

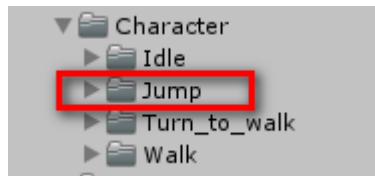
Настав час реалізувати анімацію стрибка. Вона міститиме три внутрішні стани:

Коли персонаж відштовхується від землі

Коли персонаж в польоті над землею

Коли персонаж приземляється

Завантажте в проект додану папку Jump



Налаштовуємо для всіх кадрів

**MeshType**Full rect

**Max size** 512

**Pixelperunit**550

Створюємо наступні анімації для персонажа

StartJump

Jump

EndJump

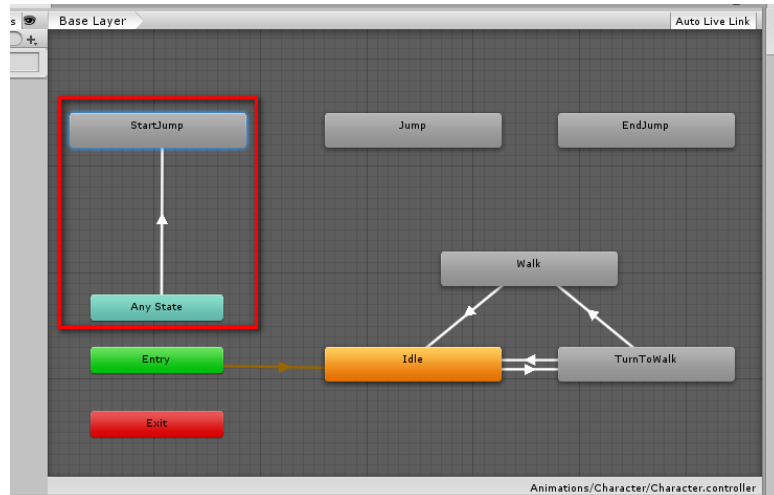
Перетягуємо в Jump один найбільш підходящий кадр анімації стрибка.



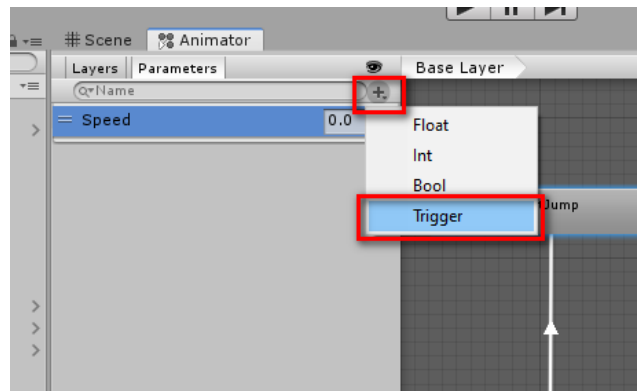
У StartJump і EndJump перетягуємо кадри, що залишилися, відповідні їм з анімації стрибка і ставимо по 10 семплів частоту кадрів

Відкриваємо аніматор. Технічно стрибнути можна із будь-якого стану анімації персонажа. Тому нам знадобиться AnyState

Створюємо перехід з AnyState в StartJump



Створюємо параметр тригер StartJump

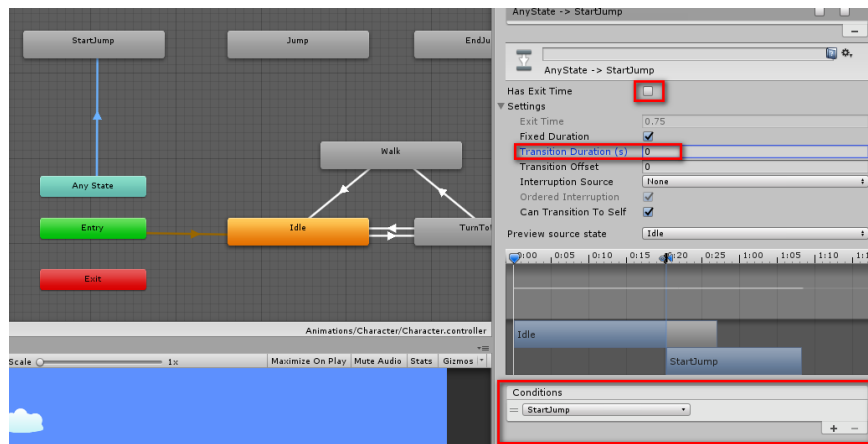


Додаємо його в умову переходу від AnyState до StartJump (За цим переходом потрібно натиснути ЛКМ). Виставляємо:

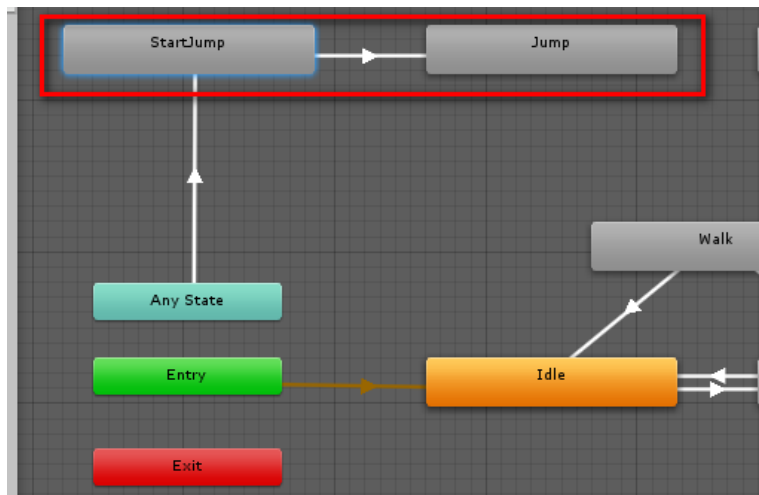
Transition Duration:0

Знімаємо прапорець у Has exit time



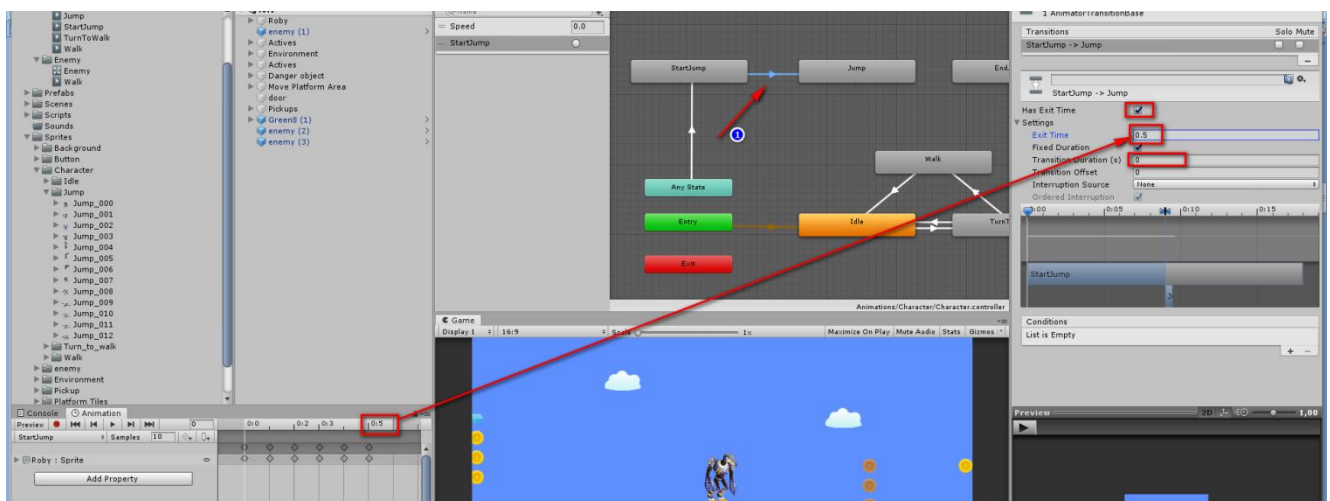


Створюємо перехід із StartJump до Jump

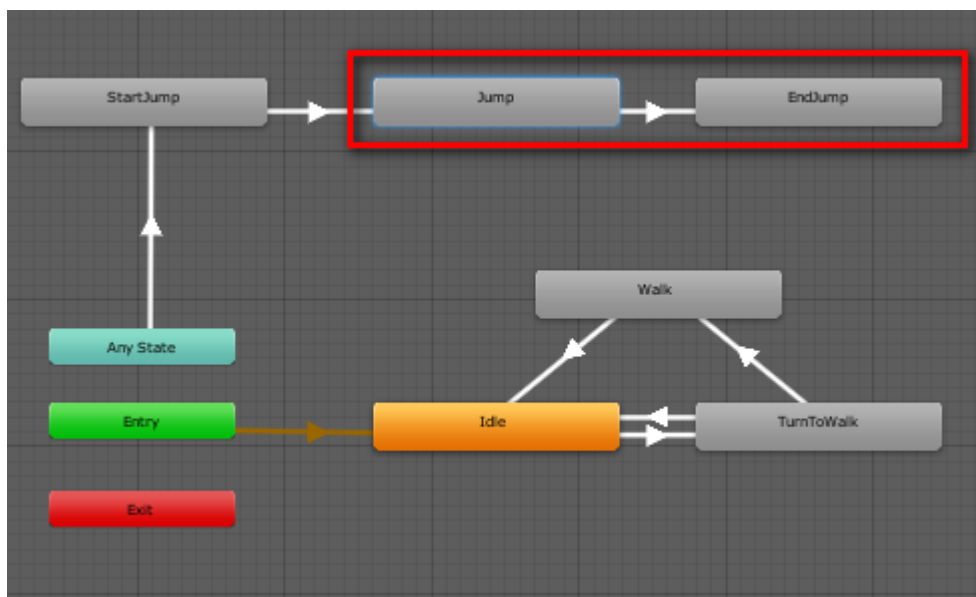


Встановлюємо Transition Duration: 0

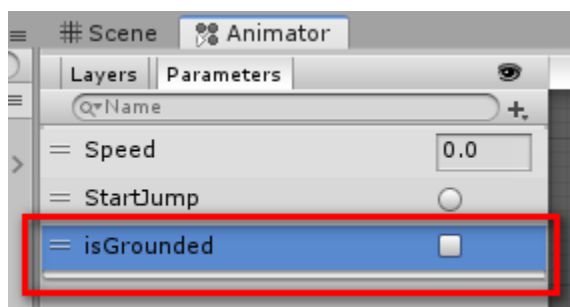
