

# RAGDOLL MECANIM MIXER

Руководство

[ВИДЕО РУКОВОДСТВО](#)

[https://assetstore.altingiran.kz/ramecan-mixer/  
nurzzz27@gmail.com](https://assetstore.altingiran.kz/ramecan-mixer/nurzzz27@gmail.com)

## СОДЕРЖАНИЕ

|  |    |
|--|----|
| ОБНОВЛЕНИЯ .....                                 | 3  |
| ДЕМО СЦЕНА .....                                 | 4  |
| СОЗДАНИЕ RAGDOLL НА ПРИМЕРЕ ПЕРСОНАЖА ETHAN..... | 5  |
| ИЗ STANDARD ASSETS.....                          | 5  |
| РАБОТА С СОСТОЯНИЯМИ В RAMECAN MIXER .....       | 10 |
| ВАЖНЫЕ ЗАМЕТКИ.....                              | 12 |

## ОБНОВЛЕНИЯ

| Версия | Дата       | Изменения   |
|--------|------------|---|
| 1.01   | 28.03.2020 | обновлен пакет PostProcess<br>фикс проблемы с Joint Drive Damper<br>фикс проблемы с сохранением состояний |

## ДЕМО СЦЕНА

### *Анимации*

Так как демо сцена использует анимации из другого ассета, нужно скачать этот пакет анимаций. Asset Store: [RPG Character Mecanim Animation Pack FREE](#) (Спасибо Explosive).

При импорте поставьте галочку только на папку «Unarmed», как на 1.1 рисунке.

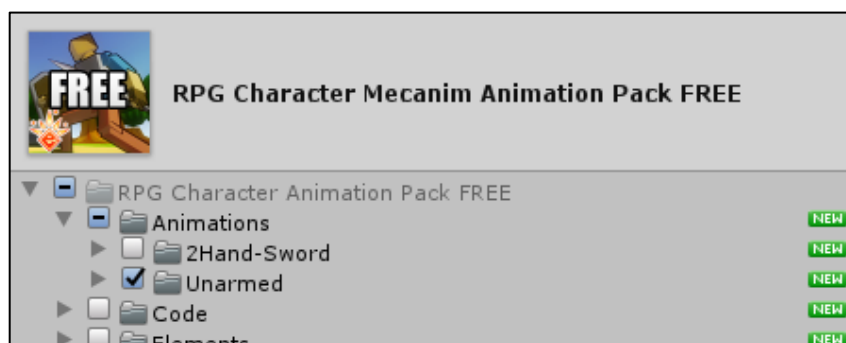


Рисунок 1.1

После импорта следует перезапустить Unity для того, чтобы анимации правильно связались с проектом. После этого можно запускать демо сцену.

### *Постэффекты*

Для того, чтобы сделать такую же картинку как на видео, нужно импортировать пакет PostProcessing из Package Manager.

После импорта у объекта «PostProcessing» появится компонент «Post Processing Volume» с пустым параметром «Profile». Туда нужно поместить готовый профайл, который показан на 1.2 рисунке.



Рисунок 1.2

## СОЗДАНИЕ RAGDOLL НА ПРИМЕРЕ ПЕРСОНАЖА ETHAN ИЗ STANDARD ASSETS

### 1. Компонент «Ragdoll Constructor» и выбор активных костей.

Компонент отвечает за создание специального рэгдолла для работы с ассетом. Этот конструктор отличается от того, что встроен в движок. Его можно найти по пути изображенном на рисунке 2.1.

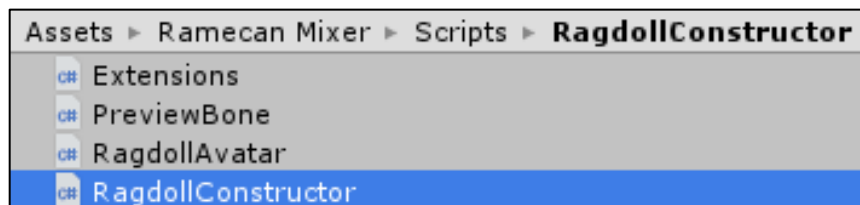


Рисунок 2.1

У Итана довольно много костей, и все они будут отображены в виде иерархии как на рисунке 2.2.

