

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента *Валах Іллі Анатолійовича*
(ПІБ)

академічної групи *122-19ск-2*
(шифр)

спеціальності *122 Комп'ютерні науки*
(код і назва спеціальності)

освітньої програми *Комп'ютерні науки*
(назва освітньої програми)

на тему: *Розробка інформаційного забезпечення синтезу архітектури
комп'ютера на основі тримірного моделювання та візуалізації*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>проф. Мещеряков Л.І.</i>			
розділів:				
спеціальний	<i>проф. Мещеряков Л.І.</i>			
економічний	<i>доц. Касьяненко Л.В.</i>			
Рецензент				
Нормоконтролер	<i>доц. Гуліна І.Г.</i>			

Дніпро
2022

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

завідувач кафедри
програмного забезпечення комп'ютерних систем

(повна назва)

І.М. Удовик

(підпис)

(прізвище, ініціали)

« » 2022 року

ЗАВДАННЯ

**на кваліфікаційну роботу
бакалавра**

(назва освітньо-кваліфікаційного рівня)

студента 122-19ск-2
(група)

Валах І.А.

(прізвище та ініціали)

тема кваліфікаційної роботи

Розробка інформаційного забезпечення

синтезу архітектури комп'ютера на основі тримірної моделювання та візуалізації

затверджена наказом ректора НТУ «ДП» від 18 травня 2022 №268-с

Розділ	Зміст виконання	Термін виконання
<i>Спеціальний</i>	<i>На основі матеріалів проектно-технологічної практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми</i>	<i>13.05.2022 р.</i>
<i>Економічний</i>	<i>Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки</i>	<i>27.05.2022 р.</i>

Завдання видав

(підпис)

проф. Мещеряков Л.І.

(посада, прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Валах І.А.

(прізвище, ініціали)

Дата видачі завдання: 14.01.2022 р.

Термін подання кваліфікаційної роботи до ЕК: 13.06.2022 р.

РЕФЕРАТ

Пояснювальна записка: 78 с., 70 рис., 3 дод., 30 джерел.

Об'єкт розробки: програмне забезпечення демонстраційного характеру по збору персонального комп'ютера.

Мета кваліфікаційної роботи: створення додатку що допоможе розвинути навички машинного навчання, та дати розуміння того як будеється ПК.

У вступі розглядається сама проблема та її аналіз. Наведено області використання та їх важливість у нашому часі.

У першому розділі проведено аналіз предметної галузі, визначено актуальність завдання та призначення розробки, розроблена постановка завдання, задані вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі виконано аналіз існуючих рішень, обрано платформу для розробки, виконано проєктування і розробка програми, наведено опис алгоритму і структури функціонування програми, наведені характеристики складу параметрів технічних засобів, зображено у візуальному вигляді процес моделювання комплектуючих та їх текстурування.

Для досягнення поставленої мети було обрано наступні програми: «Blender», «Adobe Photoshop», «Visual Studio Code», «Voxel 3D».

Практичне значення розробленої інформаційної системи полягає в створенні повноцінного працеспроможного додатка, що надає можливість користувачеві в зрозумілій для нього формі показати на базі трьохмірного моделювання та візуалізації повноцінний збір персонального комп'ютера.

Актуальність данної інформаційної системи полягає в тому, що в наш час є доцільним використання комп'ютерів у повсякденному житті для розваг, тому саме ця програма за допомогою сучасних методів розробки в середовищі Blender дає змогу користувачу зібрати свій власних ПК без додаткової допомоги.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведений підрахунок вартості роботи по створенню проєкту та розраховано час на його створення.

Список ключових слів: КОМП'ЮТЕР,МОДЕЛЮВАННЯ,ГРАФІКА,3D, BLENDER, HARD SURFACE, ДИЗАЙН.

ABSTRACT

Explanatory note: 78 p., 70 fig., 3 appendix, 30 sources.

Object of development: demonstration software for assembling a personal computer

The purpose of the qualification work: to create an application that will help develop machine learning skills, and to give an understanding of how to build a PC.

The introduction considers the problem itself and its analysis. Areas of use and their importance in our time are given.

In the first section the analysis of the subject area is carried out, the urgency of the task and purpose of development are defined, the statement of the task is developed, requirements to software realization, technologies and software are set.

The second section analyzes the existing solutions, selected a platform for development, designed and developed the program, describes the algorithm and structure of the program, gives the characteristics of the parameters of technical means, visualizes the process of modeling components and their texturing.

To achieve this goal, the following programs were selected: "Blender", "Adobe Photoshop", "Visual Studio Code", "Voxel 3D".

The practical significance of the developed information system is to create a full-fledged workable application that allows the user to show in a form understandable to him on the basis of three-dimensional modeling and visualization a complete collection of personal computer.

The relevance of this information system is that nowadays it is advisable to use computers in everyday life for entertainment, so this program with the help of modern development methods in the Blender environment allows the user to assemble their own PCs without additional help.

The economic section determines the complexity of the developed information system, calculates the cost of work to create a project and calculates the time for its creation.

Keyword list: COMPUTER, SIMULATION, GRAPHICS, 3D, BLENDER, HARD SURFACE, DESIGN.

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

ІС – інформаційна система

ОС – операційна система

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

ІТ – інформаційні технології

GPU – graphics processing unit

CPU – central processing unit

RAM – random access memory

Зміст

РЕФЕРАТ.....	3
ABSTRACT.....	4
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	9
1.1 Загальні відомості з предметної галузі.....	10
1.2 Призначення розробки та галузь застосування.....	11
1.3 Підстава для розробки.....	11
1.4 Постановка завдання.....	12
1.5 Вимоги до програми або програмного виробу.....	12
1.5.1 Вимоги до функціональних характеристик.....	12
1.5.2 Вимоги до інформаційної безпеки.....	12
1.5.3 Вимоги до складу та параметрів технічних засобів.....	12
1.5.4 Вимоги до інформаційної та програмної сумісності.....	13
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	14
2.1 Функціональне призначення системи.....	14
2.2 Опис застосованих математичних методів.....	14
2.3 Опис використаних технологій та мов програмування	16
2.4 Опис структури програми та алгоритмів її функціонування.....	21
2.5.1 Моделювання процесору.....	21
2.5.2 Моделювання відеокарти.....	27

2.5.3	Моделювання SSD.....	31
2.5.4	Моделювання HDD.....	33
2.5.5	Моделювання оперативної пам'яті.....	37
2.5.6	Моделювання повітряного охолодження для процесору.....	41
2.5.7	Моделювання повітряного охолодження для системного блоку.....	44
2.5.8	Моделювання блоку живлення.....	46
2.5.9	Моделювання монітору.....	49
2.5.10	Моделювання материнської плати.....	51
2.6	Опис розробленої системи.....	52
2.6.1	Використані технічні засоби.....	53
2.6.2	Використані програмні засоби.....	53
2.6.3	Виклик та завантаження програми.....	54
2.6.4	Сайт галерея.....	55
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ.....		56
3.1	Розрахунок трудомісткості та вартості розробки програмного продукту.....	58
3.2	Розрахунок витрат на створення програми.....	59
ВИСНОВКИ.....		62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		63
Додаток А. Код програми.....		64
Додаток Б. Відгук керівника економічного розділу.....		76
Додаток В. Перелік файлів на диску.....		78

Вступ

3D графіка або тривимірна графіка - це один з розділів комп'ютерної графіки, набір прийомів та інструментів, які дозволяють створити об'ємні об'єкти за допомогою форми та кольору.

Від двомірної вона відрізняється тим, що виконує побудову геометричної проекції трьохмірної моделі (віртуального простору), робиться це за допомогою спеціальних пакетів. Отримана модель відповідає об'єктам реального світу (наприклад, будинок, людина, автомобіль) або абстрактної (проекція чотиривимірного фрактала).

Сьогодні 3D графіка міцно укоринилась в багатьох сферах нашого життя це: будівництво (візуалізація об'ємних архітектурних зображень будівель, об'єктів, інтер'єру, екстер'єру); виробництво (об'єктне моделювання); телебачення (модельовані фото у глянцевиx журналах, відеоролики, спецефекти в кіно), ігрова індустрія (3D-анімація та віртуальні світи, розробка комп'ютерних ігор); поліграфія (реалізація поліграфічної продукції), реклама (електронні презентації та каталоги, рекламні щити тощо) тощо.

Етапи створення трихмірного зображення можна поділити на такі пункти:

1. Моделювання – створення математичної моделі об'єкта.
2. Текстурування – двухмірне зображення об'єкта, яке в подальшому наноситься на 3D модель.
3. Використання елементів світла у сцені.
4. Створення анімації.
5. Рендеринг – фінальний вигляд зображення з урахуванням текстур та світла.
6. Композитинг – постобробка заображення після виконаного рендеру.

РОЗДІЛ I

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1. Загальні відомості з предметної галузі

Найосновнішим у 3D графіці є модель. Якщо модель якісно зроблена у контексті оптимізованої сітки, то це майже половина роботи, яка в подальшому не доставить проблем, бо з якісною моделлю усі подальші етапи пайплайну пойдуть як по маслу. А якщо модель має погану сітку, то ні які тестури її не спасуть, бо вони будуть не коректно відображатися. Анімація теж буде виглядати дуже погано так як сітка буде деформуватися не коректно. В данній роботі буде використовуватися анімація, але вона буде базуватися на keyframe, а не на рігінгу, тож якщо об'єкти не будуть деформуватися – сітка не важлива. Традиційне зображення являє собою роботу у двох осях X та Y, але у трьохмірній графіці додається 3-тя вісь – Z.

Плюси цього методу в тому, що для фінального рендеру досить розмістити модель у сцені та за допомогою спеціальних програм виконати рендер. І цю операцію можна повторювати скільки завгодно разів. Нажаль у 2D така розкіш не можлива, в силу відсутності 3-ї осі. Так як світло у 3D виставляється у сцену і може редагуватися у будь-який час, в традиційному зображенні художнику кожен раз треба створювати джерела світла спочатку.

Найзнаменитішим методом моделювання є полігональне моделювання. Його суть в тому, що використовується примітив – трикутник у відеоіграх і чотирикутник у кіно. Багатокутники з яких складається модель називаються полігонами. В ігрових двигунах використовуються трикутник, так як відеокарта рендерить зображення у трикутниках.

Щоб модель максимально була схожа на оригінал вона містить багато полігонів. Але в той же час велика кількість полігонів може негативно сказатися наприклад при експорті у ігровий двигун. Тож важливо дотримуватися правил оптимізації. Величезна кількість полігонів відповідає величезному числу точок, якими вони будуються, що призводить до збільшення кількості даних, що

доводиться обробляти процесору. Отже при реалізації моделі часто доводиться йти на угоду між оптимізацією та деталізацією. У зв'язку з цим з'явилися такі терміни, як *high poly* (високополігональна модель) і *low poly* (низькополігональна модель). У комп'ютерних іграх використовуються низькополігональні моделі, так як візуалізація в іграх проводиться у реальному часі.

Після етапу моделювання виходить лише математична модель, що містить інформацію лише про геометричну форму об'єкта. Для того, щоб надати моделі необхідний колір, застосовуються текстури. Текстура це двовірне зображення, яке накладається на тривимірну модель.

Текстури бувають процедурними (тобто згенерованими за допомогою алгоритму) і намальованими в графічному редакторі. З погляду тривимірної графіки, матеріал є математична модель, що визначає параметри поверхні. Перш ніж накласти текстуру на модель, необхідно зробити її розгортку – тобто, подати поверхню моделі за допомогою проєкції на площину. Створення моделі можна поділити на такі етапи :

- Моделювання по референсам та кресленню.
- Створення UV – розгортки у спеціальній програмі.
- Створення процедурних або хендпейн текстур.
- Налаштування сцени (додання джерел світла та композиції).

Після виконання цих пунктів модель готова для візуалізації та фінального рендеру.

У 3D графіці після виготовлення моделі її необхідно розмістити в сцену до інших об'єктів, додати камеру та освітлення, і лише після цього можна отримати фінальне зображення. Це зображення прораховується за допомогою фізичної моделі, це модель поширення променя світла з урахуванням відображення, заломлення, розсіювання тощо. При традиційному малюванні художник сам зображує відблиски, відображення, тіні і т.д., а в 3D графіці автор готує сцену з урахуванням геометрії, освітлення, властивостей камери і візуалізатор сам розраховує підсумкове зображення.

1.2. Призначення розробки та галузь застосування

Останні декілька років можна спостерігати потребу використання 3D графіки майже у кожній існуючій області сучасного життя.

В ці дні важливість створення комп'ютерної графіки важко переоцінити. Світ не стоїть на місці, швидкі темпи поліпшення технічного прогресу зробили комп'ютерну графіку актуальною практично в усіх напрямках промислово-побутової сфери. 3D графіка є невід'ємним супутником архітекторів, дизайнерів, культурних діячів, рекламних експертів, експертів у галузі ігрової промисловості та важкого машинобудування. Здібності 3D графіки у світі практично нескінченні.

Досить частим явищем є розробка рекламної кампанії товару ще до появи в світ. У цьому разі єдиним виходом є використання комп'ютерної графіки. При правильному підході до процесу виробництва 3D моделі та анімації людині дуже важко визначитися, що перед ним зображення не справжнього продукту, а його комп'ютерної моделі. Спецтехнології прогресують дуже швидко, тому більше немає необхідності використовувати важкі прикраси або реквізит, які неможливо отримати звичайним методом. Тепер потрібен лише комп'ютер, який просто вирішить цю проблему.

1.3. Підстави для розробки

Підставою до розробки кваліфікаційної роботи бакалавра на тему «Розробка інформаційного забезпечення синтезу архітектури комп'ютера на основі трьохмірного моделювання та візуалізації» є наказ по Національному технічному університету «Дніпровська політехніка» від 18 травня 2022р.№268-с.

1.4. Постановка завдання

Метою кваліфікаційної роботи є аналіз доступних технологій та методів розробки для створення за допомогою моделювання проєкції комплектуючих у ПК.

Для досягнення поставленої мети, необхідно дослідити наступні завдання:

1. Проаналізувати та обрати певні комплектуючі.
2. Перевірити співвідношення серед комплектуючих.
3. В 3D пакеті змоделювати усі комплектуючі та зовнішні елементи.
4. Додати текстури до об'єктів для подальшого рендеру.
5. Створити анімацію.
6. Додати джерела світла.
7. Налаштувати сцену та зробити фінальний рендер та анімацію.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

До основних вимог належить додати тільки те, щоб сцена була оптимізована для подальшої та зручної навігації у viewport.

1.5.2. Вимоги до інформаційної безпеки

Щодо інформаційної безпеки треба зазделегіть переконатися в надійності операційної системи та мати ліцензійне програмне забезпечення. Мати встановлений на ПК антивірус, при належності також зробити архівацію даних та резервну копію ОС.

Щодо апаратної безпеки належить придбати блок безперебійного живлення.

1.5.3. Вимоги до складу та параметрів технічних засобів

До технічних параметрів з боку користувача не потребує специфічних характеристик. Так як програма буде зарендерена на спеціальному ком'ютері.

Тож нижче наведені рекомендації щодо мінімальних/рекомендованих програм для перегляду проєкту:

- Відекодек PotPlayer або його аналоги.
- Відеокарта з підтримкою Direct X.
- Аудіодрайвер Realtech.

1.5.4. Вимоги до інформаційної та програмної сумісності

Кваліфікаційна робота на тему «Розробка інформаційного забезпечення синтезу архітектури комп'ютера на основі тримірного моделювання та візуалізації» передбачає використання 3D пакету Blender, засобів програмування на мові JavaScript у середі Visual Studio Code та повинна бути сумісною з операційною системою Windows.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення системи

Метою кваліфікаційної роботи є продемонструвати користувачеві у зрозумілій візуальній формі поетапне складання ПК до робочого стану. В програмі буде розглядено у деталях кожний компонент ПК у 3D, що дасть у повній змозі спостерігати складання наживо.

2.2. Опис застосованих математичних методів

Математичні методи моделювання та текстурювання (math node рис.2.1.) зображенні на рис.2.2. у вигляді «ноду» в якій можна спостерігати базові математичні операції, такі як: add, subtract, multiply, divide та інші.

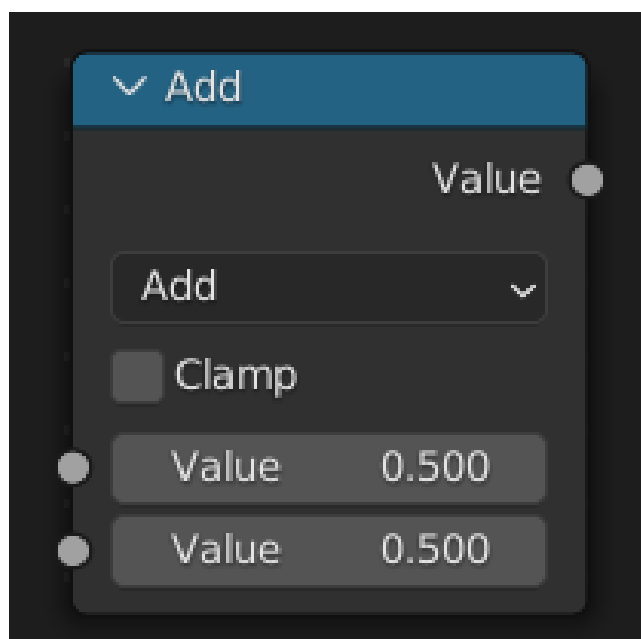


Рис.2.1. Math Node

Functions	Comparison	Rounding	Trigonometric	Conversion
Add	Minimum	Round	Sine	To Radians
Subtract	Maximum	Floor	Cosine	To Degrees
Multiply	Less Than	Ceil	Tangent	
Divide	Greater Than	Truncate	Arcsine	
Multiply Add	Sign	Fraction	Arccosine	
Power	Compare	Modulo	Arctangent	
Logarithm	Smooth Minimum	Wrap	Arctan2	
Square Root	Smooth Maximum	Snap	Hyperbolic Sine	
Inverse Square Root		Ping-Pong	Hyperbolic Cosine	
Absolute			Hyperbolic Tangent	
Exponent				

Рис.2.2. Математичні методи ноду Math

Також використовуються векторні математичні операції рис.2.3. та рис.2.4.

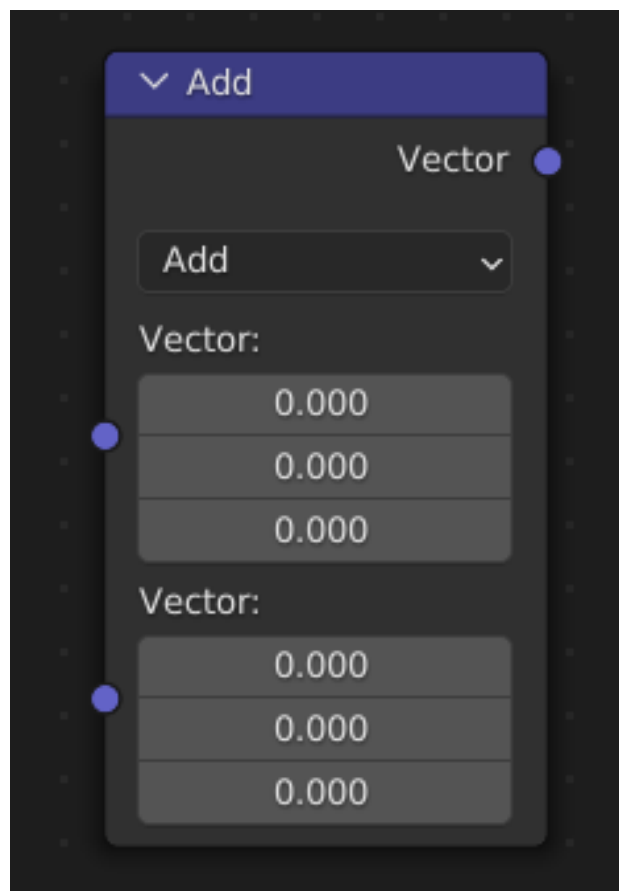


Рис.2.3. Vector math node

<u>A</u> dd	<u>N</u> ormalize
<u>S</u> ubtract	<u>A</u> bsolute
<u>M</u> ultiply	Minimum
<u>D</u> ivide	Max <u>i</u> imum
Multi <u>p</u> ly Add	Floor
<u>C</u> ross Product	Ceil
<u>P</u> roject	Fraction
<u>R</u> eflect	Modulo
<u>R</u> efract	<u>W</u> rap
<u>F</u> aceforward	Snap
<u>D</u> ot Product	Sine
<u>D</u> istance	Cosine
<u>L</u> ength	<u>T</u> angent
Scale	

Рис.2.4. Математичні методи ноду vector math node

2.3. Опис використаних технологій та мов програмування

В роботі використовується Blender. Blender це безкоштовний пакет для створення 3D з відкритим вихідним кодом. Він підтримує весь 3D-конвеєр моделювання, ригінг, анімацію, моделювання, візуалізацію, композицію та відстеження руху, навіть редагування відео та створення ігор. Досвідчені користувачі використовують API Blender для сценаріїв Python для налаштування програми та написання спеціалізованих інструментів; часто вони включені в майбутні випуски Blender. Blender добре підходить для окремих осіб і невеликих студій, які отримують переваги від його уніфікованого конвеєра та адаптивного процесу розробки.

Blender є кросплатформним і однаково добре працює на комп'ютерах Linux, Windows і Macintosh. Його інтерфейс використовує OpenGL, щоб забезпечити послідовний досвід. Щоб підтвердити конкретну сумісність, у списку підтримуваних платформ вказуються ті, які регулярно тестуються командою розробників.

Blender позиціонується як програма для виробництва та редагування тривимірної графіки, візуалізації, анімації, виробництва комп'ютерних ігор і навіть скульптингу. Схожі програми вимагають багато джерел від комп'ютера, але встановлений Blender займає близько трьохсот мегабайт і оптимізований для слабких ПК. Додатковою перевагою є стійке та швидке встановлення пакету завдяки професійній команді розробників. Як і у всіх програмах моделювання, користувач працює у свого роду сцені або в'юпорт. Тут 3D модель звичайно створюється і редагується, а також обертається, переміщається, масштабується і т.д. Тут же відображаються всі видозміни, пов'язані з процесами анімації, текстурювання та візуалізації. 3D моделювання представлено практично всіма існуючими методами виробництва та роботи з об'ємними моделями. Доступно проектування об'єктів за допомогою примітивів, полігонів, NURBS-кривих, кривих Безьє, метасфер, булевих операцій, Subdivision Surface та базових інструментів для скульптингу. Як і в 3Ds Max, програма пропонує величезну кількість різних модифікаторів, що застосовуються до моделі. Для анімації у розпорядженні користувача такі інструменти, як ригінг (скелетна анімація), інверсна кінематика, деформація сітки, обмежувачі, ключові кадри, редагування вагових показників вершин тощо. Чудово реалізована динаміка жорстких і м'яких тіл, а також анімація частиць. Програма дозволяє накладати кілька текстур на один об'єкт, і обладнана рядом інструментів для текстурювання, включаючи UV-мапінг і часткове налаштування текстур. Ряд шейдерів, що настроюються, надає еластичності в роботі з матеріалами. Пакет оснащений декількома інструментами візуалізації, а також підтримує інтеграцію з різними зовнішніми рендерерами.

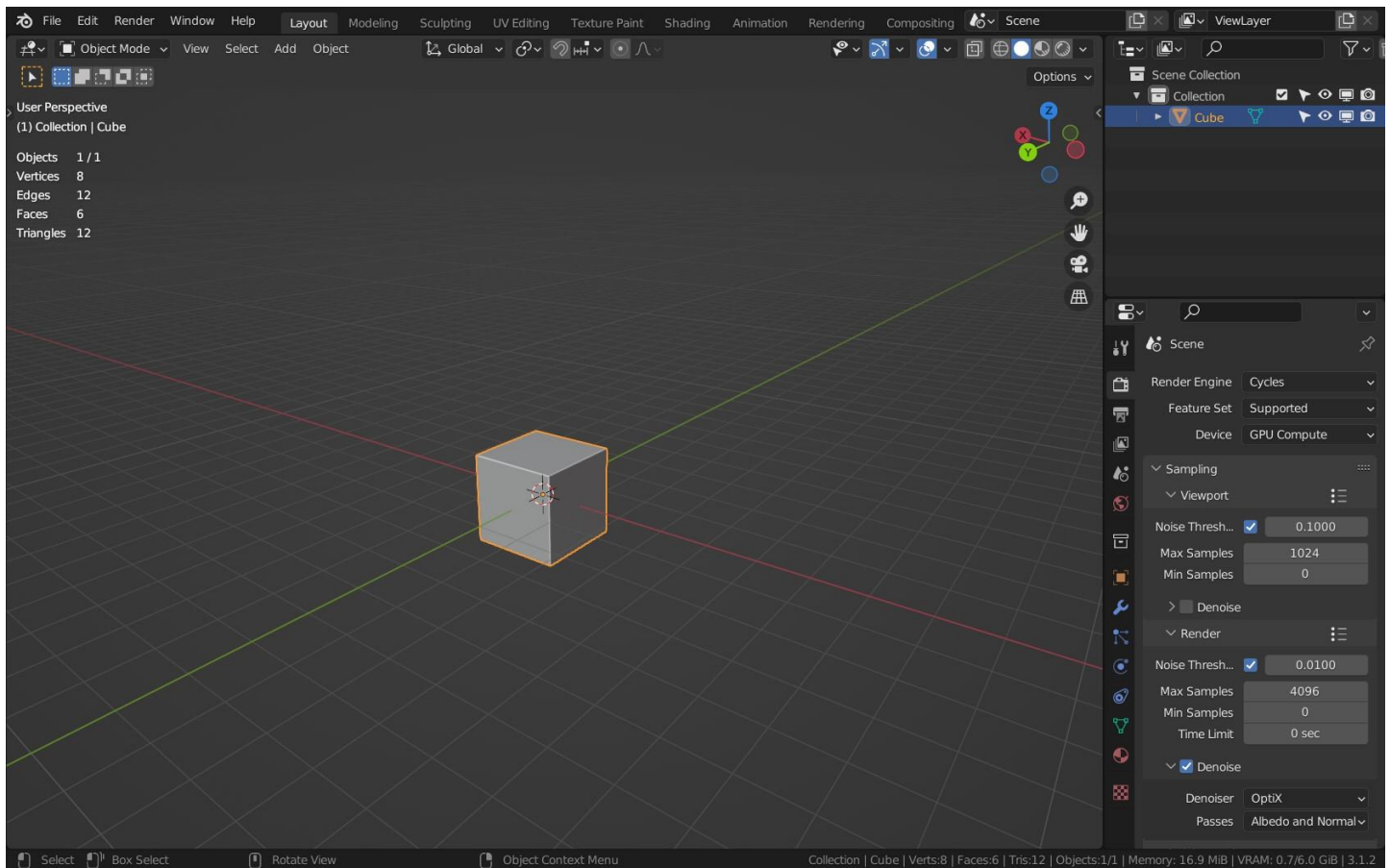


Рис.2.5. Інтерфейс Blender

В роботі використовується JavaScript. JavaScript – мова програмування, яка підтримує об'єктно-орієнтований, імперативний та функціональний стиль.

JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів програм. Найширше застосування знаходить у браузерах як мову сценаріїв для надання інтерактивності веб-сторінкам.

Средою програмування обрано Visual Studio Code. Visual Studio Code засіб для створення, редагування та завантаження сучасних вебзастосунків і програм для хмарних систем. Visual Studio Code розповсюджується безкоштовно і доступний у версіях для платформ Windows, Linux і OS X.

Компанія Microsoft представила Visual Studio Code у квітні 2015 на конференції Build 2015. Це середовище розробки стало першим кросплатформовим продуктом у лінійці Visual Studio.

За основу для Visual Studio Code використовуються напрацювання вільного проєкту Atom, що розвивається компанією GitHub. Зокрема, Visual Studio Code є надбудовою над Atom Shell, що використовує браузерний рушій Chromium і Node.js. Примітно, що про використання напрацювань вільного проєкту Atom і на сайті Visual Studio Code, і в пресрелізі, і в офіційному блозі не згадується.

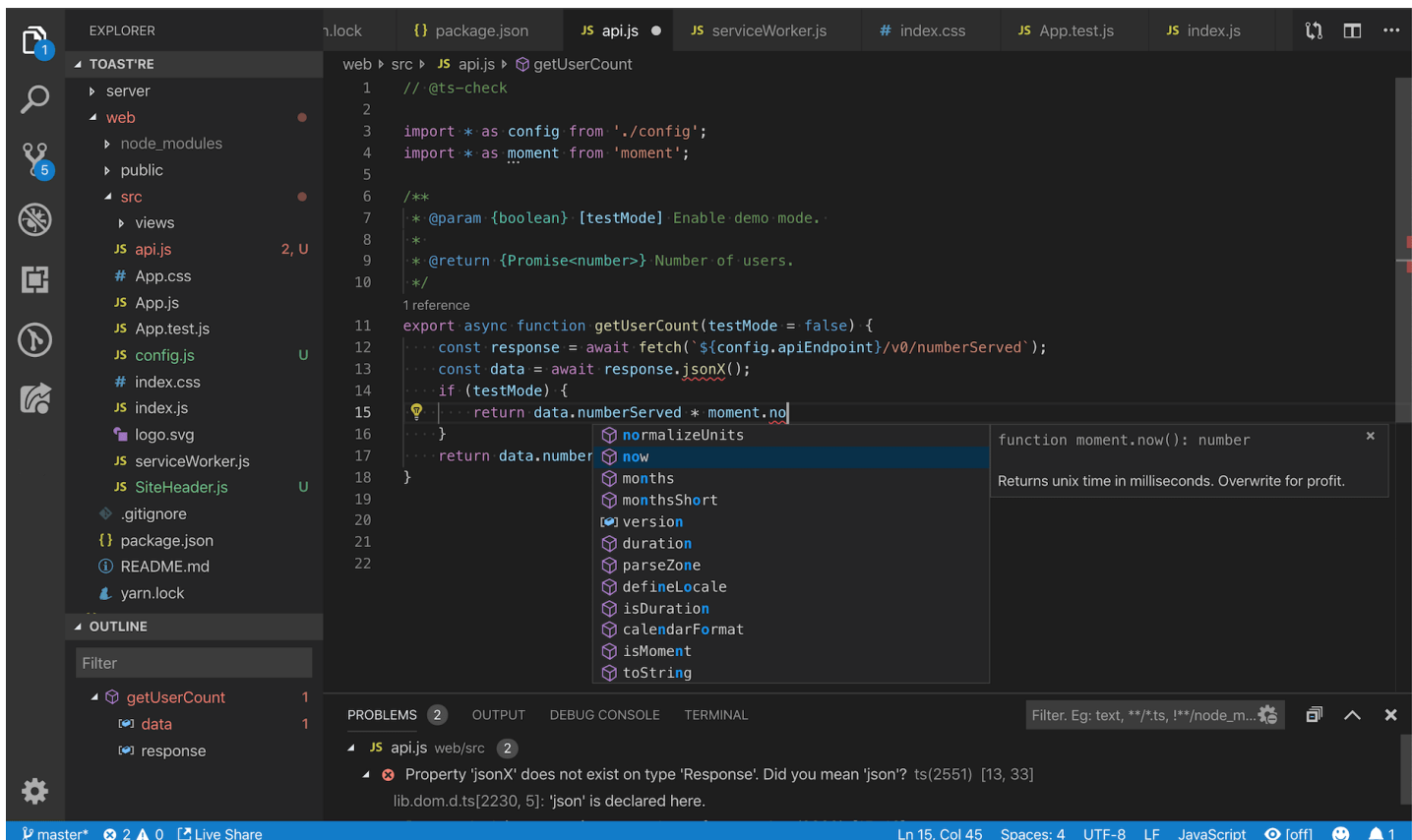


Рис. 2.6. Інтерфейс Visual Studio Code

Для редагування рендерів використовується програма Photoshop. Adobe Photoshop графічний редактор, розроблений і поширюваний фірмою Adobe Systems. Цей продукт є лідером ринку в галузі комерційних засобів редагування растрових зображень і найвідомішим продуктом фірми Adobe. Часто цю програму називають просто Photoshop (Фотошоп). У наш час Photoshop доступний на платформах Mac OS X/Mac OS і Microsoft Windows. Ранні версії редактора були портовані під SGI IRIX, але офіційна підтримка була

припинена, починаючи з третьої версії продукту. Для версії CS і CS6 можливий запуск під Linux за допомогою альтернативи Windows API — Wine.

