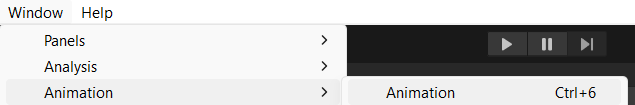
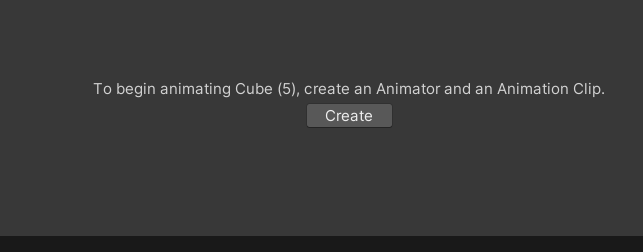
**Animatioin**

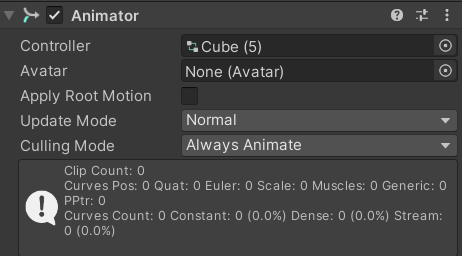
Нужно новое окно для создания анимации



Для того что создать анимационный клип, нам необходимо моздать папку, с выбранным объектом нажимаем Create



У объекта появляется новый компонент



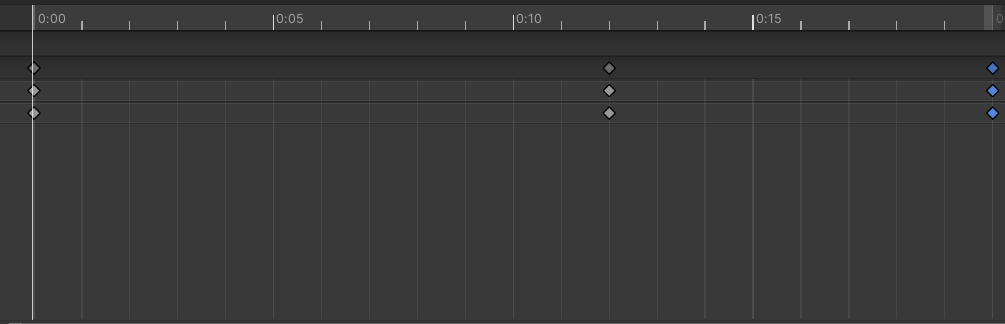
Самый быстрый спосоь начать делать анимацию нажать recorde button (он будет записывать все изменения с объектом)



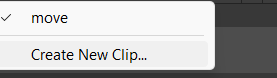
К примеру на 0 кадре я изменю положение куба, unity автоматически проставит keyframe, и когда на 15 кадре сделаю изменения он так же автоматически проставит keyframe.

Так же если выделять keyframe то мы можем их редактировать и unity автоматом заменит на новые из менения.

Мы сожем анимаровать не тлько transfor компонент, но и другие, к примеру Renderer, при включенном recorde button, мы можем отключать и включать Renderer и движок сам запомнит точки анимации, главное указывать кадры



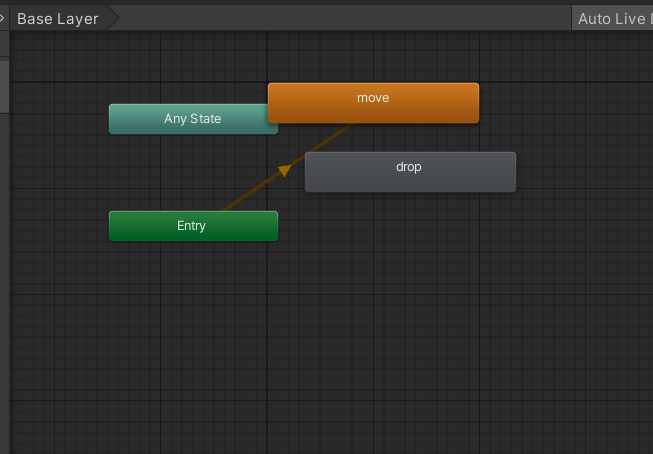
Так же мы можем создать несколько анимации на один объект



