Разработка программного модуля для функционального тестирования программ на платформе Unity

Руководитель,
к.пед.н., доцент
каф. ЭВМ
Плаксина Ю.Г.

Выполнил: студент
группы КЭ-406
Клюшин М. А.

Актуальность

В настоящее время при проведении тестирования специалистам по тестированию необходимо тратить время на рутинные задачи, такие как проверка вручную корректности работы всех объектов и компонентов, проверка старых сценариев тестирования, написание кода для новых сценариев тестирования.

В то же время многие задачи тестирования которые выполняются вручную могут быть автоматизированы, что приведет к сокращению времени на формирование и проверку сценариев тестирования.

Цель и задачи

Целью выпускной квалификационной работы является разработать инструмент для функционального тестирования программ, позволяющий создание тестов без использования средств языков программирования и обеспечивающий автоматическую проверку тестов.

Задачи:

1. Аналитический обзор научно-технической, нормативной и методической литературы по тематике работы.
2. Проектирование программной архитектуры.
3. Разработка программного решения.
4. Проведение тестирования программного решения.

Обзор аналогов

Критерии	AltTest	AutoPlay	Unit Test
Программа или программный модуль для тестирования	+	+	-
Тестирует функционал объектов и компонентов.	+	+	+/-
Автоматическая проверка тестов.	+	-	+
Формирование тестовых сценариев без использования кода.	-	-	-

Функциональные и нефункциональные требования

Функциональные требования:

1. Обнаружить и получить доступ ко всем объектам и компонентам в иерархии сцены Unity.
2. Предоставлять пользователю интерфейс поиска объектов и компонентов.
3. Предоставлять пользователю интерфейс создания сценариев тестирования.
4. Предоставлять пользователю интерфейс проверки сценариев тестирования.

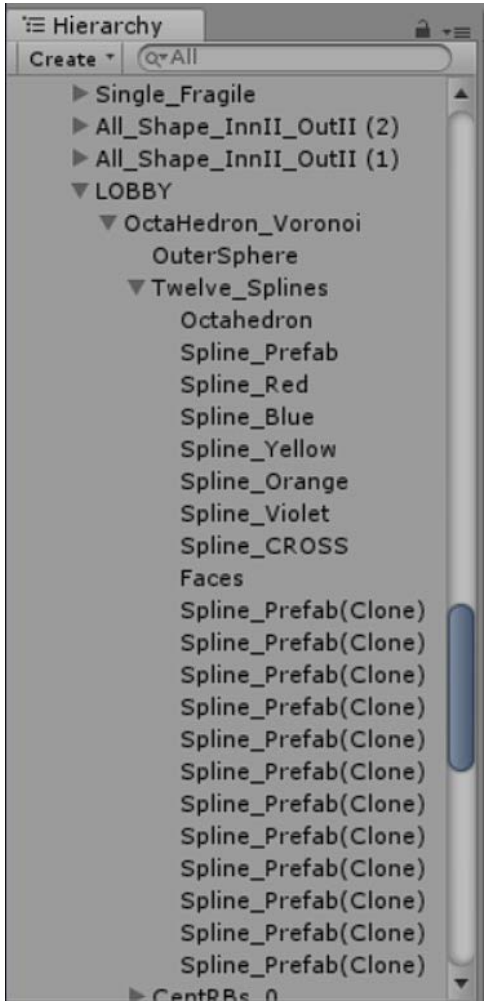
Нефункциональные требования:

1. Реализовать программный модуль в виде NPM (Node Package Manager) пакета для Unity.
2. Предусмотреть архитектурную возможность расширения функционала.
3. Программный модуль интегрируется в платформу Unity в виде отдельного окна инспектора, позволяющего взаимодействовать с его функционалом.
4. В качестве синтаксиса поисковых запросов использовать язык XPath.

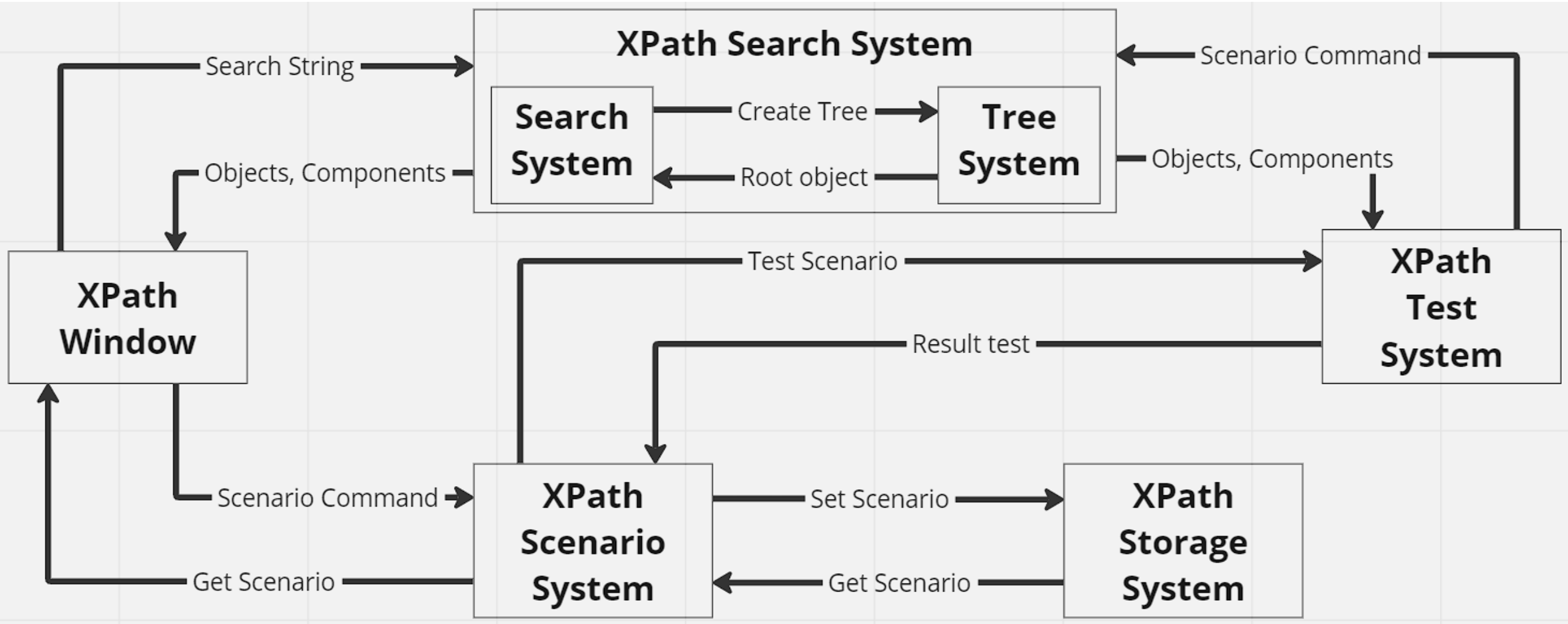
Требования к структуре и функционированию системы

