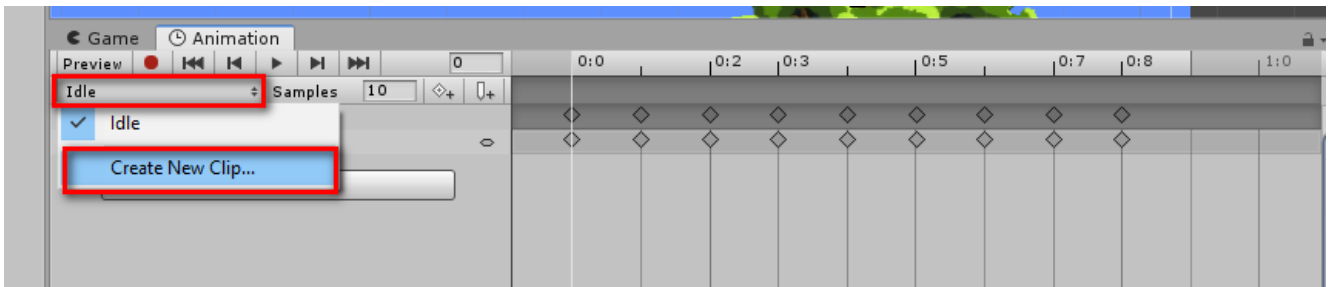
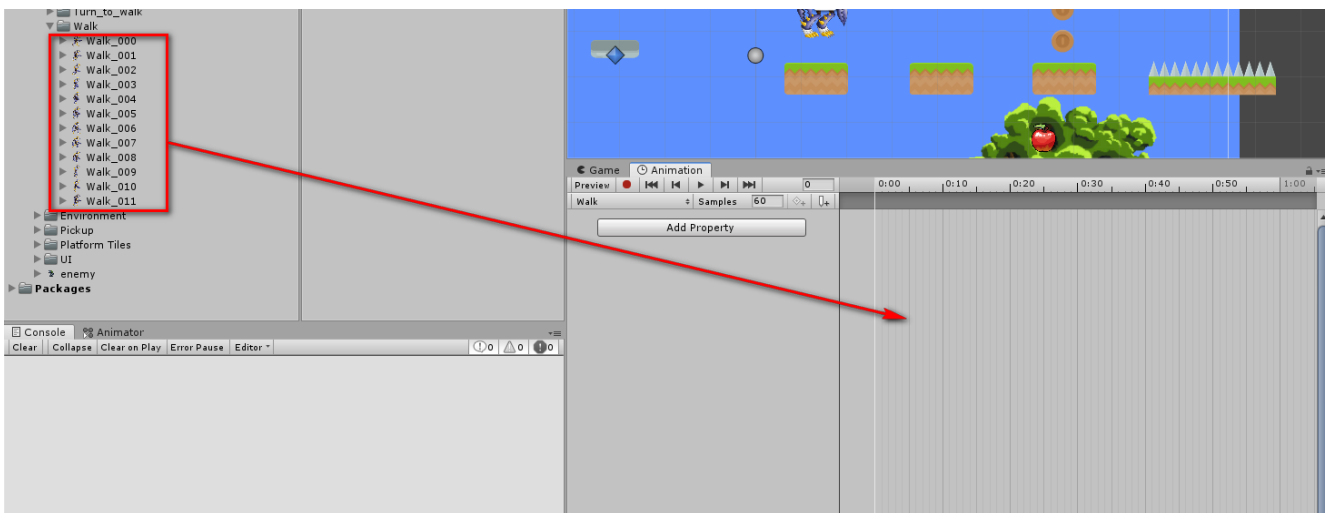


Клацаємо по назві поточної анімації на панелі Animation і створюємо ще одну анімацію:

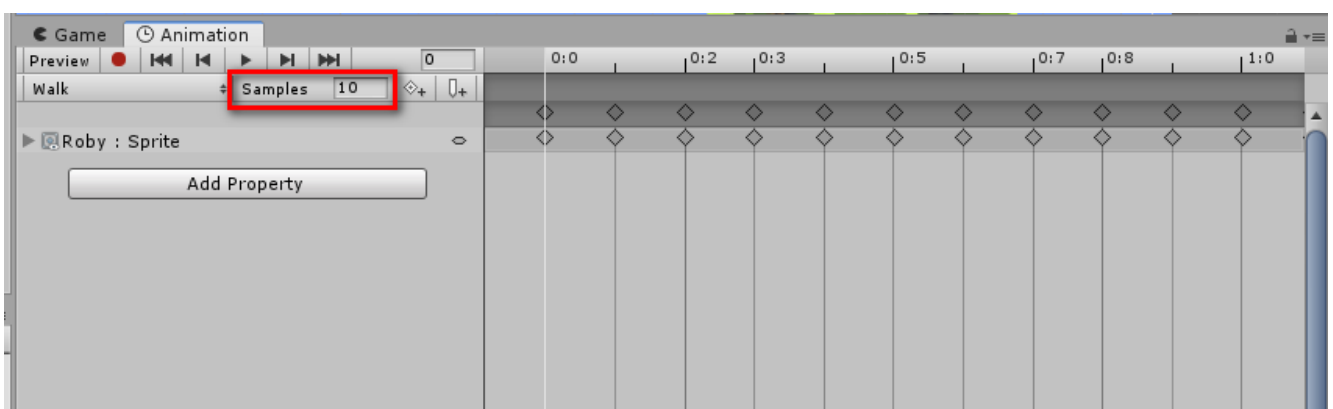


Називаємо її Walk

Перетягуємо в панель анімації всі спрайти з папки Walk

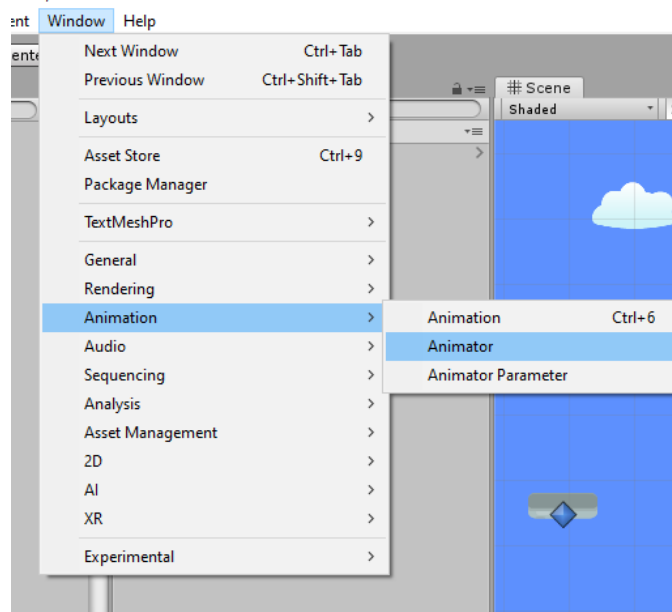


І встановлюємо швидкість 10 семплів!