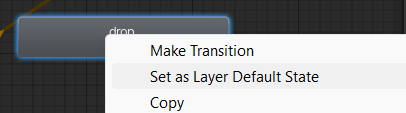
**Using the animator**

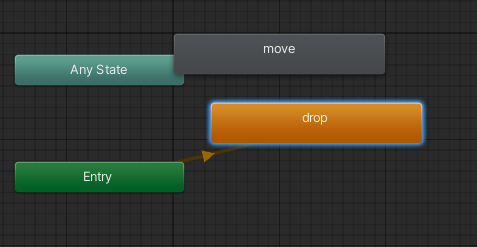
Папка в которой мы создавали анимации ест controller, в него нам нужно зайти



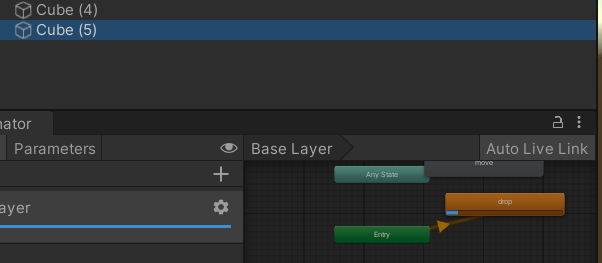


По у молчанию во времы запуска симуляции у нас будет проигрыватся анимация move, так как она была создана первой, но мы можем замменить стартовую анимацию



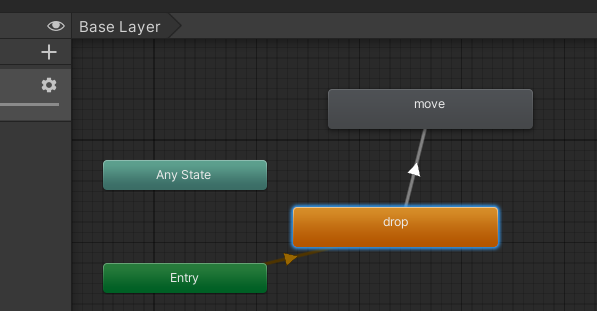


Если во время симуляции мы выберем объект с аниации, то мы можем следить что происходит с самой анимацией

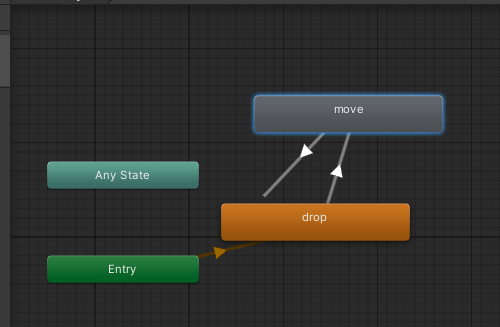


Как делать преход мещду анимациями? Выбираем анимацию и правый клик мыши и выбираем Make transition и направляем стрелку от drop к move анимации

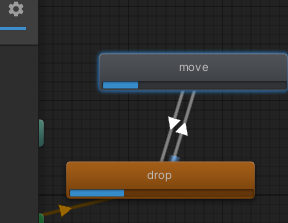




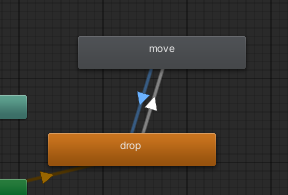
Так же чтоб объект не застревал в анимации move, теперь тоже самое только с анимацией move



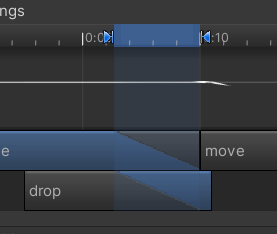
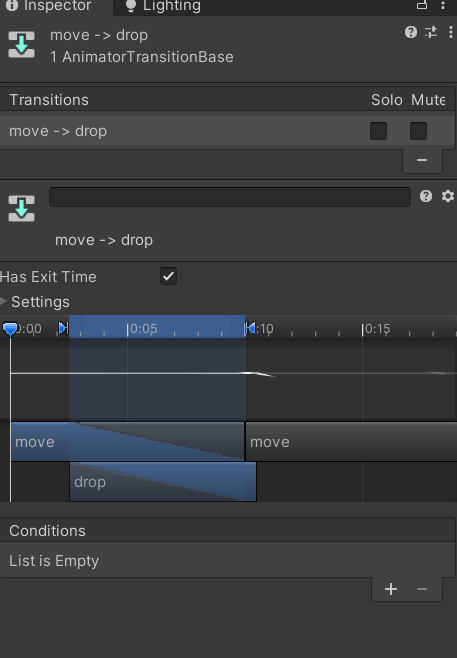
Эти две анимации будут проигрываться друг за другом каждый раз по циклу