Ставимо прапорець у Has ExitTime. Значення його дорівнює кількості кадрів на анімацію Start Jump:



Створюємо перехід з Jump до EndJump



Створюємо параметр типу bool під назвою isGrounded



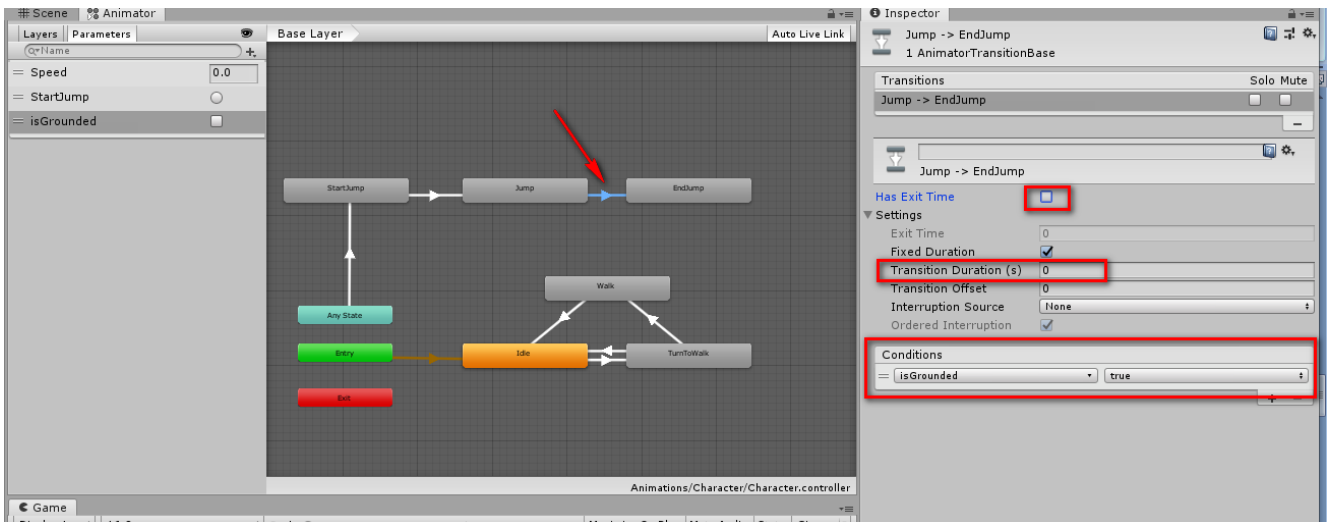
Виділяємо перехід між Jump та EndJump та налаштовуємо

