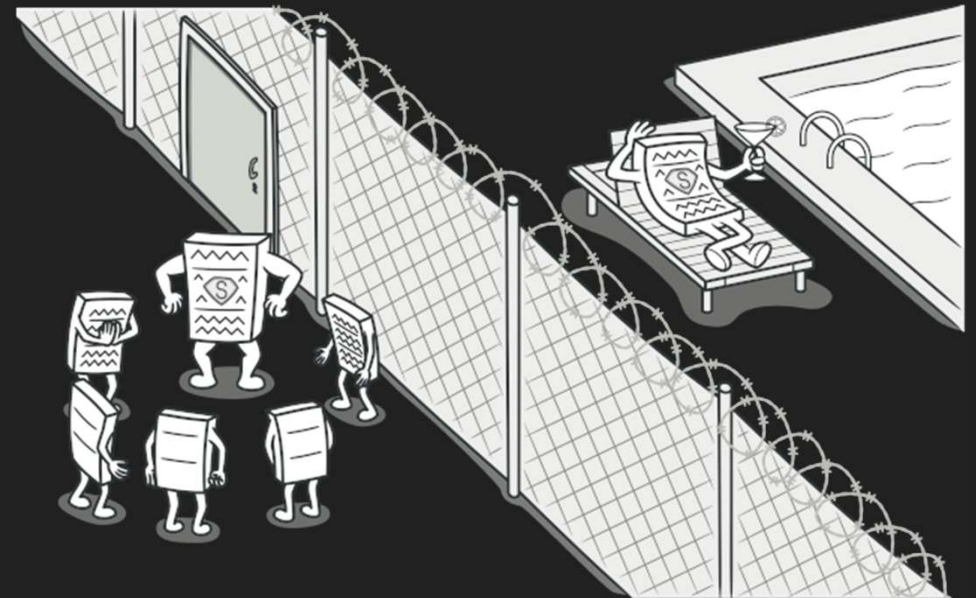


Замісник

Також відомий як: **Proxy, Surrogate**

Суть патерну

- Структурний патерн, що дає змогу підставляти замість реальних об'єктів спеціальні об'єкти-замінники. Ці об'єкти перехоплюють виклики до оригінального об'єкта, дозволяючи зробити щось до чи після передачі виклику оригіналові.

