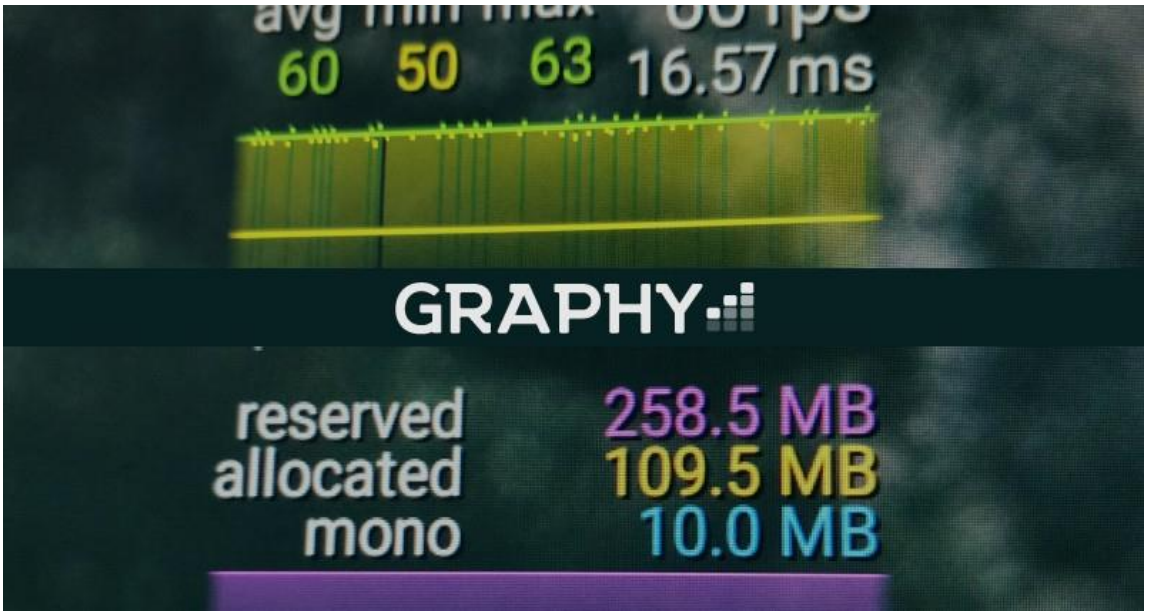
- Графический -
Ultimate Stats Monitor & Debugger

Версия актива:
v2.1.0
Автор: Мартин Пане (Студия: Tayx)

Контактная информация:

- ➔ Электронная почта (martintayx@gmail.com)
- ➔ Twitter (@martinTayx)
- ➔ Discord (https://discord.gg/2KgNEHK)



Здравствуйте, и спасибо, что скачали **Graphy**!

Я надеюсь, что вы найдете этот ресурс таким же полезным, как и я, для отладки и мониторинга ваших проектов Unity. Здесь вы найдете подробный обзор всех функций и опций, которые предлагает вам Graphy. Если после прочтения этой статьи вам что-то осталось неясным, или вы хотите запросить какую-либо функцию или сообщить об ошибке, прокрутите страницу вниз и вы найдете ссылки для связи со мной различными способами.

- ➔ **Введение:** Graphy - это набор инструментов, который предлагает следующее:
  - ◆ График и счетчик FPS.
  - ◆ График и счетчик памяти (RAM).
  - ◆ График аудиоспектрометра и счетчик дБ.
  - ◆ Расширенные данные устройства.
  - ◆ Сценарий отладчика.

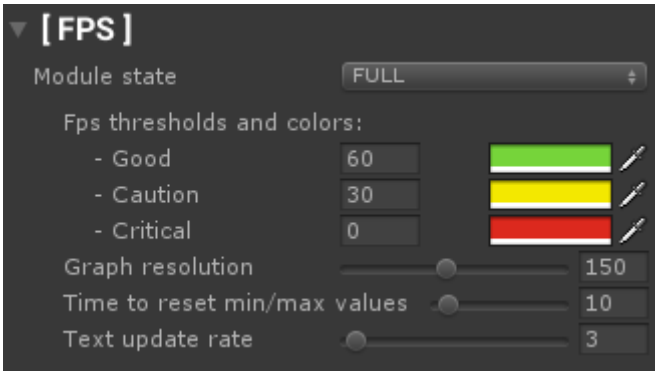
- ➔ **Как использовать:**
  - ◆ Просто перетащите префаб из "Tayx/Graphy - Ultimate Stats Monitor/Prefab" в нужную вам сцену!
  - ◆ Он переживет изменения сцены, использует паттерн Singleton и DontDestroyOnLoad.
  - ◆ Будьте осторожны и не пытайтесь получить к нему доступ из кода, если он не инстанцирован в сцене, это не сработает.

