

Leap Motion

Tvorba prvej hry (Bowling)

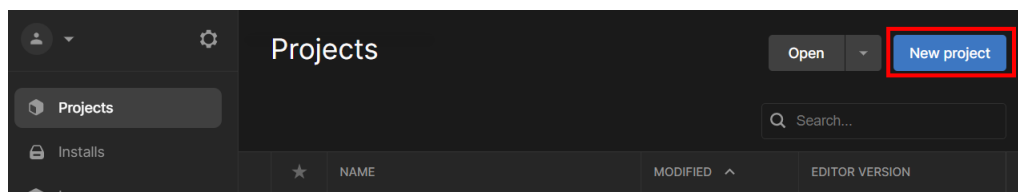
Leap Motion podporuje tvorbu hier napríklad v hernom frameworku Pygame či v hernom engine Unity/Unreal pomocou pluginu. Pri tvorbe tejto hry použijeme Unity.

Čo budeme potrebovať:

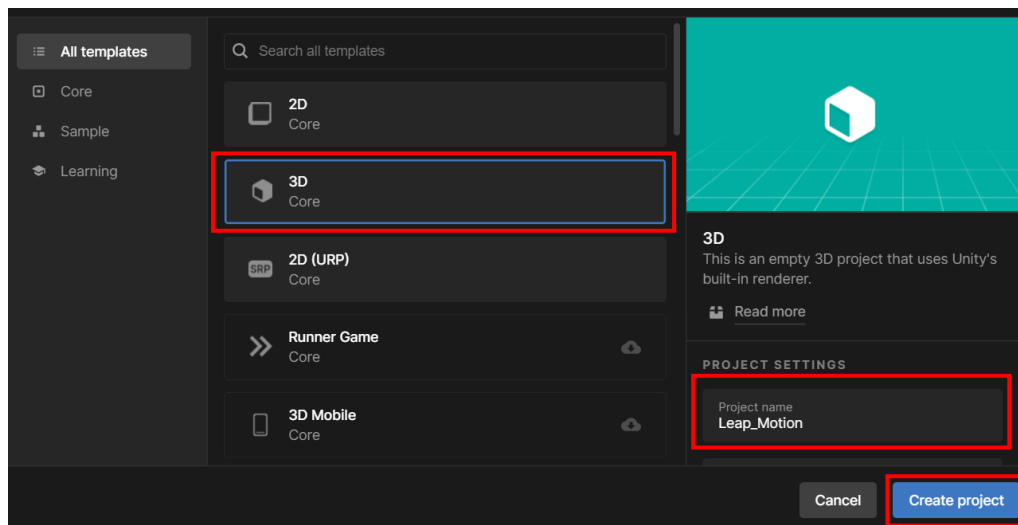
- Unity 2019.4 alebo vyšší ([odkaz na stiahnutie](#))
- Windows 10, 8.1 alebo 7
- Leap Motion software ([odkaz na stiahnutie](#))

