

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення
Дисципліна: Інженерія програмного забезпечення

Лабораторна робота №6
Тема: «СТВОРЕННЯ ФАЙЛУ ДОПОМОГИ
ДО РОЗРОБЛЕНОЇ ПРОГРАМИ»

Виконав: ст. гр. КН-24
Куріщенко П. В.
Перевірив: викладач
Смірнов О. А.

Мета: Навчитися складати технічні документи і керівництво для користувача відповідно до ДСТУ ISO/IEC 26514:2015 або ISO/IEC/IEEE 26514:2022 (Інженерія систем і програмного забезпечення. Вимоги до дизайнерів і розробників документації користувача). Отримати навички подання керівництва для користувача в електронному форматі MS HTML Help Workshop (.chm).

Назва проєкту: Розробка програми 3Д-движка.

ЗАВДАННЯ

Скласти керівництво для користувача програми розробленої в ЛР №5 відповідно до ДСТУ ISO/IEC 26514:2015 Інженерія систем і програмного забезпечення. Вимоги до дизайнерів і розробників документації користувача (ISO/IEC 26514:2008, IDT) або більш новий міжнародний стандарт: ISO/IEC/IEEE 26514:2022. Systems and software engineering – Design and development of information for users, й зробити електронну версію цього посібника у форматі довідкової системи MS HTML Help Workshop. Можливе використання іншого компілятора за узгодженням у лектора.

КЕРУВАННЯ КОРИСТУВАЧА (ISO/IEC 26514)

Назва: 3D Engine Viewer (LP5)

Автор: Куріщенко Павло Володимирович, КН-24

Версія: 1.0

Рік: 2025

Відмова від відповідальності

Дане програмне забезпечення створено виключно в навчальних цілях у межах лабораторної роботи.

Автор не гарантує повної відповідності обладнанню користувача, відсутності помилок або стабільності роботи.

Використання програми здійснюється на власний ризик. Автор не несе відповідальності за можливі втрати даних чи збої системи.

Зміст

1. Вступ
2. Огляд програми
3. Встановлення
4. Використання програми
 - 4.1. Запуск програми
 - 4.2. Основні дії
 - 4.3. Типові сценарії
5. Повідомлення про помилки
6. Вирішення типових проблем
7. Технічна підтримка

1. Вступ

Призначення

3D Engine Viewer призначений для перегляду та базового тестування роботи простого 3D-движка: рендеринг моделей, додавання освітлення, керування камерою, імпорт та експорт сцен.

Цільові користувачі

- студенти, які виконують лабораторну роботу;
- викладачі, що здійснюють оцінювання;
- початкові користувачі, які хочуть протестувати функціонал 3D-движка.

Необхідні умови

- **ОС:** Windows 10 або новіша
- **Графіка:** відеодрайвер з підтримкою OpenGL 3.0+

- **Бажано:** GPU (достатньо інтегрованої графіки)

Основні функції

- рендеринг 3D сцени з керованою камерою;
- додавання примітивних моделей: *Triangle, Cube, Pyramid, Sphere*;
- створення точкових джерел світла;
- зміна текстур моделей;
- збереження та завантаження сцен у форматі *.scene*;
- очищення сцени;
- відображення FPS та Zoom-коефіцієнта.

2. Огляд програми

Головне вікно складається з таких елементів:

Область перегляду сцени

Центральне OpenGL-поле для рендерингу графіки.

Меню програми

Scene

- *Load scene* — завантажити *.scene*
- *Import scene* — експортувати поточну сцену
- *Default scene* — очистити сцену
- *Reset camera* — повернути камеру до початкового стану

Models

- *Change texture* — змінити текстуру вибраної моделі

Lights

- *Place here* — додати точкове світло в позицію камери
- *Color* — встановити колір
- *Intensity* — задати інтенсивність світла

Samples

- *Triangle / Cube / Pyramid / Sphere* — додавання примітивів

Управління камерою

- **Ліва кнопка + рух миші** — поворот (Yaw/Pitch)