Виконання за створеним параметром isGrounded

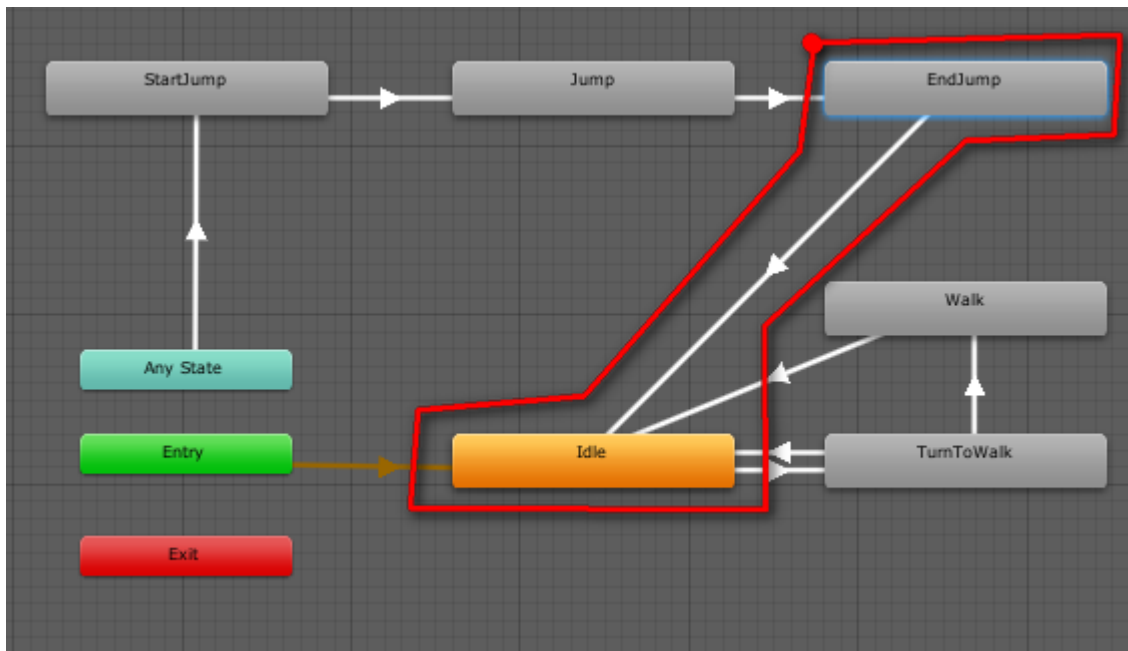
Transition Duration: 0

Знімаємо прапорець у Has exit time





Створюємо перехід з EndJump до Idle



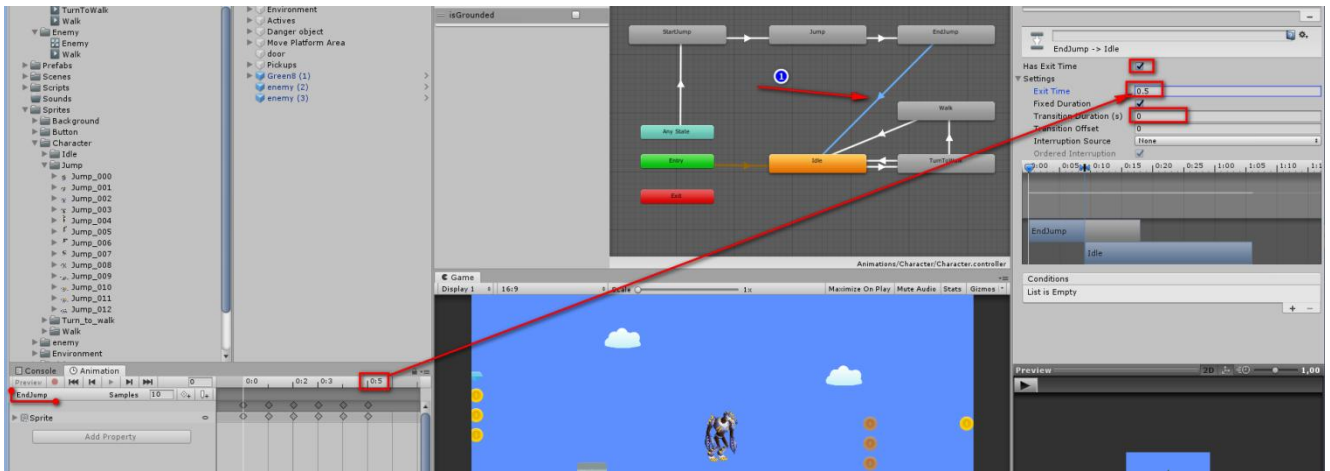
Виділяємо перехід між EndJump та Idle та налаштовуємо:

Transition Duration: 0

Ставимо прапорець у Has exit time

ExitTime = тривалість кадрів анімації EndJump на таймлайні





Тепер налаштуємо запуск анімацій у скрипті. Відкриваємо скрипт Player і на початку методу FixedUpdate записуємо:

```
Unity Message | 0 references
void FixedUpdate()
{
    animator.SetBool("isGrounded", groundDetection.isGrounded); //передаєм текущий признак соприкосновения с землей аниматору