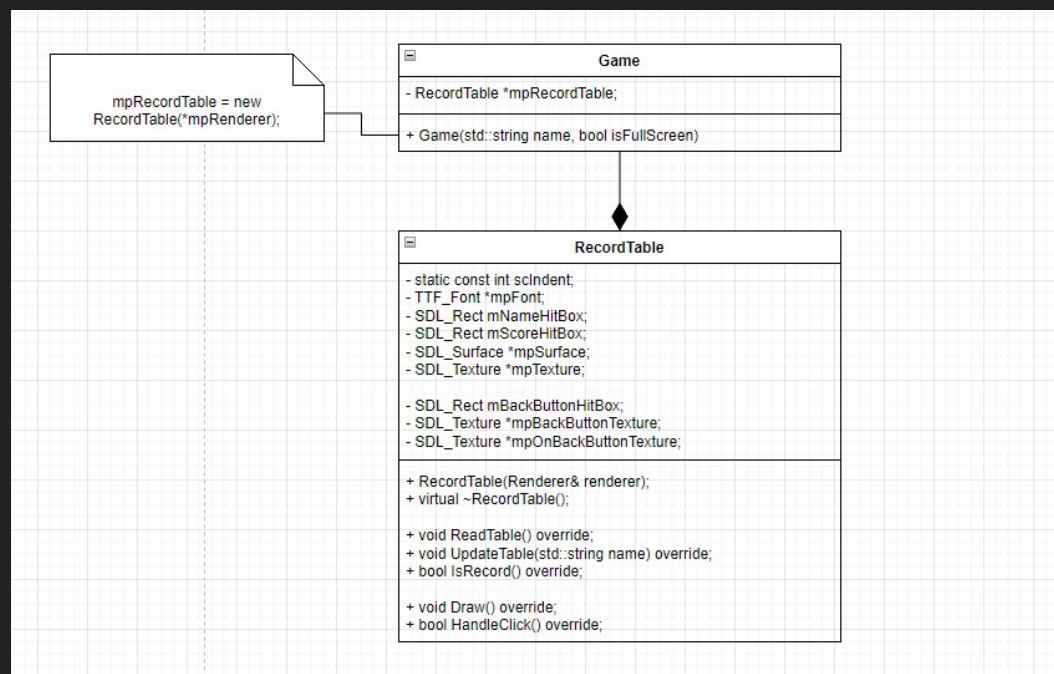


Розглянемо проблему

- В моїй грі всі текстури та екрани ініціалізуються при запуску. Користувач може навіть не дійти до деяких із них, але ресурси все ж витрачаються. Чи можливо створити об'єкт екрана тільки тоді коли він потрібний та залишити той самий інтерфейс, щоб не змінювати код в усіх місцях використання?

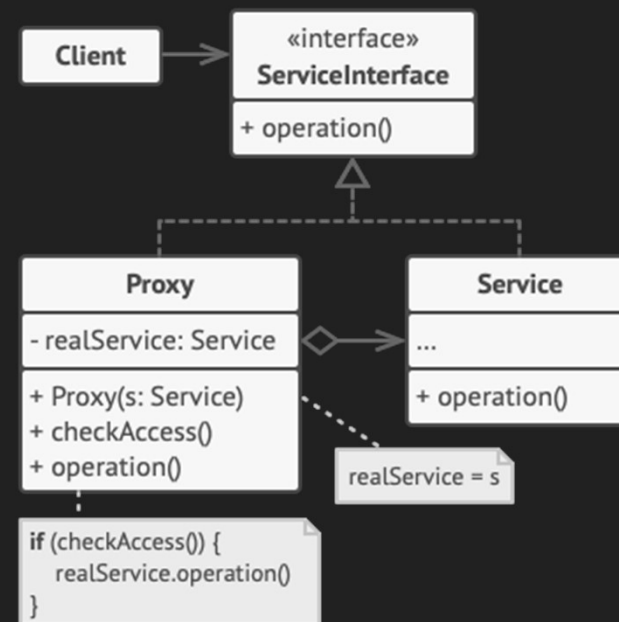


Діаграма класів



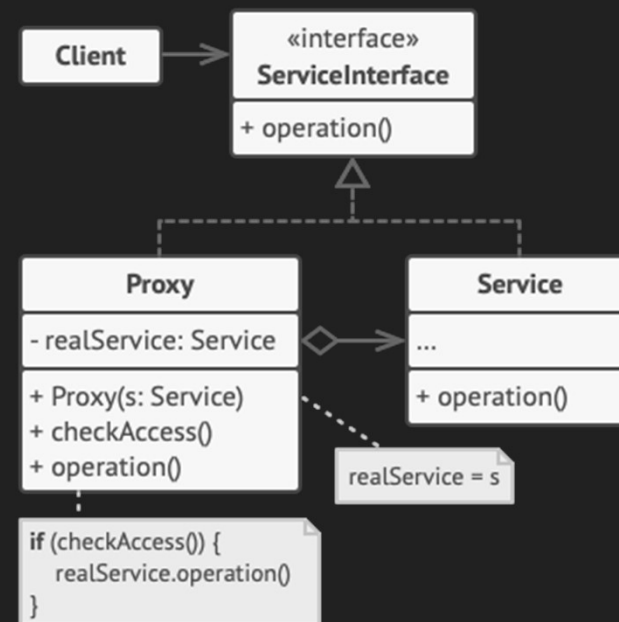
Що пропонує патерн Замісник

- **Інтерфейс сервісу** визначає загальний інтерфейс для сервісу й замісника. Завдяки цьому об'єкт замісника можна використовувати там, де очікується об'єкт сервісу.
- **Замісник** зберігає посилання на об'єкт сервісу. Після того, як замісник закінчує свою роботу (наприклад, ініціалізацію, логування, захист або інше), він передає виклики вкладеному сервісу.