Príprava Unity

Po nainštalovaní Unity si v **Unity Hub** vytvoríme nový projekt

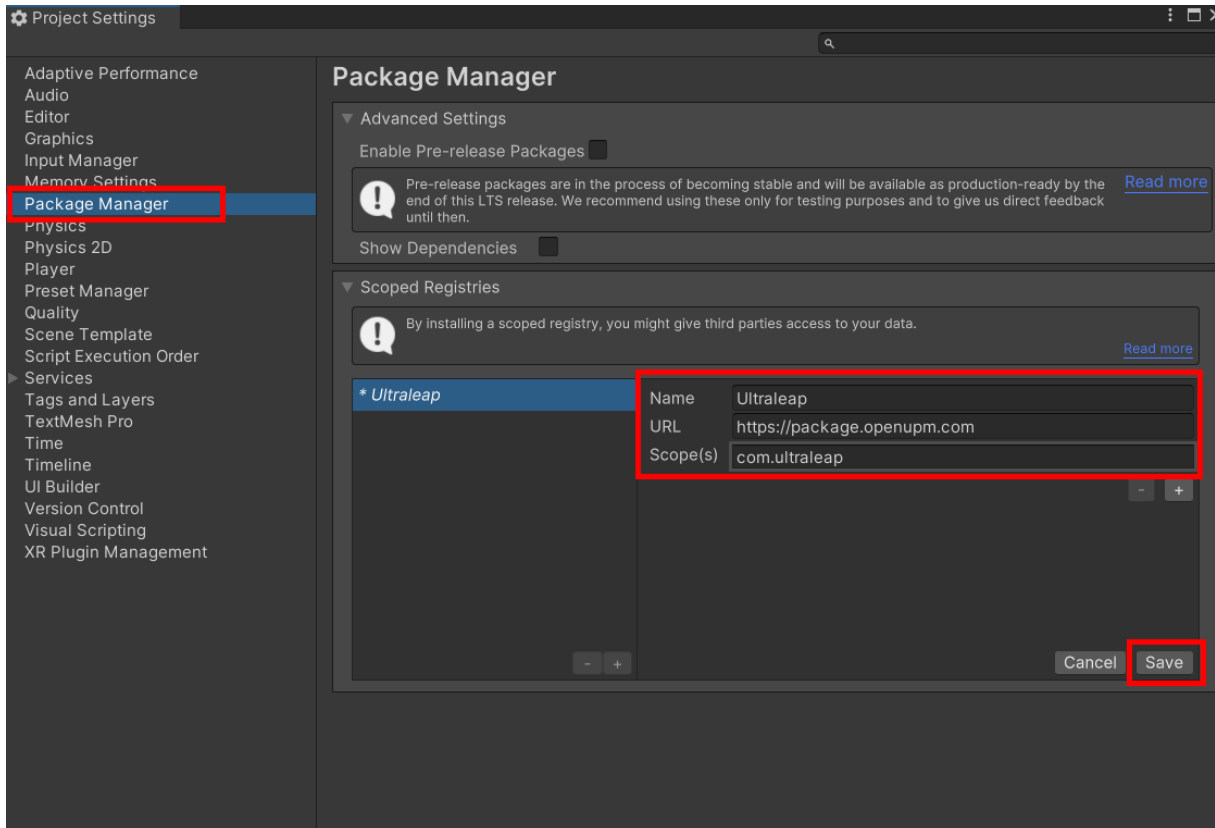


V zozname šablón zvolíme **3D**, projekt si pomenujeme a vytvoríme

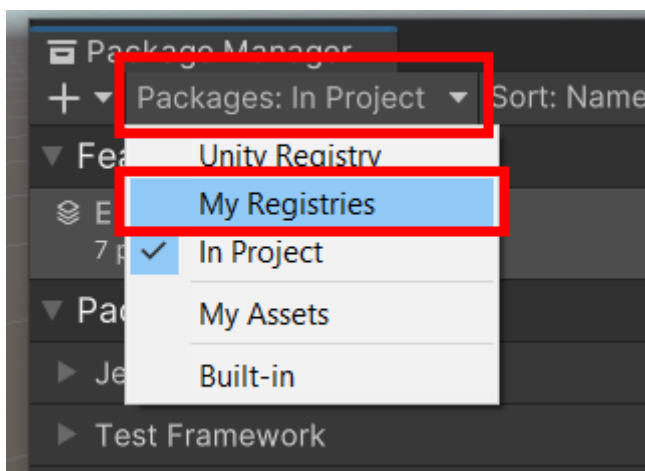


Po vytvorení projektu stiahneme **Leap Motion plugin** cez **Edit -> Project Settings -> Package Manager** kde vložíme:

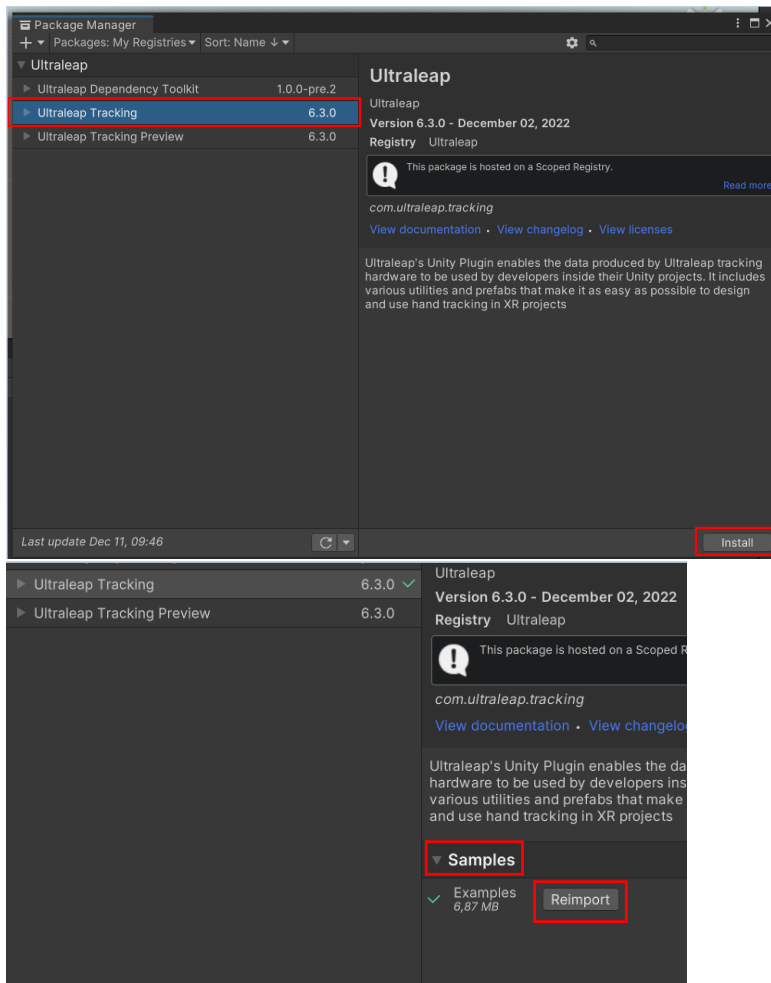
- **Name:** Ultraleap
- **URL:** <https://package.openupm.com/>
- **Scope(s):** com.ultraleap



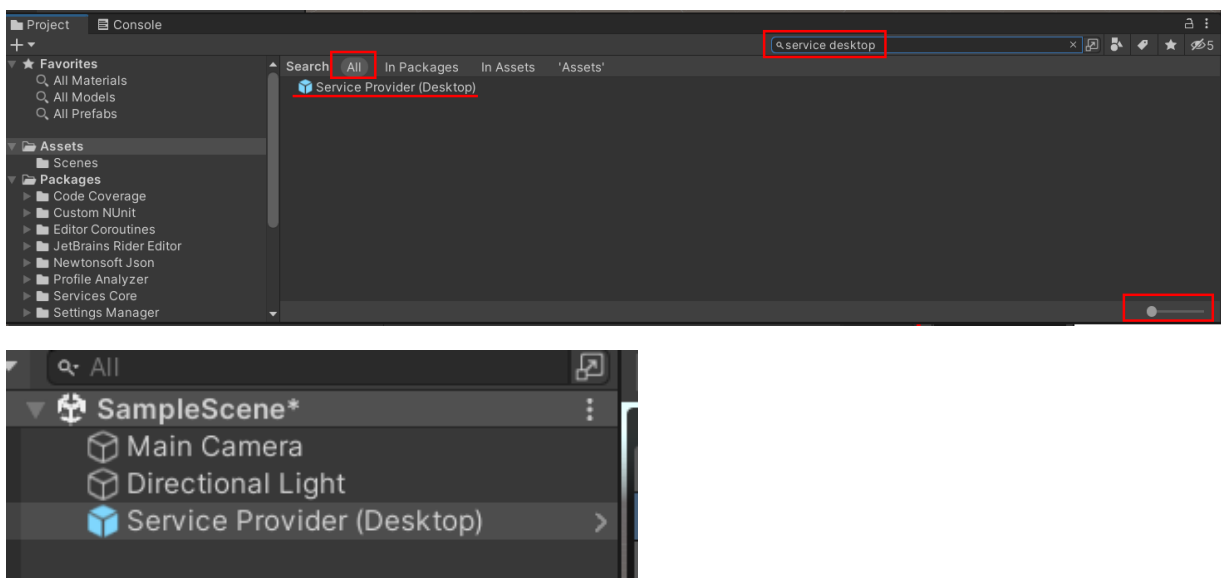
Prejdeme do ďalšieho **Package Manager** cez **Window -> Package Manager** kde v hornej časti prepne **Packages:** na **My Registries**



