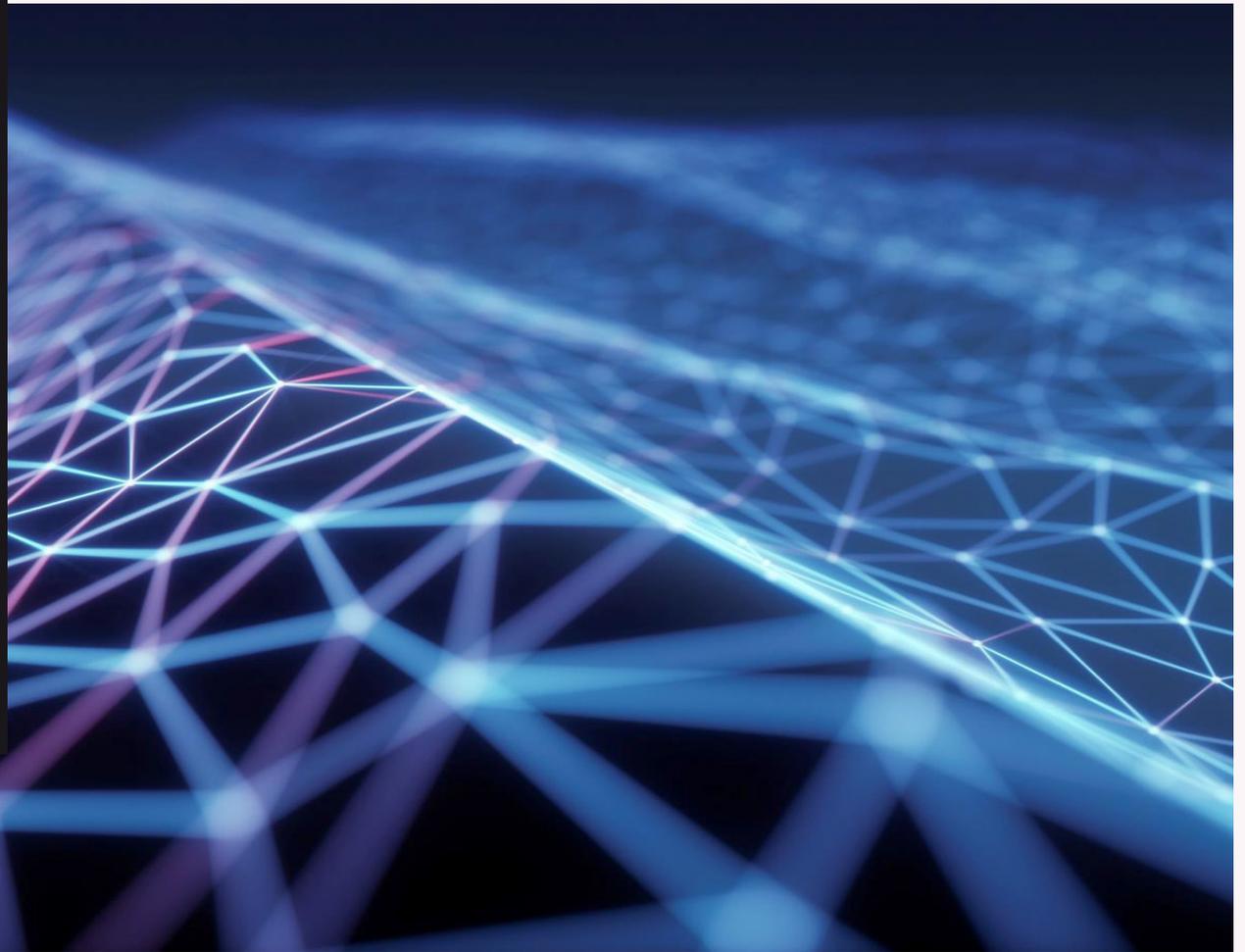


UD 06

TERRENOS & CHARACTER CONTROLLER

Desarrollo de Entornos
Interactivos Multidispositivo





**Attribution-NonCommercial-ShareAlike
4.0 International (CC BY-NC-SA 4.0)**

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).



Parte 1

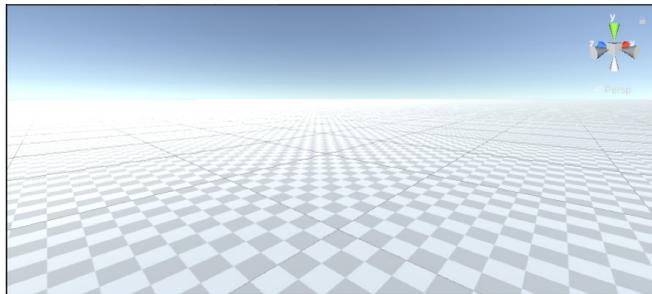
TERRENOS

Un terreno representa el suelo sobre el que se moverán nuestros personajes. Aprenderemos cómo crearlos dentro de Unity

TERRAIN TOOLS

Aunque existen programas para crearlos externamente (como Houdini o WorldMachine), Unity tiene una herramienta para generarlos internamente como un Game Object especial: GameObject ->3D Object -> Terrain.

Al crear un terreno, se generará en la escena un plano de grandes dimensiones, y en nuestro proyecto un archivo con los datos de configuración del terreno.

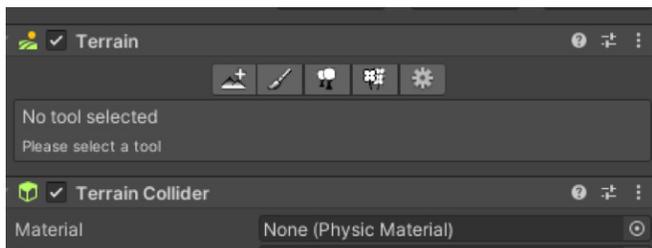


- Crear un terreno adyacente
- Pintar el terreno: añadir capas y crear elevaciones
- Añadir árboles y detalles
- Configuración, veremos las opciones básicas del terreno: detalle, sombras, distancia a la que se ven los arboles, viento, etc.

[Aquí](#) puedes ver todas las opciones

Terrain tools

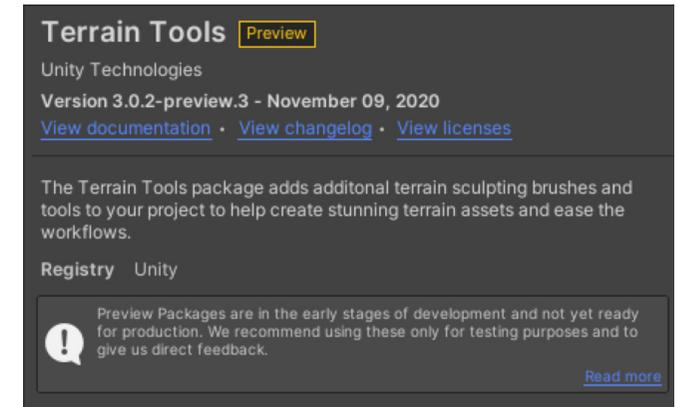
Si seleccionamos el terreno, veremos en el inspector que tiene opciones específicas de los terrenos (como un colisionador de tipo Terrain Collider). En el componente "terrain", organizadas por pestañas, tendremos las siguientes opciones para modelar y configurar el terreno, las cuales veremos más adelante:



Existe un paquete que facilita la creación y gestión de los terrenos, aunque no es -a día de hoy- un paquete oficial. Por ello, si queremos instalarlo tendremos que activar la visualización de paquetes que están en desarrollo. En "Edit > Project Settings > Package Manager" veremos la casilla donde podemos activar los paquetes en desarrollo (Enable Preview Packages), que debemos activar con cuidado tal y como nos advierte.

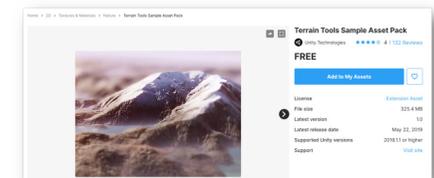


Una vez activada, podemos buscar el paquete Terrain Toolbox en la ventana de Package Manager > Unity Registry:



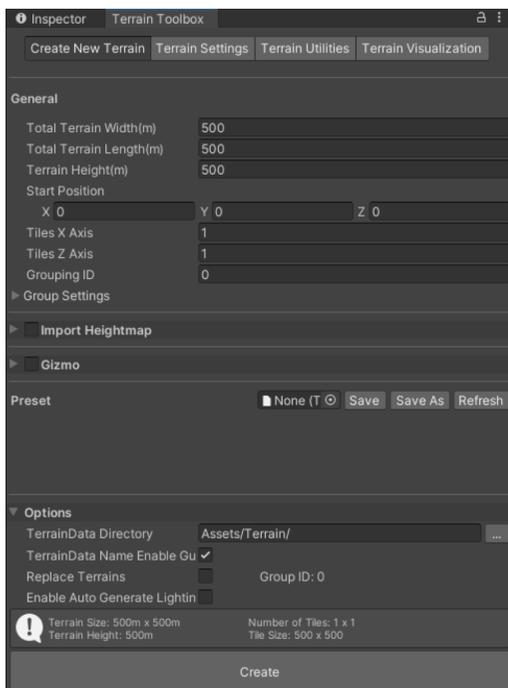
Una vez instalada, tendremos la opción en el menú de Window > Terrain > Terrain Toolbox, que nos permitirá crear terrenos nuevos (los creará dentro de un grupo de terrenos) o editar los que ya están creados a través de un panel propio.

