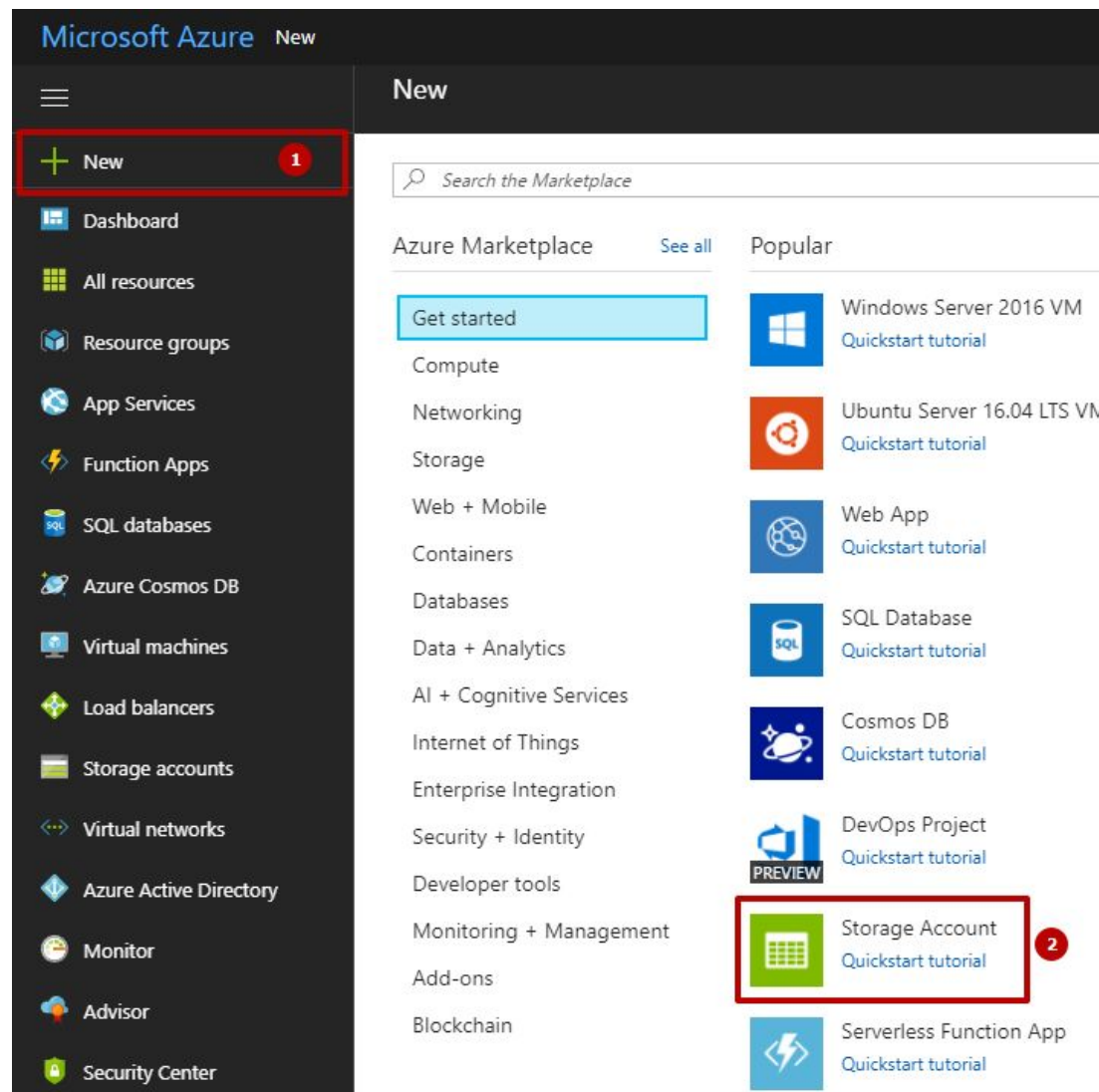


Динамическая подгрузка игрового контента в Unity из облака.



Создание и настройка необходимых сервисов.

Для начала вам необходимо авторизоваться в существующий аккаунт Azure. Если у Вас нет аккаунта, пожалуйста вернитесь в предыдущее руководство по созданию аккаунта Azure и активации Azure Pass баланса на 100\$ для использования Azure в течении месяца.

Первое что нам нужно будет сделать это **создать Storage Account**. Windows Azure Storage это система облачного хранилища, предоставляющая клиентам возможность хранить практически неограниченные объемы данных в течение любого периода времени. **Мы его будем использовать для хранения игровых ассет бандлов для нашей игры.**

Нажмите **New** в левой панели а затем выберете **Storage Account**.

Динамическая подгрузка игрового контента в Unity из облака.



Microsoft Azure New > Create storage account

Create storage account

The cost of your storage account depends on the usage and the options you choose below.
[Learn more](#)

1 * Name ⓘ
testbundle ✓
core.windows.net

Deployment model ⓘ
Resource manager Classic

2 Account kind ⓘ
Blob storage ▼

Performance ⓘ
Standard Premium

3 Replication ⓘ
Locally-redundant storage (LRS) ▼

Access tier (default) ⓘ
Cool Hot

Для создания стораджа необходимо заполнить несколько важных полей.

1 - Name название нашего хранилища.

2 - Account kind выбираем **Blob storage**. Blob расшифровывается как Binary Large Object.

3 - Replication Locally-redundant storage

Динамическая подгрузка игрового контента в Unity из облака.



Storage accounts

Virtual networks

Azure Active Directory

Monitor

Advisor

Security Center

Cost Management + Billing

Help + support

More services >

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

Create

Automation options

4

5

6

7

4 - Subscription мы выбираем тип текущей подписки. Если вы активировали Azure Pass тогда система вам предложит выбрать именно этот тип подписки.

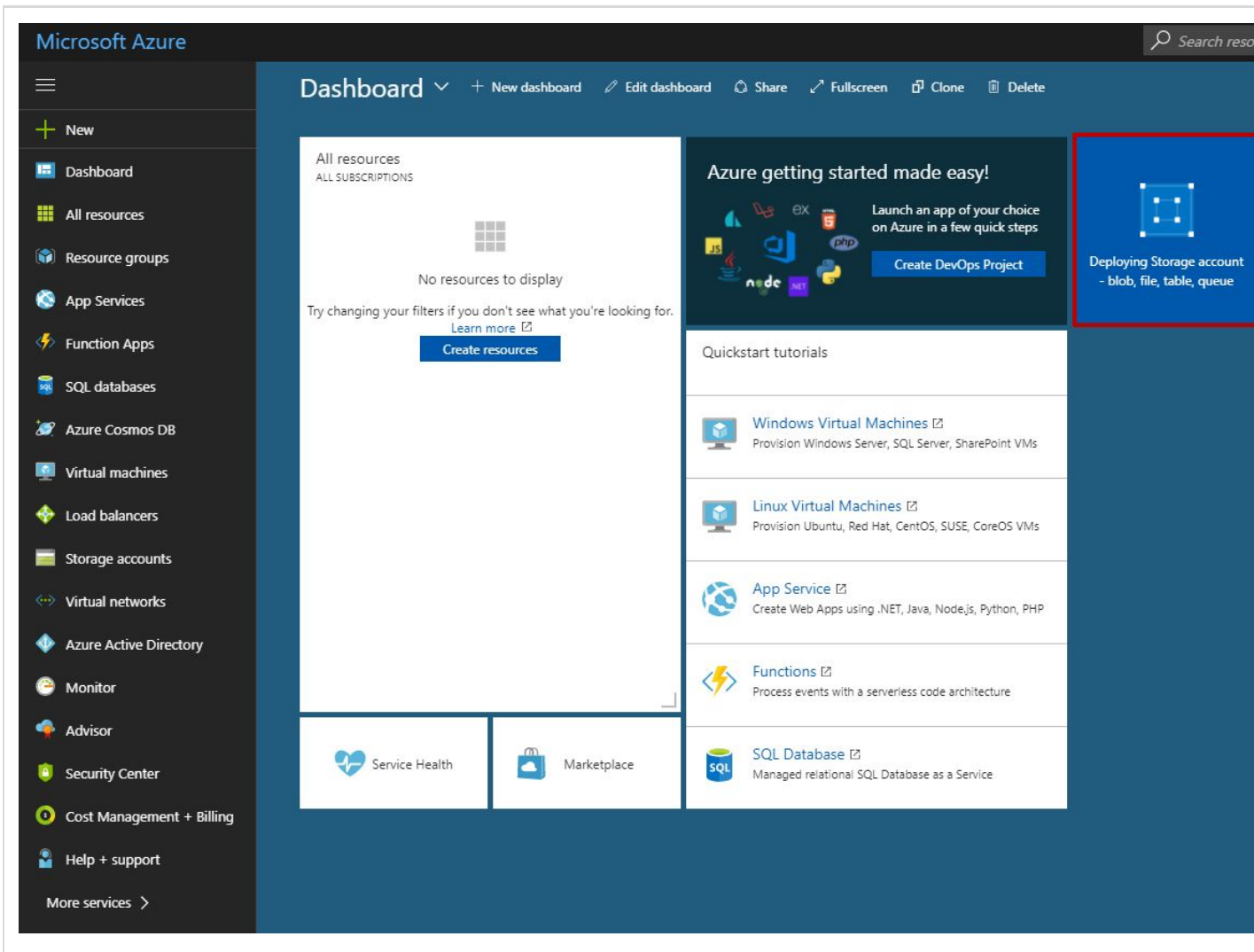
5 - Resource Group - в него нужно придумать и ввести название для текущей группы ресурсов. *Группы ресурсов (resource groups)*, в рамках их можно создавать нужные совокупности приложений, сервисов и баз данных.

6 - Location укажите местоположение серверов которые ближе находятся к вам.

7 - Pin to dashboard чтобы на главной странице появилась ссылка на наш сторедж.

Нажмите кнопку **Create** для создания.

Динамическая подгрузка игрового контента в Unity из облака.



После того как мы нажмем кнопку **Create** у нас на главной странице должна появиться ссылка на наш сторадж. На скриншоте показан процесс разворачивания стораджа на клауде.

Динамическая подгрузка игрового контента в Unity из облака.



testbundle
Storage account

Search (Ctrl+*/*)

Open in Explorer → Move Delete storage account

Resource group (change)
testbundlegroup

Status
Primary: Available

Location
West Europe

Subscription (change)
Azure Pass

Subscription ID

Performance/Access tier
Standard/Hot

Replication
Locally-redundant storage (LRS)

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

Account kind
Blob storage

В дальнейшем для работы Вам будет необходим адрес стореджа, его можно посмотреть в **Overview**.

В нашем случае это:
testbundle.blob.core.windows.net

testbundle - Containers
Storage account

Search (Ctrl+*/*)

+ Container Refresh

Storage account
testbundle

Status
Primary: Available

Location
West Europe

Subscription (change)
Azure Pass

Subscription ID

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

SETTINGS

Containers

Access keys

NAME LAST MODIFIED PUBLIC AC

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

Теперь пришло время создать контейнер для хранения наших игровых бандлов. Бандлы для каждой платформы отличаются и в Unity это разные файлы, у которых одинаковое только название.

Динамическая подгрузка игрового контента в Unity из облака.



testbundle - Containers
Storage account

Search (Ctrl+/)

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems

SETTINGS
Containers
Access keys

+ Container Refresh

New container

* Name
windows

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

OK Cancel

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

1 - Name - В данном примере мы назвали свой контейнер windows, так как в нем будут храниться бандля игры для платформы Windows. Название не имеет значения и выбрано для удобства.

2 - Public access level. Также можно настроить уровень доступа к нашему хранилищу.

Затем нажимаем кнопку **Ok**

testbundle - Containers
Storage account

Search (Ctrl+/)

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems

SETTINGS
Containers
Access keys
Configuration

+ Container Refresh

Storage account
testbundle

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

Status
Primary: Available

Location
West Europe

Subscription (change)
Azure Pass

Subscription ID
[REDACTED]

Search containers by prefix

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

После удачного создания у вас должен появиться новый контейнер.

