

Game Engines

Andrej Lúčný

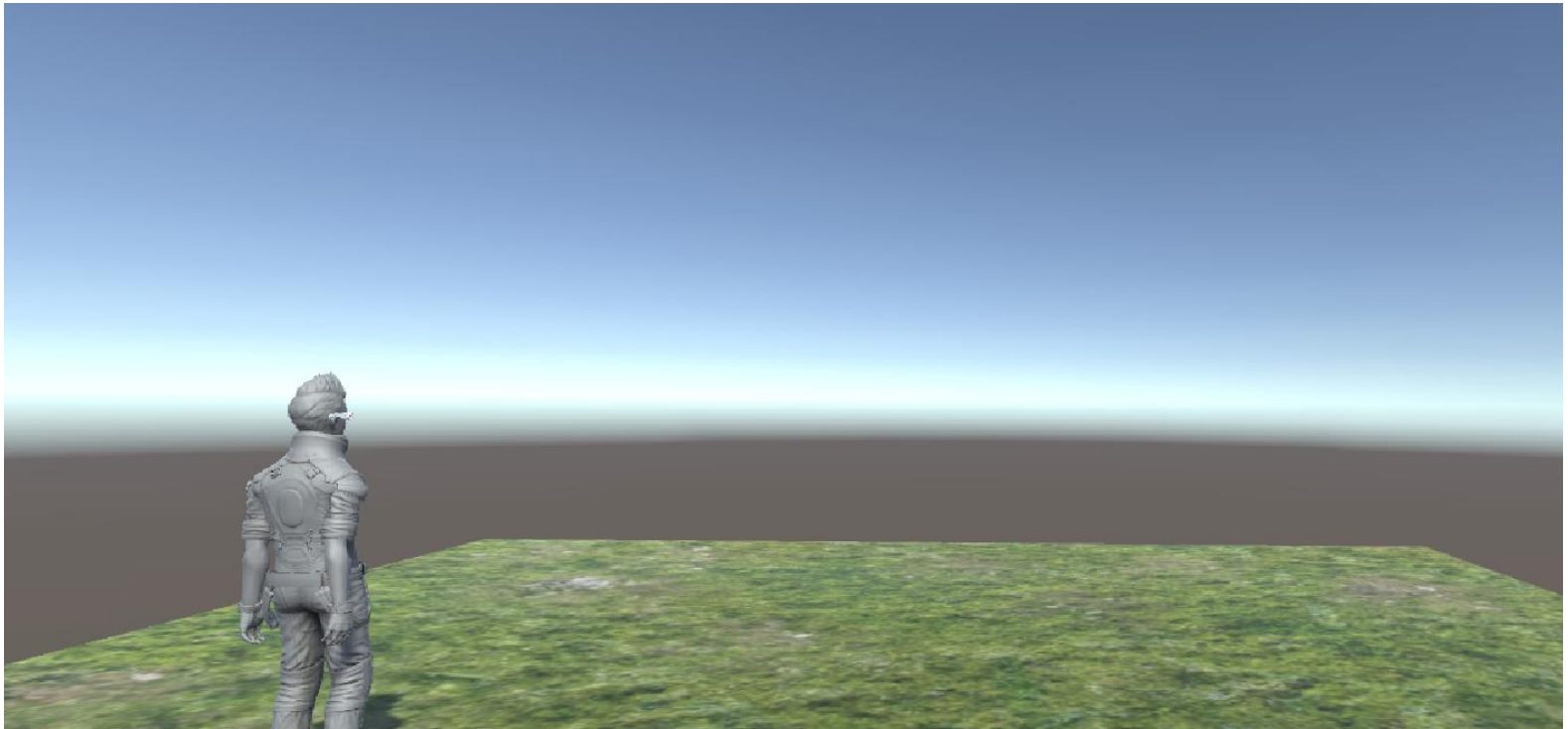
Katedra aplikovanej informatiky

lucny@fmph.uniba.sk

<http://www.agentspace.org/ge>

Networking

Vo verzii s VS17 je UNet, ktorý je už deprecated



Vyjdeme z projektu Ethan – ver5.

Abstraktný vstup

Abstrakcia nad joystickmi, šípkami, ...

Input.GetAxis(name) vracia -1.0 .. 1.0

```
void Update () {  
    float forward = Input.GetAxis("Vertical");  
    float turn = Input.GetAxis("Horizontal");  
    if (forward < 0) forward = 0;  
    anim.SetFloat("Forward", forward);  
    anim.SetFloat("Turn", turn);  
}
```