Photoshop головним чином призначений для редагування цифрових фотографій та створення растрової графіки. Особливості Adobe Photoshop полягають у багатому інструментарії для операції створення і обробки зображень, високій якості обробки графічних зображень, зручності й простоті в експлуатації, широких можливостях до автоматизації обробки растрових зображень, які базуються на використанні сценаріїв, механізмах роботи з кольоровими профілями, які допускають їх втілення в файли зображень з метою автоматичної корекції кольорових параметрів при виводі на друк для різних пристроїв, великому наборі команд фільтрації, за допомогою яких можна створювати найрізноманітніші художні ефекти.

Базові інструменти редагування дозволяють змінювати тон, насиченість зображення, обтинати його, накладати фотофільтри, виправляти перспективу тощо. Photoshop підтримує так звані шари прозорі області зображення, на яких розміщуються елементи фотомонтажу, текст, геометричні фігури. Програма містить інструменти для роботи з текстом і нескладними фігурами, дозволяє зображати робочі контури, задавати текстам і фігурам стилі оформлення. Для роботи з окремими фрагментами зображення передбачені різні типи виділення: за фігурою, в режимі «малювання» зони виділення, за діапазоном кольорів тощо. Існують різноманітні фільтри для деформації та стилізації зображення, такі як фільтри розмиття, імітації різних художніх технік. Photoshop також містить інструменти для цифрового живопису, зокрема набори пензлів. Користувач може змінювати їх розмір, кут нахилу, колір.

2.4. Опис структури системи та алгоритмів її функціонування

Розробка архітектури компонентів ПК почнемо з того, що додамо примітив, який максимально схожий на кінечний об'єкт, який ми хочемо створити. Нижче перелічено компоненти, які потрібно змоделювати:

1. CPU – центральний процесор.
2. GPU – відеокарта.
3. Материнська плата.
4. SSD – твердотільний накопичувач.
5. HDD – звичайний жорсткий диск.
6. RAM – оперативна пам'ять.
7. Блок живлення.
8. Монітор.
9. Системний блок.

2.5.1. Моделювання процесору

Першим в списку йде процесор, тож з нього й почнемо. Додамо у сцену примітив «Куб» та використовуючи команду «Scale» створимо першу частину процесора. Далі додамо примітив «Циліндр» та за допомогою модифікатора Boolean у верхній частині створимо відтвір для подальшого встановлення у сокет процесора. Також для реального вигляду додамо модифікатор «Bevel» та модифікатор «WeightNormals» для того щоб зробити фаски та підправит шейдінг у моделі. (рис. 2.8)

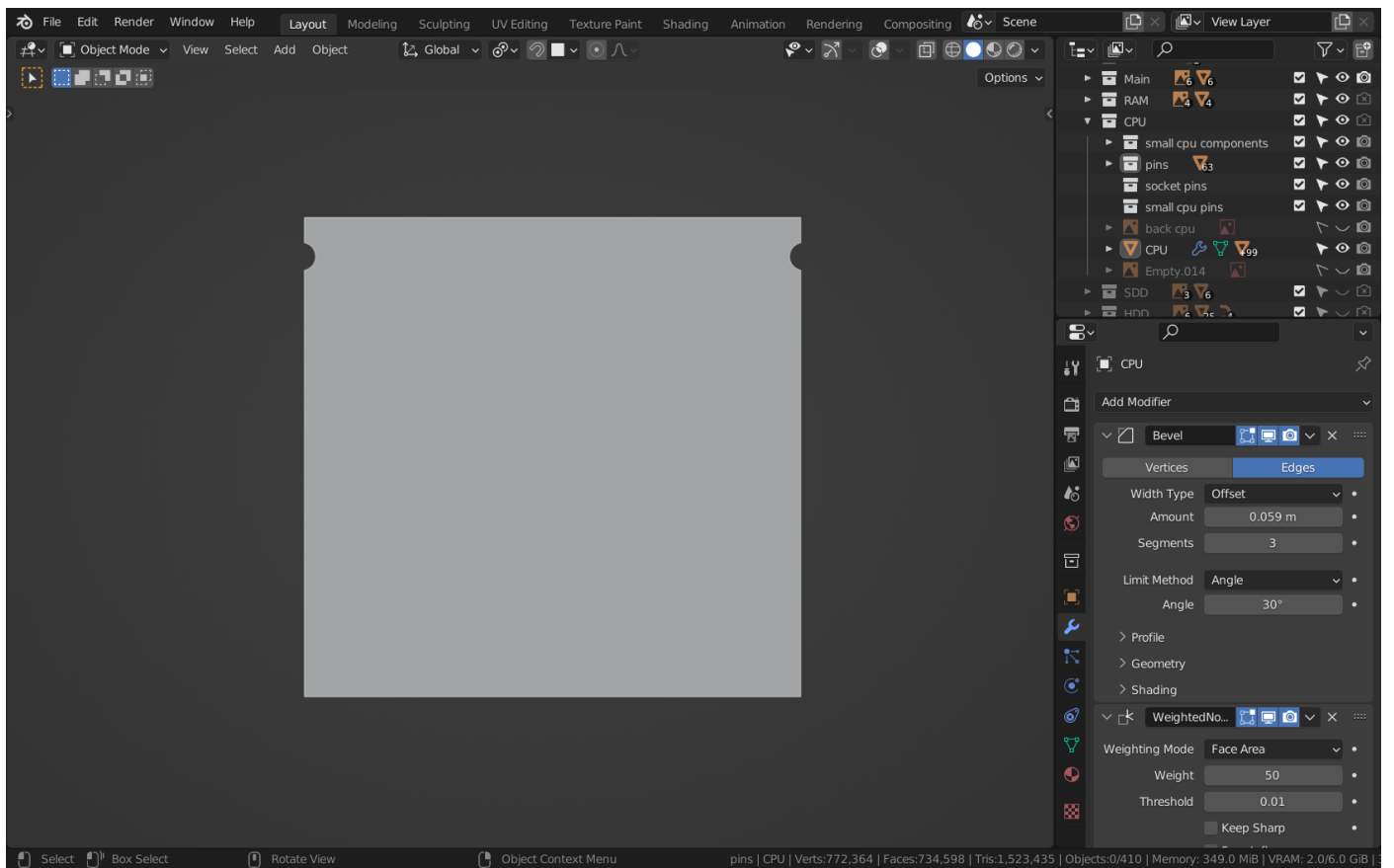


Рис.2.8. Додання примітиву та маніпуляції з ним

Наступним кроком знов додамо примітив «Куб» та проведемо операції які описані вище, але на цей раз додамо модифікатор Mirror для того щоб операції, які виконуються з однієї сторони дублювалися на іншу.

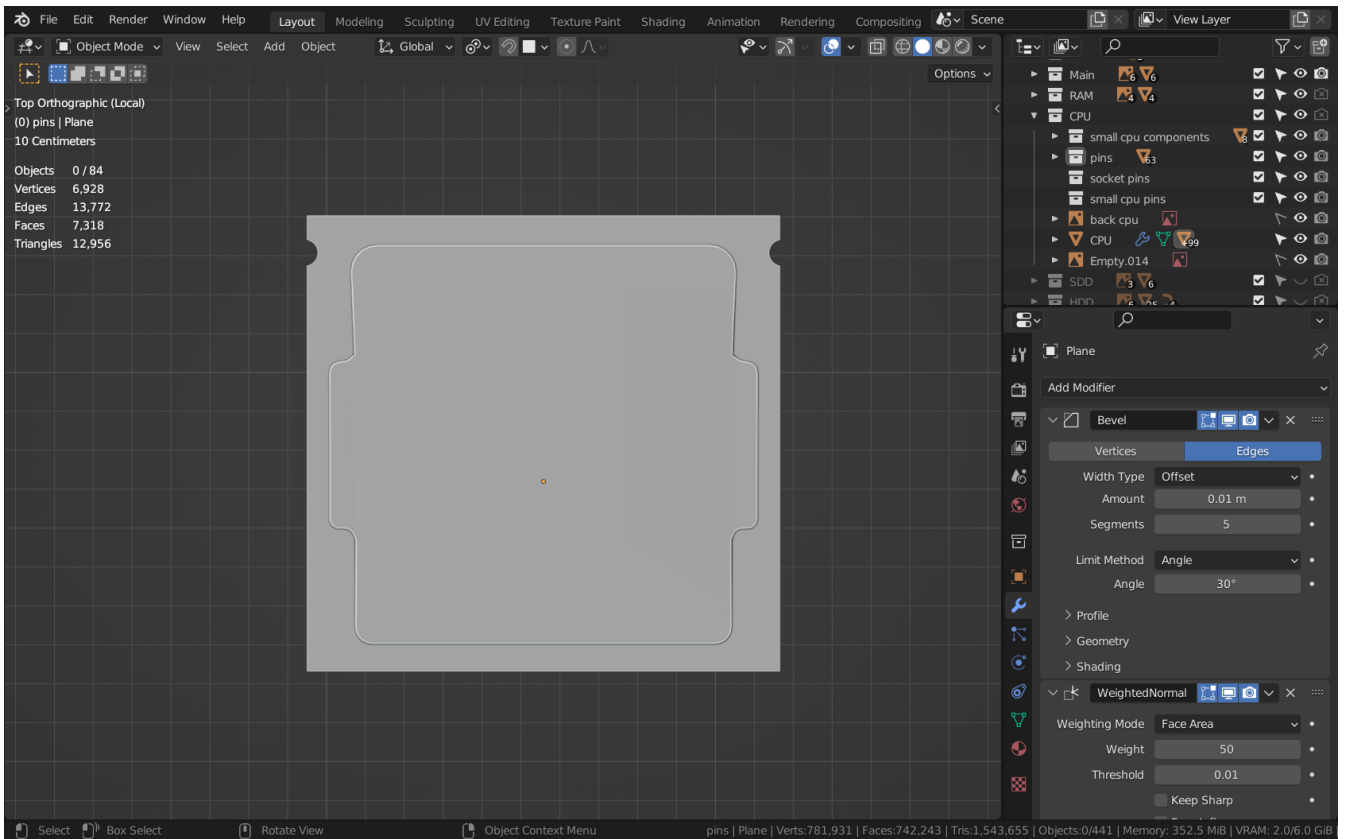


Рис.2.9. Додання примітиву та маніпуляції з ним з використанням модифікатора

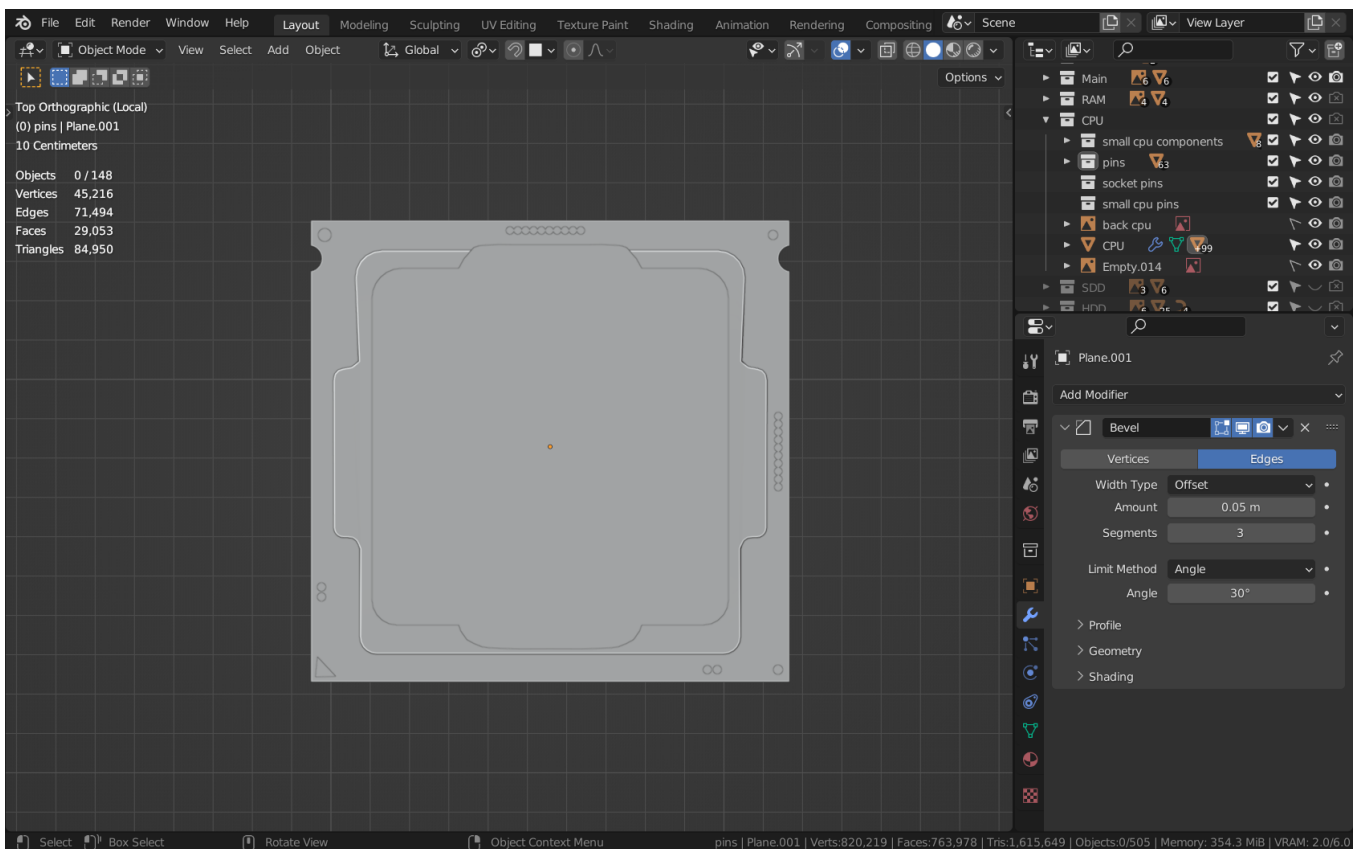


Рис.2.10. Додамо останню частину процесора

Після того як завершено моделювання процесора треба створити текстури, в даній роботі будуть використовуватися складні PBR (Physical Base Rendering) матеріали для найкращого результату. В Blender це робиться за допомогою нодової системи програмування, яка працює на базі Python.

У режимі Shading потрібно вибрати компонент до якого бажаємо додати матеріал. Для оптимізування простіру, закриємо непотрібні вікна та почнемо додавати матеріал. Але спочатку додамо до нашої сцени світло. Я буду використовувти HDRI (High Dynamic Range Imaging) простими словами це зображення реального світу зробленою людиною у форматі 360 градусів, яке передасть світло у реальному часі. Тепер можемо створити матеріал для процесору.

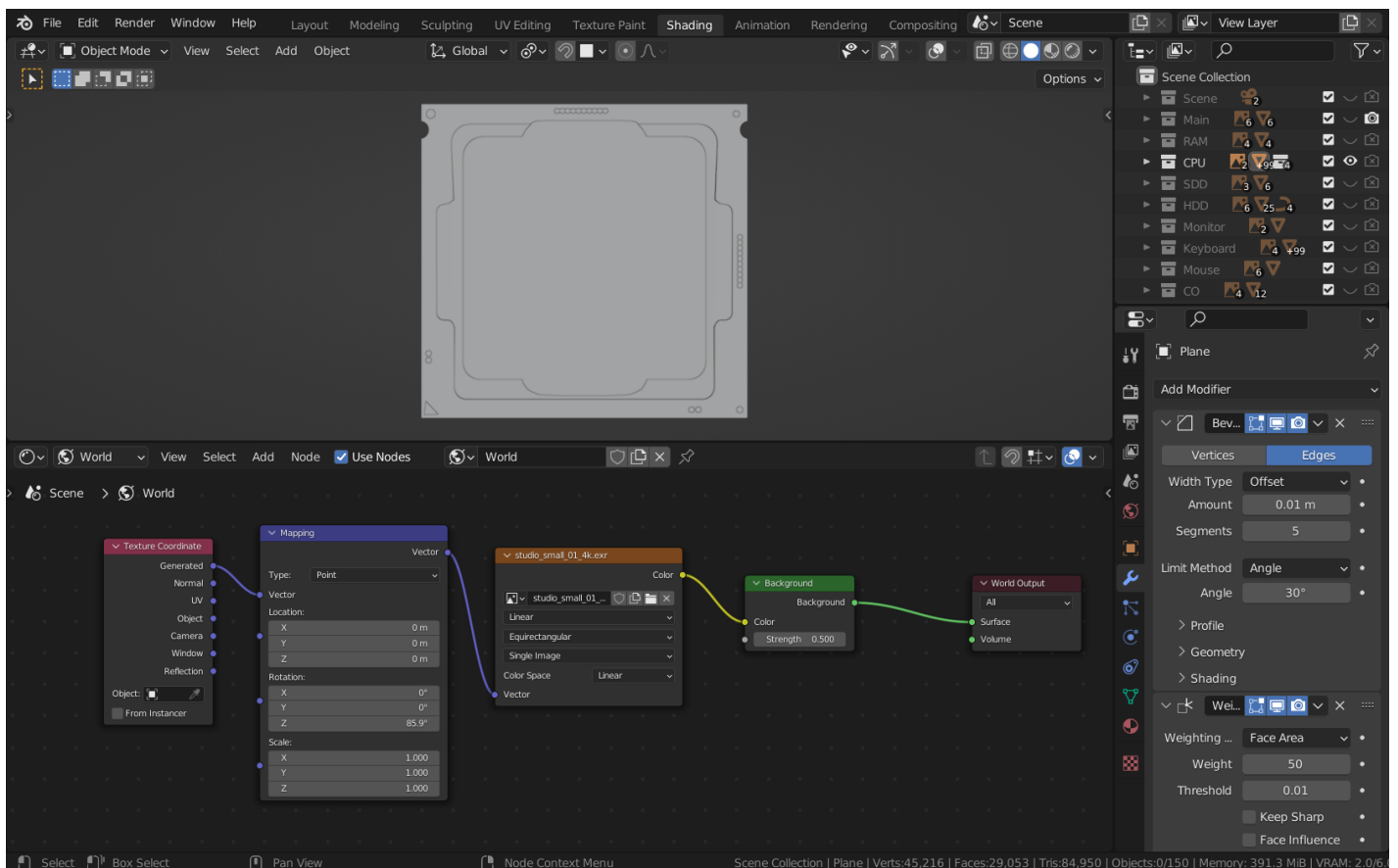


Рис.2.11. Додамо карту HDRI

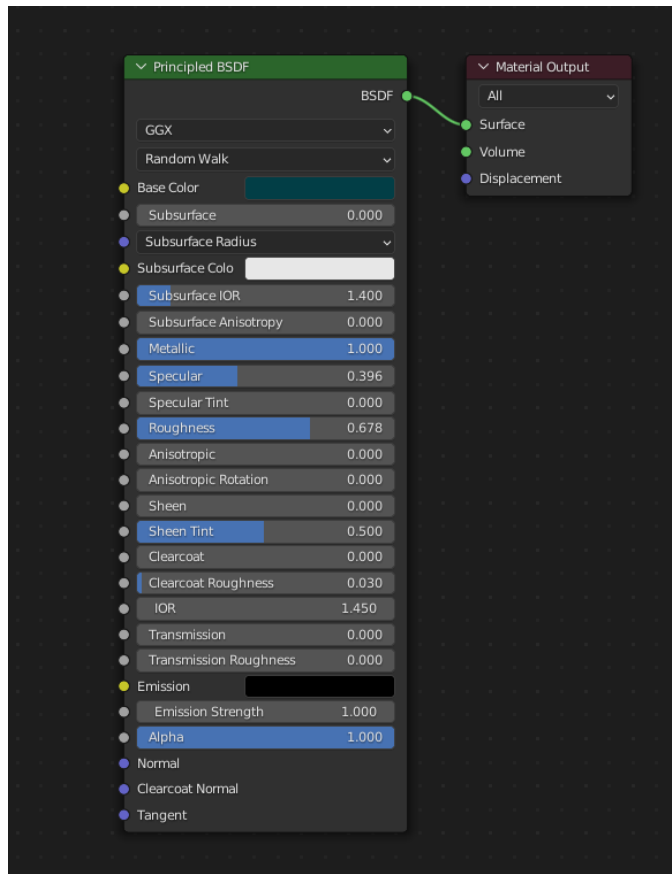


Рис.2.12. Матеріал процесору

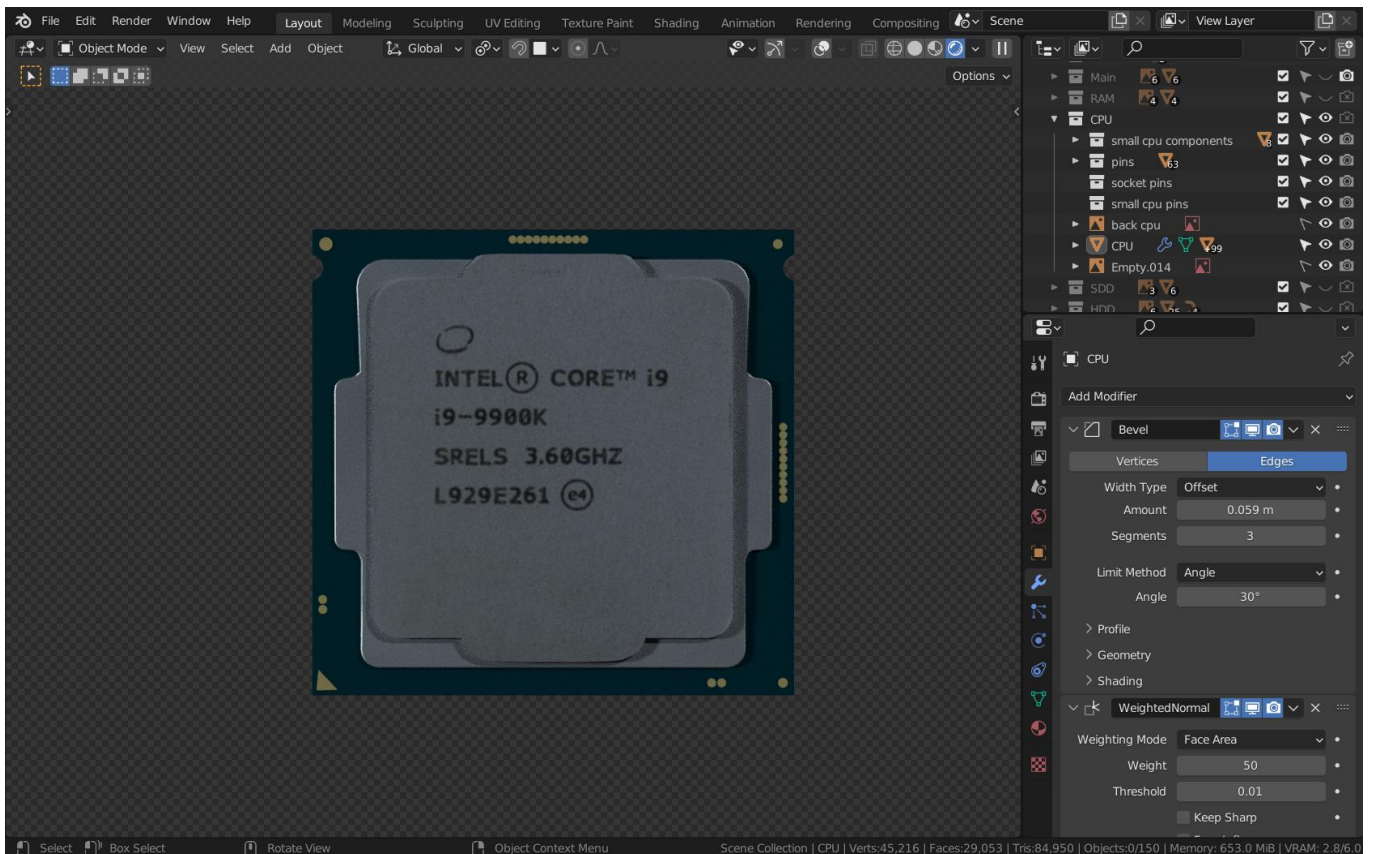


Рис.2.13. Фінальна версія процесора (вид спереду)

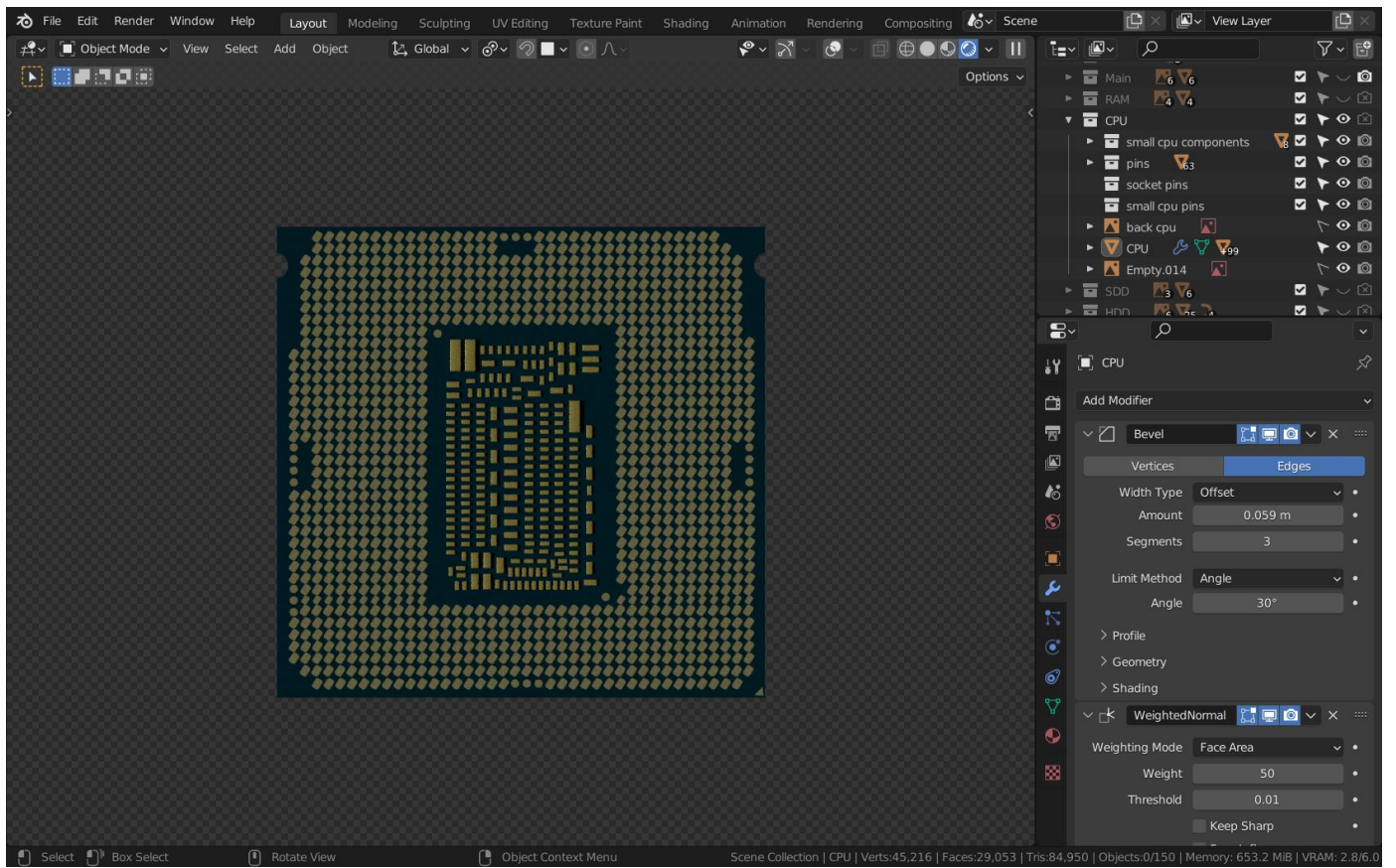


Рис.2.14. Фінальна версія процесора (вид ззаду)



Рис.2.15. 3D версія (solid mode)



Рис.2.16. 3D версія (render mode)

2.5.2. Моделювання відеокарти

Тепер почнемо моделювати відеокарту. Ця модель буде складніше ніж процесор, тому потрібно розбити її на маленькі частини, та моделювати кожную з них окремо, а потім зібрати воедино. Використовуючи інструмент «Spin» створимо зовнішню форму відеокарти. А завдяки модифікатору «Mirror» можна моделювати тільки одну частину, а друга буде відзеркалена.