Успешно создав контейнер, нам необходимо создать Bundle в Unity, который мы будем хранить в нашем контейнере.

Пожалуйста не закрывайте окно Azure, мы вернемся к нему сразу после того как пройдем необходимые этапы в Unity.

Динамическая подгрузка игрового контента в Unity из облака.



Unity 2017.2.0f3

Projects

Learn

New

Open

My Account

On Disk

In the Cloud

Unity Project 1

Unity Project 2

Unity Project 3

Unity Project 4

Unity Project 5

Unity Project 6

Unity Project 7

Unity Project 8

Unity Project 9

Unity Project 10

Unity Project 11

Unity Project 12

Unity Project 13

Unity Project 14

Unity Project 15

Unity Project 16

Unity Project 17

Unity Project 18

Unity Project 19

Unity Project 20

Unity Project 21

Unity Project 22

Unity Project 23

Unity Project 24

Unity Project 25

Unity Project 26

Unity Project 27

Unity Project 28

Unity Project 29

Unity Project 30

Unity Project 31

Unity Project 32

Unity Project 33

Unity Project 34

Unity Project 35

Unity Project 36

Unity Project 37

Unity Project 38

Unity Project 39

Unity Project 40

Unity Project 41

Unity Project 42

Unity Project 43

Unity Project 44

Unity Project 45

Unity Project 46

Unity Project 47

Unity Project 48

Unity Project 49

Unity Project 50

Unity Project 51

Unity Project 52

Unity Project 53

Unity Project 54

Unity Project 55

Unity Project 56

Unity Project 57

Unity Project 58

Unity Project 59

Unity Project 60

Unity Project 61

Unity Project 62

Unity Project 63

Unity Project 64

Unity Project 65

Unity Project 66

Unity Project 67

Unity Project 68

Unity Project 69

Unity Project 70

Unity Project 71

Unity Project 72

Unity Project 73

Unity Project 74

Unity Project 75

Unity Project 76

Unity Project 77

Unity Project 78

Unity Project 79

Unity Project 80

Unity Project 81

Unity Project 82

Unity Project 83

Unity Project 84

Unity Project 85

Unity Project 86

Unity Project 87

Unity Project 88

Unity Project 89

Unity Project 90

Unity Project 91

Unity Project 92

Unity Project 93

Unity Project 94

Unity Project 95

Unity Project 96

Unity Project 97

Unity Project 98

Unity Project 99

Unity Project 100

Подключение сервисов к чистому Unity проекту

В данном проекте мы использовали версию Unity 2017.2.0f3

Для начала запустите Unity и создайте новый проект.

Динамическая подгрузка игрового контента в Unity из облака.



Unity 2017.2.0f3

Projects Learn

New Open My Account

Project name

Asset Bundle Stan

1

Location

C:\Users\Stanislav\Desktop

...

2

3D 2D

Add Asset Package

OFF

Enable Unity Analytics

?

Cancel

Create project

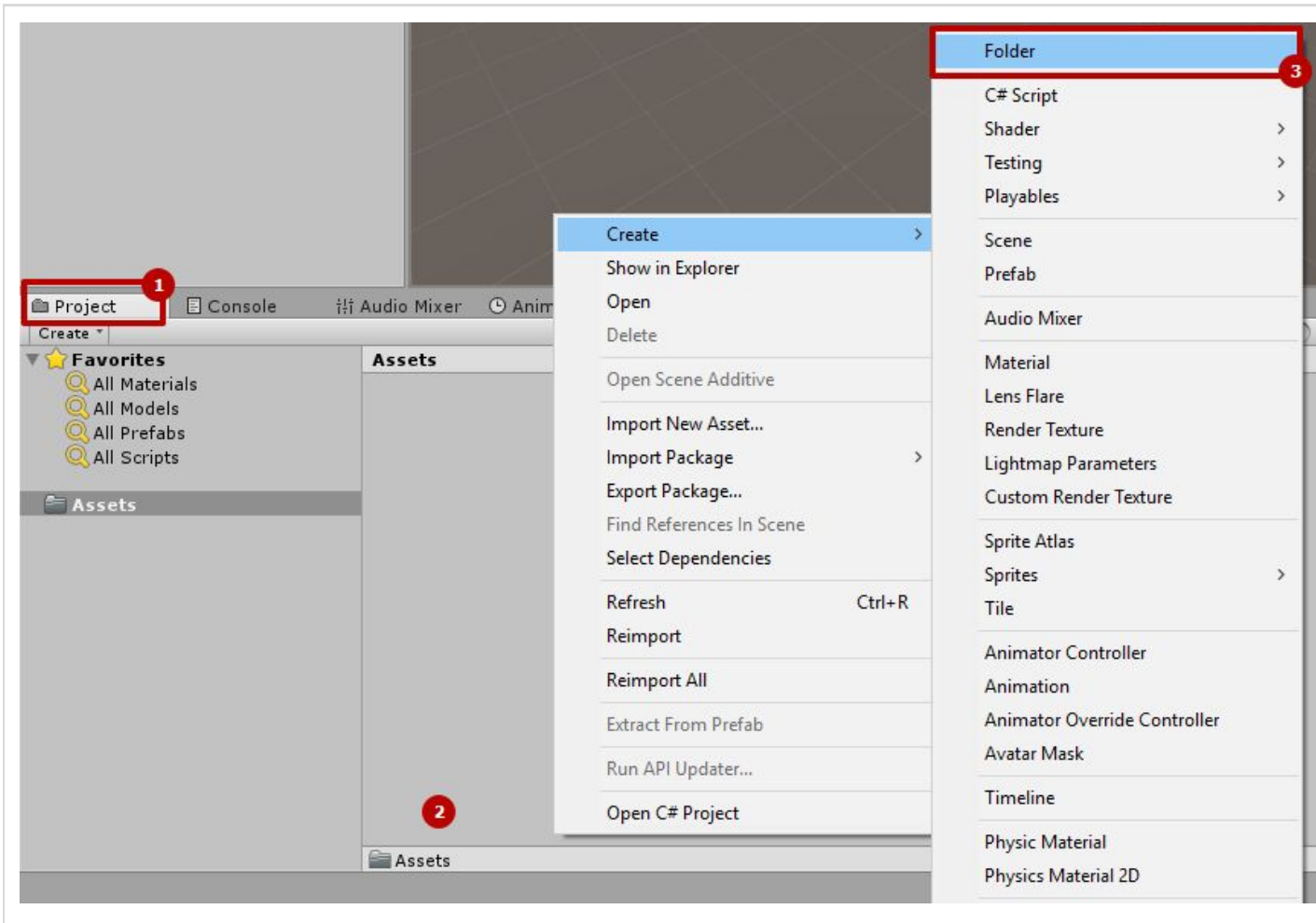
3

1 - Project name - введите уникальное название вашего нового проекта.

2 - Location - укажите директорию где будет храниться новый проект.

3 - Create project нажмите чтобы запустить создание нового проекта.

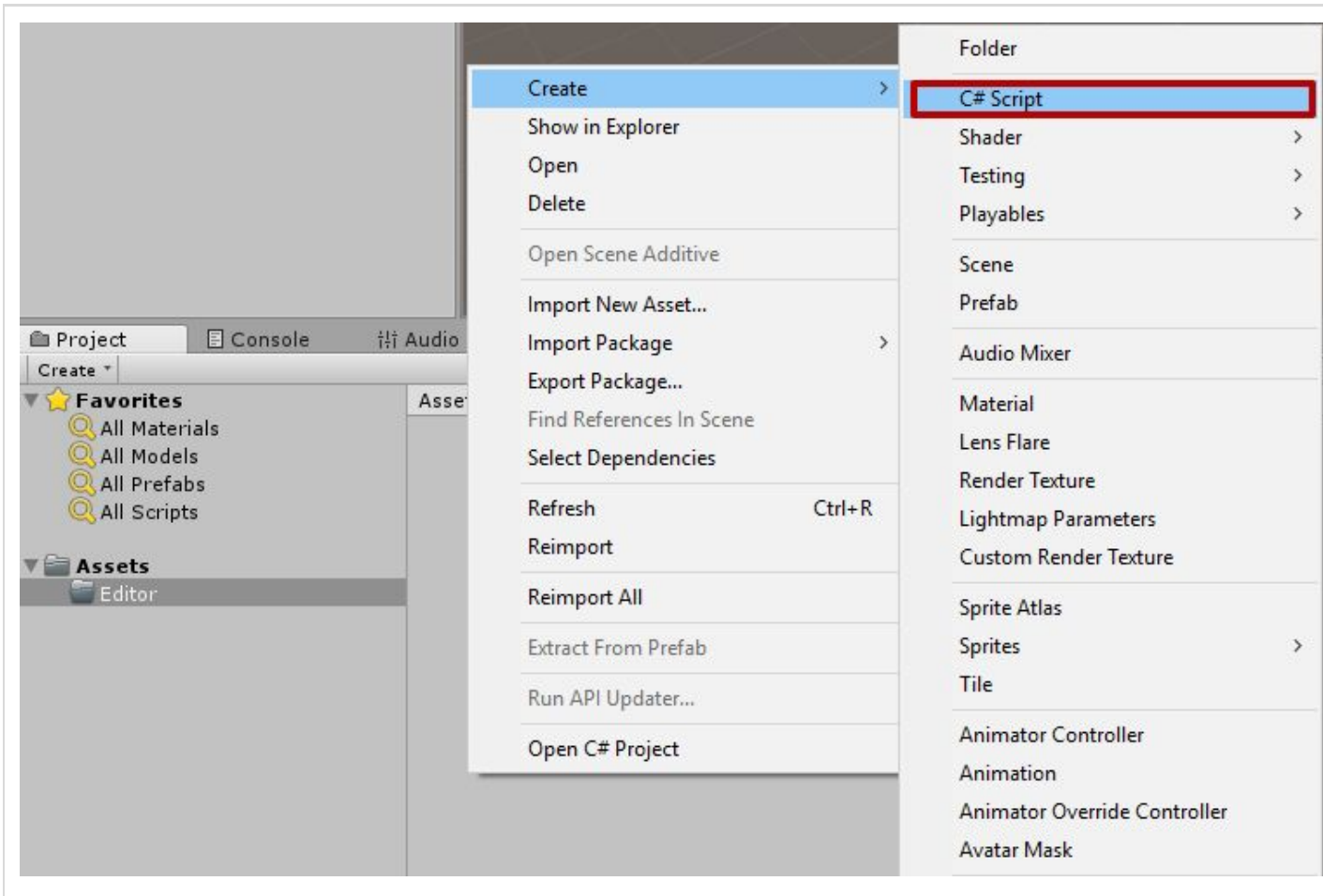
Динамическая подгрузка игрового контента в Unity из облака.



В окне **Project** создаем папку **Editor** для этого

- 1** - перейдем в окно Project
- 2** - правок кнопкой мыши на пустой области выбираем в меню Create
- 3** - Folder. Создаем новую папку с названием **"Editor"** и заходим в нашу новую папку.

Динамическая подгрузка игрового контента в Unity из облака.



В нашей папке правой кнопкой мыши, мы выбираем **"C# Script"** и создаем новый C# скрипт с названием **CreateAssetBundles**.

Благодаря этому скрипту мы сможем в редакторе создавать Asset Bundles.

Откройте редактор скрипта двойным щелчком мыши по файлу созданного скрипта.

