



TALLER

EN INTEGRACIÓN DE
ANIMACIONES EN UNITY

GUÍA DEL
USUARIO



ORGANIZADO POR: EVA ESQUIVEL MORENO Y PETER JOHN CUTINO MATA

ÍNDICE

1. ¿Qué vamos a hacer?
2. Herramientas necesarias
3. Desarrollo del Taller
 - 3.1. Creación del modelo
 - 3.2. Proceso de elección de animaciones
 - 3.3. Importación a Unity
 - 3.4. Creación de estados
 - 3.5. Creación de transiciones
4. Visualización del modelo y sus animaciones
5. Bibliografía

1. ¿Qué vamos a hacer?

Vamos a organizar un taller cuyo propósito principal es introducir a la integración de animaciones en modelos 3D.

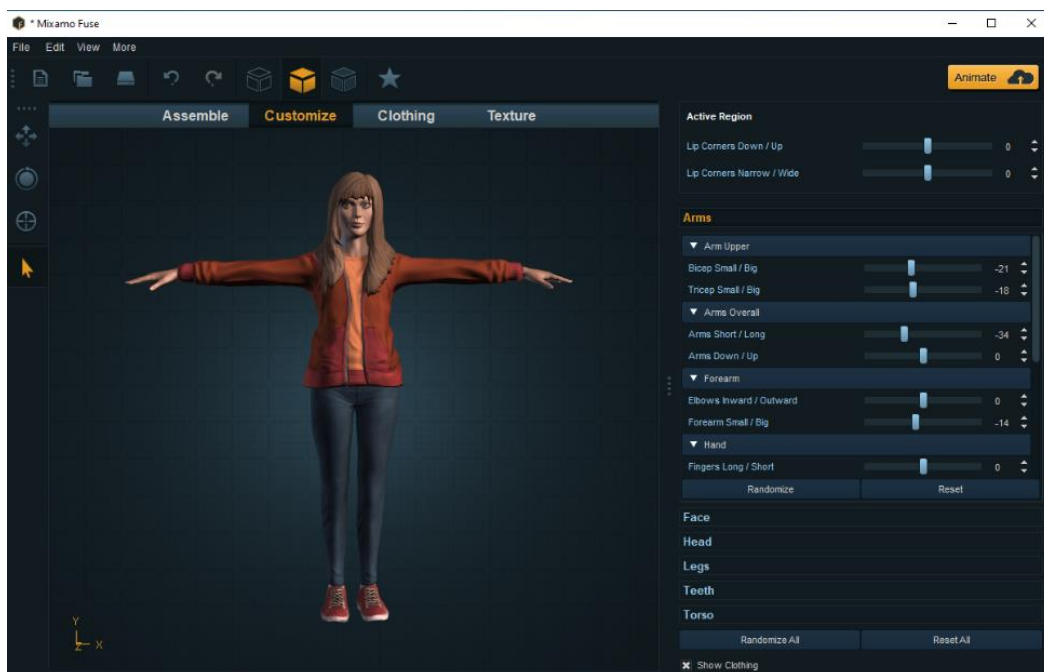
Lo principal será mostrar que con herramientas de alto nivel podemos construir un modelo completamente animado. No vamos a centrarnos en modelado ni en la creación de animaciones en sí, pero si esperamos que los asistentes al finalizar este taller sientan un mayor interés por el desarrollo de proyectos en Unity.

Para ello, trabajaremos en un ejemplo básico de integración en Unity que pueda realizarse en 1 hora y media aproximadamente.

2. Herramientas necesarias

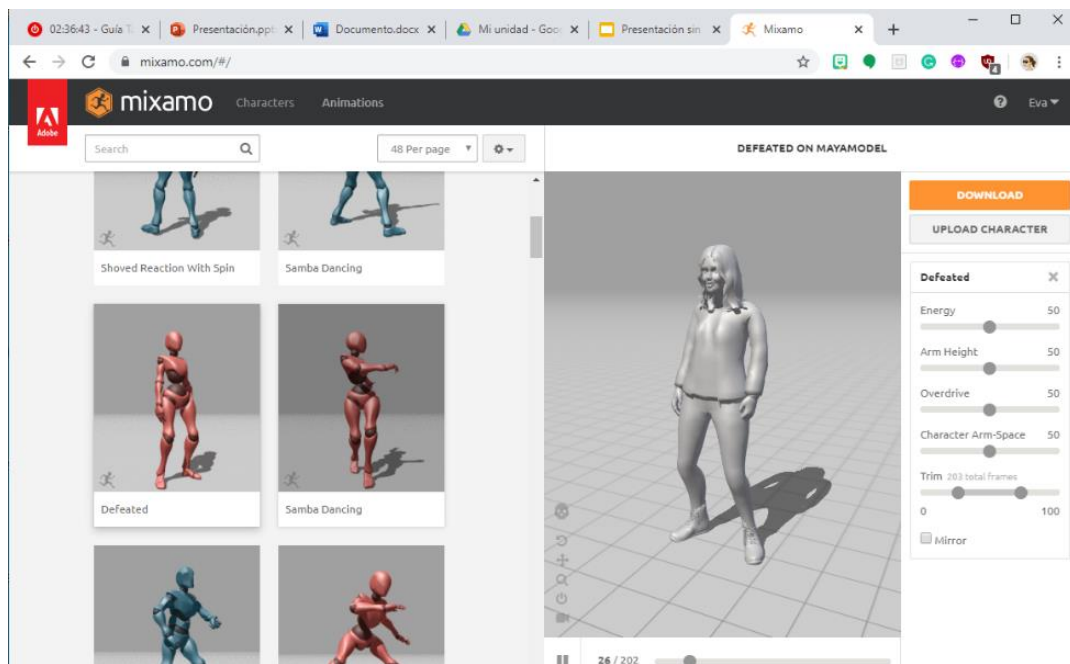
Para este taller necesitaremos:

- **Fuse CC:** Es una herramienta que podemos encontrar en Steam, y que utilizaremos para generar un avatar 3D que animaremos.

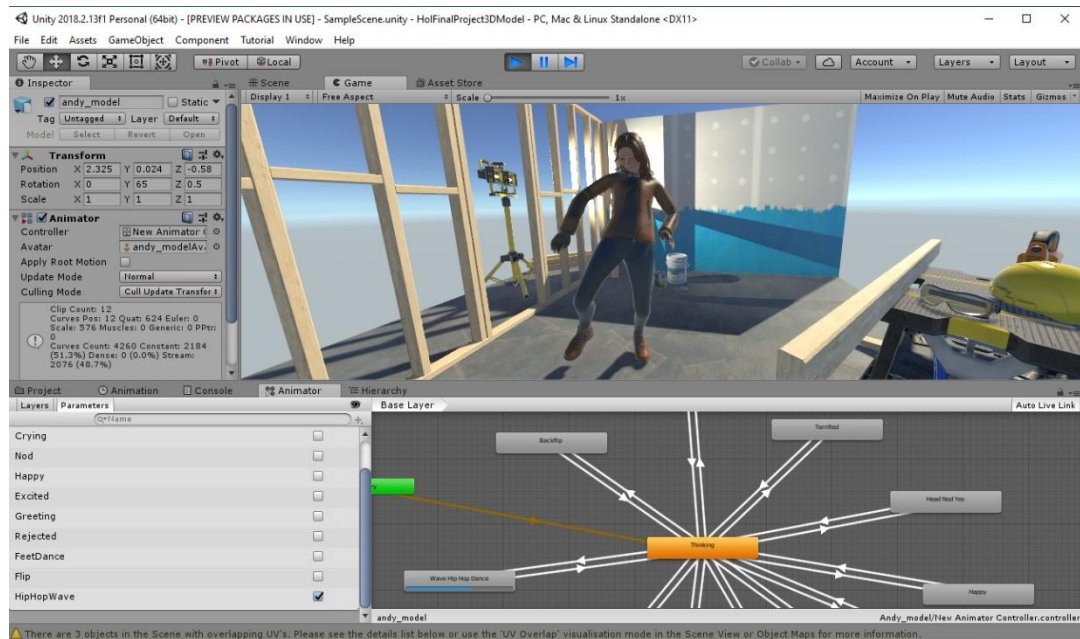


- **Mixamo:** Es una página web que conjuntamente con Fuse CC permite “dar vida” al modelo. De aquí podremos obtener las animaciones que aplicaremos a nuestro modelo 3D. (<https://www.mixamo.com/#/>)

GUÍA DEL TALLER DE INTEGRACIÓN DE ANIMACIONES EN UNITY



- **Unity:** Es un motor de videojuegos multiplataforma. Nosotros lo utilizaremos para integrar el modelo con sus respectivas animaciones.