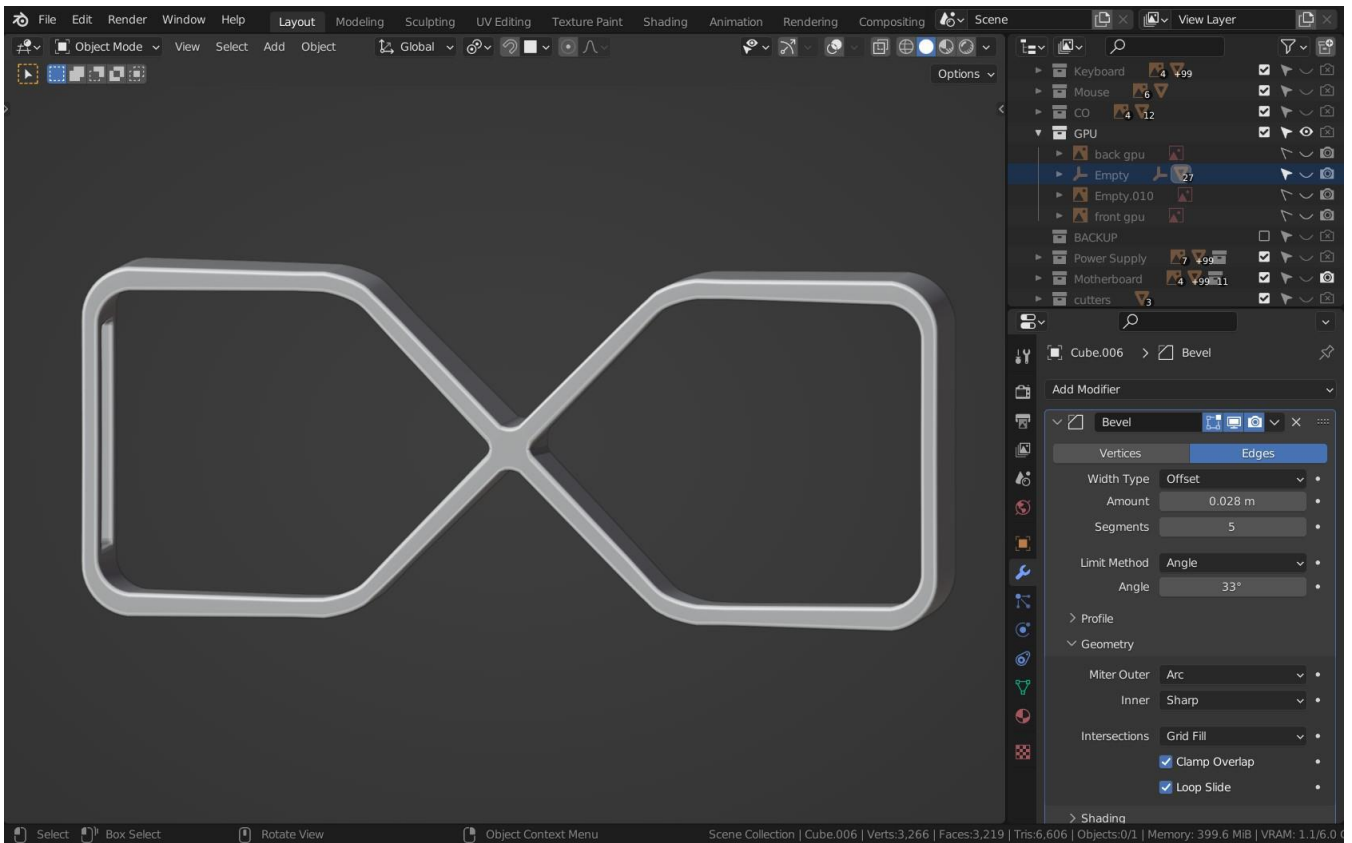


Рис.2.17. Моделювання зовнішньої частини

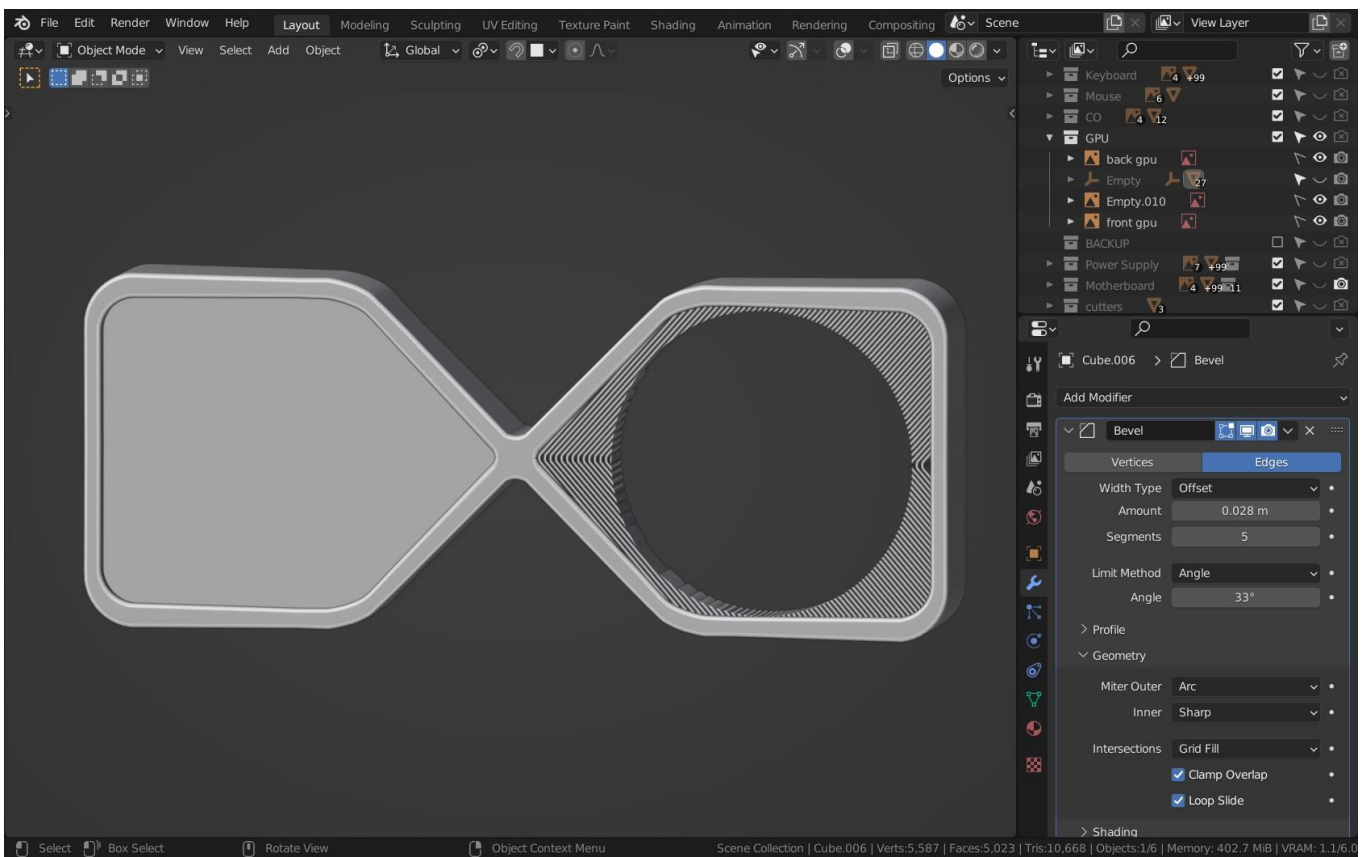


Рис.2.18. Моделювання внутрішньої частини

Далі додамо вентилятор та декілька частини по середині

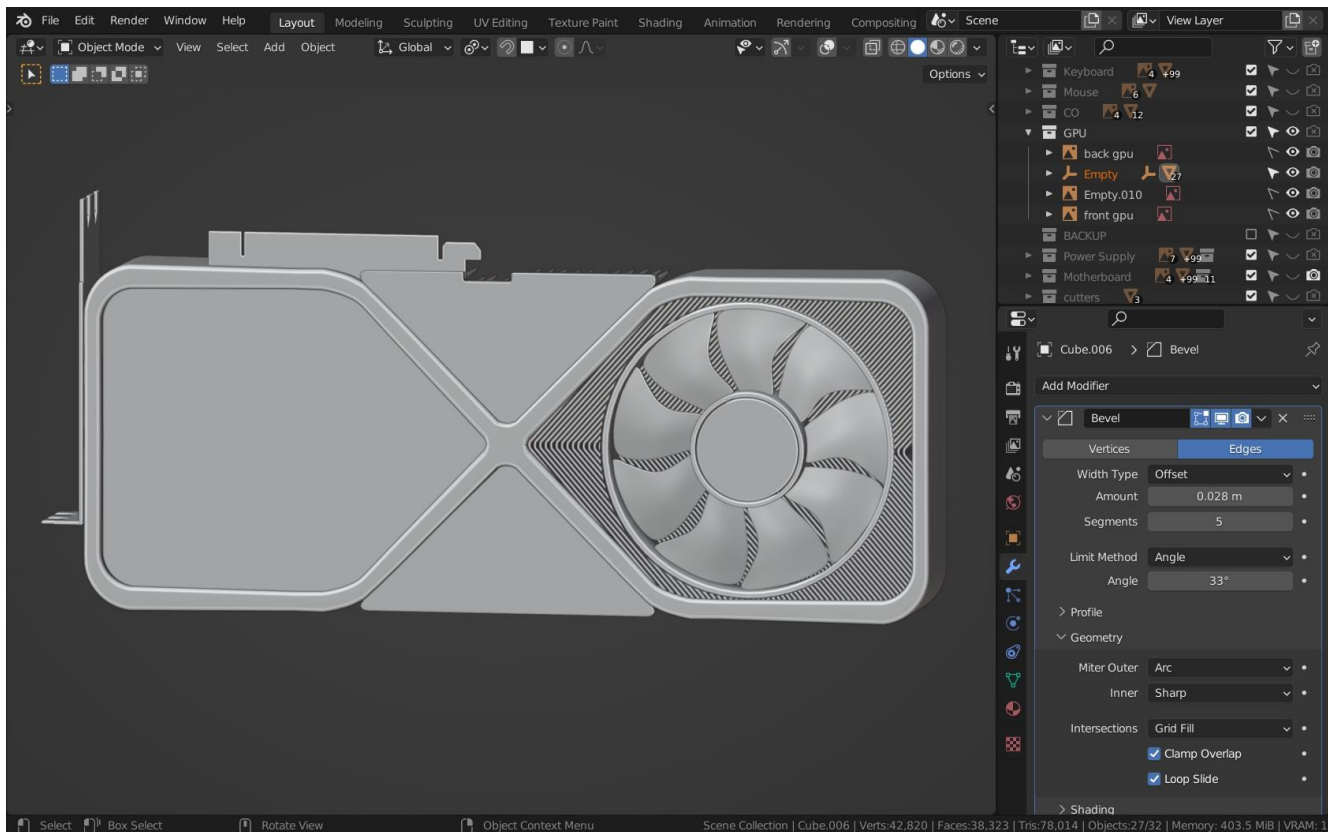


Рис.2.19. Моделювання вентилятора

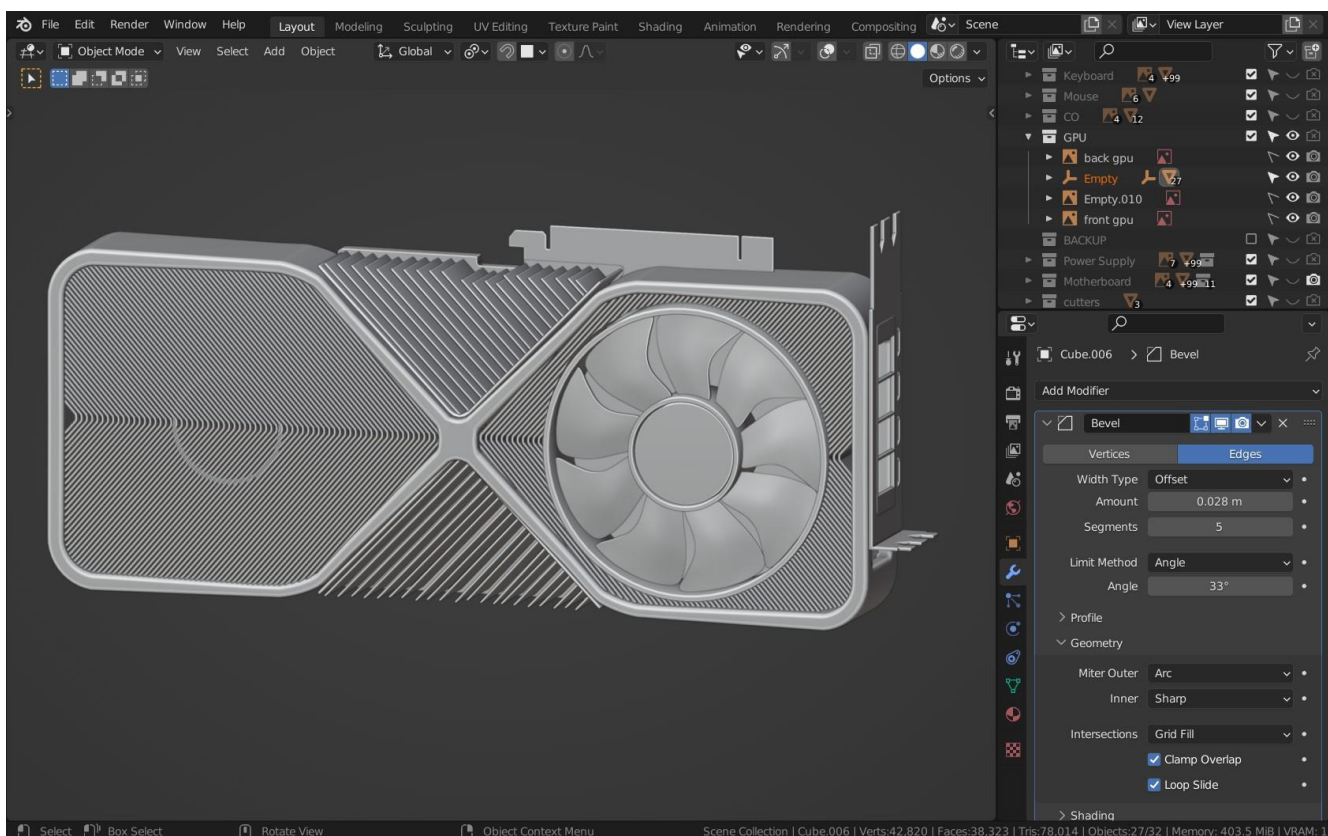


Рис.2.20. Розташування вентиляторів з обох сторін та декілька деталей

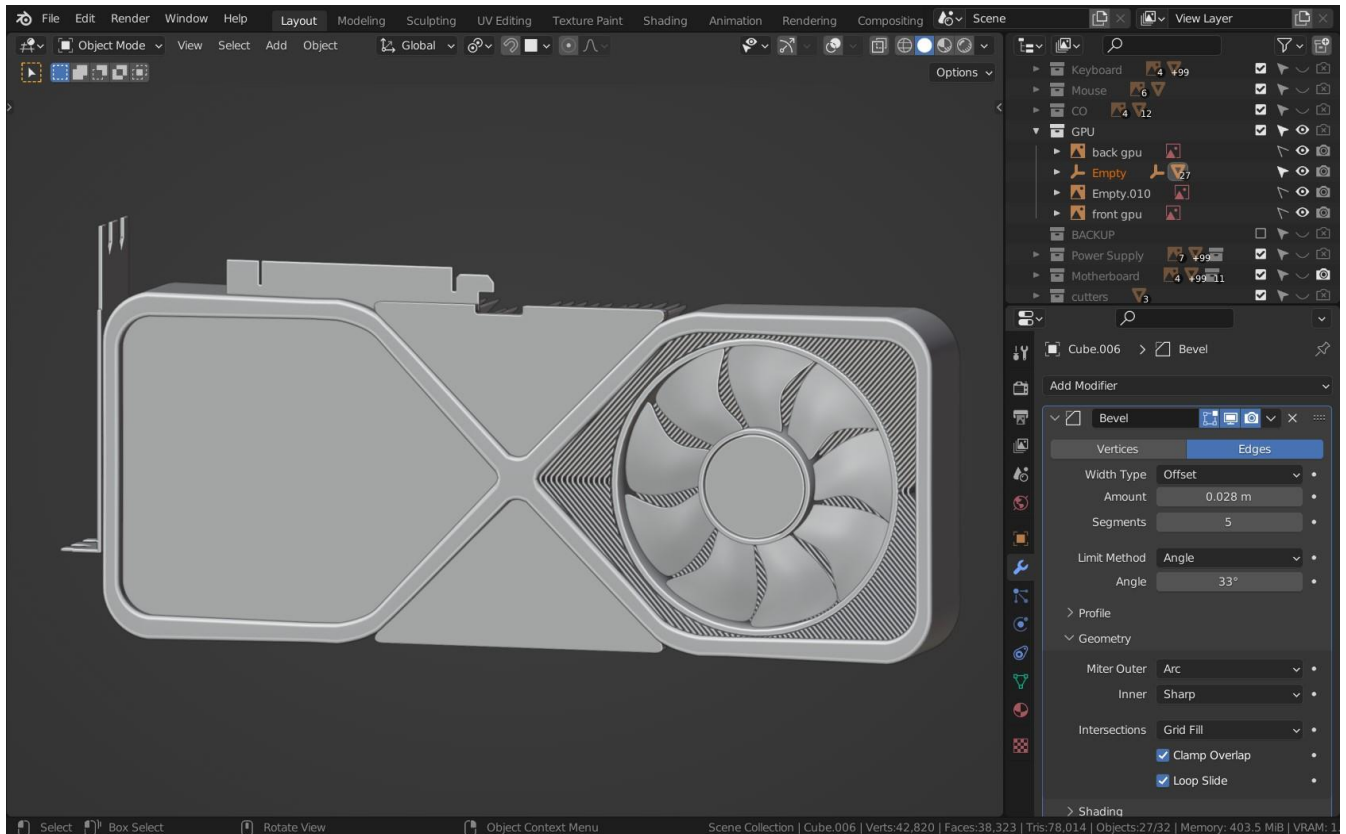


Рис.2.21. Завершуємо моделювання, додаючи останні частини

Тепер почнемо роботу з матеріалами для відеокарти. Використовується PBR текстурування, як і в прикладі з процесором. Переходимо у вкладку Shading та вибираємо елемент якому будемо додавати матеріал. Для додання надпису будемо використовувати нод «Image texturing» та нод «Bump map» для створення ілюзії об'єму. Завдяки ноду «Color ramp» який відповідає за частоту білого та чорного кольору, налаштуємо сприятливий матеріал до одного з елементів відеокарти.

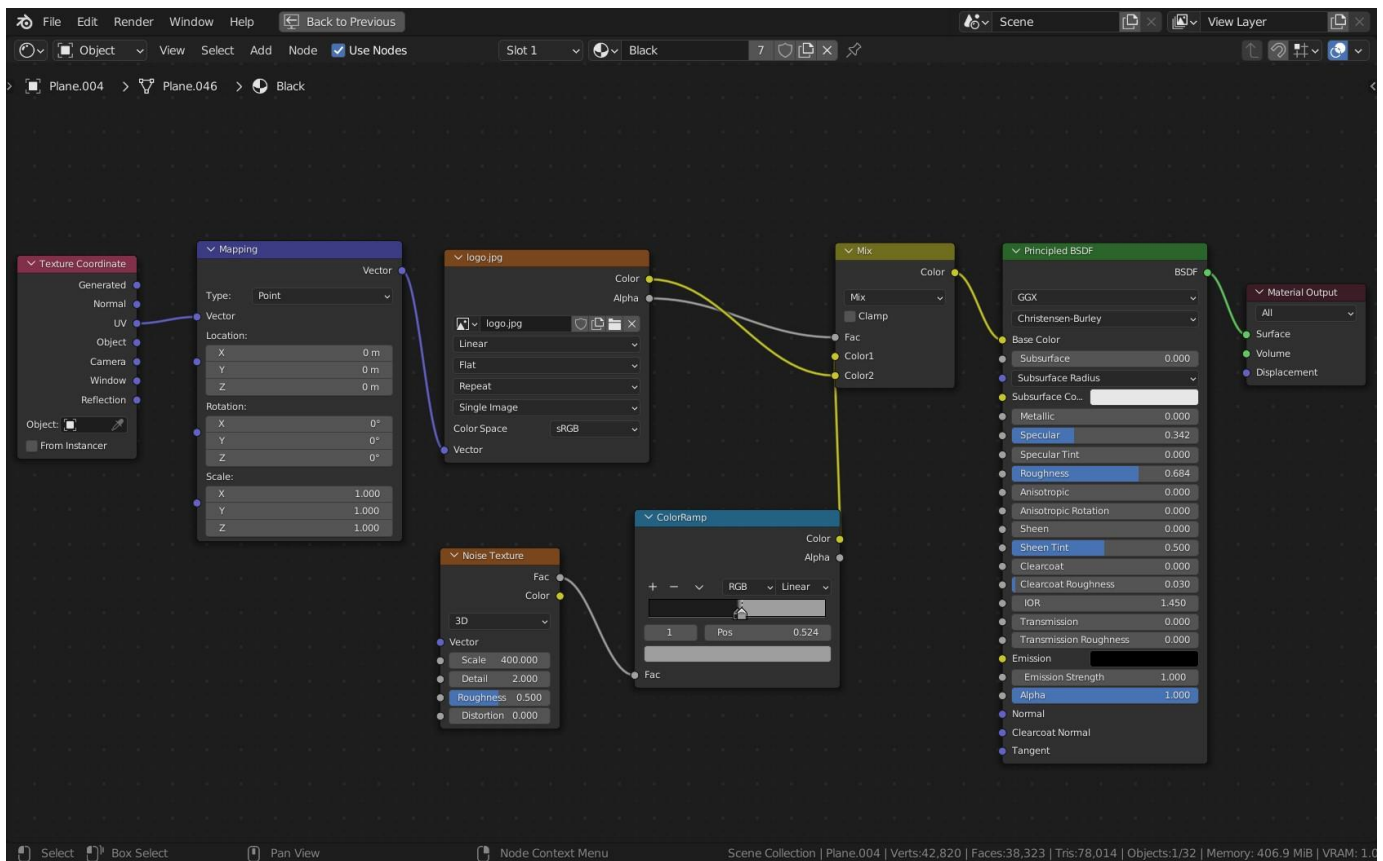


Рис.2.22. Створення матеріалу

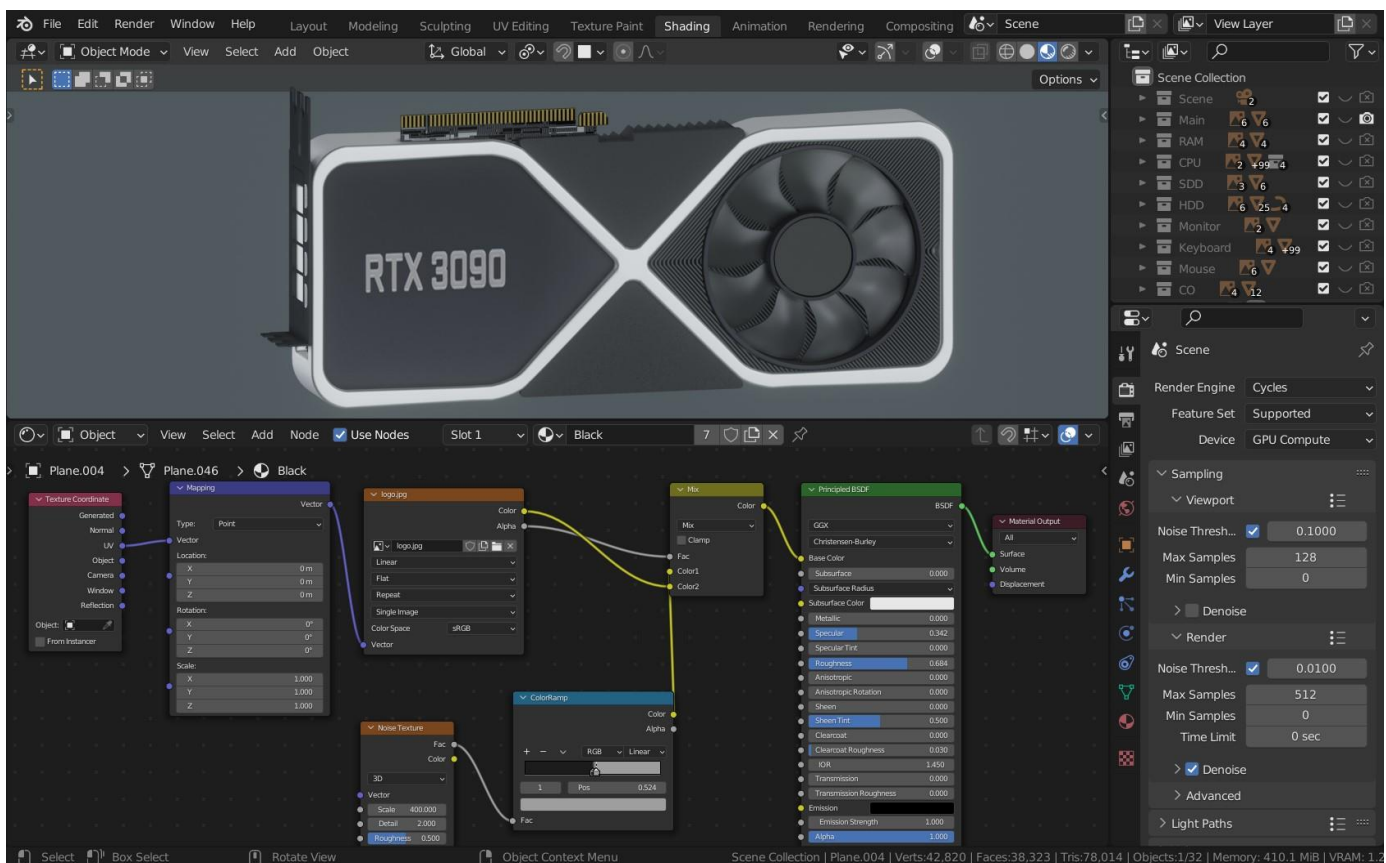


Рис.2.23. Робочий простір налаштування матеріалів

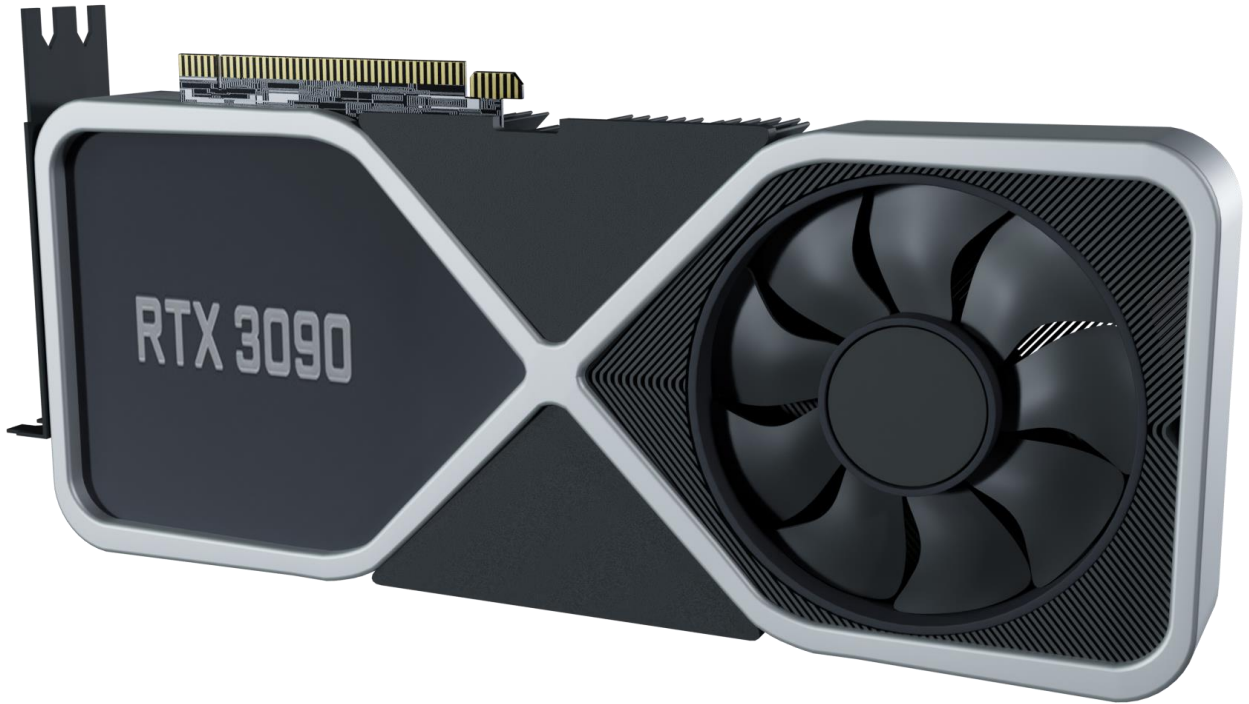


Рис.2.24. Фінальний рендер відеокарти

2.5.3. Моделювання SSD

Далі за планом SSD. По складності моделювання на рівні процесора, тому додаємо лише один примітив «Куб» та використовуємо команду «Insert face» та «Extrude». Також додамо модифікатор «Bevel» та «WeightNormal» для фотореалістичного вигляду. Для того щоб зробити внутрішні частини використовується subdivision and solidify модифікатори.