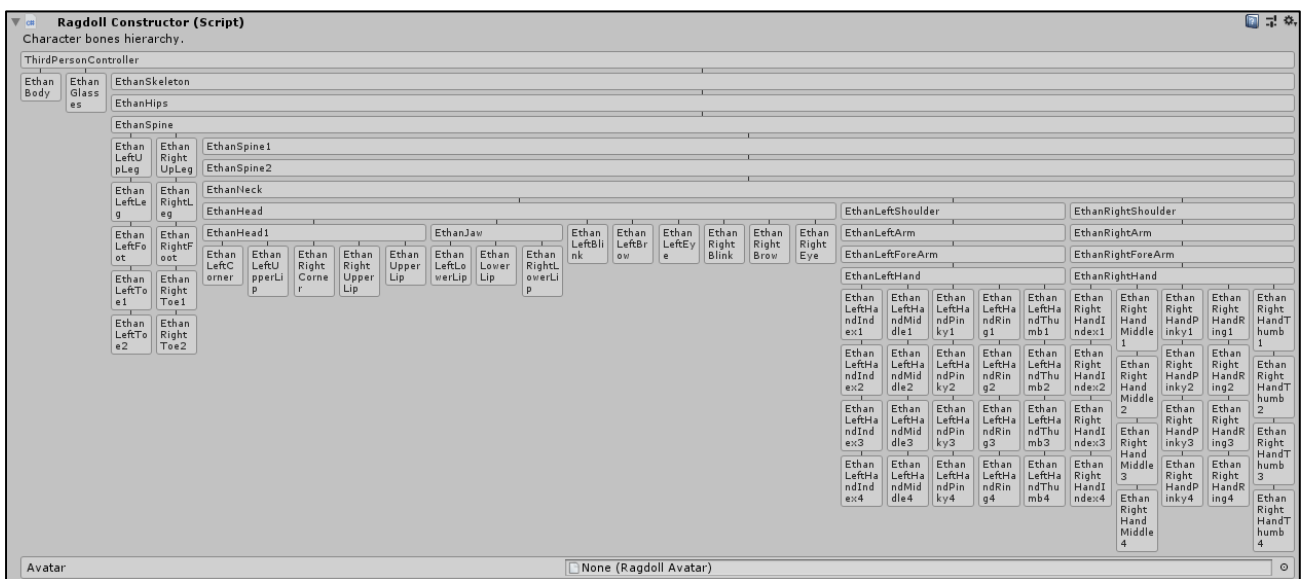


Рисунок 2.2

Нужно выбрать именно те кости, из которых будет строиться рэгдолл. Но для начала требуется установить аватар. Можно использовать один из готовых. Они находятся в папке «Ragdoll Avatars». На рисунке 2.2, внизу, есть специальный параметр. В него и нужно поместить выбранный аватар.

После установки готового аватара в инспекторе появится еще одна иерархия костей. Если конструктор определит одинаковые названия между костями персонажа и костями аватара, то он автоматически их свяжет. А если кости аватара остаются красными, значит нужно устанавливать связи вручную.

Для этого надо зажать мышь на кости аватара и протянуть соединительную нить до соответствующей кости персонажа, как на рисунке 2.3.