NOTA: Una buena opción es descargar [ejemplos de terrenos](#) con herramientas adicionales en la Asset Store:



AÑADIENDO TERRENOS A LA ESCENA

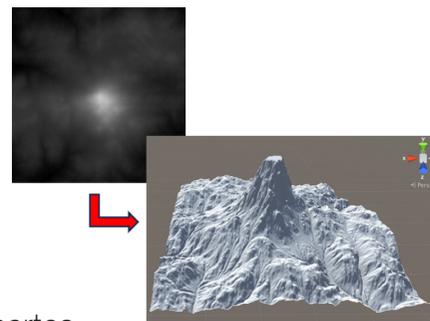
Usar el paquete "Terrain ToolBox" para crear terrenos nos permite configurar sus dimensiones, su posición, o la carpeta en la que se guardarán los datos vinculados al terreno. Cuando lo tenemos todo listo, pulsamos en "CREATE" y se añadirá el terreno y un Empty Object Padre en la escena, para agrupar los terrenos colindantes.



Un dato importante es el "Grouping ID", ya que es lo que garantiza que dos terrenos están "unidos". Para añadirlos, es preferible usar la herramienta de expandir terrenos colindantes que veremos más adelante.

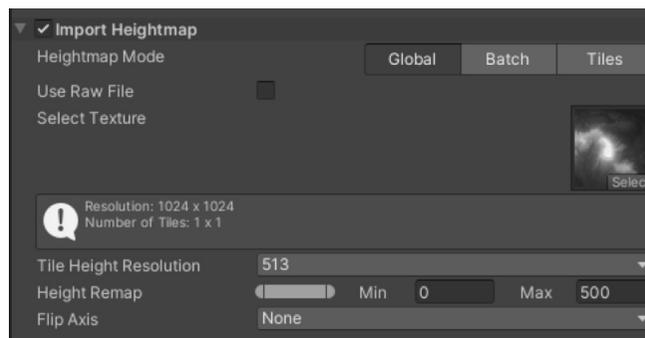
Mapa de alturas

Podemos crear terrenos a partir de imágenes bidimensionales, los llamados mapas de altura (height maps).



Unity convertirá las partes luminosas de la imagen en zonas elevadas, y las partes oscuras en zonas bajas del terreno.

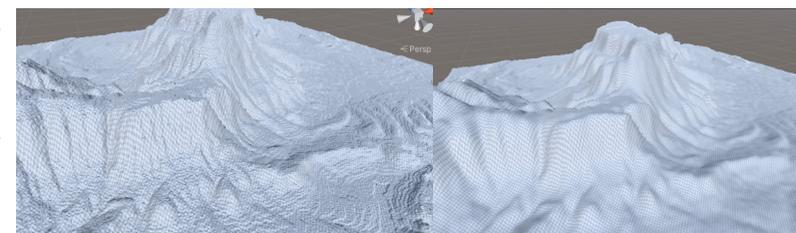
Si activamos la casilla "Import Heightmap" del Terrain ToolBox podremos usar una imagen en blanco y negro para crear el mapa.



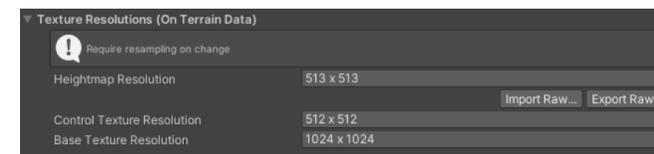
En la opción de "Height Remap" podemos indicar la altura que tendrán las zonas blancas y la que tendrán las negras.

NOTA: cuanta más información tenga la imagen en su escala de grises, mejor interpretará

las diferencias de alturas. Por eso se recomienda usar archivos RAW de 16 bits por canal (Photoshop permite guardar como "Photoshop RAW", en formato PC). Si el archivo no tiene calidad suficiente, es posible que salgan cambios bruscos de altura en forma de terrazas. Podemos evitarlo bajando la resolución de importación (Tile Height Resolution).



También podemos importar un mapa de alturas en un terreno ya creado, en la pestaña de configuración del terreno, buscaremos el apartado de Texture Resolutions, y pincharemos en el botón de "Import Raw".

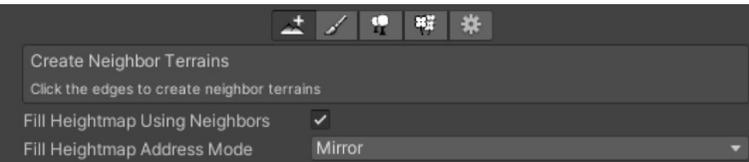


Existe también la opción de exportar un mapa de altura de un terreno que hayamos creado nosotros, por lo que podremos reproducirlo en otro terreno.

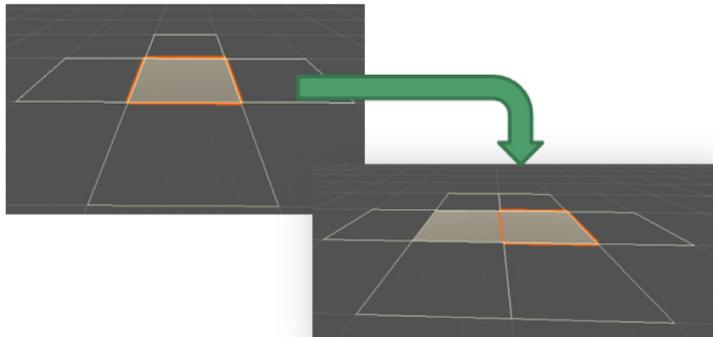
EXTENDER EL TERRENO

Vamos a ver las diferentes opciones (pestañas) que aparecen en el inspector cuando seleccionamos un terreno.

La primera opción que nos encontramos en las pestañas del componente "Terrain" es la de añadir terrenos colindantes.

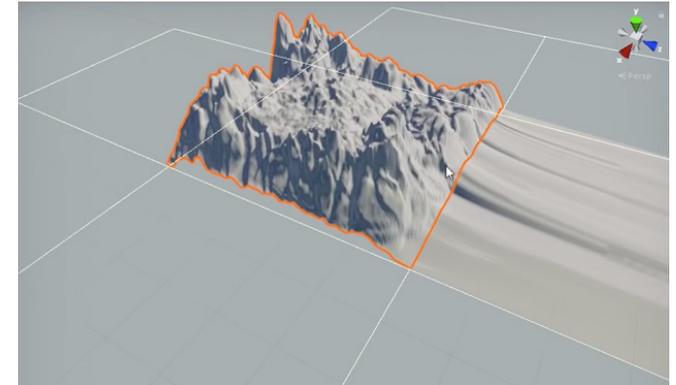


Al activarla, nos mostrará en la escena los 4 lados de nuestro terreno a los que podemos añadir uno nuevo. Si hacemos click en cualquiera de esos cuatro cuadrados señalados, aparecerá automáticamente un terreno nuevo.

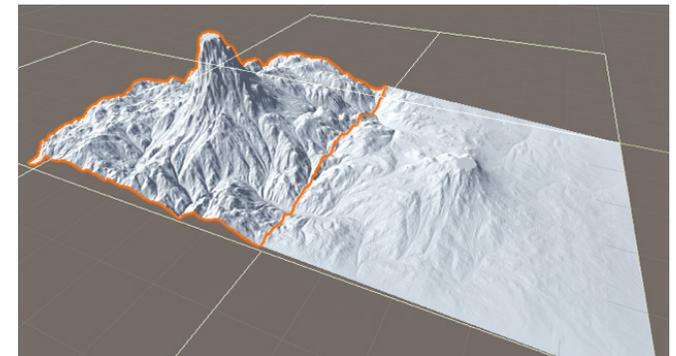


Si tenemos marcada la casilla de "Fill Heightmap Using Neighbors", el terreno colindante tomará las elevaciones del borde para continuarlas de dos maneras:

1. En la opción de "clamp" las irá suavizando
2. En la opción de "mirror" creará un duplicado que tenderá a aplanarse



Si activamos la casilla de "Auto-conect" en terrenos que están en el mismo ID, en la pestaña de configuración del terreno, al modificar uno en la zona de contacto se modifica el otro.



Si queremos que se comporten como un solo terreno, deben compartir el mismo ID, que se encuentra en la configuración del terreno, en "Grouping ID". Además, deberán tener la misma resolución

Antes de seguir: configurando los parámetros de los pinceles (fuerza, tamaño, rotación espaciado y dispersión) obtendremos el resultado deseado. Pulsa la tecla y desplaza el ratón a izquierda y derecha para ajustarlos (también puedes importar tus propios pinceles en "New Brush e importando una imagen):

"A": ajustar la fuerza del pincel .

"S": ajustar tamaño del pincel.

"D": ajustar rotación.

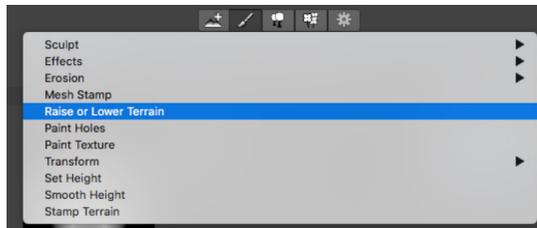
"Ctrl": invierte el efecto del pincel.

"Shift" activa la herramienta de suavizado.

ESCULPIR EL TERRENO

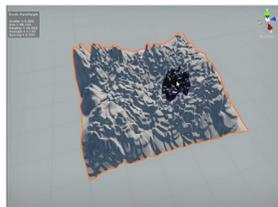
En la pestaña de pinceles tenemos varias opciones para modificar la orografía del terreno trabajando con pinceles.

