Instructions for Using Gamecloud with Unity

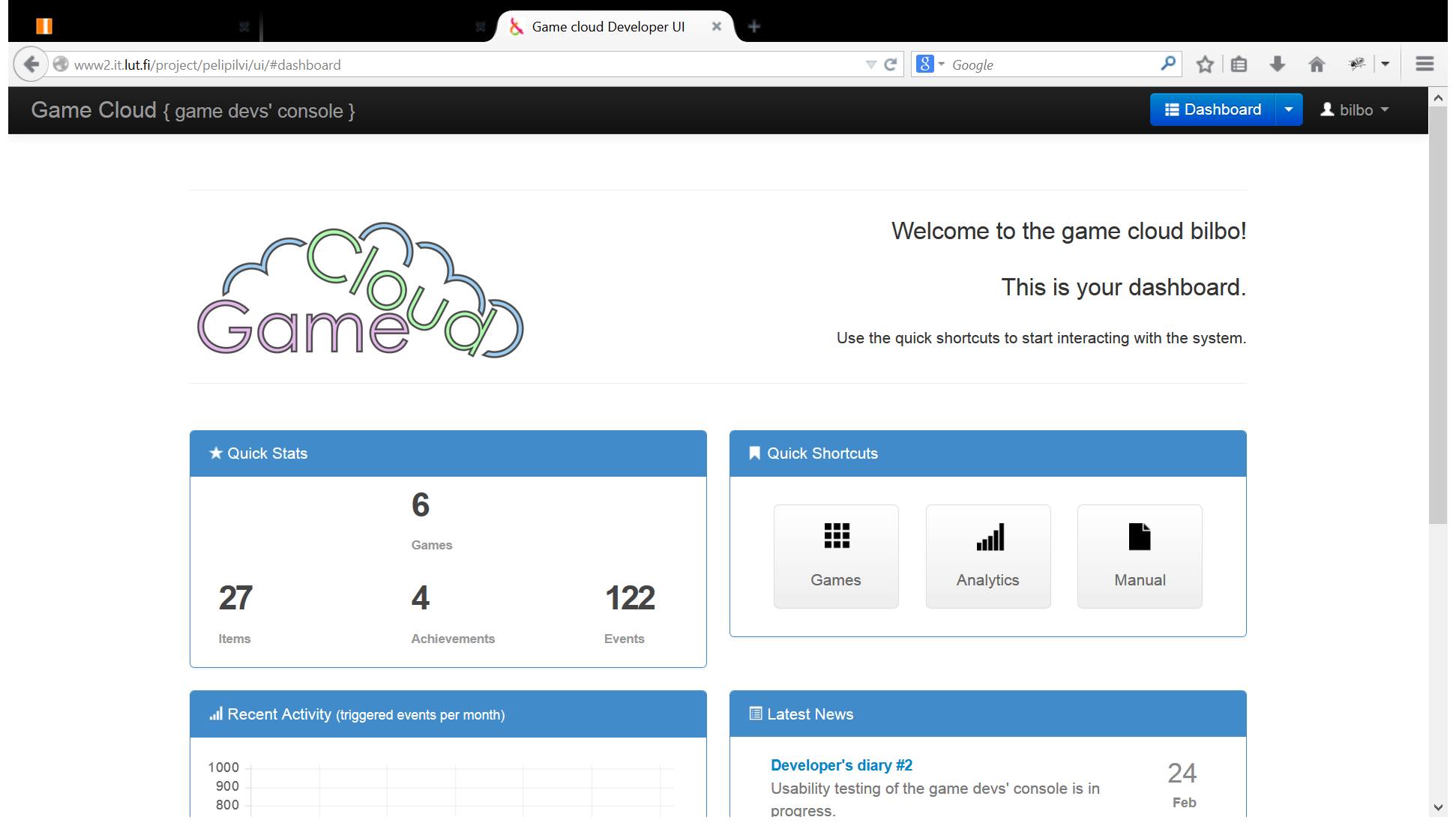
# 1) Get User Account & Password

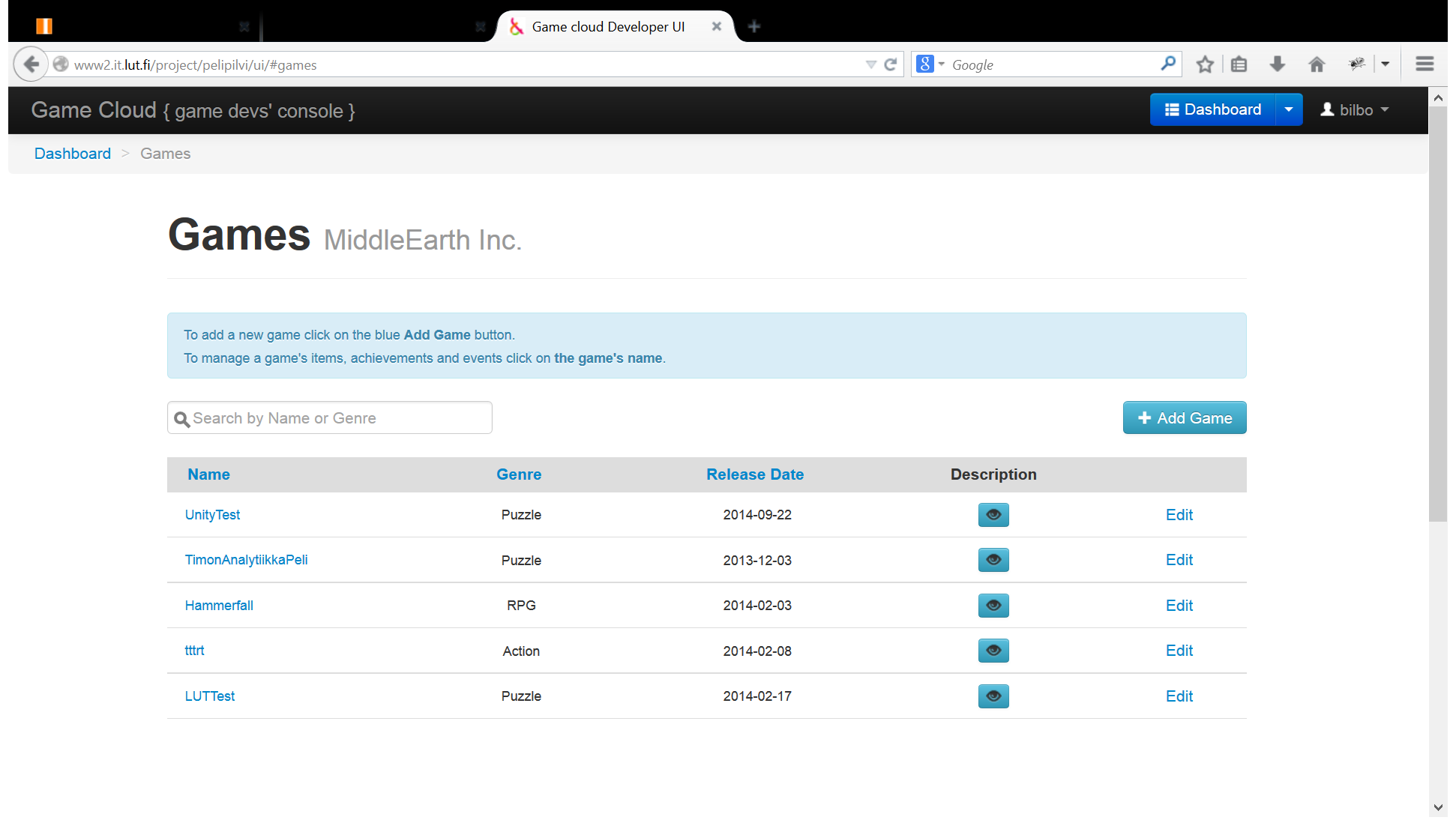
This is currently given to you on request. The system does not accept new accounts otherwise