- **Права кнопка + рух миші** — панорамування (Right/Up)
- **Колесо натиснуте + рух** — рух по осі (вперед/назад)
- **Колесо прокрутки** — Zoom (зміна FOV, 10°–120°)

Додаткові показники

- **FPS** — оновлення кожні 0.5 с
- **Zoom** — 60 / FOV

3. Встановлення

Мінімальні вимоги

- Windows 10 (64-bit)
- OpenGL 3.0+
- Qt 6.7 DLL-бібліотеки

Типовий склад дистрибутиву

- 3DEngine.exe
- platforms/ — Qt платформи
- styles/, imageformats/ — плагіни

Запуск програми

1. Подвійний клік по **3DEngine.exe**, або
2. Через термінал:
 1. cd path\to\folder
 2. 3DEngine.exe <файл сцени (опціонально)>

4. Використання програми

4.1. Запуск програми

Після старту:

- якщо сцена порожня — автоматично додається *Triangle*;
- камера встановлюється в стандартну позицію:

(4, 3, 4), yaw = −135°, pitch = −20°, zoom = 60.

4.2. Основні дії

Маніпуляції камерою

- Поворот: ліва кнопка + рух
- Панорамування: права кнопка + рух
- Рух по осі: колесо натиснуте + рух
- Зміна приближення: колесо прокрутки

Операції над сценою

- Додати модель: *Samples* → *Triangle/Cube/Pyramid/Sphere*
- Змінити колір світла: *Lights* → *Color*
- Змінити інтенсивність: *Lights* → *Intensity (0–10)*
- Додати світло: *Lights* → *Place here*
- Змінити текстуру: *Models* → *Change texture*
- Завантажити сцену: *Scene* → *Load scene*
- Зберегти сцену: *Scene* → *Import scene*
- Очистити сцену: *Scene* → *Default scene*
- Скинути камеру: *Scene* → *Reset camera*

4.3. Типові сценарії роботи

Сценарій 1: “Очистити і зібрати тестову сцену”

1. *Scene* → *Default scene*
2. *Samples* → *Cube, Pyramid, Sphere*
3. *Lights* → *Intensity* (напр. 2.5)
4. *Lights* → *Place here*
5. *Models* → *Change texture*

Сценарій 2: “Підготувати сцену та зберегти”

1. *Scene* → *Load scene* (або зібрати вручну)
2. Налаштувати освітлення
3. *Scene* → *Import scene*

Сценарій 3: “Налаштувати камеру для огляду”

1. Поворот — лівою кнопкою
2. Панорамування — правою
3. Рух по осі — колесо натиснуте + рух
4. Zoom — колесо
5. Scene → *Reset camera* при потребі

Сценарій 4: “Додати кілька джерел світла”

1. Розмістити камеру в потрібну позицію
2. Lights → *Color*
3. Lights → *Place here*
4. Повторити для інших точок

5. Повідомлення про помилки

Приклади можливих повідомлень системи:

- “Failed to save scene file.”

Немає доступу до шляху при експорті(кириличні символи в шляху).

- Порожня сцена після завантаження

Файл містить MODELS 0 та LIGHTS 0.

Формат файлу .scene (спрощено)

```
CAMERA x y z yaw pitch fov
LIGHTS N
LIGHT t px py pz dx dy dz r g b a intensity
MODELS M
NAME <ім'я>
TEXTURE <шлях або - >
MATERIAL r g b a
MESHES K
VERTICES V
v x y z
INDICES I
i index
```

