

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

«Допущено до захисту»

В.о. завідувача кафедри

_____ Моїсеєнко Н.В.

«___» _____ 2023 р.

Реєстраційний № _____

«___» _____ 2023 р.

**ПОРІВНЯЛЬНИЙ АНАЛІЗ КОНСТРУЮВАННЯ 3D-ОБ'ЄКТІВ
НА РІЗНИХ ГРАФІЧНИХ ПЛАТФОРМАХ**

Кваліфікаційна робота
студента групи І-19
ступінь вищої освіти «бакалавр»
спеціальності
014.09 Середня освіта (Інформатика)
Шестопалова Дмитра Сергійовича

Керівник: кандидат фізико-
математичних наук, доцент
Тарасова Олена Юріївна

Оцінка:

Національна шкала _____

Шкала ECTS ___ Кількість балів _____

Голова ЕК _____

Члени ЕК _____

ЗАПЕВНЕННЯ

Я, Шестопапов Дмитро Сергійович, розумію і підтримую політику Криворізького державного педагогічного університету з академічної доброчесності. Запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Я не надавав і не одержував недозволену допомогу під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають покликання на відповідне джерело. Із чинним Положенням про запобігання та виявлення академічного плагіату в роботах здобувачів вищої освіти Криворізького державного педагогічного університету ознайомлений. Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі порушення академічної доброчесності робота не допускається до захисту або оцінюється незадовільно.



ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1 Дослідження актуальності 3D-моделювання.....	6
1.2 Математичні основи 3D-графіки.....	8
1.3 Основні етапи створення 3D-моделі.....	12
1.4 Огляд та порівняльна характеристика програмних засобів 3D- моделювання.....	13
1.5 Висновки до розділу 1	17
РОЗДІЛ 2. ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЄКТУ	18
2.1 Вибір програмних засобів реалізації 3D-моделей.....	18
2.2 Розробка 3D-моделей.....	22
2.3 Вибір засобів для створення web-сайту.....	27
2.4 Розробка структури та дизайну web-сайту з інтеграцією 3D-моделей ..	29
2.5 Висновки до розділу 2	40
ВИСНОВКИ.....	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	42

ВСТУП

Актуальність теми полягає у тому, що стрімкий розвиток графічних технологій тривимірного моделювання передбачає знаходження алгоритмів для ефективного його застосування у різних галузях. Сучасні обчислювальні системи мають достатню потужність для обробки складних тривимірних сцен у реальному часі без втрати швидкості та якості відображення. Усе вищезазначене спричинило зростання інтересу до тривимірної візуалізації серед фахівців різних галузей.

Водночас, існує велика кількість різних графічних платформ та програм для створення 3D-об'єктів, і кожна з них має свій власний набір переваг та недоліків.

Проведення кваліфікаційного порівняльного аналізу конструювання 3D-об'єктів на різних графічних платформах може бути корисним для фахівців, які займаються 3D-моделюванням, оскільки вони зможуть отримати інформацію про найефективніші та найбільш зручні для використання програми.

Мета кваліфікаційного дослідження – реалізація 3D-моделей об'єктів та їх інтеграція до розробленого сайту.

Для реалізації мети визначені наступні завдання:

1. Дослідити предметну область: використання 3D-об'єктів при створенні сучасних сайтів, комп'ютерних ігор, анімацій тощо.
2. Проаналізувати програмні засоби – графічні платформи для створення 3D-моделей за наступними параметрами: швидкість роботи, якість візуалізації, доступність інструментів моделювання, рівень складності використання.
3. За допомогою обраних програмних засобів (Blender, Tinkercad, Sweet Home 3D) розробити 3D-моделі різних моніторів.
4. Обрати засоби розробки сайту.

5. Розробити структуру та дизайн web-сайту з інтеграцією 3D-моделей, візуалізувати сцени експозиції 3D-моделей об'єктів за допомогою публікації на розробленому сайті.

Об'єкт дослідження – процес розробки 3D-моделей на різних графічних платформах.

Предмет дослідження – теоретичні, методологічні, програмні та прикладні аспекти створення 3D-моделей об'єктів і варіанти їх інтеграції до структури сайту.

Для досягнення мети і розв'язання поставлених завдань використали сукупність методів дослідження, таких як: теоретичні – щоб проаналізувати інформаційні джерела; емпіричні – для вивчення та узагальнення досвіду використання 3D-моделювання у розробці сайтів.

Структура та обсяг роботи: містить вступ, два розділи, висновки, а також список використаних джерел – 41 позиція. Основний зміст роботи викладено на 44 сторінках, проілюстровано 49 рисунками, 4 таблицями. Повний обсяг роботи – 45 сторінок.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження актуальності тривимірного моделювання

Тривимірною графікою, яка ще часто називається 3D-моделюванням, можна вважати комп'ютерну графіку, що включає методи та інструменти щоб створити об'ємні об'єкти у тривимірному просторі. 3D-зображення на відміну від двовимірного передбачає створення геометричної проєкції 3D-моделі сцени на площину використовуючи спеціальні програми. Отримана модель може бути реальною, наприклад модель певного існуючого об'єкта, або абстрактною [1].

3D-графіка використовується для імітації фотографування або відеозйомки тривимірних образів об'єктів, що зазвичай створюються в пам'яті комп'ютера у наступній послідовності: підготовка, побудова геометричної моделі сцени, настроювання освітлення і знімальних камер, підготовка і налаштування матеріалів, візуалізація сцени. Отже, таким чином створюється віртуальний 3D-світ [2].

Тривимірне моделювання дає можливість отримати модель, максимально схожу на реальну. За допомогою сучасних програм досягається високий рівень деталізації. Наочність проєкту значно збільшується. Складною задачею є проєктування тривимірного об'єкта на двомірній площині, а за допомогою 3D-візуалізації маємо можливість розглянути усе детально [3].

3D-моделювання використовується у наступних галузях людської діяльності:

Реклама та маркетинг. Тривимірна графіка необхідна для демонстрації майбутнього виробу. Спочатку зображаємо, а потім створюємо 3D-модель проєктуемого об'єкту, а на основі 3D-моделі досліджуємо зразок майбутнього виробу. Після отримання 3D-моделі, отримане зображення використовується для створення зовнішньої реклами, у розробці дизайну упаковки, дизайну виставкових стендів тощо.

Міське планування. Завдяки використанню 3D-графіки спостерігається більш реальне моделювання міської архітектури та ландшафтів [4]. За допо-

могою візуалізації ландшафтного оформлення та архітектури будівель інвестори та архітектори мають можливість відчутти ефект присутності у спроектованому просторі.

Промисловість. За допомогою 3D-технологій виробники мають можливість економити матеріали і зменшувати фінансові затрати на інженерне проектування. Дизайнери-графіки можуть створювати тривимірні зображення деталей і об'єктів, які потім використовуються для створення прототипів об'єктів та прес-форм [5].

Комп'ютерні ігри. При створенні комп'ютерних ігор 3D-модельовання використовується більше двадцяти років. У спеціальних програмах фахівці створюють моделі героїв та тривимірні ландшафти, анімації 3D-об'єктів і персонажів, а також створюють концепт-арти (концепт-дизайни) [6].

Кінематограф. У сучасній кіноіндустрії для зйомок у форматі 3D використовуються спеціальні камери. Також, завдяки 3D-графіці, для кінематографу можна створити повноцінні ландшафти та окремі об'єкти [7].

Архітектура та дизайн інтер'єрів. У наш час створення 3D-моделі споруди є незамінним при проектуванні. На основі 3D-моделі можна створювати прототип будівлі. Прототип може бути або загальним обрисом будівлі, або деталізованою моделлю майбутньої споруди. За допомогою 3D-технологій, можна побачити, як буде виглядати житлове або офісне приміщення, наприклад, після ремонту.

Анімація. Використовуючи 3D-графіку створюють анімованих персонажів, які можуть рухатися, або взагалі повноцінний анімований відеоролик.

За останній час програми для створення й обробки 3D-графіки значно покращені. Це пов'язано із вдосконаленням комп'ютерної техніки, її графічної системи. Тривимірне моделювання привертає увагу як інженерів, програмістів, проектувальників, розробників комп'ютерних ігор, так і науковців (у медицині, астрономії тощо) [8].

При геологічних дослідженнях отримують геометрію залягаючих на великій глибині пластів. Методи діагностики комп'ютерної візуалізації внут-

рішніх органів людини використовуються у сучасній медицині. Двовимірна графіка використовується при енцефалограмах, міограмах, а тривимірна – при томографії. Результати дослідження виводяться на екран комп'ютера [9].

Про постійний ріст затребуваності тривимірної графіки та 3D-моделювання також свідчить динаміка популярності пошукових запитів з 3D-моделювання за останні 5 років в Україні та світі (рис. 1.1–1.2). Україна займає 5 позицію за популярністю даних запитів. Бачимо, що запитів з 3D-моделювання набагато більше ніж з тривимірної графіки, отже 3D-моделювання більш популярне.

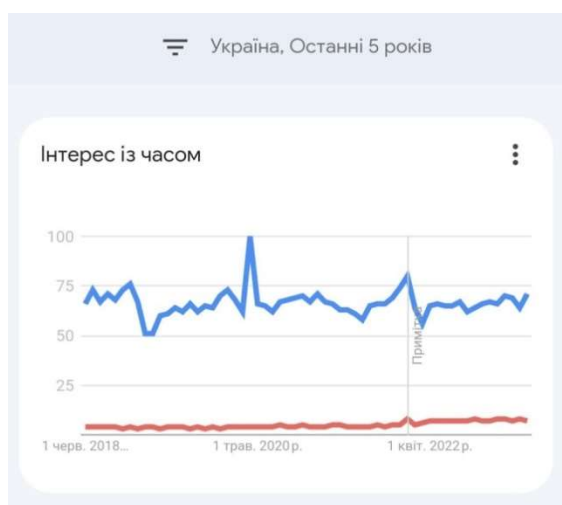


Рис.1.1 Порівняння популярності запитів в Україні: «3D-моделювання» (синя крива), «тривимірна графіка» (червона крива)

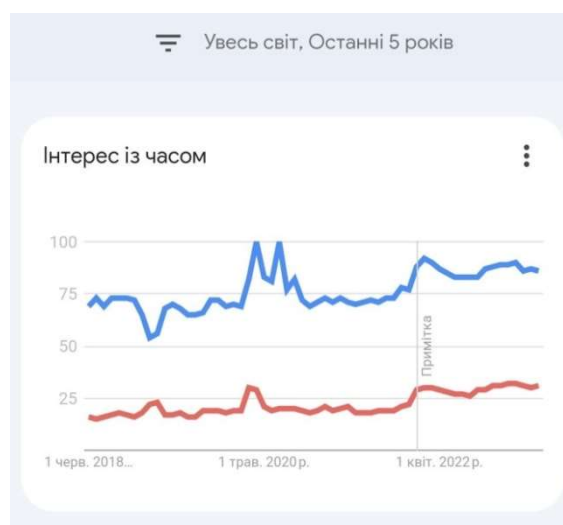


Рис.1.2 Порівняння популярності запитів в світі: «3D-моделювання» (синя крива), «тривимірна графіка» (червона крива)

1.2 Математичні основи 3D-графіки

У комп'ютерній графіці 3D-моделюванням вважається розробка математичного представлення тривимірної поверхні об'єкта через спеціальні програми.