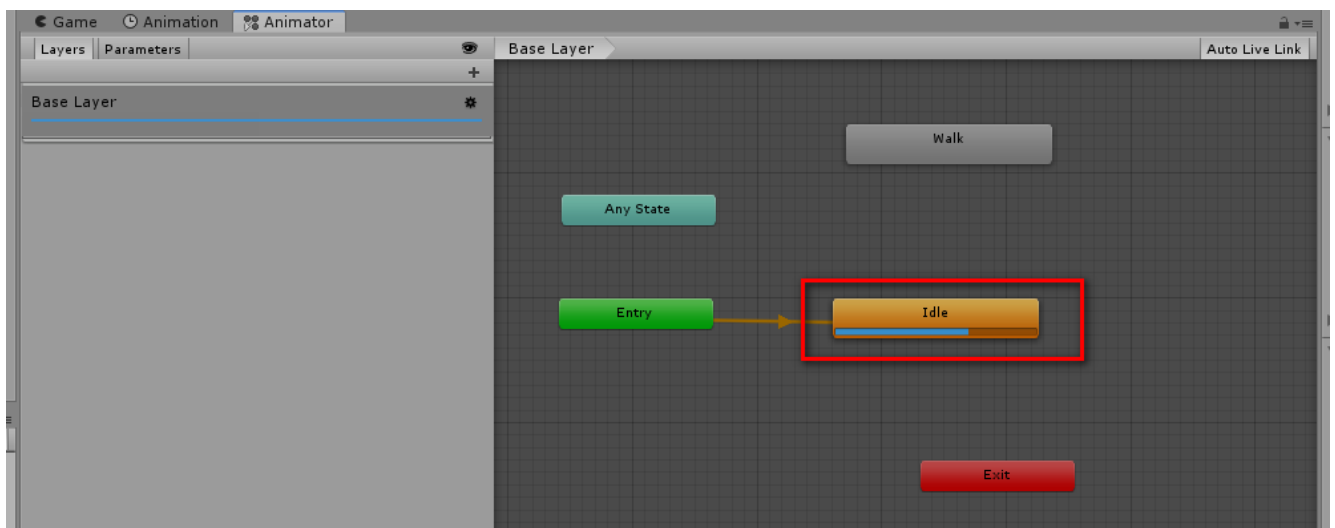


Тепер настав час налаштувати запуск цієї анімації. Виділіть робота та відкрийте панель аніматора:





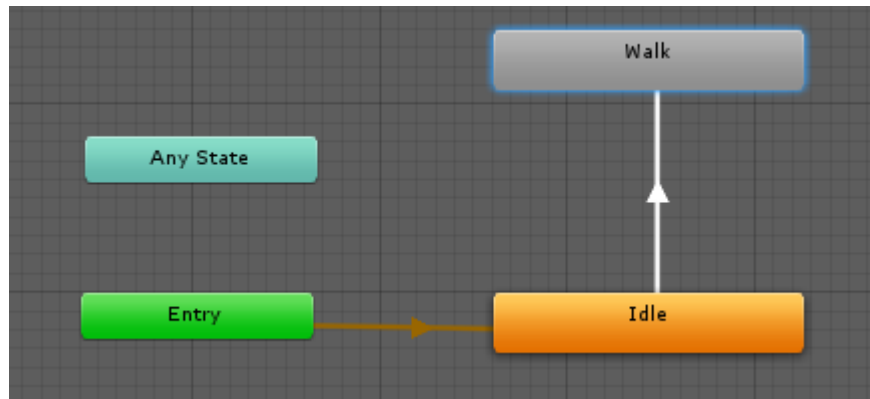
На даний момент при запуску у нас постійно працює анімація Idle



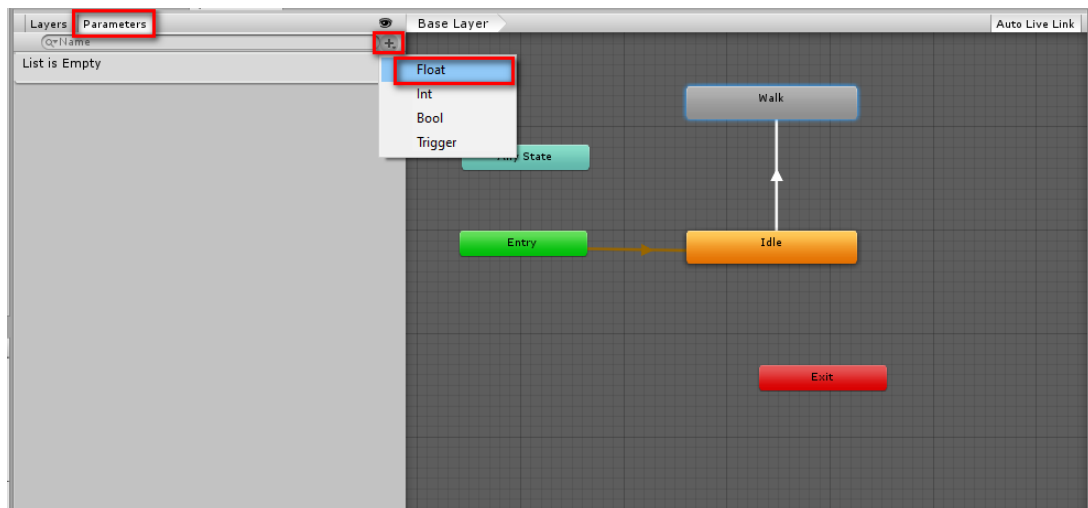
Клацаємо по цій анімації правою кнопкою миші та вибираємо:



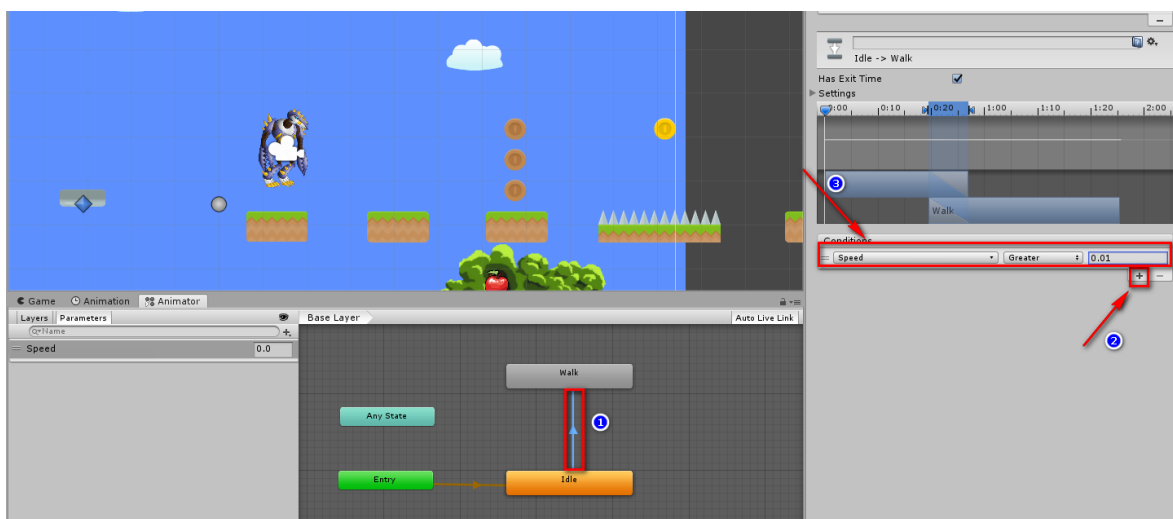
Тягнемо перехід до стану Walk



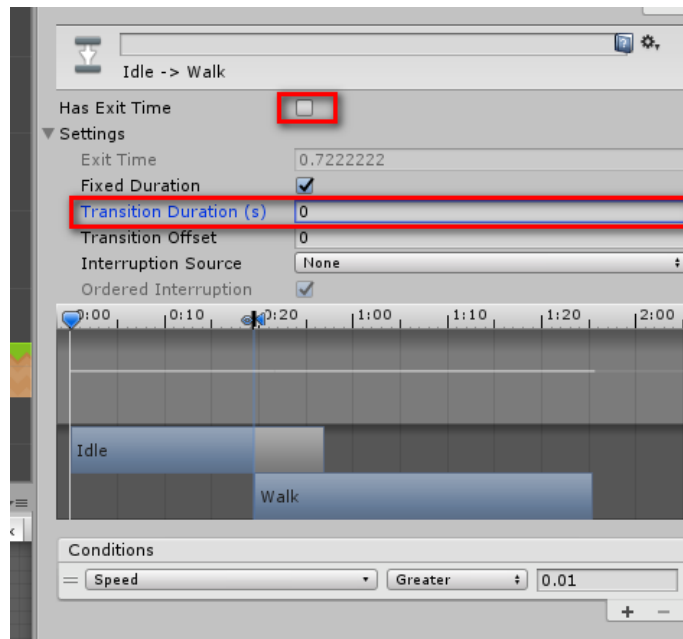
Переходимо на вкладку параметрів і створюємо один тип Float під назвою Speed