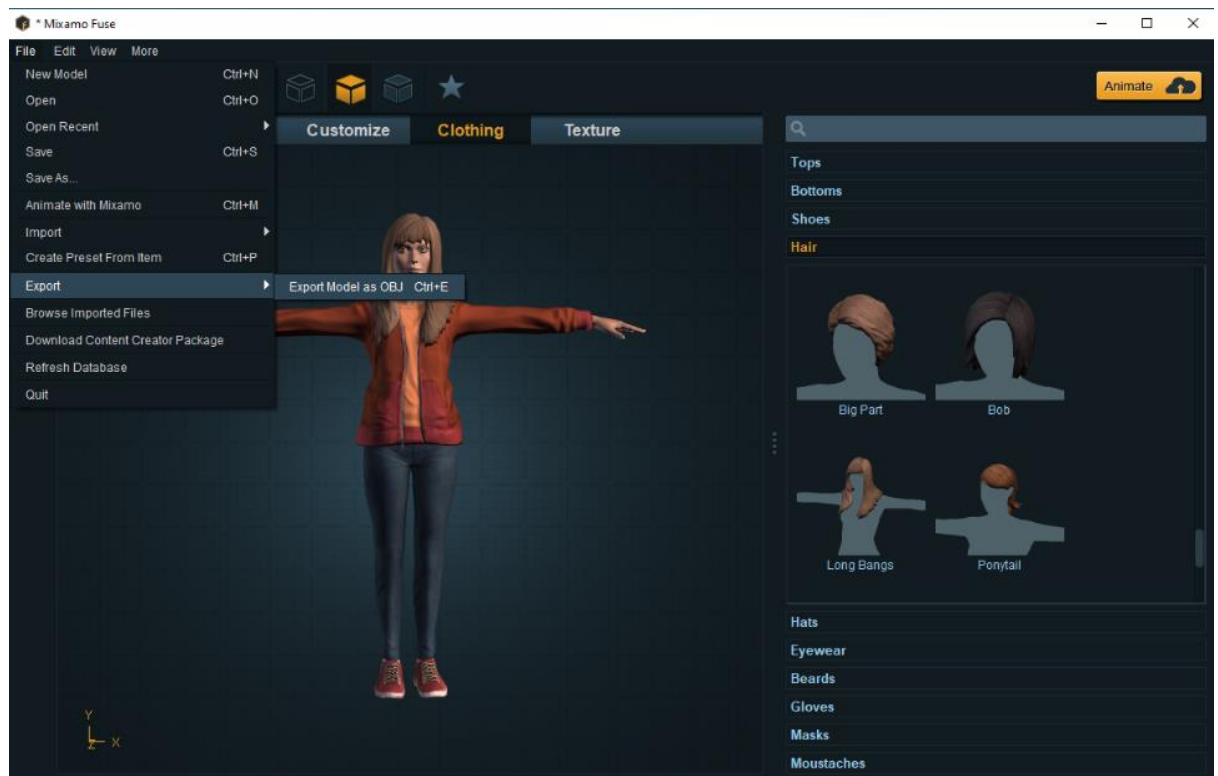
3. Desarrollo del taller

Vamos a seguir los siguientes pasos:

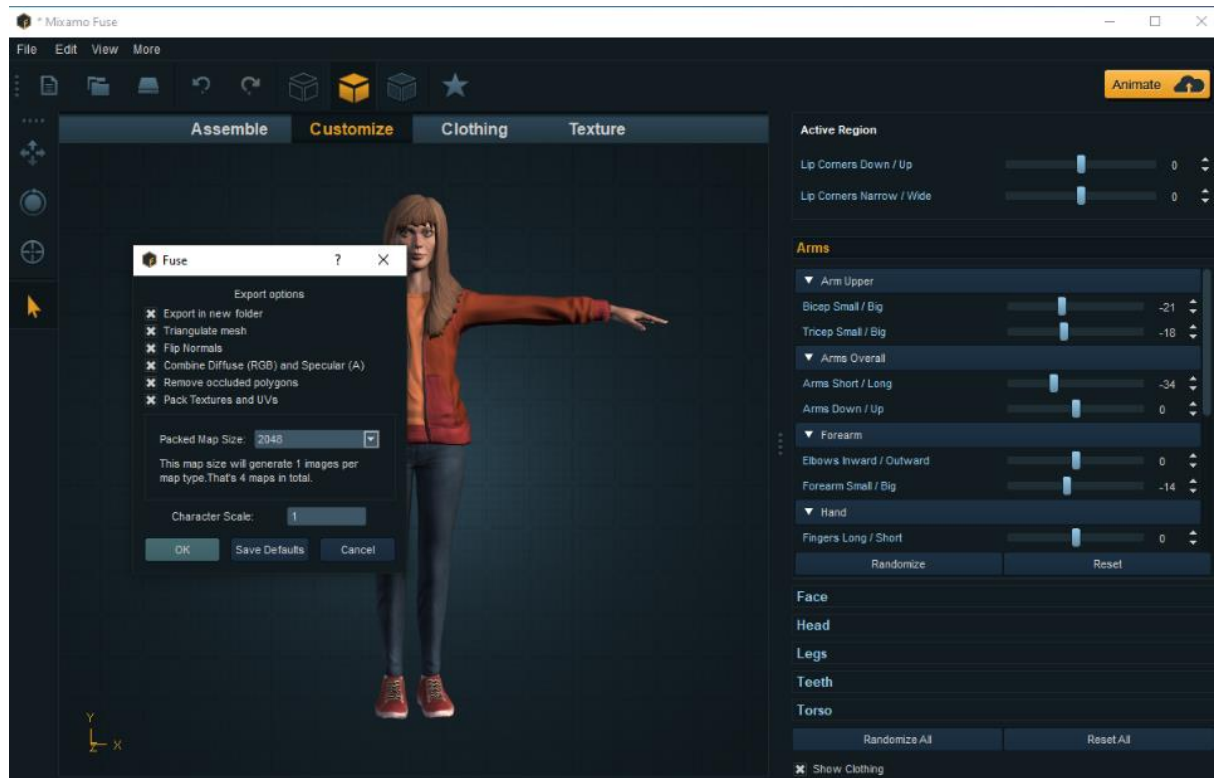
3.1. Creación del modelo

Para la creación del modelo, nosotros vamos a hacer uso de Fuse CC.

