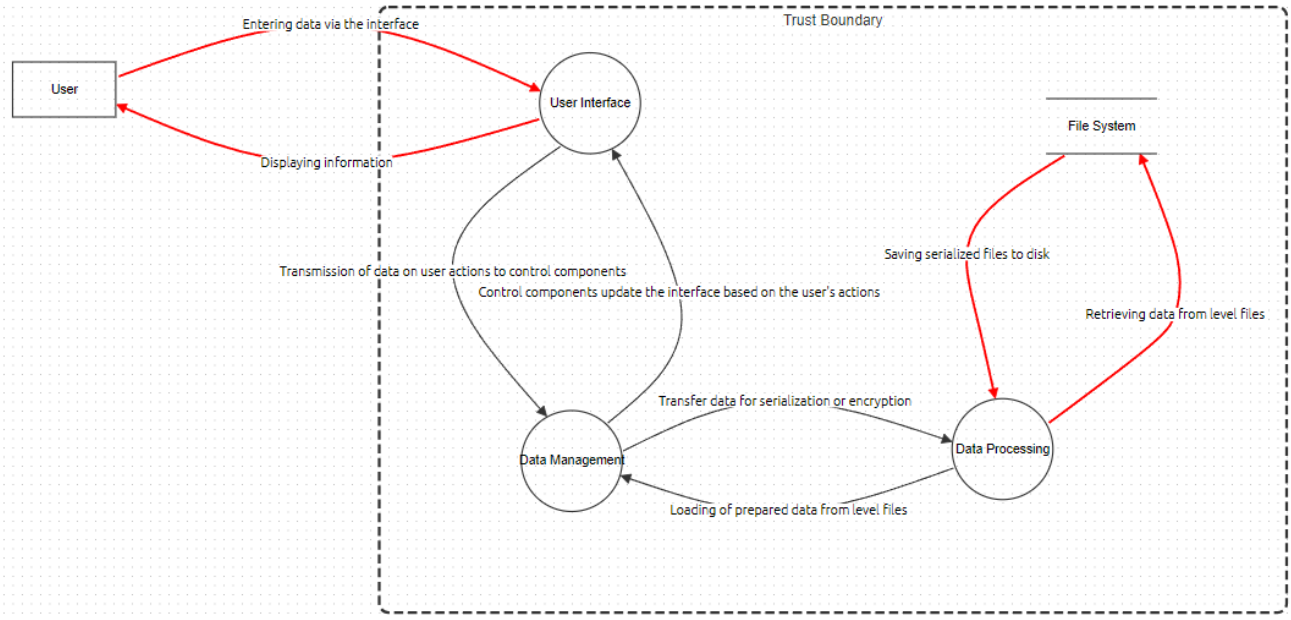

Рисунок 1 – Вигляд таблиці в OWASP Threat Dragon.

Для своєї роботи програма використовує комп’ютерний застосунок та для повноцінного функціонування не потребує підключення до мережі інтернет через те, що має монолітну архітектуру. Монолітний у цьому контексті означає зібраний у єдине ціле. Компоненти програми пов’язані та взаємозалежні, а не мають слабку зв’язаність, як у випадку модульних програм. Замість використання бази даних, найкращим рішенням стало використання локального сховища комп’ютера для імпорту усіх необхідних даних для розробки, так як основна взаємодія з інформацією відбувається у файлах ігрових рівнів, тому для збереження цих файлів буде раціональніше використовувати локальне сховище комп’ютера, а не хмарне сховище бази даних.

Опис зв’язків між сутностями:

1. User ↔ User Interface:

- User → User Interface: Користувач вводить дані через інтерфейс (наприклад, створює новий рівень, вибирає асети).
- User Interface → User: Інтерфейс надає користувачеві інформацію (наприклад, відображає сцену або інструменти для редагування).

2. User Interface ↔ Data Management:

- User Interface → Data Management: Інтерфейс передає дані про дії користувача в керуючі компоненти (наприклад, користувач додає об'єкт на сцену, і ця інформація передається менеджеру сцени).
- Data Management → User Interface: Компоненти керування оновлюють інтерфейс залежно від дій користувача (наприклад, оновлення відображення сцени після додавання об'єкта).

3. Data Management ↔ Data Processing:

- Data Management → Data Processing: Компоненти, що керують, передають дані в систему обробки для серіалізації та шифрування (наприклад, коли користувач зберігає рівень, сцену й об'єкти передають на обробку).
- Data Processing → Data Management: Система обробки повертає результат (наприклад, підтвердження успішного збереження або завантаження рівня).

4. Data Processing ↔ File System (Data Storage):

- Data Processing → File System: Обробка даних зберігає серіалізовані файли на диск (наприклад, рівні, асети, звуки).
- File System → Data Processing: Під час завантаження рівня система обробки отримує дані з локальної файлової системи (наприклад, для завантаження ассетів).

