

Game Engines

Andrej Lúčný

Katedra aplikovanej informatiky

lucny@fmph.uniba.sk

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

Použitie Animátorov

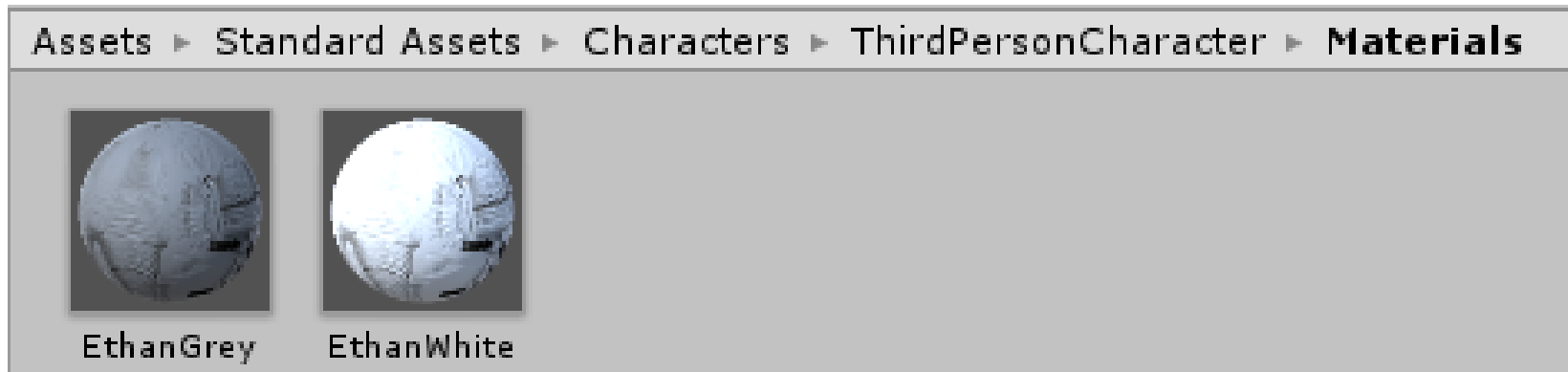
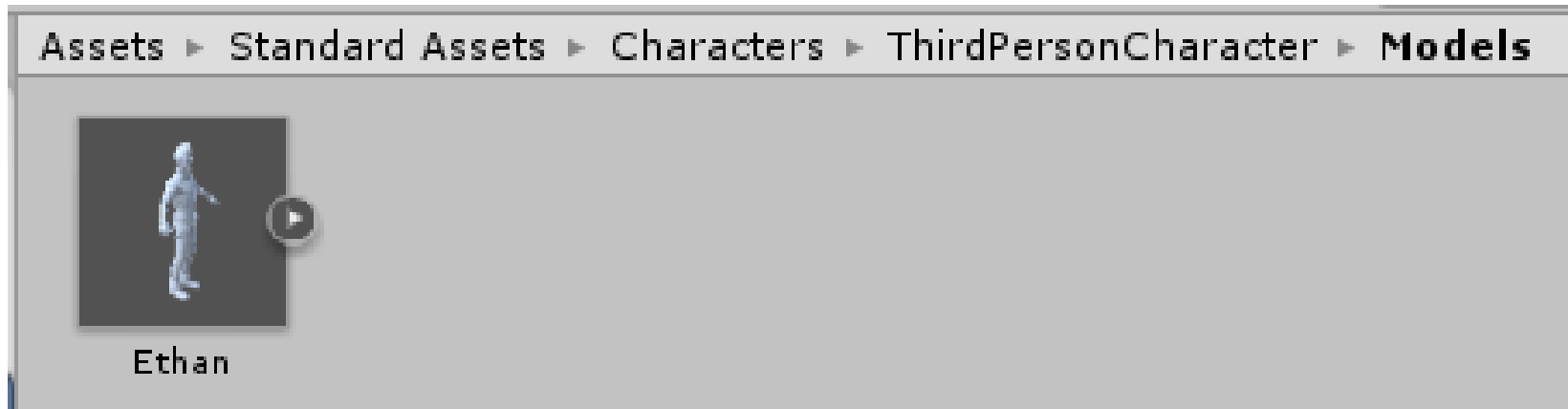
Vyjdeme z projektu Ethan – ver 1