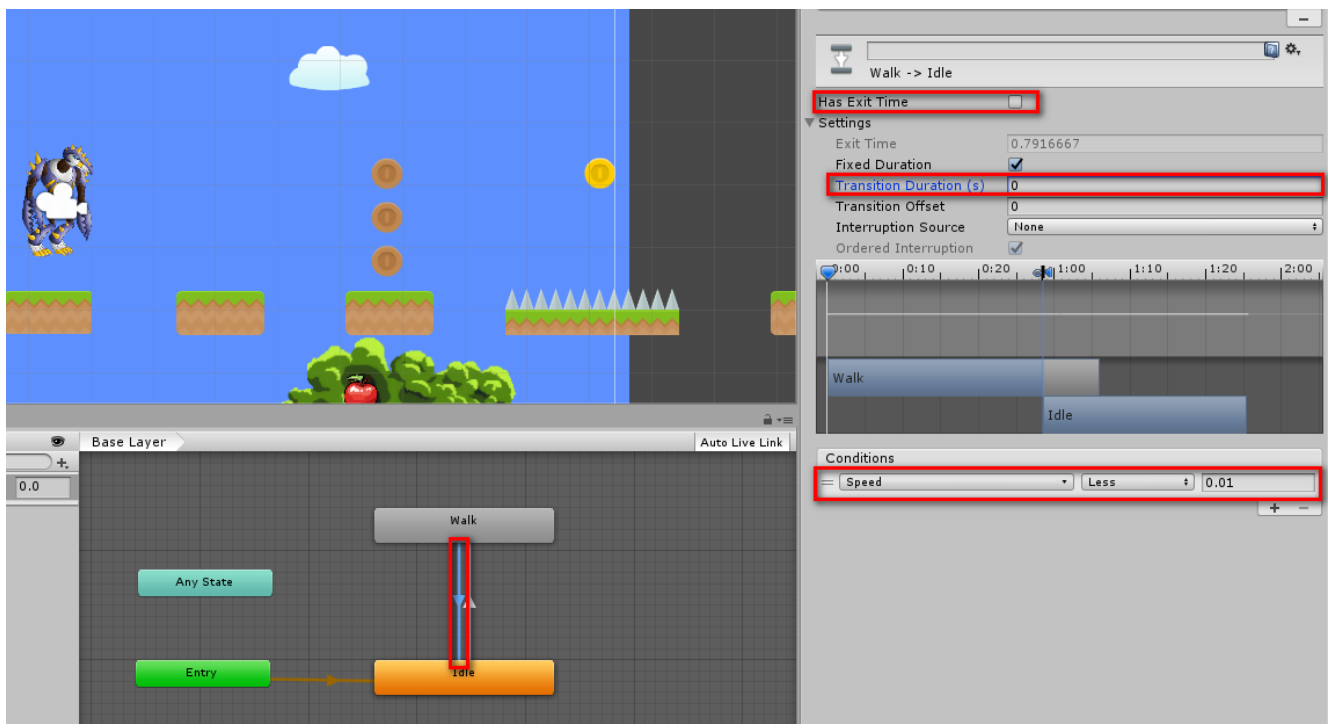
Переконуємося, що перехід виділено та створюємо для нього умову, що у разі перевищення параметром Speed значення 0.01 має запуститися цей перехід



Щоб перехід між анімаціями спрацював миттєво, вказуємо, що не будемо використовувати час виходу і час затримки прирівнюємо нулю



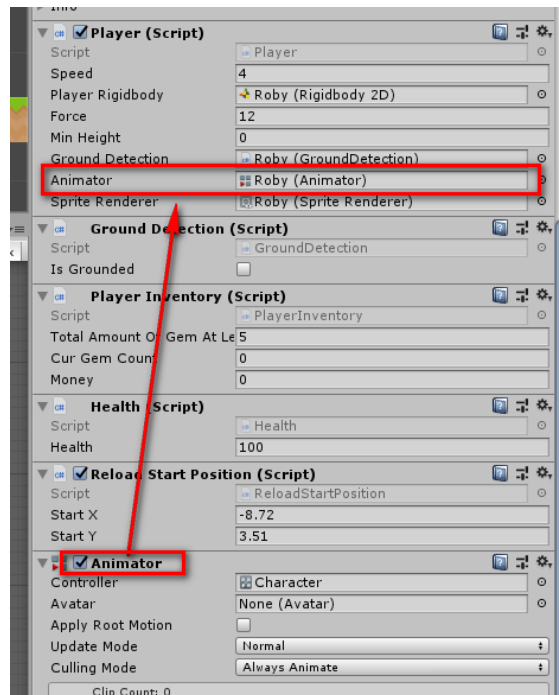
Створюємо перехід у зворотний бік і задаємо йому умову запуску, якщо параметр Speed стає менше 0.01:



Тепер потрібно запрограмувати подачу керуючого сигналу в наш параметр Speed.



У скрипті Player ми передбачили змінну для зв'язку з аніматором гравця. Давайте заповнимо її в інспекторі:



Відкриваємо скрипт Player і в методі FixedUpdate передаємо значення швидкості гравця по модулю в наш параметр:

```
animator.SetFloat("Speed", Mathf.Abs(direction.x));
```

“Speed” – Назва параметра, до якого передається значення

Direction.x – це поточна швидкість робота. Звичайно якщо вона не нульова, значить він рухається. Mathf.Abs дає нам значення швидкості за модулем. Якщо брати швидкість не за модулем, рух у лівій дасть негативне значення і анімація не запуститься!

Запустіть програму. Обидві анімації мають працювати!

