



PLUGIN YOUR GAMES 2.0

ПОДКЛЮЧАЙТЕ СВОИ ИГРЫ

[Онлайн документация \(красивая, удобная и актуальная\)](#)

[Телеграм канал](#) — новости об обновлениях, полезная и важная информация.

[Телеграм чат](#) — задать вопрос, найти информацию.

[Сайт PluginYG2](#) — преимущества и другая информация.

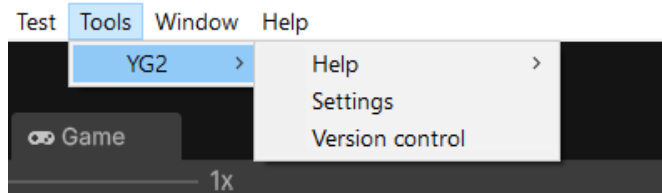
[Видео](#) — полезно для визуального понимания. Но рекомендуется всё же ориентироваться на документацию!

Начало работы

Поддерживаются версии Unity: 2021.3.18+, 2022.3+, 2023.2+ включая Unity 6

После импорта PluginYG2, вас встретит приветственное окно. В нём дублируются кнопки из контекстного меню:

6000.0.26f1) <DX11>



Откройте **настройки** плагина — это фундаментальное окно. Обратите внимание, многие поля имеют всплывающие подсказки!

Рекомендуется ознакомиться со всеми параметрами из раздела **Basic Settings**.

Если при импорте плагина вы отказались от установки оптимальных настроек для стандартной платформы Яндекс Игры, то для ЯИ вам необходимо выбрать шаблон **YandexGame** или включить галку **Auto Apply Settings**. Посмотреть какие настройки применятся можно по ссылке на настройки платформы в поле **Platform**.

Локальный хост

Тестирование игры на локальном хосте с помощью Build And Run работает. При этом инициализация SDK (получение данных платформы) будет игнорироваться.

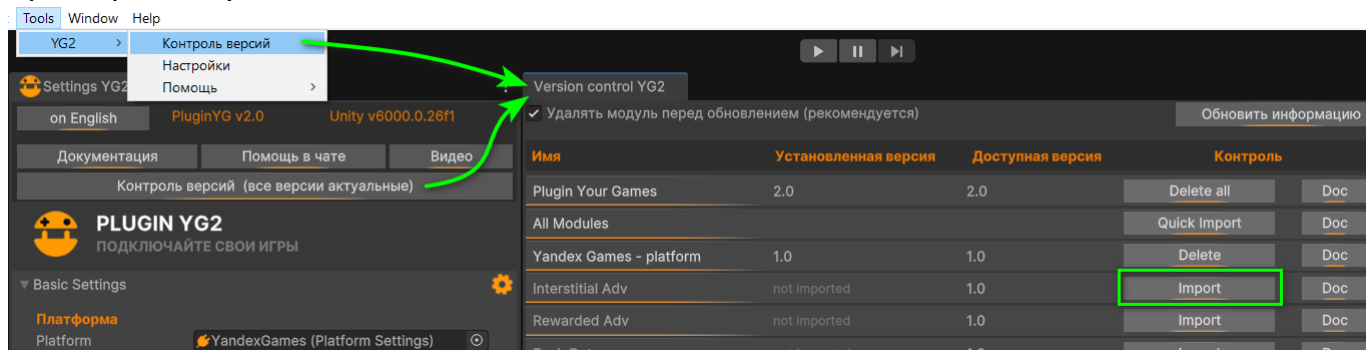
При размещении игры на своём домене на Яндекс Играх, включите опцию Own Host, она находится в настройках платформы (поле Platform в разделе Basic Settings). Данная опция добавит соответствующую строку подключения.

Обратите внимание, ваш хост может не поддерживать некоторые настройки проекта. Например, компрессию Brotli. Если при импорте плагина вы соглашались на установку оптимальных настроек, то PluginYG2 сам меняет некоторые параметры проекта. Для контроля над этим процессом, перейдите в настройки платформы и внесите изменения в соответствии с вашими предпочтениями.

Для локального тестирования игры с подключением к SDK, в Index.html измените функцию IsLocalHost под свои нужды. Данный файл можно корректировать в папке WebGLTemplates/YandexGames.