У просторі можна ввести систему координат із трьома координатними осями (не лежать в одній площині, зазвичай перпендикулярних один одному) і положення кожної точки буде описуватися трійкою чисел (x, y, z) (рис. 1.3).

Будь-яку точку в такому просторі можна задати тривимірним вектором – впорядкованою трійкою чисел, записаною у вигляді рядка або стовпця [10].

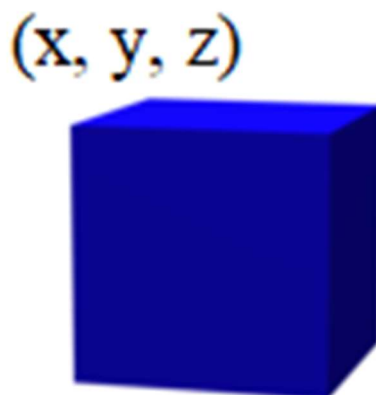


Рис.1.3 Координати в тривимірній графіці

При використанні програм тривимірної графіки необхідною умовою є знання математичних основ. Неможливо побудувати об'ємну модель, не знаючи її параметрів та не маючи уявлення про неї.

Тривимірна система координат може бути лівосторонньою та правосторонньою (рис. 1.4–1.5).

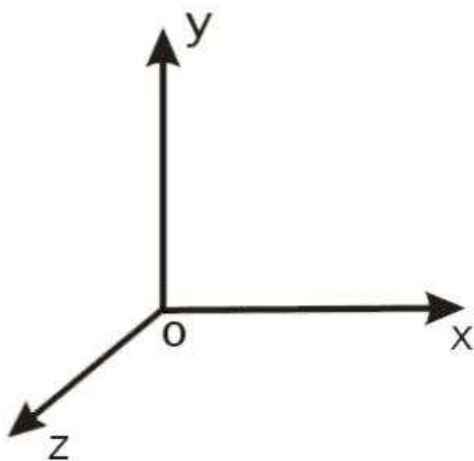


Рис.1.4 Правостороння система координат

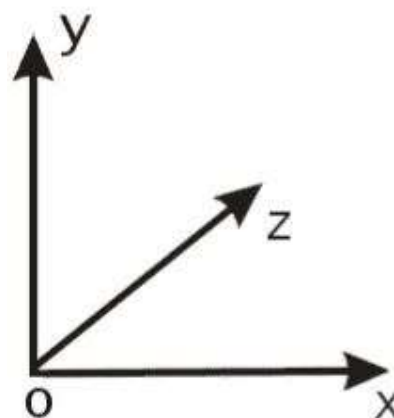


Рис.1.5 Лівостороння система координат

При використанні різних графічних редакторів використовуються й різні системи координат.

До математичних основ тривимірної графіки можна віднести дуже значний обсяг тем, наприклад: рівняння прямих, ліній та площин», тривимірні перетворення та проєкції, тіла обертання, обчислення об'ємів, площ поверхонь та перерізи тіл[11].

Види координат: локальні (координати в системі координат моделі; використовуються при моделюванні та для створення декількох копій об'єкта), світові (визначають реальне положення об'єкта в 3D-світі; використовуються при виявленні зіткнень, обробці фізики, освітленні тощо), координати камери, координати проєкції, екранні координати. (рис. 1.6–1.7).

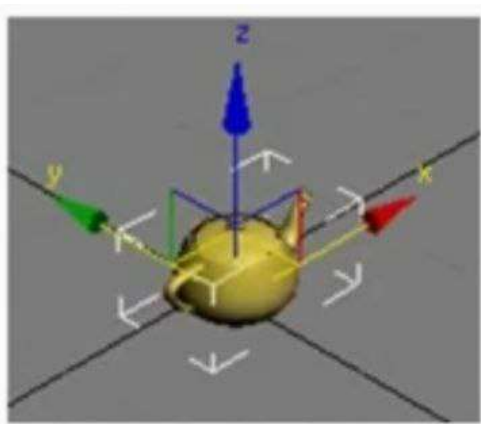


Рис.1.6 Локальні координати

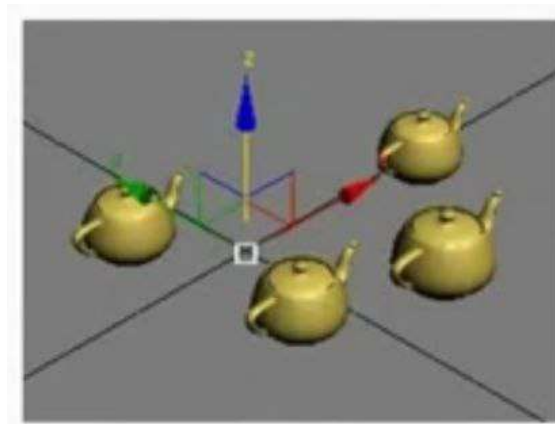


Рис.1.7 Світові координати

Коли побудували тривимірну сцену, потрібно це відобразити на моніторі. Для цього необхідно встановити камеру (рис. 1.8). Координати камери використовуються для визначення об'єктів сцени. Коли розмістили камеру, у неї є площа огляду – піраміда (Див. Рис. 1.8).

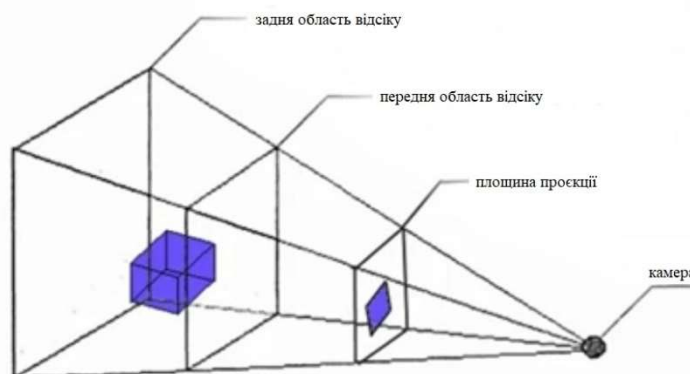


Рис. 1.8 Тривимірна сцена: камера та координати проєкції

Далі потрібно перетворити координати камери у координати проекції. Додають задню і передню області відсіку. Все, що знаходиться за їх межами буде невидимим. А те, що потрапило до відсіченої піраміди буде проектуватись на площину проекції. Площина проекції обмежена квадратом 1x1. Оскільки роздільна здатність монітора відмінна від 1x1, то потрібно переходити до екранних координат масштабуванням зображення 1x1.

Отже, 3D-модель об'єкта призначена для того, щоб на моніторі комп'ютера можна було різносторонньо розглядати деталі об'єктів.

Особливості 3D-зображень:

- просторова перспектива та ефекти об'єктів (тіні, відблиски тощо);
- текстура поверхонь (шорсткість, нерівність, пухнастість тощо).

Принцип тривимірного моделювання: побудова 3D-моделі відбувається допомогою тривимірних примітивів – об'ємних і плоских геометричних тіл, які називаються мешами (сітка). Кожен тривимірний примітив описується математично.

Основні тривимірні елементи: до плоских відносяться квадрат і коло, а до об'ємних – куб, сфера, конус, циліндр.

Поверхня 3D-моделі об'єкта ділиться на грані, які мають форму трикутника або чотирикутника). Грань обмежена ребрами. Кожне ребро має дві вершини.

Перетворення, що можуть виконуватись над елементами 3D-моделі:

- переміщення цілого об'єкта або часткове – вершин, ребер, граней;
- повертання;
- масштабування.

Властивості 3D-моделі включають:

- кількість вершин, ребер, граней;
- розміщення компонентів на поверхні моделі;
- текстура матеріалів [12].

1.3 Основні етапи створення 3D-моделі

До основних графічних функцій можна віднести [13]: функції побудови (проведення відрізків прямих, кіл та їх дуг, кривих за заданими умовами, багатокутників і т. ін.); функції редагування (видалення, вставки); функції перетворення (зсув, поворот, зміна масштабу); функції обчислення (довжин, периметрів, об'ємів, площ); функції виведення (виведення інформації до друку або креслення) [14].

Етапи розробки 3D-моделі:

1. Моделювання об'єкту [15]: вичавлювання, обертання, модифікація, полігональне моделювання.
2. Текстурування об'єкту (вибір матеріалів для об'єкту) (рис. 1.9) [16].

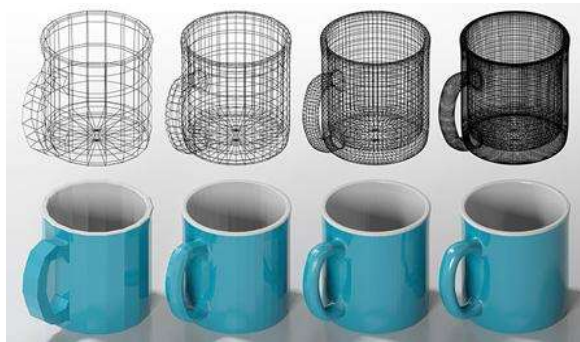


Рис.1.9 Текстурування об'єкту

3. Світлові ефекти (вибір тону світла, рівня яскравості, різкості і глибини тіней) [17]. Також треба вибрати місце спостереження за об'єктом (вид з висоти пташиного польоту, масштабування простору з досягненням ефекту присутності в ньому тощо).
4. Тривимірна візуалізація або рендеринг [18] – це останній етап 3D-моделювання. На цьому етапі відбувається деталізація налаштувань відображення 3D-моделі: додавання графічних спецефектів (відблиски, туман, сяйво. Далі визначаються параметри 3D-анімації персонажів, деталей, ландшафтів тощо та підбирається потрібна кількість кадрів в секунду і формат відео. Для отримання двовимірного растрового зображення, визначається формат і розширення зображення.

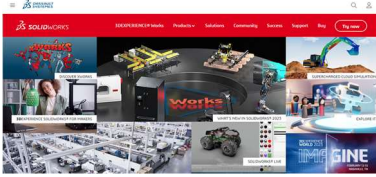

5. Постпродакшн [19]. На цьому етапі відбувається обробка та візуалізація відзнятих зображень і відео за допомогою медіа-редакторів – Adobe Photoshop, Adobe Premier Pro, GarageBand, Adobe Illustrator, Samplitude, Wavelab та ін.

