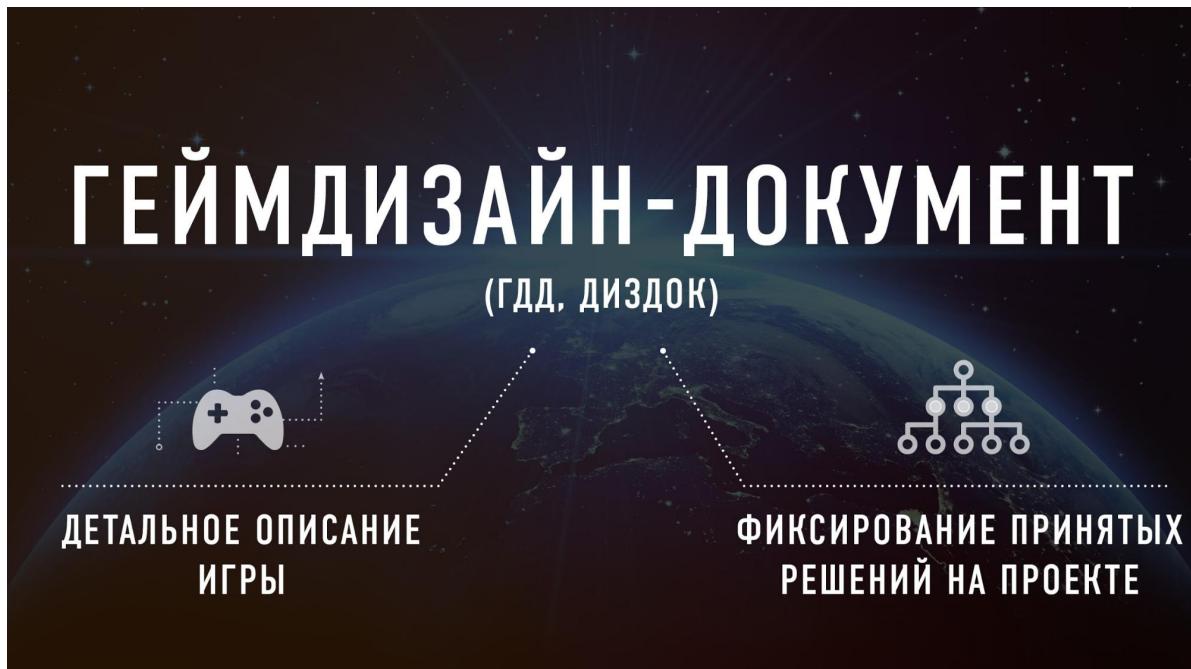


## Тема 12. Оформлення гри. Ігрова документація

Ігрова документація (ІД) – це повний опис гри з постійною фіксацією прийнятих на проекті рішень. Це документ, з якого учасник команди може отримати будь-яку потрібну інформацію по проекту.



Документація має кілька типів. Це може бути як повний опис проекту, що включає і маркетингові аспекти, так і опис однієї певної фічі. У загальному випадку гейм-дизайнерський документ може містити наступну інформацію:

- опис гри (ЦА, платформи, жанр);
- схема геймплея;
- опис механік і інтерфейсу;
- використовувані в розробці технології;
- вимоги до графіку / арт-дизайну / звуку;
- сюжет ;
- замітки гейм-дизайнера.

ІД вирішує не тільки завдання інформування команди про стан гри. Вона потрібна для грамотної та ефективної постановки завдань виконавцям

(розробникам, художникам, аніматорам). Таким чином, документація як свого роду лакмусовий папірець дозволяє відстежувати прогрес, оцінювати якість роботи і краще планувати роботу. При наявності документації, де все описано в деталях, простіше правильно оцінити витрати на розробку тієї чи іншої фічі. Крім того, саме ІД можна вважати результатом роботи гейм-дизайнера, кінцевою точкою, яка відправляє чітко сформульовану ідею на реалізацію.

Документація пиється для спрощення всіх процесів всередині команди. Так чи інакше до неї будуть звертатися представники майже всіх відділів розробки та підтримки проекту:

- гейм-дизайнер;
- продюсер;
- QA;
- програміст;
- UI-дизайнер;
- сценарист;
- художник;
- маркетолог / ком'юніті-менеджер;
- новий співробітник компанії.

У кожному випадку людина отримує з документа потрібну їй інформацію. І саме від гейм-дизайнера залежить її якість і цілісність.

Звичайно ж, дизайн-документ - не панацея. Але він значно економить як ваше час, так і час ваших колег, і, як підсумок, підвищує якість кінцевого продукту, виключаючи ефект «зламаного телефону».

Існує три помилки, які можуть звести нанівець всю користь від складання ІД:

### **Неоднозначність / незрозумілість**

Тут все просто - всі ваші тези, ідеї і результати обчислень повинні бути викладені однозначно. Не дозволяйте людям додумувати за вас, не пропонуйте кілька варіантів виконання завдання на вибір, не сподівайтесь,

що колеги вміють читати ваші думки і не плутайте колег складними літературними оборотами.

### **Неграмотність**

Читання документації сильно ускладнюється, якщо в тексті повно проблем з орфографією, граматикою і пунктуацією. Подбайте про це.

### **Інформація не по суті**

У творчому процесі гейм-дизайнеру досить легко дозволити собі ліричний відступ, коли справа стосується улюбленого персонажа, механіки або предмета. Пам'ятайте про «кістяк» - ті речі, які обов'язково повинні бути вказані в ІД (наприклад, якщо ми говоримо про юніт - дальність атаки, опис бойової механіки і т.д). І нічого більше. Всі нові ідеї та доповнення можна залишити для заміток. В основному документі їх бути не повинно.

Ще варто відзначити, що завжди треба пам'ятати, для кого ви пишете ІД. Наприклад, художниками найчастіше не потрібні точні формули розрахунку тих чи інших показників, а для програмістів вони необхідні. Для UI-дизайнерів дуже важливі схеми інтерфейсів, список необхідних елементів і переходів, в той час як майже для всіх інших відділів ця інформація буде зайвою.

## **ВИБІР ІНСТРУМЕНТАРІЮ**

Існує безліч інструментів для роботи з документацією. З них можна виділити групу wiki-движків:

- MediaWiki ;
- DokuWiki ;
- Markdoc ;
- Sphinx .

Розрізняються вони використовуваними мовами, функціональністю «з коробки» (підтримка стилів, акаунти з різними правами, заливанням і типом зберігання файлів і тому подібним), розширеністю.

Крім wiki-движків використовуються наступні системи:

- Google Docs (або аналоги - наприклад, офісний пакет iCloud);
- Evernote;
- GitHub.

Також можна користуватися звичайними офлайн-редакторами з підключеною системою контролю версій або шаринга. У цьому випадку сам документ пишеться, наприклад, в Word, а потім завантажується в хмару, щоб з ним могли працювати інші. Але це тягне за собою ряд незручностей - наприклад, відсутність нормального пошуку по всій документації та повноцінного контролю версій.

Доцільним також є використання зв'язки Confluence і JIRA. Перший сервіс - інструмент для ведення документації, другий - відомий таск-менеджер. Обидва дуже добре доповнюють один одного - наприклад, до статті в Confluence можна інтегрувати завдання з JIRA, відстежувати їх статус. При використанні інших систем можуть виникнути складності при інтеграції таск-трекера з документацією.

Основний плюс зв'язки переплетений з основним мінусом. Справа в тому, що це дуже складна в налаштуванні система (існує навіть окремий тип фахівців, які впроваджують Confluence в великих організацій). Але все це окупается вражуючою продуманістю і гнучкістю - при бажанні Confluence можна налаштувати практично під будь-які потреби. Так що тут все залежить від розміру вашої команди і проектних завдань.

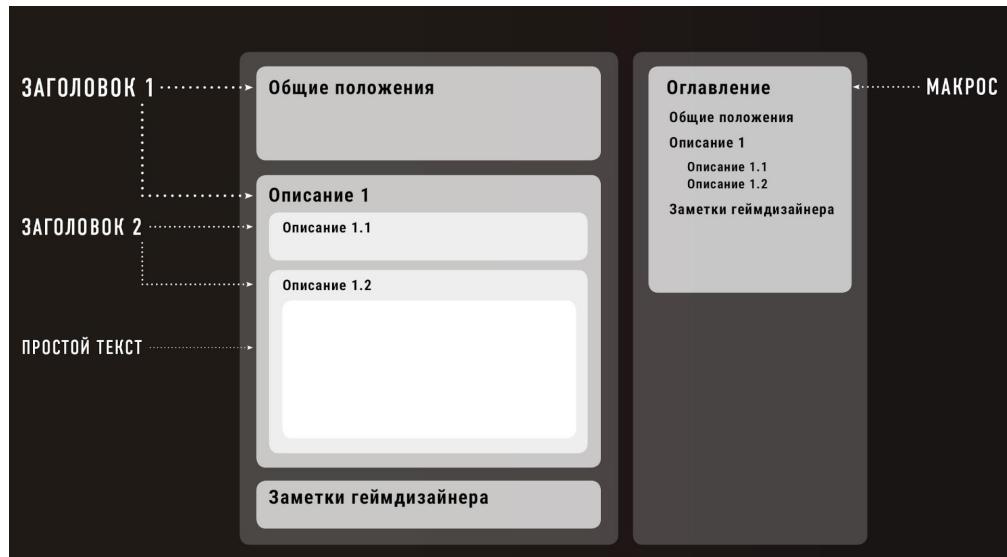
## **ПОРАДИ З ПРАКТИКИ**

**1).** Зберігайте актуальність і повноту інформації. Застаріла або недороблена ІД - майже так само погана, як і її відсутність. Щоб розуміти проект і правильно його розвивати, потрібно постійно бачити загальну, правдиву, деталізовану картину.

**2).** Для оформлення документів найкраще створити окремий невеликий регламент, за яким всі будуть розуміти, як потрібно оформляти ІД. Сюди - затверджені шрифти, тонкощі щодо форматування, що в документі означає

курсив, що потрібно виділяти «жирним» і так далі. В майбутньому це прискорить навігацію по ІД і спростить комунікацію в команді.

3). Варто розробити єдиний для вашої команди стиль оформлення ІД і його структуру. Нижче наведено спрощений варіант структури документації:



4). Працюючи з конфігураціями, пропишіть в ІД ті механізми і структури, які ви в майбутньому захочете варіювати під час налаштування механік. Якщо не зробити це заздалегідь, великий шанс, що потім доведеться просити програмістів «витягувати» з коду те, що можна було відразу вивести на поверхню. Ось приклад однієї зі сторінок документації, де наведено шматок коду, з яким може бути потрібно мати справу в майбутньому:

Примеры лотов

Лоты задаются вручную

**Лот без скидки** » Развернуть исходный код

**Лот со скидкой** ▼ Свернуть исходный код

```
<lot>
    <lot_quality quality = "2"/>
    <lot_available>
        <city_level_range>
            <comparisons>
                <greater_than_or_equal value="10" />
                <less_than value="20" />
            </comparisons>
        </city_level_range>
    </lot_available>
    <lot_price>
        <resource data="diamonds" count="15" />
    </lot_price>
    <lot_price_original>
        <resource data="diamonds" count="20" />
    </lot_price_original>
    <reward>
        <vip_pack data="vip_timer_30_days" count="1" />
    </reward>
</lot>
```

**Параметры лота**

**lot\_quality** - качество лота, 1-4 (фон подложки, см уровни качества лота)

**lot\_available** - требование лота, может содержать список

**lot\_price** - актуальная цена лота, может быть представлена только одним типом валюта

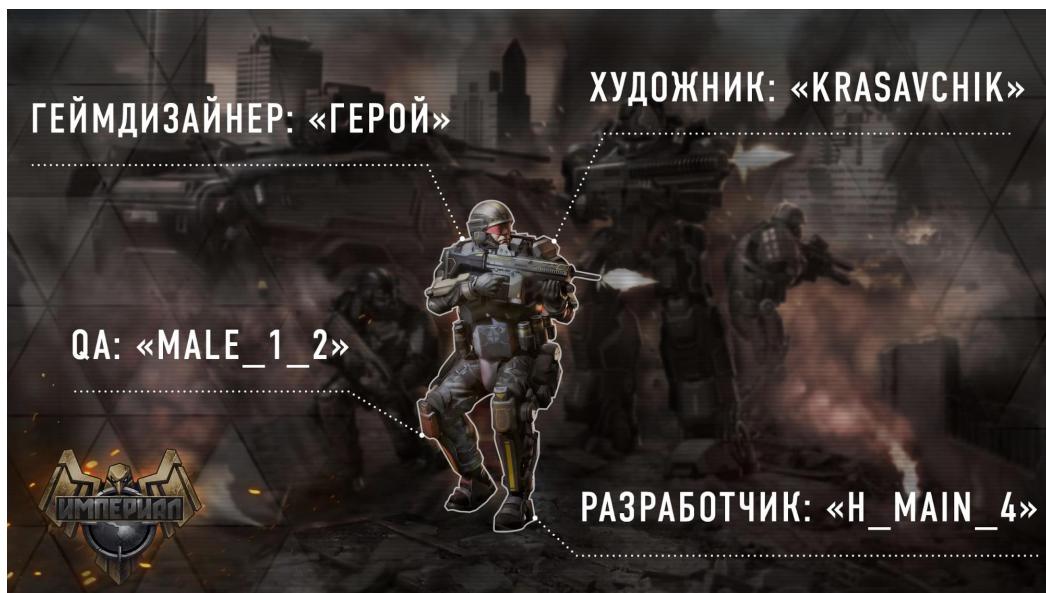
**lot\_price\_original** - «базовая» цена лота, необязательное поле если содержит значение не равное 0, лот будет показываться, как лот со скидкой. Может быть представлена только одним типом валюты, должно совпадать с типом валюты в **lot\_price**.

**reward** - товар лота, может быть представлена только одним типом награды. Количество может быть больше 1.

**5).** У роботі з таблицями рекомендується прибирати під кат згруповані елементи таблиці, а також виділяти шапки (щоб завжди було ясно, який тип даних записаний в стовпці / рядку) і грамотно структурувати дані. Тут як з кодом - якщо з вашими даними працюватиме інша людина, подбайте, щоб він не загруз там в нетрях. Ну і не варто користуватися вкладеними таблицями - це зайві ускладнення.

**6).** Прописуючи формули, обов'язково наводьте пояснення (легенду) - за що відповідає кожна змінна, і як вона взаємодіє з іншими елементами формули. Надалі, якщо ви будете використовувати одну змінну в різних точках ІД (а таке буде), це сильно спростить роботу.

**7).** Дотримуйтесь єдності іменування. Домовтеся і зафіксуйте в документації, як в команді будуть називатися юніти, предмети та інше. Нікому не хочеться напередодні збирання патча дізнатися, що один і той же юніт у чотирьох співробітників називається по-різному:



### **Ключові етапи розробки ІД:**

1) Для початку визначте мету своєї гри, ті враження, що ви хочете, щоб користувач отримав. На практиці це означає, що ваша ІД повинна позначити загальний процес гри, і що користувач зможе робити. Не потрібно описувати на цьому етапі всі дрібні деталі. Поки досить того, що ви знаєте, чого намагаєтесь досягти, і як кожен елемент і фіча впишуться в гру. Ви зможете

повернутися до деталей пізніше, вже не ризикуючи надмірно ускладнити ігровий процес або випустити з уваги мету гри.

2) Той, хто читає вашу ІД, повинен мати можливість швидко і легко зрозуміти, про що гра, і що в ній повинен робити користувач.

3) Нехай в документі буде достатньо місця для тих деталей, які можна поліпшити, доповнити або навіть змінити в міру необхідності. ІД повинна бути такою, щоб при цьому її не доводилося переписувати цілком.

У першому наближенні ваша ІД повинна включати наступне:

### **Загальний опис гри**

Те, що називається концепт-документ або медіа-китом. Обсяг - близько 2 сторінок. Включає в себе саммарі за проектом, короткий синопсис, дуже стислий опис сюжетної лінії гри, і, що найважливіше, список найбільш значущих фіч проекту.

### **Опис «ядра» гри**

Ця частина обмежена ігровим циклом, який демонструє ключові фічі гри і те, як користувач буде проходити через них. Цикл повинен описувати кожну включенну в неї фічу і пояснювати, для чого вона потрібна. Часто буває корисним включити в зображення циклу фейковий скріншот гри, оскільки це дозволить наочно продемонструвати те, до чого ви прагнете, а також те, як користувач бачитиме гру і з нею взаємодіяти.

### **Елементи гри**

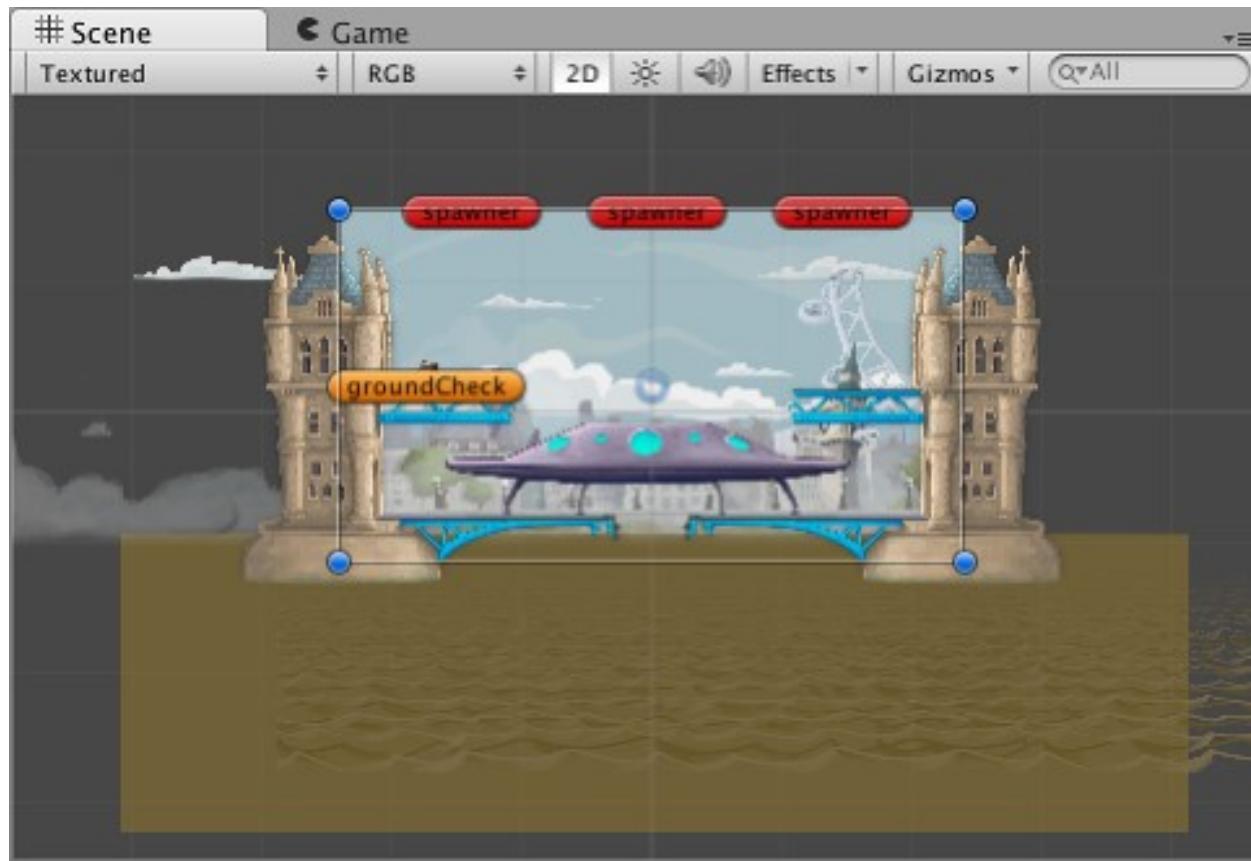
Це секція, в якій будуть детально описані всі фічі, і ось тут-то і потрібно буде додавати деталі. Вам, швидше за все, знадобиться включити таблиці, які детально опишуть фічі і інші елементи, такі, як модифікації гри, її розвиток, GUI, монетизацію, оновлюваний контент, соціальні фічі, ігровий світ, те, як змінюється ігрове меню в процесі, і будь-які інші додаткові деталі.

Якщо ви працюєте над тестовим варіантом гри, то необов'язково включати сюди абсолютно все, але потрібно буде неодмінно згадати фічі, які ви точно хочете бачити в грі, або ті, які додадуться в процесі пост-релізу.

Ця частина в результаті вийде дуже схожою на традиційний ІД. Її можна збільшувати і доповнювати в ході роботи над проектом. Усередині цієї секції будуть обов'язково відбуватися зміни; на цьому етапі вони цілком виправдані. Важливо пам'ятати, що загальний опис гри і опис «ядра» гри після остаточного затвердження повинні залишатися в основному недоторканими, оскільки зміни в них спричинять за собою фундаментальні зміни в зовнішньому вигляді гри. Вносячи поправки, спершу ретельно їх обдумайте.



# Тема 9. Основи роботи з 2D іграми



<https://docs.unity3d.com/ru/current/Manual/Overview2D.html>

# Створення 2D проекту

Unity 5.0.0f4 X

Projects    Get started

New Unity Project 2

Project name\*

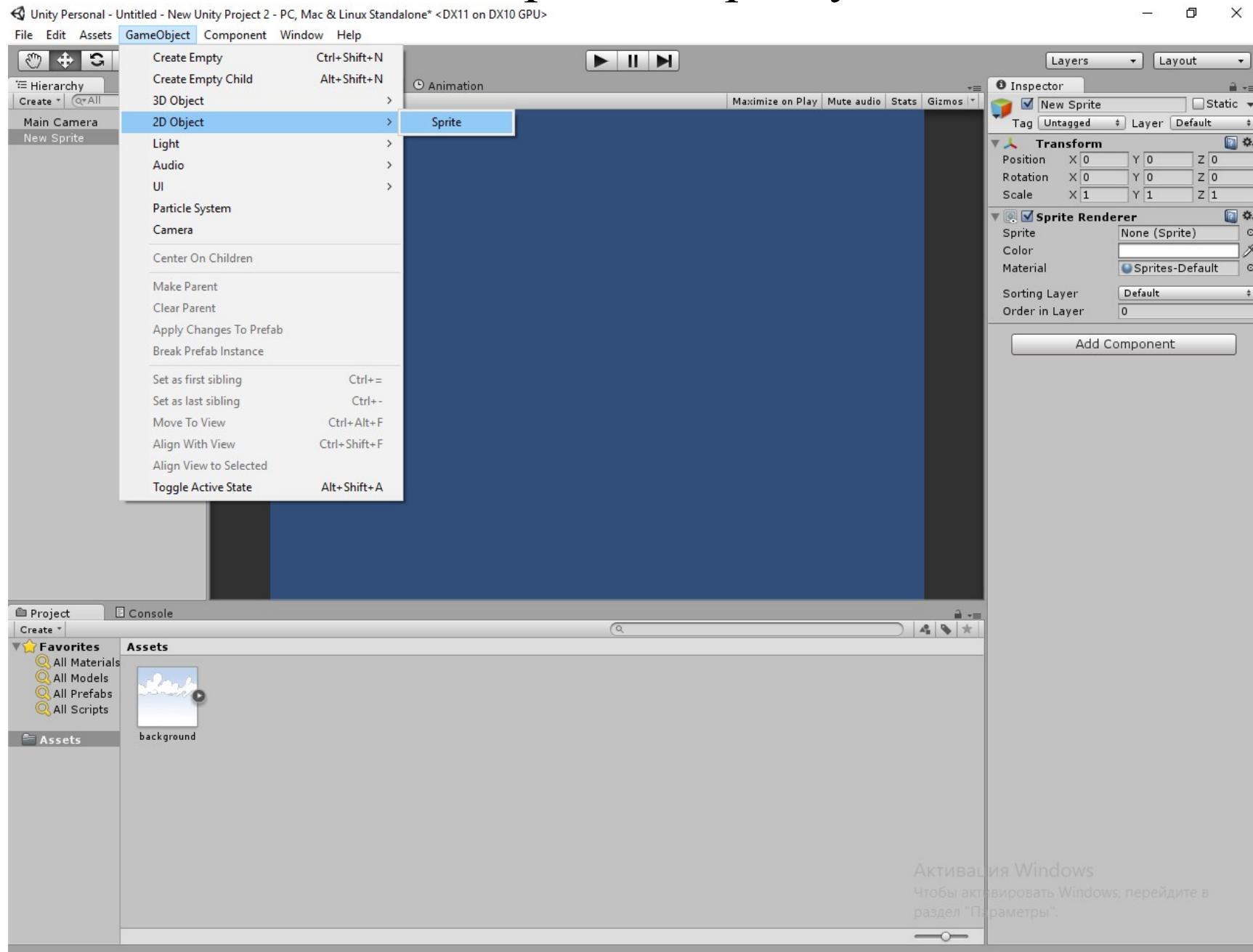
C:\Users\Roman\Documents ...

Location\*

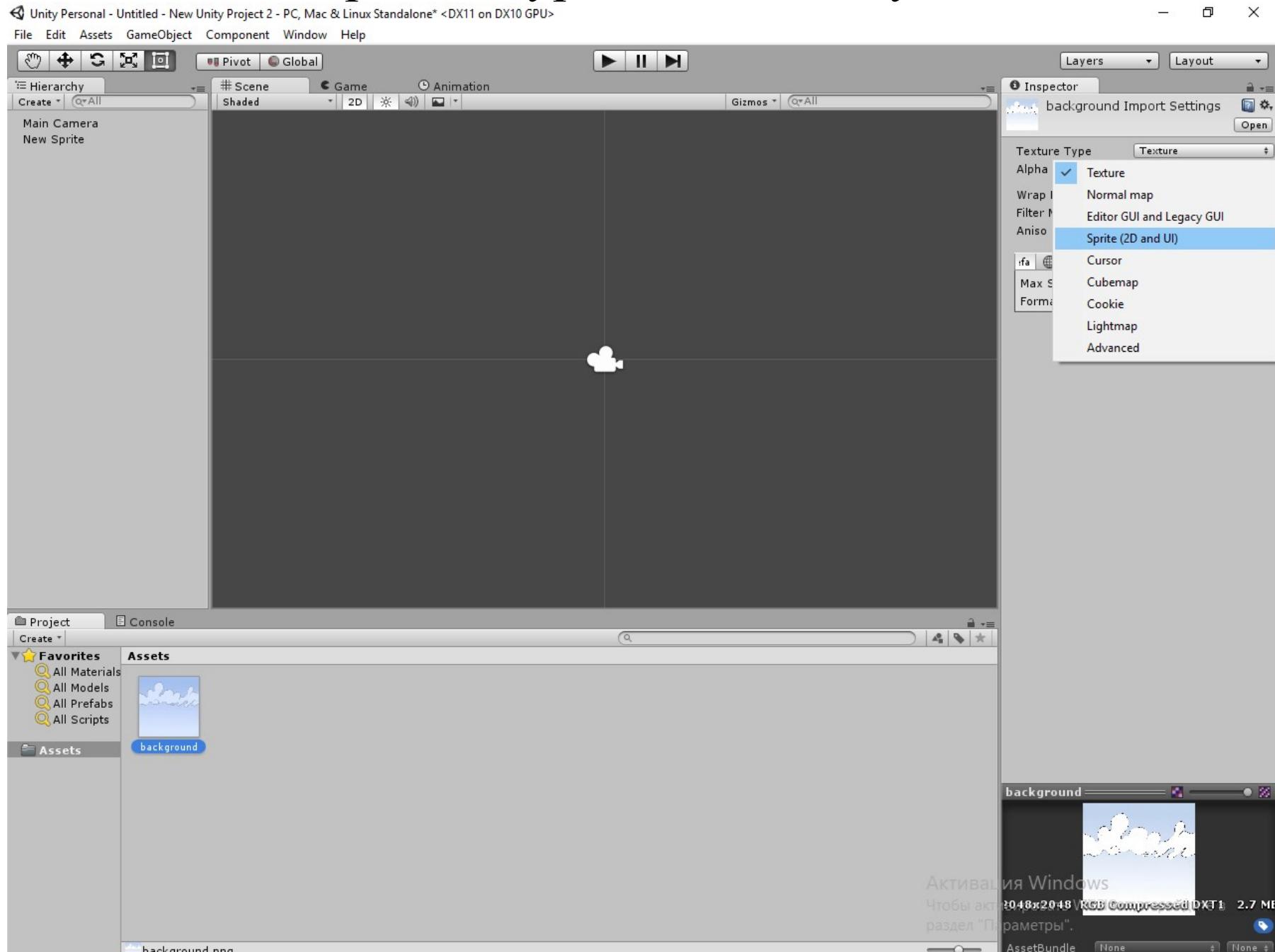
3D **2D** Back Create project

 Community Documentation Tutorials

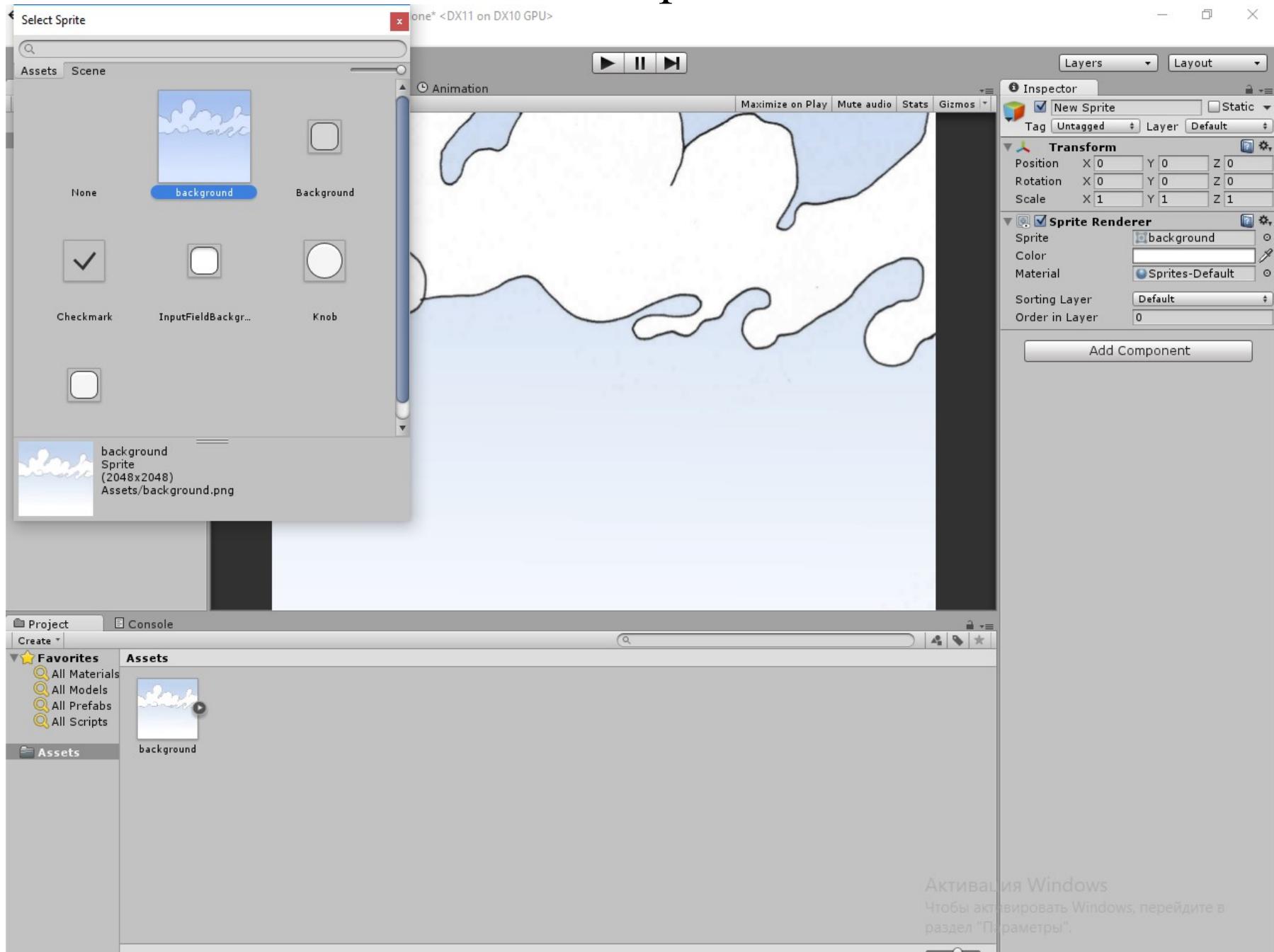
# Створення спрайту



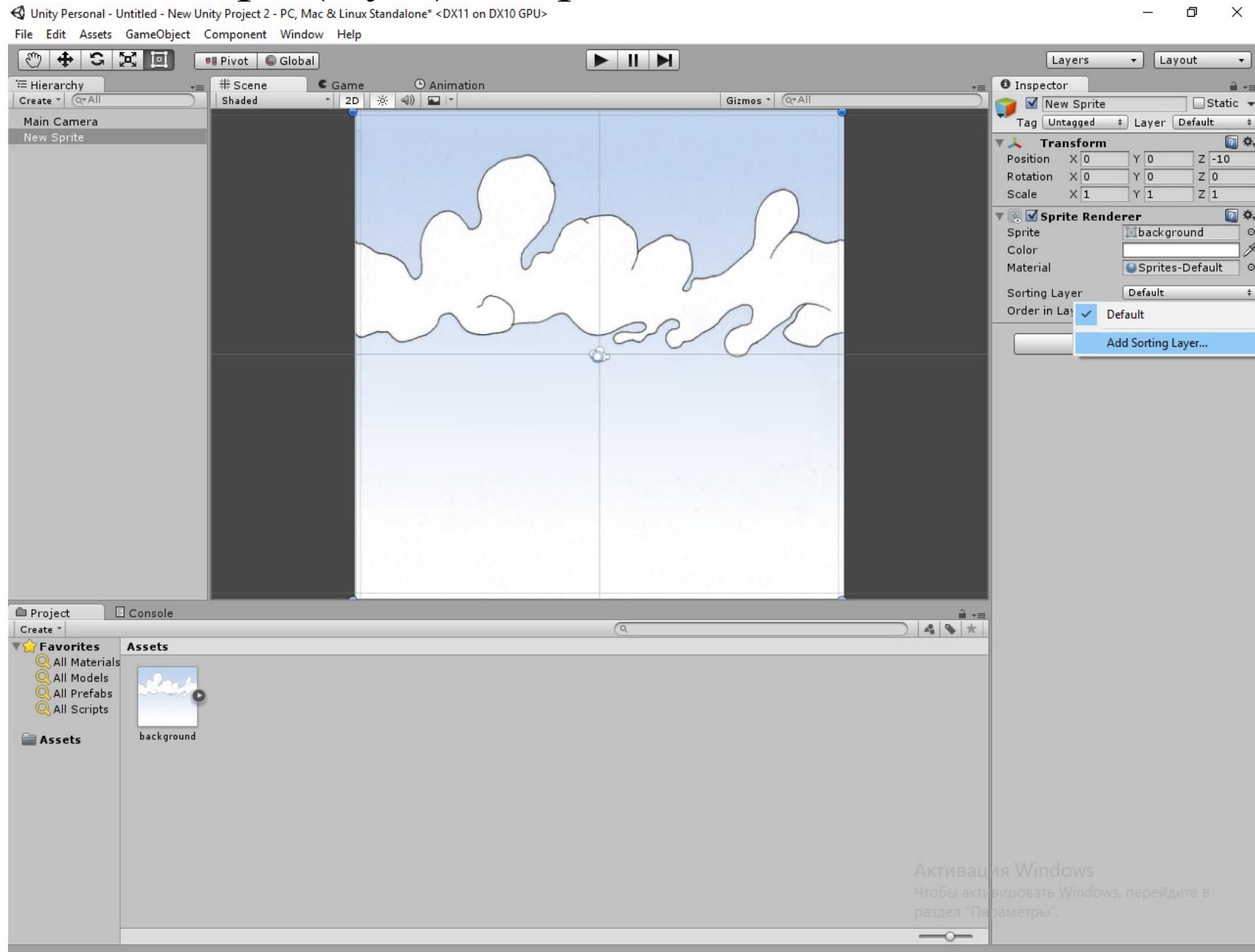
# Імпорт текстури та її налаштування



# Фон спрайта



# Шари (layer) в спрайтах та їх добавлення



File Edit Assets GameObject Component Window Help



Pivot Global



Hierarchy

Create All

Main Camera  
New Sprite

# Scene

Game

Animation

Shaded

2D



Gizmos

All



Layers

Layout

Inspector

Tags &amp; Layers



Tags

Sorting Layers

= Layer	Default
= Layer	Layer1
= Layer	Layer2
= Layer	New Layer



Layers

Project Console

Create All

Favorites

- All Materials
- All Models
- All Prefabs
- All Scripts

Assets



background

Assets

TagManager.asset

Активация Windows

Чтобы активировать Windows, перейдите в раздел "Параметры".



AssetBundle

None

None

File Edit Assets GameObject Component Window Help



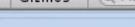
Pivot Global



Layers Layout

Hierarchy  
Create AllMain Camera  
New SpriteScene Game Animation  
Shaded 2D

Gizmos All

Inspector  
New Sprite Static  
Tag Untagged Layer DefaultTransform  
Position X 0 Y 0 Z -10  
Rotation X 0 Y 0 Z 0  
Scale X 1 Y 1 Z 1Sprite Renderer  
Sprite background  
Color  
Material Sprites-Default  
Sorting Layer Default  
Order in Layer  
Default  
Layer1  
Layer2  
New Layer  
Add Sorting Layer...

Project Console

Create All

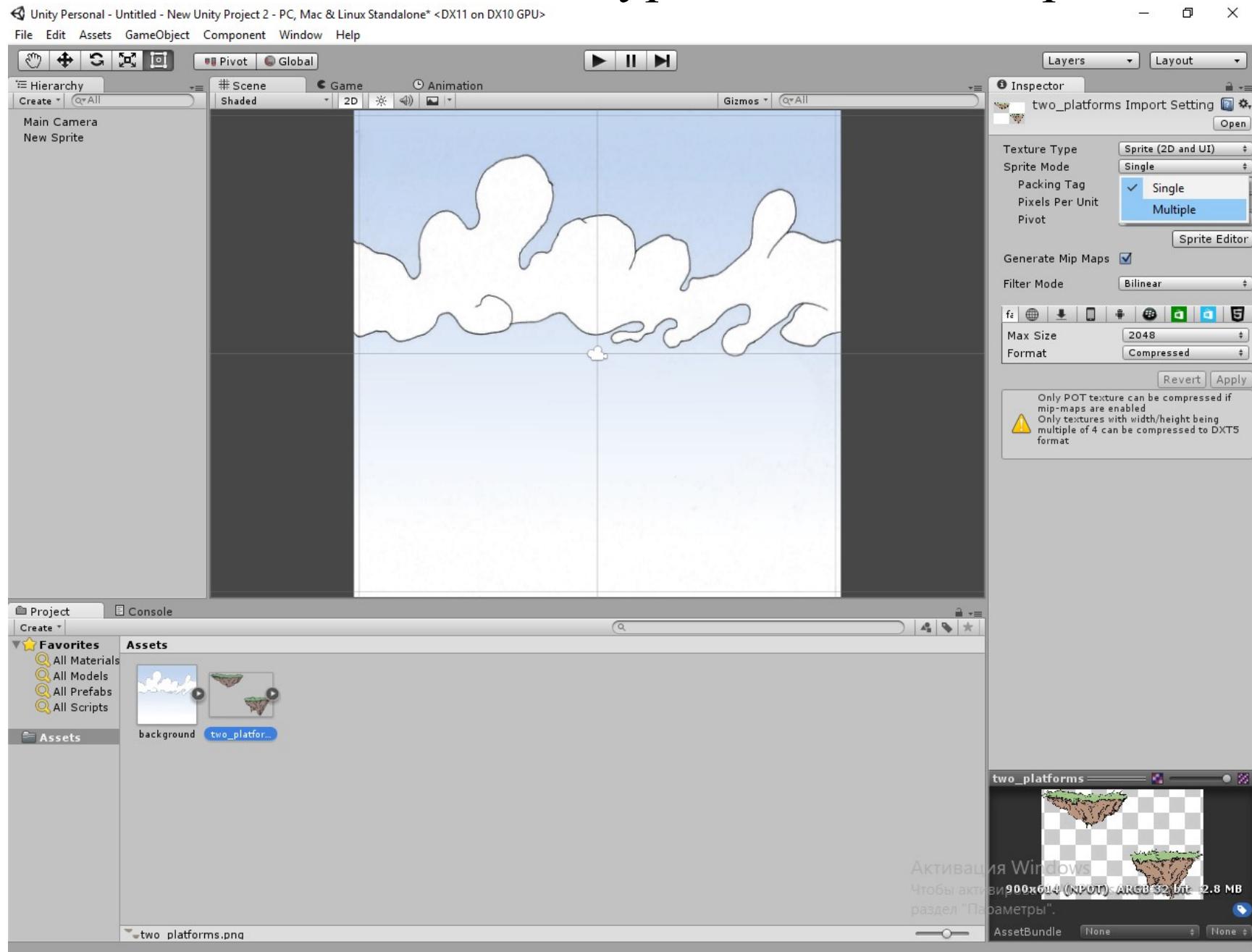
Favorites  
All Materials  
All Models  
All Prefabs  
All Scripts

background

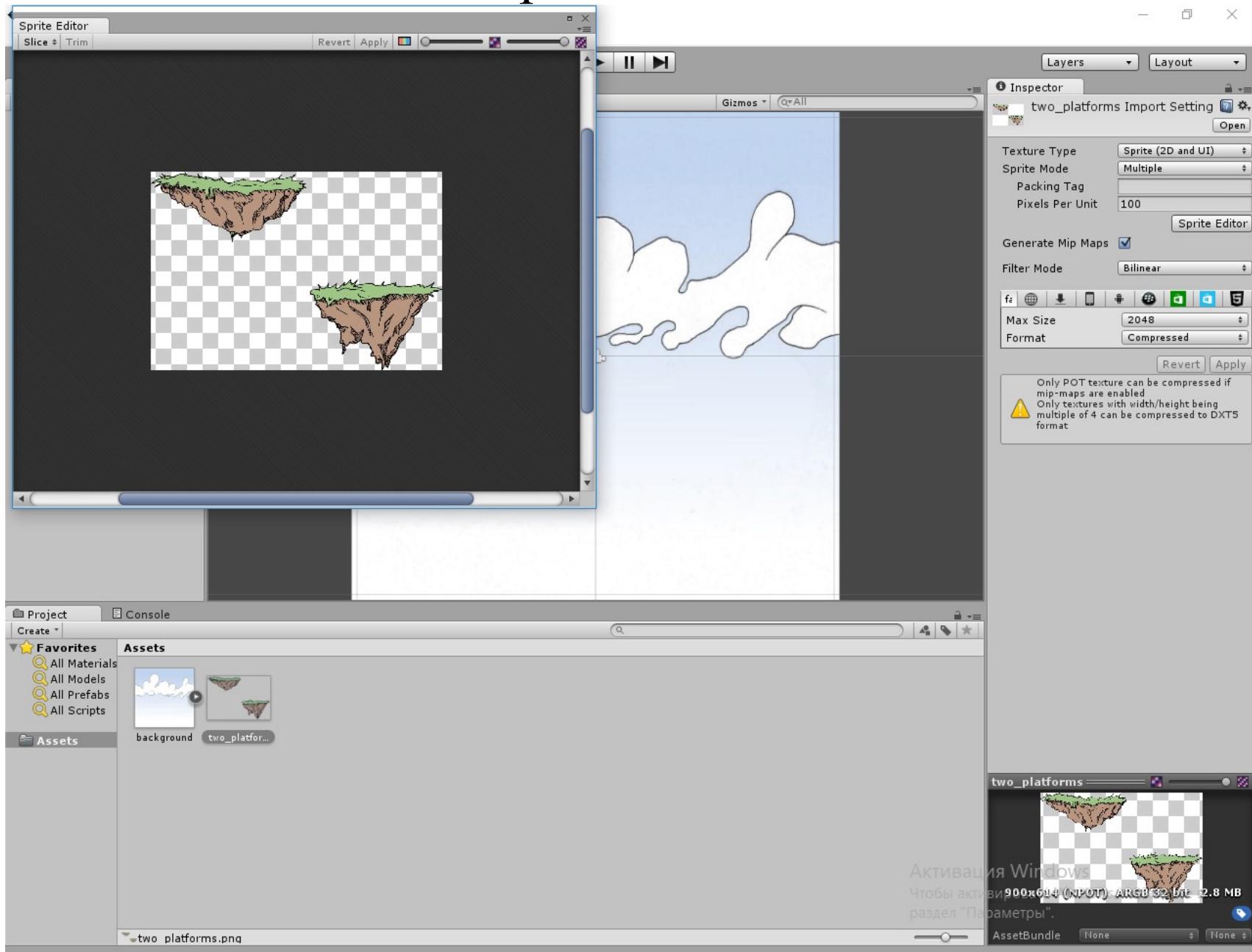
Assets

Активация Windows  
Чтобы активировать Windows, перейдите в  
раздел "Параметры".

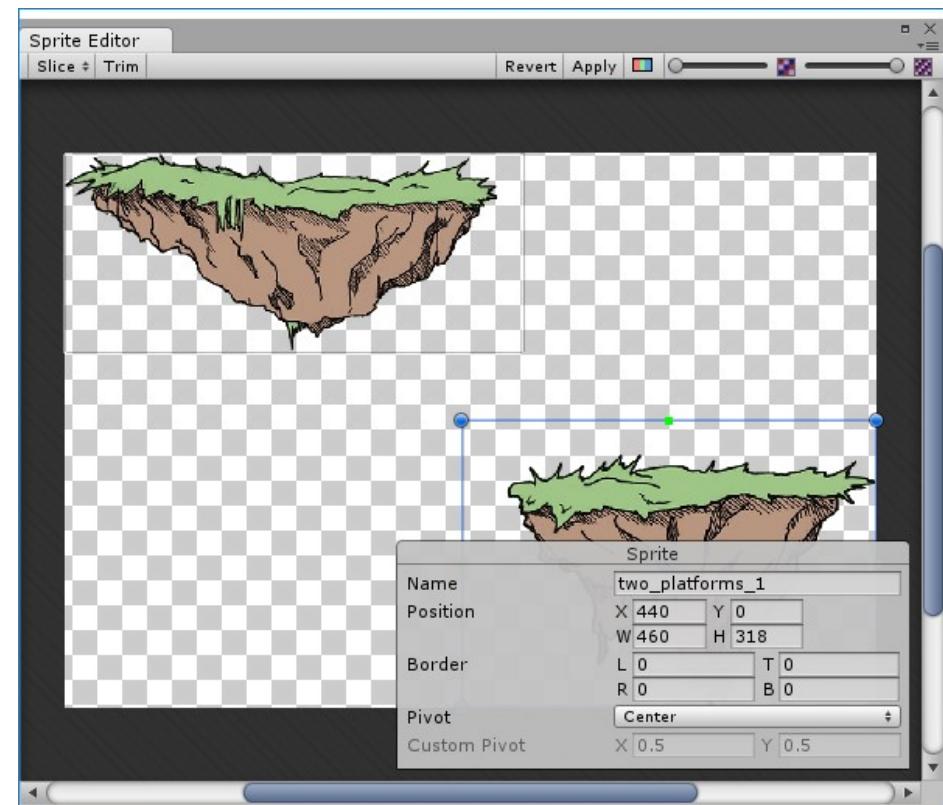
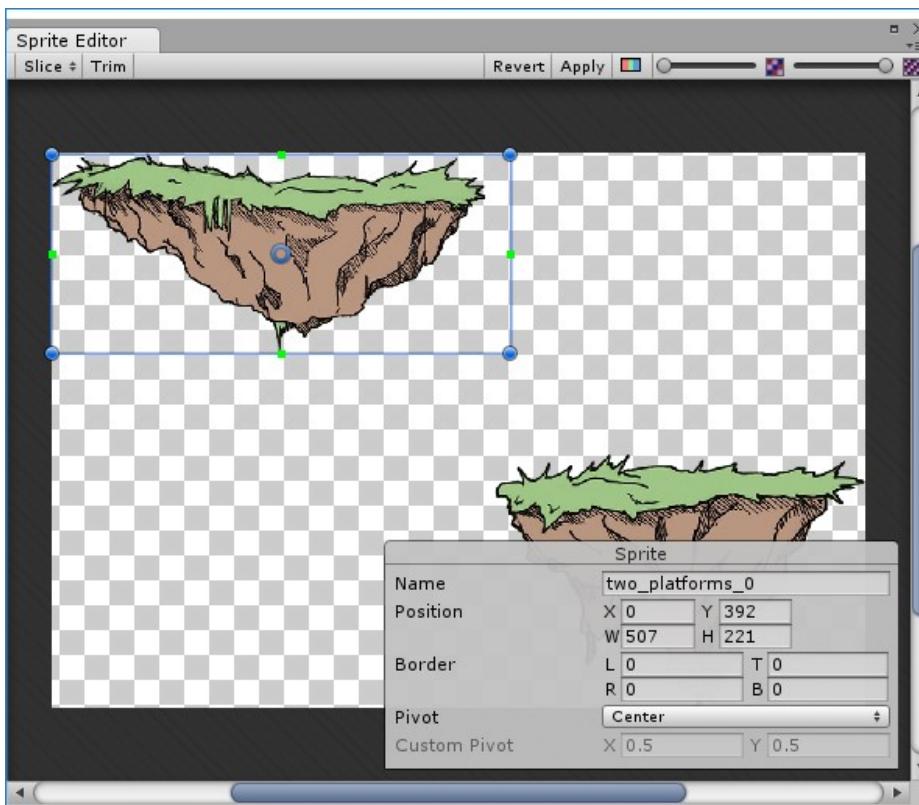
# Розділення однієї текстури на декілька спрайтів



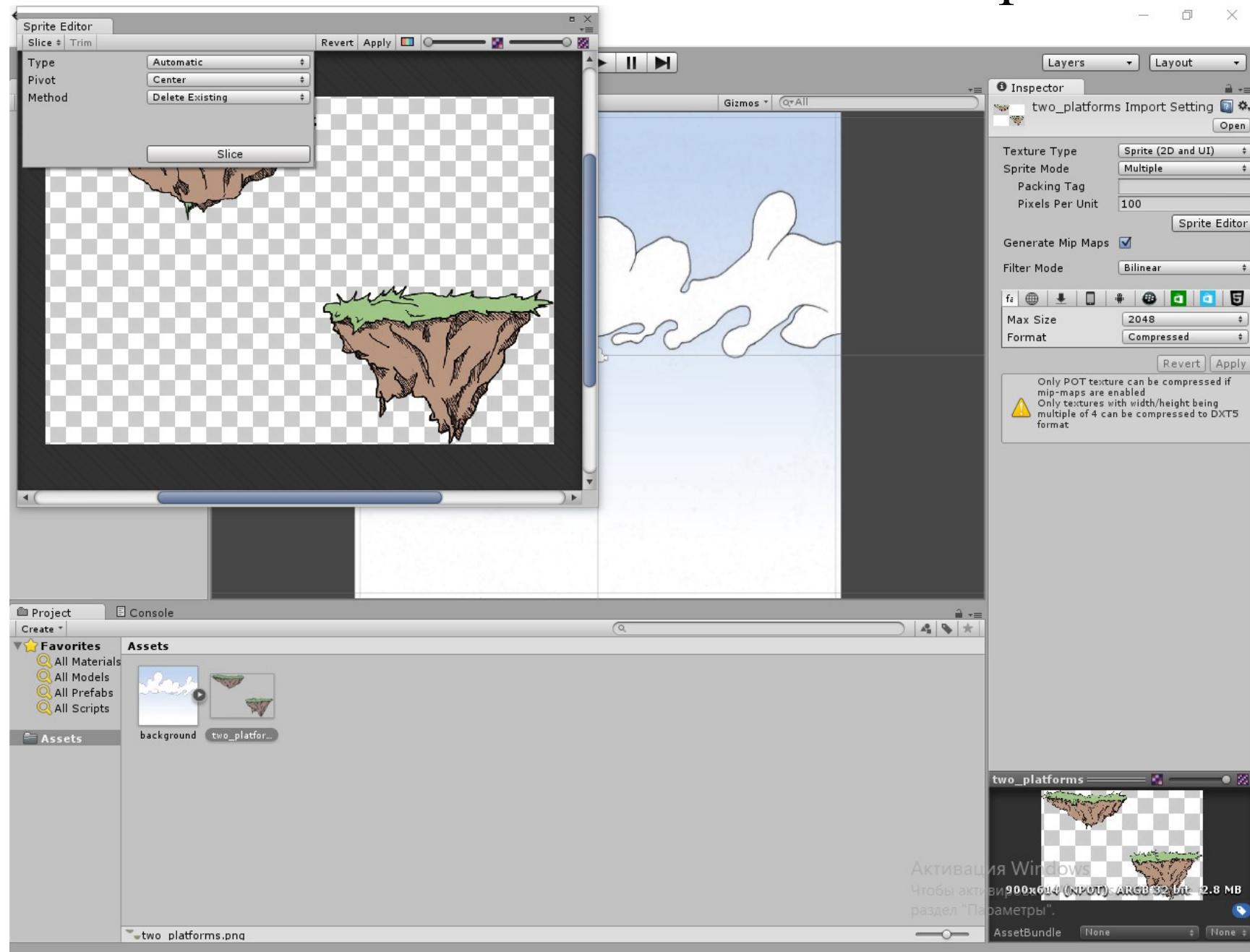
# Sprite Editor



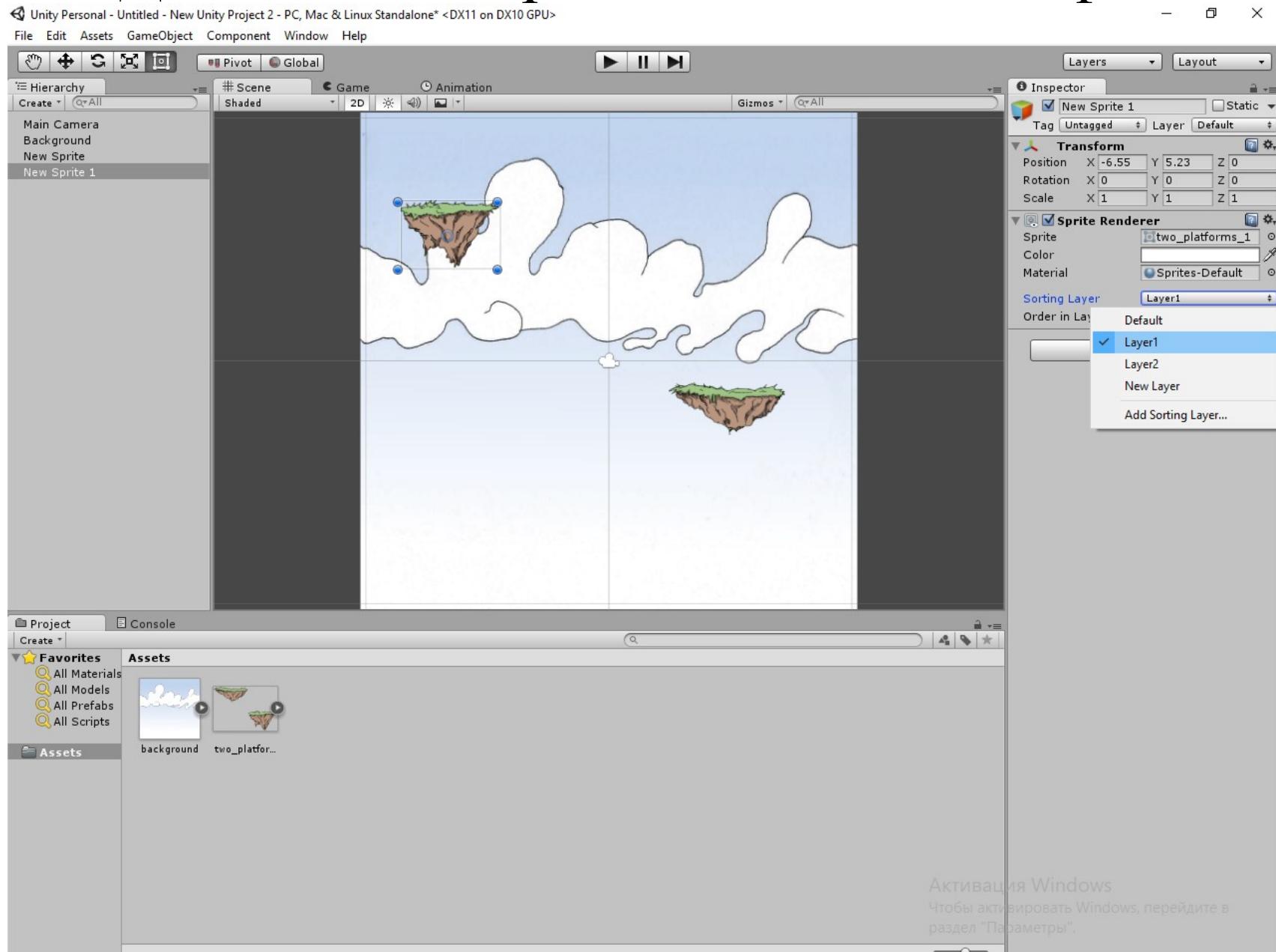
# Виділення спрайтів прямокутником



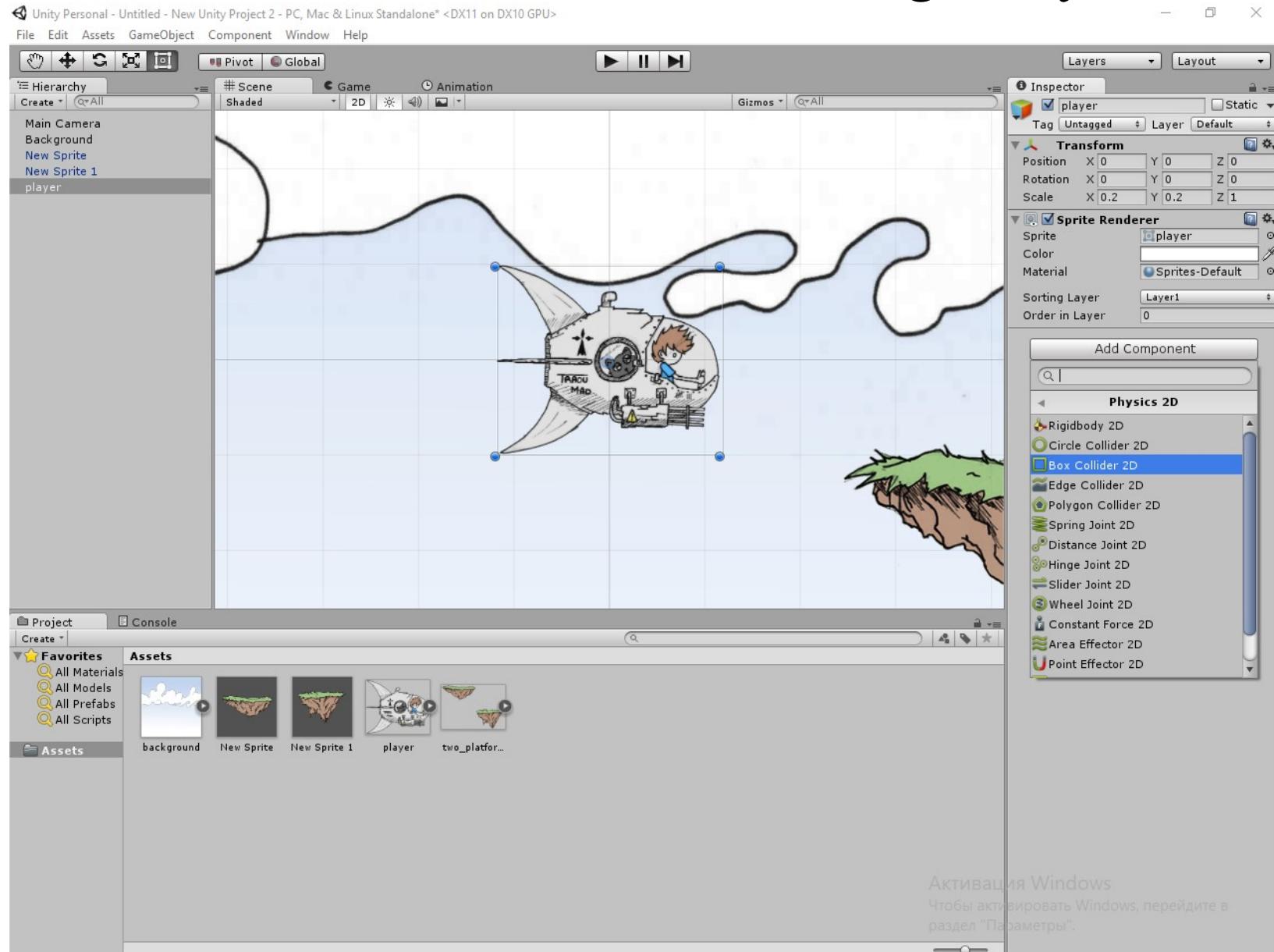
# Slice і автоматизація виділення спрайтів



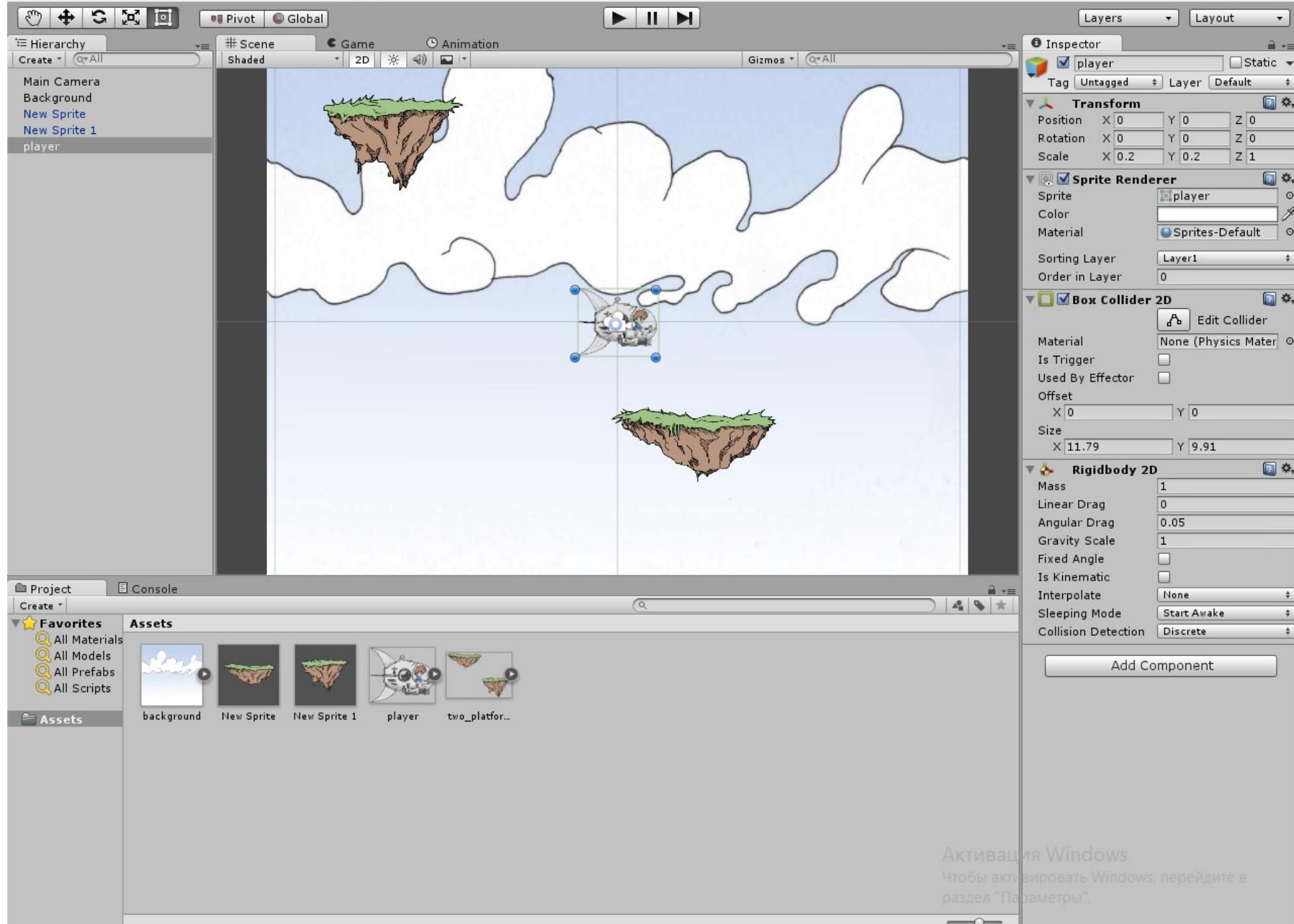
# Добавлення спрайтів на основний фон



# Добавлення гравця та встановлення для нього компонентів Collider і Rigidbody



File Edit Assets GameObject Component Window Help



Активация Windows  
Чтобы активировать Windows, перейдите в  
раздел "Параметры".