V assetoch máme avatara Ethan, v scéne podlahu

Naším cieľom bude teraz riadiť avatara klávesovými skratkami zo skriptu: kráčanie, behanie, skákanie

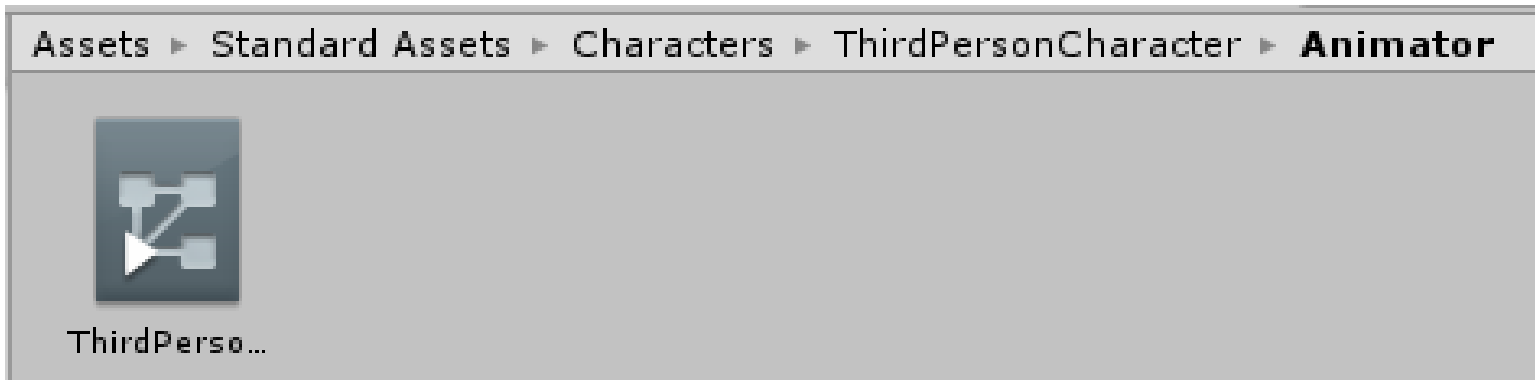


Dôležité adresáre



(Použite Ethan Grey, je to krajšie)

Dôležité adresáre



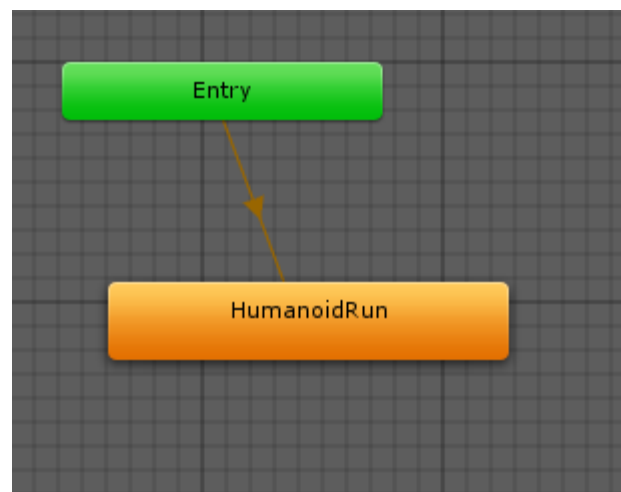
Vlastný animátor controller

Medzi Assetmi cez create spravíme vlastný animator, nazveme ho SimpleAnimator

Otvoríme (dvojklikom naň) okno Animator a potiahnutím do priestoru project window ho vytiahneme z tabu do samostaného okna, kde s nim môžeme lepšie pracovať