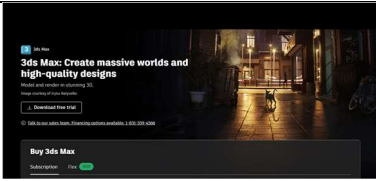


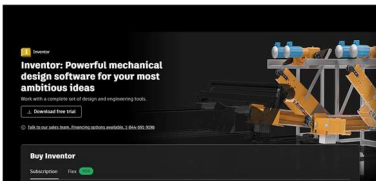
1.4 Порівняльна характеристика програмних засобів 3D-моделювання

Програмні пакети для 3D-моделювання дуже різноманітні. Обрати оптимальне програмне забезпечення для моделювання досить складно, так як важко знайти програму з необхідним функціоналом. У табл. 1.1 представлені найпопулярніші сервіси для 3D-моделювання.

Таблиця 1.1

Перелік та опис програм для 3D-моделювання

№ п/п	Назва	Опис
1	SolidWorks (SolidWorks Corporation) [20] 	Використовується для дизайну, деталізації та візуалізації систем, машин та оснащення і включає моделювання, збірку, рисування sheetmetal, зварюваних деталей та freeform surfacing functionality. Підтримує Visual Basic та C.
2	ProEngineering [21] 	Включає проектування, інженерний аналіз та підготовку виготовлення виробів різної складності і призначення. ProENGINEER є основою автоматизації підприємства, за допомогою якого здійснюється підтримка довговічності виробу відповідно до концепції CALS-технологій (Continuous Acquisition and Life cycle Support). ProENGINEER Enterprise SE (Standard Edition) забезпечує виконання задач розробки виробу і відповідає вимогам виробничо-конструкторських груп.
3	3D-MAX [22]	Включає засоби для створення тривимірних комп'ютерних моделей, реальних чи фантастичних об'єктів навколишнього світу, використовуючи різні

№ п/п	Назва	Опис
		<p>техніки, зокрема полігональне моделювання, в яке входять Editable mesh (редагована поверхня) і Editable poly (редагований полігон). Цей метод використовується для створення складних моделей і низькополігональних моделей для ігор.</p>
4	<p>SketchUp Pro [23]</p> 	<p>Програма, яка підтримує створення 2D- та 3D-моделей будівель, меблів, інтер'єру. Всі геометричні характеристики зазначаються в полі Value Control Box (поле контролю параметрів), яке знаходиться в правому нижньому кутку робочої області, справа від напису Measurements (панель вимірів). За допомогою інструменту Push/Pull площину можна «втягнути» в сторону, створивши нові бокові стінки. Рухати площину можна до наперед заданої кривої, використовуючи інструмент Follow Me («Ведення»).</p>
5	<p>AutoCAD [24]</p> 	<p>Програма для комплексного тривимірного моделювання з повним набором інструментів (підтримується твердотільне, поверхневе і полігональне моделювання). За допомогою AutoCAD ми можемо отримати якісну візуалізацію моделей використовуючи рендеринг mental ray. Результат моделювання можна відправити на 3D-принтер. Через відсутність тривимірної параметризації AutoCAD не може конкурувати з машинобудівними САПР, такими як Inventor та SolidWorks.</p>
6	<p>Inventor (Autodesk) 3D [25]</p> 	<p>САПР для створення і вивчення аналізу поведінки цифрових прототипів виробів і деталей та для 3D-дизайну механіки, емуляції продукту, створення інструментарію.</p>
7	<p>Компас-3D [26]</p>	<p>Графічний редактор для створення твердотілих</p>

№ п/п	Назва	Опис
		об'єктів з використанням набору простих параметричних тіл (паралелепіпед, циліндр та ін.). Він доступний в освоєнні та має підказки.
8	Blender [27] 	Використовується для створення 3D-графіки, включаючи засоби моделювання, анімації, рендерінгу, післяобробки відео тощо.

На основі стандарту ISO 9126:2001 можемо обрати параметри якості оцінювання, де кожна характеристика описується за допомогою кількох вхідних у неї індексів [28]. Для кожного індексу визначається набір параметрів, що дозволяють його оцінити: «функціональність», «зручність використання», «продуктивність» або «часова ефективність», та використовується: «інтерфейс програми», «кількість бібліотек 3D-моделей і матеріалів» з найважливішим параметром – «якість рендеру» – підсумкового зображення після обробки [29] (табл. 1.2).

Таблиця 1.2

Параметри оцінювання

Параметр	Індекс
Якість рендеру	4.0
Кількість бібліотек	2.0
Зручність у використанні	2.0
Часова ефективність	2.5
Функціональність	3.0
Інтерфейс	1.5

Кожний критерій не є рівнозначним, тому треба було зазначити коефіцієнти важливості кожному з них (табл. 1.3).

Таблиця 1.3

Таблиця оцінювання програмних засобів 3D-моделювання

Назва ПЗ	Якість фінальної картинки	Кількість бібліотек	Зручність у використанні	Часова ефективність	Функціональність	Інтерфейс
SolidWorks	5	3	9	9	3	3
ProEngineering	4	3	6	7	2	5
3D Max	10	10	8	8	10	9
SketchUp Pro	7	5	10	10	6	9
AutoCAD	5	6	5	6	7	6
Inventor	7	6	7	7	7	7
Компас-3D	4	1	3	5	3	3
Blender	9	8	7	8	10	9

Підсумкова оцінка еквівалентна сумі балів за кожний параметр, помножений на відповідний індекс оцінювання. Після розрахунків було сформовано рейтингову таблицю (табл. 1.4).

Таблиця 1.4

Таблиця рейтингів програмного забезпечення для 3D-візуалізації

Місце	Назва програми	Оцінка
1	3D Max	139,5
2	blender	129,5
3	sketchUp Pro	114,5
4	inventor	103
5	autoCAD	87
6	solidWorks	80
7	proEngineering	65
8	компас-3D	50

1.5 Висновки до розділу 1

Було розглянуто основні поняття тривимірної графіки (3D-модельювання), теоретичні, методологічні, програмні та прикладні аспекти створення 3D-моделей. Проведено аналіз предметної області та визначено актуальність даної теми. Розглянуто області застосування 3D-модельювання та етапи створення 3D-моделі. Здійснено порівняльний аналіз програмних засобів для створення 3D-моделей, виявлено їх переваги та недоліки.

РОЗДІЛ 2. ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЄКТУ

2.1 Вибір програмних засобів реалізації 3D-моделей

Згідно результатів порівняльної характеристики, представленої у вигляді рейтингової таблиці (див. Табл. 1.4) лідерами серед програмних засобів для 3D моделювання є 3D Max та Blender, але останній має відкриту ліцензію тому наш вибір зупинився саме на цьому програмному інструментарії. Додатково, у якості альтернативних програмних засобів для створення 3D-моделей були обрані Sweet Home 3D та Tinkercad.

Коротко зупинимось на ключових аспектах роботи з обраними редакторами.

Blender – редактор тривимірної графіки, призначений для об’ємного моделювання, візуалізації (рендера), створення статичних і динамічних сцен, анімації, а також створення [30].

Blender містить примітиви, які складаються з безлічі з’єднаних у просторі точок і утворюють фігуру [31]. До першого примітиву відноситься площина. Другий примітив – це сфера, яка складається з більшої кількості полігонів, ніж ICO Sphere. Далі – звичайний куб та циліндр. Примітив Monkey – мавпочка. Додатковими об’єктами у сцені є камера, джерела світла та інші допоміжні інструменти.

3D-просторі має три вісі: X (червона), Y (зелена) та Z (синя).

Маніпулятор – засіб для зміни положення, кута повороту та розмірів об’єкта. Існує три типи маніпуляторів: маніпулятор для зміни положення, маніпулятор для зміни кута повороту, маніпулятор для зміни розмірів об’єкта. Але важливо розуміти, що таке є глобальні та локальні вісі для кожного типу маніпулятора.

Режим редагування об’єктів використовується для редагування елементів лише одного об’єкта [32]. Елементами об’єкта є: «вершини» – точки в просторі з певними координатами, «ребра» – лінії, що з’єднують «вершини», «грані» – площини, натягнуті на ребра.

Текстури – це зображення, що характеризує одну з властивостей поверхні матеріалу. Текстурою можна показати рельєфність, малюнок, блиск тощо. Для редагування текстури необхідна наявність матеріалу на виділеному об'єкті. Без матеріалу текстури немає. Для редагування текстур служить вкладка «Texture» у вікні властивостей об'єкту.

До переваг цієї програми відноситься:

- вільне ПЗ;
- малий розмір;
- зручний україномовний інтерфейс;
- велика кількість вільних додатків, які розширюють можливості програми;
- можливість створювати самостійні ігри;
- невибагливість до апаратної частини комп'ютера;
- експорт та імпорт 3D-форматів;
- мобільність.

До недоліків цієї програми відноситься:

- недостатність навчальної літератури;
- незвичний вигляд.

Sweet Home 3D – вільне програмне забезпечення, яке має відкритий код і призначене для візуалізації житлових споруд і планів будинків та моделювання інтер'єру. Sweet Home 3D доступний російською, англійською, французькою, португальською, італійською, німецькою, іспанською, шведською, чеською, польською та угорською мовами і може бути запущений на Windows, MacOS, Linux і Solaris [33].

Програма містить каталог зразків меблів, розташований за категоріями. Цей каталог постійно оновлюються, а моделі інтер'єру можна безкоштовно завантажити з офіційного сайту.

У цьому програмному додатку можна зберегти проєкт кімнати або квартири в форматі PDF і роздрукувати на папір, щоб відразу надати замовнику.

На сайті збережена велика кількість текстур і моделей меблів для цієї програми, які можна завантажувати і потім використовувати при розробках проєктів. Також є можливість сфотографувати кімнату або квартиру чи записати на відео [34].

Можливості:

- двовимірні та тривимірні плани;
- імпорт тривимірних моделей в різних форматах, таких як: OBJ (Wavefront), 3DS (3D Studio MAX), LWS (LightWave Scene) DAE (COLLADA);
- імпорт фонових зображень у форматах, таких як: BMP, GIF, JPEG, PNG;
- експорт плану у форматах PDF і SVG, експорт тривимірного вигляду у форматі OBJ;
- створення демонстраційних графічних та відео файлів проєкту інтер'єру;
- використовувати на плануванні можна лише об'єкти, передбачені програмою. Каталоги об'єктів доступні для завантаження на офіційному сайті.

Крім запуску програми з будь-якого носія, також можливе використання програми у веб-браузері за умови встановленого в операційній системі Java Runtime Environment та її підтримки цим браузером.

До основних переваг програми можемо віднести наступне:

- можливість працювати на комп'ютерах з операційними системами, такими як: Windows, Linux, Mac OS і Solaris;
- програма є безкоштовною та русифікованою;
- легкість і простота у використанні, наявність підказок;
- малий розмір.

До недоліків Sweet Home 3D можемо віднести:

- відсутність можливості складання фінансових витрат на реалізацію проєкта;
- невелика кількість фурнітури у каталозі фурнітури;
- відсутня можливість будувати розрізи і моделювати ландшафти.