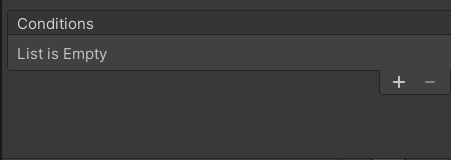
Мы отдельно можем просматривать стрелки перехода анимации



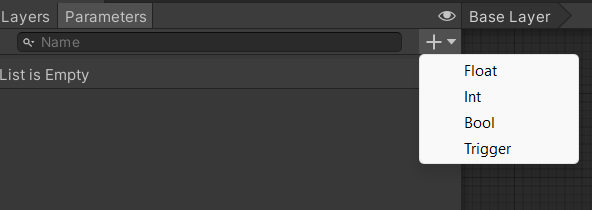
И можем просматривать их параметры, можем изменять, time line



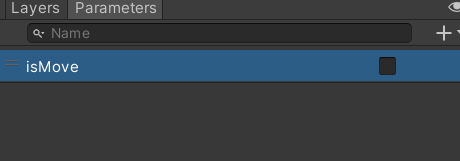
Так же есть условия при которых перехот от анимации к анимации должен случиться, пока условие пустое



У нас есть вкладка Parametres

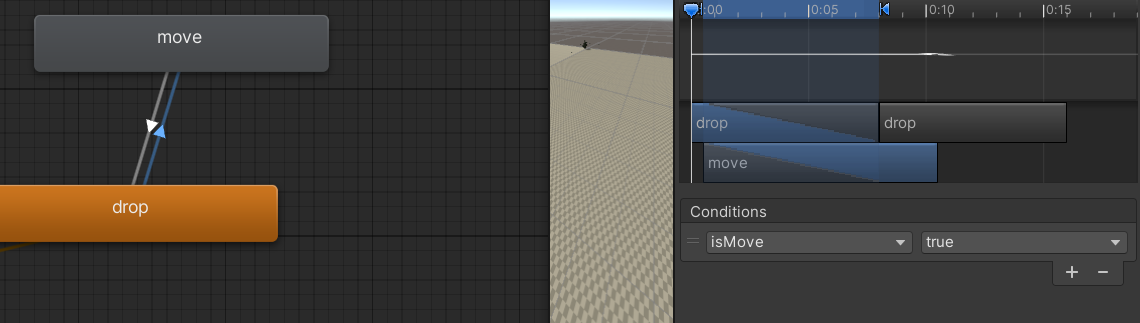


Я бы хотела добавить bool и назвать isMove



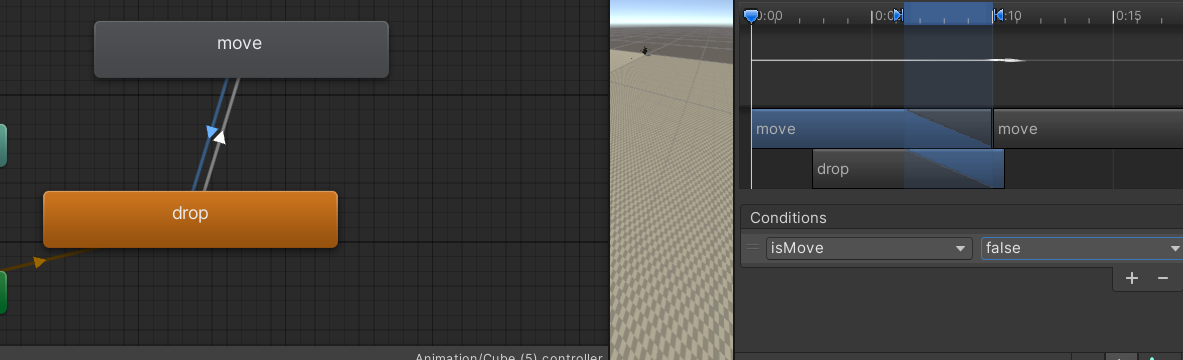
B далее я выделяю переход который идет в анимацию *Move* и добавляю условие (true), при этом условии будет проигрываться анимация Move