Динамическая подгрузка игрового контента в Unity из облака.



```
CreateAssetBundles.cs
Asset Bundle Stan.Editor
CreateAssetBundles
BuildAllAssetBundles()

1 using System.IO;
2 using UnityEditor;
3
4 public class CreateAssetBundles
5 {
6     [MenuItem("Assets/Build AssetBundles")]
7     static void BuildAllAssetBundles()
8     {
9         string assetBundleDirectory = "Assets/AssetBundles";
10        if (!Directory.Exists(assetBundleDirectory))
11        {
12            Directory.CreateDirectory(assetBundleDirectory);
13        }
14        BuildPipeline.BuildAssetBundles(assetBundleDirectory, BuildAssetBundleOptions.None,
15        BuildTarget.StandaloneWindows);
16    }
17 }
18
```

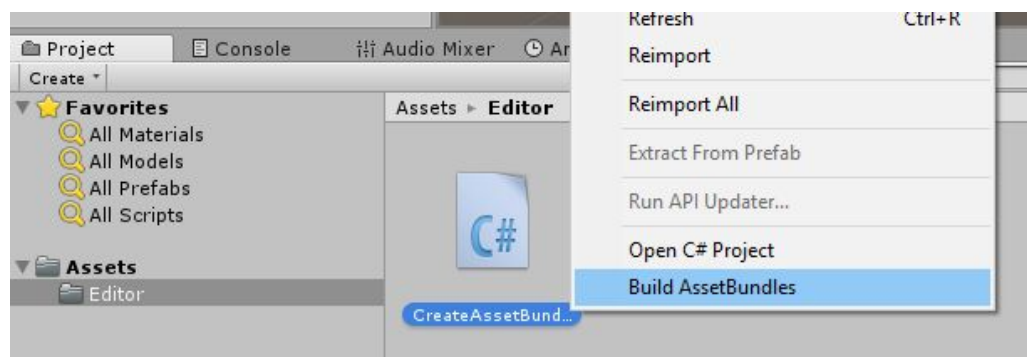
Обратите внимание: в методе

`BuildPipeline.BuildAssetBundles(assetBundleDirectory, BuildAssetBundleOptions.None, BuildTarget.StandaloneWindows);`
третьим параметром указана платформа для которой создается бандл в данном случае это **Windows**.

Замените код по умолчанию, кодом который указан ниже и нажмите **CTRL + S** чтобы сохранить:

```
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);
    }
}
```

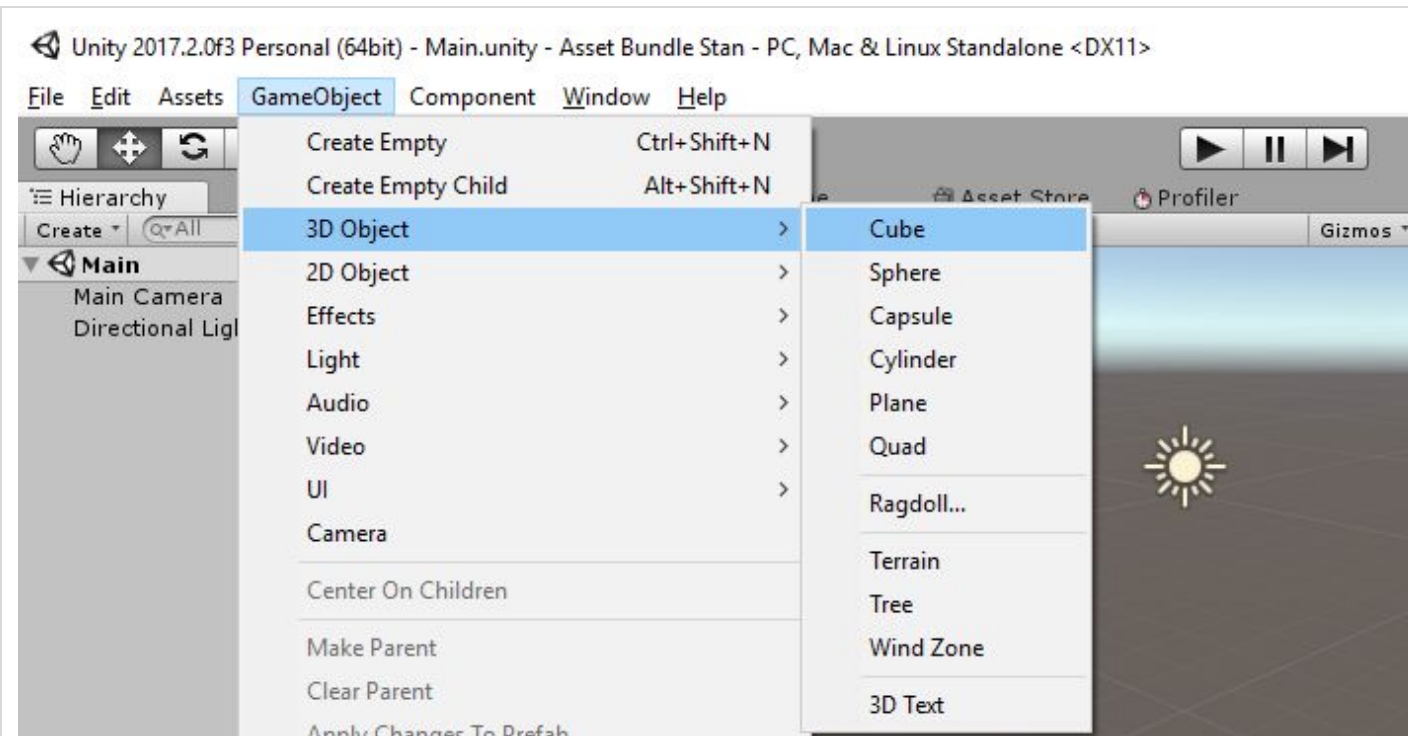


Можно обратить внимание что в меню редактора в **Assets** появится кнопочка **Build AssetBundles**. При нажатии на эту кнопку сработает метод `BuildAllAssetBundles`. В теле этого метода создается папочка **AssetBundles** куда и будут помещены бандлы. Эти бандлы мы и будем загружать потом в облако Azure.

На всякий случай мы поместили ссылку на документацию для текущей версии юнити:

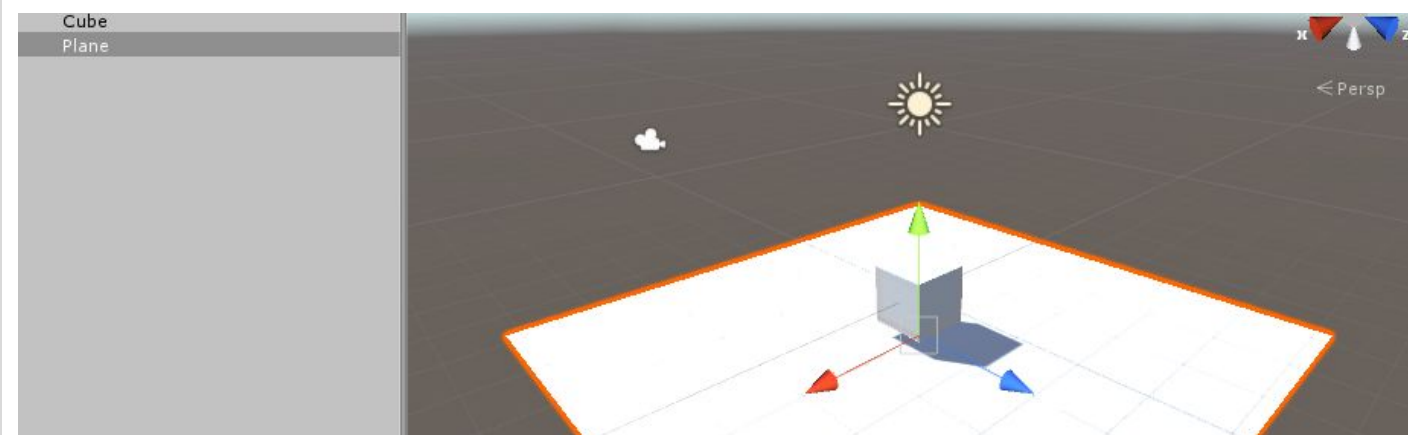
<https://docs.unity3d.com/Manual/AssetBundles-Workflow.html>

Динамическая подгрузка игрового контента в Unity из облака.



Теперь приступим к созданию наших игровых бандлов. Первый наш бандл сделаем из префаба кубика. Сохраняем и называем нашу сцену проекта. **(CTRL+SHIFT+S)** В нашем случае мы ее назвали **Main**.

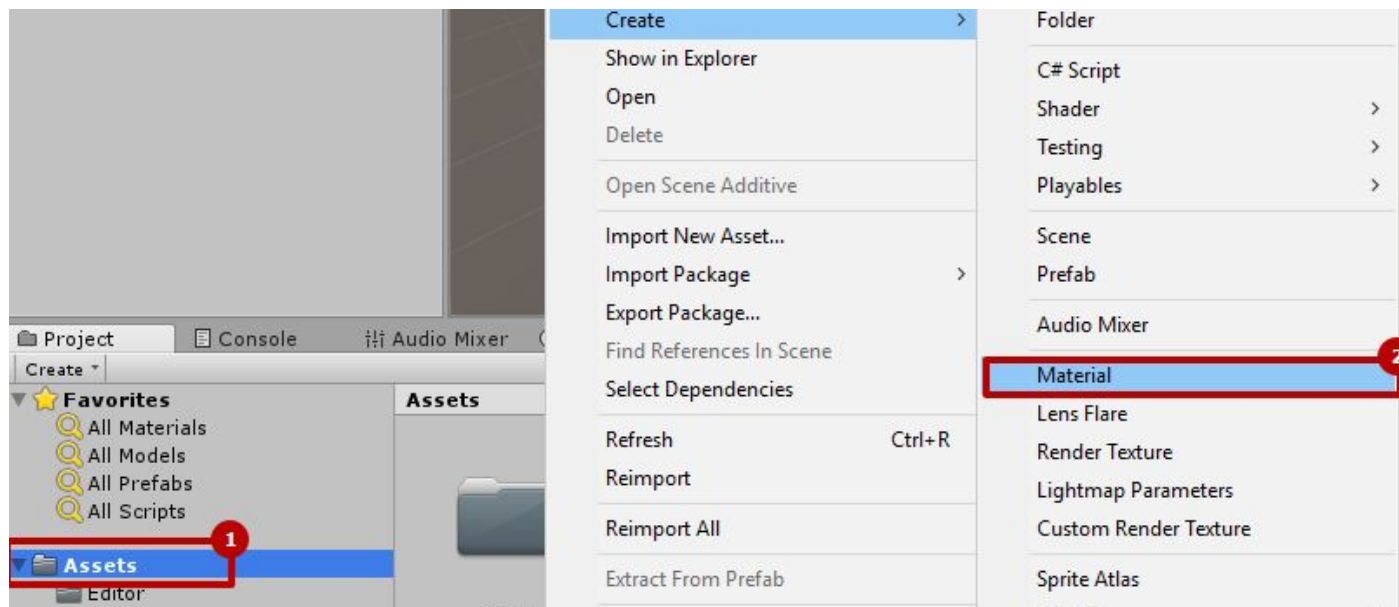
Добавляем на сцену кубик (с которого потом будем делать bundle) для этого в верхней панели меню нажимаем **GameObject -> 3D Object -> Cube**



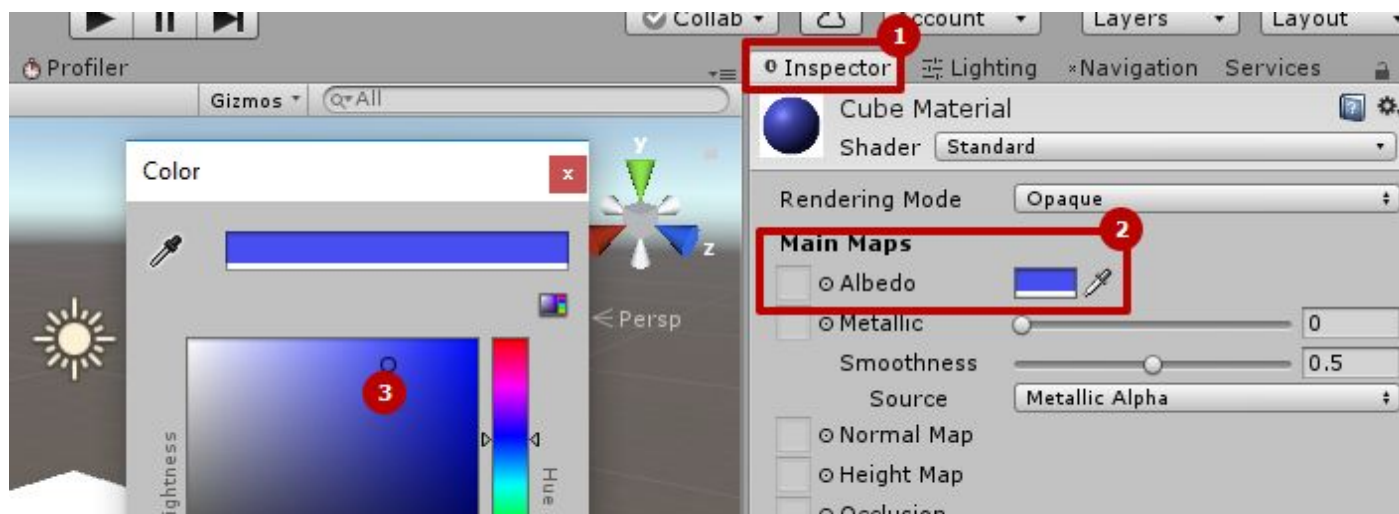
Добавляем на сцену плейн (Это наша карта, она будет статическая) **GameObject -> 3D Object -> Plane**

Для того чтобы визуально разделить куб и карту потяните за зеленую стрелочку, которая отвечает за вертикальное позиционирование объектов.