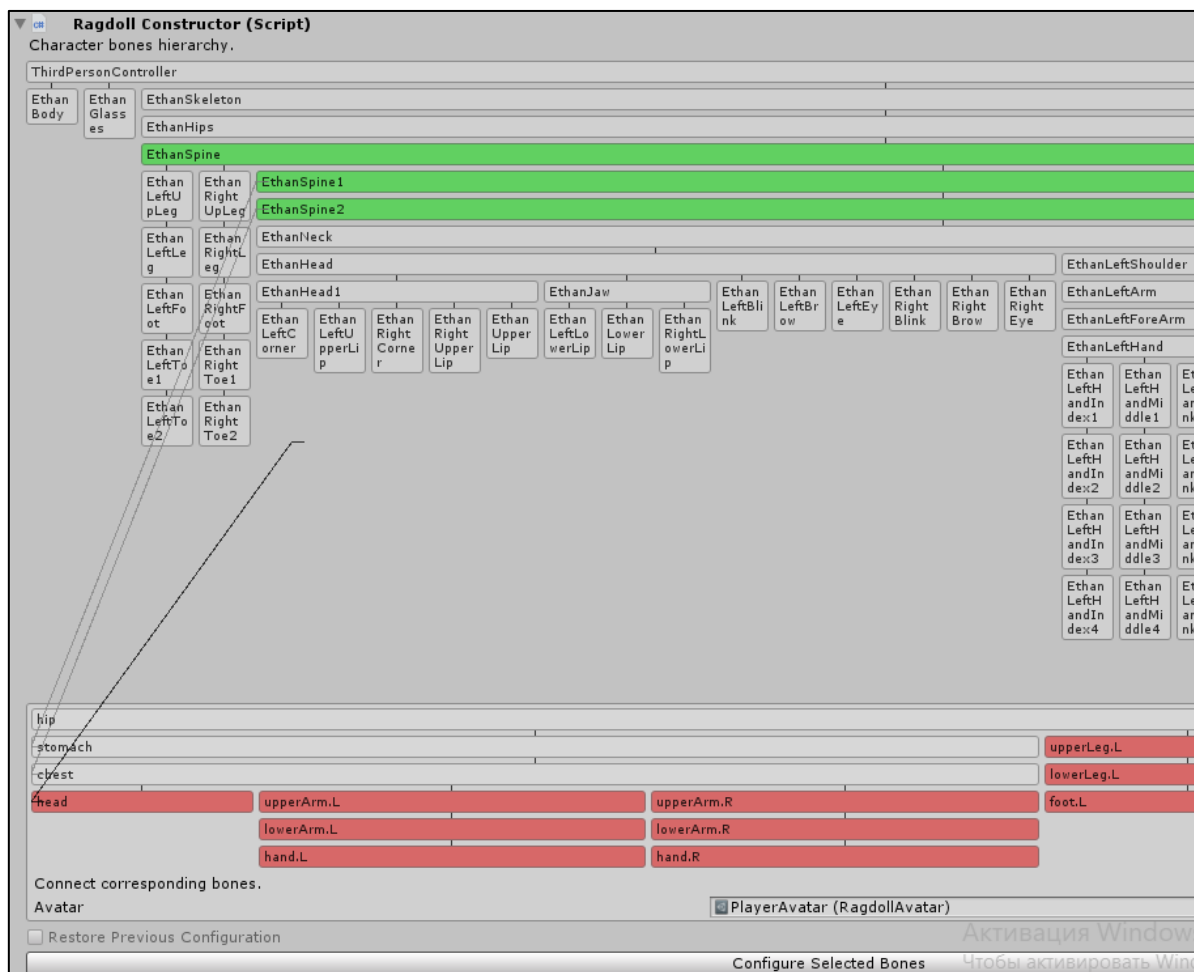


Рисунок 2.3

Но так как для Итана есть готовый аватар, то удобнее взять его. Тогда соединения настроятся автоматически, по названиям костей.

*Зеленым цветом подсвечиваются активные кости, из них будет собран рэгдолл. По нажатию на кости персонажа, можно включать и выключать их.*

Если вы хотите создать новый аватар, то достаточно нажать правой кнопкой мыши в области вкладки «Project» и выбрать в открывшемся контекстном меню «Create – Ragdoll Avatar», как на рисунке 2.4. Он находится почти в самом верху списка.