# 2) Login to gamecloud account

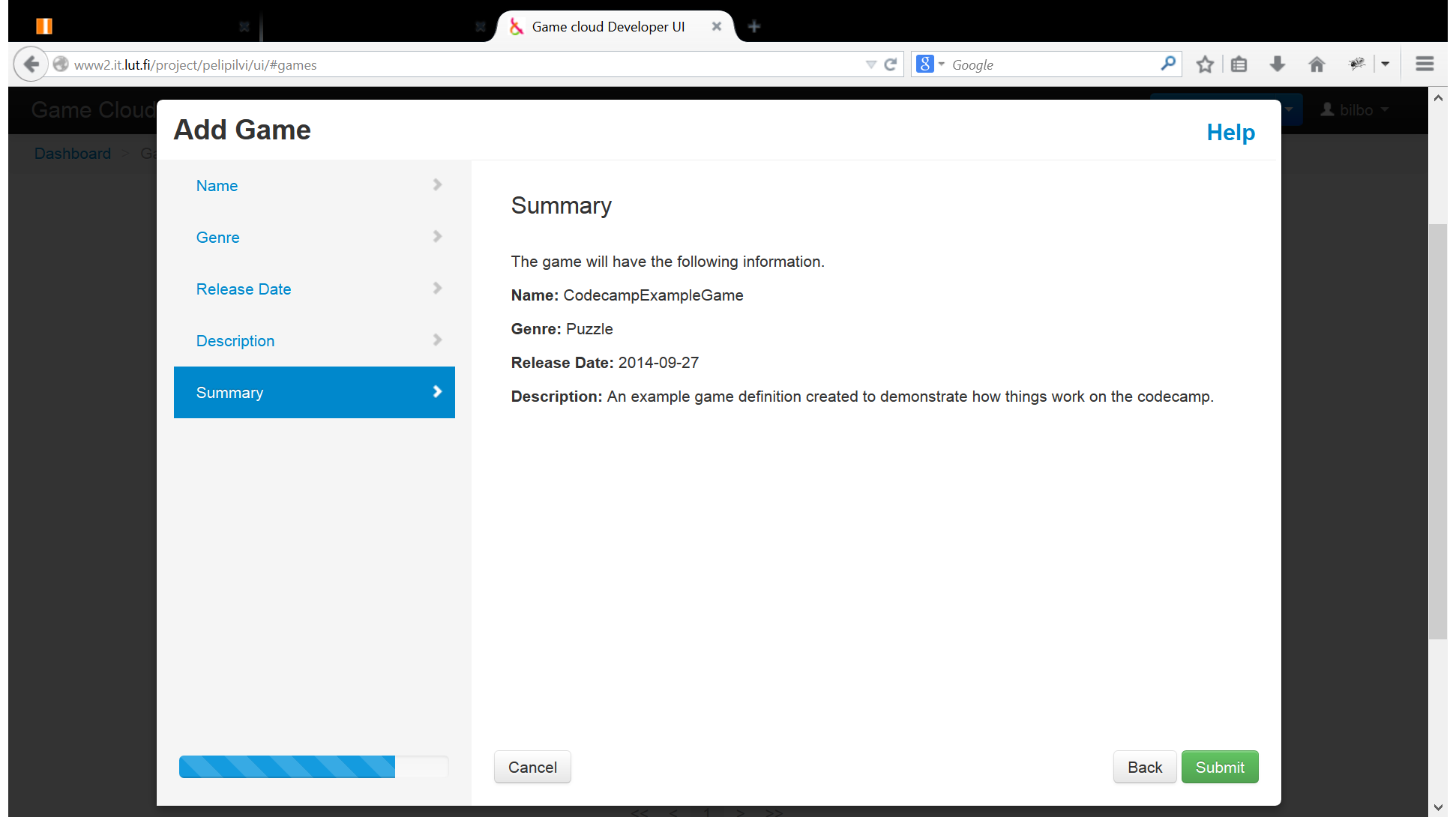
The address for development UI is: <http://www2.it.lut.fi/project/pelipilvi/ui/>

# 3) Create a new game





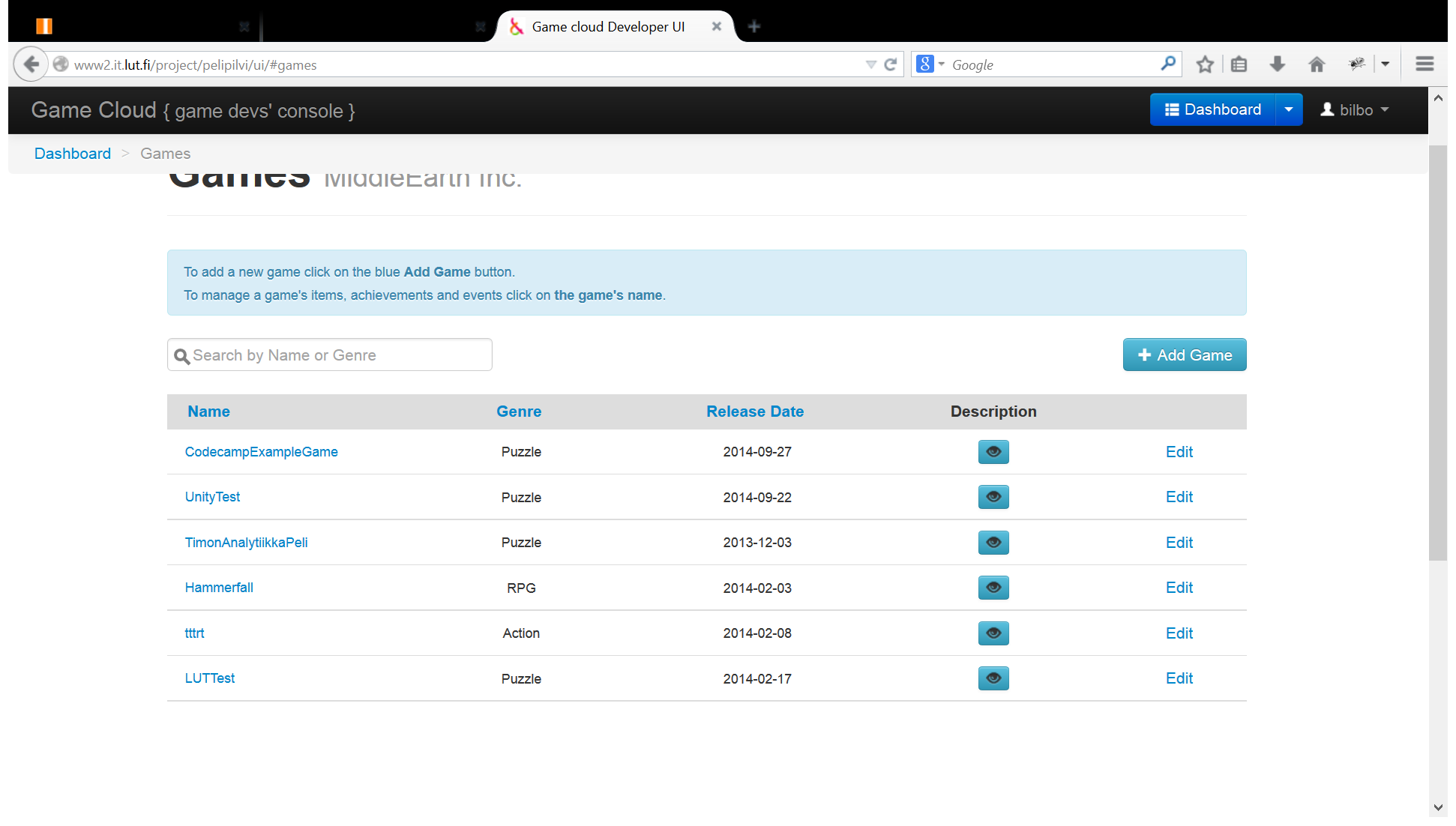
Adding game asks few basic information of the game you are making. The name of the game, Genre of the game, release date and a short descriptive summary.