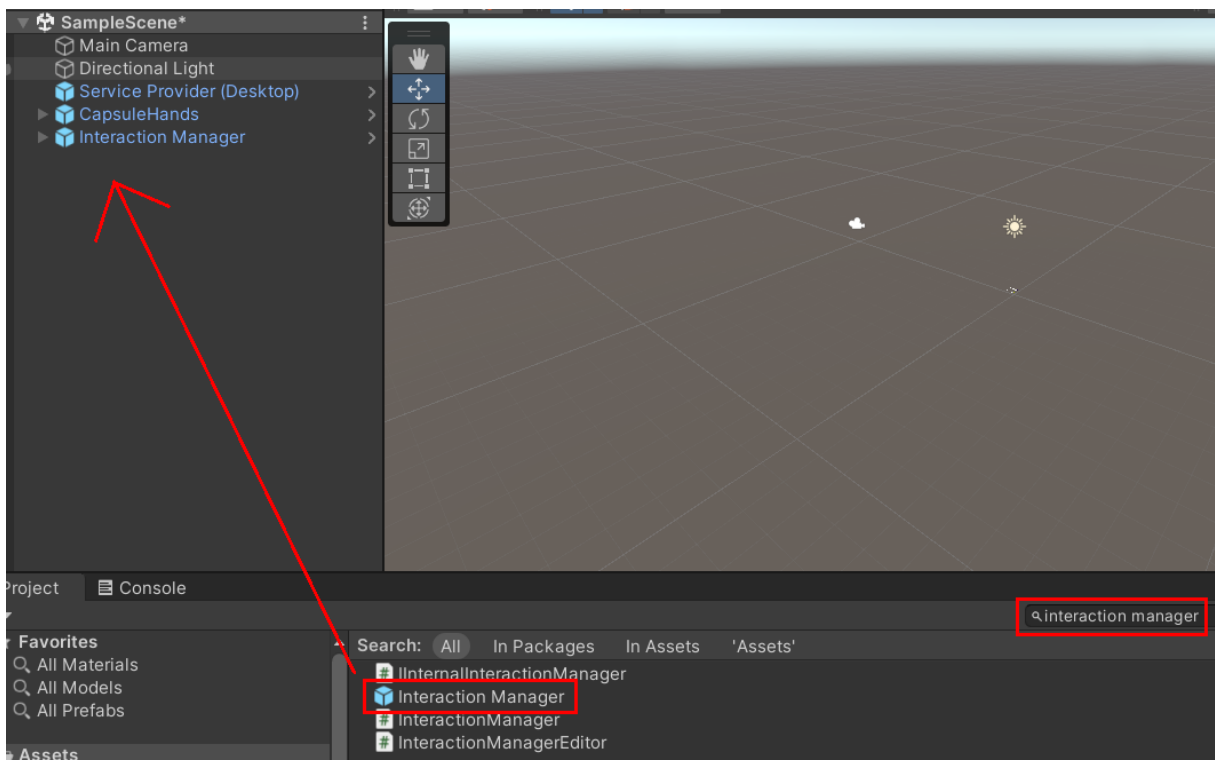
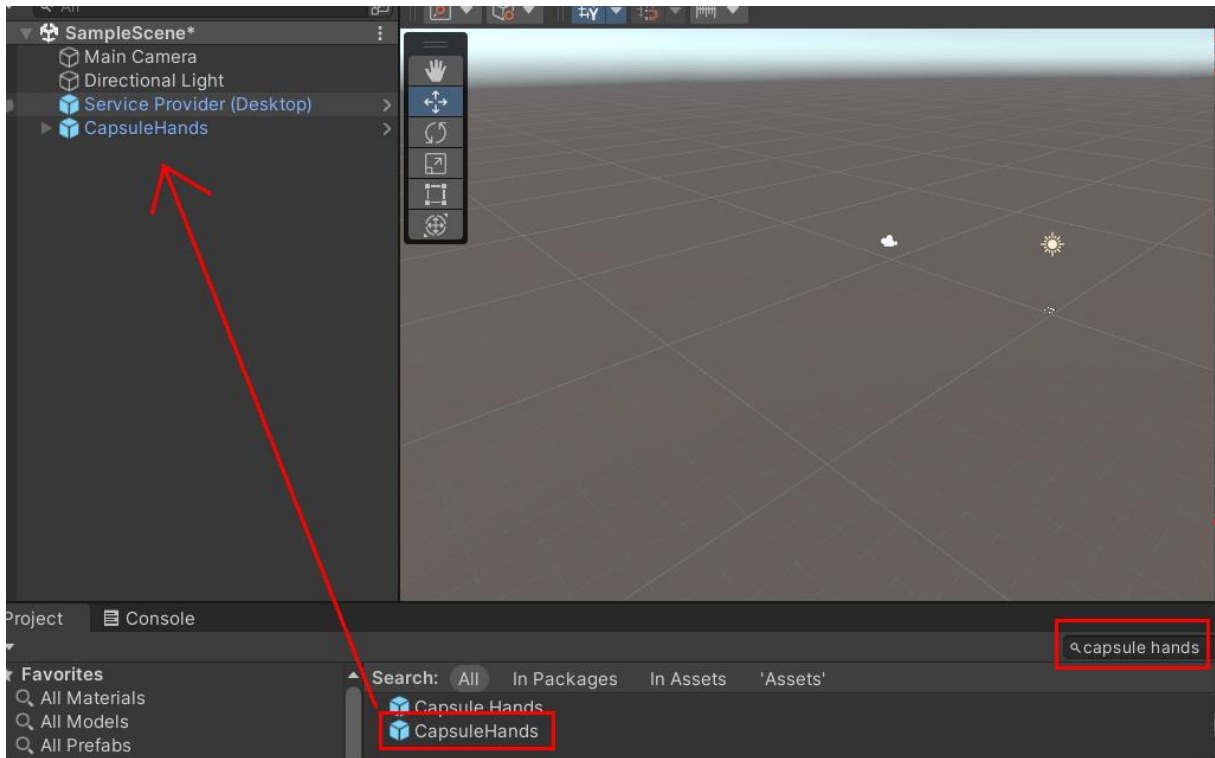
Zvolíme **Ultraleap Tracking** a dáme inštalovať. Po nainštalovaní importujeme **Examples**.