6. Вирішення типових проблем

Програма не запускається

- Перевірити наявність Qt DLL

Миша не працює

- Клікнути на область перегляду

Файл не завантажується

- Помилка формату
- Непідтримувані символи в шляху

Світло не видно

- Модель може бути поза межами освітлення

Текстура не застосовується

- Спробувати *.png* або *.jpg*
- Рекомендується розмір зображення до 256 КБ

7. Технічна підтримка

З питаннями звертайтеся на email: pkurishchenkoukr.net05@gmail.com

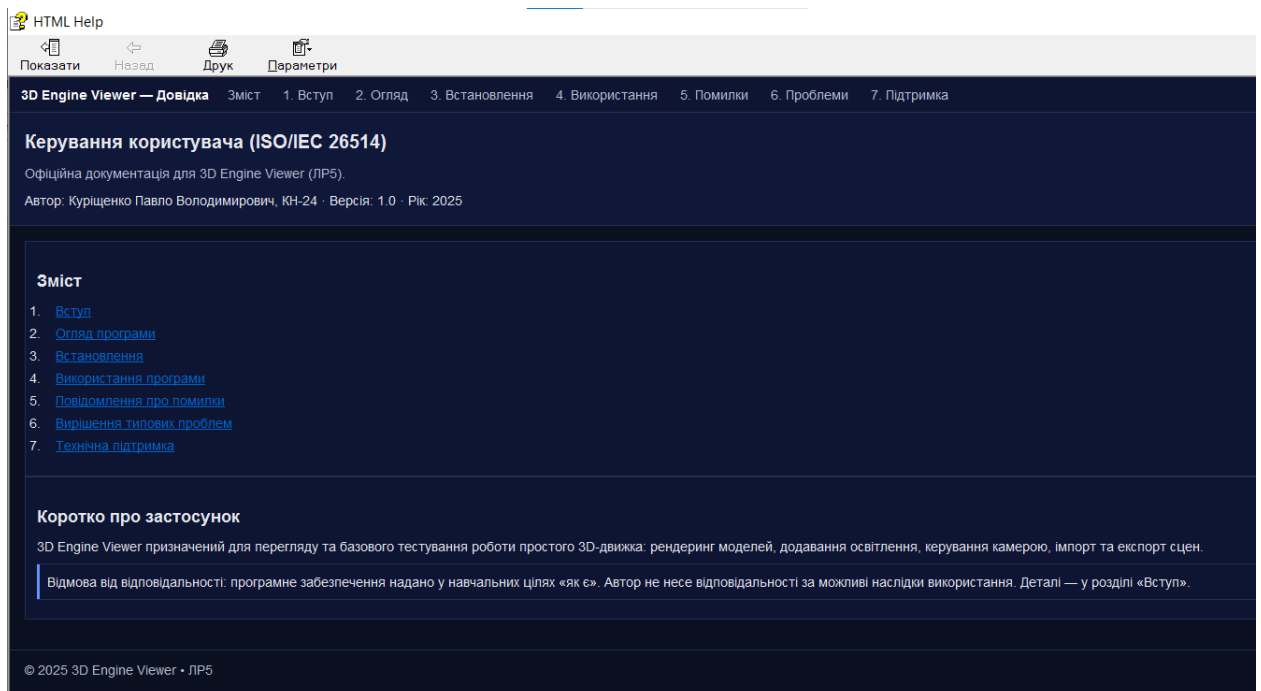


Рис. 1 – Сторінка «Зміст»