And once you click submit, the things will get saved to the system. **Note! You cannot change this information currently, so make sure you don’t have big typos!**

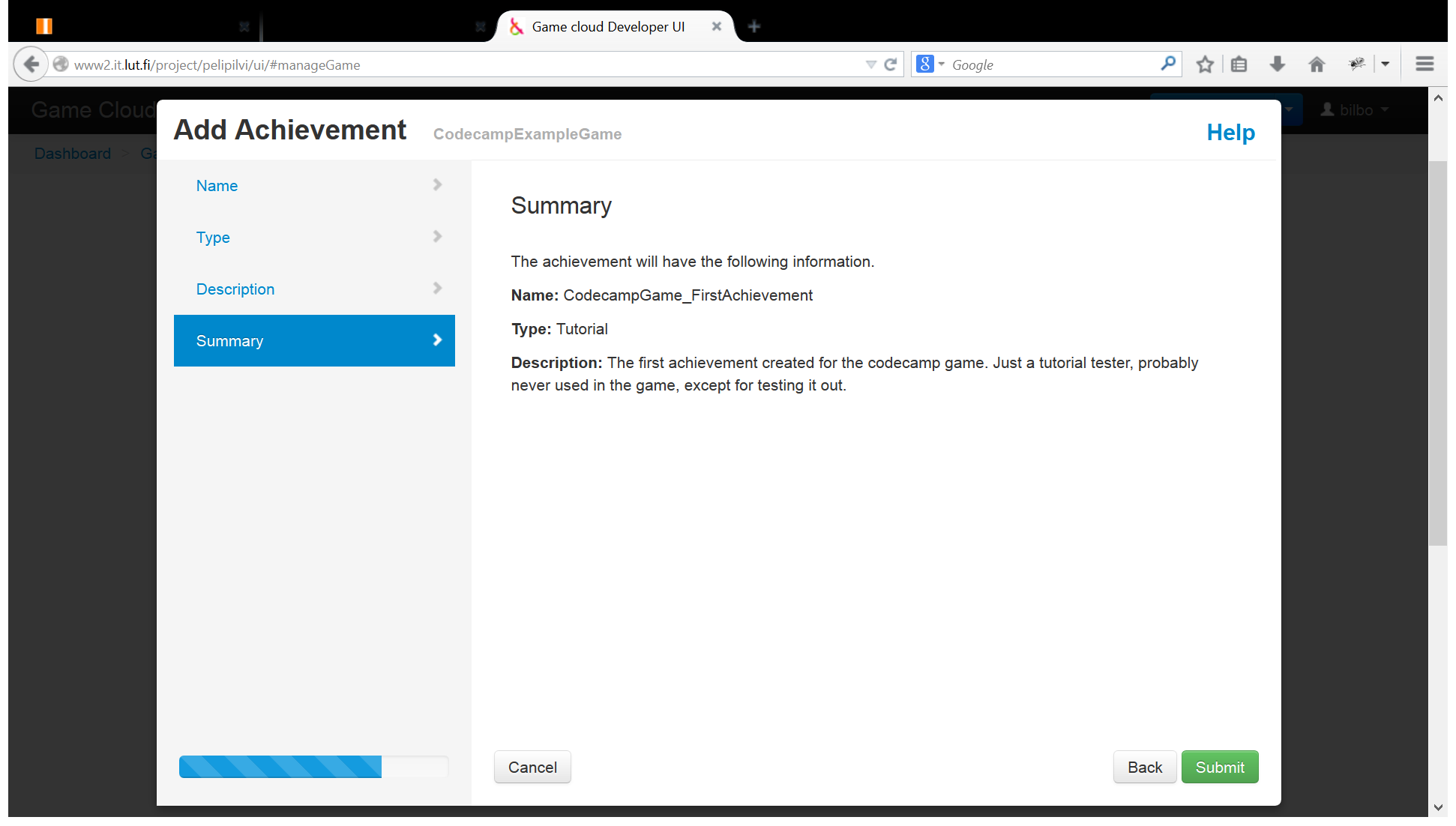
# 4) Defining content to the system

- First, select the game you just have created

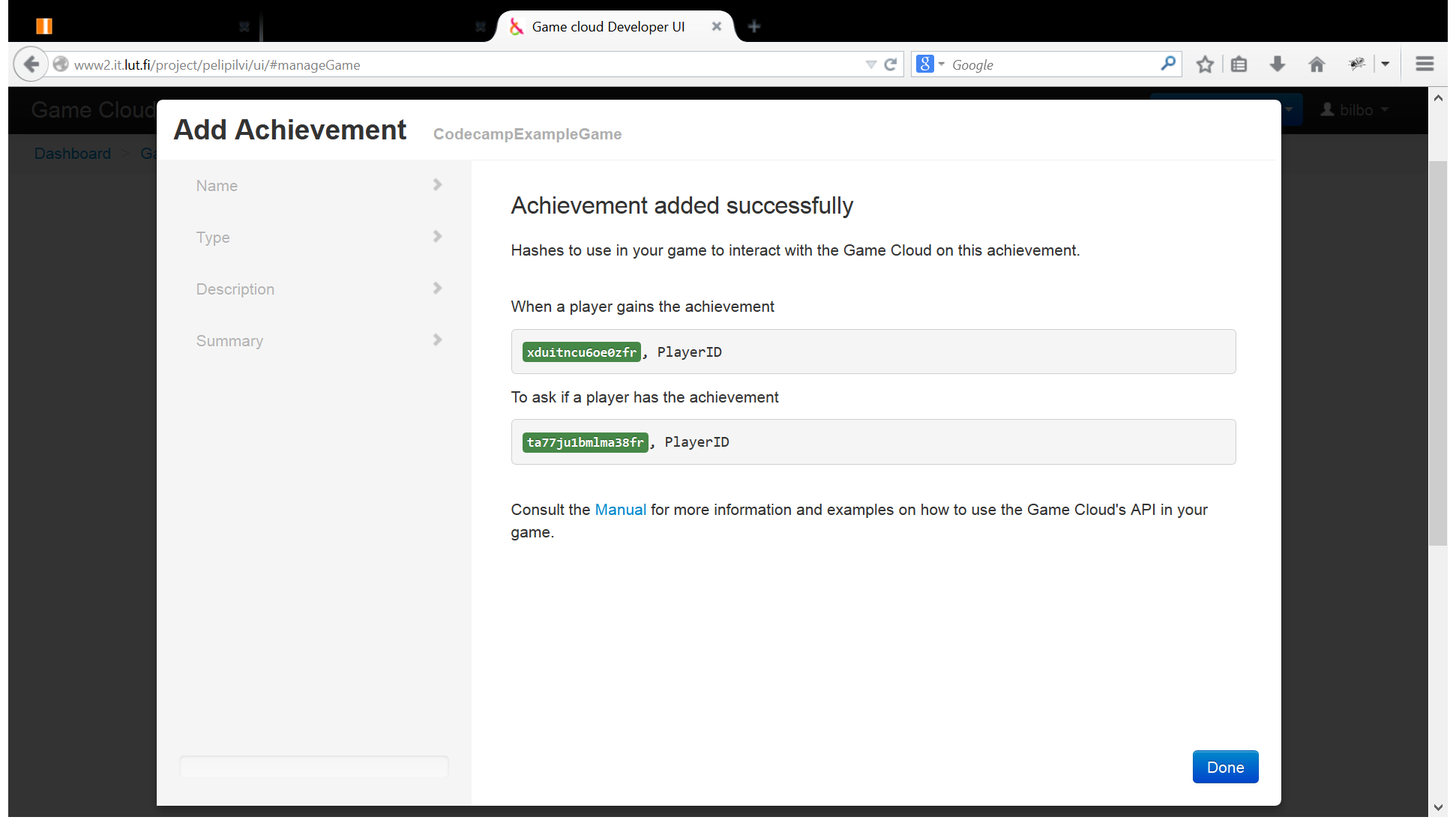