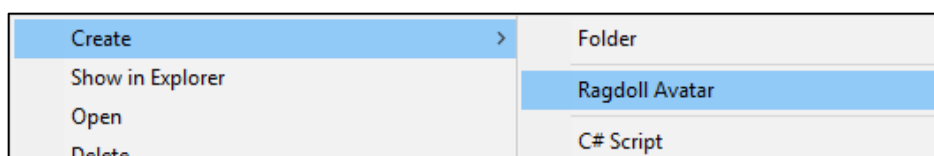


Рисунок 2.4

После того, как все нужные кости выбраны и подсвечены зеленым цветом можно приступить к конфигурированию костей. Для этого нужно нажать на кнопку «Configure Selected Bones», изображенную на рисунке 2.3, внизу.

## 2. Конфигурирование костей.

На этом этапе происходит основная работа по настройке костей. При нажатии на кость, внизу компонента появляются ее настройки, а в сцене будет отображен коллайдер или джоинт этой кости. Это можно увидеть на 2.5 рисунке. Изменять параметры костей можно как во вкладке инспектора, так и на сцене.

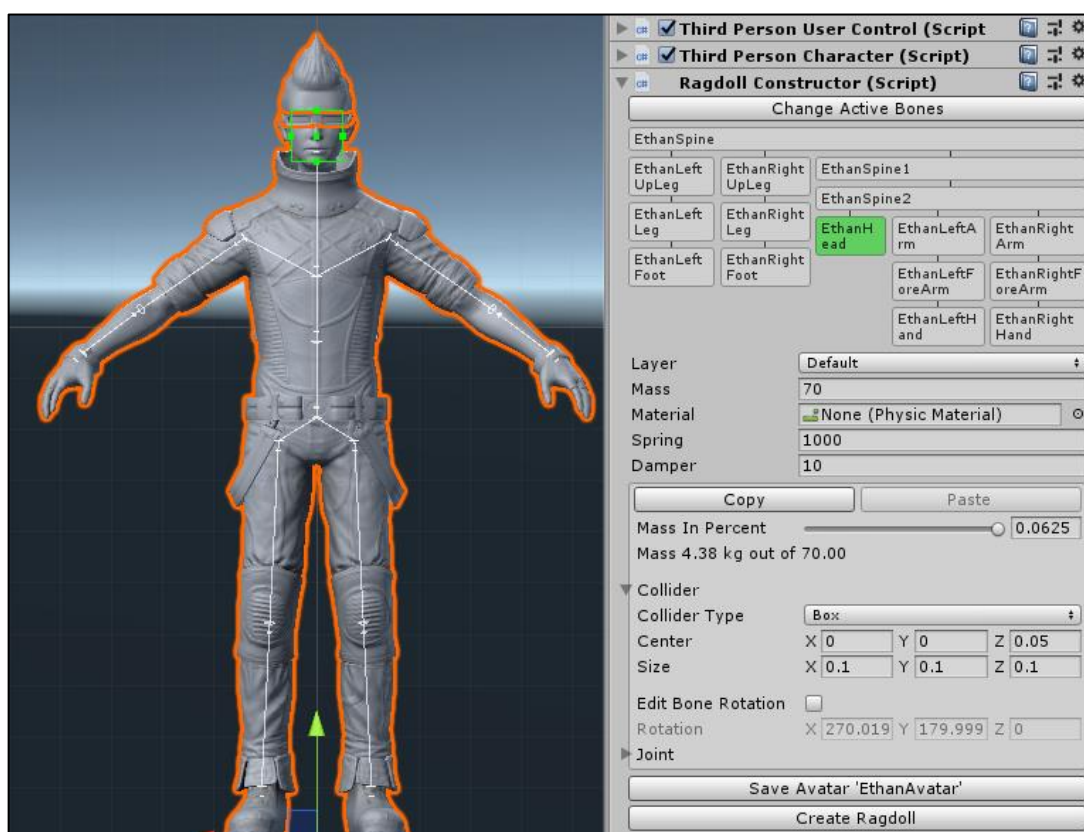


Рисунок 2.5

После настройки одной кости вы с легкостью можете скопировать эти параметры в другую кость. Это может понадобиться при настройке одинаковых или зеркальных частей тела. Скелет рэгдолла немного отличается от скелета персонажа ориентациями костей. Он строится так, чтобы при копировании зеркальных костей, большая часть настроек была идентичной. Но есть два параметра, которые, необязательно, нужно будет изменить вручную, при отзеркаливании костей. Первый, «Collider – Rotation» – этот параметр не копируется, а остается таким же, каким был. Второй, «Collider – Center» – этот параметр копируется, но одну из осей (обычно X) надо будет инвертировать, добавив или убрав знак минус(-) перед значением параметра. Но в большинстве случаев можно обойтись без исправления этих параметров.