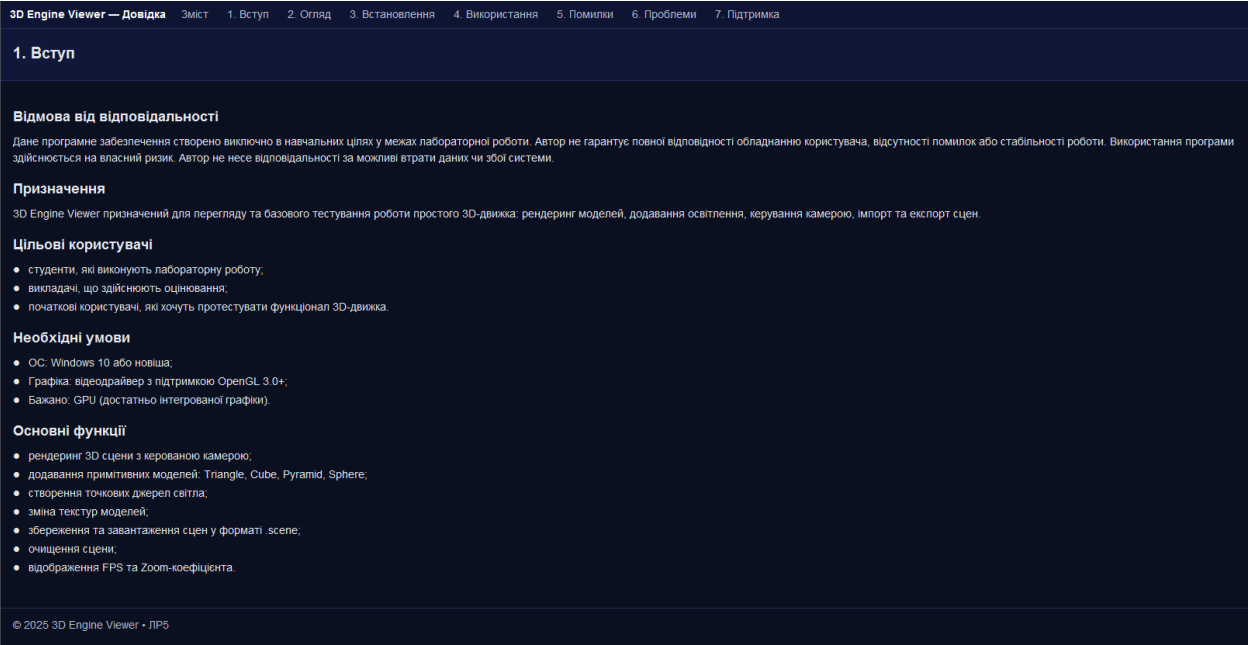


Рис. 2 – Сторінка «Вступ»