Выбираю переход в анимацию drop и выбираю условие которое будет равно *false*



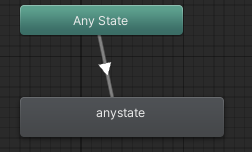


И в симуляции когда я захочу что б анимации менялись я буду изменять параметр





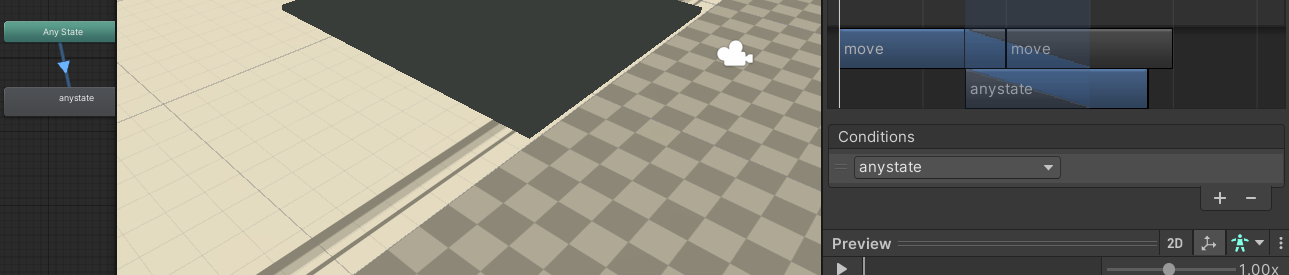
У нас есть состояние которое поможет взывать нам анимацию из любым состояния, для примера я создала анимацию Anystate



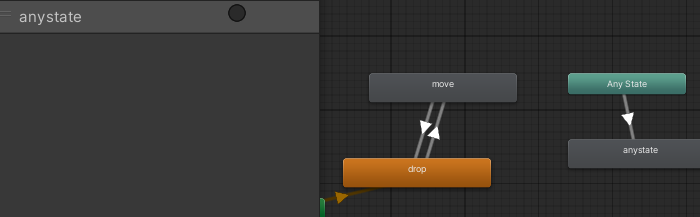
И на переходе (на стрелке) я хочу сделать условие с триггером



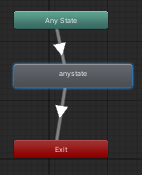




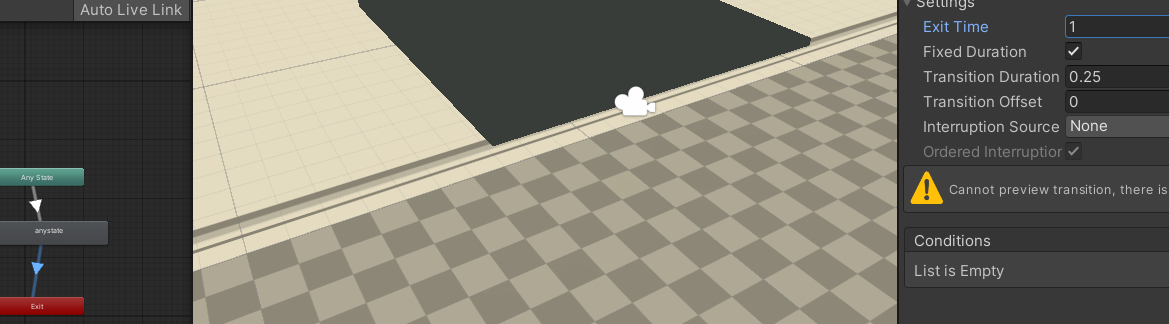
И когда мы будем нажимать на параметр во время симуляции то будет срабатывать анимация Anystate



И что бы объект не застрял в анимации Anystate мы сделаем переход в Exit и тогда объект начнет анимацию с Start



Подправим парамет, что б у нас полностью отработала анимация и потом был из нее выход



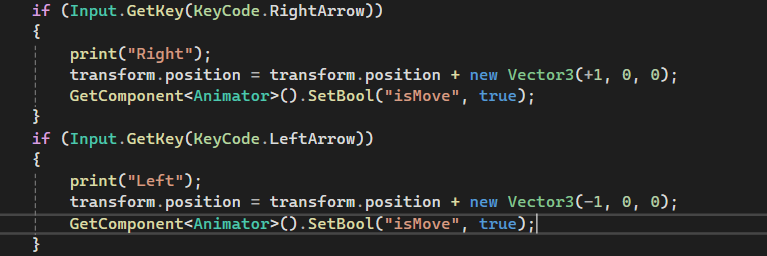
**Training Animation via Code**

Выше мы рассмотрели анимации и условия и переключали их вручную(*parametres*), тут же мы рассмотрим как это сделаь с помощью кода