Динамическая подгрузка игрового контента в Unity из облака.

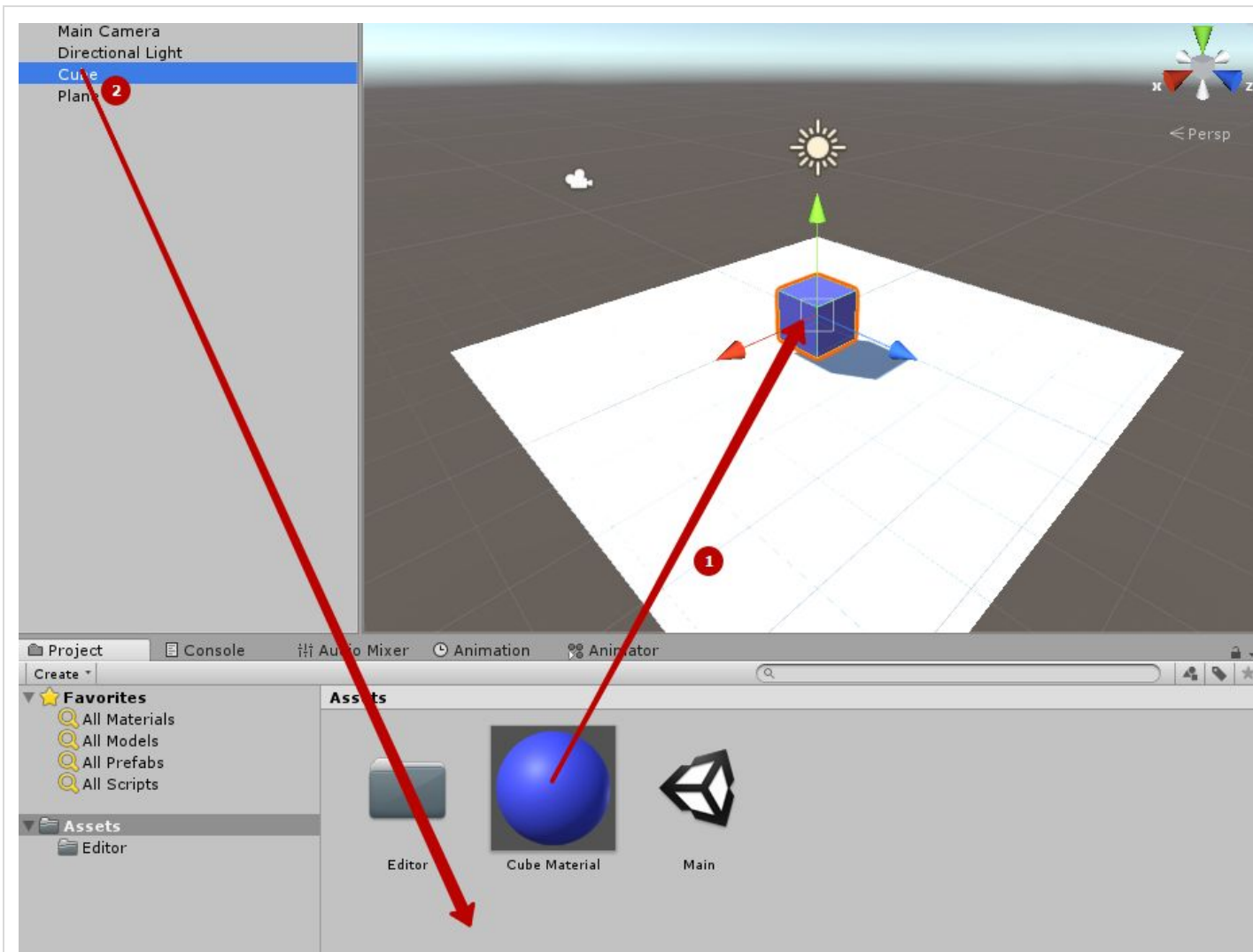


Для того чтобы наш куб был особенным мы создадим для него материал любого цвета. Для этого перейдем в папку **Assets** и правой кнопкой мыши создадим **Material** с названием **"Cube Material"**. При клике на новосозданный материал вы сможете изменить его цвет.



Для этого в правом верхнем углу в окне **Inspector** нажмите на окошко рядом с названием Albedo - оно поможет изменить цвет. А так же выберите в палитре цвет который вам больше всего по душе.

Динамическая подгрузка игрового контента в Unity из облака.

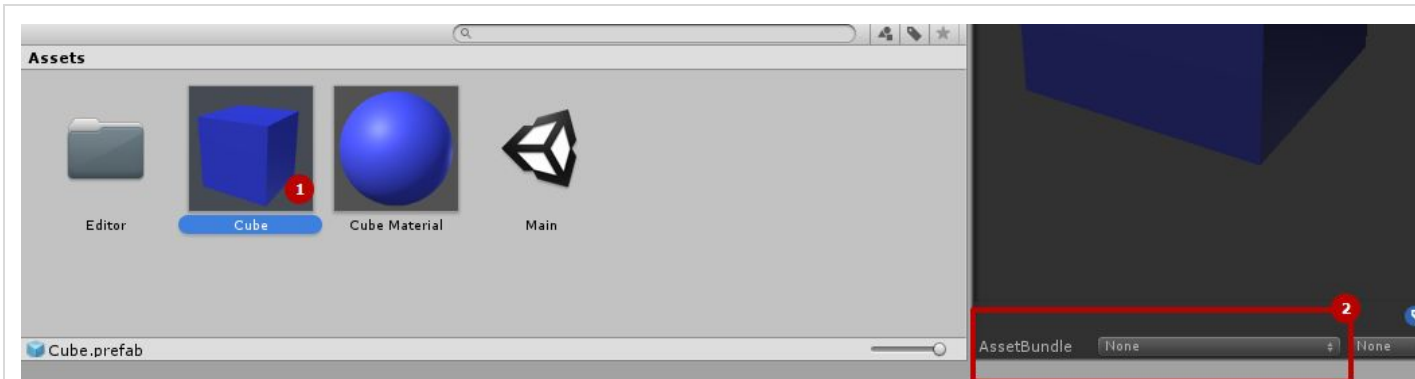


1 - Для того чтобы применить материал к нашему кубу необходимо просто перетащить материал на сам куб.

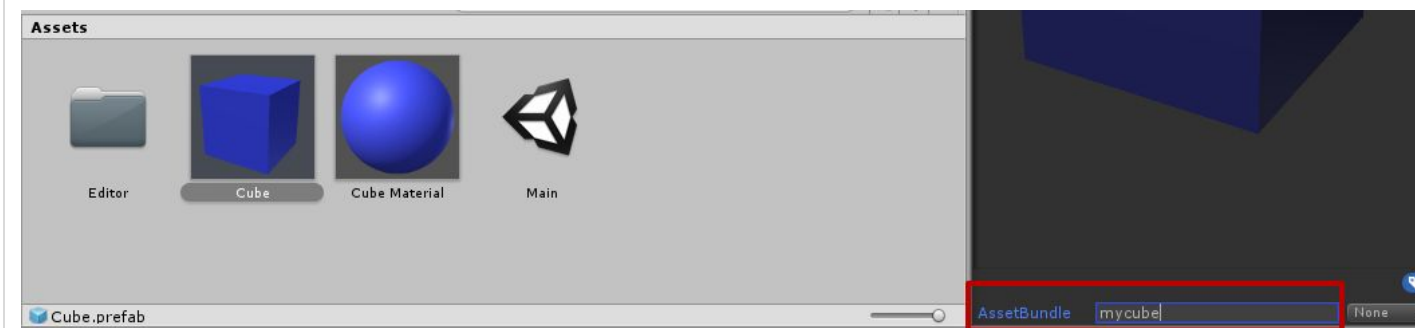
2 - Делаем из кубика префаб (Просто перетянув его из окошка Hierarchy на пустое место окошка Project) Префаб это особый тип ассетов, позволяющий хранить весь GameObject со всеми компонентами и значениями свойств. Префаб выступает в роли шаблона для создания экземпляров хранимого объекта в сцене.

Снова Сохраняем нашу сцену
(CTRL+SHIFT+S)

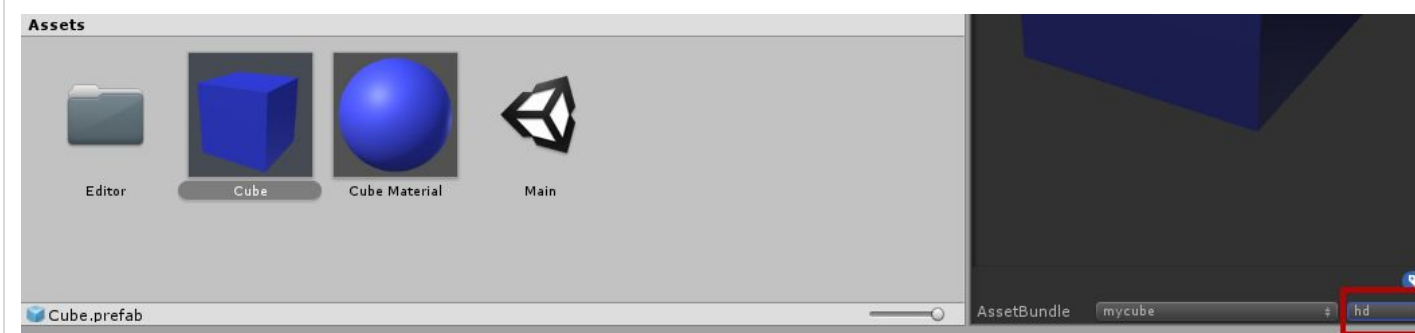
Динамическая подгрузка игрового контента в Unity из облака.



Выберите созданный префаб **"Clube"** и обратите внимание на маленькое AssetBundle меню, справа экрана.

