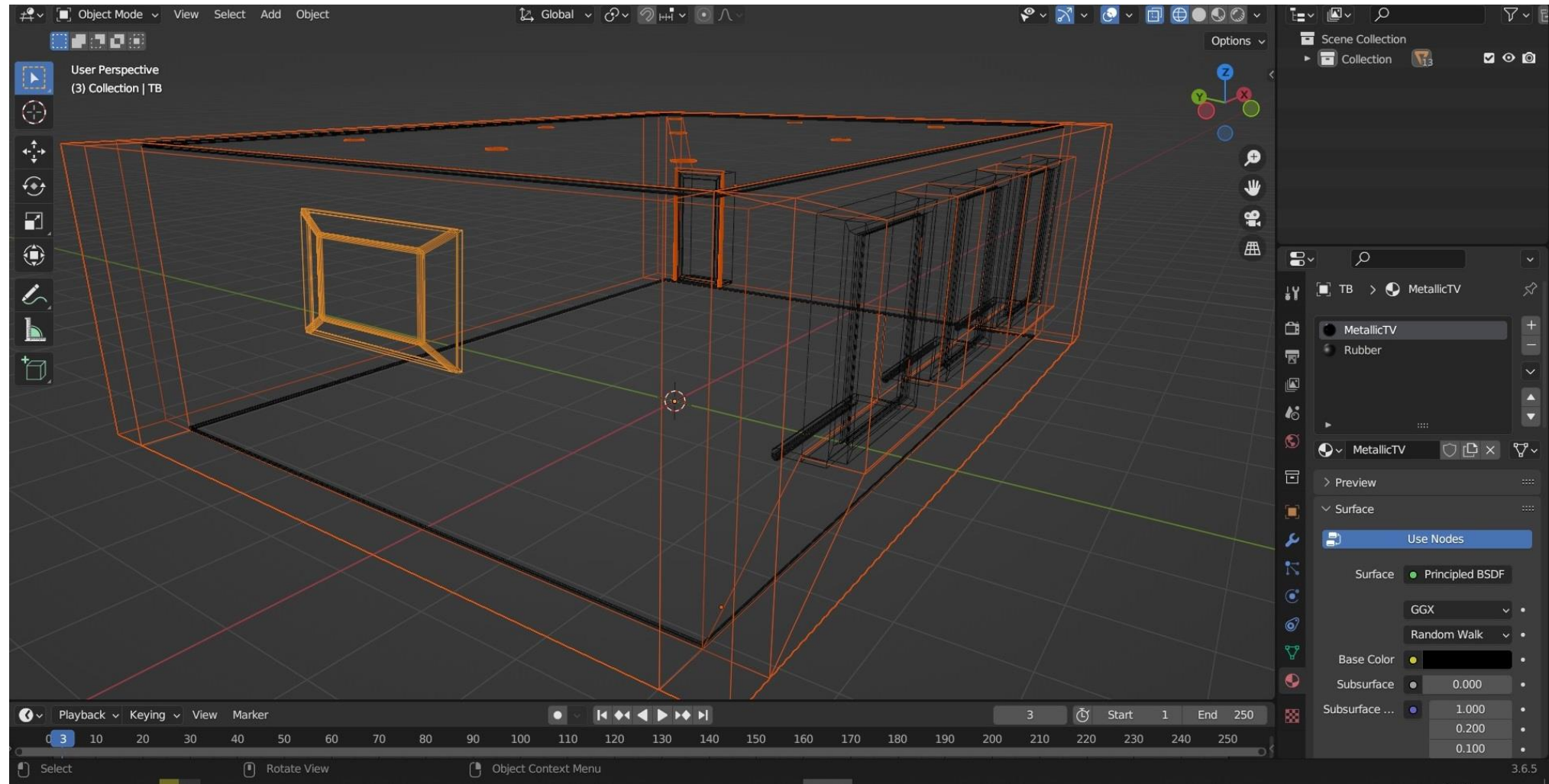
## 3D РЕДАКТОРИ

Назва ПЗ	Призначення ПЗ	Вартість ПЗ
Maya	Професійне ПЗ	215 \$ на місяць
Cinema 4D	Професійне ПЗ	999 \$ на рік
Autodesk 3ds Max	Професійне ПЗ	215 \$ на місяць
Blender 3D	Професійне ПЗ	Безкоштовно
Free CAD	Професійне ПЗ	Безкоштовно
K3DSurf	Професійне ПЗ	Безкоштовно
Google SketchUp	ПЗ для загального користування	Безкоштовно
Wings 3D	ПЗ для загального користування	Безкоштовно
Sweet Home 3D	ПЗ для загального користування	Безкоштовно