Выделяется четыре функциональные подсистемы:

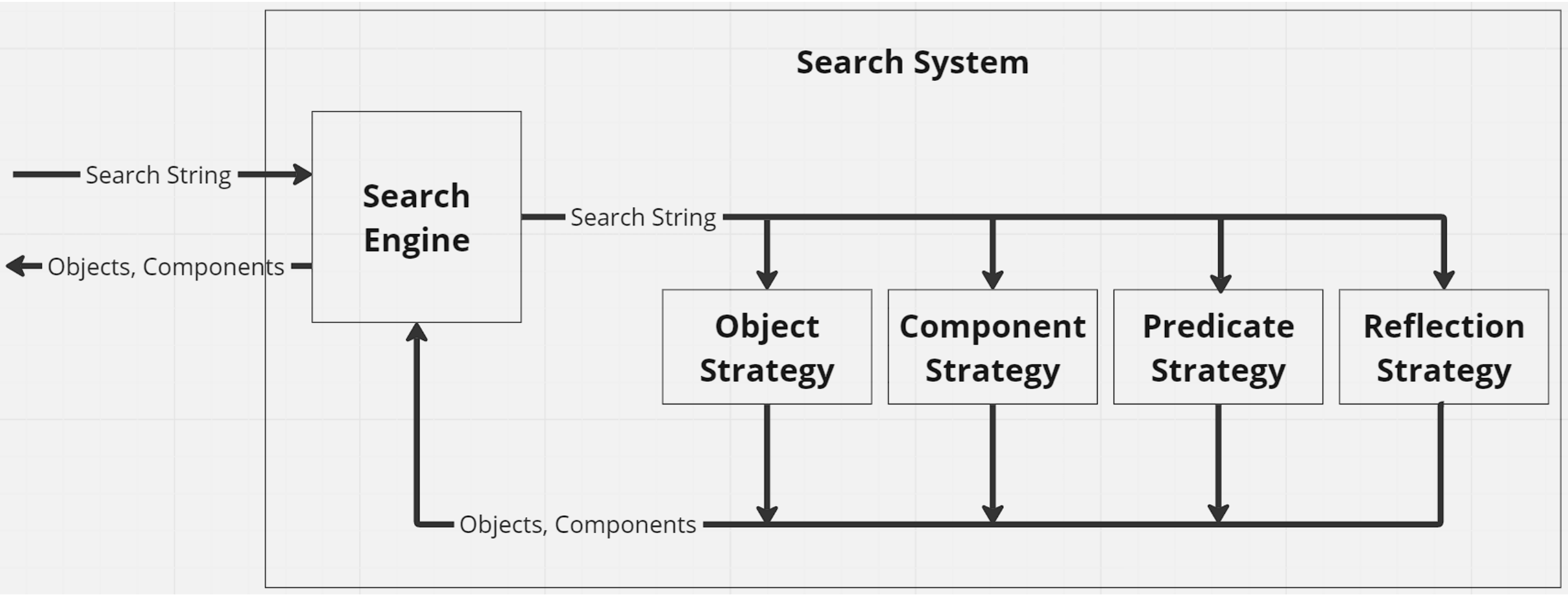
- подсистема поиска и редактирования;
- подсистема формирования сценариев;
- подсистема хранения сценариев;
- подсистема проверки сценариев.