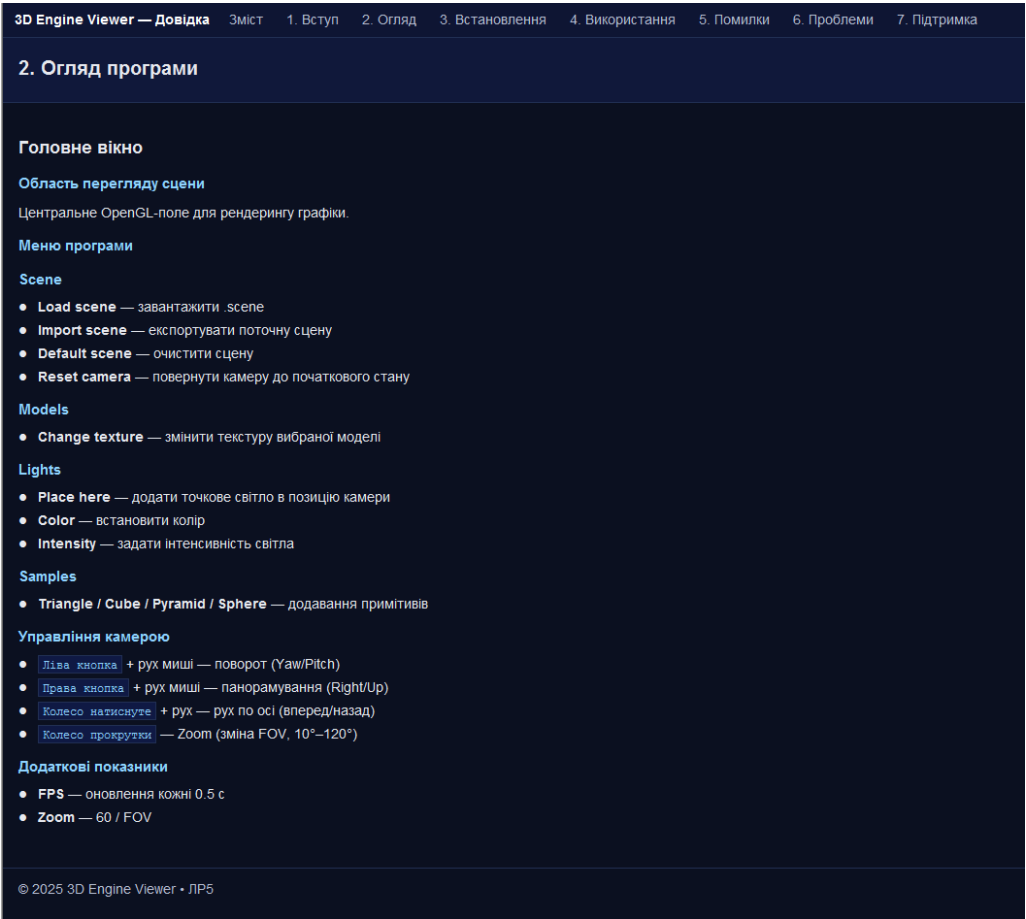


Рис. 3 – Сторінка «Огляд»

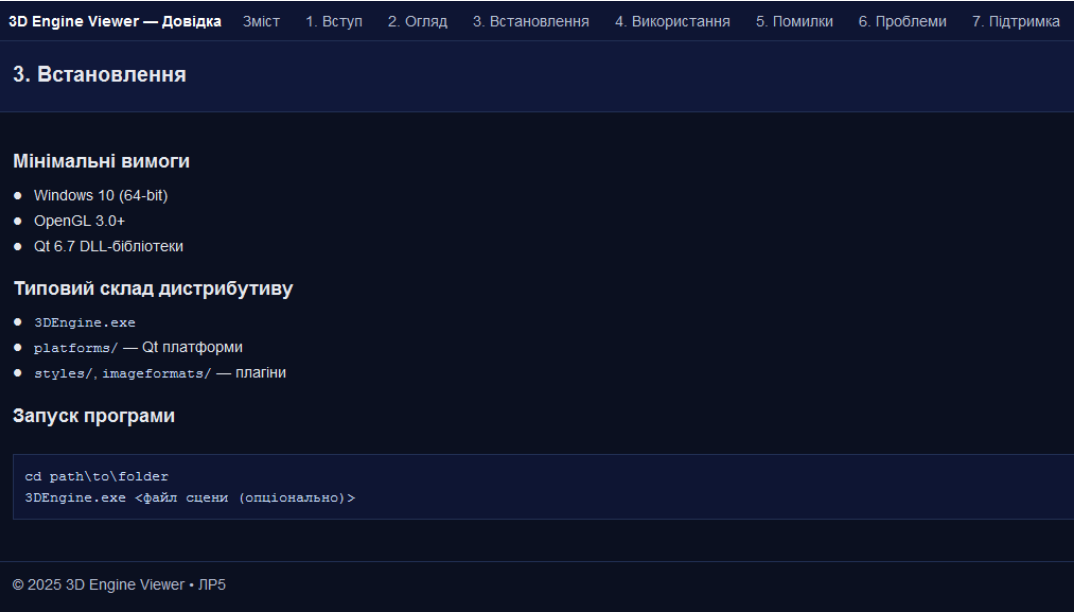


Рис. 4 – Сторінка «Встановлення»