Autodesk Tinkercad – це безкоштовна онлайн-програма, створена для 3D-моделювання. Відтоді, як вона стала доступною у 2011 році, вона є однією з найпопулярніших платформ для створення моделей для 3D-друку та для початкового рівня роботи з твердотільною геометрією в школах.

Конструкція складається з примітивних форм, які є або «суцільними», або «дірковими». При комбінації твердих тіл та отворів, за допомогою даної програми можна створювати нові форми із призначенням властивостей. Користувачі мають можливість створювати власні генератори форм за допомогою вбудованого редактора JavaScript.

Фігури можуть імпортуватися у трьох форматах (STL, OBJ) для 3D, а також двовимірні форми SVG для екструдювання у 3D-фігури. Tinkercad може експортувати моделі у формати STL або OBJ, які готові до 3D-друку. Tinkercad також має можливість експортувати 3D-моделі до Minecraft Java Edition і проєктувати конструкції використовуючи віртуальні кубики Lego.

Після запуску програми ви побачите спеціальне вікно, де можна створювати тривимірну модель. У цьому вікні можна додавати нові об'єкти, з'єднувати їх, масштабувати, розтягувати, стискати та робити ще багато інших процесів.

Вгорі є багатофункціональні кнопки. Вони дозволяють створювати групи об'єктів, вирівнювати їх, відображати дзеркально і робити багато інших речей.

Ключові особливості: багато інструментів для створення моделей; можна відразу відтворити модель за допомогою принтера; зберігає процес у файлах OBJ та STL; можна об'єднувати та роз'єднувати файли у групах; є

веб-версія, а також спеціальний UMP-додаток; Є можливість створювати двомірний SVG креслення [35].

Основні переваги програми TinkerCad:

- доступність програми;
- не потребує встановлення (не використовує робочого дискового простору);
- простий інтерфейс;
- простий, інтуїтивно зрозумілий набір інструментів;
- велика співдружність користувачів;
- простота реєстрації;
- наявність вбудованого засобу для навчання (покрокові інструкції із завданнями);
- можливість зберігати моделі у хмарі;
- імпорт готових моделей;
- велика кількість готових проєктів, які можна завантажити та редагувати.

Але програма має і недоліки:

- обмеженість варіантів перетворення фігур через простота роботи з ними;
- невелика кількість налаштувань камери;
- залежить від інтернет-з'єднання;
- велика ресурсомісткість (менш потужним комп'ютерам важко пропрацювати контури деталей).

2.2 Розробка 3D-моделей

Розглянемо результати створення 3D-моделі сучасних моніторів в обраних у попередньому пункті редакторах 3D-графіки: Blender, Tinkercad та Sweet Home 3D.

Почнемо з Blender. Об'ємне зображення можна розглядати з усіх сторін за допомогою функції обертання. В налаштуваннях об'єкту можна вибрати

масштаб майбутньої 3D-моделі, її положення та обертання. Будь-які зміни з моделлю виконуються в режимі редагування. Переглядати отриману модель можна в об'єктному режимі (рис. 2.1). Далі необхідно задати розміри майбутнього монітора (рис. 2.2).

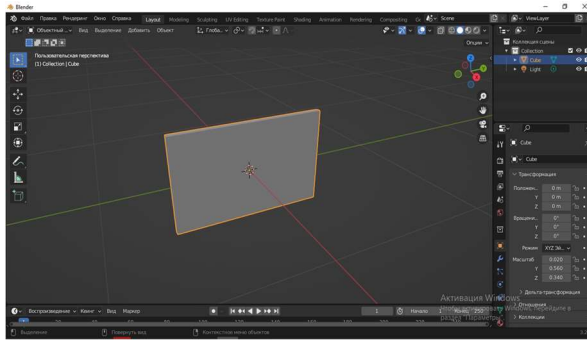


Рис. 2.1 Створення моделі куба

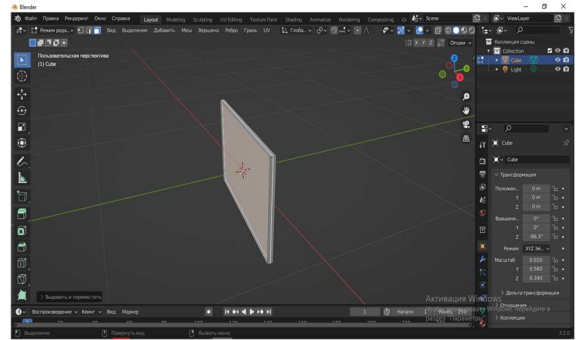


Рис. 2.2 Задавання розмірів

Далі переходимо в режим редагування і працюємо з полігонами (рис. 2.3).

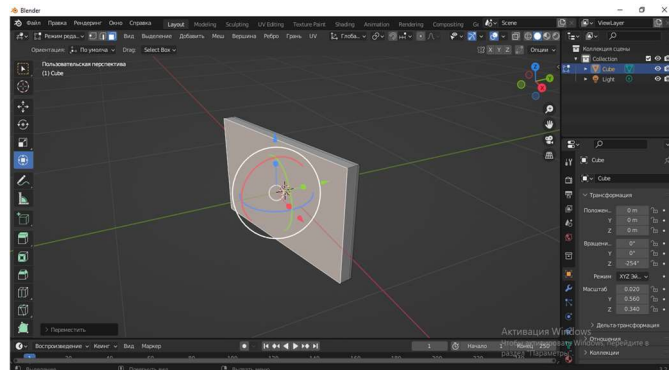


Рис.2.3 Робота з полігонами

Повертаємо наш монітор в інший бік та масштабуємо (рис. 2.4–2.5).

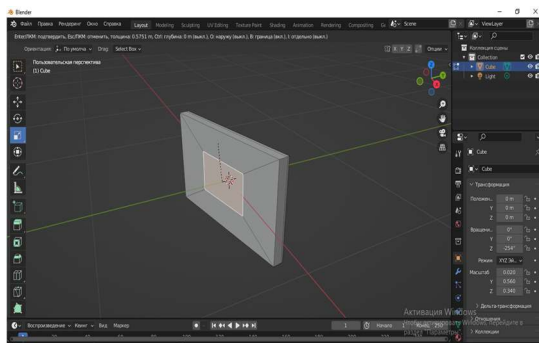


Рис.2.4 Масштабування

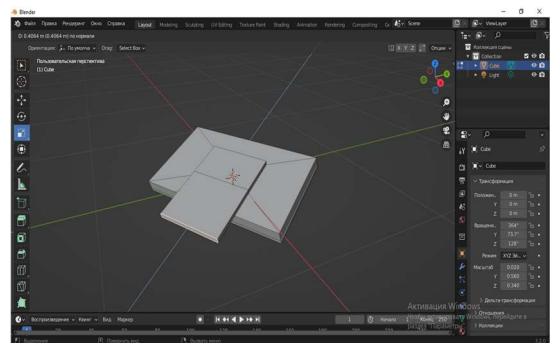


Рис.2.5 Вигляд ззаду

Моделюємо підставку (рис. 2.6–2.7).

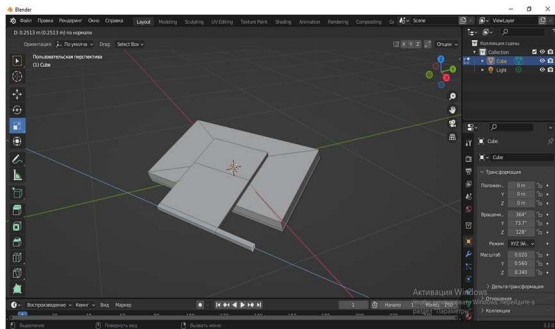


Рис.2.6 Моделювання підставки

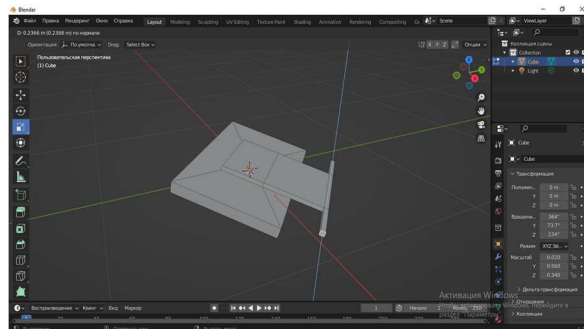


Рис.2.7 Моделювання підставки

Таким чином вийшла 3D-модель сучасного монітора (рис. 2.8).

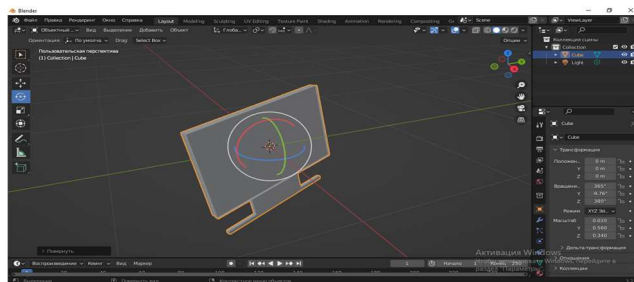


Рис.2.8 ПК-монітор

Накладаємо текстуру та змінюємо колір (рис. 2.9).

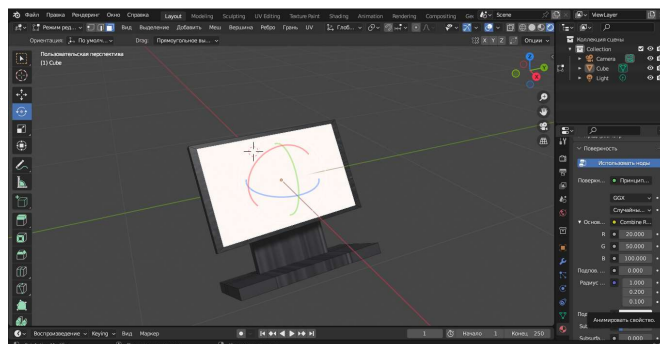


Рис.2.9 Накладання текстури

Використаємо Tinkercad для створення ЖК-монітора. В «основних формах» оберемо паралелепіпед (рис. 2.10).

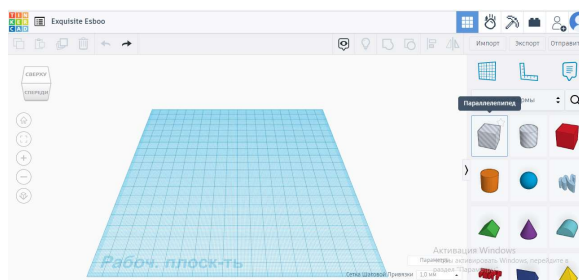


Рис. 2.10 Робоча поверхня

Розміщуємо його на робочій поверхні (рис. 2.11).