- This will take you to a page that displays all the Events, Items and Achievements defined in the system. Here you can define all this information for your game.

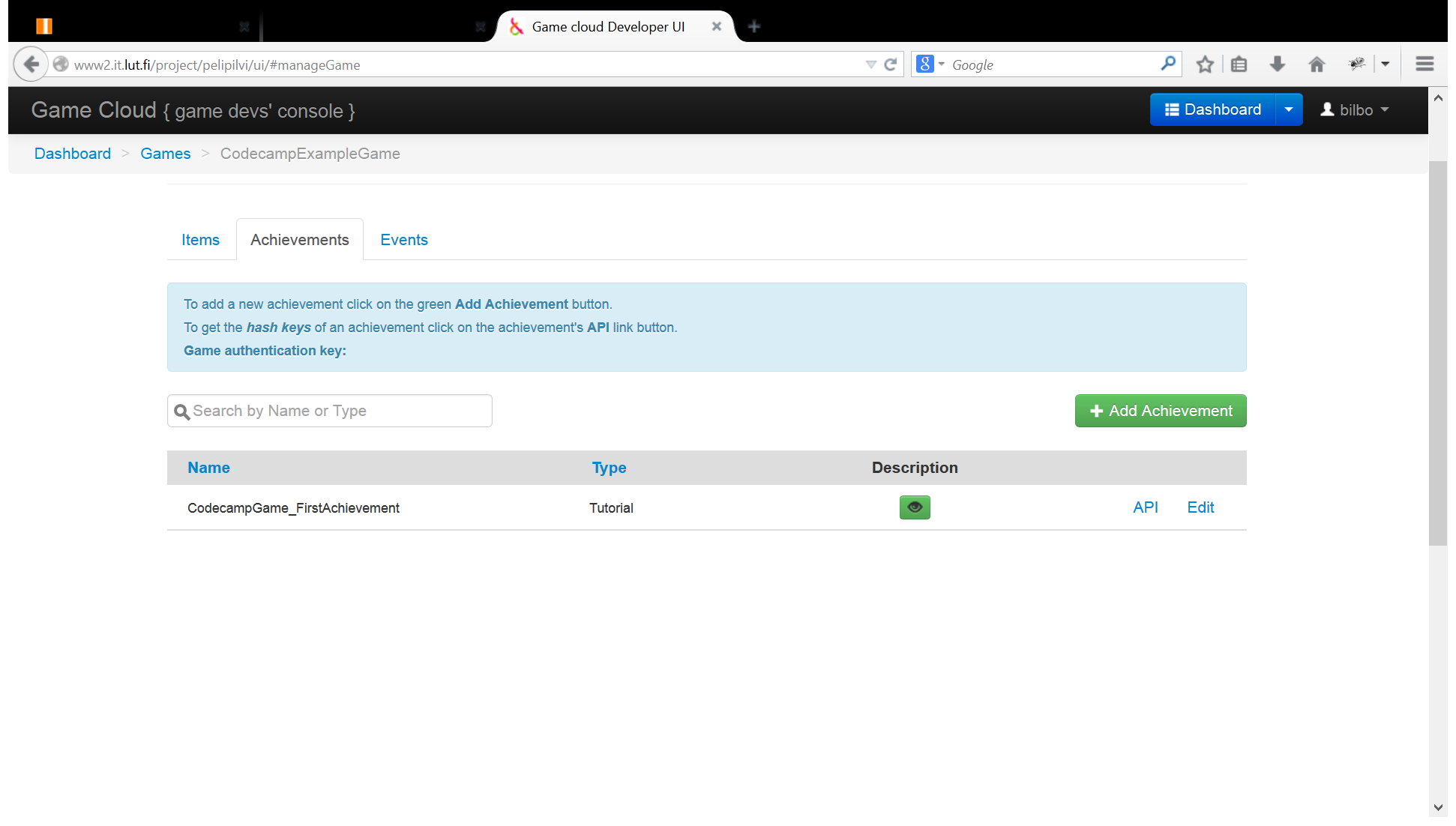
As an example, let’s create an achievement to the system



And then you will see a page containing information on how to ask if the player has the defined achievement and how to give the achievement to a certain player.