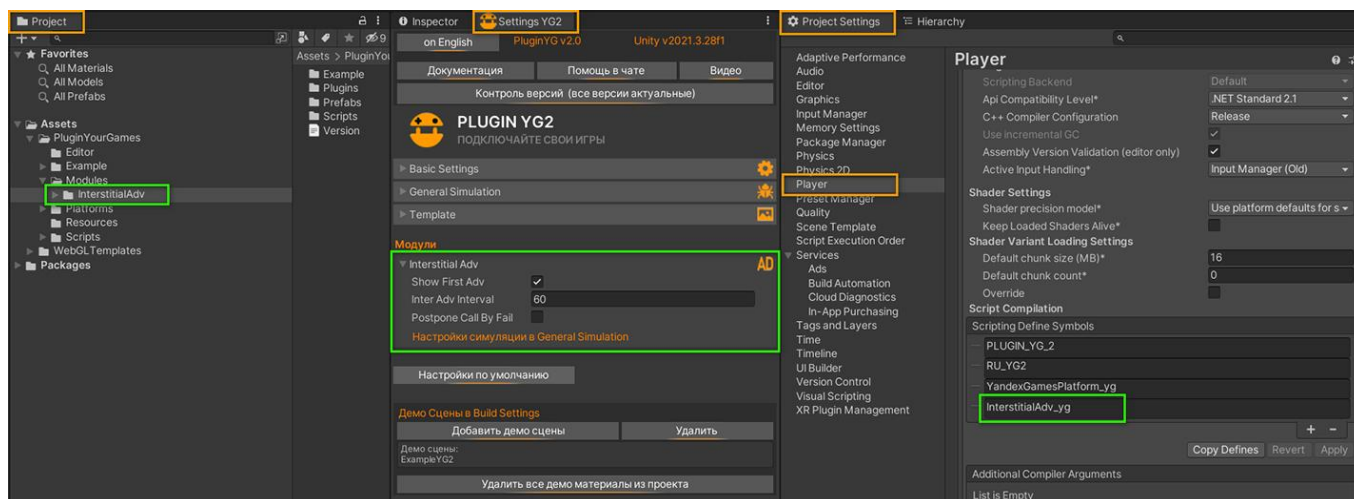
Модульная система

Все функции плагина разделены на отдельные модули. Подробнее о преимуществах такого подхода и в целом о плагине вы можете почитать на [Откройте инструмент контроля версий](#). Давайте скачаем и импортируем для примера модуль **InterstitialAdv**:



Когда модуль импортирован:

1. Он попадает в папку **Modules**. Любая папка в директории Modules будет определена как модуль плагина.
2. Если модуль имеет глобальные настройки, то они отобразятся в настройках плагина.
3. Для всех модулей будет создан свой дефайн ([директива препроцессора](#)). Вы можете использовать его в своём коде.



У дефайнов модулей в конце есть подпись `yg`. Модули - платформы ещё обозначаются надписью `Platform`. При возникновении проблем, вы можете проверить все ли дефайны правильно установлены и скорректировать их. Ещё

можно попробовать обновить данные о модулях в проекте — удалив файл `PluginYourGames/Editor/ModulesListYG2.txt`

Обнаружив **явные** проблемы с плагином, пожалуйста, сообщите об этом в чате [Plugin Your Games](#). Там вы сможете найти обсуждения уже с готовыми решениями, общаться по тематике разработки игр, следить за обновлениями плагина и другой важной информацией.

Импорт всех модулей

Есть «быстрый импорт». Нажмите на **Quick Import** и выберите модули для импорта. У некоторых модулей есть зависимости (другие модули). При импорте нескольких модулей одновременно надо это учитывать.

При критических ошибках в проекте, когда невозможно открыть окно контроля версий, можете вручную скачать и воспользоваться [пакетом](#), который содержит все модули.

Работа с кодом

PluginYG находится в пространстве имён YG, импортируйте его в начале вашего скрипта: `using YG;`

Практически все функции находятся в классе YG2. Так, например, мы можем найти в данном классе метод вызова рекламы: `YG2.InterstitialAdvShow();`

Как работать с рекламой и другими модулями смотрите в соответствующих разделах.

Другие объекты, которые содержатся в PluginYG2 без отдельных модулей:

Основные

`platform` — пример: `YandexGames`, `CrazyGames`.

`onGetSDKData` — событие, сигнализирующее о том, что SDK платформы инициализирован (может пригодиться при использовании опции синхронной загрузки).

`isSDKEnabled` — поле равно true — когда SDK инициализирован.

`isFirstGameSession` — поле, информирующее о том, первая ли сейчас игровая сессия. Если первая — будет равно true всю текущую игровую сессию.

Пауза игры

`onPauseGame` — событие паузы игры. Возвращает bool. Подробнее в разделе Interstitial Adv.

`PauseGame` — метод паузы игры. Принимает параметр bool: true - установить паузу, false - продолжить игру.