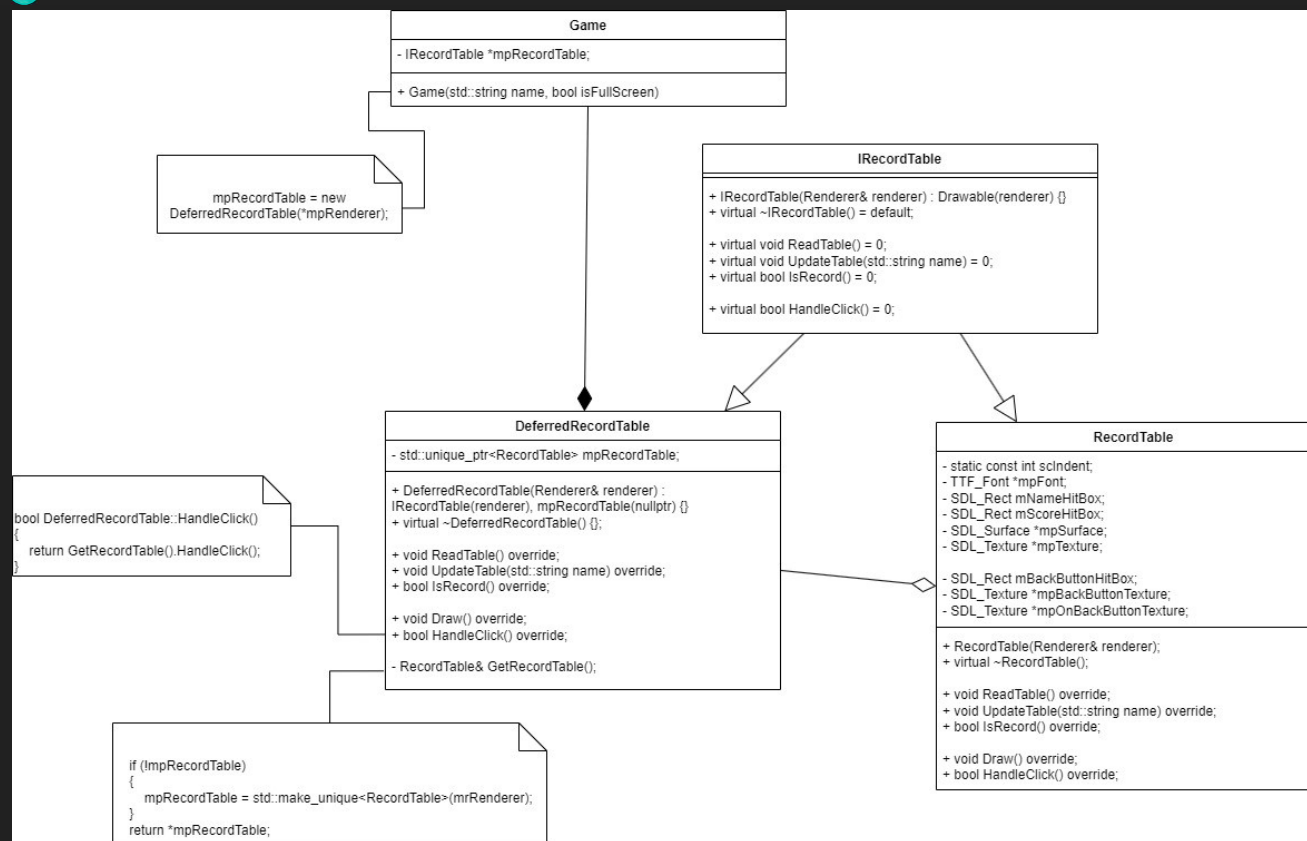


Що пропонує патерн Замісник

- Клієнт працює з об'єктами через інтерфейс сервісу. Завдяки цьому його можна «обдурити», підмінивши об'єкт сервісу об'єктом замісника.



Застосуємо патерн до нашої гри



Переваги та недоліки патерну

Переваги

- Дозволяє контролювати сервісний об'єкт непомітно для клієнта.
- Може працювати, навіть якщо сервісний об'єкт ще не створено.

Недоліки

- Ускладнює програмний код
- Збільшує час отримання відклику від сервісу.

Можливі застосування

- Лінива ініціалізація (віртуальний проксі)
- Захист доступу (захисний проксі)
- Локальний запуск сервісу (віддалений проксі)
- Логування запитів (логуючий проксі)
- Кешування об'єктів («розумне» посилання)

Зв'язок з іншими патернами

- **Адаптер** надає альтернативний інтерфейс, тоді коли **Замісник** надає той самий інтерфейс
- **Декоратор** має схожу структуру, але інше призначення. **Декоратор** додає нові обов'язки, в той час, як **Замісник** контролює доступ до сервісного об'єкта. Також **Замісник** сам керує життям сервісного об'єкта, а обгортання **Декораторів** контролюється клієнтом.