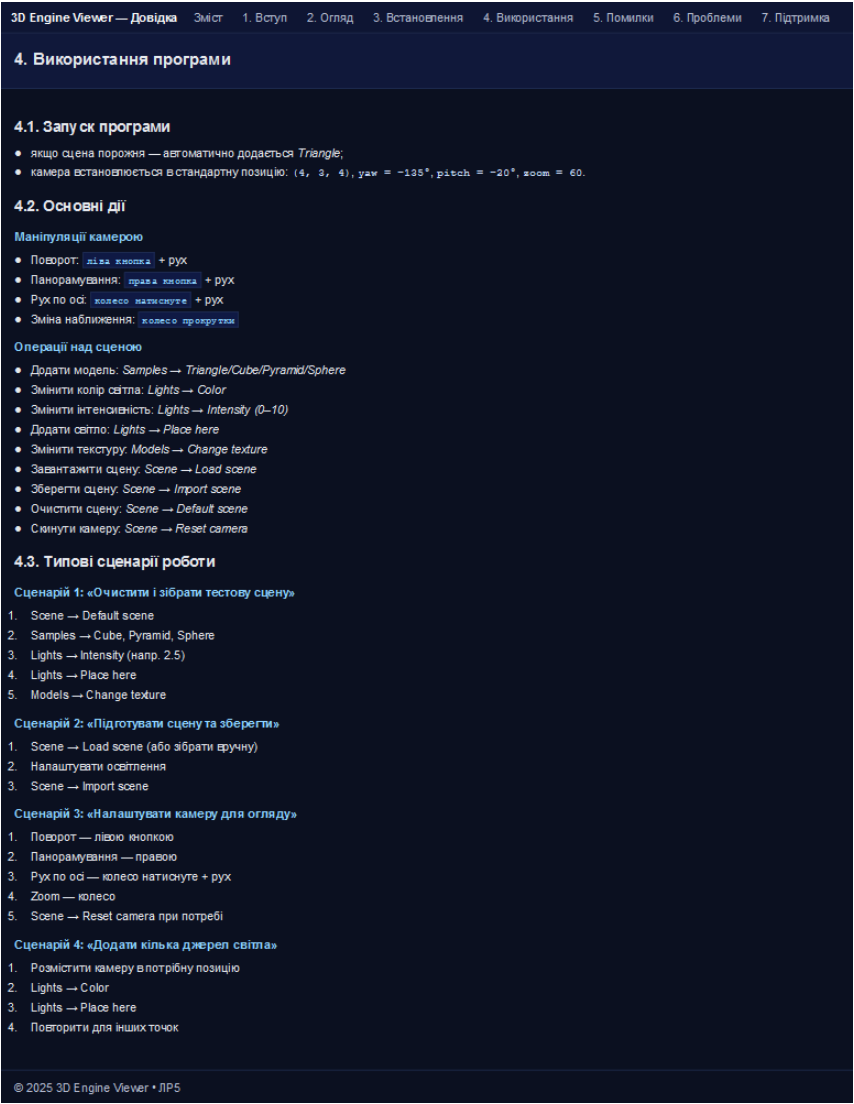


Рис. 5 – Сторінка «Використання»