На данном этапе желательно выбрать слой, в котором будут находиться кости. Советую создать специальный слой «Ragdoll», и выбрать его. Установить слой можно будет и после создания рэгдолла.

Что бы создать новый слой, нужно нажать «Layer – Add Layer», в самом верху инспектора. А после этого выбрать созданный слой в конструкторе рэгдолла.

Внизу 2.5 рисунка есть кнопка «Save Avatar», предназначенная для сохранения всех произведенных настроек, отдельно, в аватаре, для последующего использования при создании рэгдолла других персонажей.

После произведения настроек остается только нажать на кнопку «Create Ragdoll», чтобы создать все нужные объекты и компоненты рэгдолла.

### 3. Проверка рэгдолла и модификация кода «Third Person Character».

После нажатия на кнопку создания рэгдолла на сцене создается объект «ThirdPersonController Container». Это сделано для того, что бы не потерять связанных между собой персонажа и его рэгдолл. Они находятся внутри контейнера. Так же создается объект «Ragdoll», внутри которого находятся все созданные кости с компонентами. Отличие от стандартного рэгдолла в том, что рэгдолл существует отдельно от самого персонажа, а не в его собственных костях. Все это изображено на 2.6 рисунке.

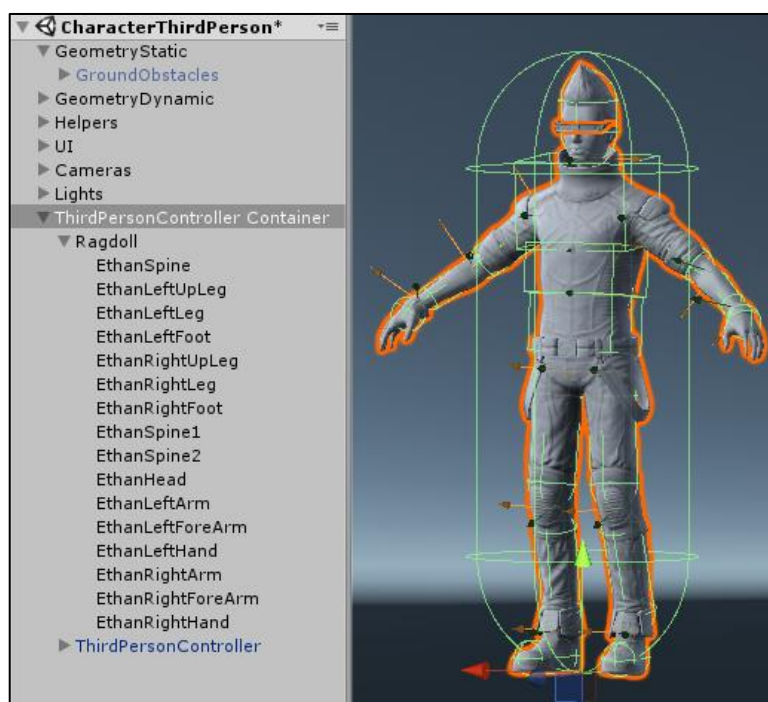


Рисунок 2.6

Фактически рэгдолл уже готов, им можно пользоваться. Но в данном случае, в коде «Third Person Character» имеется Raycast луч, который натывается на коллайдеры рэгдолла. Поэтому персонаж приседает, так как думает, что