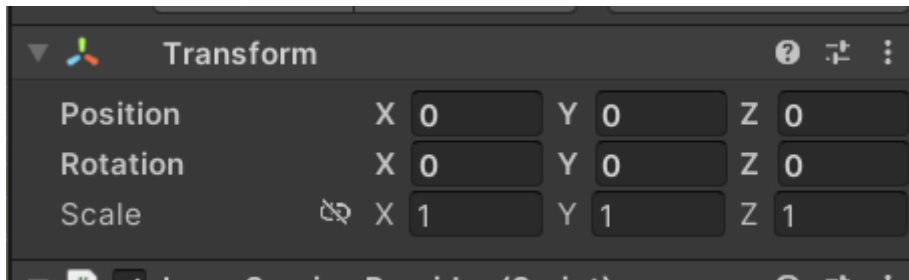
Vyhľadáme **service desktop** (slúži na prepojenie Leap Motion a Unity assets) v **All** (kvôli prehľadnosti súborov si môžeme cez pravý dolný scrollbar zmeniť veľkosť zobrazenia) a presunieme ho do **Hierarchy**.



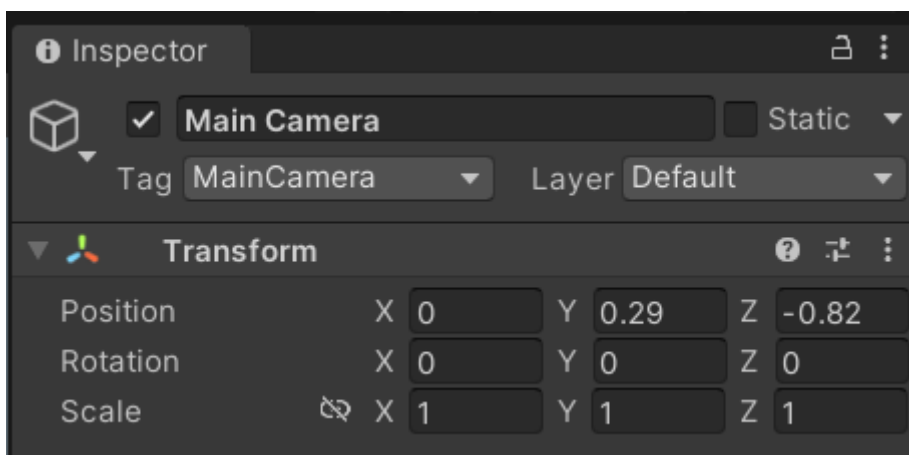
Taktiež pridáme **CapsuleHands** (samotné ruky, ktoré budeme môcť ovládať) a **Interaction Manager** (slúži na interakciu objektov – možnosť s nimi pohybovať)



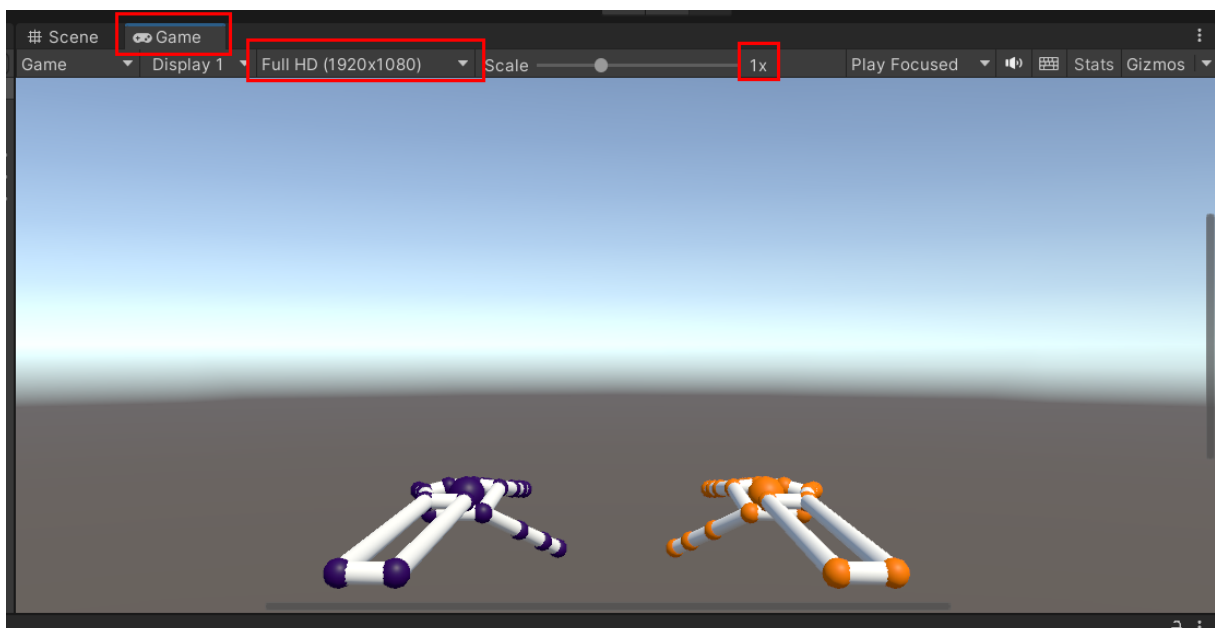
Skontrolujeme u **Service Provider, CapsuleHands, Interaction Manager** či majú default nastavené hodnoty pre **Position, Rotation a Scale**.