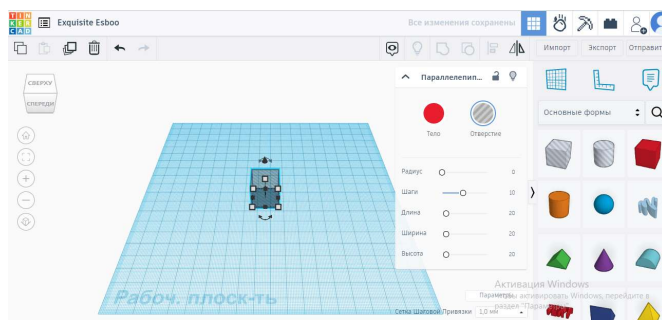


Рис.2.11 Розміщення паралелепіпеда

Далі задаємо розміри (рис. 2.12).

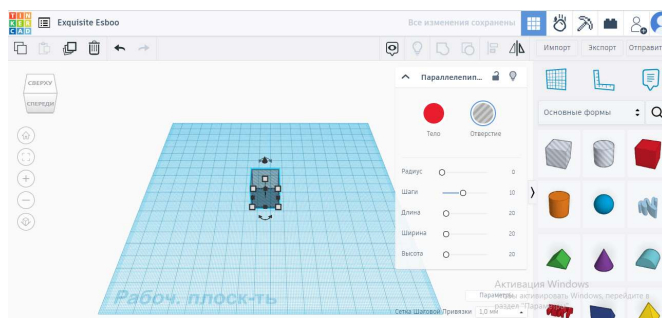


Рис. 2.12 Задання розмірів

Моделюємо підставку (рис. 2.13–2.14).

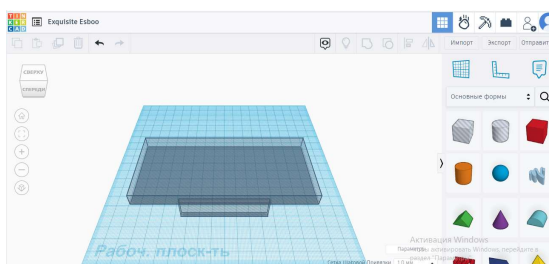


Рис. 2.13 Моделювання підставки

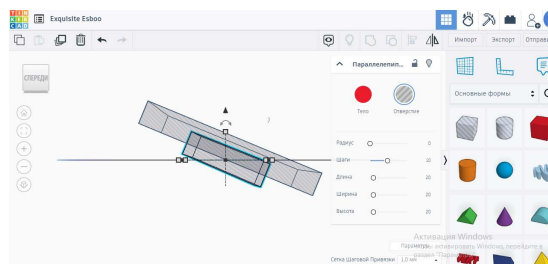


Рис. 2.14 Моделювання підставки

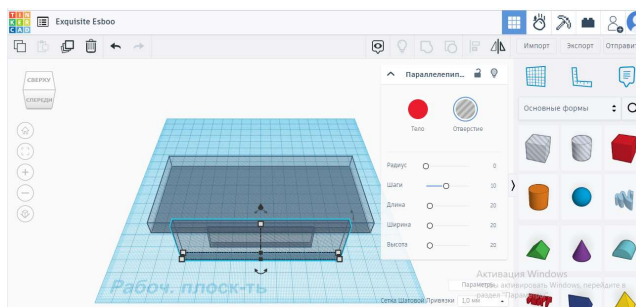


Рис. 2.15 Моделювання підставки

Отримаємо 3D-модель монітора (рис. 2.16).

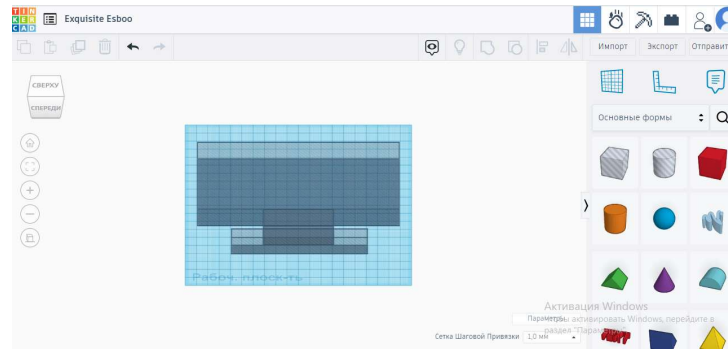


Рис. 2.16 ЖК-монітор

З програмою Sweet Home 3D набагато простіше, оскільки до неї можна імпортувати завантажені 3D-моделі (рис. 2.17–2.18).

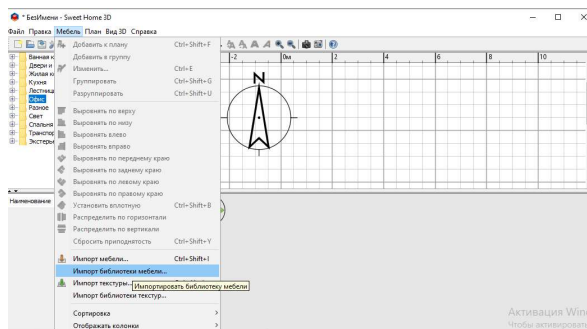


Рис. 2.17 Імпорт бібліотеки

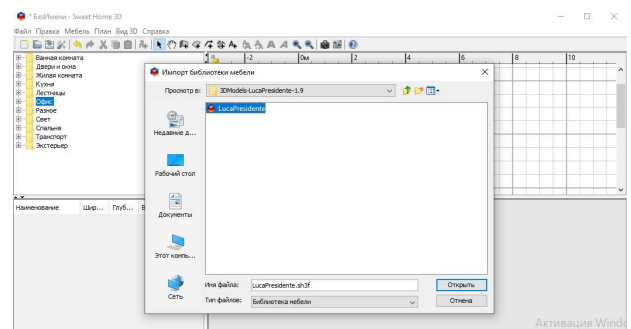


Рис. 2.18 Вибір файлів для імпортування

У розділі «Офіс» обираємо ЖК-монітор та додаємо до плану (рис. 2.19–2.20).

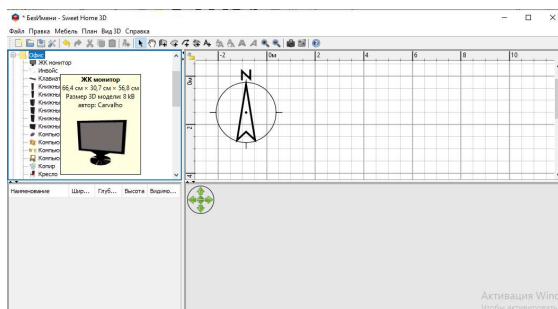


Рис. 2.19 Обирання об'єкта

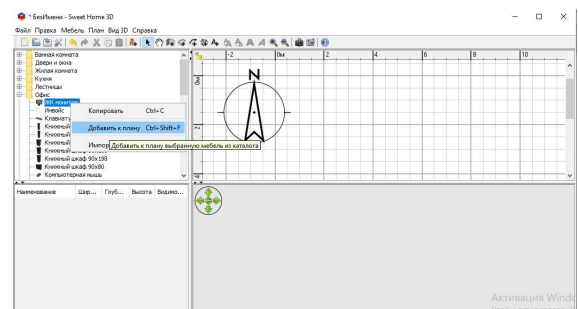


Рис. 2.20 Додавання об'єкта

У результаті маємо готову 3D-модель (рис. 2.21). За допомогою цієї програми, нажаль, не зможемо самостійно створювати різні предмети. Ця програма розрахована здебільшого для моделювання інтер'єру. У ній не можна обертати моделі, можна зробити тільки фото або відео тримірного вигляду.

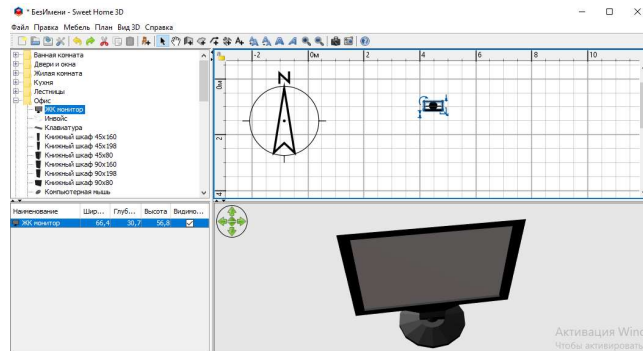


Рис. 2.21 ЖК-монітор

Серед програмних засобів, за допомогою яких моделювали монітори, на наш погляд краще обрати Blender – він має простий інтерфейс, має великий функціонал: і 3D-моделювання, і анімація, і монтаж відео зі звуком. Крім того, Blender – популярний, постійно розвивається та має технічну підтримку.

2.3 Вибір засобів для створення web-сайту

В якості основного фреймворка нами був обраний Django фреймворк [36], написаний на мові Python [37]. Однією із причин вибору Django є те, що він має вбудовану адмін-панель, яка дозволяє кінцевому користувачу досить легко маніпулювати контентом та сутностями нашого додатку, а також самостійно створювати, редагувати та видаляти 3D-моделі. Також у даного фреймворка є вже вбудована ORM система що дозволяє нам маніпулювати даними у базі даних, шаблонна система – за допомогою якої ми побудували веб-сайт, а також велике ком'юніті, що також може відіграти значну роль при пошуку інформації в інтернет.

Для відображення 3D-моделей на сайті можна використовувати різні додатки та бібліотеки, наприклад, такі, як Blend4Web [38], Three.js [39] та Sketchfab [40]. Проаналізуємо їх більш детально.

Blend4web – це веб-орієнтований 3D-движок – програмна основа для створення та інтерактивного рендерингу тривимірної графіки та аудіо в браузерах. Платформа призначена для візуалізацій, презентацій, онлайн-магазинів, ігор та інших інтернет-додатків. Фреймворк Blend4Web тісно інтегрований з Blender (звідси і назва).

Контент відтворюється за допомогою WebGL та інших технологій браузера без використання плагінів. Технічно Blend4Web – це бібліотека для веб-сторінок, надбудова Blender і деякі інструменти для налагодження та оптимізації.

З недоліків цієї технології можна назвати досить невелику кількість документації, обмежену кількість форматів 3D-моделей з якими вона може працювати.

Sketchfab – це веб-платформа 3D-моделювання для публікації, обміну та торгівлі 3D, VR і AR-контентом. Він надає засіб перегляду на основі технологій WebGL і WebXR, який дозволяє користувачам відображати 3D-моделі в Інтернеті для перегляду в будь-якому мобільному браузері, настільному браузері або гарнітурі віртуальної реальності.

Основним продуктом Sketchfab є програма для перегляду моделей 3D, віртуальної реальності (VR) і доповненої реальності (AR). Це дозволяє користувачам вільно переміщатися навколо або всередині 3D-сцени за допомогою миші, сенсорних маніпуляцій, VR або AR. Крім статичних 3D-моделей, програма перегляду може відтворювати 3D-анімацію та керувати нею. Глядачі можуть увімкнути режим VR, щоб побачити модель у гарнітурах віртуальної реальності.