Cross platform Input Manager

Slúži na ovládanie herných objektov cez sieť

```
using UnityStandardAssets.CrossPlatformInput;
```

miesto:

```
Input.GetAxis("Vertical")
```

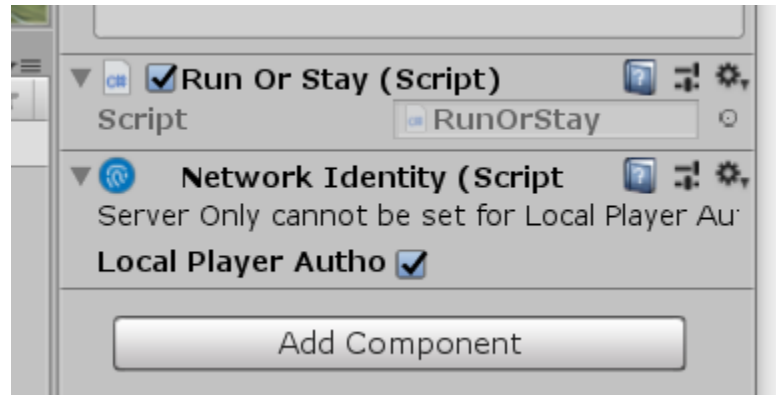
použijeme:

```
CrossPlatformInputManager.GetAxis("Vertical")
```

Funguje to rovnako, ale tým pádom je objekt schopný prijať vstup nielen z lokálnej ale aj vzdialenej klávesnice

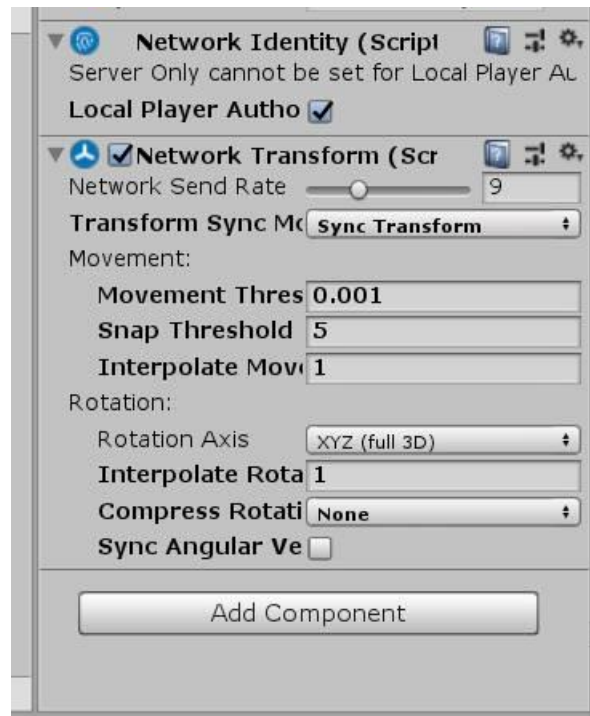
Network Identity

Pridáme komponent Network identity:



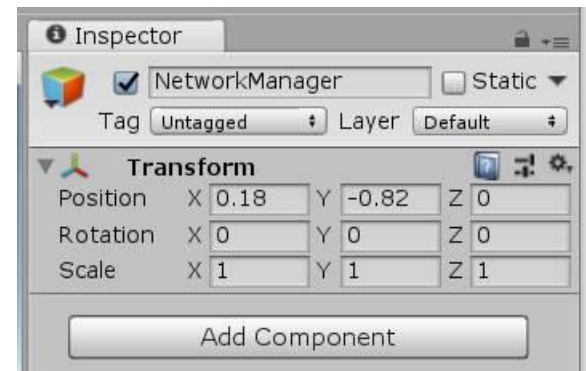
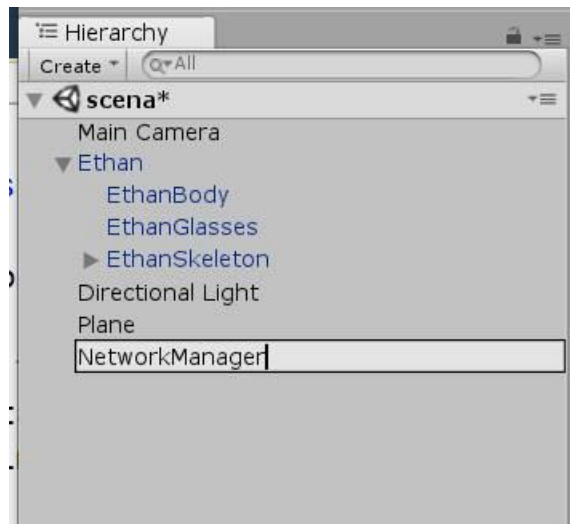
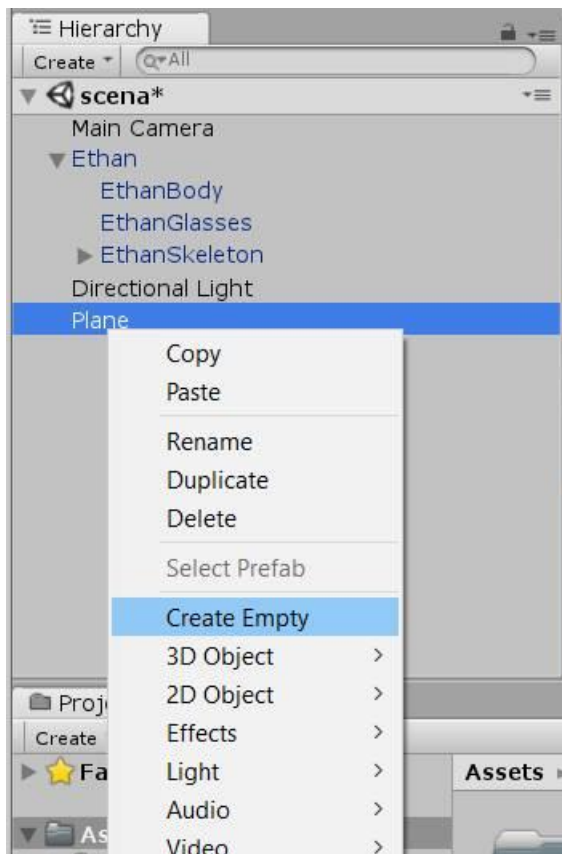
Network Transform