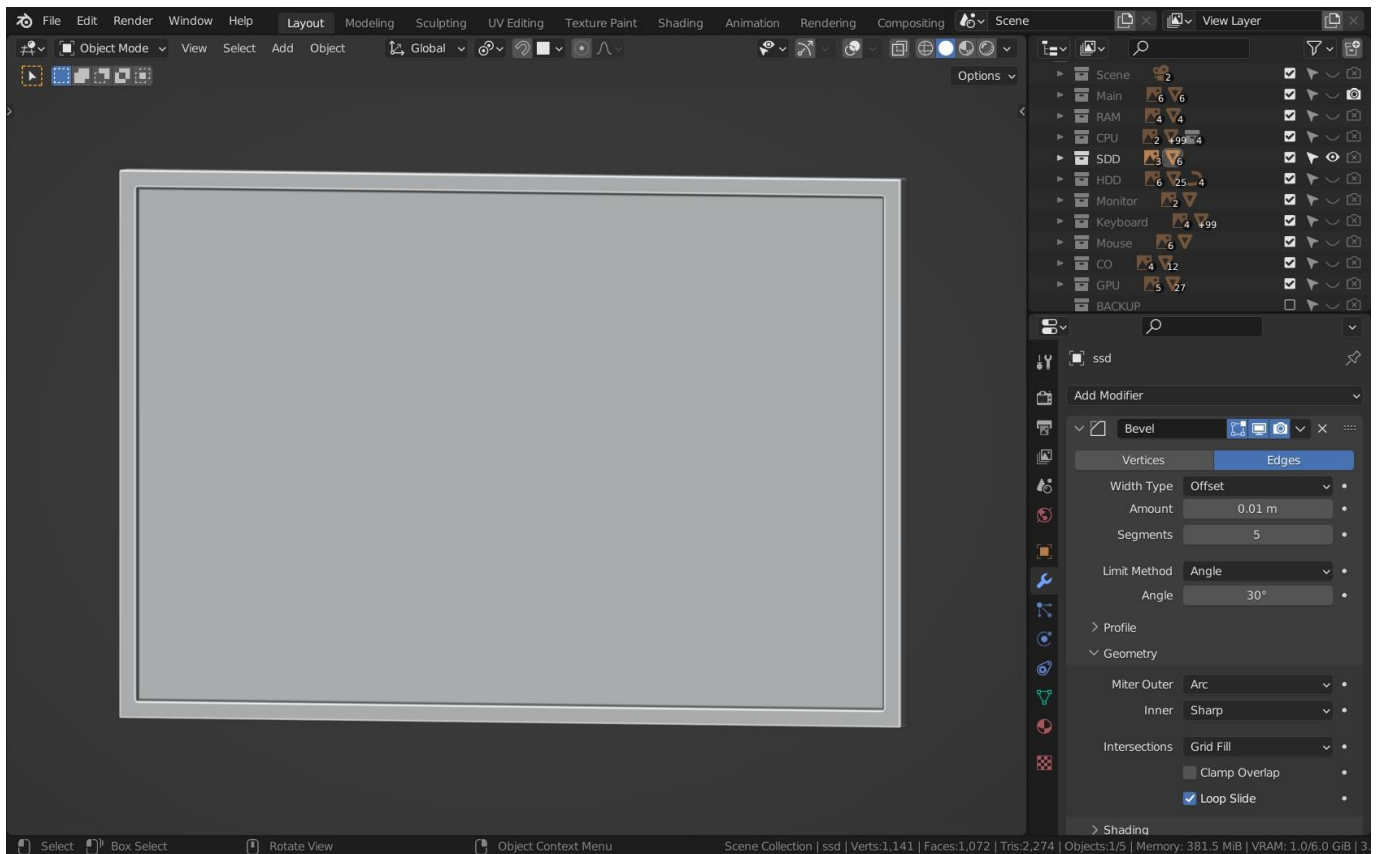


Рис.2.25. Моделювання SSD

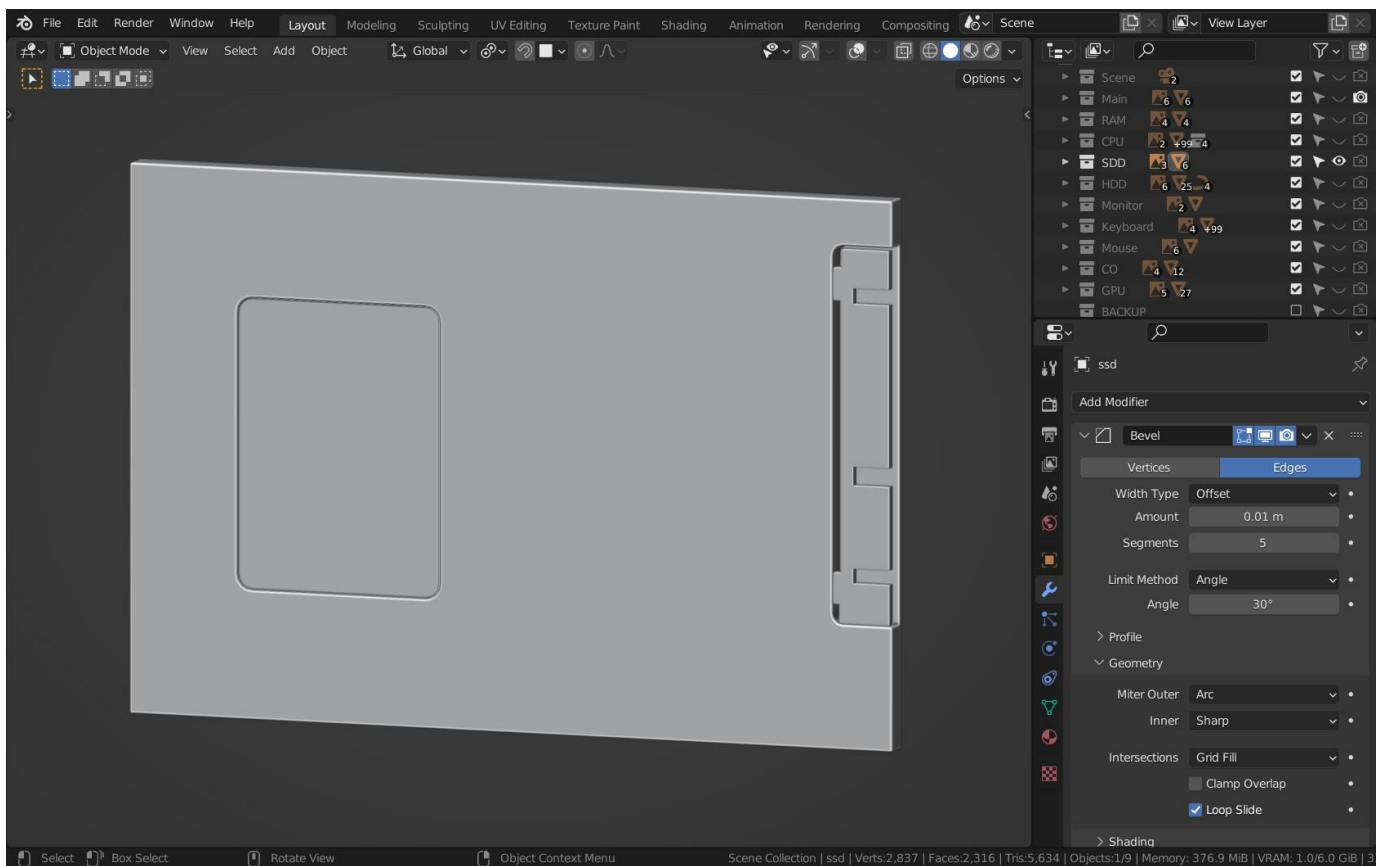


Рис.2.26 Моделювання SSD (вид сзаду)

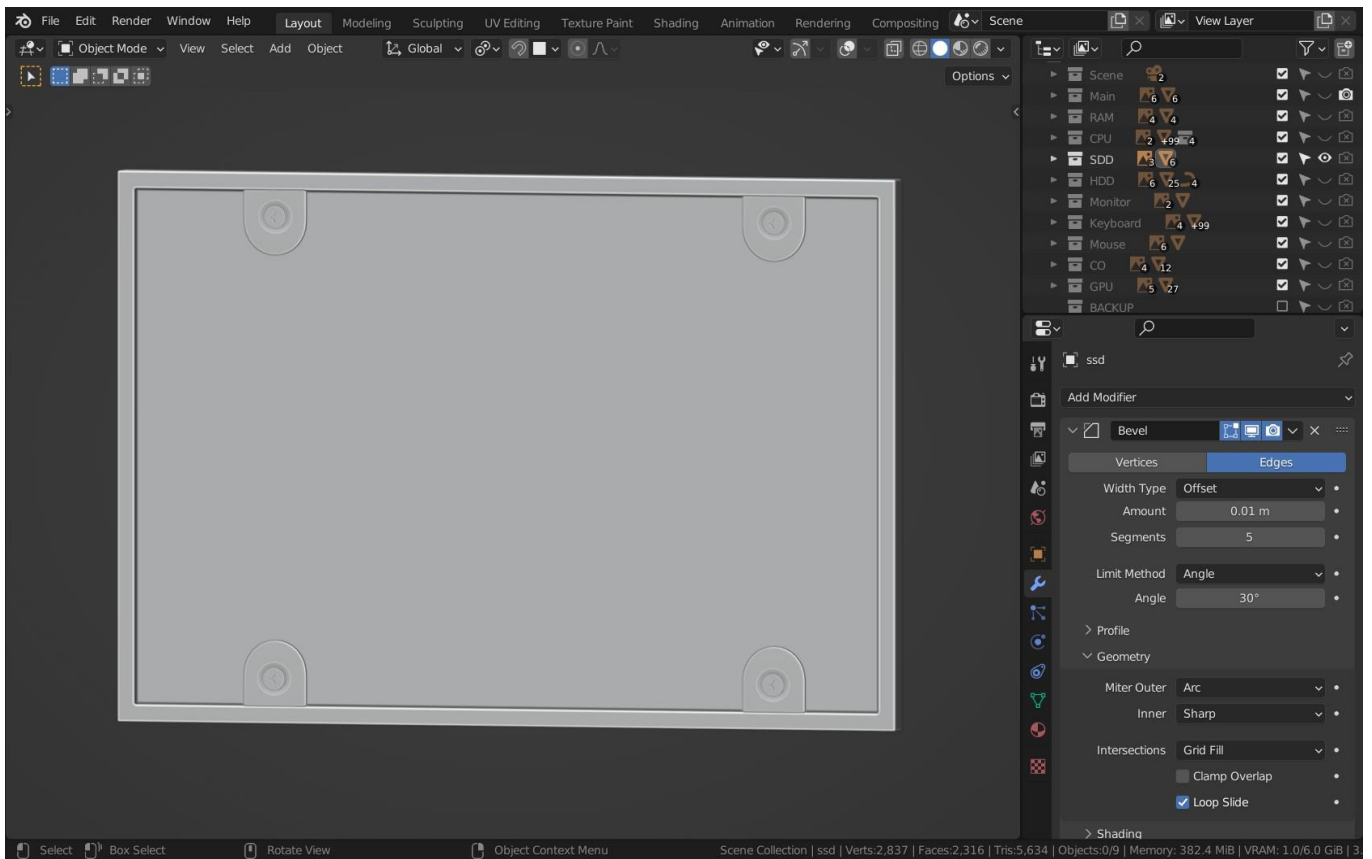


Рис.2.27 Додамо маленькі деталі

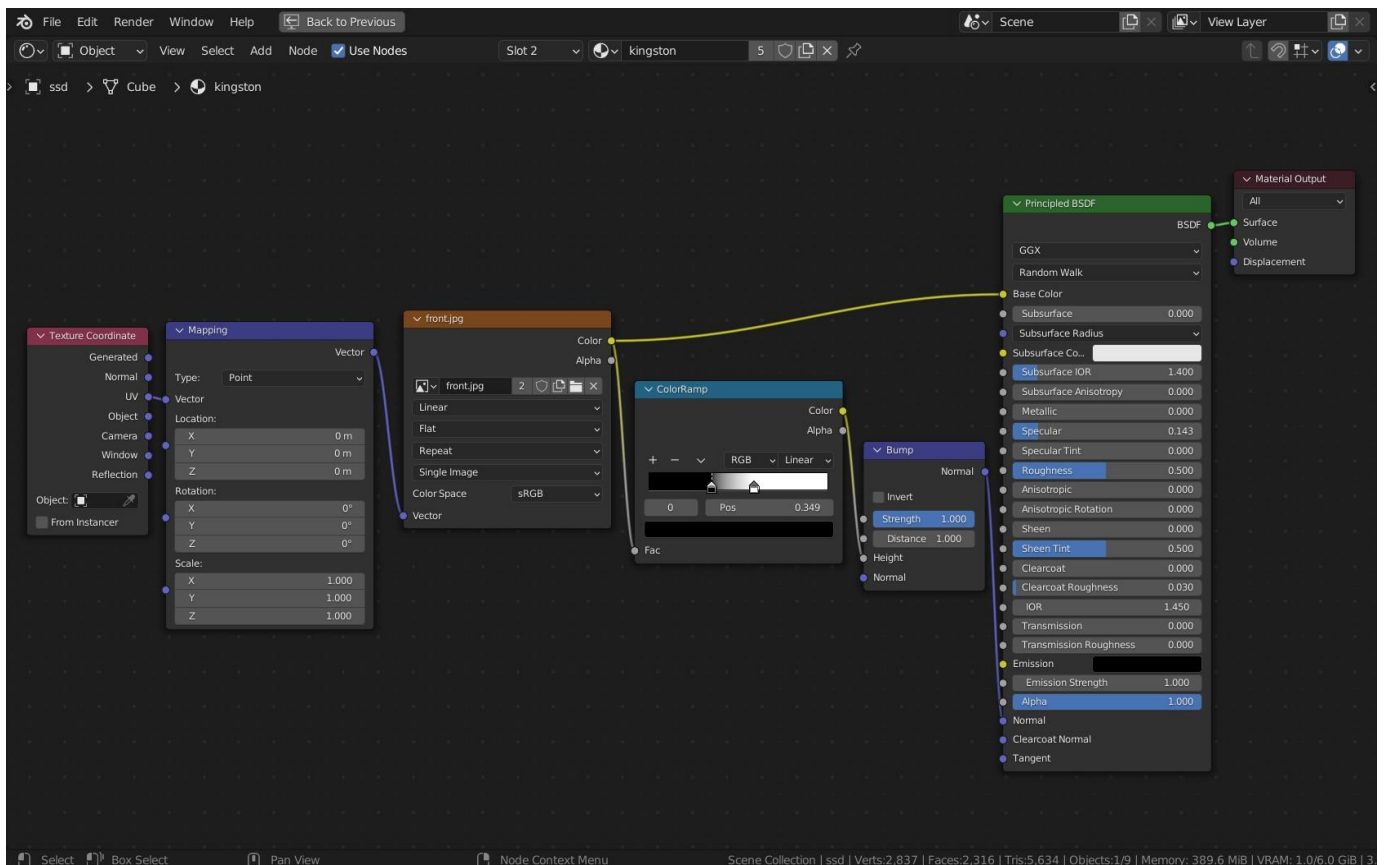


Рис.2.28 Створення матеріалу для SSD



Рис.2.29 Фінальний рендер SSD

2.5.4. Моделювання HDD

Перейдемо до HDD. Нічого складного тут також нема. Для моделювання HDD будуть використовуватись тіж самі операції, як і при створенні SSD. Тож щоб не повторюватися, просто нижче буде зображено сам процес. (рис. 2.30)

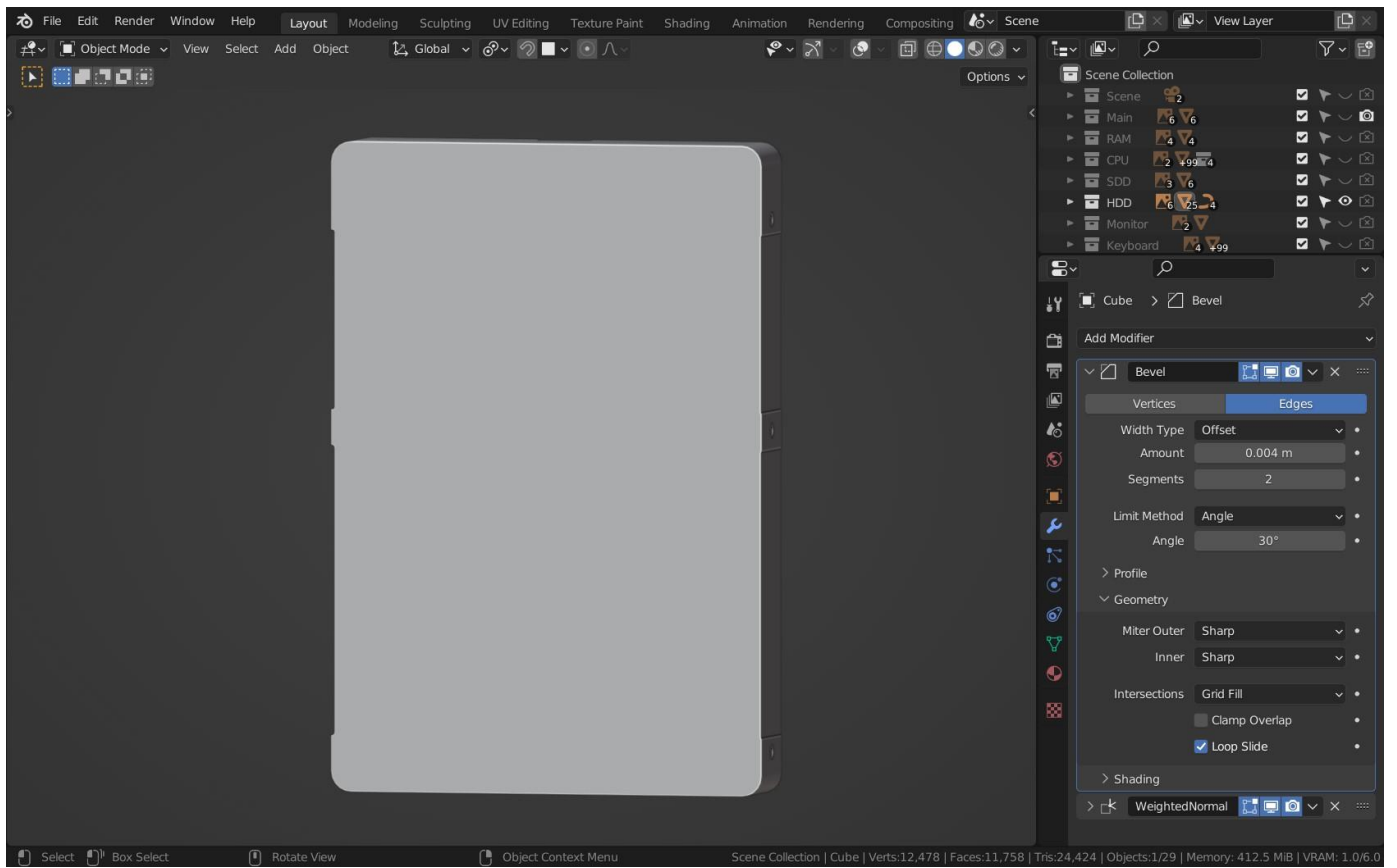


Рис.2.30. Додання модифікатору Bevel до примитива «Куб»

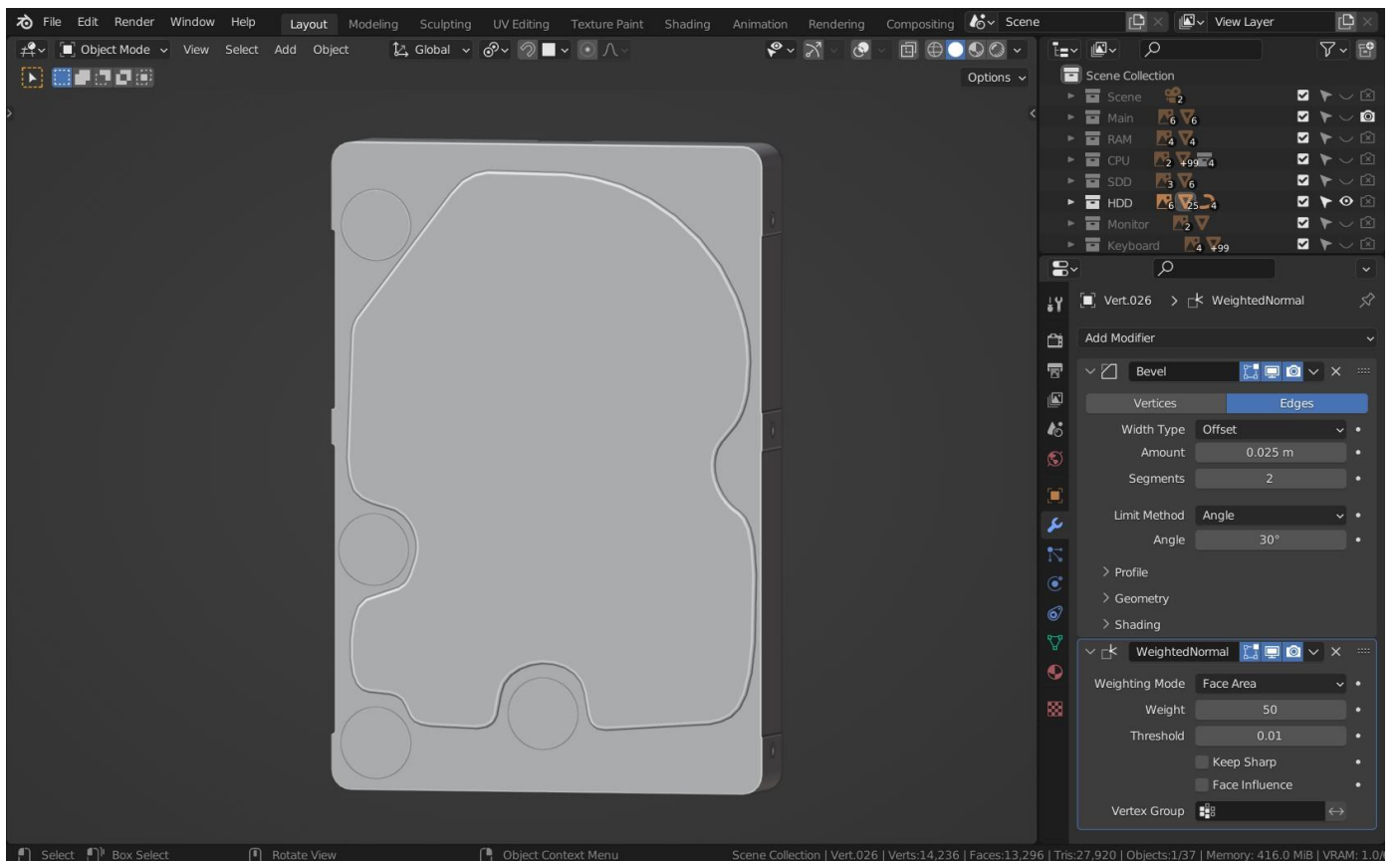


Рис.2.31. Моделювання інших форм

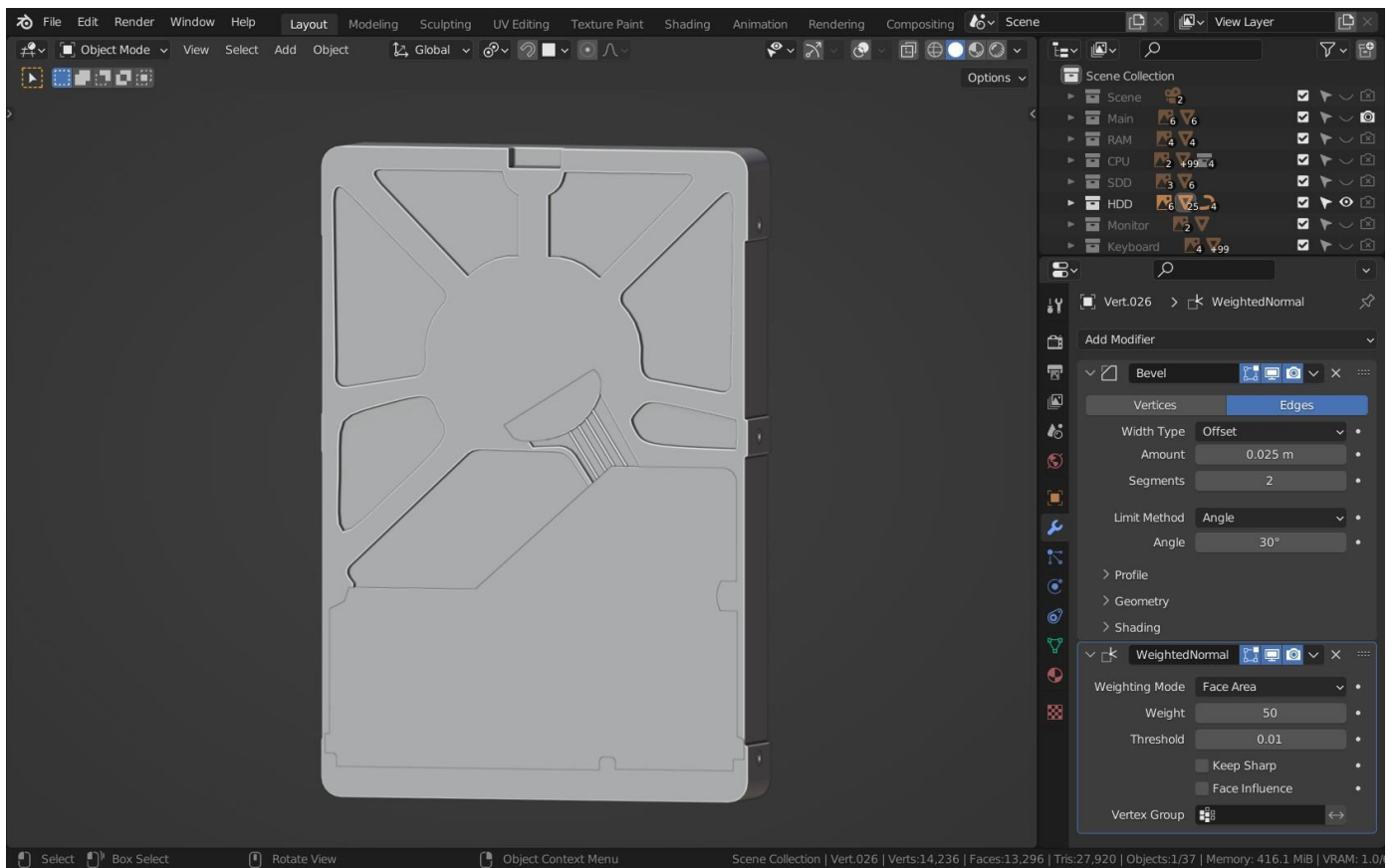


Рис.2.32. Додання деталей зсуду. Використовування модифікатора Boolean

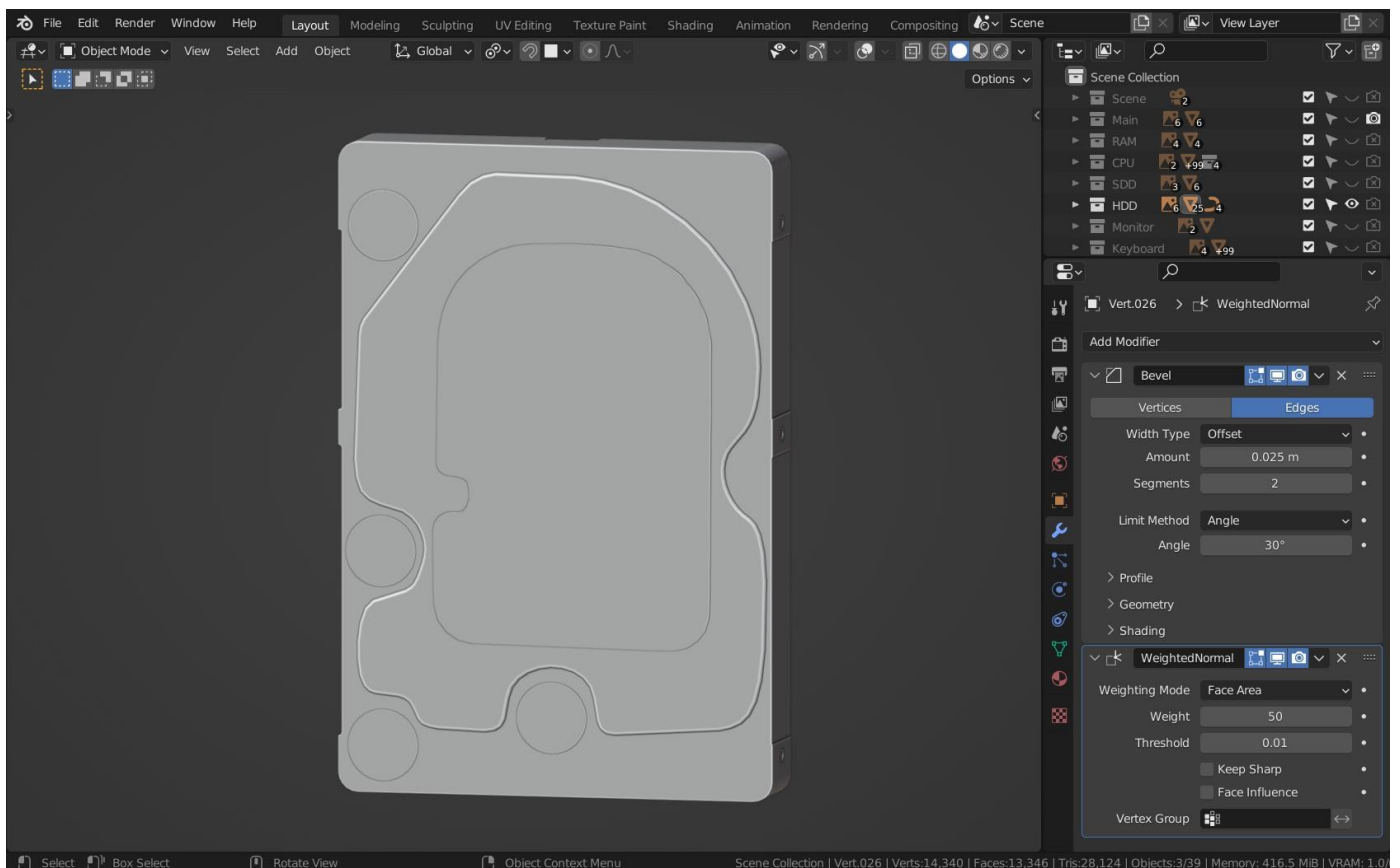


Рис.2.33. Моделювання останніх деталей спереду

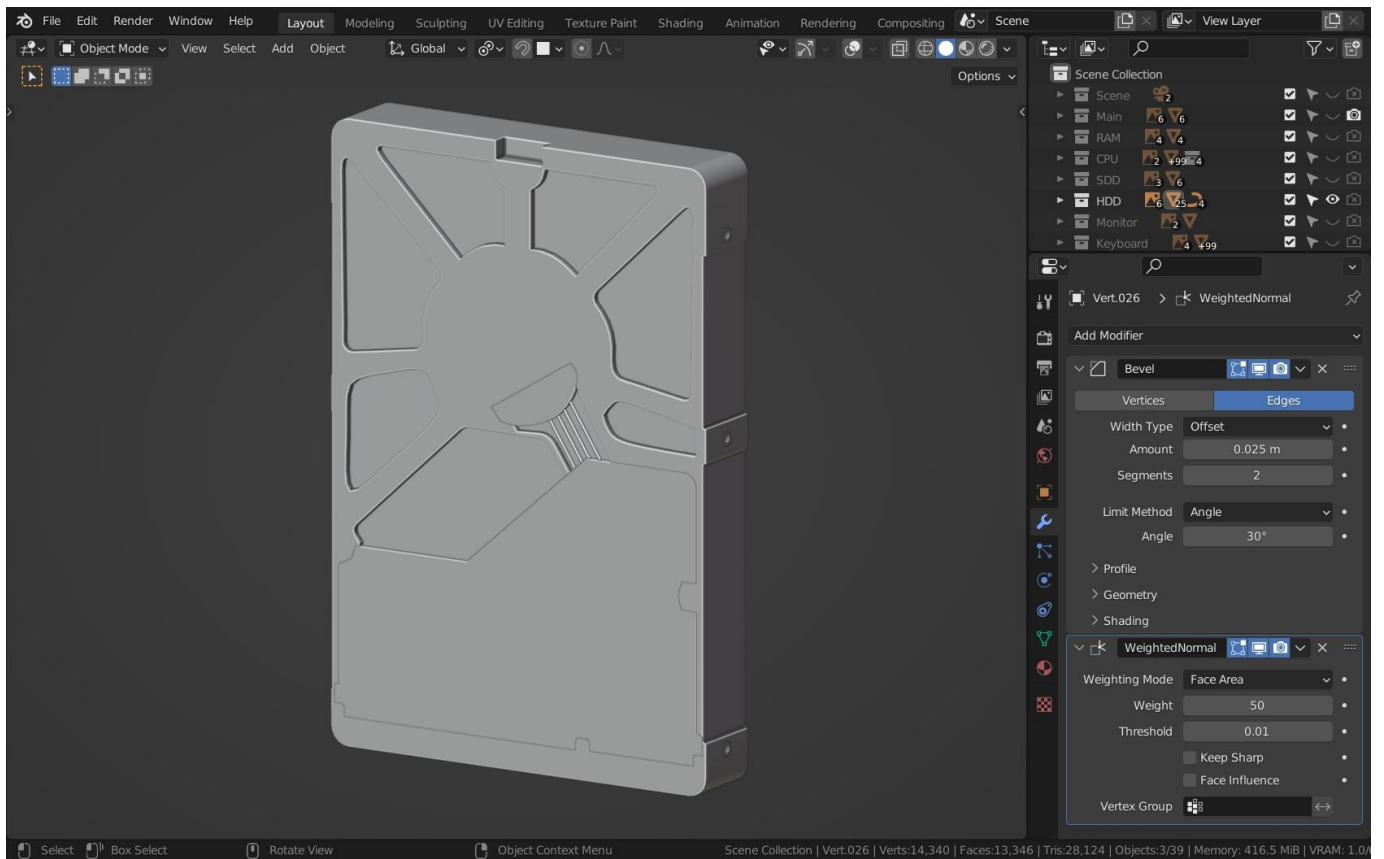


Рис.2.34. Моделювання останніх деталей сзаду

Приступаємо до стадії текстурування. Використовується знов ж таки PBR технології.