Veamos ahora diferentes herramientas disponibles para esculpir el terreno:



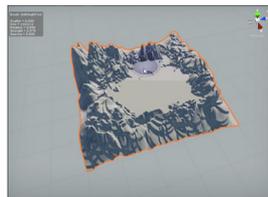
Raise or Lower Terrain

Crea elevaciones en el terreno. Como todas las herramientas, si pulsamos Ctrl se invierte la función (en lugar de elevar, hunde el terreno).



Set height

Ajustando la altura (height) del pincel, pondremos una altura homogénea a toda la zona del terreno.

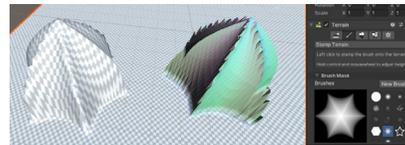


Smooth Height

Suaviza las irregularidades del terreno. Una herramienta muy habitual, por lo que se accede también pulsando la tecla "shift". Como todas, es bueno ajustar una fuerza adecuada

Stamp Terrain

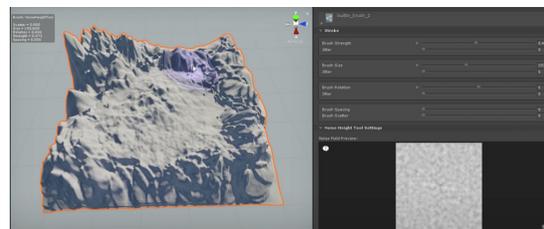
Configurando la altura, al hacer click se "estampará" una vez la forma del terreno que tengamos configurado en los pinceles. Esto nos permite configurar nuestros propios pinceles con formas complejas y añadirlos al terreno de forma fácil.



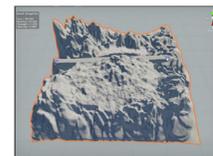
Sculpt

Permite opciones adicionales:

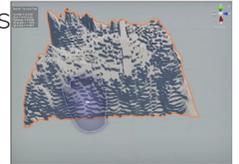
- **Noise:** agrega ruido al terreno. Se puede ajustar los parámetros de ese ruido



- **Bridge:** seleccionando un punto de inicio al hacer "Ctrl. + Click" y uno de final, crea "caminos" con la forma del pincel seleccionado.



- **Terrace:** divide las alturas del terreno en "terrazas".

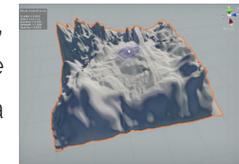


Effects

Permite varias funciones: aumentar diferencias (contrast), afilar picos (sharpen peaks), suavizar pendientes (slope flatten).

Erosión

Simula erosión de tres formas, por agua, viento o calor. Tiene parámetros propios bajo la opción de "Simulation Scale"



Mesh stamp

Debemos seleccionar una malla de nuestro proyecto, y "estamparla" en nuestro terreno.

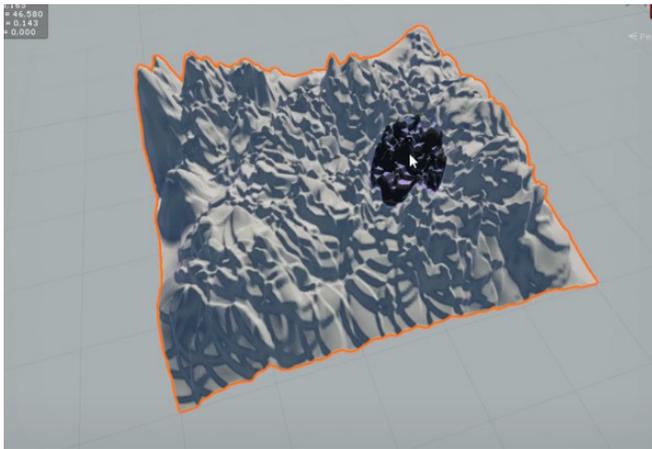
Transform

Herramientas precisas para modelar el terreno, como pinzar, retorcer, difuminar...

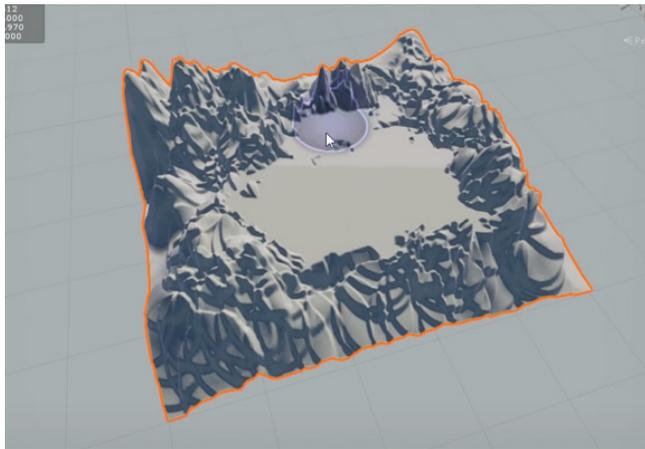
Paint holes

Permite abrir agujeros en el terreno donde colocar nuestras propias mallas, por ejemplo, una cueva.

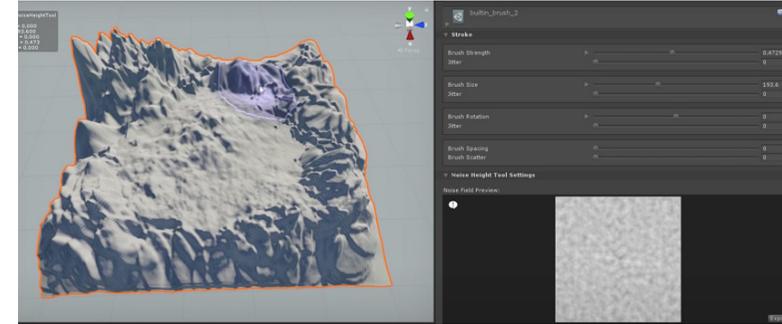
ESCULPIR EL TERRENO (ejemplos)



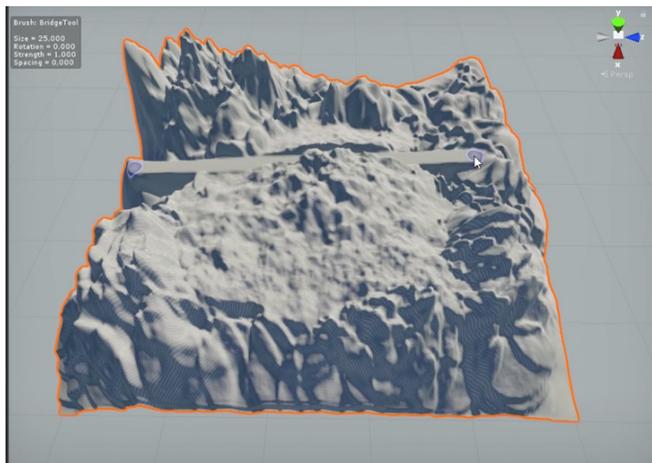
Rise or lower Terrain



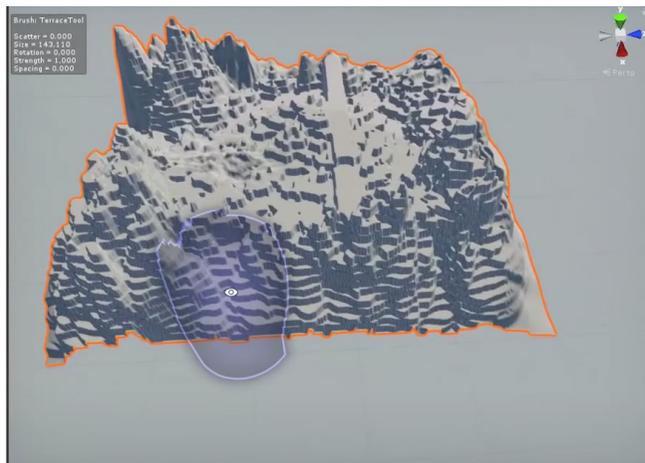
Set height



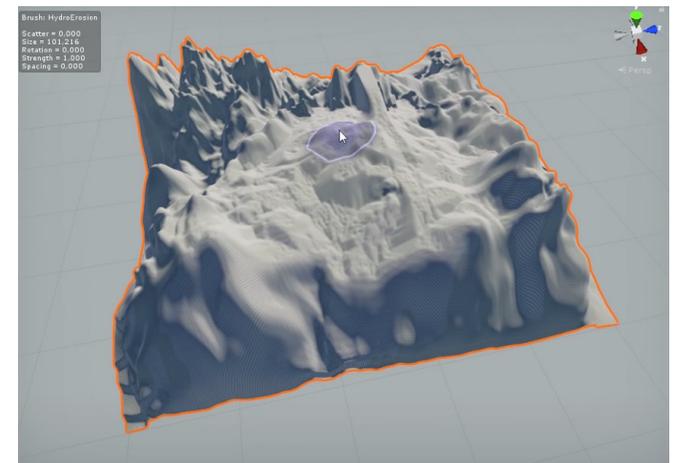
Noise



Bridge



Terrace

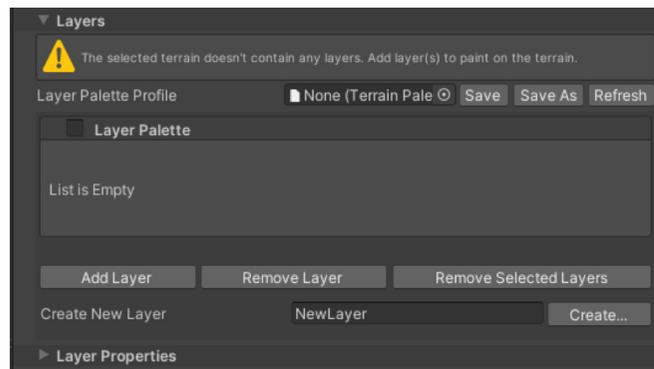


Erosion

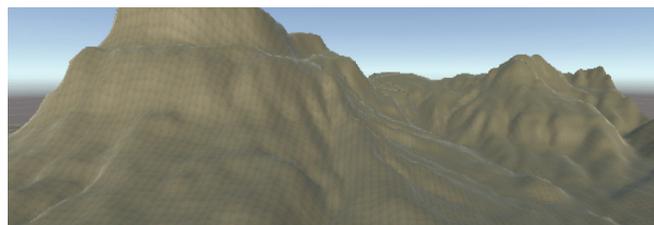
AÑADIR TEXTURAS

Lógicamente, nuestro terreno tiene que tener una textura -o conjunto de texturas- que le den realismo. Para ello se utiliza la herramienta de "Paint Texture" de la pestaña de pinceles.