➔ **Общие характеристики модуля:**

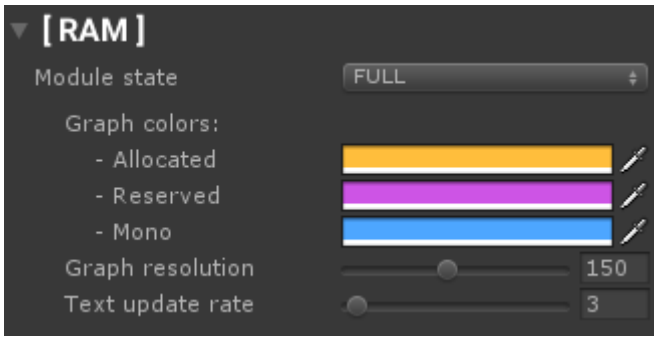
- **Module Position:** определяет позицию, в которой будет расположен модуль.
- **Состояние модуля:**
  - **ПОЛНЫЙ:** полный модуль со всеми его функциями.
  - **ТЕКСТ:** скрывает графики и показывает только текст.
  - **BASIC:** показывает только основную информацию о модуле.
  - **BACKGROUND:** отслеживает данные в фоновом режиме, делая их доступными из кода.
  - **OFF:** полностью отключает модуль, делая его недоступным из кода. Осторожно, это также нарушает функциональность **отладчика** (который использует мониторы для чтения данных).
- **Цвет графика:** устанавливает цвет графика.
- **Разрешение графика:** устанавливает количество точек, на которые делится график. Будьте осторожны с очень высоким разрешением графика на экранах с низким разрешением, у графика могут возникнуть проблемы с отрисовкой всех точек.
- **Частота обновления текста:** определяет, сколько раз в секунду обновляется текст.

➔ **Модуль FPS:**

- **Пороги:** значения, используемые для переключения между различными цветами, связанными с текстом и графиком.
- Нижние 1% и 0,1%: показывает постоянство времени кадров. Нижний 1% и 0,1% показывает, как часто и сильно игра будет заикаться



➔ **Модуль оперативной памяти:** не имеет специальных средств управления.



➔ **Аудио модуль:**

- **Найти аудиопрослушиватель:** применяется, только если AudioListener равен null.
  - **ALWAYS:** постоянно сканирует наличие компонента AudioListener в основной камере.
  - **ON SCENE LOAD:** сканирует AudioListener в основной камере каждый раз, когда загружается сцена.
  - **NEVER:** никогда не проверяет наличие AudioListener.
- **Audio Listener:** ссылка на AudioListener, который вы хотите использовать.
- **Окно БПФ:** Используется для уменьшения утечки между частотными бинами/полосами. Обратите внимание, чем сложнее тип окна, тем лучше качество, но ниже скорость. Самое простое - прямоугольное. Самое сложное - BlackmanHarris.
- **Размер спектра: Должен** быть равен степени 2 в диапазоне 128-8192. Чем выше частота дискретизации, тем меньше точность, но и больше влияние на производительность. Осторожно с мобильными устройствами.

➔ **Модуль дополнительных данных:** отображает информацию об устройстве.

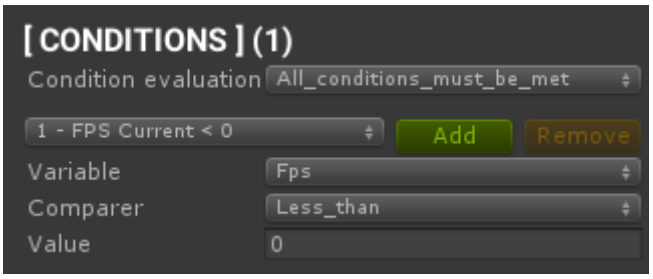
- **Данные экрана:** разрешение, частота обновления.
- **Данные окна:** разрешение, частота обновления, dpi.
- **Графический API:** OpenGL/DirectX/etc.
- **GPU:** VRAM, максимальный размер текстуры, уровень шейдеров.
- **RAM:** доступная память системы.
- **ОС:** операционная система.

➔ **Отладчик:** этот компонент представляет собой мощную функцию, позволяющую задать ряд условий, выполнение которых запустит цепочку определенных вами действий. Он разработан на основе отладочных пакетов. Каждый отладочный пакет может иметь различные условия и действия.