The same view can be seen by clicking the API link on the game information screen



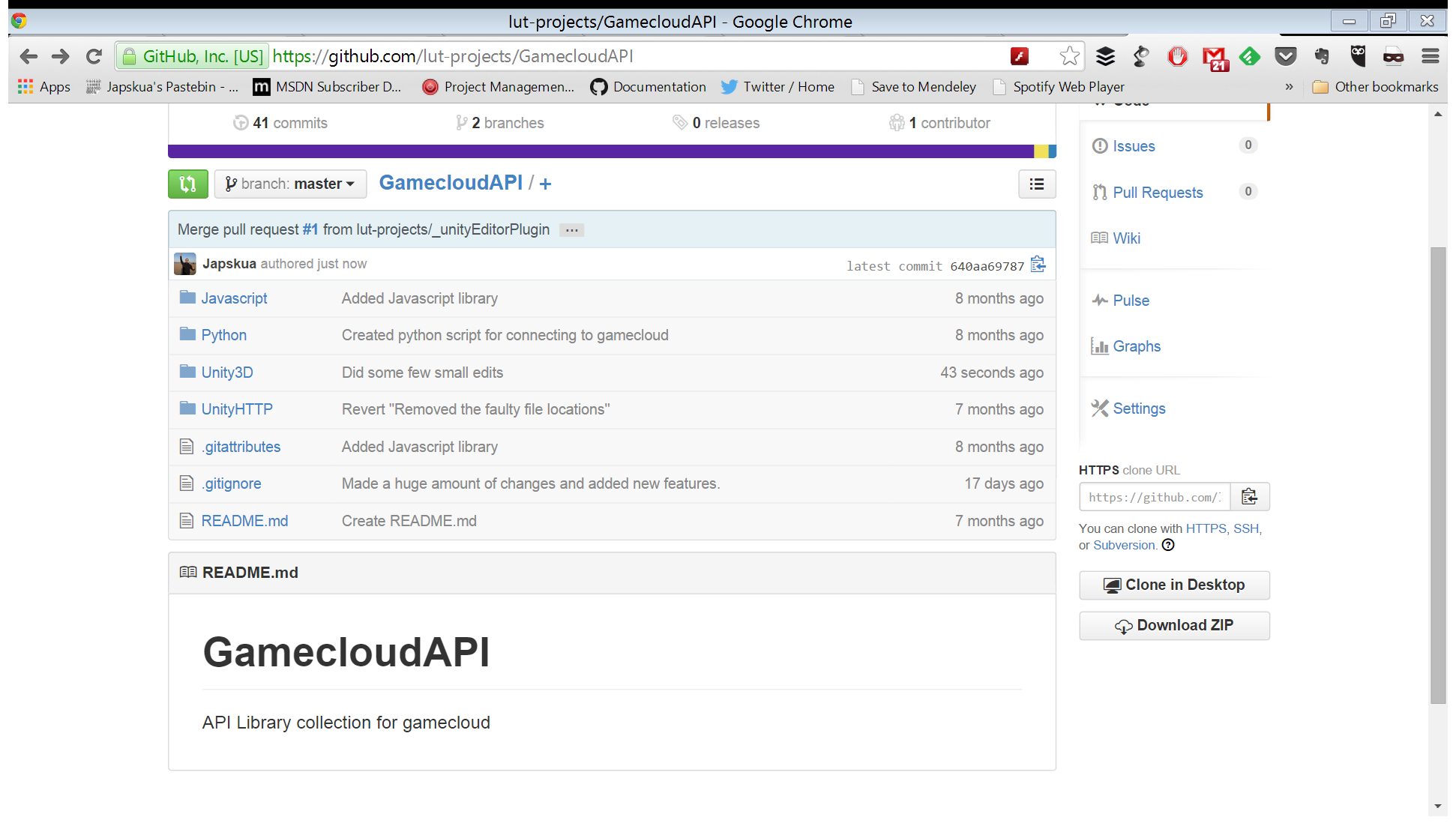
# 5) Calling the functions inside MY own code

**The manual on Developer UI is the best source for information**

However, here is a short demonstration how to use it inside Unity

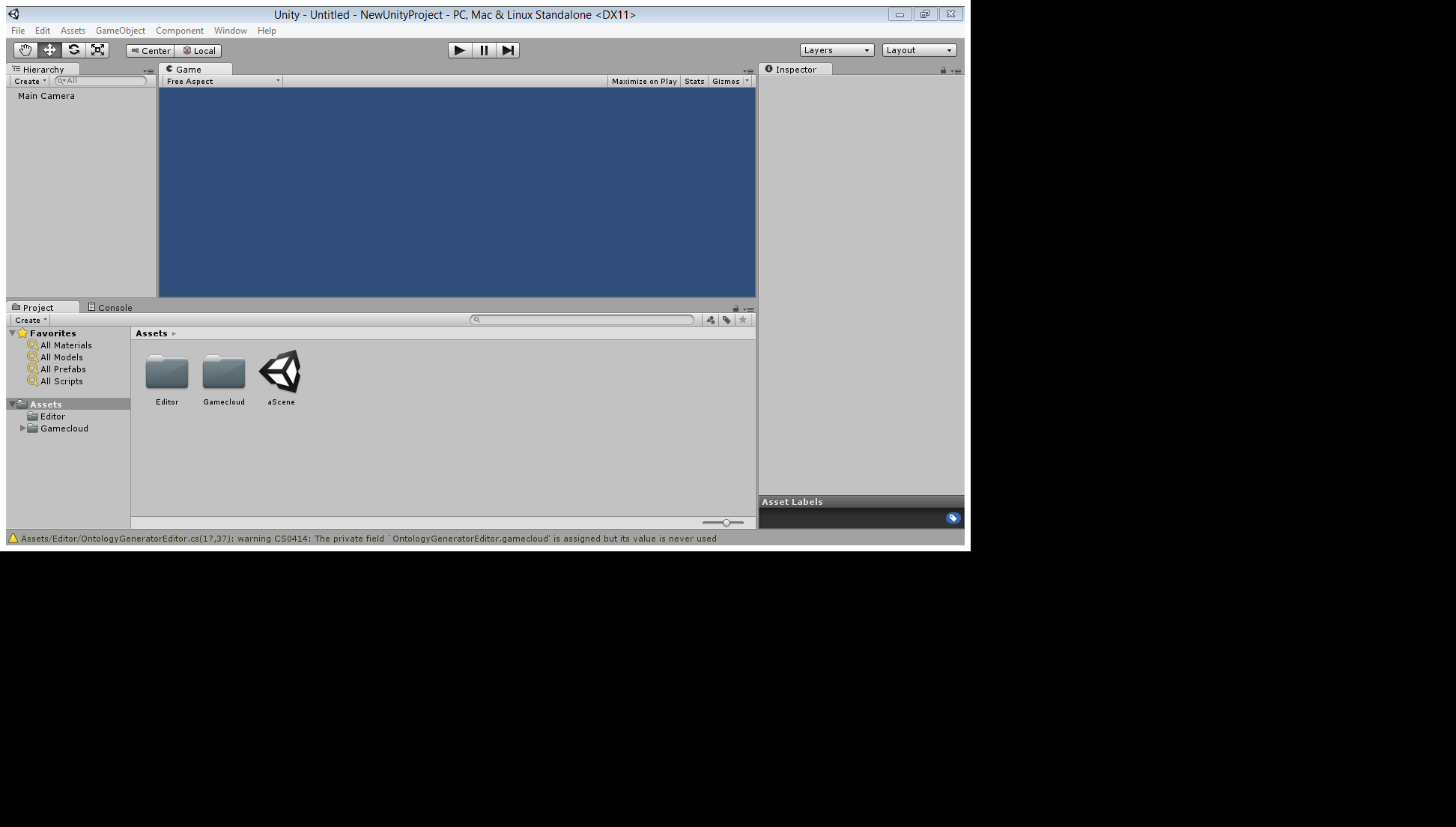
First, go to: <https://github.com/lut-projects/GamecloudAPI>

And download .zip file, if you don’t have installed on your computer



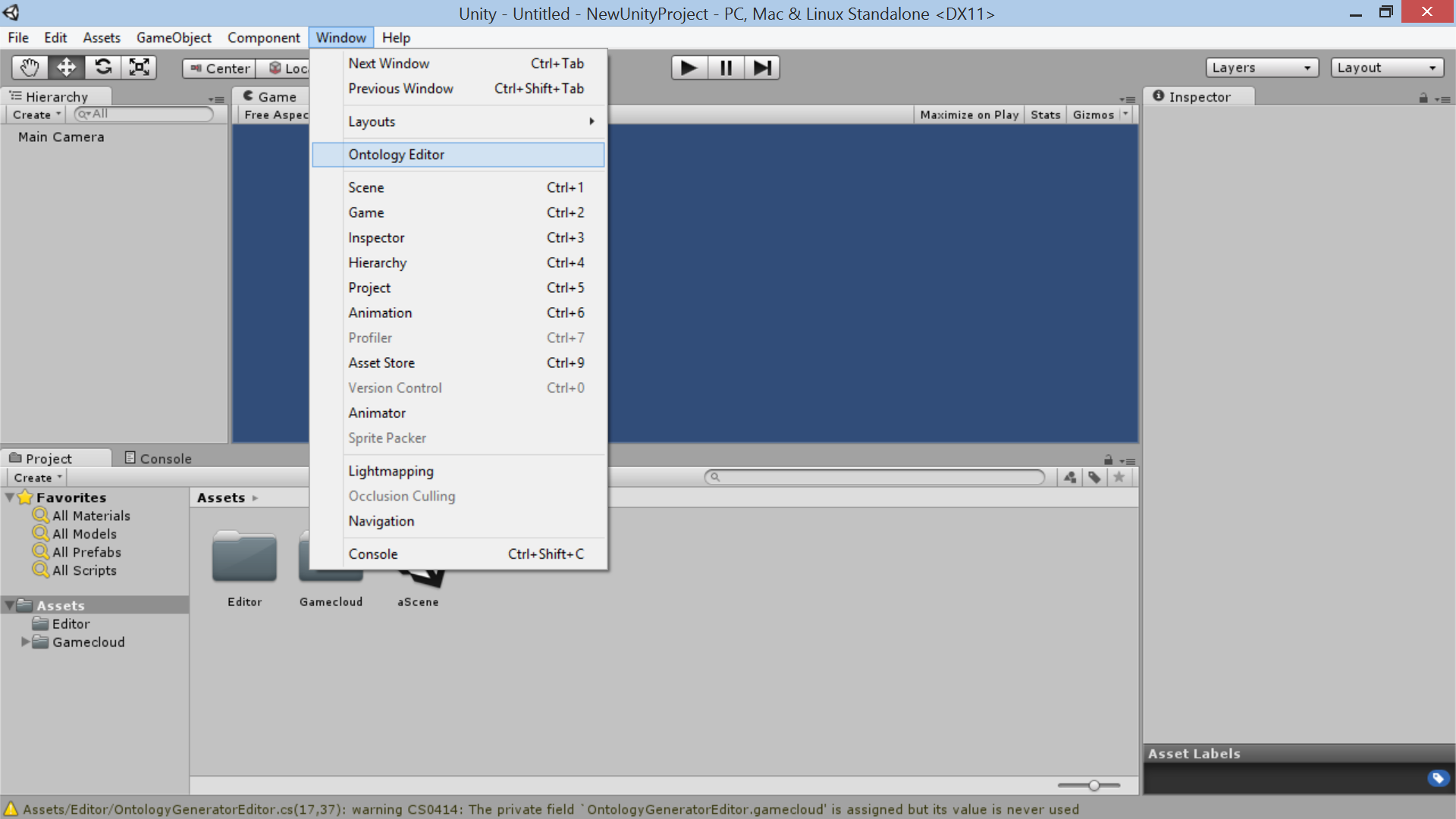
## Next up: Add the files to your Unity3D project

* First, extract the Unity3D/Assets contents into the /Assets/ folder of your Unity Project



Next, you can load the OntologyEditor Window straight in Unity by selecting

Window -> Ontology Editor



This will open a new Window for you, that can be used to manipulate Unity stuff

## Login & Functions of Ontology Window

### 1 Gamecloud Connection Settings

* Gamecloud Adress: The complete address to the system
  + <http://54.220.223.184:8888/api/1/>
* Test Connection button should show ”Nothing Happens” message in the Console
* Username & password should be quite self-explanatory
* Login connects to the system (and shows a bunch of debug info in the Console)

After successful login, you can use stuff in the system

## 2 Select/Create Current Game

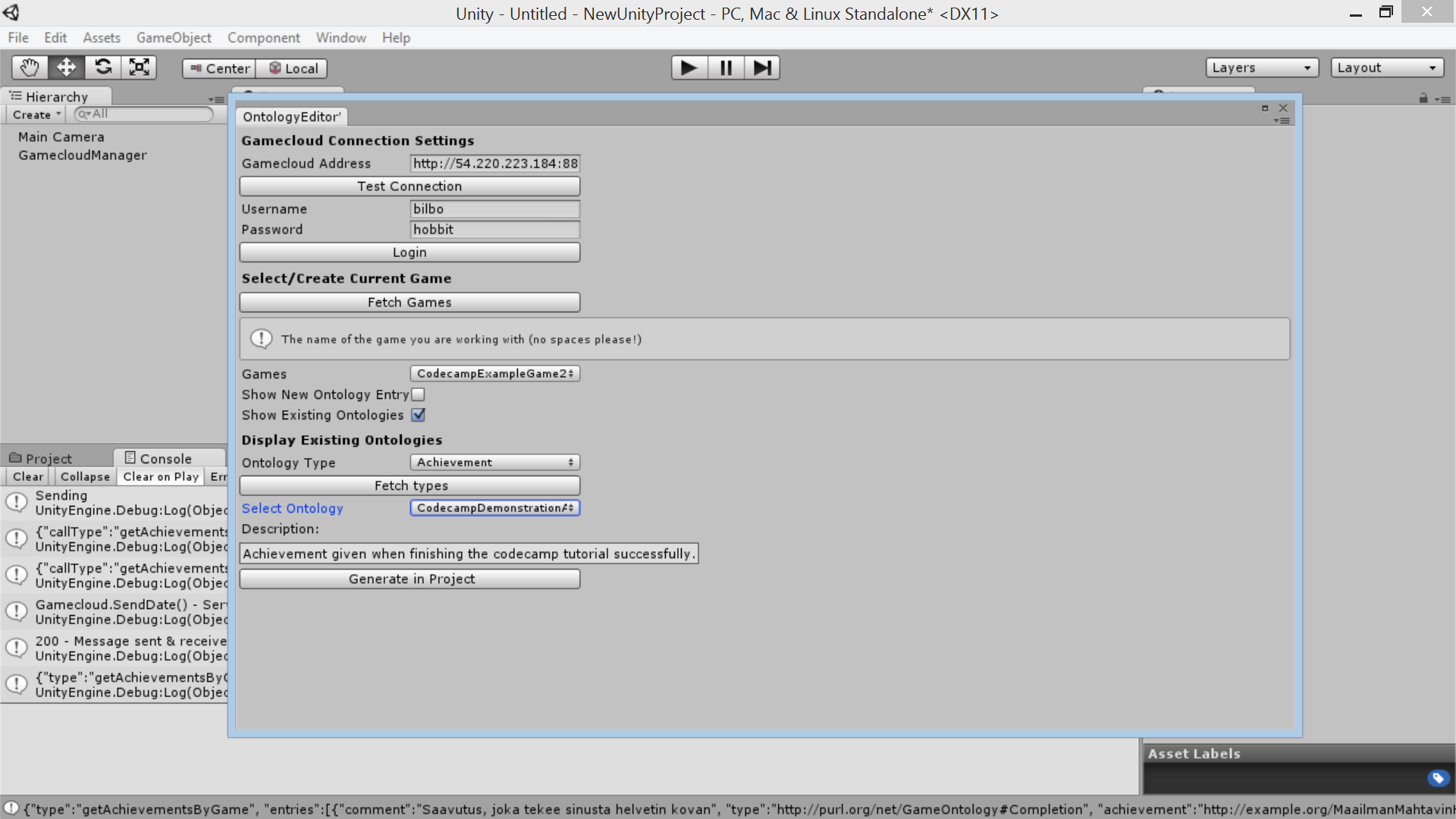
This part is used to fetch the games that you have created in the system. The CREATE part is not available currently, so the Games need to be created through Developer UI Website

* Fetch games gets all your games from the gamecloud

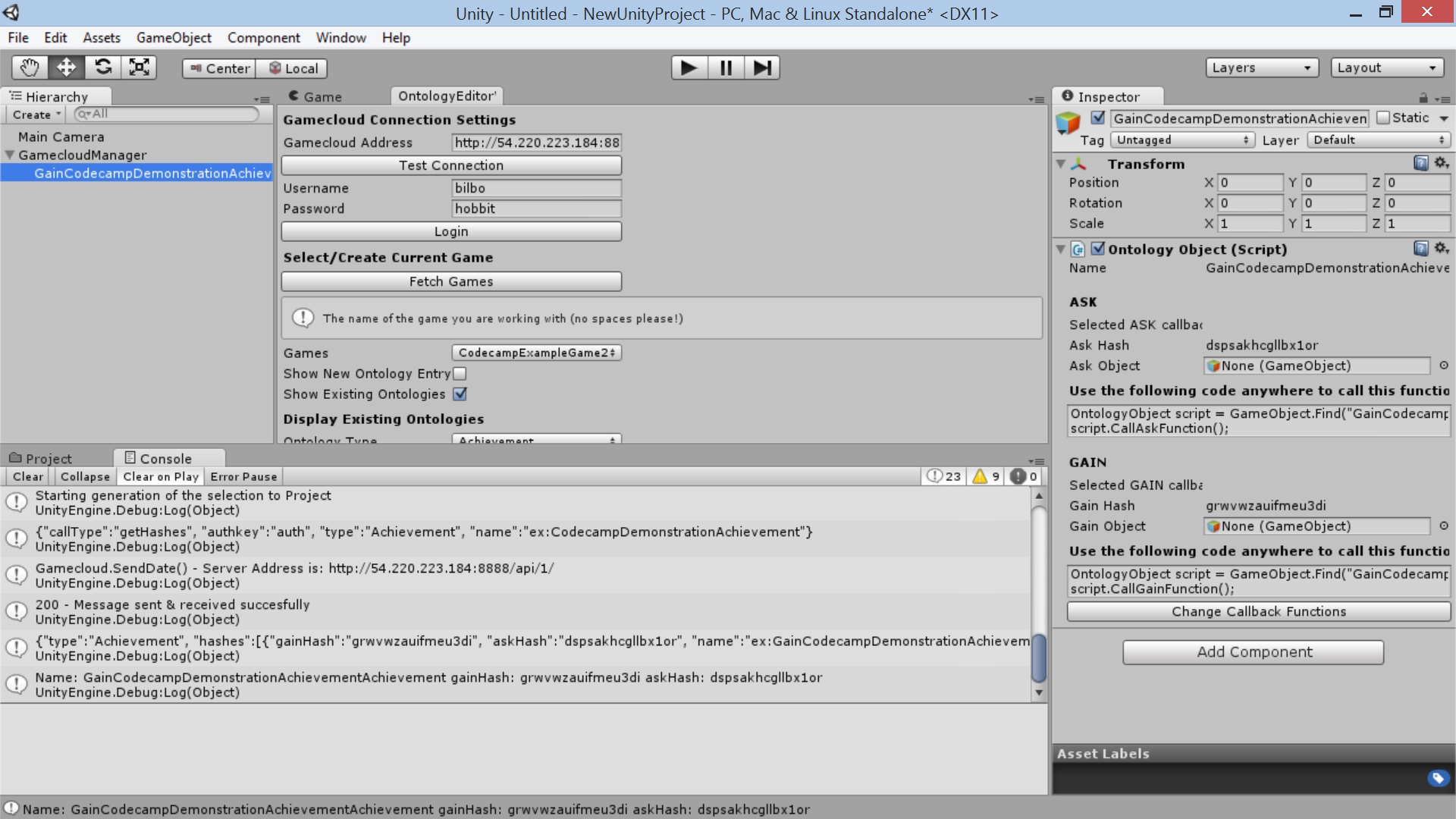
After fetching the games, you will see a dropdown list of all your games. Select one and tick “Show Existing Ontologies”

## 3 Display Existing Ontologies

This view is used to generate the abstract ontology information in your Unity project. Select the type from the list and click “Fetch” to get all entries of that type.

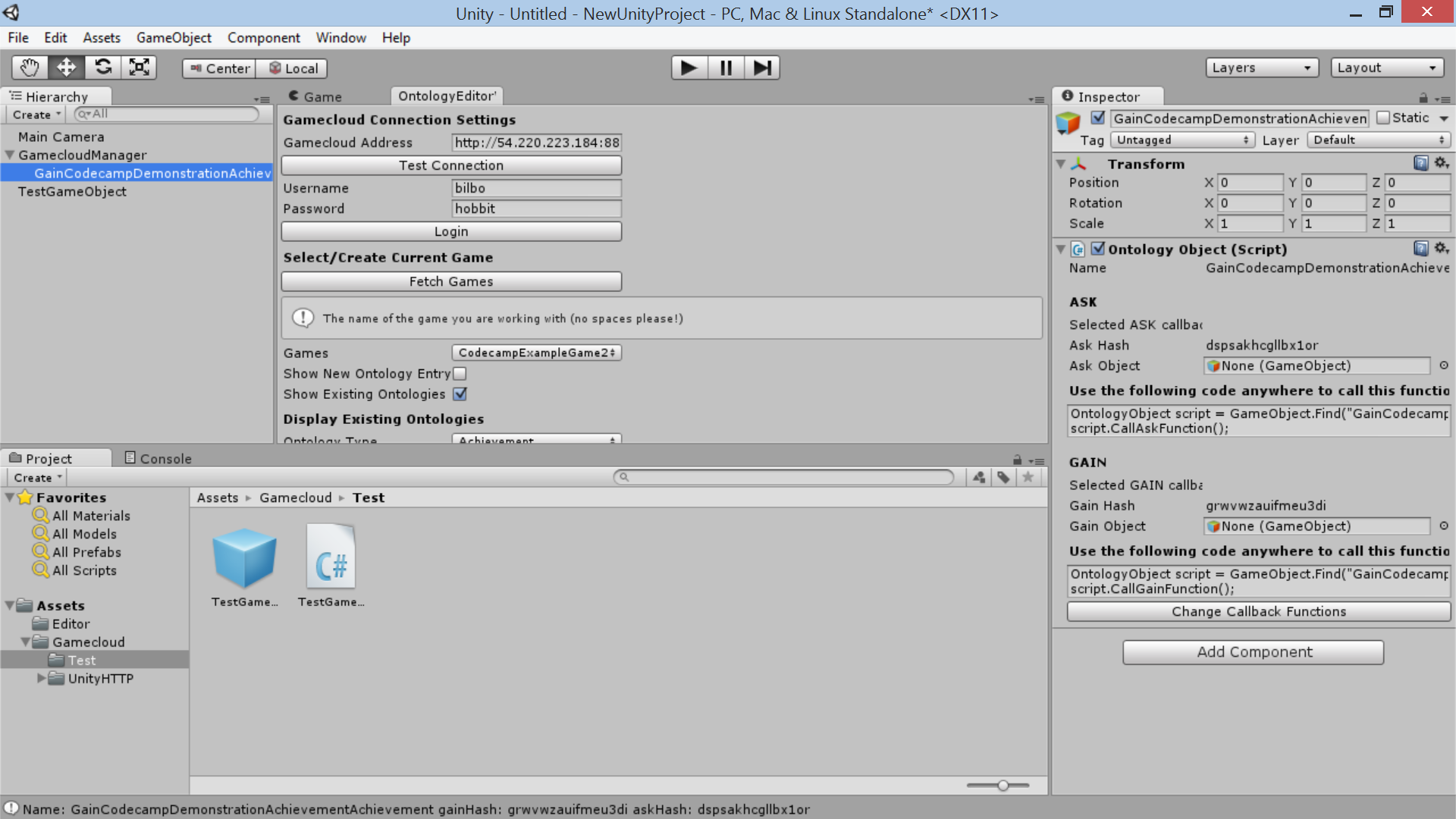


* Once you click “Generate in Project”, it will create a Unity GameObject that contains all the information of the Entry and how to call it.



# Calling the Functions in Unity3D

The created new GameObject contains information on how to do ASK/GAIN/LOSE for the given entry. Not all of these options are always available. For example, Achievements can never be lost, so there is no LOSE call for achievements.

The Inspector view in Unity shows all the required information and enable you to manipulate the object.

The ASK & GAIN hash have been automatically populated in this example.

Whenever you want to call a certain method, for example ASK if the current player has this achievement, you can copy-paste the shown code into your own program code.

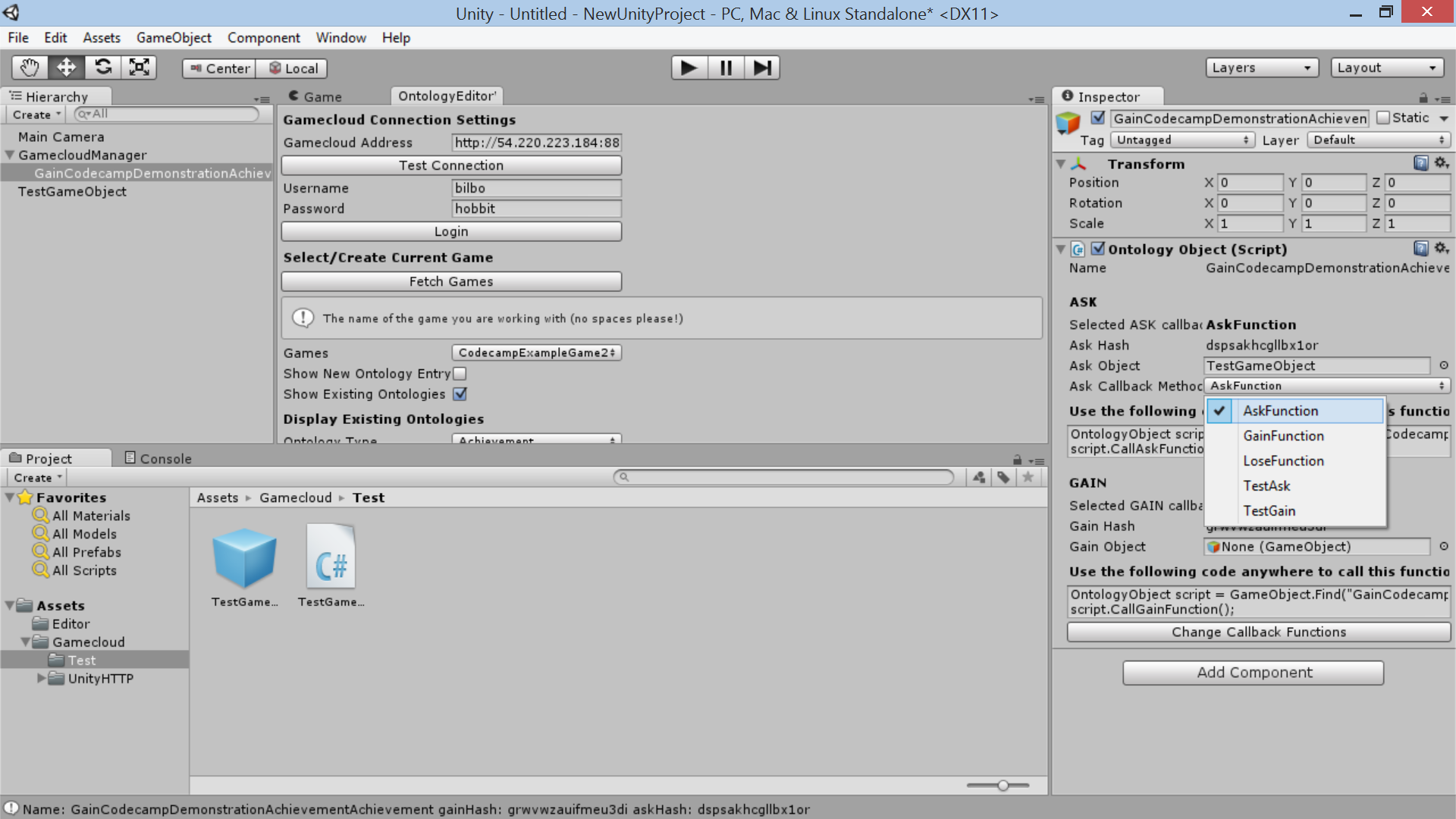
## Callbacks when receiving results from the Gamecloud

Notice there is place for GameObject at the both ASK & GAIN.

This is the place to drop ANY gameobject in Unity. When droped here, a new dropdown menu will open, displaying all public functions that are available in the scripts of the dropped GameObject.

The selected function will then be called, whenever the ASK method is called and a result is returned from Gamecloud.

AN EXAMPLE VIEW OF SELECTING FUNCTION



In the example, the message returned by Gamecloud is then processed by TestGameObject.AskFunction().

The limitation here is, that the selected function must be of format void MyFunction(HashTable result). **So it must accept HashTable as the result.**