El programa nos permite personalizar el modelo, modificando desde la vestimenta, hasta los parámetros más específicos que definen los rasgos faciales y textura de la ropa. Una vez creado nuestro modelo personalizado, debemos exportarlo como un objeto:

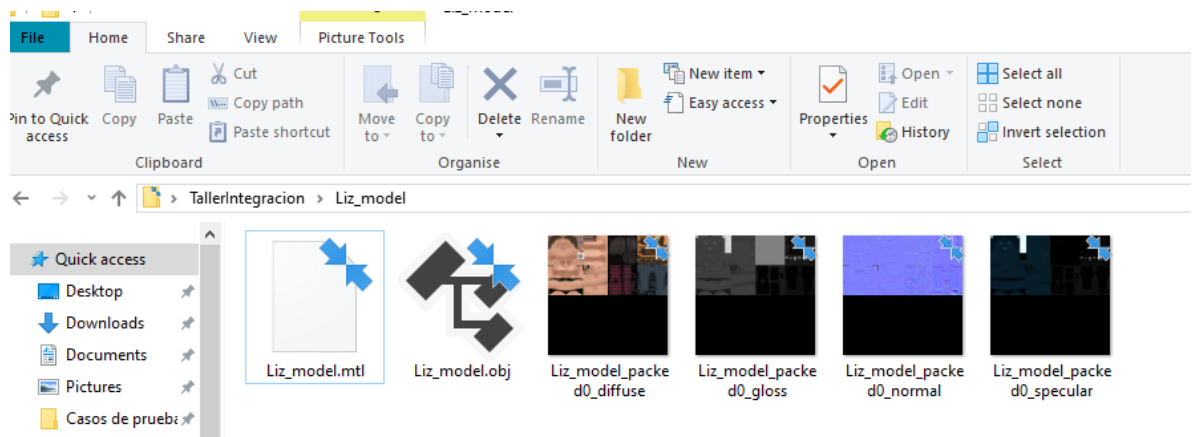


File > Export> Export Model as OBJ



A continuación, **marcar todos los checkboxes** y pulsar Ok. Se abrirá el sistema de archivos para guardar el modelo en una ruta. Aconsejamos crear una carpeta en una ruta fácil de encontrar para guardar el modelo.

En la carpeta deben haberse guardado todos esos elementos:

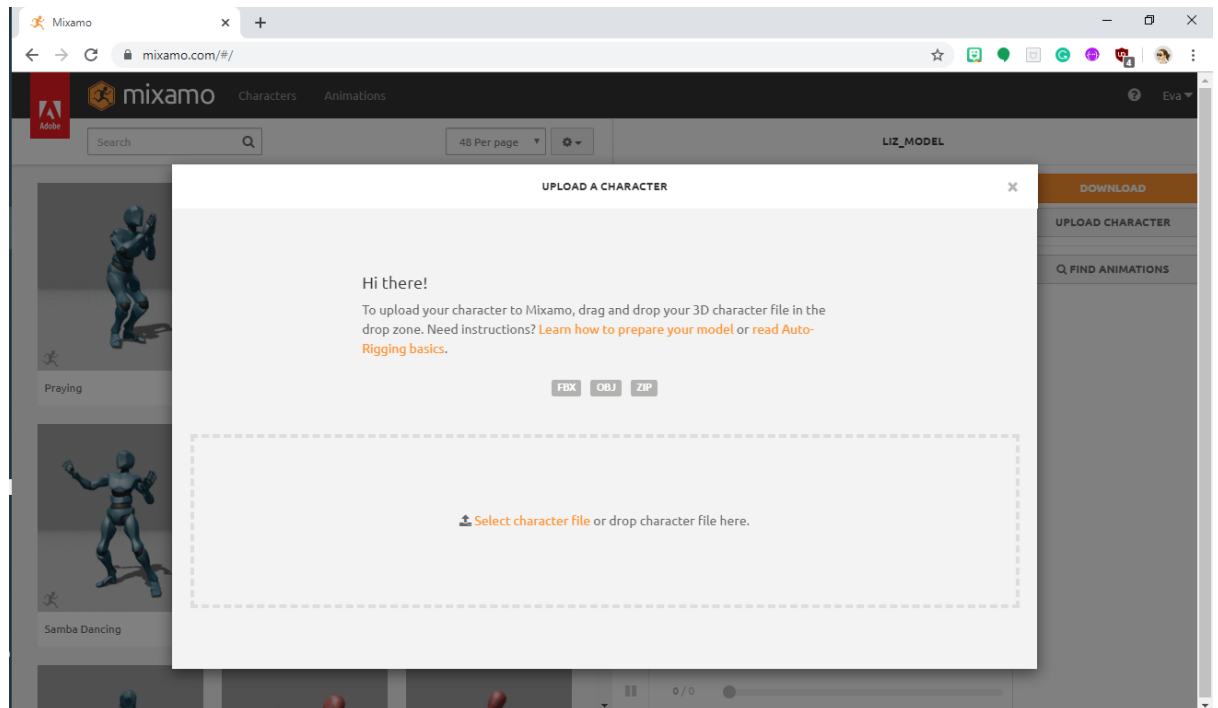


3.2. Proceso de elección de animaciones

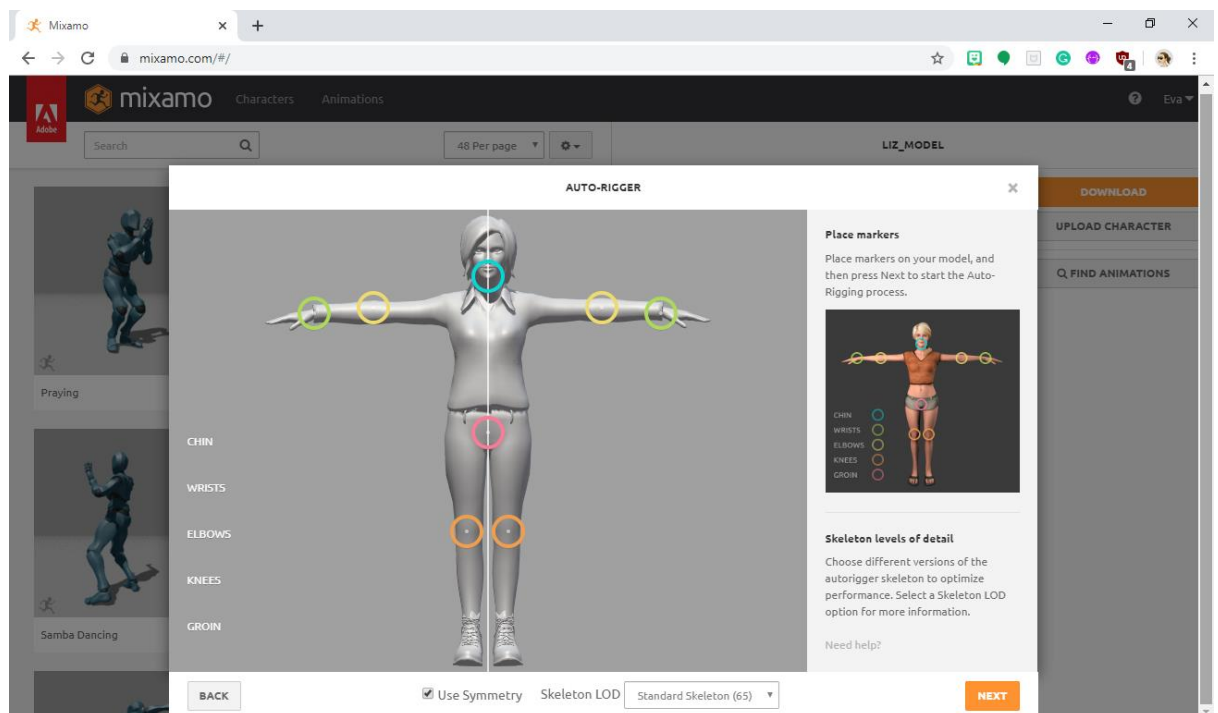
Accedemos a <https://www.mixamo.com/#/> para generar animaciones para el modelo.

Será necesario subir el modelo que ya hemos creado con FuseCC, para poder adaptar a él las animaciones.