- ◆ **Общие параметры:**
  - **Активный:** Если не активен, то будет пропущен при проверке отладчиком отладочных пакетов.
  - **ID:** Необязательно, но используется для работы с отладочными пакетами из кода.
  - **Выполнить один раз:** Если true, то он будет выполнен только один раз, после чего удалится.
  - **Время начального сна:** Время ожидания перед проверкой этого отладочного пакета.
  - **Время сна после выполнения:** Применяется, только если "Execute Once" равно false. Время ожидания перед проверкой данного

отладочного пакета после его выполнения.

- ◆ **Условия:**
  - **Оценка условий:** определяет, все ли условия должны быть выполнены, или только одно из них.
  - **Переменная:** какую переменную сравнивать с указанным значением.
  - **Компаратор:** <, <=, ==, >=, >.
  - **Значение:** плавающее значение, используемое для сравнения с указанной

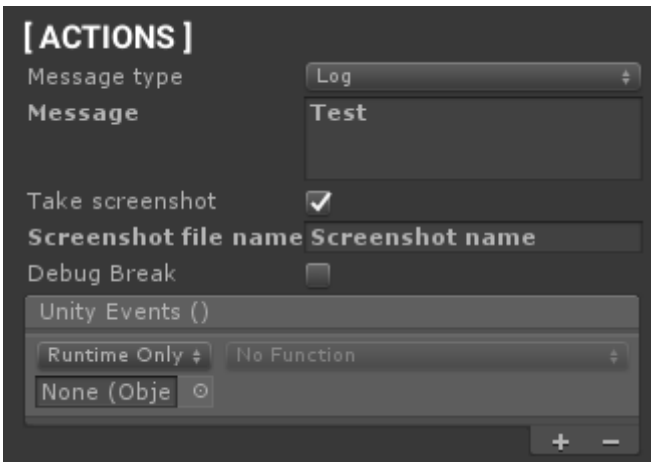


переменной.

- ◆ **Действия:**
  - **Тип сообщения:** стандартные типы консольного журнала: Журнал, Предупреждение, Ошибка.
  - **Сообщение:** текстовое сообщение, которое вы хотите отправить на консоль.
  - **Сделать снимок экрана:** делает снимок экрана.
  - **Имя файла скриншота:** имя скриншота. Избегайте этих символов: "* . \" / \\ [ ] :
    ; | = ," (без кавычек). В конце будет добавлена дата, чтобы избежать перезаписи. Место записи изображения может включать список каталогов/папок. При отсутствии списка каталогов/папок изображение будет записано в папку Project. На мобильных платформах имя файла добавляется к пути к постоянным данным.

- **Прерывание отладки:** приостанавливает работу редактора Unity.
- **События Unity:** обертка событий Unity, которая может быть сериализована и изменена из редактора Unity.

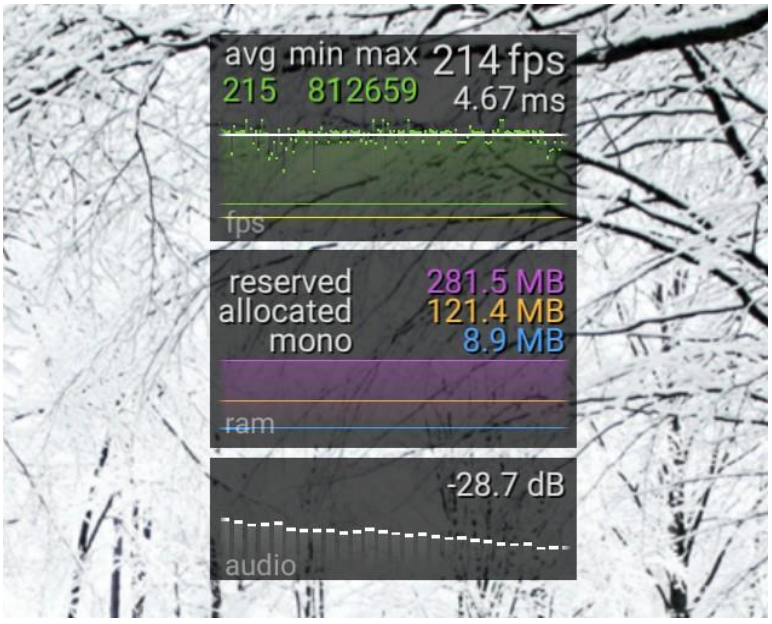
- **System.Action (доступен только из кода):** вызывайте собственные методы из кода, добавляя System.Action к нужному вам отладочному пакету, получая его по ID.