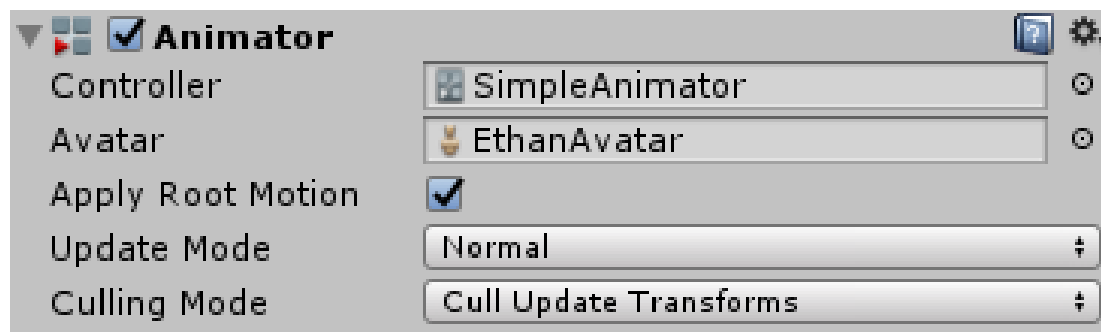
Vložíme do neho Animáciu HumanoidRun

(že ide o animáciu sa môžeme presvedčiť keď na ju klikneme v project window a v inspectore zvolíme animation a dáme si ju prehrať)



Vlastný animátor controller

**V animatore avatara nastavime tento nami vytvoreny
AnimationControler**



Spustíme hru

Vlastný animátor controller

Aby nám avatar neušiel, museli by sme skriptom hýbať s kamerou a sledovať avatara (poznáme z predchádzajúcich cvičení)

Môžeme ho ale nechať bežať na mieste a to vypnutím

Apply Root Motion

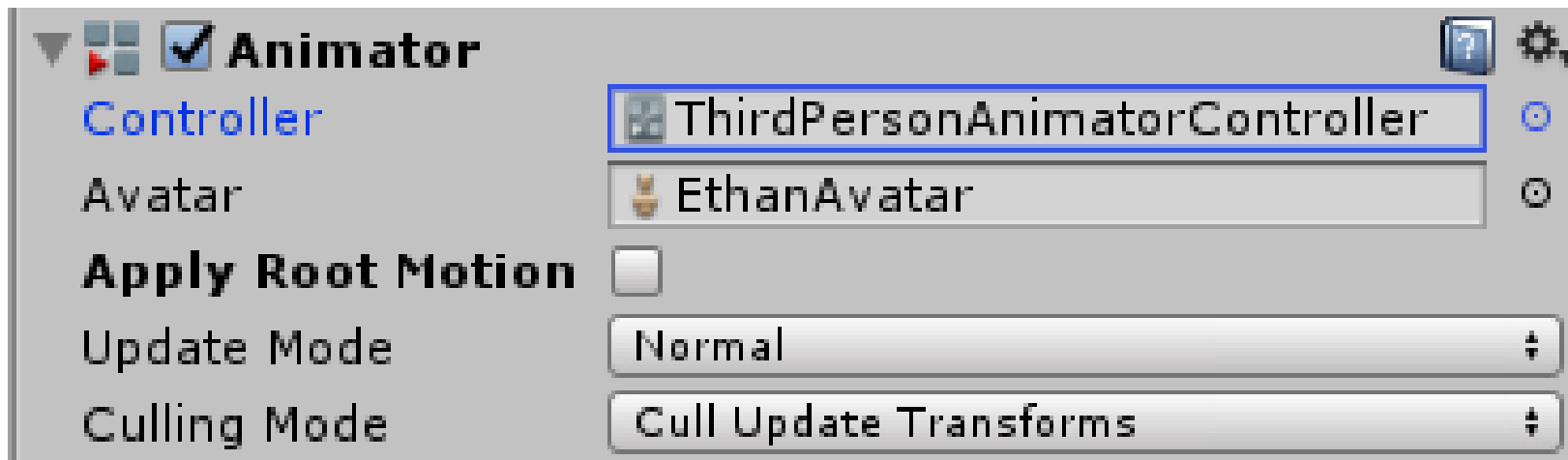
v animátore avatara

Pustíme hru, beží na mieste.