GUÍA DEL TALLER DE INTEGRACIÓN DE ANIMACIONES EN UNITY



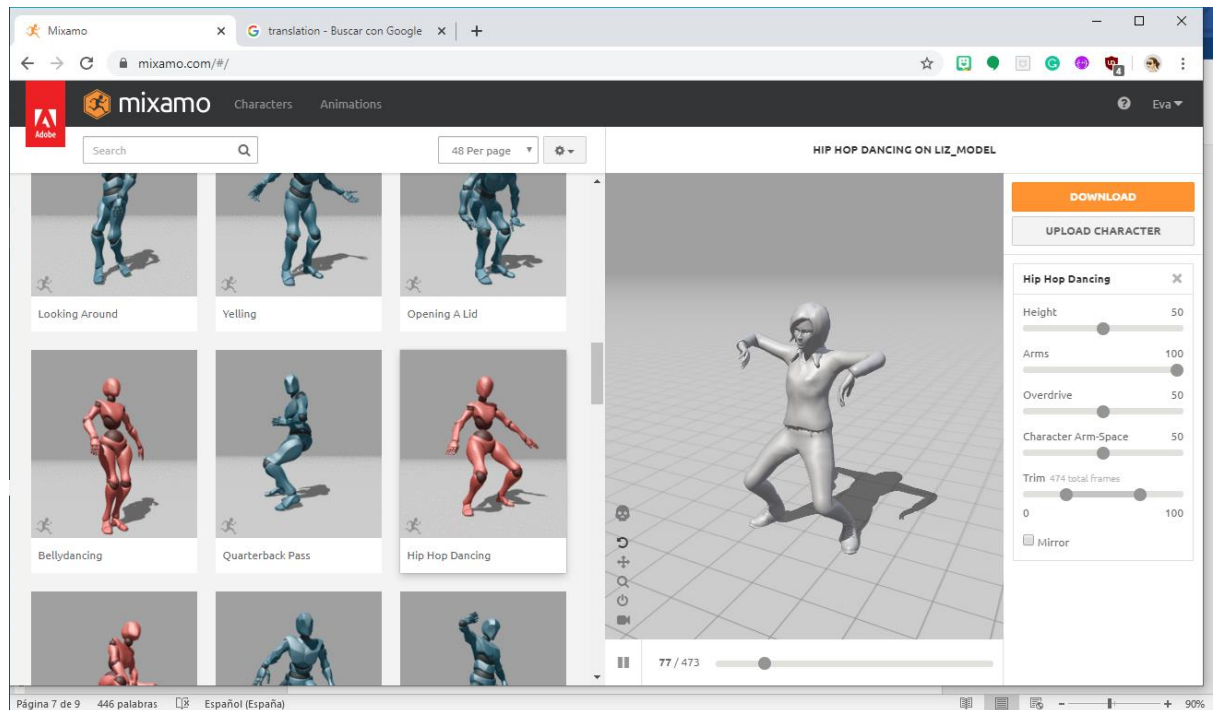
Si pulsamos en Upload Character nos aparecerá esta ventana. *Select character file* abrirá nuestro sistema de archivos. Debemos elegir el modelo en formato .obj y abrirlo.



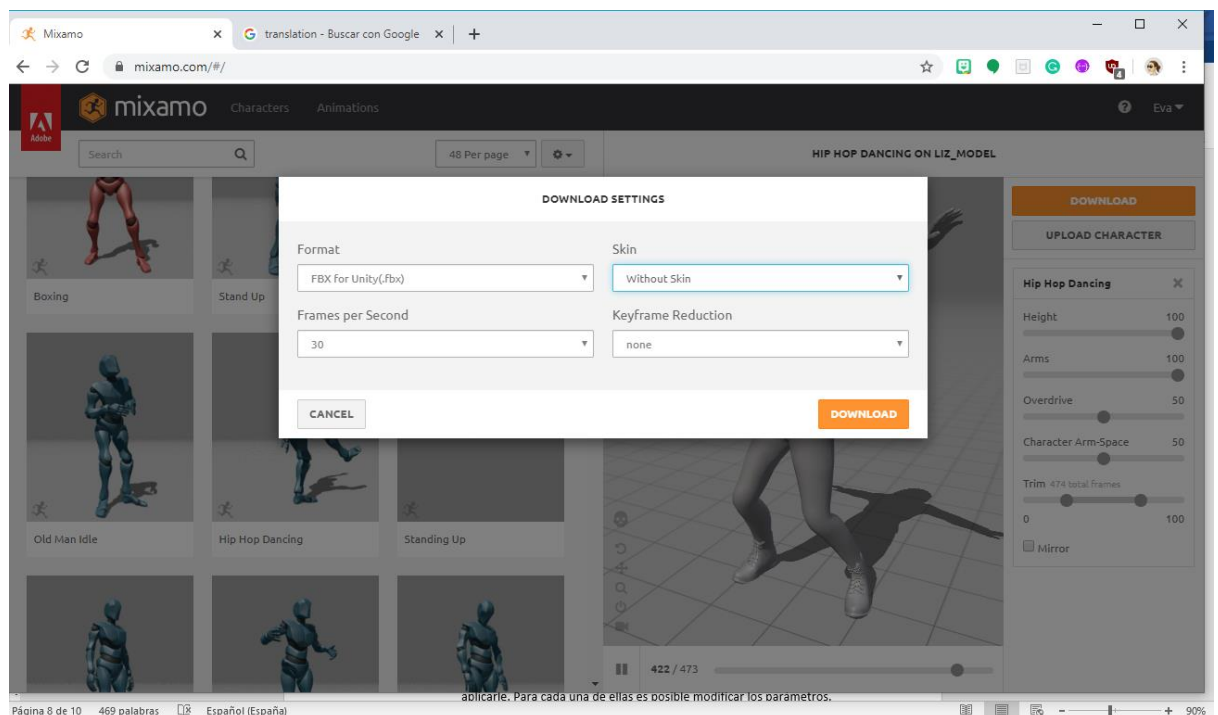
Es necesario colocar los distintos marcadores en el modelo para que las animaciones puedan aplicarse correctamente en el modelo tomando estos puntos como referencia.

También se puede modificar el número de detalles del esqueleto elegido.

GUÍA DEL TALLER DE INTEGRACIÓN DE ANIMACIONES EN UNITY



Una vez cargado el modelo, podemos elegir las animaciones que vamos a aplicarle. Para cada una de ellas es posible modificar los parámetros.



Nos descargamos la animación estableciendo:

- *Format: FBX for Unity (.fbx)*
- *Skin: Without Skin*

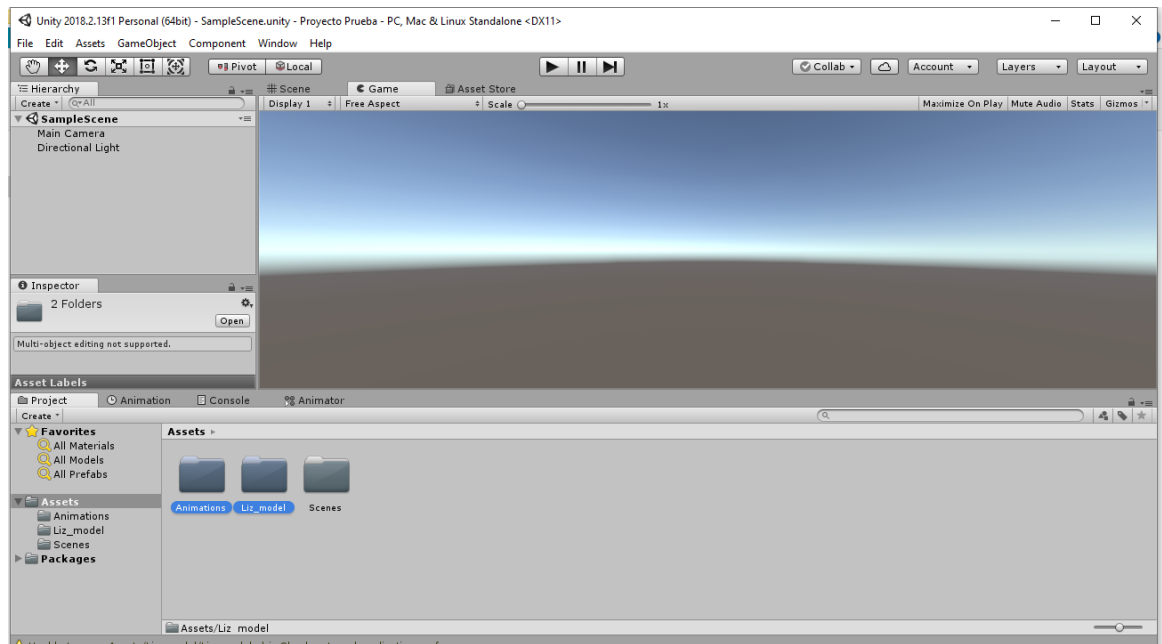
Y la guardamos, a ser posible, en una subcarpeta de la que creamos para guardar el modelo

