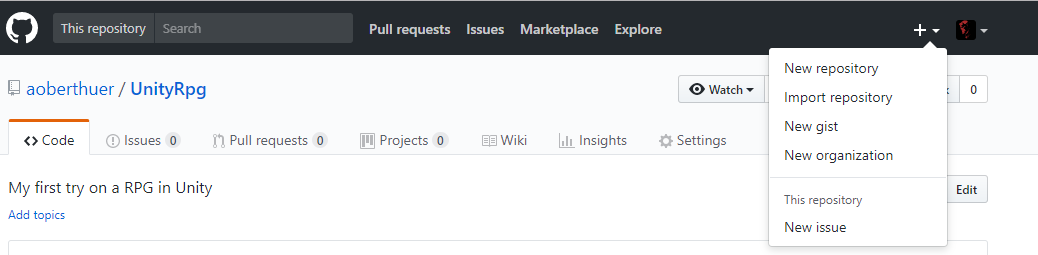
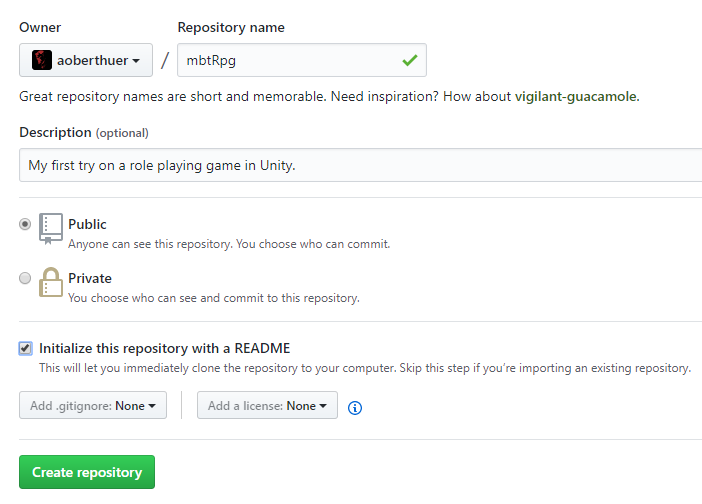
## General setup

#### Create repository on GitHub

Create you repository on github.com

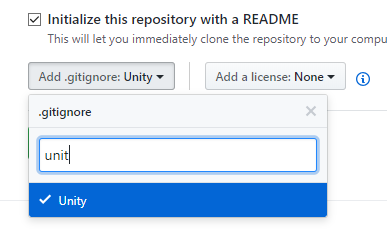
The option can be found in the upper right corner under the ‘+’ menu.





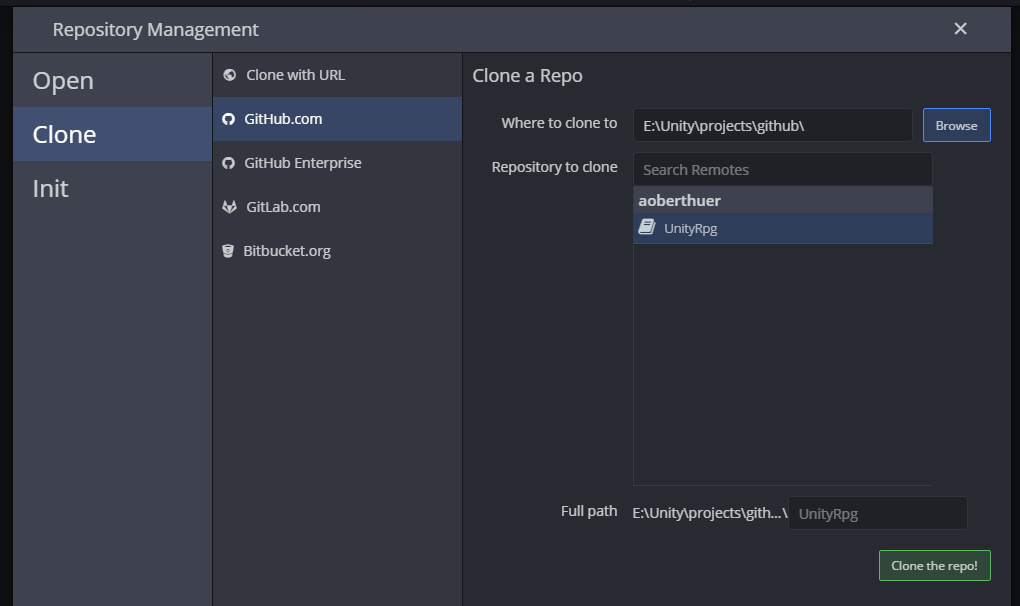
On creation do check the ‘Initialize this repository with a README’ as otherwise you cannot clone it to your local computer. It also says so in the text ;-)

Also use the Add .gitignore option, in this case for Unity:



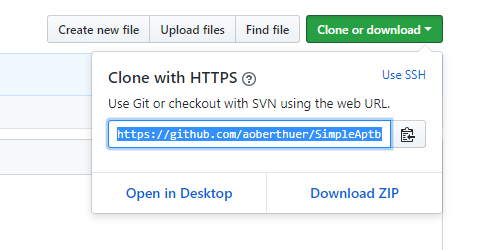
### Clone repository

Clone Git repository to your local computer. See settings from GitKraken here for clone repository action. This will work out of the box if you check the initialize with readme box on creation.