Pridáme komponent ktorý umožňuje sieťovú synchronizáciu transform-ov vo všetkých inštanciách hry:



Network Manager

Vytvára sa trochu zvlášťne: v hierarchy window vytvoríme empty objekt a premenujeme ho na „Network Manager“



Network Manager

**Objektu „Network Manager“
pridáme komponenty
„NetworkManager“ a
„NetworkManagerHUD“**

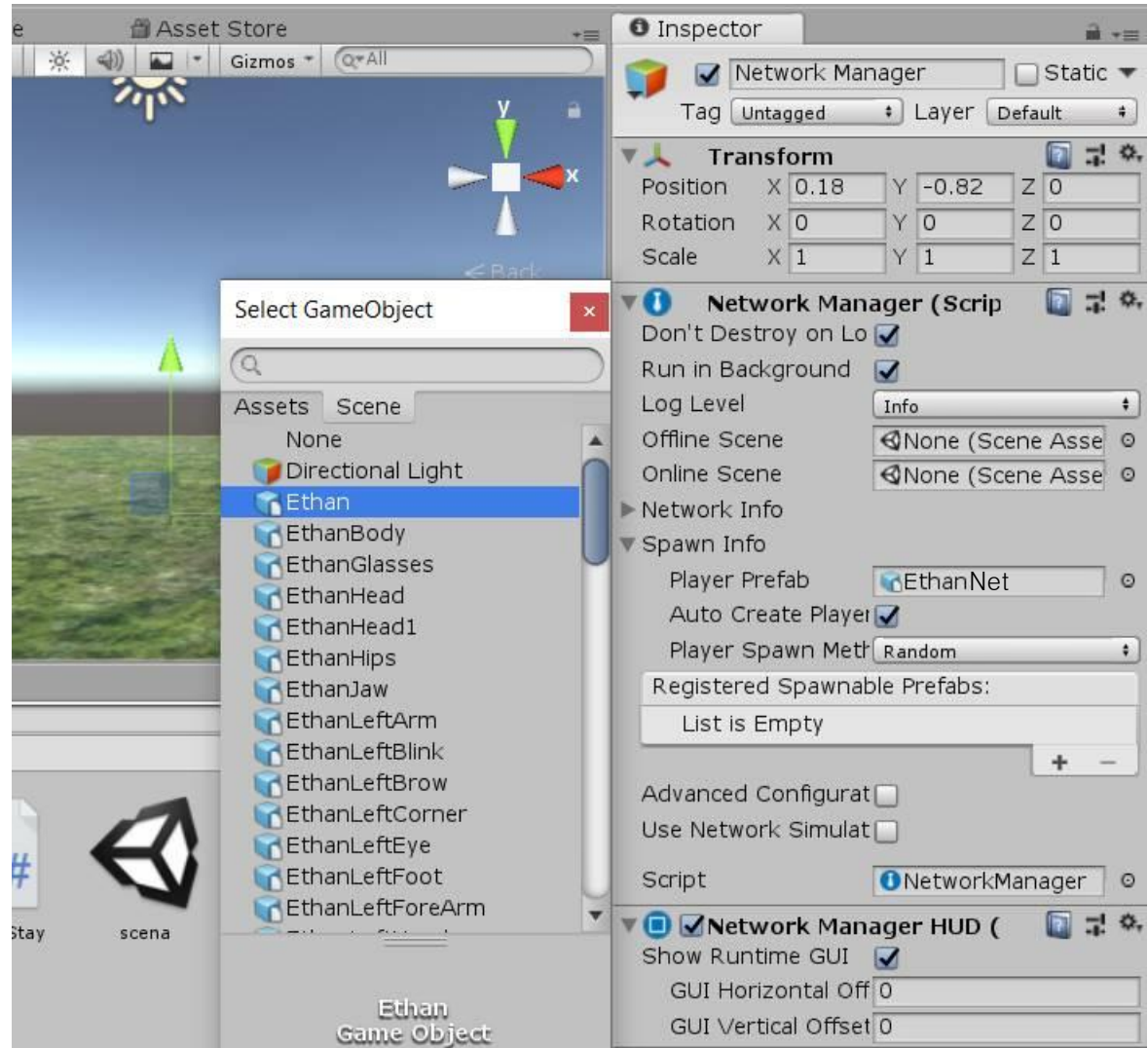


Network Manager

Z Ethana v
scéne
urobíme
Prefab (z
Hierarchy
windows ho
stiahneme do
Assetov)

Vymažeme
ho zo scény.