Znovu zapneme Apply Root Motion

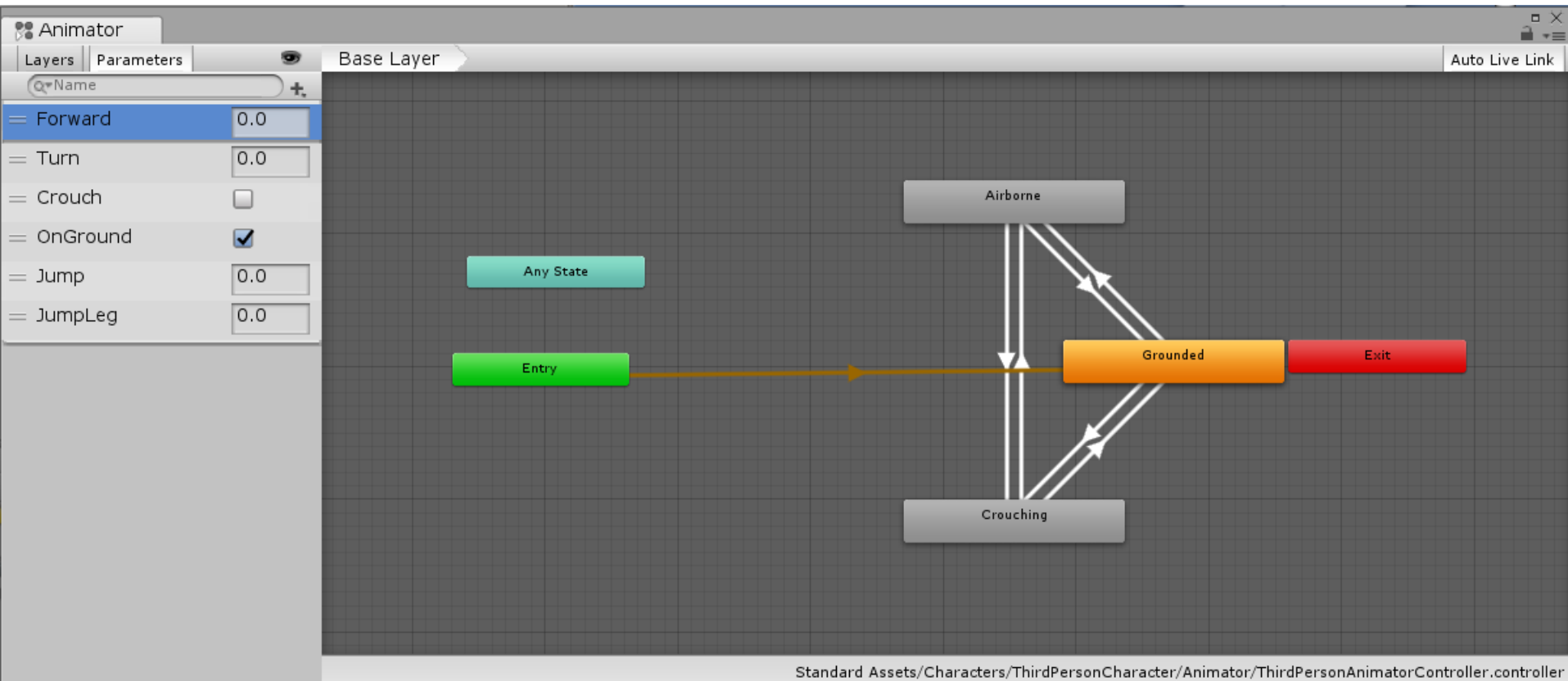
Použitie hotového animator controllera

Nastavme teraz hotový animator



A dvojklikom naň v projekt window si v Animator windows pozrime tento controller

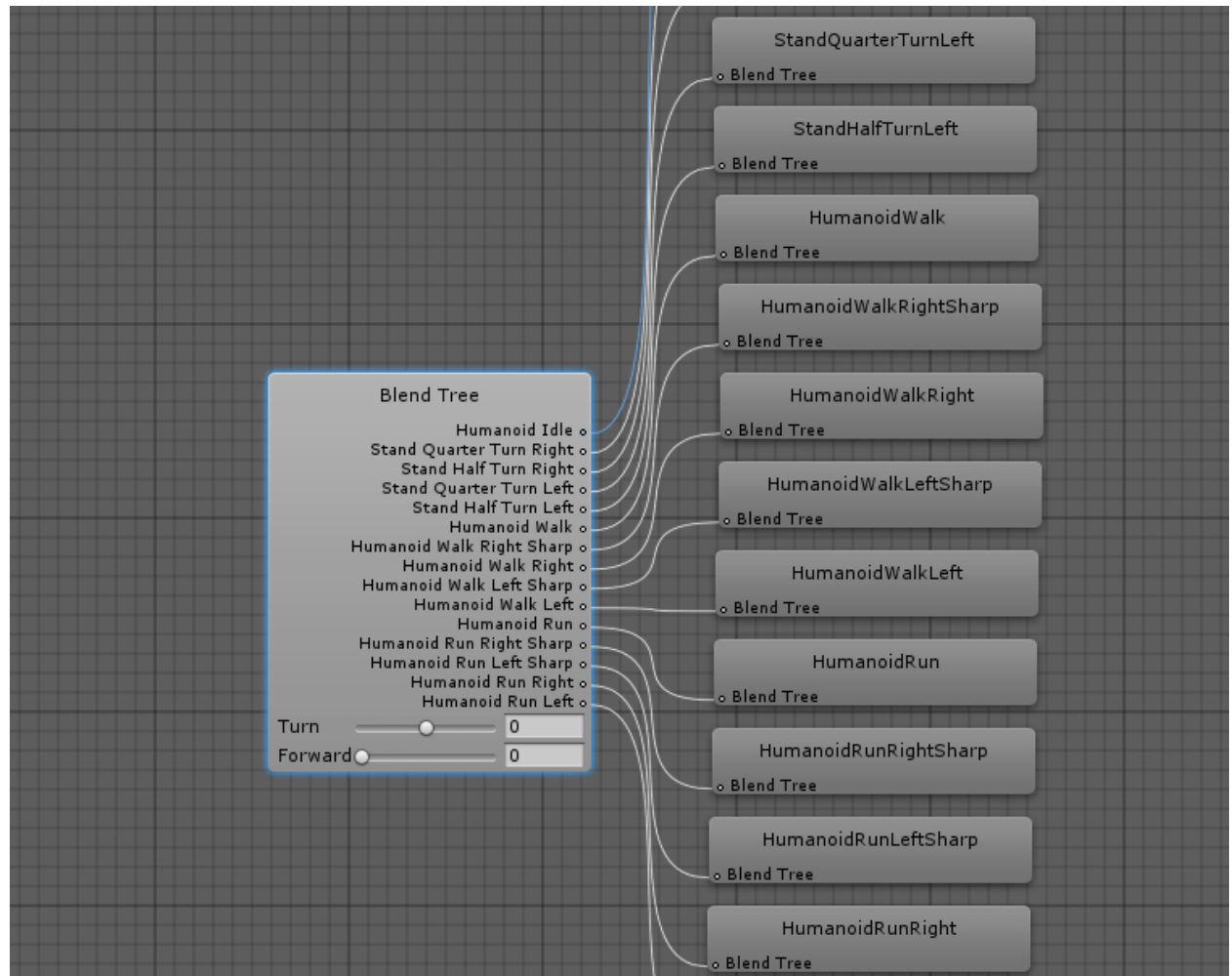
Base Layer



Použite Alt + myš na manipuláciu so stavovým diagramom
Na základnej úrovni sú tri stavy: Grounded, Airborne a Crouching

Blend tree

Vidíme, že je každý stav je založený na tzv. Blend tree, ktorý umožňuje namiešať výsledný pohyb z viacerých animácií, podľa zadaných parametrov