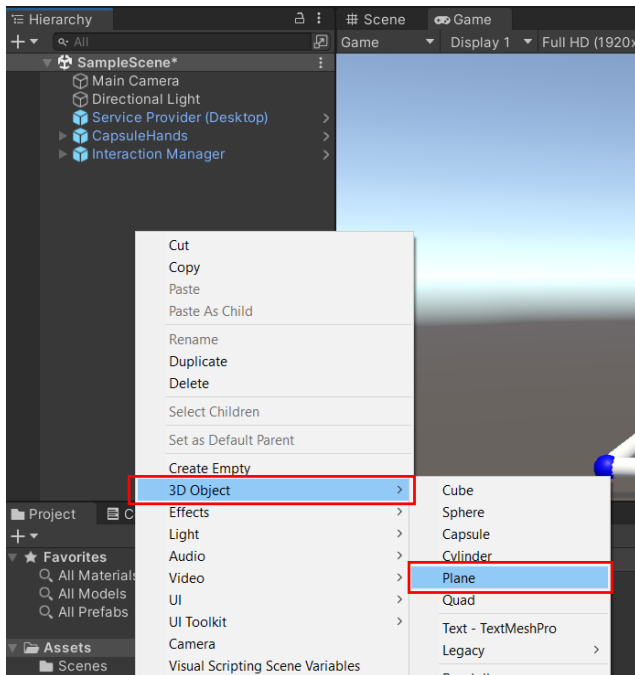
V **Main Camera** nastavíme Transform hodnoty podľa obrázku nižšie



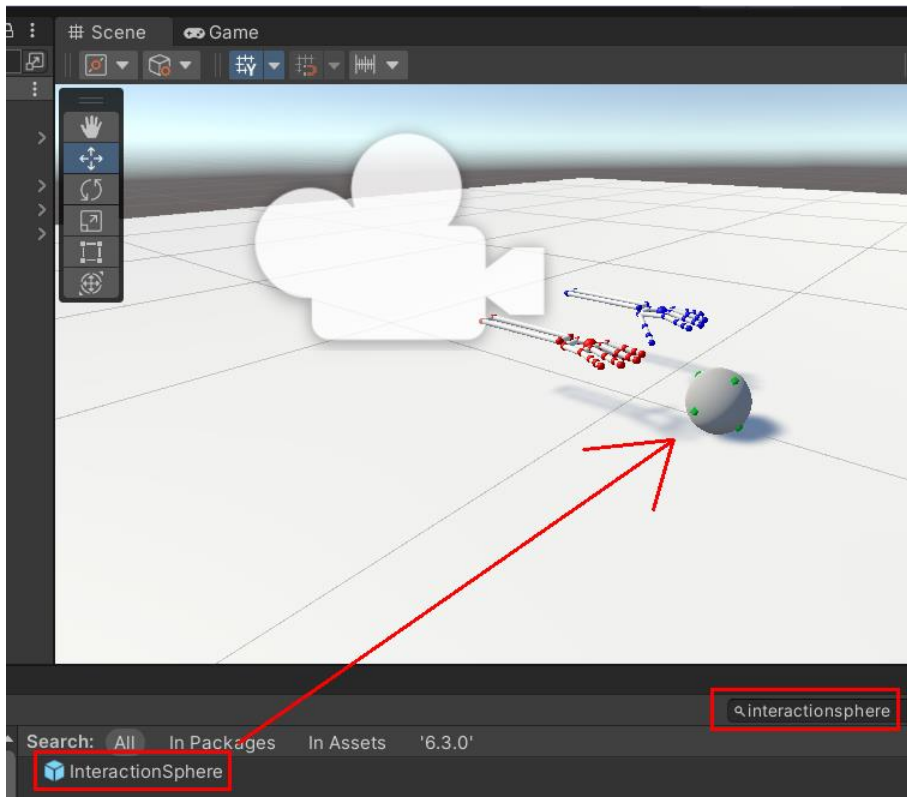
Keď si v hornej časti prepne **Game** a nastavíme scale na **1x**, mali by sme vidieť ruky.



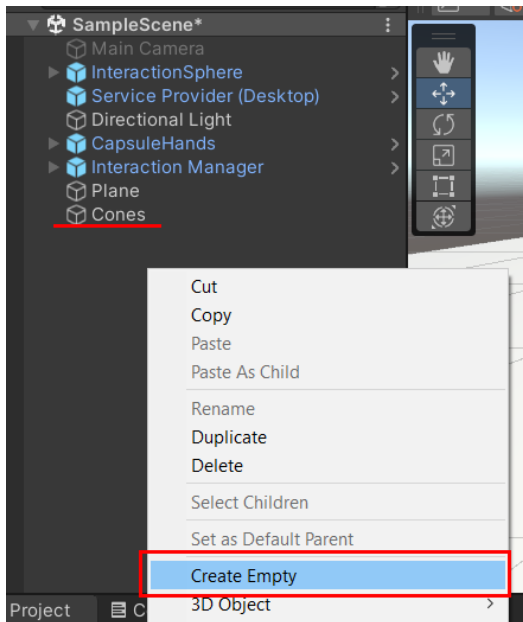
Pridanie platformy – v **Hierarchy** cez pravé tlačidlo pridáme **3D Object** -> **Plane**