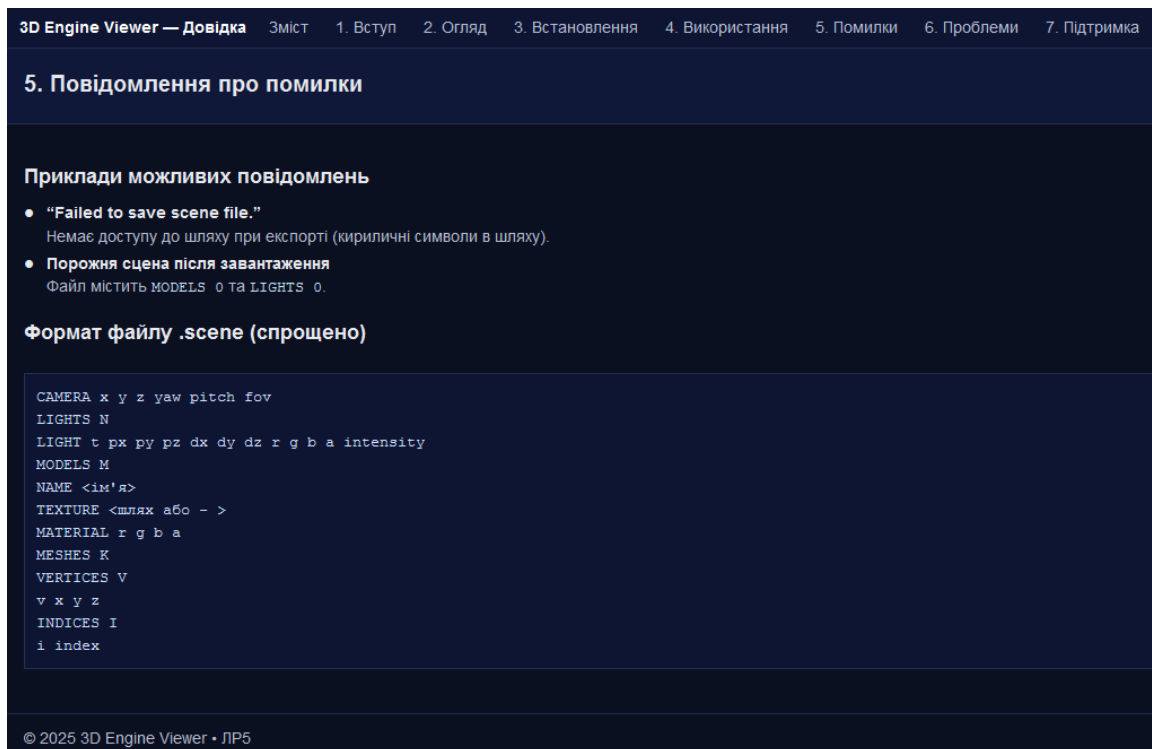


Рис. 6 – Сторінка «Помилки»

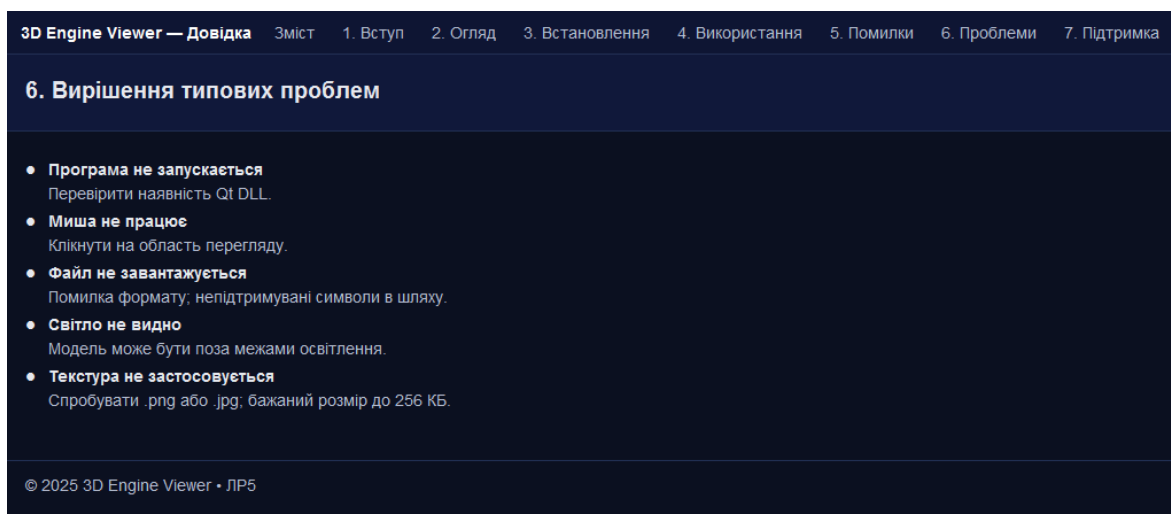


Рис. 7 – Сторінка «Проблеми»

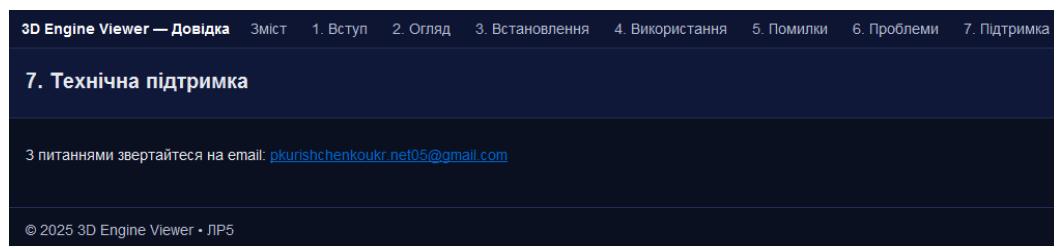


Рис. 8 – Сторінка «Підтримка»