# Створення та підключення скрипта руху

using UnityEngine;

```
public class PlayerScript : MonoBehaviour
```

```
{
```

```
    public Vector2 speed = new Vector2(5, 5);
```

```
    private Vector2 movement;
```

```
    private Rigidbody2D rigidbody2D;
```

```
    void Start()
```

```
{
```

```
        rigidbody2D = GetComponent<Rigidbody2D>();
```

```
}
```

```
    void Update()
```

```
{
```

```
        float inputX = Input.GetAxis("Horizontal");
```

```
        float inputY = Input.GetAxis("Vertical");
```

```
        movement = new Vector2(
```

```
            speed.x * inputX,
```

```
            speed.y * inputY);
```

```
}
```

```
    void FixedUpdate()
```

```
{
```

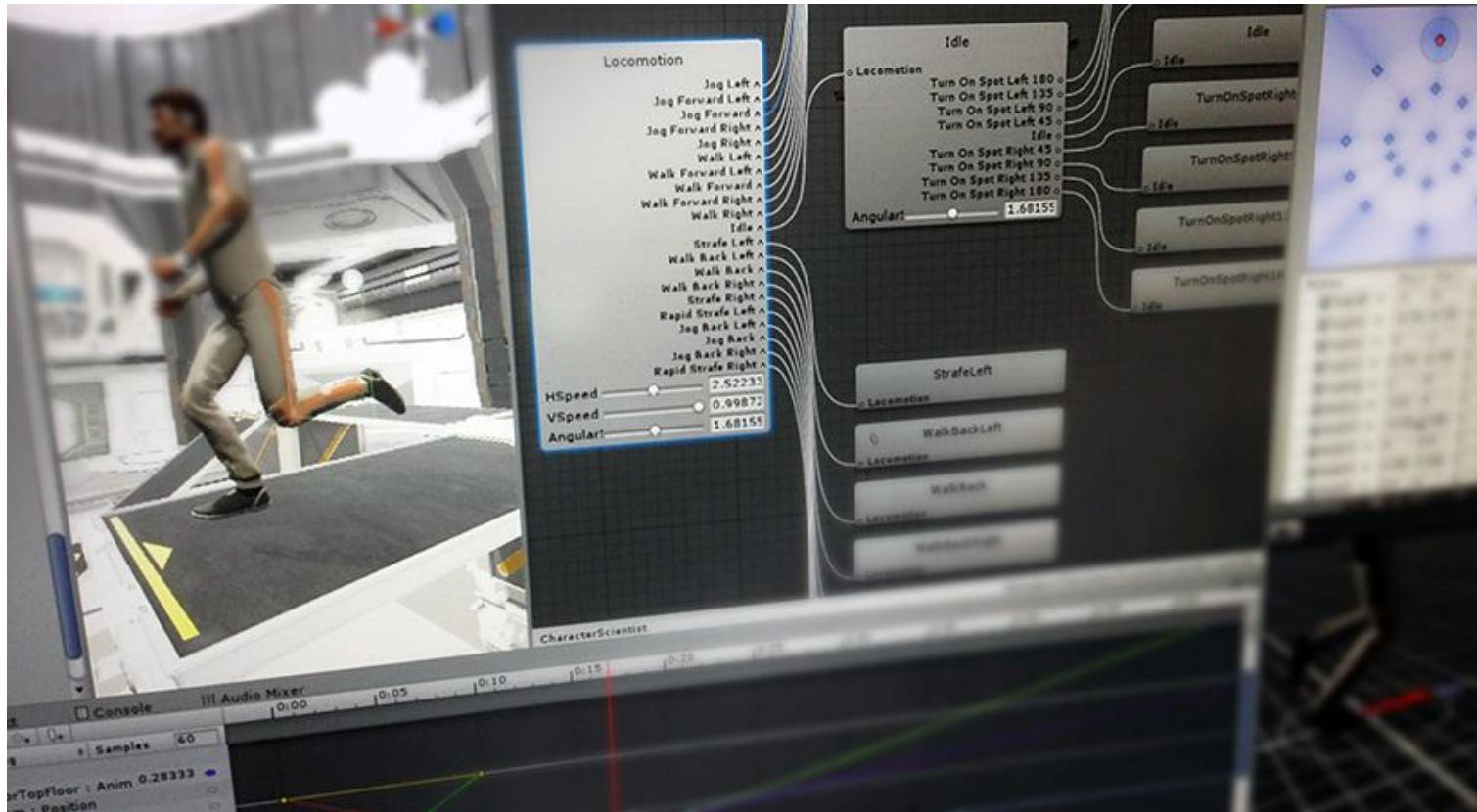
```
        rigidbody2D.velocity = movement;
```

```
}
```

Дякую за увагу!



# Тема 8. Анімація

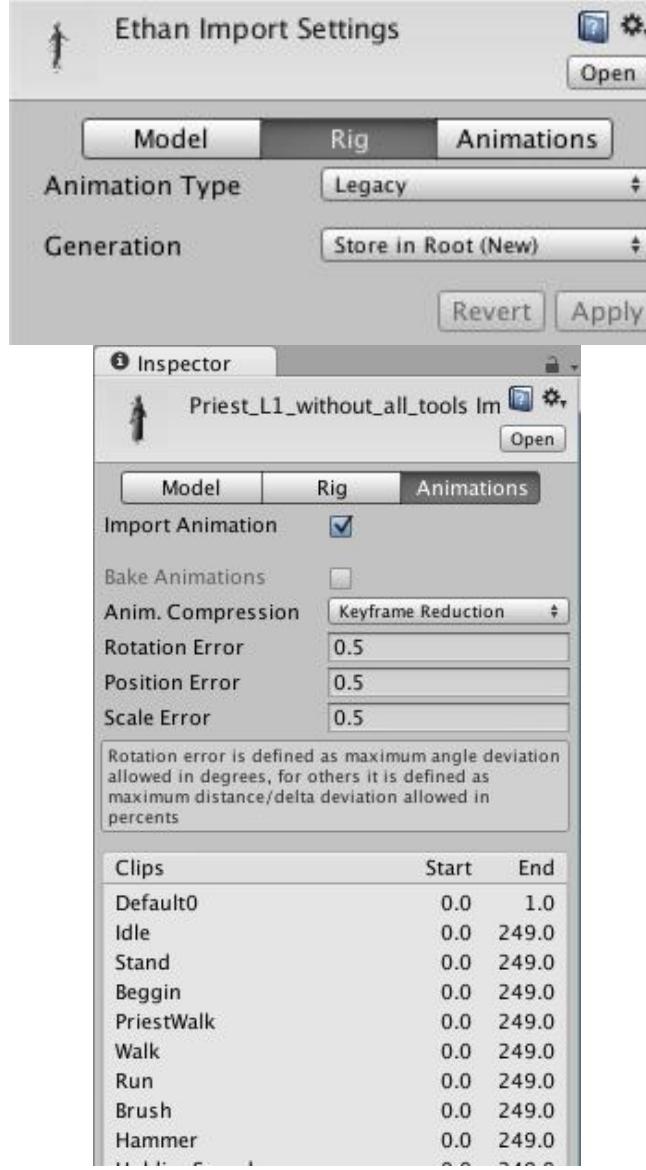


<https://docs.unity3d.com/ru/current/Manual/AnimationSection.html>

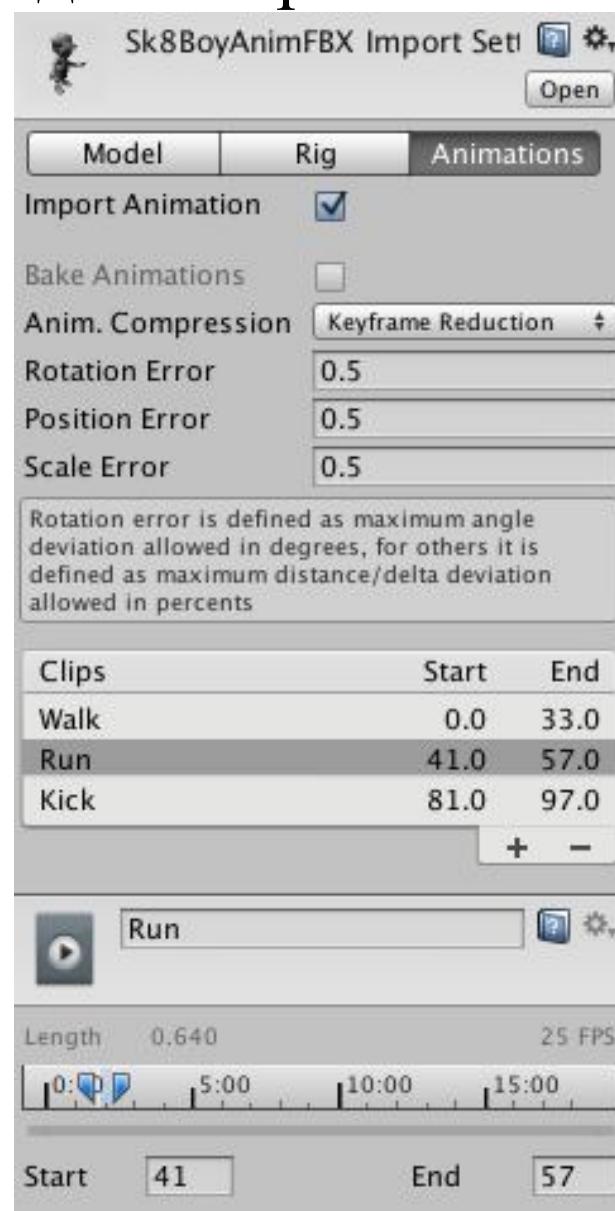
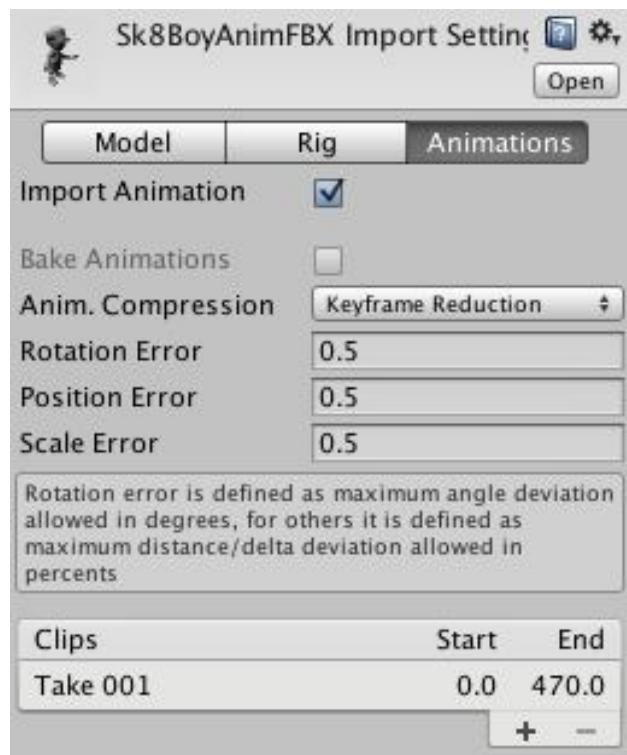
<https://docs.unity3d.com/ru/current/Manual/Animations.html>

# Анімація

<https://docs.unity3d.com/ru/current/Manual/Animations.html>

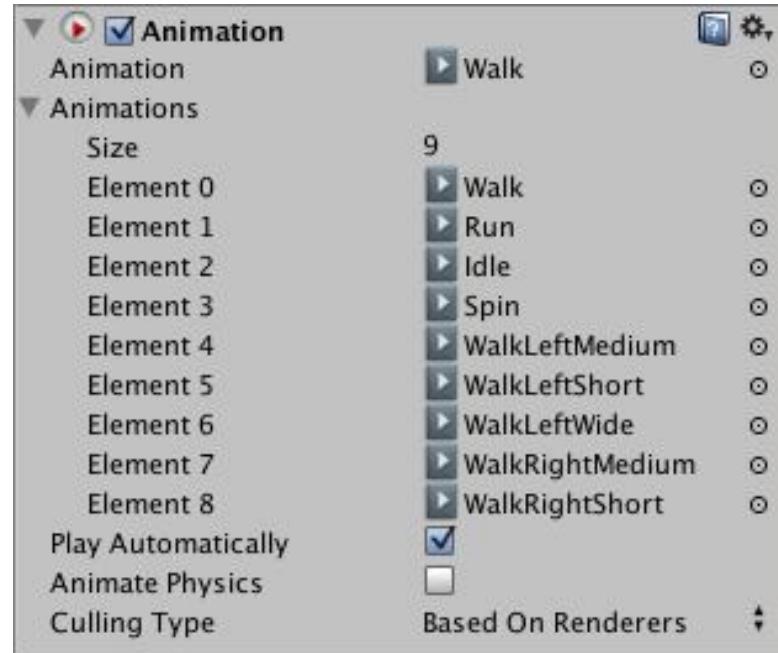


# Робота з моделями без попередньо нарізаної анімації



Import Animation	Вибір – імпортування анімації
Wrap Mode	Визначає що робити після закінчення анімації:
Default	Використовує установки зазначені в кліпі анімації
Once	Програти кліп до кінця і зупинитися
Loop	Програти кліп до кінця і тут же почати з початку
PingPong	Програється до кінця, потім від кінця до початку в зворотному порядку і т.д.
Forever	Грає до кінця, потім нескінченно повторює останній кадр
Anim. Compresion	Встановлюється для оптимізації кліпу анімації
Off	Без оптимізації
Keyframe reduction	Спроба видалити ключові кадри, де зміни надто непомітні для ока
Keyframe reduction and compression	Працює як Keyframe reduction, але дані кліпу також стискаються.
Rotation error	Мінімальні зміни в значеннях повороту, нижче яких два ключових кадри вважаються рівними
Position error	Мінімальні зміни в положенні (у відсотках до значення координати), нижче яких два ключових кадри вважаються рівними
Scale error	Мінімальні зміни в масштабі (у відсотках до значення координати), нижче яких два ключових кадри вважаються рівними.

# Компонент Animation



Animation	Анімація, яка буде програватися за замовчуванням, якщо відзначений пропорець Play Automatically
Animations	Список доступних анімацій з коду
Play Automatically	Чи повинна анімація автоматично програватися при запуску гри?
Animate Physics	Чи повинен анімуємий об'єкт оброблятися фізичним движком?
Culling Type	Визначає, коли не повинна відтворюватись анімація.
Always Animate	Завжди програвати.

Based on Renderers	На підставі пози за замовчуванням.
Based on Clip Bounds	На підставі кордонів кліпу (розраховуються при імпорті анімації), тобто анімація не буде відтворюватись, якщо кордону кліпу не потрапляють в поле зору.
Based on User Bounds	На підставі кордонів, обраних користувачем, тобто, анімація не буде відтворюватись, якщо обрані користувачем кордону не потрапляють в поле зору.

# Скриптінг анімації

<https://docs.unity3d.com/ru/current/ScriptReference/Animation.html>

## Змішування анімацій

```
void Update () {  
    if (Input.GetAxis("Vertical") > 0.2)  
        animation.CrossFade ("walk");  
    else  
        animation.CrossFade ("idle");}
```

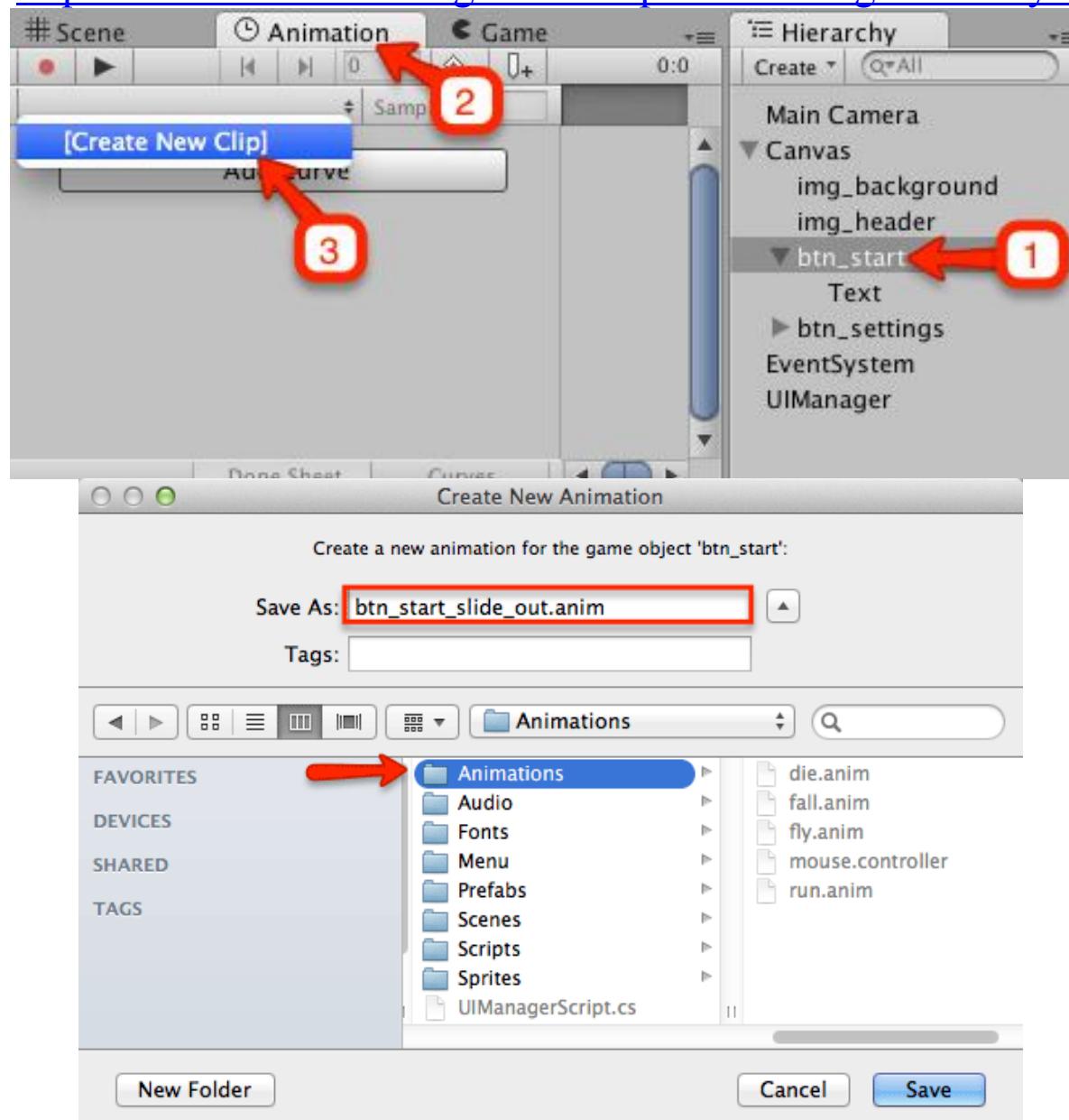
## Слой анімації

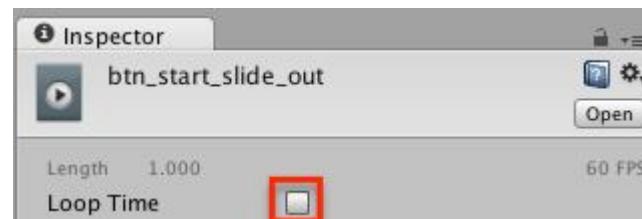
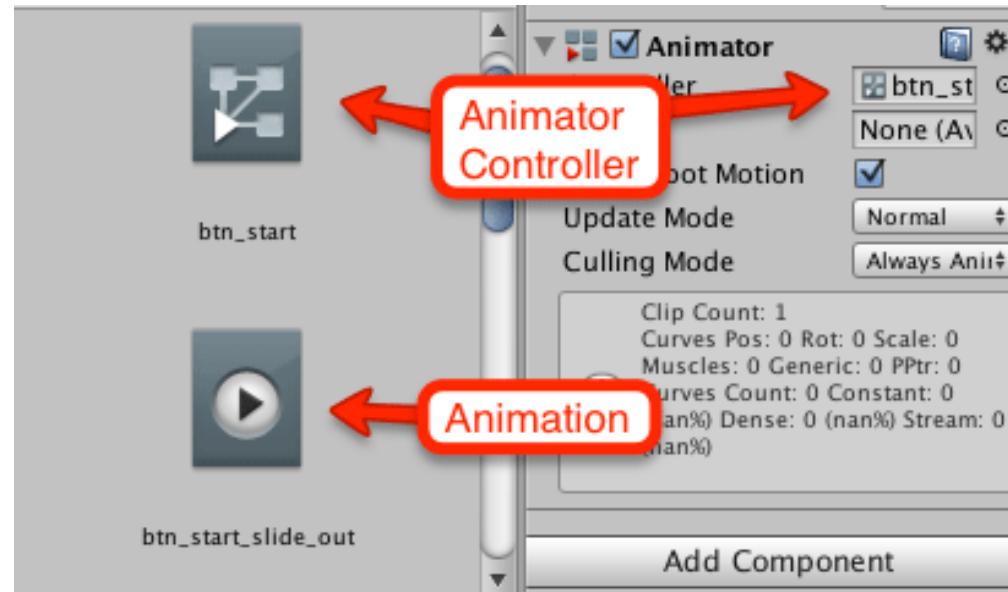
```
void Start () {  
    animation.wrapMode = WrapMode.Loop;  
    animation["shoot"].wrapMode = WrapMode.Once;  
    animation["shoot"].layer = 1;  
    animation.Stop();}  
}
```

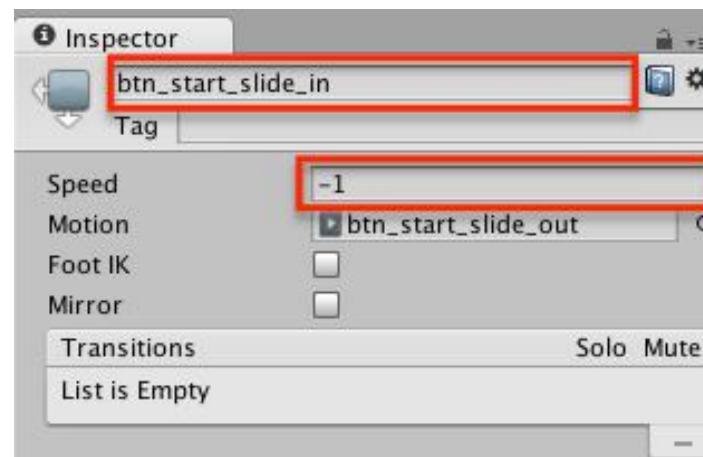
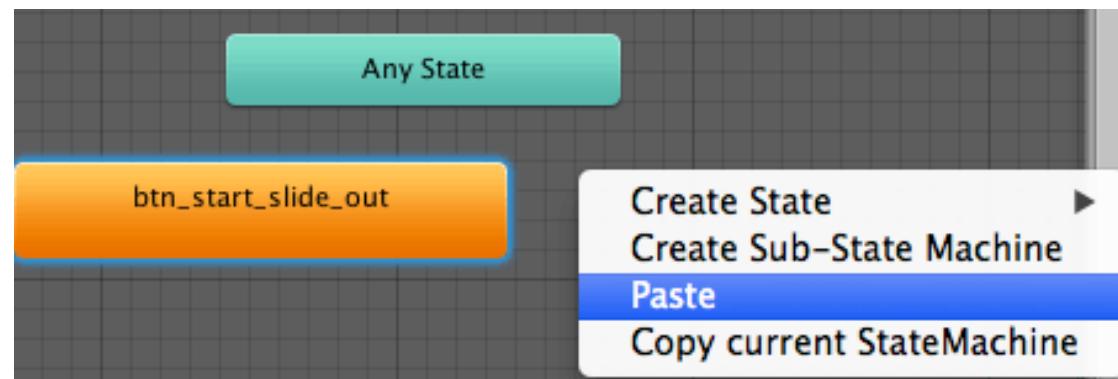
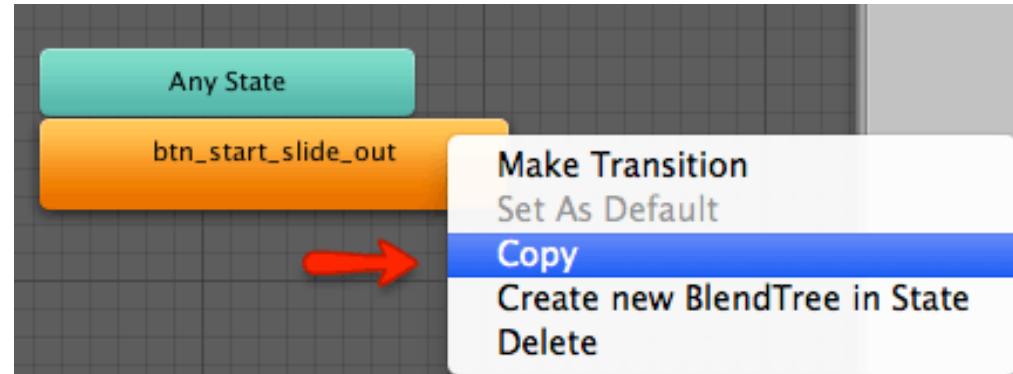
```
void Update () {  
    if (Mathf.Abs(Input.GetAxis("Vertical")) > 0.1)  
        animation.CrossFade("walk");  
    else  
        animation.CrossFade("idle");  
    if (Input.GetButtonDown ("Fire1"))  
        animation.CrossFade("shoot");}
```

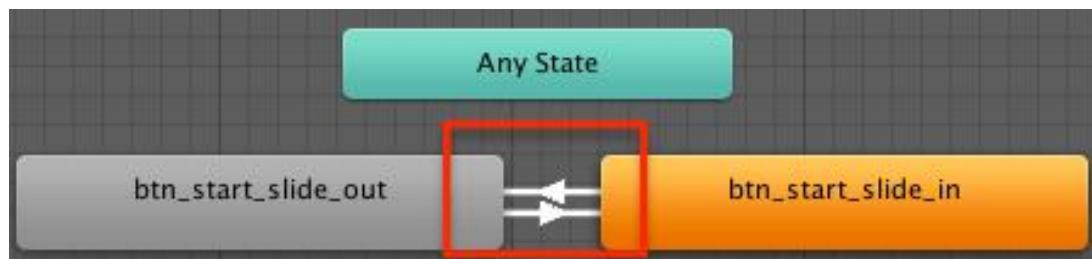
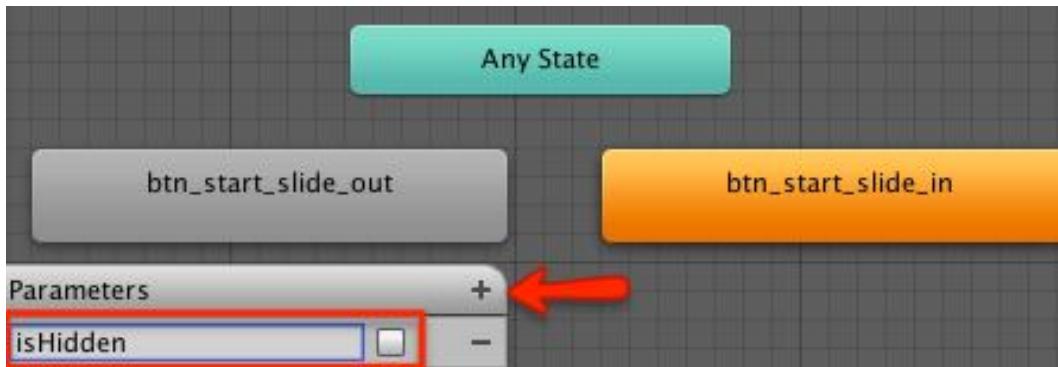
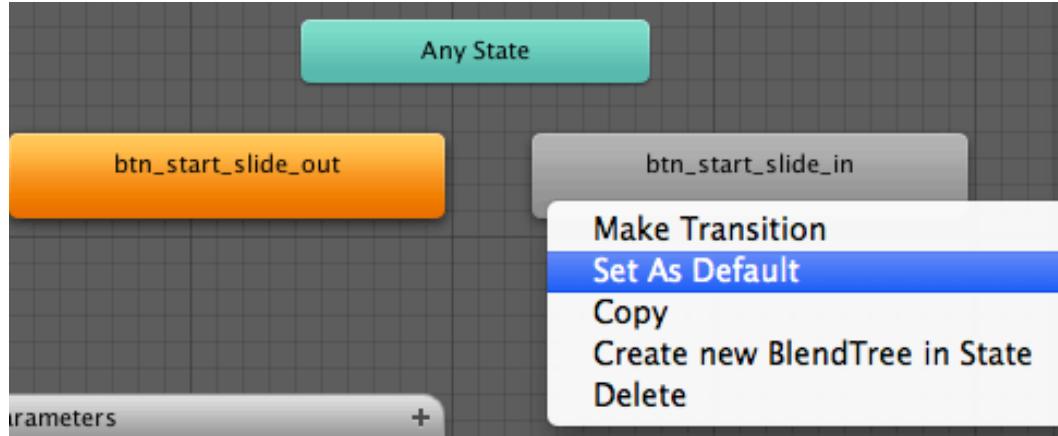
# Приклад анімації меню з минулої лекції

<http://websketches.ru/blog/uhebnik-po-novomu-gui-v-unity-2>







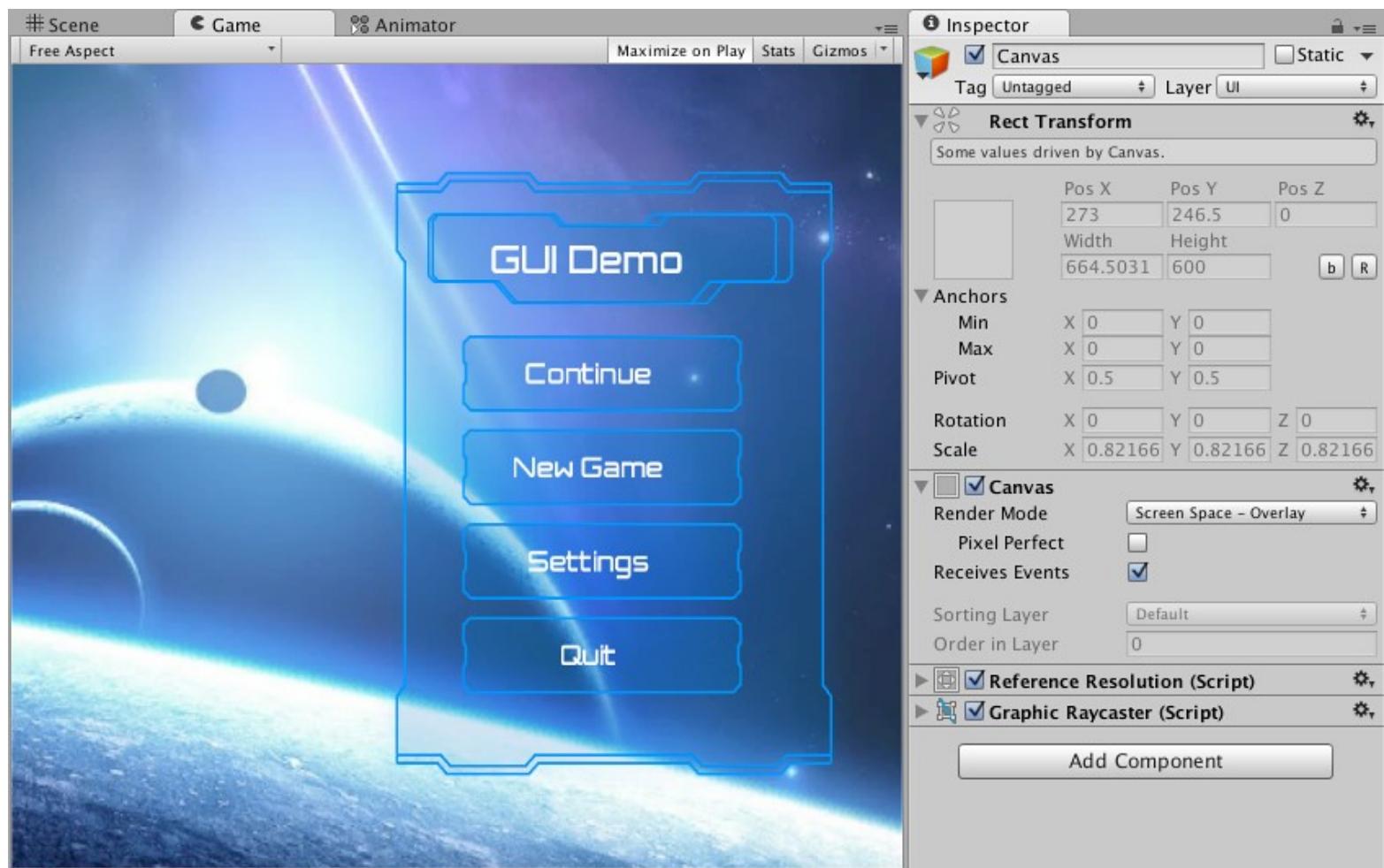




Дякую за увагу!



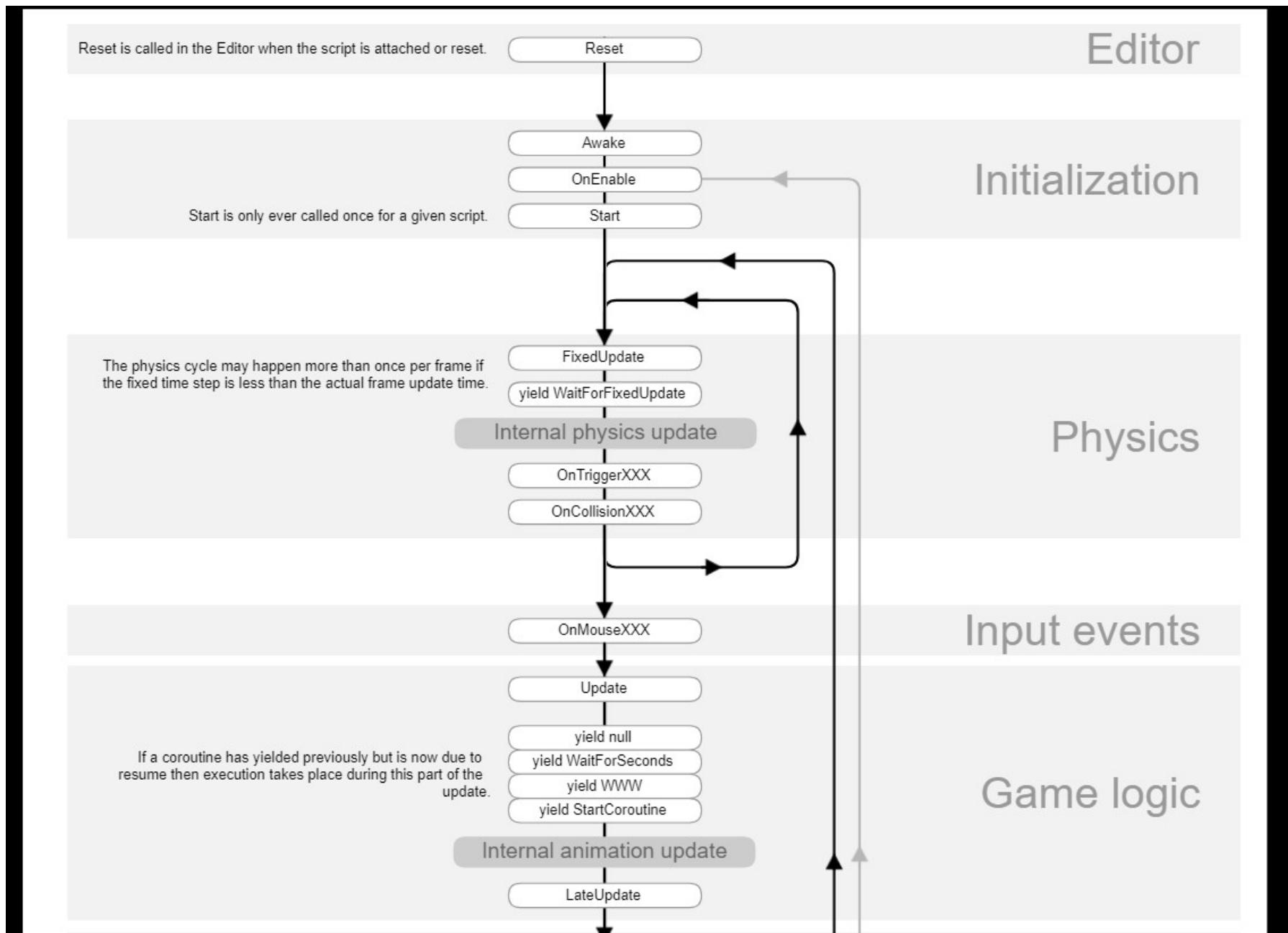
# Тема 7. Інтерфейс користувача

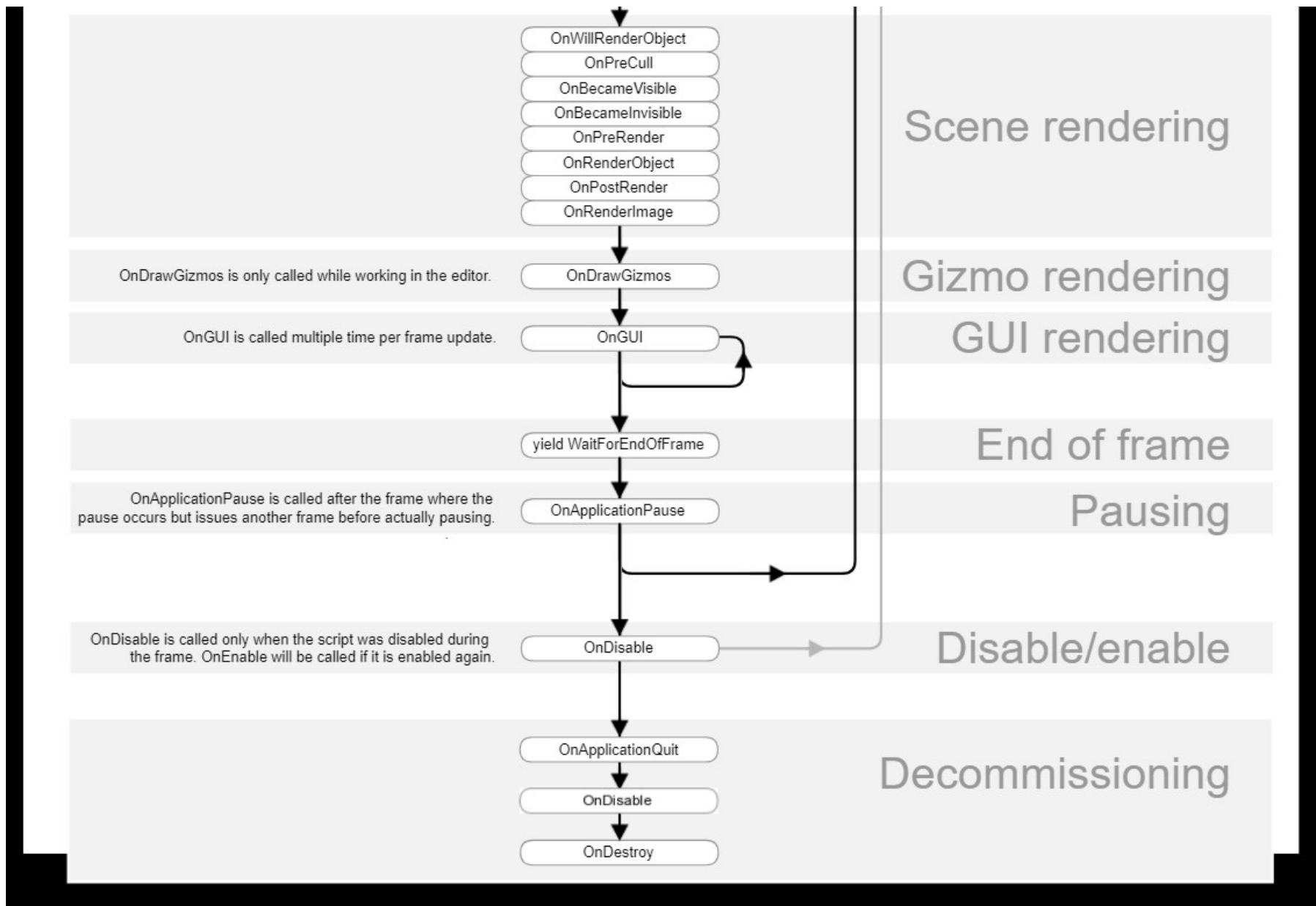


<https://docs.unity3d.com/ru/current/Manual/UISystem.html>

<https://unity3d.com/ru/learn/tutorials/s/user-interface-ui>

# БЛОК-СХЕМА ЖИТТЕВОГО ЦИКЛУ СКРИПТА





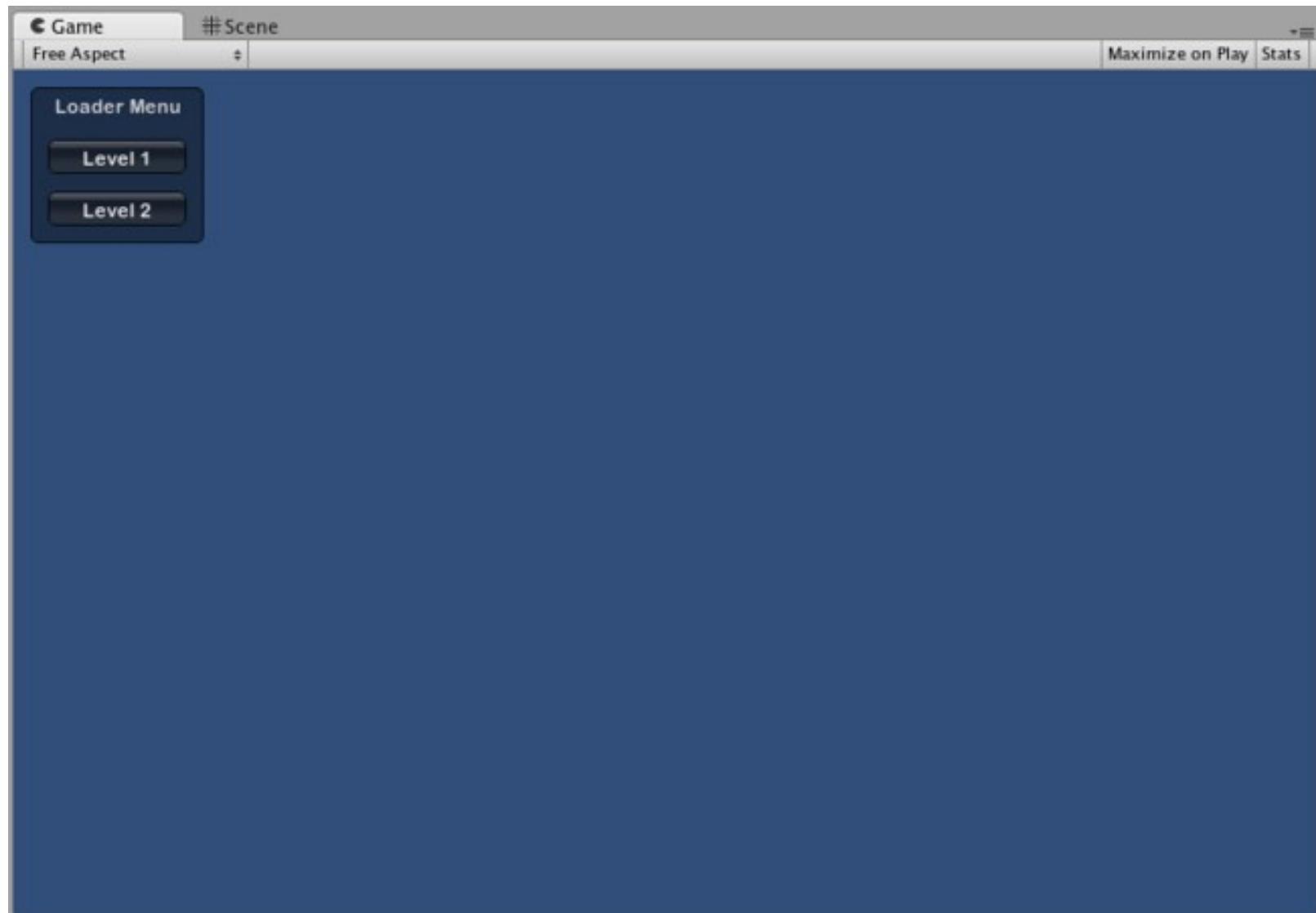
<https://docs.unity3d.com/ru/current/Manual/ExecutionOrder.html>

# Розробка GUI з використанням застарілої системи

<https://docs.unity3d.com/ru/500/Manual/GUIScriptingGuide.html>

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

```
public class GUITest : MonoBehaviour {  
    void OnGUI () {  
        GUI.Box(new Rect(10,10,100,90), "Loader Menu");  
  
        if(GUI.Button(new Rect(20,40,80,20), "Level 1")) {  
            Application.LoadLevel(1);  
        }  
  
        if(GUI.Button(new Rect(20,70,80,20), "Level 2")) {  
            Application.LoadLevel(2);  
        }  
    }  
}
```



```
using UnityEngine;
using System.Collections;

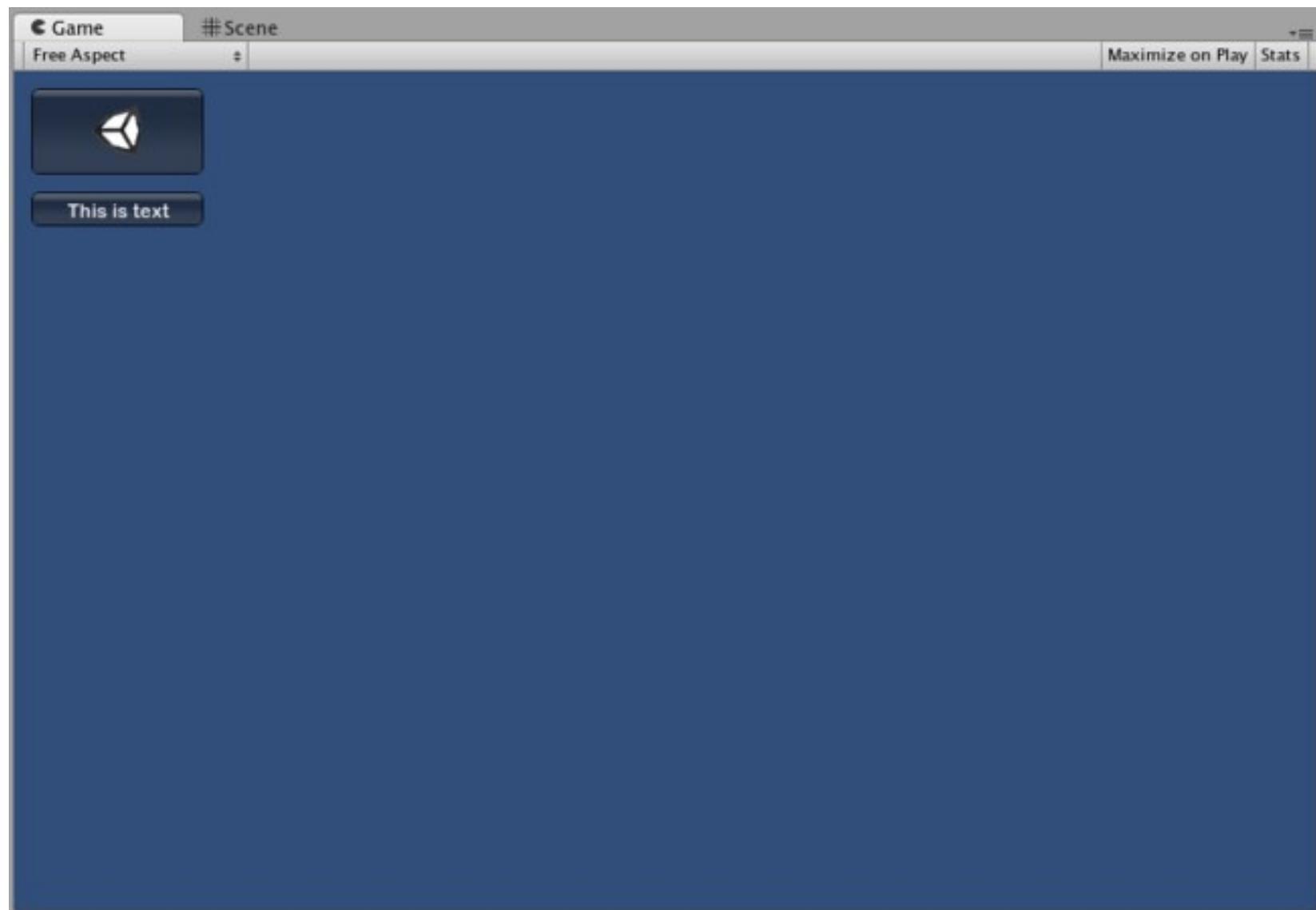
public class GUITest : MonoBehaviour {

    public Texture2D icon;
    string textFieldString="Hello";

    void OnGUI () {

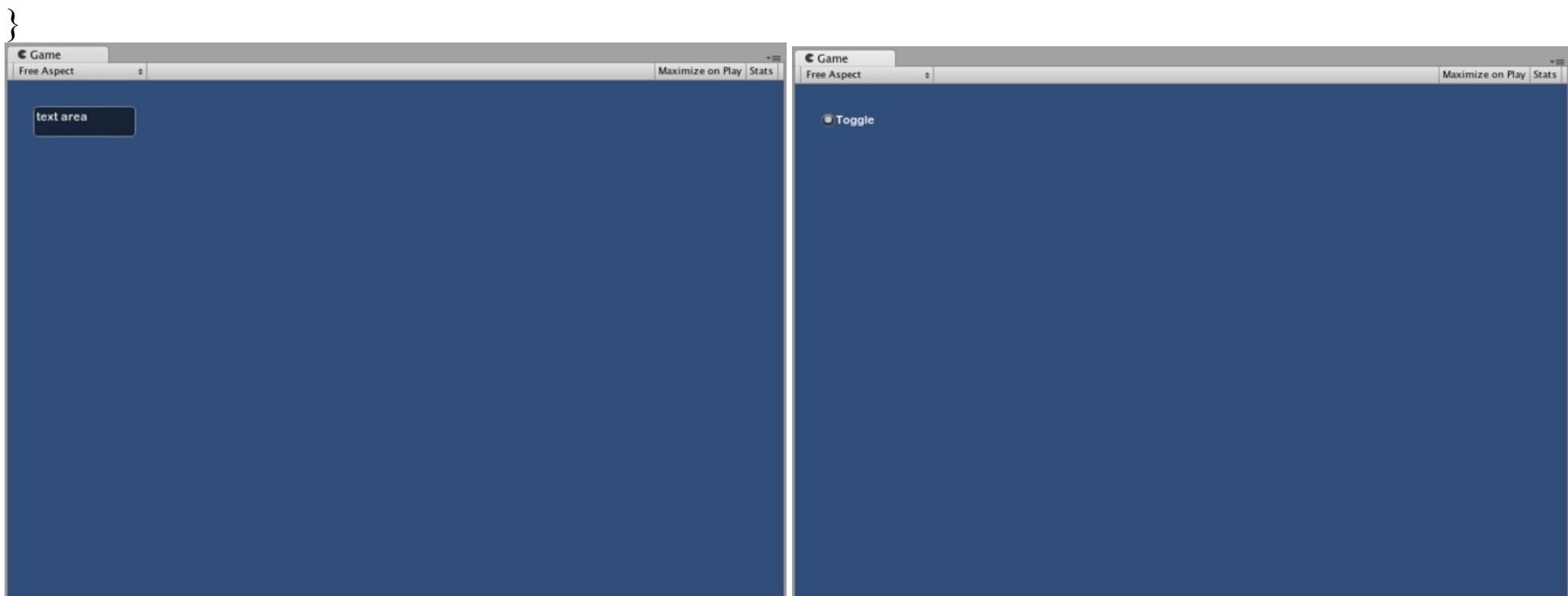
        if (GUI.Button (new Rect (10,10, 100, 50), icon)) {
            print ("you clicked the icon");
        }

        //GUI.Box (new Rect (10,10,100,50), new GUIContent("This is text", icon));
        //GUI.Button (new Rect (10,10,100,20), new GUIContent ("Click me", "This is the tooltip"));
        //GUI.Label (new Rect (10,40,100,20), GUI.tooltip);
        //textFieldString = GUI.TextField (new Rect (25, 25, 100, 30), textFieldString);
        if (GUI.Button (new Rect (10,70, 100, 20), "This is text")) {
            print ("you clicked the text button");
        }
    }
}
```

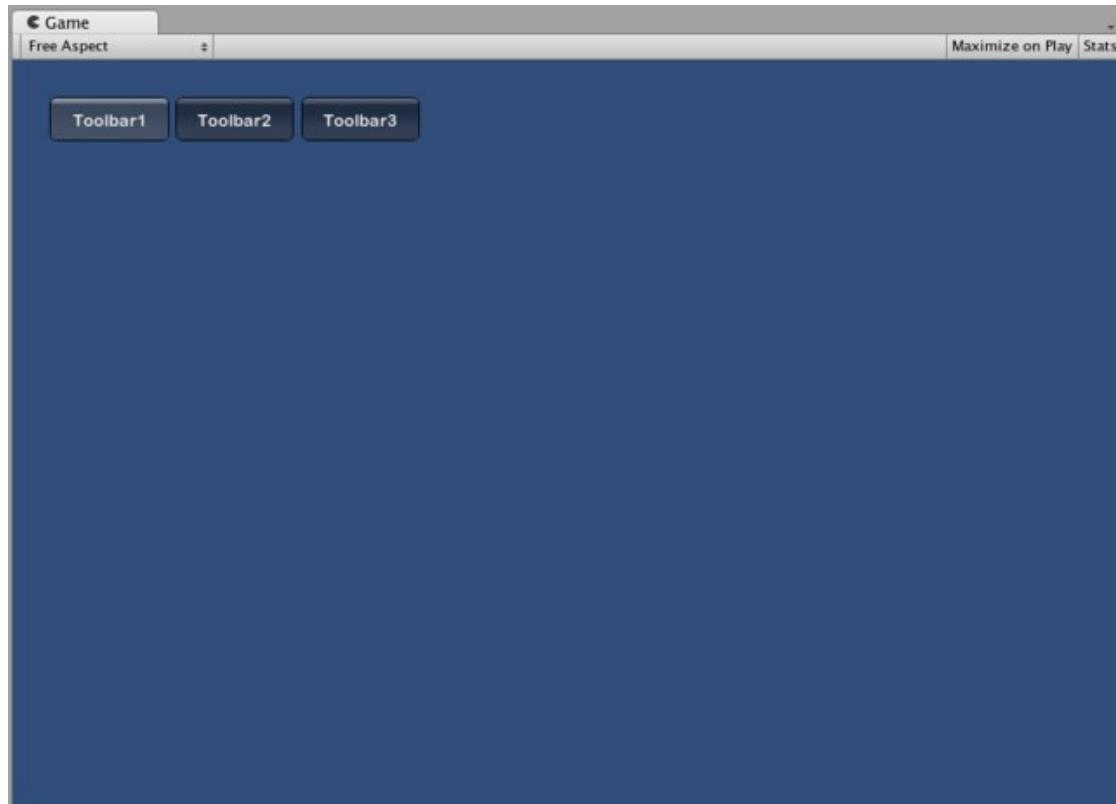


```
using UnityEngine;
using System.Collections;
public class GUITest : MonoBehaviour {

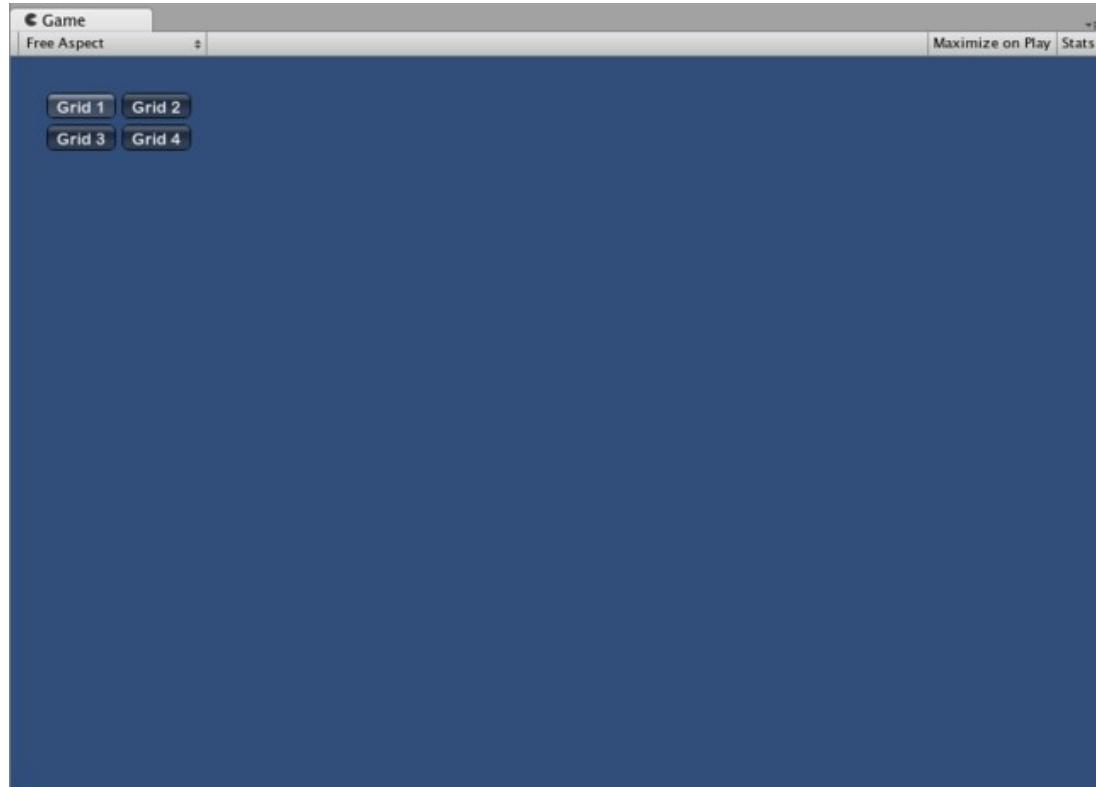
    private string textAreaString = "text area";
    //private bool toggleBool = true;
    void OnGUI () {
        textAreaString = GUI.TextArea (new Rect (25, 25, 100, 30), textAreaString);
        //toggleBool = GUI.Toggle (new Rect (25, 25, 100, 30), toggleBool, "Toggle");
    }
}
```



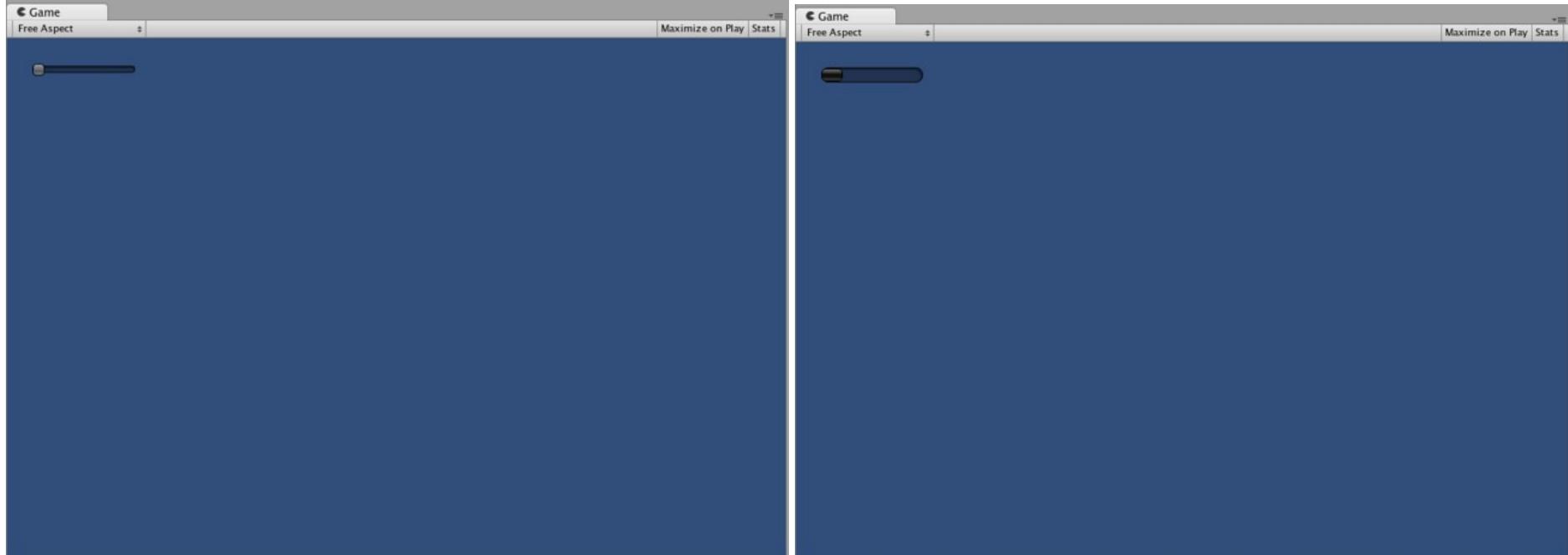
```
public class GUITest : MonoBehaviour {  
  
    private int toolbarInt = 0;  
    private string[] toolbarStrings = {"Toolbar1", "Toolbar2", "Toolbar3"};  
  
    void OnGUI () {  
        toolbarInt = GUI.Toolbar (new Rect (25, 25, 250, 30), toolbarInt, toolbarStrings);  
    }  
}
```



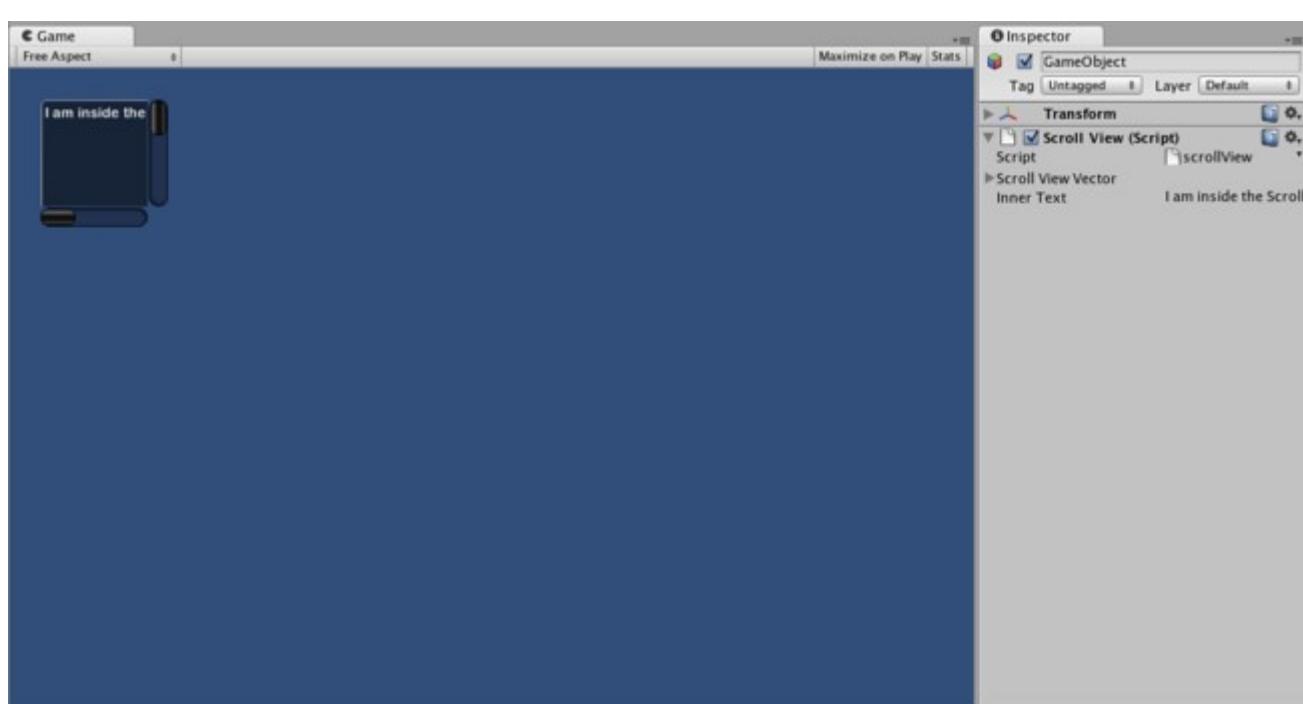
```
public class GUITest : MonoBehaviour {  
    private int selectionGridInt = 0;  
    private string[] selectionStrings = {"Grid 1", "Grid 2", "Grid 3", "Grid 4"};  
    void OnGUI () {  
        selectionGridInt = GUI.SelectionGrid (new Rect (25, 25, 300, 60), selectionGridInt, selectionStrings,  
2);  
    }  
}
```



```
public class GUITest : MonoBehaviour {  
    private float hSliderValue = 0.0f;  
    void OnGUI () {  
        hSliderValue = GUI.HorizontalSlider (new Rect (25, 25, 100, 30), hSliderValue, 0.0f, 10.0f);  
        //vSliderValue = GUI.VerticalSlider (new Rect (25, 25, 100, 30), vSliderValue, 10.0f, 0.0f);  
        //hScrollbarValue = GUI.HorizontalScrollbar (Rect (25, 25, 100, 30), hScrollbarValue, 1.0, 0.0, 10.0);  
        //vScrollbarValue = GUI. VerticalScrollbar (new Rect (25, 25, 100, 30), vScrollbarValue, 1.0f, 10.0f,  
        0.0f);  
    }  
}
```