3.3. Importación a Unity

Abrimos Unity y lo primero que tenemos que hacer es crear un proyecto.


Arrastramos tanto la carpeta con el modelo y demás como las animaciones que nos hemos descargado.

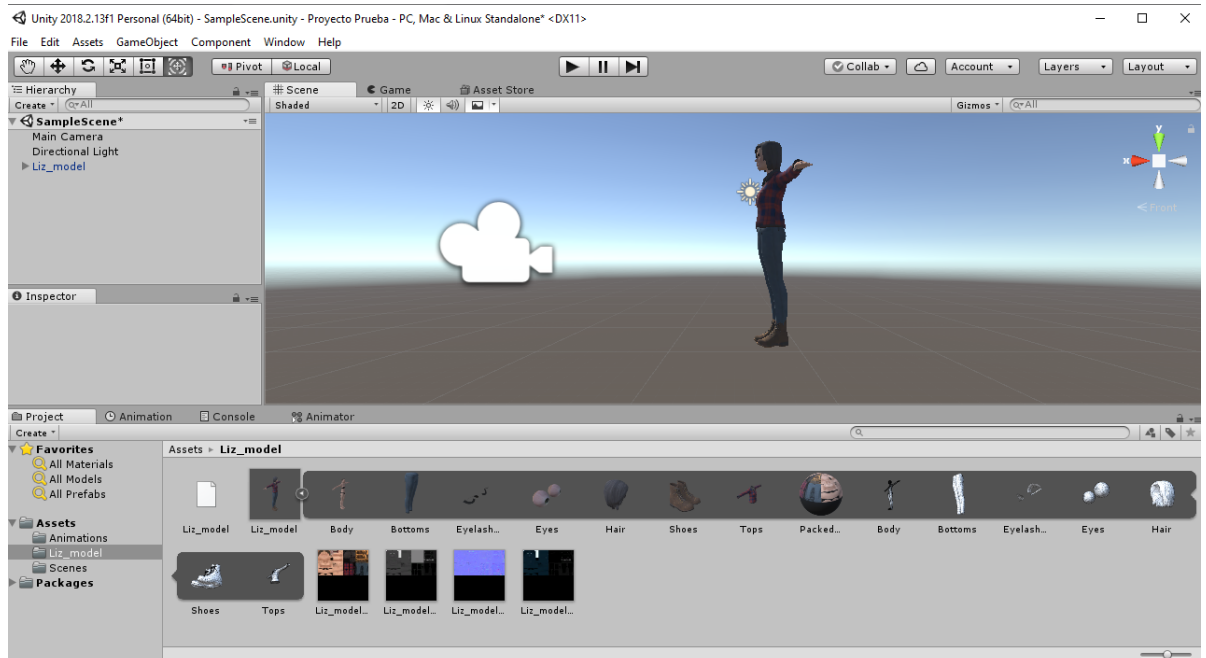
Lo guardamos en carpetas en la ventana *Project*:



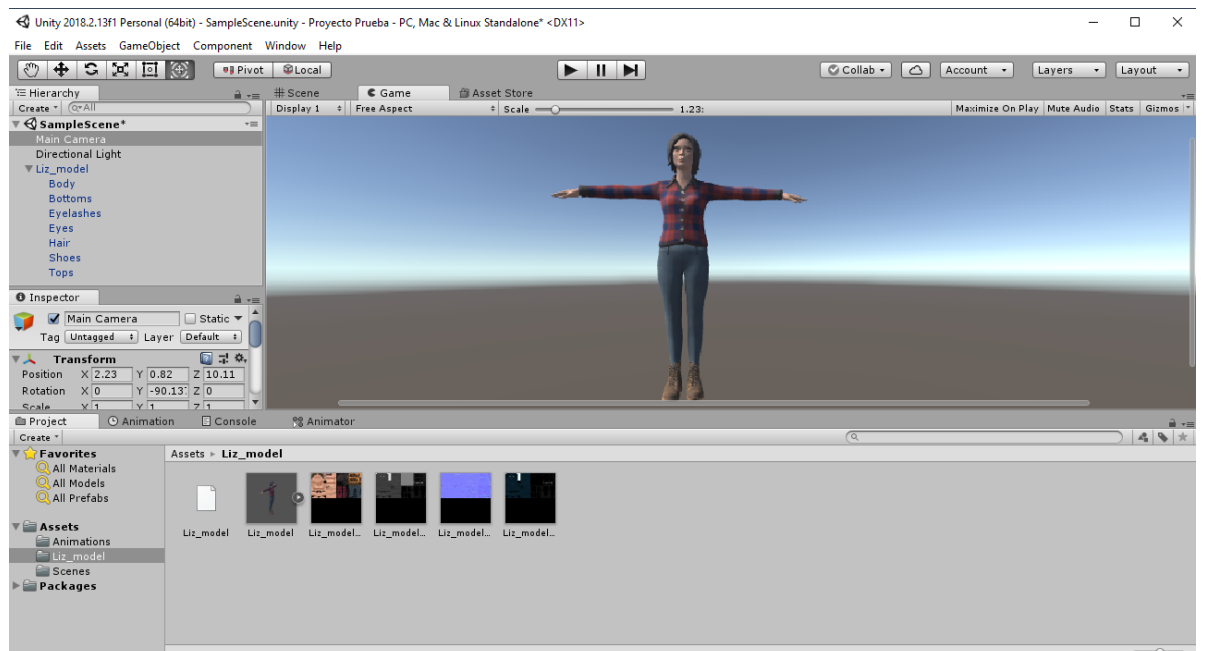
Después de eso, arrastramos el modelo a la escena, y si fuera necesario ajustamos el valor de la escala en la ventana Inspector, habiendo seleccionado antes el modelo en la ventana *Hierarchy*.