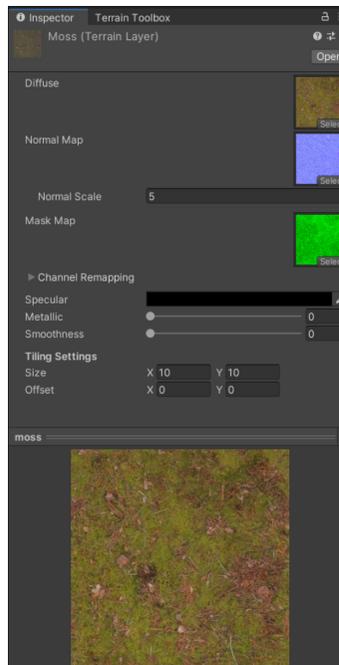
Para poder aplicar una textura, debemos añadir al terrenos al menos una capa. Para ello, seleccionamos en la herramienta de pincel, y en el menú de "capas" elegimos un nombre para nuestra capa y pulsamos en "Create" (también podemos hacerlo directamente en el proyecto Create. > Terrain Layer). Si ya tenemos capas creadas, podemos añadir las en "Add Layer".



Nos pedirá asignar una imagen que usará como textura Diffuse. Al ser la primera capa, la tomará como la textura base del terreno:



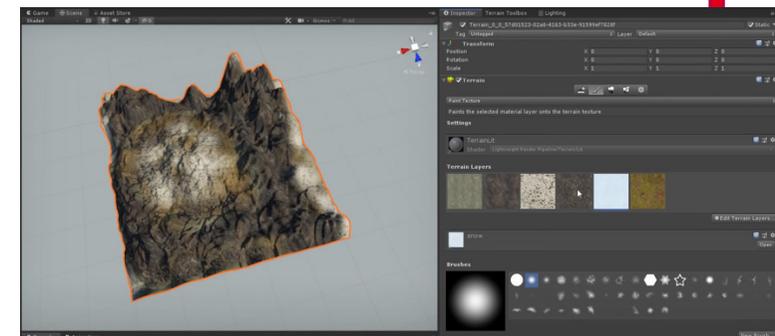
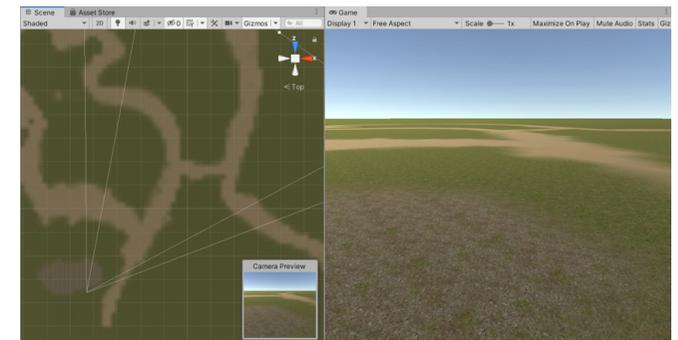
Podemos ir añadiendo más capas, creadas por nosotros o importadas.



Esas capas las podemos editar de forma independiente si las seleccionamos en el proyecto, añadiendo mapa de normales o incluso de máscara, además de ajustar ciertos parámetros del material como si de una textura cualquiera se tratara.

La combinación de mapas y parámetros de la textura determinará su realismo.

Cuando "pintamos" sobre una capa, determinamos las zonas que se ven de esa textura. El nº de capas máximo recomendado está entre 4 y 8, dependiendo del método de compilación.

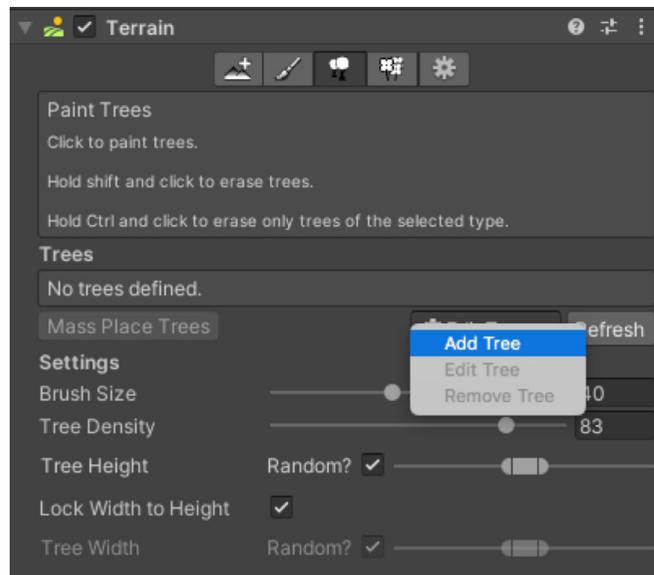


AÑADIR DETALLES: ÁRBOLES

Una vez “pintado” con texturas nuestro terreno, es el momento de añadir algunos detalles para darle realismo.

En los terrenos se pueden “pintar” árboles igual que se pintan elevaciones o detalles.

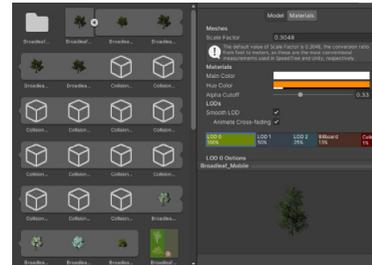
Si vamos a la pestaña de “árboles” en nuestro terreno, antes de comenzar a pintar debemos añadir un tipo de árbol (Edit Trees > Add Tree):



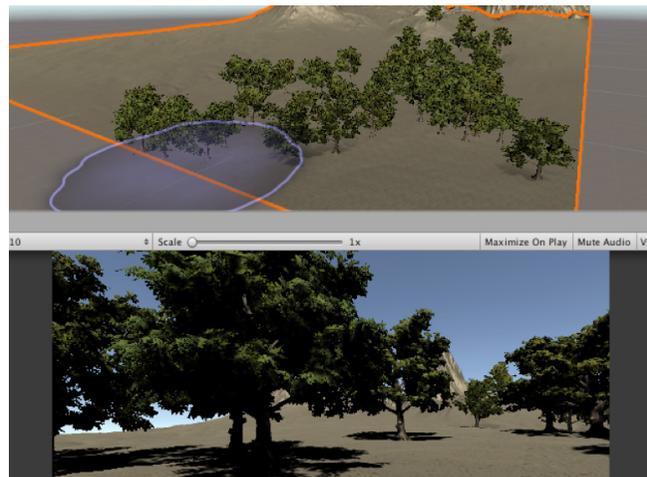
En la ventana que se abre de “Add Tree” deberemos buscar el árbol en la casilla de Tree Prefab y pulsar en “Add”, ya que los árboles son un tipo específico de prefabs.

Los árboles podemos editarlos de forma independiente a través de su ventana de inspector,

en el proyecto. Verás que tiene un gran número de opciones vinculados al prefab que contiene el árbol.



Ahora, ajustando el tamaño del pincel, y la densidad de árboles, podemos añadir el tipo de árbol seleccionado. Podemos ajustar otros parámetros como la aleatoriedad de la altura, el giro o el color:



TRUCO: Para borrar árboles se pintará pulsando la tecla “shift” (añadiendo “Ctrl” si solo queremos borrar los árboles del tipo seleccionado).

Una opción es la de añadir árboles de una vez a todo el terreno mediante el botón “Mass Place Trees”.

Si queremos que los árboles no puedan atravesarse, debemos activar “Enable Tree Collider” en el componente “Terrain Collider” del terreno.

Una alternativa es crear tus propios árboles. Unity permite hacerlo de dos maneras:

1. Usando la herramienta de “Speed Tree”, un software de terceros disponible para PC y para MAC
2. Mediante el “Tree Editor”, que partiendo desde un tronco sencillo (Game Object > 3D Object > Tree) podemos ir añadiendo elementos de forma nodal. Este método no soporta LOD's.

Unity cuenta con escenas completas de terrenos que puedes descargar gratuitamente. Eso sí, están realizadas para [Universal Render Pipeline](#) y para [High Definition Render Pipeline](#) (que los veremos más adelante). Pero puedes descargar los paquetes (cuidado que pesan bastante) e importar solo lo que desees:

AÑADIR OTROS DETALLES

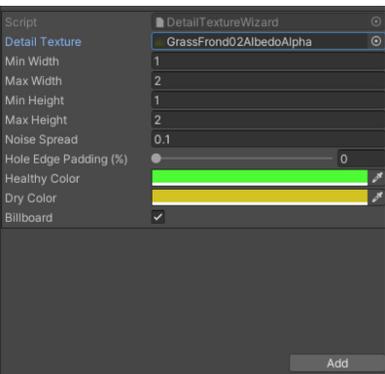
Hierba

En la última pestaña de edición de terrenos nos permite añadir elementos pequeños, como rocas o hierba, que doten de mayor realismo a nuestra escena

Para añadir hierba deberemos pulsar en "Edit Details" y "Add Grass Texture".