2D Freeform Cartesian

Blend Tree

- Humanoid Idle
- Stand Quarter Turn Right
- Stand Half Turn Right
- Stand Quarter Turn Left
- Stand Half Turn Left
- Humanoid Walk
- Humanoid Walk Right Sharp
- Humanoid Walk Right
- Humanoid Walk Left Sharp
- Humanoid Walk Left
- Humanoid Run
- Humanoid Run Right Sharp
- Humanoid Run Left Sharp
- Humanoid Run Right
- Humanoid Run Left

Turn: 0

Forward: 0

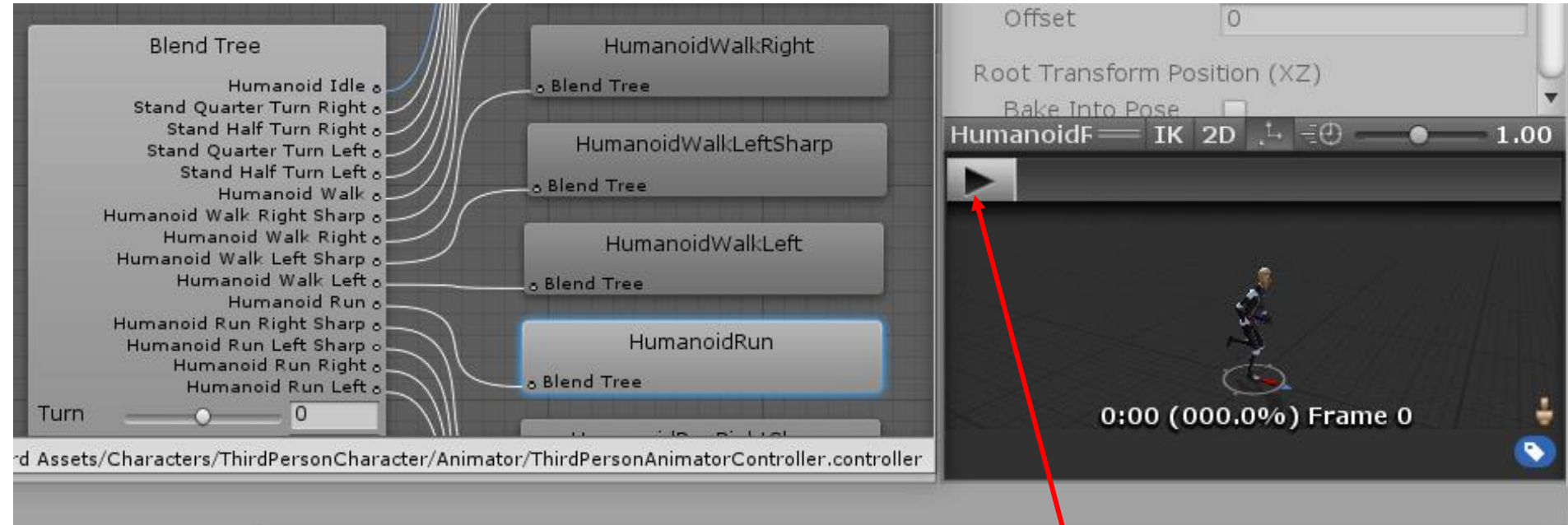
Parameters

Turn Forward

Motion	Pos X	Pos Y		
HumanoidIdle	0	0	1	<input type="checkbox"/>
StandQuarterTurnR	0.5	0	1	<input type="checkbox"/>
StandHalfTurnRight	1	0	1	<input type="checkbox"/>
StandQuarterTurnL	-0.5	0	1	<input type="checkbox"/>
StandHalfTurnLeft	-1	0	1	<input type="checkbox"/>
HumanoidWalk	0	0.5	1	<input type="checkbox"/>
HumanoidWalkRight	1	0.5	1	<input type="checkbox"/>
HumanoidWalkRight	0.5	0.5	1	<input type="checkbox"/>
HumanoidWalkLeftS	-1	0.5	1	<input type="checkbox"/>
HumanoidWalkLeft	-0.5	0.5	1	<input type="checkbox"/>
HumanoidRun	0	1	1	<input type="checkbox"/>
HumanoidRunRights	1	1	1	<input type="checkbox"/>