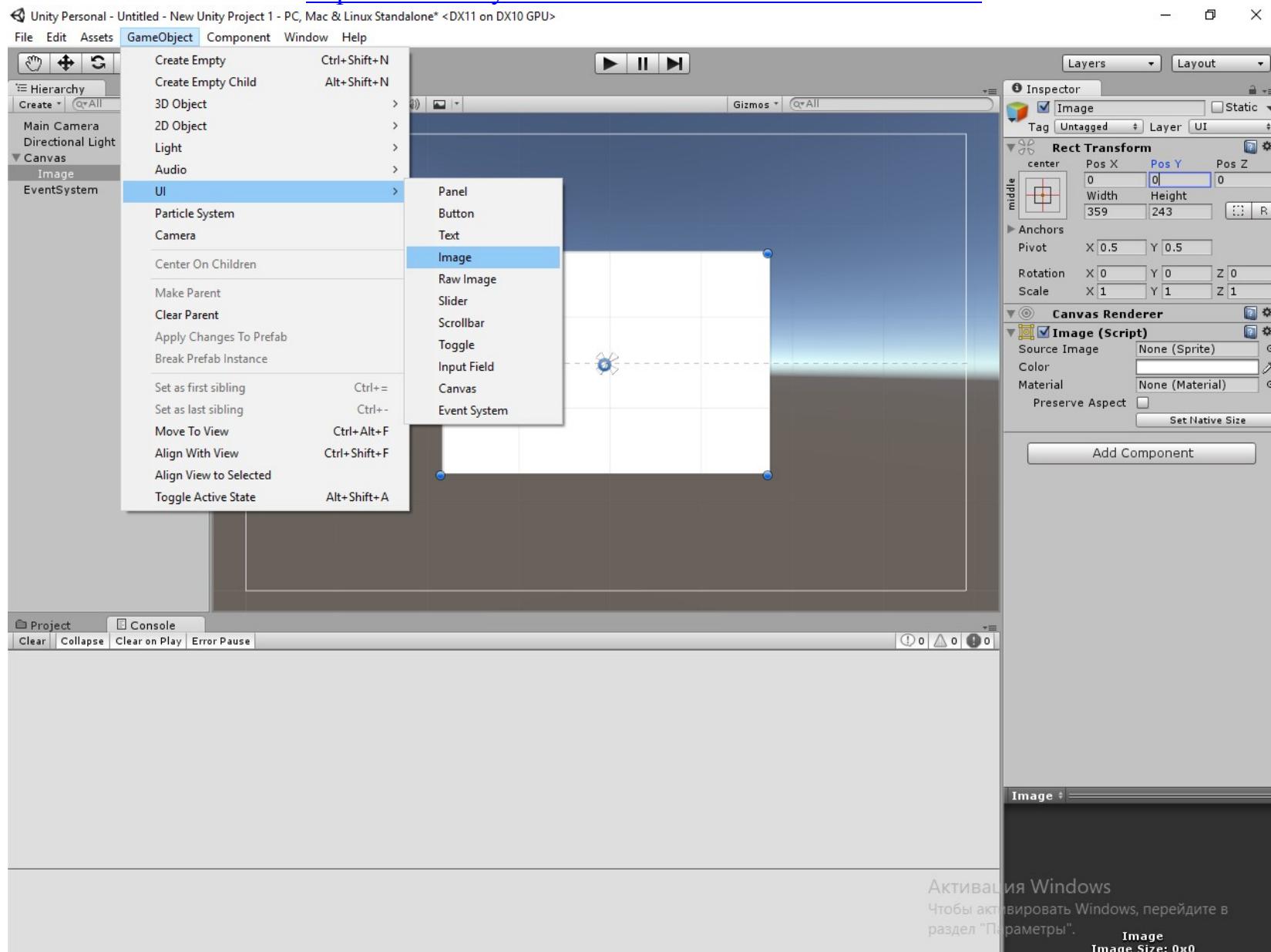


```
public class GUITest : MonoBehaviour {
    private Vector2 scrollViewVector = Vector2.zero;
    private string innerText = "I am inside the ScrollView";
    void OnGUI () {
        scrollViewVector = GUI.BeginScrollView (new Rect (25, 25, 100, 100), scrollViewVector, new Rect (0, 0, 400, 400));
        innerText = GUI.TextArea (new Rect (0, 0, 400, 400), innerText);
        // End the ScrollView
        GUI.EndScrollView();
    }
}
```

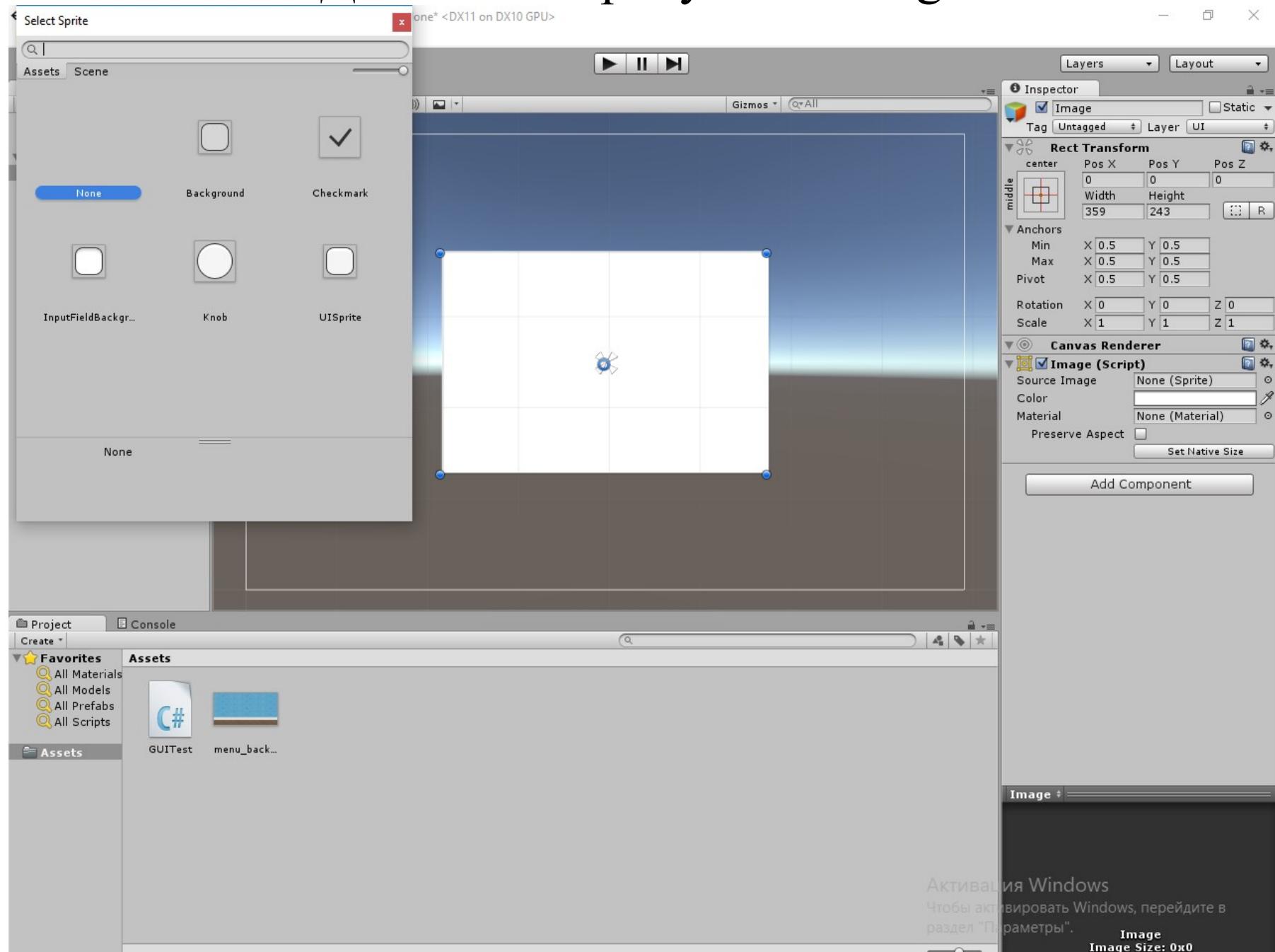


# Розробка GUI з використанням нової системи

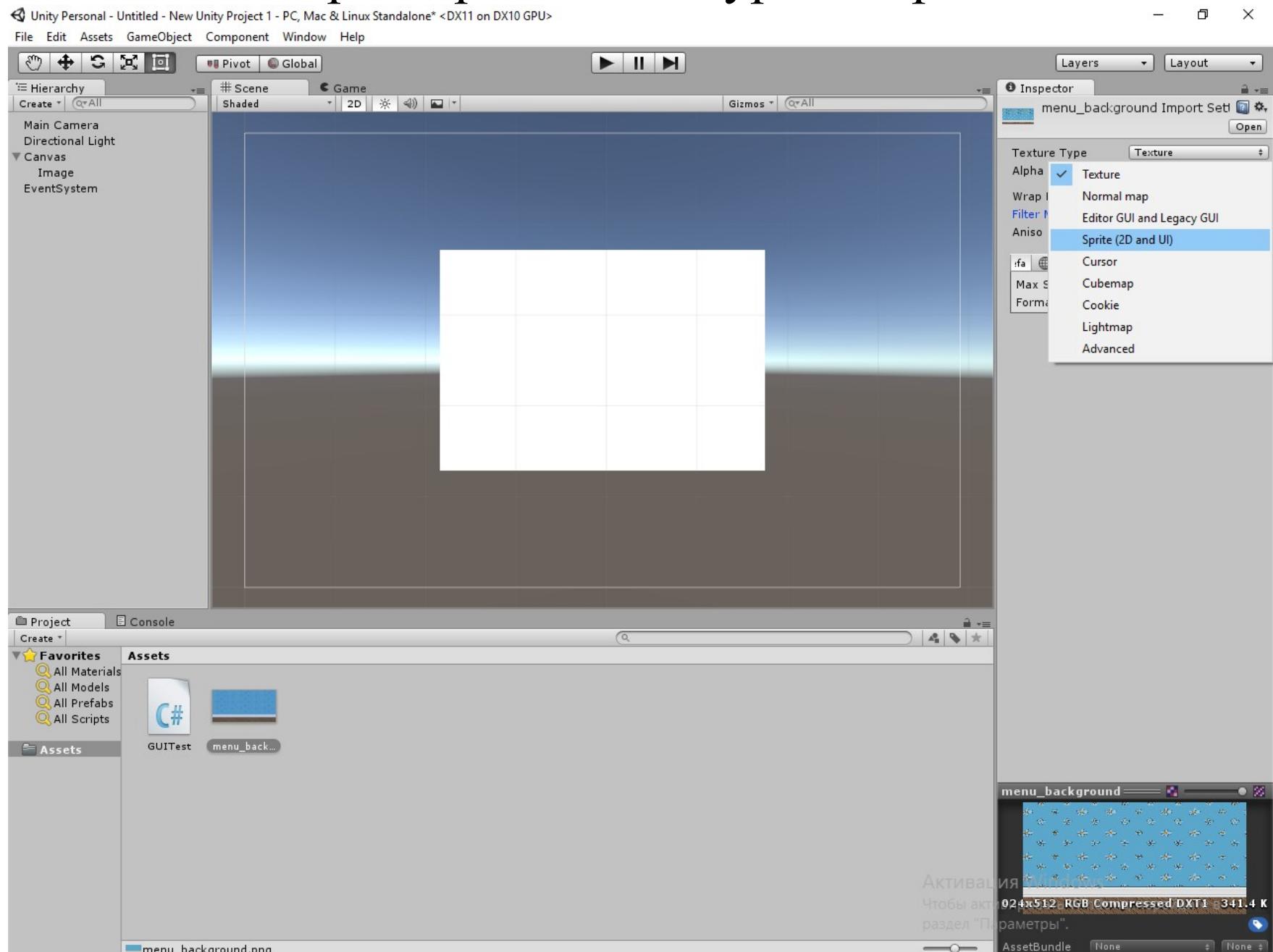
<https://docs.unity3d.com/ru/500/Manual/UIOverview.html>



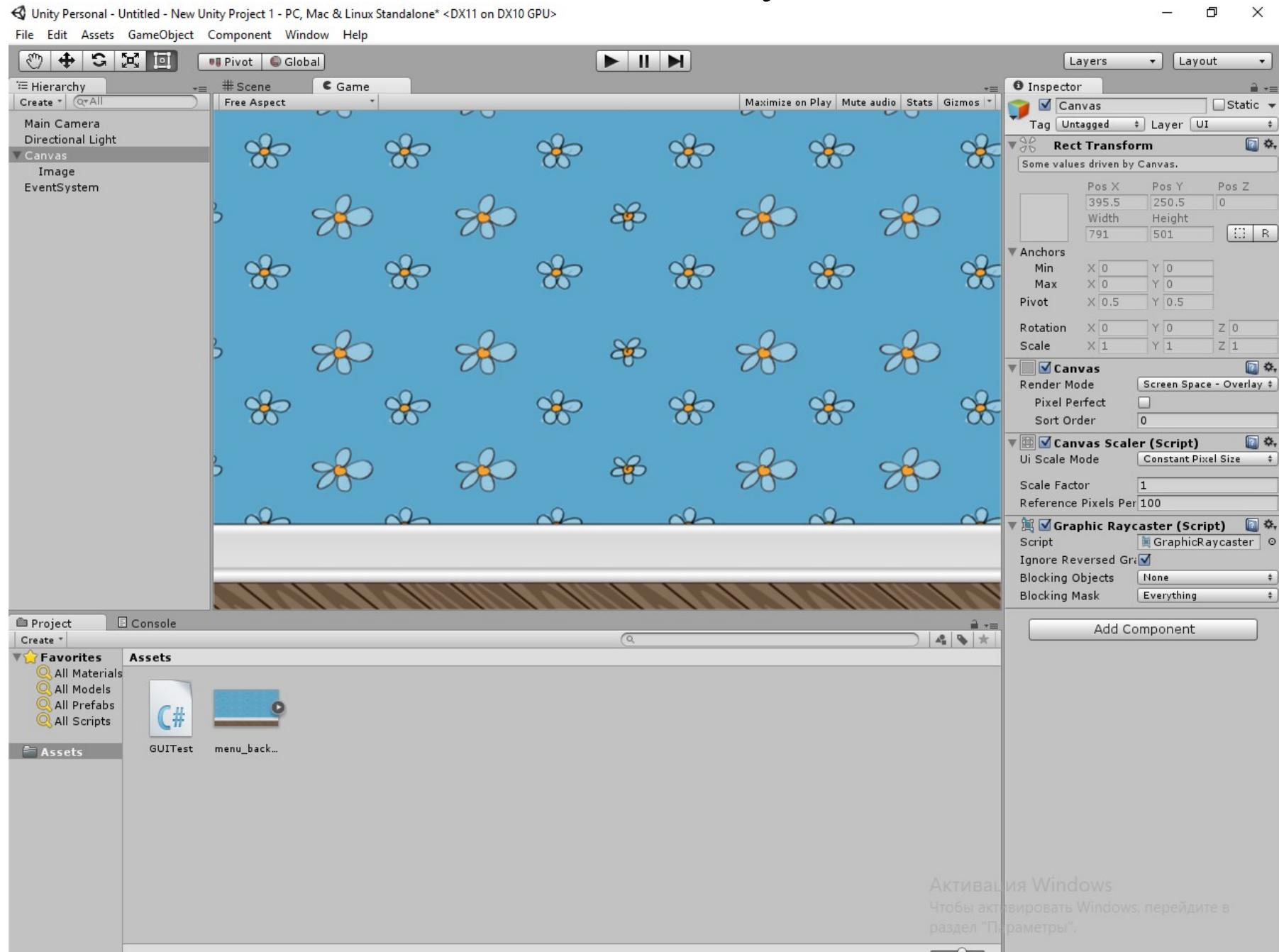
# Добавлення рисунка в Image

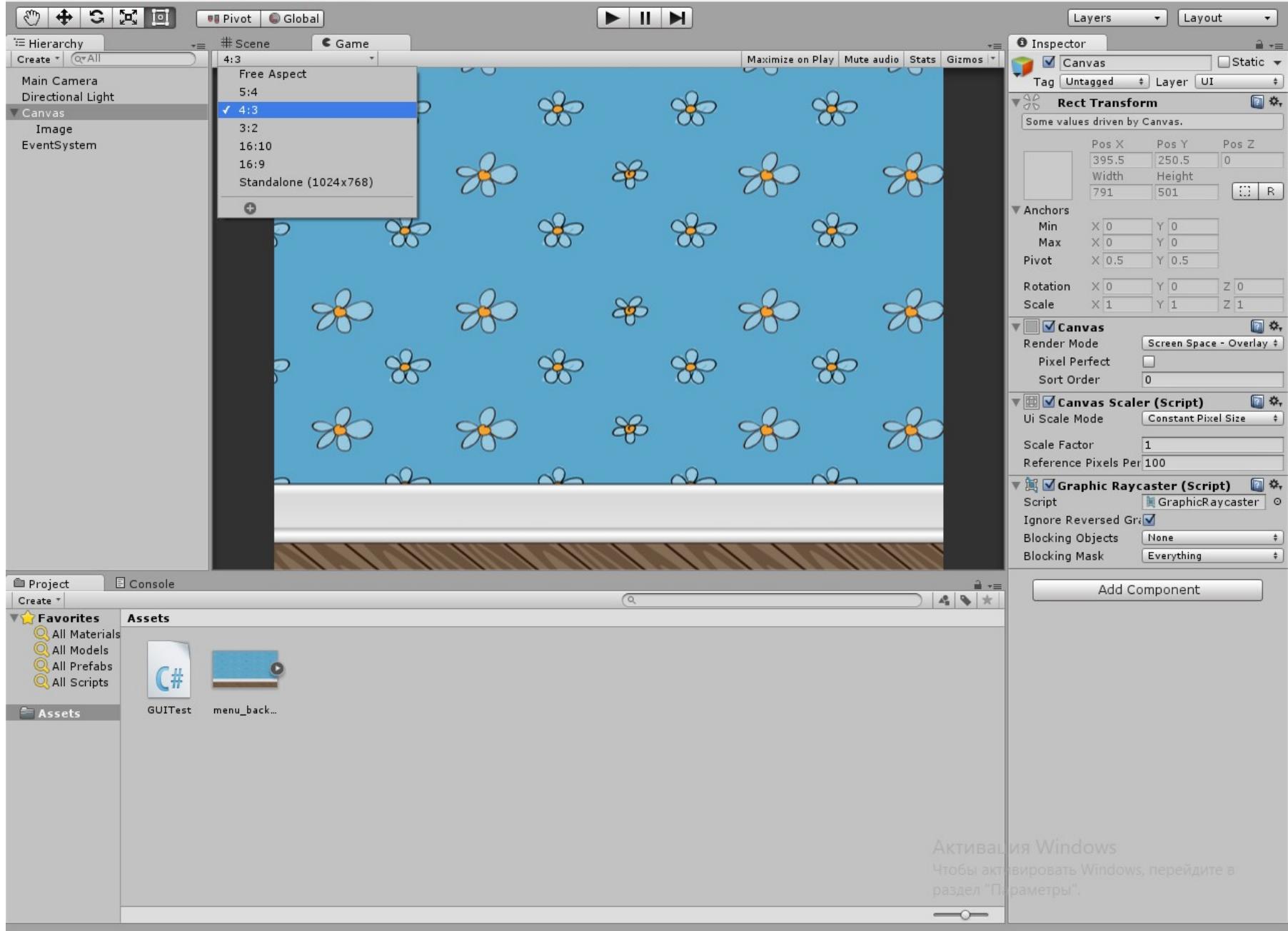


# Перетворення текстури в спрайт

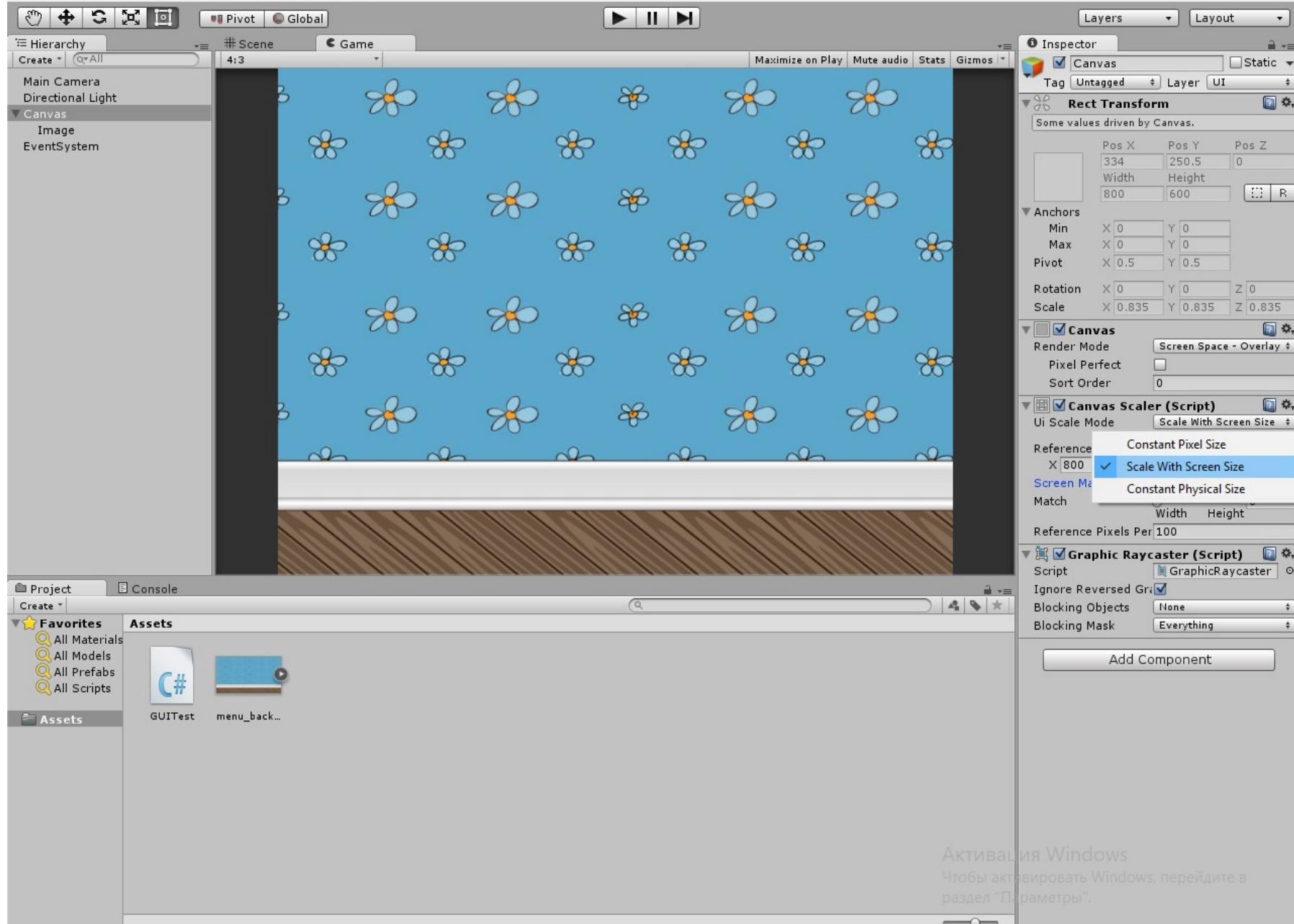


# Режими масштабування

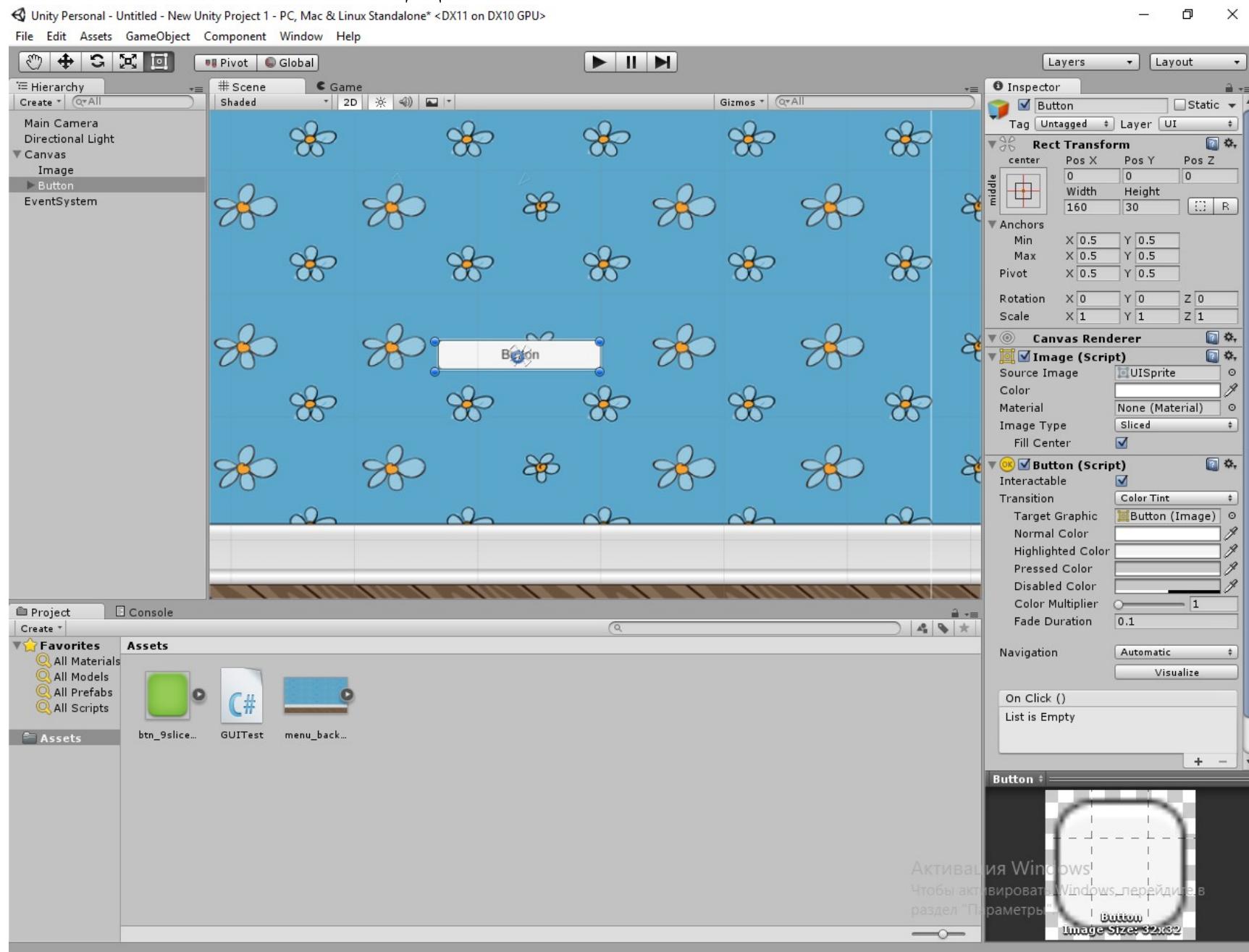




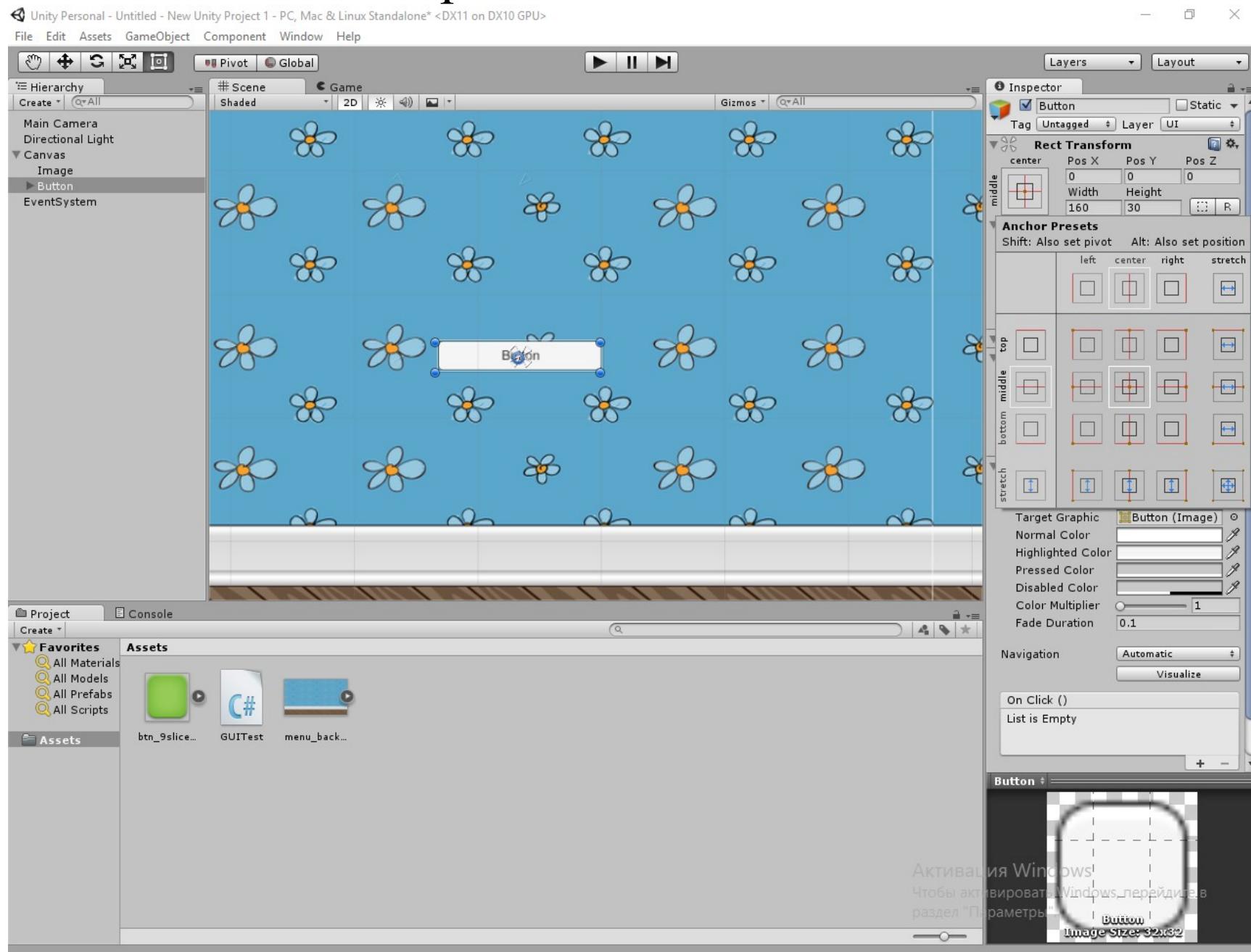
File Edit Assets GameObject Component Window Help



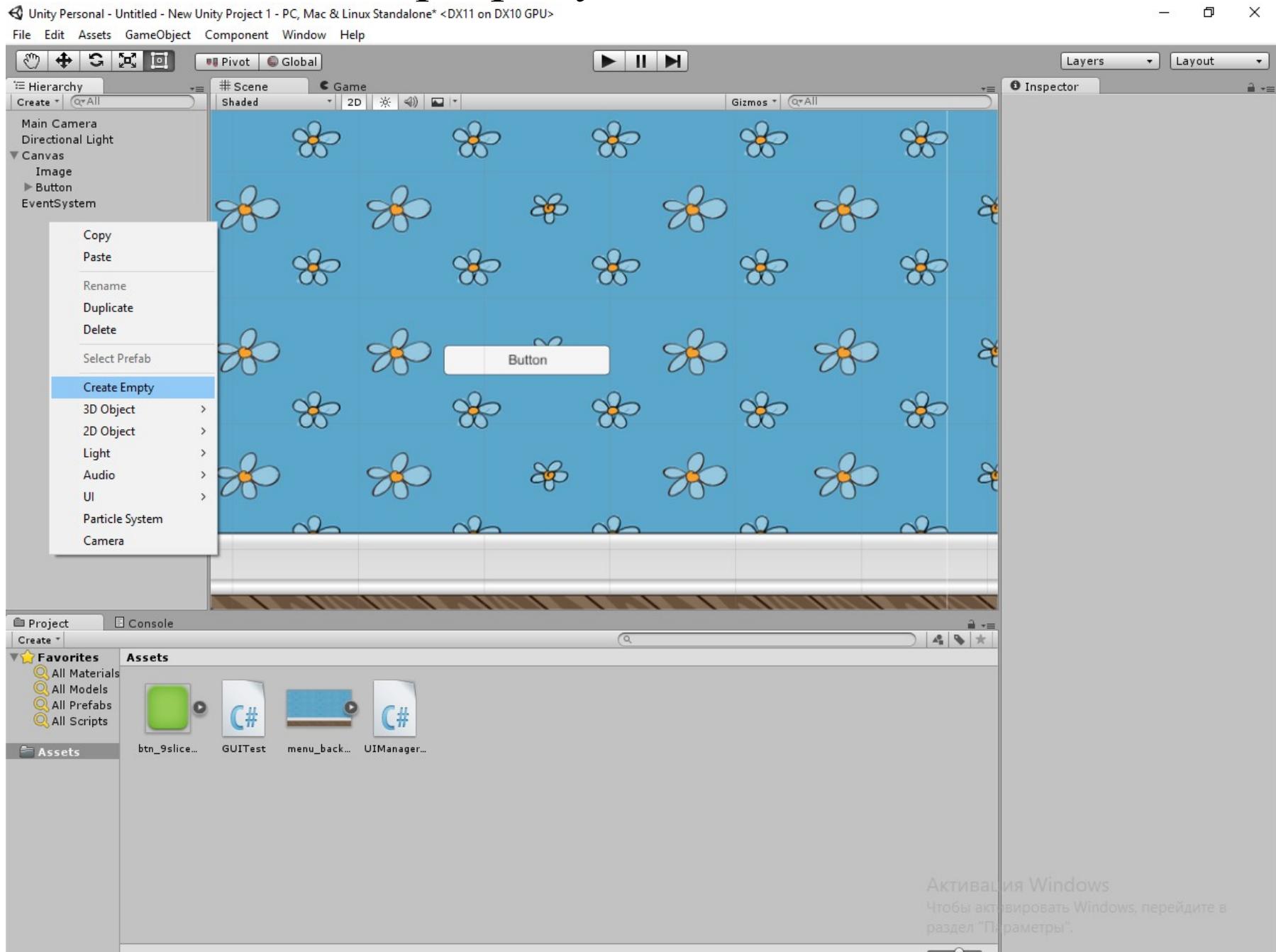
# Добавлення кнопки



# Вирівнювання кнопки



# Програмування кнопки



File Edit Assets GameObject Component Window Help



Pivot Global



Layers Layout

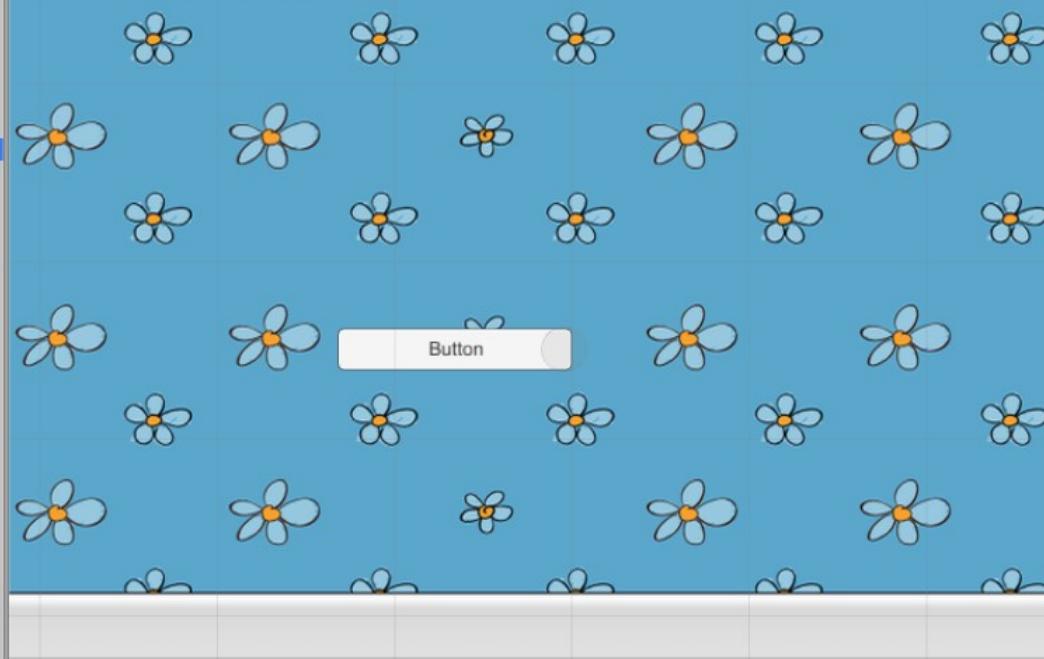
Hierarchy Create Q All

Main Camera  
Directional Light  
Canvas  
Image  
Button  
EventSystem  
UIManager

# Scene Game

Shaded 2D

Gizmos Q All



Inspector Layers Layout

UIManager Static

Tag Untagged Layer Default

Transform Position X 395.5 Y 250.5 Z 0  
Rotation X 0 Y 0 Z 0  
Scale X 1 Y 1 Z 1

UI Manager Script (Script) UIManagerScript

Script UIManagerScript

Add Component

Project Create

Favorites

All Materials  
All Models  
All Prefabs  
All Scripts

Assets



btn\_9slice... GUITest menu\_back... UIManager...

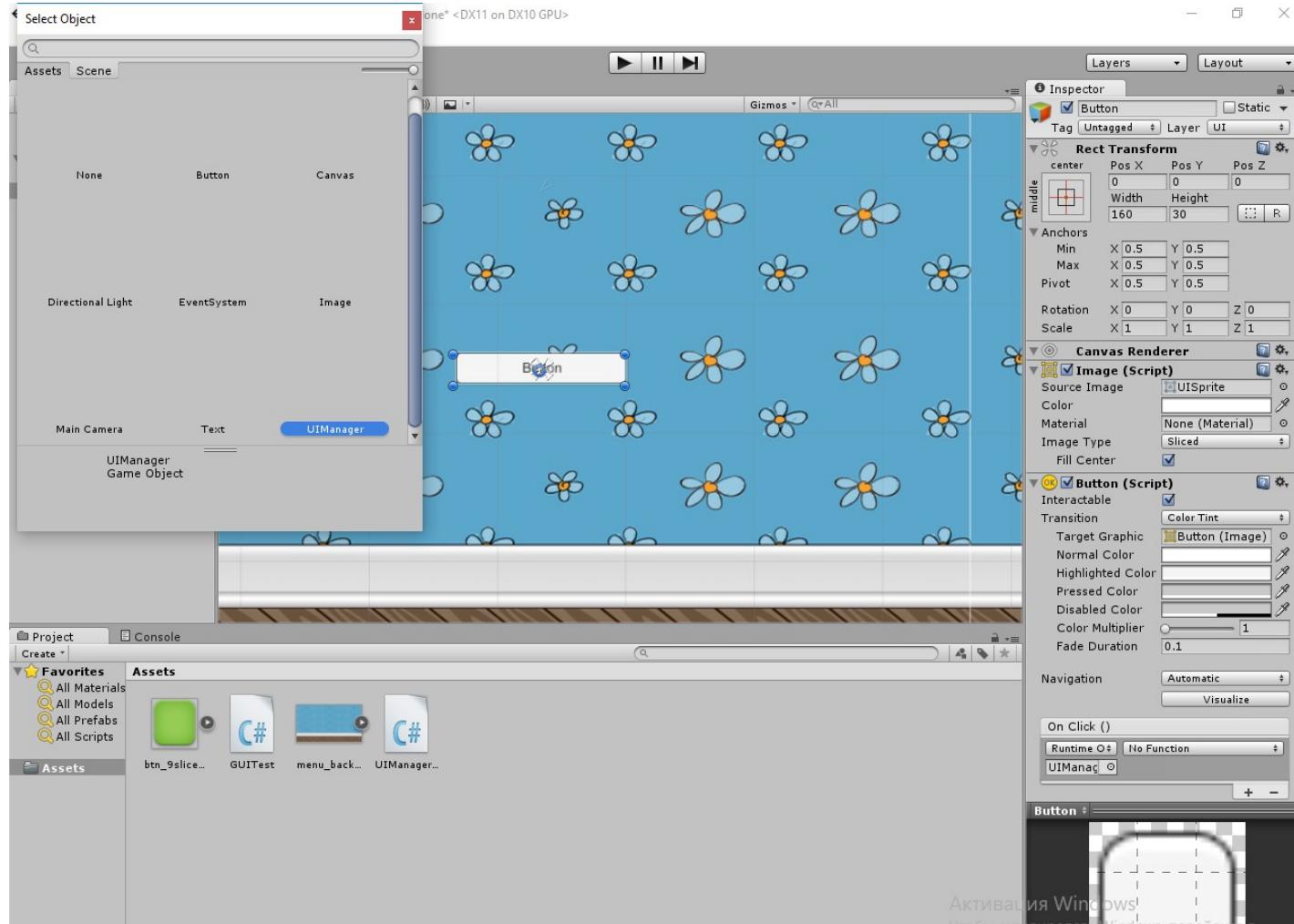
Assets

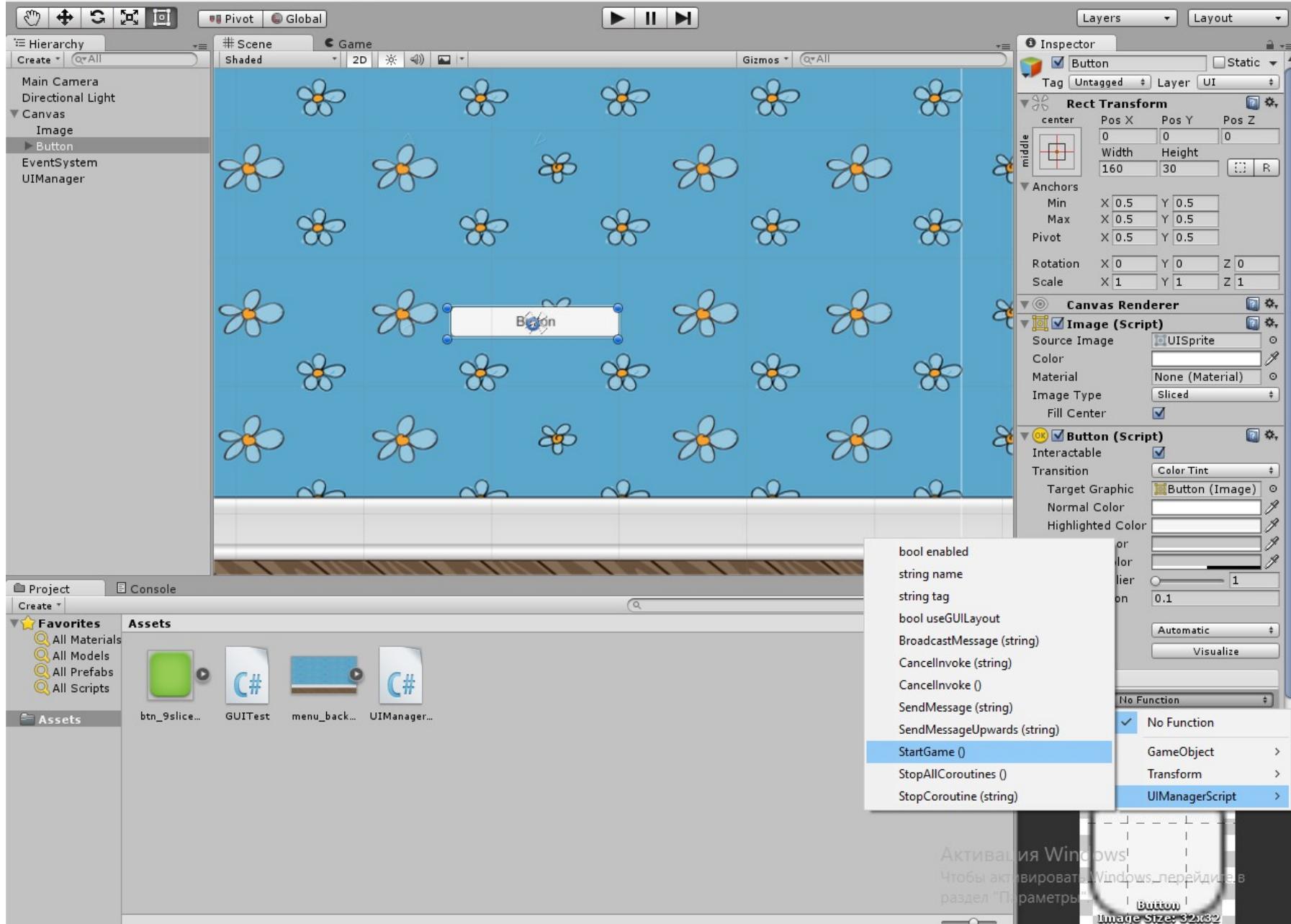
Активация Windows  
Чтобы активировать Windows, перейдите в раздел "Параметры".

```

using UnityEngine;
using System.Collections;
public class UIManagerScript : MonoBehaviour {
    public void StartGame() {
        //Application.LoadLevel("RocketMouse");
        print("Start");
    }
}

```





Дякую за увагу!



# Тема 6. Керування гравцем. Тригери

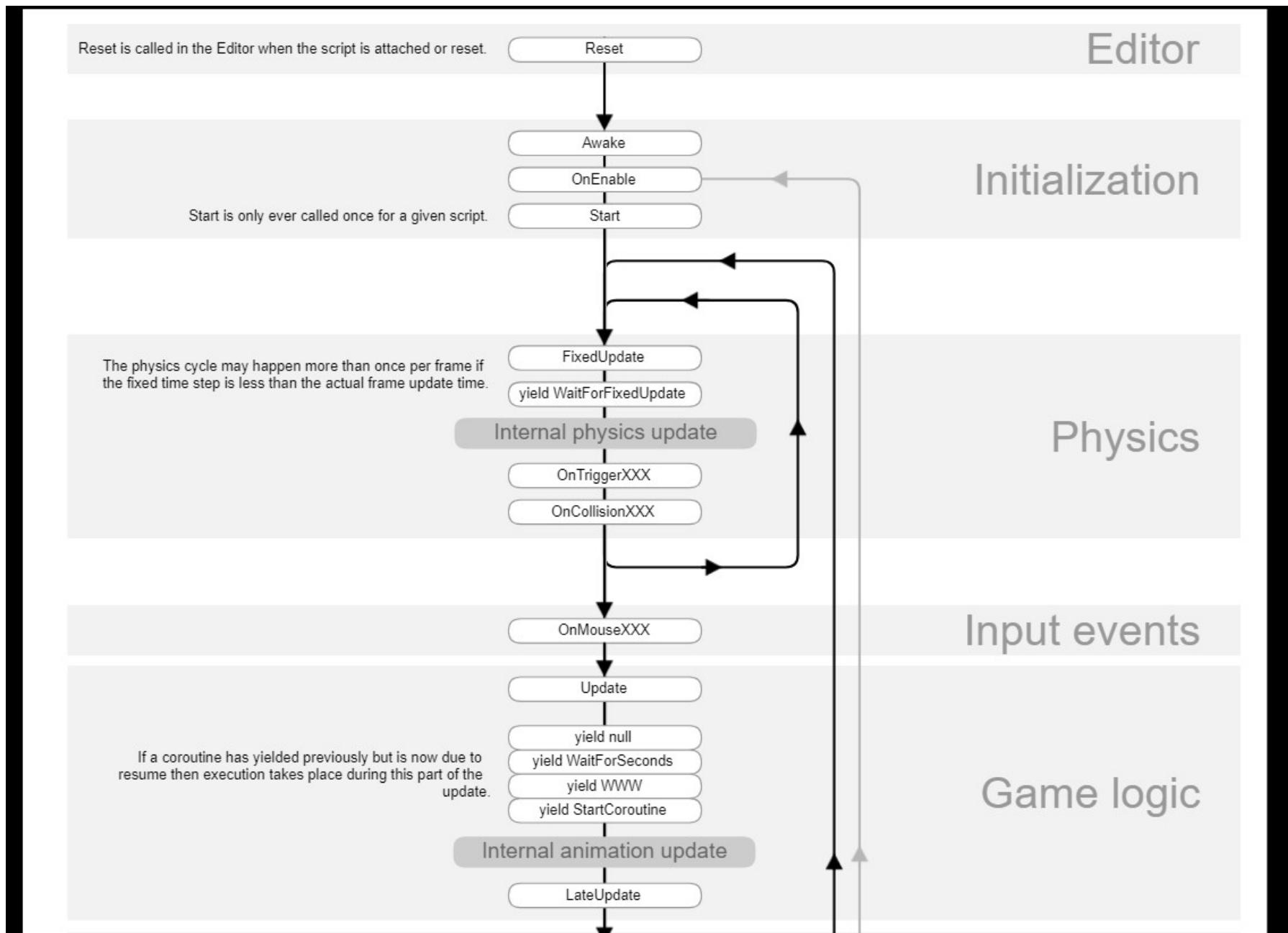
The screenshot shows the Unity Editor interface. On the left, the Project pane lists various scripts: Character, AutoBackToTitle.cs, ClickToStart.cs, Explosion.cs, Explosive.cs, Fire.cs, FloorSection.cs, GameControl.cs, GameGUI.cs, Hose.cs, MapIcons.cs, MessageGUI.cs, MoveBetweenPoints, Player.cs (selected), Priority Particle Add., PriorityAlphaParticle, SceneChanger.cs, SmokeParticles.cs, WaterHoseParticles, WaterSplash.cs, and World.cs. On the right, the Player.cs script is displayed in the Editor window:

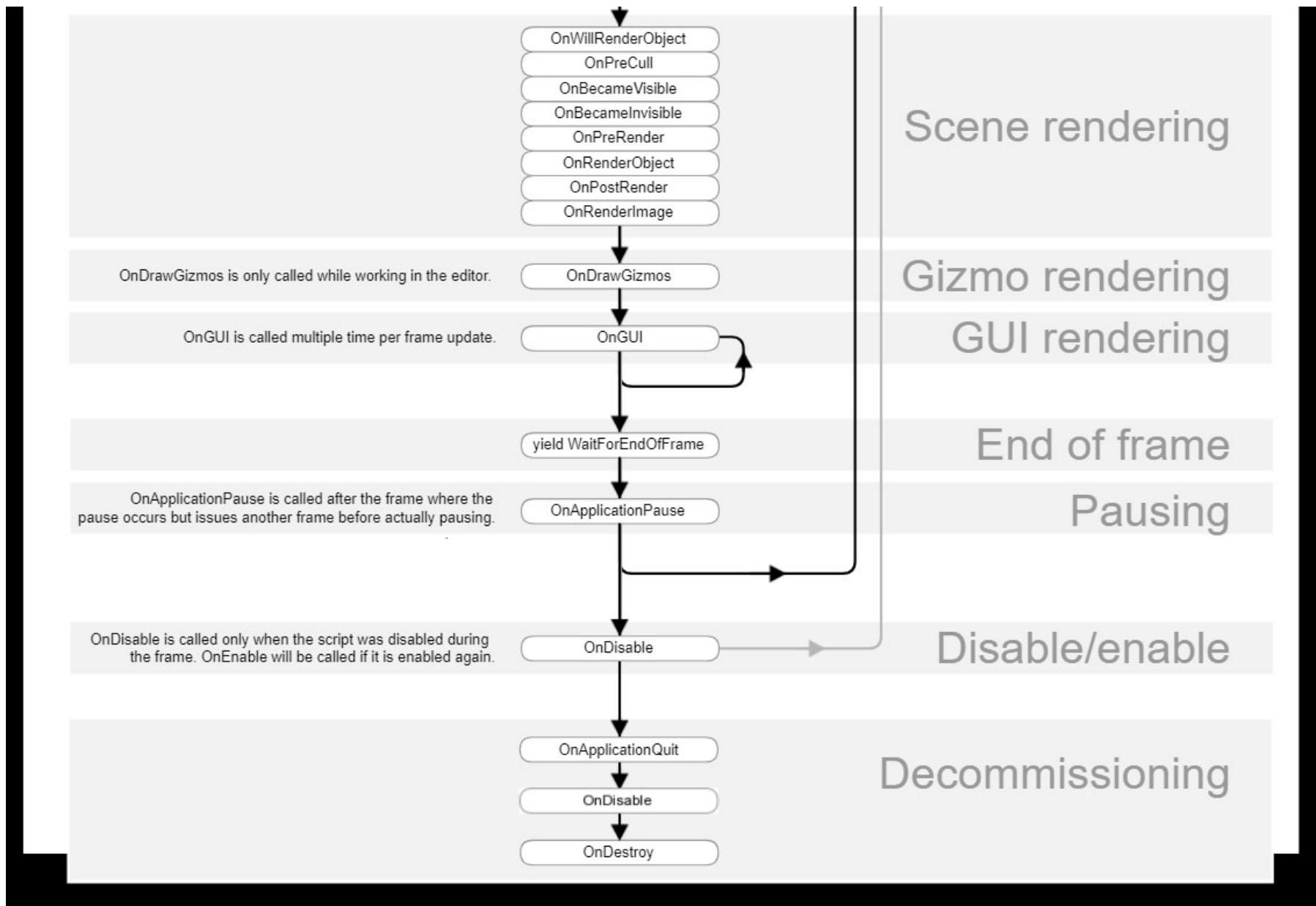
```
50     vignette.blur = (1-health) * 2 + smokeEffect * 30 * (health - 1);
51     vignette.blurDistance = (1-health) * 2 + smokeEffect * 30 * (health - 1);
52     vignette.chromaticAberration = heatEffect * 30;
53 }
54
55
56 void OnTriggerEnter(Collider c)
57 {
58     var fire = c.GetComponent<Fire>();
59     if (fire && fire.alive)
60     {
61         float dist = 1 - ((transform.position - fire.transform.position).magnitude);
62         NearHeat(dist);
63     }
64
65     var smoke = c.GetComponent<ClickToStart>();
66     if (smoke && smoke.GetComponent<ParticleEmission>().alive)
67     {
68         float dist = 1 - ((transform.position - smoke.transform.position).magnitude);
69         NearSmoke(dist);
70     }
71 }
72
73 void OnCollisionEnter(Collision c)
74 {
75     var healthBox = c.gameObject.GetComponent<HealthBox>();
76     if (healthBox)
77     {
78         if (healthBox.health <= 0)
```

<https://docs.unity3d.com/ru/current/Manual/ScriptingSection.html>

<https://docs.unity3d.com/ru/current/ScriptReference/index.html>

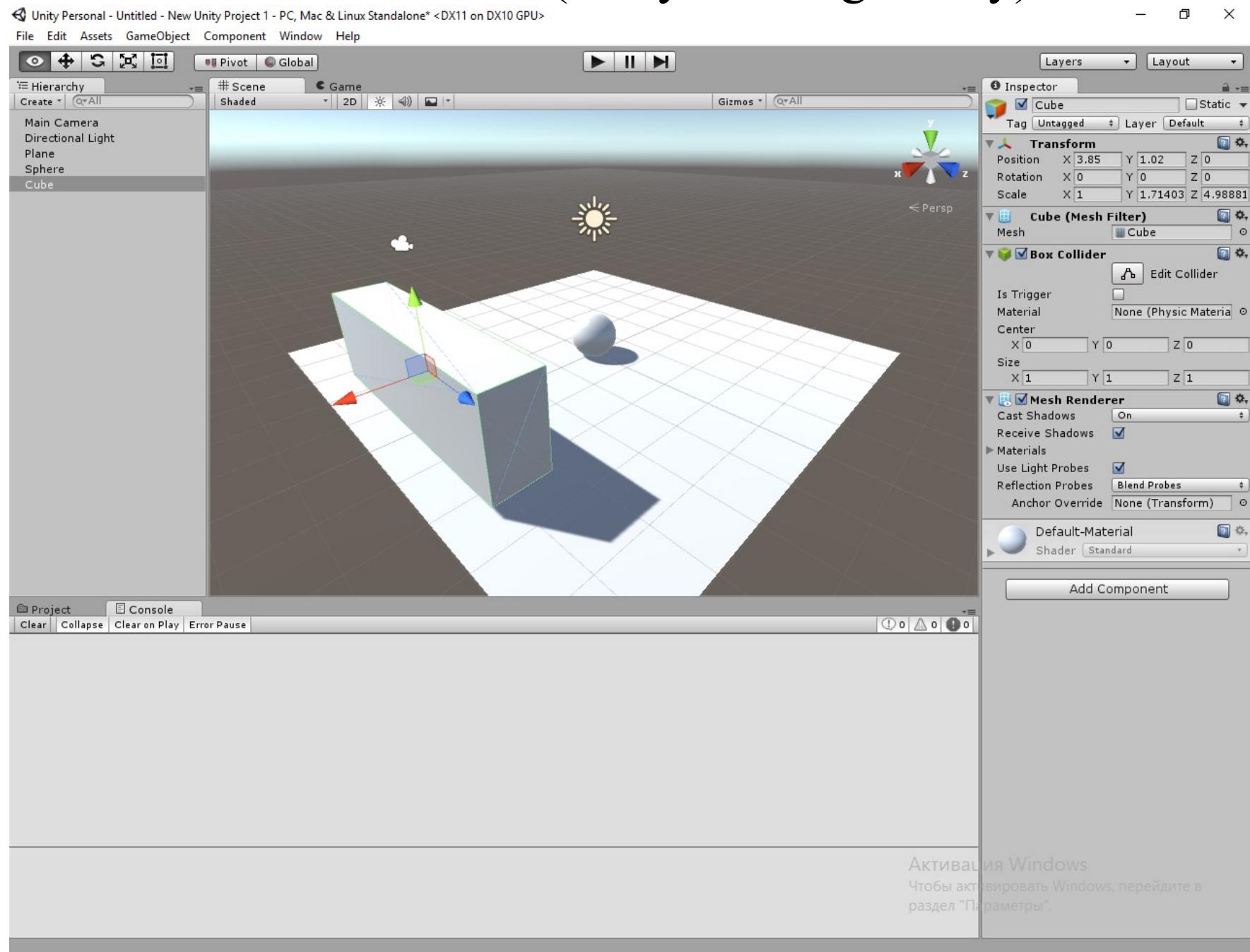
# БЛОК-СХЕМА ЖИТТЕВОГО ЦИКЛУ СКРИПТА





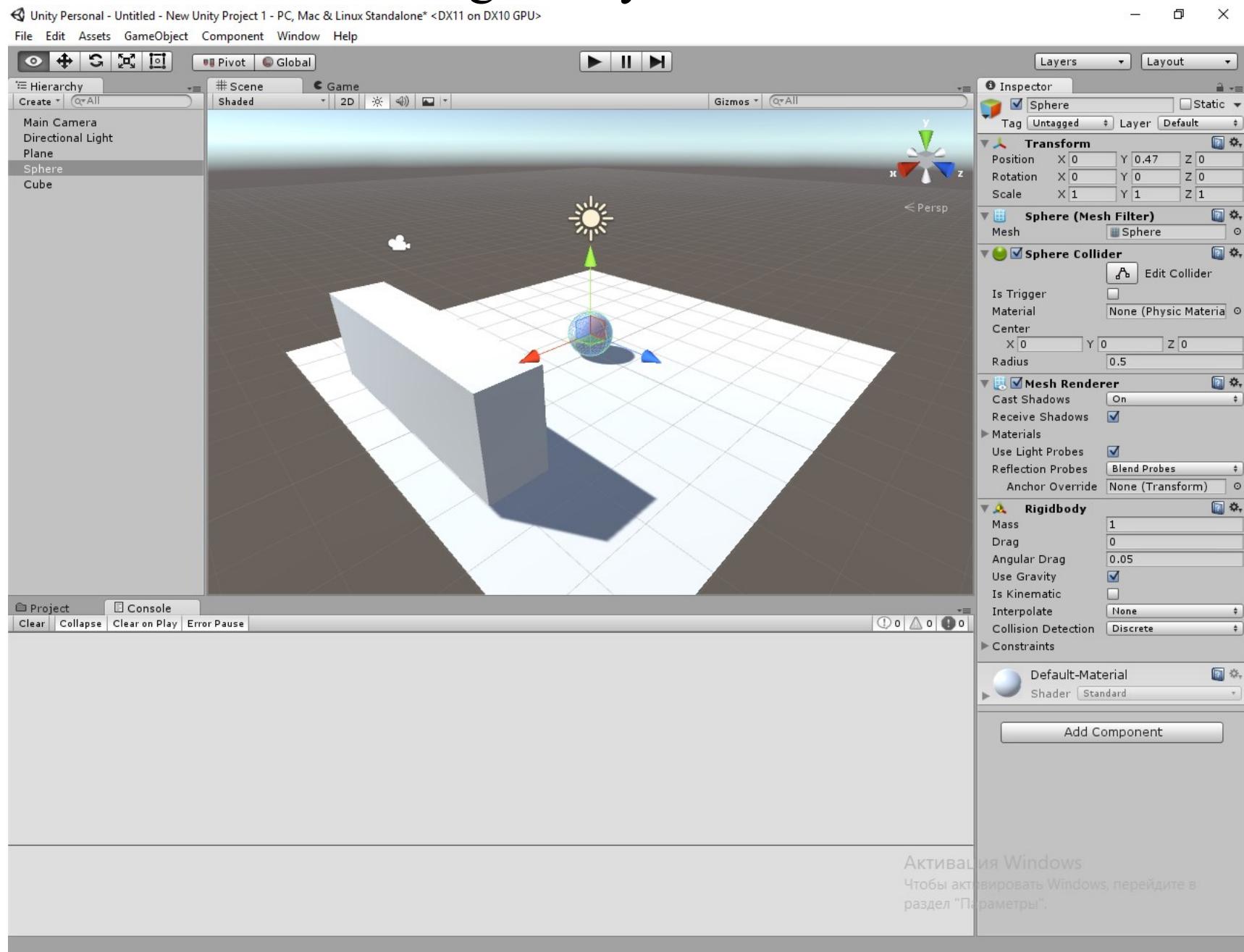
<https://docs.unity3d.com/ru/current/Manual/ExecutionOrder.html>

# Static Collider (відсутній Rigidbody)

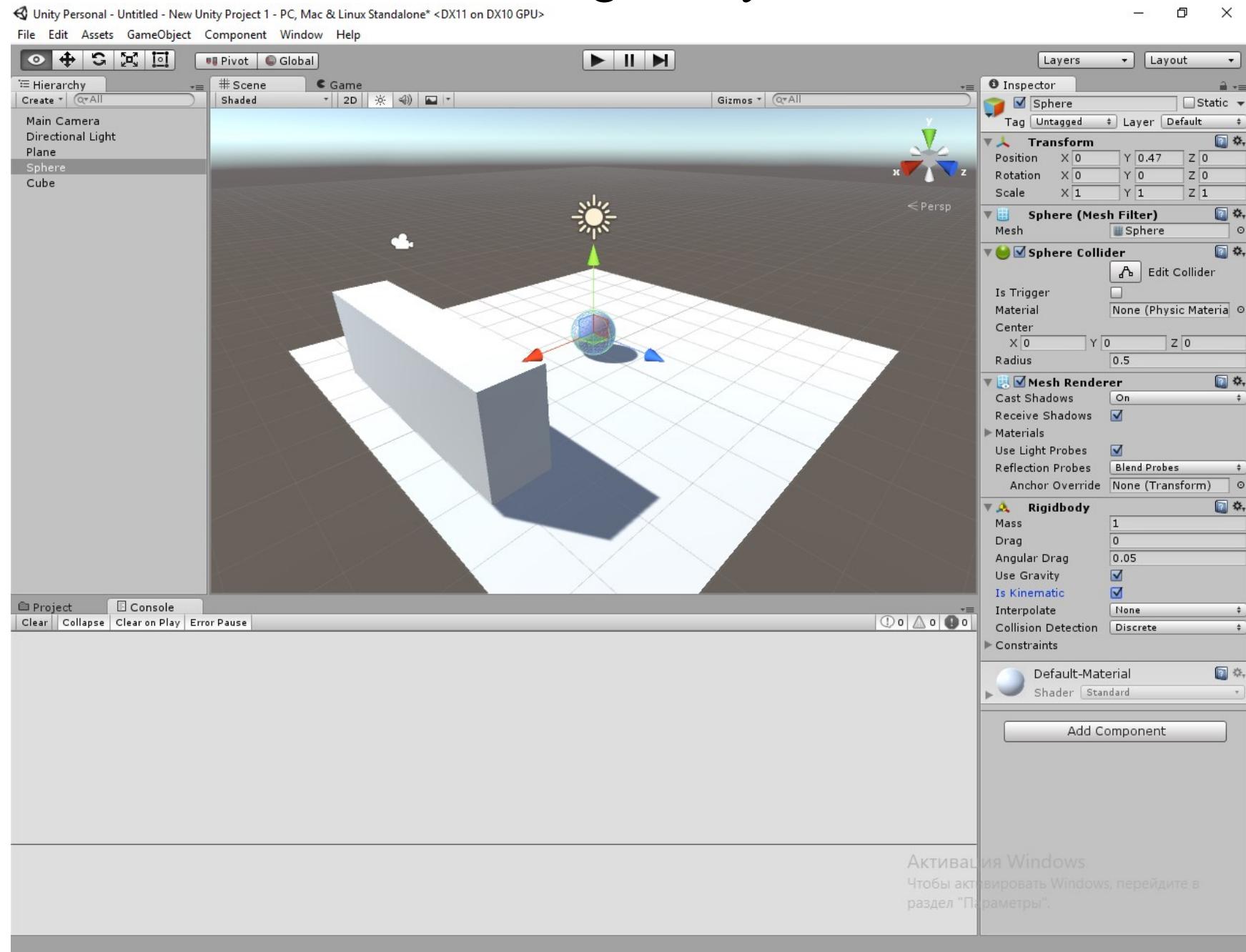


Активация Windows  
Чтобы активировать Windows, перейдите в раздел "Параметры".