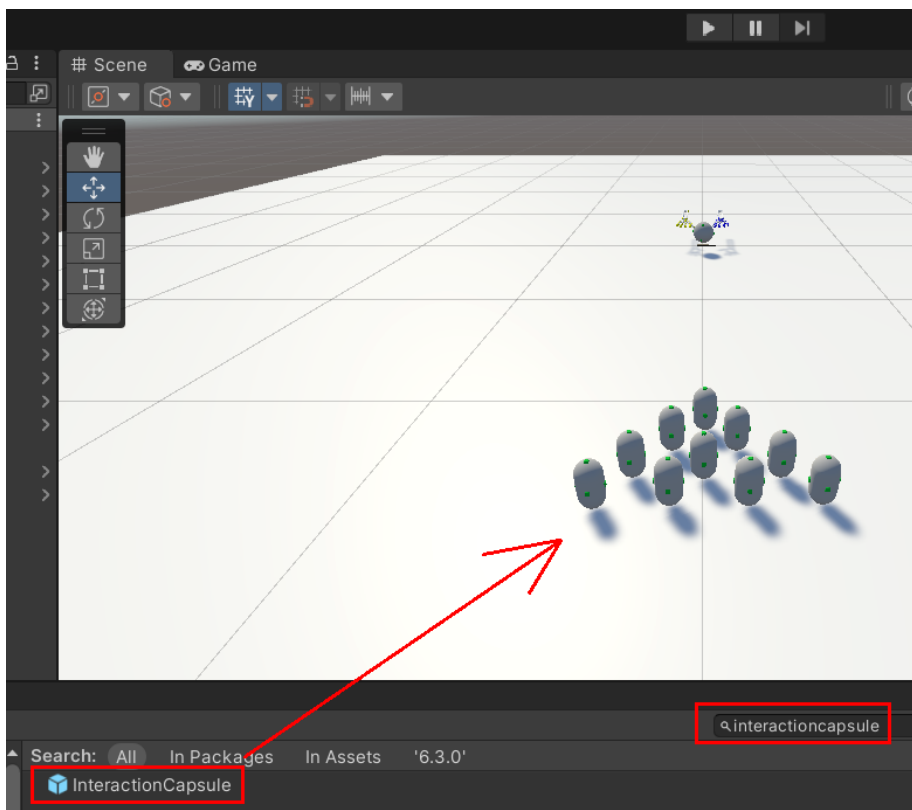
Pridanie objektu – v súboroch vyhľadáme objekt **InteractionSphere**, ktorý následne presunieme priamo do scény tak, aby bol v malej vzdialenosti od rúk.



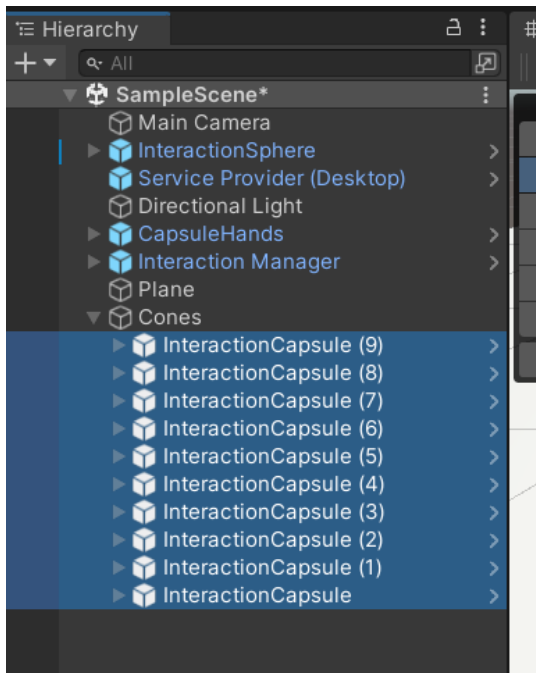
Pridanie kužeľov – v **Hierarchy** vytvoríme empty objekt (**Create Empty**), ktorý pomenujeme napríklad Cones



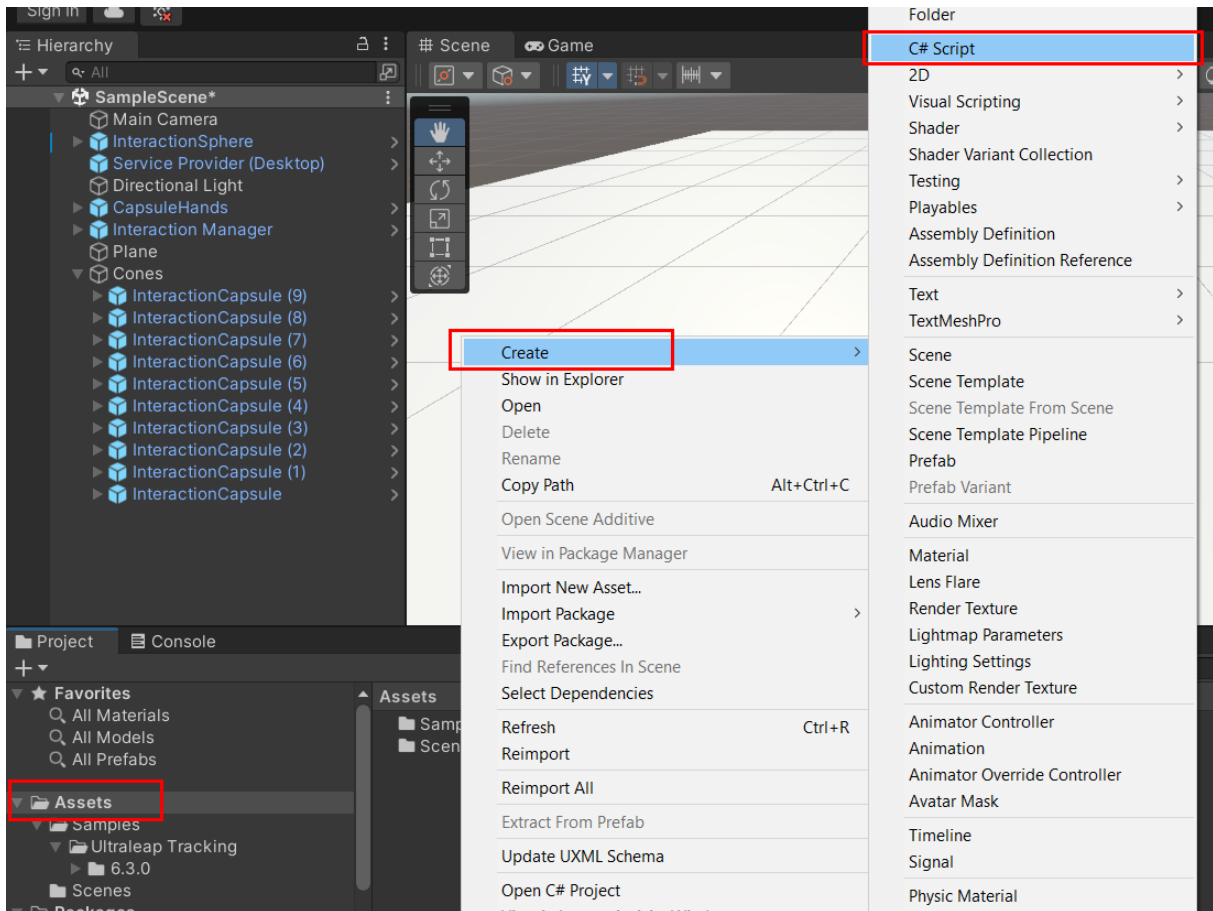
Následne si vyhľadáme objekt **InteractionCapsule** a postupne budeme pridávať tento objekt priamo do scény (tak ako vidieť na obrázku).