Метод `PauseGame` имеет и другие параметры, чтобы при включении паузы можно было выбрать контролировать ли звук, время, курсор мыши, Event System.

`PauseGameNoEditEventSystem` — метод паузы без остановки EventSystem. Его удобно выполнять через EventsYG2 (о нём ниже). Контроль над EventSystem можно полностью отключить с помощью опции Edit Event System (настройки плагина → Basic Settings → Pause Game).

`isPauseGame` — поле равно true — когда игра остановлена.

Фокус игры

`onShowWindowGame` — событие срабатывает, когда игра становится в фокусе.

`onHideWindowGame` — выход из фокуса.

`onFocusWindowGame` — возвращает bool значение.

`isFocusWindowGame` — с помощью этого поля можно узнать в фокусе ли сейчас игра.

Разметка геймплея

GameReadyAPI — метод, который нужно вызывать когда игра загрузилась и пользователь может приступить к геймплею. Необходимо использовать, если в Basic Settings отключена галка Auto GRA.

По умолчанию данный метод выполняется автоматически после загрузки игры, потому что опция AutoGRA в настройках плагина включена. Если в вашей игре после её загрузки есть ещё дополнительные задержки - выключите AutoGRA и вызывайте метод вручную.

GameplayStart — используйте этот метод, чтобы дать сигнал платформе о начале геймплея.

GameplayStop — используйте этот метод, чтобы дать сигнал платформе об остановке геймплея.

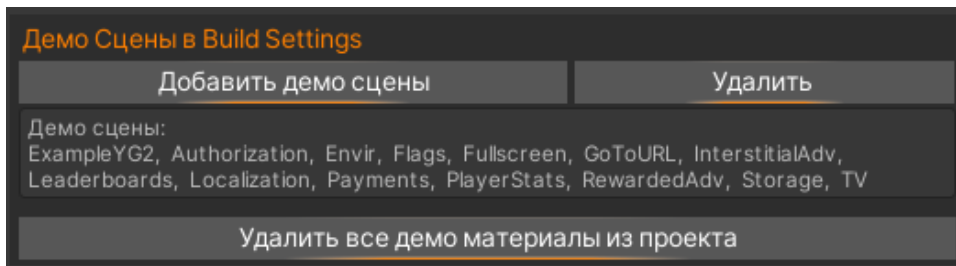
При открытии и закрытии рекламы или других паузах - выполняются методы **GameplayStart** и **GameplayStop**. Но только если раньше в игре был вызван метод **GameplayStart**.

То есть, если в коде использовать методы разметки, то они будут автоматически выполнять при просмотре рекламы и других схожих паузах. А после паузы будет установлено состояние геймплея такое, какое было до паузы

isGameplaying — поле, информирующее о том, идёт ли сейчас геймплей.

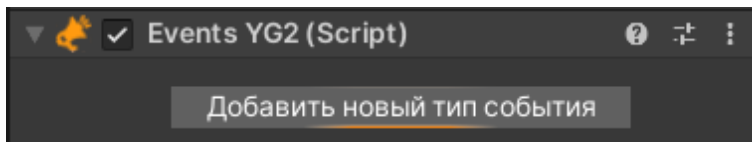
YG2.optionalPlatform.HappyTime — счастливое время.

Демо материалы



Все демонстрационные материалы находятся в папках **Example**. Откройте главную демо-сцену: PluginYourGames/Example/Scenes/**ExampleYG2**. Главная сцена отображает демо-объекты всех модулей. Если у модуля есть сцена с демо-объектом, то он отобразится. Но сначала нужно нажать на кнопку "**Добавить демо сцены**". Тогда сцены модулей добавятся в [Build Settings](#). Перед релизом игры или по завершению изучения плагина можно удалить демо сцены из Build Settings нажав на кнопку "**Удалить**" или выполнить удаление всех демо материалов из проекта. При удалении какого-либо модуля — сцена модуля останется в Build Settings и будет неопределённой. Перед удалением модуля желательно в разделе демо сцен нажать на кнопку "Удалить".

EventsYG2 компонент



- Компонент EventsYG2 хранит в себе большинство доступных методов плагина. С его помощью без кода можно, например, повесить на кнопку метод вызова рекламы.
- Второе назначение EventsYG2 — выполнять методы по событию. Для этого нажмите на кнопку "**Добавить новый тип события**" и выберите событие, при котором необходимо выполнить какой-либо метод. Это работает также, как вы привязываете методы на клик по кнопке в компоненте **Button**.