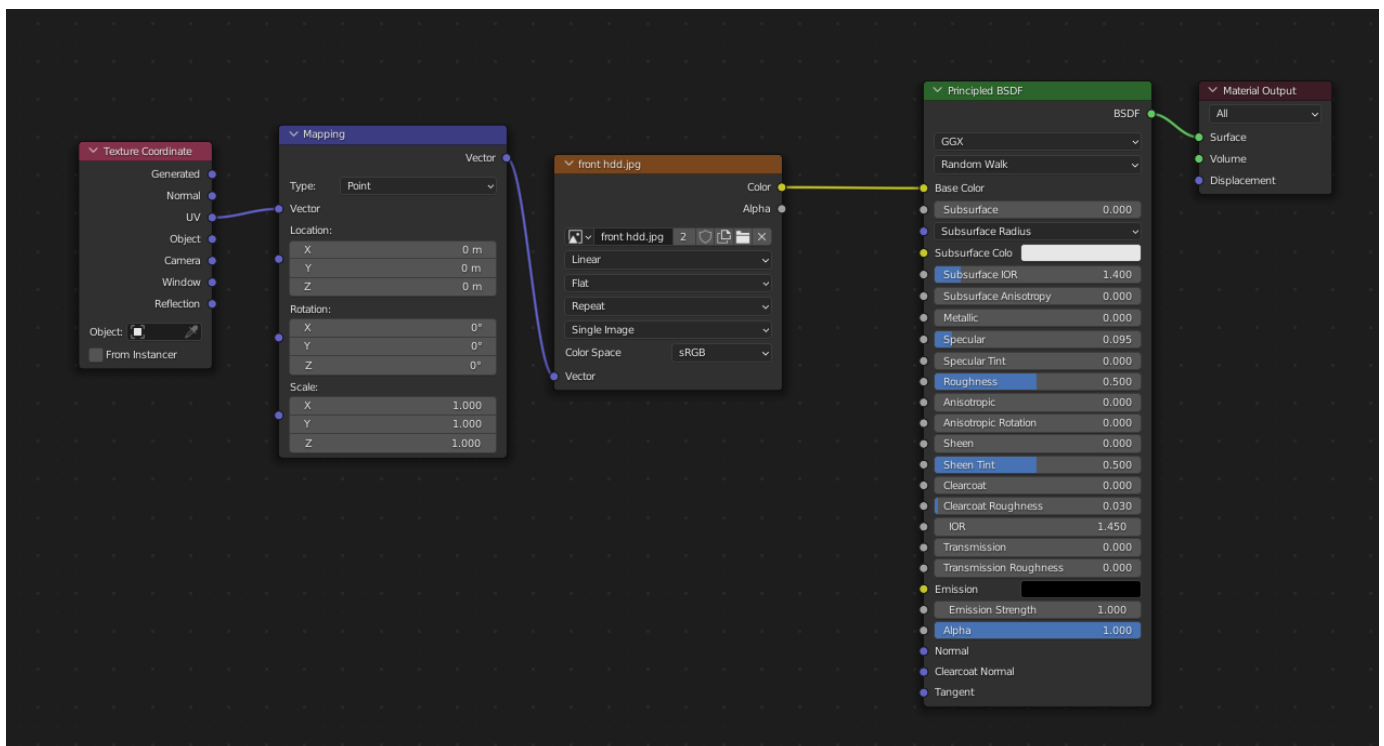


Рис.2.35. Створення матеріалу емблеми WD

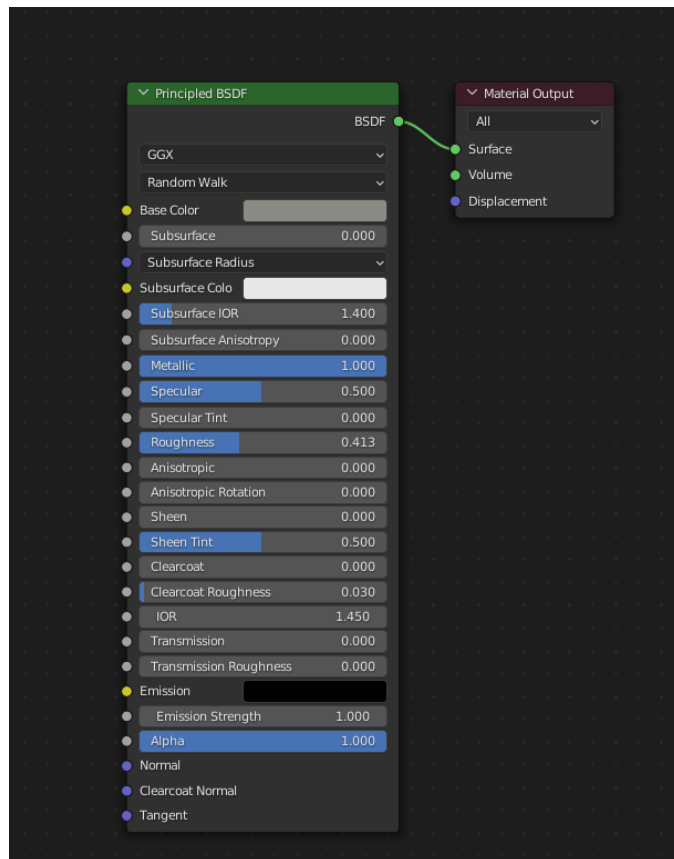


Рис.2.36. Створення металевої частини



Рис.2.37. Фінальний рендер HDD

2.5.5. Моделювання оперативної пам'яті

Настав час моделювання одного з найважливішого компонента ПК – оперативна пам'ять. Почнемо як за звичай з примитиву «Куб» та використовуючи модифікатор Boolean зробимо отвори по обох боках. Для спрощення даної процедури використовується аддон Bool tool, який йде з блендером за замовчуванням. Для активації аддона потрібно перейти у «Preferences» у вкладці «Addons» ввести «Bool Tool» та підтвердити активацію аддона. У вкладці «Tools» модифікую аддон для подальшого використання. (рис. 2.38).

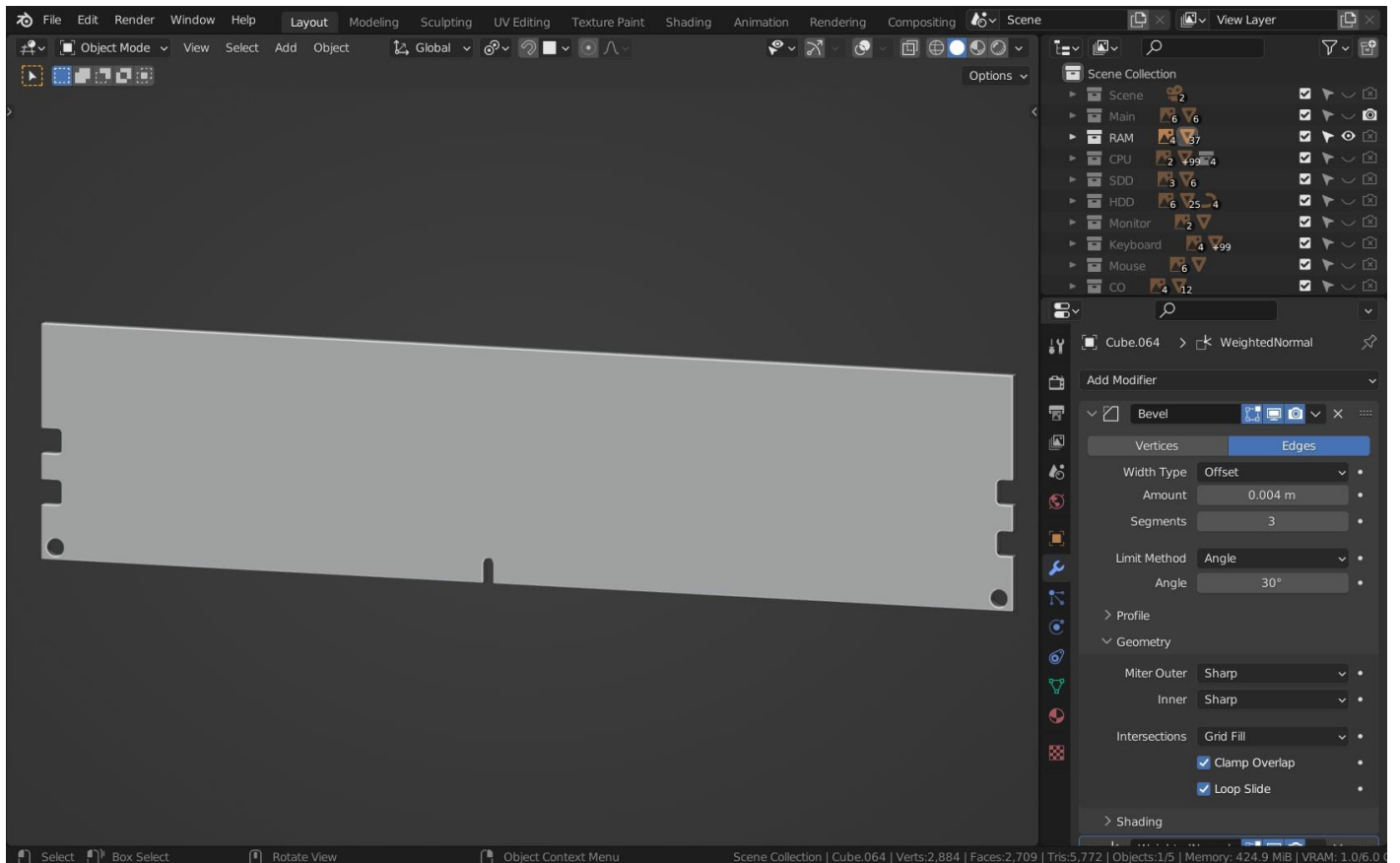


Рис.2.38. Використання Boolean до куба

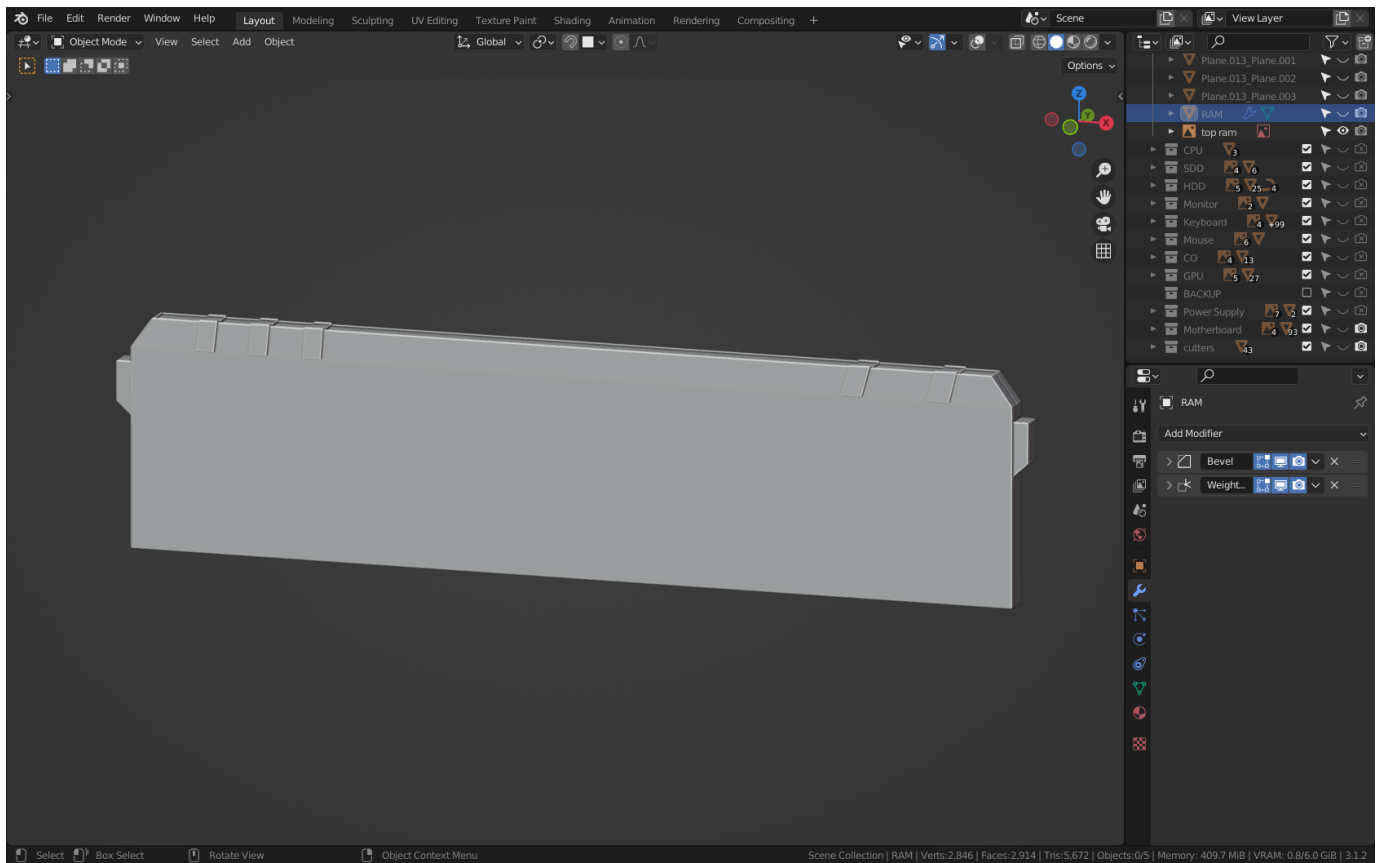


Рис.2.39. Моделювання верхньої частини оперативної пам'яті

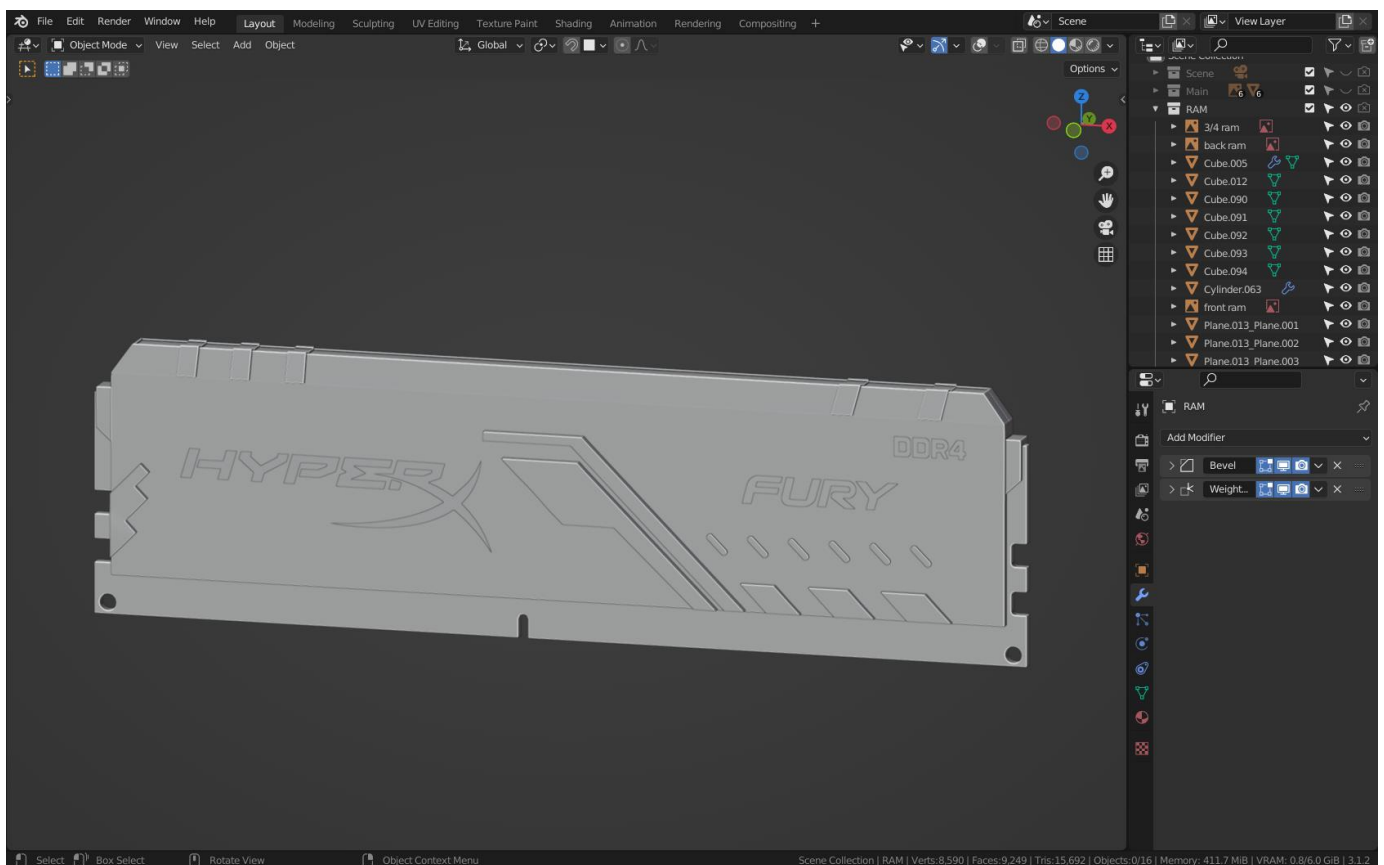


Рис.2.40. Додання надпису та іншої деталізації

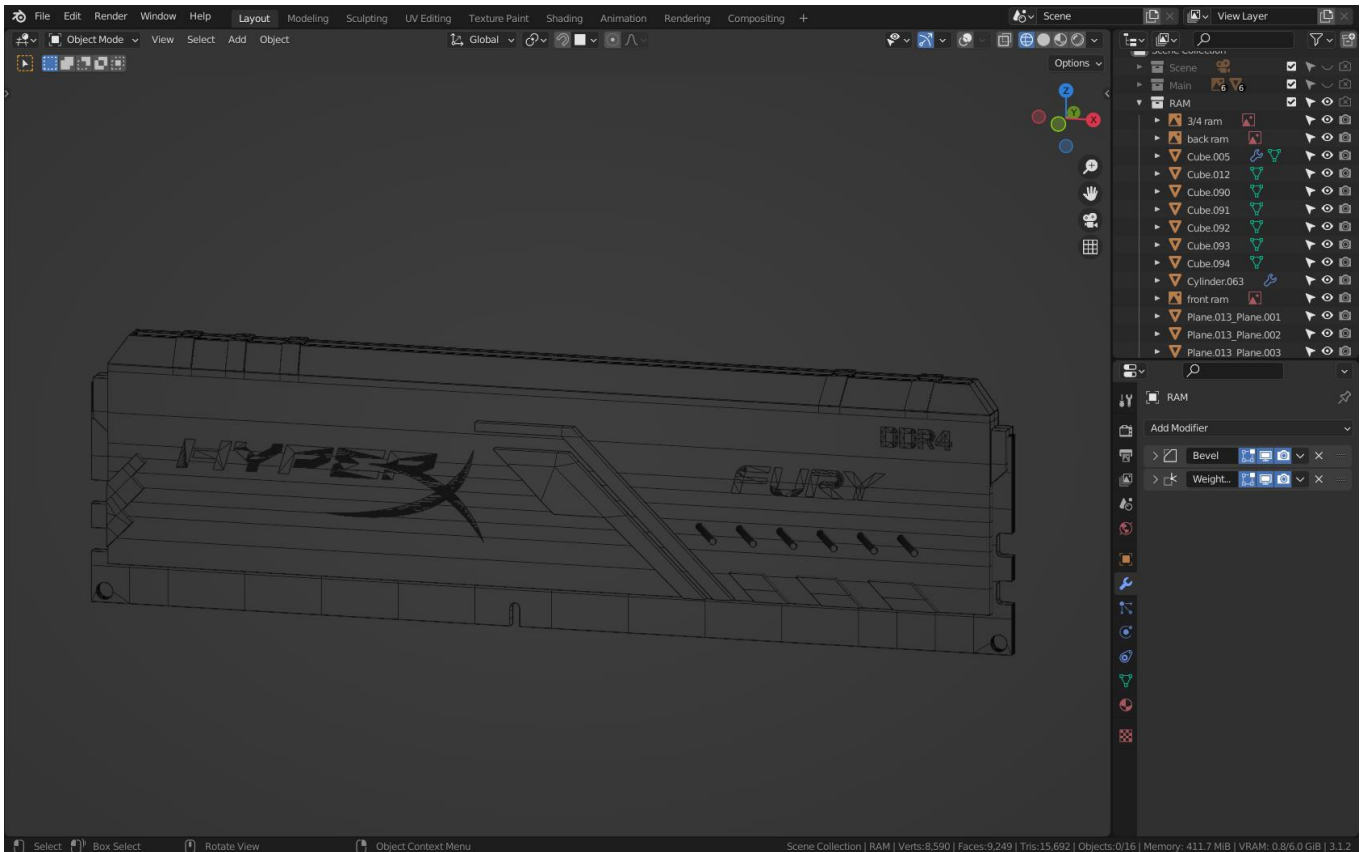


Рис.2.41. Wireframe mod оперативної пам'яті

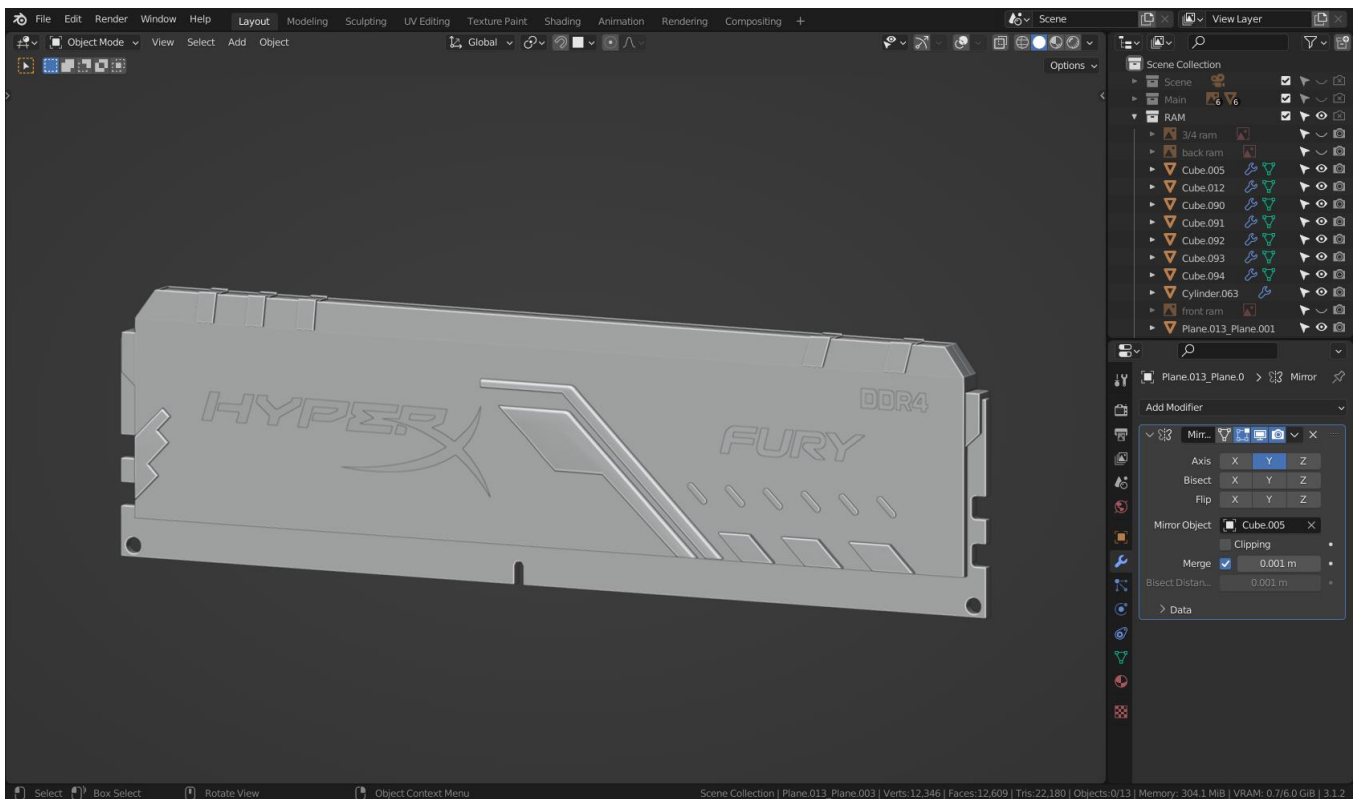


Рис.2.42. Фінальний вигляд у solid mode

На етапі текстурування все також використовуємо вище перелічені методи.

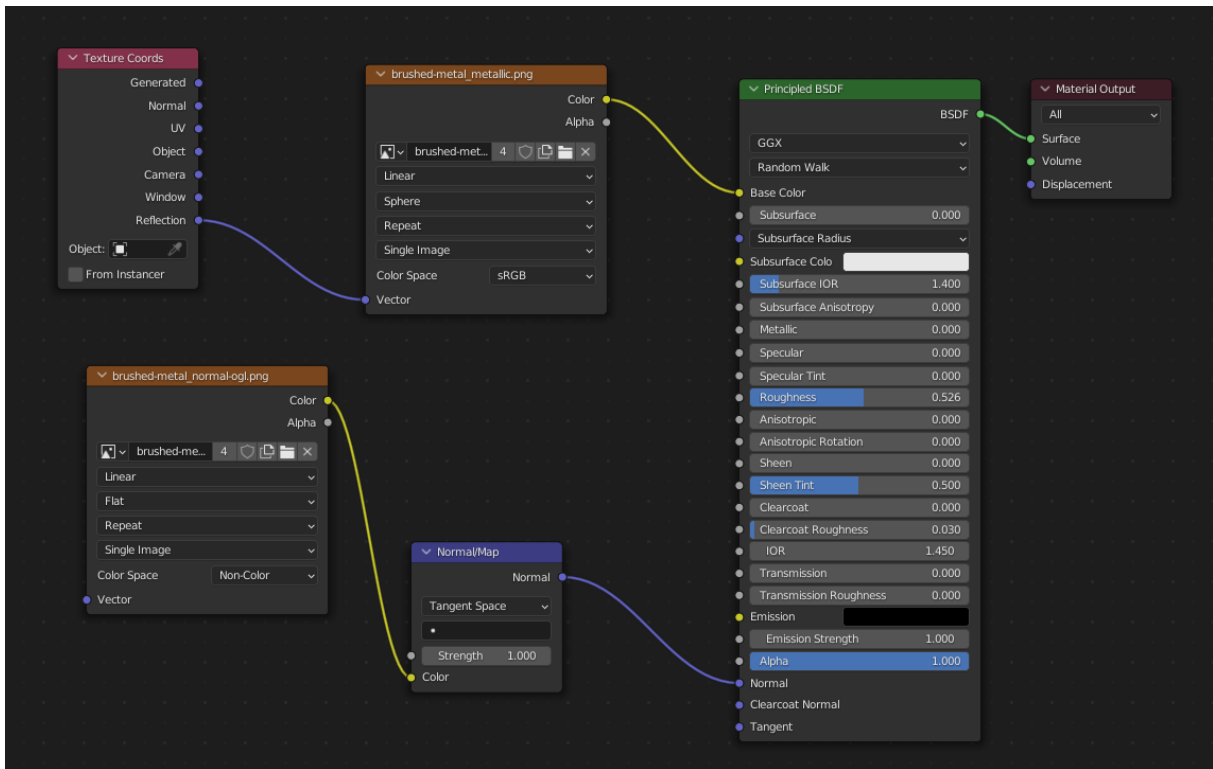


Рис.2.43. Матеріал плашки

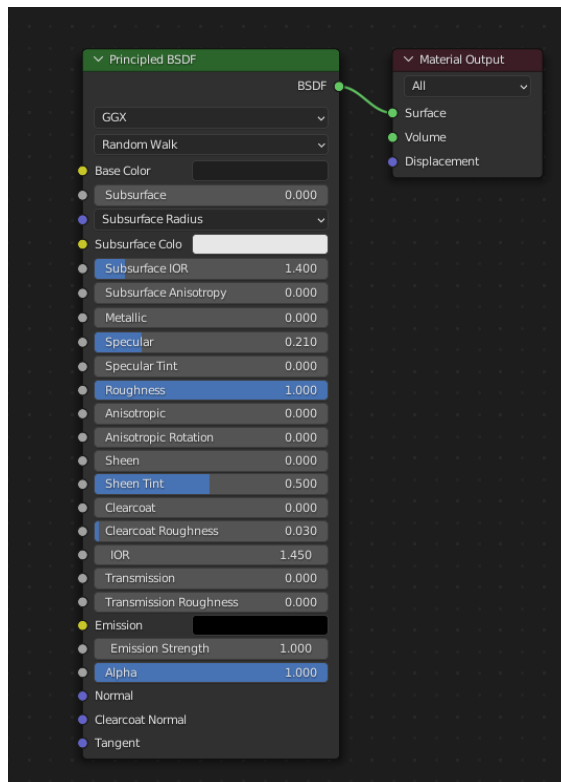


Рис.2.44. Матеріал основної частини

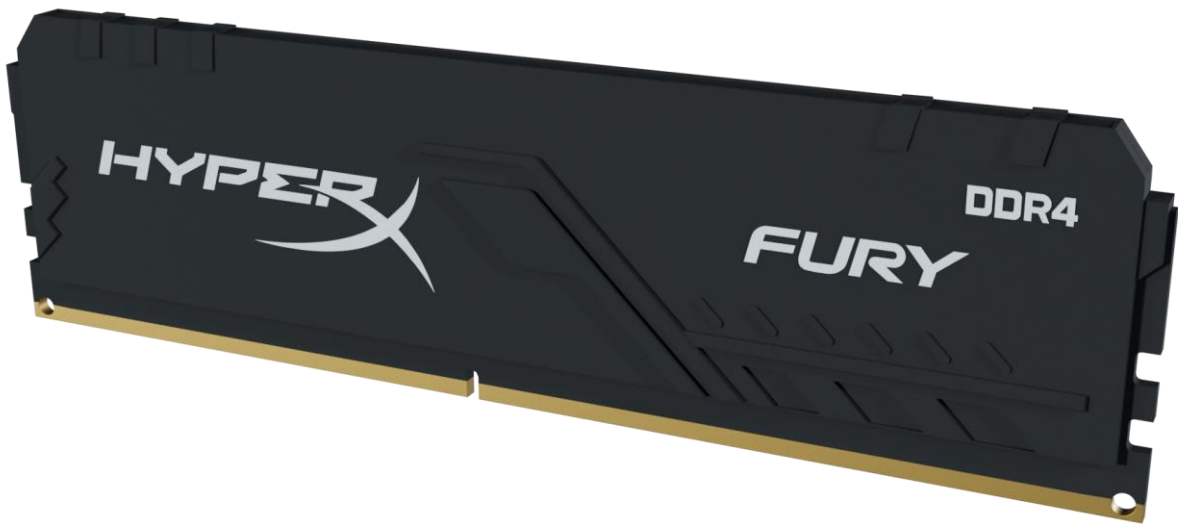


Рис.2.45. Фінальний рендер оперативної пам'яті

2.5.6. Моделювання повітряного охолодження для процесору

Для того щоб зробити повітряне охолодження для CPU використовуємо креслення. (рис. 2.46).