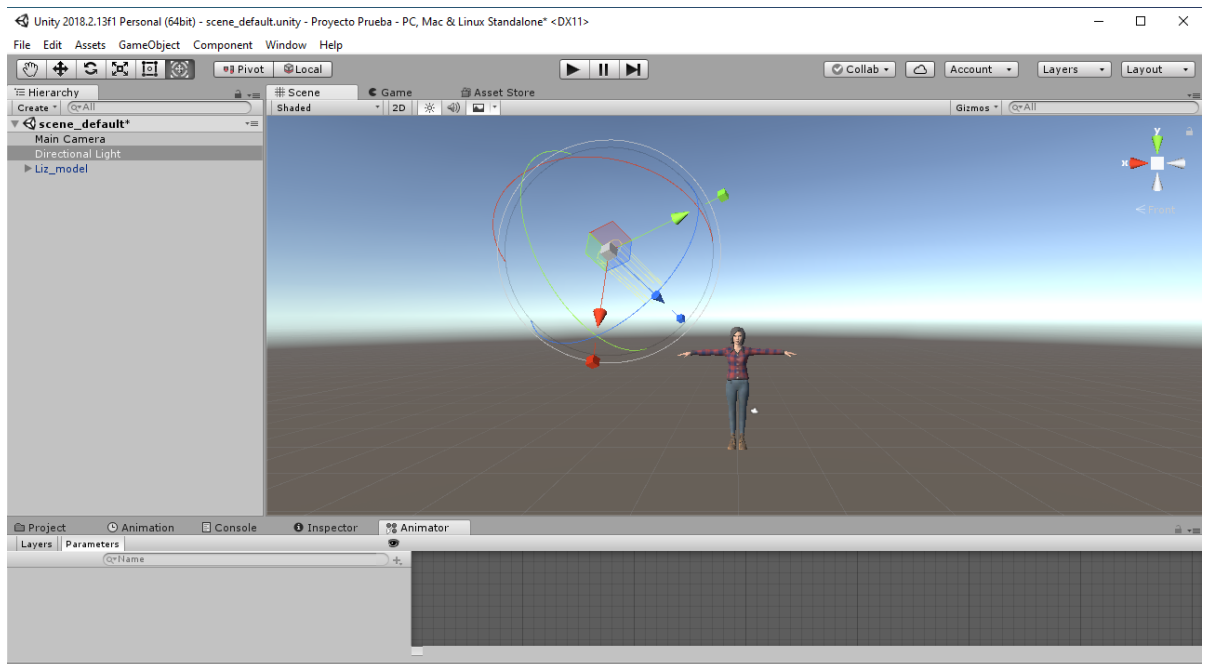
A continuación, para que cuando reproduzcamos las animaciones se observe bien el modelo (perspectiva en la pestaña *Game*), debemos mover la cámara. Esto se hace pulsando el botón  y después seleccionando la cámara en la venta *Hierarchy*. Solo es necesario ir girando la cámara, hasta al mirar el modelo a través de la ventana *Game*, se vea completo.



La visión desde esa ventana debería ser la siguiente:

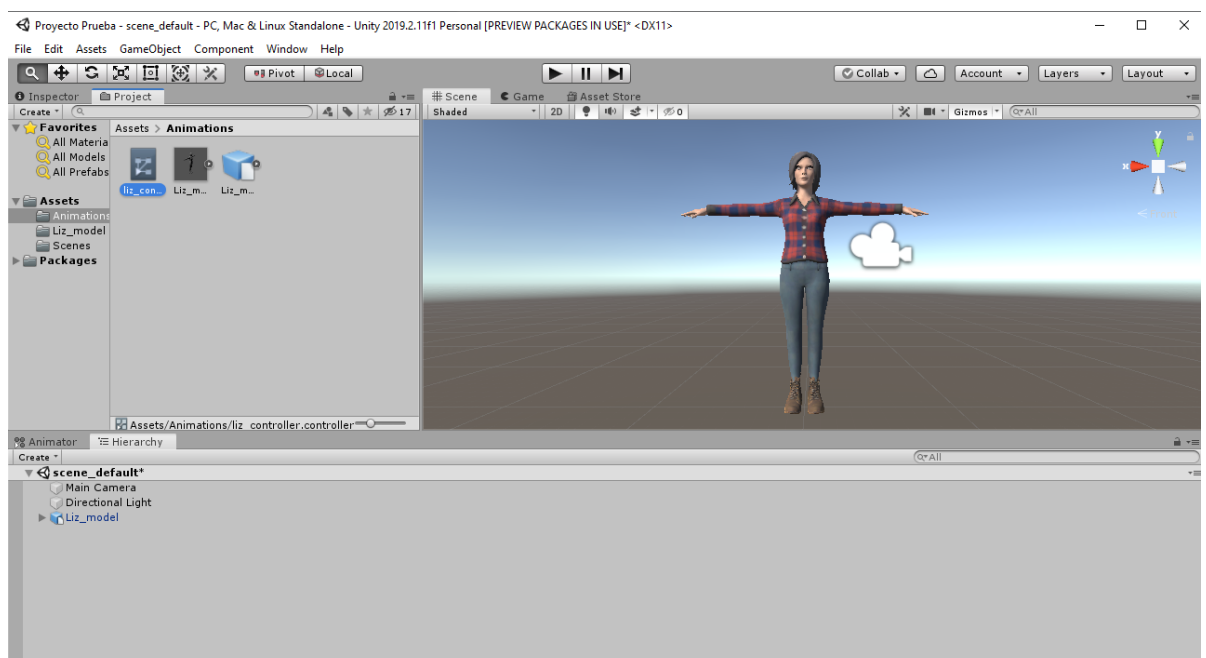


De la misma forma movemos la luz direccional, para que el modelo pueda visualizarse correctamente.



3.4. Creación de estados

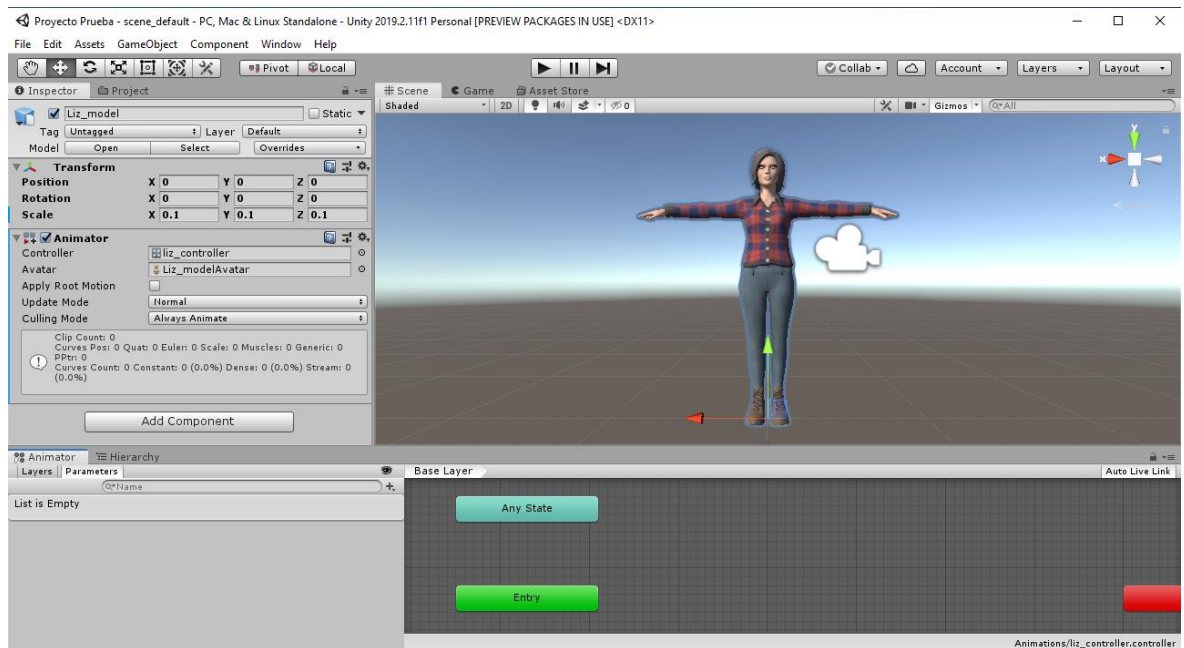
Para definir las animaciones del modelo debemos crear un controlador, el cual nos permitirá realizar transiciones entre los distintos estados.