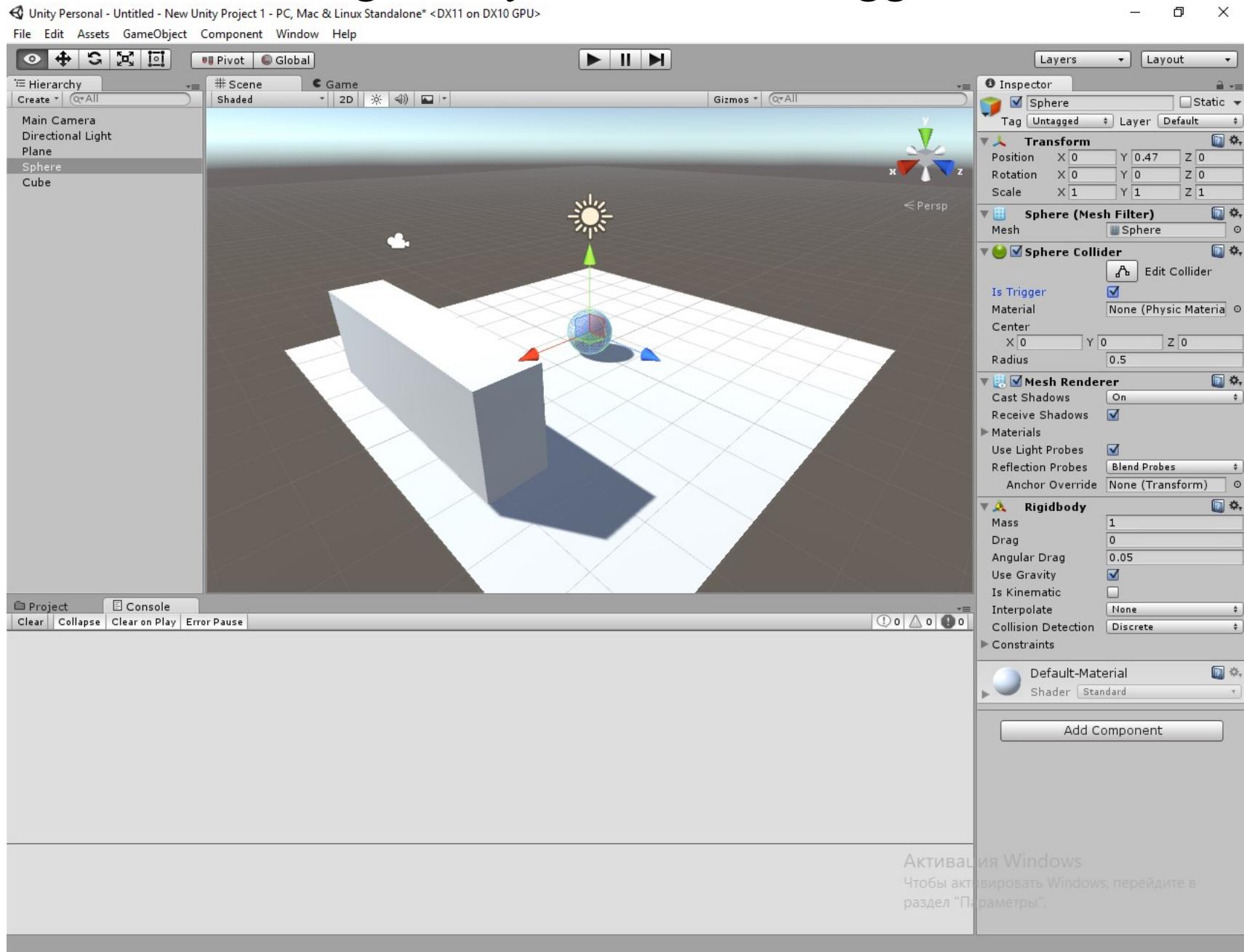
# Rigidbody Collider



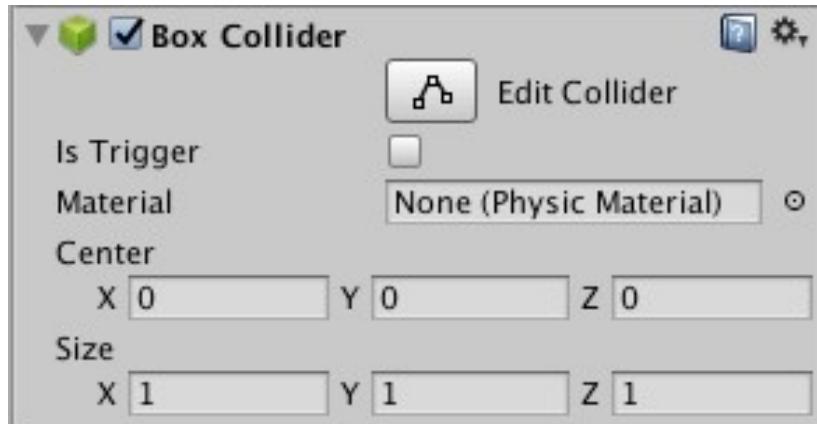
# Kinematic Rigidbody Collider



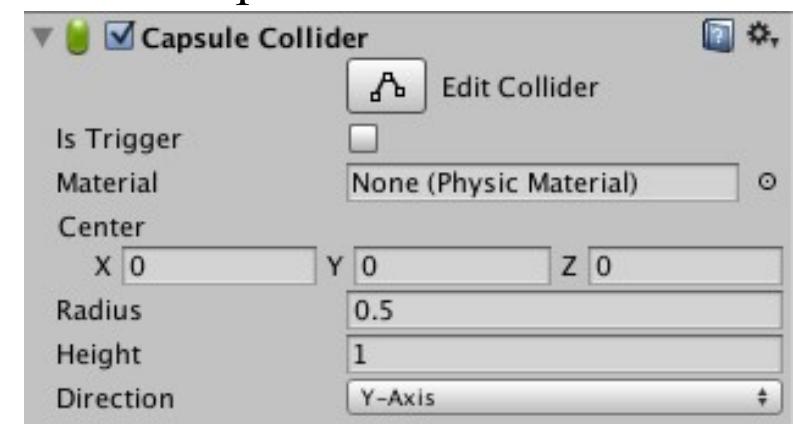
# Rigidbody Collider Is Trigger



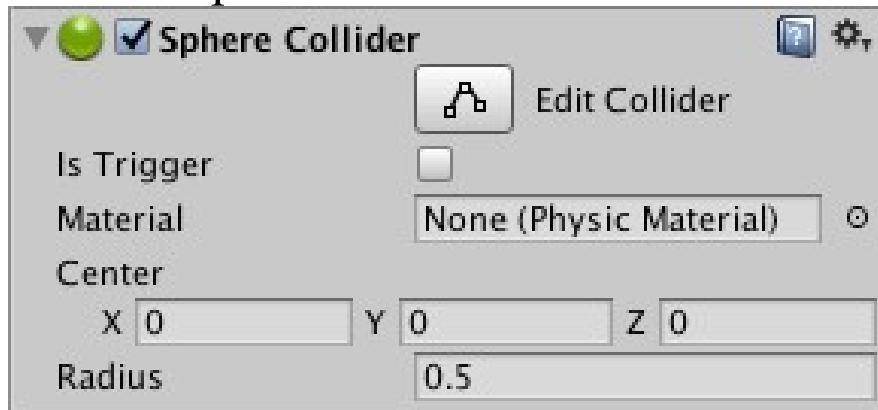
## Box Collider



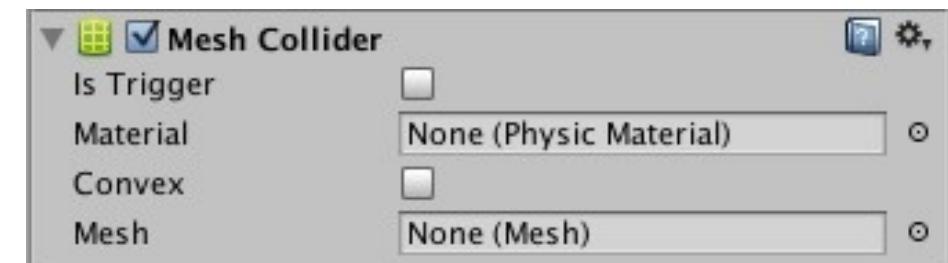
## Capsule Collider



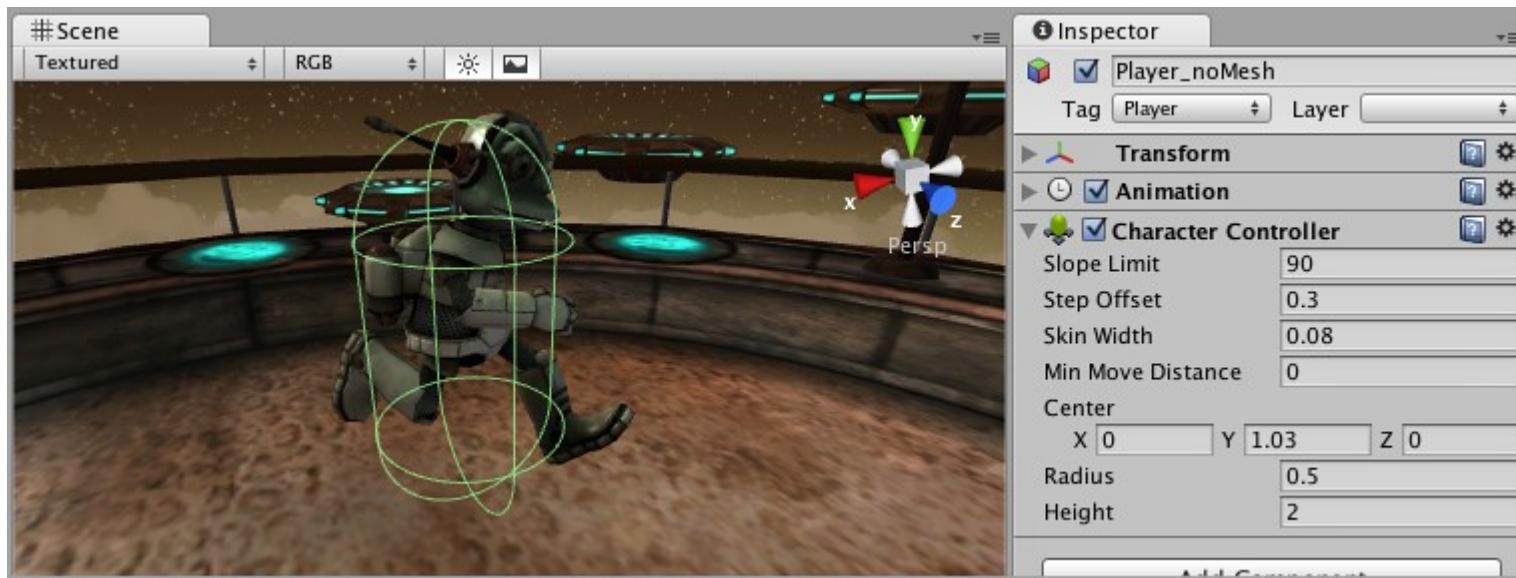
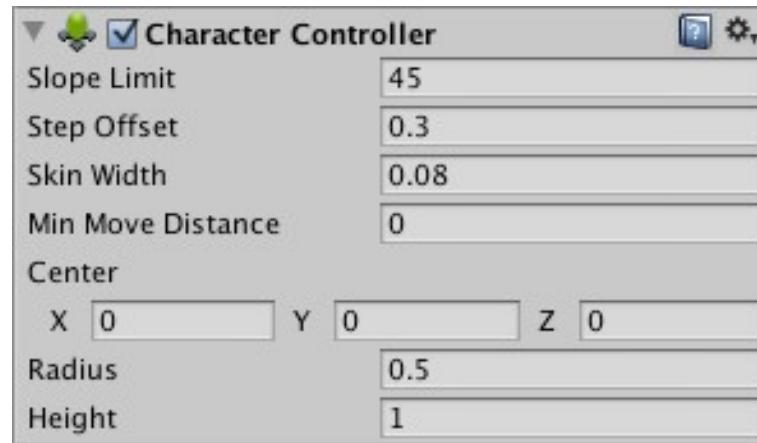
## Sphere Collider



## Mesh Collider



# Character Controller



<https://docs.unity3d.com/ru/current/ScriptReference/CharacterController.html>

# Код переміщення об'єкту з використанням CharacterController

```
using UnityEngine;
using System.Collections;
public class ExampleClass : MonoBehaviour {
    public float speed = 6.0F;
    public float jumpSpeed = 8.0F;
    public float gravity = 20.0F;
    private Vector3 moveDirection = Vector3.zero;
    void Update() {
        CharacterController controller = GetComponent<CharacterController>();
        if(controller.isGrounded) {
            moveDirection = new Vector3(Input.GetAxis("Horizontal"), 0, Input.GetAxis("Vertical"));
            moveDirection = transform.TransformDirection(moveDirection);
            moveDirection *= speed;
            if (Input.GetButton("Jump"))
                moveDirection.y = jumpSpeed;
        }
        moveDirection.y -= gravity * Time.deltaTime;
        controller.Move(moveDirection * Time.deltaTime);
    }
}
```

# Властивість isTrigger

```
public class Example : MonoBehaviour
{
    Collider m_ObjectCollider;

    void Start()
    {
        m_ObjectCollider = GetComponent<Collider>();
        m_ObjectCollider.isTrigger = false;
        Debug.Log("Trigger On : " + m_ObjectCollider.isTrigger);
    }

    void OnMouseDown()
    {
        m_ObjectCollider.isTrigger = true;
        Debug.Log("Trigger On : " + m_ObjectCollider.isTrigger);
    }
}
```

## Подія Collider.OnTriggerEnter(Collider)

```
using UnityEngine;
using System.Collections;

public class ExampleClass : MonoBehaviour {
    void OnTriggerEnter(Collider other) {
        if(other.tag=="BotSphere")
            Destroy(other.gameObject);
    }
}
```

## Подія Collider.OnTriggerExit(Collider)

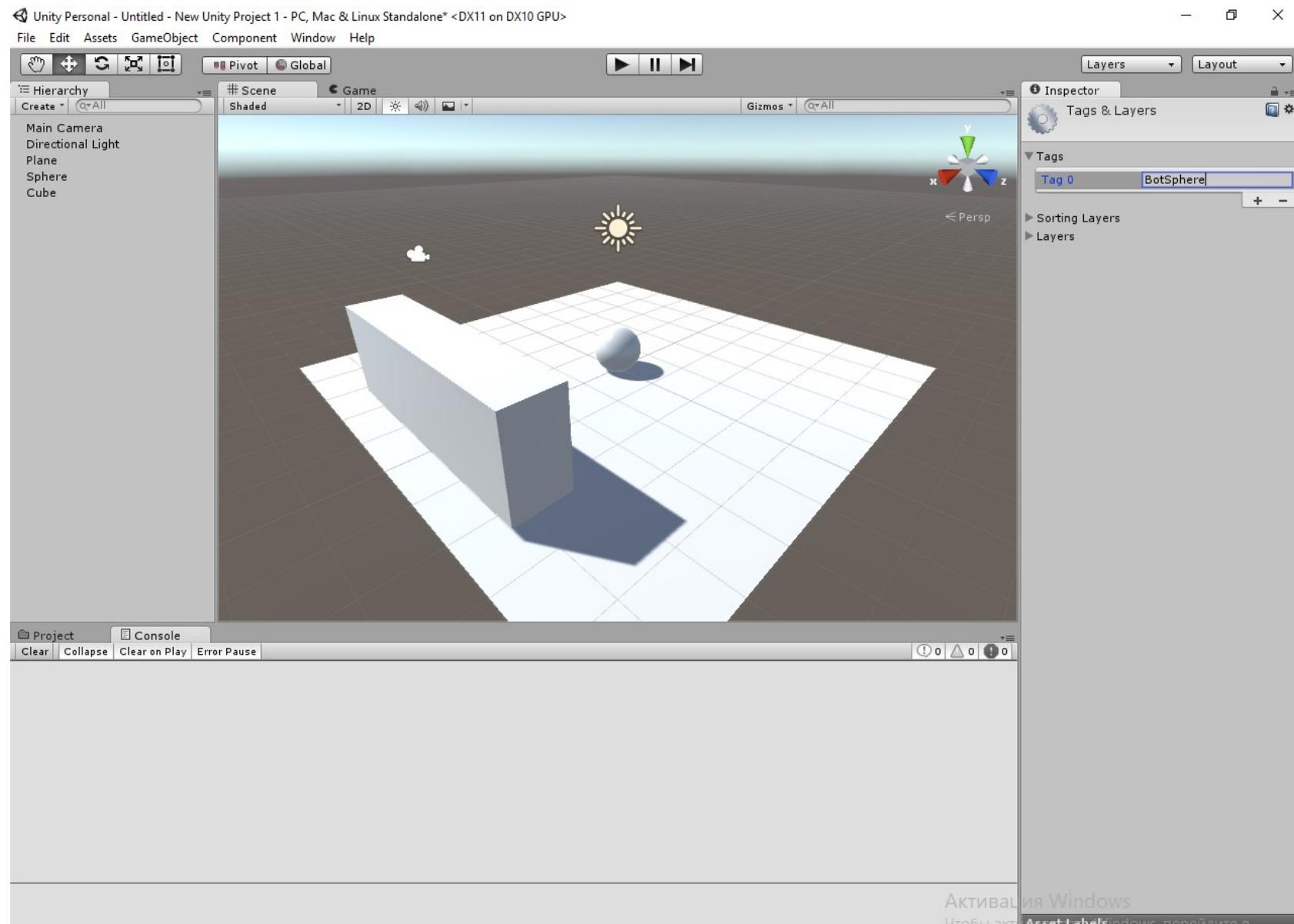
```
using UnityEngine;
using System.Collections;

public class ExampleClass : MonoBehaviour
{
    void OnTriggerExit(Collider other)
    {
        Destroy(other.gameObject);
    }
}
```

## Подія Collider.OnTriggerStay(Collider)

```
using UnityEngine;
using System.Collections;
public class ExampleClass : MonoBehaviour
{
    void OnTriggerEnter(Collider other)
    {
        if (other.attachedRigidbody)
            other.attachedRigidbody.AddForce(Vector3.up * 10);
    }
}
```

# Ініціалізація властивості Tag ігрового об'єкта





Pivot Global



## Hierarchy

Create All

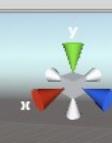
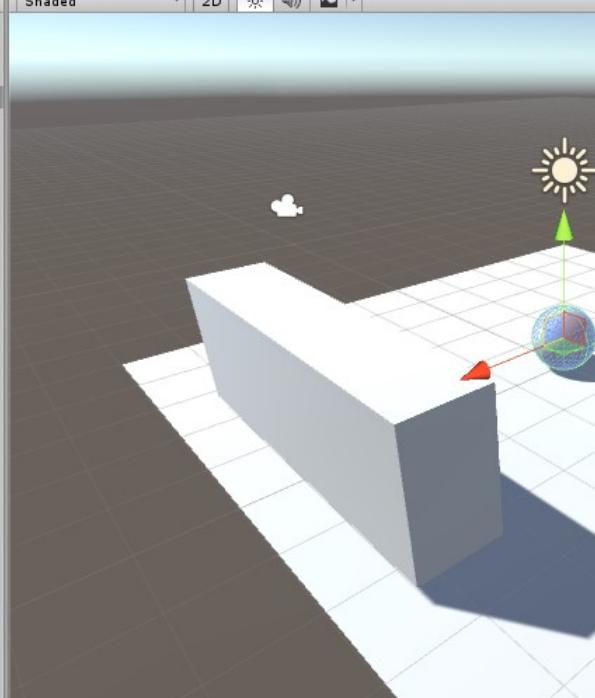
Main Camera  
Directional Light  
Plane  
**Sphere**  
Cube

## Scene

Shaded 2D

## Game

Gizmos All



## Project

## Console

Clear Collapse Clear on Play Error Pause

0 0 0

0 0 0

0 0 0

Layers Layout

## Inspector

Sphere

Static

Untagged

Respawn

Finish

EditorOnly

MainCamera

Player

GameController

Center

X 0 Y 0 Z 0

Radius 0.5

Mesh Renderer

Cast Shadows On

Receive Shadows

Materials

Use Light Probes

Reflection Probes Blend Probes

Anchor Override None (Transform)

Rigidbody

Mass 1

Drag 0

Angular Drag 0.05

Use Gravity

Is Kinematic

Interpolate None

Collision Detection Discrete

Constraints

Default-Material

Shader Standard

Add Component

Активация Windows

Чтобы активировать Windows, перейдите в раздел "Параметры".



Pivot Global



## Hierarchy

Create

Main Camera  
Directional Light  
Plane  
**Sphere**  
Cube

# Scene

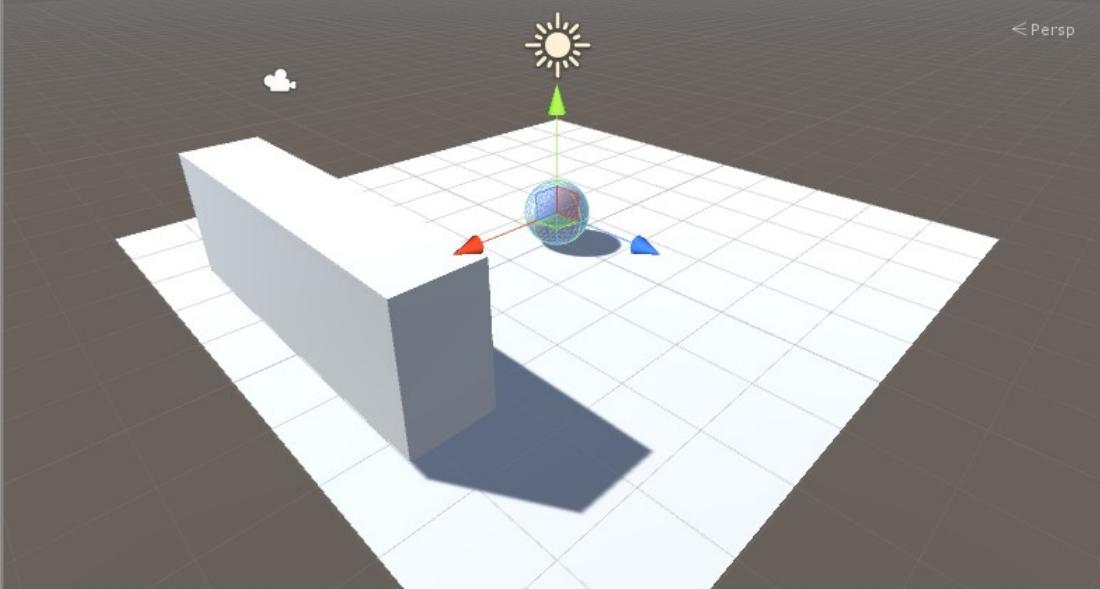
Game

Shaded 2D

Gizmos All

Persp

x y z



Project Console

Clear Collapse Clear on Play Error Pause

0 0 0

Layers Layout

## Inspector

Sphere Static

Untagged Layer Default

Untagged Respawn

Finish

EditorOnly

MainCamera

Player

GameController

BotSphere

Add Tag...

Center X 0 Y 0 Z 0

Radius 0.5

Mesh Renderer

Cast Shadows On

Receive Shadows

Materials

Use Light Probes

Reflection Probes Blend Probes

Anchor Override None (Transform)

Rigidbody

Mass 1

Drag 0

Angular Drag 0.05

Use Gravity

Is Kinematic

Interpolate None

Collision Detection Discrete

Constraints

Default-Material

Shader Standard

Add Component

Активация Windows

Чтобы активировать Windows, перейдите в раздел "Параметры".

# Подія Collider.OnCollisionEnter(Collision)

```
using UnityEngine;
using System.Collections;

public class ExampleClass : MonoBehaviour
{
    public Transform explosionPrefab;

    void OnCollisionEnter(Collision collision)
    {
        ContactPoint contact = collision.contacts[0];
        Quaternion rot = Quaternion.FromToRotation(Vector3.up, contact.normal);
        Vector3 pos = contact.point;

        Instantiate(explosionPrefab, pos, rot);
        Destroy(gameObject);
    }
}
```

## Подія Collider.OnCollisionExit(Collision)

```
using UnityEngine;
using System.Collections;

public class ExampleClass : MonoBehaviour {
    void OnCollisionExit(Collision collisionInfo) {
        print("No longer in contact with " + collisionInfo.transform.name);
    }
}
```

## Подія Collider.OnCollisionStay(Collision)

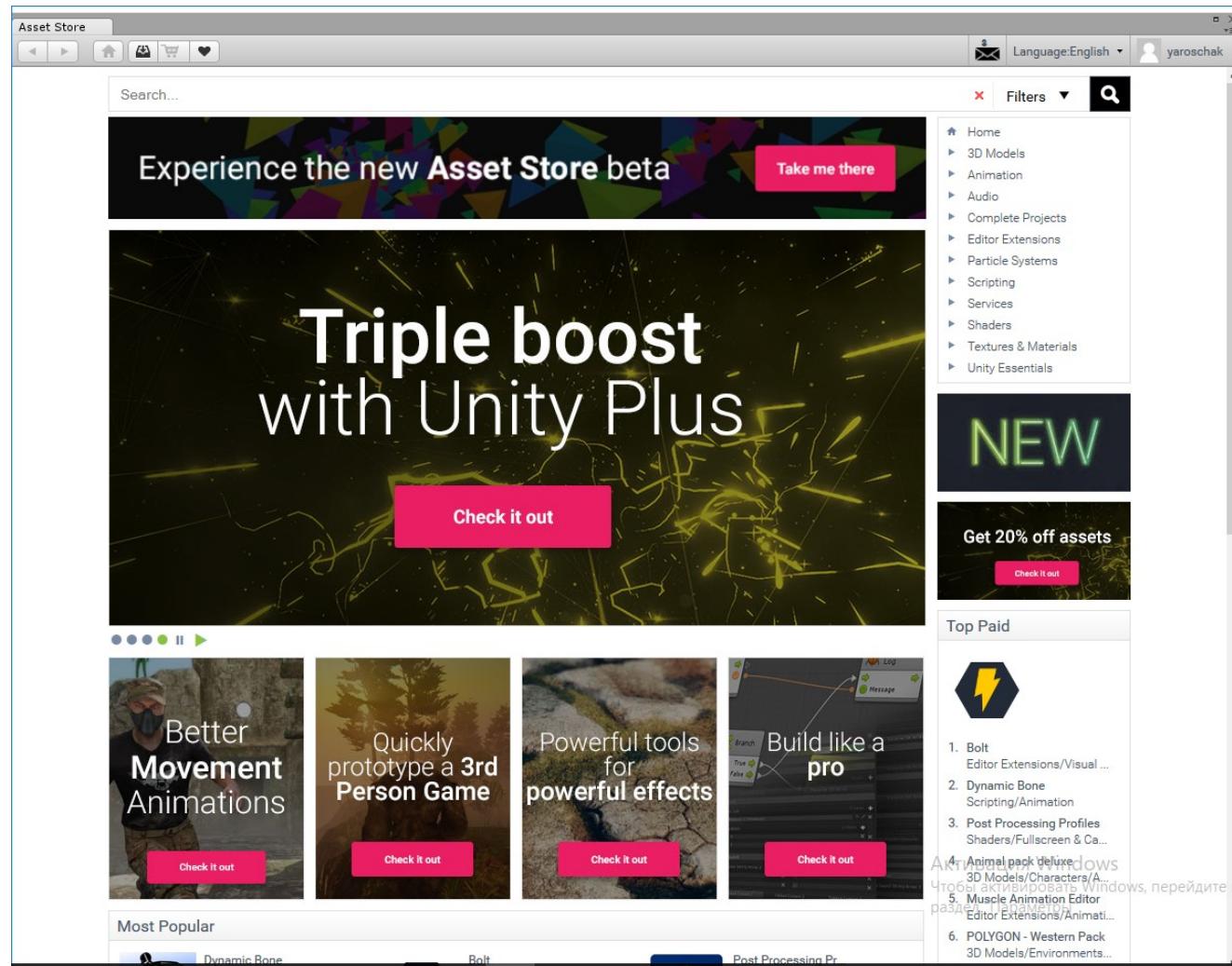
```
using UnityEngine;
using System.Collections;

public class ExampleClass : MonoBehaviour {
    void OnCollisionStay(Collision collisionInfo) {
        foreach (ContactPoint contact in collisionInfo.contacts) {
            Debug.DrawRay(contact.point, contact.normal * 10, Color.white);
        }
    }
}
```

Дякую за увагу!

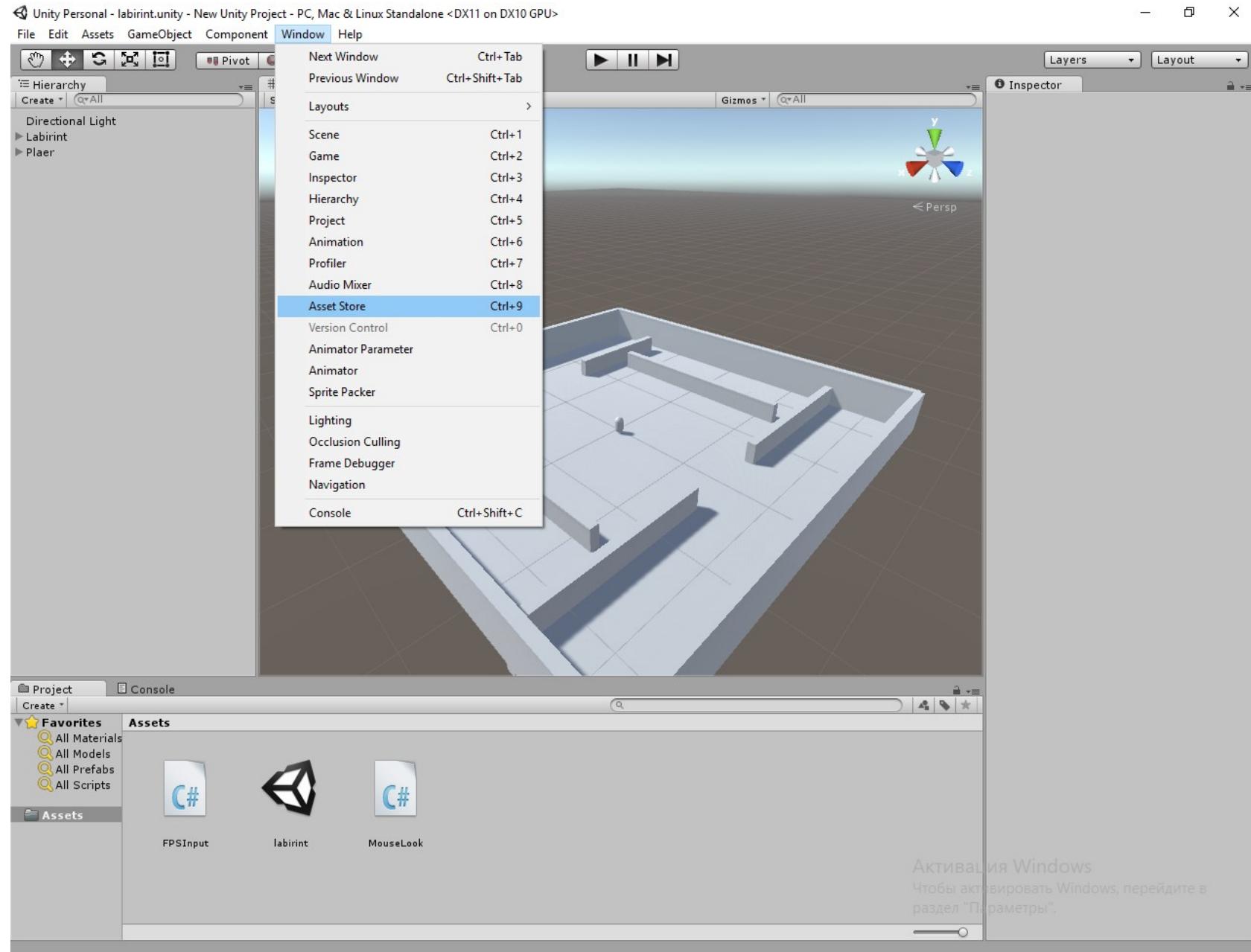


# Тема 5. Asset store та система контролю версіями

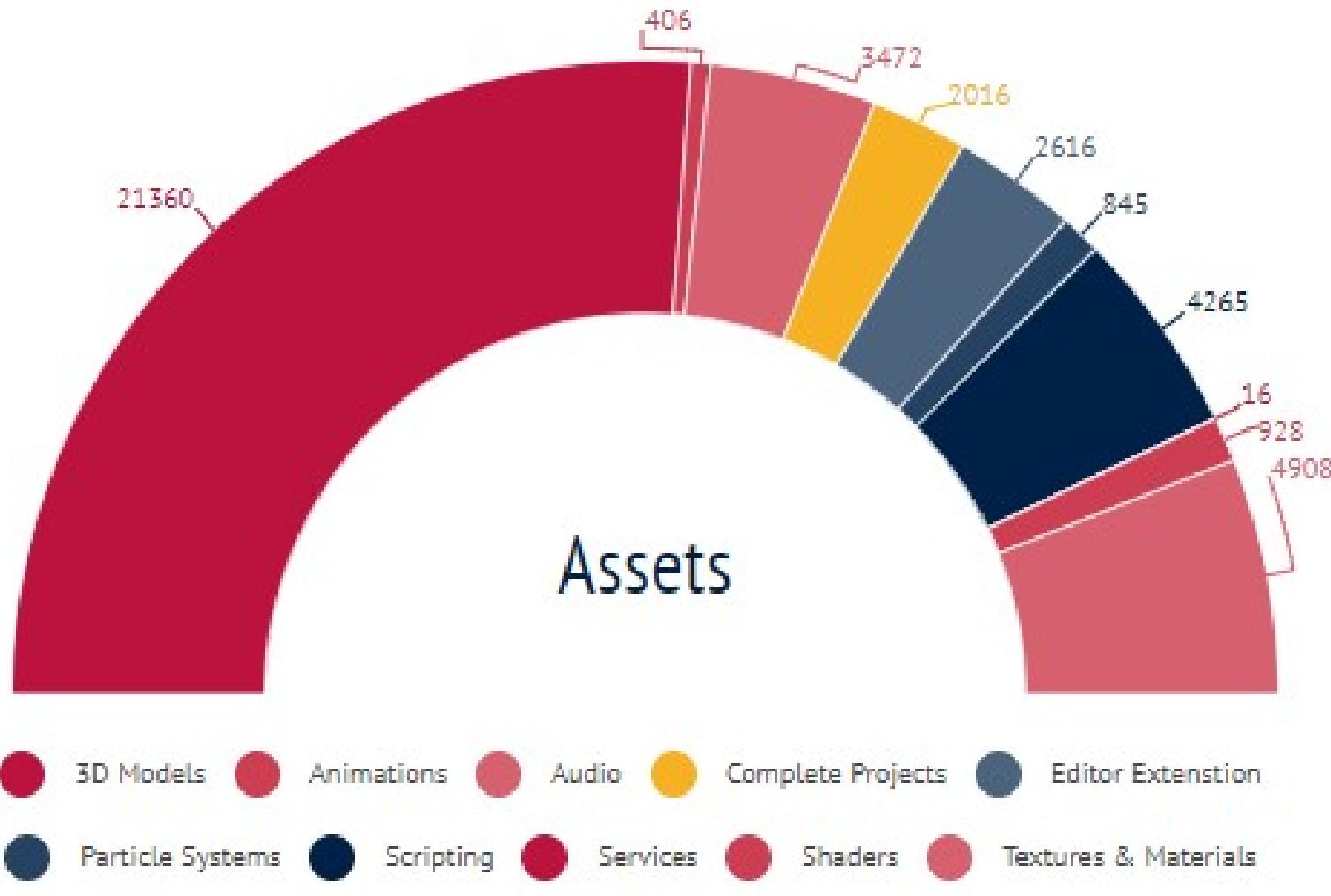


<https://docs.unity3d.com/ru/500/Manual/AssetStore.html>

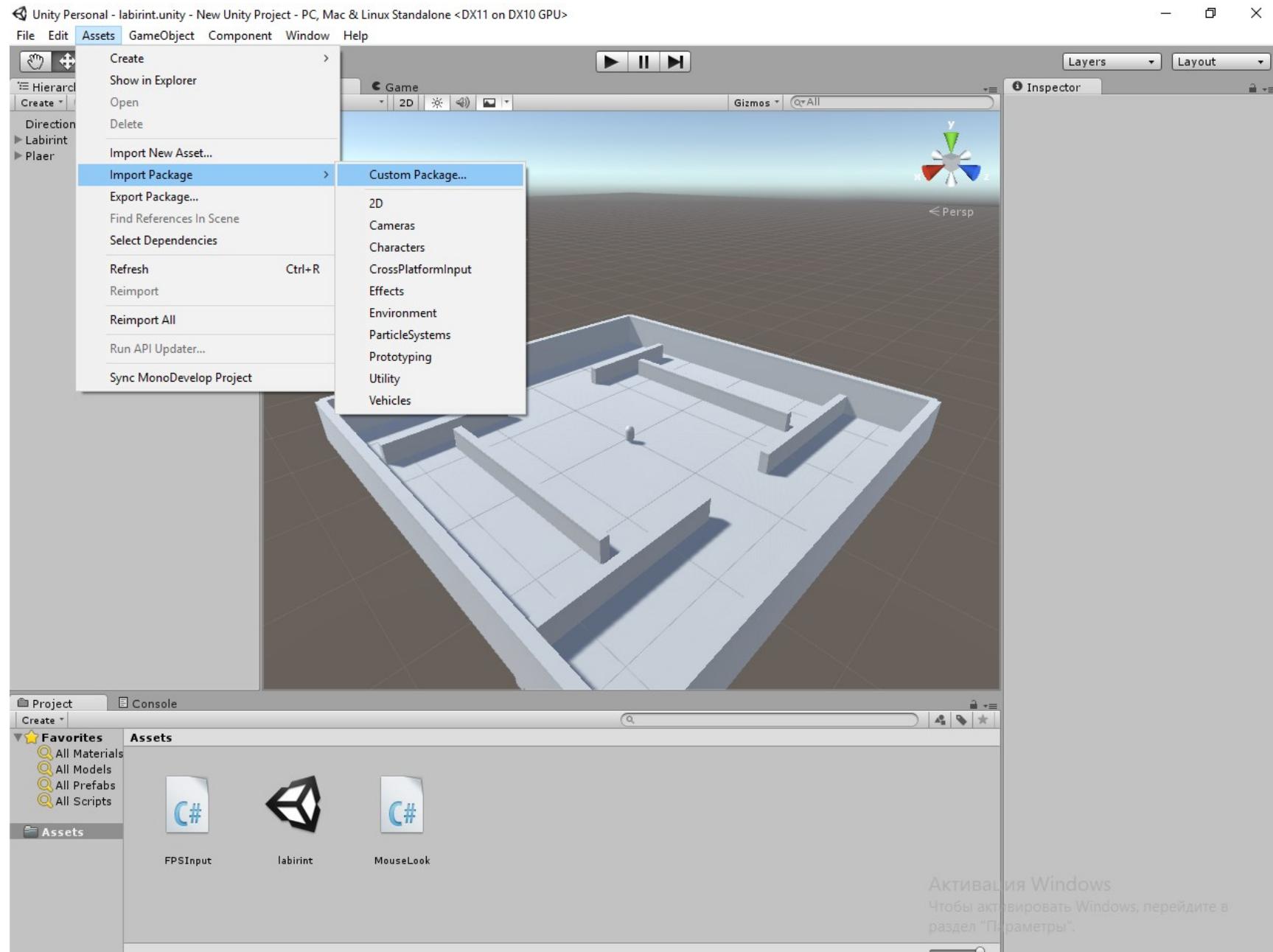
# Доступ до AssetStore

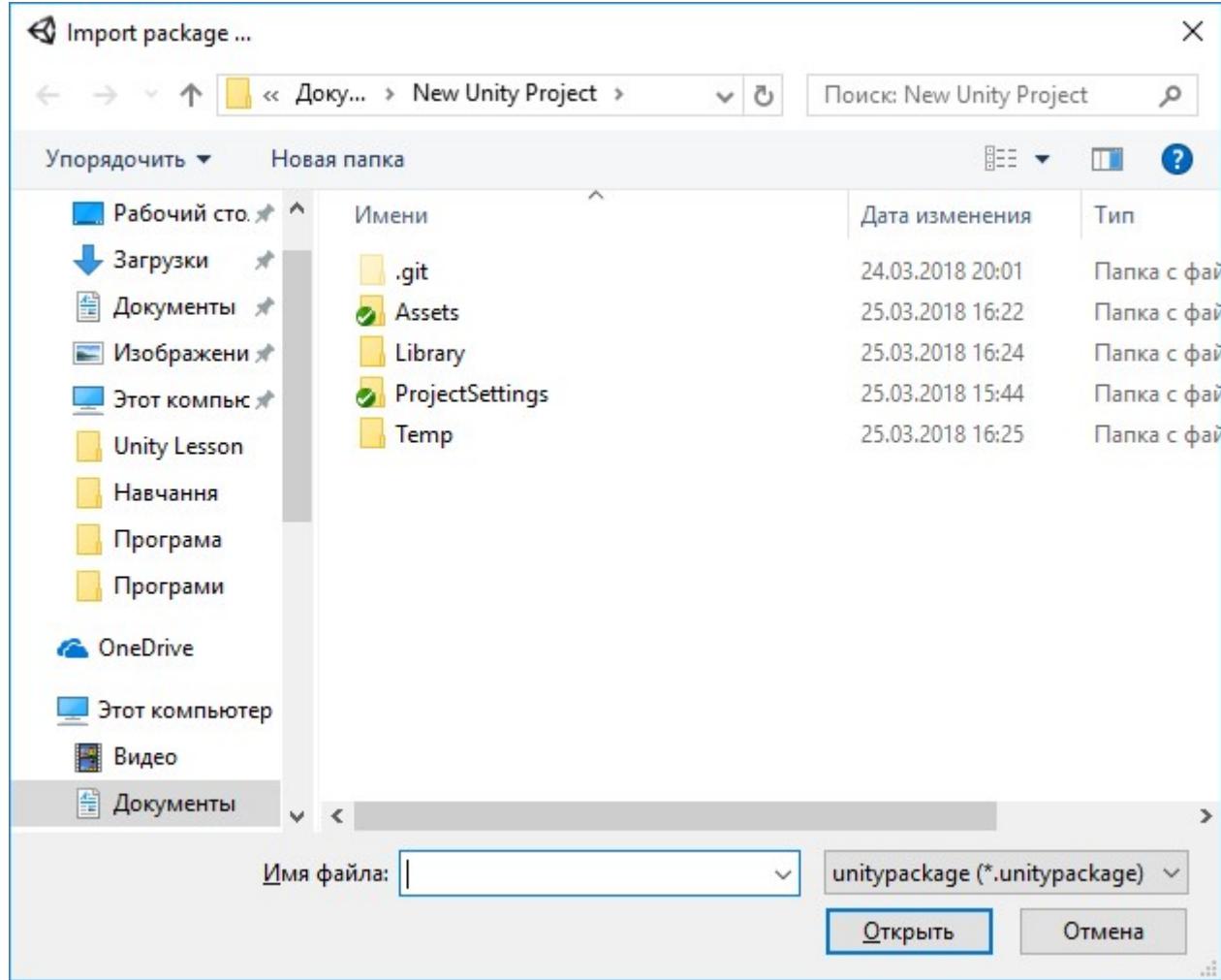


# Діаграма розподілу ресурсів в AssetStore

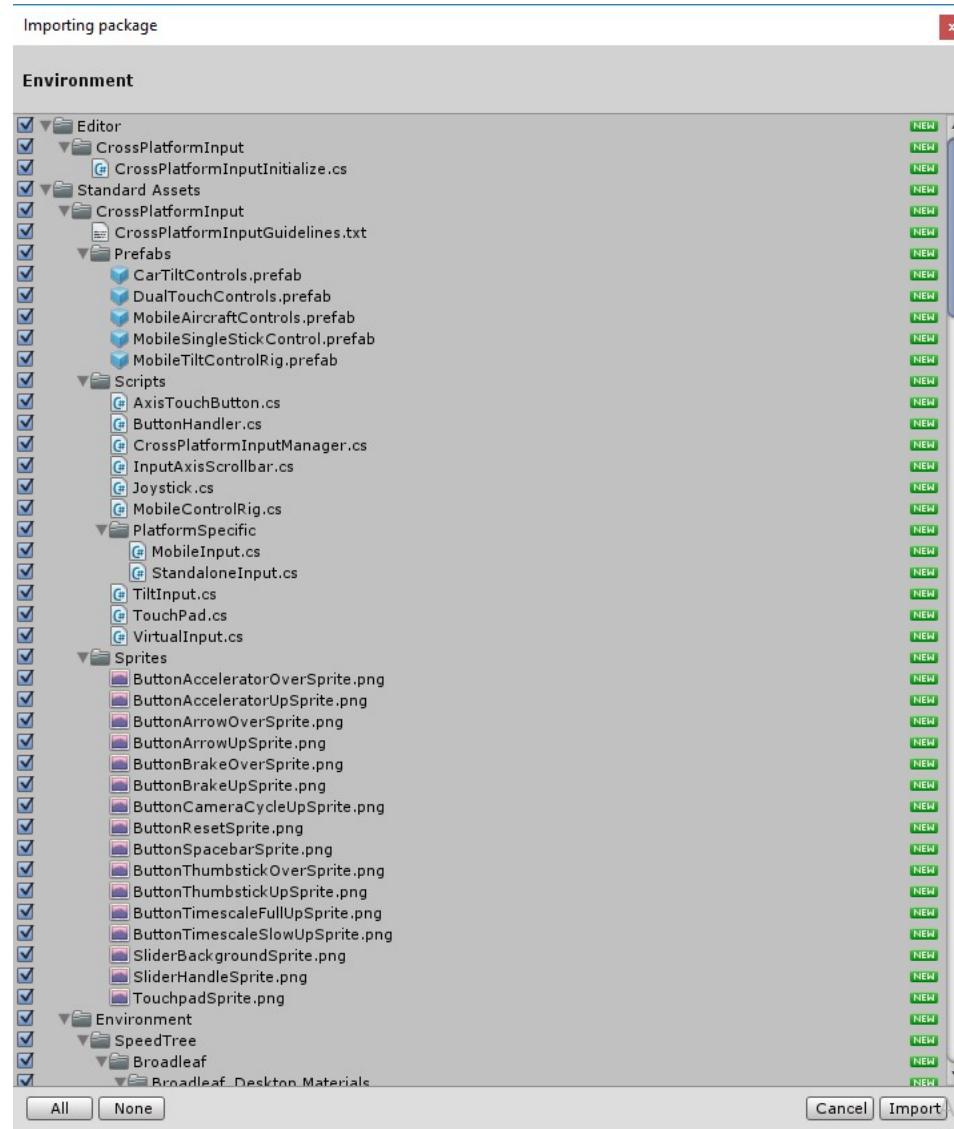


# Імпорт пакетів (unitypackage)





# Вибір стандартних асsetів (Середовище)



# Встановлення стандартних асsetів

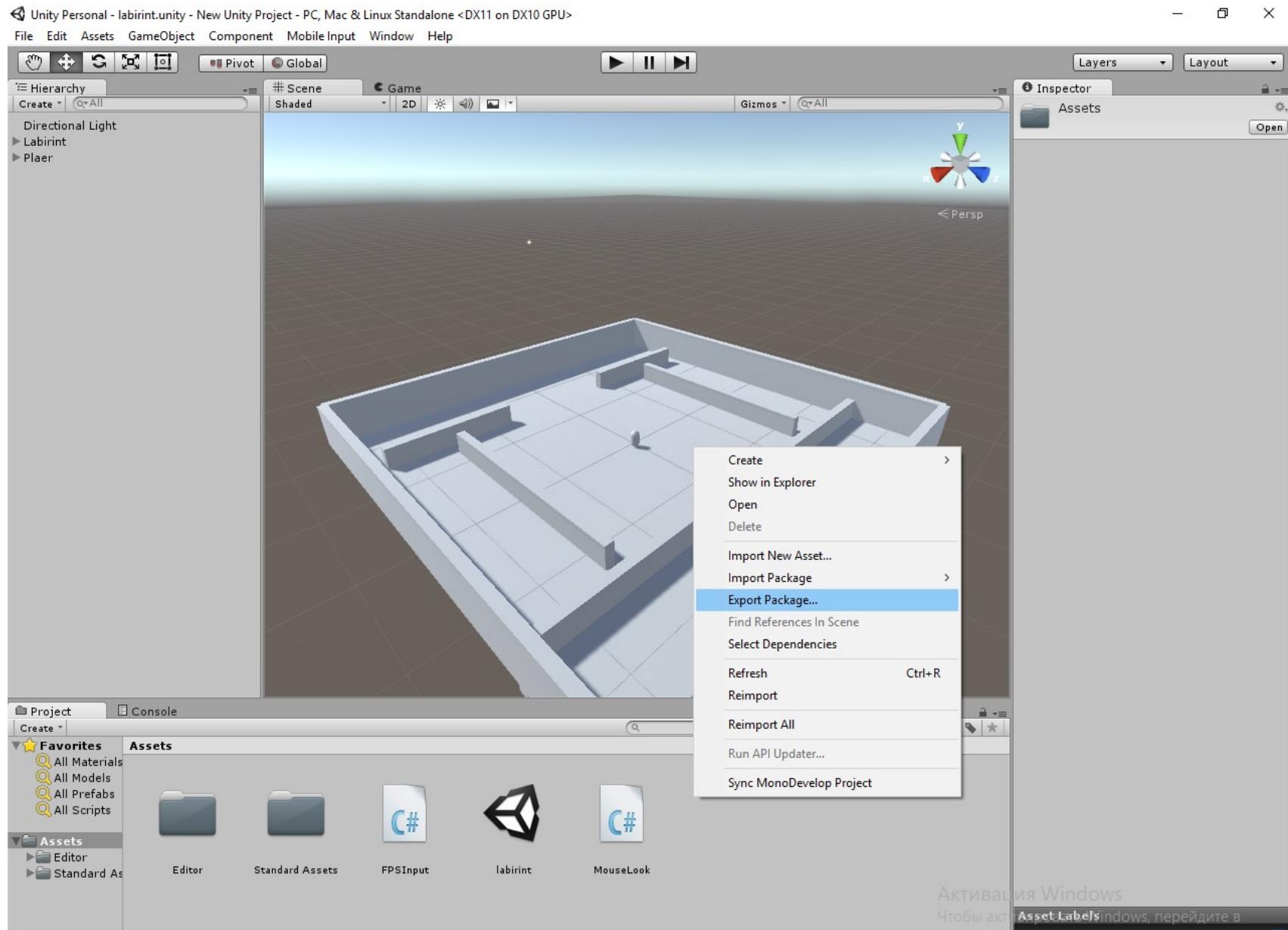
The LTS version will not have any new features, API changes or improvements. It will address crashes, regressions, and issues that affect the wider community, such as Enterprise Support customer issues, console SDK/XDK, or any major changes that would prevent a large section of users from shipping their game. Each LTS stream will be supported for a period of two years.

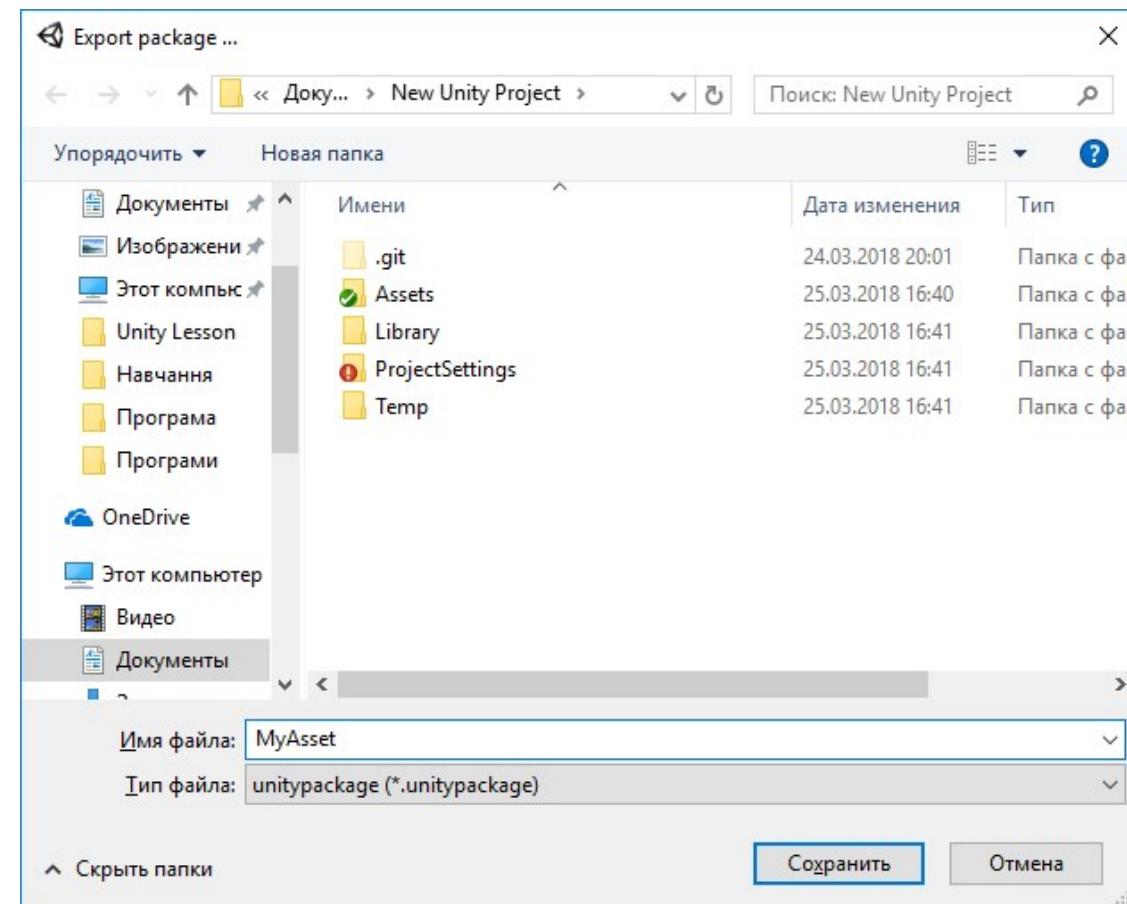
[Download LTS Releases](#)

Unity 2017.x   Unity 5.x   Unity 4.x   Unity 3.x

Unity версія	Дата випуску	Загрузки (Win)	Загрузки (Mac)	Примечания к выпуску
Unity 2017.3.1	5 Feb, 2018	<a href="#">Unity</a>	<a href="#">Cache Server</a>	<a href="#">Примечания к выпуску</a>
Unity 2017.3.0	19 Dec, 2017	<a href="#">Встроенные шейдеры</a>	<a href="#">Стандартные активы</a>	<a href="#">Примечания к выпуску</a>
Unity 2017.2.2	26 Feb, 2018	<a href="#">Пример проекта</a>	<a href="#">Установщик поддержки</a>	<a href="#">Примечания к выпуску</a>
Unity 2017.2.1	12 Dec, 2017	<a href="#">Загрузки (Win)</a>	<a href="#">Загрузки (Mac)</a>	<a href="#">Примечания к выпуску</a>

# Експорт асsetів





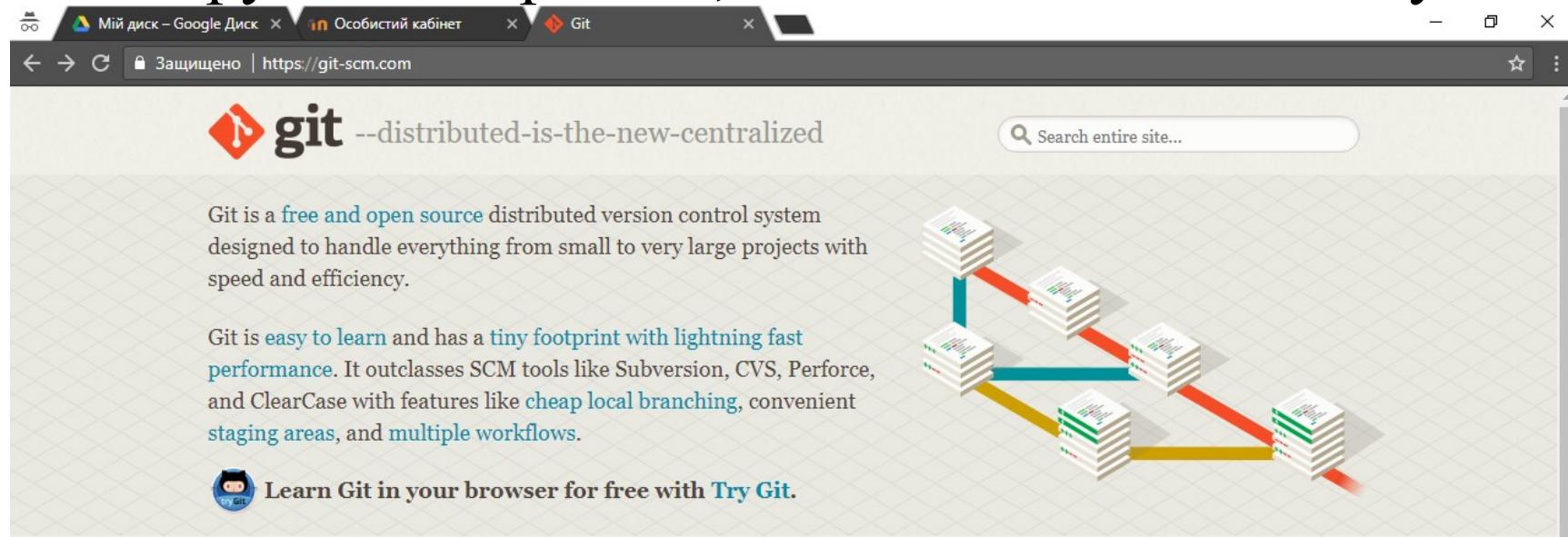
Быстрый доступ	Имя	Дата изменения	Тип	Размера
Рабочий стол	.git	24.03.2018 20:01	Папка с файлами	
Загрузки	Assets	25.03.2018 16:40	Папка с файлами	
Документы	Library	25.03.2018 16:41	Папка с файлами	
Изображения	ProjectSettings	25.03.2018 16:41	Папка с файлами	
Этот компьютер	Temp	25.03.2018 16:42	Папка с файлами	
Unity Lesson	.gitignore	24.03.2018 18:50	Текстовый докум...	1 КБ
Навчання	Assembly-CSharp.csproj	25.03.2018 16:40	Visual C# Project f...	4 КБ
Програма	Assembly-CSharp-Editor.csproj	25.03.2018 16:24	Visual C# Project f...	7 КБ
Програми	Assembly-CSharp-Editor-vs.csproj	25.03.2018 16:24	Visual C# Project f...	7 КБ
OneDrive	Assembly-CSharp-firstpass.csproj	25.03.2018 16:24	Visual C# Project f...	8 КБ
Этот компьютер	Assembly-CSharp-firstpass-vs.csproj	25.03.2018 16:24	Visual C# Project f...	8 КБ
	MyAsset.unitypackage	25.03.2018 16:42	Unity package file	5 КБ
	New Unity Project.sln	25.03.2018 16:40	Microsoft Visual S...	2 КБ

# Asset Store Tools для продажи своїх асsetів

The screenshot shows a web browser displaying the Unity Asset Store at <https://assetstore.unity.com/packages/KATEGORY-MISSING/asset-store-tools-115>. The page features a large central image of the Unity logo (a stylized 'U'). At the top, there's a navigation bar with links for 3D, 2D, Add-Ons, Audio, Templates, Tools, and VFX. On the right side of the header, there are links for 'Create a Unity Account' and 'Log In'. Below the header, a search bar contains the placeholder 'Type here to search assets'. To the right of the search bar are buttons for 'Plus/Pro', 'Impressive New Assets', 'Learn Game Development', and 'Shop On Old Store'. The main content area displays a product card for 'Asset Store Tools' by 'UNITY TECHNOLOGIES'. The card includes a 'FREE' badge, a '5-star rating' icon with '183 user reviews', and a 'Add to My Assets' button. Below the card, there's a section titled 'Upload your content to the Unity Asset Store' with a descriptive paragraph about sharing submissions. At the bottom of the page, there are sections for 'Package contents', 'Releases', and 'Supported Unity versions', along with 'Share' and 'Add to List' buttons.

<https://docs.unity3d.com/ru/500/Manual/AssetStorePublisherAdministration.html>  
<https://docs.unity3d.com/ru/500/Manual/AssetStorePublishingGuide.html>

# Системи керування версіями, завантаження і налаштування Git



Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.

[Learn Git in your browser for free with Try Git.](#)

## About

The advantages of Git compared to other source control systems.

## Documentation

Command reference pages, Pro Git book content, videos and other material.

## Downloads

GUI clients and binary releases for all major platforms.

## Community

Get involved! Bug reporting, mailing list, chat, development and more.

**Pro Git** by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

[Windows GUIs](#) [Tarballs](#)

[Mac Build](#) [Source Code](#)

## Companies & Projects Using Git

<https://git-scm.com/download/win>

microsoft NETFLIX

Активация Windows  
PostgreSQL  
Чтобы активировать Windows, перейдите в раздел "Параметры".

