



# Software Engineering Conference Russia

November 14-15, 2019. Saint-Petersburg

## Система автоматизации функционального тестирования для приложений на игровом движке Unity

**Олег Бедрин**  
**Влада Кугуракова**  
Digital Media Lab



Казанский федеральный  
УНИВЕРСИТЕТ  
ВЫСШАЯ ШКОЛА  
информационных технологий  
и интеллектуальных систем

# План

1. Существует много Unity проектов которые используют Input если не большинство
  - Весь VR который использует XR Legacy Input System
  - VRTK
    - Биотехнологическая лаба, криминалистика, симуляции техники
  - Стимовские игры однопользовательские и нормальные игры
2. Есть проблема с тем что каждый раз перепроходить свою игру на юньке
  - Личная боль: биотехнолаба построена так, что нужно постоянно было перепроходить весь сценарий, чтобы удостовериться что всё работает.
  - Триггер к началу разработки: В одном из новых проектов необходимо было реализовать систему, которая могла бы записывать и проигрывать вещи которые делает игрок.
3. Как я это сделал?
  - DI контейнер - изъезженная тема, была уже реализована, но отказались тк в DOTS это не надо, но как же зависимости между системами?
  - Архитектура основанная на DI
    - Основные системы
  - Принцип работы перехвата
  - Окна
  - Демо - видео

# Три основных системы ввода Unity



# Интерфейсы-маркеры

[Unity User Manual \(2019.2\)](#) / [Scripting](#) / [Event System](#) / Supported Events



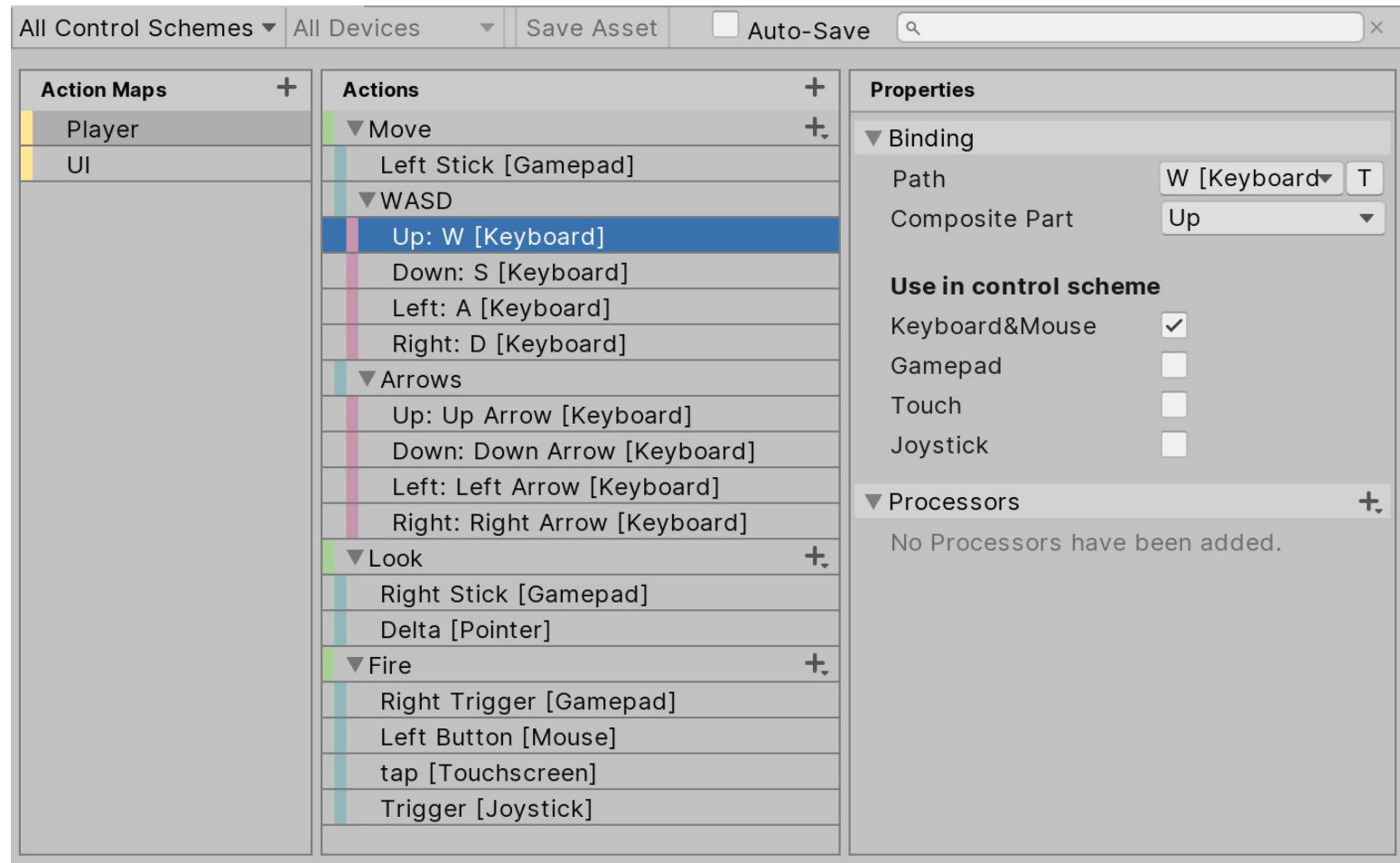
## Supported Events

The **Event System** supports a number of events, and they can be customised further in user custom user written Input Modules.

The events that are supported by the Standalone Input Module and Touch Input Module are provided by interface and can be implemented on a MonoBehaviour by implementing the interface. If configured the events will be called at the correct time.

- IPointerEnterHandler - OnPointerEnter - Called when a pointer enters the object
- IPointerExitHandler - OnPointerExit - Called when a pointer exits the object
- IPointerDownHandler - OnPointerDown - Called when a pointer is pressed on the object
- IPointerUpHandler - OnPointerUp - Called when a pointer is released (called on the GameObject that the pointer is clicking)

# Input Ассеты (Превью версия)

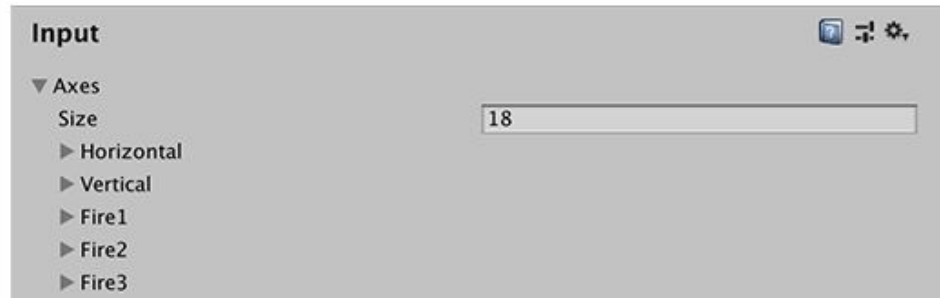


# Класс Input

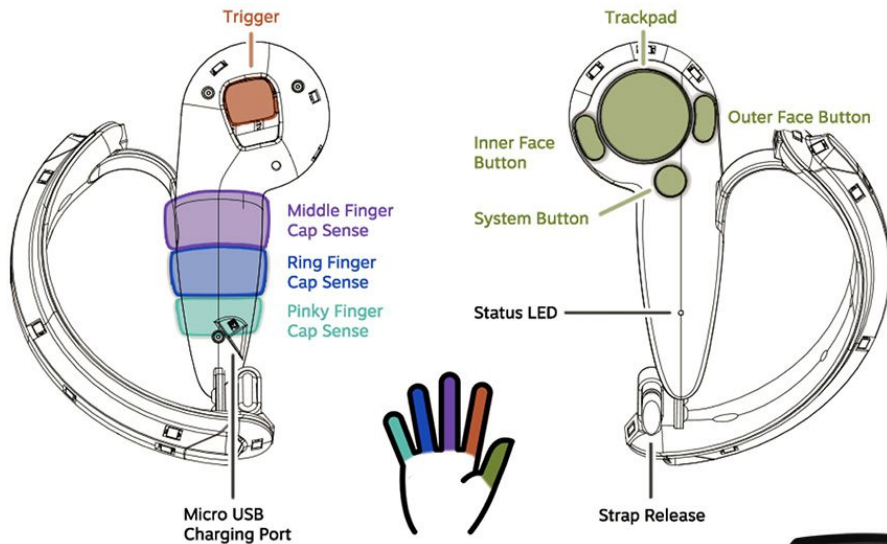


## Input

Use the **Input** settings (top menu: **Edit** > **Project Settings**, then select the **Input** category) to [define the input axes and game actions](#) for your Project.