V **Hierarchy** presunieme všetky vytvorené **InteractionCapsule** do **Cones**



V **Assets** vytvoríme **C# Script**, pomenujeme ho **Bowling** a otvoríme.



Na začiatku si pridáme knižnicu **TMPro**

```
using TMPro;
```

Do class **Bowling** pridáme:

```
private int score = 0;
private int counter = 0;
public TMP_Text ScoreText;

// List, do ktoreho sa bude vkladat position kuzelov
private List<Vector3> objects = new List<Vector3>();

// Premenna, do ktorej bude position bowlingovej gule
private Vector3 ballVector = new Vector3();
```

Do funkcie **Start** vložíme:

```
// zapise position bowlingovej gule
ballVector = GameObject.Find("InteractionSphere").transform.position;

// vyhlada vsetky kuzele a ulozi do Listu ich position
for(int i = 0; i < gameObject.transform.childCount; i++) {
    objects.Add(gameObject.transform.GetChild(i).position);
}

ScoreText.text = "Skóre: 0\nPokusov: 0";
score = 0;
```

Do funkcie **Update** vložíme:

```
// stlacenie medzernika
if (Input.GetKeyDown("space"))
{
    counter++;

    for(int i = 0; i < gameObject.transform.childCount; i++)
    {
        // kontrola rotacie kuzelov (ci spadli alebo nie)
        if (gameObject.transform.GetChild(i).rotation.x > 0.40 ||
            gameObject.transform.GetChild(i).rotation.x < -0.40 ||
            gameObject.transform.GetChild(i).rotation.z > 0.40 ||
            gameObject.transform.GetChild(i).rotation.z < -0.40
        ) {
            score++;
        }

        // vylistuje vsetky kuzele a nastavi im prvotnú position
        // nastavi aj default velocity a rotation
        gameObject.transform.GetChild(i).position = objects[i];
        gameObject.transform.GetChild(i).GetComponent<Rigidbody>().velocity =
Vector3.zero;
        gameObject.transform.GetChild(i).GetComponent<Rigidbody>().angularVelocity = Vector3.zero;
        gameObject.transform.GetChild(i).rotation = Quaternion.Euler(0, 0, 0);
    }

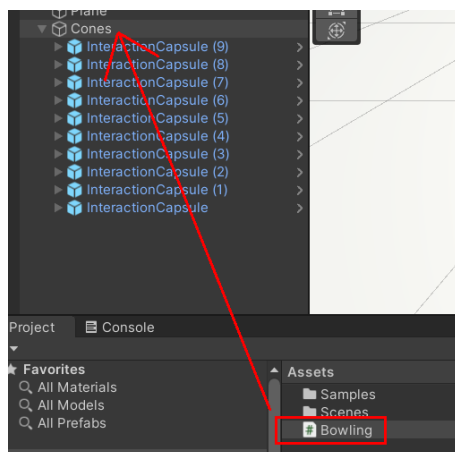
    // nastavi text na obrazovke
    ScoreText.text = "Skóre: " + score + "\nPokusov: " + counter;

    // vyhlada bowlingovu gulu nastavi jej prvotnú position
    // nastavi aj default velocity a rotation
    GameObject.Find("InteractionSphere").transform.position = ballVector;
    GameObject.Find("InteractionSphere").transform.GetComponent<Rigidbody>().velocity = Vector3.zero;
    GameObject.Find("InteractionSphere").transform.GetComponent<Rigidbody>().angularVelocity = Vector3.zero;
    GameObject.Find("InteractionSphere").transform.rotation =
Quaternion.Euler(0, 0, 0);
}
```