<https://git-scm.com>

# Вибір графічних оболонок

Мій диск – Google Диск | Особистий кабінет | Git - GUI Clients | Защищено | https://git-scm.com/download/gui/win

**About**

**Documentation**

**Downloads**

GUI Clients  
Logos

**Community**

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

## GUI Clients

Git comes with built-in GUI tools for committing ([git-gui](#)) and browsing ([gitk](#)), but there are several third-party tools for users looking for platform-specific experience.

If you want to add another GUI tool to this list, just [follow the instructions](#).

All Windows Mac Linux Android iOS

**SourceTree**  
Platforms: Mac, Windows  
Price: Free  
License: Proprietary

**GitHub Desktop**  
Platforms: Mac, Windows  
Price: Free  
License: MIT

**TortoiseGit**  
Platforms: Windows  
Price: Free  
License: GNU GPL

**Git Extensions**  
Platforms: Linux, Mac, Windows  
Price: Free  
License: GNU GPL

electron + update-docs

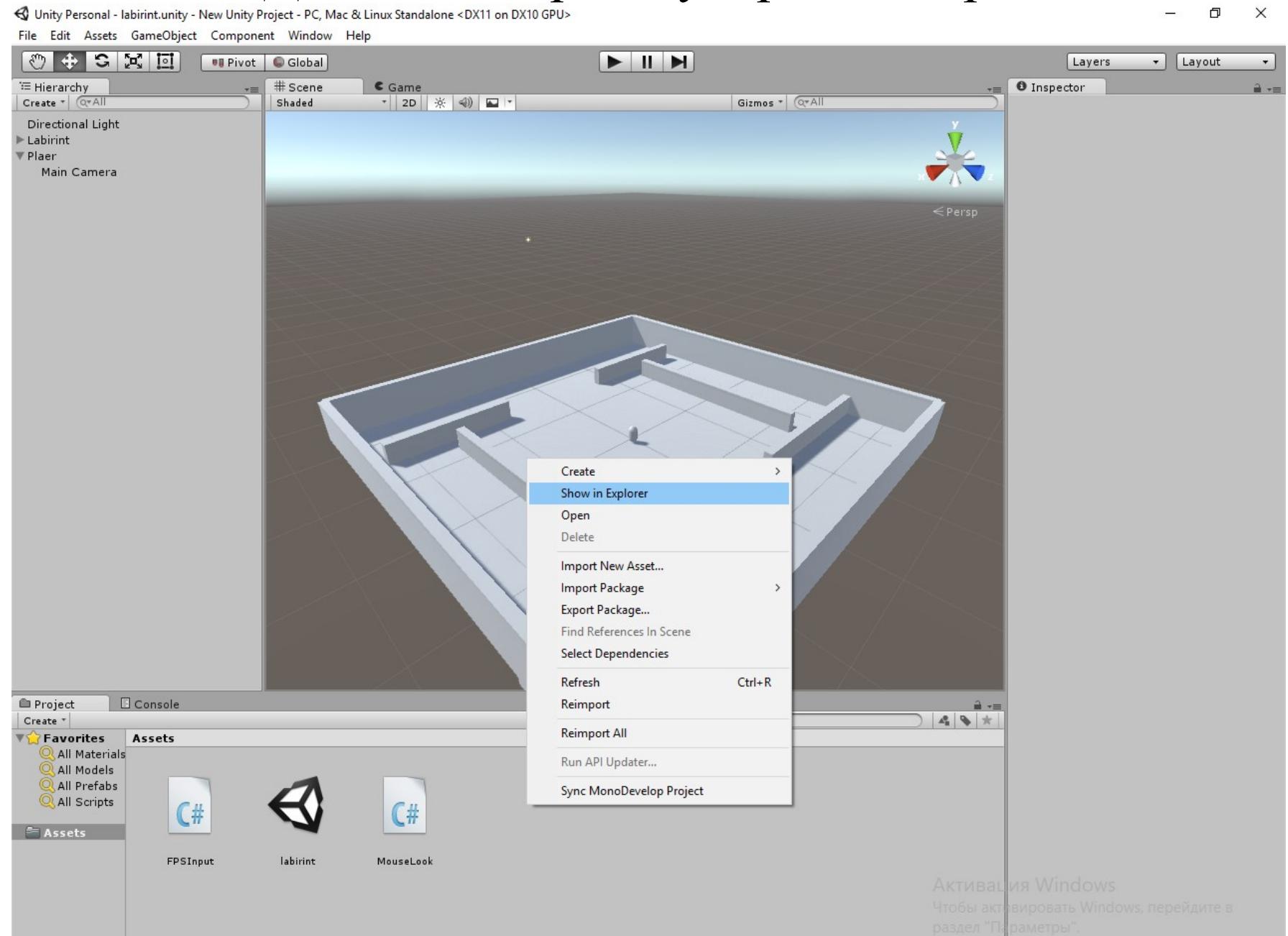
Viewing 298 files Show All  
Filter (38+ entries) + auto-launch-required...  
LOCAL  
master  
read-disk

First pass at auto-launch! Get us compiling on Windows! Get us compiling on Windows! Fix build config

Commit Message  
0 [electron] fixed IDEA warnings.  
0 fixed compilation errors  
0 applied changes from SmartSVN (62142)  
0 [master] SU-3263: Widget is disposed exception after update dialog disposed  
0 [master] SU-3235: File table display remote state in "Name" column

Активация Windows  
Чтобы активировать Windows, перейдите в раздел "Параметры".

# Додавання проекту в репозиторій



	Имя	Дата изменения	Тип	Размера
★ Быстро́й досту́п				
Рабочий стол	Assets	24.03.2018 18:15	Папка с файлами	
Загрузки	Library	24.03.2018 18:15	Папка с файлами	
Документы	ProjectSettings	24.03.2018 18:15	Папка с файлами	
Изображения	Temp	24.03.2018 18:15	Папка с файлами	
Этот компьютер	Assembly-CSharp.csproj	03.02.2018 12:59	Visual C# Project f...	4 КБ
Google Drive File	Assembly-CSharp-vs.csproj	03.02.2018 12:59	Visual C# Project f...	4 КБ
Test	New Unity Project.sln	24.03.2018 17:34	Microsoft Visual S...	2 КБ
Unity Lesson	New Unity Project.userprefs	18.03.2018 18:00	Файл "USERPREFS"	1 КБ
Программа	New Unity Project-csharp.sln	24.03.2018 17:34	Microsoft Visual S...	2 КБ
Программы				
OneDrive				
Этот компьютер				
Видео				
Документы				
Загрузки				
Изображения				
Музыка				
Объемные объекты				
Рабочий стол				
Ло́кальный диск (C:)				
My (D:)				
Book (E:)				
Video1 (F:)				
Video2 (G:)				
Новый том (H:)				
OTHER (I:)				
DVD-дисковод (J:)				
Google Drive File St				
Сеть				

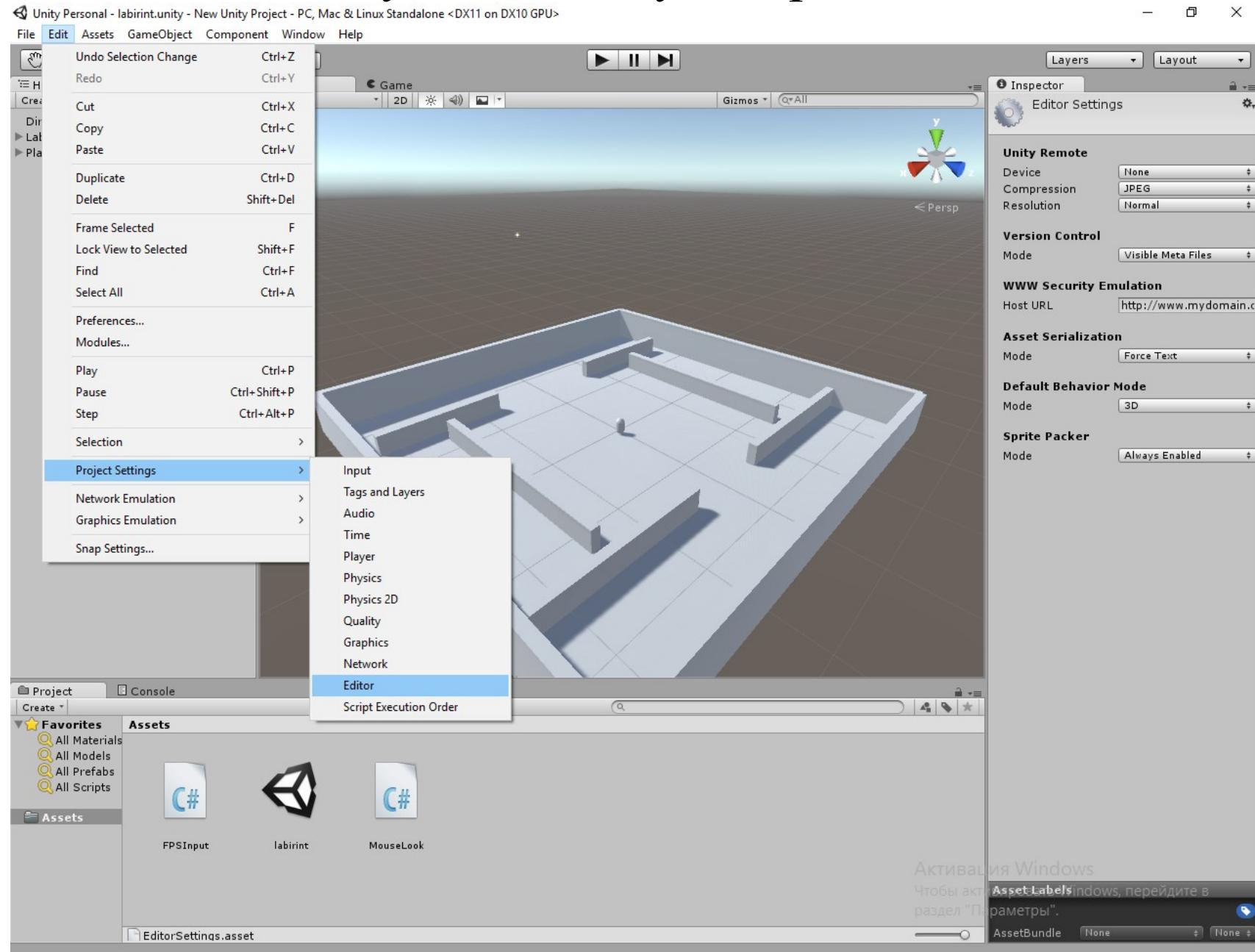
Элементов: 9

## Активация Windows

Чтобы активировать Windows, перейдите в раздел "Параметры".



# Налаштування Unity для роботи з Git



# Створення репозиторію

Файл Главная Поделиться Вид

← → ↑ ↓ | Документы

Этот компьютер > Локальный диск (C:) > Пользователи > Roman > Документы >

Поиск: Документы

Имя Дата изменения Тип Размера

123 03.03.2018 21:51 Папка с файлами

bbb 11.03.2018 19:55 Папка с файлами

New Unity Project 24.03.2018 18:30 Папка с файлами

Открыть Открыть в новом окне Закрепить на панели быстрого доступа

Git GUI Here

Git Bash Here

Проверка с использованием Windows Defender...

Предоставить доступ к

Git Sync...

Git Commit -> "master"...

TortoiseGit

Добавить в архив...

Добавить в архив "New Unity Project.rar"

Добавить в архив и отправить по e-mail...

Добавить в архив "New Unity Project.rar" и отправить по e-mail

Восстановить прежнюю версию

Добавить в библиотеку

Закрепить на начальном экране

Отправить

Файл Главная Поделиться Вид

← → ↑ ↓ | New Unity Project

Этот компьютер > Локальный диск (C:) > Пользователи > Roman > Документы > New Unity Project >

Поиск: New Unity Project

Имя Дата изменения Тип Размера

.git 24.03.2018 18:30 Папка с файлами

Assets 24.03.2018 18:15 Папка с файлами

Library 24.03.2018 18:27 Папка с файлами

ProjectSettings 24.03.2018 18:15 Папка с файлами

Temp 24.03.2018 18:15 Папка с файлами

Assembly-CSharp.csproj 03.02.2018 12:59 Visual C# Project f... 4 КБ

Assembly-CSharp-vs.csproj 03.02.2018 12:59 Visual C# Project f... 4 КБ

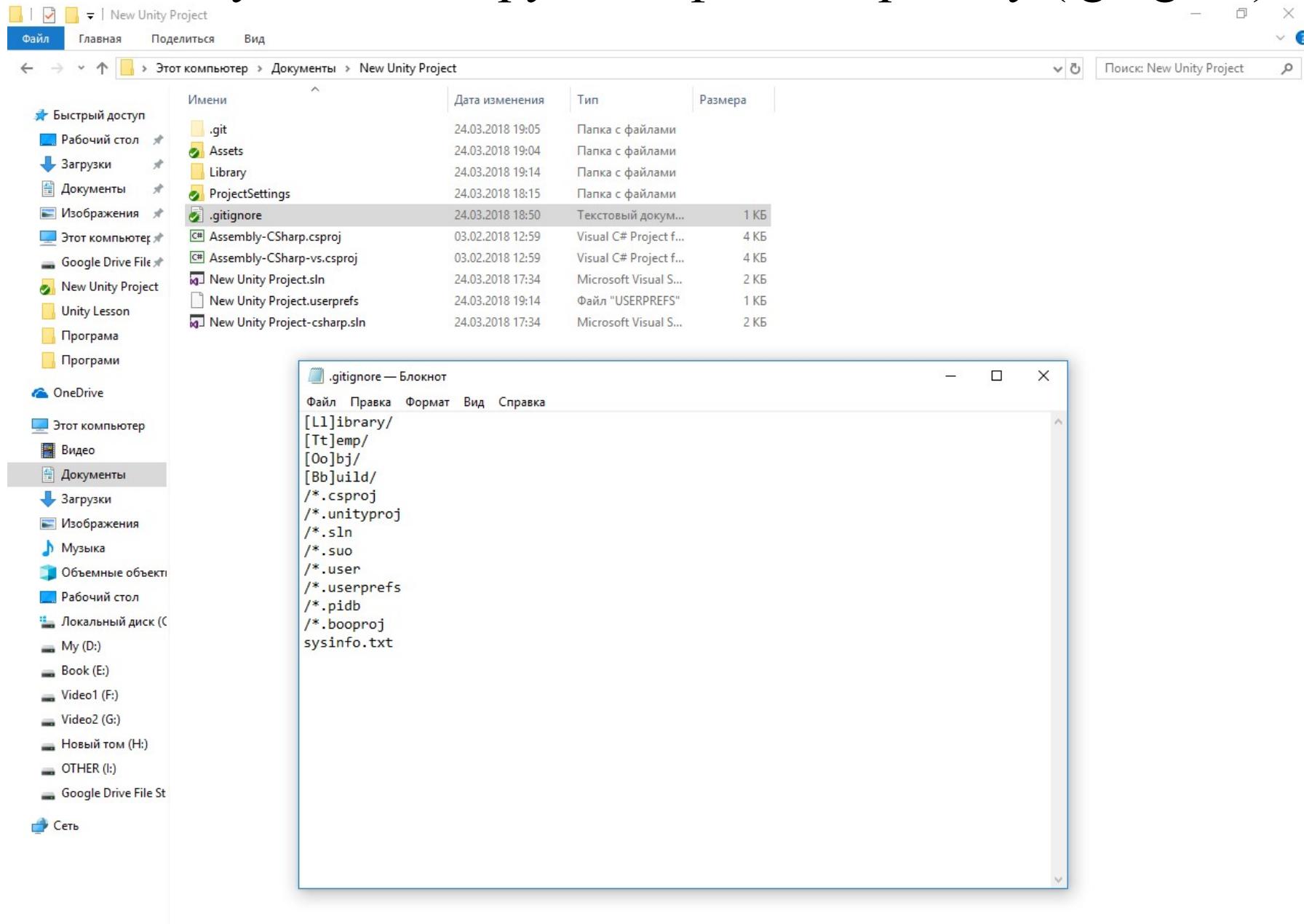
New Unity Project.sln 24.03.2018 17:34 Microsoft Visual S... 2 КБ

New Unity Project.userprefs 18.03.2018 18:00 Файл "USERPREFS" 1 КБ

New Unity Project-csharp.sln 24.03.2018 17:34 Microsoft Visual S... 2 КБ

OneDrive

# Налаштування ігноруємих файлів проекту (.gitignore)



# Перша фіксація змін

New Unity Project

Начальна Порядок виду

↑ > Этот компьютер > Документы > New Unity Project

Пошук New Unity Project

Имя	Дата изменения	Тип
.git	24.03.2018 19:05	Папка с файлами
Assets	24.03.2018 19:04	Папка с файлами
Library	24.03.2018 19:14	Папка с файлами
ProjectSettings	24.03.2018 18:15	Папка с файлами
.gitignore	24.03.2018 18:50	Текстовый
Assembly-CSharp.csproj	03.02.2018 12:59	Visual C#
Assembly-CSharp-vs.csproj	03.02.2018 12:59	Visual C#
New Unity Project.sln	24.03.2018 17:34	Microsoft
New Unity Project.userprefs	24.03.2018 19:14	Файл "USER"
New Unity Project-csharp.sln	24.03.2018 17:34	Microsoft

Commit to: master  new branch

Message:  
Перший коміт! Початок розробки!

Amend Last Commit  
 Set author date  
 Set author

Add Signed-off-by

Changes made (double-click on file for diff):  
Check: All None Unversioned Versioned Added Deleted Modified Files Submodules

Path	Extension	Status	Lines added	Lines removed
.gitignore	.gitignore	Unknown		
Assets/FPSInput.cs	.cs	Unknown		
Assets/FPSInput.cs.meta	.meta	Unknown		
Assets/MouseLook.cs	.cs	Unknown		
Assets/MouseLook.cs.meta	.meta	Unknown		
Assets/labyrinth.unity	.unity	Unknown		
Assets/labyrinth.unity.meta	.meta	Unknown		
ProjectSettings/ AudioManager.asset	.asset	Unknown		
ProjectSettings/ DynamicsManager.asset	.asset	Unknown		
ProjectSettings/ EditorBuildSettings.asset	.asset	Unknown		
ProjectSettings/ EditorSettings.asset	.asset	Unknown		
ProjectSettings/ GraphicsSettings.asset	.asset	Unknown		
ProjectSettings/ InputManager.asset	.asset	Unknown		
ProjectSettings/ NavMeshAreas.asset	.asset	Unknown		
ProjectSettings/ NetworkManager.asset	.asset	Unknown		
ProjectSettings/ Physics2DSettings.asset	.asset	Unknown		
ProjectSettings/ ProjectSettings.asset	.asset	Unknown		

Not Versioned Files

21 files selected, 21 files total

View Patch>>

Show Unversioned Files  
 Do not autoselect submodules  
 Show Whole Project  
 Message only

Commit Cancel Help

Активізація Windows

Вид  
Сортировка  
Группировка  
Обновить  
Настроить папку...  
Вставить  
Вставить ярлык  
Git GUI Here  
Git Bash Here  
Предоставить доступ к...  
Git Sync...  
Git Commit -> "master"...  
TortoiseGit  
Создать  
Свойства

# Зміна файлів

Project

делиться Вид

от компьютера > Документы > New Unity Project >

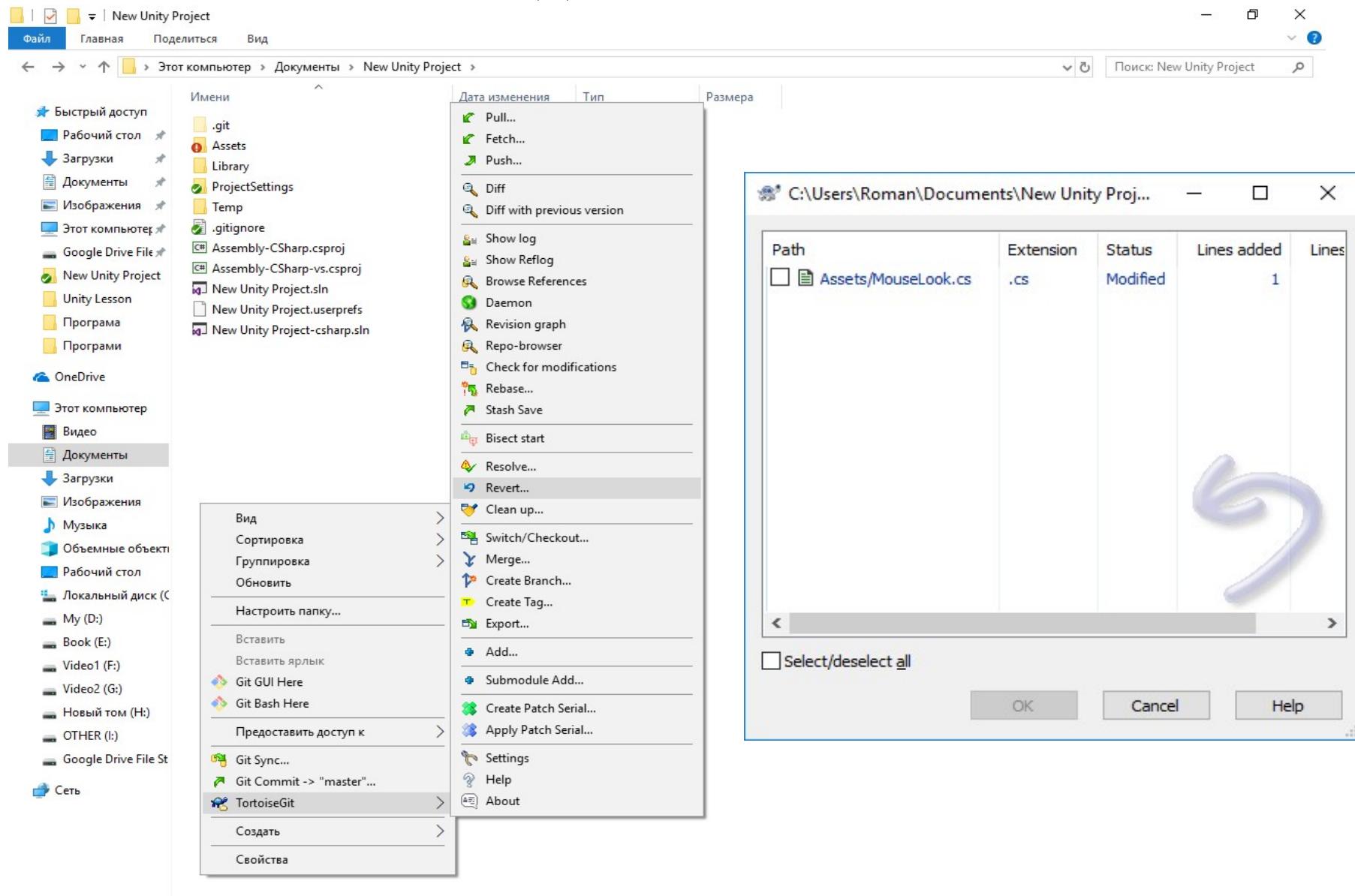
Имена	Дата изменения	Тип	Размера
.git	24.03.2018 19:24	Папка с файлами	
Assets	24.03.2018 19:34	Папка с файлами	
Library	24.03.2018 19:26	Папка с файлами	
ProjectSettings	24.03.2018 19:26	Папка с файлами	
Temp	24.03.2018 19:26	Папка с файлами	
.gitignore	24.03.2018 18:50	Текстовый докум...	1 КБ
C# Assembly-CSharp.csproj	03.02.2018 12:59	Visual C# Project f...	4 КБ
C# Assembly-CSharp-vs.csproj	03.02.2018 12:59	Visual C# Project f...	4 КБ
Microsoft Visual Studio Solution File (.sln)	24.03.2018 19:26	Microsoft Visual S...	2 КБ
New Unity Project.userprefs	24.03.2018 19:14	Файл "USERPREFS"	1 КБ
Microsoft Visual Studio Solution File (.sln)	24.03.2018 19:26	Microsoft Visual S...	2 КБ

делиться Вид

от компьютера > Документы > New Unity Project > Assets

Имена	Дата изменения	Тип	Размера
FPSInput.cs	03.02.2018 14:19	Visual C# Source f...	1 КБ
FPSInput.cs.meta	03.02.2018 12:59	Файл "META"	1 КБ
labirint.unity	24.03.2018 18:15	Unity scene file	31 КБ
labirint.unity.meta	27.01.2018 12:02	Файл "META"	1 КБ
MouseLook.cs	24.03.2018 19:34	Visual C# Source f...	2 КБ
MouseLook.cs.meta	27.01.2018 11:08	Файл "META"	1 КБ

# Відкат зміни



# Погодження змін

Скріншот демонструє процес погодження змін в Unity Project за допомогою TortoiseGit.

Лівий екран: Розгорнута структура папок проекту New Unity Project в проводнику Windows. Папка .git та інші залежності видно в лівому меню.

Правий екран: Окно TortoiseGit Commit. Вікно відкрито на папці C:\Users\Roman\Documents\New Unity Project - Commit - TortoiseGit.

Вікно Commit відкрите на вкладині "Commit to: master".

Сообщение в поле Message:

У файл Assets/MouseLook.cs добавлено комментарий.

Параметри погодження:

- new branch
- Amend Last Commit
- Set author date
- Set author

Панель змін (Changes made):

Check: All None Unversioned Versioned Added Deleted Modified Files Submodules

Path	Extension	Status	Lines added	Lines removed
Assets/MouseLook.cs	.cs	Modified	1	1

Іконка зеленої стрілки вказує на погодження змін.

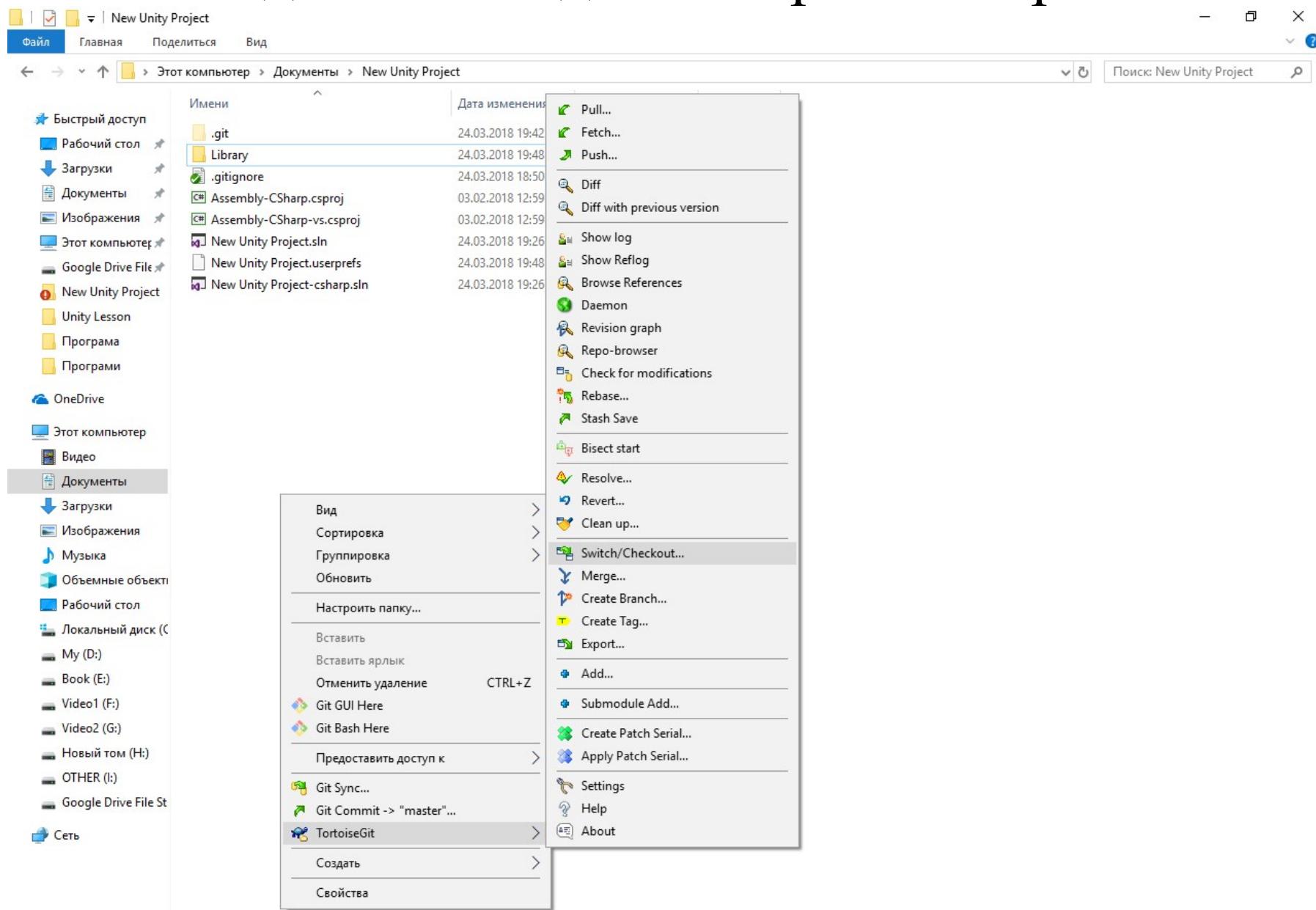
Нижній рядок панелі змін:

- Show Unversioned Files
- Do not autoselect submodules
- Show Whole Project
- Message only

Кнопки дій:

- Commit
- Cancel
- Help

# Відновлення даних із репозиторію



New Unity Project

Файл Главная Поделиться Вид

Этот компьютер > Документы > New Unity Project

Положение: C:\Users\Roman\Documents\New Unity Project - Switch/Checkout - Тип: Рабочий стол

Имя Дата изменения Размер

.git 1 КБ  
Lib 4 КБ  
.gitignore 4 КБ  
Assets 2 КБ  
Assets 1 КБ  
Assets 2 КБ

Быстрый доступ

- Рабочий стол
- Загрузки
- Документы
- Изображения
- Этот компьютер
- Google Drive File
- New Unity Project
- Unity Lesson
- Программа
- Программы
- OneDrive
- Этот компьютер
- Видео
- Документы
- Загрузки
- Изображения
- Музыка
- Объемные объекты
- Рабочий стол
- Локальный диск (C:)
- My (D:)
- Book (E:)
- Video1 (F:)
- Video2 (G:)
- Новый том (H:)
- OTHER (I:)
- Google Drive File St
- Сеть

Switch To

Branch: master

Tag:

Commit:

Option

Create New Branch Branch\_

Overwrite working tree changes (force)  Merge

Track

Override branch if exists

OK Cancel

C:\Users\Roman\Documents\New Unity Project - Log Messages - TortoiseGit

master From: 24.03.2018 To: 24.03.2018 Aut Author Email

Graph	Actions	Message	Author	Date
		master У файл Assets/MouseLook.cs добав... Y_S_V 24.03.2018	Y_S_V	24.03.2018
Перший коміт! Початок розробки.				

SHA-1: a0c93f14c2d302ef54d1db6dee17930ac62ebb7f

\* У файл Assets/MouseLook.cs добавлено коментарий.

Path	Extension	Status	Lines added	Lines removed
Assets/MouseLook.cs	.cs	Modified	1	1

Showing 2 revision(s), from revision cc0d881 to revision a0c93f1 - 1 revision(s) selected, 0 file(s) selected; line: 1(+ 1(-)

Show Whole Project  All Branches Filter paths Help

Refresh Statistics Walk Behaviour View OK Cancel

# Огляд вмісту репозиторію

The screenshot shows the TortoiseGit Repository Browser interface. The main window displays the file structure of a Unity project located at `C:\Users\Roman\Documents\New Unity Project`. The current revision is HEAD. The tree view on the left shows the root directory `/` containing `Assets` and `ProjectSettings`. The details view on the right lists the files under `/Assets`:

Filename	Extension	Size
FPSInput.cs	.cs	819 байт
FPSInput.cs.meta	.meta	263 байт
labyrinth.unity	.unity	30,5 КБ
labyrinth.unity.meta	.meta	175 байт
MouseLook.cs	.cs	1,31 КБ
MouseLook.cs.meta	.meta	263 байт

At the bottom of the details view, it says "Showing 6 files, 0 submodules and 0 folders, 6 items in total".

The context menu for the `/Assets` folder is open, showing options like Pull..., Fetch..., Push..., Diff, Show log, Show Reflog, Browse References, Daemon, Revision graph, Repo-browser, Check for modifications, Rebase..., Stash Save, Bisect start, Resolve..., Revert..., Clean up..., Switch/Checkout..., Merge..., Create Branch..., Create Tag..., Export..., Add..., Submodule Add..., Create Patch Serial..., Apply Patch Serial..., Settings, Help, and About.

The status bar at the bottom of the TortoiseGit window shows "Assets" and buttons for "OK", "Cancel", and "Help".

Дякую за увагу!



# Тема 4. Використання скриптів руху в Unity3d

The screenshot shows the Unity Editor interface. On the left, there is a list of scripts in the Project tab, including Character, AutoBackToTitle.cs, ClickToStart.cs, Explosion.cs, Explosive.cs, Fire.cs, FloorSection.cs, GameControl.cs, GameGUI.cs, Hose.cs, MapIcons.cs, MessageGUI.cs, MoveBetweenPoints, Player.cs (selected), Priority Particle Add., PriorityAlphaParticle, SceneChanger.cs, SmokeParticles.cs, WaterHoseParticles, WaterSplash.cs, and World.cs. On the right, the Script Editor window displays the following C# code:

```
    vignette.blur = (1-health) * 2 + smokeEffect * 30 * (health - 1);
    vignette.blurDistance = (1-health) * 2 + smokeEffect * 30 * (health - 1);
    vignette.chromaticAberration = heatEffect * 30;
}

void OnTriggerStay(Collider c)
{
    var fire = c.GetComponent<Fire>();
    if (fire && fire.alive)
    {
        float dist = 1 - ((transform.position - fire.transform.position).magnitude);
        NearHeat(dist);
    }

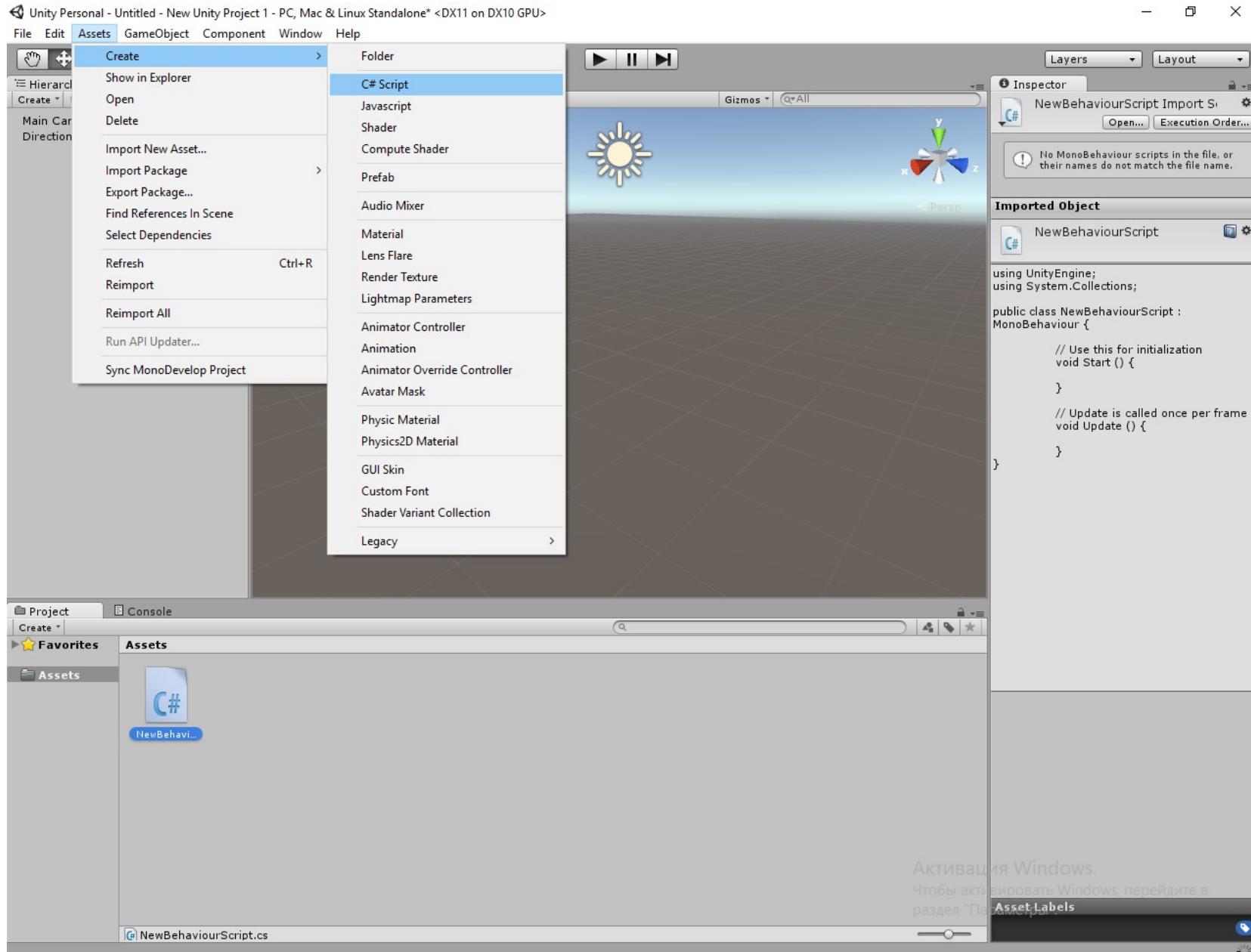
    var smoke = c.GetComponent<ClickToStart>();
    if (smoke && smoke.GetComponent<ParticleFunction>())
    {
        float dist = 1 - ((transform.position - smoke.transform.position).magnitude);
        NearSmoke(dist);
    }
}

void OnCollisionEnter(Collision c)
{
    var healthBox = c.gameObject.GetComponent<HealthBox>();
    if (healthBox)
    {
        if (healthBox.health <= 0)
```

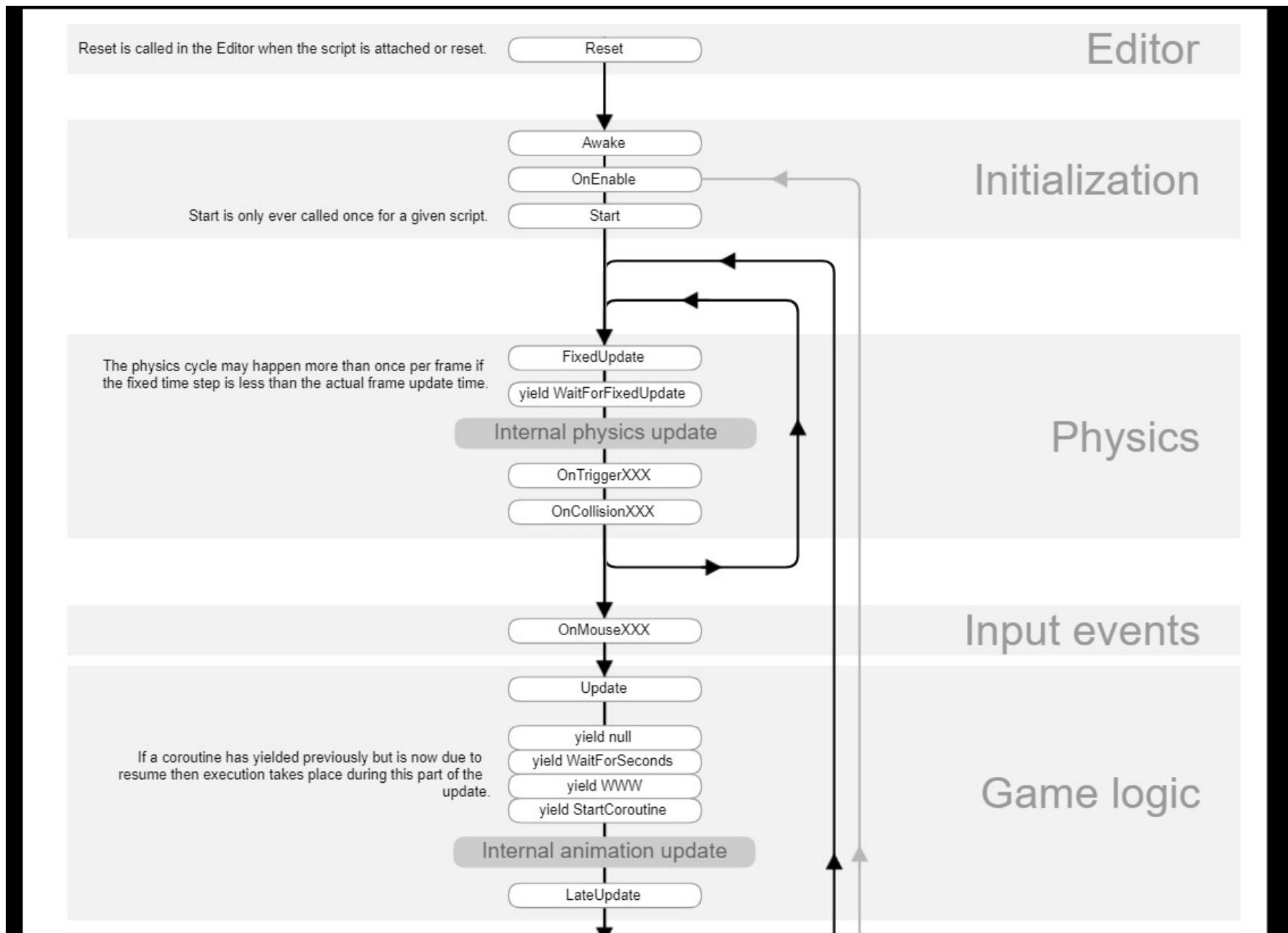
<https://docs.unity3d.com/ru/current/Manual/ScriptingSection.html>

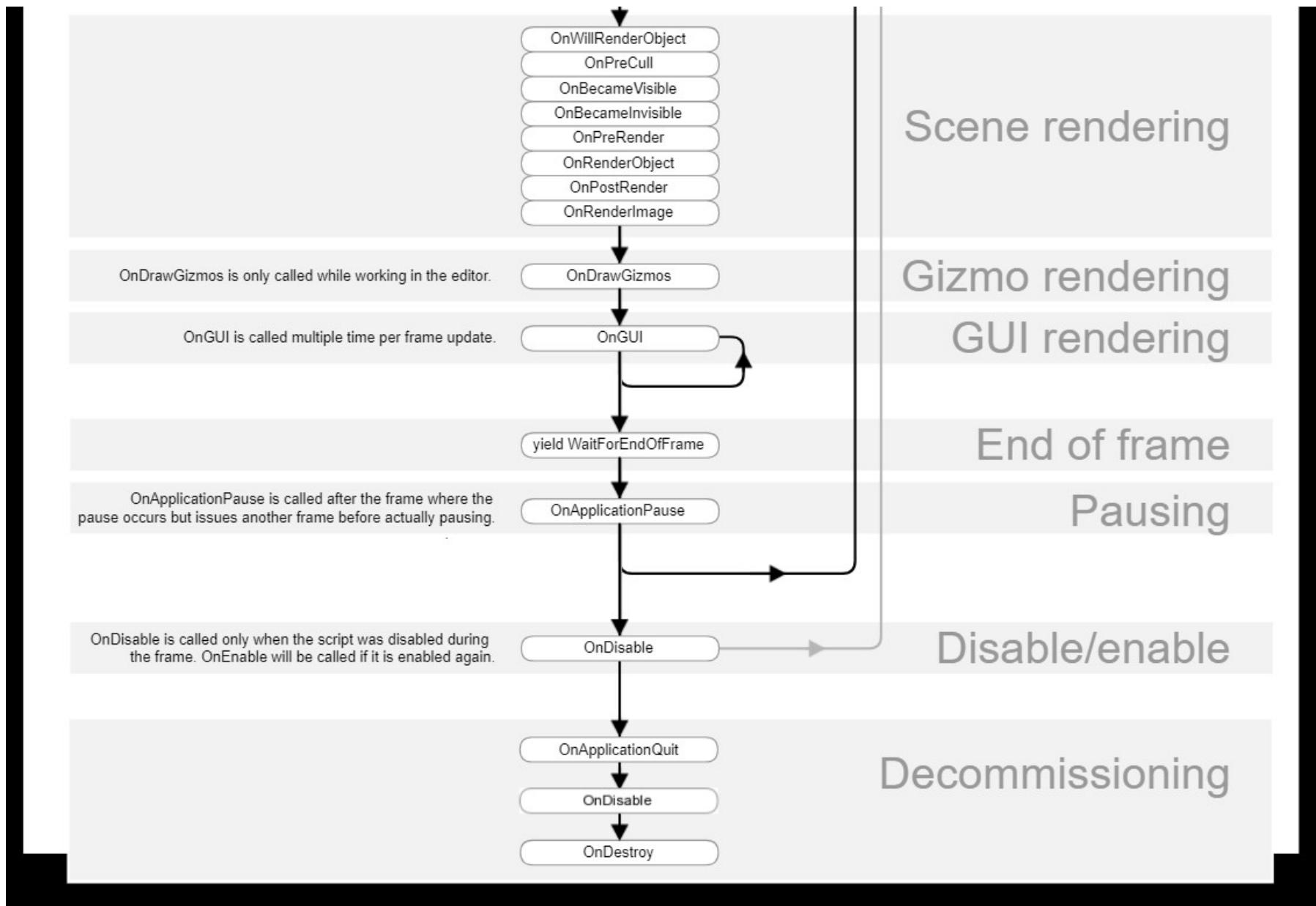
<https://docs.unity3d.com/ru/current/ScriptReference/index.html>

# Створення скрипта



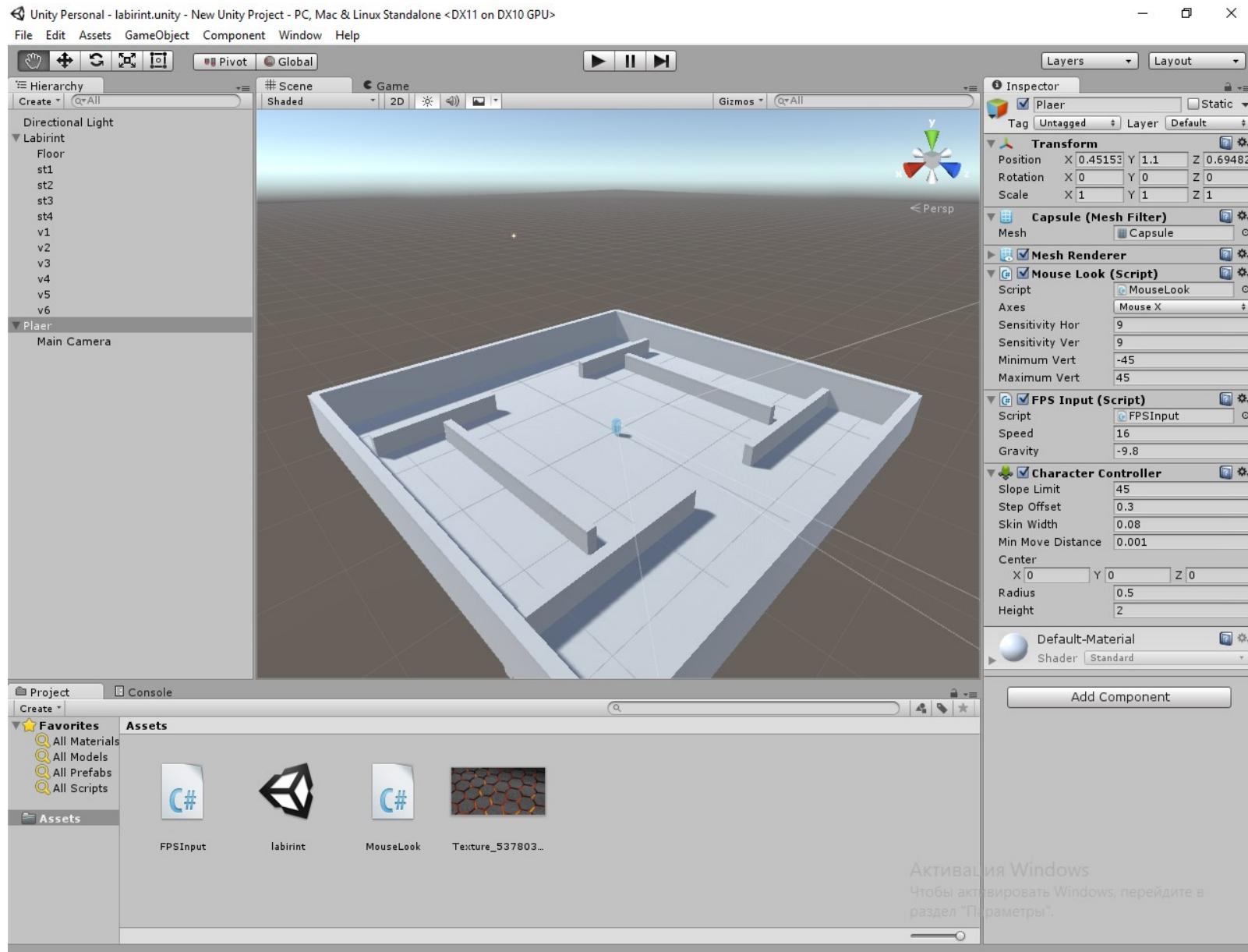
# БЛОК-СХЕМА ЖИТТЕВОГО ЦИКЛУ СКРИПТА

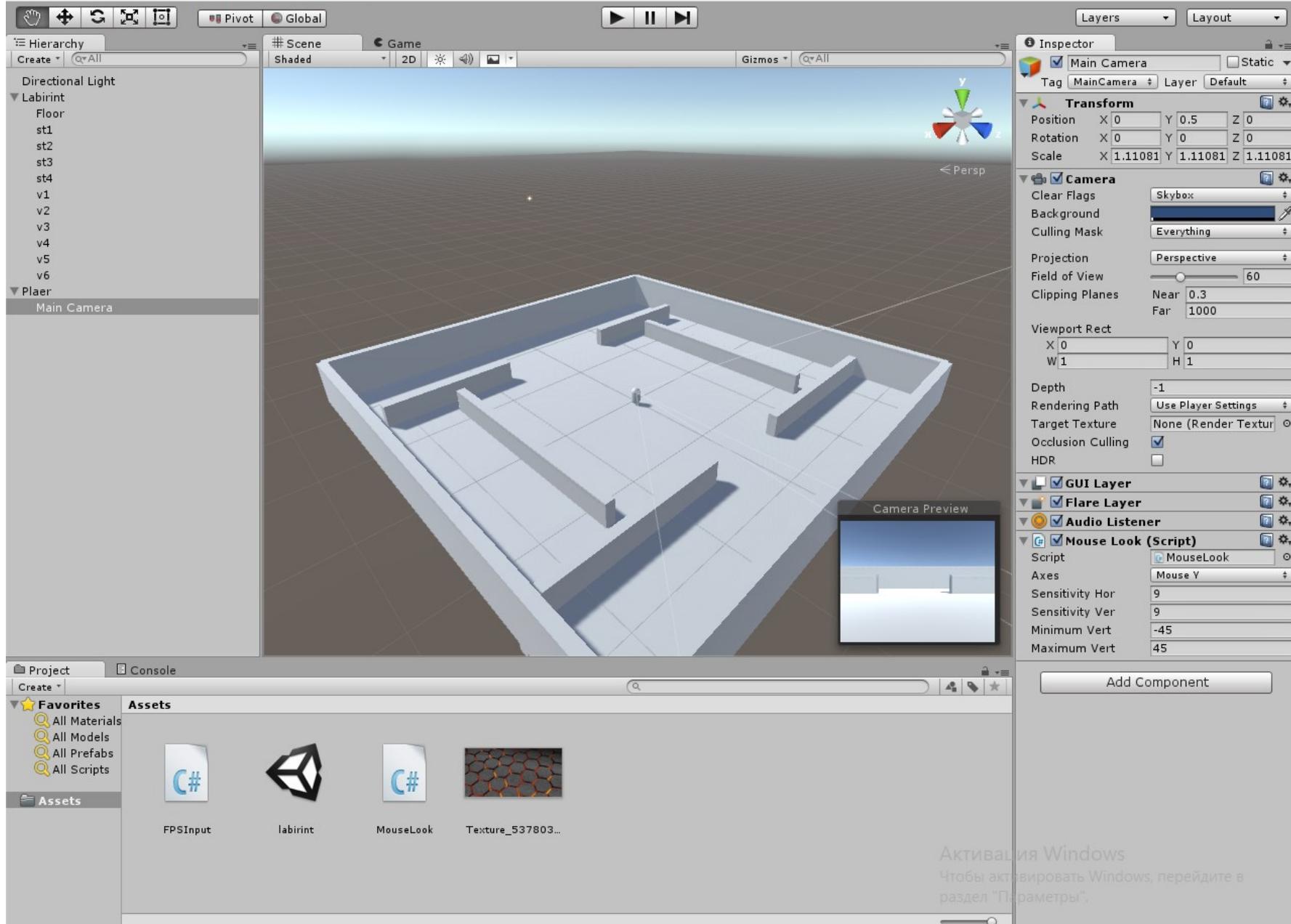




<https://docs.unity3d.com/ru/current/Manual/ExecutionOrder.html>

# Створення шутера від першої особи



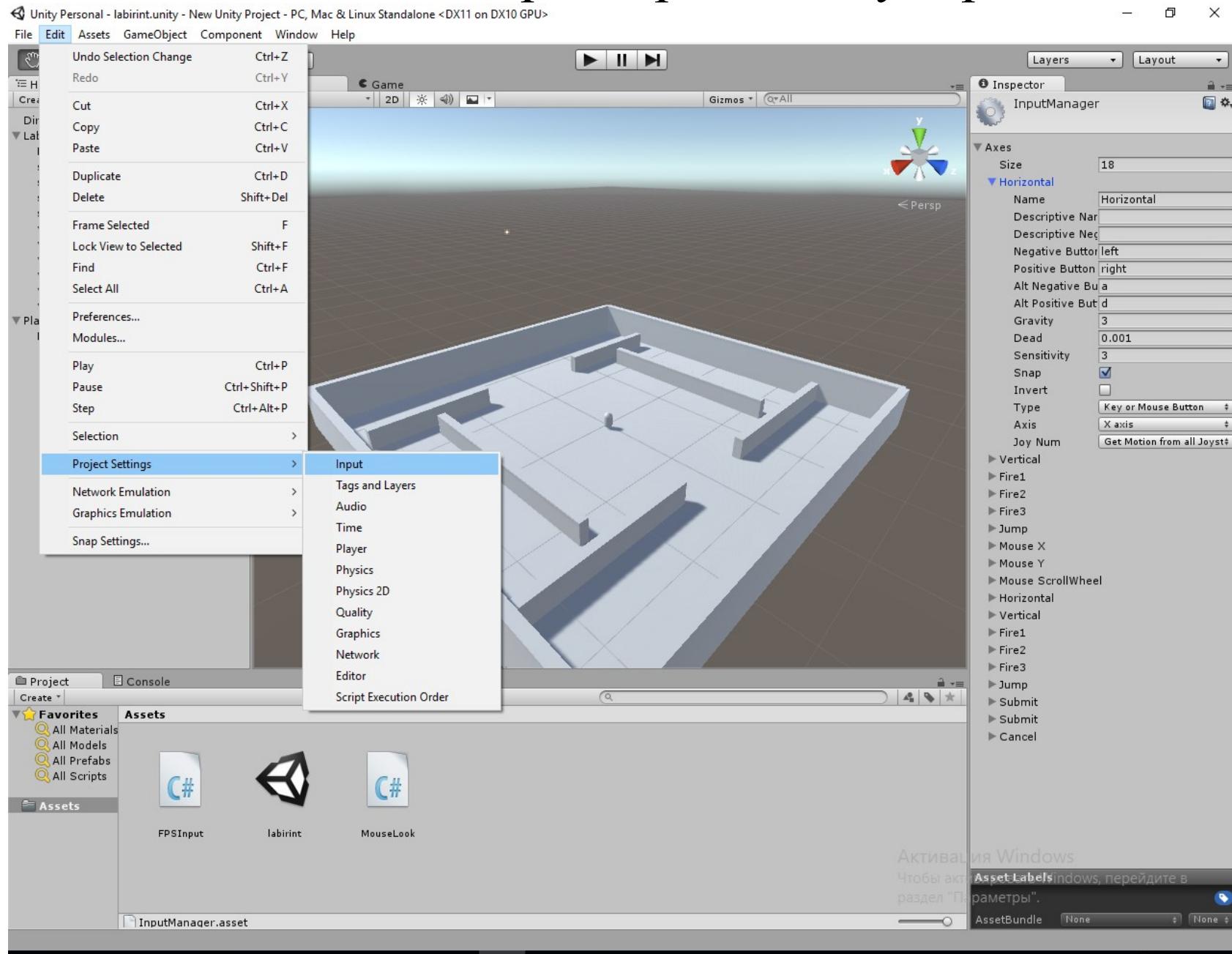


# Скрипт управління гравцем. Клас Input

<https://docs.unity3d.com/ScriptReference/Input.html>

```
using UnityEngine;
public class FPSInput : MonoBehaviour {
    public float speed=6.0f;
    public float gravity = -9.8f;
    private CharacterController _charController;
    // Use this for initialization
    void Start () {
        _charController=GetComponent<CharacterController>();      }
    // Update is called once per frame
    void Update () {
        float deltaX = Input.GetAxis ("Horizontal") * speed;
        float deltaZ = Input.GetAxis ("Vertical") * speed;
        Vector3 movement = new Vector3 (deltaX,0, deltaZ);
        movement = Vector3.ClampMagnitude (movement, speed);
        movement.y = gravity;
        movement *= Time.deltaTime;
        movement = transform.TransformDirection (movement);
        _charController.Move(movement);
    }
}
```

# Можливі значення параметрів методу Input.GetAxis



# Скрипт управління камерою

```
using UnityEngine;
public class MouseLook : MonoBehaviour {
    public enum RotationAxes
    {MouseXandY=0,
     MouseX=1,
     MouseY=2
    }
    public RotationAxes axes=RotationAxes.MouseX;

    public float sensitivityHor = 9.0f;
    public float sensitivityVer = 9.0f;

    public float minimumVert = -45.0f;
    public float maximumVert = 45.0f;

    private float _rotationX=0;
    // Use this for initialization
    void Start () {
        Rigidbody body = GetComponent<Rigidbody>();
        if (body != null)
            body.freezeRotation = true;
    }
}
```

```
// Update is called once per frame
void Update () {
    if (axes == RotationAxes.MouseX) {
        transform.Rotate (0, Input.GetAxis ("Mouse X") * sensitivityHor, 0);
    }
    else if (axes == RotationAxes.MouseY) {
        _rotationX -= Input.GetAxis ("Mouse Y") * sensitivityVer;
        _rotationX=Mathf.Clamp(_rotationX,minimumVert,maximumVert);
        float rotatyonY=transform.localEulerAngles.y;
        transform.localEulerAngles= new Vector3(_rotationX,rotatyonY,0);
    }
    else {
        _rotationX -= Input.GetAxis ("Mouse Y") * sensitivityVer;
        _rotationX=Mathf.Clamp(_rotationX,minimumVert,maximumVert);
        float delta=Input.GetAxis ("Mouse X") * sensitivityHor;
        float rotatyonY=transform.localEulerAngles.y+delta;

        transform.localEulerAngles= new Vector3(_rotationX,rotatyonY,0);
    }
}
```

# ОСНОВА ВСІХ ОСНОВ клас Vector3

<https://docs.unity3d.com/ScriptReference/Vector3.html>

The screenshot shows the Unity Documentation interface for the Vector3 class. The left sidebar lists various Unity classes, with 'Vector3' highlighted in a dark box. The main content area is divided into sections: 'Constructors', 'Public Methods', and 'Static Methods'. Each section contains a list of methods with their descriptions.

**Constructors:**

- [Vector3](#) Creates a new vector with given x, y, z components.

**Public Methods:**

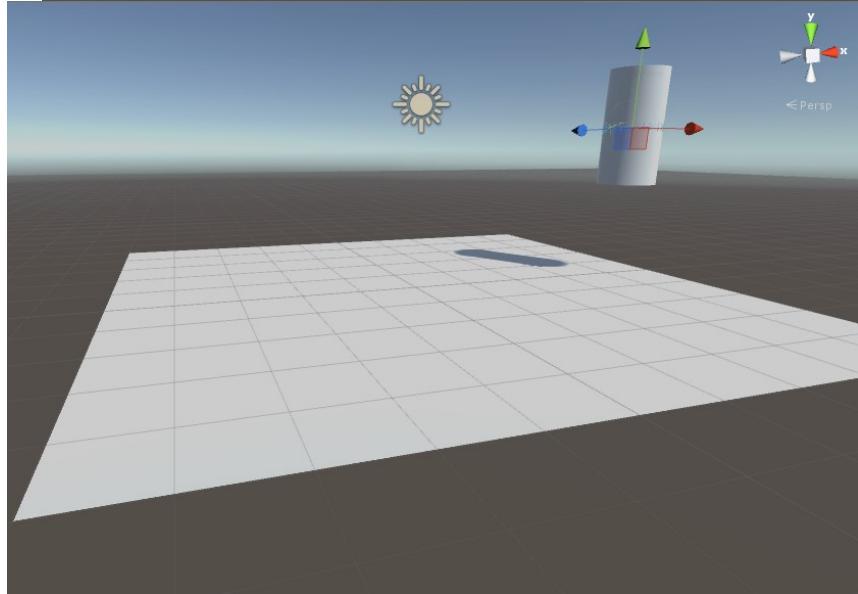
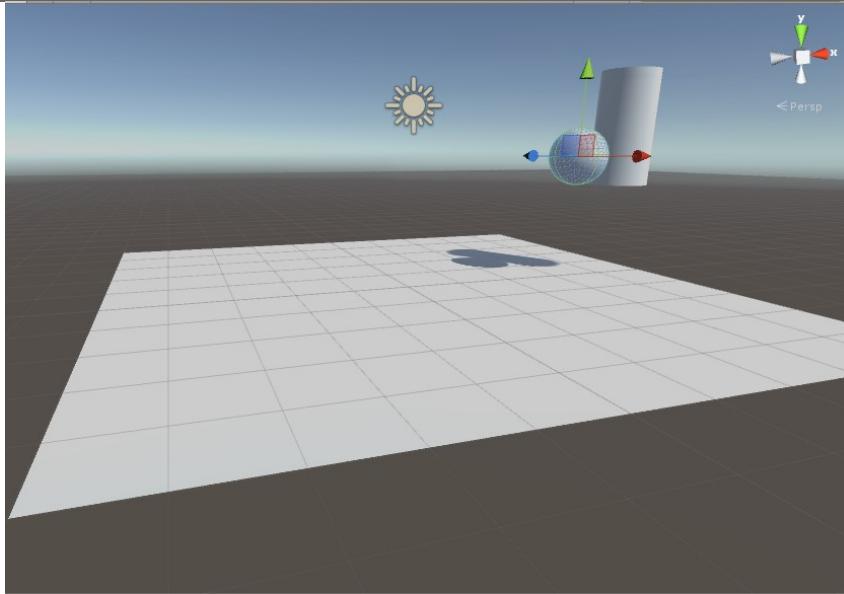
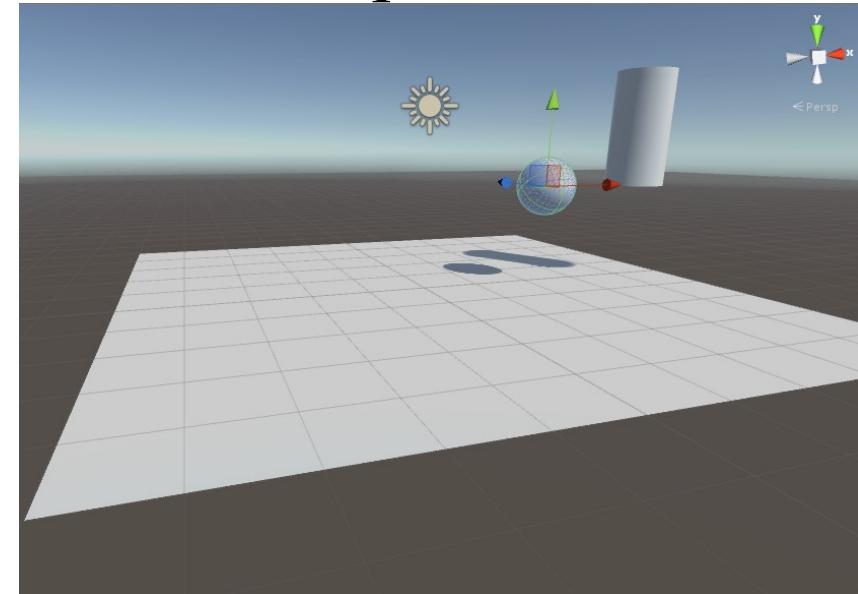
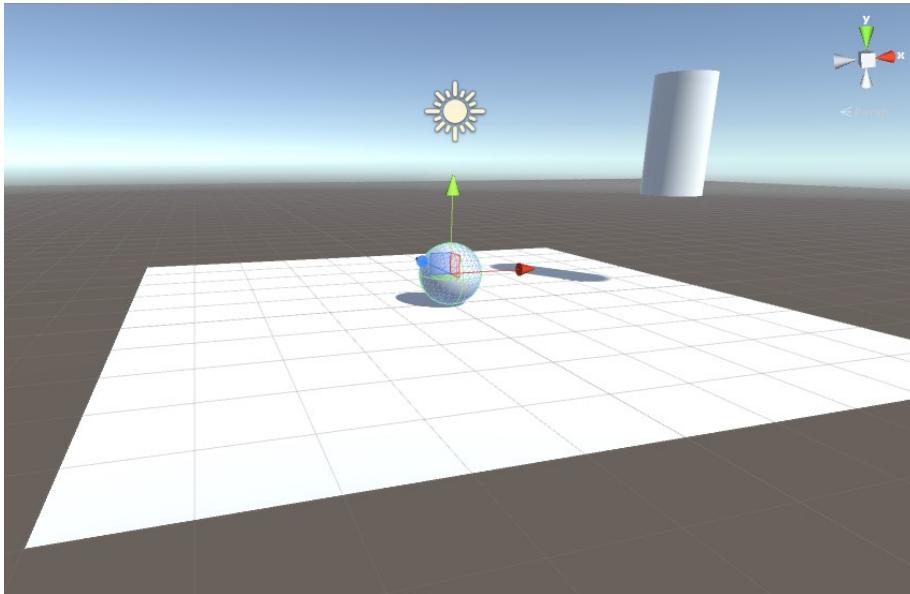
- [Equals](#) Returns true if the given vector is exactly equal to this vector.
- [Set](#) Set x, y and z components of an existing Vector3.
- [ToString](#) Returns a nicely formatted string for this vector.