No загрози - ID	Назва зв’язку (елемент Interaction з екрану аналізу)	STRIDE-клас загрози	Назва загрози (елемент Title з екрану аналізу)	Опис загрози (елемент Description з екрану аналізу)
1 - 15	Retrieving data from level files	Spoofing	Spoofing of Destination Data Store File System	File System may be spoofed by an attacker and this may lead to data being written to the

No загрози - ID	Назва зв'язку (елемент Interaction з екрану аналізу)	STRIDE-клас загрози	Назва загрози (елемент Title з екрану аналізу)	Опис загрози (елемент Description з екрану аналізу)
				attacker's target instead of File System. Consider using a standard authentication mechanism to identify the destination data store.
2 - 16	Retrieving data from level files	Denial Of Service	Potential Excessive Resource Consumption for Data Processing or File System	Does Data Processing or File System take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.
3 - 17	Saving serialized files to disk	Spoofing	Spoofing of Destination Data Store File System	File System may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of File System. Consider using a standard authentication mechanism to identify the destination data store.
4 - 18	Saving serialized files to disk	Information Disclosure	Weak Access Control for a Resource	Improper data protection of File System can allow an attacker to read information not intended for disclosure. Review authorization settings.
5 - 26	Displaying information	Repudiation	External Entity Human User Potentially Denies Receiving Data	Human User claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.
6 - 27	Displaying information	Denial Of Service	Data Flow Displaying information Is Potentially Interrupted	An external agent interrupts data flowing across a trust boundary in either direction.

No загрози - ID	Назва зв’язку (елемент Interaction з екрану аналізу)	STRIDE-клас загрози	Назва загрози (елемент Title з екрану аналізу)	Опис загрози (елемент Description з екрану аналізу)
7 - 28	Entering data via the interface	Spoofing	Spoofing the User Interface Process	User Interface may be spoofed by an attacker and this may lead to information disclosure by Human User. Consider using a standard authentication mechanism to identify the destination process.
8 - 32	Entering data via the interface	Denial Of Service	Potential Process Crash or Stop for User Interface	User Interface crashes, halts, stops or runs slowly; in all cases violating an availability metric.
9 - 34	Entering data via the interface	Elevetion Of Privilege	User Interface May be Subject to Elevation of Privilege Using Remote Code Execution	Human User may be able to remotely execute code for User Interface.
10 - 35	Entering data via the interface	Elevetion Of Privilege	Elevation by Changing the Execution Flow in User Interface	An attacker may pass data into User Interface in order to change the flow of program execution within User Interface to the attacker’s choosing.

Опис загроз записаних вище:

1. Spoofing of Destination Data Store (File System): Зловмисник може підмінити файлову систему, змушуючи застосунок записувати дані в інше сховище. Для захисту можна використовувати аутентифікацію під час взаємодії з файловою системою та обмежити доступ за допомогою прав користувача – не реалізовано.

2. Potential Excessive Resource Consumption for Data Processing or File System: Зловмисник може викликати надмірне споживання ресурсів, що призведе до відмови в обслуговуванні. Необхідно обмежити розмір даних, контролювати тайм-аути процесів і використовувати механізми контролю обсягу пам'яті – реалізовано.

3. Spoofing of Destination Data Store (File System): Загроза повторюється, і методи захисту аналогічні – обмеження прав доступу та використання перевірок цілісності даних для запобігання підміни файлової системи – не реалізовано.

4. Weak Access Control for a Resource (File System): Недостатній захист файлової системи може дозволити атакуючому прочитати або змінити дані. Потрібно налаштовувати правильні права доступу або використовувати шифрування для захисту важливих даних – реалізовано.

5. External Entity Human User Potentially Denies Receiving Data: Користувач може заперечувати отримання даних. Такого відбутися не може, у коді опрацьовані усі можливі випадки пов’язані з такою вразливістю.

6. Data Flow Displaying Information Is Potentially Interrupted: Потік даних може бути перерваний зовнішнім агентом. Хоча це малоймовірно для офлайн-додатків, необхідно забезпечити надійну обробку помилок і відновлення критичних процесів – можливо лише якщо зловмисник має доступ до комп’ютера користувача.

7. Spoofing the User Interface Process: Атакуючий може підмінити процес інтерфейсу користувача, що призведе до витоку даних. Щоб цього уникнути потрібно використовувати аутентифікацію процесів і контроль цілісності файлів інтерфейсу – реалізовано на половину, для додатку не передбачалося створення функції аутентифікації процесів або користувача.

8. Potential Process Crash or Stop for User Interface: Інтерфейс може зависнути або зупинитися через перевантаження, порушуючи доступність. Для захисту слід використовувати багатопоточність і механізми відновлення інтерфейсу після помилок – не реалізовано.

9. User Interface May be Subject to Elevation of Privilege Using Remote Code Execution: Зловмисник може виконати довільний код через інтерфейс. Виконання цієї вразливості можливе лише тоді, коли у зловмисника є доступ до комп’ютера користувача, наприклад DLL Inject.

10. Elevation by Changing the Execution Flow in User Interface: Атакуючий може змінити логіку виконання програми через інтерфейс. Важливо валідувати вхідні дані та використовувати механізми контролю цілісності коду – реалізовано.