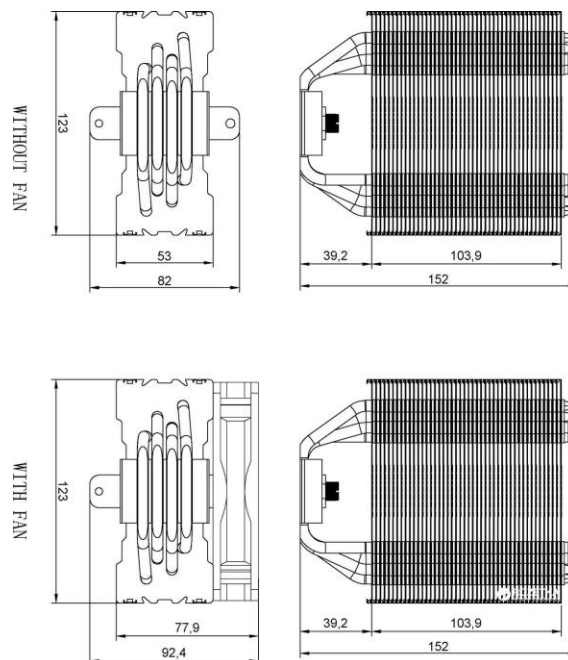


Рис.2.46. Креслення

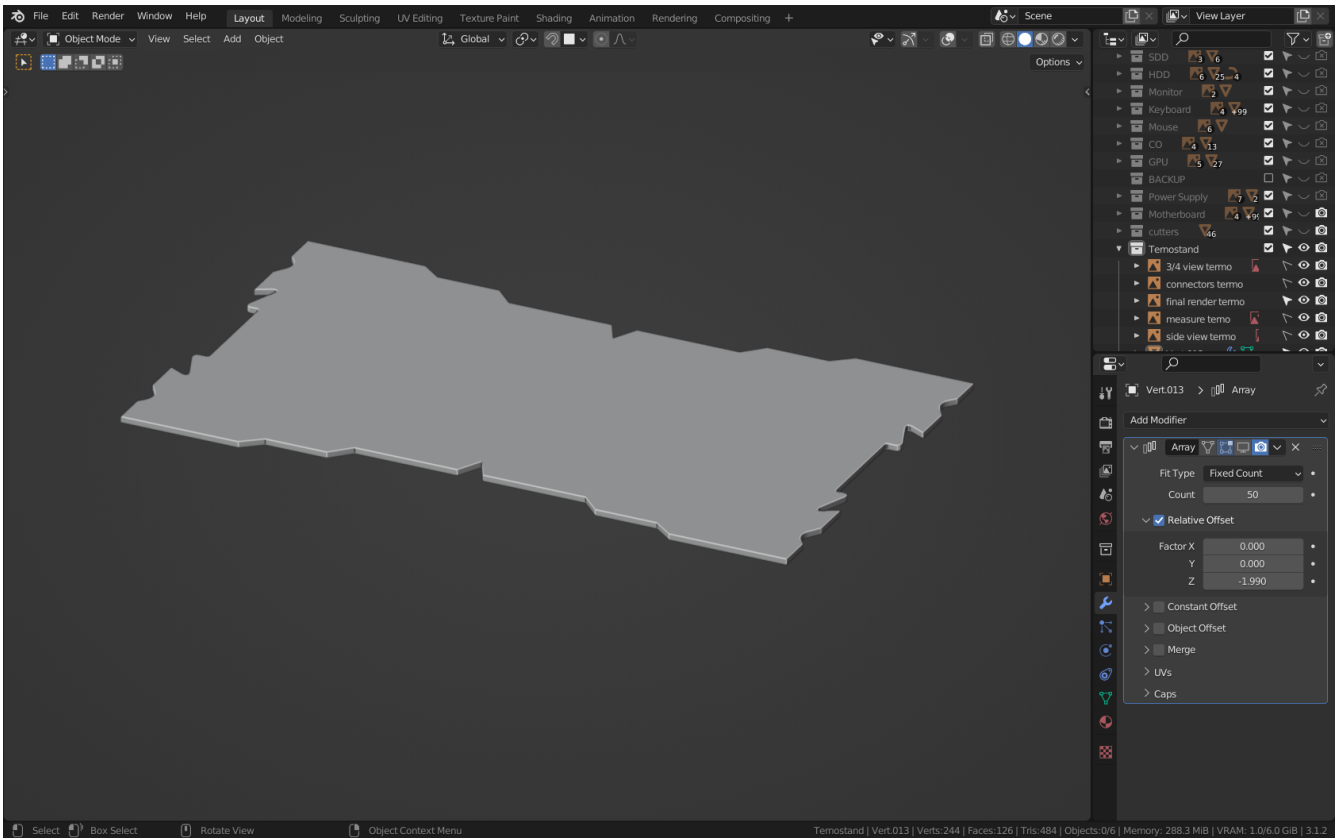


Рис.2.47. Створення форми та екструду

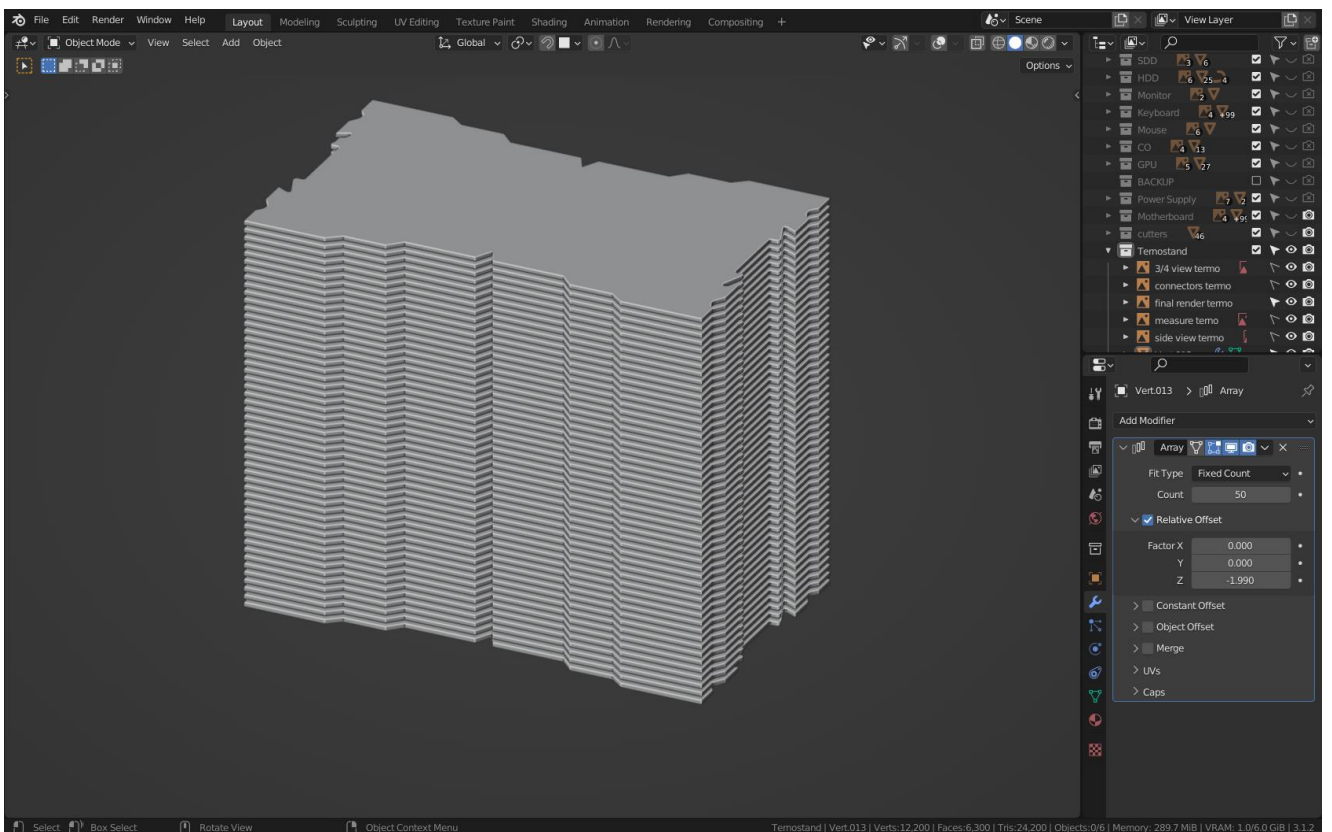


Рис.2.48. Використання модифікатора Array

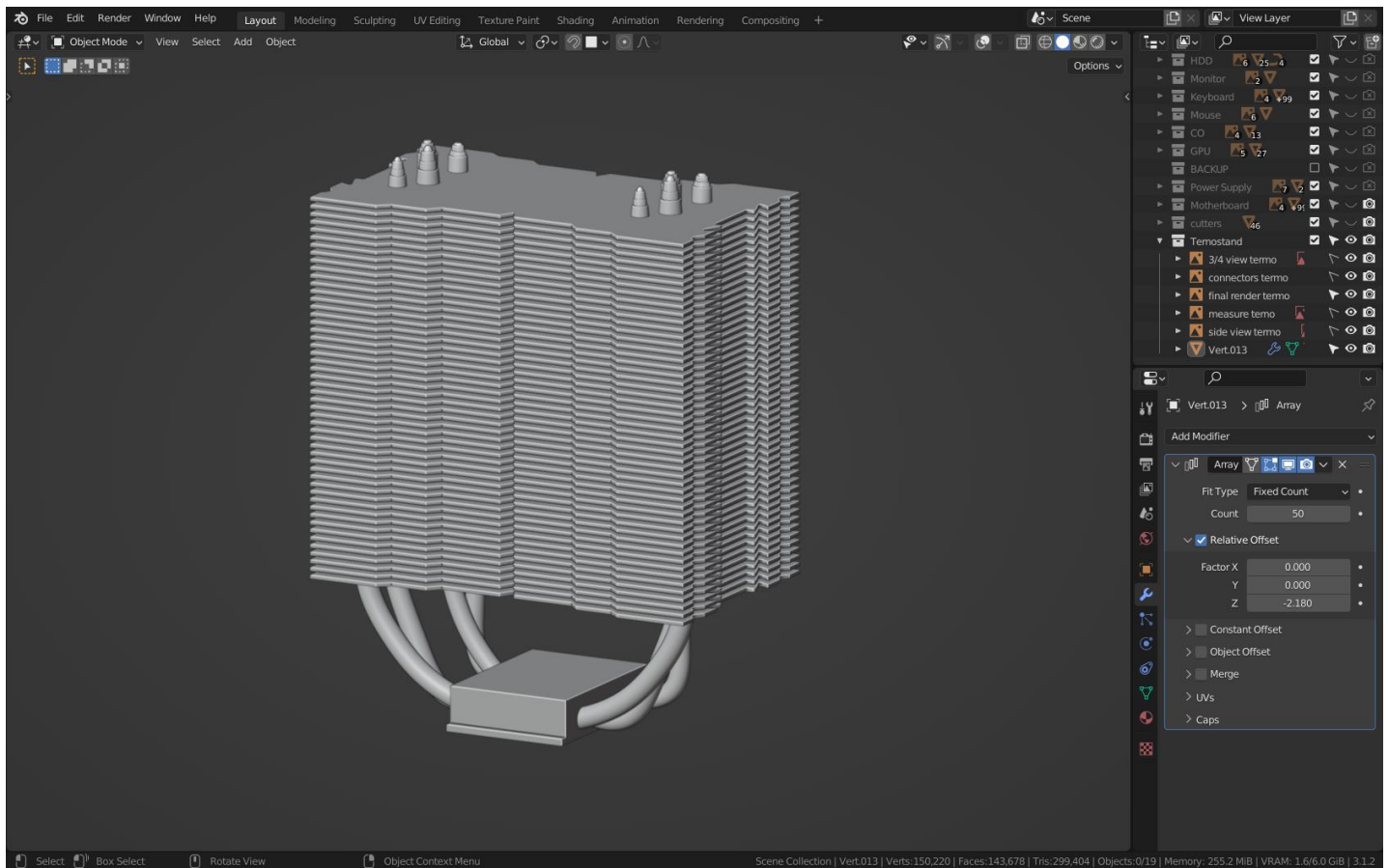


Рис.2.49. Завершення моделювання



Рис.2.50. Фінальний рендер повітряного охолодження для CPU

2.5.7. Моделювання повітряного охолодження для системного блоку

Тепер змоделюємо більш складну форму повітряного охолодження для системного блоку. В данному випадку будемо моделювати і методом Boolean, і методом Subdivision (моделювання під сабдив).

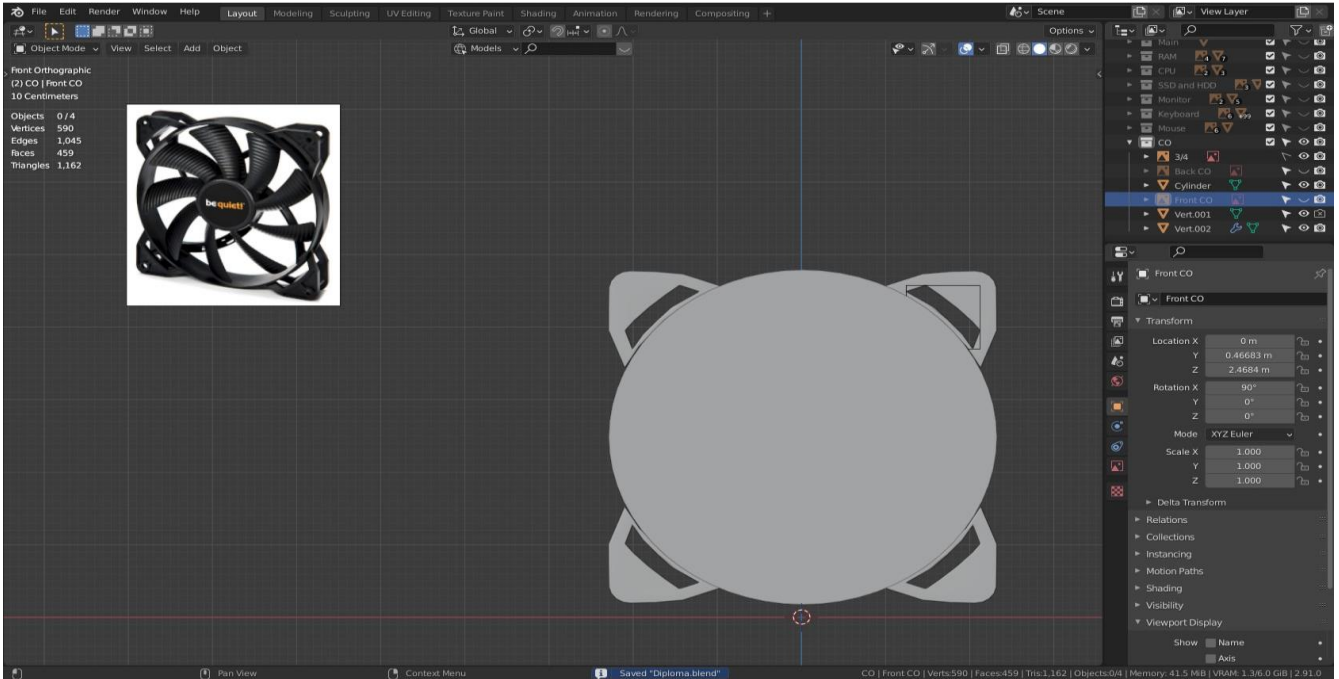


Рис.2.51. Boolean моделювання

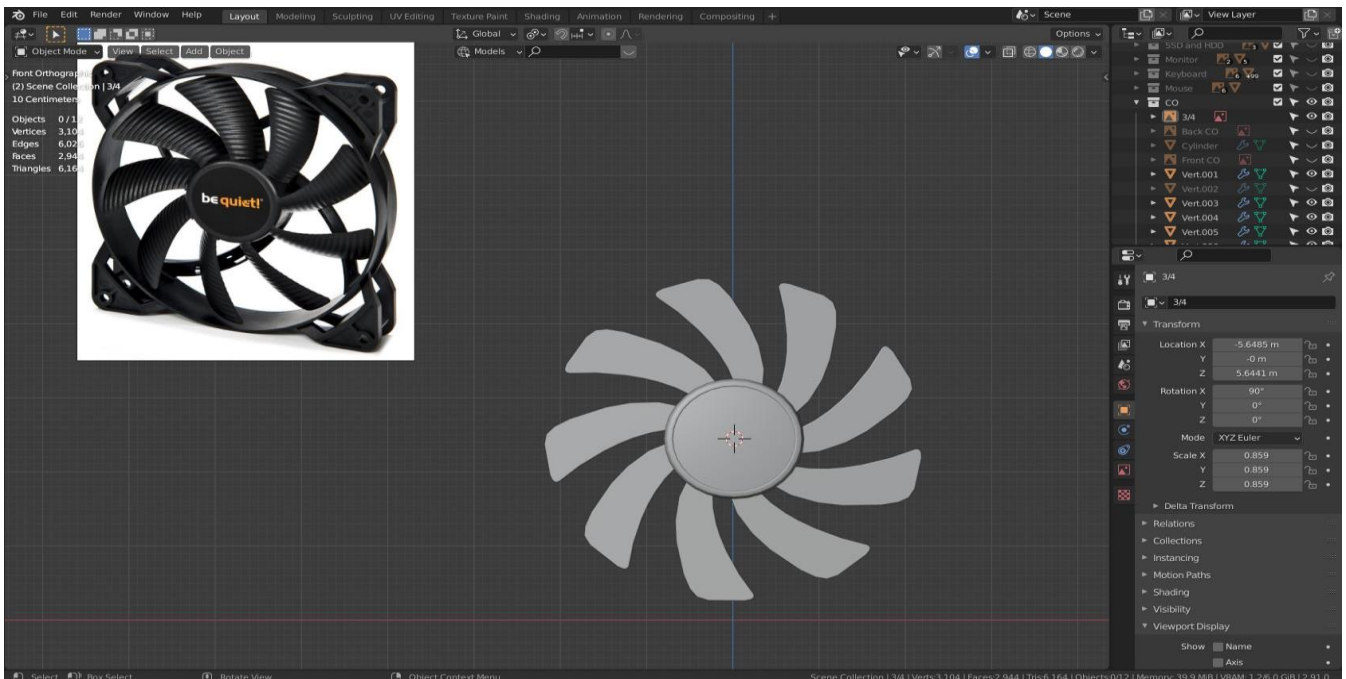


Рис.2.52. Моделювання лопаті

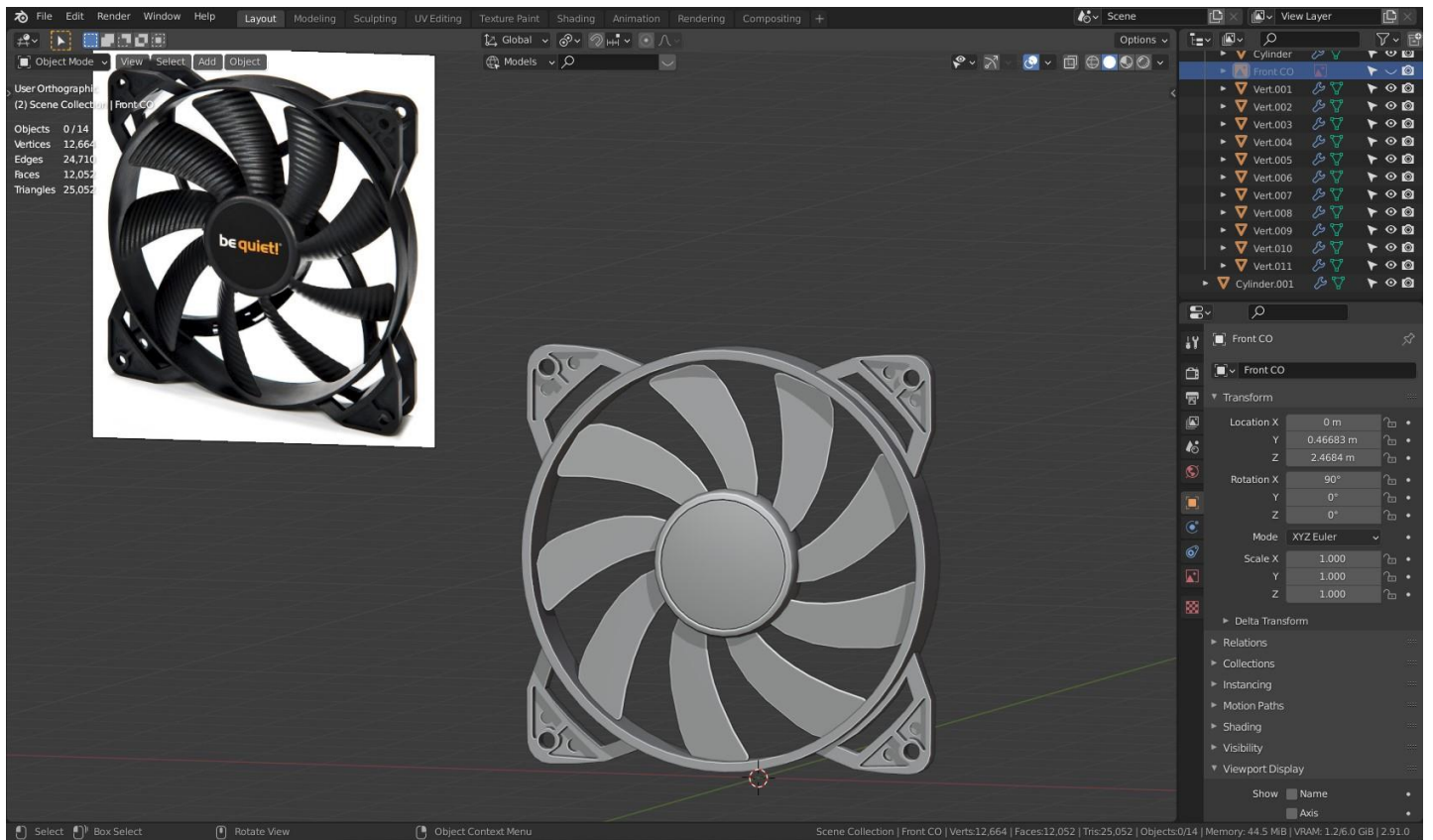


Рис.2.53. Фінальна стадія моделювання

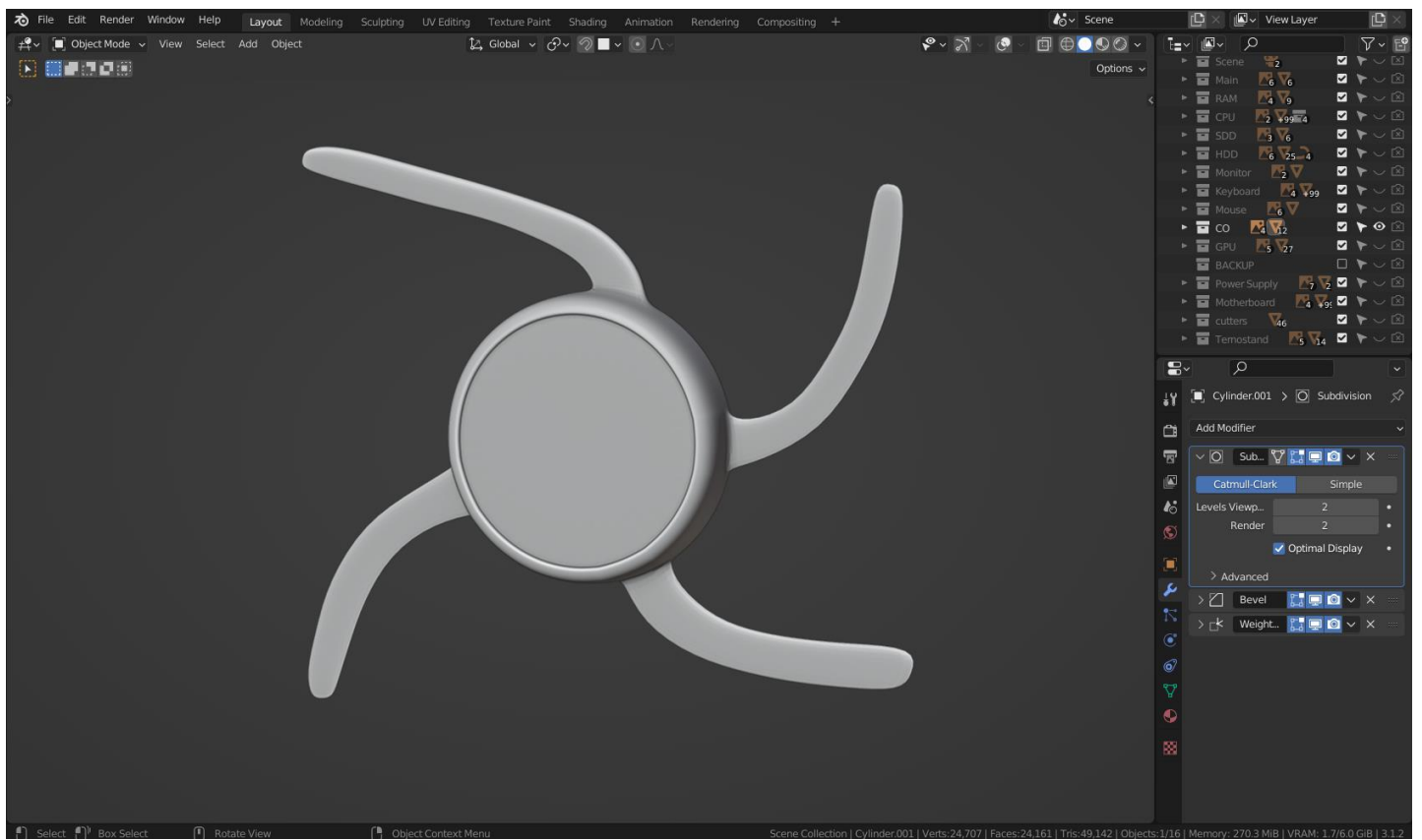


Рис.2.54. Subdivision моделювання додаткових частин



Рис.2.55. Фінальний рендер повітряного охолодження (кулера)

2.5.8. Моделювання блоку живлення

Перейдемо до моделювання блоку живлення. Принцип моделювання залишається той самий, але тут буде також використовуватися новий модифікатор Wireframe для створення сітки.