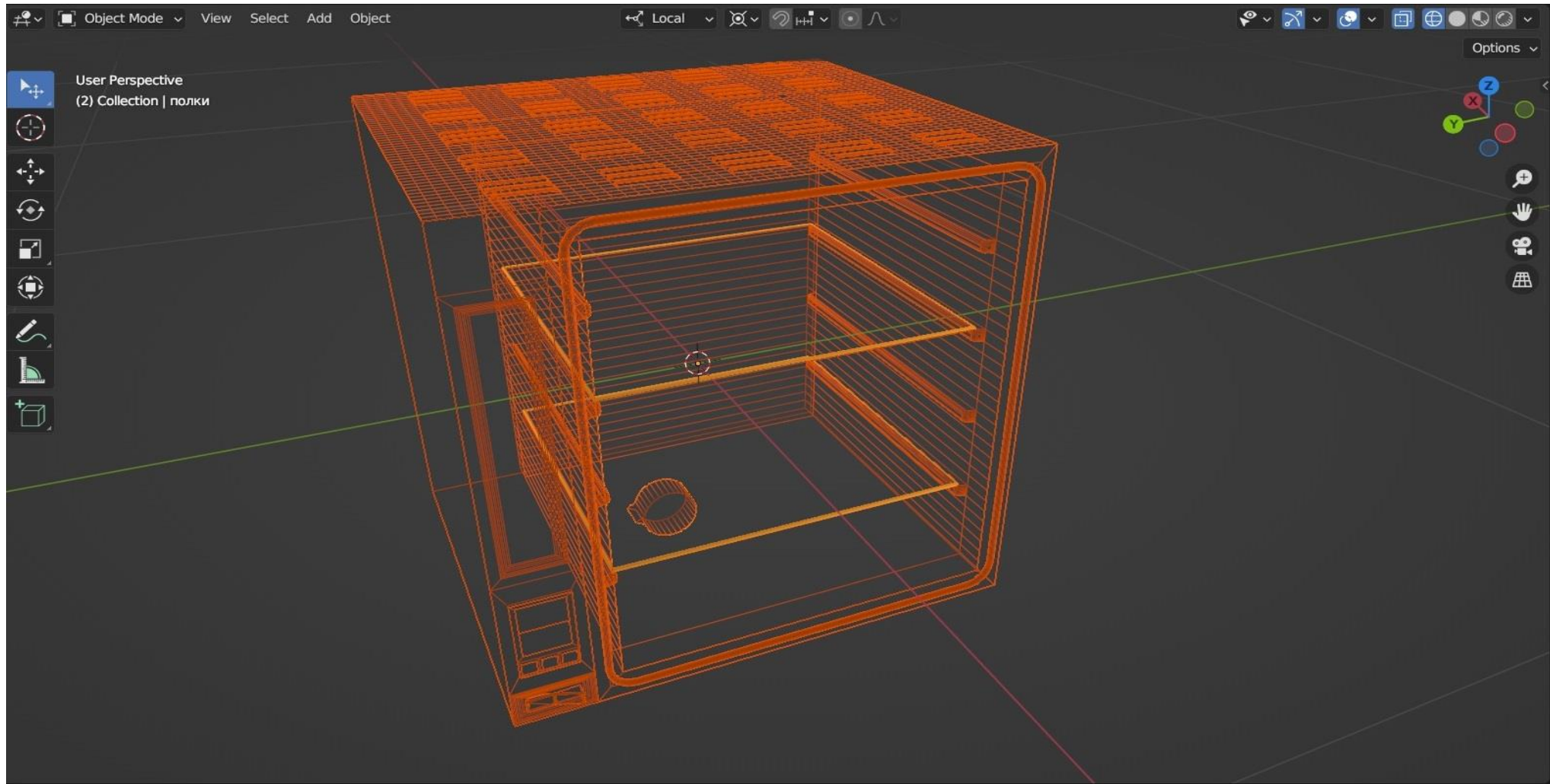
# 3D ENGINE

Назва	Можливість розробки під платформу WebGL	Студентська безкоштовна ліцензія
Unity	є	є
Unreal Engine	немає	немає