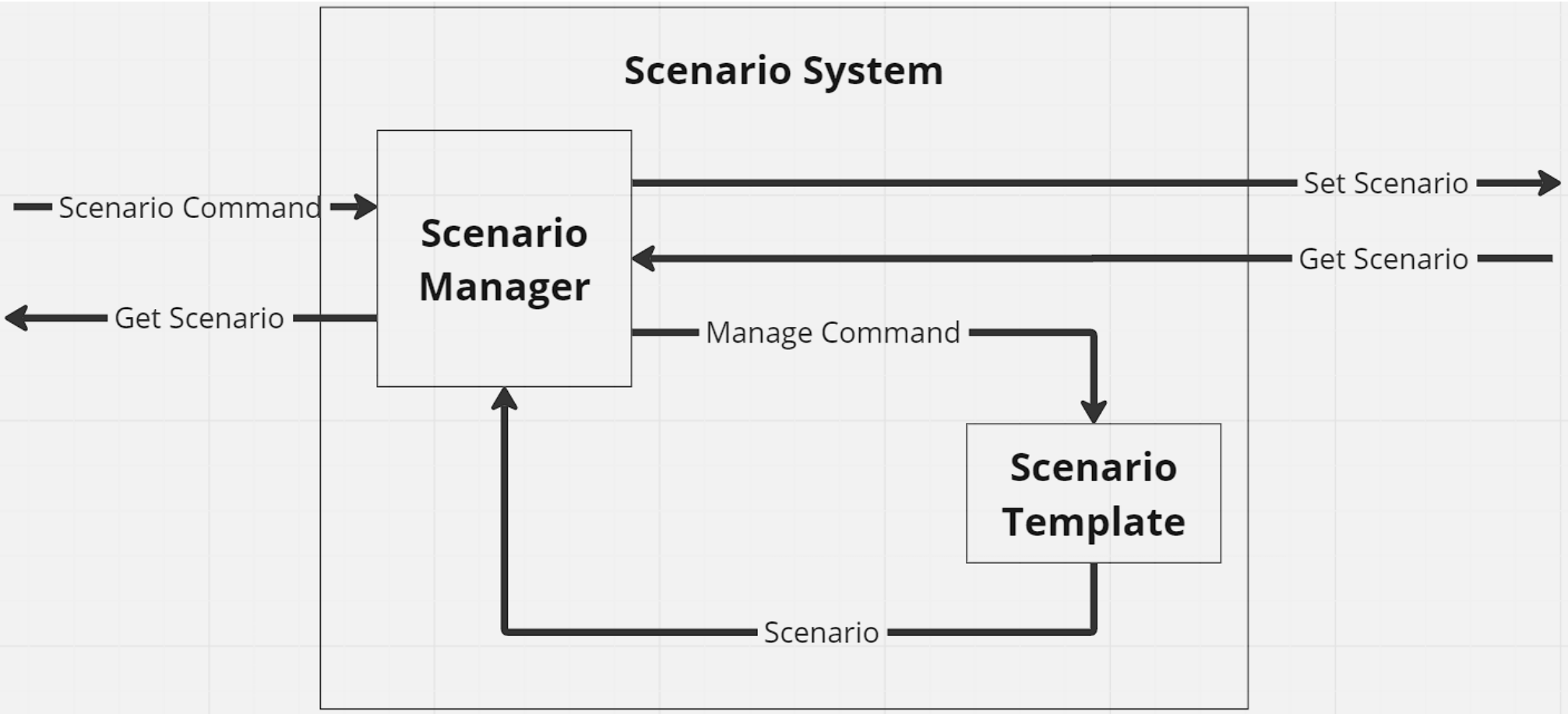
Концептуальная модель программного модуля



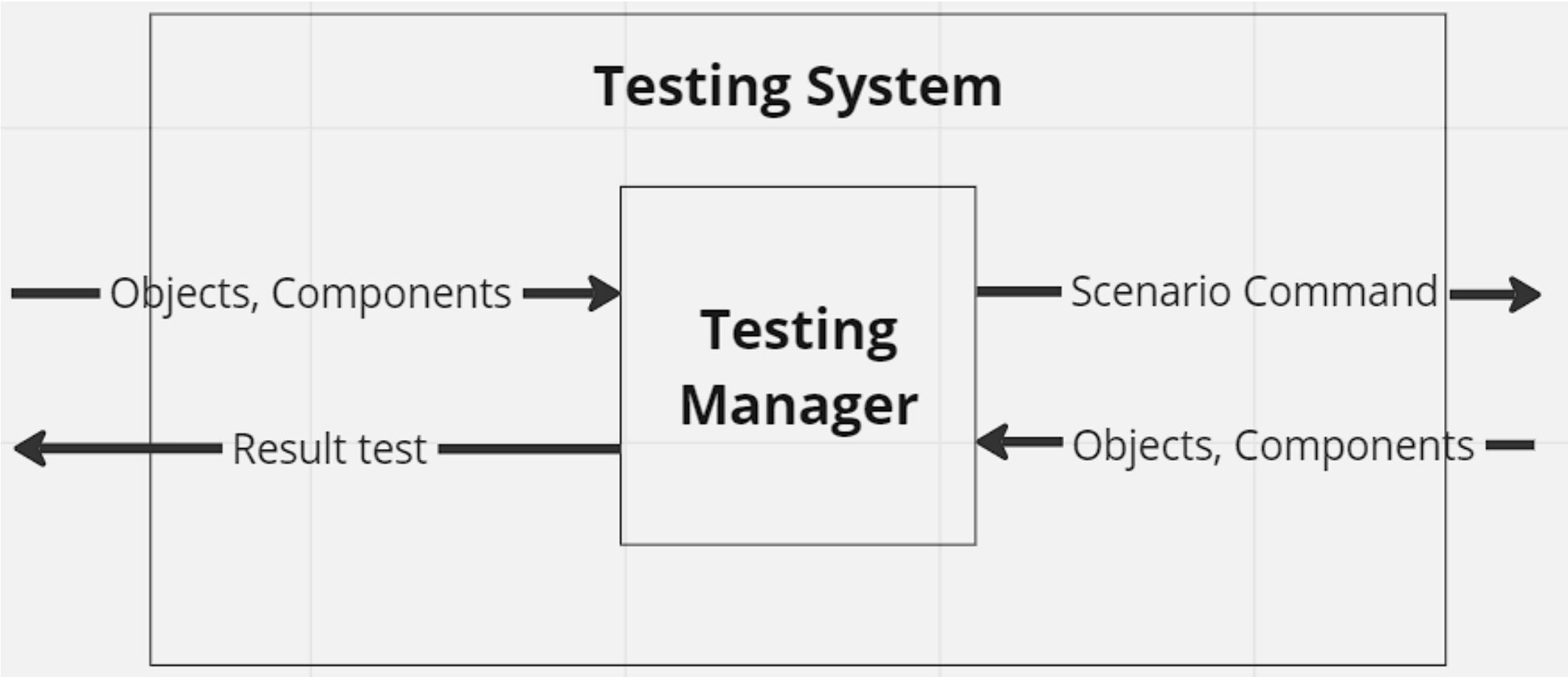
Концептуальная модель подсистемы поиска программного модуля



Концептуальная модель подсистемы формирования сценариев программного модуля



Концептуальная модель подсистемы проверки сценариев программного модуля



Команды поисковых запросов

Команды прямого поиска:

- / – поиск следующего объекта в иерархии;
- /@ – поиск следующего компонента в иерархии.