                                                                //Если после прыжка вернулись на землю, это включит анимацию приземления

    direction = Vector3.zero; //Это нужно, т.к. мы не обрабатываем отжатие кнопки
                             //Без этого игрок будет перемещаться на огромные расстояния
                             //(0,0)
}
```

В умову стрибка додаємо

```
//Прыжок на space
if (Input.GetKeyDown(KeyCode.Space) && groundDetection.isGrounded)
{
    rb2d.AddForce(Vector2.up * force, ForceMode2D.Impulse);
    animator.SetTrigger("StartJump"); //Передаем в аниматор сигнал о запуске анимации прыжка
}
```

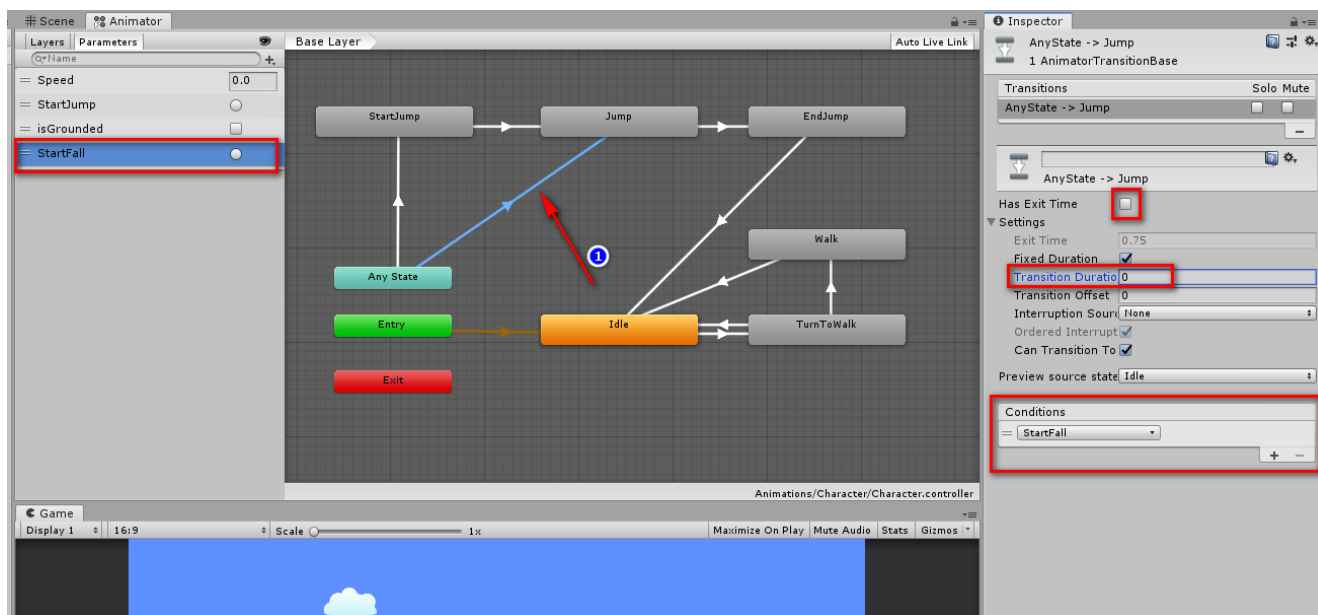
Якщо персонаж падає з платформи на іншу платформу, а не стрибає, анімація не програтиться. Виправимо це!