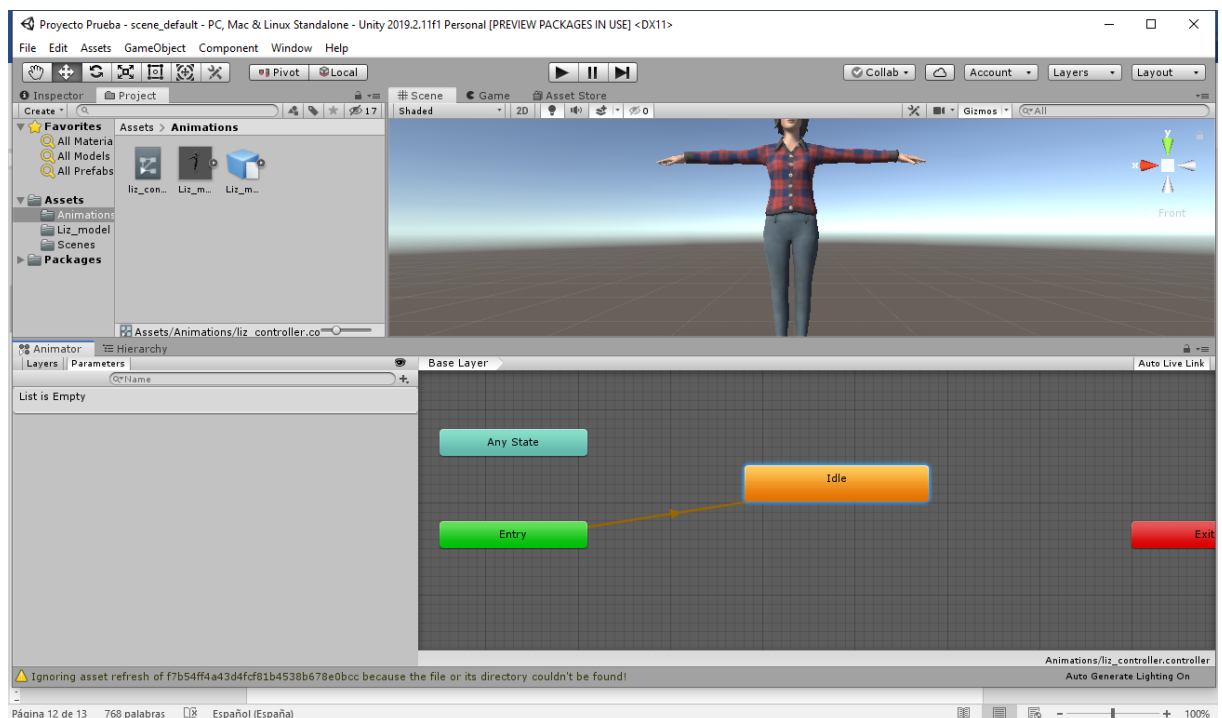
Lo primero es crear el controlador: en la ventana Project, *Botón derecho* > *Create* > *Animation Controller*.

Seleccionando el modelo en *Hierarchy*, en la ventana Inspector debemos visualizar el componente *Animator*. Ahí debemos establecer el Avatar que vamos a utilizar, y el controlador que acabamos de crear.

GUÍA DEL TALLER DE INTEGRACIÓN DE ANIMACIONES EN UNITY



A continuación, arrastramos las animaciones que nos hemos descargado al Animator Controller. Podemos decir que la animación que ocurrirá justo después del Entry state, será el estado por defecto. En nuestro caso, Idle será el estado inicial del avatar.

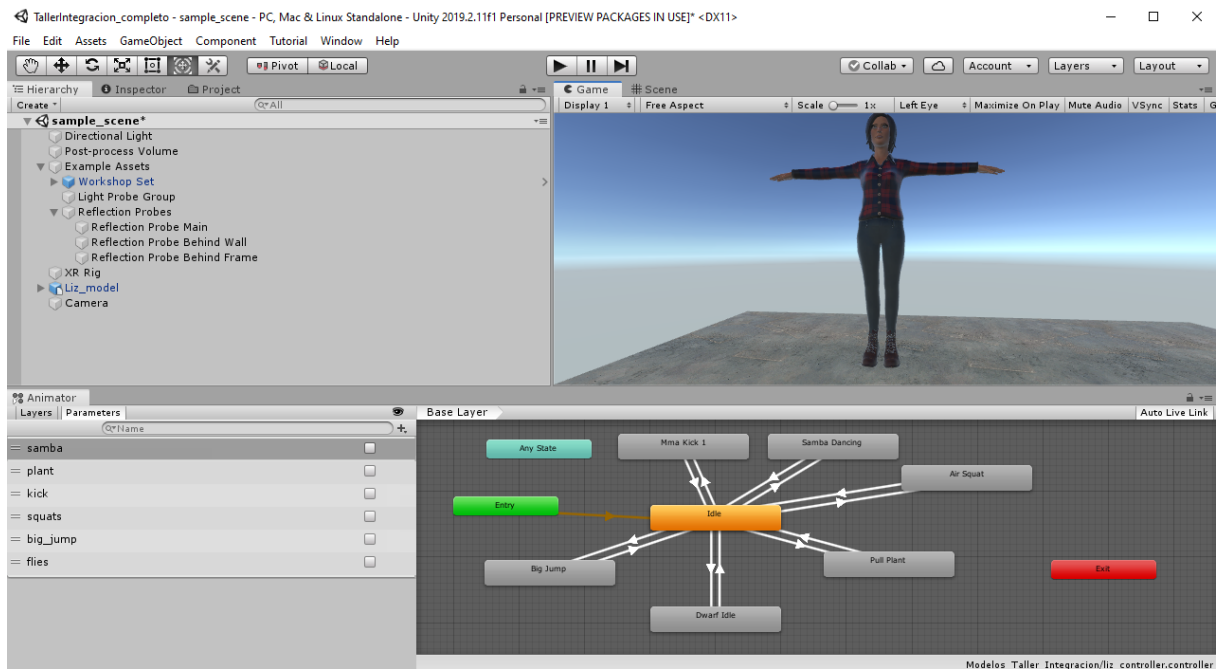


Seleccionamos la animación Idle, y en Inspector marcamos el checkbox **Loop Time**. De esta forma la animación se reproducirá en bucle.