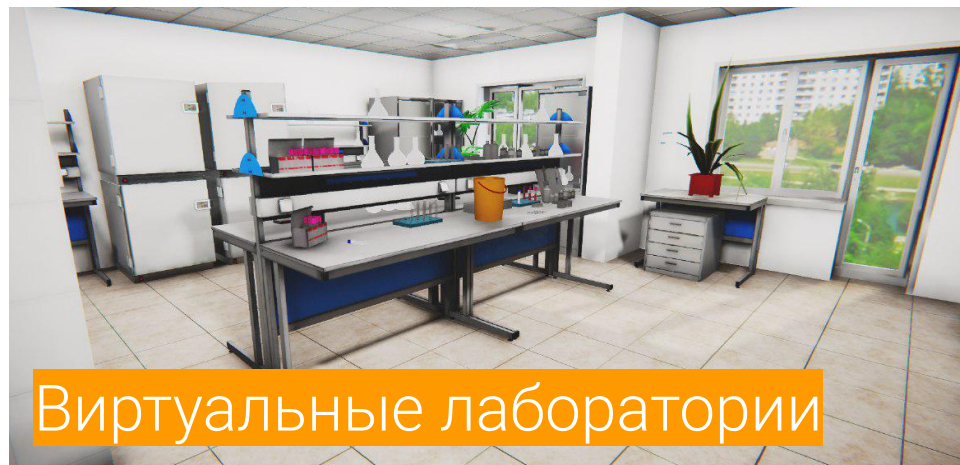


# Кто использует Input класс?





# Где мы его используем?



Виртуальные лаборатории



Болгар XIV



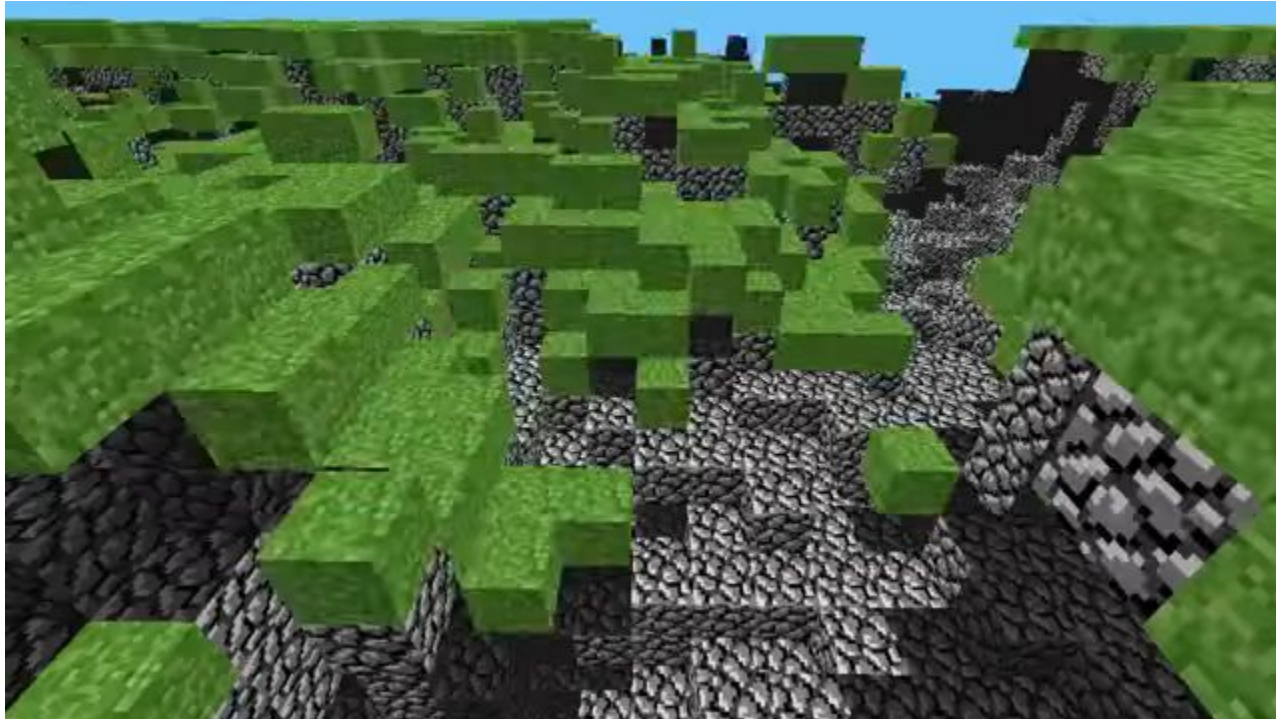
Art City VR



Камаз-Дакар



# Представим, что мы сделали игру



# И все unit-тесты проходят, но...



# Нужно провести ещё один тест



Казанский федеральный  
УНИВЕРСИТЕТ  
ВЫСШАЯ ШКОЛА  
информационных технологий  
и интеллектуальных систем

# Нужно провести ещё один тест

- Нужно проверить крутится ли вентиль...

# Нужно провести ещё один тест

- Нужно проверить крутится ли вентиль...
- Качаются ли качели...



# Нужно провести ещё один тест

- Нужно проверить крутится ли вентиль...
- Качаются ли качели...
- Меняется ли UI при взаимодействии с НИМ...