➔ **Графический режим:**
- ◆ **ПОЛНЫЙ:** разблокирует все функции до их максимального потенциала.
- ◆ **LIGHT:** ограничивает некоторые мелочи для улучшения совместимости со старым оборудованием (например, уменьшено максимальное разрешение графики).

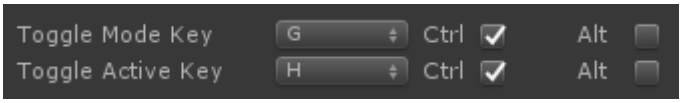
➔ **Keep Alive:** если верно, Graphy переживет изменения сцены. Осторожно, если Graphy установлен как дочерний объект другого объекта, оба переживут изменения сцены.

➔ **Справочная информация**
- ◆ Если флажок установлен, ко всем модулям будет применено фоновое наложение. Цвет будет соответствовать выбранному.



➔ **Горячие клавиши**

- **Режимы переключения:** переключение между различными макетами и модулями.
- **Переключение активно:** включает/выключает Graphy



➔ **Скриптинг**

- **Добавьте пространство имен:** Добавьте "using Tayx.Graphy;" в верхнюю часть вашего файла .cs.
- **GraphyManager:** через этот класс вы можете получить доступ ко всем переменным, связанным со статистикой, таким как CurrentFPS, AllocatedMemory и т.д.
- **GraphyDebugger:** через этот класс вы можете получить доступ к различным отладочным пакетам, чтобы деактивировать их, добавить обратные вызовы

```
// Use this for initialization
0 references
void Start ()
{
    var fps = GraphyManager.Instance.CurrentFPS;

    GraphyDebugger.Instance.RemoveAllDebugPacketsWithId(2);
}
```

System.Action и т.д.

- **Продвинутый материал**:
  - Добавьте обратные вызовы к существующему пакету отладки:

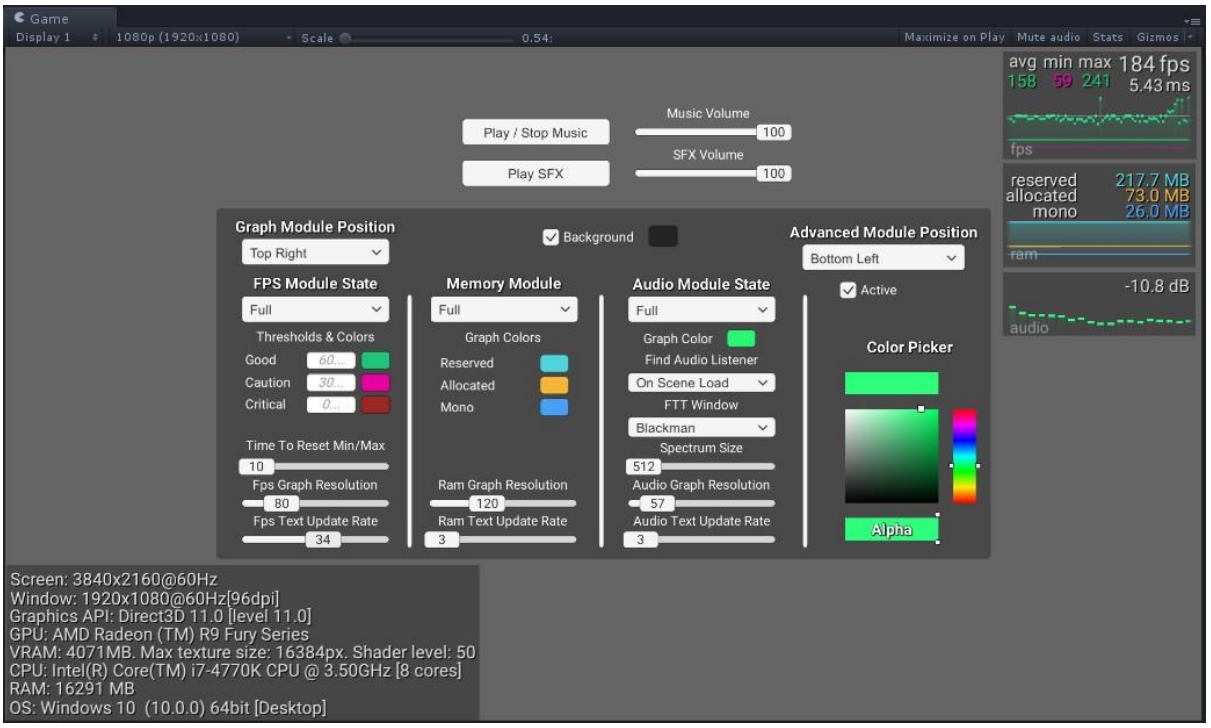
```
Action TestAction = null;

TestAction += TestMethod;

GraphyDebugger.Instance.AddCallbackToFirstDebugPacketWithId(TestAction, 5);
```

