# Еще один способ интеграции с Unity3D

В предыдущем разделе мы проделали достаточно много действий, чтобы интегрировать Code и Unity3D. Но есть и более простой механизм. Это использование [Unity plugin](https://github.com/dotBunny/VSCode/) от @reapazor, что позволит значительно упростить работу.

Как Вам известно, интерфейс Unity легко расширить, используя сторонние компоненты, исходный код которых можно разместить в дереве Вашего проекта. При этом Unity тут же откомпилирует его и на основании атрибутов добавит новые пункты меню, расширив стандартны набор.

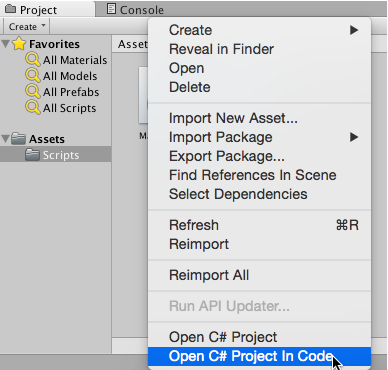
В нашем случае, исходный код расширения можно получить, клонировав проект:

**git clone https://github.com/dotBunny/VSCode.git**

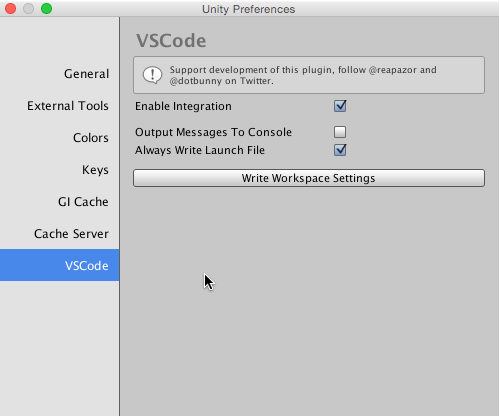
или просто скачав zip архив с сайта.

Само расширение состоит всего из одного файла, который мы должны добавить в наш проект. Если Вы уже перешли на Unity 5.2, то можете добавить этот файл в любое место. В более ранних версиях Вы должны разместить его в каталоге Assests/Plugins/Editor.

Как только Вы добавили файл расширения, то сможете найти меню **Open C# Project in Code**:



Используя этот пункт меню Вы можете открыть «правильный» проект в Code не прилагая больших усилий. Но это не все расширения, которые позволяет использовать plugin. Давайте откроем окно Unity->Preferences и перейдем на вкладку VSCode:



Сделаем обзор основных опций в этом окне:

* **Enable Integration** – этот переключатель, выбран по умолчанию. Если Вы выключите интеграцию, то Unity будет использовать стандартный редактор кода по умолчанию;
* **Write Launch File** – по умолчанию Вы не можете отлаживать проект. Чтобы включить эту возможность, Вам необходимо создать и настроить launch.json файл. Используя этот переключатель Вы можете включить автоматическое создание файла launch.json. Обратите внимание, что файл будет создан, как только Вы запустите свою игру в плейере Unity;
* **Write Workspace Settings** – С помощью этой кнопки Вы можете скрыть все ненужные файлы, которые присутствуют в каталоге, но являются рабочими;

Таким образом, мы затратили минимум времени, а получили больше чем в предыдущем разделе, выполняя настройку в ручном режиме.