находится в ограниченном пространстве. Это очень легко исправить. Для начала следует открыть код «Third Person Character» и добавить переменную «layerMask», как на 2.7 рисунке.

```
17 [SerializeField] float m_GroundCheckDistance = 0.1f;  
18 [SerializeField] LayerMask layerMask;  
19  
20 Rigidbody m_Rigidbody;
```

Рисунок 2.7

Эта переменная будет отвечать за то, какие слои будут видны для рэйкаста. В коде есть три использования рэйкаста, в них нужно будет добавить параметр «layerMask», точно так же как на 2.8 рисунке.

```
93 if (Physics.SphereCast(crouchRay, m_Capsule.radius * k_Half, crouchRayLength, layerMask, QueryTriggerInteraction.Ignore))  
111 if (Physics.SphereCast(crouchRay, m_Capsule.radius * k_Half, crouchRayLength, layerMask, QueryTriggerInteraction.Ignore))  
212 if (Physics.Raycast(transform.position + (Vector3.up * 0.1f), Vector3.down, out hitInfo, m_GroundCheckDistance, layerMask))
```

Рисунок 2.8

Теперь остается только убрать галочку рядом со слоем рэгдолла в компоненте «Third Person Character» в инспекторе.

Компонент «Ragdoll Mixer» выполняет всю работу по смешиванию физики рэгдолла и анимации meanim. Он автоматически добавляется персонажу после завершения создания рэгдолла в конструкторе. *А компонент конструктора можно удалить, если он вам больше не нужен. Но следует уточнить, что после удаления, рэгдолл можно будет изменить только напрямую, через соответствующие компоненты костей.*

Уменьшив коллайдер самого персонажа, можно добиться взаимодействия рэгдолла с окружающими объектами, как на рисунке 2.9.

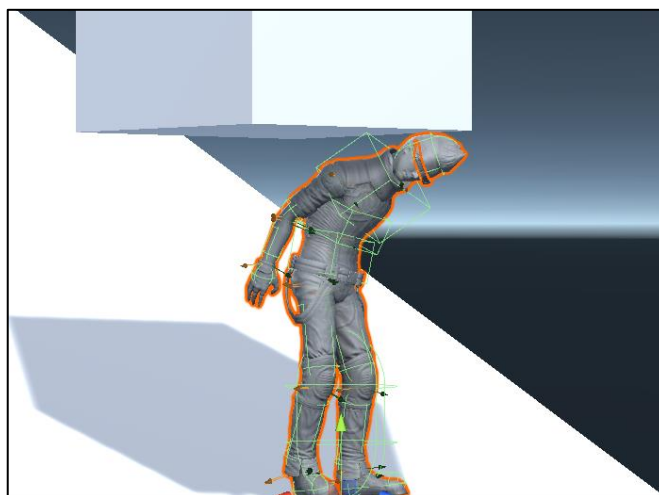


Рисунок 2.9

## РАБОТА С СОСТОЯНИЯМИ В RAMECAN MIXER

Компонент «Ramecan Mixer» представляет из себя форму для работы с состояниями костей. Как можно увидеть на рисунке 3.1, в верхней части есть возможность добавлять, удалять и изменять состояния, а немного ниже присутствует иерархия костей рэгдолла. А при нажатии на кость, в самом низу появляется текущее состояние параметров кости. Можно выбирать несколько костей и изменять их одновременно.