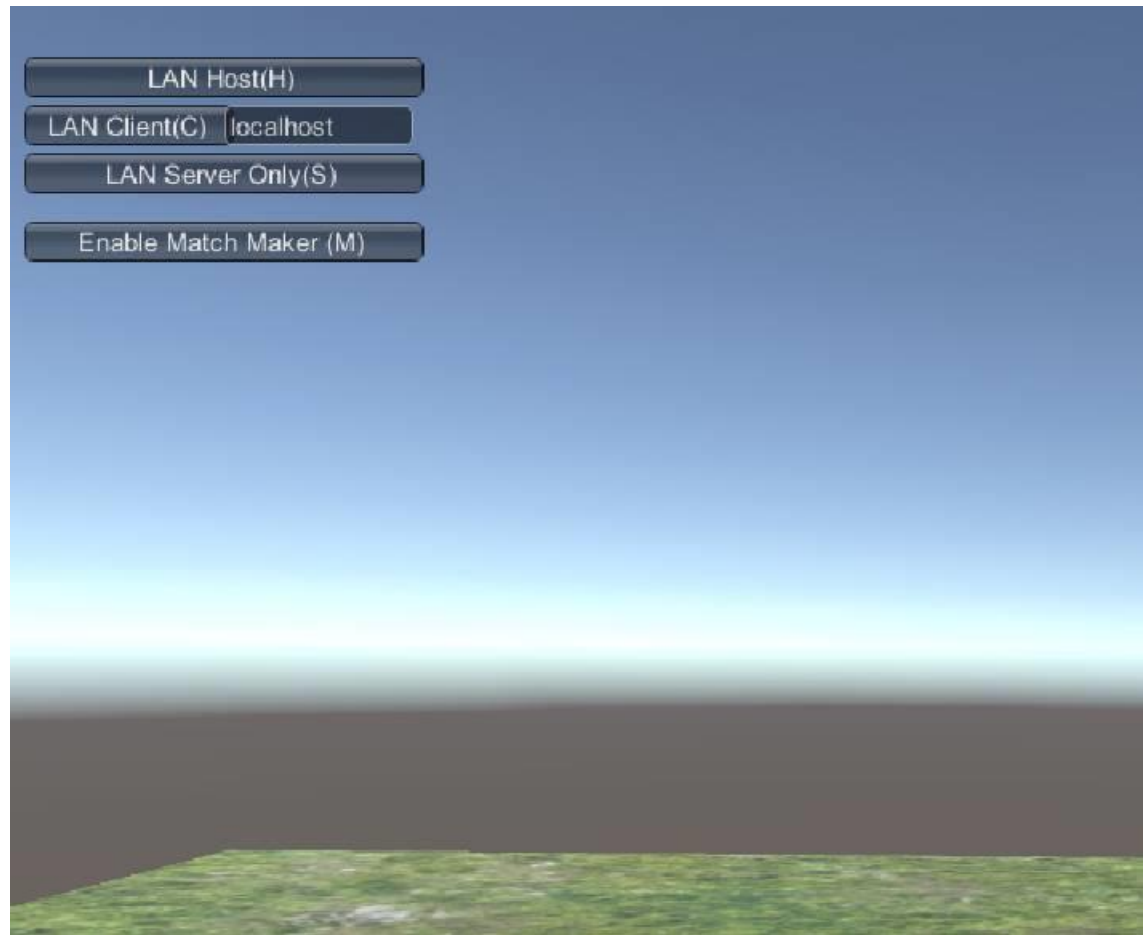
Nastavíme v
Network
Manageri
Player-a



Network Manager

**Následkom
toho pri
spustení hry
bude
ponúknuté či
hra beží ako
host alebo ako
client. Prvý
musí byť host.**

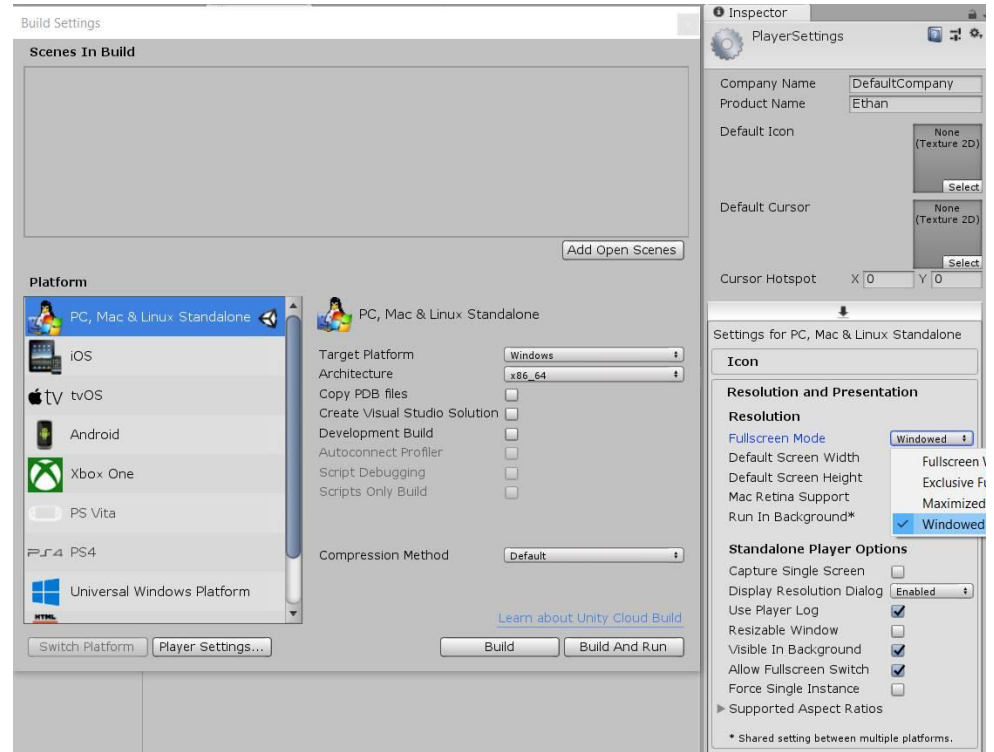
**Kým
nezvolíme,
žiadny avatar v
hre nie je**



Build

Ked' teraz chceme spustit' hru raz ako host a raz ako client, najjednoduchšie je zbuildovat' ju do inštalačného balíka, spustit' host v Unity a cez exe v balíku spustit' client-a

V Build Settings / Player Settings / Resolution musíme akurát vypnúť Full screen mode