Створюємо перехід з AnyState в Jump. У перехід між ними створюємо та додаємо умовою тригер StartFall

Знімаємо прапорець Has Exit Time

Transition Duration = 0





Додаємо до скрипту Player ще одну змінну

```
private bool isJumping; //Данная переменная нужна, чтобы сохранять признак прыжка персонажа
                        //Если персонаж не прыгает, но находится в воздухе, значит он падает
```

В умову стрибка додаємо

```
//Прыжок на space
if (Input.GetKeyDown(KeyCode.Space) && groundDetection.isGrounded)
{
    rb2d.AddForce(Vector2.up * force, ForceMode2D.Impulse);
    animator.SetTrigger("StartJump"); //Передаем в аниматор сигнал о запуске анимации прыжка
    isJumping = true; //Сохраняем признак того, что персонаж прыгнул для аниматора
}
```

На початку методу залишилося перевірити. Якщо ця змінна залишилася false, то стрибка не було. Якщо в цьому випадку зберігається теж false, то ми знаходимося в повітрі. Відповідно падаємо

```
void FixedUpdate()
{
    animator.SetBool("isGrounded", groundDetection.isGrounded); //передаем текущий признак соприкосновения с землей аниматору
                                                                //Если после прыжка вернулись на землю, это включает анимацию приземления

    //Если мы не прыгаем, но находимся в воздухе - значит мы падаем
    if (!isJumping && !groundDetection.isGrounded)
    {
        animator.SetTrigger("StartFall");
    }
}
```



Можна запустити та перевірити результат. Бачимо, що падіння спрацьовує лише один раз. Так відбувається, тому що ми не повертаємо значення змінної `isJumping` назад в `false`. Дописуємо наступне:

```

Unity Message | 0 references
void FixedUpdate()
{
    animator.SetBool("isGrounded", groundDetection.isGrounded); //Передаем текущий признак соприкосновения с землей аниматору
                                                                    //Если после прыжка вернулись на землю, это включит анимацию приземления

    //Если мы не прыгаем, но находимся в воздухе - значит мы падаем
    if (!isJumping && !groundDetection.isGrounded)
    {
        animator.SetTrigger("StartFall");
    }

    isJumping = groundDetection.isGrounded; //Сработает на следующем кадре после запуска прыжка
                                            //Нужно для того, чтобы вернуть признак прыжка в false после проигрывания анимации
}
    
```

Тепер падіння працює правильно. Але якщо звернути увагу, анімація `StartJump` працює дуже швидко. Відчувається швидкий перехід із неї в стан `Jump`. І `StartJump` мало видно.