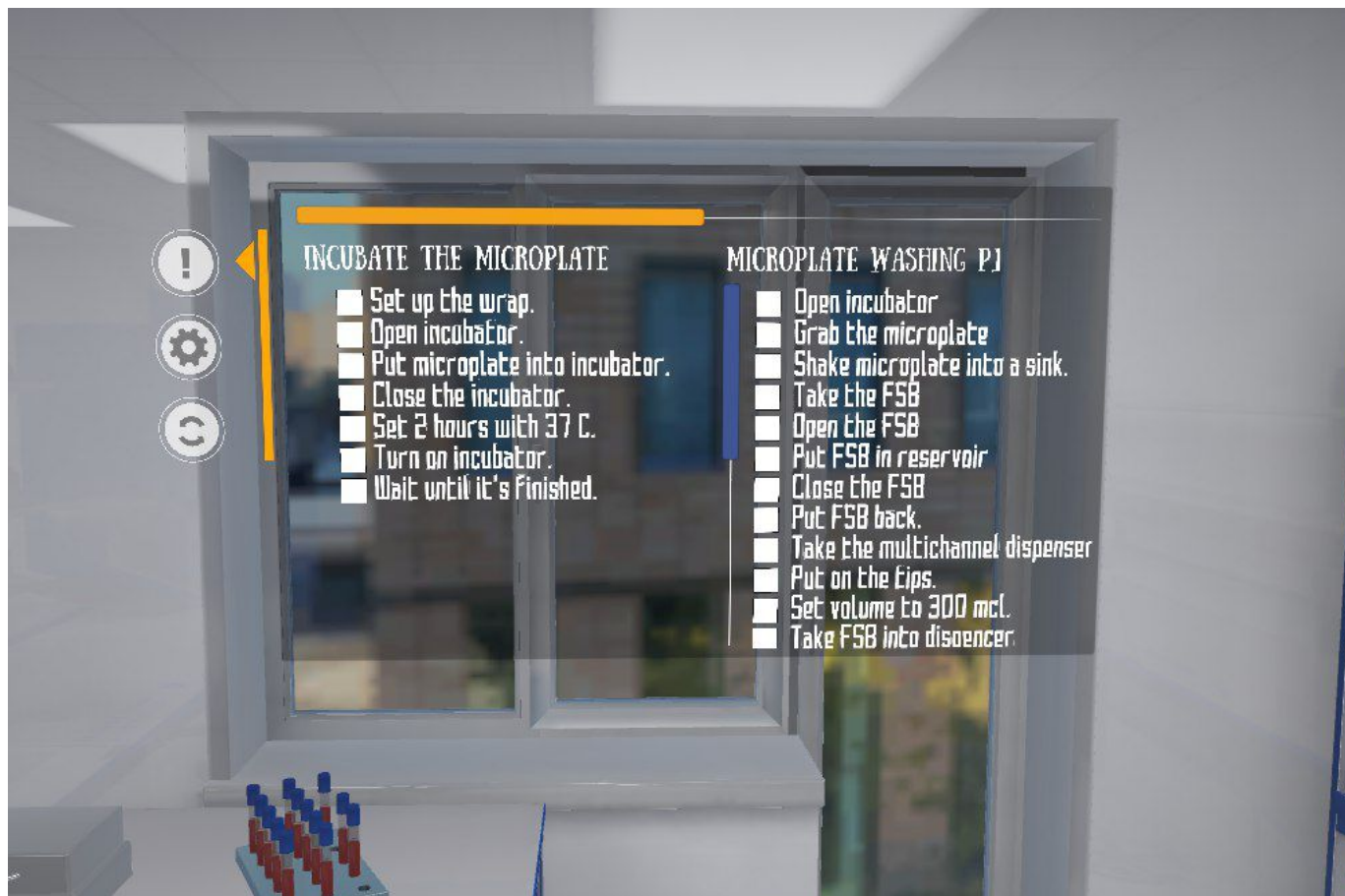
# Нужно провести ещё один тест

- Нужно проверить крутится ли вентиль...
- Качаются ли качели...
- Меняется ли UI при взаимодействии с НИМ...
- И ещё куча всего...

# Нужно провести ещё один тест

- Нужно проверить крутятся ли вентиль...
- Качаются ли качели...
- Меняется ли UI при взаимодействии с НИМ...
- И ещё куча всего...
- **И всё это нужно будет делать руками или тестерам.**

# Личная боль



# Решение относительно старое



Казанский федеральный  
УНИВЕРСИТЕТ  
ВЫСШАЯ ШКОЛА  
информационных технологий  
и интеллектуальных систем



# Решение относительно старое

- Автоматизация действий

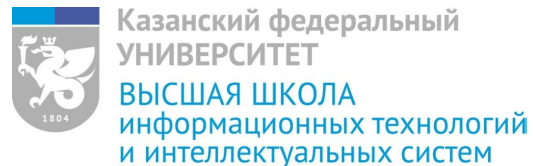
# Решение относительно старое

- Автоматизация действий



# Решение относительно старое

- Автоматизация действий



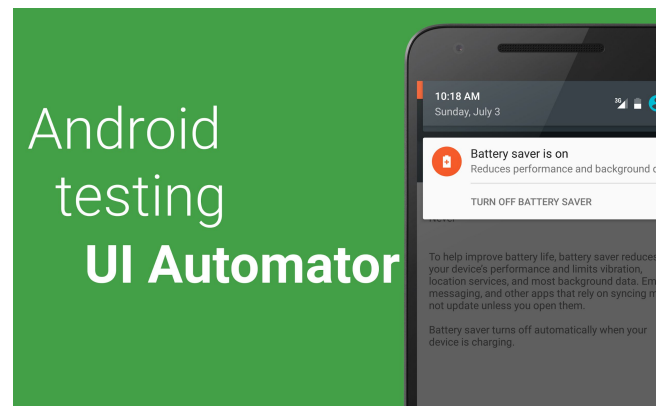
# Решение относительно старое

- Автоматизация действий



# Решение относительно старое

- Автоматизация действий



Казанский федеральный  
УНИВЕРСИТЕТ  
ВЫСШАЯ ШКОЛА  
информационных технологий  
и интеллектуальных систем