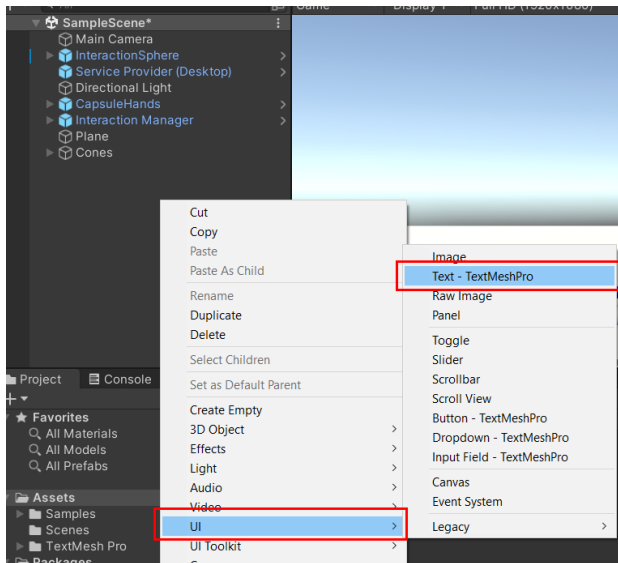
Celý kód:

```
6 public class Bowling : MonoBehaviour
7 {
8     private int score = 0;
9     private int counter = 0;
10    public TMP_Text ScoreText;
11
12    // List, do ktoreho sa bude vkladat position kuzelov
13    private List<Vector3> objects = new List<Vector3>();
14
15    // Premenna, do ktorej bude position bowlingovej gule
16    private Vector3 ballVector = new Vector3();
17
18    void Start() {
19        // zapise position bowlingovej gule
20        ballVector = GameObject.Find("InteractionSphere").transform.position;
21
22        // vyhlada vsetky kuzele a ulozi do Listu ich position
23        for(int i = 0; i < gameObject.transform.childCount; i++) {
24            objects.Add(gameObject.transform.GetChild(i).position);
25        }
26        ScoreText.text = "Skóre: 0\nPokusov: 0";
27        score = 0;
28    }
29
30    void Update() {
31        // stlacenie medzernika
32        if (Input.GetKeyDown("space")) {
33            counter++;
34
35            for(int i = 0; i < gameObject.transform.childCount; i++) {
36                if (gameObject.transform.GetChild(i).rotation.x > 0.40 ||
37                    gameObject.transform.GetChild(i).rotation.x < -0.40 ||
38                    gameObject.transform.GetChild(i).rotation.z > 0.40 ||
39                    gameObject.transform.GetChild(i).rotation.z < -0.40
40                ) {
41                    score++;
42                }
43
44                // vylistuje vsetky kuzele a nastavi im prvotnú position, nastavi aj default velocity a rotation
45                gameObject.transform.GetChild(i).position = objects[i];
46                gameObject.transform.GetChild(i).GetComponent<Rigidbody>().velocity = Vector3.zero;
47                gameObject.transform.GetChild(i).GetComponent<Rigidbody>().angularVelocity = Vector3.zero;
48                gameObject.transform.GetChild(i).rotation = Quaternion.Euler(0, 0, 0);
49            }
50
51            ScoreText.text = "Skóre: " + score + "\nPokusov: " + counter; // nastavi text na obrazovke
52
53            // vyhlada bowlingovu gulu nastavi jej prvotnú position, nastavi aj default velocity a rotation
54            GameObject.Find("InteractionSphere").transform.position = ballVector;
55            GameObject.Find("InteractionSphere").transform.GetComponent<Rigidbody>().velocity = Vector3.zero;
56            GameObject.Find("InteractionSphere").transform.GetComponent<Rigidbody>().angularVelocity = Vector3.zero;
57            GameObject.Find("InteractionSphere").transform.rotation = Quaternion.Euler(0, 0, 0);
58        }
59    }
60 }
```

Script uložíme a presunieme do **Cones** v *Hierarchy*

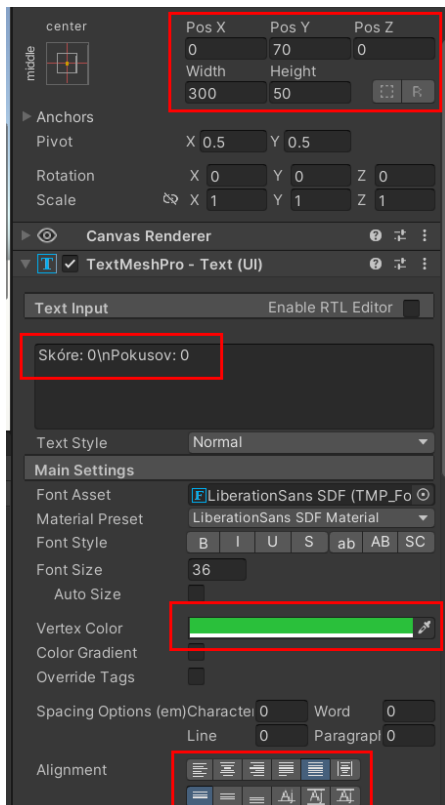


Pridanie textu – v Hierarchy vytvoríme Text cez UI -> Text - TextMeshPro

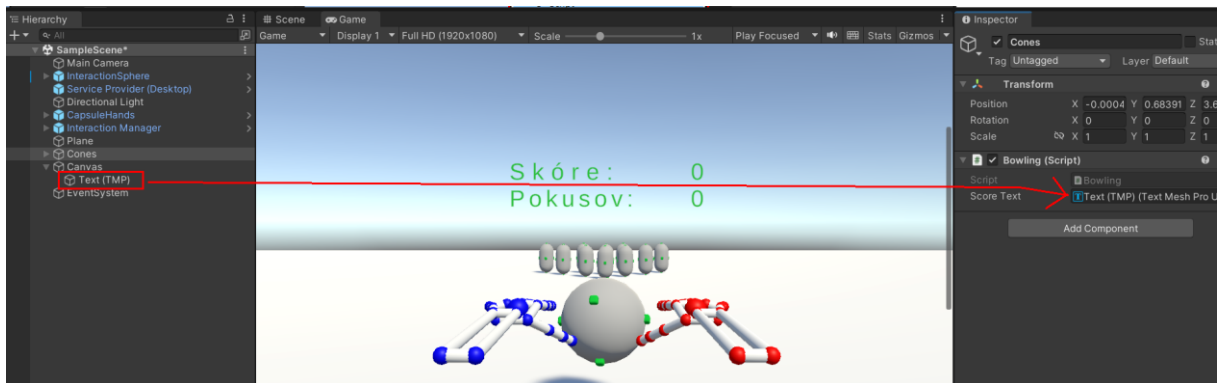


V Hierarchy klikneme na **Text (TMP)** a nastavíme:

- Súradnice X, Z: **0**
- Y: **70**
- Width: **300**
- Text Input: **Skóre: 0\nPokusov: 0**
- Vertex Color napríklad: **2BC13B** (hexadecimal)
- Alignment: **Flush**



V **Hierarchy** klikneme na **Cones** a do **Score Text** (pod scriptom) vložíme z Hierarchy **Text (TMP)**.



Typy:

- Reštart sa vyvolá medzerníkom
- Najlepšie je „bowlingovú“ guľu odrážať pästou

Hru už len otestujeme