Сборка билда

Теперь вы знаете основу работы с Plugin Your Games 2.0 и с этими знаниями уже можно собирать рабочий билд. Но есть ещё пару интересных моментов. Далее я расскажу о номере билда и немного о правильной настройке проекта под WebGL.

Номер билда

PluginYG2 хранит информацию о билде в файле

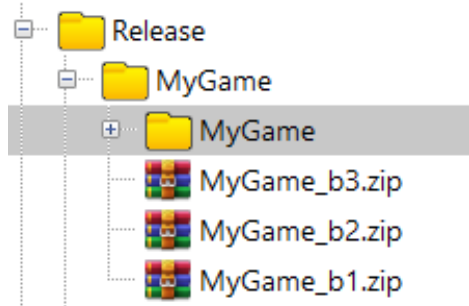
PluginYourGames/Editor/**BuildLogYG2.txt**. Туда записывается номер билда.

Например, если вы собираете проект при импортированном PluginYG третий раз, то номер билда будет = 3. Этот номер отображается в:

- Loge Unity Editor после сборки проекта
- В названии zip архива билда (b3)
- В консоли браузера (в первых логах)
- В углу экрана игры — при включенном **Developer Build** режиме (подробнее далее)

(Для обнуления информации о билде можно удалять файл **BuildLogYG2.txt** и даже всю папку PluginYourGames/**Editor**)

После сборки проекта, если включена галка **Archiving Build** в настройках плагина в разделе **Basic Settings**, то билд игры будет автоматически заархивирован в формате **zip**, чтобы каждый раз не архивировать вручную. PluginYG не удаляет оригинальный билд и не помещает внутрь него архив, т.к. плагин не нарушает какую-либо базовую логику Unity при сборке билда. Поэтому расположение созданного архива может показаться странным. Для удобства — внутри папки, в которую собираетесь построить билд — создайте ещё одну папку с таким же названием игры. И уже внутри неё собирайте билд игры:

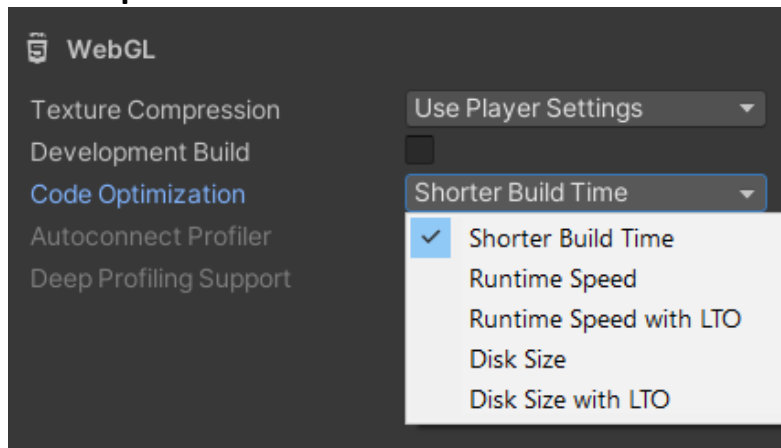


Настройки проекта

В настройках плагина в разделе **Template** вы можете включить галку **Developer Build** во время тестирования игры. Тогда в сборку попадут скрипты, отображающие номер билда в углу экрана игры. Это даёт возможность мгновенно убедиться, что на сервере платформы актуальный билд и сигнализирует о том, что данный билд является не финальным. На плашку с номером билда можно нажать, чтобы она исчезла.



Перед финальным билдом необходимо выключить **Developer Build** режим плагина или одноимённый режим от Unity. Также в **Build Settings** для WebGL есть опция **Code Optimization**:



Для экономии времени при сборке тестового билда установите параметр **Shorter Build Time**. Для релизного билда **Runtime Speed with LTO**. Подробнее смотрите в [официальной документации](#).

Другие основные настройки проекта под WebGL плагин устанавливает автоматически, если вы это ему позволили, [подробнее выше](#).

По умолчанию не применяется опция [Color Space](#). Для WebGL рекомендуется использовать значение **Gamma**. Игры для IOS будут работать только в Gamma режиме. Другие устройства также могут не поддерживать иной Linear режим. Это же касается и для опции **Auto Graphic API** (рекомендуется включить).

Из основных рекомендаций ещё можно отметить опцию **DSP Buffer Size** для звуков — настройка оптимизации звука, которая ещё может помочь убрать проблему с посторонними шумами если такая имеется. А в настройках самих звуков для WebGL параметр [Load Type](#) рекомендуется выставлять на **Decompress On Load**.

Ещё проблема у новичков — не отображается шрифт в билде на WebGL. Нельзя использовать стандартный Unity шрифт, замените его.