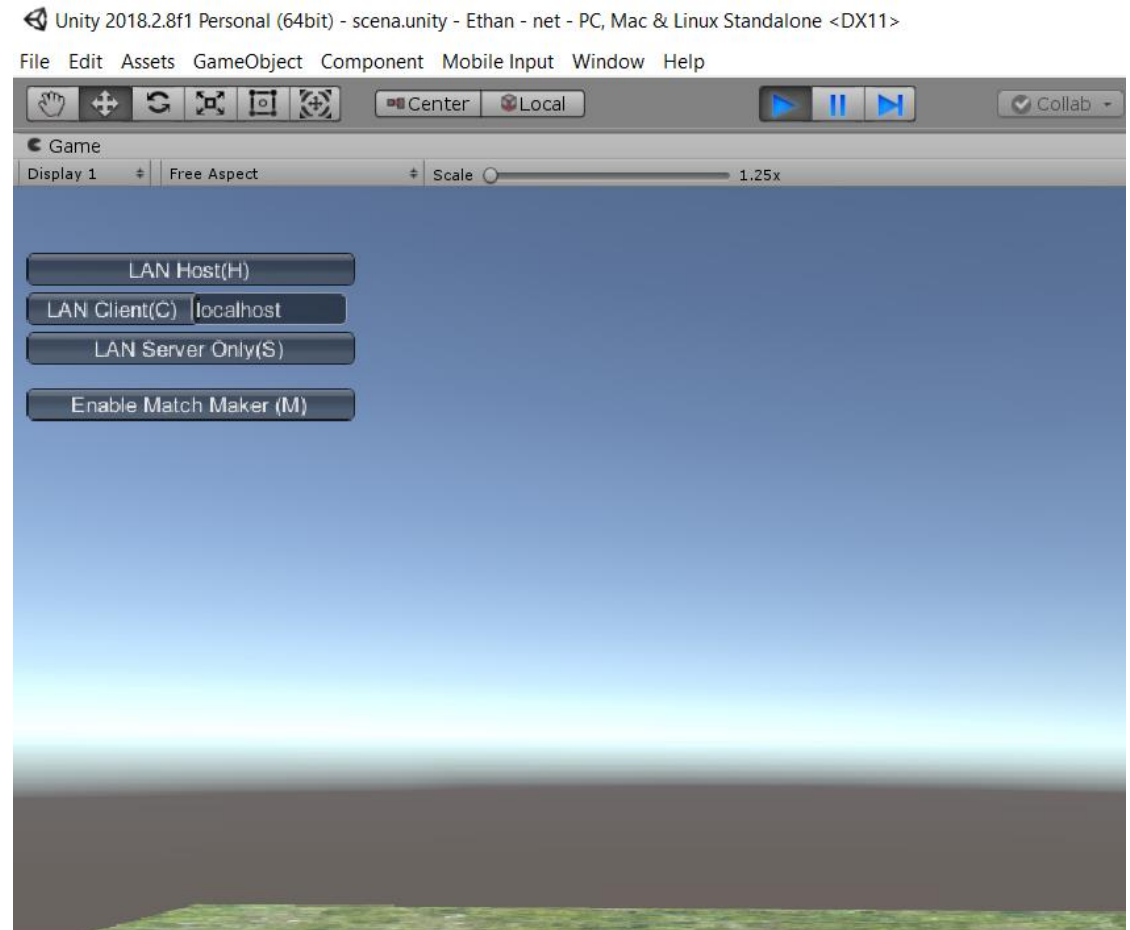


**Build
(uložíme niekde mimo projekt
do deployment directory)**

Run

**Spustíme hru v
prostredí Unity a
zvolíme Host**

**Tým pádom sa
objaví aj avatar.
Urobme s ním jeden
krok dopredu.**

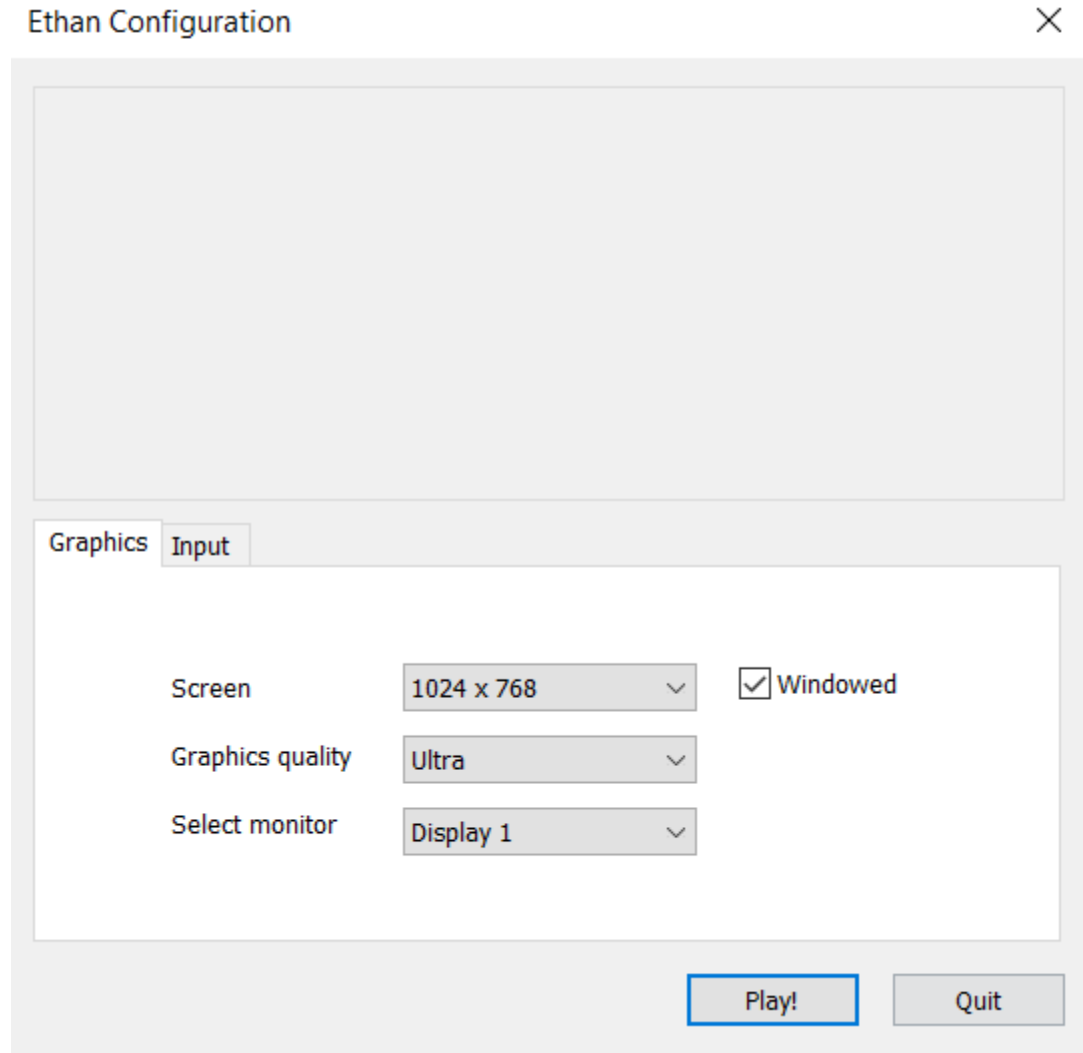


**Spustíme zbuidované
Ethan.exe v
deployment directory,
dáme Play a zvolíme
Client**

**Tým pádom v oboch
scénach
spriahnutých cez sieť
budú dva avatary
a z obidvoch hýbeme
obomi**

**Keď dáme Stop v
jednej druhej,
zmiznú**

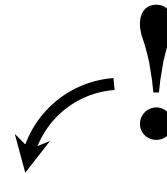
Play



SetupLocal Player

**Prefabu EthanNet vypneme skript RunOrStay
A pridáme mu skript SetupLocalPlayer:**

```
using UnityEngine.Networking;
```