**Static Methods:**

- [Angle](#) Returns the angle in degrees between from and to.
- [ClampMagnitude](#) Returns a copy of vector with its magnitude clamped to maxLength.
- [Cross](#) Cross Product of two vectors.
- [Distance](#) Returns the distance between a and b.
- [Dot](#) Dot Product of two vectors.
- [Lerp](#) Linearly interpolates between two vectors.
- [LerpUnclamped](#) Linearly interpolates between two vectors.
- [Max](#) Returns a vector that is made from the largest components of two vectors.
- [Min](#) Returns a vector that is made from the smallest components of two vectors.
- [MoveTowards](#) Moves a point current in a straight line towards a target point.
- [Normalize](#) Makes this vector have a magnitude of 1.

At the bottom right of the page, there is a watermark that says 'Активация Windows'.

# Методи MoveTowards i Lerp



# Приклади використання MoveTowards і Lerp

```
using UnityEngine;
public class Move : MonoBehaviour {
    private Transform target;
    private float speed = 1f;
    private Vector3 speedRot = Vector3.right * 50f;

    // Use this for initialization
    void Start () {
        target = GameObject.FindGameObjectWithTag("Player").transform;
    }

    // Update is called once per frame
    void Update () {
        transform.Rotate (speedRot * Time.deltaTime);
        transform.position=Vector3.Lerp(transform.position,target.position,speed*
Time.deltaTime);

    }
}
```

```
public static Vector3 Lerp (Vector3 from, Vector3 to, float t)
{
    t = Mathf.Clamp01 (t);
    return new Vector3 (from.x + (to.x - from.x) * t, from.y + (to.y - from.y) * t, from.z + (to.z - from.z) * t);
}
```

```
public static float Clamp01 (float value)
{
    if (value < 0f)
    {
        return 0f;
    }
    if (value > 1f)
    {
        return 1f;
    }
    return value;
}
```

```
public static Vector3 MoveTowards (Vector3 current, Vector3 target, float maxDistanceDelta)
{
    Vector3 a = target - current;
    float magnitude = a.magnitude;
    if (magnitude <= maxDistanceDelta || magnitude == 0f)
    {
        return target;
    }
    return current + a / magnitude * maxDistanceDelta;
}
```

```
public float magnitude
{
    get
    {
        return Mathf.Sqrt (this.x * this.x + this.y * this.y + this.z * this.z);
    }
}
```

Дякую за увагу!



# Тема 3. Використання скриптів в Unity3d

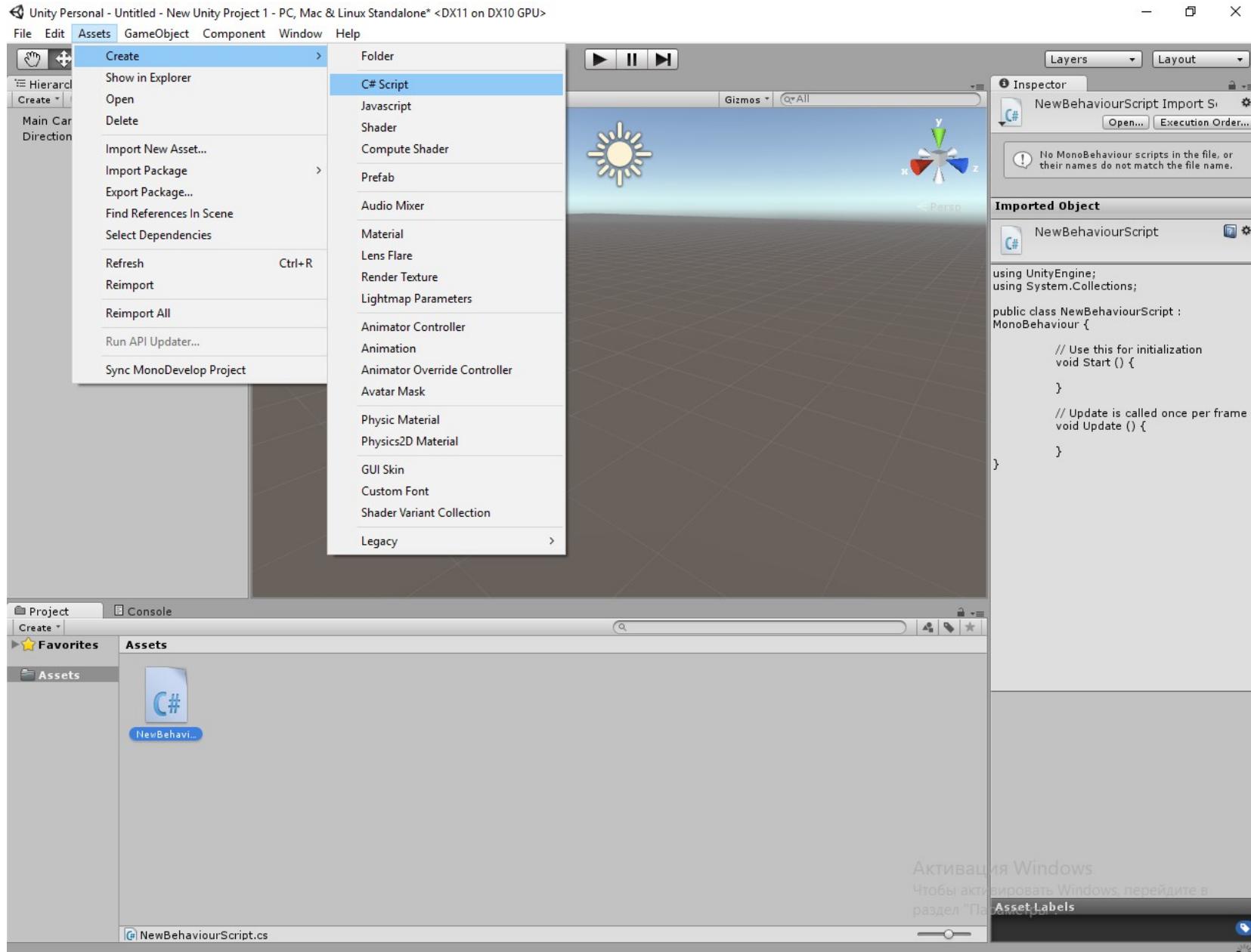
```
Character
AutoBackToTitle.cs
ClickToStart.cs
Explosion.cs
Explosive.cs
Fire.cs
FloorSection.cs
GameControl.cs
GameGUI.cs
Hose.cs
MapIcons.cs
MessageGUI.cs
MoveBetweenPoints
Player.cs
Priority Particle Add.
PriorityAlphaParticle
SceneChanger.cs
SmokeParticles.cs
WaterHoseParticles
WaterSplash.cs
World.cs
Assets

50     vignette.blur = (1-health) * 2 + smokeEffect * 30 * (health - 1);
51     vignette.blurDistance = (1-health) * 2 + smokeEffect * 30 * (health - 1);
52     vignette.chromaticAberration = heatEffect * 30;
53 }
54
55
56 void OnTriggerStay(Collider c)
57 {
58     var fire = c.GetComponent<Fire>();
59     if (fire && fire.alive)
60     {
61         float dist = 1-((transform.position - fire.transform.position).magnitude);
62         NearHeat(dist);
63     }
64
65     var smoke = c.GetComponent<ClickToStart>();
66     if (smoke && smoke.GetComponent<ParticleFunction>())
67     {
68         float dist = 1-((transform.position - smoke.transform.position).magnitude);
69         NearSmoke(dist);
70     }
71 }
72
73 void OnCollisionEnter(Collision c)
74 {
75     var healthBox = c.gameObject.GetComponent<HealthBox>();
76     if (healthBox)
77     {
78         if (healthBox.health <= 0)
```

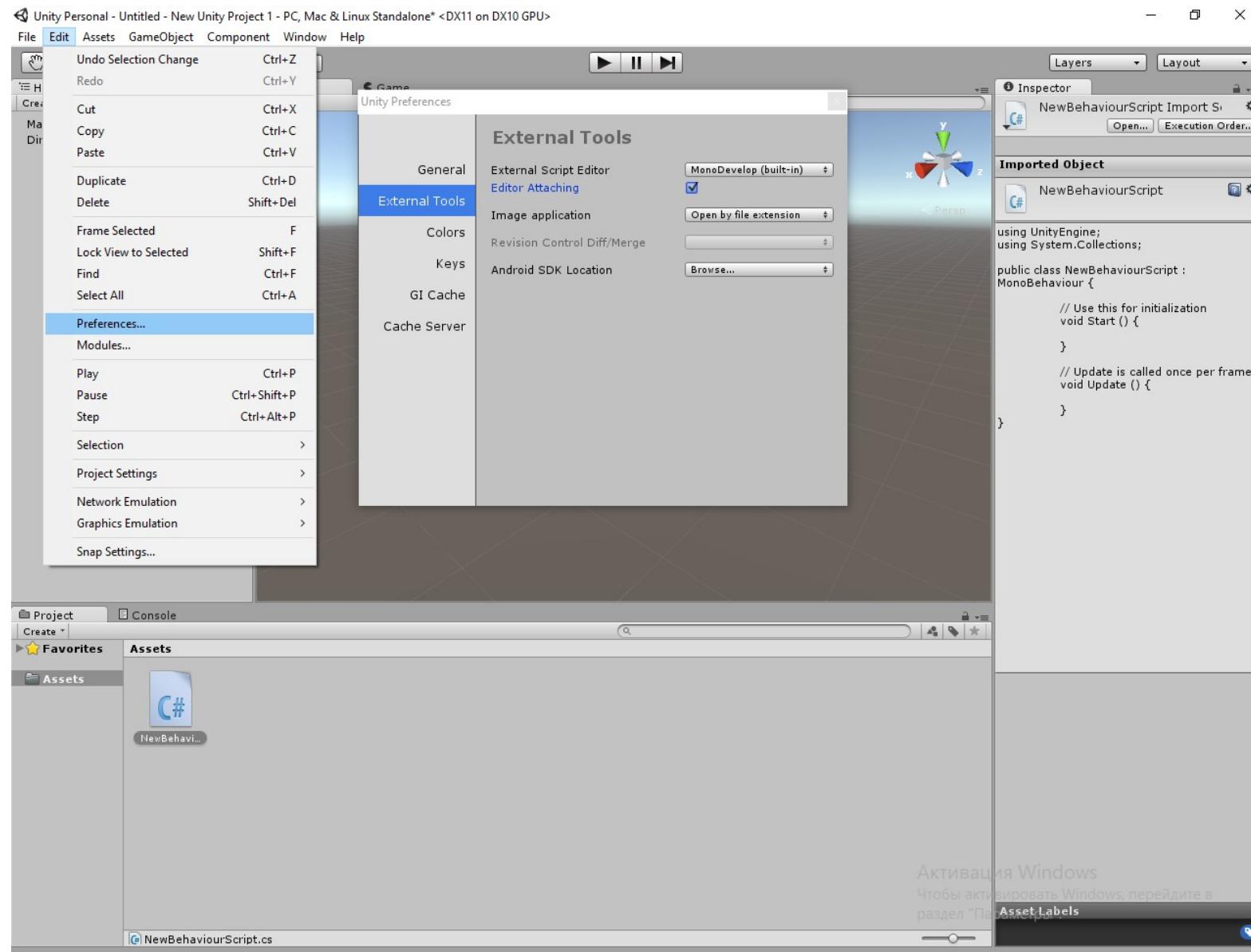
<https://docs.unity3d.com/ru/current/Manual/ScriptingSection.html>

<https://docs.unity3d.com/ru/current/ScriptReference/index.html>

# Створення скрипта



# Вибір редактора для роботи з скриптами



# Код автоматично згенерованого скрипта

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

```
public class MainPlayer : MonoBehaviour {
```

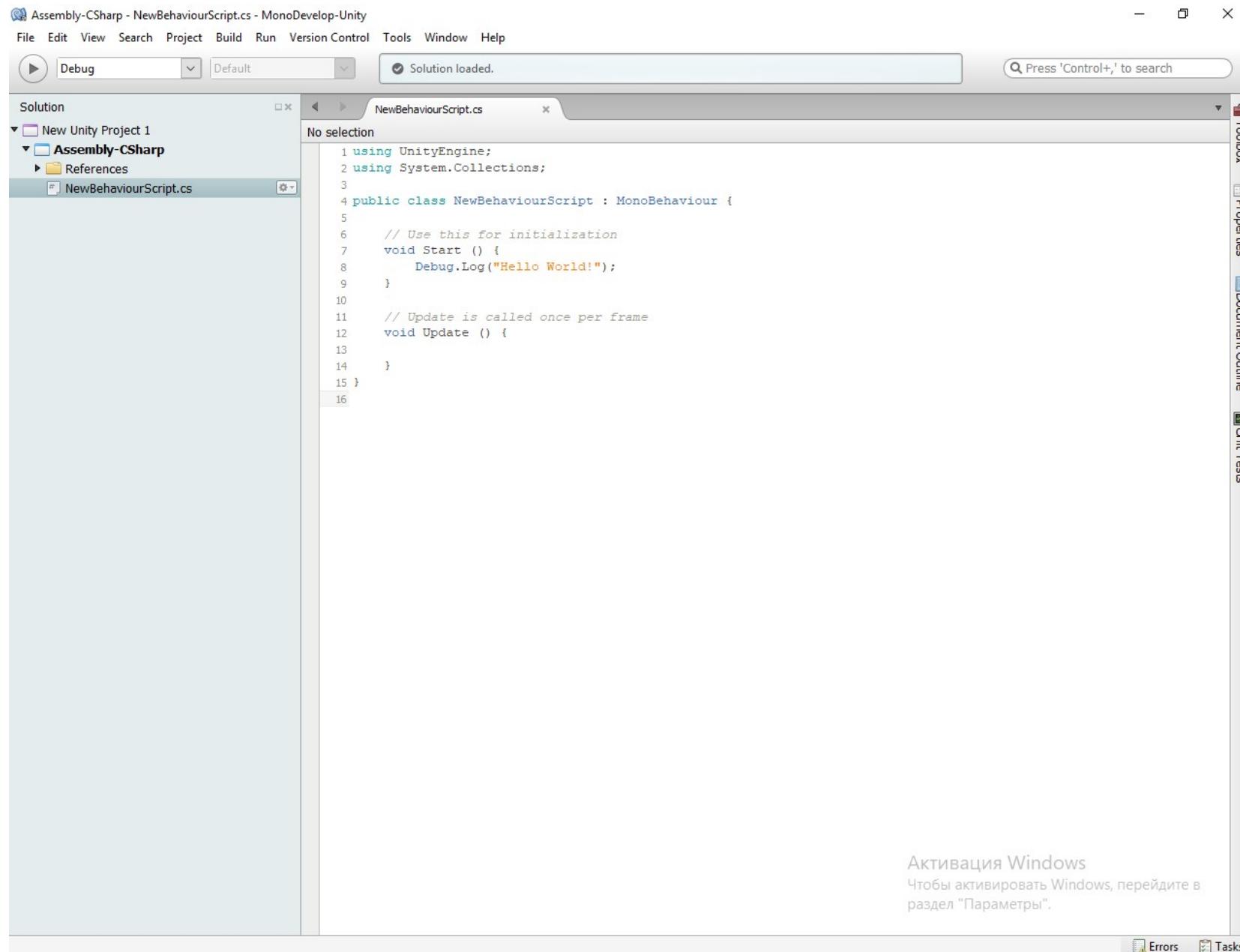
```
    // Use this for initialization  
    void Start () {
```

```
}
```

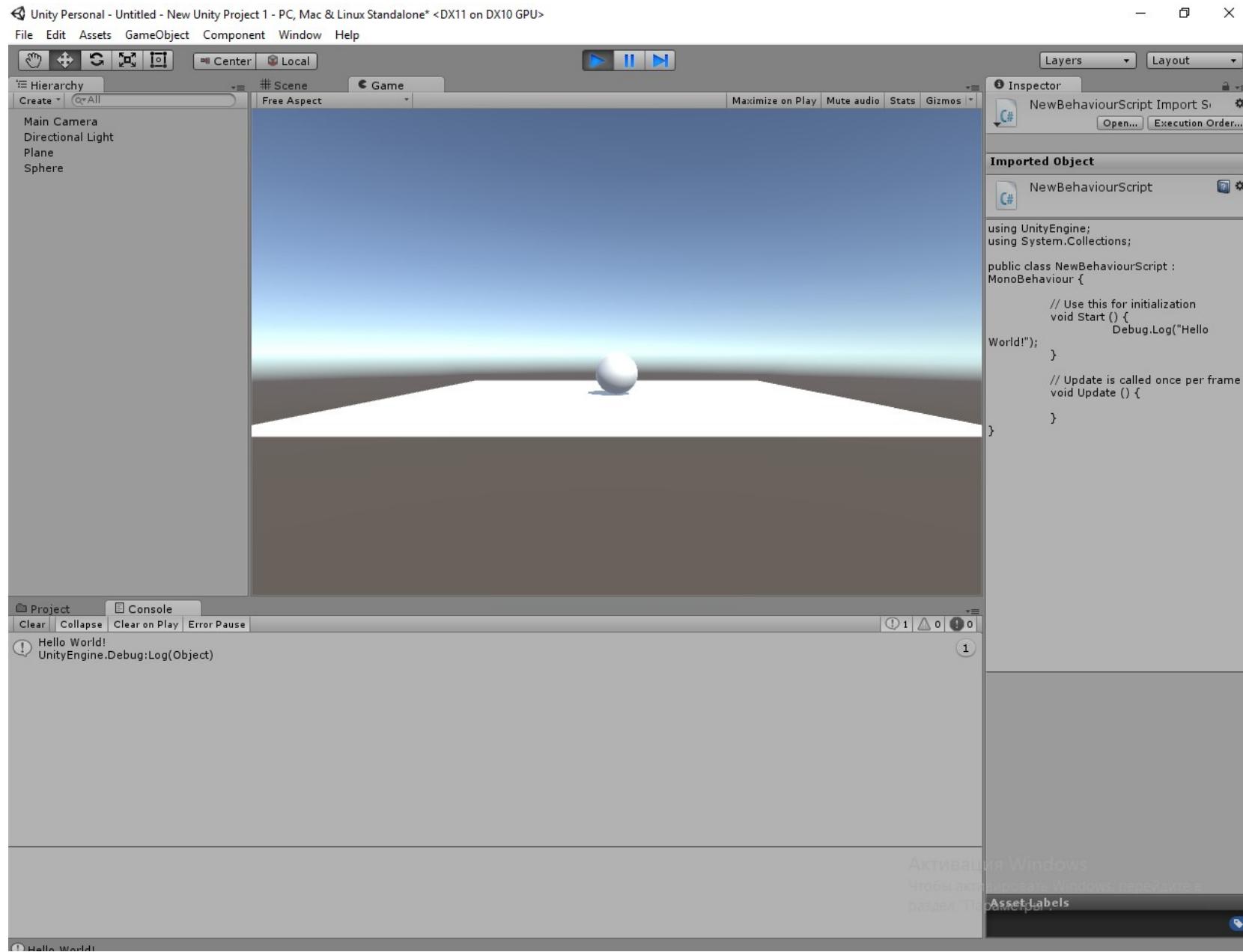
```
    // Update is called once per frame  
    void Update () {
```

```
    }
```

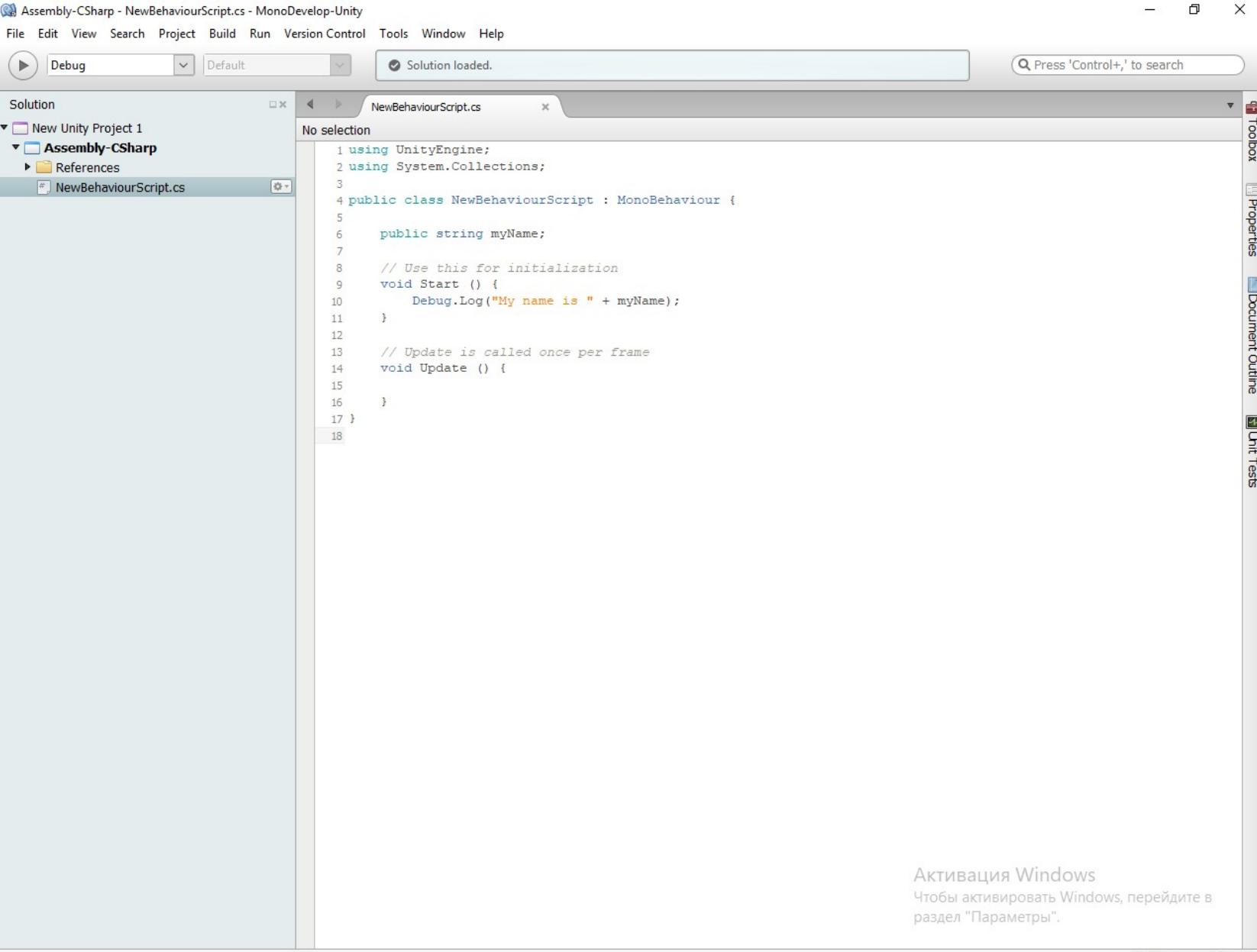
# Середовище розробки MonoDevelop



# Робота з консоллю



# Публічні поля класів

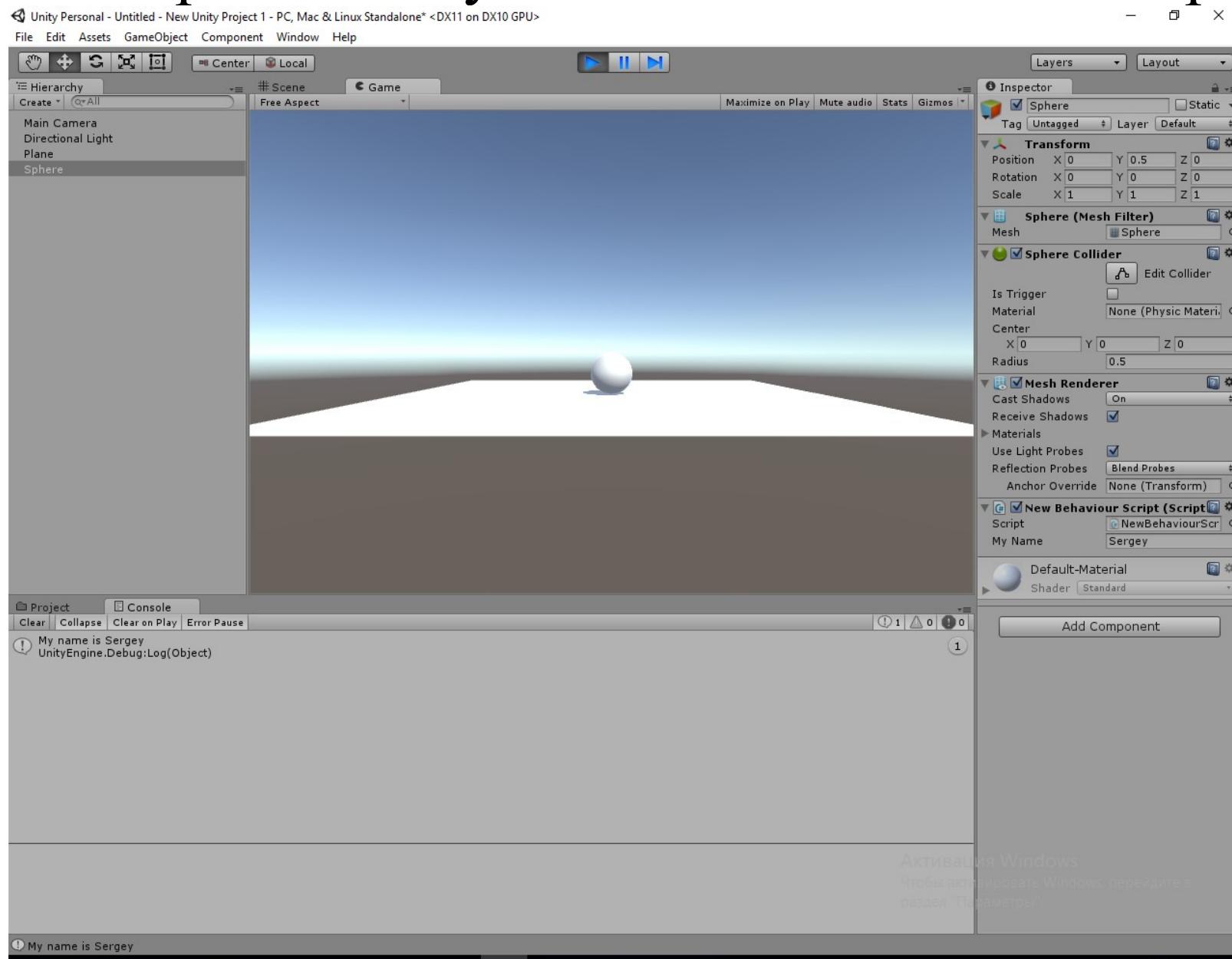


The screenshot shows the MonoDevelop-Unity IDE interface. The title bar reads "Assembly-CSharp - NewBehaviourScript.cs - MonoDevelop-Unity". The menu bar includes File, Edit, View, Search, Project, Build, Run, Version Control, Tools, Window, and Help. The toolbar has "Debug" and "Default" buttons. A status bar at the top right says "Solution loaded." and "Press 'Control+,' to search". The left sidebar shows the "Solution" tree with "New Unity Project 1" and "Assembly-CSharp" selected, containing "References" and "NewBehaviourScript.cs". The main editor window displays the following C# code:

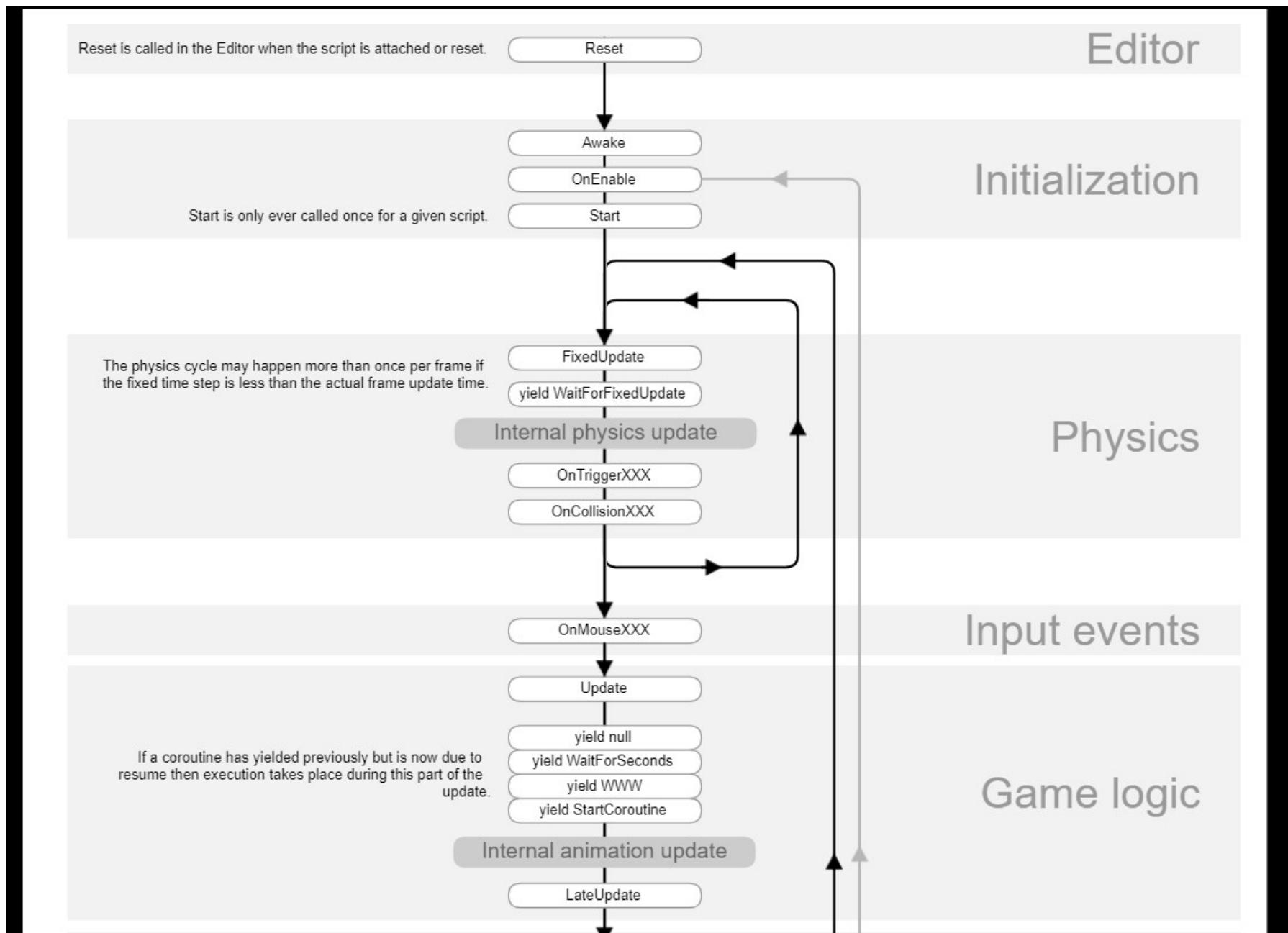
```
1 using UnityEngine;
2 using System.Collections;
3
4 public class NewBehaviourScript : MonoBehaviour {
5
6     public string myName;
7
8     // Use this for initialization
9     void Start () {
10         Debug.Log("My name is " + myName);
11     }
12
13     // Update is called once per frame
14     void Update () {
15
16     }
17 }
18
```

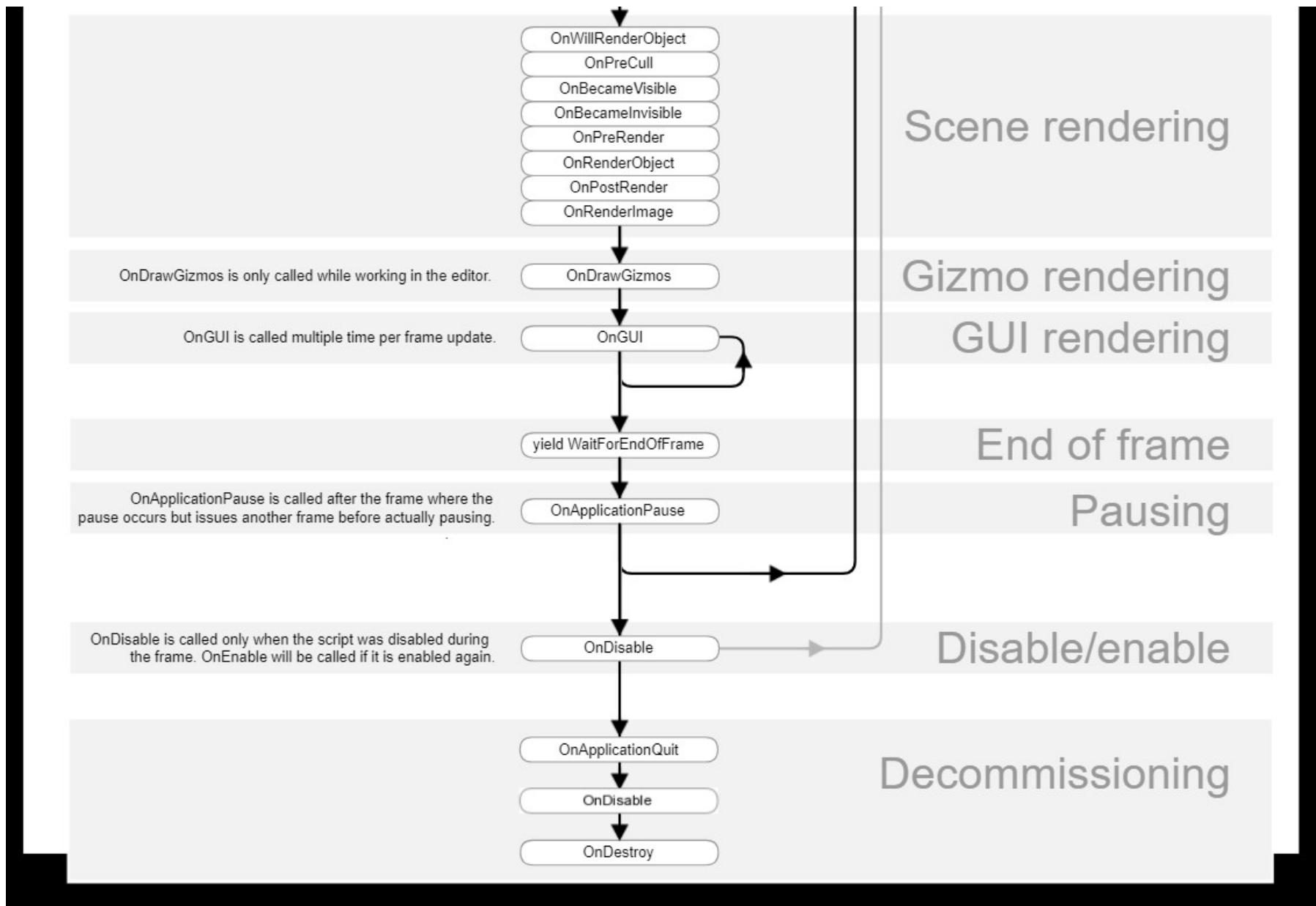
The right sidebar contains tabs for Toolbox, Properties, Document Outline, and Unit Tests. A message at the bottom right says "Активация Windows" and "Чтобы активировать Windows, перейдите в раздел \"Параметры\"." The bottom navigation bar includes Errors and Tasks buttons.

# Відображення публічних полів в інспекторі



# БЛОК-СХЕМА ЖИТТЕВОГО ЦИКЛУ СКРИПТА





<https://docs.unity3d.com/ru/current/Manual/ExecutionOrder.html>

# Основні функції подій

**Start** викликається до оновлення першого кадру чи фізики об'єкту.

**Awake** викликається для кожного об'єкту в сцені в момент її завантаження. Всі функції Awake повинні відпрацювати до першого виклику функції Start.

**Update** викликається перед розрахунком анімації та перед відображенням кадру.

**FixedUpdate** викликається перед кожним оновленням фізичних даних.

**LateUpdate** викликається після функції Update

**OnGUI** викликається при взаємодії з користувальським інтерфейсом.

Функції **OnCollisionEnter**, **OnCollisionStay** і **OnCollisionExit** будуть викликані на початку, продовженні і завершенні контакту ігрових об'єктів.

Функції **OnTriggerEnter**, **OnTriggerStay** і **OnTriggerExit** будуть викликані коли колайдер об'єкта налаштований як Trigger.

# Звернення до компонентів об'єкту

```
Rigidbody rb = GetComponent<Rigidbody>();
```

```
rb.mass = 10f;
```

```
using UnityEngine;
```

```
public class Pole : MonoBehaviour {
```

```
    public int a=3;
```

```
    void Start () { }
```

```
    void Update () { }
```

```
}
```

```
using UnityEngine;

public class NewBehaviourScript : MonoBehaviour {

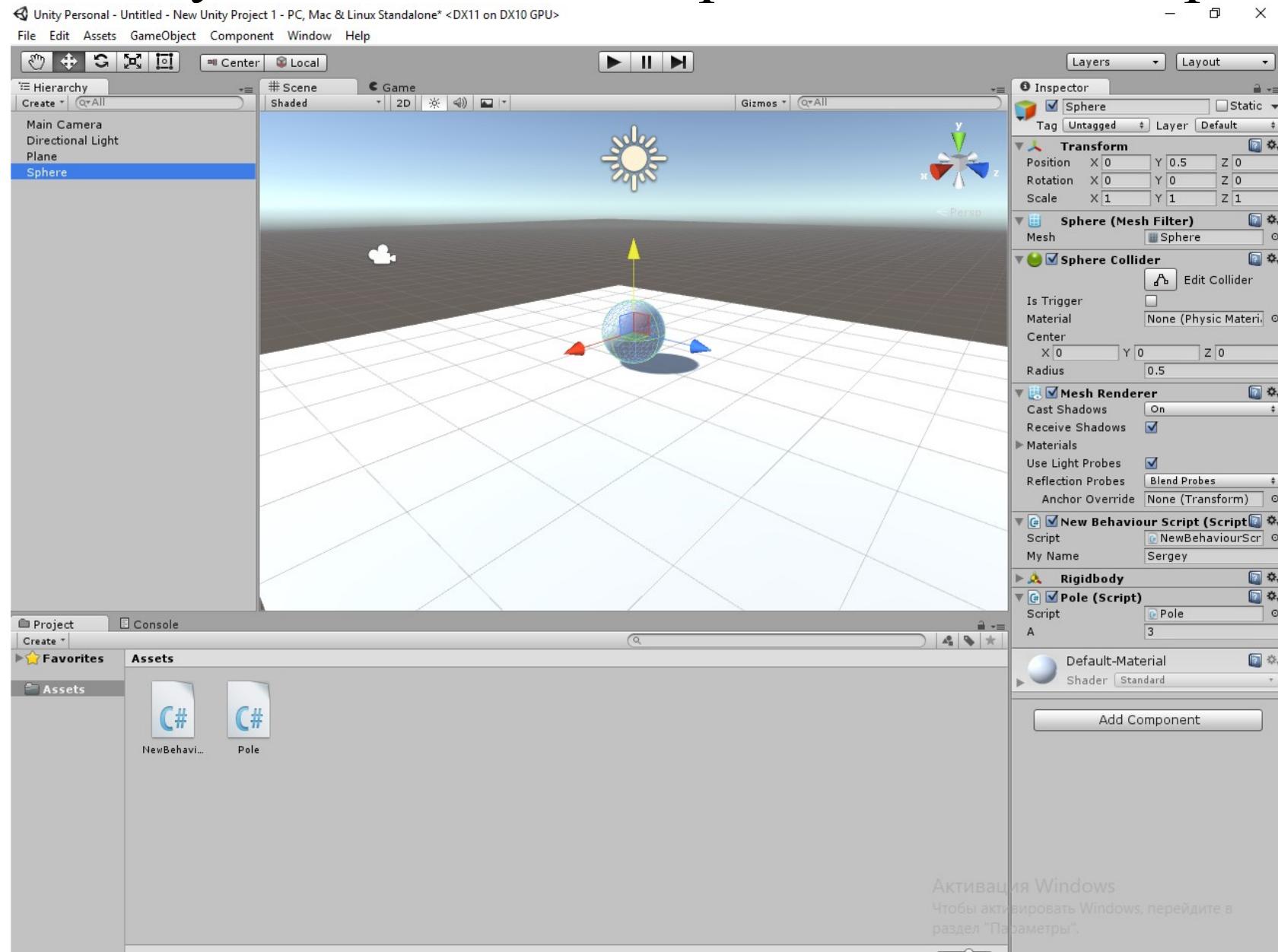
    public string myName;

    // Use this for initialization
    void Start () {
        Debug.Log("My name is " + myName);
        var rb = GetComponent<Pole>();
        Debug.Log(rb.a);
    }

    // Update is called once per frame
    void Update () {

    }
}
```

# Зв'язування об'єктів через поле інспектора



# Доступ до дочірніх об'єктів через компонент Transform батьківського об'єкту

using UnityEngine;

```
public class WaypointManager : MonoBehaviour {
```

```
    public Transform[] waypoints;
```

```
    void Start() {
```

```
        waypoints = new Transform[transform.childCount];  
        int i = 0;
```

```
        foreach (Transform t in transform) {
```

```
            waypoints[i++] = t;
```

```
        }
```

```
}
```

```
}
```

# Клас Transform

The screenshot shows the MonoDevelop-Unity IDE interface. The title bar reads "Assembly Browser - MonoDevelop-Unity". The menu bar includes File, Edit, View, Search, Project, Build, Run, Version Control, Tools, Window, and Help. The toolbar has a "Debug" button and a dropdown set to "Default". The assembly browser tab is active, showing the "Assembly-CSharp" assembly. The left sidebar shows the "Solution" tree with "New Unity Project 1" and "Assembly-CSharp" selected, containing "References", "NewBehaviourScript.cs", and "Pole.cs". The main code editor displays the source code for the "Transform" class. The code is as follows:

```
using ...
namespace UnityEngine
{
    public class Transform : Component, IEnumerable
    {
        // Properties
        public extern int childCount...
        public Vector3 eulerAngles...
        public Vector3 forward...
        public extern bool hasChanged...
        public Vector3 localEulerAngles...
        public Vector3 localPosition...
        public Quaternion localRotation...
        public Vector3 localScale...
        public Matrix4x4 localToWorldMatrix...
        public Vector3 lossyScale...
        public Transform parent...
        internal extern Transform parentInternal...
        public Vector3 position...
        public Vector3 right...
        public extern Transform root...
        public Quaternion rotation...
        public Vector3 up...
        public Matrix4x4 worldToLocalMatrix...
    }
}
```

A tooltip "Активация Windows" and text "Чтобы активировать Windows, перейдите в раздел \"Параметры\"." are visible at the bottom right. The right sidebar includes tabs for Properties, Document Outline, and Unit Tests.

# Пошук об'єктів за їх ім'ям або тегом

Transform.Find

GameObject.Find

GameObject.FindWithTag

GameObject.FindGameObjectsWithTag

## Приклад коду

```
GameObject player;
```

```
GameObject[] enemies;
```

```
void Start() {
```

```
    player = GameObject.FindGameObjectWithTag("Player");
```

```
    enemies = GameObject.FindGameObjectsWithTag("Enemy");
```

```
}
```

# Створення клонів та видалення ігрових об'єктів

```
Public GameObject enemy;
```

```
void Start() {  
    for (int i = 0; i < 5; i++) {  
        Instantiate(enemy);  
    }  
}
```

```
void OnCollisionEnter(Collision otherObj) {  
    if (otherObj.gameObject.tag == "Missile") {  
        Destroy(gameObject,.5f);  
    }  
}
```

```
Destroy(this);
```

Дякую за увагу!



# Тема 2. Робота з текстурами і матеріалами.

## Редактор ландшафтів Terrain



<https://docs.unity3d.com/ru/current/Manual/Shaders.html>

<https://docs.unity3d.com/ru/current/Manual/script-Terrain.html>

# Текстури

Текстура - растрове зображення, яке накладається на поверхню полігональної моделі для надання їй кольору, окраски чи ілюзії рельєфу.

Текстуру можна уявити собі як рисунок на поверхні скульптури.

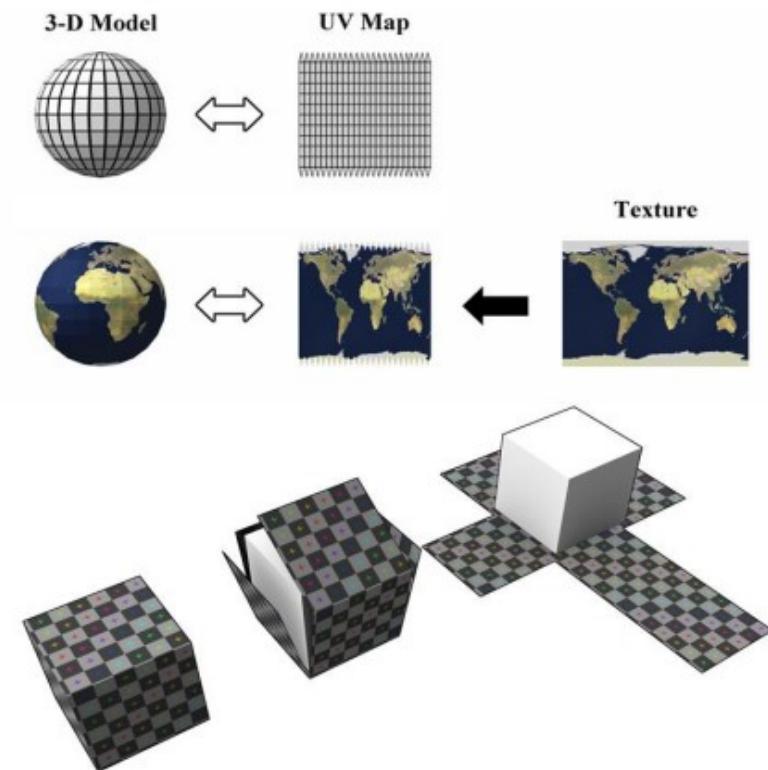
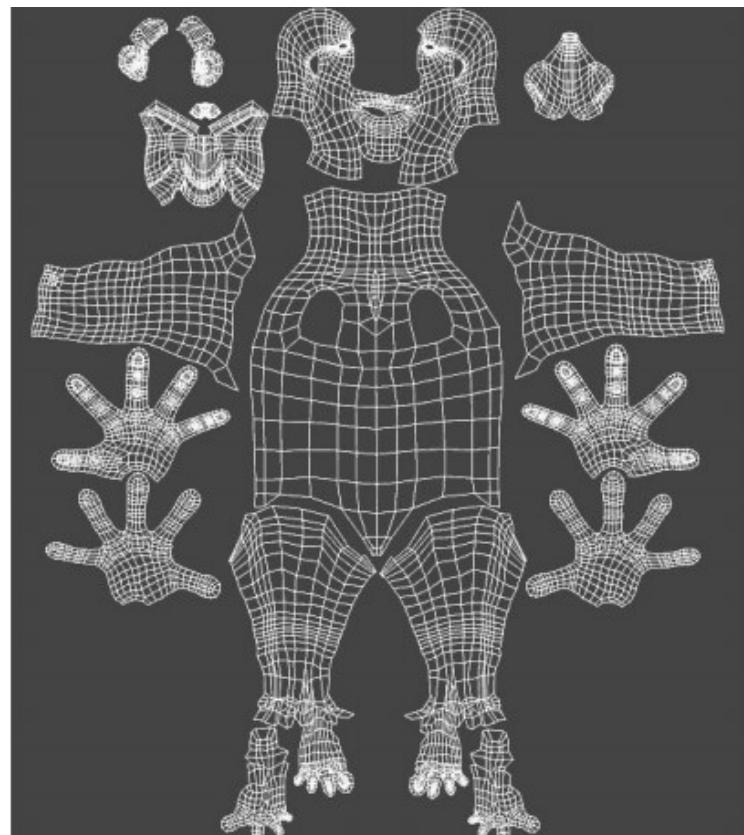
Використання текстур дозволяє відтворити малі об'єкти поверхні, створення яких полігонами виявилося б дуже затратним з точки зору ресурсів. Наприклад, шрами на шкірі, складки на одязі тощо.



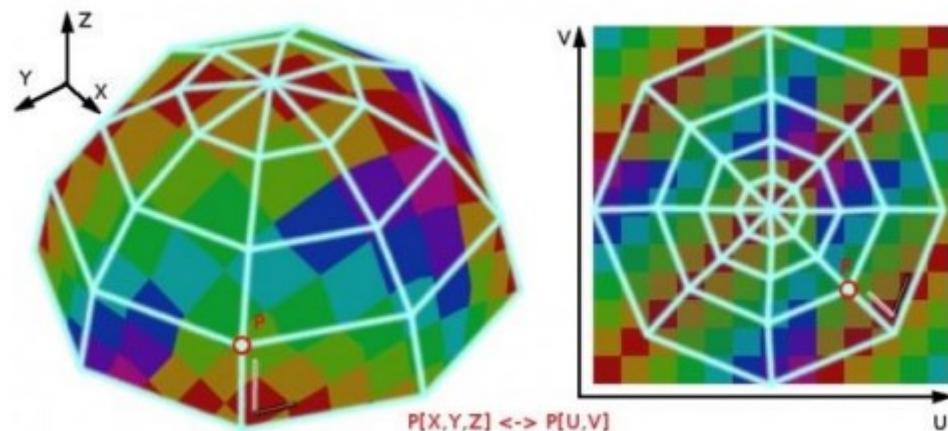
# UV mapping

Для правильного нанесення текстур на поверхню моделі необхідно спочатку зробити її UVW розгортку: оболонка об'єкта поділяється на окремі фрагменти, з'єднані між собою швами, тобто модель практично «Розкроюється», як плаття в ательє.

Найкращий спосіб натягування 2D текстур на 3D-об'єкти називається "UV mapping". Ви берете тривимірний (координати X, Y, Z) об'єкт і розвертаєте його на двовимірну площину-зображення (координати U, V). кольори картинки проектуються на меш, і полігони набувають забарвлення.

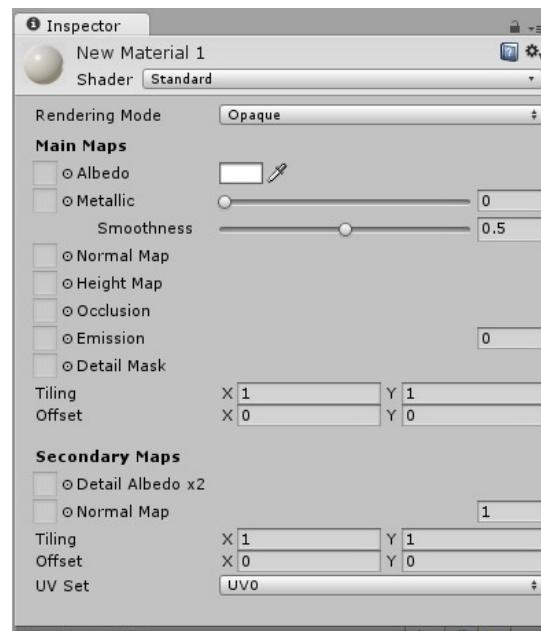


На цьому прикладі ви можете бачити, що розміри і форма квадратиків на 3d моделі відрізняються від 2d площини. Ця різниця обумовлена розтягуванням (mapping) текстури (UV карти) по полігонам. Виділена точка на малюнку показує, як точка на об'ємному об'єкті може бути відображенна на плоскій карті.



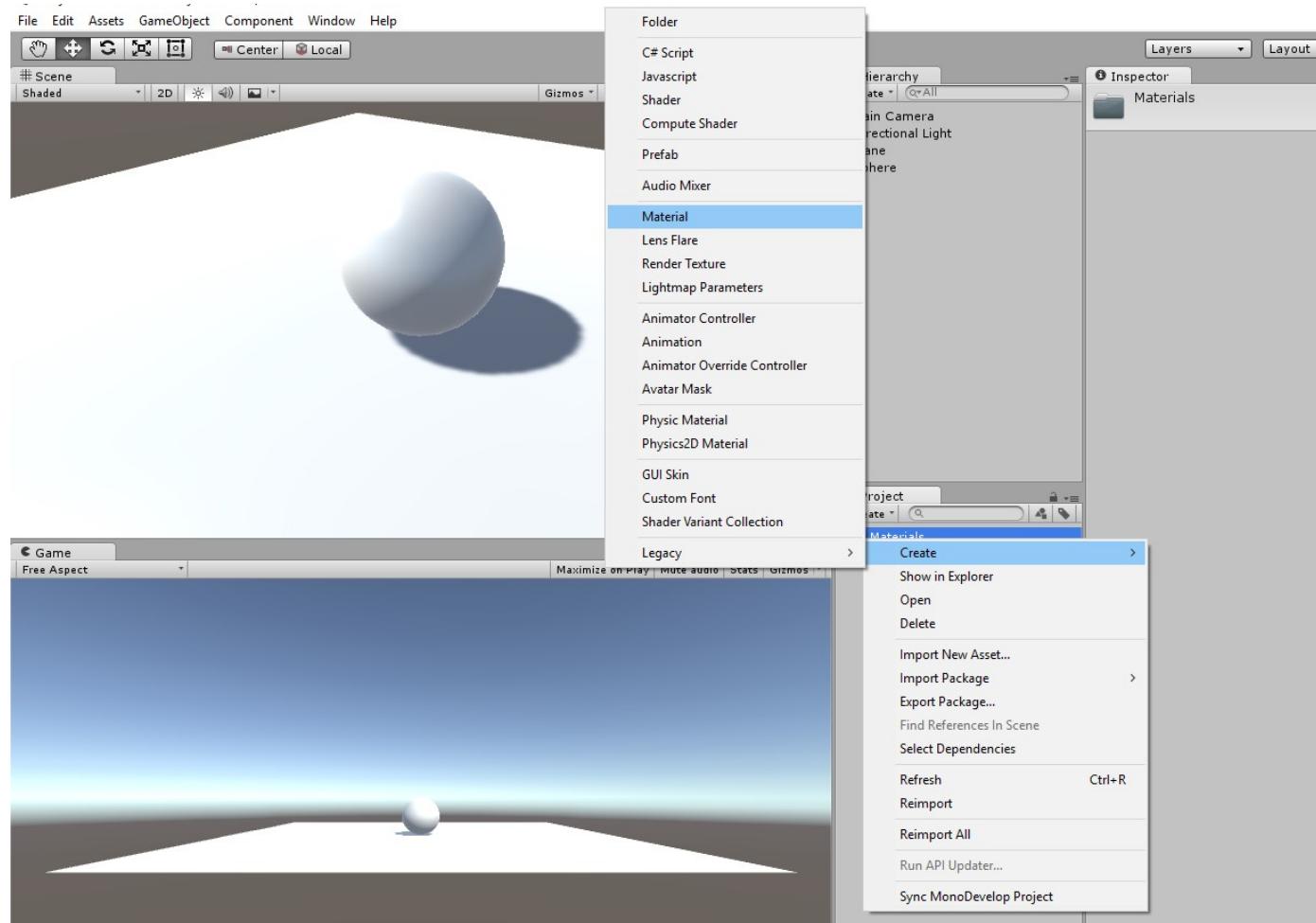
## Карти

- Карта відблисків (specular map)
- Карта нормалей (normal map)
- Карта висот (height map)
- Карта затінення (light map)
- Карта емісії (emission map)
- Карта шорсткості (roughness map)
- Карта аборбції (occlusion map)
- Карта металічності (metallic map)



# Матеріали

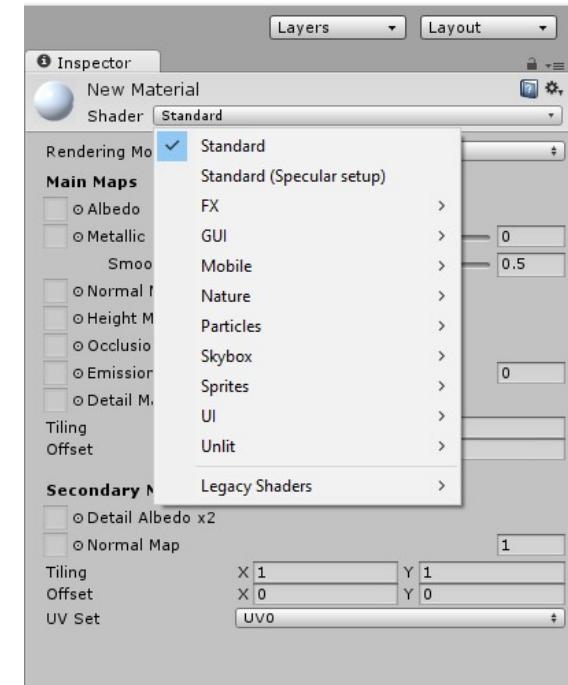
Матеріал - математична модель, що описує параметри поверхні (текстури, заломлення, відображення, прозорість). Модель описується за допомогою програми – Shader. За замовчуванням для нових матеріалів присвоюється «Стандартний шейдер», при цьому всі властивості карти порожні.



# Вбудовані шейдери:

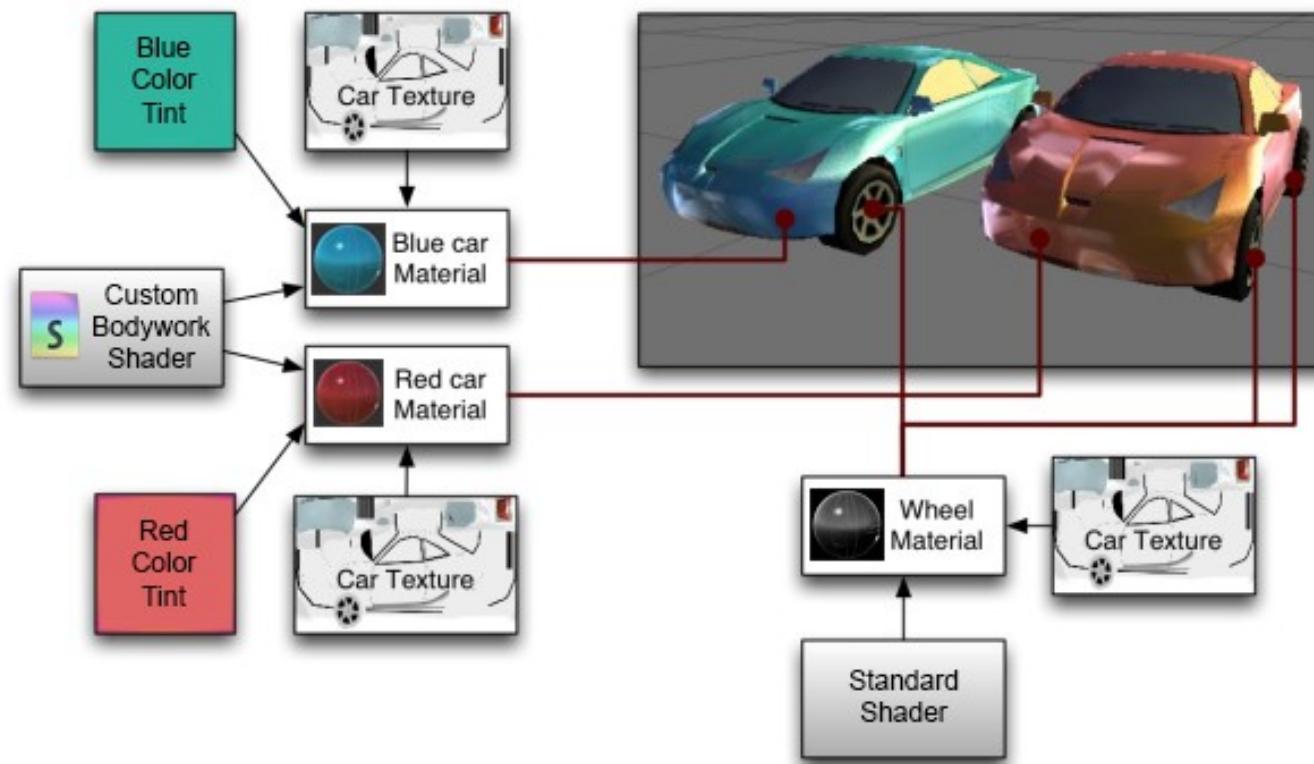
- FX : водні та світлові ефекти.
- GUI : відображення графічного інтерфейсу користувача користувача.
- Mobile : спрощений високопродуктивний шейдер для мобільних пристроїв.
- Nature : дерева і поверхня землі.
- Particles : ефекти системи частинок.
- Skybox : Для відтворення фонових середовищ за всією геометрією.
- Sprites : Для використання з 2D спрайтами.
- Unlit : для відтворення.
- Legacy : велика колекція застарілих шейдерів, які були замінені стандартним

рейдером.



Можна також створювати свої шейдери з використанням мови ShaderLab.