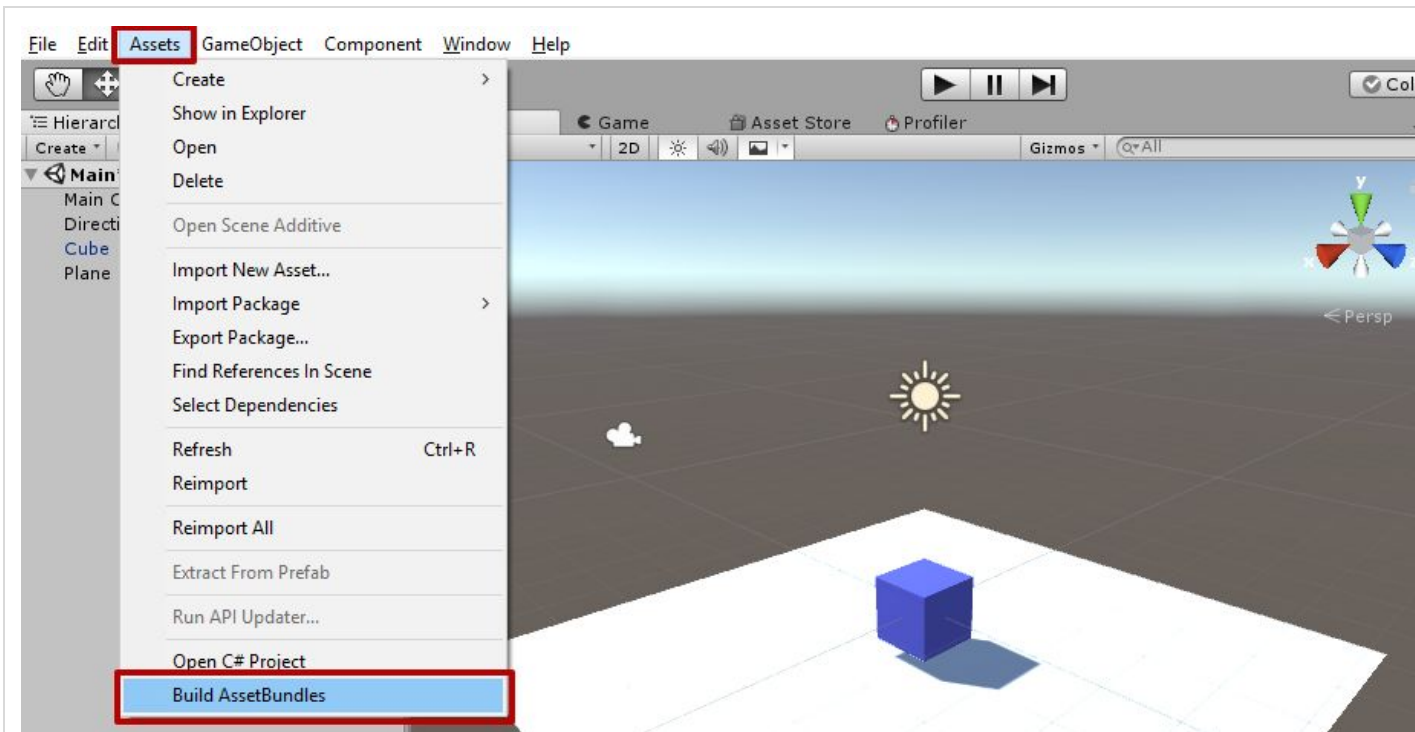


Нажав **New** создадим новое название для нашего ассет бандла, в моем случае я назвал **"mycube"**. По этому названию мы будем в дальнейшем загрузить файл из облака Azure.

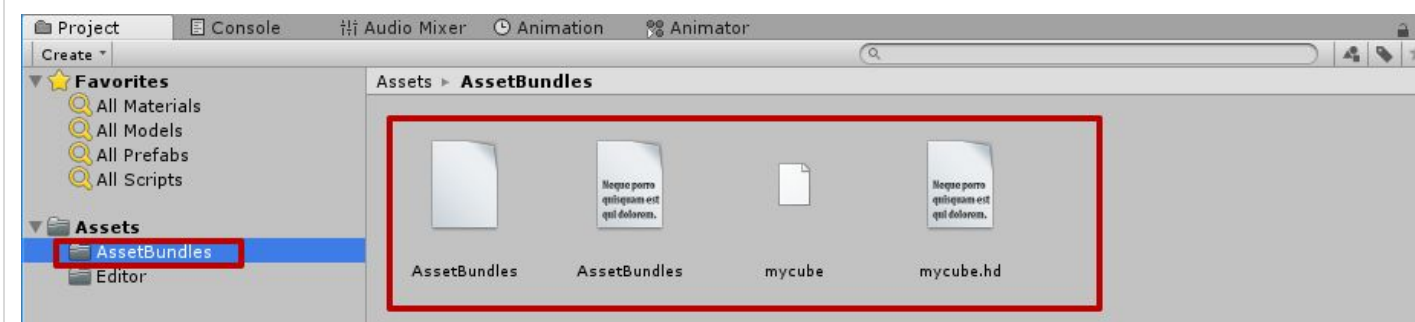


Обратите внимание на поле которое находится возле имени. По умолчанию оно хранит значение **None**. В этом поле создается индекс, в итоге он дописывается к названию бандла. Индекс можно оставлять пустым, но я для примера указал там название **"hd"**.

Динамическая подгрузка игрового контента в Unity из облака.

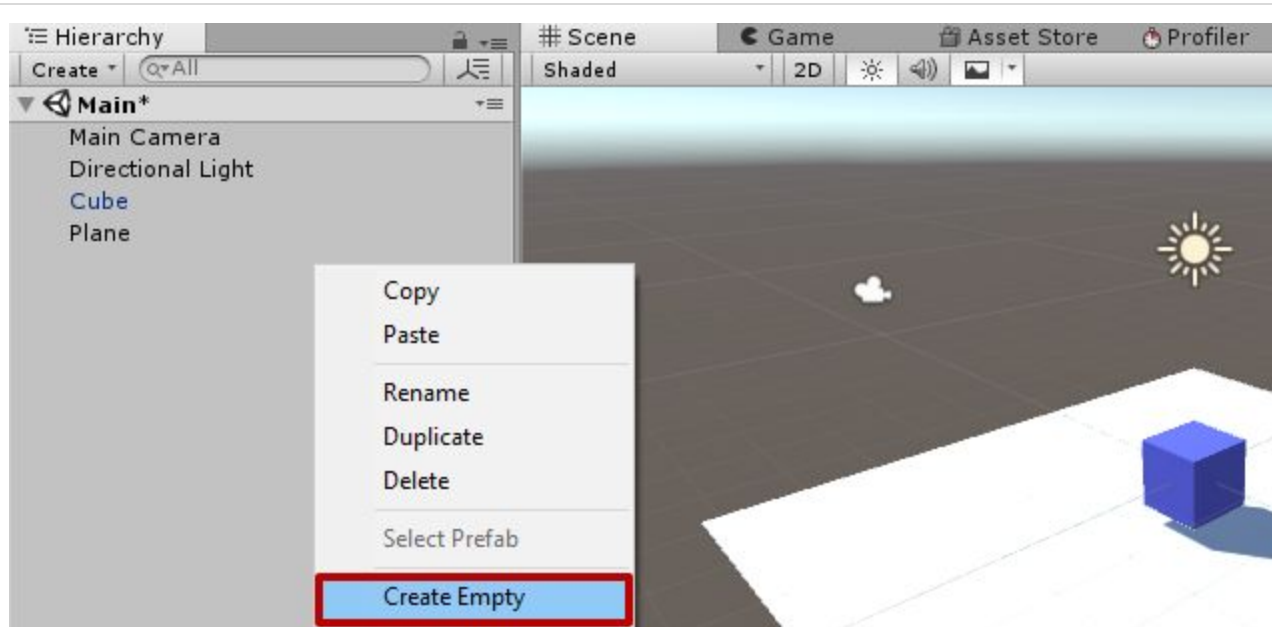


После того как мы дали бандлу название, пришла пора воспользоваться скриптом **BuildAllAssetBundles**, для этого просто нужно зайти в меню **Assets** в верхней панели меню и нажать кнопку **Build AssetBundles**.



После того как процесс создания закончится в папке **AssetBundles** появятся файлы с нашими бандлами. Именно их мы будем загружать в облако Azure.

Динамическая подгрузка игрового контента в Unity из облака.

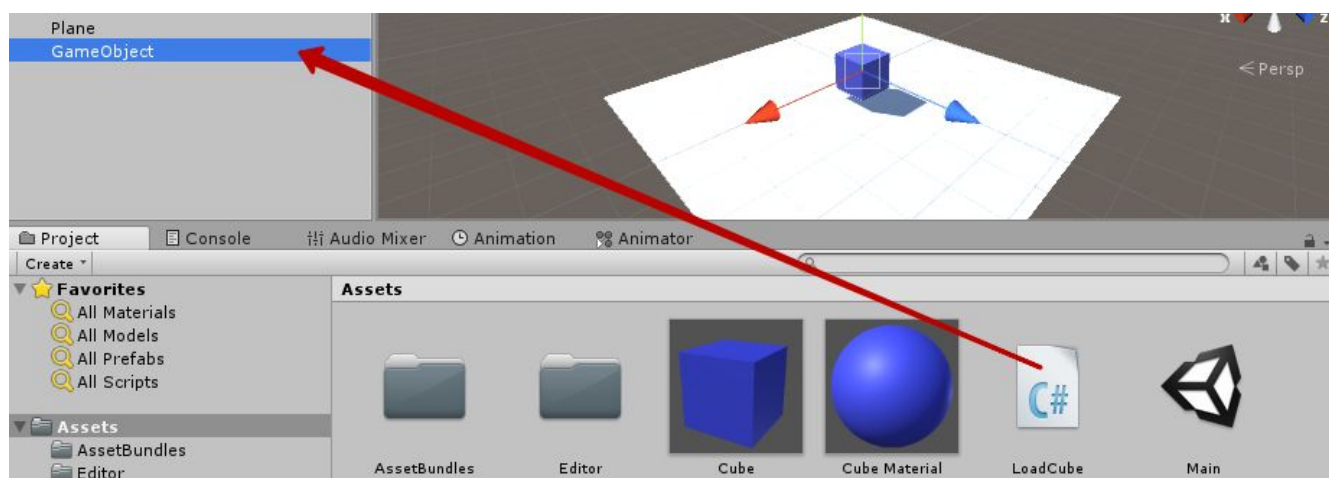


В каждой игре во время процесса разработки бандлы могут сильно меняться, на облако грузятся только чистовые варианты. Это делается для того чтобы сэкономить время.

Мы должны подготовить скрипт который будет отвечать за загрузку и создание объекта.

Для того чтобы это было удобно тестировать начальную загрузку сделаем из папки AssetBundles.

Для начала создаем пустой объект **"GameObject"**. Для этого правой кнопкой мыши нажимаем в пустой области окна **Hierarchy** и выбираем **Create Empty**



Создаем скрипт с названием **"LoadCube"**.

Скрипт помещаем на новосозданный **GameObject**.

Динамическая подгрузка игрового контента в Unity из облака.