Команды рекурсивного поиска:

- // – поиск всех объектов в иерархии;
- //@ – поиск всех компонентов в иерархии;
- имя объекта – поиск всех объектов в иерархии с указанным именем;
- @имя компонента – поиск всех компонентов в иерархии с указанным именем;
- * – любое имя объекта или компонента.

Команды поисковых запросов

Для сужения результатов поиска были разработаны команды предикаты, позволяющие установить условия для результатов поиска. Ключевой парой символов обозначающие предикат являются [ ] (квадратные скобки).

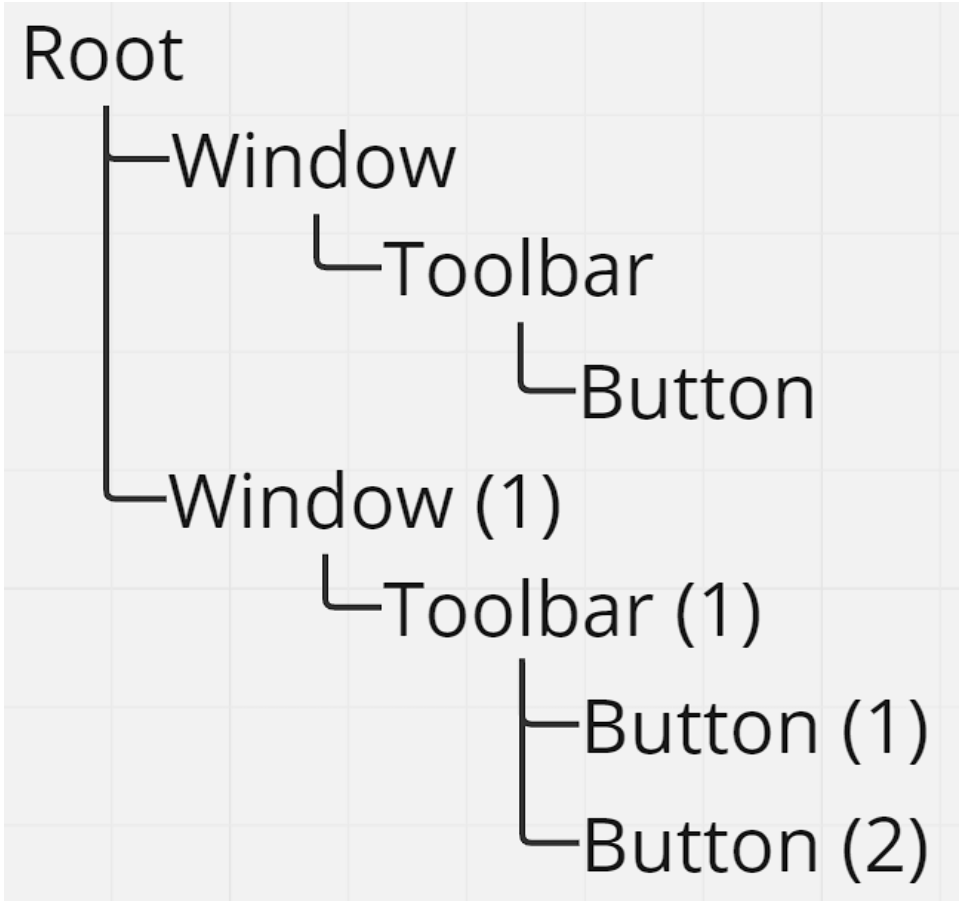
Команды предикаты поиска:

- last() – поиск последнего элемента в результате поиска;
- position() – позволяет обратиться к позиции элемента в результате поиска;
- номер элемента – получение элемента по его позиции в результате поиска;
- <, >, >=, <=, ==, != – операторы сравнения;
- –, + – математические операторы.

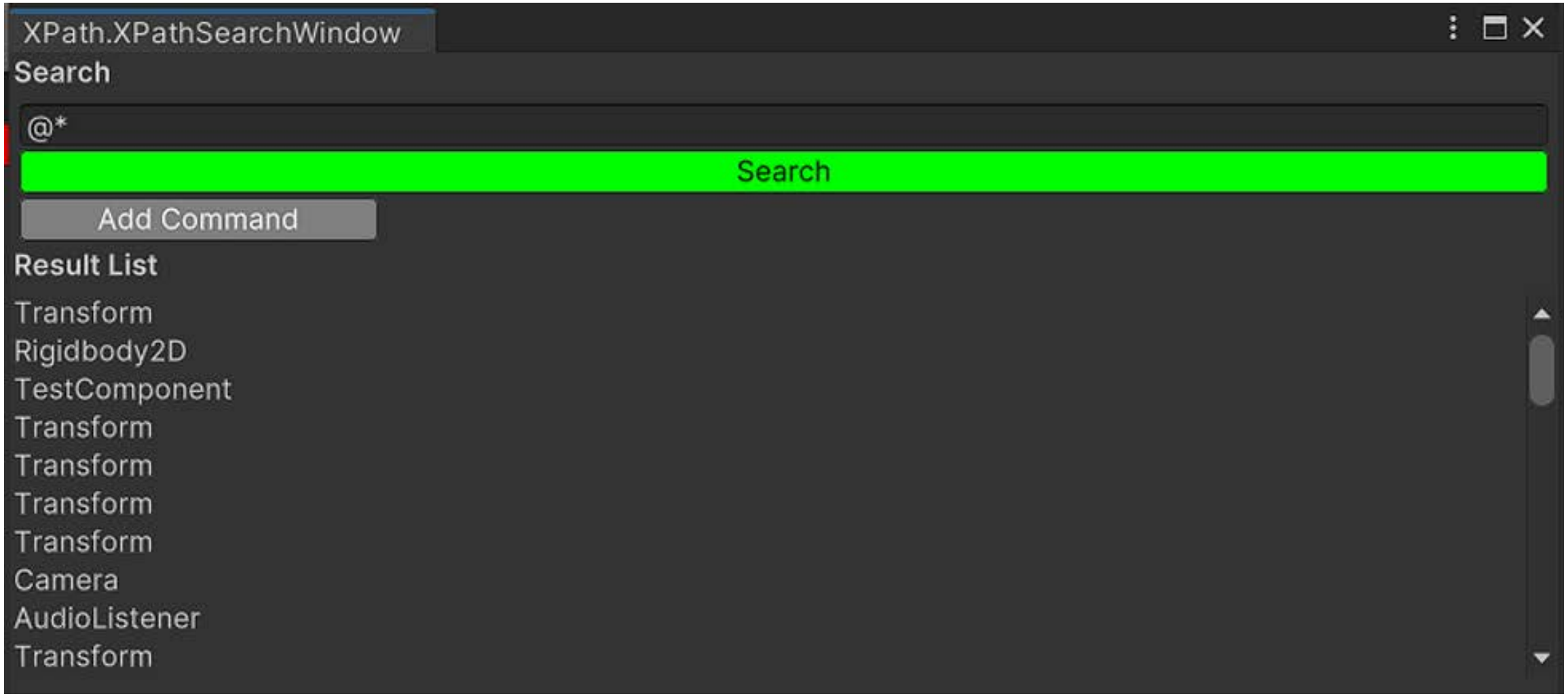
Для вызова функционала Reflection был введен ключевой символ . (точка)