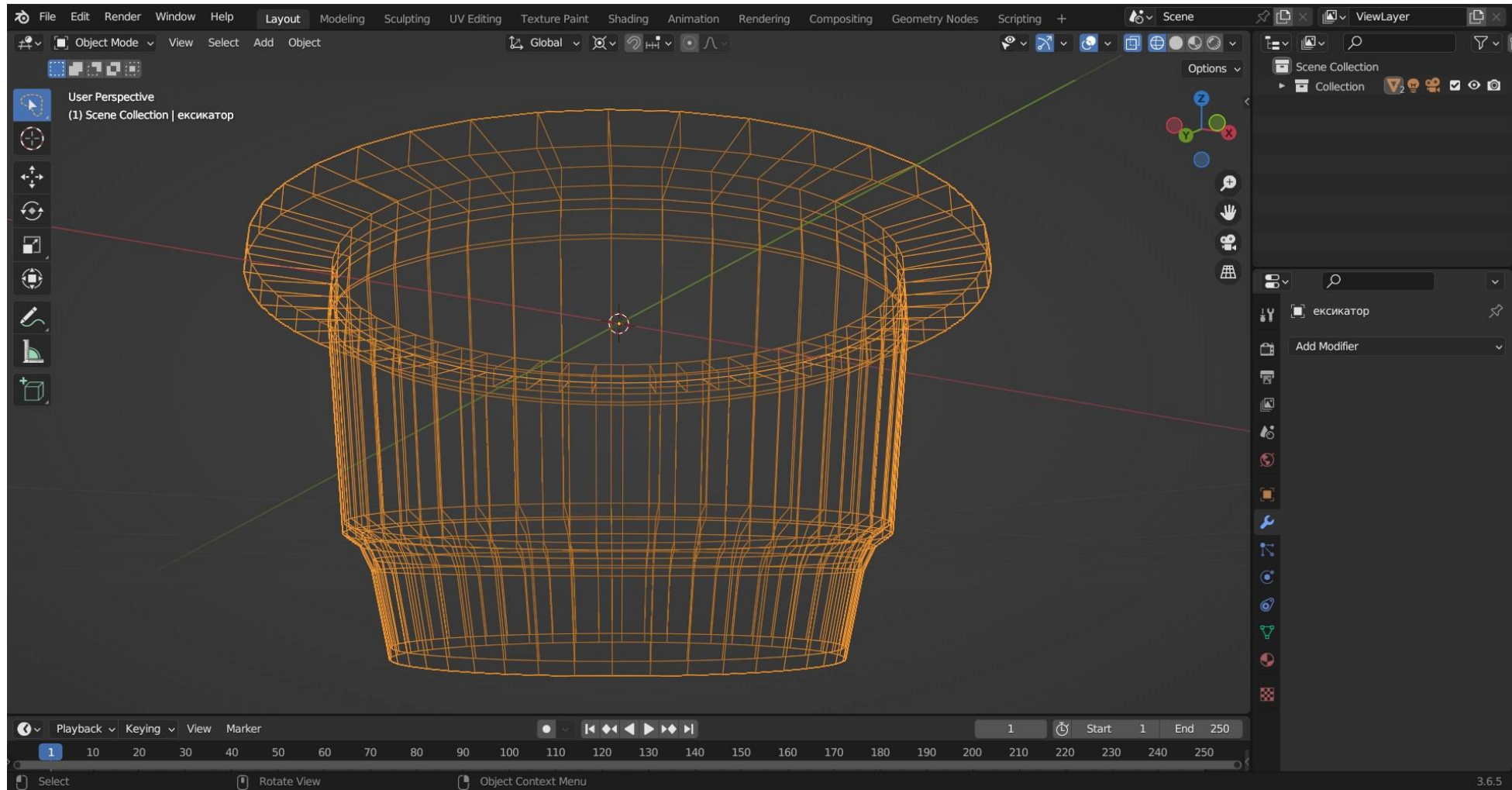
# 3D МОДЕЛЬ ЛАБОРАТОРІЇ



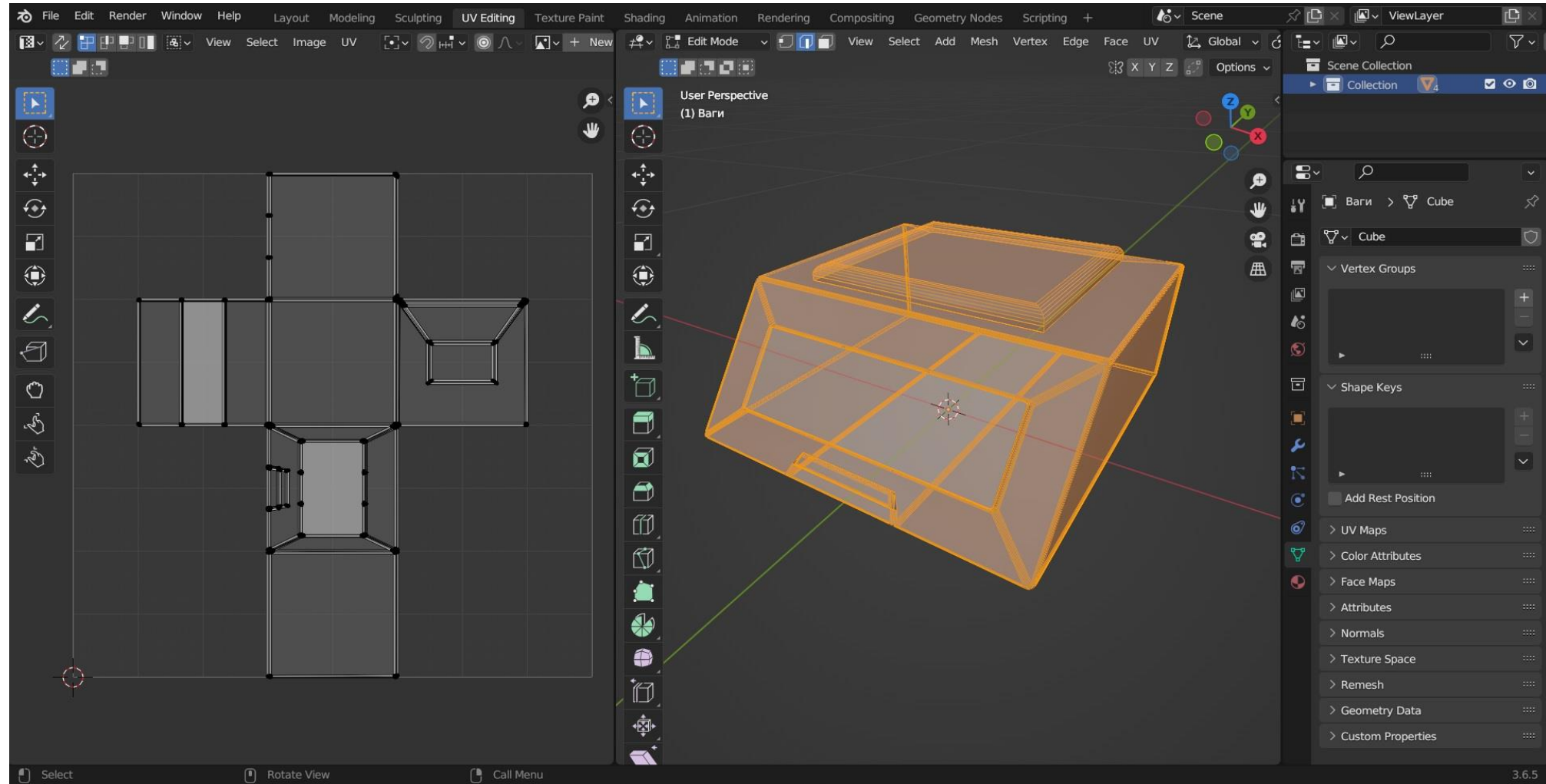
# 3D МОДЕЛЬ СУШИЛЬНОЇ ШАФИ



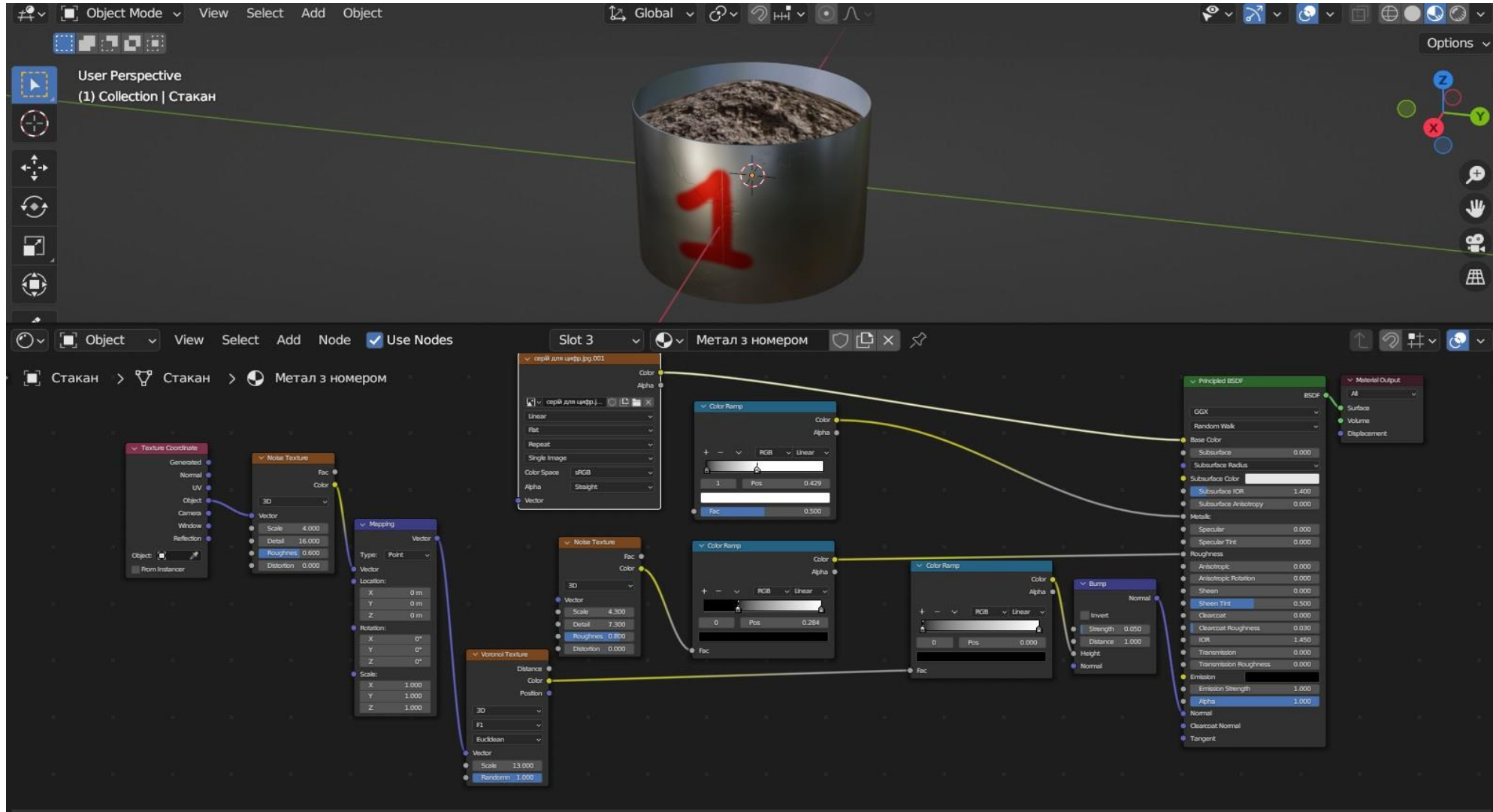
# 3D МОДЕЛЬ ЭКСИКАТОРА



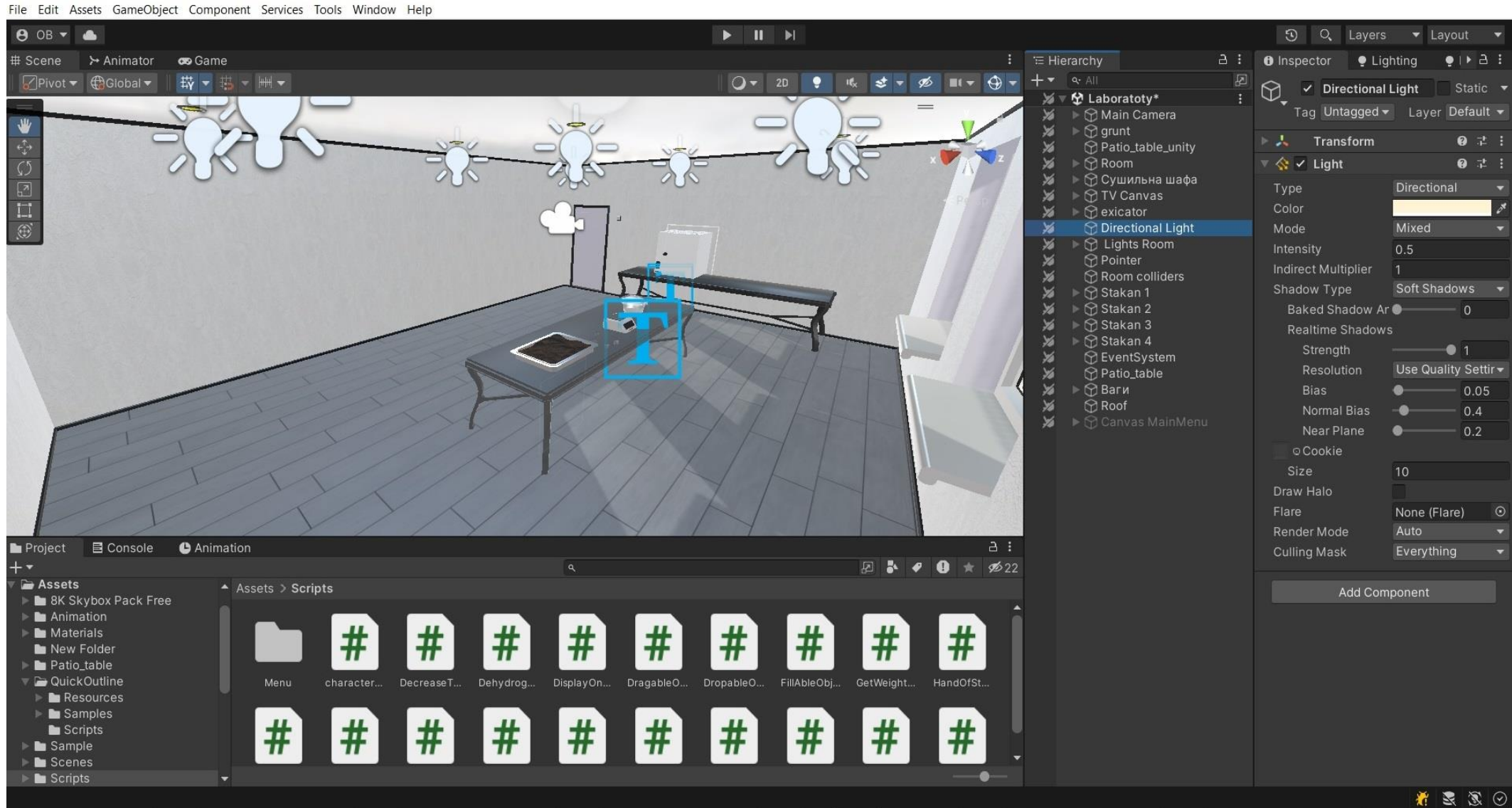
# СТВОРЕННЯ UV РОЗГОРТКИ



# ТЕКСТУРОВАНА 3D МОДЕЛЬ СТАКАНУ З ҐРУНТОМ



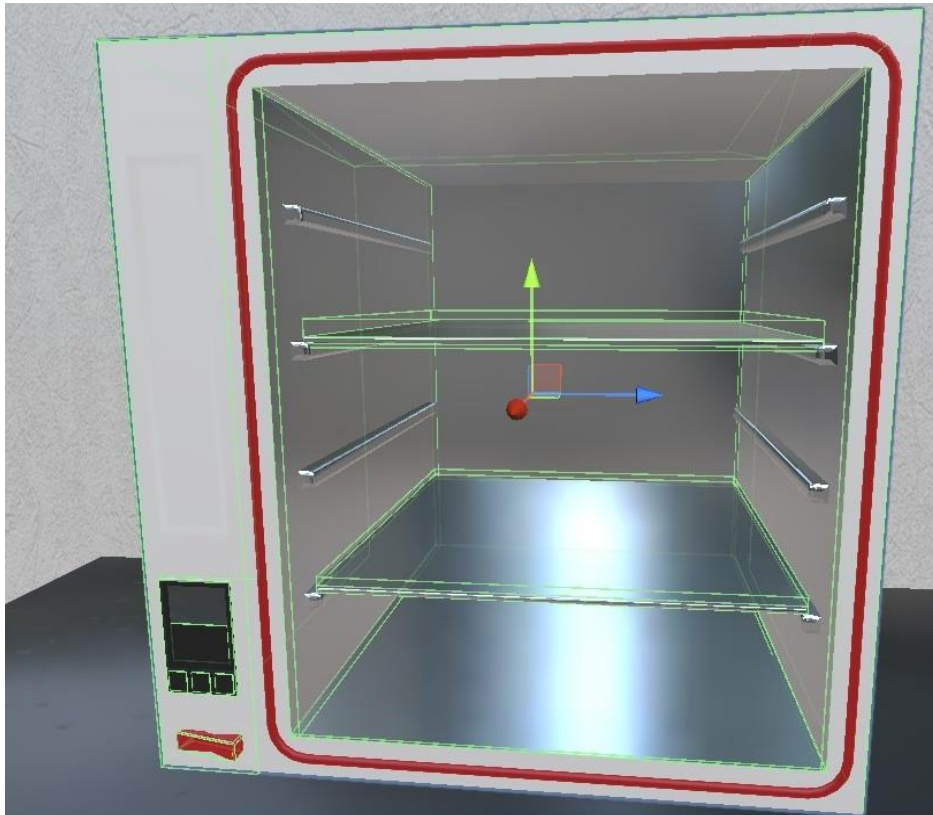
# ІМПОРТ СТВОРЕНИХ МОДЕЛЕЙ У UNITY



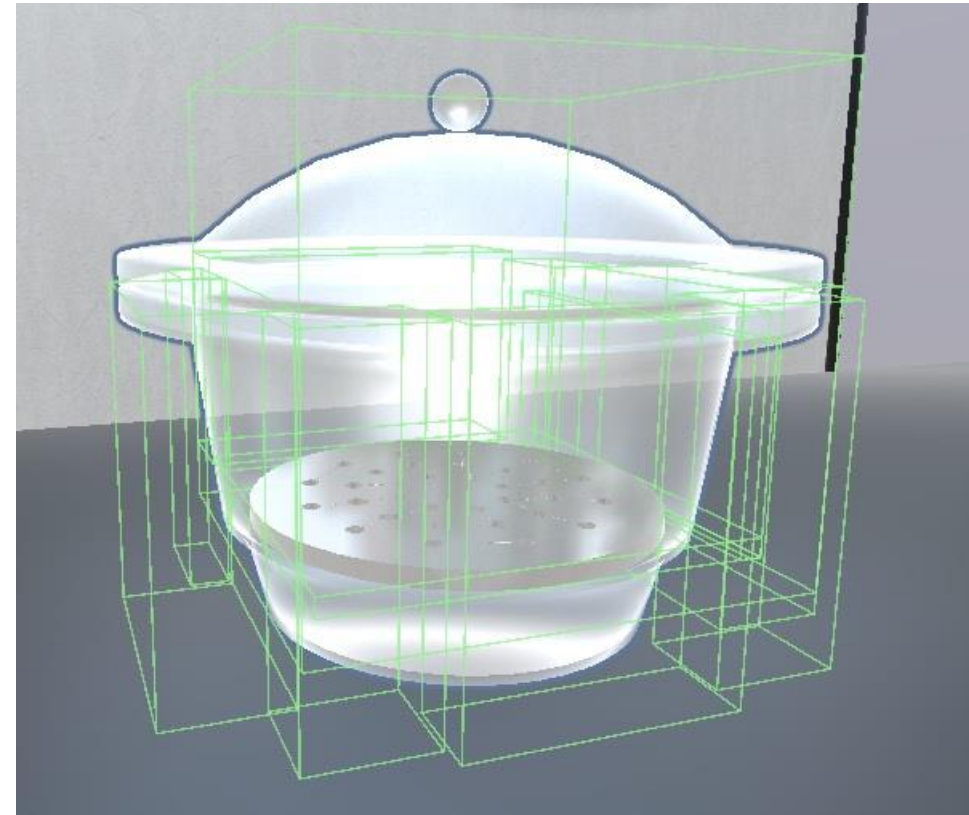