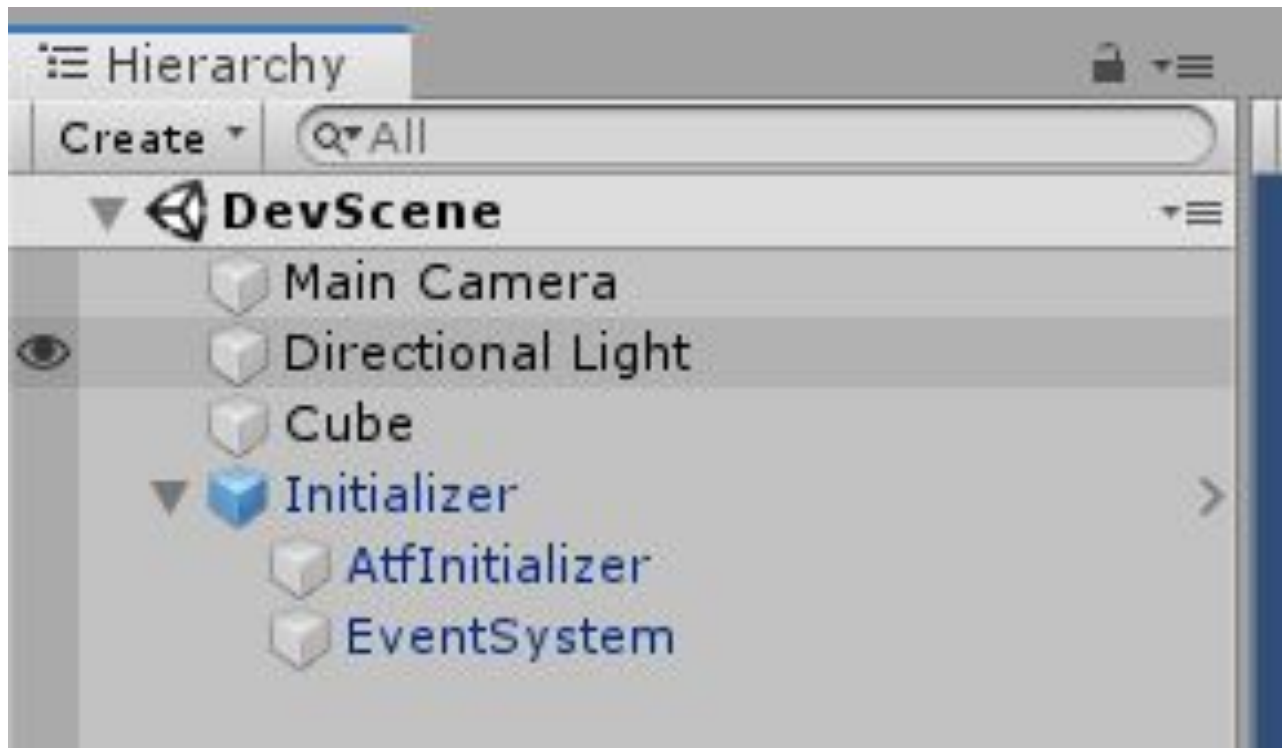
# Необходимо решение для Unity

- Легкая интеграция в проект
- Готовность к DOTS
- Кастомный Unity Editor UI

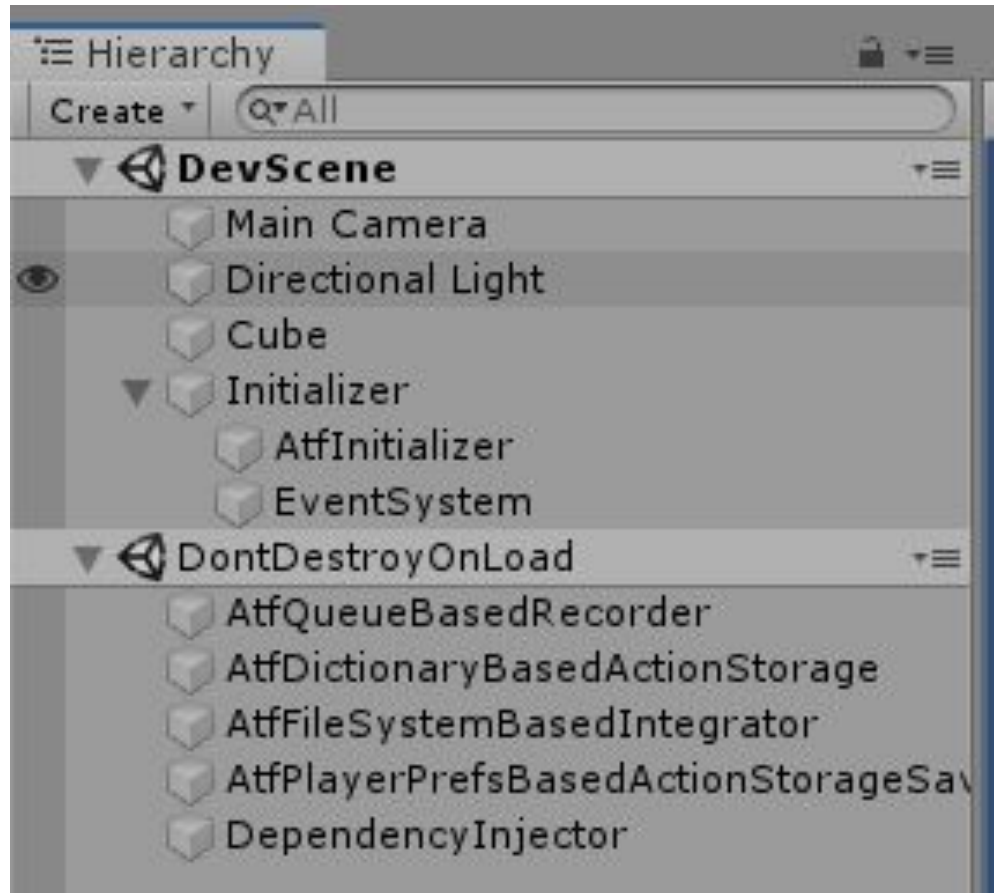
# Концептуально должен быть похож

- Автоматизация действий
- Генерация и хранение сценариев
- Графический интерфейс