Примеры поисковых запросов

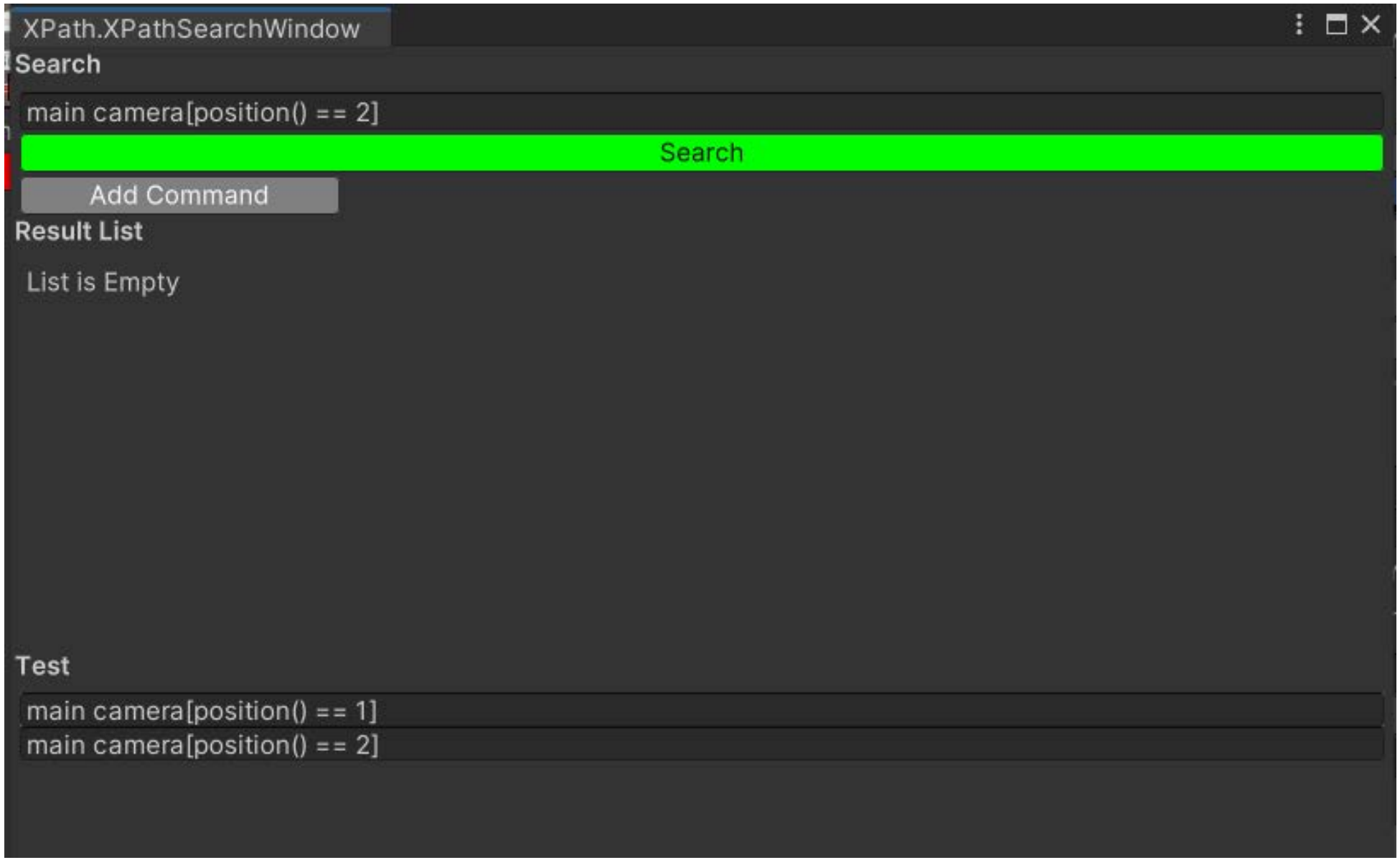
- Window/Toolbar/Button
- /Window (1)/Toolbar/Button (1)
- //@Transform
- Window/@Transform[1]
- Window/@Transform[last() - 1]
- Window/Toolbar/Button[position() < 3]
- Window/Toolbar/Button.Click()