A continuación, nos pedirá que añadamos alguna textura.

Podemos encontrar tanto texturas de hierba como de detalles en la Asset Store de unity.



Antes de añadirla, podemos configurar algunos parámetros como color, tamaño, etc.

Si activamos la casilla de "Billboard" nos aseguramos de que siempre esté mirando a cámara.

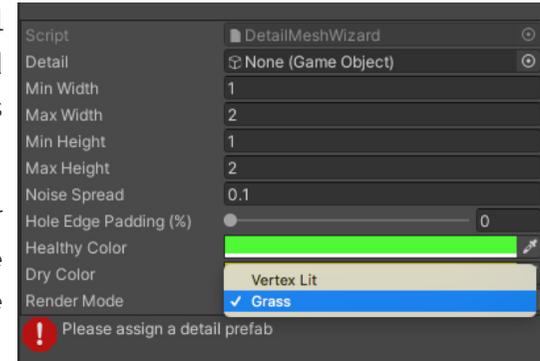
Ahora solo tenemos que ajustar los parámetros del pincel y pintar estos detalles por el terreno:



Rocas y detalles

Para añadir detalles como rocas, el proceso es el mismo, seleccionando "Add Detail Mesh", pero en lugar de texturas deberemos elegir prefabs.

Adicionalmente, en la ventana para añadir el elemento, nos aparecerá la opción de Render Mode (en lugar de la opción de Billboard), con dos opciones:



1. **Grass:** renderiza el elemento de una forma similar a la hierba, como una textura bidimensional.
2. **Vertex Lit:** renderizará el detalle como un objeto 3D. Más apropiado para elementos como rocas.

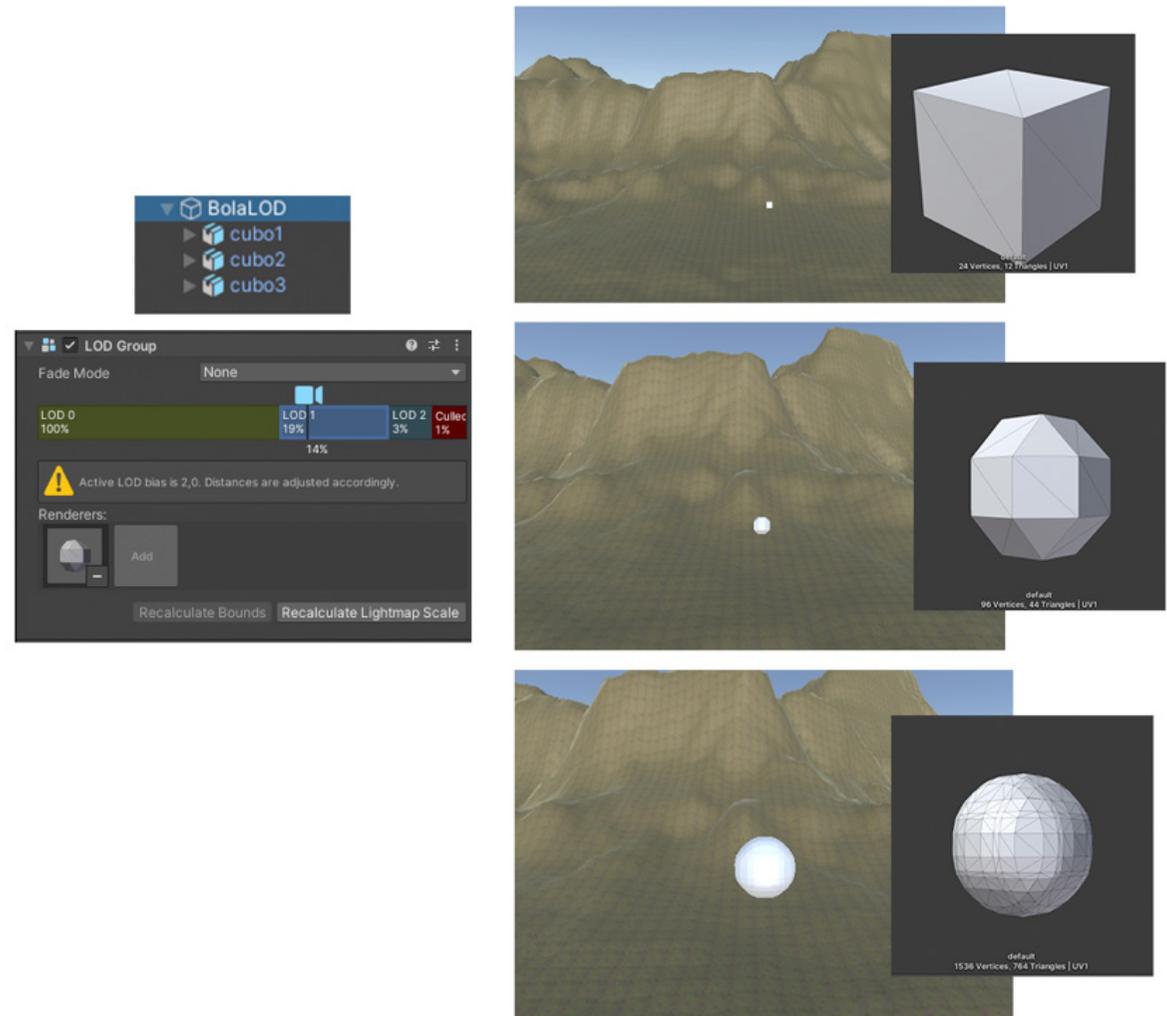
UNA PUNTE SOBRE LOS LOD's

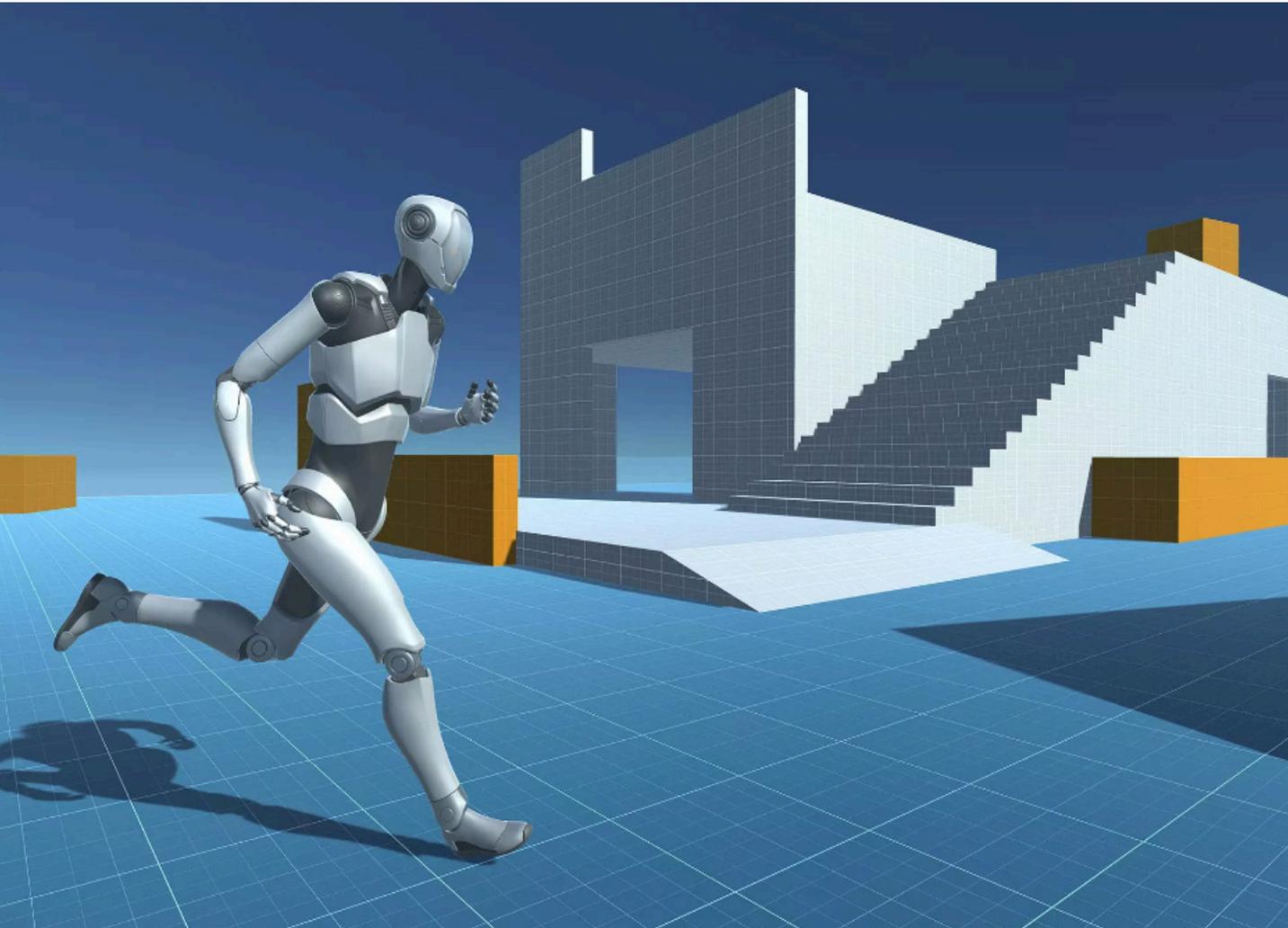
Como posiblemente sepas, cuando se modela un objeto para un videojuego, se suele hacer con varias resoluciones para que el motor de videojuegos pueda intercalar entre las mallas de menor número de vértices cuando está lejos y las que tienen más detalle cuando la cámara está cerca. Esto garantiza un mayor rendimiento.