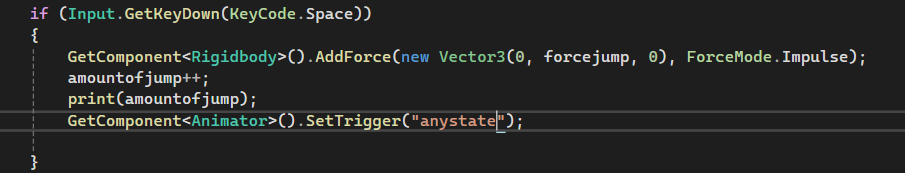


Для этого нам нужно будет получить доступ к *Animator* компоненту, так как он является контролером всех анимации и преходов между ними

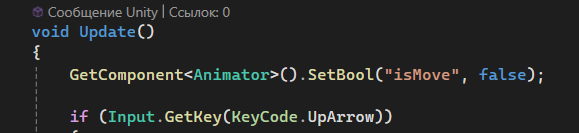
В фаиле *MyComponent* в условия где я получаю нажатие для предвижения right/left я могу получить компонент анимации, для этого в пределах того же объекта на котором скрипт и из которого мы хотим получить этот компонент мы используем GetComponent



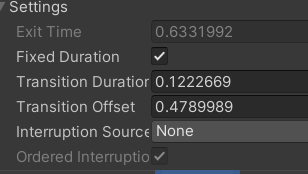
И для клавиши *space* определим анимацию anystate



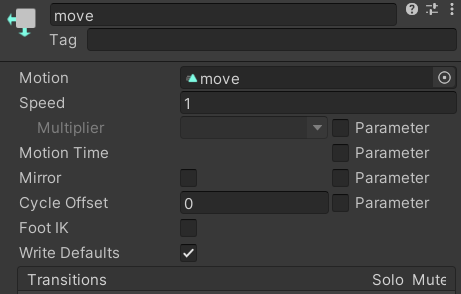
Если мы запустим симуляцию то увиди что даже после того как мы не нажимаем на клавиши, наш персонаж застрянет в цикле движения, и поэтому вначале необходимо прописать



В переходах между анимациями мы можем корректировать настроики

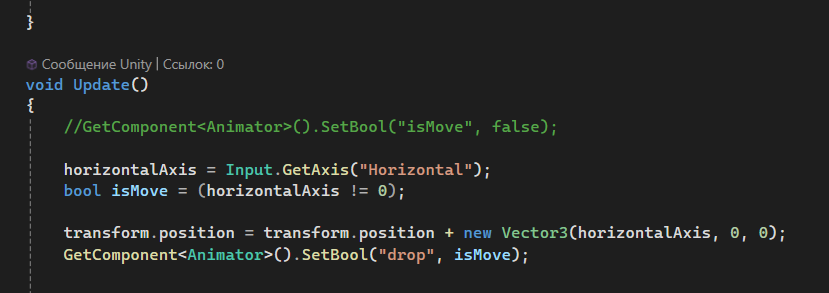


Выбрав анимацию мы можем редактировать и ее настройки

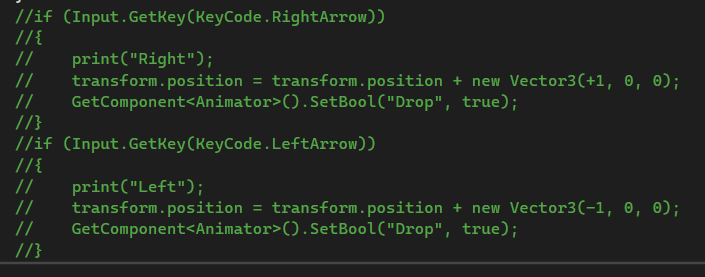


**Addition**

Мы можем упростить код анимации, код ниже позволяет нам заменить множество строк связанных left/right перемещением, horizontal возвращает значение от 1 до -1 (для стрелок на клавиатуре/джостиков), а если мы не нажимаем клавиши (вправо / влево) то он возвращает нуль(0), поэтому мы можем пременную horizontalAxis вставить в Vetor3 и так как это все в функции update нам не нужно условие if, и после зауска симуляции у нас все работает как и до оптипизации кода, если нажимаем вправо или влево то отрабатывает анимация drop (возвращается либо 1 или -1)



Убираем это





С таким написание анимация не будет сглаженна



**Curve animation**