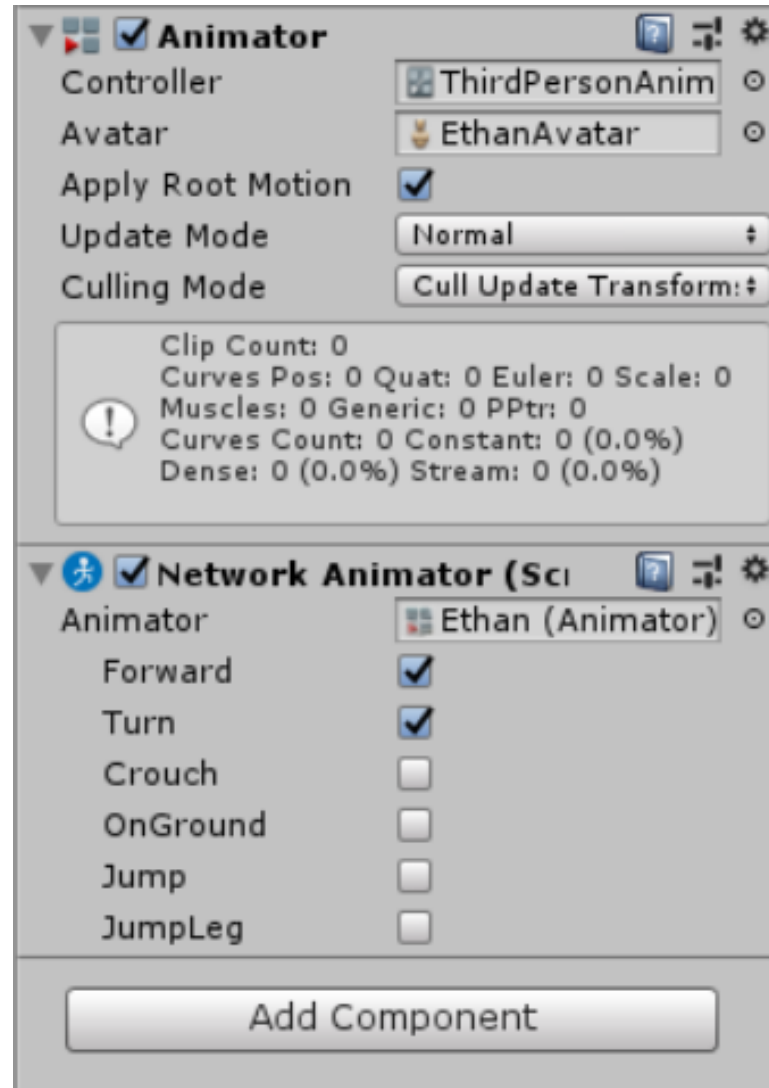


```
public class SetupLocalPlayer : NetworkBehaviour {  
  
void Start () {  
    if (!isLocalPlayer)  
    {  
        GetComponent<RunOrStay>().enabled = false;  
    }  
}  
  
}
```

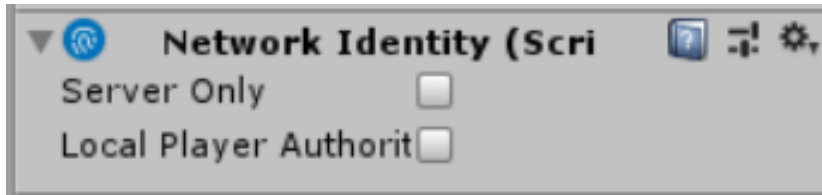
**Zbuildujeme, spustíme oba a teraz už
ovláda každý hráč len toho svojho**