Son los llamados "Levels of Detail" (LOD's), y Unity cuenta con una herramienta para configurarlos.

Si quieres comprobarlo, añade a una escena un Empty Object y pon como hijos las diferentes mallas con niveles de detalle. Incorporando el componente "LOD Group" podrás configurar a qué distancia "cambia" de malla, y qué mallas se ven en cada grupo.

En la siguiente imagen verás un ejemplo muy básico de una esfera que cuando se ve de lejos no es más que un cubo de 6 lados y 24 vértices (no necesita más detalle) pero cuando se acerca pasa a ser una esfera de 1500 vértices:





Controlando un personaje **3D**

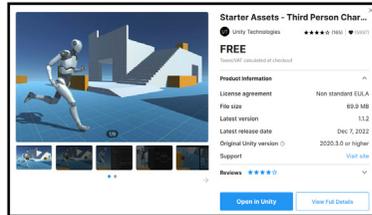
Vamos a seguir ampliando nuestros conocimientos del Animator Controller, y para ello controlaremos un personaje 3D.

IMPORTANDO UN PERSONAJE 3D

Aunque Unity incluye algunos paquetes que permiten riggear y animar elementos dentro del programa, lo normal es traerlo ya con las animaciones creadas en programas específicos para ello. Lo más estándar: importar un FBX

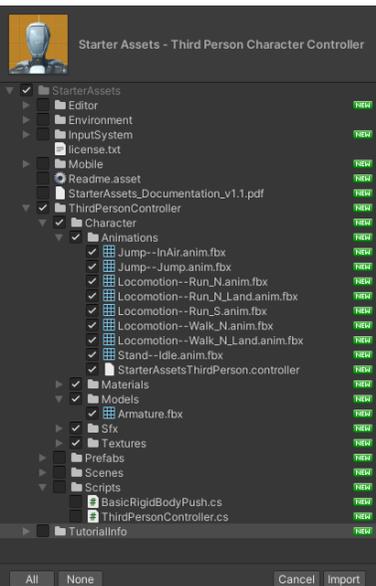
Una vez hemos creado nuestro personaje 3D, riggeado y animado, y hemos exportado esas animaciones (normalmente a .fbx) es el momento de importarlo a Unity.

Para estas prácticas, usaremos el [personaje 3D](#) que tenemos disponible en los Assets de Unity.



Importaremos solo lo necesario: dentro de la carpeta "Third Person Controller" encontraremos

las animaciones y el modelo, y opcionalmente los materiales, las texturas, los y los sonidos. Cuidado porque algunos elementos tienen dependencias con otros paquetes y entre ellos, así como avisos de compatibilidad. Consulta la documentación del asset para más información.



Ahora, arrastra el modelo "Armature.fbx" directamente al escenario (añade un terreno si quieres para que quede más realista):

NOTA: si no te aparecen las texturas, quizás tengas que editar los materiales que vienen en el paquete y añadirselos manualmente, o crear unos nuevos.

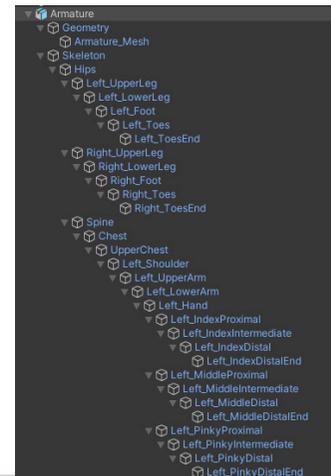


Al importar el FBX al proyecto, verás que tiene un icono específico:



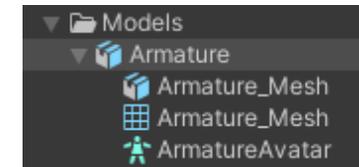
Como sabrás, los archivos .fbx son contenedores capaces de albergar múltiples mallas, a menudo organizadas de forma jerárquica, esqueletos, materiales e incluso animaciones. Funcionan de una forma similar a un prefab en Unity.

Si despliegas el FBX en la ventana de jerarquía, verás todas las mallas y sus jerarquías:

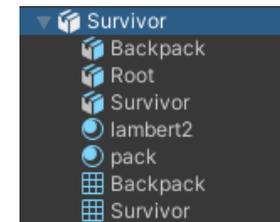


Puedes añadir objetos hijos (por ejemplo, un arma en la mano), pero no puedes cambiar esa jerarquía, ni borrar mallas, pero como cualquier otro prefab, puedes pulsar con el botón derecho al FBX en la jerarquía y marcar Prefab > Unpack Completely y dejará de ser dependiente del fbx original.

Si despliegas el FBX importado al proyecto, verás que contiene una malla y un Avatar:



NOTA: un fbx puede contener múltiples mallas, así como múltiples materiales, todos contenidos en el FBX. Aquí se muestra un ejemplo de FBX que contiene dos mallas y dos materiales:



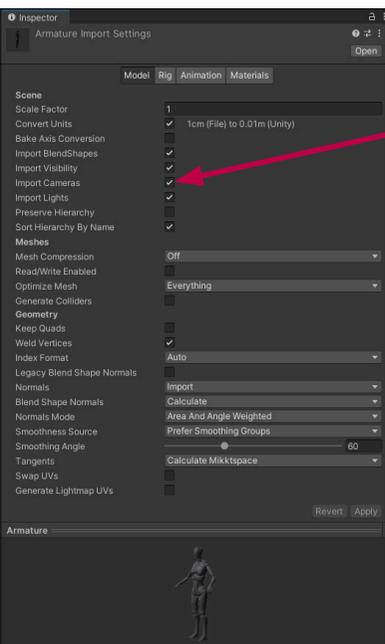
IMPORTANDO UN FBX (parte 1)

Recuerda que todos los elementos del FBX pueden configurarse en la importación, afectando de esa manera a cómo se comportan en la escena.

Si seleccionas el FBX en el proyecto, verás varias pestañas

Model

Si el modelador se ha equivocado a la hora de crear las unidades de medida, este es el sitio para corregirlo, ya sea indicando que el modelo se hizo en cms., y no en metros, o escalándolo en la importación para no tener que hacerlo en la escena, algo no recomendable.



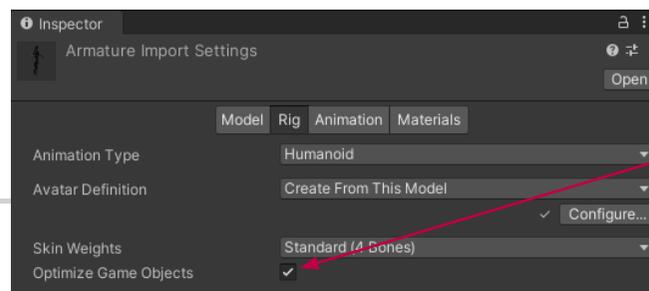
Otras opciones importantes son las de desactivar la importación de luces o de cámaras, por si el que ha creado el FBX se ha olvidado quitarlas al exportar (dos cámaras en la escena pueden provocar fallos inesperados).

Rig

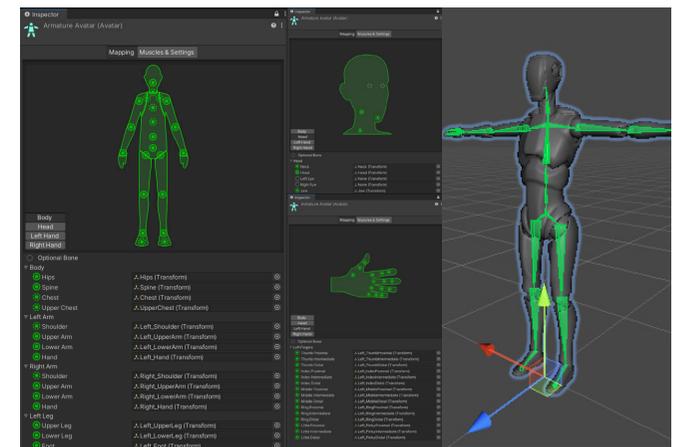
La configuración del Avatar, es decir, cómo debe interpretarse la distribución de los huesos de cara a una animación. Debemos indicar qué tipo de animación es, que en este caso es de Humanoide. En cuanto al Avatar, normalmente viene incrustada en el modelo. Si no es así, le podemos decir que lo cree a partir del propio modelo (generará un Avatar en el fbx), o importarlo desde otro modelo.

A menudo las animaciones no incluyen un Avatar, y por tanto no podemos ver cómo quedará en el modelo original. En ese caso, le decimos que lo coga desde otro modelo. Si te fijas, las animaciones importadas aquí usan el Avatar del FBX "Armature".

En este caso, el "Animation Type" es humanoide. Asegúrate de que las animaciones que le apliques más adelante tienen el mismo Rig, o se "desordenará" el esqueleto al aplicar la animación.

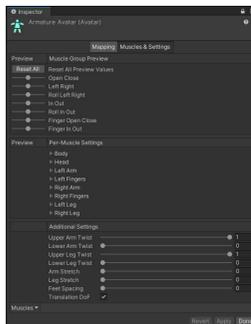


Si pulsamos en "Configure" se abrirá la herramienta Mecanim de Unity, donde indicamos a qué malla del modelo se corresponde cada hueso, del cuerpo, la cabeza o las manos. Veremos que cada hueso tiene su correspondencia en la jerarquía del objeto en la escena, e incluso podemos reasignarlos (tendremos que aplicar los cambios)



En la pestaña de "Muscles&Settings" puedes ajustar el rango de movimiento de cada hueso.