↑ Turn ↑ Forward

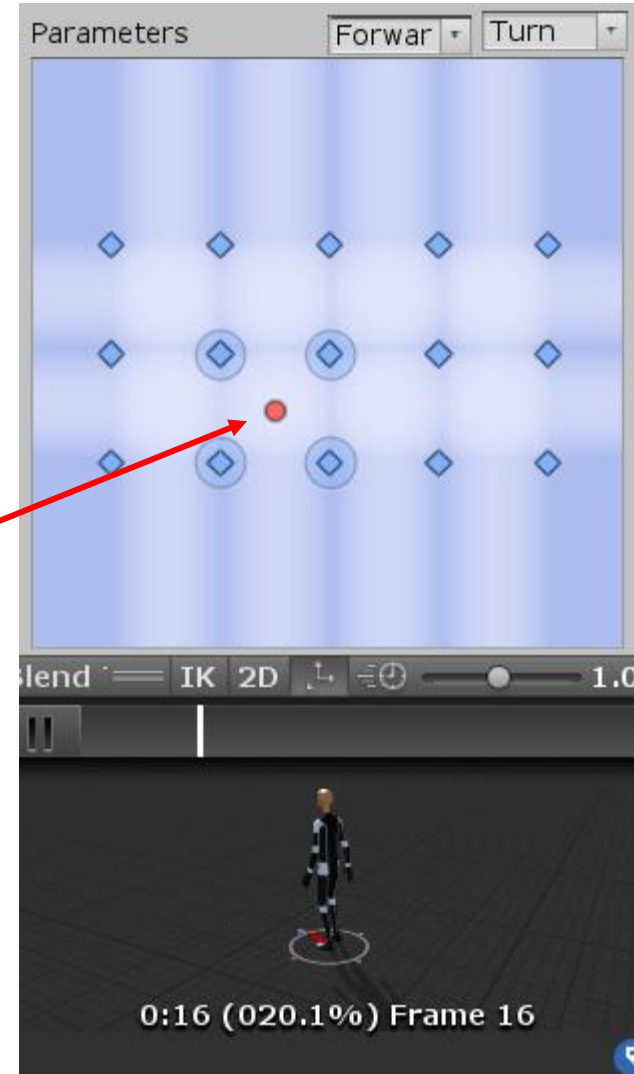
Blend Tree prelína animácie podľa nastavenia Forward and Turn



Prehratie animácie

Blend Tree prelína animácie podľa nastavenia Forward and Turn

**Prehratie animácie
Podľa parametrov**



Úloha 1

Pridajte do blend tree v stave Grounded ďalší motion field, ktorý bude utekať 2x rýchlejšie ako obyčajné utekanie.

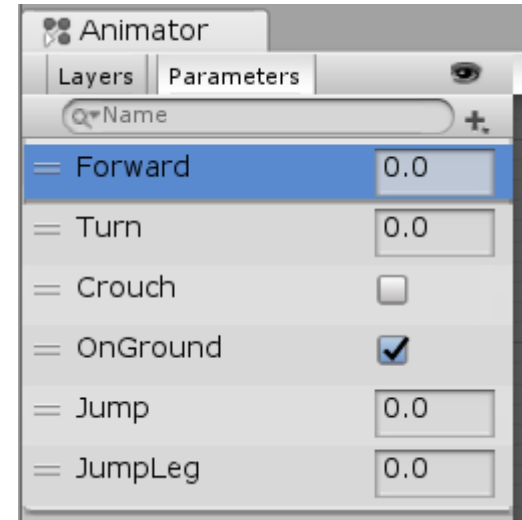
Na ten účel musíte do blend tree pridať ďalšiu položku ktorá púšťa animáciu Run s dvojnásobnou rýchlosťou a zodpovedá hodnote Forward 2.0 a Turn 0.0

Ovládanie zo skriptu

```
private Animator anim;
```

```
void Start () {  
    anim = GetComponent<Animator>();  
}
```