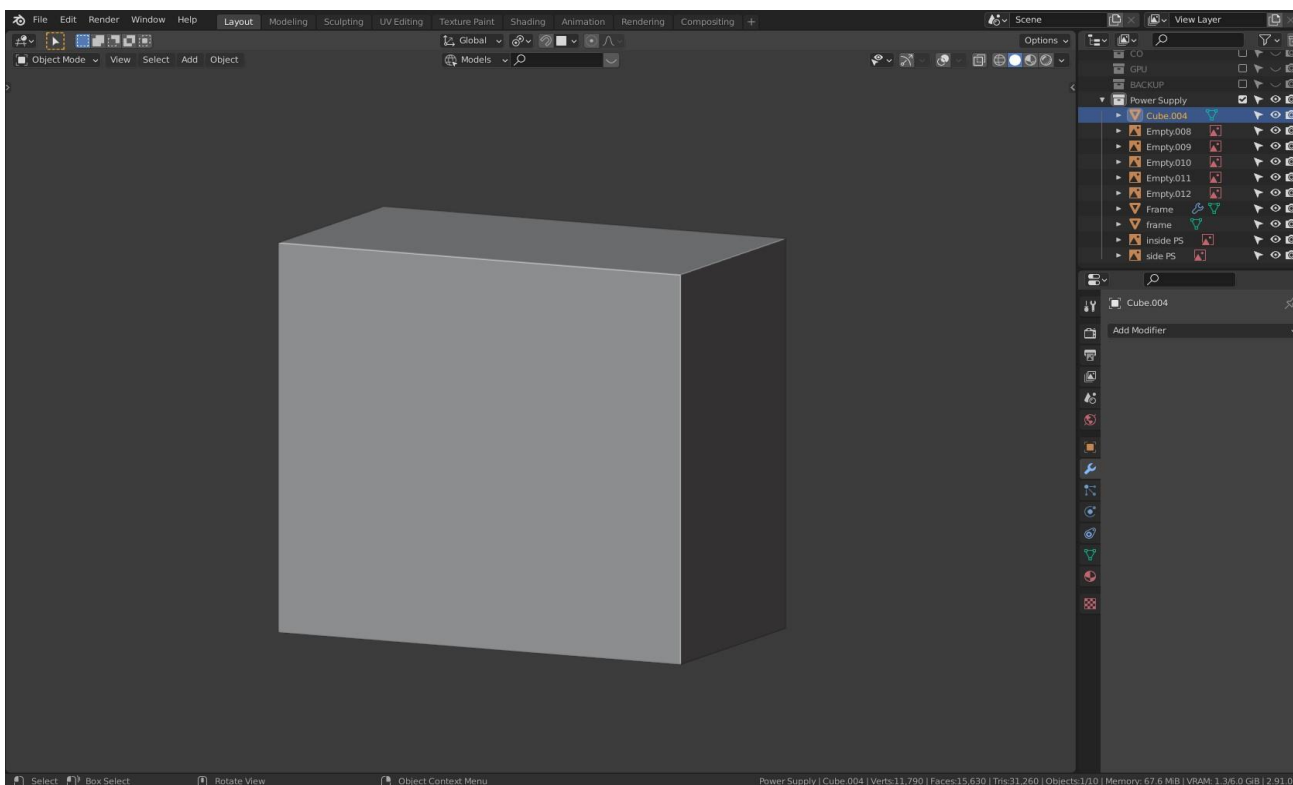


Рис.2.56. Додання у сцену примітив «Куб»

Використовуючи інструмент Insert face та Extrude робимо таку форму і додаємо модифікатор Wireframe. (рис. 2.57).

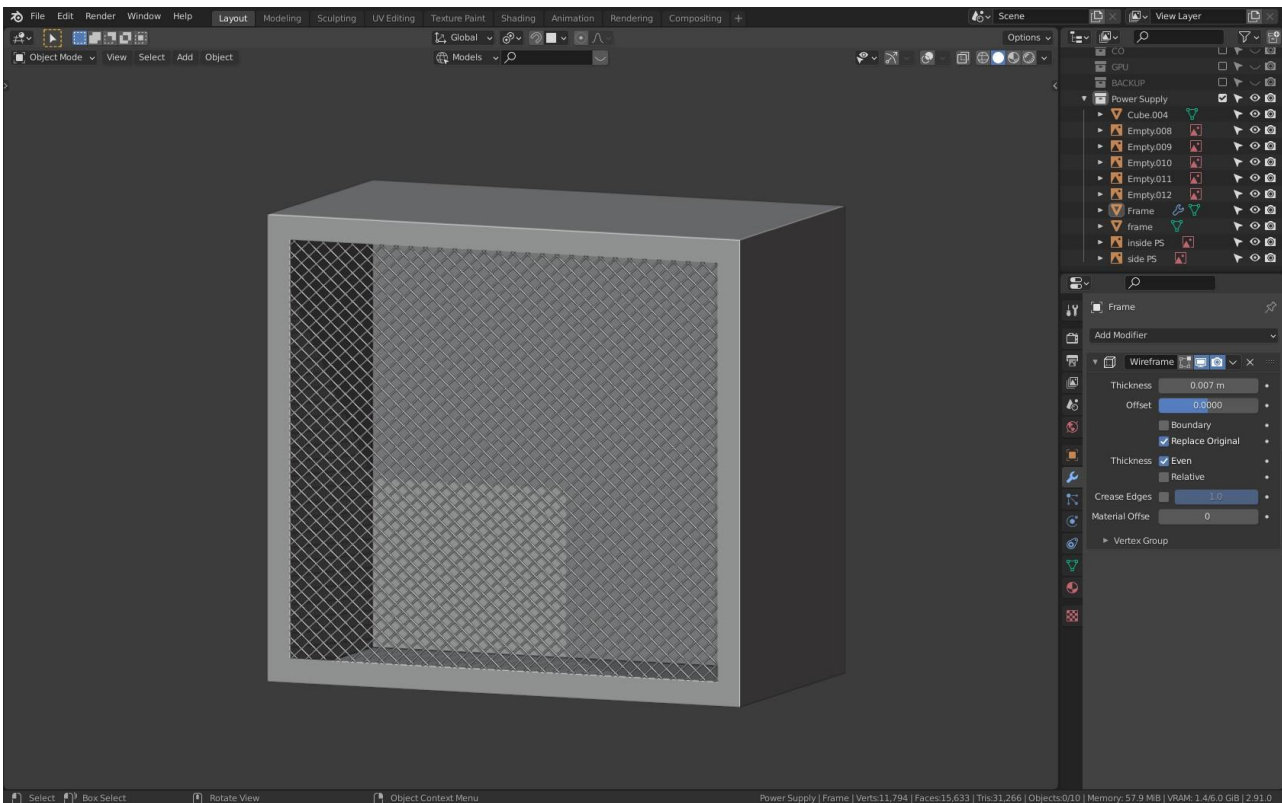


Рис.2.57. Модифікатор Wireframe

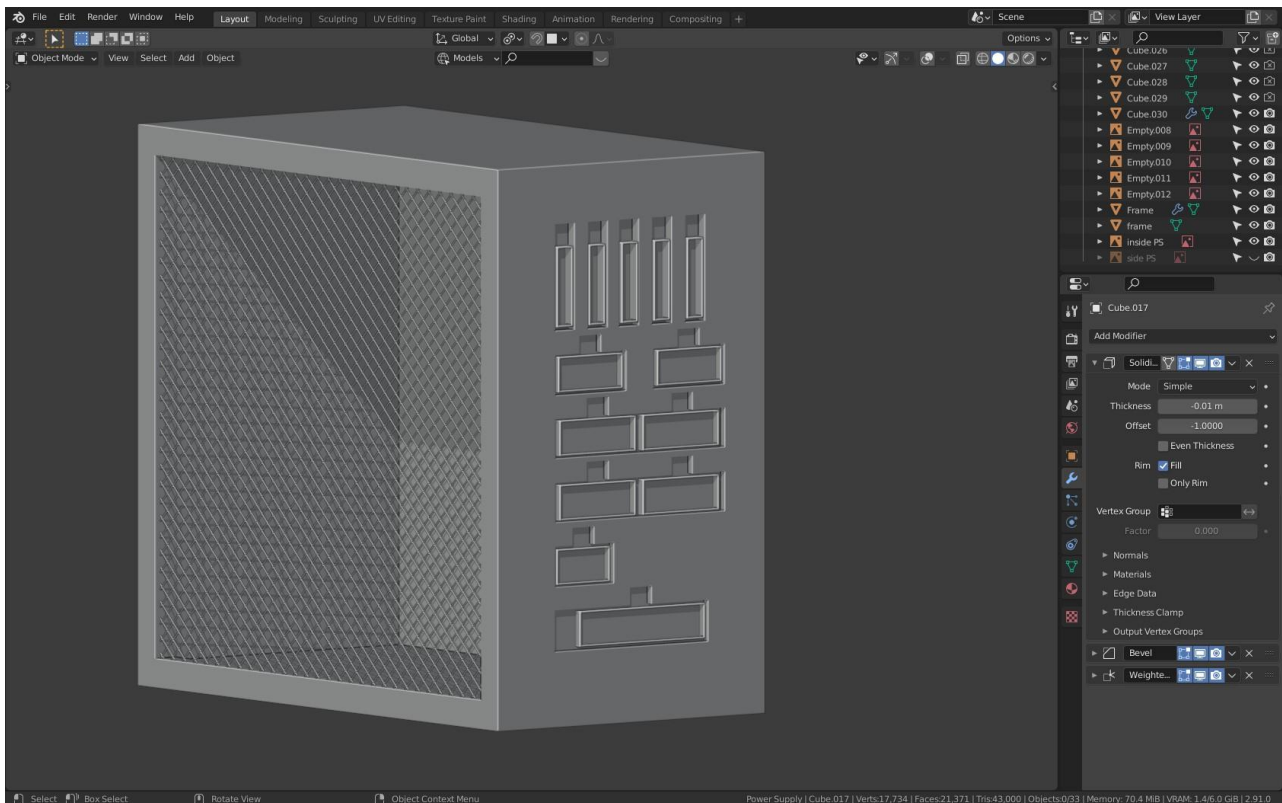


Рис.2.58. За допомогою Boolean робимо відтвори для підключення

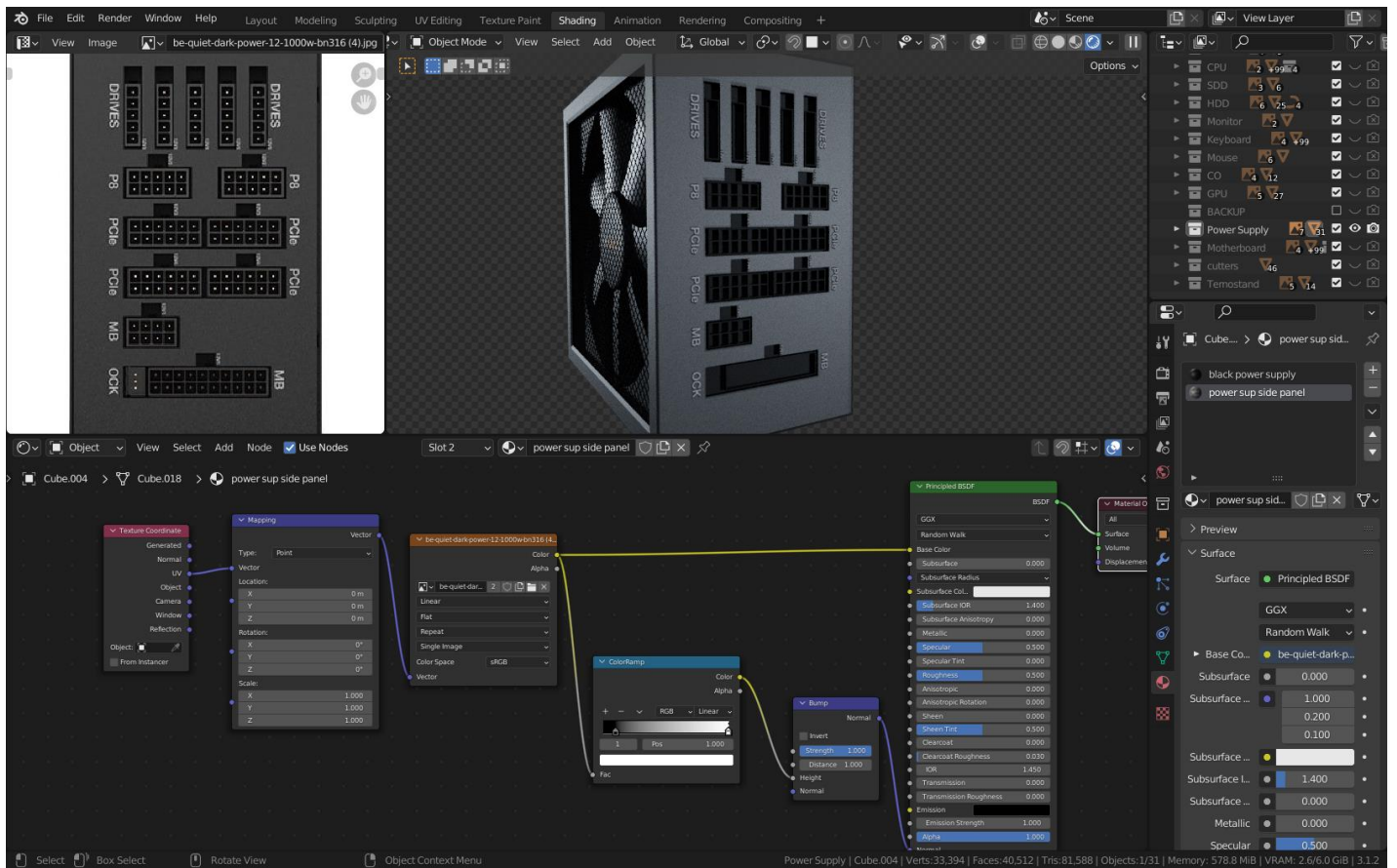


Рис.2.59. Створення матеріалу для блоку живлення



Рис.2.60. Фінальний рендер блоку живлення

2.5.9. Моделювання монітору

Тепер змоделюємо один із самих простіших компонентів ПК – монітор. Хоч монітор і прост у моделюванні, завжди треба орієнтуватися на референси.

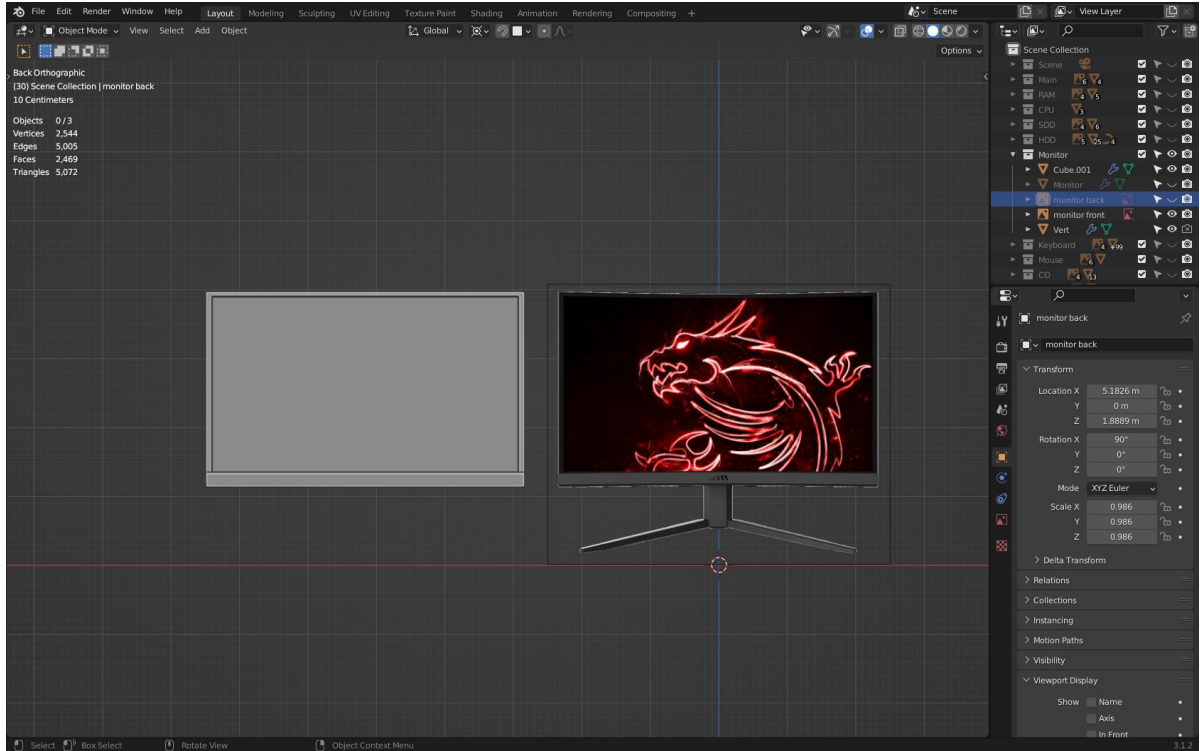


Рис.2.61. Моделювання монітор по референсу

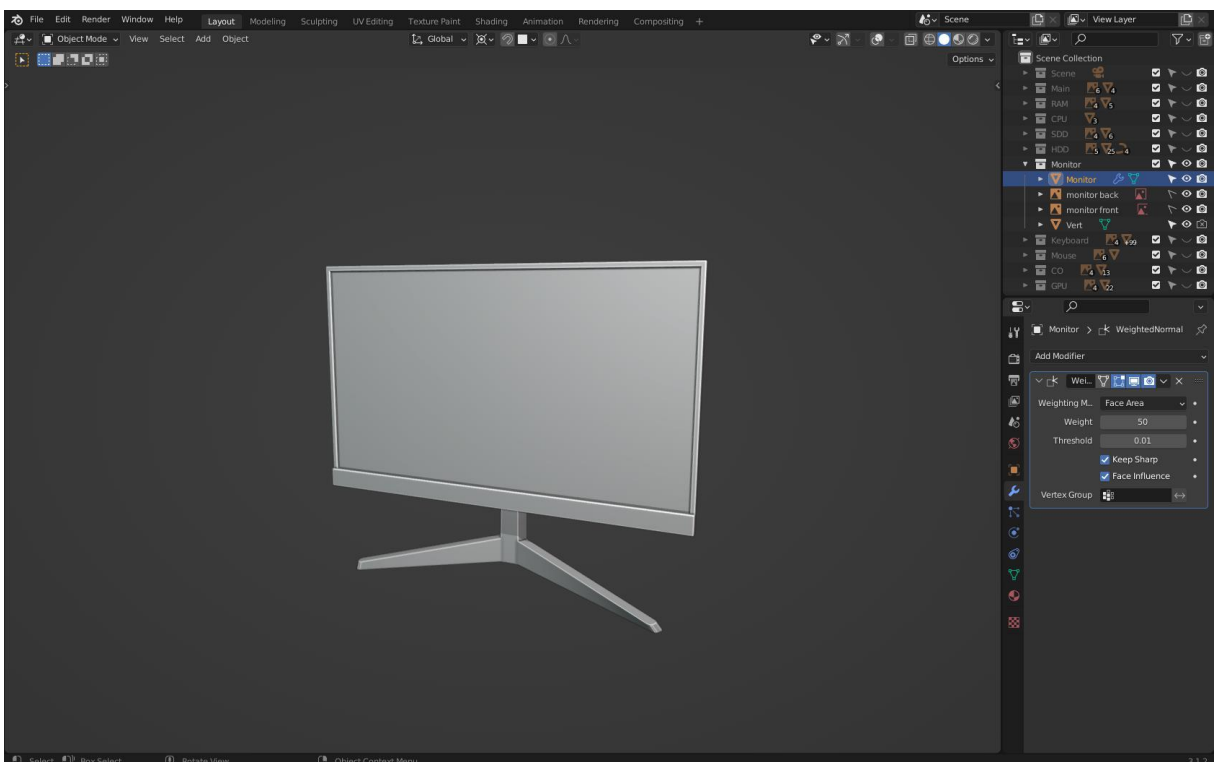


Рис.2.62. Моделювання ніжки монітора

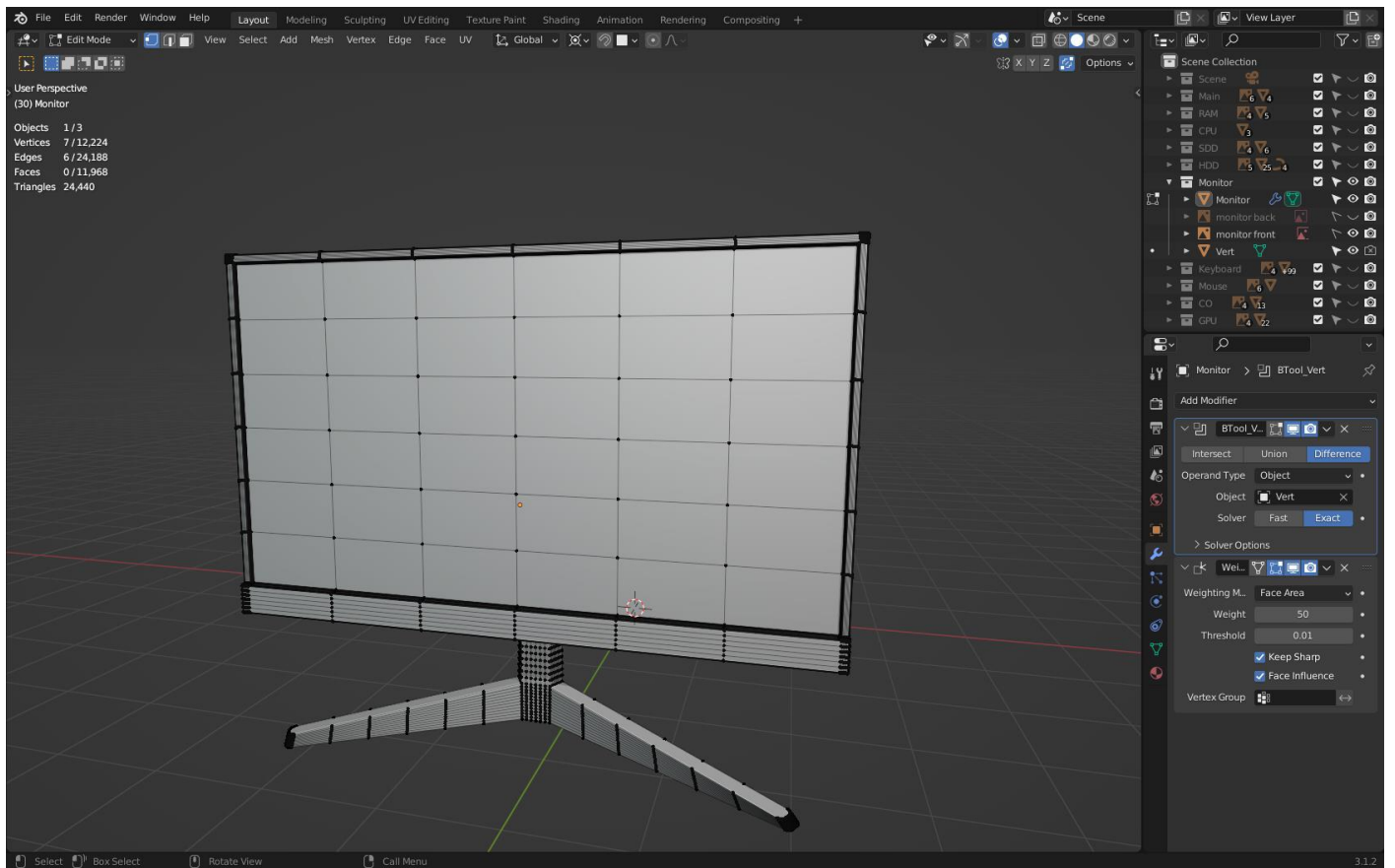


Рис.2.63. Топологія монітора

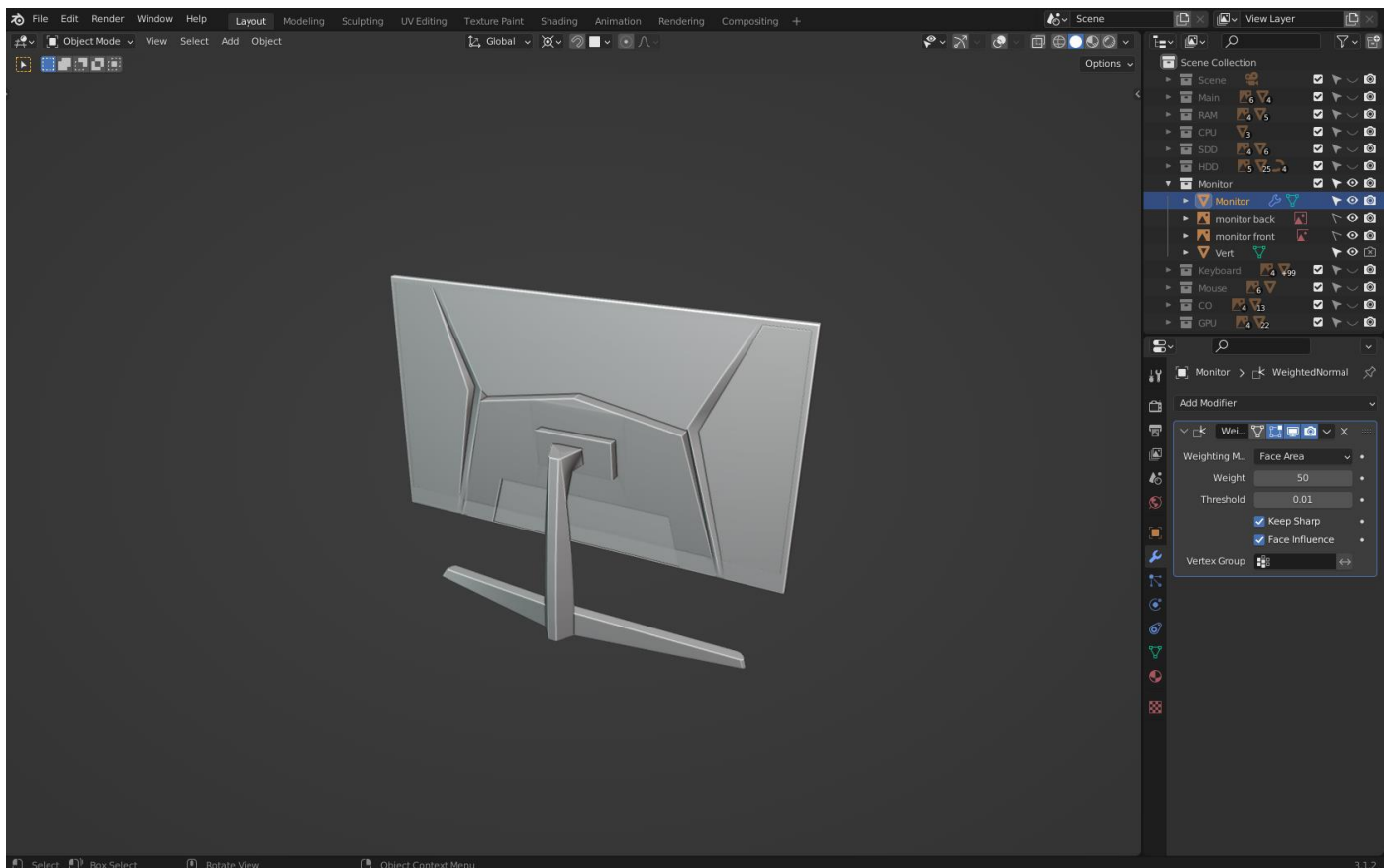


Рис.2.64. Вид ззаду



Рис.2.65. Фінальний рендер монітору

2.5.10. Моделювання материнської плати

Напевно найскладнішим компонентом є материнська плата. Так як вона має у декілька разів більше компонентів ніж будь-який інший компонент. Тож розіб'ємо процес моделювання на такий план:

1. Моделювання основи, тобто додання примітиву plane та подальшого його екструдювання або модифікатора solidify.
2. Моделювання бокової панелі, тобто портів підключення.
3. Моделювання конденсаторів та блочних елементів.
4. Моделювання слотів оперативної пам'яті та слотів для відеокарти та M2.
5. Моделювання мікросхем.
6. Моделювання сокету процесора та радіаторів.
7. З'єднання усіх компонентів до основної частини материнської плати.