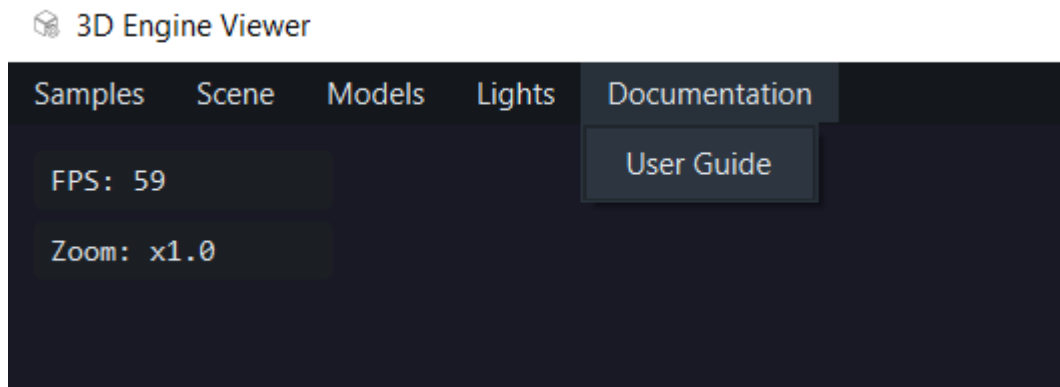


Рис. 9 – Інтеграція в програму

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Навіщо необхідна довідкова система в програмі?

Довідкова система потрібна для:

- надання користувачеві **оперативної та структурованої інформації** про роботу програми;
- пояснення **функцій, параметрів, меню та інтерфейсу**;
- зменшення навантаження на розробника та технічну підтримку;
- **підвищення зручності використання** програмного забезпечення;
- швидкого вирішення типових проблем без сторонньої допомоги;
- забезпечення відповідності програмного продукту **стандартам якості та документації**.

Таким чином довідкова система є обов'язковим елементом будь-якого програмного забезпечення, яке має інтерфейс або складний функціонал.

2. Які відомості потрібні в довідковій системі, а які ні?

У довідковій системі повинні бути:

- призначення програми та сфера застосування;
- опис інтерфейсу та можливостей;
- типові сценарії роботи;

- вимоги до системи та встановлення;
- повідомлення про помилки та способи їх вирішення;
- контактна інформація або посилання на підтримку.

У довідковій системі НЕ повинно бути:

- вихідного коду програми або внутрішньої реалізації алгоритмів;
- інформації для розробників (архітектура, UML, API);
- матеріалів, не пов'язаних напряду з роботою програми;
- документів планування, технічних завдань, специфікацій;

Уся інформація має бути **корисною та компактною**.

3. На Вашу думку, який оптимальний розмір довідкової системи?

(Обґрунтувати)

Оптимальний розмір залежить від складності програми, але для типової навчальної або лабораторної роботи він повинен бути:

5–10 сторінок тексту у форматі СНМ або HTML-сторінок.

Обґрунтування:

- менший обсяг (1–3 сторінки) — недостатній для покриття функцій;
- надто великий (>20 сторінок) — перевантажує користувача та суперечить принципам ISO/IEC 26514 про компактність та орієнтацію на завдання;
- 5–10 сторінок достатньо, щоб:
 - описати інтерфейс,
 - пояснити використання,
 - дати інструкції та вирішення проблем,
 - не перевантажити читача.

4. Які існують формати довідкових систем?

Найпоширеніші формати довідкових систем:

1. CHM (Compiled HTML Help) - класичний Windows-формат, створюється через HTML Help Workshop;

- зручна навігація, зміст, індекс, пошук.

2. HTML / Web Help - набір HTML-сторінок, доступних у браузері (локально або онлайн);

- сучасний і універсальний варіант.

3. PDF - статичний документ, часто використовується як інструкція користувача;

- легко розповсюджується, але не інтерактивний.

4. Markdown - застосовується в GitHub, Confluence, GitLab, DokuWiki.