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

1 using System.Collections;
2 using UnityEngine;
3 using UnityEngine.Networking;
4
5 public class LoadCube : MonoBehaviour
6 {
7     IEnumerator Start()
8     {
9         string uri = "file:/// " + Application.dataPath + "/AssetBundles/mycube.hd";
10         UnityWebRequest request = UnityWebRequest.GetAssetBundle(uri, 0);
11         yield return request.Send();
12         AssetBundle bundle = DownloadHandlerAssetBundle.GetContent(request);
13         /// Обратите внимание, на имя ассета, оно такое же как имя префаба
14         GameObject cube = bundle.LoadAsset<GameObject>("Cube");
15         Instantiate(cube);
16     }
17 }
18
```

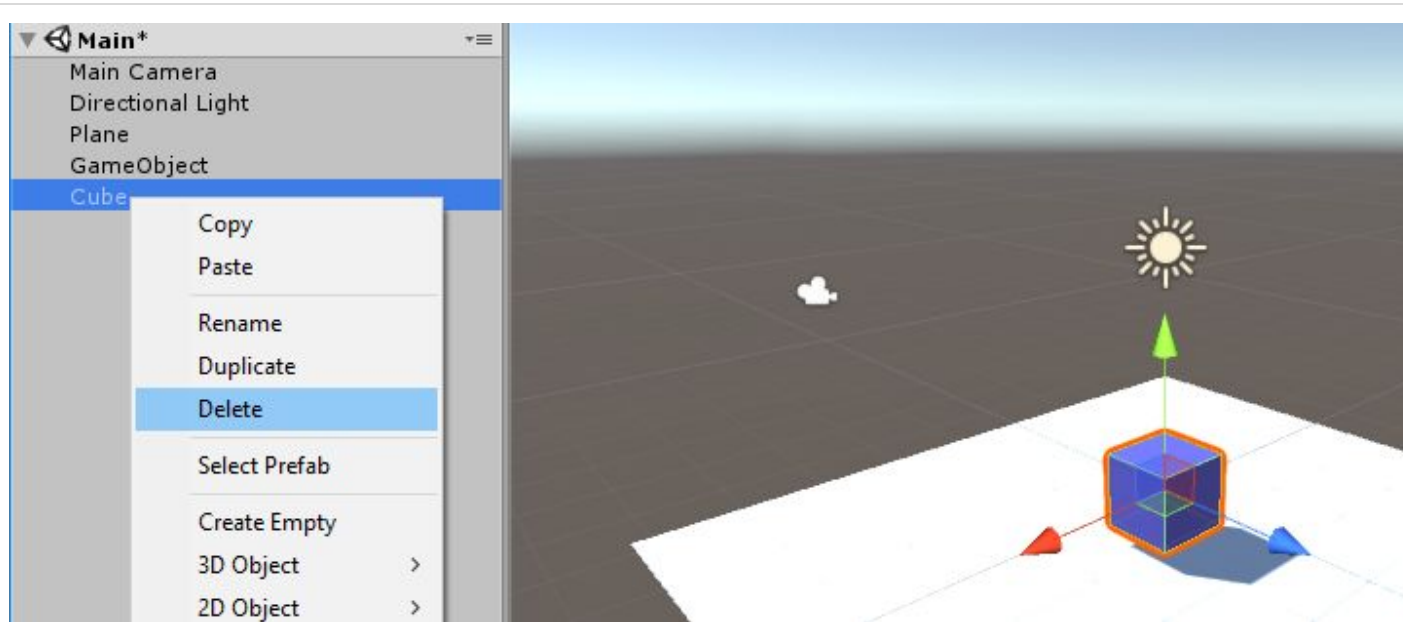
Теперь приступим к написанию логики **LoadCube** скрипта. Для этого откроем скрипт и вставим в него код который находится ниже ниже.

Обратите внимание на название загружаемого бандла и путь к нему. Сейчас он называется mycube.hd важно что через точку к нему добавлен индекс. **В Unity редакторе в имени индекса не видно, он считается как расширение и не отображается в редакторе.**

```
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);
    }
}
```

Динамическая подгрузка игрового контента в Unity из облака.



Давайте теперь удалим наш кубик со сцены, чтобы мы смогли увидеть как появится кубик созданный из ассет бандлов после запуска игры.

Выберите объект Cube в Hierarchy и правой кнопкой мыши укажите Delete.

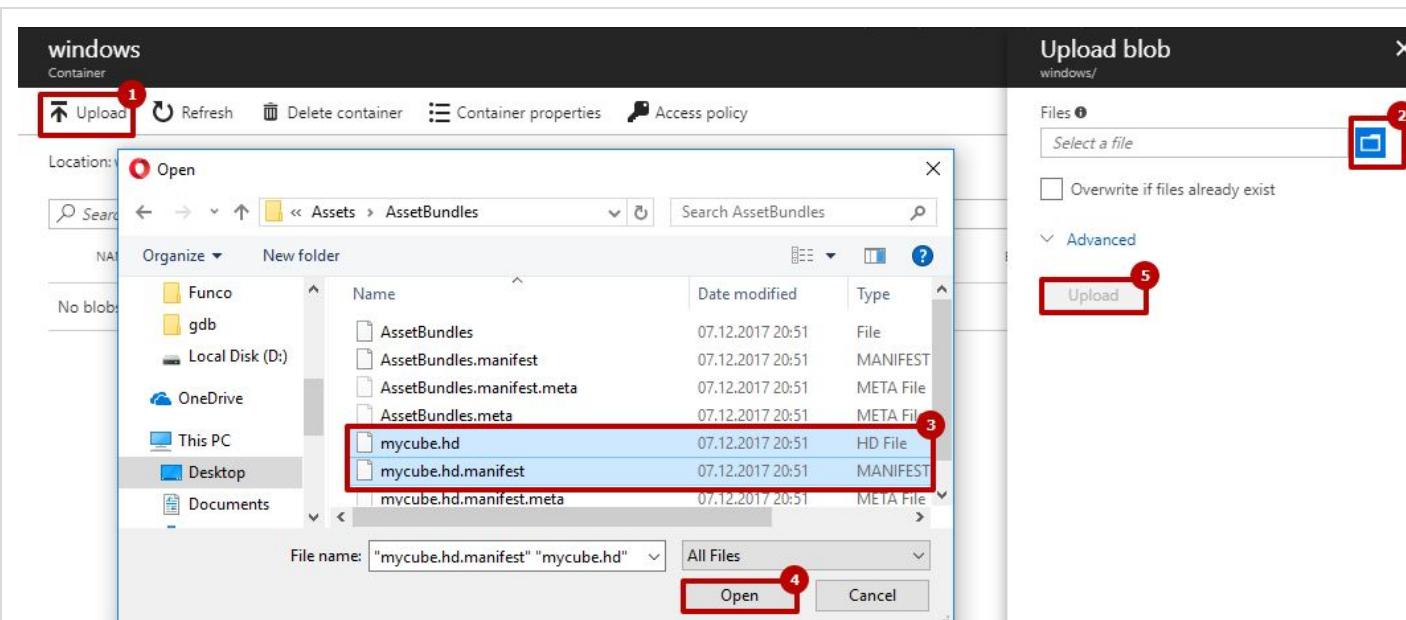
Можем сохранить нашу сцену
CTRL+Shift+S



Запускаем нашу игру нажатием стрелочки в верхней панели.

На пустой сцене должен появиться кубик как на скриншоте.

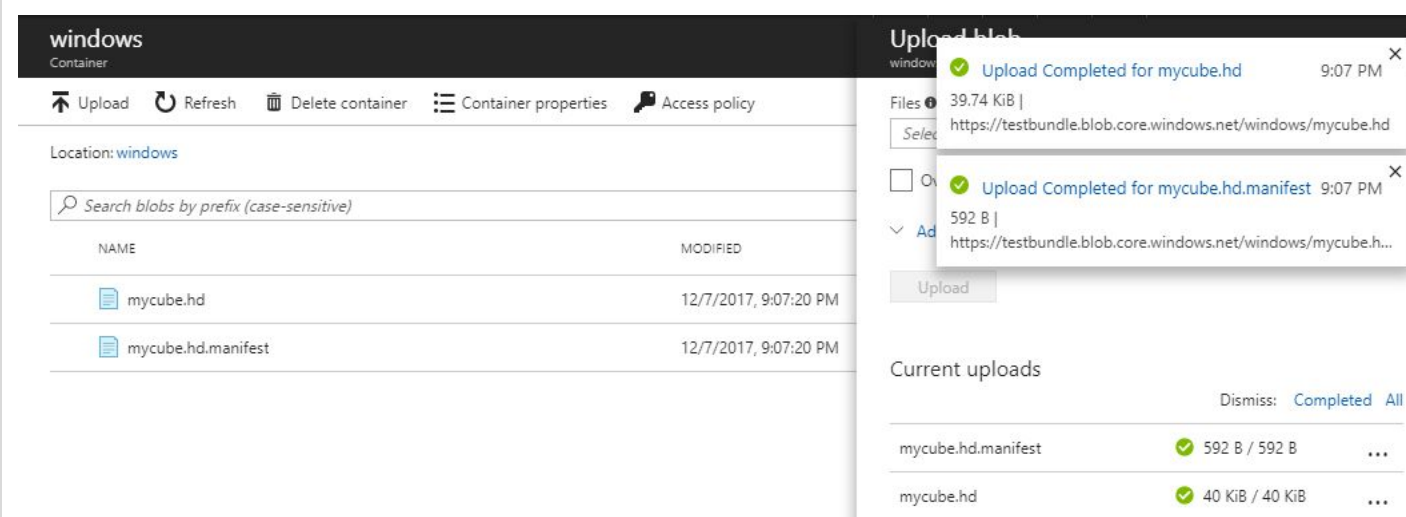
Динамическая подгрузка игрового контента в Unity из облака.



Грузим бандл в облако

Теперь давайте сделаем чтобы этот bundle, грузился из облака Azure а не из локальной директории.

Для начала вернемся к Azure и открыв наш контейнер **"windows"** нажмем кнопку **Upload** затем как на скриншоте ниже выберем наши бандлы (если вы сохраняли проект на рабочем столе вы увидите папку с Unity проектом. Далее **Assets\AssetBundles**) и нажмем кнопку **Upload** в меню справа.



После этого вы увидите как ваши бандлы загрузились в облако.

Динамическая подгрузка игрового контента в Unity из облака.