Розглянемо, як працює стрибок поетапно

1) Відштовхнулися від землі - аніматор прийняв `SetTrigger (StartJump)`

```

//Прыжок на space
if (Input.GetKeyDown(KeyCode.Space) && groundDetection.isGrounded)
{
    rb2d.AddForce(Vector2.up * force, ForceMode2D.Impulse);
    animator.SetTrigger("StartJump"); //Передаем в аниматор сигнал о запуске анимации прыжка
    isJumping = true; //Сохраняем признак того, что персонаж прыгнул для аниматора
}
    
```

2) Ознака стрибка `isJumping` став дорівнює `true`

```

//Прыжок на space
if (Input.GetKeyDown(KeyCode.Space) && groundDetection.isGrounded)
{
    rb2d.AddForce(Vector2.up * force, ForceMode2D.Impulse);
    animator.SetTrigger("StartJump"); //Передаем в аниматор сигнал о запуске анимации прыжка
    isJumping = true; //Сохраняем признак того, что персонаж прыгнул для аниматора
}
    
```

3) Персонаж перебуває у стані стрибка

4) У наступному кадрі після спрацювання умови стрибка аніматору передається поточний стан торкання землі

```

Unity Message | 0 references
void FixedUpdate()
{
    animator.SetBool("isGrounded", groundDetection.isGrounded); //Передаем текущий признак соприкосновения с землей аниматору
                                                                    //Если после прыжка вернулись на землю, это включит анимацию приземления

    //Если мы не прыгаем, но находимся в воздухе - значит мы падаем
    if (!isJumping && !groundDetection.isGrounded)
    {
        animator.SetTrigger("StartFall");
    }
}
    
```





5) Так як землі ми поки що не торкаємося, isGrounded зараз дорівнює false і анімація падіння не запускається

6) Далі йде перевірка, що ми не здійснили стрибок

```

animator.SetBool("isGrounded", groundDetection.isGrounded); //Передаем текущий признак соприкосновения с землей аниматору
//Если после прыжка вернулись на землю, это включит анимацию приземления

//Если мы не прыгаем, но находимся в воздухе - значит мы падаем
if (!isJumping && !groundDetection.isGrounded)
{
    animator.SetTrigger("StartFall");
}
    
```

7) Так як насправді ми стрибнули і isJumping зараз дорівнює true, то дана перевірка проходить не успішно і дії за умови не виконуються.

8) Далі isJumping змінює своє значення на false, т.к. їй присвоюється значення змінної isGrounded, а ми досі не доторкнулися до землі

```

//Если мы не прыгаем, но находимся в воздухе - значит мы падаем
if (!isJumping && !groundDetection.isGrounded)
{
    animator.SetTrigger("StartFall");
}

isJumping = groundDetection.isGrounded; //Сработает на следующем кадре после запуска прыжка
//Нужно для того, чтобы вернуть признак прыжка в false после проигрывания анимации
    
```

9) На наступному кадрі знову запускається метод fixed update і на цей раз у перевірці isJumping вже false і ми все ще в повітрі, відповідно умова виконується і спрацьовує тригер StartFall, який запускає перехід у стан стрибка

```

@ Unity Message | 0 references
void FixedUpdate()
{
    animator.SetBool("isGrounded", groundDetection.isGrounded); //Передаем текущий признак соприкосновения с землей аниматору
    //Если после прыжка вернулись на землю, это включит анимацию приземления

    //Если мы не прыгаем, но находимся в воздухе - значит мы падаем
    if (!isJumping && !groundDetection.isGrounded)
    {
        animator.SetTrigger("StartFall");
    }
}
    
```

10) Таким чином, виходить, що стан StartJump програється всього 2 кадри, після чого насильно переводиться в стан Jump

Для вирішення цієї проблеми змінюємо присвоєння isJumping на наступне

```

//Если мы не прыгаем, но находимся в воздухе - значит мы падаем
if (!isJumping && !groundDetection.isGrounded)
{
    animator.SetTrigger("StartFall");
}

//isJumping - если мы сейчас находимся в прыжке
//!groundDetection.isGrounded - если мы еще не прикоснулись к земле
//То в isJump остается true которое не позволяет запуститься триггеру StartFall в условии выше
isJumping = isJumping && !groundDetection.isGrounded;
    
```



Тепер `isJumping` стане `false` тільки після приземлення і анімація `StartJump` буде програватися із запланованою швидкістю. Якщо ж ми не натискали пропуск для стрибка, то `isJumping` залишається `false` і ми потрапляємо в конструкцію стрибка за допомогою тригера `StartFall` минаючи стан `StartJump`

