**Facade**

Есть игровые движки, где вся работа создания приложения скрыта, то есть программист просто пишет код, затем нажимает на кнопку запуска, и готово, у него есть готовое приложение.  
Задача реализовать эту систему. Причем вся система должна быть скрыта от пользователя. Он только нажимает на кнопку запуска. То есть нам нужен класс, который будет реализовывать всю работу создания приложения, а пользователь это не видит. У него есть лишь доступ к этому классу. Ну и вся система будет разбита на подсистемы, чтобы сделать все структурированно, а не все в кучу. Таким образом, есть система, разбитая на подсистемы, чью работу пользователь не видит, но он взаимодействует с ней через специально выделенный класс для работы с системой. Никакой абстракции, наследования, древовидной структуры, нет динамического подключения новых функций. Собственно, это явно паттерн фасад. Что нам и нужно.

Фасад (Facade) представляет шаблон проектирования, который позволяет скрыть сложность системы с помощью предоставления упрощенного интерфейса для взаимодействия с ней.

Когда использовать фасад?

Когда имеется сложная система, и необходимо упростить с ней работу. Фасад позволит определить одну точку взаимодействия между клиентом и системой.

Когда надо уменьшить количество зависимостей между клиентом и сложной системой. Фасадные объекты позволяют отделить, изолировать компоненты системы от клиента и развивать и работать с ними независимо.

Все, что доктор прописал для нашей задачи.  
  
Иные паттерны:  
Декоратор (Decorator) представляет структурный шаблон проектирования, который позволяет динамически подключать к объекту дополнительную функциональность.  
Паттерн Адаптер (Adapter) предназначен для преобразования интерфейса одного класса в интерфейс другого. Благодаря реализации данного паттерна мы можем использовать вместе классы с несовместимыми интерфейсами.  
Паттерн Компоновщик (Composite) объединяет группы объектов в древовидную структуру по принципу "часть-целое и позволяет клиенту одинаково работать как с отдельными объектами, так и с группой объектов.  
Паттерн Заместитель (Proxy) предоставляет объект-заместитель, который управляет доступом к другому объекту. То есть создается объект-суррогат, который может выступать в роли другого объекта и замещать его.  
Мост (Bridge) - структурный шаблон проектирования, который позволяет отделить абстракцию от реализации таким образом, чтобы и абстракцию, и реализацию можно было изменять независимо друг от друга.