# СТВОРЕННЯ КОЛАЙДЕРІВ

Колайдер – геометрична «оболонка» 3D моделі, що відповідає в першу чергу за фізичну взаємодію з іншими 3D моделями на сцені, що мають свої колайдери. В ідеальному випадку має повторювати форму 3D моделі. При розрахунку фізичної взаємодії на колайдери складної форми накладено обмеження: вони не можуть бути вгнутими.

Колайдер для сушильної шафи



Колайдер для ексикатора



# СТВОРЕННЯ СКРИПТІВ ДЛЯ ВЗАЄМОДІЇ ОБ'ЄКТІВ

- Скрипт – програма, що описує «поведінку» 3D об'єктів на сцені. За для вирішення наявної задачі було створено 17 скриптів, що відповідають за
  - Переміщення по сцені, її огляд та пошук предметів
  - Переміщення предметів по сцені
  - Взаємодія з об'єктами, що можуть бути в стані відчинено\зачинено
  - Наповнення стакану
  - Збереження інформації про актуальний стан стакану
  - Взаємодія з вагами
  - Взаємодія з сушильною шафою
  - Взаємодія з освітленням
  - Меню взаємодії з користувачем

# ВИСНОВКИ

1. Особлива актуальність створення системи дистанційної освіти сьогодні обумовлена в цілому рядом факторів: військовий стан, епідеміологічна ситуація, посилення міграції населення, розвиток науково-технічних центрів у великих містах, формування нових потреб населення у розвитку.
2. Дистанційне навчання забезпечує ефективну взаємодію між студентами та викладачами. Водночас дистанційне навчання має ряд серйозних недоліків серед яких є питання організації практичних та лабораторних робіт в онлайн-форматі. Мало здобути нові знання. Потрібно навчитися застосовувати їх у житті.
3. Таким чином, щоб допомогти студентам у навчанні, підвищити їх інтерес до професійної галузі, а також покращити розуміння матеріалу, необхідні віртуальні лабораторні комплекси, які не лише дозволяють наочно пояснити та показати роботу на обладнанні, але є безпечнішими та дешевшими. Звичайно, робота з реальним обладнанням все одно має бути головним у цьому питанні, але з урахуванням ситуації, що склалася в Україні, використання віртуальних лабораторних робіт актуальне і досить перспективне.
4. Завдяки обраним програмним продуктам було створено програмне забезпечення, що дозволяє студентам вивчати фізичні властивості ґрунтів у випадку, коли відвідування лабораторії з тих чи інших причин неможливе.