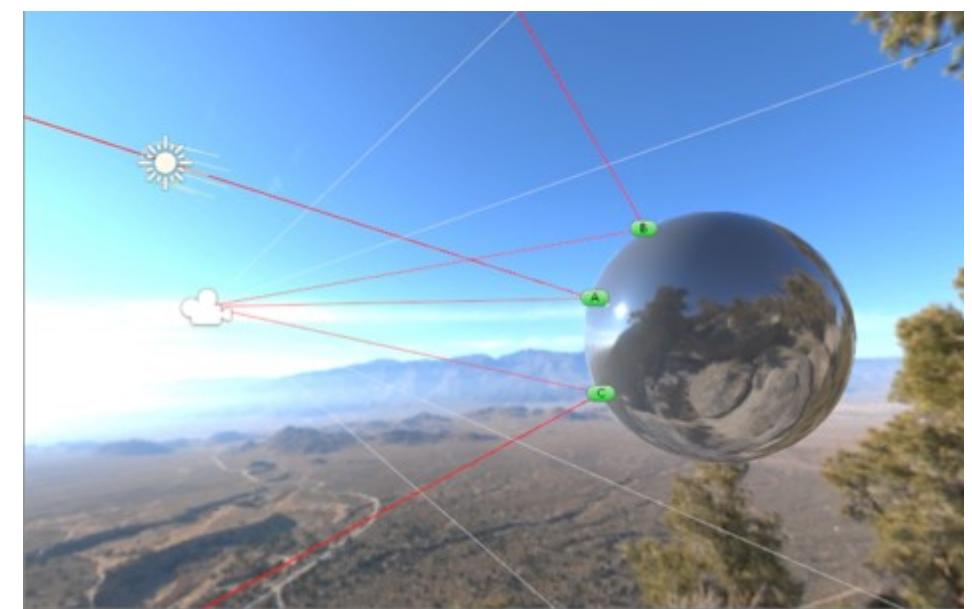
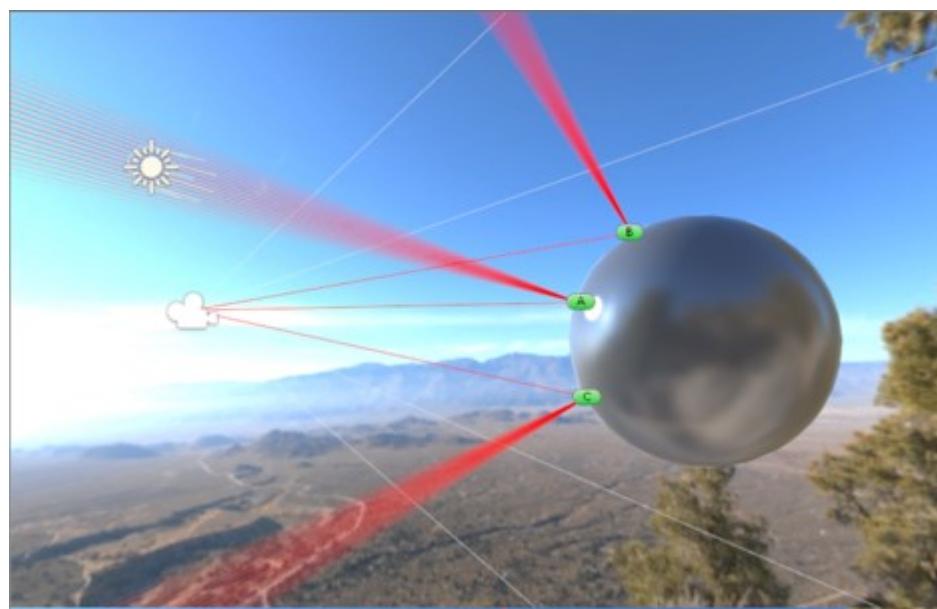
Приклад можливої комбінації з використанням трьох матеріалів, двох шейдерів і однієї текстури.



# Стандартний шейдер

Стандартний шейдер Unity - це вбудований шейдер із повним набором функцій. Він може бути використаний для виготовлення "реальних" об'єктів, таких як камінь, дерево, скло, пластик та метал, і підтримує широкий спектр шейдерних типів і комбінацій. Його функції вмикаються або вимикаються, просто використовуючи або не використовуючи різні слоти текстури та параметри у редакторі матеріалу.

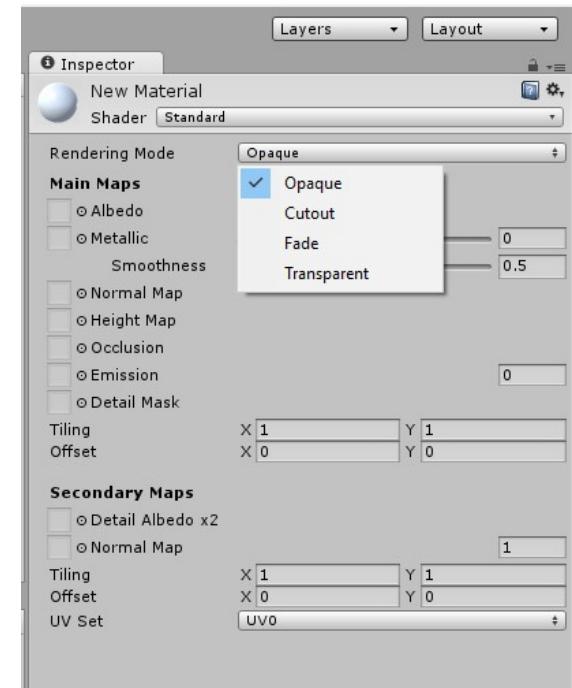
Стандартний шейдер також включає в себе вдосконалену модель освітлення, що називається Physical Based Shading. Фізичне затінення (PBS) імітує взаємодію між матеріалами та світлом, що імітує реальність. Він найкращим чином працює в ситуаціях, коли освітлення та матеріали повинні існувати разом інтуїтивно та реалістично.

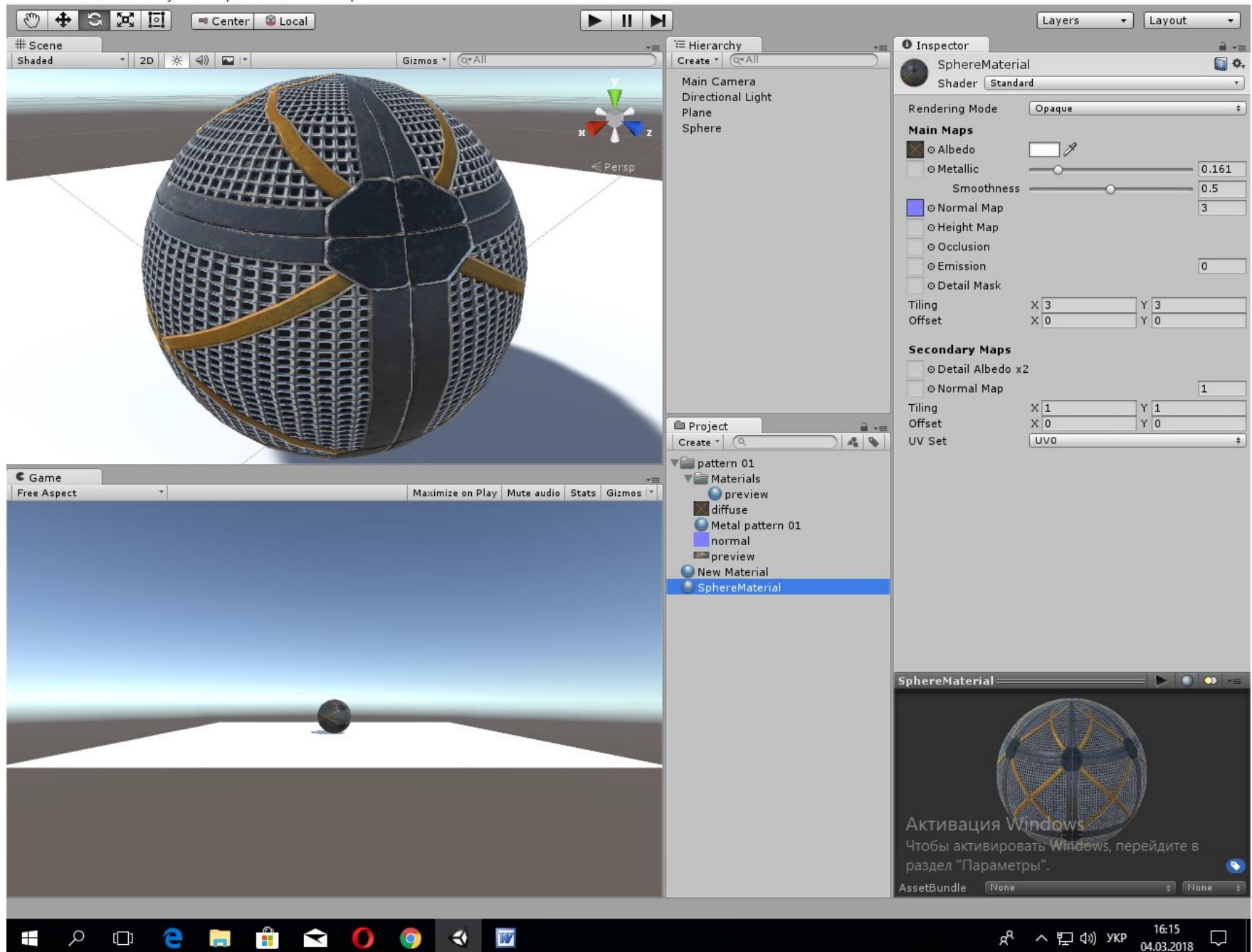


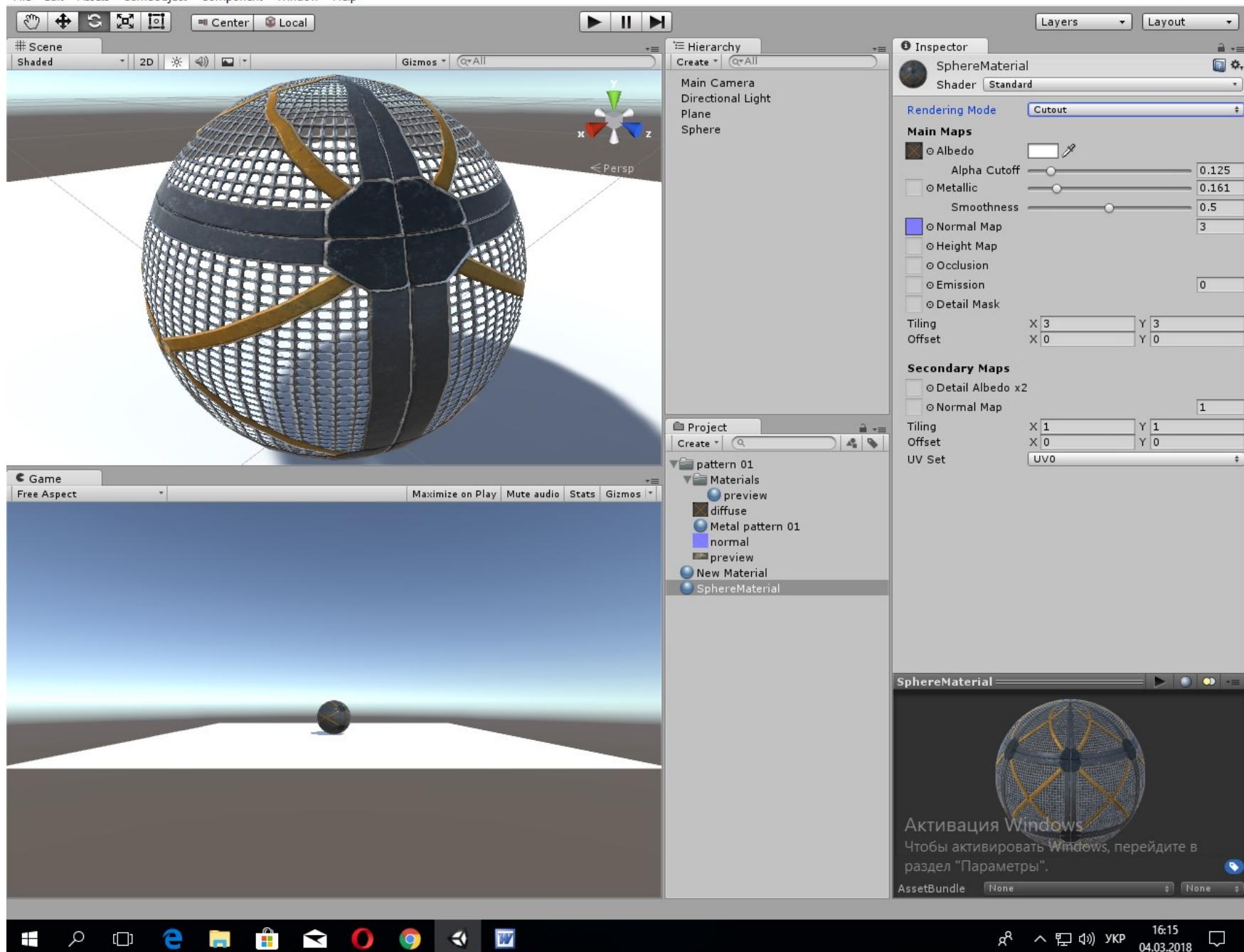
# Режим рендеринга

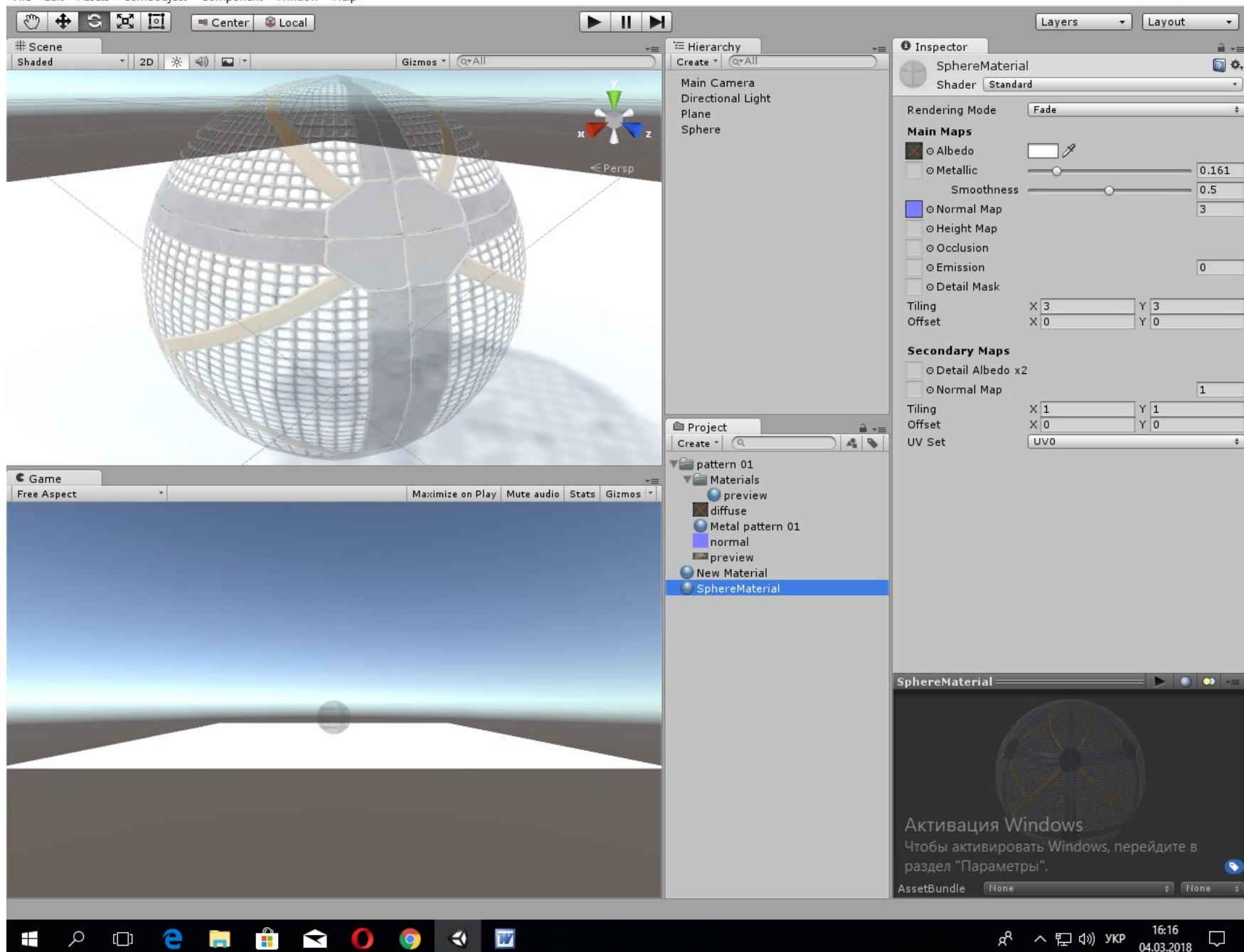
Перший параметр матеріалу в стандартному шейдері – це режим візуалізації. Цей режим дозволяє вам вибрати, чи використовується об'єкт прозорість, і якщо так, то який тип режиму накладання використовувати.

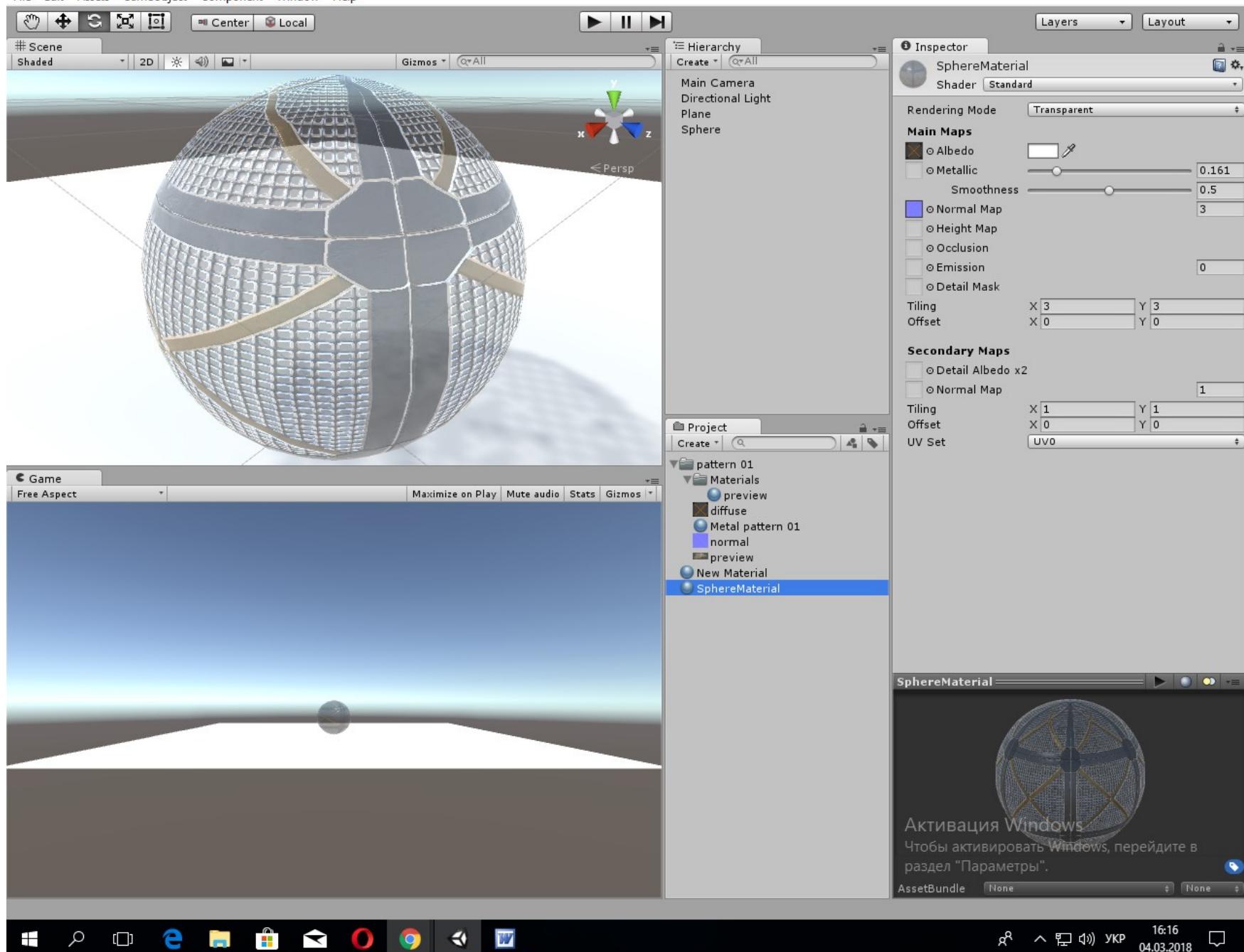
- Opaque – режим за замовчуванням і підходить для звичайних твердих об'єктів без прозорих областей.
- Cutout - Дозволяє створювати прозорі ефекти, які переходи між непрозорими та прозорими областями. У цьому режимі немає напівпрозорих областей, текстура яких або 100% непрозора, або невидима. Це корисно при використанні прозорості для створення форми таких матеріалів, як листя, або тканина з отворами.
- Transparent - Підходить для візуалізації прозорих матеріалів, таких як прозорий пластик або скло. У цьому режимі сам матеріал буде приймати значення прозорості (на основі альфа-каналу фактури та альфа колірного відтінку), однак відблиски та освітлення висвітлюються в повній мірі, як у випадку з реальними прозорими матеріалами.



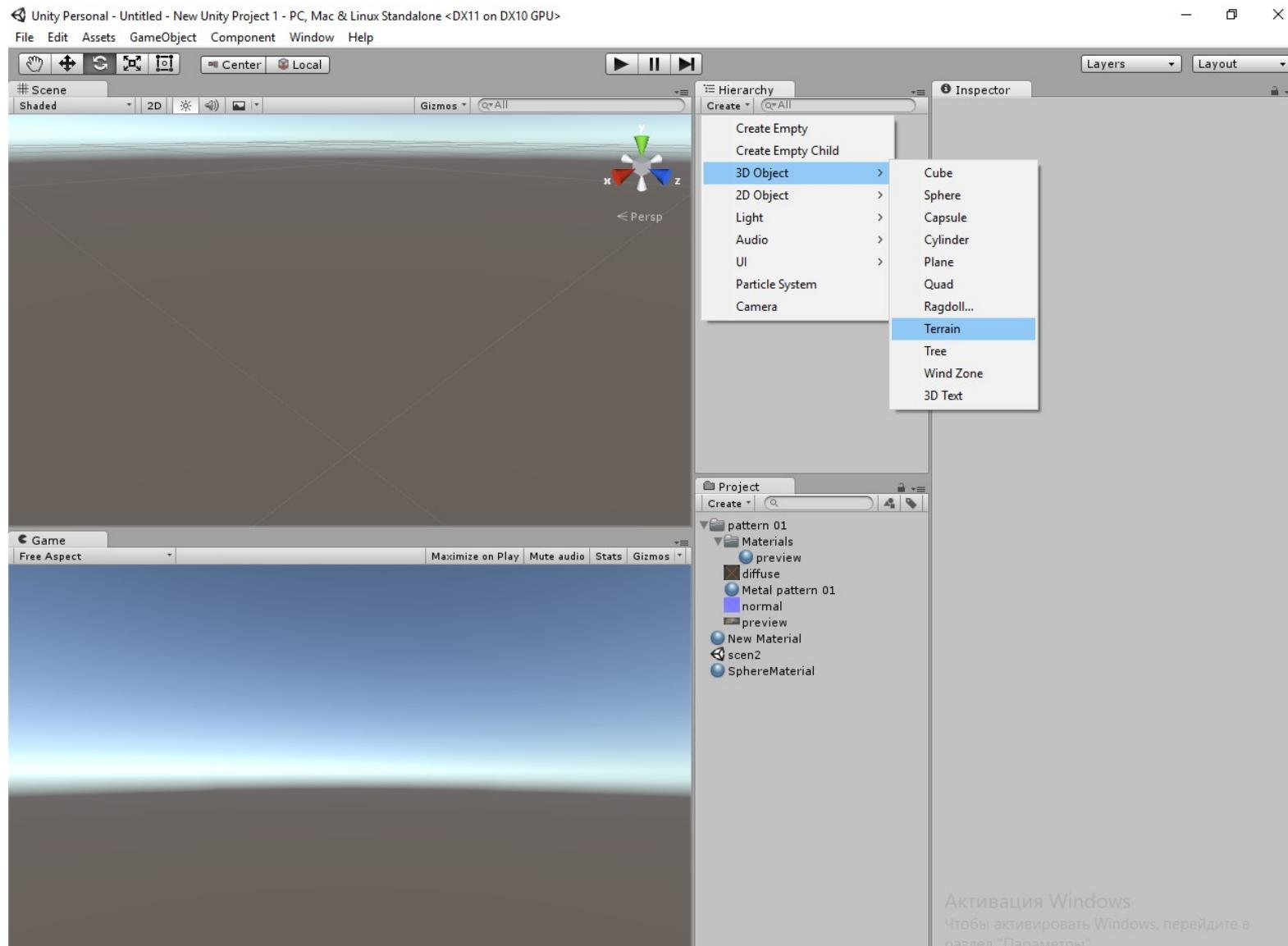


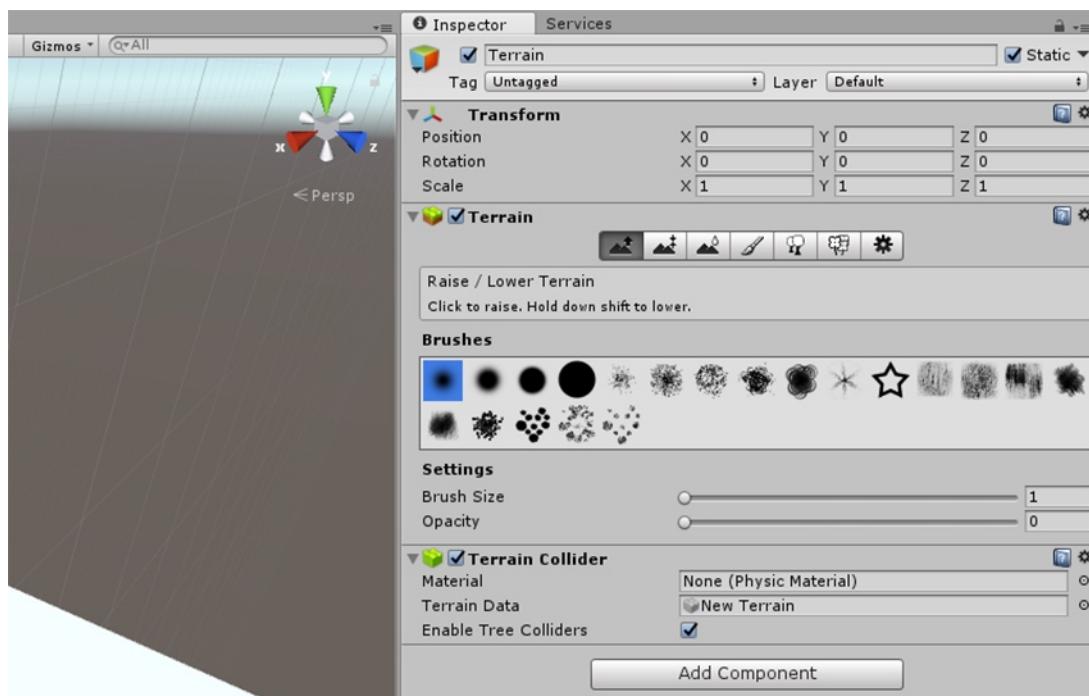
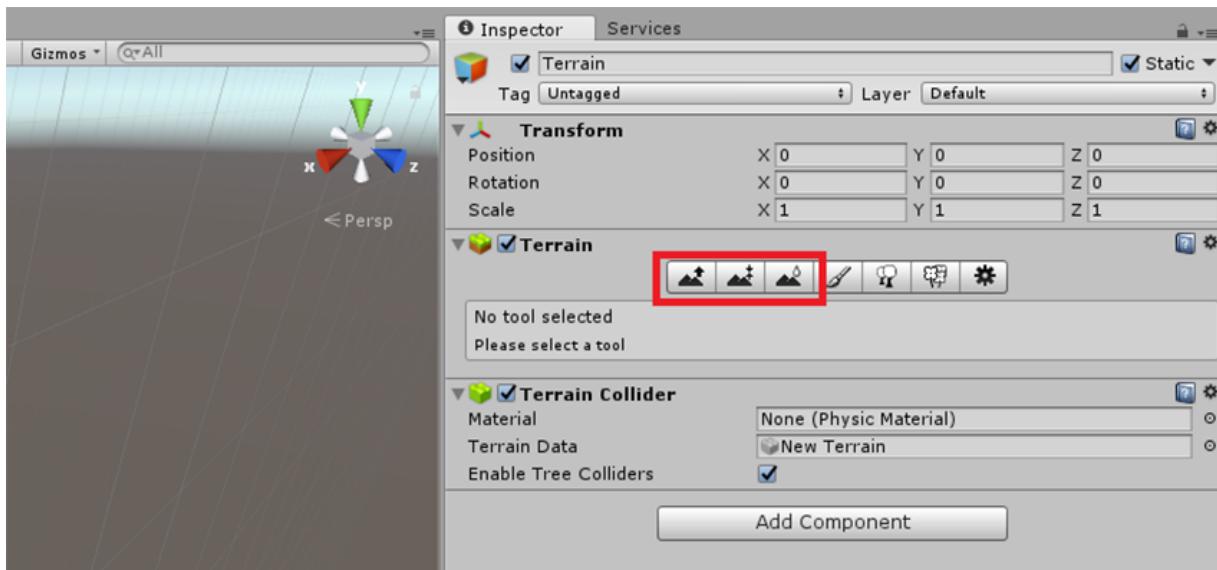


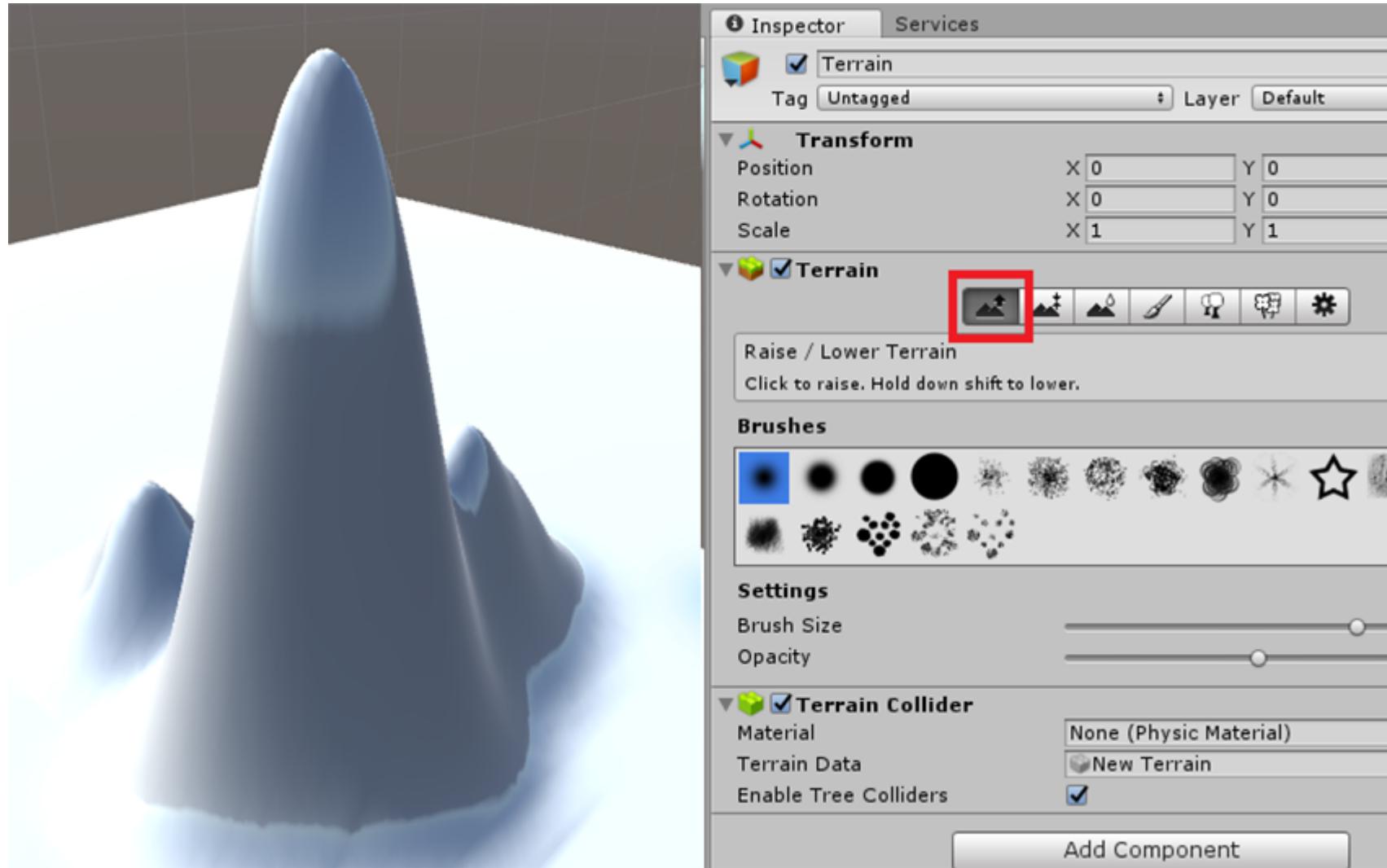


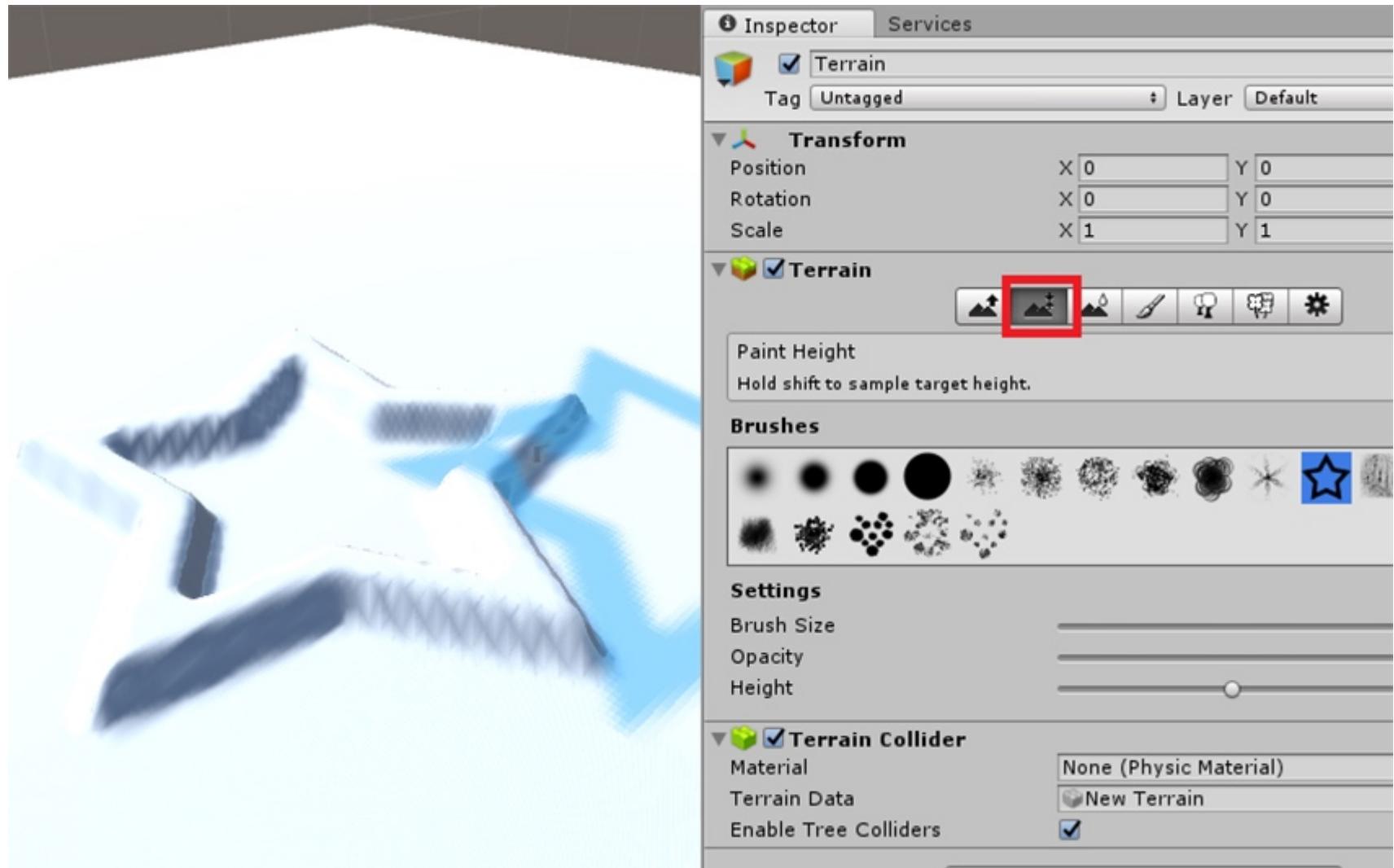


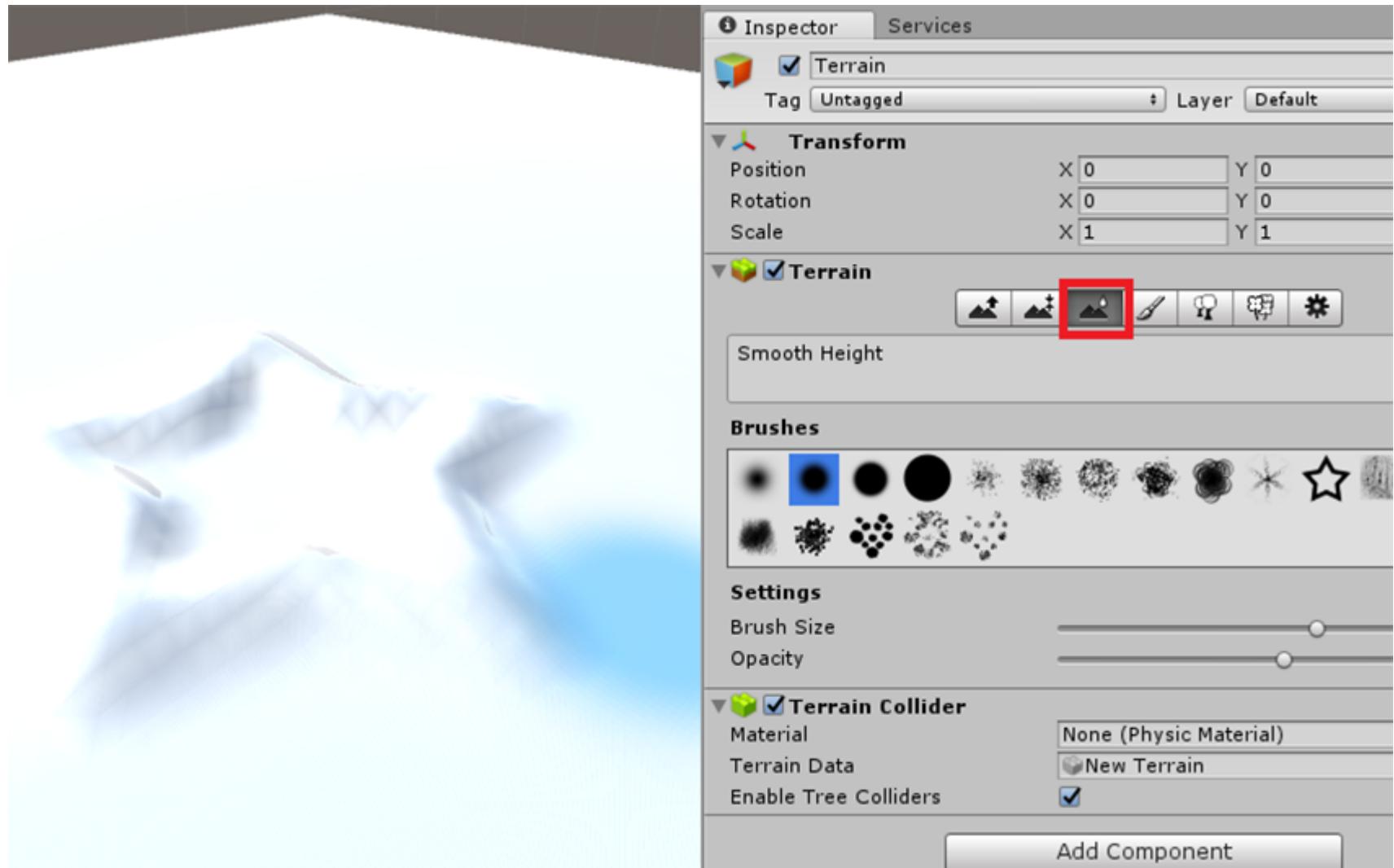
# Редактор ландшафтів Terrain

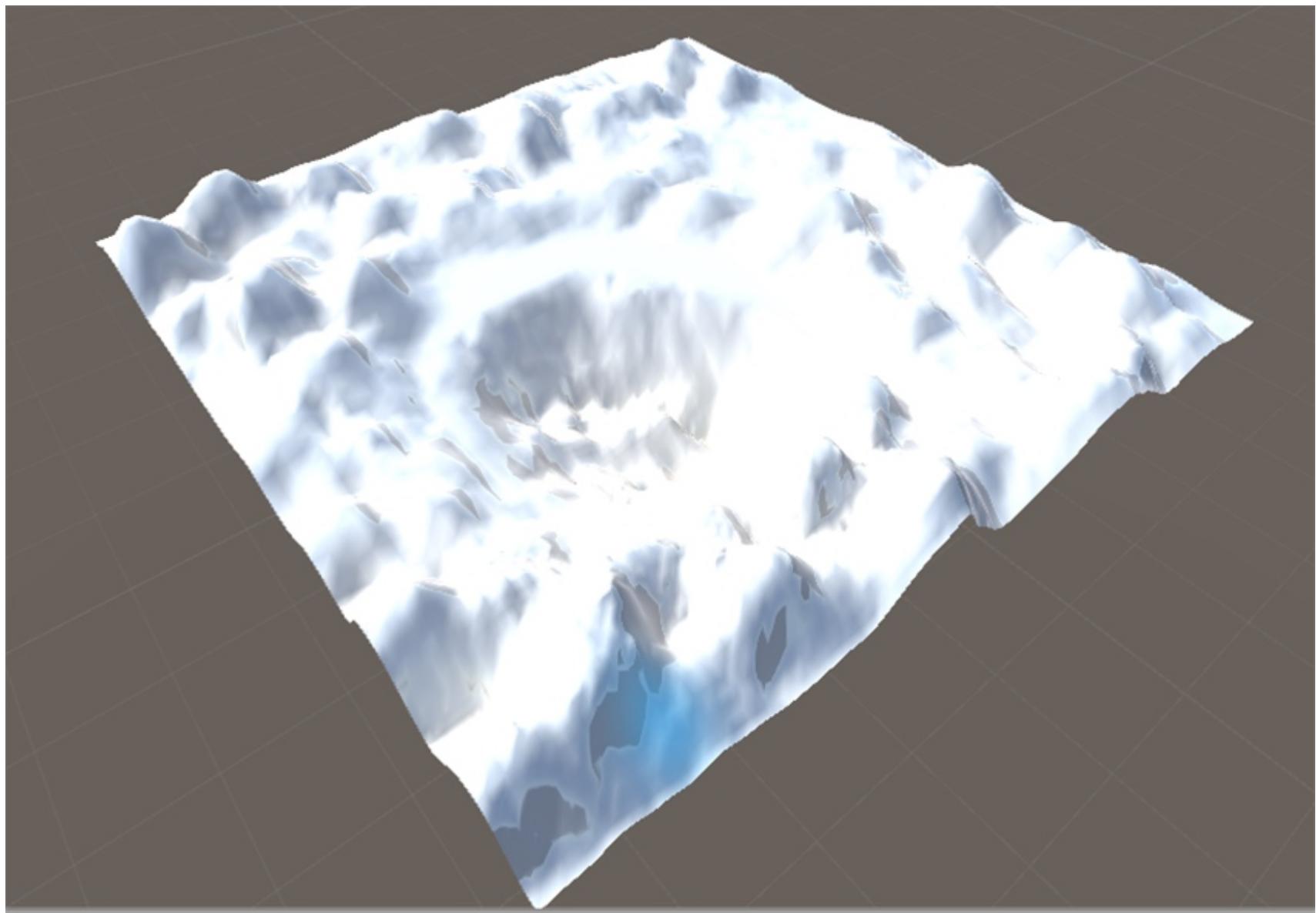


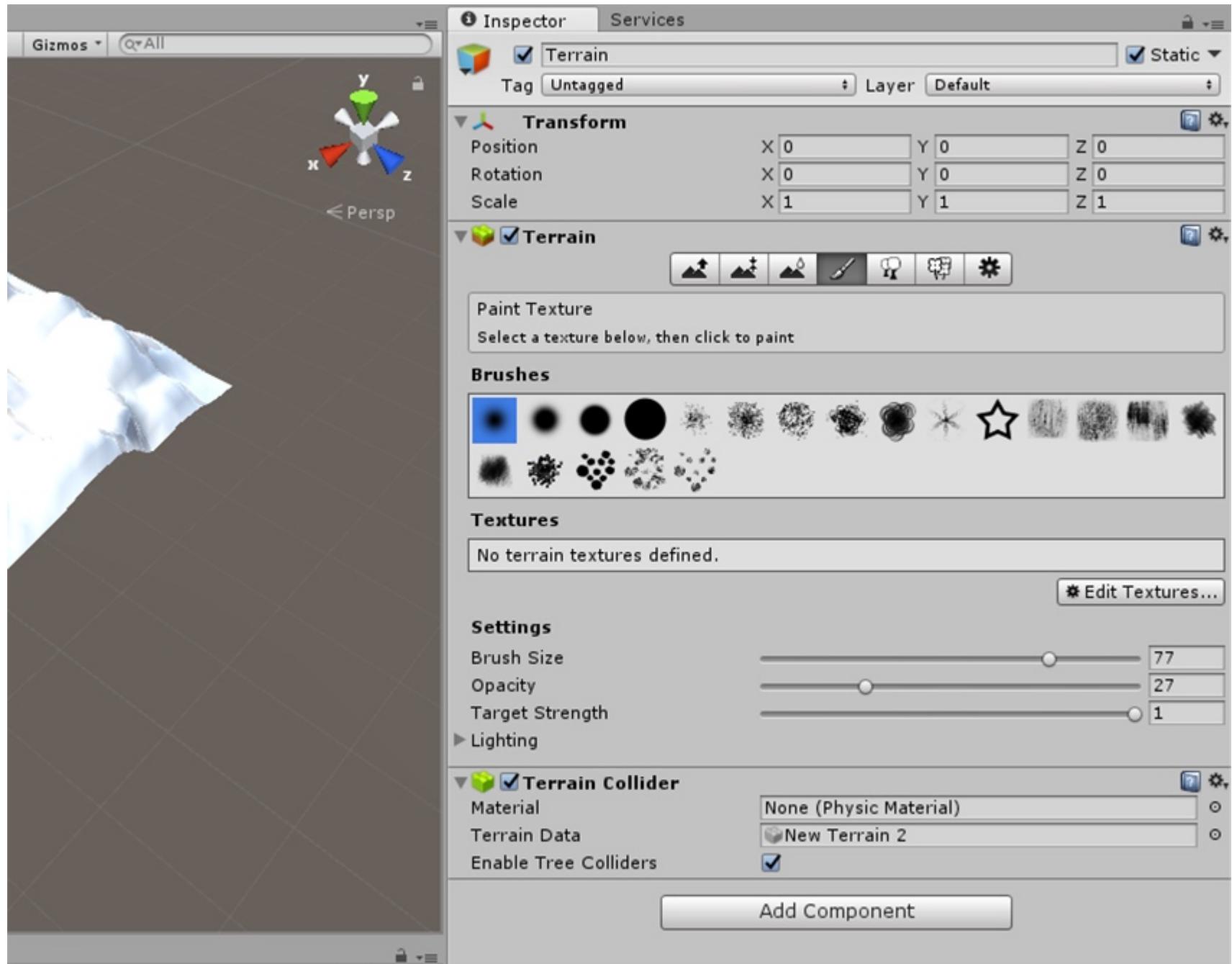


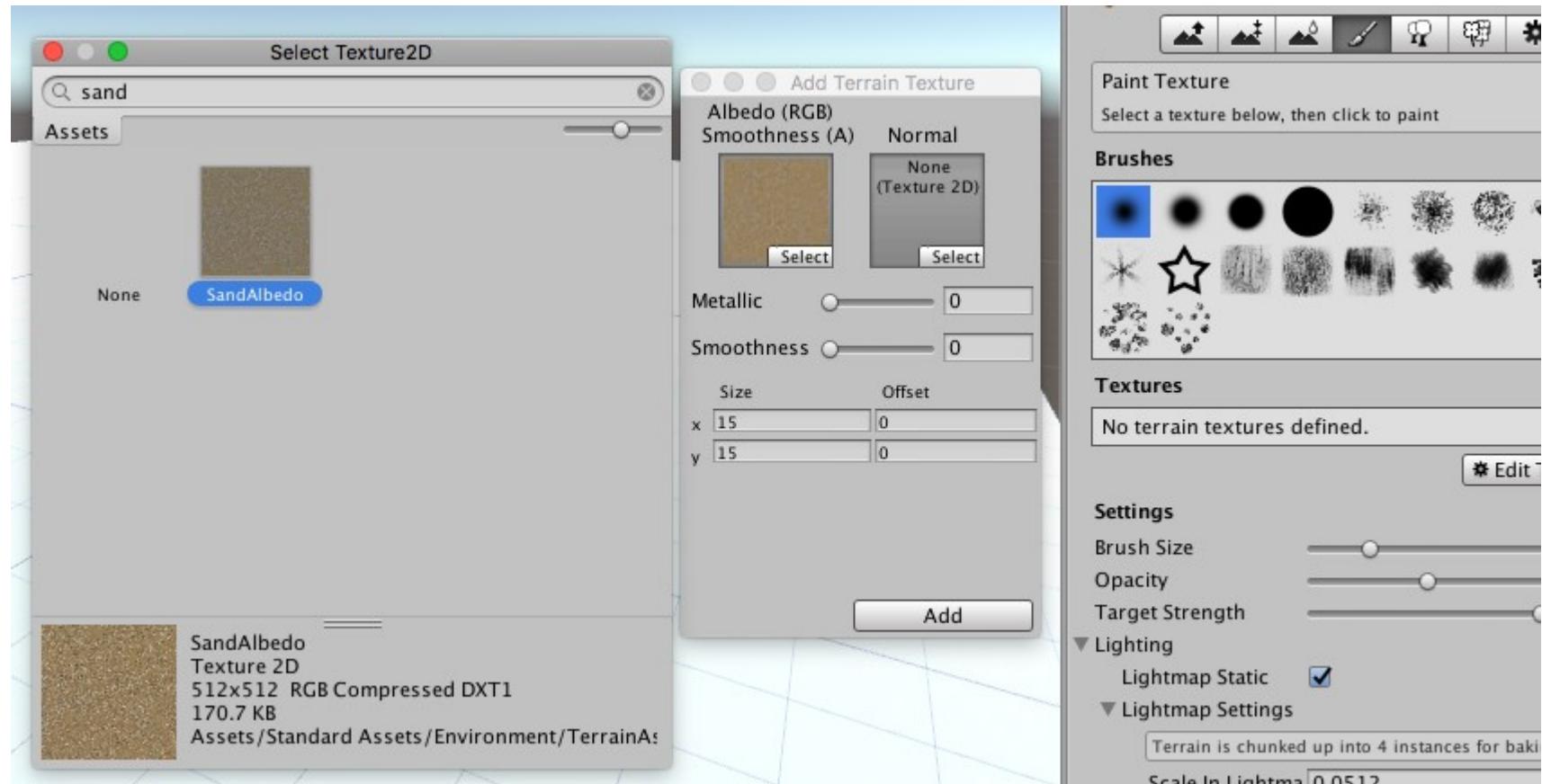


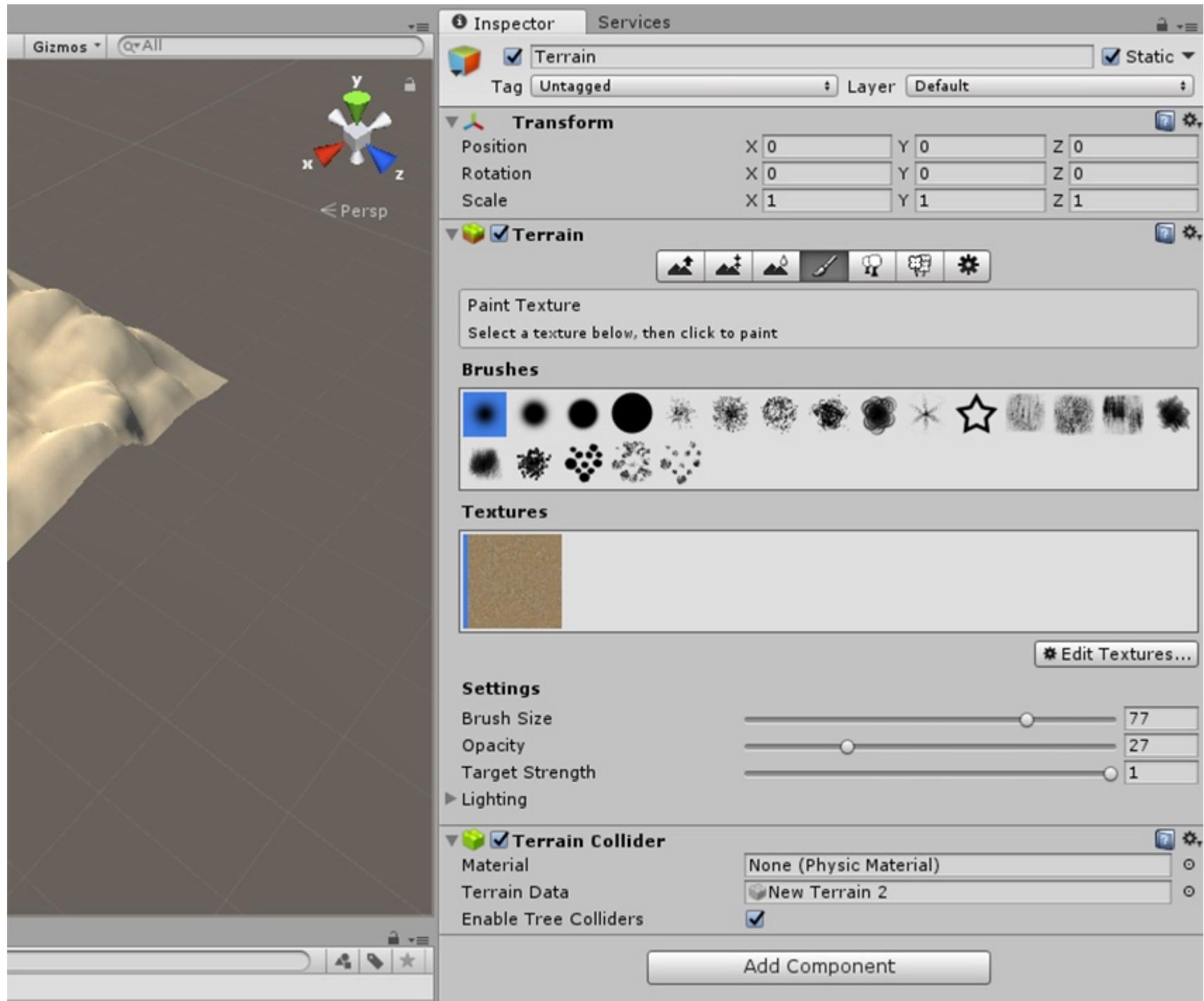












**Gizmos**  **All**

**Inspector** Services

**Terrain**  **Static**

Tag Untagged Layer Default

**Transform**

Position X 0 Y 0 Z 0  
Rotation X 0 Y 0 Z 0  
Scale X 1 Y 1 Z 1

**Terrain**

**Paint Texture**  
Select a texture below, then click to paint

**Brushes**

**Textures**

\* Edit Textures...

**Settings**

Brush Size 77

Opacity 27

Target Strength 1

**Lighting**

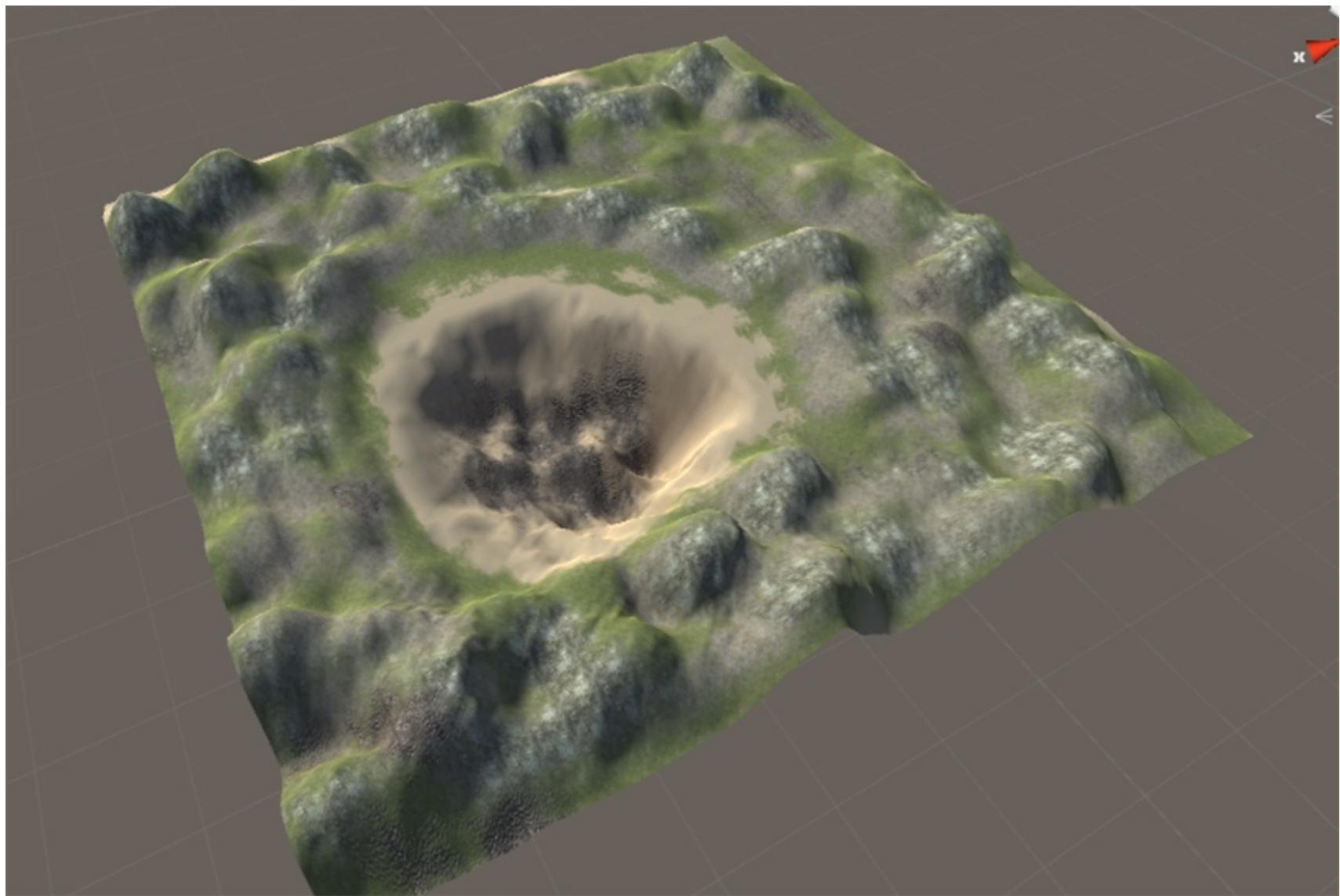
**Terrain Collider**

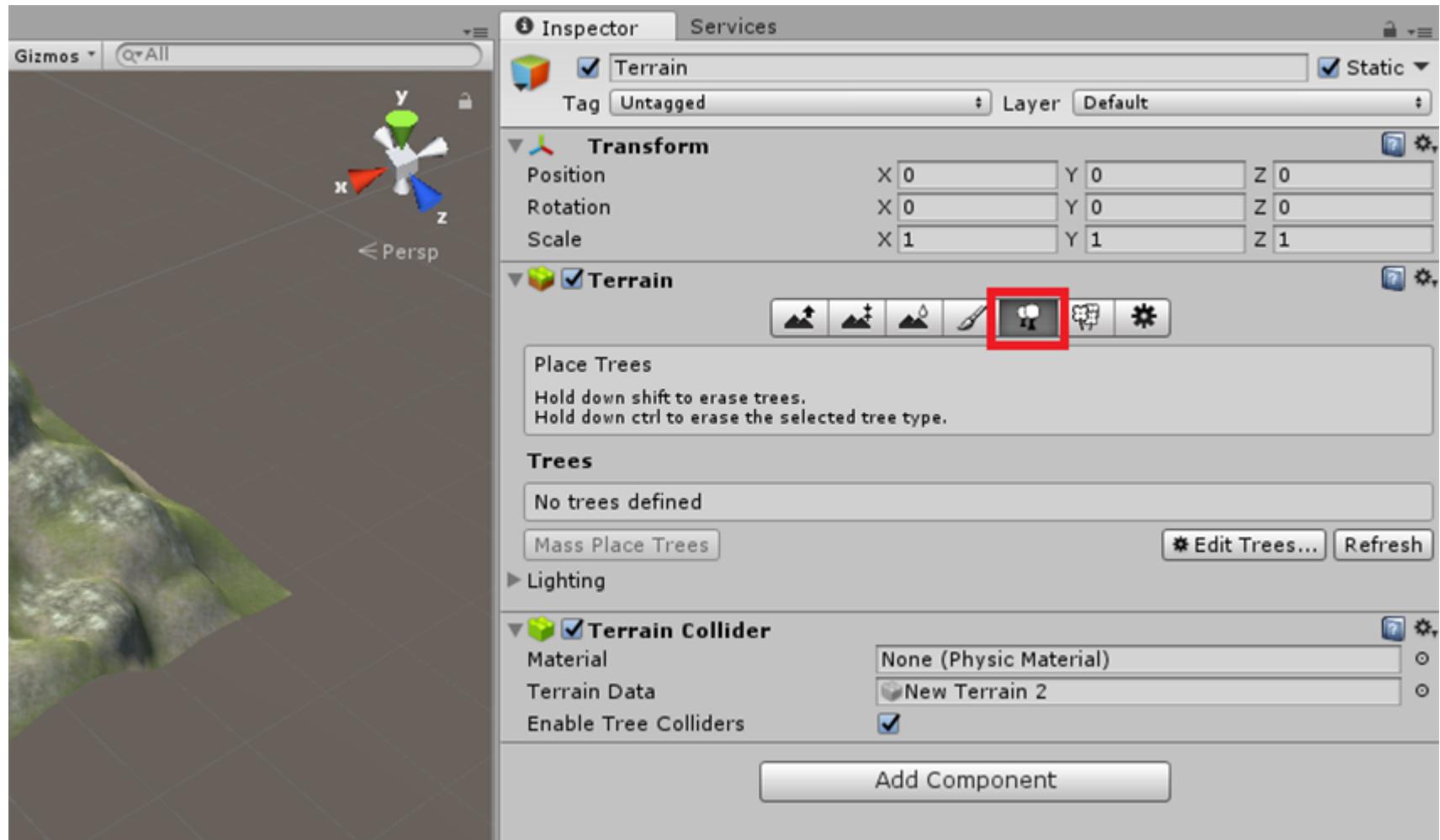
Material None (Physic Material)

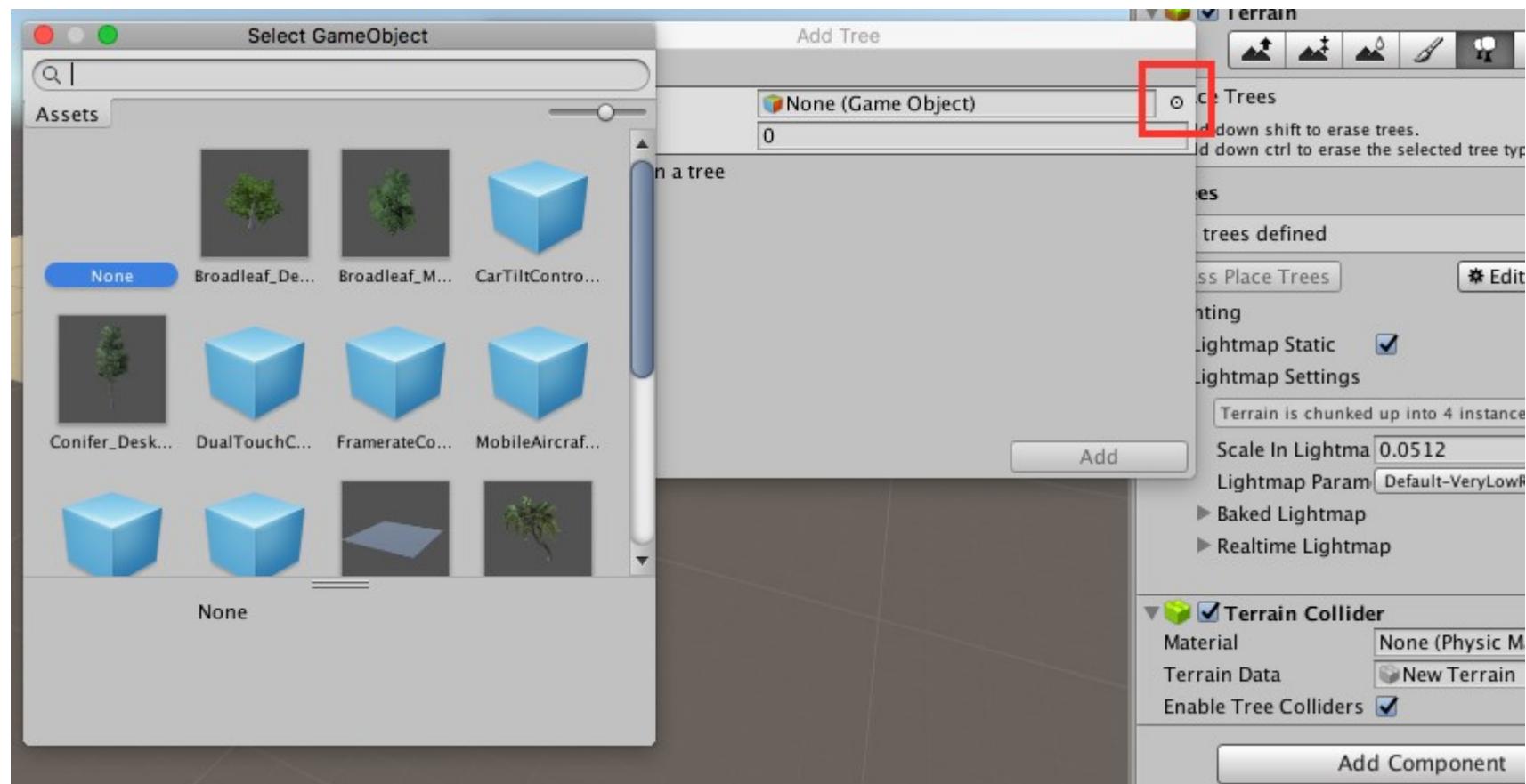
Terrain Data New Terrain 2

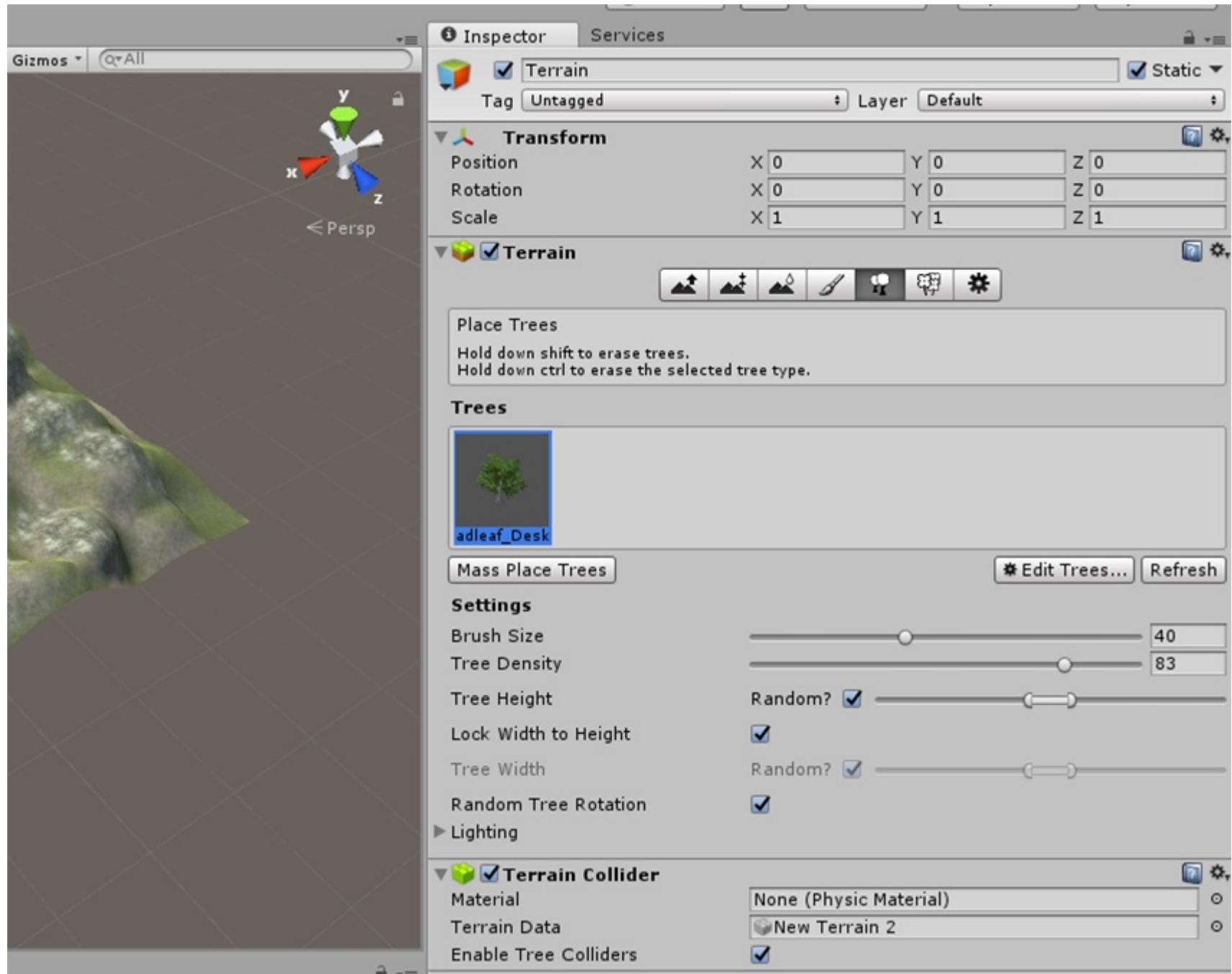
Enable Tree Colliders

Add Component











# Unity 3D



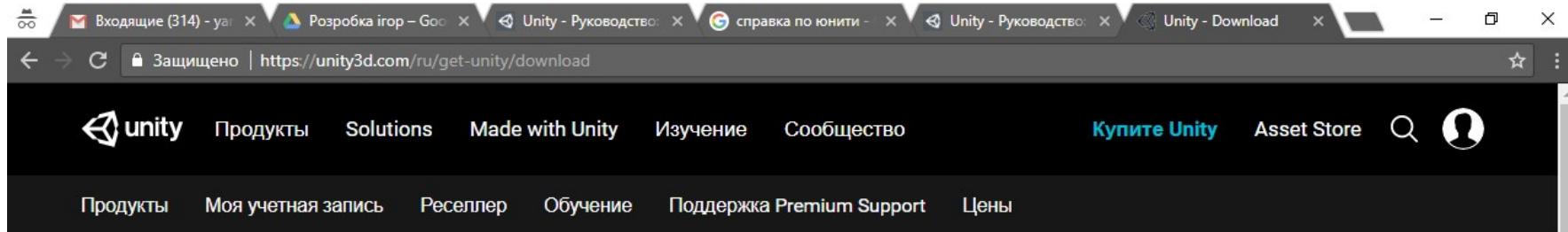
Посилання на скачування:

<https://unity3d.com/ru/get-unity/download>

Посилання на документацію:

<https://docs.unity3d.com/ru/current/Manual/LearningtheInterface.html>

# Сторінка завантаження та опису вимог до системи



The screenshot shows a browser window with multiple tabs open, including 'Входящие (314) - yar', 'Розробка igor - Goo', 'Unity - Руководство', 'справка по юнити - X', 'Unity - Руководство - X', 'Unity - Download - X', and 'Unity - Download'. The main content area is a dark-themed page for downloading Unity. At the top, there's a navigation bar with links like 'unity', 'Продукты', 'Solutions', 'Made with Unity', 'Изучение', 'Сообщество', 'Купите Unity', 'Asset Store', a search icon, and a user profile icon. Below the navigation bar is a secondary menu with links 'Продукты', 'Моя учетная запись', 'Реселлер', 'Обучение', 'Поддержка Premium Support', and 'Цены'. The main heading 'Загрузить Unity' is prominently displayed. A paragraph below it says: 'Добро пожаловать! Вы оказались здесь, поскольку хотите загрузить Unity, самую популярную в мире платформу разработки для создания многоплатформенных 2D- и 3D-игр и интерактивного контента.' A green button labeled 'Выберите Unity + загрузку' is visible. To the right, there's a section titled 'Системные требования' with details about OS and graphics requirements, and a 'Подробнее' button. Another section titled 'Загрузите бета-версию Unity' with a 'Загрузите бета-версию' button is also present.