Si no vas a cambiar nada en la jerarquía del FBX, puedes colapsarlo al activar la casilla de "Optimize Game Objects", mejorando así el rendimiento.



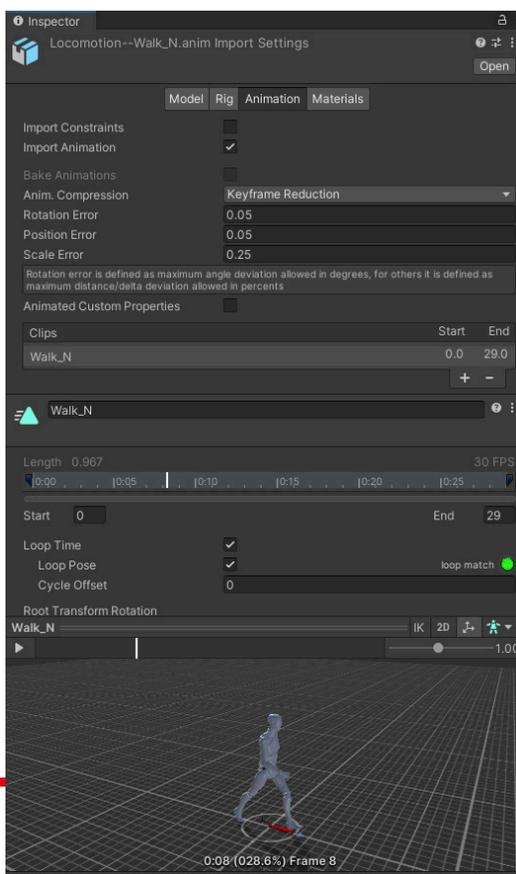
IMPORTANDO UN FBX (parte 2)

Animation

Si el FBX contiene animaciones aparecerán aquí.

Es en esta ventana donde puedes cambiar su nombre y sus parámetros, como duraciones, ponerlas en bucle, puntos de entrada y salida, velocidades, etc.

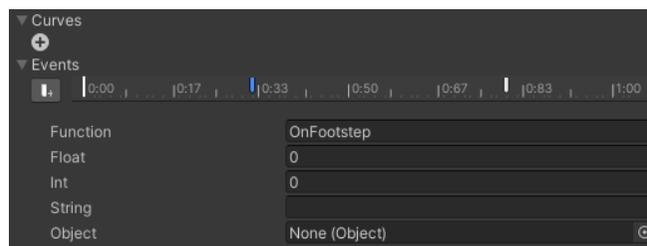
Como el FBX del modelo no tiene animaciones, puedes seleccionar las animaciones importadas para verlo:



En la parte inferior, puedes desplegar el previsualizador de la animación, e incluso moverte por ese entorno 3D. Verás que aparece un Humanoide estándar, pero si despliegas el icono del Avatar que aparece en la esquina superior derecha, y seleccionas "Other", puedes hacer que use otro modelo.

TRUCO: una misma animación puedes duplicarla en el proyecto y cambiar sus parámetros para tener dos animaciones distintas, por ejemplo cambiando sus puntos de inicio y de final.

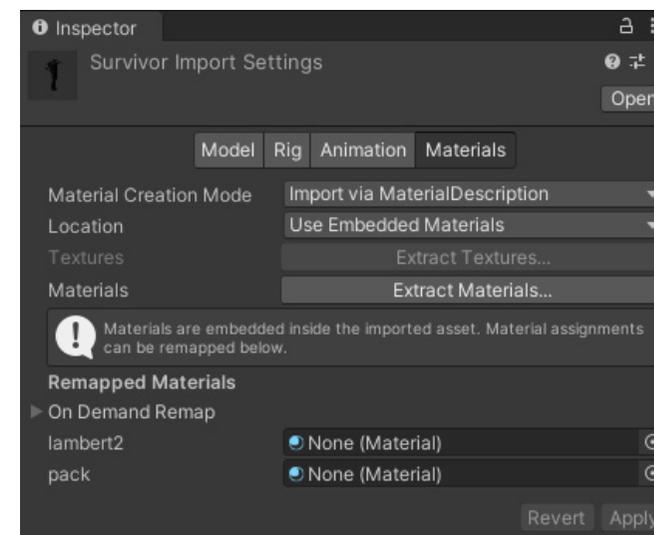
NOTA: algunas de las animaciones importadas tienen eventos asociados dentro de la animación, tal y como aprendimos en la unidad anterior. Al ser FBX de solo lectura (a menos que los desempaquetemos), no podemos cambiar la animación, pero sí podemos editar o borrar los eventos en esta pestaña, en el apartado "Events". Al seleccionarlos veremos la función que llaman y sus variables, y podemos borrarla. Si te fijas, son eventos pensados para ejecutar sonidos al caminar:



Materials

Aparecerán los materiales que incluye el FBX.

De nuevo, como no podemos modificar un FBX, si queremos editar esos materiales deberemos "extraerlos", y una vez fuera del FBX cualquier cambio que hagamos se aplicarán al original.

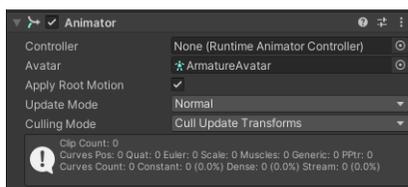


IMPORTANTE: cualquier cambio que hagamos en la configuración de los parámetros de un FBX importado deben ser guardados al pulsar en el botón "Apply", y se aplicarán a todas las "instancias" de ese FBX que haya en el juego (como un prefab).

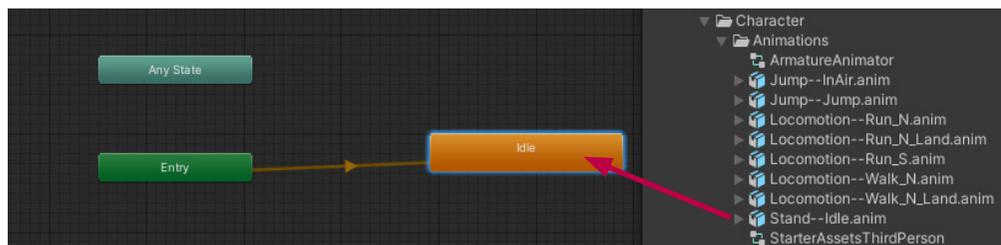
ANIMANDO UN PERSONAJE 3D

Si nos fijamos, el personaje Aramture en el escenario ya tiene un componente "Animator" (si no lo tuviera, se lo podemos añadir), pero no se le ha asignado ningún "Animator Controller". Lo haremos nosotros desde cero: en el panel de proyecto pulsamos Botón Derecho > Create > Animator Controller, le damos un nombre y se lo asignamos al componente Animator de Armature.

NOTA: entre los materiales importados, dentro de la carpeta de animaciones, encontraremos un Animator Controller ya creado llamado "StarterAssetsThirdPerson". Aunque podemos usarlo o consultarlo, es recomendable practicar creando uno desde cero.



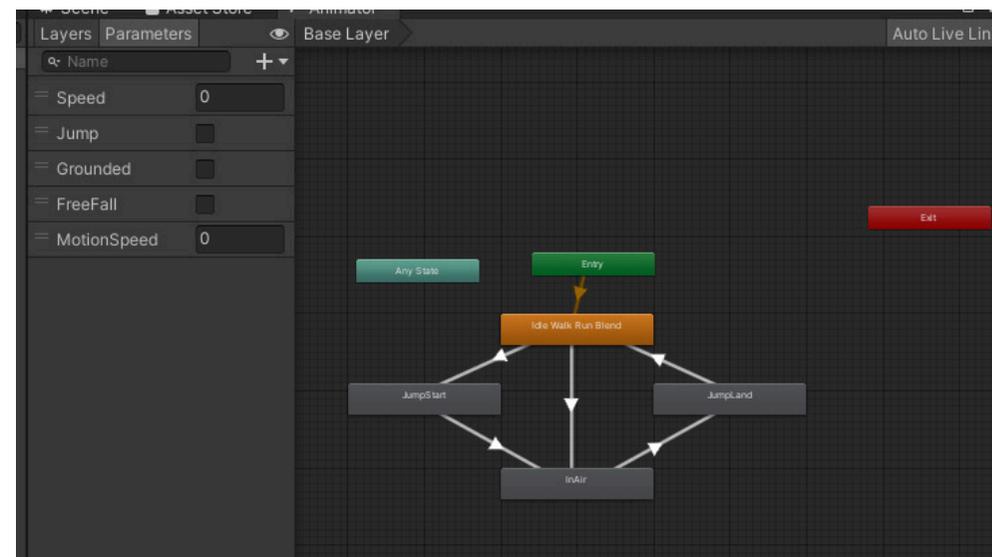
Ahora tenemos un Animator Controller vacío, listo para recibir los nodos de animación. Si arrastramos las animaciones importadas al panel del Animator, las reconocerá como nodos nuevos de animación, y la primera la asignará como el estado de entrada:



Acuérdate de asignar este Animator Controller al componente Animator de tu personaje en la escena.

Si ahora lanzamos el juego, veremos que el personaje inicia la animación de Idle por defecto.

Prueba a añadir más animaciones y crear transiciones entre ellas con sus correspondientes parámetros, igual que hicimos en la Unidad Anterior. Puedes "inspirarte" en el controller incluido en los materiales.