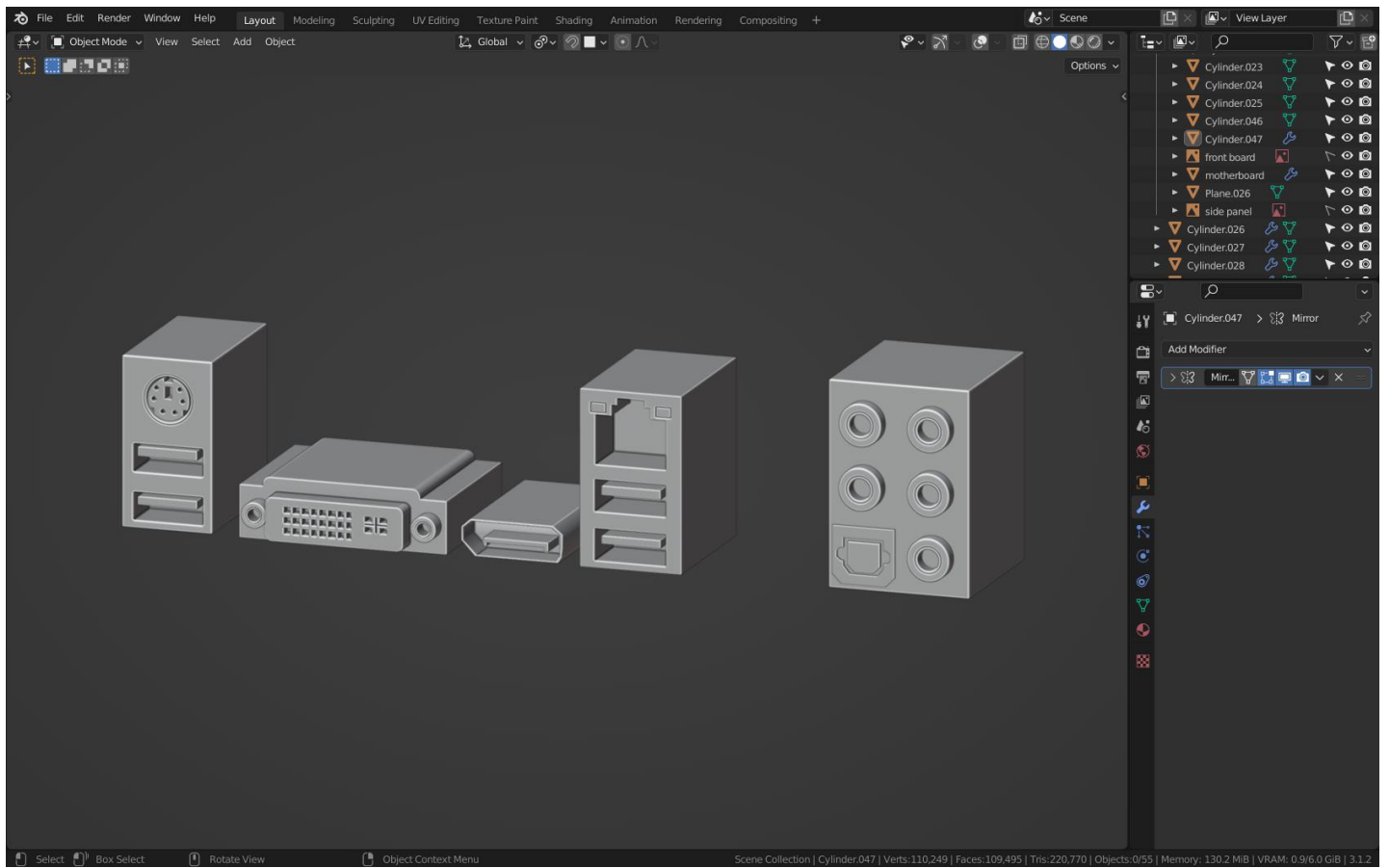


Рис.2.66. Side panel материнської плати

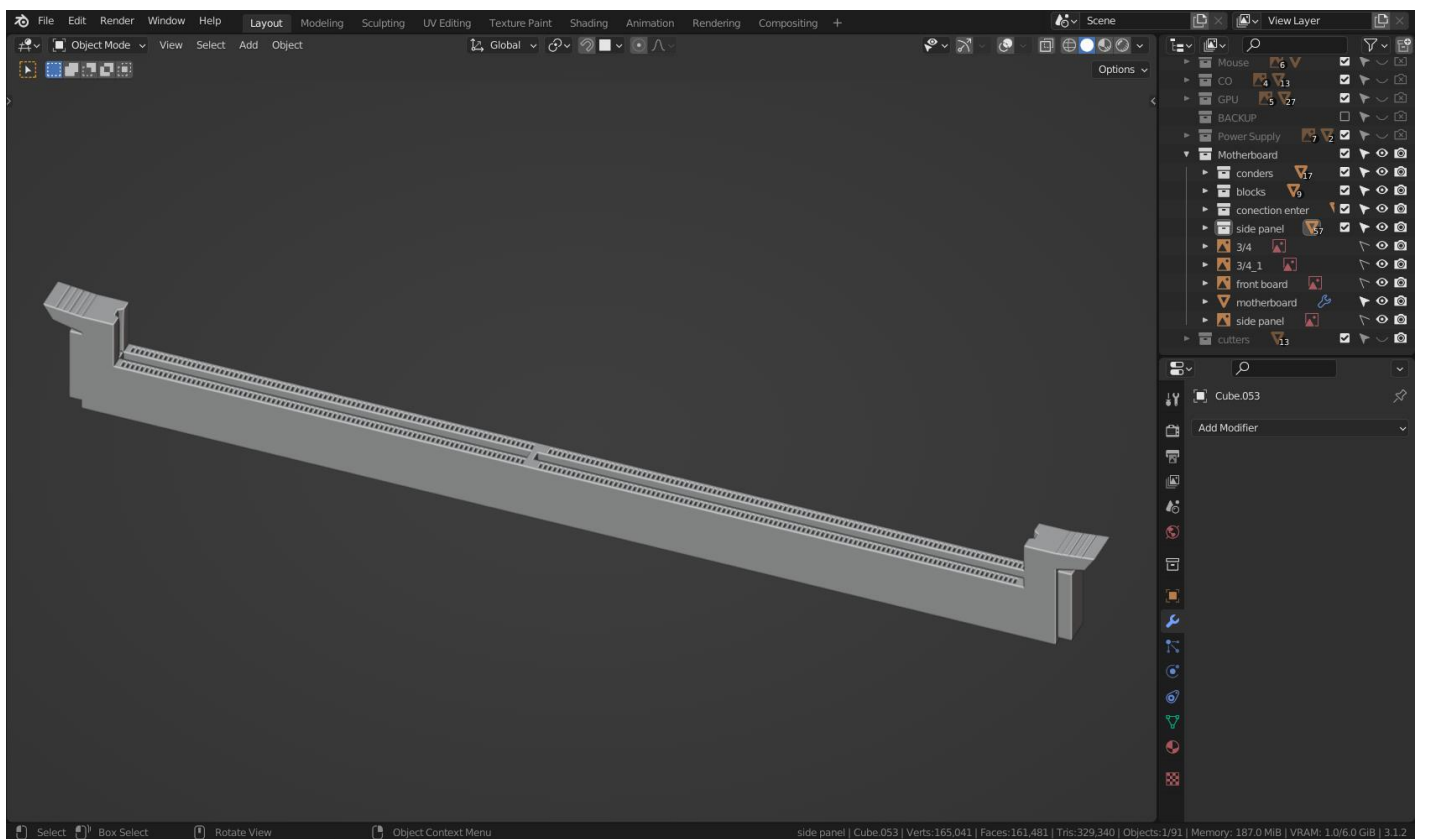


Рис.2.67. RAM slot



Рис.2.68. Clay version



Рис.2.69. Rendered version

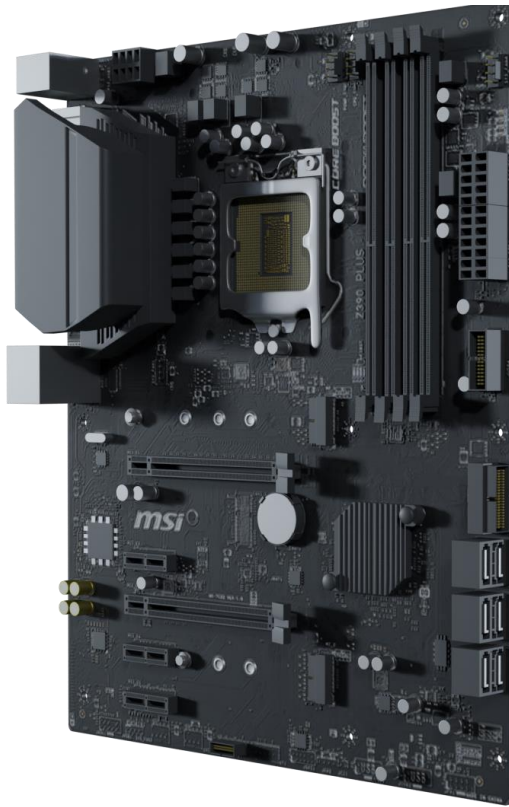


Рис.2.70. Фінальний рендер материнської плати

2.6. Опис розробленої системи

2.6.1. Використані технічні засоби

Для створення данної інформаційної безпеки використовувався 6-ти ядерний процесор Intel i5-9600k 4,0 Mz. Щоб вільно користуватися вьюпортом у рендер режимі використовувалися 6-ти гігова відеокарта Nvidia RTX 2060 6 GB та 16-ть гігабайт оперативної пам'яті Нурег X Fury 16 GB. Система створювалась на базі операційної системи Windows 10 Pro.

2.6.2. Використані програмні засоби

В данній кваліфікаційній роботі використовувався 3D пакет для моделювання та анімації Blender, HTML, CSS та JavaScript для створення сайту та Python у нодовій системі для текстуровання.

2.6.3. Виклик та завантаження програми

Виклик програми виконується запуском виконуючого файлу 3D_Animation_Timelapse.exe

2.6.4. Створення сайту галереї

Для простішого та нагляднішого перегляду рендерів буде створено сайт галерею. Писати сайт будемо на HTML та CSS з використання JavaScript. Код будемо писати у Visual Studio Code.



Рис.2.71. Сайт галерея

РОЗДІЛ 3

ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Вхідні дані:

q - передбачуване число операторів – 400

C - коефіцієнт складності програми – 1,25

p - коефіцієнт корекції програми в ході її розробки – 0,1

B - коефіцієнт збільшення витрат – 1,2

k - коефіцієнт кваліфікації програміста – 0,8

$C_{\text{ПР}}$ - годинна заробітна плата програміста, грн/год – 250.

B_k – число виконавців – 1 виконавець

F_p – місячний фонд робочого часу – 80 год

$C_{\text{МВ}}$ – вартість машинно-години ЕОМ – 50 грн/год

Оскільки створення данної програми ускладнено завдяки творчого характеру виконавця, можуть бути невеликі відклонення від норм.

Трудомісткість розробки розраховується за формулою:

$$t = t_{\text{и}} + t_{\text{а}} + t_{\text{п}} + t_{\text{отл}} + t_{\text{д}}, \text{ людино} - \text{годин}, \quad (3.1)$$

де t_0 – витрати праці на підготовку й опис поставленої задачі

$t_{\text{и}}$ - витрати праці на дослідження алгоритму рішення задачі

$t_{\text{а}}$ - витрати праці на розробку блок-схеми алгоритму

$t_{\text{п}}$ - витрати праці на програмування по готовій схемі

$t_{\text{отл}}$ - витрати праці на налагодження програми на ЕОМ

$t_{\text{д}}$ - витрати праці на підготовку документації

Умовне число операторів:

$$Q = q * C * (1+p), \text{ людино-годин} \quad (3.2)$$

де q – передбачуване число операторів

C – коефіцієнт складності програми

p – коефіцієнт кореляції програми в ході її розробки

$$Q = 400 * 1.25 * (1+0.1) = 550 \text{ людино-годин}$$

Витрати праці на вивчення опису задачі $t_{и}$ визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_{и} = \frac{QB}{(75...85)K}, \text{ людино} - \text{годин} \quad (3.3)$$

де B – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k – коефіцієнт кваліфікації програміста, обумовлений стажем роботи.

Витрати праці на розробку алгоритму рішення задачі:

$$t_{и} = \frac{550 * 1.2}{80 * 0.8} = \frac{660}{64} = 10, \text{ людино} - \text{годин}$$

Витрати на складання програми по готовій блок-схемі:

$$t_a = \frac{Q}{(20...25)K} \text{ людино} - \text{годин} \quad (3.4)$$

$$t_a = \frac{550}{20 * 0.8} = 34 \text{ людино} - \text{годин}$$

Витрати на складання програми по готовій блок-схемі:

$$t_{п} = \frac{Q}{(20...25)K} \text{ людино} - \text{годин} \quad (3.5)$$

$$t_{п} = \frac{550}{(25 * 0.8)} = 27 \text{ людино} - \text{годин}$$

Витрати праці на налагодження програми на ЕОМ:

$$t_{отл} = \frac{Q}{(4...5)K} \text{ людино} - \text{годин} \quad (3.6)$$

$$t_{отл} = \frac{550}{(4 * 0.8)} = 171 \text{ людино} - \text{годин}$$

За умови комплексного налагодження завдання:

$$t_{отл}^k = 1.5 * t_{отл} \quad (3.7)$$

$$t_{отл} = 171 * 1.5 = 256$$

Витрати праці на підготовку документації:

$$t_d = t_{др} + t_{до}, \text{ людино – годин} \quad (3.8)$$

де $t_{др}$ – трудоміскість підготовки матеріалів і рукопису.

$$t_{др} = \frac{q}{(15...20)K}, \text{ людино – годин} \quad (3.9)$$

$$t_{др} = \frac{550}{(15 * 0.8)} = 45, \text{ людино – годин}$$

де $t_{до}$ – трудоміскість редагування, печатки й оформлення документації

$$t_{до} = 0.75 * t_{др}, \text{ людино – годин} \quad (3.10)$$

$$t_{до} = 0.75 * 45 = 33, \text{ людино – годин}$$

$$t_d = 45 + 33 = 78, \text{ людино – годин}$$

Отримуємо трудоміскість розробки програмного забезпечення:

$$t = 50 + 10 + 34 + 27 + 256 + 78 = 455 \text{ людино-годин}$$

3.2 Розрахунок витрат на створення програми

Витрати на створення ПЗ Кпо включають витрати на заробітну плату виконавця програми Зз/п і витрат машинного часу, необхідного на налагодження програми на ЕОМ.

$$K_{по} = Z_{зп} + Z_{мв}, \text{ грн} \quad (3.11)$$

Заробітна плата виконавців визначається за формулою:

$$Z_{зп} = t * C_{пр}, \text{ грн} \quad (3.12)$$

де t – загальна трудоміскість, людино-годин

де $C_{пр}$ – середня годинна заробітна плата програміста, грн/година

$$Z_{зп} = 455 * 250 = 113750 \text{ грн}$$

Вартість машиного часу, необхідного для налагодження програми на ЕОМ:

$$З_{\text{мв}} = t_{\text{отл}} * C_{\text{м}}, \text{ грн} \quad (3.13)$$

де $t_{\text{отл}}$ – трудомісткість налагодження програми на ЕОМ

де $C_{\text{мв}}$ – вартість машино-години ЕОМ, 50 грн/год

Визначений в такий спосіб витрати на створення програмного забезпечення є частиною одноразових капітальних витрат АСУП.

$$З_{\text{мв}} = 256 * 50 = 12800 \text{ грн}$$

$$K_{\text{по}} = 113750 + 12800 = 126550 \text{ грн}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k * F_p} \text{ міс.} \quad (3.14)$$

де B_k – число виконавців

де F_p – місячний фонд робочого часу

$$T = \frac{455}{1 * 80} = 5.6 \text{ міс.}$$

Визначено трудомісткість розробленої інформаційної системи (455 людино-годин), проведений підрахунок вартості роботи по створенню програми (126550 грн) та розраховано час на його створення (5,6 міс).

ВИСНОВКИ

Метою кваліфікаційної роботи було розроблення інформаційного забезпечення синтезу архітектури комп'ютера на основі трьохмірного моделювання та візуалізації.

Для розробки компонентів використовувався 3D пакет Blender. Текстурування було виконано у тому же пакеті за допомогою нодової системи програмування на базі Python.

Як можна побачити у данній роботі, Blender може якісно відобразити моделі архітектури комп'ютера. Також завдяки внутрішнім інструментам було виконано текстурування та анімацію. Завдяки сучасним рендер двигунам, такі як Eevee та Cycles X вийшло зробити фотореалістичні моделі. У режимі «Compositing» була зроблена пост обробка рендерів для більш деталізованого вигляду. Для створення сайту використовувався JavaScript, написаний у середі Visual Studio Code.

Для досягнення поставленої мети було обрано наступні програми: «Blender», «Adobe Photoshop», «Visual Studio Code», «Voxel 3D».

В результаті було створене інформаційне забезпечення з детальним оглядом сбору персонального комп'ютера у реальному часі.

В економічному розділі визначено трудомісткість розробленого забезпечення (455 людино-годин), проведений підрахунок вартості роботи створення забезпечення (126550 грн.) та розраховано час на його створення (5,6 міс.)

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Офіційний сайт Blender Foundation. URL <https://www.blender.org/about/>
2. Методичні вказівки з виконання економічного розділу в дипломних проектах студентів спеціальностей «Комп'ютерні науки» Національна політехніка.
3. Методичні вказівки до виконання кваліфікаційної роботи бакалаврів напряму підготовки 122 «Комп'ютерні науки» О.С.Шевцова; Національна політехніка.
4. Воган В. - Посібник з топології – 2018 – 267 с.
5. Різ Е. Як зробити красиво в 3D-дизайні / Е. Різ. 2012. - 288 с.
6. Прахів А. Blender. 3D-моделювання та анімація. Керівництво для початківців/А. Прахов. 2009. - 272 с.
7. Голованов Н.М. Геометричне моделювання– К., 2002. – 630 с.
8. Color Contrast Analyzer URL: <http://tools.cactusflower.org/analyzer/>
9. Розендаль Т. Інструменти моделювання в Blender / Тон Розендаль., 2016. – 135 с.
10. Guevarra Ezra Thess Mendoza. Modeling and Animation Using Blender Blender. Blender 2.80: The Rise of Eevee 2015. - p. 584.
11. Valenza Enrico, Kuhn Christopher, Nicq Pierre-Armand, Caudron Romain. Blender 3D: Characters, Machines and Scenes for Artist 2018. – p. 99.
12. Myers Mark. Blender 3D for Graphics Designers 2018. – p. 758.
13. Hollister B. Core Blender Development: Understanding the Essential Source Code.
14. Moioli G. Introduction to Blender 3.0: Learn Organic and Architectural Modeling, Lighting, Materials, Painting, Rendering, and Compositing with Blender.
15. Simonds B. Blender Master Class: A Hands-On Guide to Modeling, Sculpting, Materials, and Rendering.

16. McDermott W. Creating 3D Game Art for the iPhone with Unity: Featuring modo and Blender pipelines, 2009, p.785.
17. Larson Joe. 3d Printing Designs: Fun And Functional Projects
18. Valenza E. Blender 3D Cookbook, 2021. – p. 345.
19. Kuhn Christopher. Blender 3D Incredible Machines, 2020., p.500.
20. David Buliko JavaScript for Noobs, 2018. - p.180.
21. Patterson James. JAVA: A Beginner's Guide to Learning the Basics of Java Programming.
22. Freeman E., Robson E. Head First HTML5 Programming: Building Web Apps with JavaScript.
23. Van der Spuy R. Foundation Game Design with HTML5 and JavaScript.
24. Clack J.E., Johnson B.P. Sencha Touch Mobile JavaScript Framework.
25. Roberts Eric. Introduction to JavaScript Programming. The 'Nothing but a Browser' Approach.
26. Fu Pinde. Getting to know Web GIS.
27. Ranjan A., Sinha A., Battewad R. JavaScript for Modern Web Development: Building a Web Application Using HTML, CSS, and JavaScript.
28. Hu Yang. Easy Learning HTML CSS Javascript: Step by step to lead to learn HTML CSS Javascript better and fast.
29. McGrath M. HTML, CSS & JavaScript In Easy Steps.
30. Kyrnin J., Meloni J. HTML, CSS, and JavaScript: All in One.

КОД ПРОГРАМИ

```

<!DOCTYPE html>
<html>

<head>

  <meta charset="utf-8">

  <title>3D Gallery</title>
  <meta name="description" content="Showcase 3D Effect of computer components">

  <meta name="viewport" content="width=device-width, initial-scale=1.0, minimum-scale=1.0,
maximum-scale=1.0, user-scalable=no">

  <link rel="icon" href="images/favicon.png">
  <meta property="og:image" content="images/preview.jpg">

  <link rel="stylesheet" href="libs/bootstrap/css/bootstrap-reboot.min.css">
  <link rel="stylesheet" href="libs/swiper/swiper-bundle.min.css">

  <link rel="stylesheet" href="css/main.css">
  <link rel="stylesheet" href="css/media.css">

</head>

<body>

  <section class="showcase">

    <h2 class="showcase__header">3D <span>GALLERY</span></h2>

    <div class="showcase__content-wrapper">
      <div class="showcase__content">

        <div class="showcase-carousel">

          <div class="swiper-wrapper">

            <div class="swiper-slide showcase-carousel__item">
              <div class="showcase-carousel__image-wrapper">
                <div class="showcase-carousel__image-left">
                  <div class="showcase-carousel__image" style="background-image:
url(images/CPU.png);"></div>
                </div>
                <div class="showcase-carousel__image-right">
                  <div class="showcase-carousel__image" style="background-image:
url(images/CPU.png);"></div>

```

```
</div>
</div>
<p>CPU</p>
</div>
```

```
<div class="swiper-slide showcase-carousel__item">
  <div class="showcase-carousel__image-wrapper">
    <div class="showcase-carousel__image-left">
      <div class="showcase-carousel__image" style="background-image:
url(images/GPU.png);"></div>
    </div>
    <div class="showcase-carousel__image-right">
      <div class="showcase-carousel__image" style="background-image:
url(images/GPU.png);"></div>
    </div>
  </div>
  <p>GPU</p>
</div>
```

```
<div class="swiper-slide showcase-carousel__item">
  <div class="showcase-carousel__image-wrapper">
    <div class="showcase-carousel__image-left">
      <div class="showcase-carousel__image" style="background-image:
url(images/Motherboard.png);"></div>
    </div>
    <div class="showcase-carousel__image-right">
      <div class="showcase-carousel__image" style="background-image:
url(images/Motherboard.png);"></div>
    </div>
  </div>
  <p>Motherboard</p>
</div>
```

```
<div class="swiper-slide showcase-carousel__item">
  <div class="showcase-carousel__image-wrapper">
    <div class="showcase-carousel__image-left">
      <div class="showcase-carousel__image" style="background-image:
url(images/HDD.png);"></div>
    </div>
    <div class="showcase-carousel__image-right">
      <div class="showcase-carousel__image" style="background-image:
url(images/HDD.png);"></div>
    </div>
  </div>
  <p>HDD</p>
</div>
```

```
<div class="swiper-slide showcase-carousel__item">
  <div class="showcase-carousel__image-wrapper">
    <div class="showcase-carousel__image-left">
      <div class="showcase-carousel__image" style="background-image:
url(images/SSD.png);"></div>
```

```

    </div>
    <div class="showcase-carousel__image-right">
      <div class="showcase-carousel__image" style="background-image:
url(images/SSD.png);"></div>
    </div>
  </div>
  <p>SSD</p>
</div>

```

```

<div class="swiper-slide showcase-carousel__item">
  <div class="showcase-carousel__image-wrapper">
    <div class="showcase-carousel__image-left">
      <div class="showcase-carousel__image" style="background-image:
url(images/RAM.png);"></div>
    </div>
    <div class="showcase-carousel__image-right">
      <div class="showcase-carousel__image" style="background-image:
url(images/RAM.png);"></div>
    </div>
  </div>
  <p>RAM</p>
</div>

```

```

<div class="swiper-slide showcase-carousel__item">
  <div class="showcase-carousel__image-wrapper">
    <div class="showcase-carousel__image-left">
      <div class="showcase-carousel__image" style="background-image:
url(images/POWER\ SUPPLY.png);"></div>
    </div>
    <div class="showcase-carousel__image-right">
      <div class="showcase-carousel__image" style="background-image:
url(images/POWER\ SUPPLY.png);"></div>
    </div>
  </div>
  <p>Power Supply</p>
</div>

```

```

<div class="swiper-slide showcase-carousel__item">
  <div class="showcase-carousel__image-wrapper">
    <div class="showcase-carousel__image-left">
      <div class="showcase-carousel__image" style="background-image:
url(images/TERMO\ TOWER.png);"></div>
    </div>
    <div class="showcase-carousel__image-right">
      <div class="showcase-carousel__image" style="background-image:
url(images/TERMO\ TOWER.png);"></div>
    </div>
  </div>
  <p>Termo Tower</p>
</div>

```

```

<div class="swiper-slide showcase-carousel__item">

```

```

        <div class="showcase-carousel__image-wrapper">
            <div class="showcase-carousel__image-left">
                <div class="showcase-carousel__image" style="background-image:
url(images/SVO.png);"></div>
            </div>
            <div class="showcase-carousel__image-right">
                <div class="showcase-carousel__image" style="background-image:
url(images/SVO.png);"></div>
            </div>
            </div>
            <div>
                <p>SVO</p>
            </div>

        </div>

</div>

<div class="showcase-navigation">
    <div class="showcase-navigation__prev"></div>
    <div class="showcase-navigation__next"></div>
</div>

</div>
</div>

<video src="videos/smoke-background-optimized.mp4" class="showcase__video" autoplay loop
muted></video>

</section>

<script src="libs/swiper/swiper-bundle.min.js"></script>

<script src="js/common.js"></script>

</body>

</html>
:root {
    --accent: #CC8869;
    --text: #333;
    --regular-text: 16px;
    --lineheight: 1.65;
    --userfont: roboto-st, sans-serif;
    --systemfont: -apple-system, BlinkMacSystemFont, Arial, sans-serif;
    --padding: 120px;
    --transition: cubic-bezier(.4, 0, 0, 1);
}

/* BOOTSTRAP SETTINGS SECTION */

/* gutter 20px (10px + 10px). Comment this code for default gutter start at 1.5rem (24px) wide. */

```

```
.container, .container-fluid, .container-lg, .container-md, .container-sm, .container-xl, .container-xxl
{ --bs-gutter-x: .625rem; }
.row, .row > * { --bs-gutter-x: 1.25rem; }
```

```
/* FONTS LOAD SECTION */
```

```
@font-face { src: url("../fonts/roboto-regular-webfont.woff2") format("woff2"); font-family:
"roboto-st"; font-weight: 400; font-style: normal; }
@font-face { src: url("../fonts/roboto-italic-webfont.woff2") format("woff2"); font-family: "roboto-
st"; font-weight: 400; font-style: italic; }
@font-face { src: url("../fonts/roboto-bold-webfont.woff2") format("woff2"); font-family: "roboto-
st"; font-weight: 700; font-style: normal; }
@font-face { src: url("../fonts/roboto-bolditalic-webfont.woff2") format("woff2"); font-family:
"roboto-st"; font-weight: 700; font-style: italic; }
@font-face { src: url("../fonts/assassin.woff2") format("woff2"); font-family: "assassin-st"; font-
weight: 700; font-style: italic; }
```

```
/* GENERAL CSS SETTINGS */
```

```
::placeholder { color: #666; }
::selection { background-color: var(--accent); color: #fff; }
input, textarea { outline: none; }
input:focus:required:invalid, textarea:focus:required:invalid { border-color: red; }
input:required:valid, textarea:required:valid { border-color: green; }
```

```
body {
  font-family: var(--userfont);
  font-size: var(--regular-text);
  line-height: var(--lineheight);
  color: var(--text);
  min-width: 320px;
  position: relative;
  overflow-x: hidden;
}
```

```
/* USER STYLES */
```

```
body, html {
  height: 100%;
}
```

```
.showcase {
  background-color: #000;
  height: 100%;
  position: relative;
  color: #fff;
  /* overflow: hidden; */
}
```

```
.showcase::before {
  content: "";
  width: 100%;
  height: 100%;
}
```

```

position: absolute;
z-index: 1;
box-shadow: inset 0 0 500px #000;
}
.showcase::after {
content: "";
position: absolute;
top: 0;
right: 0;
bottom: 0;
left: 0;
background-image: url(../images/light.png);
background-size: cover;
background-repeat: no-repeat;
background-position: center -65px;
animation: k-light 3s ease-in-out infinite;
}
.showcase__video {
position: absolute;
top: 0;
left: 0;
width: 100%;
height: 100%;
object-fit: cover;
opacity: .48;
}
.showcase__content-wrapper {
position: relative;
z-index: 2;
height: 100%;
padding: var(--padding) 0;
}
.showcase__content-wrapper::before {
content: "";
position: absolute;
top: 0;
right: 0;
bottom: 0;
left: 0;
background-image: url(../images/glow.png);
background-position: center;
background-repeat: no-repeat;
background-size: cover;
animation: k-glow 1.6s ease-in-out infinite;
}
.showcase__header {
position: absolute;
text-align: center;
width: 100%;
z-index: 1;
top: 40px;
font-family: assassin-st, sans-serif;

```

```

font-size: 32px;
color: rgba(255, 255, 255, .75);
}
.showcase__header span {
color: var(--accent);
}
.showcase-carousel .swiper-wrapper {
transition: var(--transition);
}
.showcase-carousel__item {
position: relative;
height: calc(100vh - var(--padding)*2);
text-align: center;
opacity: .25;
transform: scale(.75);
transition: opacity 1.8s var(--transition), transform 1.8s var(--transition)
}
.showcase-carousel__item::after {
content: "";
width: 120px;
height: 0;
position: absolute;
box-shadow: 0 0 45px 10px #010101;
bottom: 5px;
left: calc(50% - 60px);
}
.showcase-carousel__item p {
position: absolute;
bottom: -85px;
width: 100%;
font-family: assassin-st, sans-serif;
font-size: 32px;
color: rgba(255, 255, 255, .5);
text-shadow: rgb(58 78 94) 0 0 10px;
}
.showcase-carousel__item.swiper-slide-active {
opacity: .8;
transform: scale(.99);
}
.showcase-carousel__image-wrapper > * {
position: absolute;
width: 100%;
height: 100%;
perspective: 150px;
transform-style: preserve-3d;
}
.showcase-carousel__image {
position: absolute;
width: 100%;
height: 100%;
background-size: contain;
background-position: center;

```



```

background-repeat: no-repeat;
}
.showcase-carousel__image-left {
  perspective-origin: left center;
  clip-path: polygon(0 0, 50% 0, 50% 100%, 0 100%);
}
.showcase-carousel__image-right {
  perspective-origin: right center;
  clip-path: polygon(50% 0, 100% 0, 100% 100%, 50% 100%);
  /* Фикс вертикальной полосы на некоторых дисплеях */
  margin-left: -.55px
}
.showcase-carousel__image-left .showcase-carousel__image {
  animation: k-left-side 2s ease-in-out infinite;
  animation-direction: alternate;
}
.showcase-carousel__image-right .showcase-carousel__image {
  animation: k-right-side 2s ease-in-out infinite;
  animation-direction: alternate;
}

/* NAVIGATION */

.showcase-navigation {
  overflow: hidden;
  position: absolute;
  z-index: 2;
  top: 0;
  right: 0;
  bottom: 0;
  left: 0;
}
.showcase-navigation > * {
  position: absolute;
  height: 100%;
  width: 50%;
  outline: none;
  cursor: pointer;
}
.showcase-navigation__next {
  right: 0;
}
.showcase-navigation > *::before {
  content: '?';
  font-family: assassin-st;
  opacity: 0;
  transition: opacity .75s ease-out;
  position: absolute;
  transform: rotate(-90deg);
  left: 50px;
  top: 50%;
  font-size: 45px;
}

```

```

}
.showcase-navigation > *:hover::before {
  opacity: .25;
}
.showcase-navigation__next::before {
  transform: rotate(90deg);
  right: 50px;
  left: auto;
}
.showcase-navigation > *.swiper-button-disabled {
  display: none;
}

```

```

/* ANIMATIONS */

```

```

@keyframes k-light {
  0% {
    opacity: .2;
  }
  50% {
    opacity: .3;
  }
  100% {
    opacity: .2;
  }
}
@keyframes k-glow {
  0% {
    opacity: .6;
  }
  50% {
    opacity: .8;
  }
  100% {
    opacity: .6;
  }
}
@keyframes k-left-side {
  0% {
    transform: rotateY(-1deg) scaleX(.92);
  }
  100% {
    transform: rotateY(0deg) scaleX(1);
  }
}
@keyframes k-right-side {
  0% {
    transform: rotateY(0deg) scaleX(1);
  }
  100% {
    transform: rotateY(1deg) scaleX(.92);
  }
}

```

```

}

/* xl <= Extra extra large (xxl) */
@media (max-width : 1400px) {

    .showcase__content {
        max-width: 1200px;
        margin: auto;
    }

}

/* lg <= Extra large (xl) */
@media (max-width : 1200px) {

}

/* md <= Large (lg) */
@media (max-width : 992px) {

}

/* sm <= Medium (md) */
@media (max-width : 768px) {

}

/* xs <= Small (sm) */
@media (max-width : 576px) {

}

/* ==== MOBILE FIRST ==== */

/* Custom (xs) */
@media (min-width : 0) {

}

/* Small (sm) */
@media (min-width : 576px) {

}

/* Medium (md) */
@media (min-width : 768px) {

}

/* Large (lg) */
@media (min-width : 992px) {

```

```

}

/* Extra large (xl) */
@media (min-width : 1200px) {
  .text {
    font-size: 18px;
  }
}

/* Extra extra large (xxl) */
@media (min-width : 1400px) {
  .text {
    font-size: 22px;
  }
}

document.addEventListener('DOMContentLoaded', function() {

  const showSlider = new Swiper('.showcase-carousel', {
    loop: true,
    slidesPerView: 3,
    speed: 1800,
    centeredSlides: true,
    navigation: {
      nextEl: '.showcase-navigation__next',
      prevEl: '.showcase-navigation__prev'
    }
  })

  document.querySelector('video').playbackRate = 2

})

```

ДОДАТОК Б

ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файлу	Опис
Пояснювальні документи	
Диплом_Валах.doc	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Диплом_Валах.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Program.rar	Архів. Містить коди програми і откомпільовану програму.
Презентація	
Презентація_Валах.ppt	Презентація кваліфікаційної роботи