```
LoadCube.cs* X CreateAssetBundles.cs
Asset Bundle Stan LoadCube Start()
1 using System.Collections;
2 using UnityEngine;
3 using UnityEngine.Networking;
4
5 public class LoadCube : MonoBehaviour
6 {
7     IEnumerator Start()
8     {
9         string uri = "https://testbundle.blob.core.windows.net/windows/mycube.hd";
10         UnityWebRequest request = UnityWebRequest.GetAssetBundle(uri, 0);
11         yield return request.Send();
12         AssetBundle bundle = DownloadHandlerAssetBundle.GetContent(request);
13         /// Обратите внимание, на имя ассета, оно такое же как имя префаба
14         GameObject cube = bundle.LoadAsset<GameObject>("Cube");
15         Instantiate(cube);
16     }
17 }
18
```

Теперь нужно немного отредактировать скрипт загрузки **LoadScript**. А точнее изменить путь к нашему бандлу.


Microsoft Azure **testbundle** Containers > windows

windows
Container

Upload Refresh Delete container Container properties Access policy

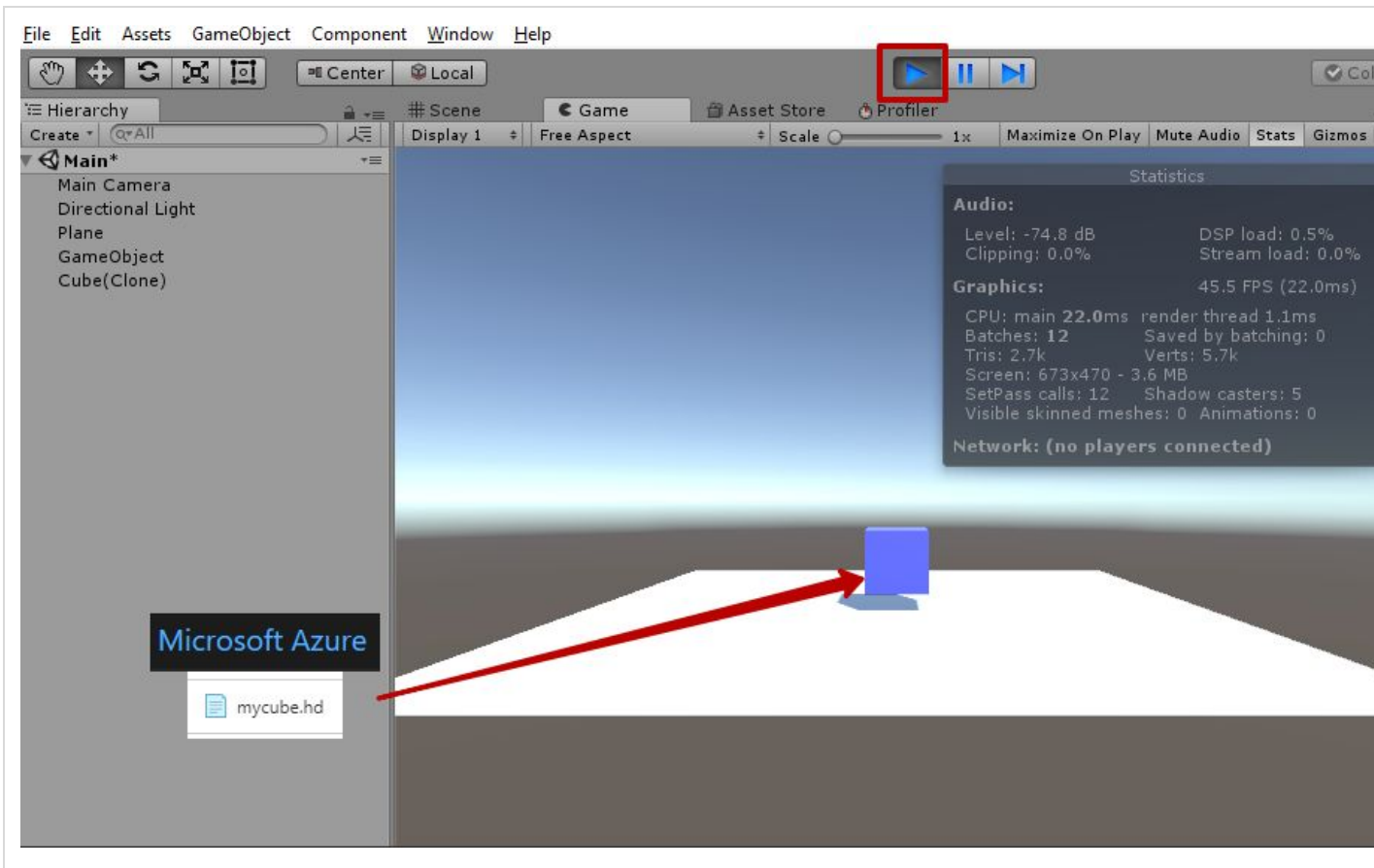
Location: windows

Search blobs by prefix (case-sensitive)

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

Путь составляется так: имя нашего хоста, в моем случае **https://testbundle.blob.core.windows.net** / название хранилища (**windows**) / название самого бандла (**mycube.hd**). На скриншоте выше показан полный путь.

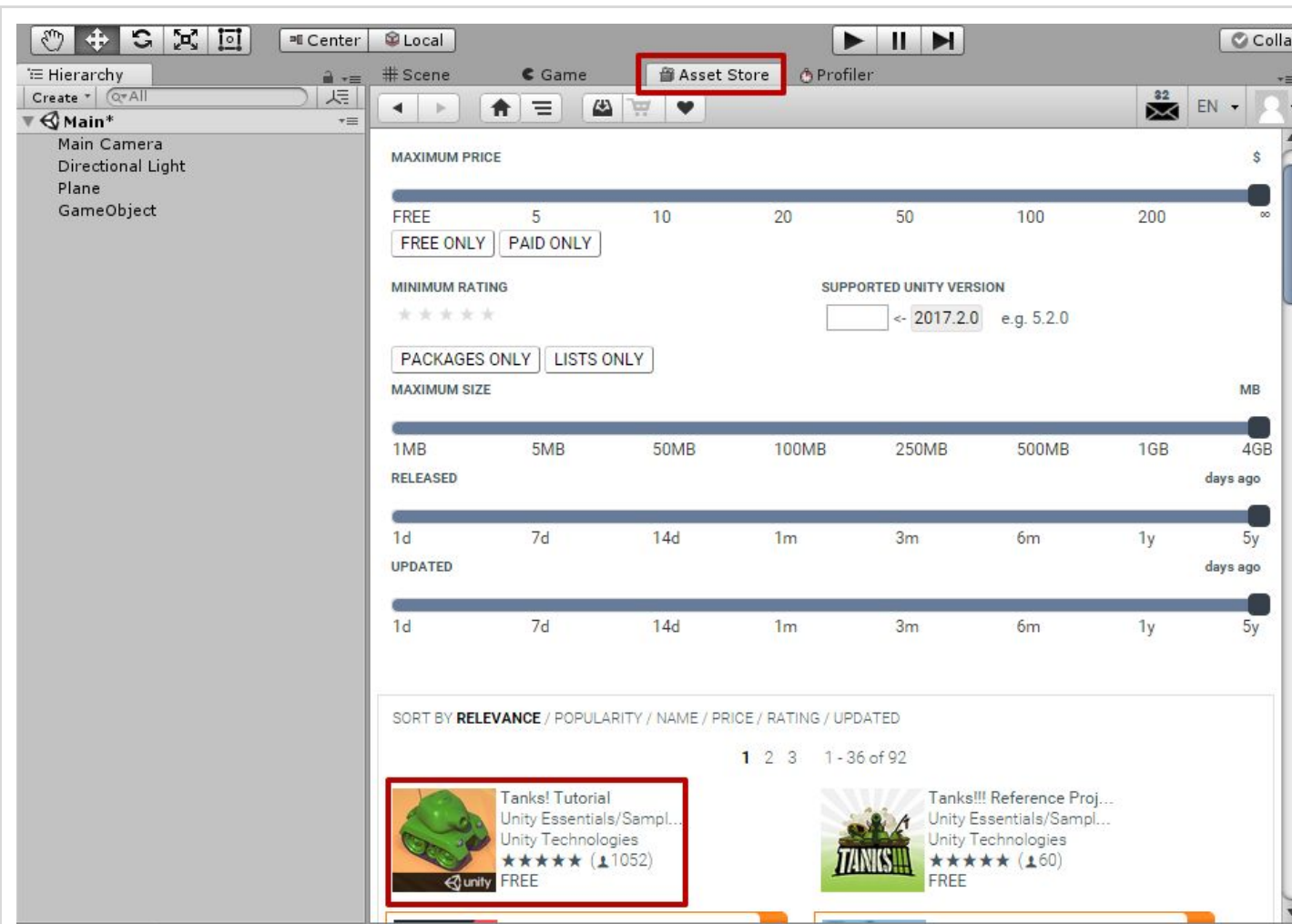
Динамическая подгрузка игрового контента в Unity из облака.



Теперь можно запускать нашу игру и посмотреть как с облака загрузится кубик.

Если кубик загрузился поздравляем, вы смогли подгрузить объект из облака Azure.

Динамическая подгрузка игрового контента в Unity из облака.



Загрузка и настройка игры Tanks

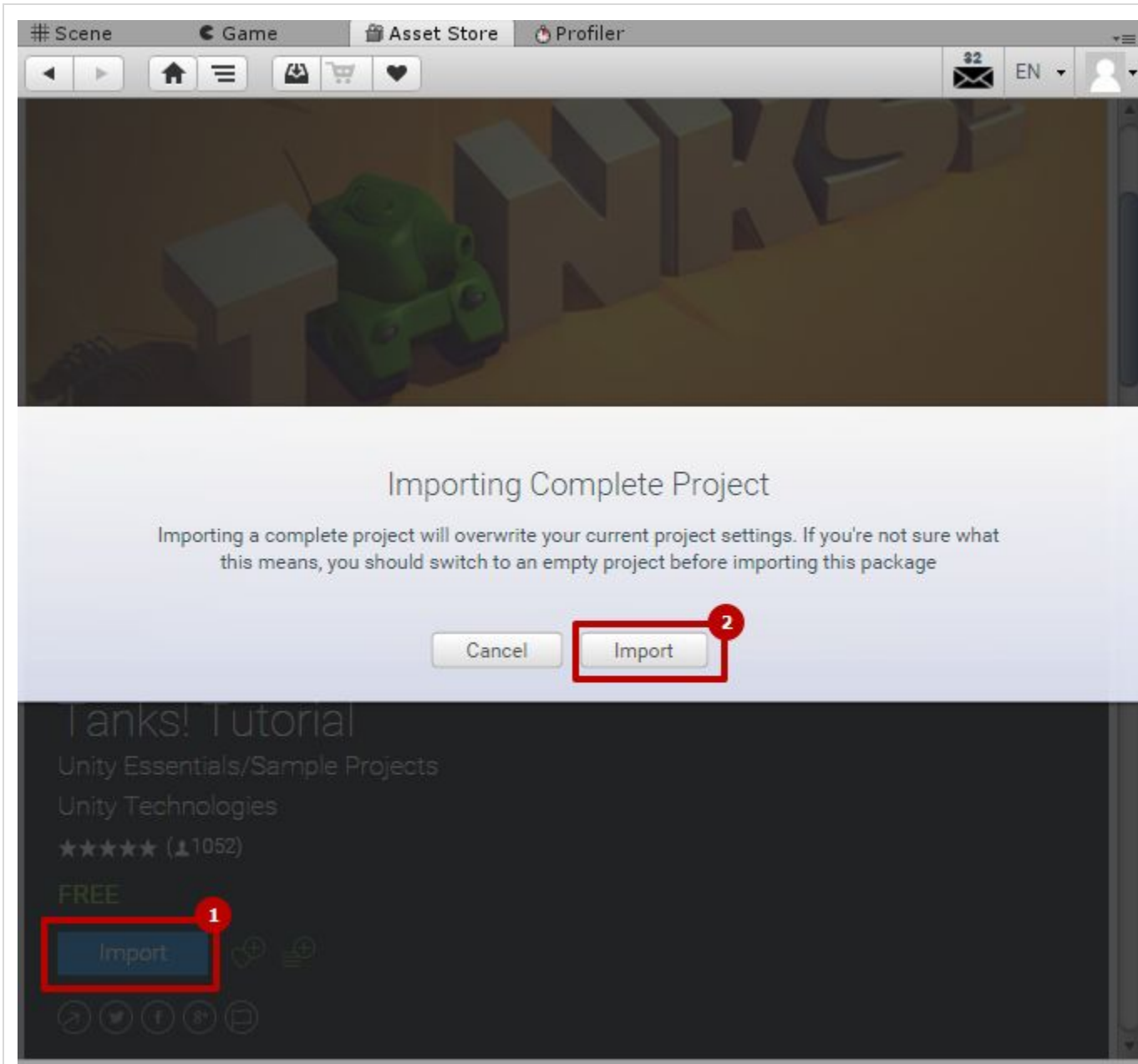
Давайте теперь используем ассет бандлы на примере реальной игры. Отличным примером станет официальный бесплатный tutorial unity под названием **"Tanks! Tutorial"** который мы скачаем с **Unity Asset Store**.

Нажмите на вкладку **"Asset Store"** и введите в строке поиска **"Tanks! Tutorial"**

Система может запросить Unity ID который вы можете создать тут <https://id.unity.com/account/new>

Выберите ассет из списка результатов поиска.

Динамическая подгрузка игрового контента в Unity из облака.



Нажмите Download для загрузки ассета.

Когда ассет скачается то появится кнопка **Import**.

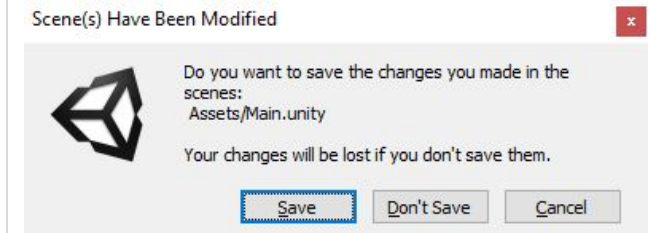
Затем нажмите еще одну кнопку **Import** и подождите когда все загрузится в ваш проект.

Динамическая подгрузка игрового контента в Unity из облака.

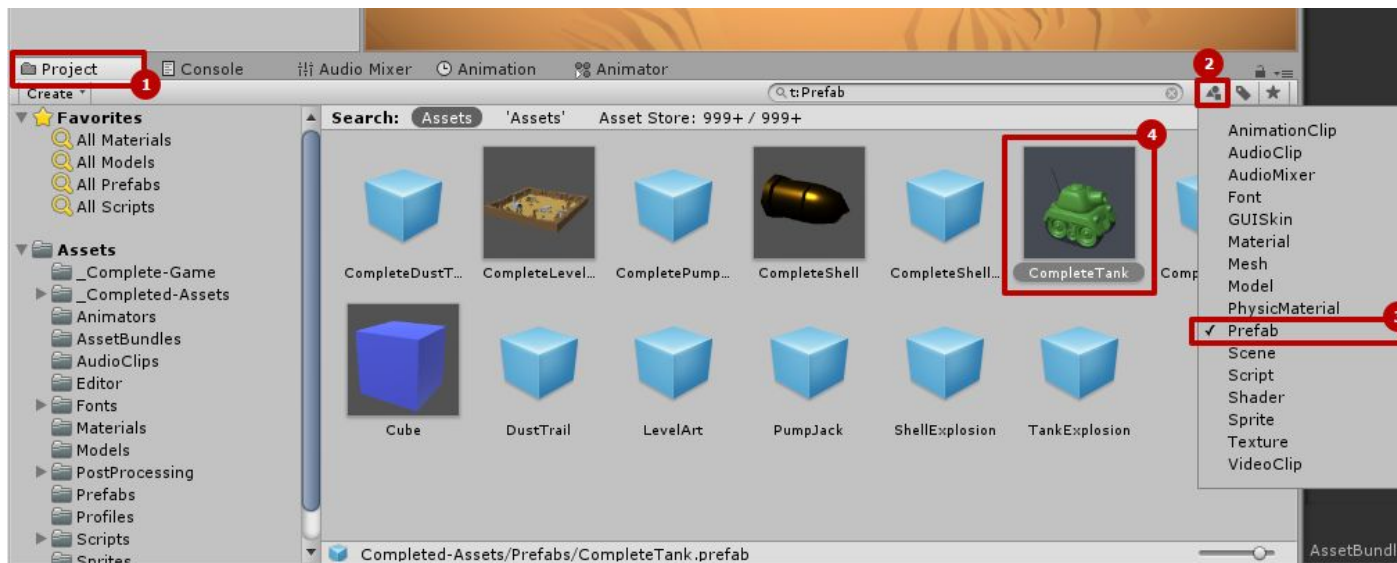


Давайте теперь найдем главную сцену и проверим чтобы все работало. На скриншоте ниже показано где находится главная сцена ("Complete-Game") это корень папки Assets.

Если система запросит диалоговое окно про сохранение сцены нажмите **Save**



Динамическая подгрузка игрового контента в Unity из облака.



Ищем префаб танка чтобы сделать из него bundle.

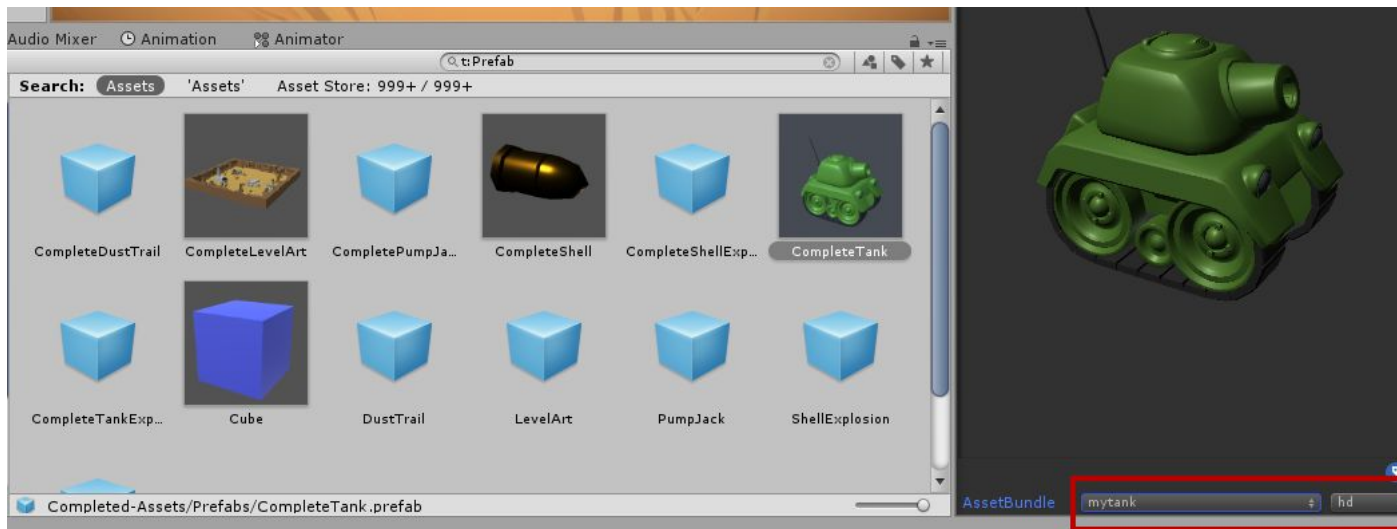
Делать bundle мы будем из префаба танка. Игрок когда скачает нашу игру, получит ее без танков, а сами танки мы будем грузить из облака Azure а потом создавать их на сцене.

Для того чтобы найти префаб танка нам необходимо

1 - выбрать окно **Project**

2-3 - Указать в фильтрах объектов **Prefab**

4 - Найти **CompleteTank** префаб.



Теперь нам необходимо сделать bundle из префаба, по тому же принципу что и кубик.

Прописываем нашему префабу AssetBundle **mytank** и в качестве типа **hd**.

Динамическая подгрузка игрового контента в Unity из облака.



Upload blob

Files

Select a file

Overwrite if files already exist

Advanced

Upload

Open

Assets > AssetBundles

Search AssetBundles

Organize New folder

This PC Desktop Documents Downloads Music Pictures Videos OS (C:)

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

File name: "mytank.hd.manifest" "mytank.hd"

All Files

Open Cancel

Теперь загрузим новый bundle нашего танка в Azure.

Upload blob

NAME

mytank.hd

URL

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

LAST MODIFIED

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

Для простого получения пути нашего бандла, можно просто нажать на файл который загружен и скопировать путь из меню справа, поле URL

Динамическая подгрузка игрового контента в Unity из облака.