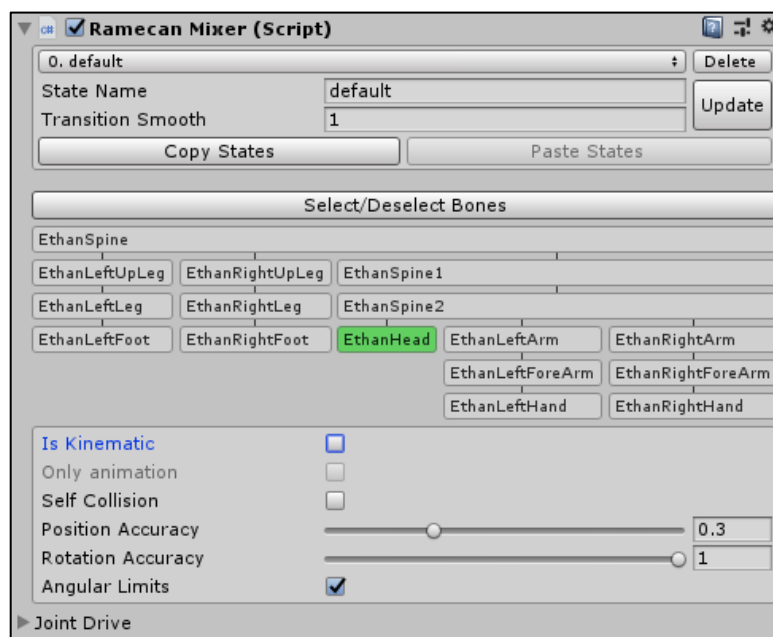


Рисунок 3.1

При смене состояний, параметры костей меняется соответственно выбранному состоянию. Но для начала нужно добавить новое. Для этого просто введите его название, например «dead». Тогда кнопка рядом изменится на «Add», ее и нужно нажать. Хорошо. Следующий параметр, это плавность смены состояний. Чем меньше значение, тем быстрее и реще будет переход. Если значение будет «0», то переход будет мгновенным.

При выбранном состоянии «dead», нажмите на кнопку «Select/Deselect Bones», чтобы выбрать все кости. Вы увидите, что параметры связанные с вращением стали неактивными. Это потому, что выделена корневая кость, а на ней отсутствует джоинт. Если вы нажмете на корневую кость, то с нее снимется выделение, и параметры джоинта станут активными. Здесь нужно изменить параметр «Rotation Accuracy» на значение 0,1. Теперь снова выберите все кости, и измените параметр «Position Accuracy» на значение 0, для того чтобы персонаж не стоял вертикально в пространстве. Еще можно включить параметр «Self Collision». Для сохранения состояния, нажмите кнопку «Update».

Если запустить сцену, то выбранное состояние автоматически активируется. Можно попробовать сменить состояние, и посмотреть на реакцию персонажа. При ручной смене состояния оно сменяется мгновенно, без плавного перехода. Мы будем изменять состояние программно, это будет именно так плавно, как нужно.