Synchronizácia animácii

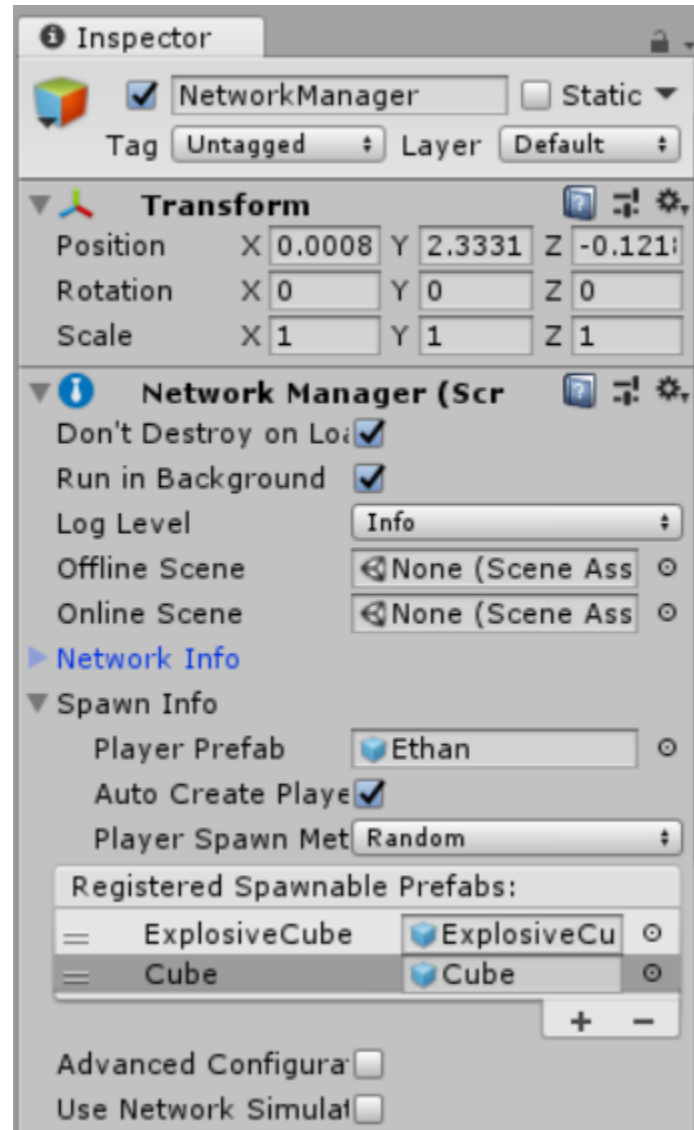
Nestačí synchronizovať transform avatara, musíme synchronizovať aj jeho animačný stav.



Synchronizácia scény



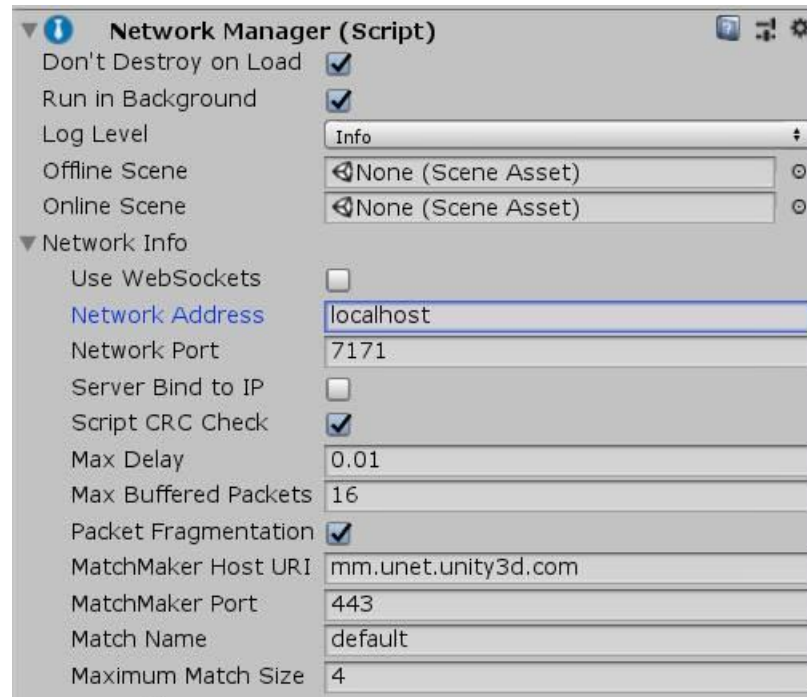
Synchronizovať možno nielen playera ale aj objekty zo scény. Stačí ich vyrobiť ako inštancie prefabov, ktoré majú network identity a zaregistrovať ich ako spawnable prefab v Network manageri



Spustenie na rôznych uzloch siete

Defaultný port Unity je 7777

V terminálke však tento port nie je povolený na firewalle, povolený je 7171



Zbuildujeme

Spustenie na rôznych uzloch siete

V cmd cez ipconfig si zistíme IP adresu svojho stroja. Na ňom v Unity spustíme hru a zvolíme Host. Urobíme s avatarom krok dopredu

Obsah deployment directory prenesieme na iný počítač v sieti

Na ňom spustíme exe a pred zvolením Client, nahradíme „localhost“ za zistenú IP adresu

Následne by mala hra fungovať cez sieť.

HUD = heads-up display



Synchronizácia premenných

UNet podporuje synchronizáciu na úrovni buffrov niekoľkých základných typov:

- `SyncListString`
- `SyncListFloat`
- `SyncListInt`
- `SyncListUInt`
- `SyncListBool`

Tieto získate zámenou `MonoBehaviour` za jeho odvodeninu `NetworkBehavior`

možno si definovať aj vlastnú synchronizovanú štruktúru pomocou `SyncListStruct`

```
public class MyScript : NetworkBehaviour
{
    public struct Buf
    {
        public int id;
        public string name;
        public float timer;
    };

    public class TestBufs : SyncListStruct<Buf> {}
    TestBufs m_bufs = new TestBufs();

    void BufChanged(Operation op, int itemIndex)
    {
        Debug.Log("buf changed:" + op);
    }

    void Start()
    {
        m_bufs.Callback = BufChanged;
    }
}
```



Moderná náhrada od 10.11.2020: free asset Mirror