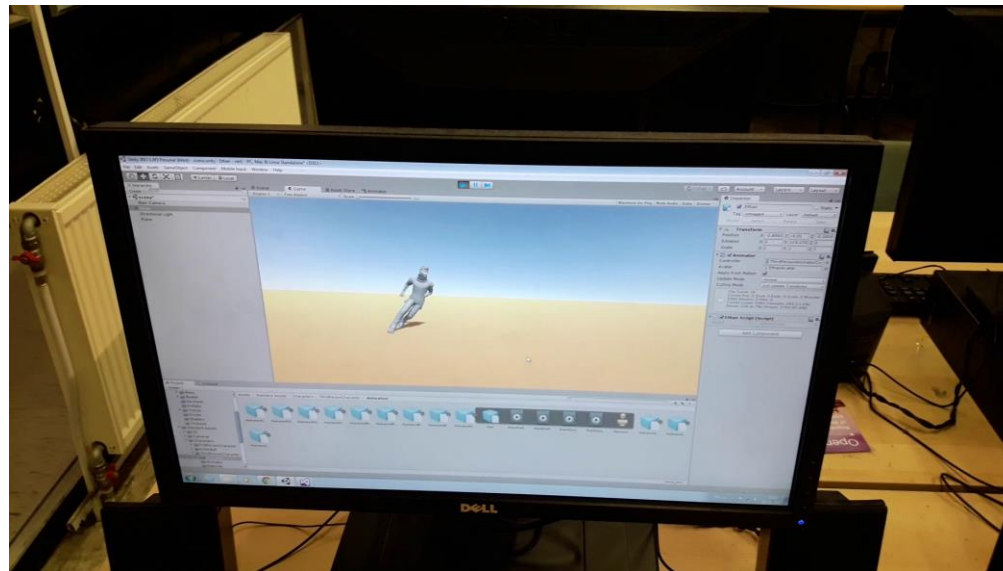
```
void Update () {  
    if (Input.GetKey(KeyCode.RightArrow))  
    {  
        anim.SetFloat("Forward", 1);  
    }  
    if (Input.GetKey(KeyCode.LeftArrow))  
    {  
        anim.SetFloat("Forward", 0);  
    }  
}
```



Úloha 2

Skúsme teraz vytvoriť ovládanie avatara z klávesnice, aby bežal, alebo kráčal, otáčal sa podľa našej vôle vyjadrenej stláčaním klávesových skratiek.

Využite priložený script `RunOrStay.cs`



Triggre (Spúšte)

Pokiaľ ste vyskúšali aj Jump, tak prechod k skákaniu je riadený cez OnGround. Ak ho však len vypneme, avatar ostane vyskočený vo vzduchu. Pridajme radšej k parametrom controllera trigger Hop a zmeňme transition z Grounded na Airborne (len) na tento trigger.

**V skripte potom skok vyvoláme:
(Aby neskákal viac krát, trigger treba resetovať)**

```
if (Input.GetKey(KeyCode.UpArrow))  
{  
    anim.SetTrigger("Hop");  
}
```

Stavy

Zo skriptu môžeme aj priamo manipulovať stav animátora, bez použitia ovládacích parametrov

Aký máme stav vieme testovať takto (musíme dať pozor na layer stavu):

```
AnimatorStateInfo stateInfo = anim.GetCurrentAnimatorStateInfo(0);  
if (stateInfo.IsName("Base Layer.Airborne"))  
{  
    anim.ResetTrigger("Hop");  
}
```

Stav môžeme aj priamo nastaviť:

```
anim.Play("Base Layer.Airborne");
```

Viaceré úrovne

Tieto správania sú na úrovni Base Layer

Úrovní môže byť viac, napr. triašku môžeme vložiť do nami definovanej podradenej úrovne a bude sprevádzať všetky pohyby na základnej úrovni.

Úloha 3

**Skúste vytvoriť v SimpleControlleri vlastný Blend Tree -
Create from new Blend Tree**

**Parametre pridávate v Animator window v záložke
Parameters**

**To, ktorá animácia sa viaže na aký parameter určuje
bodová poloha animácie v konceptuálnom priestore
parametrov (nastaviteľná po jednom a po dvojiciach
parametrov)**