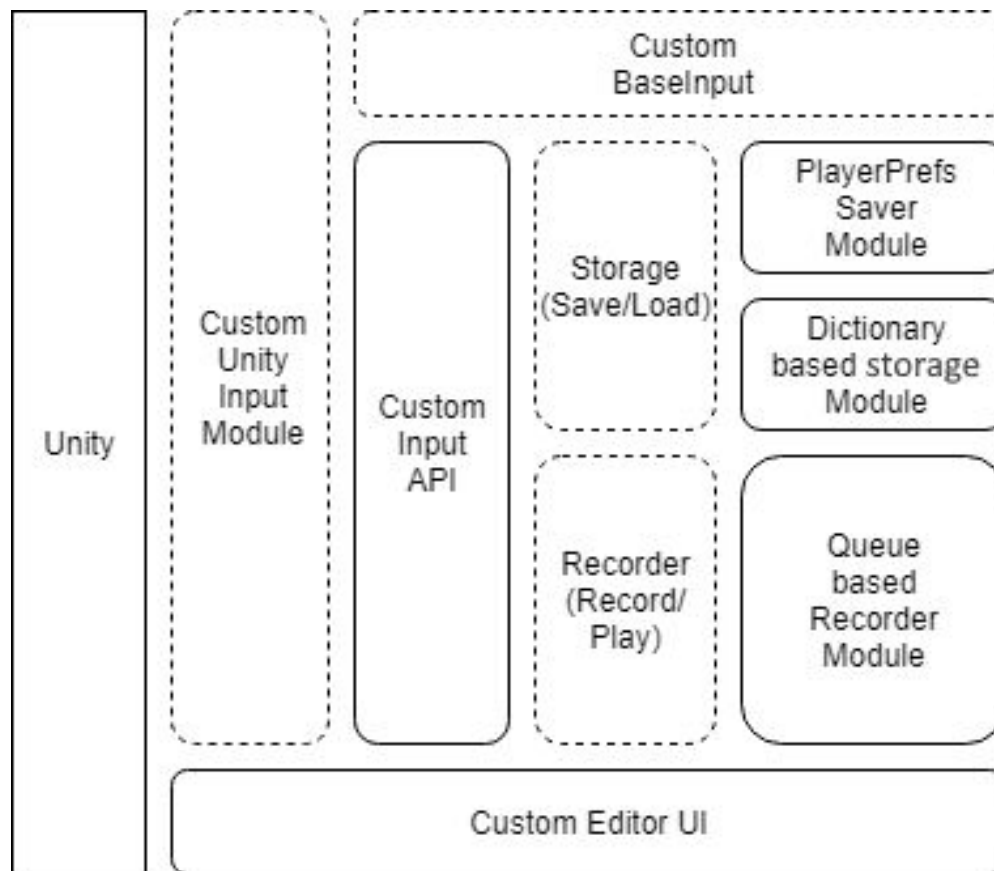
# Интеграция одним префабом



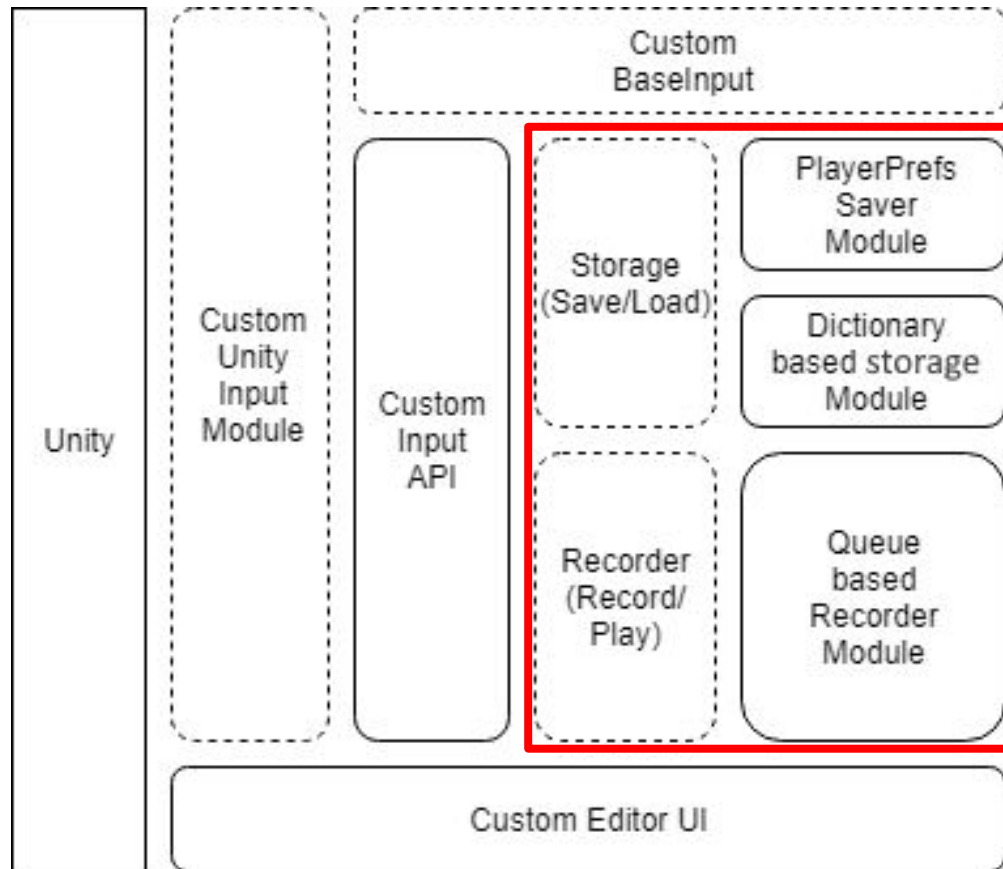
# При запуске создаются все системы



# Архитектура решения ATF



# Абстракции и реализации систем



# Основной механизм ATF

🔥 Event function

```
private void Update()  
{  
    if (AtfInput.GetKey(KeyCode.Space))
```

# Основной механизм АТФ

🔥 Frequently called  1 usage

```
public static bool GetKey(KeyCode key)
{
    return Intercept( realInput: Input.GetKey(key),
        fakeInputKind: FakeInput.GET_KEY,
        defaultValue: false, fakeInputParameter: key );
}
```



# Основной механизм ATF

Frequently called 16 usages

```
private static T Intercept<T>(object realInput,
    FakeInput fakeInputKind, T defaultValue, object fakeInputParameter = null)
{
    if (fakeInputParameter == null) fakeInputParameter = new object();
    return IfExceptionReturnDefault(
        function: () => (T) RealOrFakeInputOrRecord(realInput,
            fakeInput: GetCurrentFakeInput(fakeInputKind, fakeInputParameter),
            fakeInputParameter, fakeInputKind),
        defaultValue
    );
}
```

# Принципы построения архитектуры

- Single responsibility principle
- Liskov substitution principle
- Interface segregation principle
- Dependency inversion principle