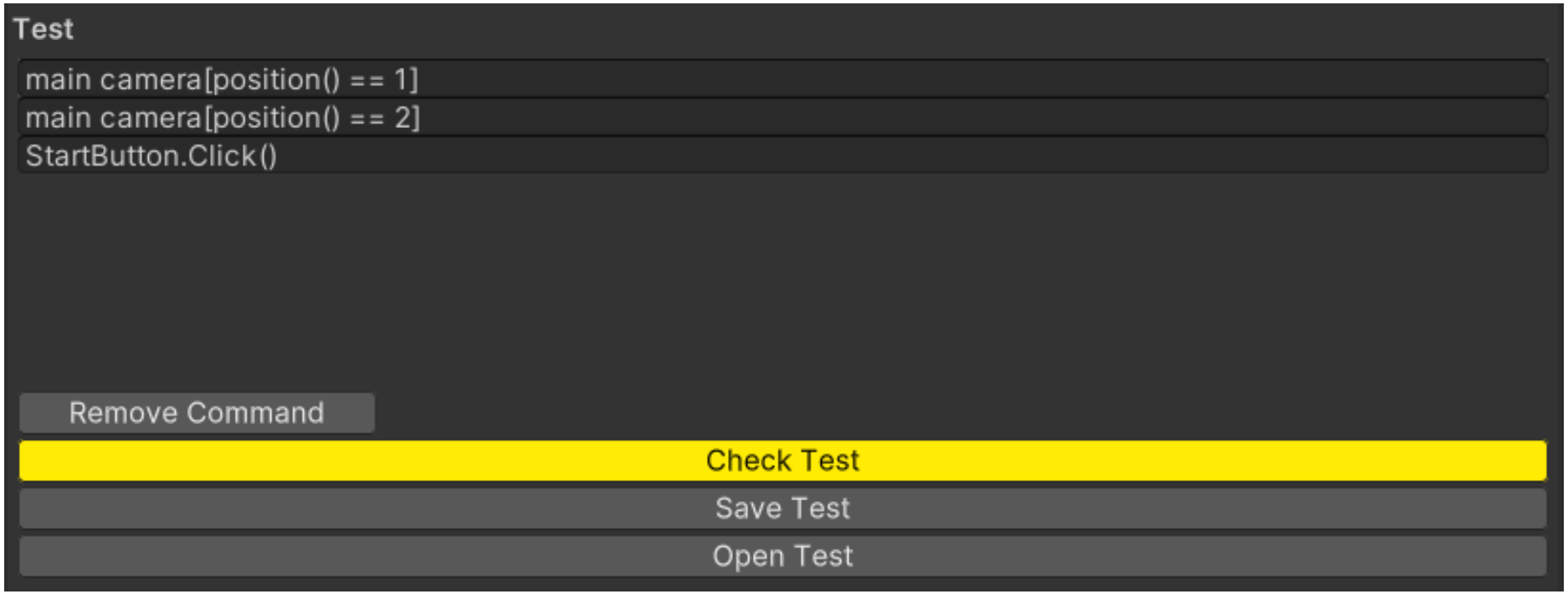
Выполнение поискового запроса

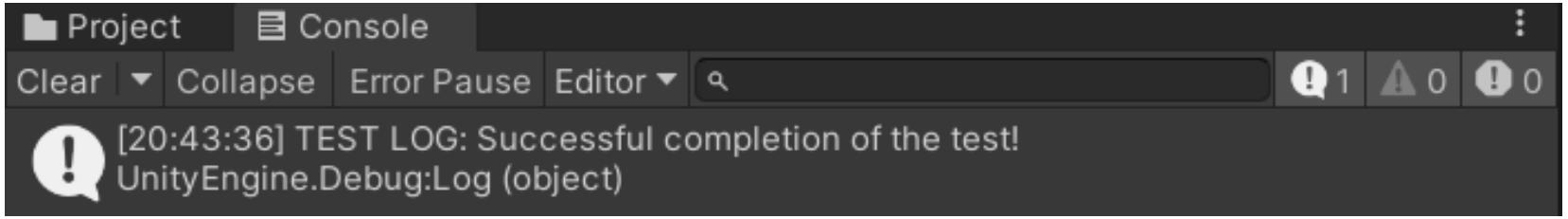


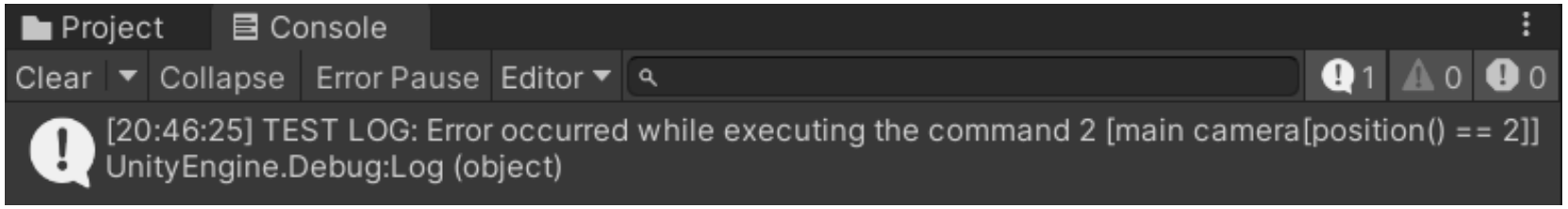
Добавление запроса в тест



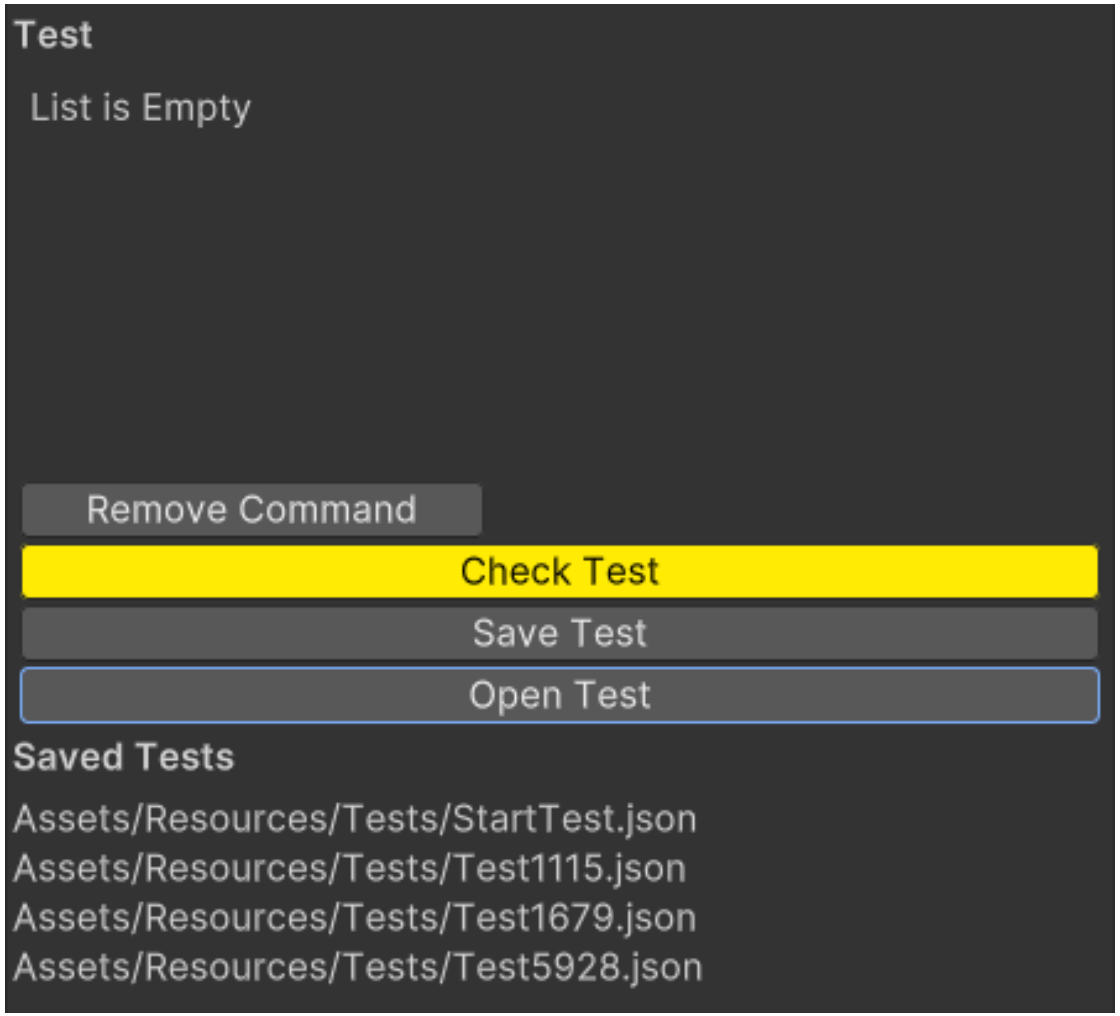
Выполнение проверки теста

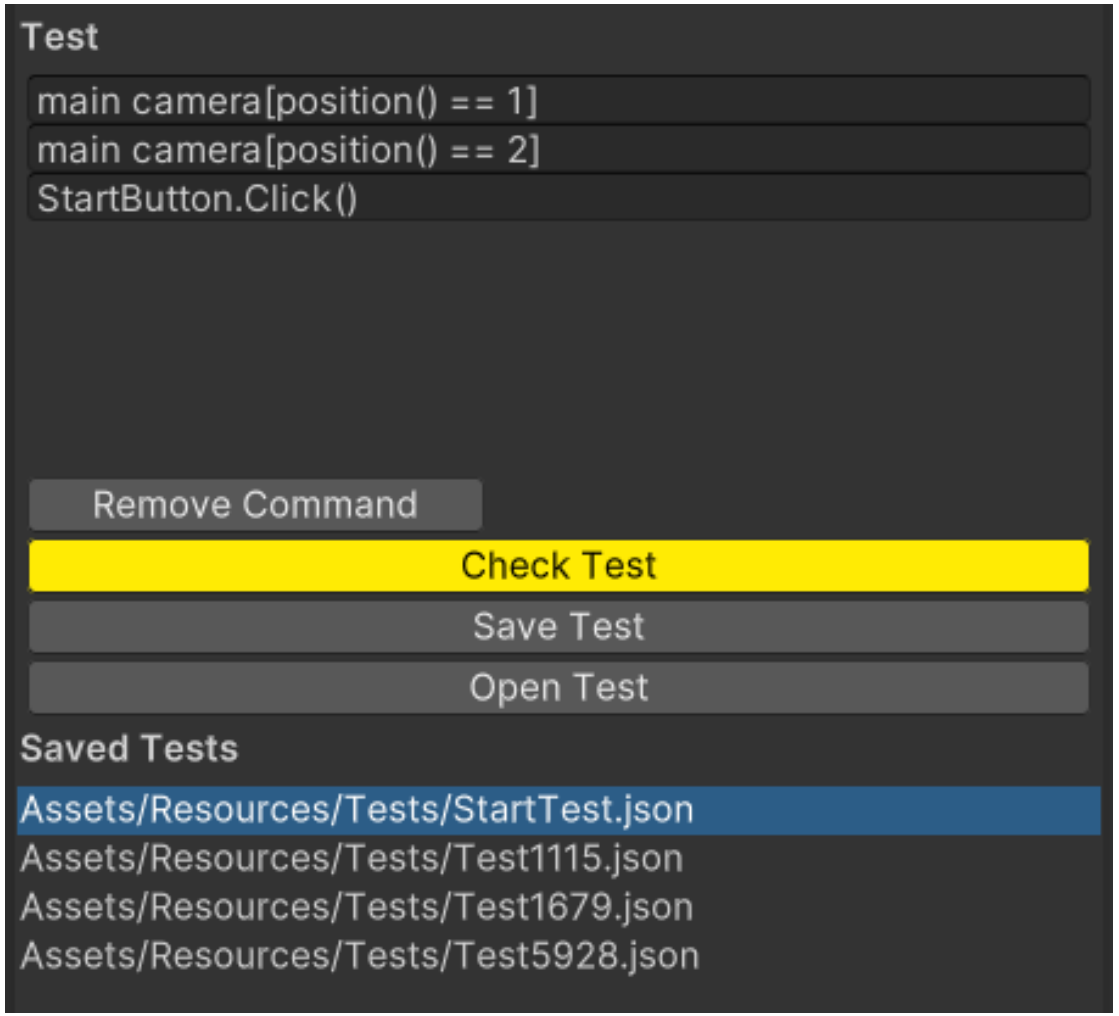




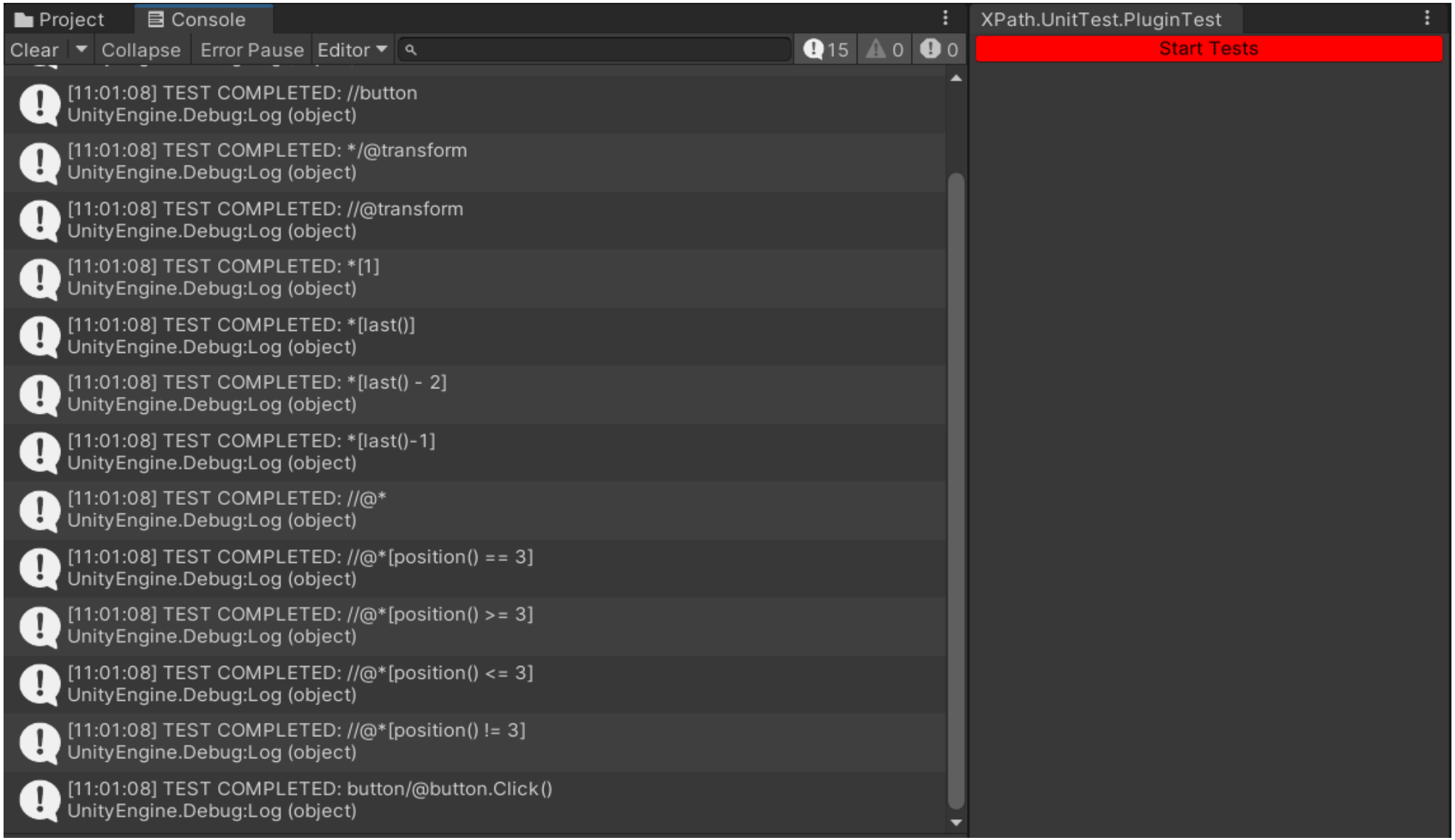


Открытие сохраненного теста





Проведение тестирования программного модуля



Заключение

В процессе выполнения выпускной квалификационной работы был разработан инструмент для функционального тестирования графического интерфейса программ, позволяющий создание тестов без использования средств языков программирования и обеспечивающий автоматическую проверку тестов.

На данный момент программный модуль находится на стадии интеграции в более крупный модуль по тестированию программ на платформе Unity.

Архитектура программного модуля позволяет расширение функционала, в дальнейшей работе планируется расширить возможность программного модуля тестировать не только графический интерфейс, но и логику поведения объектов и компонентов моделируя поведение пользователя через использование команд устройств ввода.

Спасибо за внимание!