Програма 3D-перегляду використовується на веб-сайті Sketchfab і у мобільних додатках але також може бути вбудована на зовнішніх веб-сайтах.

Перевагами sketchfab можна назвати підтримку багатьох форматів, а саме: obj, blend, fbx, gltf, легку інтеграцію на веб-сторінку. З недоліків можна відмітити те, що користувачу треба завантажити власні 3D-моделі до sketchfab, що не завжди підходить, а також обмежені можливості по взаємодії та налаштуванню сцени.

Three.js – це рушій WebGL написаний JavaScript, за допомогою якого можна запускати ігри та інші графічні програми через браузер. Three.js має багато функцій і API для візуалізації 3D-сцен у браузері.

Three.js надає можливість створювати 3D-анімацію із прискоренням

графічного процесора (GPU), використовуючи мову JavaScript як частину веб-сайту, не розраховуючи на фірмові плагіни браузера. Така можливість з'явилась завдяки WebGL, низькорівневого графічного API, які були створені для Інтернету.

За допомогою Three.js можна легко створювати складні 3D-анімації для відображення у браузері.

З переваг Three.js можна відмітити те, що це open-source бібліотека і з її кодом можна ознайомитись на github, досить хороша документація з прикладами інтеграцій, велике «ком'юніті», що полегшує пошук інформації, велика свобода в реалізації 3D-сцени, досить гарна розширюваність завдяки плагінам, які додають багато екстра-функціональності.

Отже, аналізуючи документацію та спираючись на описану порівняльну характеристику зазначених веб-сервісів відображення 3D-моделей, для досягнення мети нашої роботи ми обрали Django та Three.js.

2.4 Розробка структури та дизайну web-сайту з інтеграцією 3D-моделей

Розробку сайту починаємо з того, що створюємо віртуальне середовище python, щоб встановити необхідні пакети Python (рис. 2.22).

```
dm > view3d > requirements.txt
1  asgiref==3.6.0
2  Django==4.2
3  sqlparse==0.4.4
4
```

Рис. 2.22 Опис необхідних пакетів Python

Після встановлення пакетів отримуємо проєкт з наступною структурою файлів та директорій (рис. 2.23):

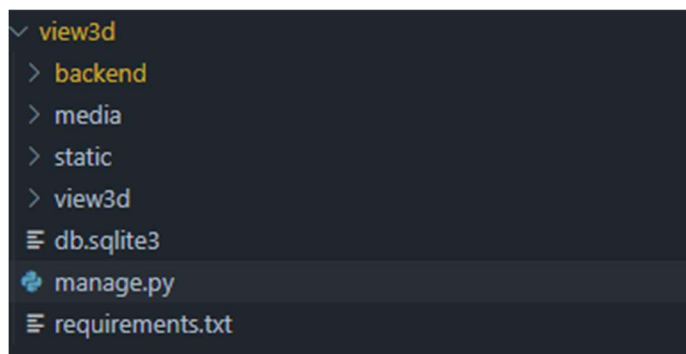


Рис. 2.23 Структура файлів та директорій проєкту

Наступним кроком необхідно створити моделі бази даних для додатку, які б дали змогу користувачеві завантажувати 3D-моделі.

Для цього спочатку треба описати базові моделі та їх поля. Почнемо з моделі яку ми назвали Model3D, яка і буде відповідальна за зберігання даних 3D-моделей (рис. 2.24).

```

> view3d > backend > models.py > Model3D > __str__
1  from django.db import models
2
3
4  class Model3D(models.Model):
5      title = models.CharField(max_length=100)
6      model_file = models.FileField()
7
8  def __str__(self):
9      return f"{self.pk}-{self.title}"
9

```

Рис. 2.24 Опис моделі Model3D, яка відповідає за збереження даних 3D-моделей

Після чого можемо згенерувати міграцію, яка дозволить швидко застосовувати або відмінити певні зміни які вносились до бази даних, а також прослідкувати історію змін. Результат такої міграції представлений нижче (рис. 2.25):

```

from django.db import migrations, models

class Migration(migrations.Migration):

    initial = True

    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name='Model3D',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
                ('title', models.CharField(max_length=100)),
                ('model_file', models.FileField(upload_to='')),
            ],
        ),
    ],

```

Рис. 2.25 Опис міграції відслідковування змін у базі даних

Після того, як застосували міграцію та таблицьки у базі даних були створені, можемо зайти у адмін-панель нашого сайту та почати створювати наші 3D-моделі (рис. 2.26).

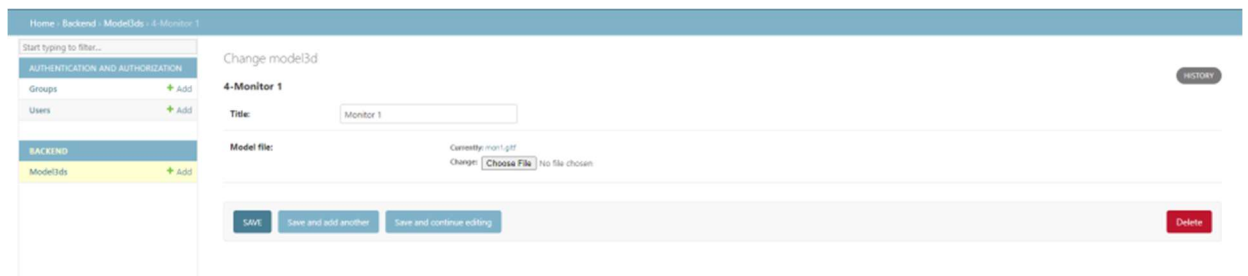


Рис. 2.26 Скріншот адмін-панелі сайту для створення 3D-моделі

Після того, як створили декілька моделей, настав час відобразити їх на нашому веб-сайті.

Для того щоб побудувати каркас веб-сайту використовуємо Bootstrap framework, який має набір HTML компонентів та CSS стилів що суттєво допоможе нам створити динамічний веб-сайт з хорошим зовнішнім виглядом, який буде коректно працювати і на комп'ютері, і на мобільних пристроях.

Отже, спочатку завантажуюмо javascript бібліотеки з веб-сайту bootstrap та css стилі та поміщаємо у директорію static, а також додаємо туди логотип нашого веб-сайту. У результаті отримаємо наступний контент (рис. 2.27):

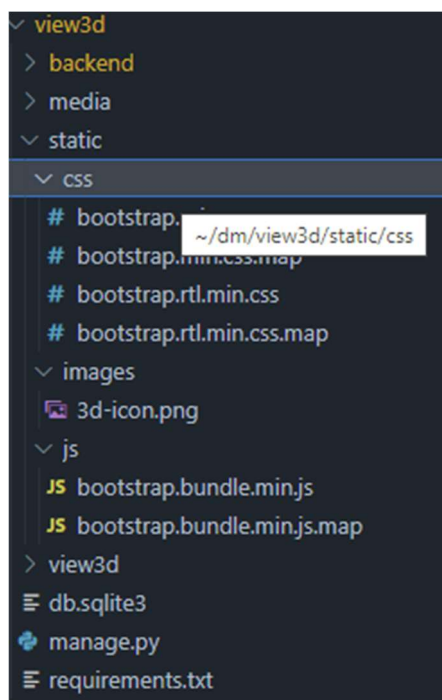


Рис. 2.27 Контент змісту створюваного сайту

Далі почнемо працювати над шаблоном головної сторінки сайту (рис. 2.28).

```

dm > view3d > backend > templates > index.html > html
1 <html>
2 <head>
3   {% load static %}
4   <link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet"/>
5   <script src="{% static 'js/bootstrap.bundle.min.js' %}"></script>
6 </head>
7 <body>
8   <header>
9     <div class="collapse bg-dark" id="navbarHeader">
10      <div class="container">
11        <div class="row">
12          <div class="col-sm-8 col-md-7 py-4">
13            <h4 class="text-white">About</h4>
14            <p class="text-muted">This project has been created to demonstrate the way we could visualize 3D models in
15          </div>
16          <div class="col-sm-4 offset-md-1 py-4">
17            <h4 class="text-white">Contact</h4>
18            <ul class="list-unstyled">
19              <li>Dmytro Shestopalov</li>
20              <li>dmytro.shestopalov@gmail.com</li>
21              <li>+38 096 123 12 12</li>
22            </ul>
23          </div>
24        </div>
25      </div>
26    </div>
27    <div class="navbar navbar-dark bg-dark shadow-sm">
28      <div class="container">
29        <a href="/" class="navbar-brand d-flex align-items-center">
30          
31          <strong>3D View</strong>
32        </a>
33        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarHeader" aria-cont
34          <span class="navbar-toggler-icon"></span>
35        </button>
36      </div>
37    </div>
38  </header>

```

Рис.2.28 Шаблон головної сторінки сайту

На головній сторінці веб-сайту буде відображатись список усіх

доступних 3D-моделей для перегляду. Якщо вивести туди одразу 3D-моделі, то завантаження сторінки може зайняти багато часу, оскільки файли 3D-моделей мають відносно великий розмір.

Тому було вирішено додати статичне зображення до наших моделей, яке використовувалось би як preview на головній сторінці, також додаємо поле description щоб можна було додати якийсь додатковий опис нашим моделям.

У результаті наша вдосконалена модель даних зміниться на наступну (рис. 2.29):

```
from django.db import models

class Model3D(models.Model):
    title = models.CharField(max_length=100)
    model_file = models.FileField()
    material_file = models.FileField(null=True)
    thumbnail = models.FileField()

    def __str__(self):
        return f"{self.pk}-{self.title}"
```

Рис. 2.29 «Вдосконалена» модель даних

І згенеруємо оновлену міграцію (рис. 2.30).

```
3 from django.db import migrations, models
4
5
6 class Migration(migrations.Migration):
7
8     dependencies = [
9         ('backend', '0001_initial'),
10    ]
11
12    operations = [
13        migrations.AddField(
14            model_name='model3d',
15            name='description',
16            field=models.TextField(default=''),
17            preserve_default=False,
18        ),
19        migrations.AddField(
20            model_name='model3d',
21            name='thumbnail',
22            field=models.FileField(default='', upload_to=''),
23            preserve_default=False,
24        ),
25    ]
26
```

Рис. 2.30 Генерація оновленої міграції

Після чого ми можемо оновити наш html шаблон та додати відповідні

поля для виводу (рис. 2.31).

```

<div class="album py-5 bg-light">
  <div class="container">

    <div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 g-3">
      {% for m in models %}
        <div class="col">
          <div class="card shadow-sm">
            

            <div class="card-body">
              <h6 class="card-text">{{m.title}}</h6>
              <p class="card-text">{{m.description}}</p>
              <div class="d-flex justify-content-between align-items-center">
                <div class="btn-group">
                  <a href="model/{{m.pk}}" class="btn btn-info" role="button">View</a>
                </div>
              </div>
            </div>
          </div>
        </div>
      {% endfor %}
    </div>
  </div>

```

Рис. 2.31 Оновлений html шаблон із полями для виведення

Для того щоб відобразити дані з нашої бази даних, використовуючи модель даних на веб-сторінці, треба створити View, який буде займатись тим, що буде передавати усі необхідні дані до шаблону.