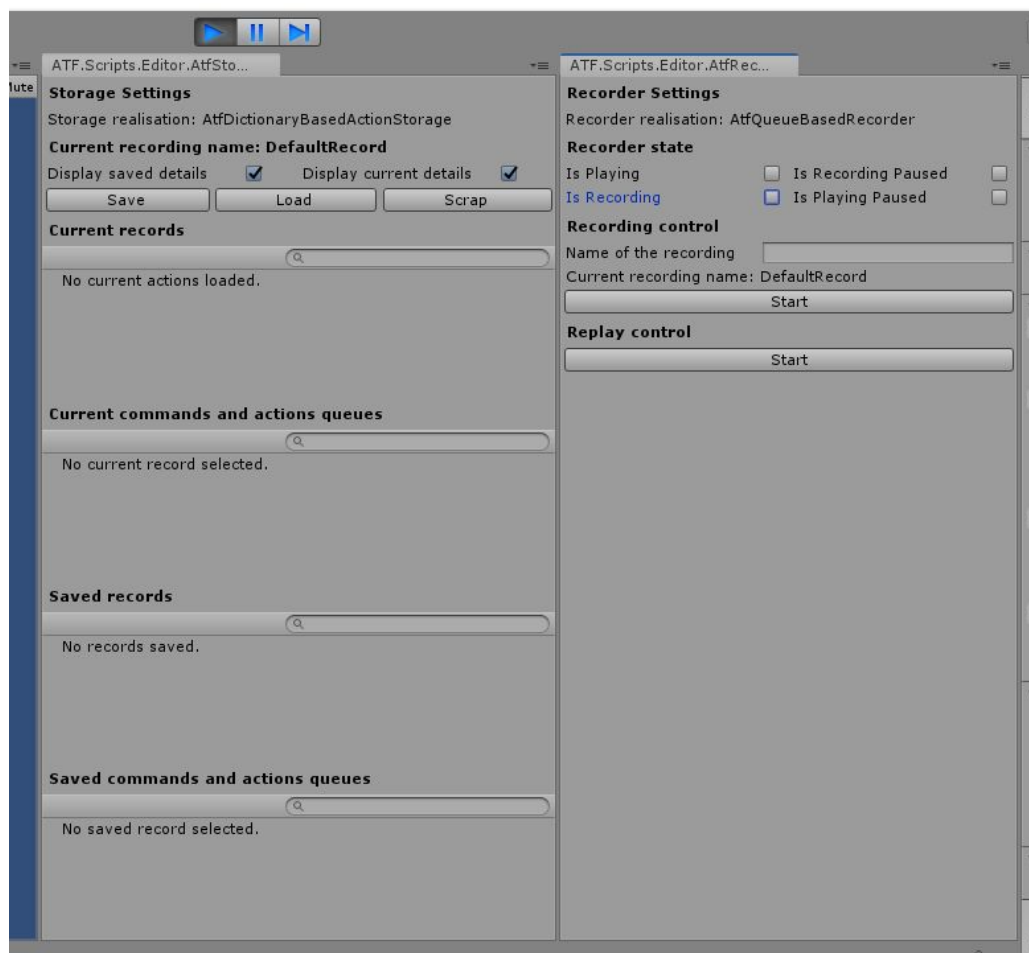
# Использование логов поведения

- Генерация сценариев
- Сериализация в документы

# Возможности построенного UI

- Управление хранилищем записей
- Управление проигрывателем

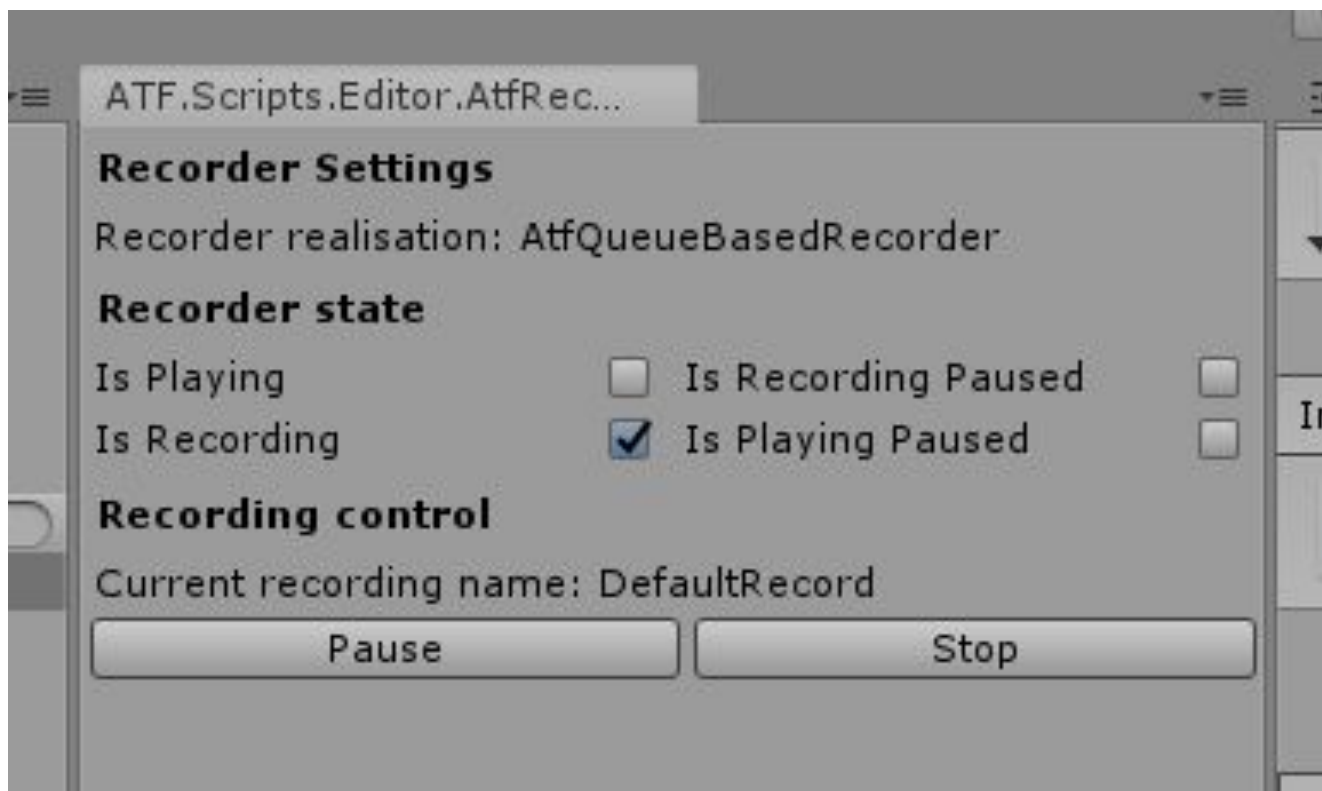
# Пользовательский Unity Editor UI



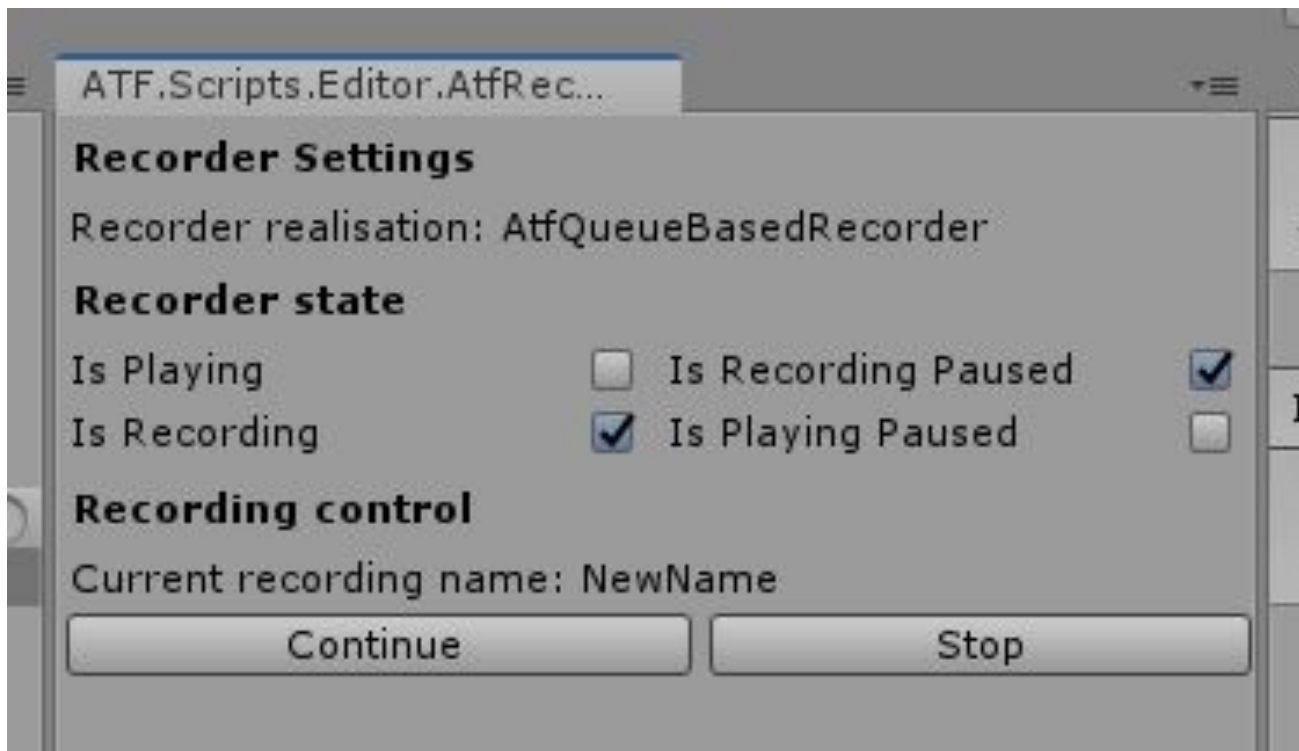
# Пользовательский Unity Editor UI



# Пользовательский Unity Editor UI

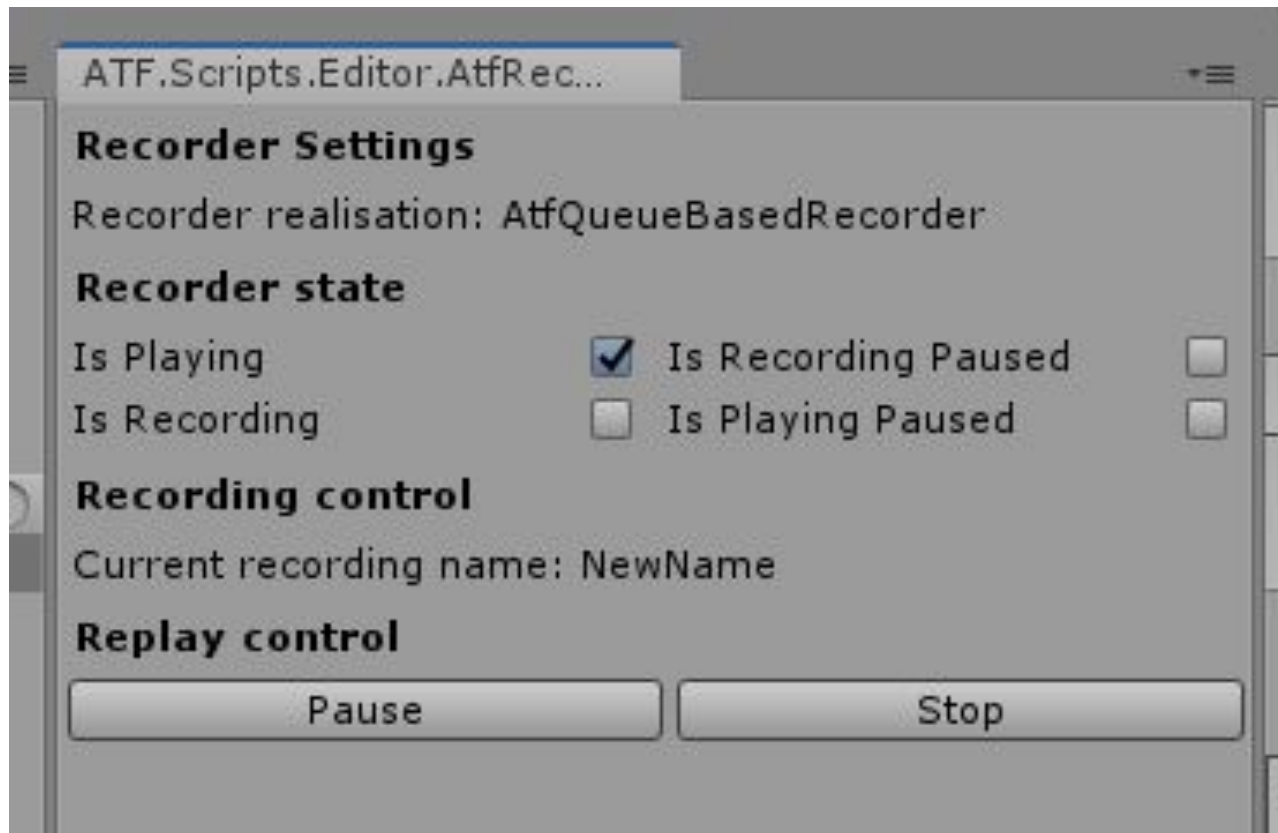


# Пользовательский Unity Editor UI

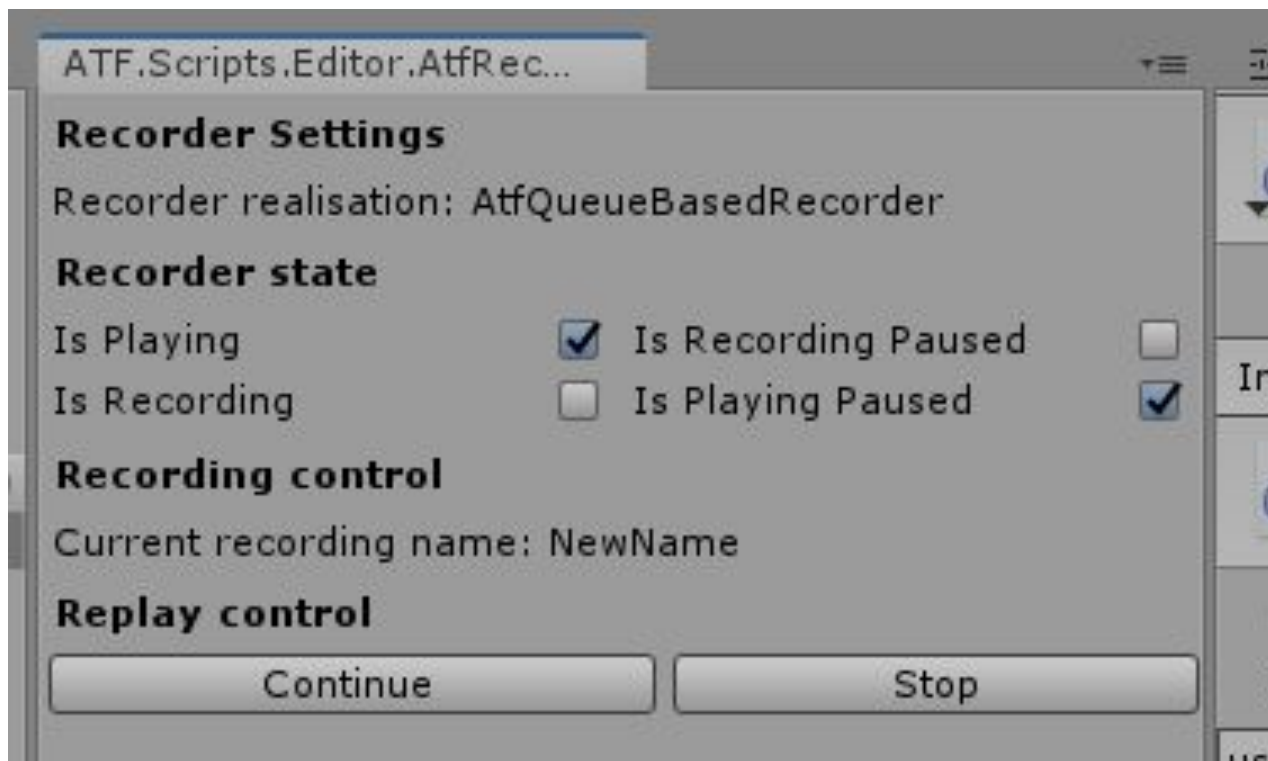




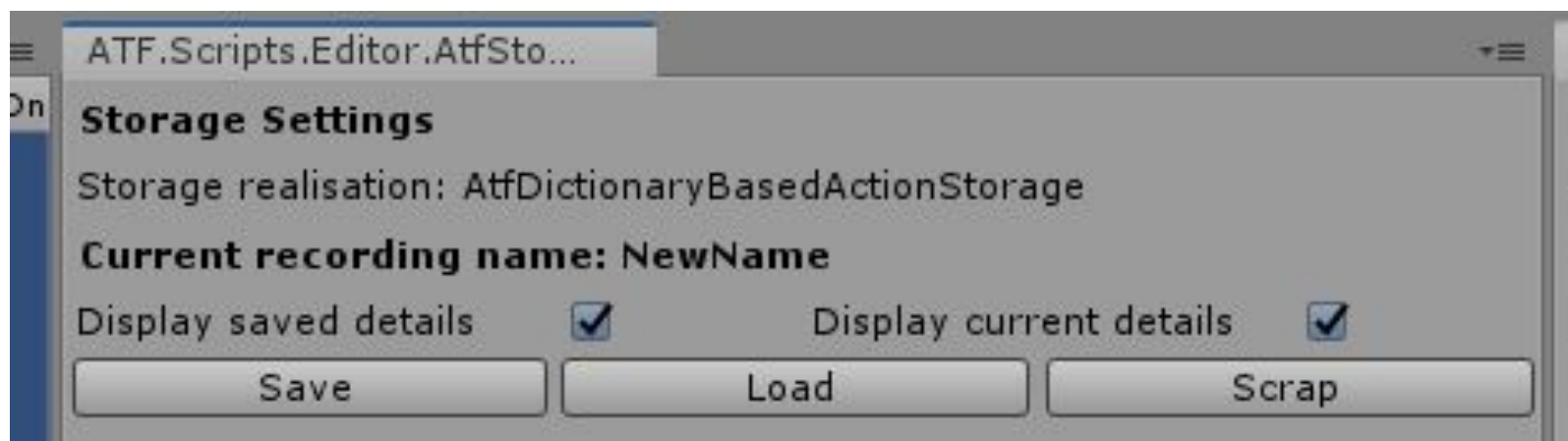
# Пользовательский Unity Editor UI



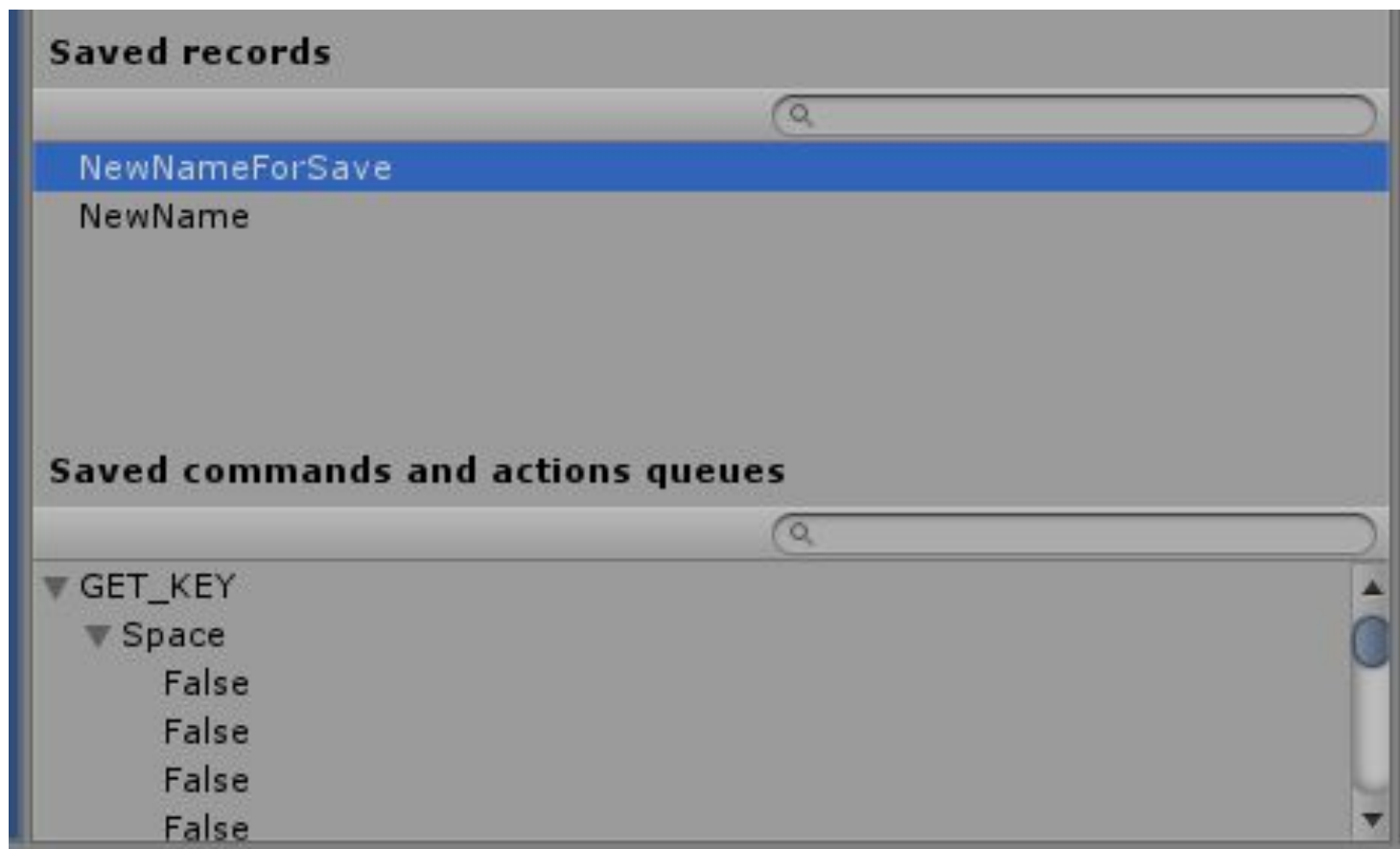