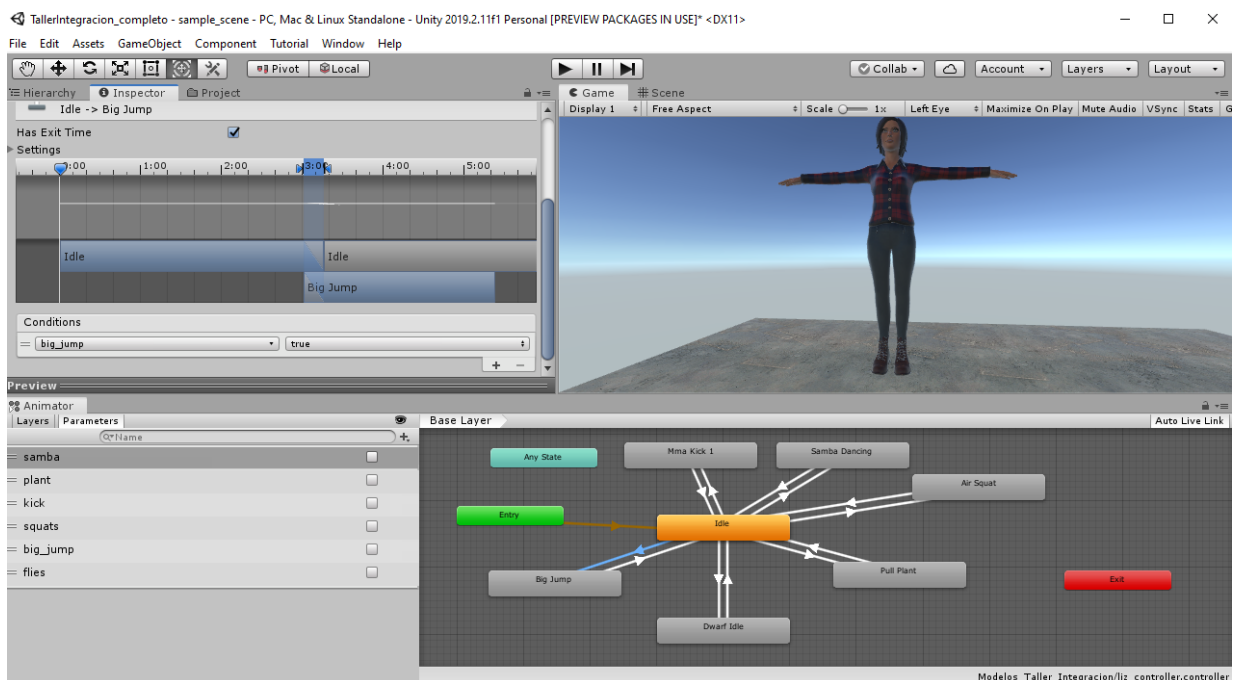
3.5. Creación de transiciones

¿Cómo añadir el resto de animaciones? Es muy simple.

De nuevo, hay que arrastrar las animaciones descargadas al *Animator*. La animación que añadimos al principio será nuestra animación inicial, a la cual vamos a conectar las demás.



También vamos a crear las transiciones. Esto significa que vamos a pasar de un estado a otro, haciendo uso de una variable *boolean* que vamos a crear para cada par de estados, tal y como se ve en la imagen anterior.

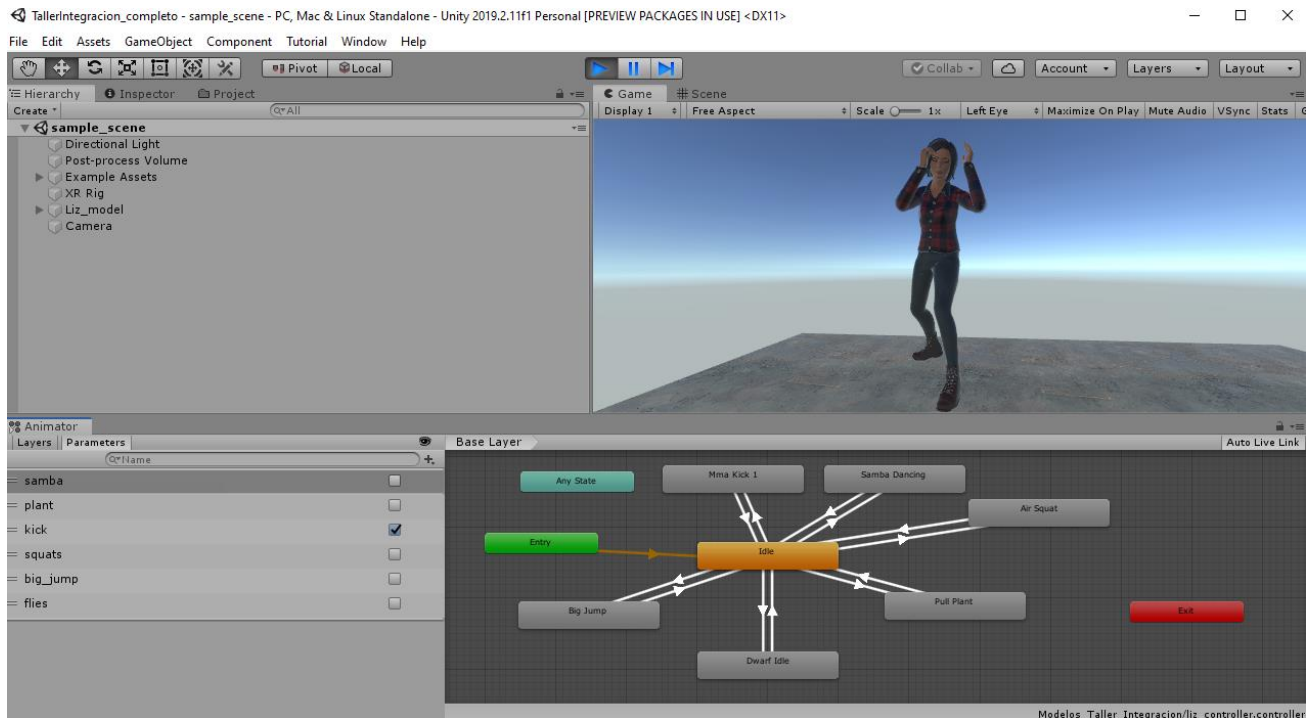


Por ejemplo, para ir de Idle a Big Jump, la variable *big_jump* será *true* al pasar de Idle a Big Jump y *false* al volver a Idle.

También es necesario establecer que la animación se reproduzca en bucle. Esto se hace seleccionando la animación en la ventana Project, y al acceder al Inspector seleccionar Loop Time y Loop Pose, además de guardar los cambios.

4 Visualización del modelo y sus animaciones

Por último, para visualizar las animaciones del modelo, pulsamos Play:



EXTRA: Cambiar estados por tecla

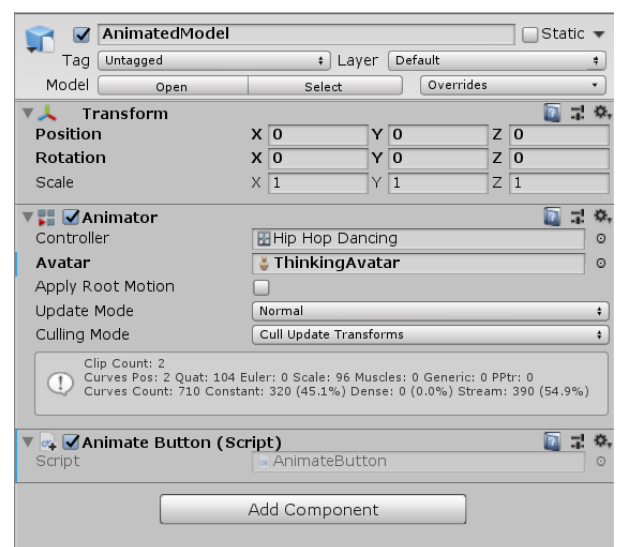
Seleccionamos **Add Component** y creamos un Script:

Se crearán por defecto dos funciones: **Start()** y **Update()**, pero sólo necesitamos introducir el código de la siguiente imagen dentro de **Update()**

```
void Update()
{
    if (Input.GetKeyDown(KeyCode.T)) {
        GetComponent<Animator>().SetBool("isDancing", true);
    }
}
```

Detrás de KeyCode puedes escribir la tecla preferida.

Una vez realizado esto al presionar la tecla especificada cambiará de estado a la animación seleccionada (puedes crear otro if en caso de querer introducir más animaciones).



4. Bibliografía

Exportación de Fuse y Mixamo: <https://aprendiendoadisenyarjuegos.blogspot.com/2017/11/fuse-para-generar-personajes-en-unity3d.html>

Animación de modelos con Mixamo: <https://www.youtube.com/watch?v=RbqyYorjUIs>

Importación de Mixamo a Unity: <http://www.walterpalladino.com/how-to-import-mixamo-animations-in-unity/>

Transiciones en Unity: <https://docs.unity3d.com/Manual/class-Transition.html>

Cambiar booleans de Animator desde Script en Unity: <https://forum.unity.com/threads/set-bool-to-opposite-if-i-push-a-key.264670/>

Funcionamiento de Animator.SetBool:
<https://docs.unity3d.com/ScriptReference/Animator.SetBool.html>