Сделаем простой скрипт, который будет менять состояния по нажатию кнопки. Для того, что бы сменить состояние, нужен всего лишь один метод – «BeginStateTransition». В него нужно вписать имя требуемого состояния, либо порядковый номер.

```
using RagdollMecanimMixer;
using UnityEngine;

public class ChangeStateExample : MonoBehaviour {

    private RamecanMixer ramecanMixer;

    void Start () {
        ramecanMixer = GetComponent<RamecanMixer>();
    }

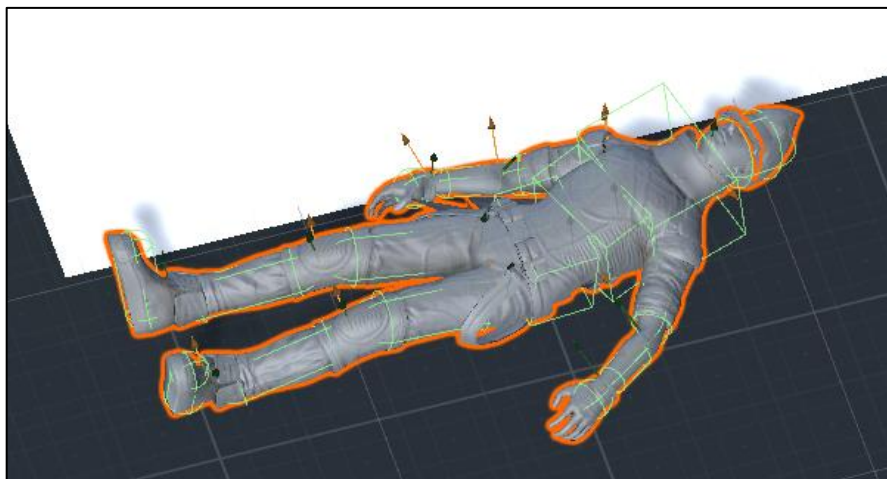
    void Update () {
        if (Input.GetKeyDown(KeyCode.UpArrow))
            ramecanMixer.BeginStateTransition("default");

        if (Input.GetKeyDown(KeyCode.DownArrow))
            ramecanMixer.BeginStateTransition("dead");
    }
}
```

Осталось только добавить этот скрипт персонажу и запустить сцену. Теперь при нажатии на стрелки вверх и вниз состоянию будут меняться.

Это был элементарный пример использования состояний рэгдолла. С помощью него вы сможете создавать такое поведение рэгдолла, которое вам требуется. В ассет так же включена демонстрационная сцена, в которой показан один из возможных вариантов использования.

Желаю успешного и приятного использования «Ragdoll Mecanim Mixer»!



## ВАЖНЫЕ ЗАМЕТКИ

1. У компонента Animator персонажа параметр **Culling Mode** должен быть выбран **Always Animate**. В этом случае анимация будет проигрываться даже если персонаж находится вне поля зрения камеры. Это позволит избежать странных рывков, когда персонаж появляется в поле зрения камеры.

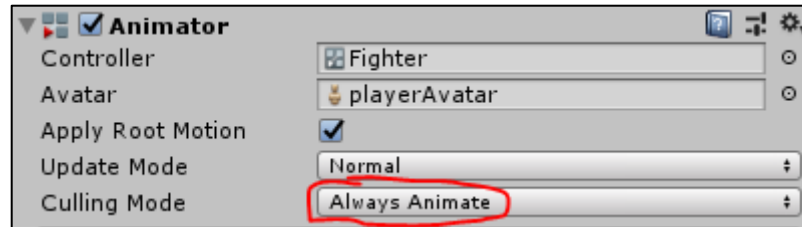


Рисунок 4.1

2. В некоторых случаях может понадобиться использование разных слоев для самого персонажа и его рэгдолла. Слой для рэгдолла можно назначить заранее. В компоненте **Ragdoll Constructor** имеется соответствующий параметр.

Обратите внимание, что после импорта ассета, у персонажа и у костей пустые слои. Это происходит потому, что слои не подлежат экспорту.

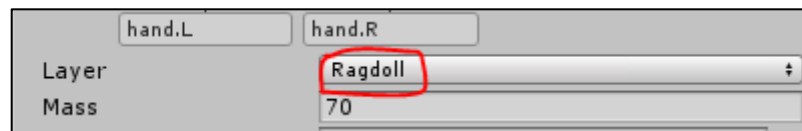


Рисунок 4.2

3. Для передвижения персонажа следует перемещать основной объект персонажа, а не его контейнер. При создании рэгдолла, конструктор создает контейнер для персонажа и его рэгдолла, как на 4.3 рисунке. Рэгдолл находится отдельно, для улучшения физических параметров. А контейнер создается для того, что бы не потерять связанные объекты.

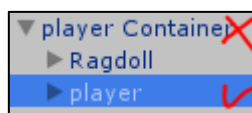


Рисунок 4.3