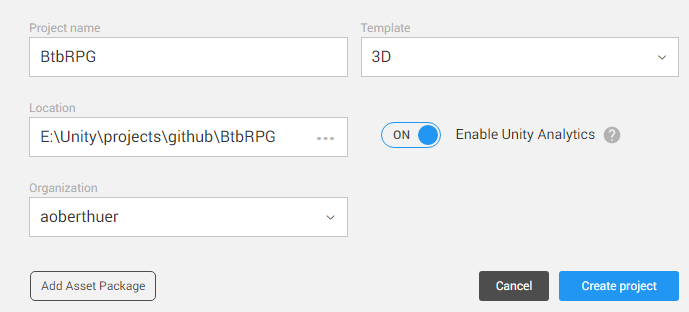
Select the repository you just created in step 01 as repository to clone. Select local path and click button.

**Update:** An easy alternative is the option clone with URL. Just open the ‘Clone or download’ button in GitHub and copy the URL from GitHub;

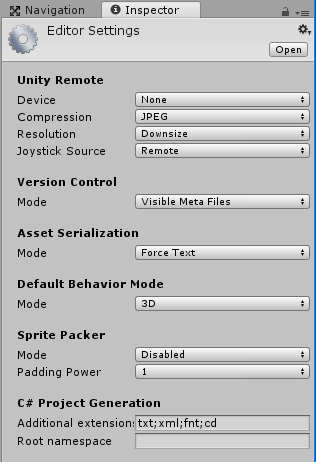


### Start new Unity project

Start a new project for Unity. Select path to repository. The project will live with its project top folder within the repository folder.



Next setup asset serialization to be text only: Edit / Project Settings / Editor / Asset Serialization and in the same place Version Control Mode to ‘Visible Meta-Files’. From the docs:



See this [link](https://docs.unity3d.com/Manual/ExternalVersionControlSystemSupport.html). Here’s the excerpt:

Before checking your project in, you have to tell Unity to modify the project structure slightly to make it compatible with storing assets in an external version control system. This is done by selecting Edit->Project Settings->Editor in the application menu and enabling External Version Control support by selecting Visible Meta Files in the dropdown for Version Control. This will show a text file for every asset in the Assets directory containing the necessary bookkeeping information required by Unity. The files will have a .meta file extension with the first part being the full file name of the asset it is associated with. Moving and renaming assets within Unity should also update the relevant .meta files. However, if you move or rename assets from an external tool, make sure to synchronize the relevant .meta files as well.

### Git ignore

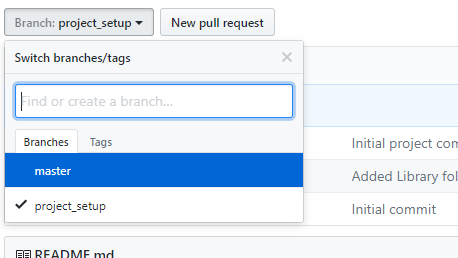
If not done in step 01, next add Library folder to git ignore (you can only do this on staged changes, also set this folder to stop tracking). This will create a .gitignore file. You can now commit your files and push them to the central repository.

Edit the .gitignore file on the file system and replace content with latest content from:

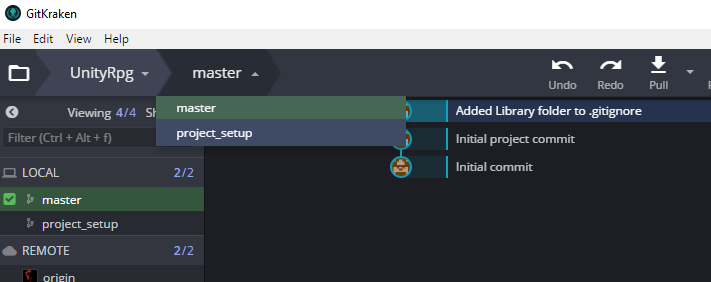
<https://github.com/dvcs/gitignore/blob/master/templates/Unity.gitignore>

### Create project setup branch (optional)

Create GitHub branch for basic-project-setup and push project into GitHub branch.



At GitHub just click on the Branch: [current branch name] - button and type name of new branch. Use ‘project\_setup’ or any other fitting name here. Locally switch to new branch in GitKraken.



Select new project\_setup branch here. Result is that project\_setup is now the current branch for setting up your project like e.g. importing standard assets. This is how it looks once pulling the new branch is done:

