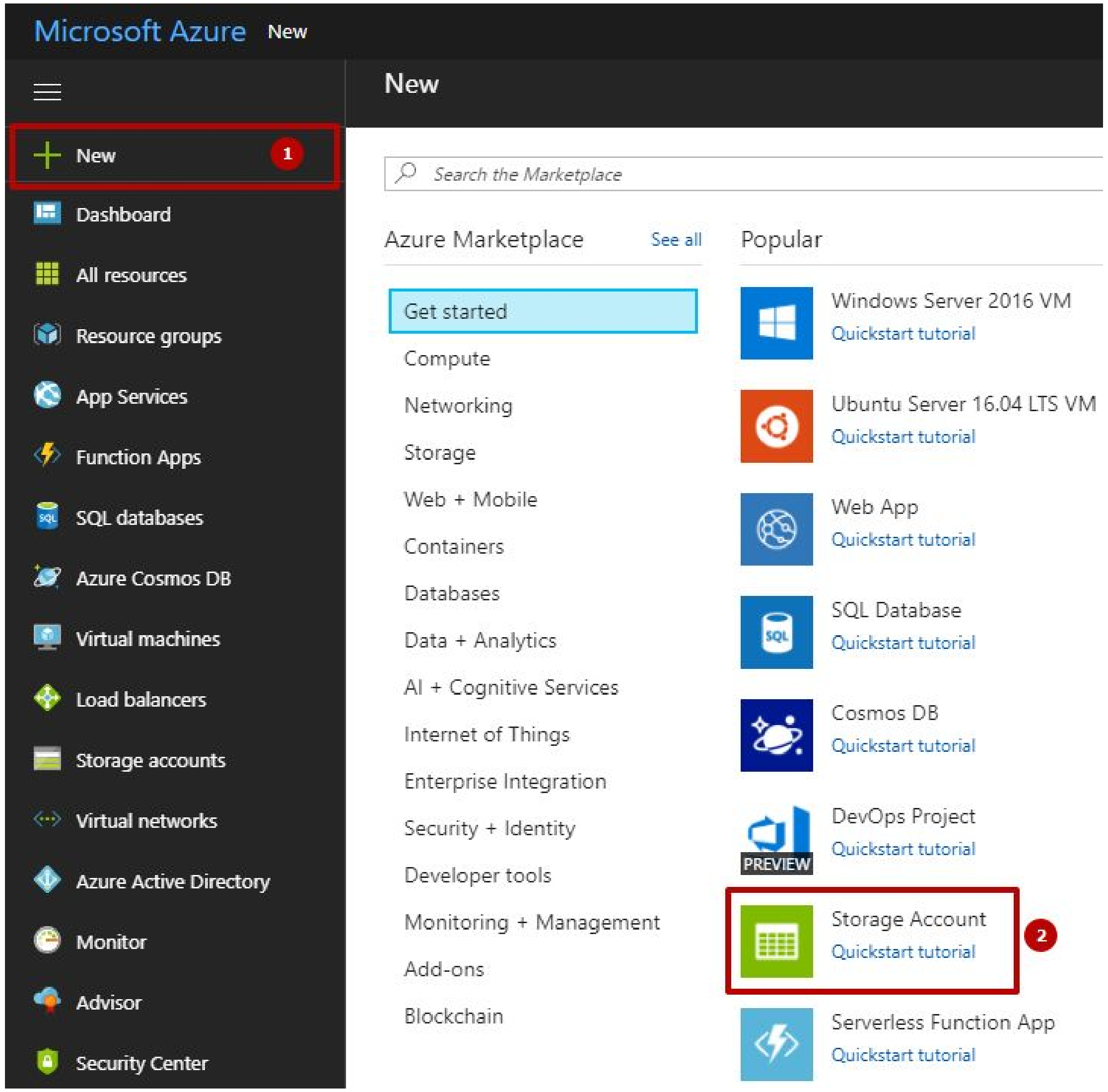
Dynamic loading of game content in Unity from the cloud.

Azure



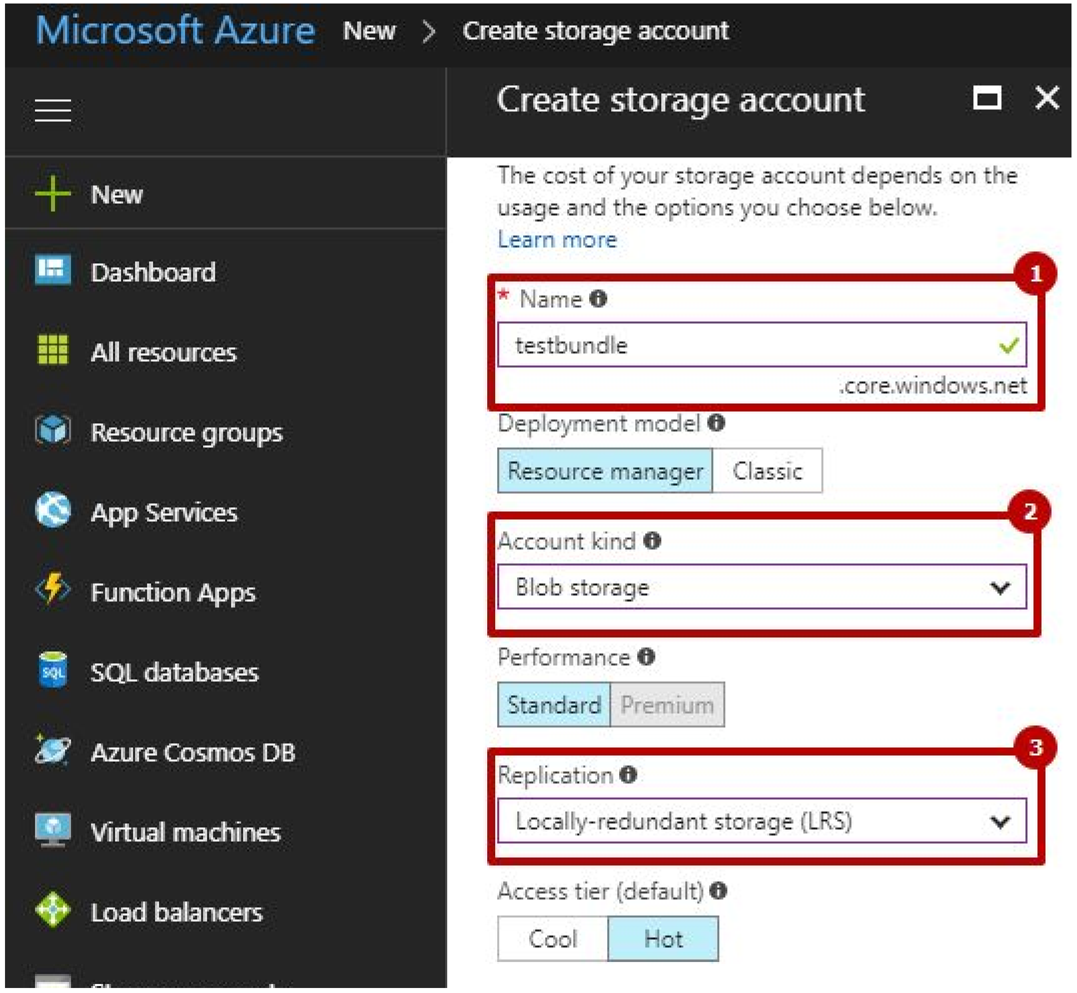
Creation and configuration of the required services.

First, you need to log in to an existing Azur account. If you do not have an account, ple follow this link https://www.microsoftazurepass.com/ ar activate Azure Pass balance for $ 100 usin pass code provided you by menthors to use Azure within a month.

The first thing we need to do is create a Storage Account. Windows Azure Storage i a cloud storage system that allows custom to store virtually unlimited amounts of dat any period of time. We will use it to store game asset bundles for our game.

Click New in the left panel and then sel Storage Account.

Dynamic loading of game content in Unity from the cloud.



The cost of your storage account depends on the usage and the options you choose below.
Learn more

* Name: testbundle .core.windows.net

Deployment model: [✓ Resource manager] Classic

Account kind: Blob storage

Performance: [✓ Standard] Premium

Replication: Locally-redundant storage (LRS)

Access tier (default): Cool [✓ Hot]

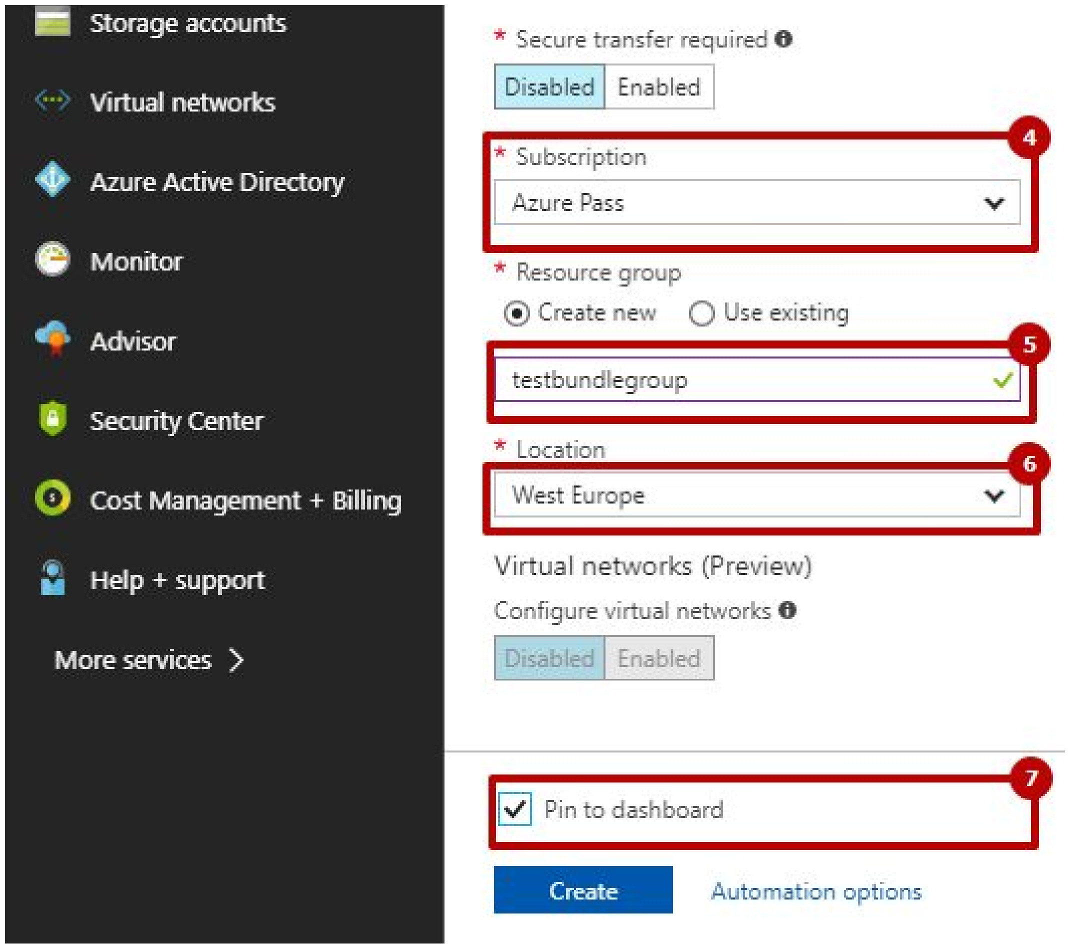
To create a storage, you need to fill in sever important fields.

1 - Name the name of our repository.

2 - Account kind choose Blob storage. Blob stands for Binary Large OBject.

3 - Replication Locally-redundant storage

Dynamic loading of game content in Unity from the cloud.



* Secure transfer required: [✓ Disabled] Enabled

* Subscription: Azure Pass

* Resource group: [✓ Create new] Use existing

testbundlegroup

* Location: West Europe

Virtual networks (Preview)

Configure virtual networks: [✓ Disabled] Enabled

[✓] Pin to dashboard

4 - Subscription we choose the type of current subscription. If you activated Azure Pass then the system will offer you to choos this type of subscription.

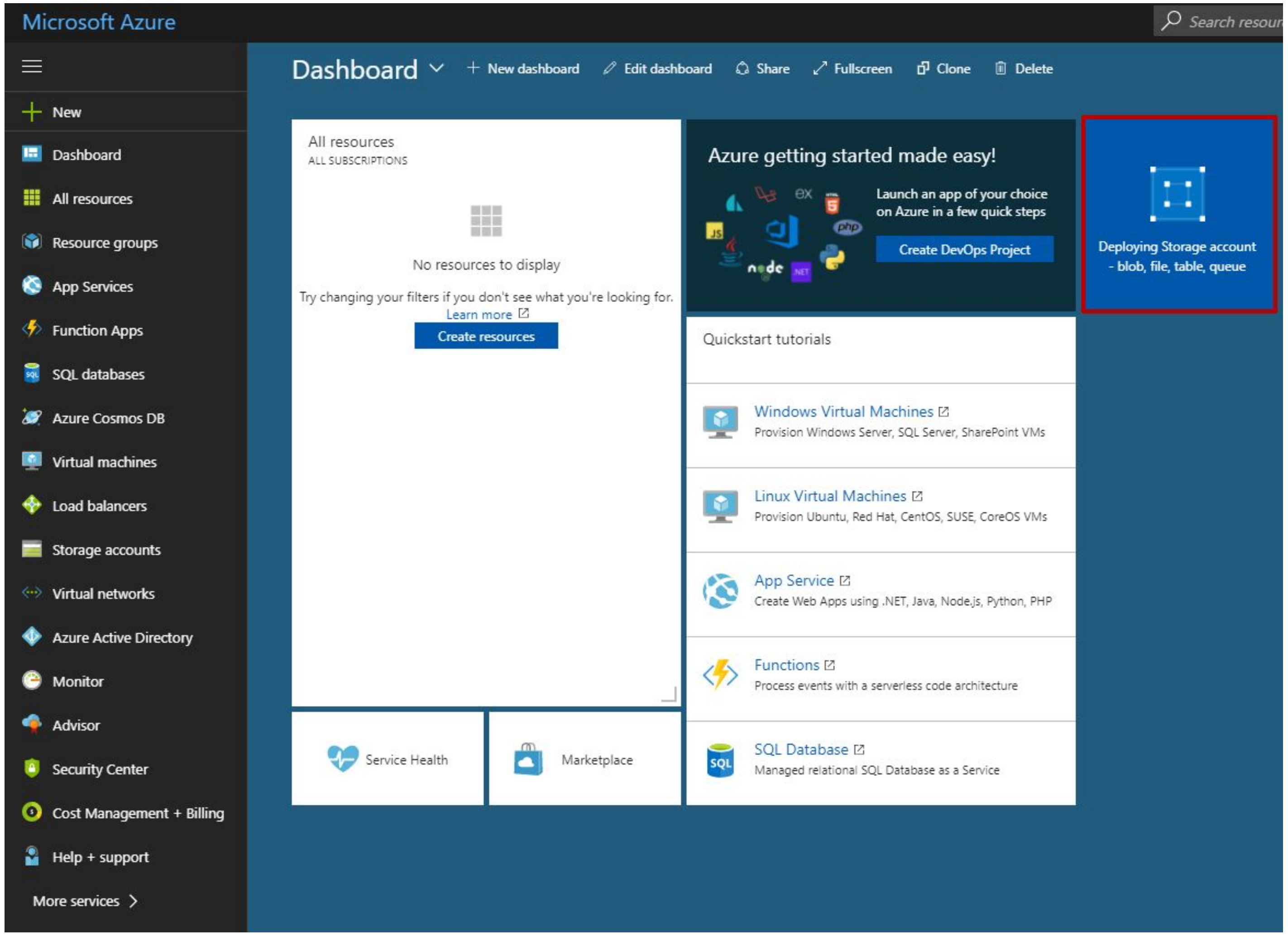
5 - Resource Group - you should name the current resource group. Resource groups, within which you can create the required se applications, services and databases.

6 - Location specify the location of the serv that are closer to you.

7 - Pin to dashboard to place a link to our storages on the main page.

Click the Create button to create.

Dynamic loading of game content in Unity from the cloud.



No resources to display

Try changing your filters if you don't see what you're looking for.

Launch an app of your choice on Azure in a few quick steps

Provision Windows Server, SQL Server, SharePoint VMs

Provision Ubuntu, Red Hat, CentOS, SUSE, CoreOS VMs

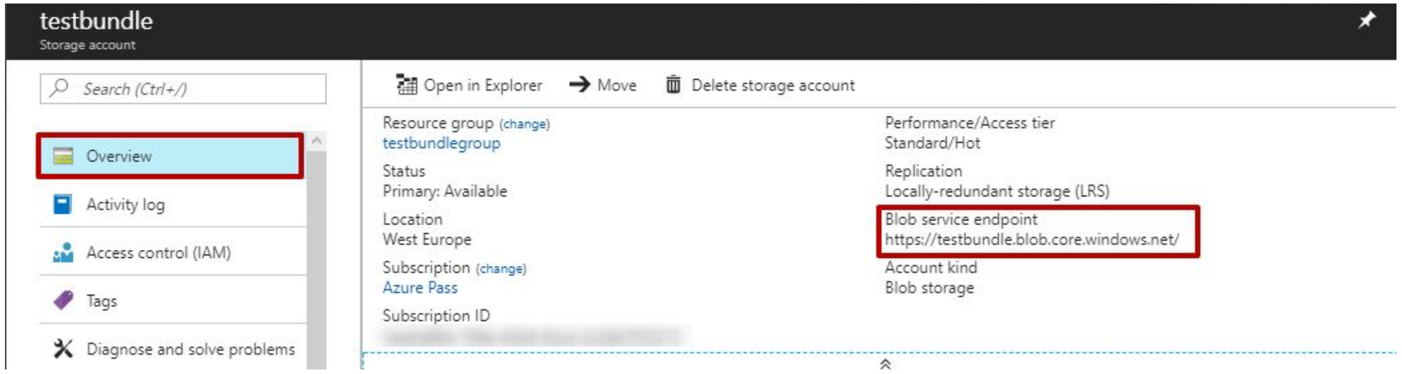
Create Web Apps using .NET, Java, Node.js, Python, PHP

Process events with a serverless code architecture

Managed relational SQL Database as a Service

After we click on the Create button, a link t our storage should appear on the main pag The screenshot shows the process of deploy a storage on the Cloud.

Dynamic loading of game content in Unity from the cloud.


testbundle
Storage account

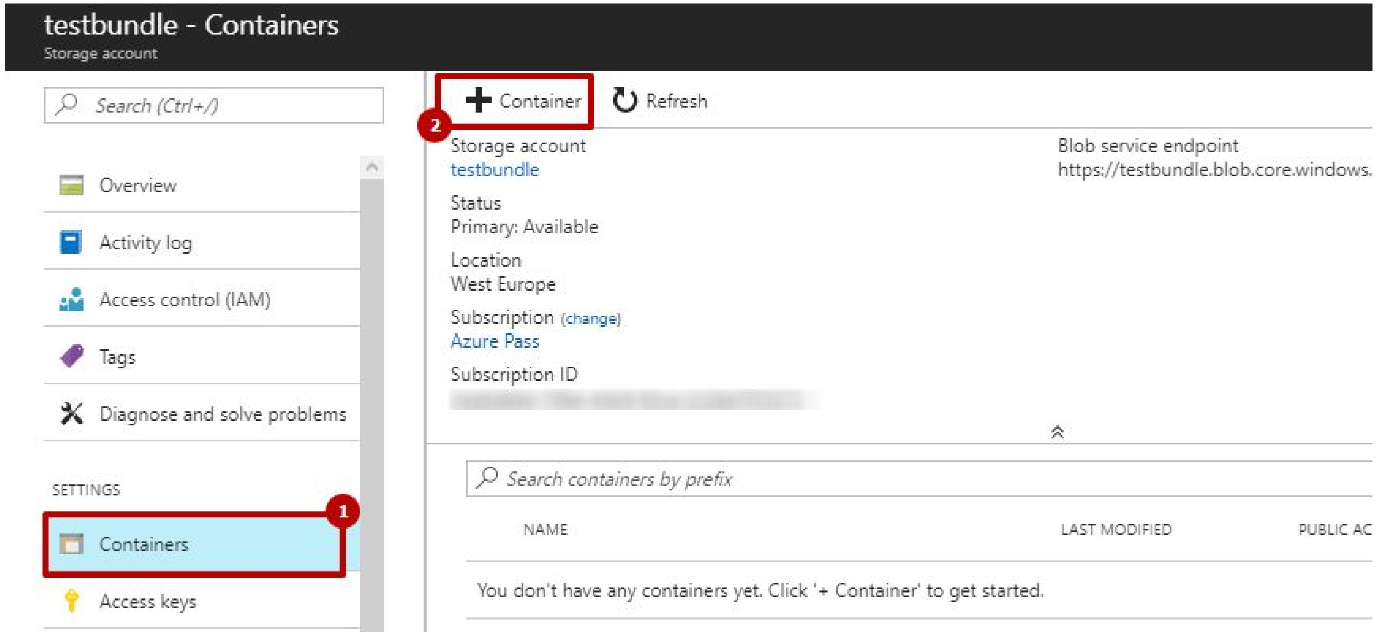
Open in Explorer | Move | Delete storage account

Resource group (change)
testbundlegroup
Status
Primary: Available
Location
West Europe
Subscription (change)
Azure Pass
Subscription ID
[redacted]

Performance/Access tier
Standard/Hot
Replication
Locally-redundant storage (LRS)
Blob service endpoint
https://testbundle.blob.core.windows.net/
Account kind
Blob storage

We will use in our project Blob service endpoint address, you can view it in Overview tab.

In our project this is: testbundle.blob.core.windows.net


testbundle - Containers
Storage account

Container | Refresh

Storage account
testbundle
Status
Primary: Available
Location
West Europe
Subscription (change)
Azure Pass
Subscription ID
[redacted]

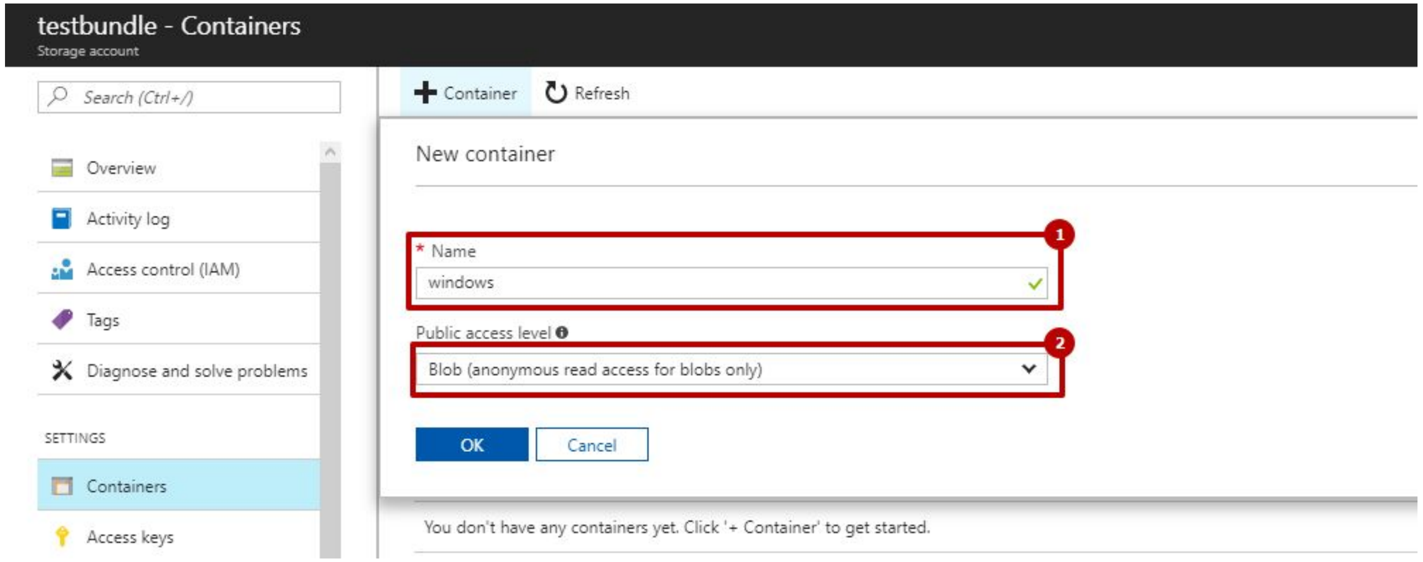
Blob service endpoint
https://testbundle.blob.core.windows.

NAME	LAST MODIFIED	PUBLIC AC
You don't have any containers yet. Click '+ Container' to get started.		

Now it's time to create a container for stori our gaming bundles. Bandles for each platf are different and in Unity it is different files which have the same name only.

Dynamic loading of game content in Unity from the cloud.

Azure



testbundle - Containers
Storage account

New container

1

* Name: windows

2

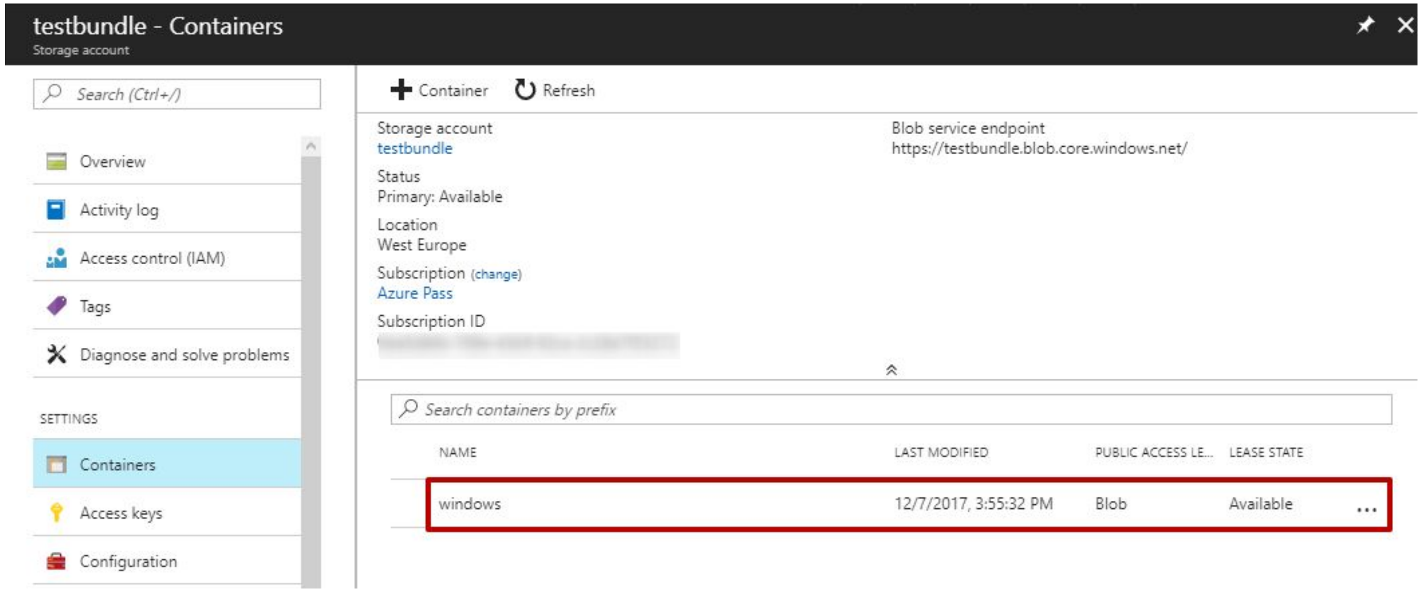
Public access level: Blob (anonymous read access for blobs only)

You don't have any containers yet. Click '+ Container' to get started.

1 - Name - In this example, we named our windows container, since it will store the game's bandles for the Windows platform. T name does not matter and is chosen for convenience.

2 - Public access level. You can also configure the level of access to our reposito

Then click the Ok button



testbundle - Containers
Storage account

Storage account: testbundle

Status: Primary: Available

Location: West Europe

Subscription (change): Azure Pass

Subscription ID: [redacted]

Blob service endpoint: https://testbundle.blob.core.windows.net/

Search containers by prefix

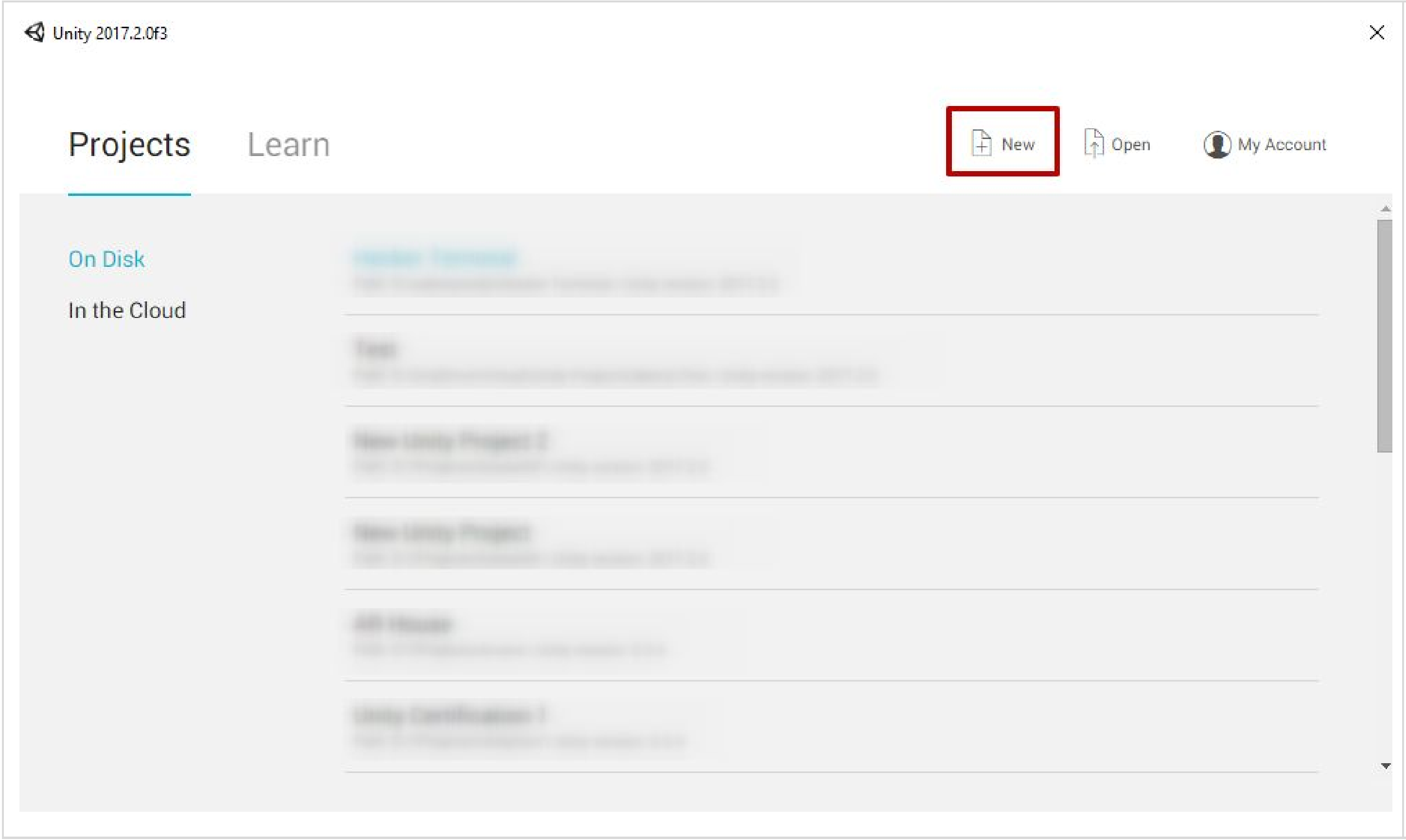
NAME	LAST MODIFIED	PUBLIC ACCESS LE...	LEASE STATE
windows	12/7/2017, 3:55:32 PM	Blob	Available

After a successful creation, you should have new container.

Successfully creating a container, we need t create a Bundle in Unity, which we will stor our container.

Please do not close the Azure window, we w return to it right after we go through the necessary steps in Unity.

Dynamic loading of game content in Unity from the cloud.

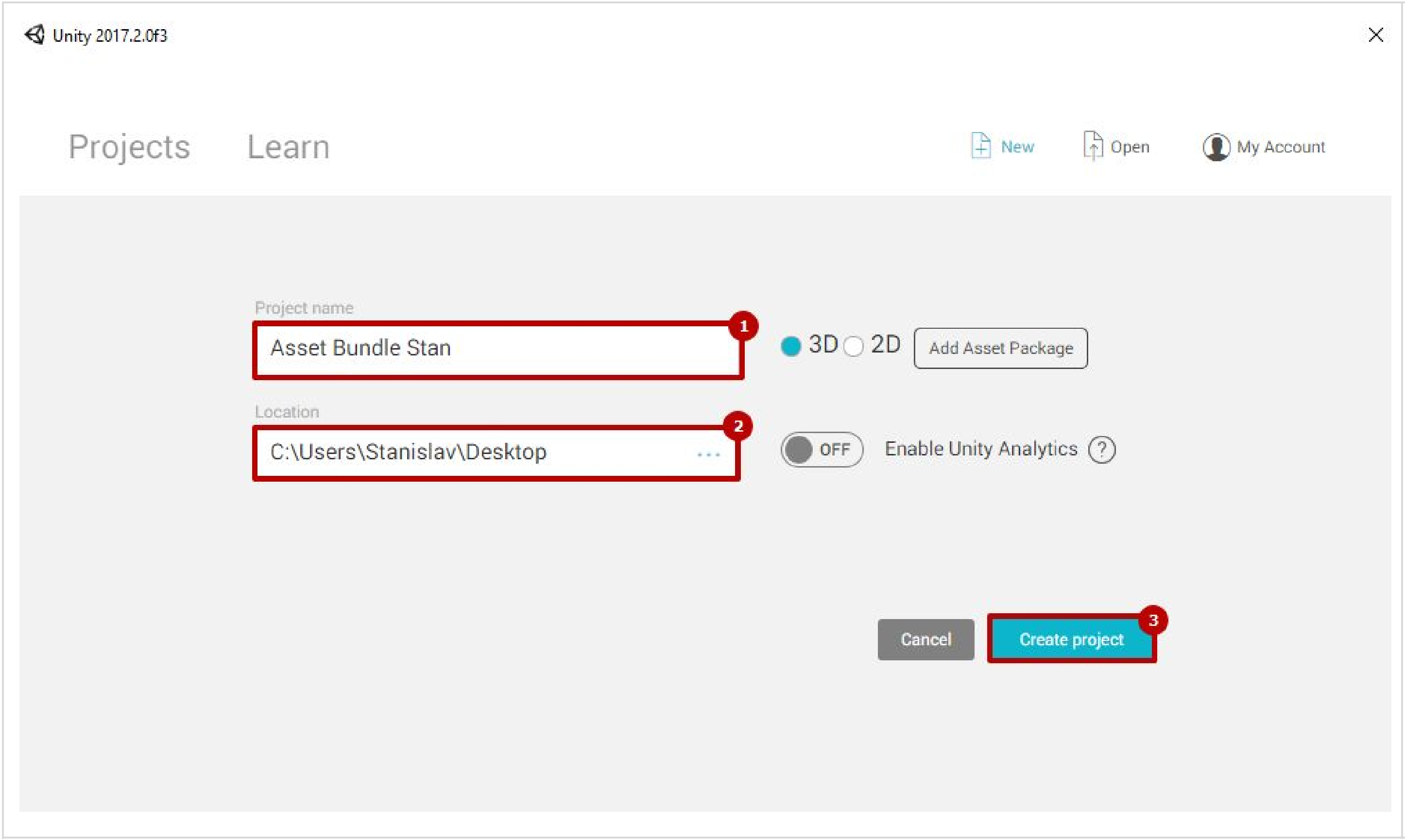


Connecting Services to a New Unity Project

In this project we used version Unity 2017.

First, run Unity and create a new project.

Dynamic loading of game content in Unity from the cloud.



Project name: Asset Bundle Stan

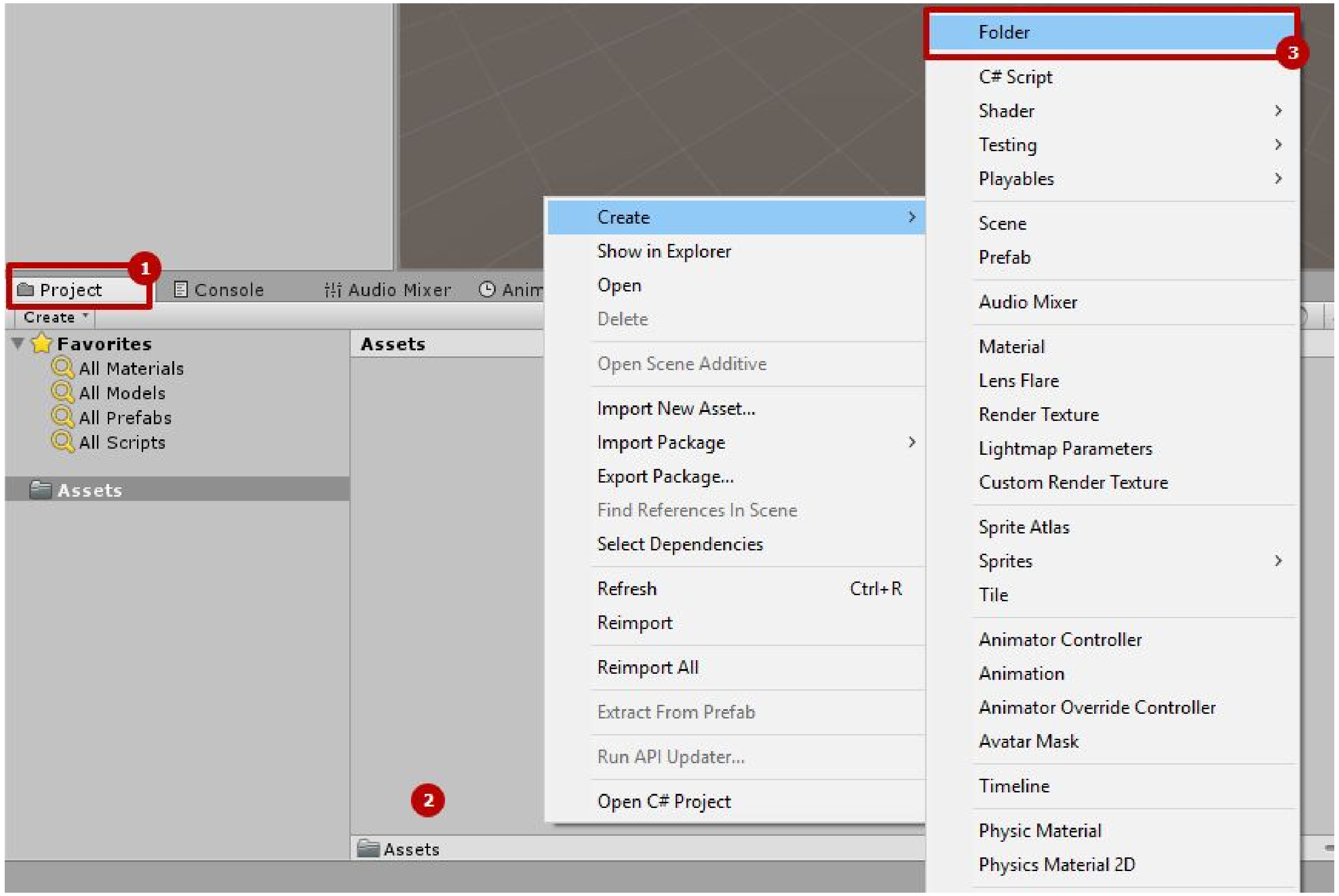
Location: C:\Users\Stanislav\Desktop

1 - Project name - enter the unique name o your new project.

2 - Location - specify the directory where th new project will be stored.

3 - Create project Click to start creating a new project.

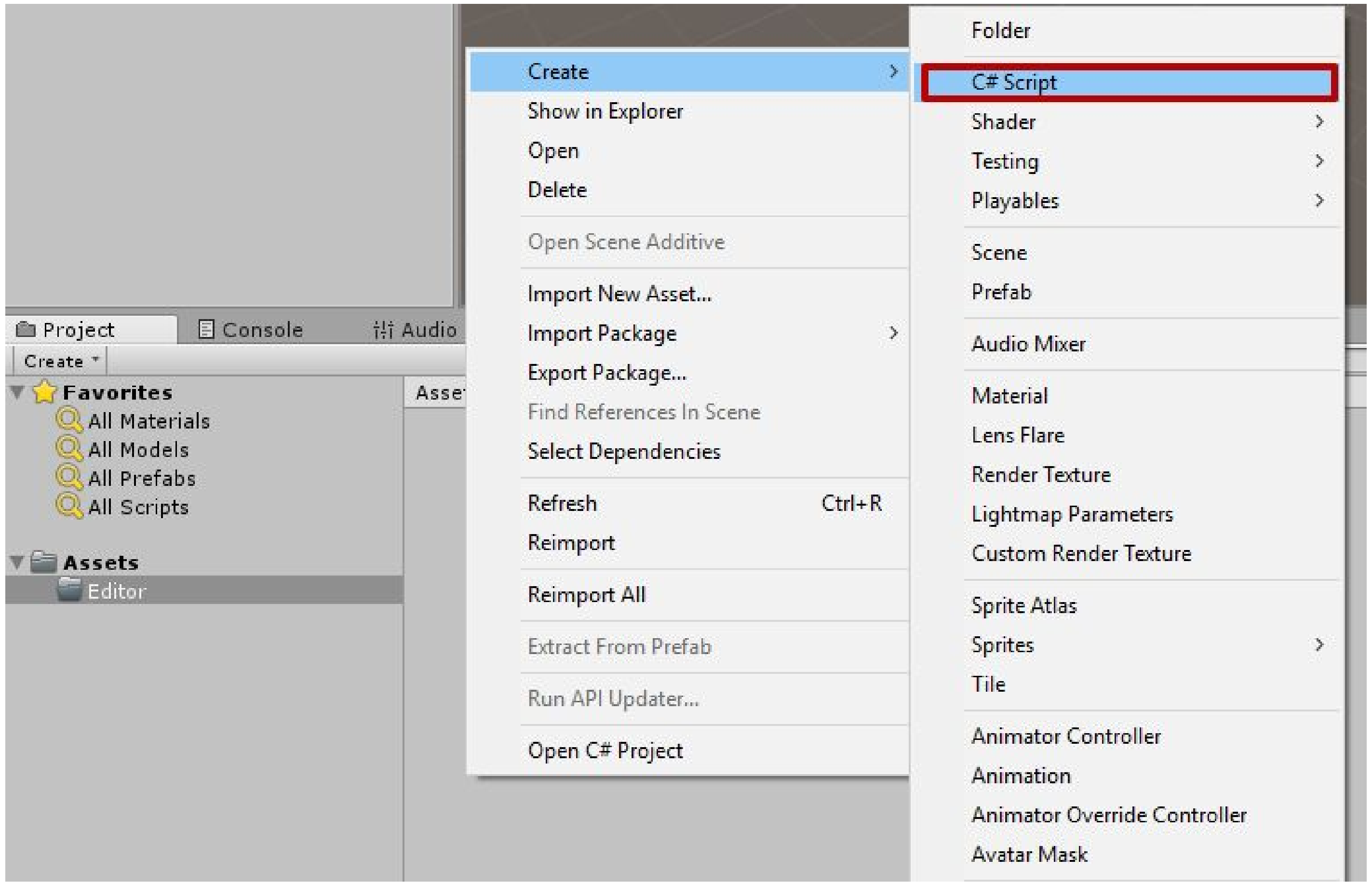
Dynamic loading of game content in Unity from the cloud.



In the Project window, create the Editor folder:
1 - go to the Project window
2 - right-click on the empty area and choos
the Create menu
3 - Folder. Create a new folder named "Editor" and go to our new folder.

Dynamic loading of game content in Unity from the cloud.

Azure

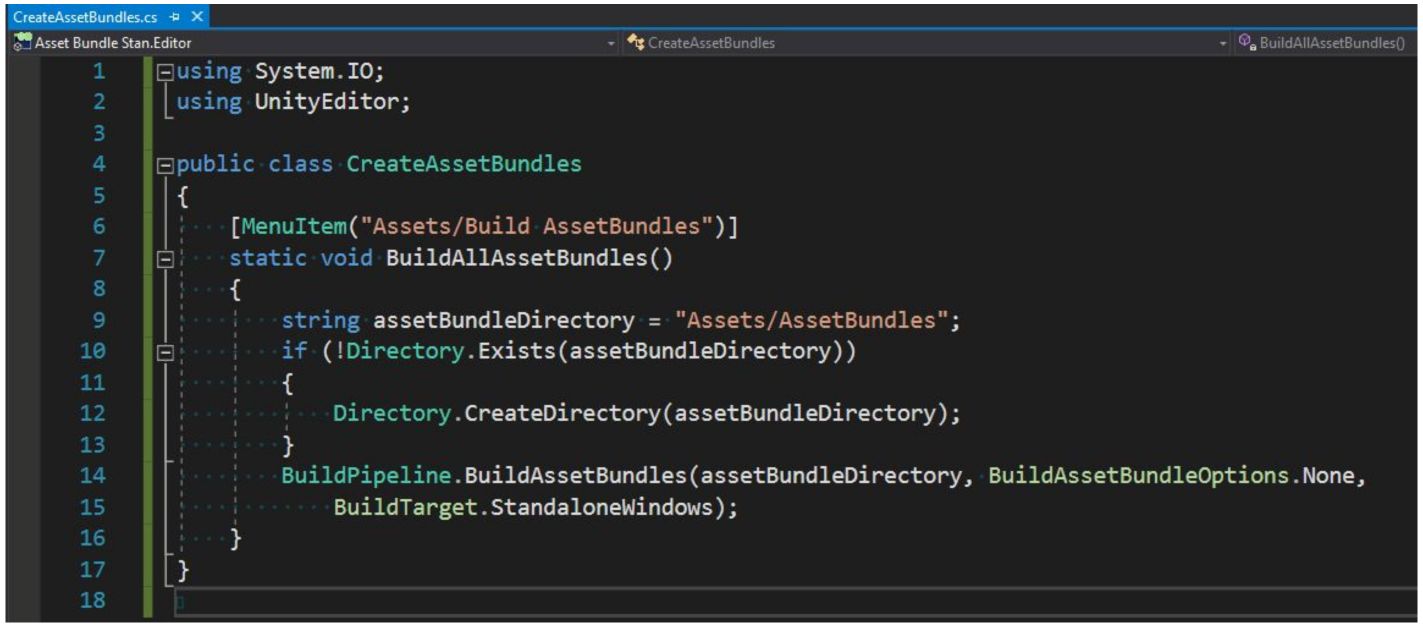


In our folder, right-click, and select "C# Script" and create a new C# script called CreateAssetBundles.

With this script, we can create Asset Bundle the editor.

Open the script editor by double clicking on file of the created script.

Dynamic loading of game content in Unity from the cloud.

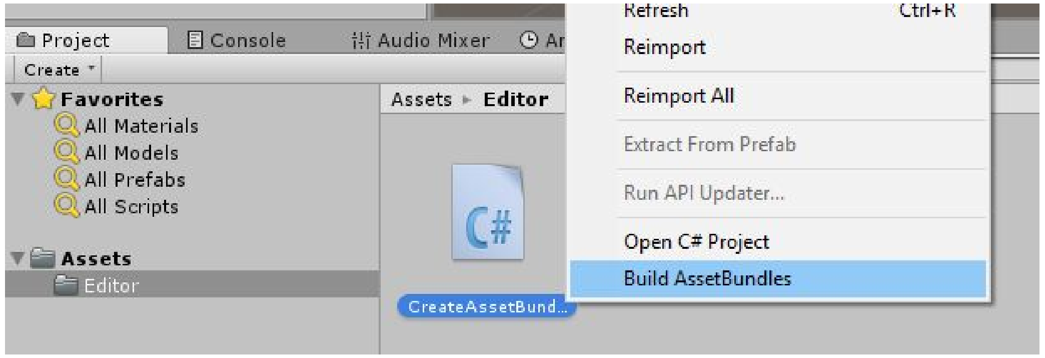


```
using System.IO;
using UnityEditor;

public class CreateAssetBundles
{
    [MenuItem("Assets/Build AssetBundles")]
    static void BuildAllAssetBundles()
    {
        string assetBundleDirectory = "Assets/AssetBundles";
        if (!Directory.Exists(assetBundleDirectory))
        {
            Directory.CreateDirectory(assetBundleDirectory);
        }
        BuildPipeline.BuildAssetBundles(assetBundleDirectory, BuildAssetBundleOptions.None,
            BuildTarget.StandaloneWindows);
    }
}
```

Note: in the method:
BuildPipeline.BuildAssetBundles (assetBundleDirectory, BuildAssetBundleOptions.None, BuildTarget.StandaloneWindo
the third parameter specifies the platform for which the bundle is created. In our case it is Windo

Replace the default code with the code belo
and press CTRL + S to save:

```
using System.IO;
using UnityEditor;

public class CreateAssetBundles
{
   [MenuItem("Assets/Build AssetBundles")]
   static void BuildAllAssetBundles()
   {
      string assetBundleDirectory = "Assets/AssetBundles";
      if (!Directory.Exists(assetBundleDirectory))
      {
         Directory.CreateDirectory(assetBundleDirectory);
      }
      BuildPipeline.BuildAssetBundles(assetBundleDirectory,
BuildAssetBundleOptions.None,
         BuildTarget.StandaloneWindows);
   }
}
```

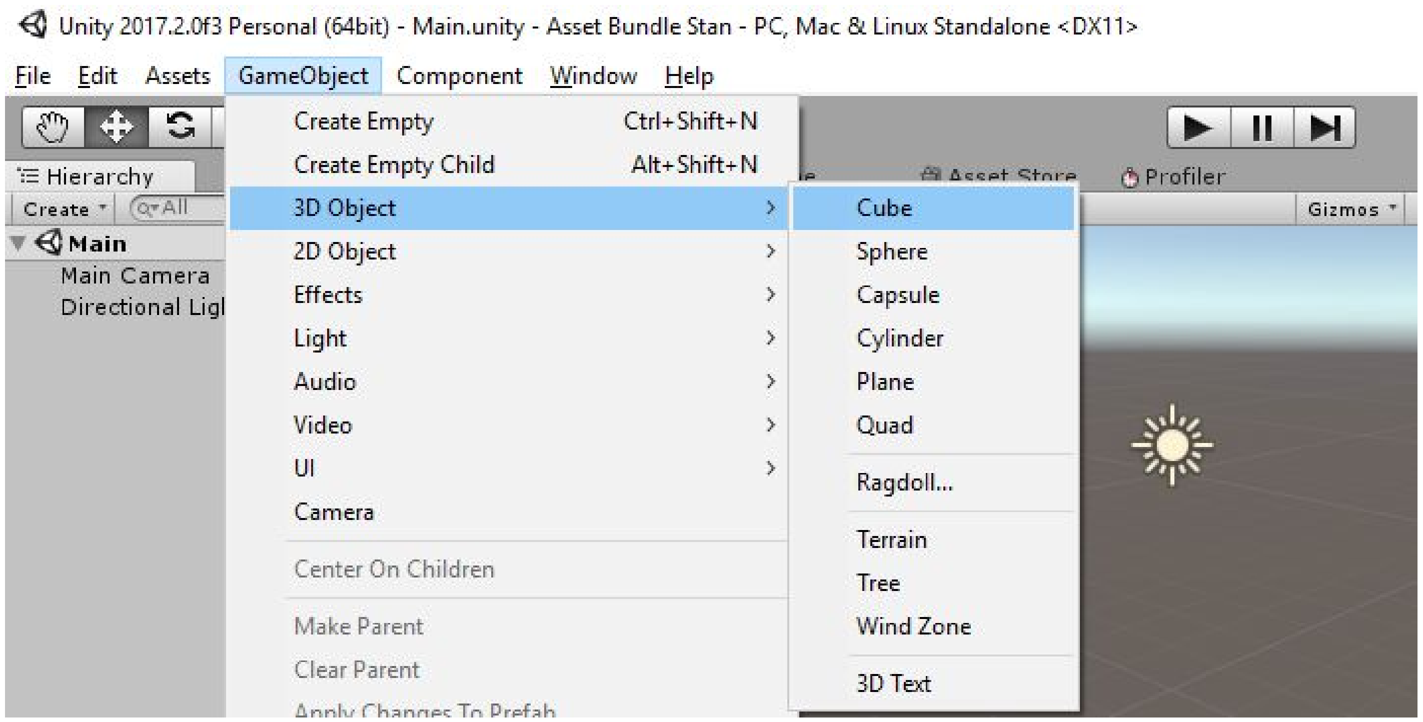


Just in case, we put a link to the documentation for the current version of the Unity:
https://docs.unity3d.com/Manual/AssetBundles-Workflow.html

You may notice that in the editor's menu in Assets there will be a button Build AssetBundles. When you click this button, t BuildAllAssetBundles method will start working. In the body of this method, the AssetBundles folder is created and where tl bundles are placed. We will upload this bun later in to the Azure cloud.

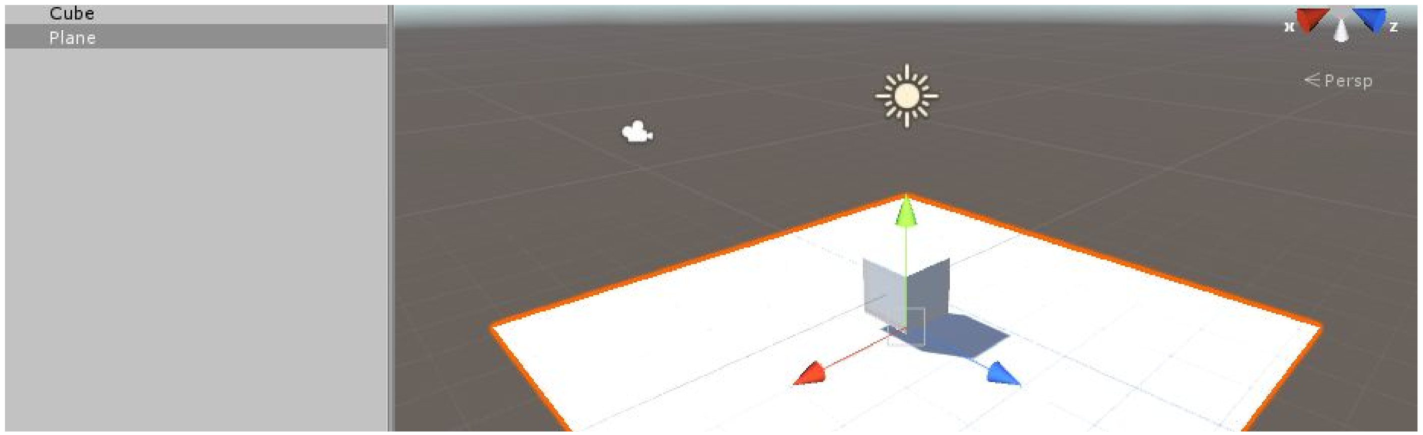
Dynamic loading of game content in Unity from the cloud.

Azure



Now let's start creating our gaming bundles
Our first bandle will be a prefab made out o
cube.
We save and call our scene of the project.
(CTRL + SHIFT + S) In our case, we called i
Main.

Add to the scene a cube (from which we wil
then make a bundle) for this in the upper m
bar, click GameObject -> 3D Object ->
Cube

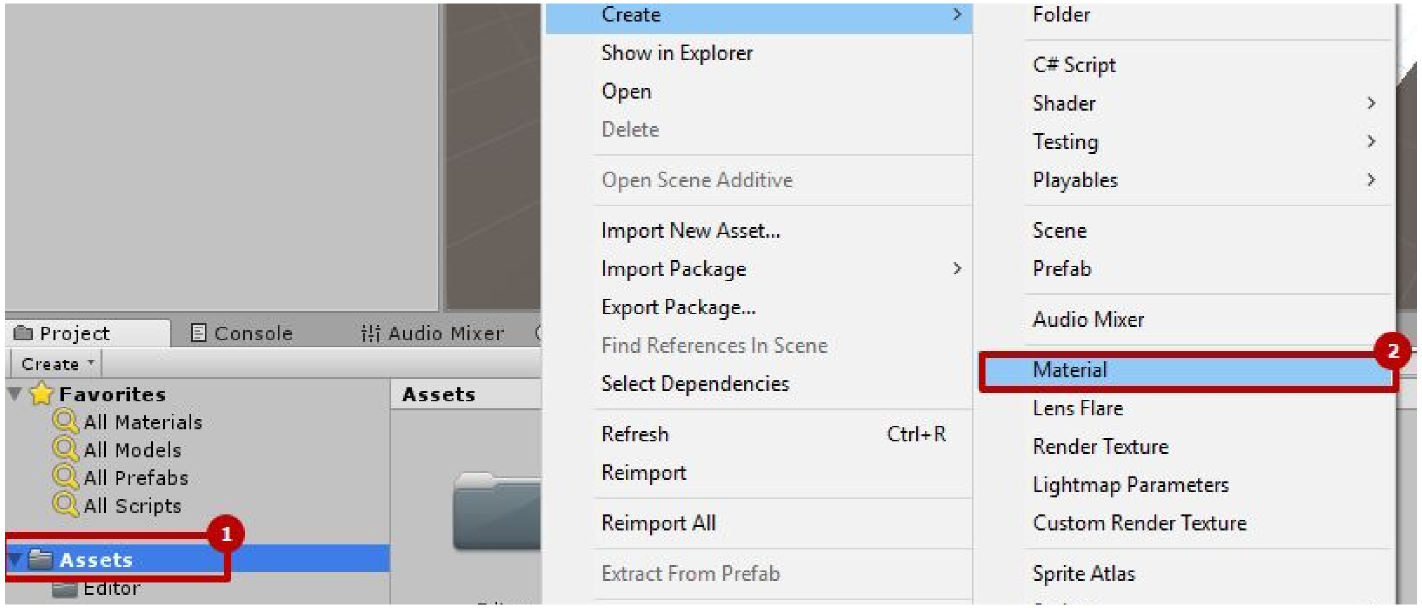


Add to the stage a plane (This is our map, it
be static) GameObject -> 3D Object ->
Plane

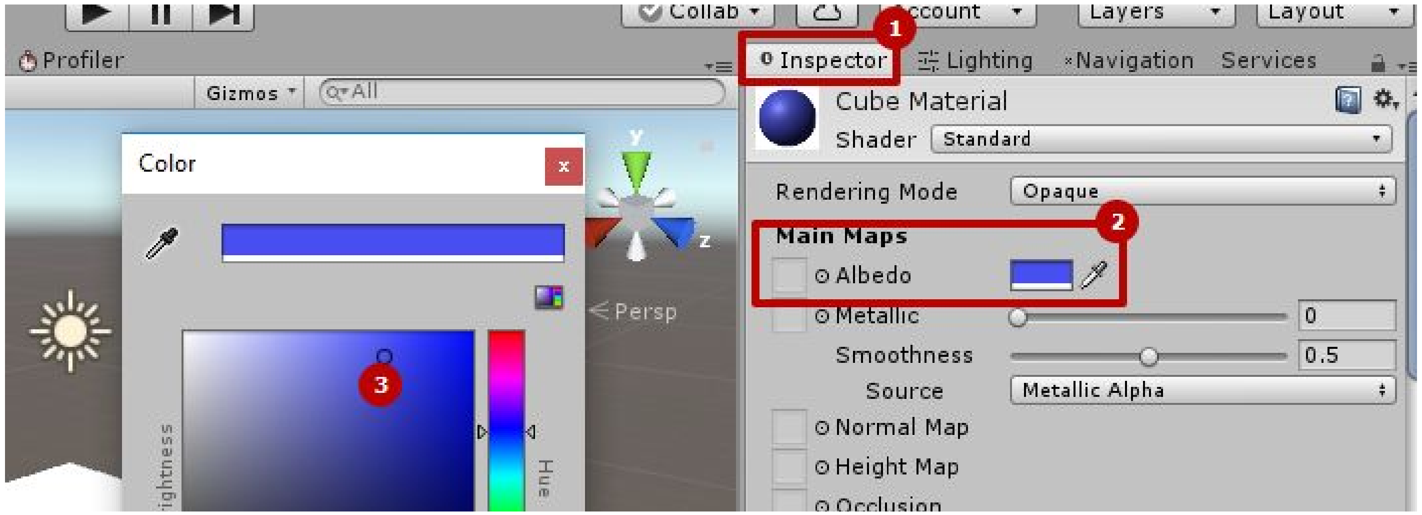
In order to visually divide the cube and map
pull the green arrow, which is responsible f
the vertical positioning of objects.

Dynamic loading of game content in Unity from the cloud.

Azure

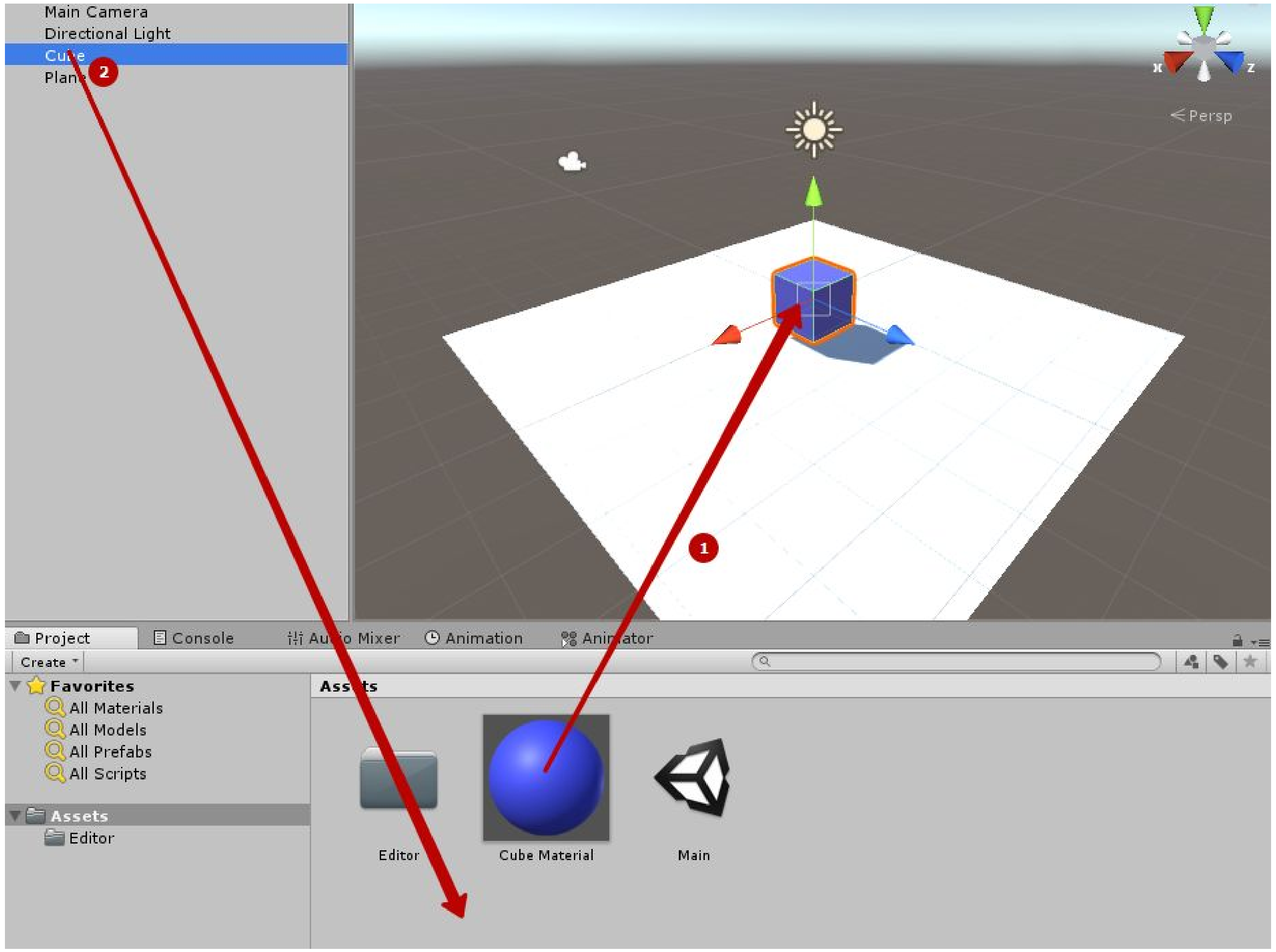


In order for our cube to be special, we will create material of any color for it. To do this to the Assets folder and right click to create Material named "Cube Material". When you click on a newly created material you can change its color.



To do this, in the upper right corner of the Inspector window, click on the window nex the name Albedo - it will help to change the color. And also choose the color that you lik most in the palette.

Dynamic loading of game content in Unity from the cloud.

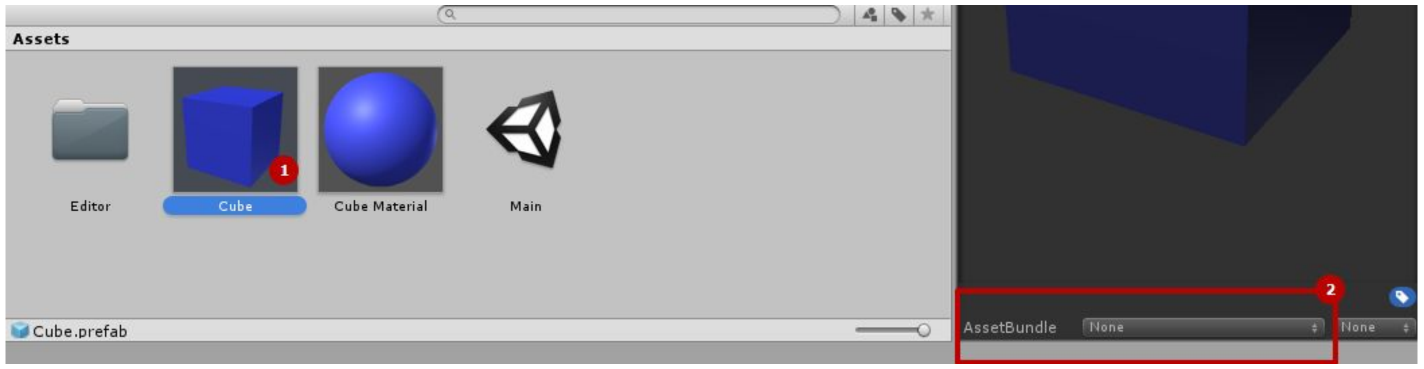


1 - In order to apply the material to our cub
you just need to drag the material onto the
cube itself.

2 - We make a prefab from the cube (Just
dragging it from the Hierarchy window to tl
empty place of the Project window) Prefab
special type of asses that allows storing the
entire GameObject with all components anc
property values. Prefab acts as a template f
creating instances of a stored object in the
scene.

Saving our scene (CTRL + SHIFT + S)

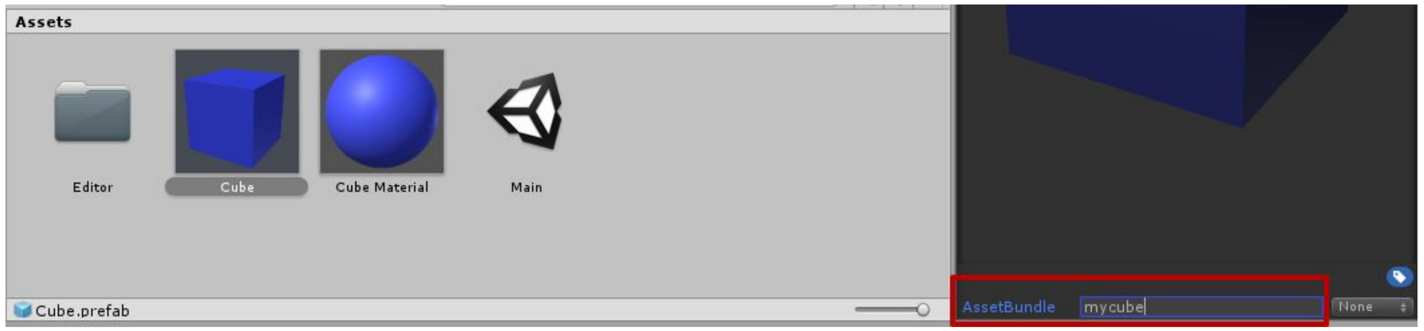
Dynamic loading of game content in Unity from the cloud.



Cube.prefab

AssetBundle None None

Select the pre-created "Clube" and find the small AssetBundle menu, on the right of the screen.

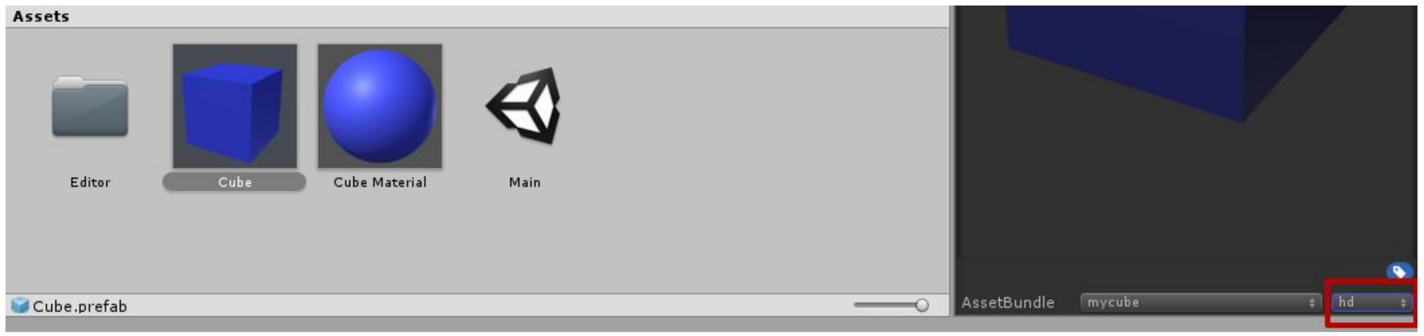


Cube.prefab

AssetBundle mycube None

By clicking New option creates a new name our Asset Bundle, in my case I called "mycube".

Using this name we will further download t file from the Azure cloud.

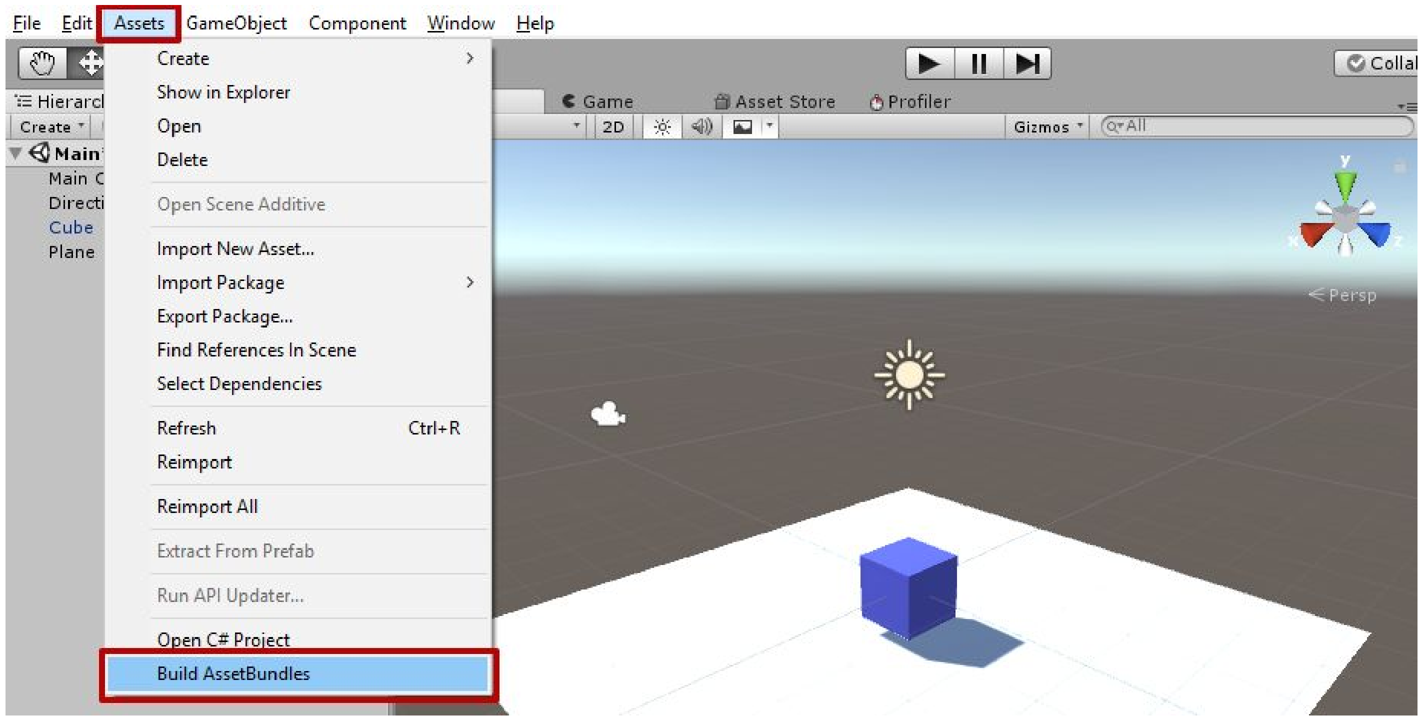


Cube.prefab

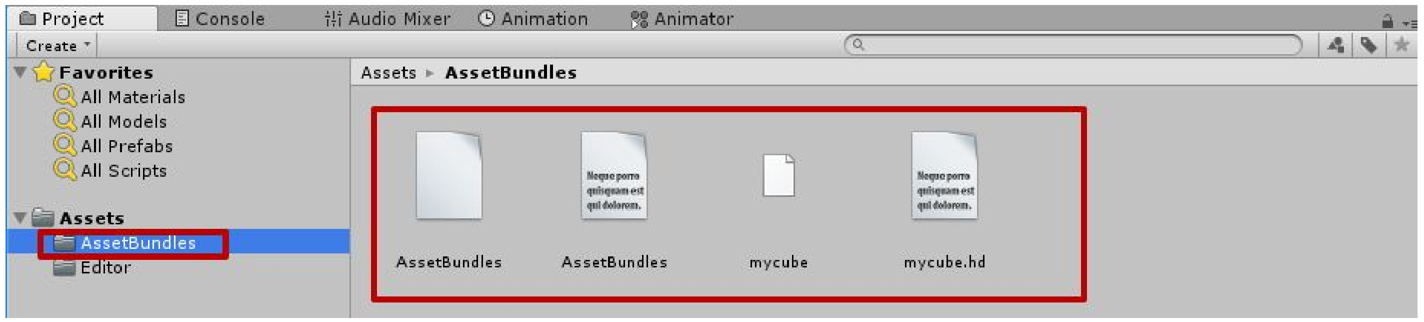
AssetBundle mycube hd

Please view the field next to the name. By default, it stores the value None.
In this field, an index is created, as a result, is appended to the name of the bundle. The index can be left blank, but for example I marked it as "hd".

Dynamic loading of game content in Unity from the cloud.



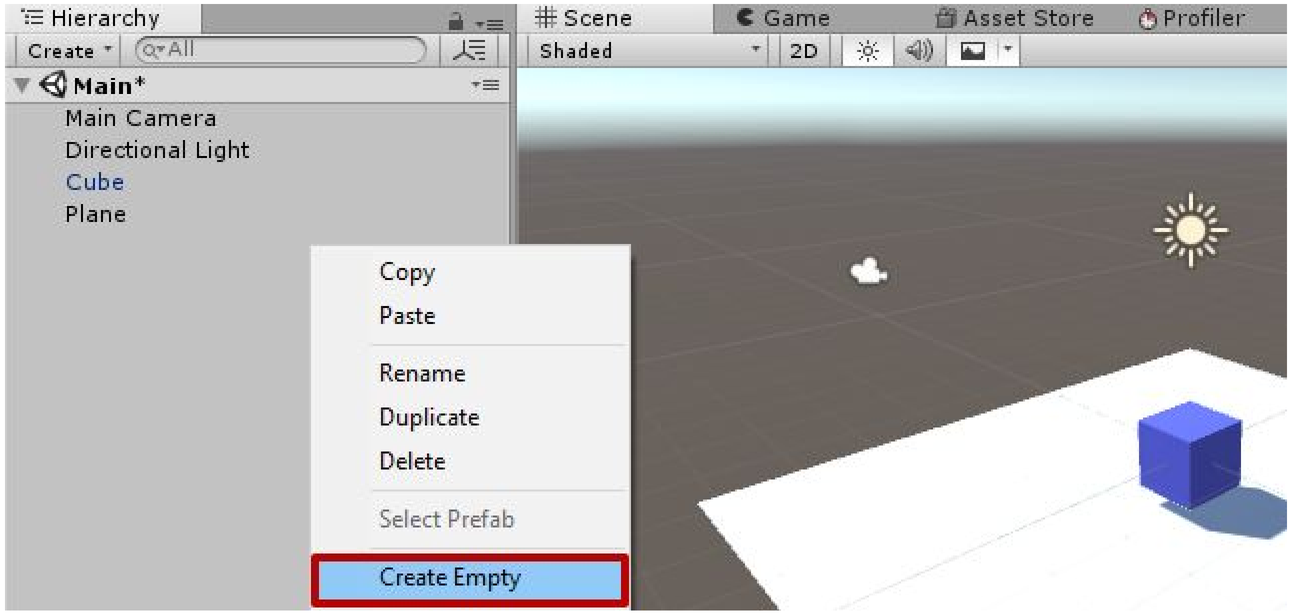
After we gave the name to the bandle, it's t
to use the BuildAllAssetBundles script, just
go to the Assets menu in the top menu bar
and click the Build AssetBundles button.



After the creation process is finished we wil
a new folder AssetBundles with our bundles
We will upload them to the cloud Azure.

Dynamic loading of game content in Unity from the cloud.

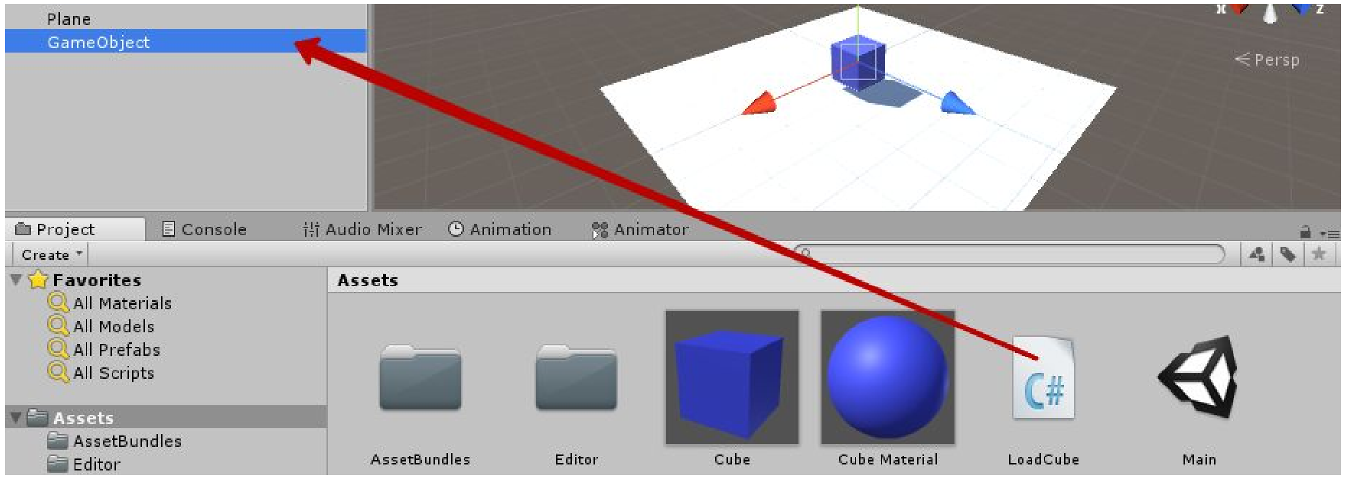
Azure



In each game during the development proc
the bundles can change a lot, only the clear
variants are loaded onto the cloud. This is d
in order to save time.

We need to prepare a script that will be
responsible for downloading and creating tl
object.

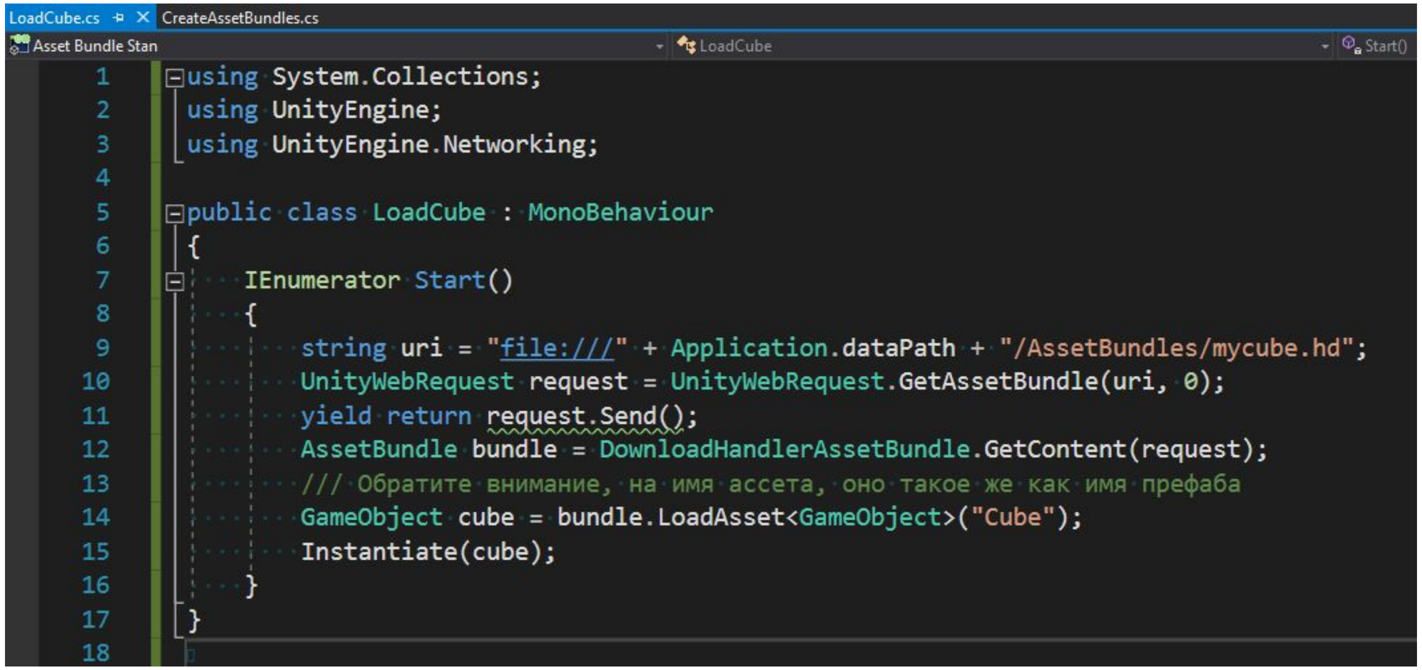
First, create an empty "GameObject" object
To do this, right-click in the empty area of t
Hierarchy window and select Create Empty



Create a script called "LoadCube".

The script is placed on the newly created
GameObject

Dynamic loading of game content in Unity from the cloud.

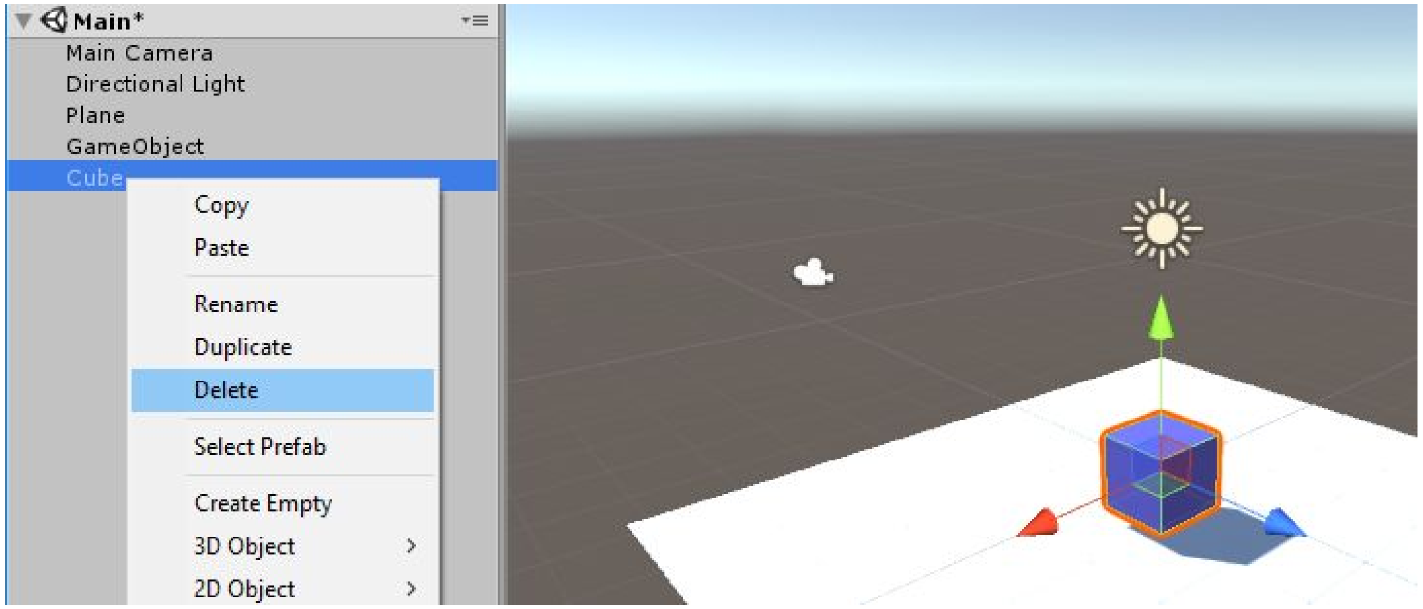


```
using System.Collections;
using UnityEngine;
using UnityEngine.Networking;

public class LoadCube : MonoBehaviour
{
    IEnumerator Start()
    {
        string uri = "file:///" + Application.dataPath + "/AssetBundles/mycube.hd";
        UnityWebRequest request = UnityWebRequest.GetAssetBundle(uri, 0);
        yield return request.Send();
        AssetBundle bundle = DownloadHandlerAssetBundle.GetContent(request);
        /// Обратите внимание, на имя ассета, оно такое же как имя префаба
        GameObject cube = bundle.LoadAsset<GameObject>("Cube");
        Instantiate(cube);
    }
}
```

Now let's start writing the logic of the LoadCube script. To do this, open the script and paste the code into it, which is below. Pay attention to the name of the downloada bundle and the path to it. Now it's called mycube.hd it's important that an index is added through the point to it. In the Unity editor, the index name is not visible, it is considered as an extension and is not displa in the editor.

```
using System.Collections;
using UnityEngine;
using UnityEngine.Networking;

public class LoadCube : MonoBehaviour
{
    IEnumerator Start()
    {
        string uri = "file:///" + Application.dataPath +
"/AssetBundles/mycube.hd";
        UnityWebRequest request =
UnityWebRequest.GetAssetBundle(uri, 0);
        yield return request.Send();
        AssetBundle bundle =
DownloadHandlerAssetBundle.GetContent(request);
        /// Обратите внимание, на имя ассета, оно такое же
имя префаба
        GameObject cube =
bundle.LoadAsset<GameObject>("Cube");
        Instantiate(cube);
    }
}
```

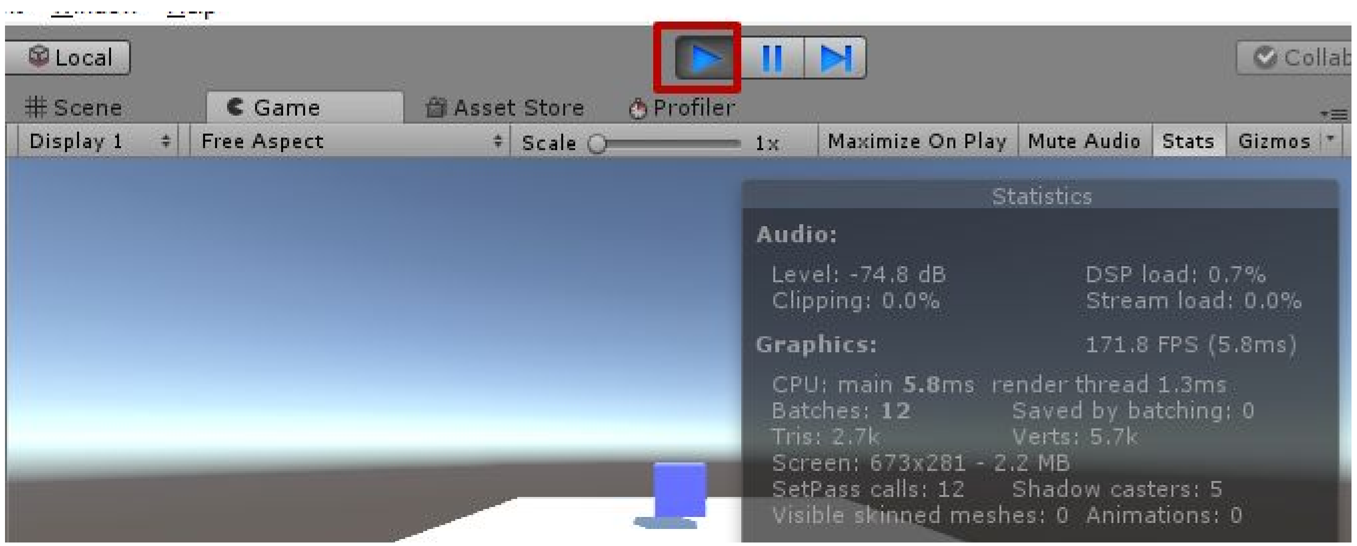
Dynamic loading of game content in Unity from the cloud.



Let's now remove our cube from the scene s that we can see how the cube created from Asset's bundles appears after the game is launched.

Select the Cube object in the Hierarchy and right-click Delete.

We can save our scene CTRL + Shift + S



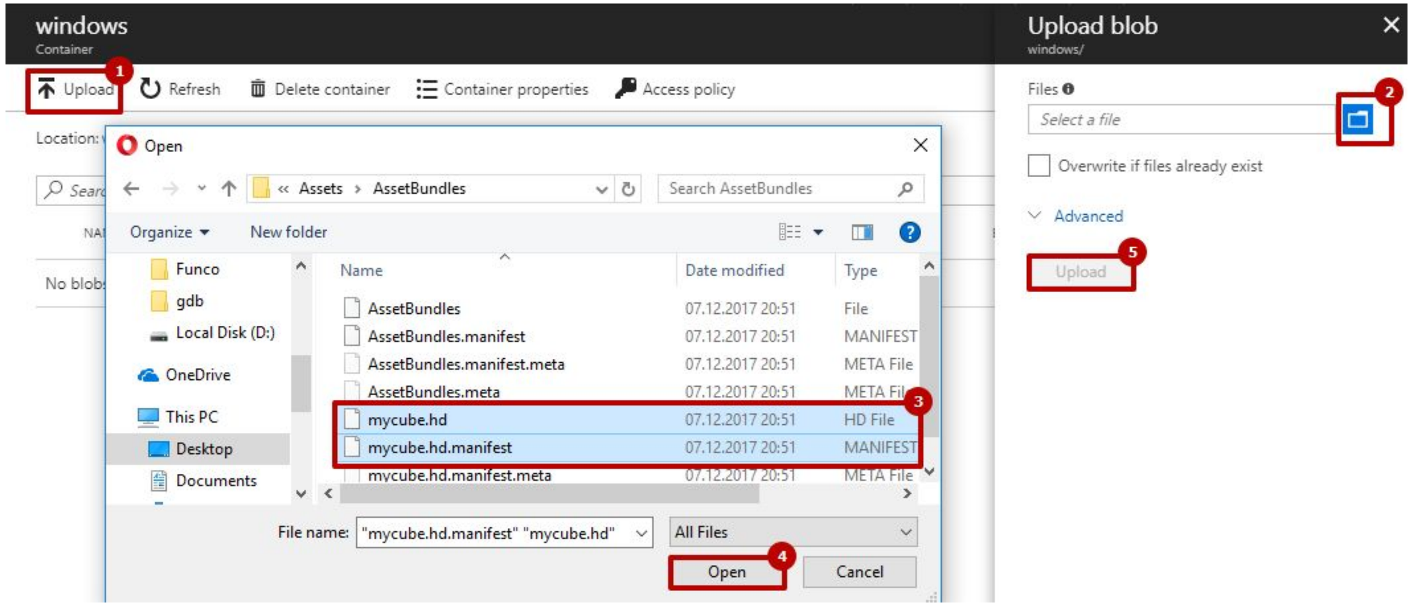
```
Audio:
  Level: -74.8 dB            DSP load: 0.7%
  Clipping: 0.0%             Stream load: 0.0%
Graphics:                    171.8 FPS (5.8ms)
  CPU: main 5.8ms  render thread 1.3ms
  Batches: 12         Saved by batching: 0
  Tris: 2.7k          Verts: 5.7k
  Screen: 673x281 - 2.2 MB
  SetPass calls: 12      Shadow casters: 5
  Visible skinned meshes: 0  Animations: 0
```

Start our game by clicking the arrow in the panel.

On the empty scene, the cube should appea in the screenshot.

Dynamic loading of game content in Unity from the cloud.

Azure


windows
Container

1 Upload | Refresh | Delete container | Container properties | Access policy

Location:

No blobs

Open

« Assets › AssetBundles

Search AssetBundles

Organize ▾ | New folder

Funco
gdb
Local Disk (D:)
OneDrive
This PC
Desktop
Documents

Name	Date modified	Type
AssetBundles	07.12.2017 20:51	File
AssetBundles.manifest	07.12.2017 20:51	MANIFEST
AssetBundles.manifest.meta	07.12.2017 20:51	META File
AssetBundles.meta	07.12.2017 20:51	META File
mycube.hd	07.12.2017 20:51	HD File
mycube.hd.manifest	07.12.2017 20:51	MANIFEST
mycube.hd.manifest.meta	07.12.2017 20:51	META File

3

File name: "mycube.hd.manifest" "mycube.hd"
All Files

4 Open | Cancel

Upload blob
windows/

Files
Select a file 2

Overwrite if files already exist

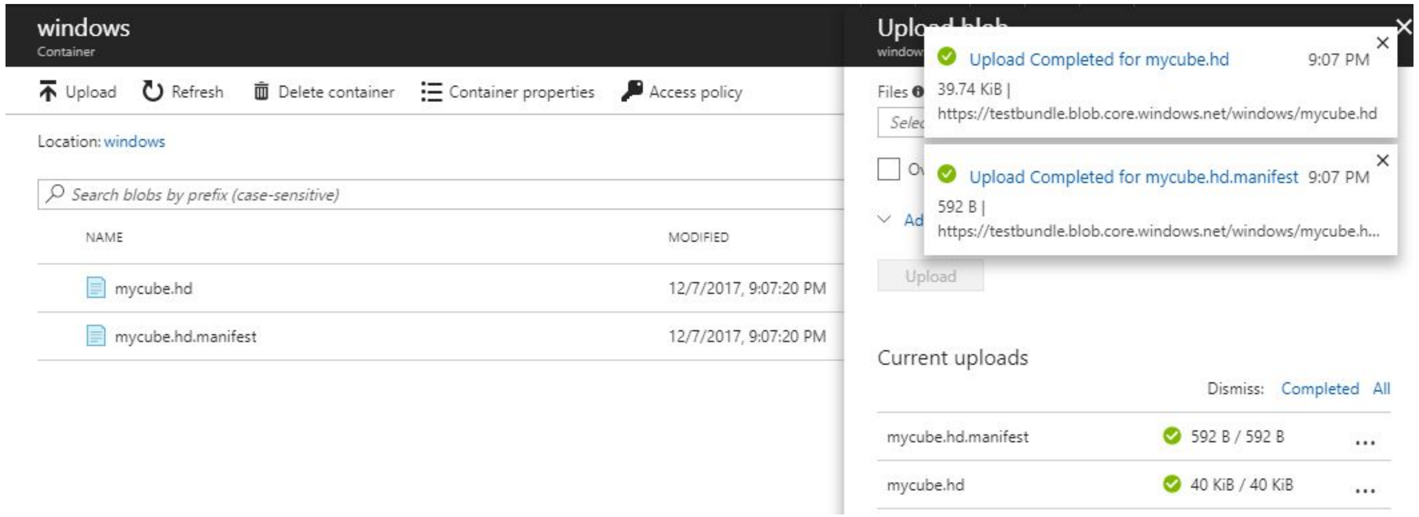
Advanced

5 Upload

Loading the bundle into the cloud

Now let's make this bundle, loaded from the Azure cloud and not from the local directory

To get started, go back to Azure and open t "windows" container, then click the Upload button, then in the screenshot below, select bundles (if you saved the project on the Azu homepage you will see the folder with the U project.) Next, click AssetBundles .


windows
Container

Upload | Refresh | Delete container | Container properties | Access policy

Location: windows

Search blobs by prefix (case-sensitive)

NAME	MODIFIED
mycube.hd	12/7/2017, 9:07:20 PM
mycube.hd.manifest	12/7/2017, 9:07:20 PM

Upload blob
windows

Upload Completed for mycube.hd 9:07 PM
39.74 KiB |
https://testbundle.blob.core.windows.net/windows/mycube.hd

Upload Completed for mycube.hd.manifest 9:07 PM
592 B |
https://testbundle.blob.core.windows.net/windows/mycube.h...

Upload

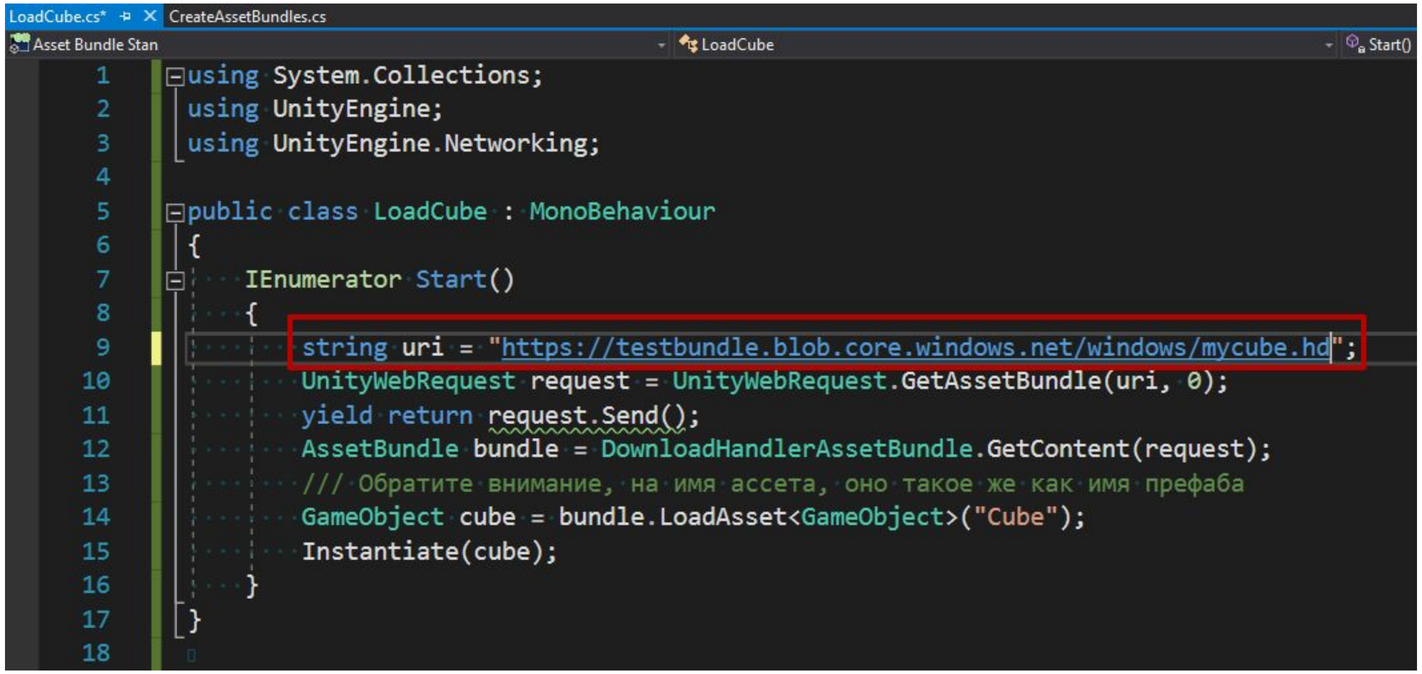
Current uploads

Dismiss: Completed All

mycube.hd.manifest	✓ 592 B / 592 B	...
mycube.hd	✓ 40 KiB / 40 KiB	...

After that, you will see how your bundles a loaded into the cloud.

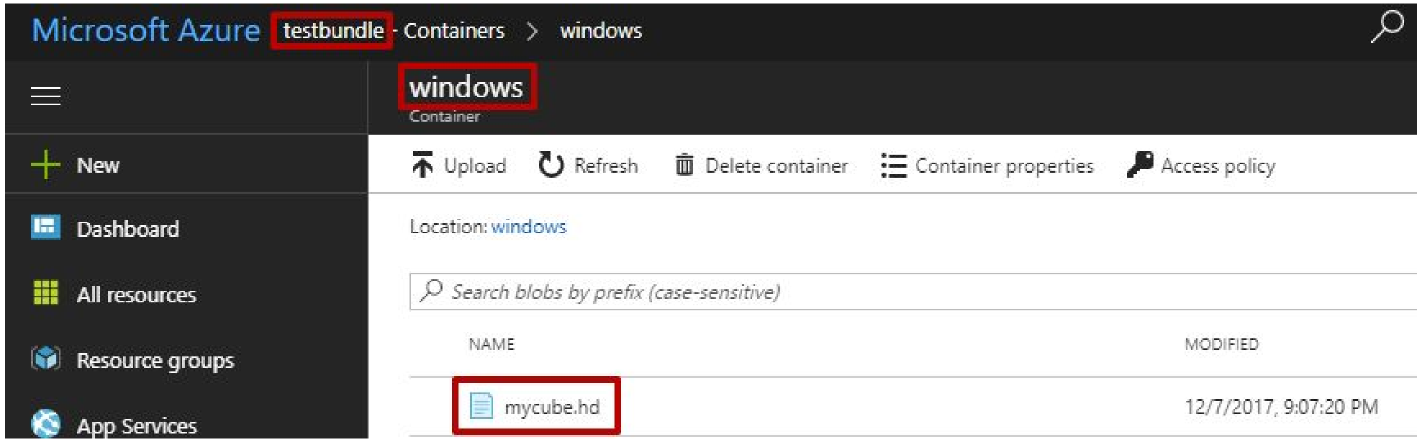
Dynamic loading of game content in Unity from the cloud.



LoadCube.cs* CreateAssetBundles.cs
Asset Bundle Stan LoadCube Start()

```
using System.Collections;
using UnityEngine;
using UnityEngine.Networking;

public class LoadCube : MonoBehaviour
{
    IEnumerator Start()
    {
        string uri = "https://testbundle.blob.core.windows.net/windows/mycube.hd";
        UnityWebRequest request = UnityWebRequest.GetAssetBundle(uri, 0);
        yield return request.Send();
        AssetBundle bundle = DownloadHandlerAssetBundle.GetContent(request);
        /// Обратите внимание, на имя ассета, оно такое же как имя префаба
        GameObject cube = bundle.LoadAsset<GameObject>("Cube");
        Instantiate(cube);
    }
}
```

Now you need to edit the LoadScript. Or rather, change the path to our bandl.



windows
Container

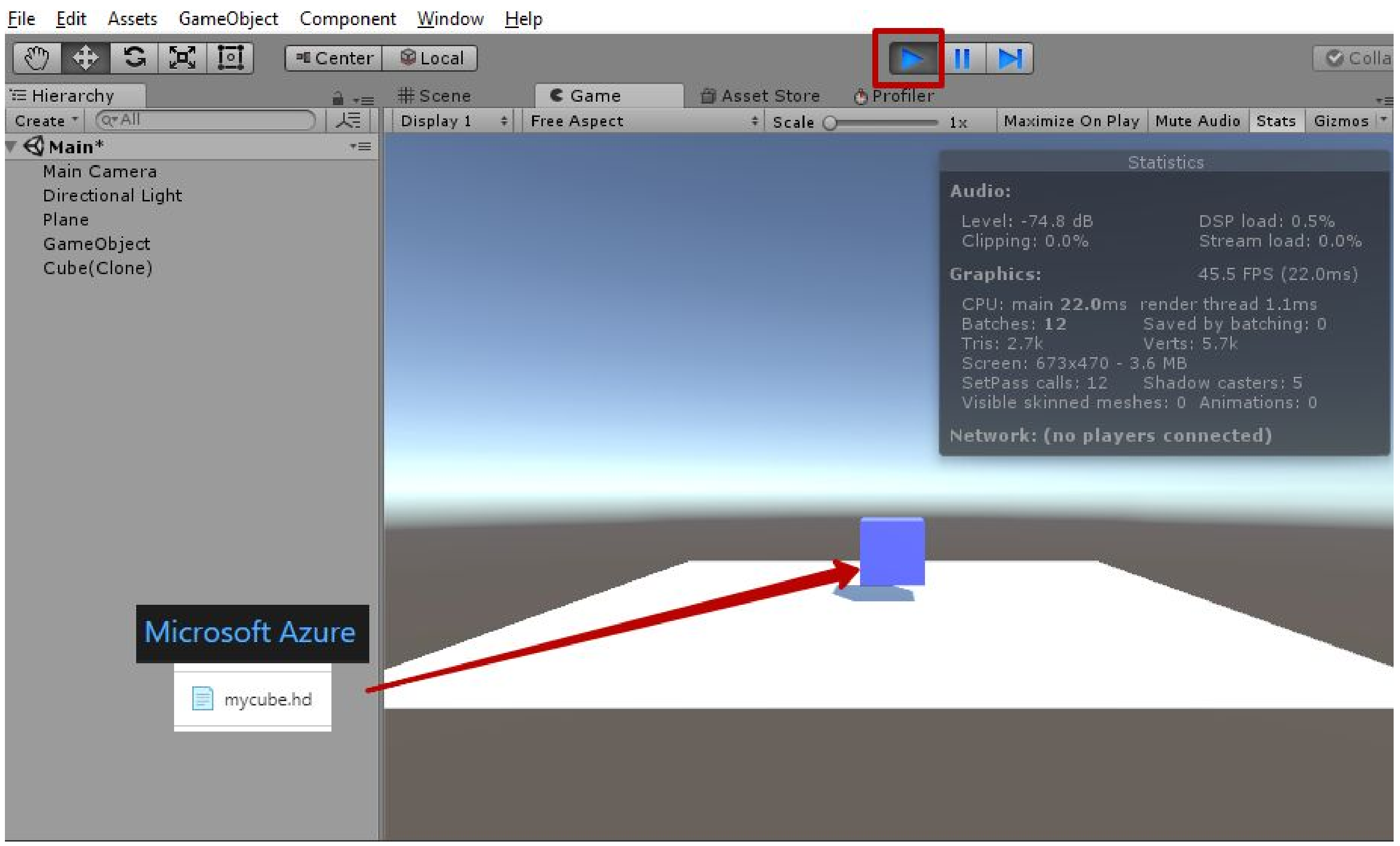
Location: windows

Search blobs by prefix (case-sensitive)

NAME	MODIFIED
mycube.hd	12/7/2017, 9:07:20 PM

The path: the name of our host, in my case https://testbundle.blob.core.windows.net/ name of the repository (windows) / the nar of the bundle itself (mycube.hd). The screenshot above shows the full path.

Dynamic loading of game content in Unity from the cloud.

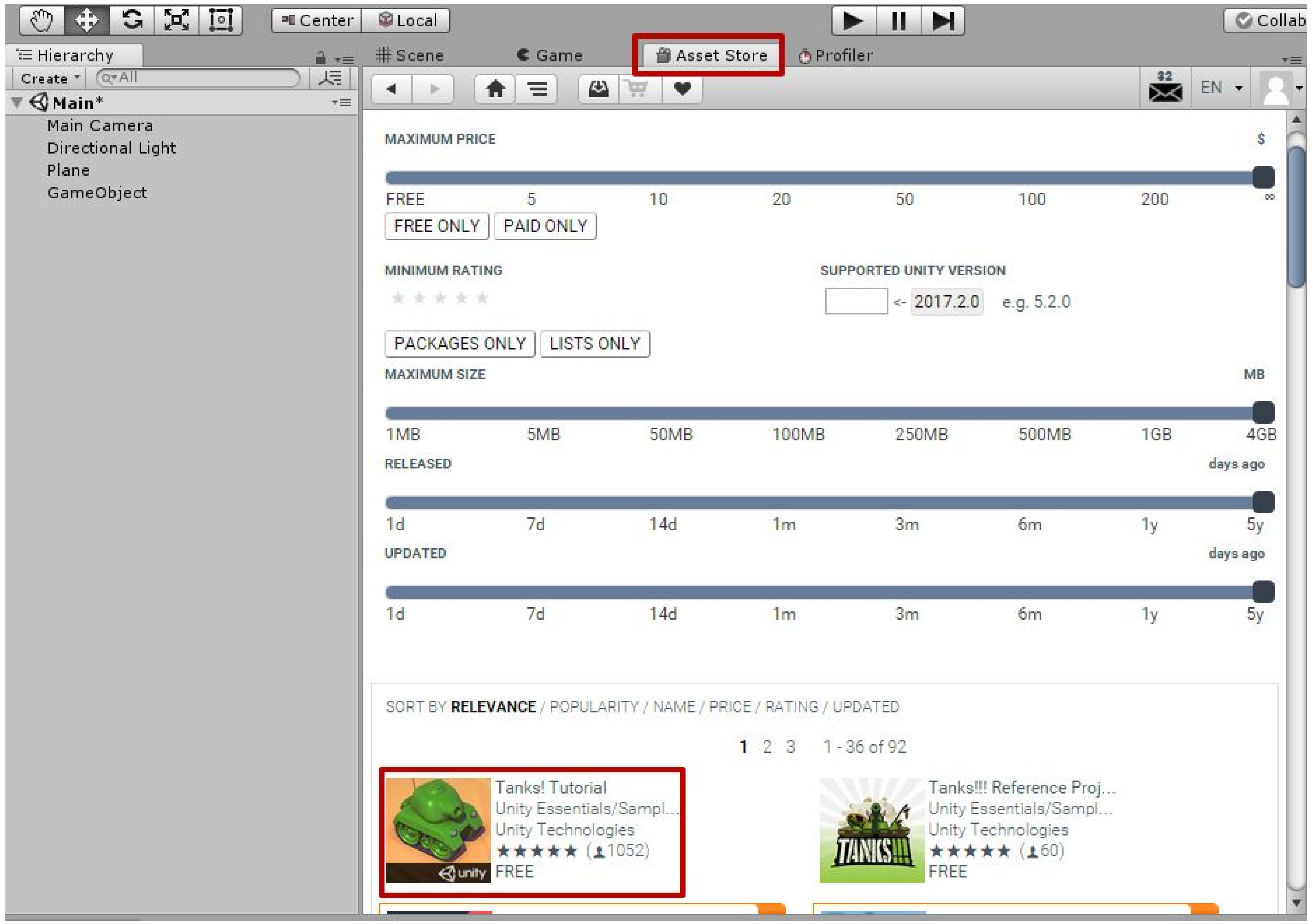


```
Statistics
Audio:
  Level: -74.8 dB              DSP load: 0.5%
  Clipping: 0.0%               Stream load: 0.0%
Graphics:                      45.5 FPS (22.0ms)
  CPU: main 22.0ms  render thread 1.1ms
  Batches: 12          Saved by batching: 0
  Tris: 2.7k           Verts: 5.7k
  Screen: 673x470 - 3.6 MB
  SetPass calls: 12    Shadow casters: 5
  Visible skinned meshes: 0  Animations: 0
Network: (no players connected)
```

Now you can run our game and see how the cube is loaded from the cloud.

If the cube was loaded, congratulations, yo were able to load the object from the Azure cloud.

Dynamic loading of game content in Unity from the cloud.



Downloading and Configuring Tanks Project

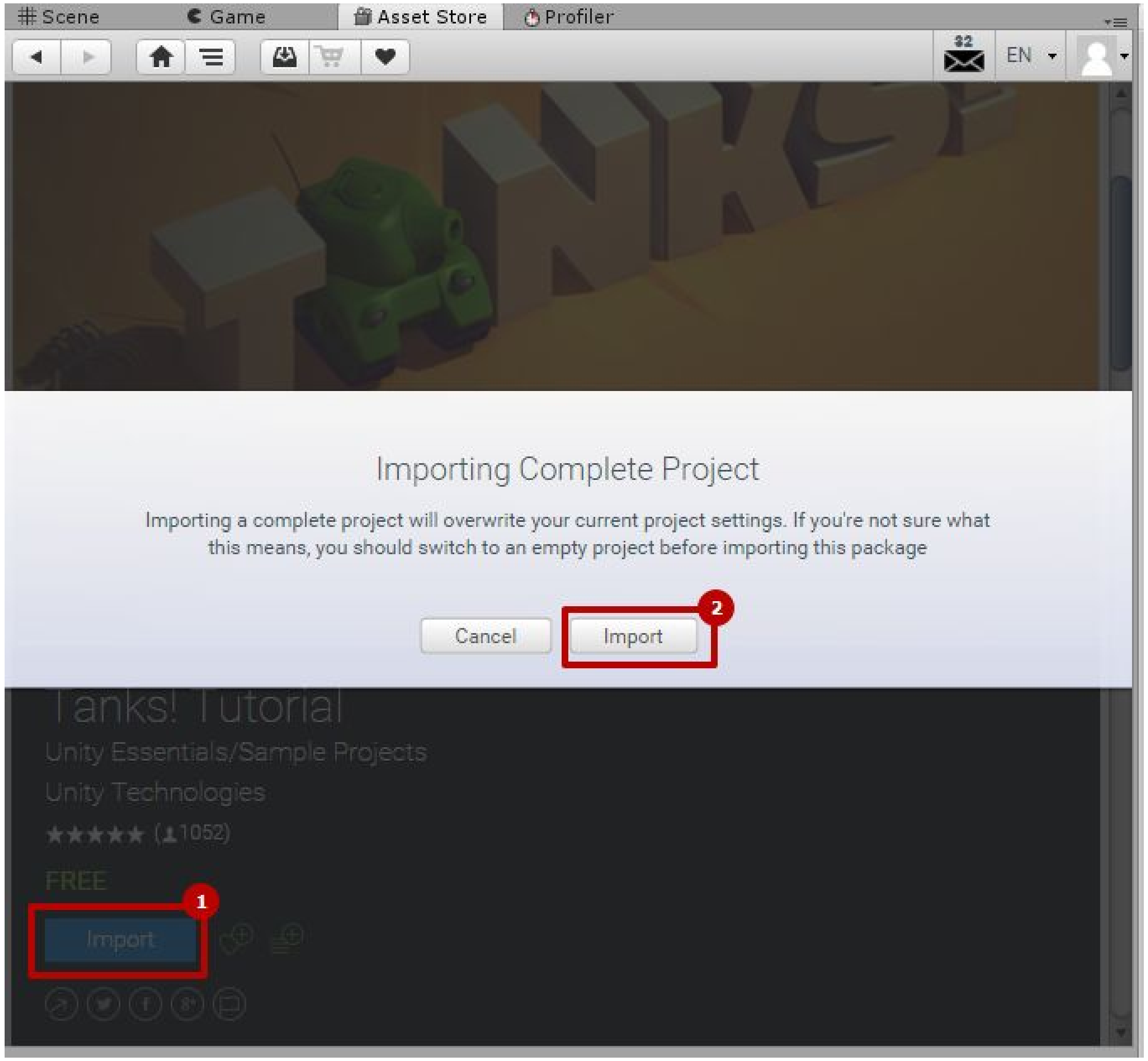
Let's now use Asset Bundles on the exampl a real game. An excellent example would be official free tutorial unity called "Tanks! Tutorial"which we download from the Unity Asset Store.

Click on the "Asset Store" tab and enter "Tanks! Tutorial "

The system can request the Unity ID which can create here https://id.unity.com/account/new

Select an Asset from the list of search resul

Dynamic loading of game content in Unity from the cloud.



Importing Complete Project

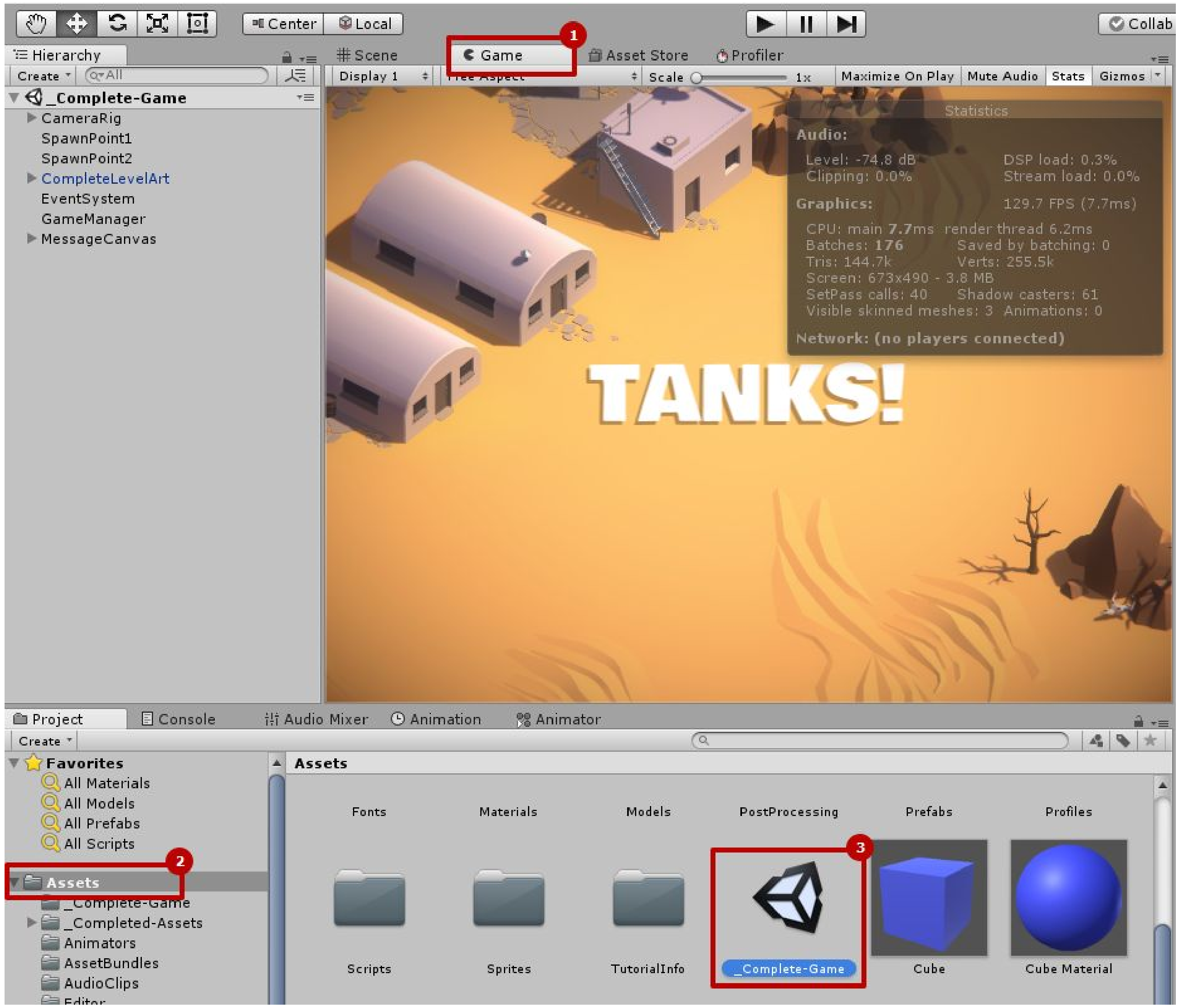
Importing a complete project will overwrite your current project settings. If you're not sure what this means, you should switch to an empty project before importing this package

Click Download to download the Asset.

When the Asset is downloaded, the Import button appears.

Then click another Import button and wait it to load into your project.

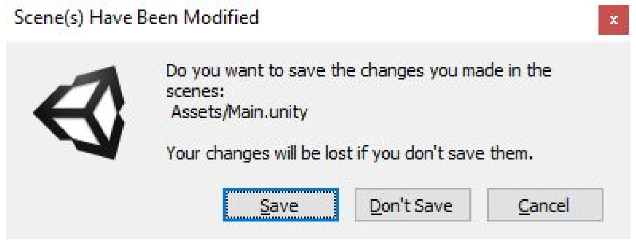
Dynamic loading of game content in Unity from the cloud.



```
Statistics
Audio:
  Level: -74.8 dB            DSP load: 0.3%
  Clipping: 0.0%             Stream load: 0.0%
Graphics:                    129.7 FPS (7.7ms)
  CPU: main 7.7ms  render thread 6.2ms
  Batches: 176        Saved by batching: 0
  Tris: 144.7k        Verts: 255.5k
  Screen: 673x490 - 3.8 MB
  SetPass calls: 40   Shadow casters: 61
  Visible skinned meshes: 3  Animations: 0
Network: (no players connected)
```

Let's now find the main scene and check th everything works. The screenshot below sh where the main scene is located ("_Complete-Game") is the root of the Assets folder.

If the system asks you to save the scene, cl Save.

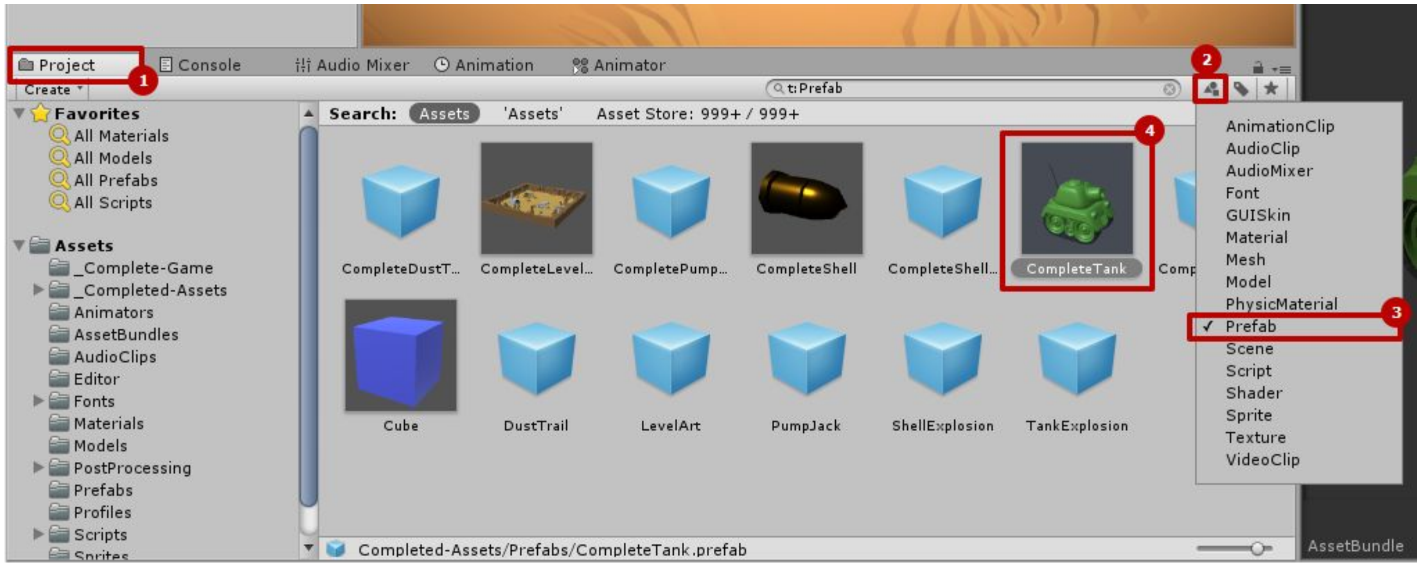


Scene(s) Have Been Modified

Do you want to save the changes you made in the scenes:
Assets/Main.unity

Your changes will be lost if you don't save them.

Dynamic loading of game content in Unity from the cloud.

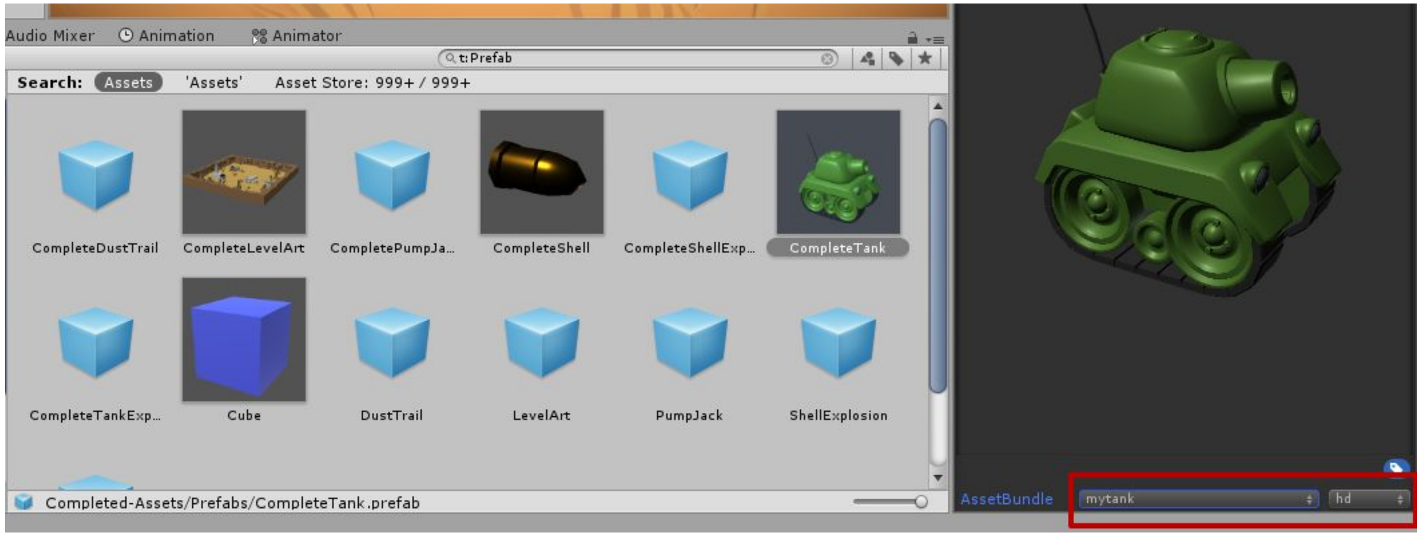


Completed-Assets/Prefabs/CompleteTank.prefab

Now we should look for the prefab of the ta
to make a bundle of it.

We will create bundle out of the prefab of t
tank.
The player will download our game without
tanks. The tanks will load from the Azure cl
and then spawn on the scene.

In order to find the prefab of the tank:
1 - select the Project window
2-3 - Define Prefab object filters
4 - Find CompleteTank Prefab.



Completed-Assets/Prefabs/CompleteTank.prefab

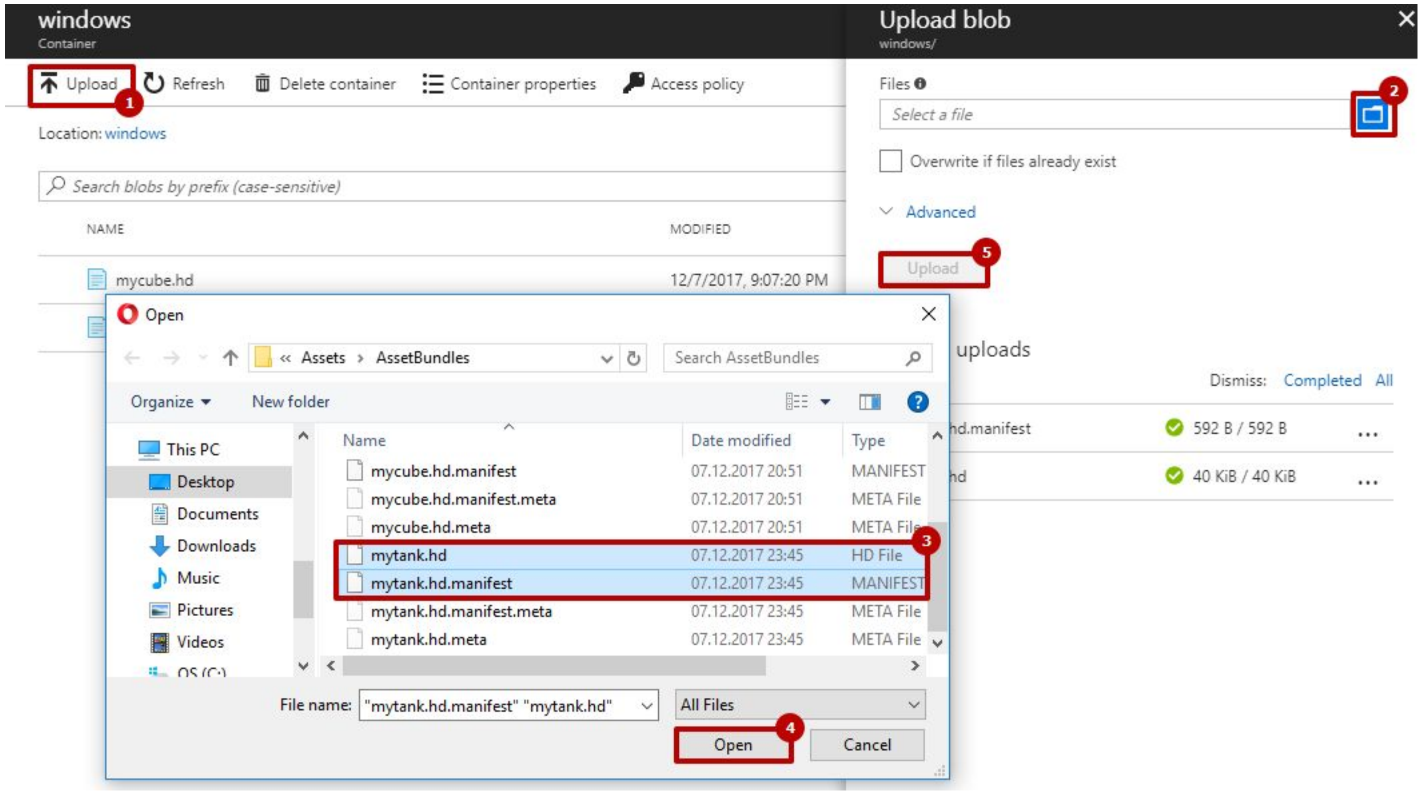
AssetBundle mytank hd

Now we need to make a bundle from the
prefab, on the same principle as the cube.

Let's set AssetBundle mytank and as the ty
hd to our tank.

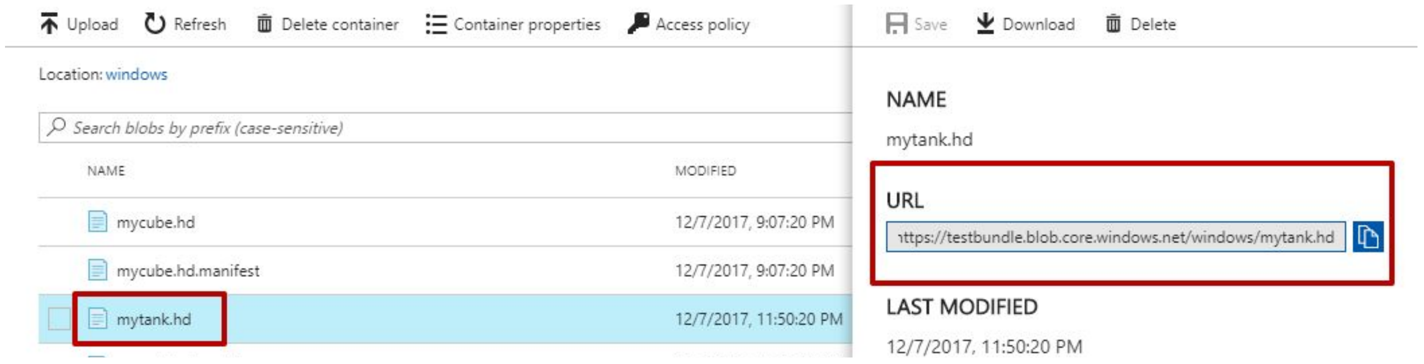
Dynamic loading of game content in Unity from the cloud.

Azure



Name	Date modified	Type
mycube.hd.manifest	07.12.2017 20:51	MANIFEST
mycube.hd.manifest.meta	07.12.2017 20:51	META File
mycube.hd.meta	07.12.2017 20:51	META File
mytank.hd	07.12.2017 23:45	HD File
mytank.hd.manifest	07.12.2017 23:45	MANIFEST
mytank.hd.manifest.meta	07.12.2017 23:45	META File
mytank.hd.meta	07.12.2017 23:45	META File

Now we should load the new bundle of our into Azure.



NAME	MODIFIED
mycube.hd	12/7/2017, 9:07:20 PM
mycube.hd.manifest	12/7/2017, 9:07:20 PM
mytank.hd	12/7/2017, 11:50:20 PM

NAME
mytank.hd

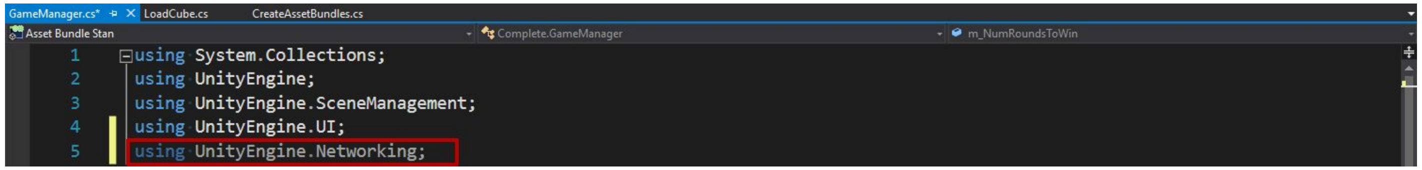
URL
https://testbundle.blob.core.windows.net/windows/mytank.hd

LAST MODIFIED
12/7/2017, 11:50:20 PM

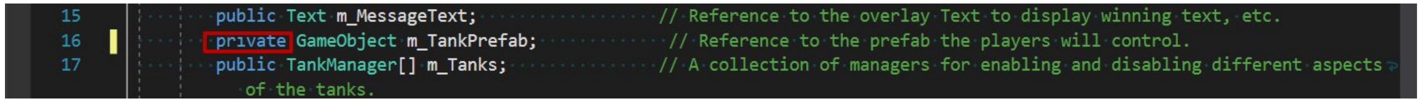
In order to easily get the path of our bundle you can simply click on the file that is downloaded and copy the path from the me to the right, the URL field.

Dynamic loading of game content in Unity from the cloud.

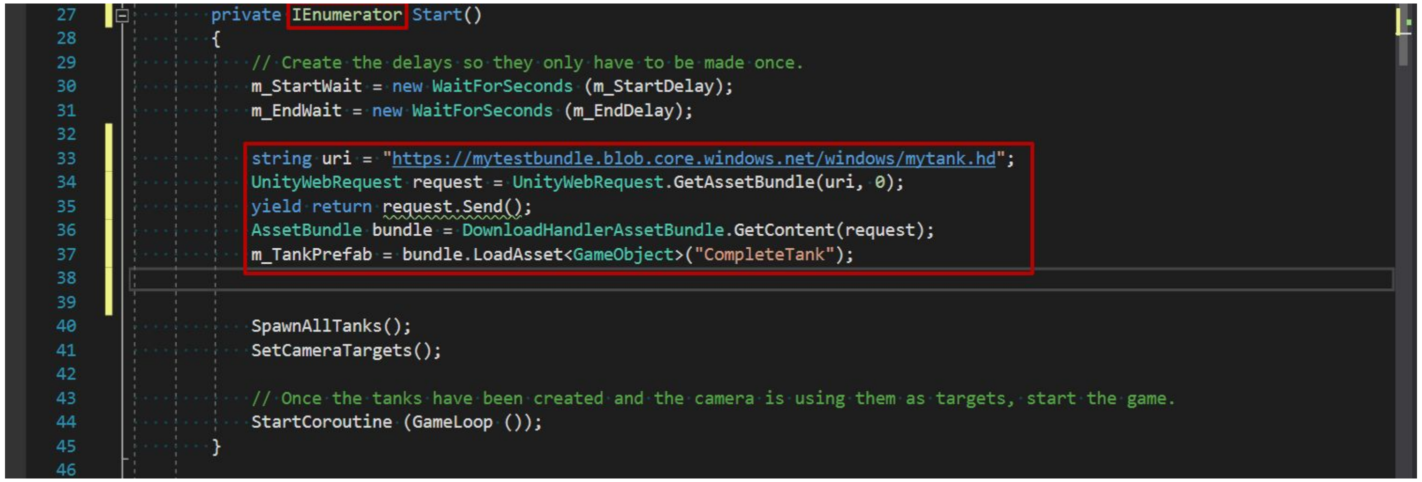
Azure



```
using System.Collections;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;
using UnityEngine.Networking;
```



```
        public Text m_MessageText;                  // Reference to the overlay Text to display winning text, etc.
        private GameObject m_TankPrefab;             // Reference to the prefab the players will control.
        public TankManager[] m_Tanks;               // A collection of managers for enabling and disabling different aspects
          of the tanks.
```



```
        private IEnumerator Start()
        {
            // Create the delays so they only have to be made once.
            m_StartWait = new WaitForSeconds (m_StartDelay);
            m_EndWait = new WaitForSeconds (m_EndDelay);

            string uri = "https://mytestbundle.blob.core.windows.net/windows/mytank.hd";
            UnityWebRequest request = UnityWebRequest.GetAssetBundle(uri, 0);
            yield return request.Send();
            AssetBundle bundle = DownloadHandlerAssetBundle.GetContent(request);
            m_TankPrefab = bundle.LoadAsset<GameObject>("CompleteTank");


            SpawnAllTanks();
            SetCameraTargets();

            // Once the tanks have been created and the camera is using them as targets, start the game.
            StartCoroutine (GameLoop ());
        }

```

Now we should write the code to download tank from the cloud.

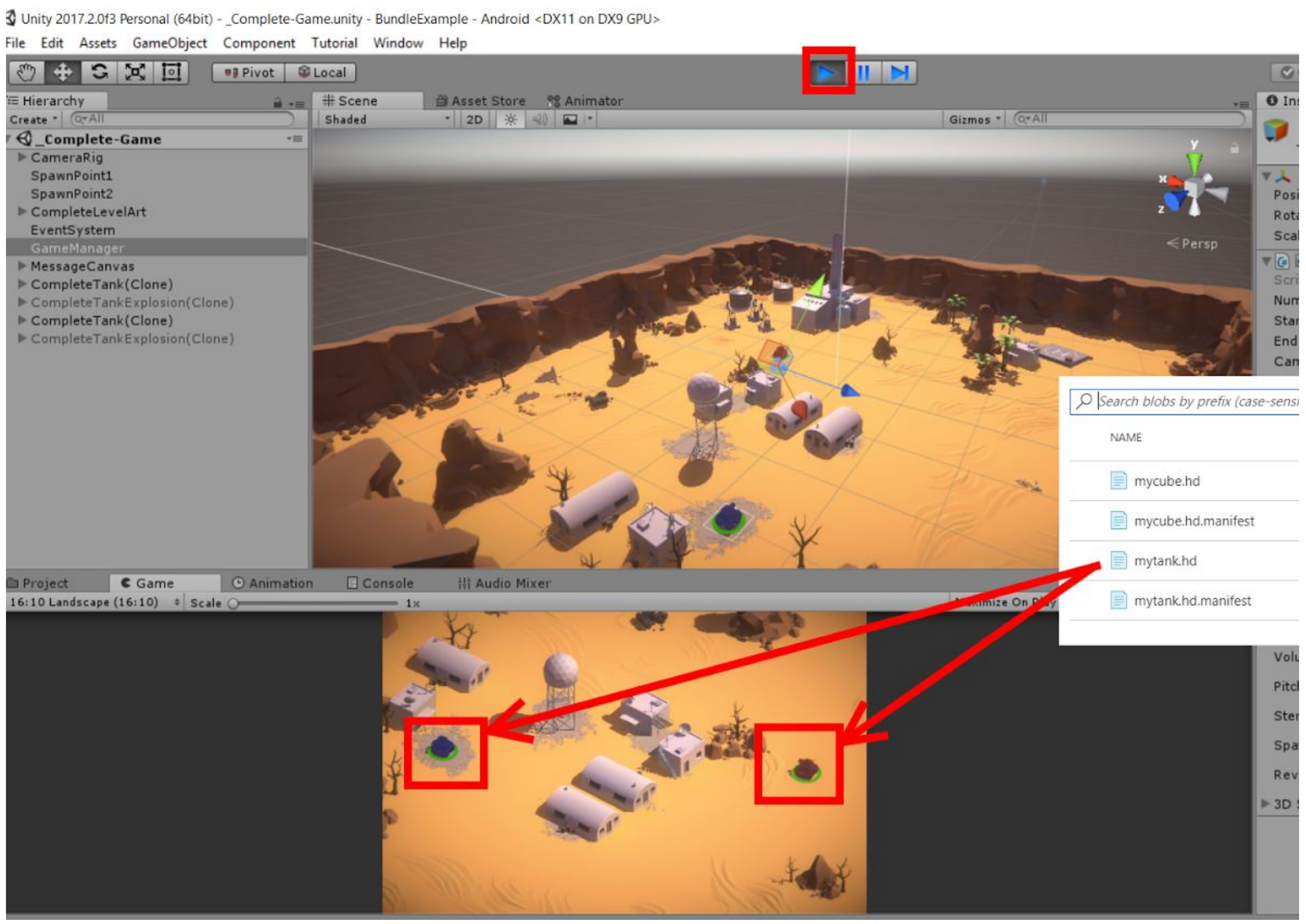
We will edit the script Assets / Scripts / Managers / GameManager.cs

The first thing we should do is to make m_TankPrefab field private, since now we c not need a prefix link from the editor. (in fa we make it impossible to display the tank without using Azure cloud)
Now we should connect the namespace usi UnityEngine.Networking;

Instead of the main Start Void method, plac Start IEnumerator.

Now you can put a piece of code to load the bundle of the tank from the Azure cloud.
Boot code of the bundle of the tank:
string uri = "https://mytestbundle.blob.core.windows.net/windo tank.hd";
UnityWebRequest request = UnityWebRequest.GetAssetBundle(uri, 0);
yield return request.Send();
AssetBundle bundle = DownloadHandlerAssetBundle.GetContent(request);
m_TankPrefab = bundle.LoadAsset<GameObject>("CompleteTank");

Dynamic loading of game content in Unity from the cloud.


Unity 2017.2.0f3 Personal (64bit) - _Complete-Game.unity - BundleExample - Android <DX11 on DX9 GPU>
File Edit Assets GameObject Component Tutorial Window Help
Hierarchy
_Complete-Game
CameraRig
SpawnPoint1
SpawnPoint2
CompleteLevelArt
EventSystem
GameManager
MessageCanvas
CompleteTank(Clone)
CompleteTankExplosion(Clone)
CompleteTank(Clone)
CompleteTankExplosion(Clone)
Scene
Asset Store
Animator
Game
Search blobs by prefix (case-sensi
NAME
mycube.hd
mycube.hd.manifest
mytank.hd
mytank.hd.manifest

Excellent work, now our tanks are loaded o the scene right from the Azure cloud.

https://github.com/rio900/unityazurebunc