```
GameManager.cs LoadCube.cs CreateAssetBundles.cs
Asset Bundle Stan Complete.GameManager m_NumRoundsToWin

1 using System.Collections;
2 using UnityEngine;
3 using UnityEngine.SceneManagement;
4 using UnityEngine.UI;
5 using UnityEngine.Networking;

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

27 private IEnumerator Start()
28 {
29     // Create the delays so they only have to be made once.
30     m_StartWait = new WaitForSeconds (m_StartDelay);
31     m_EndWait = new WaitForSeconds (m_EndDelay);
32
33     string uri = "https://mytestbundle.blob.core.windows.net/windows/mytank.hd";
34     UnityWebRequest request = UnityWebRequest.GetAssetBundle(uri, 0);
35     yield return request.Send();
36     AssetBundle bundle = DownloadHandlerAssetBundle.GetContent(request);
37     m_TankPrefab = bundle.LoadAsset<GameObject>("CompleteTank");
38
39     SpawnAllTanks();
40     SetCameraTargets();
41
42     // Once the tanks have been created and the camera is using them as targets, start the game.
43     StartCoroutine (GameLoop ());
44 }
45
46
```

Осталось самое интересное, теперь пишем код загрузки нашего танка.

Мы будем редактировать скрипт Assets/Scripts/Managers/GameManager.cs

Первое что мы делаем это поле **m_TankPrefab** должно быть **private**, так как теперь нам не нужна ссылка префаба из редактора. (по сути мы делаем невозможным отображение танка без использования Azure cloud)

Подключаем пространство имен **using UnityEngine.Networking;**

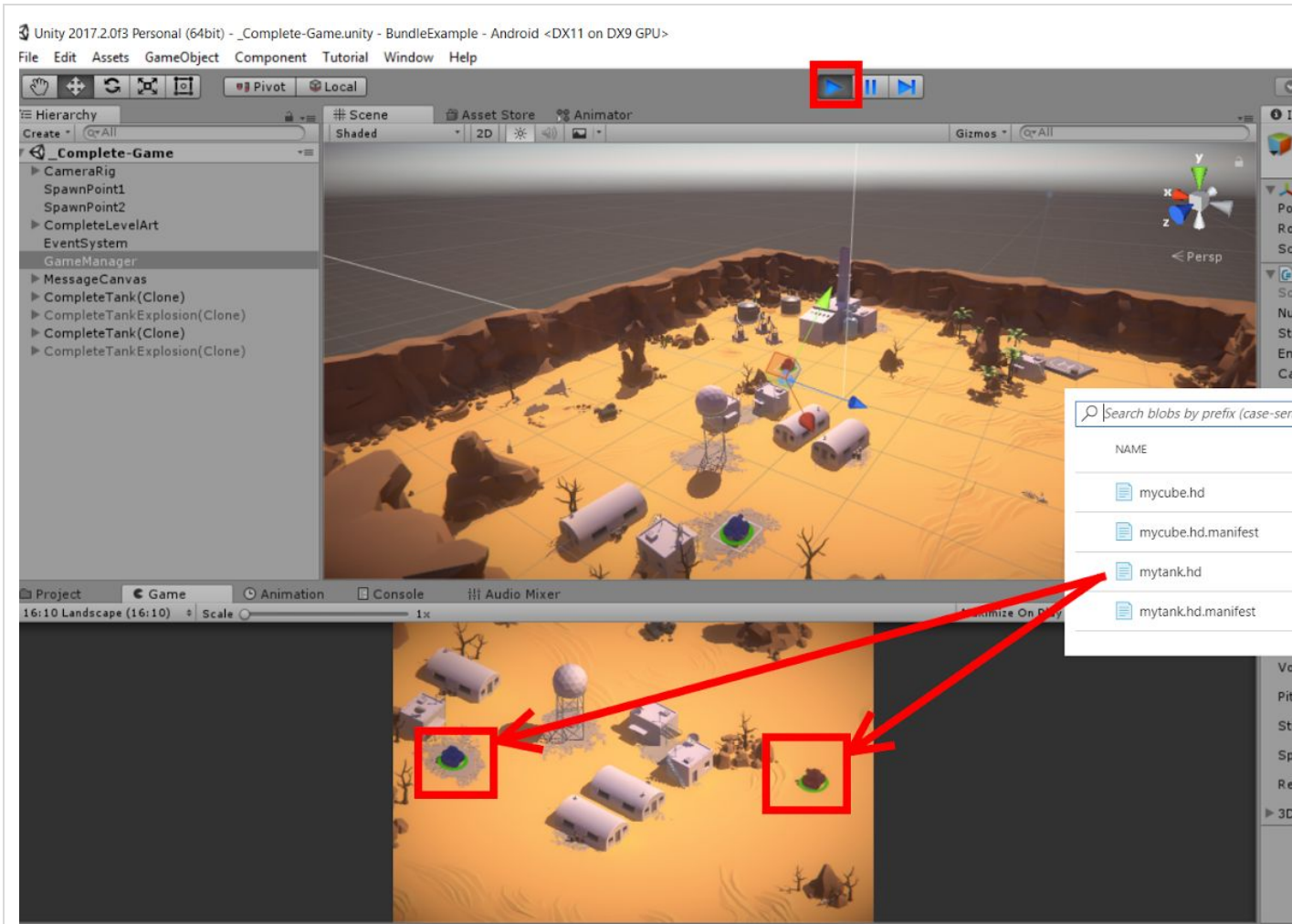
Вместо главного метода Start Void делаем **Start IEnumerator.**

Теперь можно поместить кусок кода для загрузки bundle танка из облака Azure.

Код загрузки bundle танка:

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

Динамическая подгрузка игрового контента в Unity из облака.



Отличная работа, теперь наши танки подгружаются на сцену прямо из облака Azure.

<https://github.com/rio900/unityazurebundle>