# Пользовательский Unity Editor UI



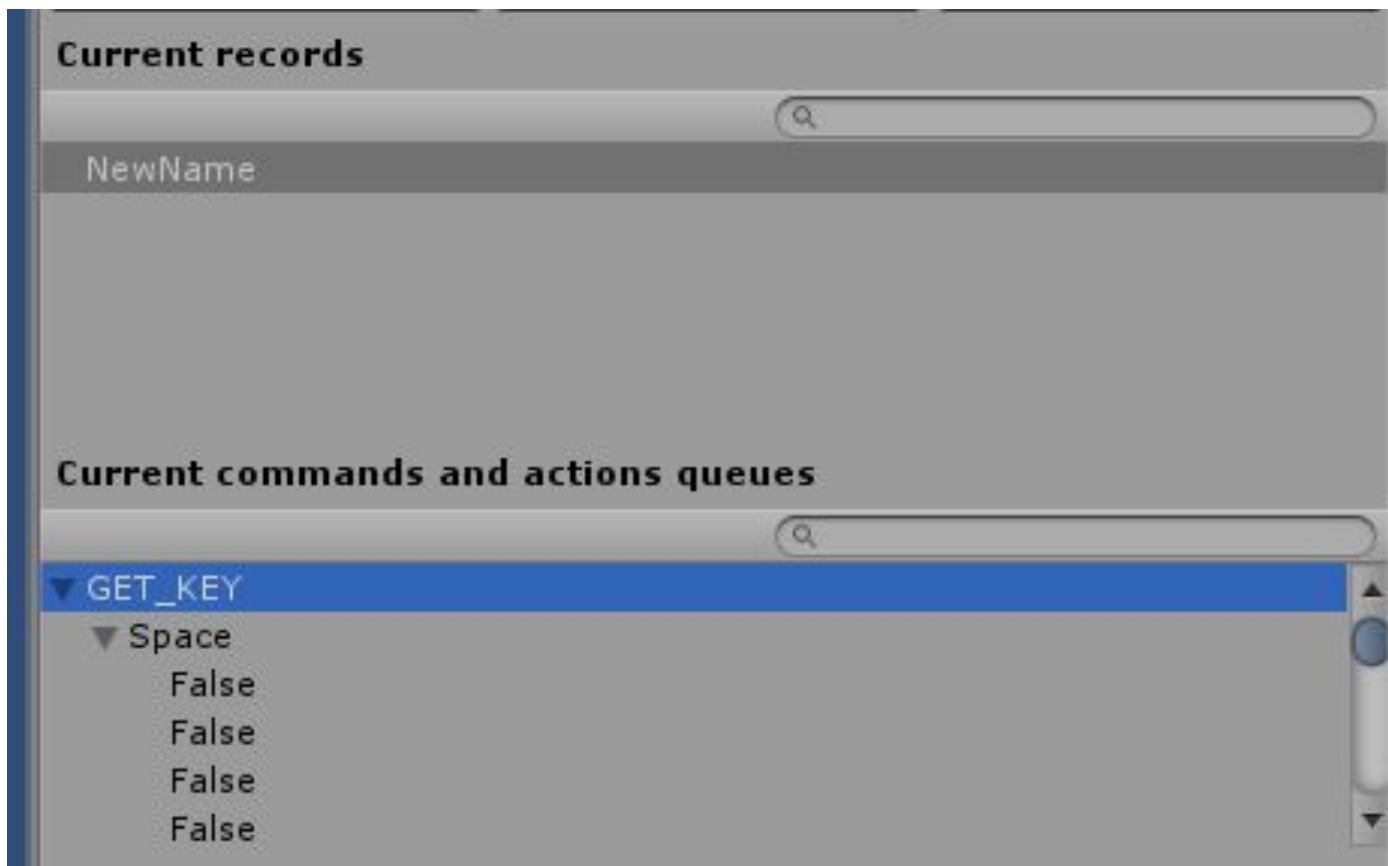
# Пользовательский Unity Editor UI



# Пользовательский Unity Editor UI



# Пользовательский Unity Editor UI



# Пример использования

## Листинг 4: Демонстрационный скрипт использования ATF

---

```
namespace ATF.Scripts.Example
{
    public class Mover : MonoBehaviour
    {
        private const float SPEED = 100f;

        private void Update()
        {
            if (ATFInput.GetKey(KeyCode.Space))
            {
                transform.Rotate(new Vector3(0, 1, 1),
                    Time.deltaTime * SPEED);
            }
        }
    }
}
```

---

Project Explorer

- Project
- Source Files
- Resource Files
- Properties



Properties

General

Advanced

Appearance

Behavior

Events

Layout

Resources

Properties

General

Advanced

Appearance

Behavior

Events

Layout

Resources

Project Explorer

- Project
- Source Files
- Resource Files
- Properties

Properties

General

Advanced

Appearance

Behavior

Events

Layout

Resources

Properties

General

Advanced

Appearance

Behavior

Events

Layout

Resources



# Контакты

1. Олег Александрович Бедрин  
[oabedrin@stud.kpfu.ru](mailto:oabedrin@stud.kpfu.ru)
2. Влада Владимировна Кугуракова  
[vlada.kugurakova@gmail.com](mailto:vlada.kugurakova@gmail.com)