  - Добавьте новый пакет отладки:

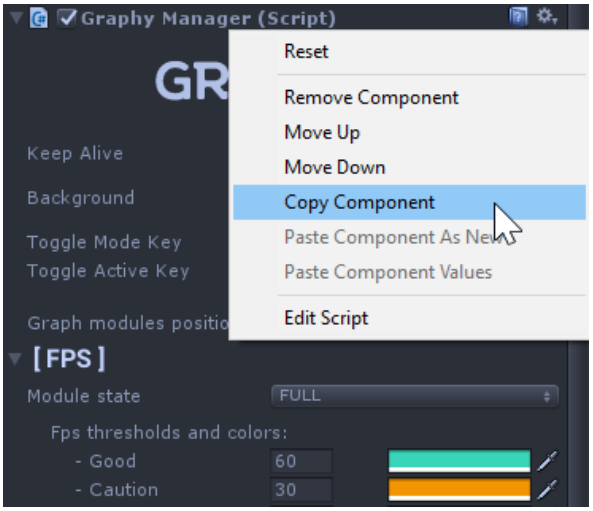
```
Action TestAction = null;

TestAction += TestMethod;

GraphyDebugger.Instance.AddNewDebugPacket
(
    5,
    new GraphyDebugger.DebugCondition()
    {
        Comparer = GraphyDebugger.DebugComparer.Equals_or_greater_than,
        Value = 45f,
        Variable = GraphyDebugger.DebugVariable.Fps_Avg
    },
    GraphyDebugger.MessageType.Warning,
    "Message Warning Test",
    true,
    TestAction
);
```

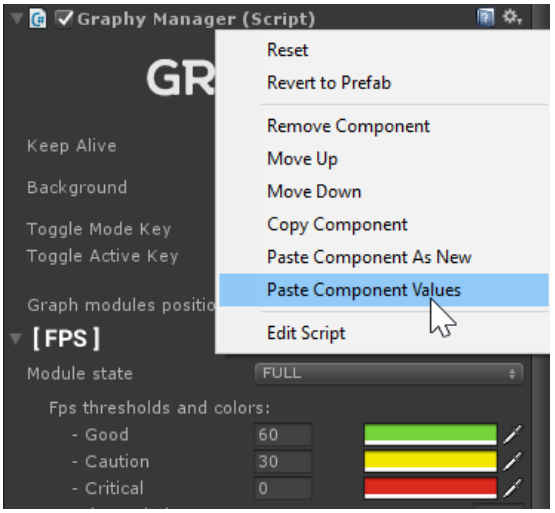
➔ **Сцена настройки:**

- Его можно найти по адресу: *"Assets\Tayx\Graphy - Ultimate Stats Monitor\Scene\Customize Graphy".*
- Позволяет установить все параметры во время выполнения программы



- Как только вы получите нужные значения, просто скопируйте компонент, выйдите из режима воспроизведения и вставьте значения компонента, как показано ниже:





Спасибо, что дочитали до этого места! Я надеюсь, что вам понравится Graphy и что он облегчит вашу жизнь разработчика. Я буду очень признателен, если вы оставите отзыв в Asset Store, а если вы хотите **связаться со мной,** не стесняйтесь:

- ➔ Электронная почта (martintayx@gmail.com)
- ➔ Twitter (@martinTayx)
- ➔ Discord (https://discord.gg/2KgNEHK)

ВОПРОСЫ И ОТВЕТЫ:

- ➔ Почему значение FPS в окне "Статистика" на вкладке "Игра" отличается от значения FPS в Graphy?
  - ◆ *Это значение является оценкой FPS, который вы могли бы получить в сборке, без накладных расходов редактора. Graphy сообщает текущее реальное значение FPS.*
- ➔ Есть ли у вас приблизительный показатель мс на типичном настольном оборудовании среднего класса, со всеми включенными функциями?
  - ◆ *Да, менее 0,1 мс в сборке и около 0,3-0,7 мс в редакторе из-за дополнительных накладных расходов редактора.*
- ➔ Как создается Graphy?
  - ◆ *Он визуализируется через пользовательский интерфейс Unity. Graphy содержится в стандартном Canvas, и у вас есть полный контроль над ним.*