Для цього створимо функцію у файлі views.py, яка буде вибирати усі існуючі моделі з бази даних та передавати до шаблону (рис. 2.32).

```

from django.shortcuts import render

from .models import Model3D

def index(request):
    context = {
        'models': Model3D.objects.all()
    }
    return render(request, 'index.html', context=context)

```

Рис. 2.32 Опис функції, яка буде вибирати усі існуючі моделі з бази даних та передавати до шаблону

У результаті отримаємо вигляд нашої головної сторінки, представлений нижче (рис. 2.33):

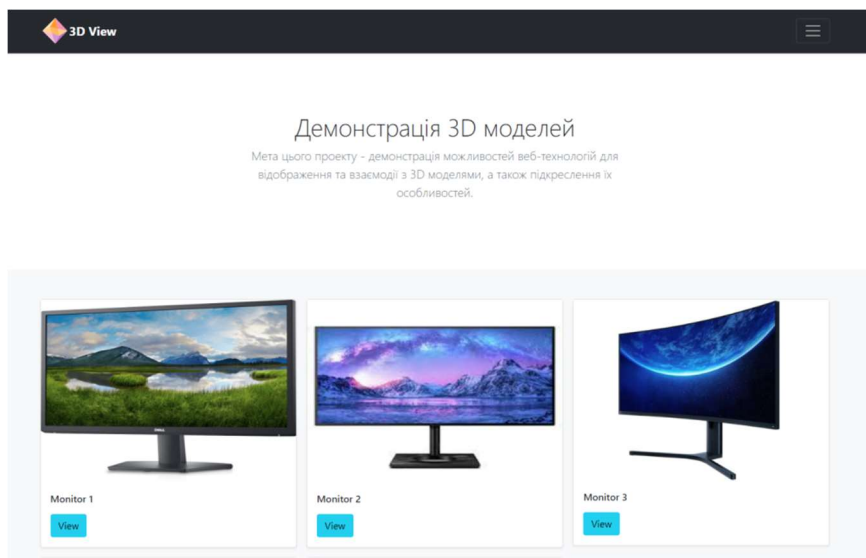


Рис. 2.33 Вигляд головної сторінки сайту

Наступним кроком буде відображення конкретної 3D-моделі на окремій сторінці. Для цього створюємо додатковий HTML шаблон (рис. 2.34), де буде основний каркас з bootstrap, а також підключена бібліотека three.js.

```
{% extends "base.html" %}

{% block extra_head %}
<script src="https://cdn.jsdelivr.net/npm/three@0.128.0/build/three.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/three@0.128.0/examples/js/controls/OrbitControls.js"></script>
<script src="https://cdn.jsdelivr.net/npm/three@0.128.0/examples/js/loaders/GLTFLoader.js"></script>
</style>
#model3d {
  overflow: hidden;
  margin: 0;
}
</style>
{% endblock extra_head %}
{% block content %}
<div class="album py-5 bg-light">
  <div class="container">
    <div class="row mb-3">
      <div class="col-8 themed-grid-col">
        <div id="model3d"></div>
      </div>
      <div class="col-4 themed-grid-col">
        <h6>{{model3d.title}}</h6>
        <p>
          {{model3d.description}}
        </p>
        <h6>Specifications</h6>
        <table class="table">
          <tbody>
            <tr>
              <td>{{spec.name}}</td>
              <td>{{spec.value}}</td>
            </tr>
          </tbody>
        </table>
      </div>
    </div>
  </div>
</div>
```

Рис. 2.34 Додатковий HTML шаблон, де буде основний каркас з bootstrap, а також підключена бібліотека three.js

Також було вирішено додати нову модель даних суто для технічних характеристик – Model3DSpecification (рис. 2.35).

```

class Model3DSpecification(models.Model):
    model_3d = models.ForeignKey(Model3D, on_delete=models.CASCADE, related_name="specifications")
    name = models.CharField(max_length=255)
    value = models.CharField(max_length=255)

    def __str__(self):
        return f"{self.model_3d.title}-{self.name}"

```

Рис. 2.35 Model3DSpecification

І створюємо відповідну міграцію (рис. 2.36).

```

2
3 from django.db import migrations, models
4 import django.db.models.deletion
5
6
7 class Migration(migrations.Migration):
8
9     dependencies = [
10         ('backend', '0002_model3d_description_model3d_thumbnail'),
11     ]
12
13     operations = [
14         migrations.CreateModel(
15             name='Model3DSpecification',
16             fields=[
17                 ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
18                 ('name', models.CharField(max_length=255)),
19                 ('value', models.CharField(max_length=255)),
20                 ('model_3d', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, to='backend.model3d')),
21             ],
22         ),
23     ]
24

```

Рис. 2.36 Створення міграції

Будемо відтворювати два типи моделей – obj та gltf. Вони будуть мати окремі шаблони.

Блок відображення gltf буде мати наступний вигляд (рис. 2.37):

```

{% block extra_js %}
<script>
  let scene = new THREE.Scene();
  let camera = new THREE.PerspectiveCamera(45, window.innerWidth / window.innerHeight, 0.1, 1000);
  camera.position.z = 3; // Отдаление камеры

  let renderer = new THREE.WebGLRenderer();
  renderer.setClearColor(0x404040);
  renderer.setSize(800, 600);

  renderer.domElement.setAttribute("id", "Church3DObj");
  document.getElementById('model3d').appendChild(renderer.domElement);

  let controls = new THREE.OrbitControls(camera, renderer.domElement);

  const aLight = new THREE.DirectionalLight(0xffffff, 1.5);
  aLight.position.setScalar(10);
  scene.add(aLight, new THREE.AmbientLight(0xffffff, 0.5));

  let loader = new THREE.GLTFLoader();
  let obj = null;

  loader.load('{{model3d.model_file.url}}', function (gltf) {
    obj = gltf.scene;
    scene.add(obj);
  });

  renderer.setAnimationLoop(_ => {
    renderer.render(scene, camera);
  });
</script>
{% endblock extra_js %}

```

Рис. 2.37 Блок відображення gltf

Додаємо можливість до сцени освітлення, можливість вільного керування камерою та переміщення. Сама ж модель рендериться за допомогою GLTFLoader-а, якому ми передаємо на вхід посилання на нашу 3D-модель.

Для obj файлу процес рендерингу виглядає трошки інакше (рис. 2.38).

```

52 <script>
53   let scene = new THREE.Scene();
54   let camera = new THREE.PerspectiveCamera(45, window.innerWidth / window.innerHeight, 0.1, 1000);
55   camera.position.z = 3; // Отдаление камеры
56
57   let renderer = new THREE.WebGLRenderer();
58   renderer.setClearColor(0x404040);
59   renderer.setSize(800, 600);
60
61   renderer.domElement.setAttribute("id", "Church3DObj");
62   document.getElementById('model3d').appendChild(renderer.domElement);
63
64   let controls = new THREE.OrbitControls(camera, renderer.domElement);
65
66   const aLight = new THREE.DirectionalLight(0xffffff, 1.5);
67   aLight.position.setScalar(10);
68   scene.add(aLight, new THREE.AmbientLight(0xffffff, 0.5));
69
70   var mesh = null;
71
72   var mtlLoader = new THREE.MTLLoader();
73   mtlLoader.load('{model3d.material_file.url}', function (materials) {
74
75     materials.preload();
76
77     const loader = new THREE.OBJLoader();
78     loader.setMaterials(materials);
79     let obj = null;
80     // load a resource
81     loader.load(
82       // resource URL
83       '{model3d.model_file.url}',
84       // called when resource is loaded
85       function (object_1) {
86         obj = object_1;
87         console.log(obj)
88         scene.add(obj);
89       },
90       // called when loading is in progresses
91       function (xhr) {
92
93         console.log((xhr.loaded / xhr.total * 100) + '% loaded');
94
95       },
96       // called when loading has errors
97       function (error) {
98
99         console.log('An error happened');
100
101       }
102     );
103

```

Рис. 2.38 Опис процесу рендерингу для об'єкту файлу

Тут також встановили світло та можливість керування та пересування камерою, відмінність полягає у тому, що Об'єкту моделі мають ще окремий файл з текстурами, які накладаються на 3D-модель, отже треба додатковий крок, щоб накласти їх на модель, використовуючи MTLLoader і після цього вони передаються у OBJLoader, який завантажує 3D-модель та накладає текстури.

Тепер треба ще створити view у нашому python-додатку, який би

передав необхідну модель у контекст шаблону. Вигляд view (рис. 2.39):

```
def model_details(request, pk):
    model_3d = Model3D.objects.get(id=pk)
    context = {
        'model3d': model_3d,
        'specifications': Model3DSpecification.objects.filter(model_3d_id=pk).all()
    }
    template_name = 'gltf-model-detail.html' if model_3d.model_file.url.endswith('.gltf') else 'obj-model-detail.html'
    return render(request, template_name, context=context)
```

Рис.2.39 Вигляд опису структури переглядача view

Тепер можемо перевірити результат нашої роботи (Рис. 2.40).