## Загрузить Unity

Добро пожаловать! Вы оказались здесь, поскольку хотите загрузить Unity, самую популярную в мире платформу разработки для создания многоплатформенных 2D- и 3D-игр и интерактивного контента.

Перед загрузкой выберите подходящую для себя версию Unity.

[Выберите Unity + загрузку](#)

## Загрузите бета-версию Unity

Получите доступ к нашей новейшей функциональности раньше всех. Ваши отзывы помогут нам повысить ее качество.

[Загрузите бета-версию](#)

## Системные требования

**OS:** Windows 7 SP1+, 8, 10, только 64-разрядные версии; Mac OS X 10.9+.

### Графический процессор:

графическая карта с поддержкой DX9 (шейдерная модель 2.0).

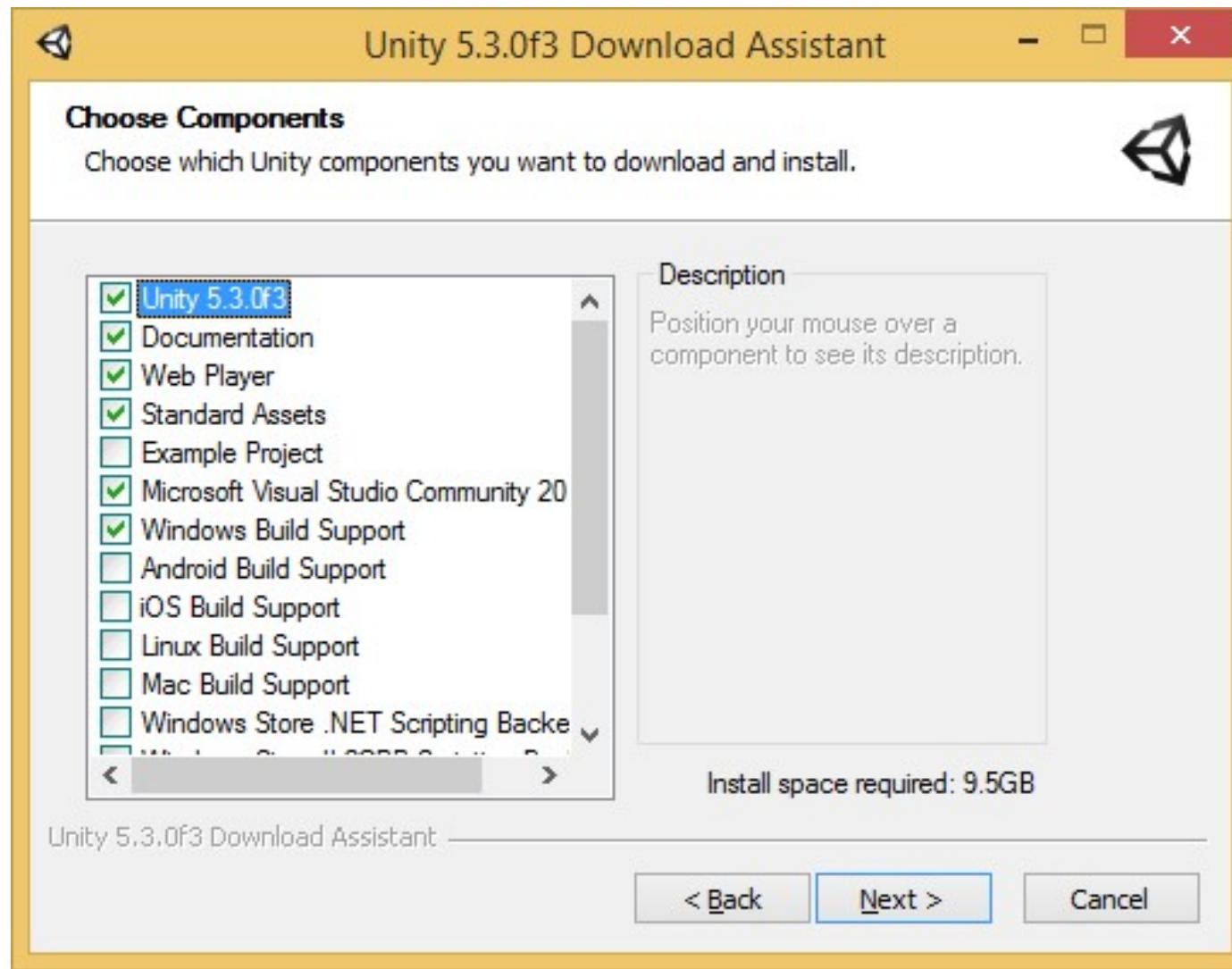
Должны работать любые карты, выпущенные с 2004 года.

[Подробнее](#)

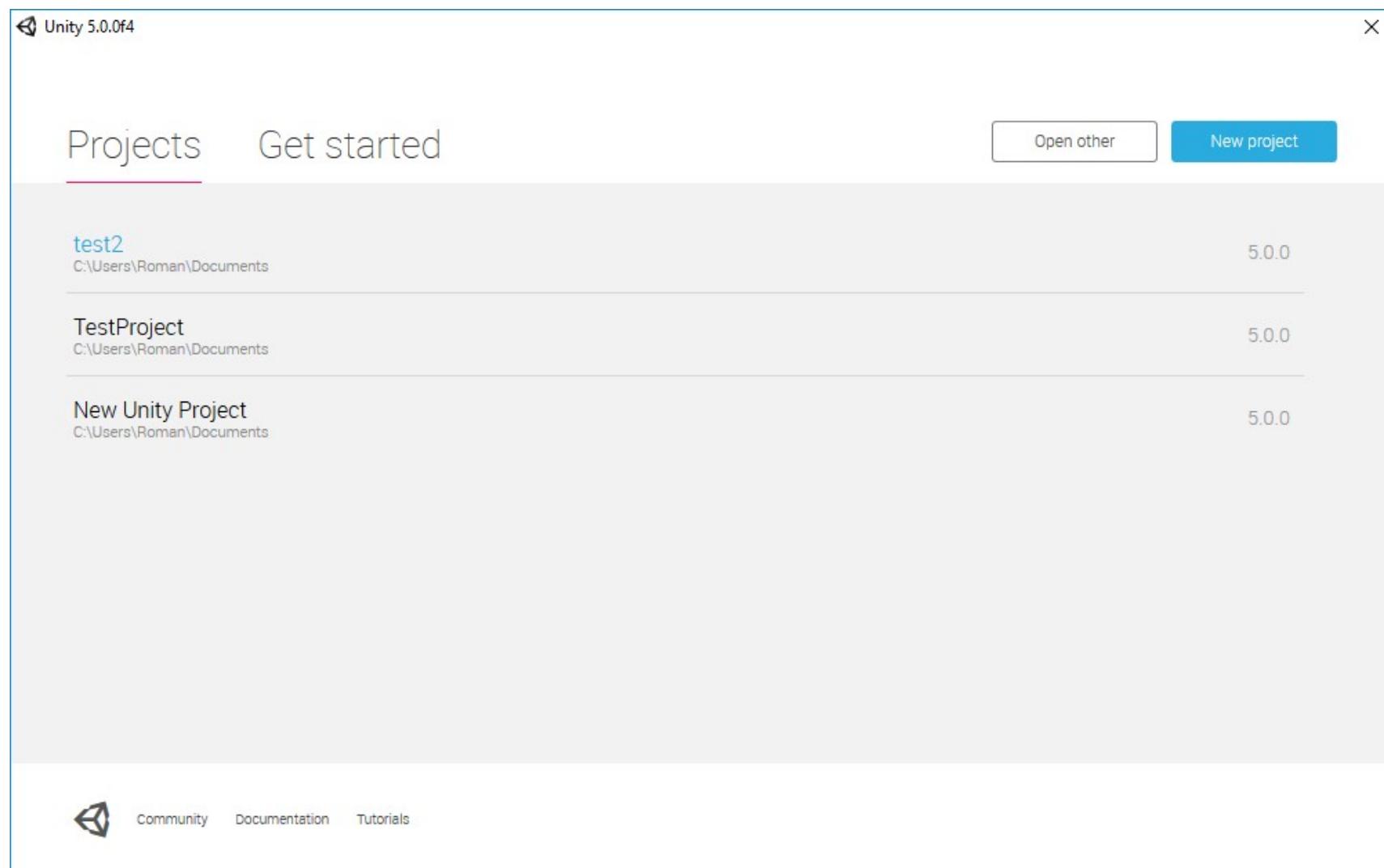
## Ресурсы

- [Более старые версии Unity](#)
- [Руководство по обновлению Unity 2017.3](#)
- [Выпуски патчей](#)

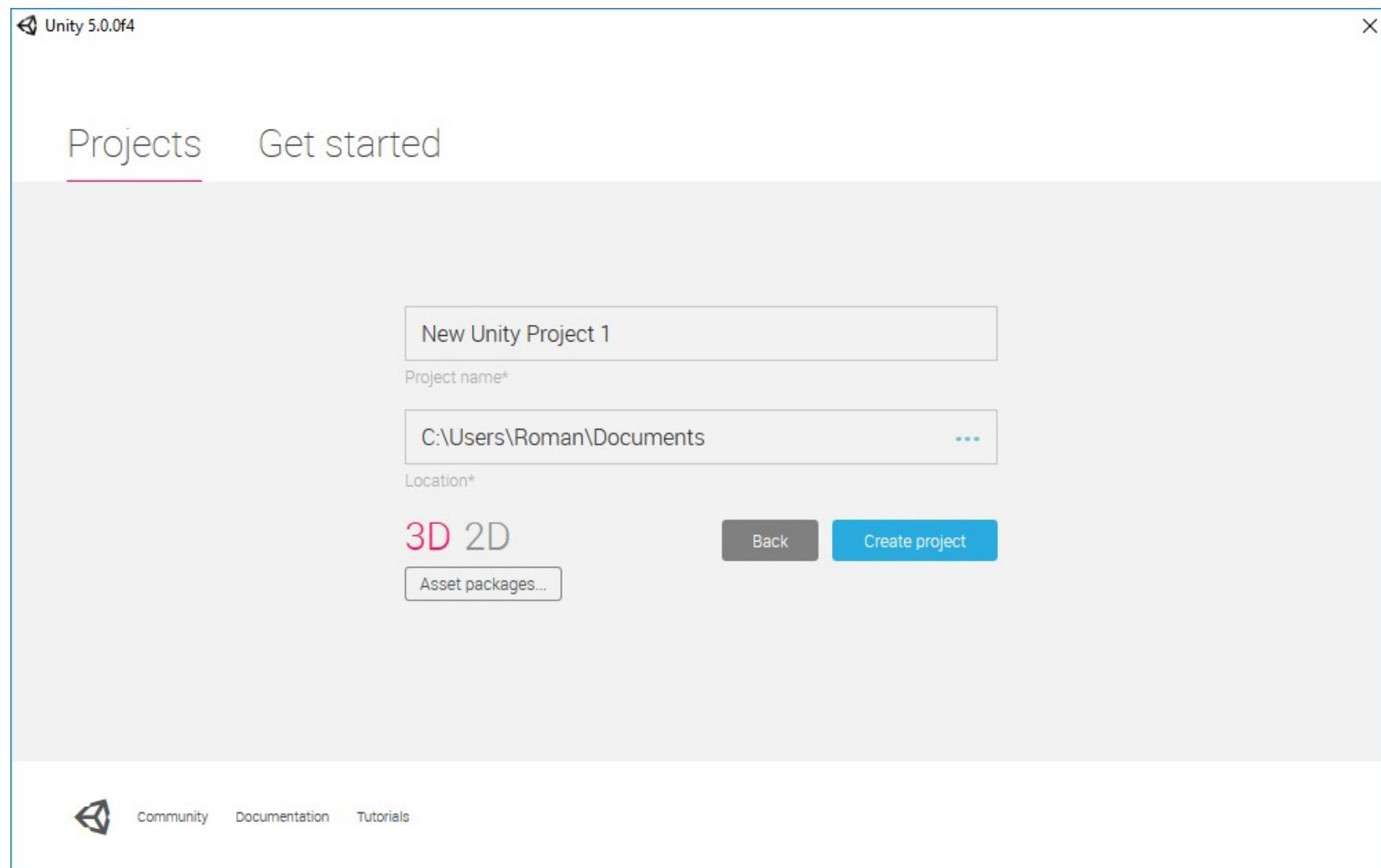
# Вікно встановлення за допомогою асистента



# Стартове вікно



# Вікно створення проекту



## Підтримуємі платформи



iOS

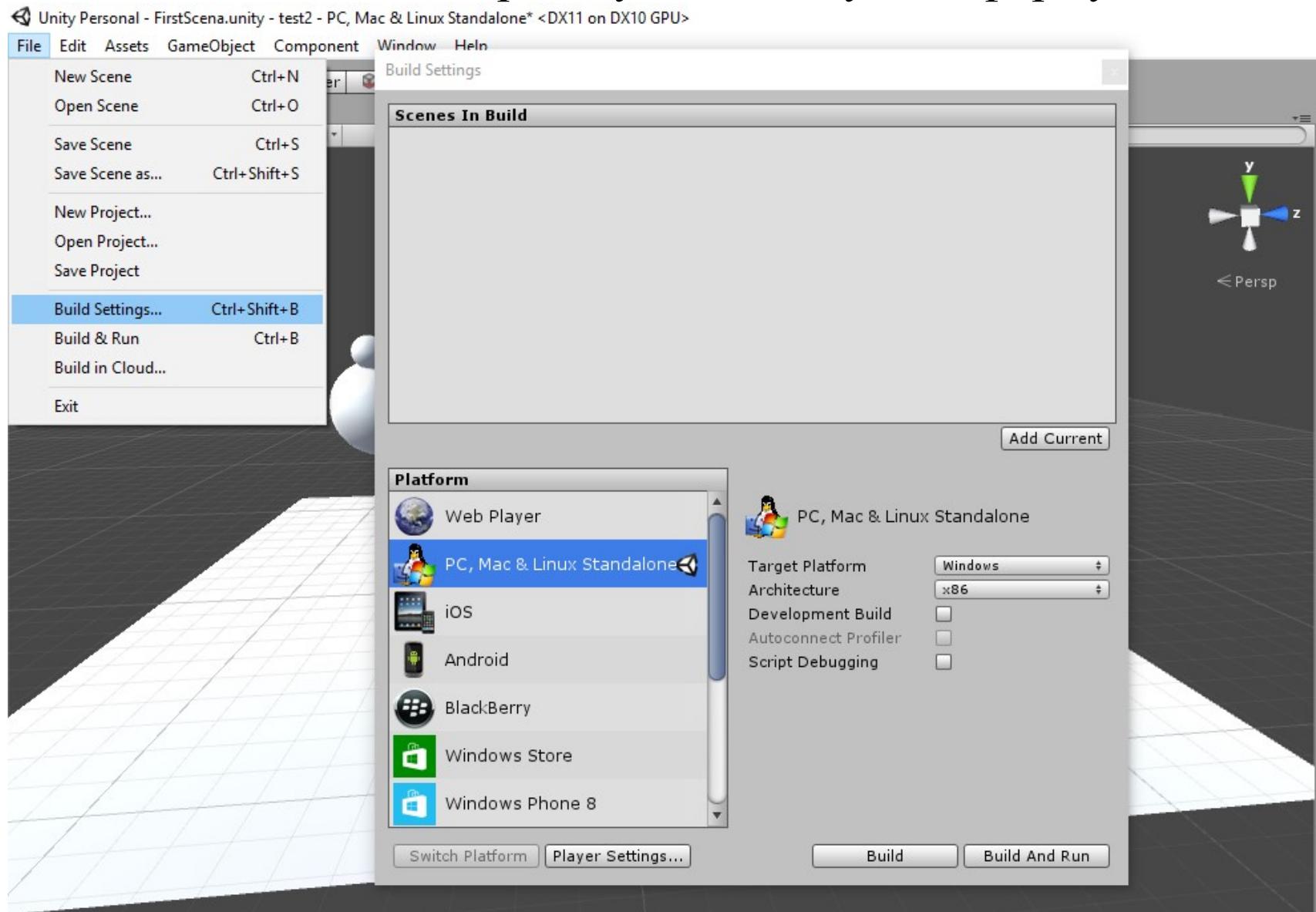


PS3

Wii U™



# Компіляція проекту під задану платформу



# Підтримуємі мови програмування

JavaScript

```
var explosion : Transform;

function OnCollisionEnter() {
    Destroy(gameObject);
    Instantiate(explosion, transform.position, transform.rotation);
}
```

C#

```
using UnityEngine;
using System.Collections;

public class Example : MonoBehaviour {
    public Transform explosion;

    void OnCollisionEnter() {
        Destroy(gameObject);
        Instantiate(explosion, transform.position, transform.rotation);
    }
}
```



Boo

```
import UnityEngine
import System.Collections

class Example(MonoBehaviour):

    public explosion as Transform

    def OnCollisionEnter():
        Destroy(gameObject)
        Instantiate(explosion, transform.position, transform.rotation)
```

## Основні складові проекту

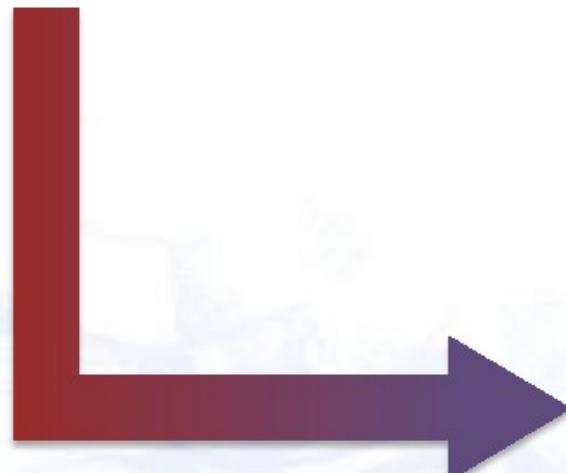
Game

Assets

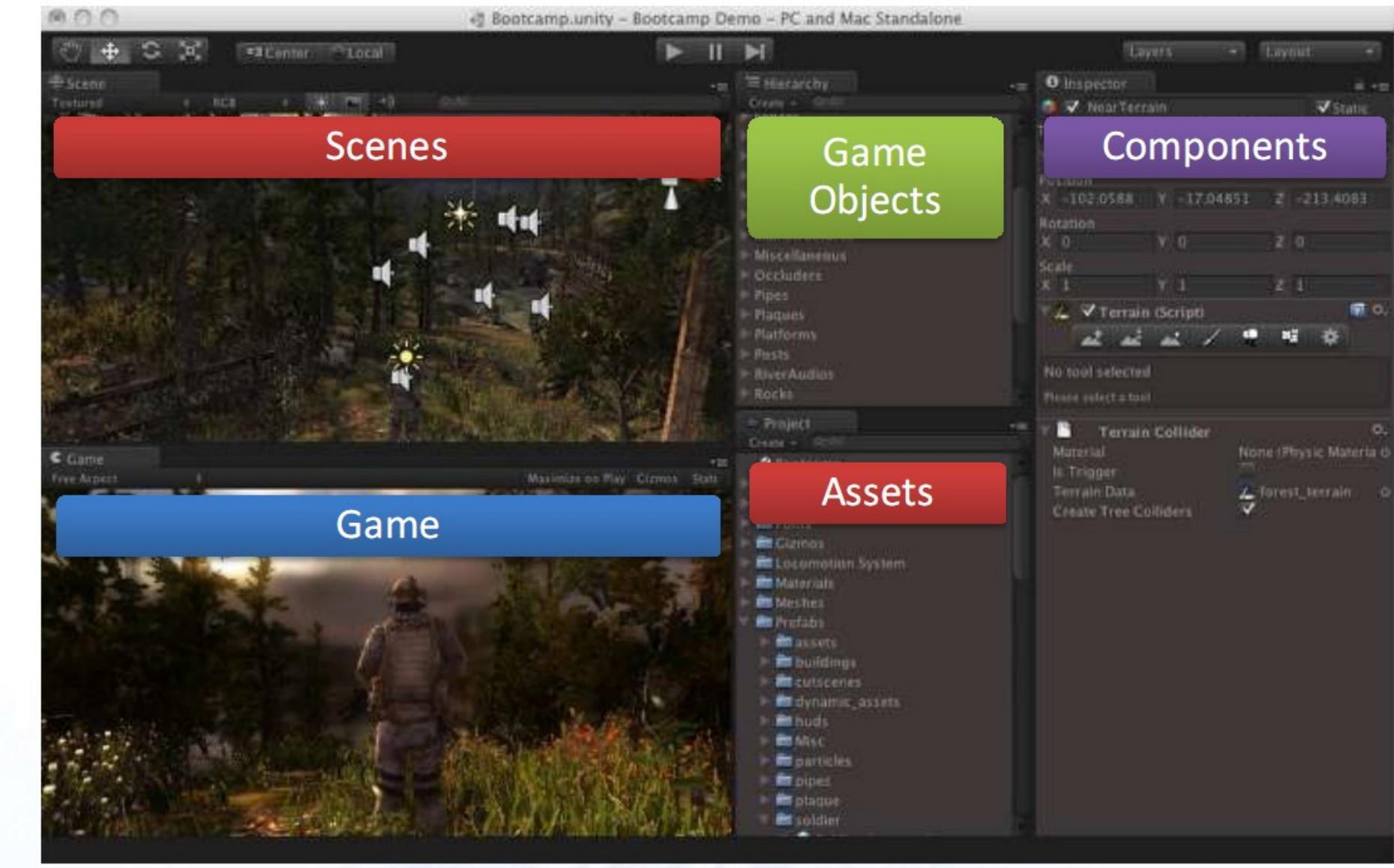
Scenes

Game Objects

Components



# Робочі області

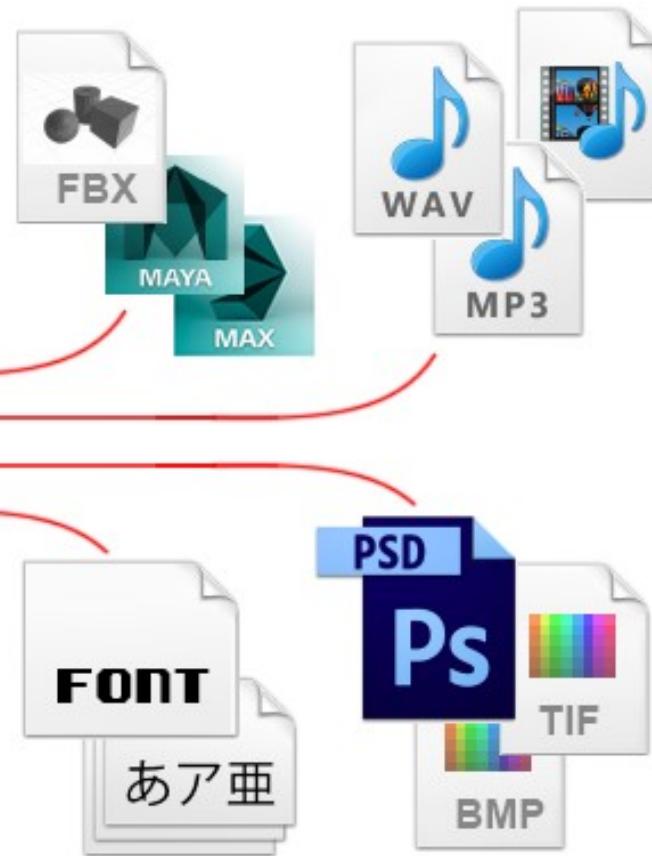
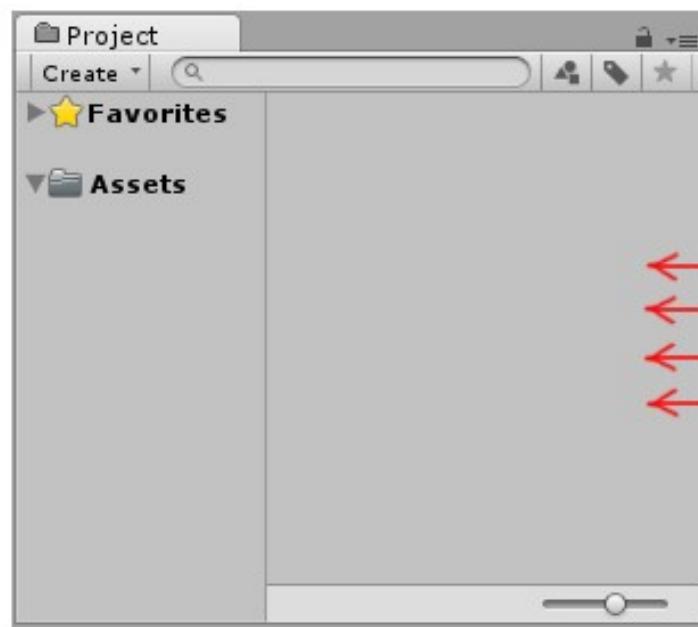


# Assets

```
public class SphereScript : MonoBehaviour
{
    public Color clickColor = Color.green;

    private void OnMouseDown()
    {
        this.renderer.material.color = clickColor;
    }
}
```





# Asset store

Unity Personal - Untitled - 123 - PC, Mac & Linux Standalone <DX11 on DX10 GPU>

File Edit Assets GameObject Component Window Help

Scene Game Shaded

Asset Store

MAXIMUM PRICE

FREE 5 10 20  
FREE ONLY PAID ONLY

MINIMUM RATING

★★★★★

PACKAGES ONLY LISTS

3D Radar Builder ...  
3D Models/Characters  
DaiMangou  
★★★★★ (17)  
\$60.00

Eye Advanced  
3D Models/Characters  
Tanuki Digital  
★★★★★ (131)  
\$35.00

MOBA Elf Characte...  
3D Models/Characters  
Pior Oberson  
★★★★★ (15)  
\$19.00

Easter Edition add-...  
3D Models/Characters  
ASSETCREW  
★★★★★ (16)  
FREE

SORT BY RELEVANCE / POPULARITY / NAME / PRICE / RATING / UPDATED

1 2 3 4 5 6 7 8 9 10 Next Last 1 - 36 of 5322

Animal pack deluxe  
3D Models/Characters  
janpec  
★★★★★ (145)  
\$5.00

Viking Ulf  
3D Models/Characters  
Maksim Bugrimov  
Not enough ratings  
\$1.99

PBR Creatures (Pa...  
3D Models/Characters  
Maksim Bugrimov  
★★★★★ (134)  
\$4.99

Kinect v2 Example...  
3D Models/Characters  
RF Solutions  
★★★★★ (1255)  
\$25.00

Realistic Hands - a...  
3D Models/Characters  
Mixail  
★★★★★ (16)  
\$3.00

Poly Art: Bear  
3D Models/Characters  
MalberS Animations  
Not enough ratings  
\$14.30

Liam | Stylized char...  
3D Models/Characters

Underwater Animal...  
3D Models/Characters

Primitive Plus  
3D Models/Characters

Layers

Gizmos Q All

Language: English Create Account Log In

Scene Game Inspector

Layouts

Scene Ctrl+1  
Game Ctrl+2  
Inspector Ctrl+3  
Hierarchy Ctrl+4  
Project Ctrl+5  
Animation Ctrl+6  
Profiler Ctrl+7  
Audio Mixer Ctrl+8  
Asset Store Ctrl+9  
Version Control Ctrl+0  
Animator Parameter  
Animator  
Sprite Packer  
Lighting  
Occlusion Culling  
Frame Debugger  
Navigation  
Console Ctrl+Shift+C

MAXIMUM SIZE MB

8 5MB 50MB 100MB 250MB 500MB 1GB 4GB

UPDATED days ago

7d 14d 1m 3m 6m 1y 5y

7d 14d 1m 3m 6m 1y 5y

Home

3D Models

Characters

- ▶ Animals
- ▶ Creatures
- ▶ Humanoids
- ▶ Robots
- ▶ Toons
- ▶ UMA Avatars
- ▶ Other

Environments

Props

Vegetation

Vehicles

Other

Animation

Audio

Complete Projects

Editor Extensions

Particle Systems

Scripting

Services

Shaders

Textures & Materials

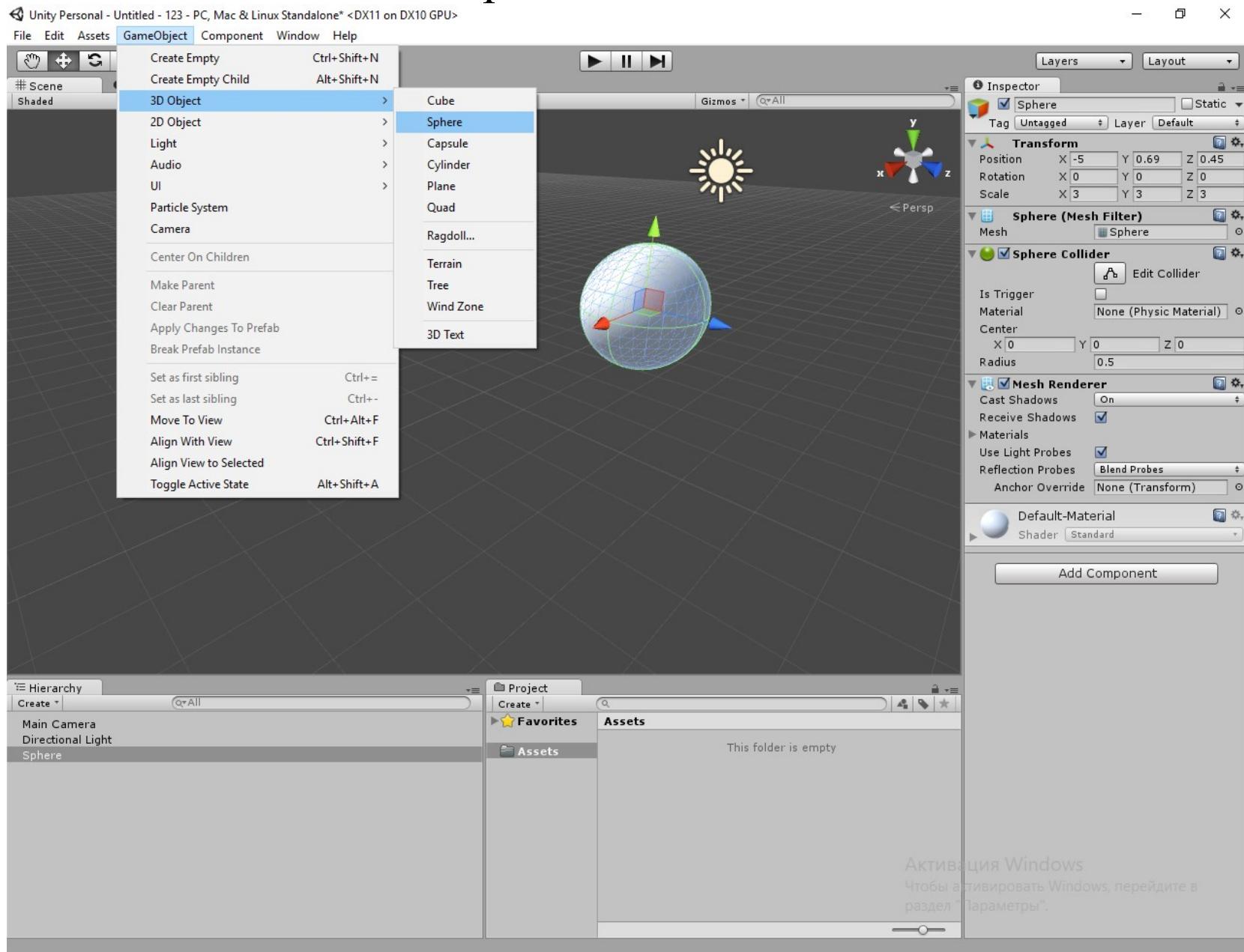
Unity Essentials

NEW

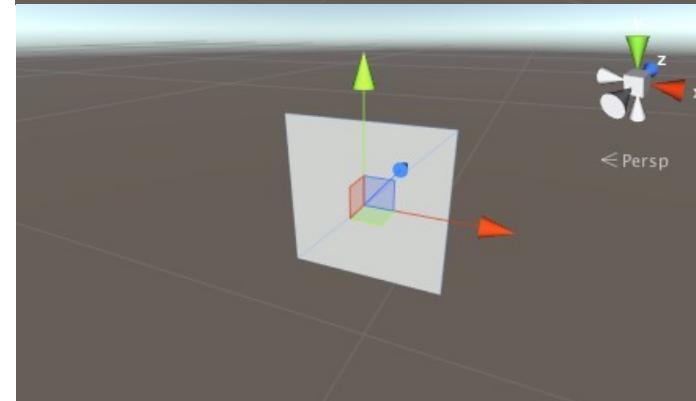
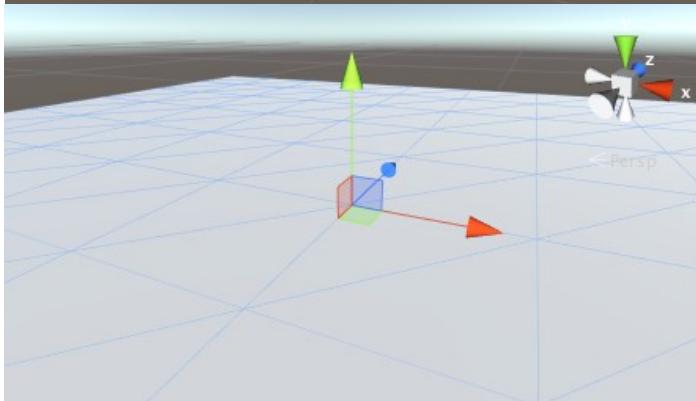
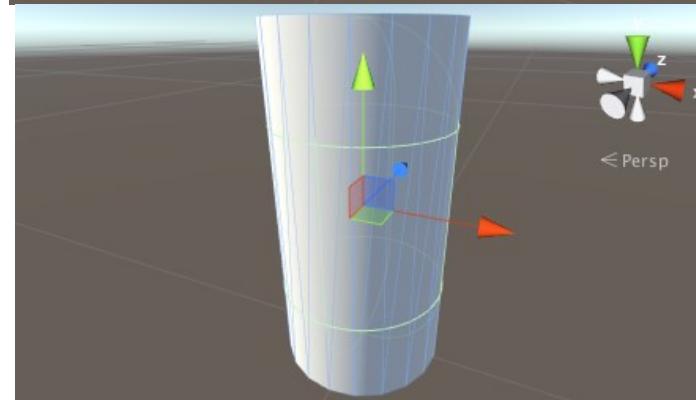
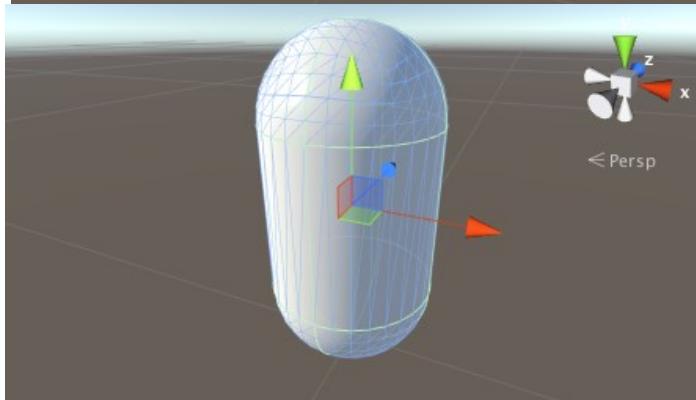
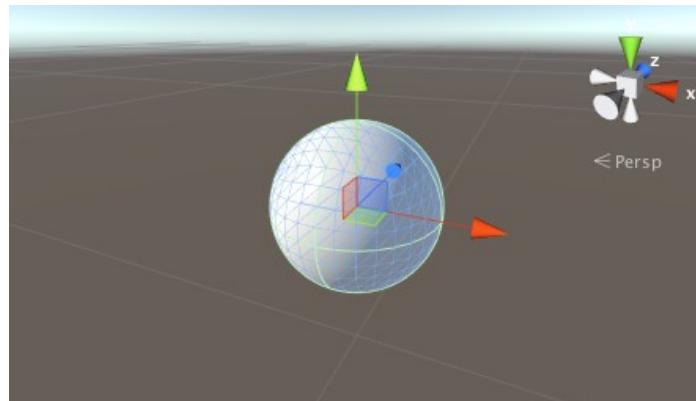
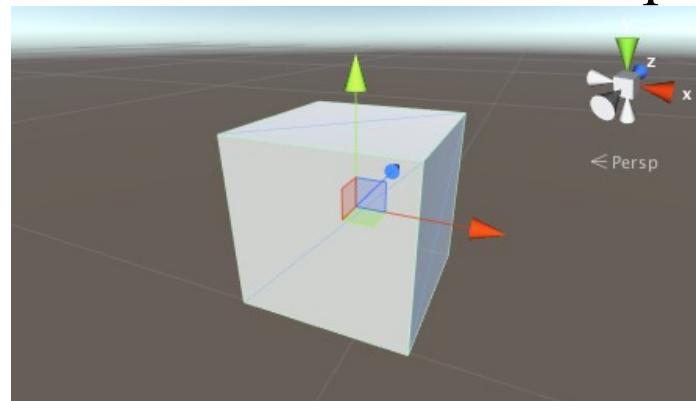
Top Paid

Windows

# Створення компонентів



# Примітиви



Входящие (314) X Розработка iop... X Unity - Руководство X Google справка по юни... X Unity - Руководство X Unity - Download X Google Перевод... X

Защищено | https://docs.unity3d.com/ru/current/Manual/Toolbar.html

unity | DOCUMENTATION Руководство API скриптов Поиск по справке... unity3d.com Язык: Русский ▾

Version: 5.3 (switch to 5.4b)

Использование окна Scene View

- Игровое окно
- Иерархия
- + Using the Inspector  
(Использование панели Inspector)
- Панель инструментов**
- Поиск
- Другие окна
- Customizing Your Workspace  
(Настройка рабочего пространства)
- Сочетания клавиш Unity
- + Создание геймплея.
- + Editor Features
- + Advanced Development
- + Advanced Editor Topics
- + Licenses and Activation
- + Upgrade Guides
- + Unity 2D
- + Графика
- + Физика
- + Скриптинг
- + Multiplayer and Networking
- + Аудио
- + Анимация
- + Пользовательский интерфейс
- + Навигация и поиск пути
- + Unity Services
- + Virtual Reality
- + Open-source repositories
- + Специфичные платформы

Руководство Unity / Working In Unity / The Main Windows / Панель инструментов

## Панель инструментов

Данная панель содержит пять основных элементов. Каждый из них относится к разным частям редактора.

**Transform Tools – used with the Scene View**

**Transform Gizmo Toggles (переключатели гизмо)** – влияют на содержимое окна Scene View

**Play/Pause/Step Buttons – used with the Game View**

**Кнопки Play/Pause/Step** – используются в окне Game View

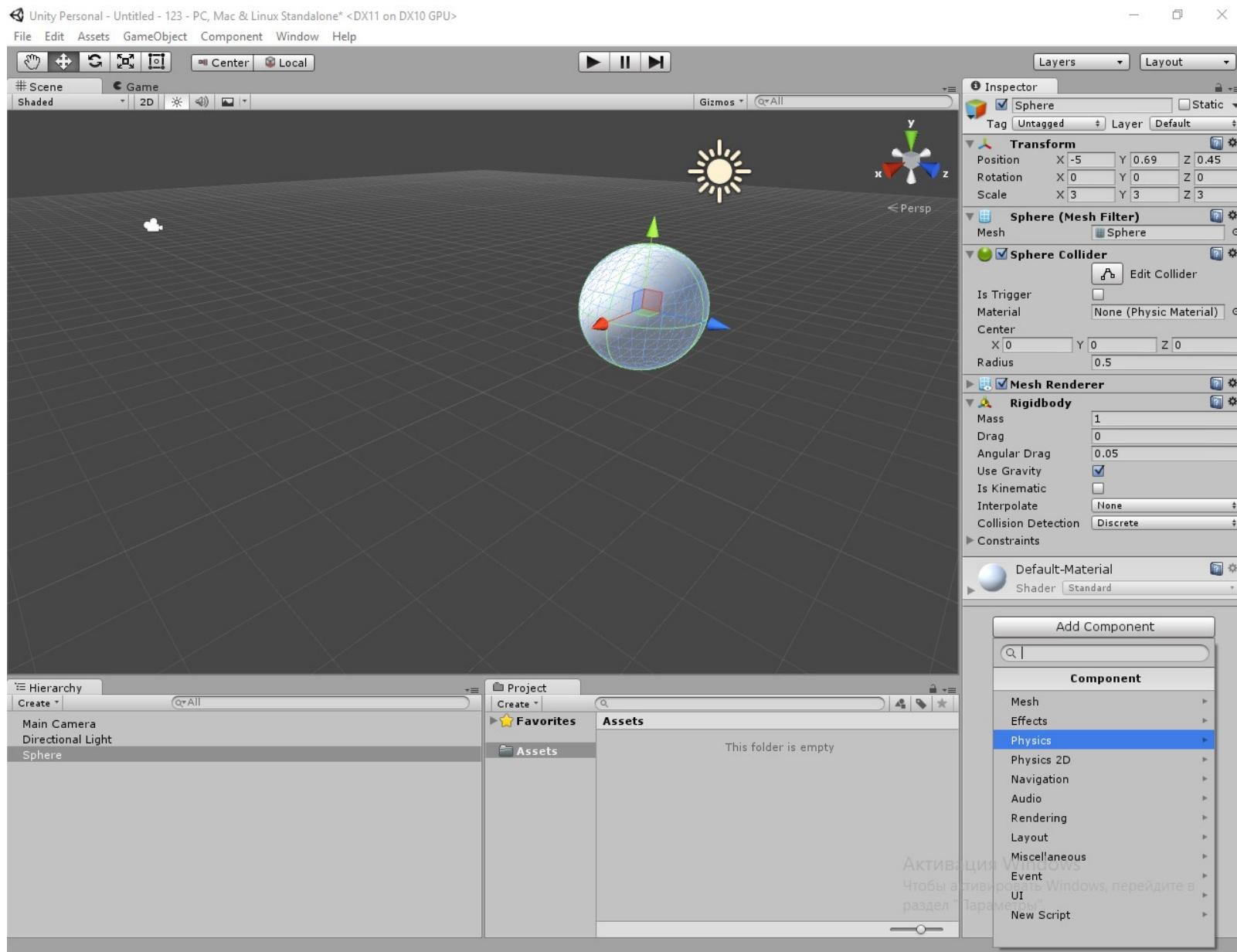
**Layout** – Выпадающее меню Layout – контролирует расположение всех окон

**Layers** – Выпадающее меню [Layers] – отвечает за то, какие объекты должны быть отображены в окне Scene View

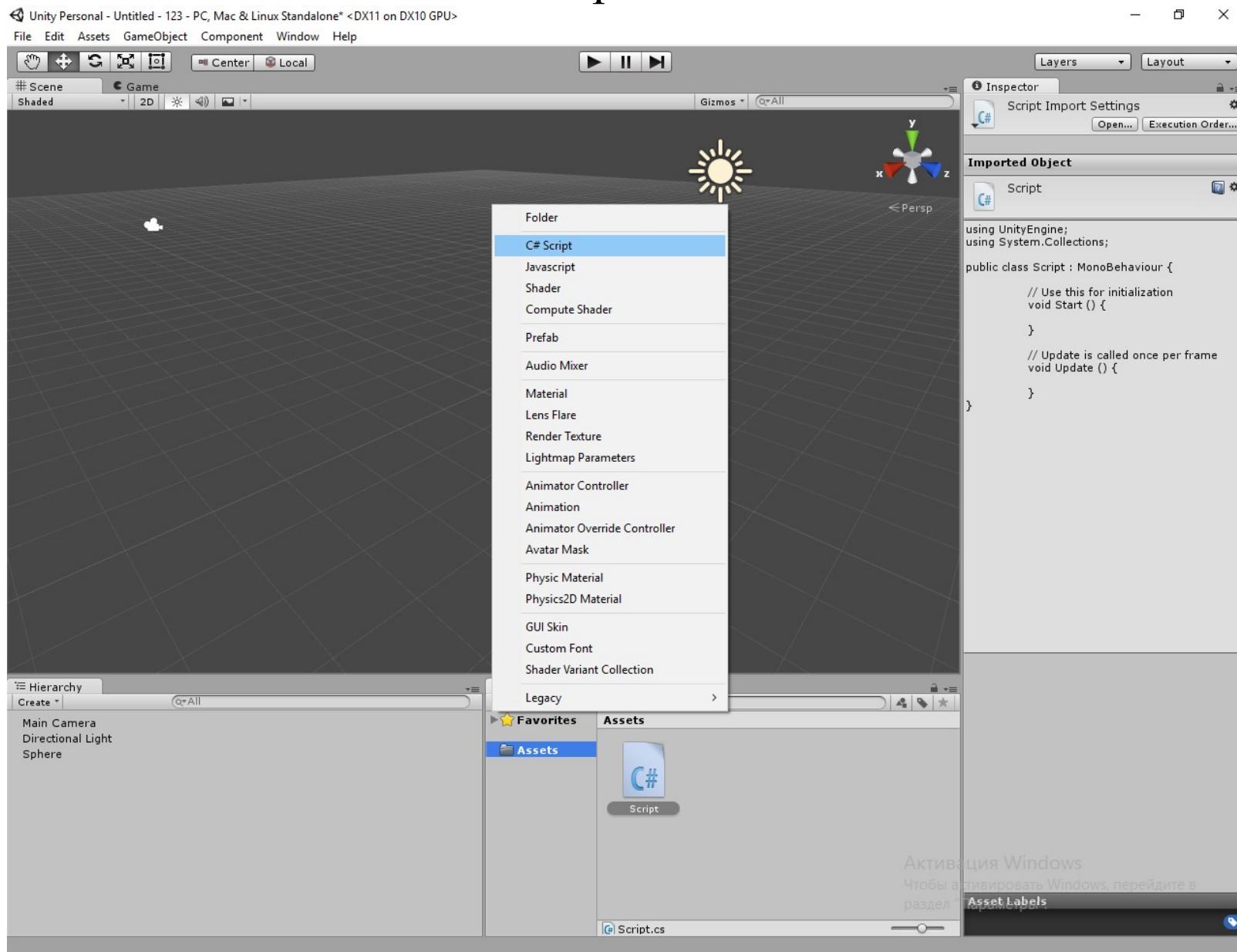
**Layout** – Выпадающее меню Layout – контролирует расположение всех окон

Активация Windows  
Чтобы активировать Windows, перейдите в раздел "Параметры".

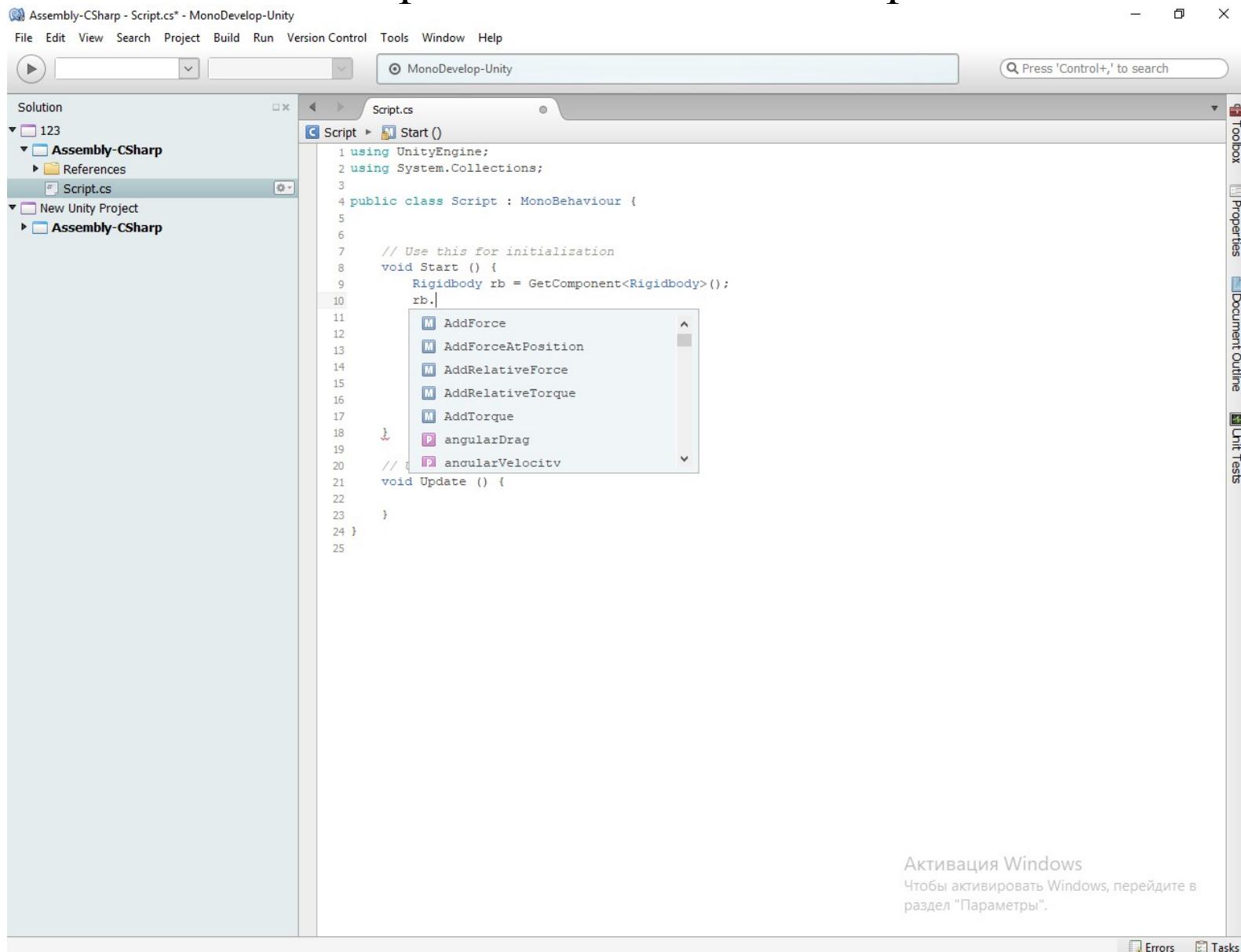
# Компоненты



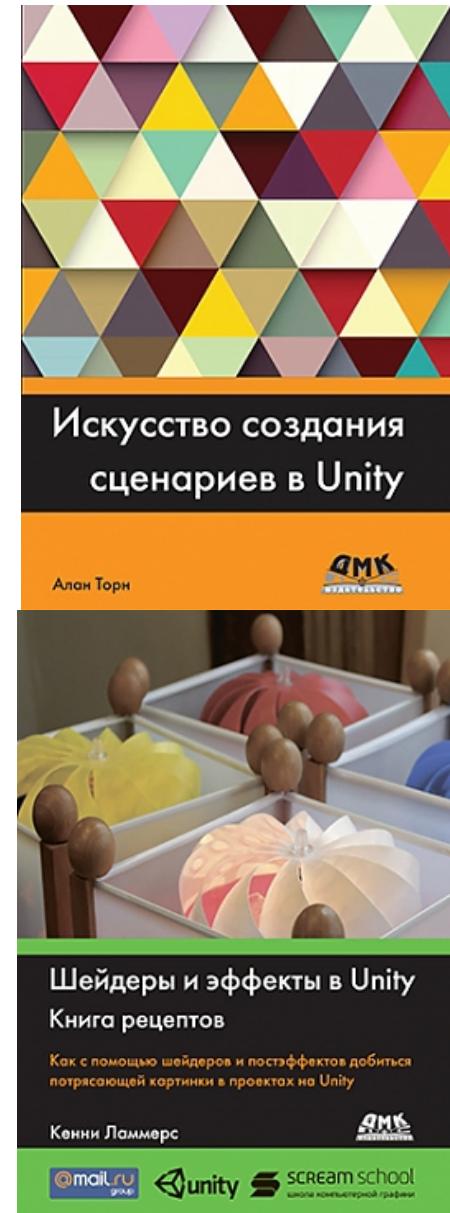
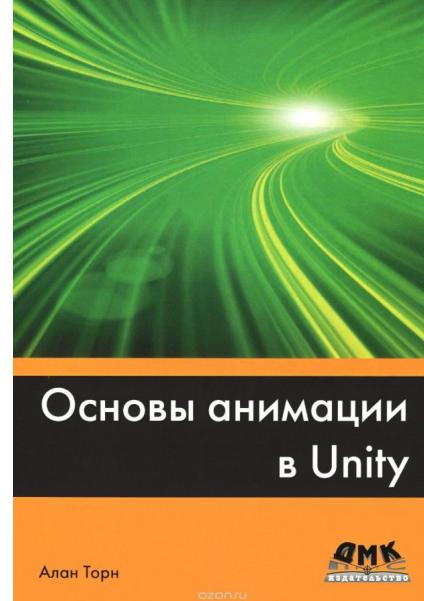
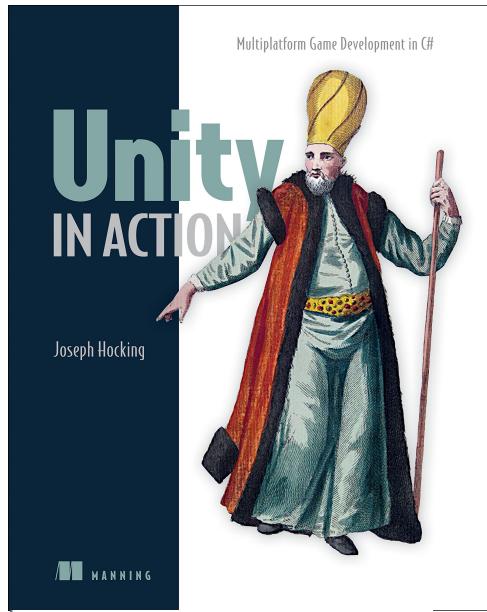
# Скрипти



# Середовище MonoDevelop



# Література

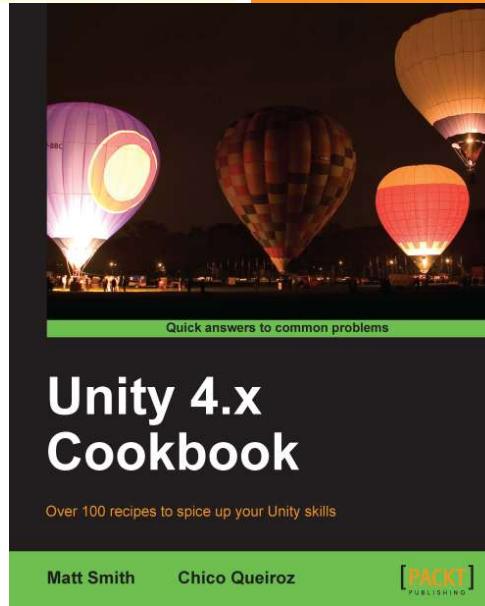
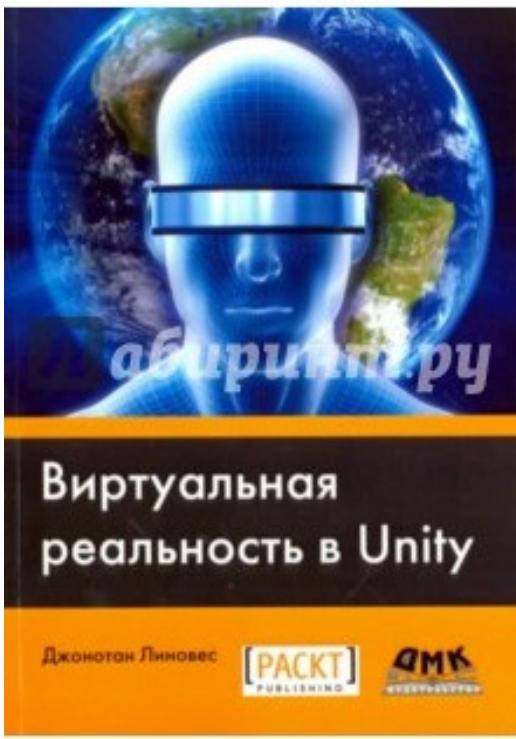


Шейдеры и эффекты в Unity  
Книга рецептов

Как с помощью шейдеров и постэффектов добиться потрясающей картинки в проектах на Unity

Кенни Ламмерс

@mail.ru group unity SCREAM school школа компьютерной графики



**ДЯКУЮ ЗА УВАГУ!**