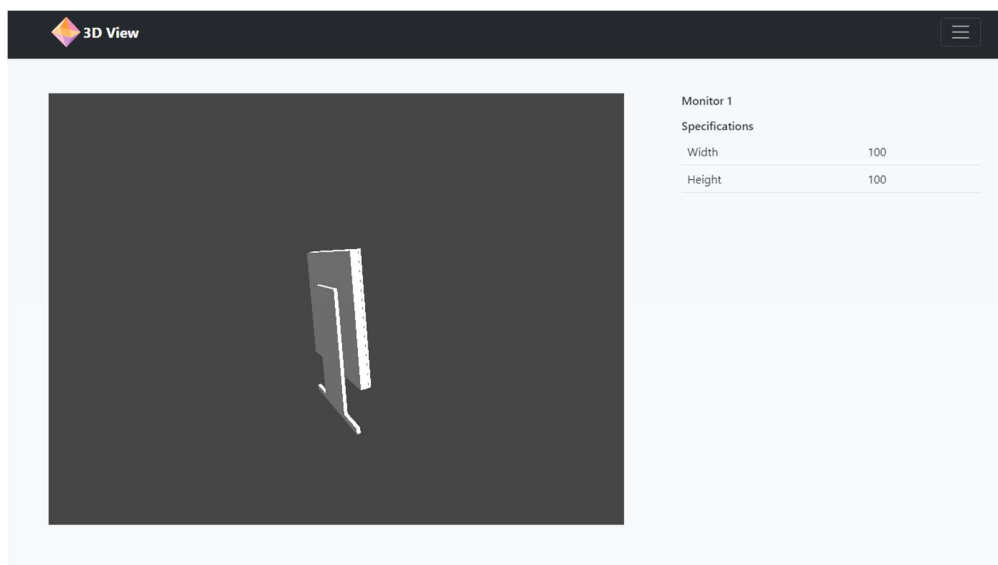


Рис. 2.40 Скріншот роботи переглядача view

З усіх видів контенту, представленого на різних сайтах, зазначимо залучення графічних 3D-об'єктів. За допомогою використання об'ємних графічних 3D-моделей є можливість досягти кращого сприйняття інформації з сайту, та забезпечити сам сайт великим функціоналом. Використовуючи 3D-моделі можемо продемонструвати досліджуваний об'єкт згідно тематики сайту. Отже, можна уникнути завантаження великої кількості графічних об'єктів, (фотографій, схем, малюнків тощо), тому що через використання інструментів маніпуляції користувач може обертати 3D-модель, розглянути її та дослідити з усіх сторін та обрати найзручніший для нього ракурс для дослідження об'єкту. Також при використанні графічних 3D-об'єктів є можли-

вість підвищення ступеня зацікавленості користувача конкретним програмним продуктом, а саме веб-сайтом, оскільки користувач першочергово оцінює зовнішній вигляд сайту, його інноваційність, сучасність та цікавість. Сайти, які містять безцінну інформацію, але оформленні безграмотно і не відповідають естетичним нормам, будуть відкинуті на задній план, а перевага буде спрямована на інші програмні ресурси. Використання 3D-об'єктів дозволяє скоротити обсяги інформації, необхідної для розміщення на сайті, і тим самим полегшують розробникам процес упорядкування, стилізування інформації та модерації на інформаційних ресурсах.

При самостійному створенні сайту графічні 3D-об'єкти можуть інтегруватись на сайт за допомогою використання допоміжних програмних інструментів у вигляді різноманітних додатків та бібліотек. Інструментами для інтеграції 3D-моделей на розроблений в даній роботі сайт є бібліотека ThreeJs та завантажувач JLTFloder, а основний процес відбувається за допомогою Java script.

2.5 Висновки до розділу 2

Створення 3D-моделі – це складний процес, за допомогою якого можна візуалізувати будь-який об'єкт. На основі проведеного у першому розділі аналізу різних редакторів 3D-моделювання було обрано Blender, адже він є дійсно потужним програмним засобом. Додатково використали Tinkercad та Sweet Home 3D. Програмні засоби, які обрали дозволили реалізувати поставлені завдання. У даному розділі також був описаний інструментарій та процес розробки сайту (прототипу віртуального музею) для завантаження 3D-моделей [41].

ВИСНОВКИ

За результатами виконаної роботи можемо сформулювати загальні висновки:

1. Створення 3D-моделей задля інтеграції на веб-сайт – це комплексна багатогранна робота, яка має враховувати інформацію про об'єкт, який необхідно графічно змоделювати, про характеристики та особливості, моделювання об'єкту та web-програмування для додавання моделі на веб-сайт. Проаналізовано стан предметної області, а саме використання 3D-об'єктів при створенні сучасних сайтів, комп'ютерних ігор, анімацій тощо.

2. Досліджено програмні засоби – графічні платформи для створення 3D-моделей за наступними параметрами: швидкість роботи, якість візуалізації, доступність інструментів моделювання, рівень складності використання. Складений порівняльний аналіз програмних платформ 3D-моделювання.

3. Розроблено 3D-моделі моніторів за допомогою обраних графічних редакторів, а саме: Blender, Tinkercad та Sweet Home 3D. Після створення моделей за спеціальною процедурою здійснено їх підготовку до інтеграції на веб-сайт, а також проведений експорт у відповідні формати. Як результат, отримано 3D моделі для наповнення контенту розробленого власноруч веб-сайту [41].

4. Досліджені засоби розробки сайтів та обрані інструменти для розробки Django та Three.js.

5. Розроблена структура та дизайн web-сайту [41] з інтеграцією 3D-моделей. Візуалізовані сцени експозиції 3D-моделей об'єктів за допомогою публікації на розробленому сайті.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Використання 3D-графіки в різних галузях людської діяльності. URL: <https://shaiu21.blogspot.com/>.
2. Скорюкова Я. Г., Жиганов В. А. Особливості 3D-моделювання в середовищі Blender: матеріали XLVIII науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2019) : збірник доповідей. – Вінниця : ВНТУ, 2019. С. 1440–1444. URL: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKewjOhOC2u6D_AhXQpIsKHXueDyUQFnoECBQQAQ&url=https%3A%2F%2Fconferences.vntu.edu.ua%2Findex.php%2Fall-fbtegp%2Fall-fbtegp-2021%2Fpaper%2Fdownload%2F12177%2F10156&usg=AOvVaw3XAGZwlVwCVovjJraYzSm4.
3. Гаврилов В. П. 3D-графіка: навчальний посібник. Харків: ХНЕУ ім. С. Кузнеця, 2018. 127 с.
4. Драченко, Д. О. Сучасне 3-D моделювання / Д. О. Драченко, О. Є. Тесленко // Перспективні напрямки розвитку сучасних інформаційних систем та технологій: зб. тез доп. всеукр. наук.-практ. студ. конференція, 18 квіт. 2018р., м. Кропивницький. Кропивницький: ЦНТУ, 2018. С. 23-24.
5. 3D-моделювання обладнання, складу, заводу чи промислового приміщення - KONSORT. KONSORT. URL: <https://konsort.com.ua/service/3d-modelyuvannya-obladnannya-skladu-zavodu-chy-promyslovogo-prymishhennya/>.
6. Використання комп'ютерних технологій 3D-моделювання у професійній діяльності фахівців. Головна | EDU Blog. URL: <https://edublog.com.ua/blog/id1335942996/posts/статті/використання-компютерних-технологій-3d-моделювання-у-професійній-діяльності-фахівців>.

7. Популярні програми для 3D моделювання в Україні. DOI: <https://doi.org/194.44.12.92:8080/xmlui/bitstream/handle/123456789/6627/V1-%20Конференція-2-8-2022-133-135.pdf?sequence=1>.
8. Осадча К. П., Чемерис Г. Ю. Добір засобів тривимірного моделювання для формування графічної компетентності майбутніх бакалаврів комп'ютерних наук. URL: <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiSob2sgPz-AhWFtYsKHcH5A9MQFnoECDQQAQ&url=https%3A%2F%2Fjournal.iit-ta.gov.ua%2Findex.php%2Fittl%2Farticle%2Fview%2F1713%2F1273&usg=AOvVaw3PN0tv9bKBBv0iIT1Na97s>.
9. Тривимірна графіка – 3D-моделювання. URL: <https://easy3dprint.com.ua/uk/trivimirna-grafika-3d-modelyuvannya/>.
10. Математичні основи тривимірної графіки. Перетворення в просторі. Проектування. Однорідні координати. URL: https://stud.com.ua/156184/informatika/matematichni_osnovi_trivimirnoyu_grafikoju_peretvorennya_prostori_proektuvannya_odnoridni_koordinati.
11. Кулій Д. Розробка факультативного курсу для старшокласників «Математичні основи комп'ютерної графіки»: дипломна робота. Чернівці: ЧНУ ім. Ю. Федьковича, 2021. 86 с.
12. Тривимірна графіка. Принципи тривимірного моделювання. URL: <https://informatik.pp.ua/uroky/9-klas/konspekty-uchnia/tryvumirna-hrafika-pryntsypu-tryvumirnoho-modeliuvannia>.
13. Пальчевський Б. О., Валецький, Б. П., Вараніцький Т. Л. Системи 3D-моделювання: навчальний посібник. Луцьк, 2016. 176 с.
14. Інженерна та комп'ютерна графіка: підручник / Михайленко В. Є., Найдис В. М., Підкоритов А. М. та ін.; за ред. В. Є. Михайленка. Київ: Вища школа, 2000. 342 с.
15. Геометричне моделювання. URL: https://uk.wikipedia.org/wiki/Геометричне_моделювання.

16. Текстура (тривимірна графіка). URL: [https://uk.wikipedia.org/wiki/Текстура_\(тривимірна_графіка\)](https://uk.wikipedia.org/wiki/Текстура_(тривимірна_графіка)).
17. Методика формування практичних навичок графічної реконструкції засобами тривимірної візуалізації. URL: <https://science.uipa.edu.ua/wp-content/uploads/2021/04/Graphic-reconstruction-1.pdf>.
18. Що Таке 3D-Рендеринг? URL: <https://gudvil.com.ua/ua/blog/scho-take-3d-rendering-6-grundlegende-prinzipien/>.
19. ProEngineering. URL: <https://www.ptc.com/en/products/creo/pro-engineer>.
20. 3D-MAX. URL: <https://www.autodesk.com/products/3ds-max/overview>.
21. SketchUp Pro. URL: <https://www.sketchup.com/ru/products/sketchup-pro>.
22. AutoCAD. URL: <https://www.autodesk.com/products/autocad/overview>.
23. Inventor. URL: <https://www.autodesk.com/products/inventor/overview>.
24. Компас-3D. URL: <https://kompas.ru/>.
25. Blender. URL: <https://www.blender.org>.
26. ISO 9126. URL: https://ru.wikipedia.org/wiki/ISO_9126.
27. SolidWork. URL: <https://solidworks.softico.ua>.
28. Примітиви форм сітьової поверхні Blender. URL: http://wiki.b3d.org.ua/index.php/Примітиви_форм_сітьової_поверхні_Blender.
29. Дундяк С. Р. Розробка інтерактивної 3D візуалізації: дипломна робота. Київ: Національний технічний університет України «Київський політехнічний інститут» ім. Ігоря Сікорського, 2017. 50 с. URL: https://cad.kpi.ua/attachments/093_2017d_Dundiak.pdf.
30. Blender для початківців. URL: https://ru.wikibooks.org/wiki/Blender_для_начинающих.
31. Рендеринг. URL: <https://uk.wikipedia.org/wiki/Рендеринг>.
32. Sweet Home 3D. URL: <https://www.sweethome3d.com/ru/>.
33. Вчимося користуватися Sweet Home 3D URL: <https://uk.soringprepair.com/use-sweet-home-3d/>.

34. Сайти з 3D-графіки: 7 найкращих онлайн ресурсів CGI індустрії. URL:
<https://cgischool.ua/saity-z-3d-grafiky-top-7/>.
35. Tinkercad. URL: <https://softdroids.com/1450-tinkercad.html>.
36. Django. URL: <https://www.djangoproject.com/>.
37. Python.org. URL: <https://www.python.org>.
38. Blend4Web. URL: <https://www.blend4web.com/ru>.
39. Three.js. URL: <https://threejs.org/>.
40. Sketchfab.com. URL: <https://sketchfab.com/>.
41. Демонстрація 3D моделей / Розробник Шестопапов Д. С. URL:
<http://209.38.208.108/>.