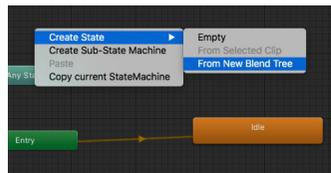
Verás que la transición entre unas animaciones y otras se crean interpolando las posiciones de los huesos en base a la duración establecida en la transición. Esto es posible porque las animaciones comparten Avatar, y es algo imposible de lograr en juegos 2D.

Es el momento de decidir qué animaciones tendrá nuestro personaje, y qué parámetros las controlarán, pero antes de hacerlo, conozcamos una herramienta muy útil: el [blend tree](#).

BLEND TREE

Hemo visto cómo cada nodo contiene una animación independiente, pero existe un tipo de nodo que permite almacenar varias animaciones a la vez, creando transiciones entre una y otra en base al valor de un parámetro. Son los Blend Tree.

Pulsamos con el botón derecho en el Animator Controller > Create State > From New Blend Tree



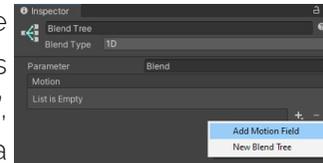
Crearé un nodo nuevo, que si hacemos doble click en él lo abrirá en un panel nuevo, donde mostrará un slider que cambia un parámetro float. Si no tenías ningún parámetro float, te creará uno llamado "Blend" que será el que usa, y que puedes cambiar por otro.



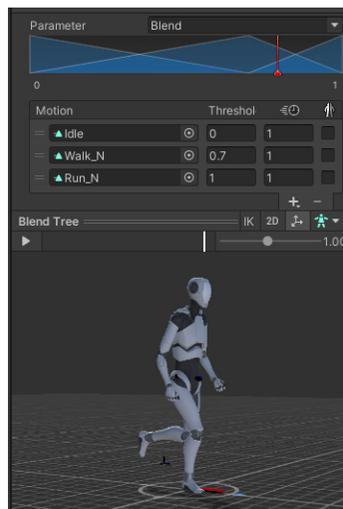
Al seleccionar el nodo creado veremos varias opciones:

- **Nombre del nodo.** Mejor cambiarlo para identificarlo mejor.
- **"Blend Type":** de momento usaremos un nodo de 1D (1 dimensión), pero debemos saber que existen Blend Trees de 2 dimensiones donde se puede controlar el estado del personaje en base a 2 parámetros (los 2 ejes de un joystick)

- **"Parameter":** el parámetro que gestiona el estado del nodo. Podemos seleccionar otro de los creados en nuestro Animator Controller.
- **"Motion":** en este panel podemos añadir "Motion Fields", cada uno vinculado a una animación.



Para ver un ejemplo de un Blend Tree funcionando, vamos a añadir 3 motion fields, y les vincularemos con las animaciones disponibles en los materiales de esta forma: **Idle - Walk_N - Run_N**. Para arrastrarlas deberás desplegar el FBX que las contiene.

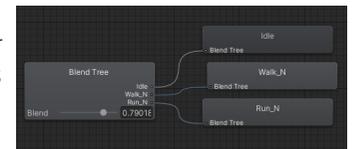


Desactiva la casilla de "Automate Thresholds" para poder introducir tus propios valores, y así ajustarlos a cómo quieres que se comporten o, llegado el caso, vincularlos a valores que vayan de -1 a 1 como en un Axis de un Joystick. En este caso, lo llevaremos de 0 a 1.

Si desplazas la barra roja, o el parámetro asociado (en este caso "Blend") verás cómo va pasando de una animación a otra.

Junto al parámetro de Threshold podrás cambiar la velocidad de ejecución de la animación (1 por defecto) y la posibilidad de hacerla en espejo.

Verás que en el panel aparecen las animaciones vinculadas:



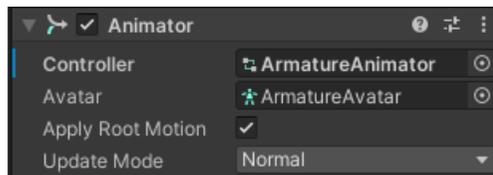
Prueba a dar al botón de "play" en el previsualizador de la animación (en la parte inferior del inspector) y cambiar el parámetro asociado al Blend Tree, pasando de 0 a 1.

Para volver al panel normal del Animator Controller, pulsa en el texto de "Base Layer" que hay en la parte superior del panel. Verás que ahora puedes crear transiciones al Blend Tree como si de otro nodo se tratara.

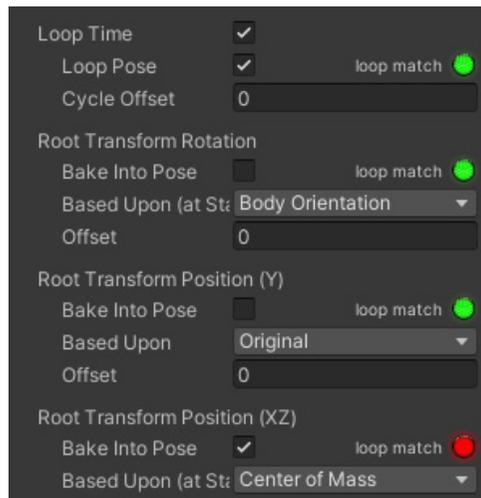
NOTA: existe otro nodo diferente, que es el Sub-State Machine, que básicamente nos permite agrupar una serie de animaciones en un solo nodo, como un Animator Controller dentro de otro. Es útil para organizar las animaciones en caso de tener muchos nodos.

ROOT MOTION

Algunas animaciones incluyen desplazamiento, algo que a menudo no es recomendable (por ejemplo para desplazarse por un terreno con desniveles). Para evitarlo, podemos marcar la casilla "Apply Root Motion" en el componente Animator de nuestro personaje.



También podemos bloquear las posiciones de la animación mediante las opciones "Bake Into Pose" que contienen en diferentes ejes.



MIXAMO

La página de Mixamo ofrece múltiples animaciones que podemos usar en nuestros proyectos.

Si subimos un modelo, nos permitirá buscar y descargarlas en formato fbx que podremos incorporar a nuestro controlador de animaciones

Es importante que una vez descargado, asignemos el tipo de Rigg "Humanoid", si es necesario, las pongamos en bucle.

Proceso:

1.- Acceder a la página web y registrarnos: <https://www.mixamo.com/>

2.- Ir al apartado de Rig->Auto-rigger y subir nuestro modelo. Requisitos:

- Debe estar en .obj o .fbx. En fbx debemos exportarlo con la opción

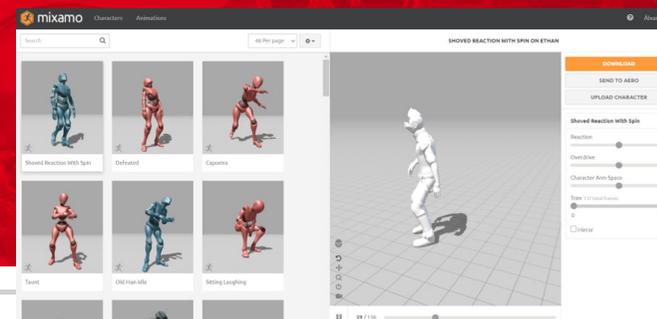
de "embed media" y el obj con la de "materials"

- Debe ser un bípedo y estar en posición de T ó A
- Debemos subir un ZIP con el OBJ, el MTL y las texturas

3.- Nos mostrará una pantalla donde ajustaremos la posición del modelo (debe tocar el suelo) y a continuación ajustar los puntos clave: barbilla, muñeca, codos, rodillas, ingle

- Se recomienda activar el uso de simetría.

4.- A continuación, nos mostrará el modelo con varias animaciones. Podemos descargar ya nuestro modelo animado, o ir a la opción de "Animate" para crear alguna animación extra (gratuita o de pago).



EJERCICIO

Como ya sabemos vincular los parámetros a entradas del Input System, así como crear transiciones entre nodos, vamos a diseñar nuestro primer Controlador de Jugado.



Revisa las animaciones que tienes disponibles, decide cuáles se ejecutan en un nodo sencillo y cuáles en Blend Tree. Aquí van unas pistas:

- Si hemos creado un Blend Tree para determinar si estamos quietos, caminando o corriendo, y tienes animaciones para caer y caminar, "caer y correr" y "caer sin más" (esta está dentro del mismo FBX que la de saltar), lo suyo sería que esas tres estuvieran en su propio Blend Tree conectado al ya creado.
- El paso de saltar y estar en el aire, así como el de caer y seguir caminando/corriendo, debería ser automático, es decir, sin condiciones (pero obligando a terminar la animación)

Piensa qué parámetros necesitarás para controlarlas y añade las entradas necesarias en el Input System para controlar esos parámetros. Ejemplo: podemos decidir que el personaje camina al mover el joystick izquierdo en su eje "Y", y gira al moverlo en su eje X. No hay animación para girar, pero podemos hacerlo por código, por ejemplo:

```
transform.Rotate(Vector3.up * rotate * rotationSpeed);
```

Recuerda que es bueno crear un parámetro de tipo Bool que indica si está tocando tierra, para pasar de salto a aterrizaje.

Crema las transiciones y las condiciones.

Es posible que para algunas transiciones necesites más de una condición. Y se activa la casilla de "Has Exit Time" donde no se necesite.

Crema un script llamado "PlayerController" en el que vincularás esos valores de entrada con los parámetros del Animator Controller, y comprueba si controlas bien a tu personaje.

PARA NOTA

Puedes buscar en Internet animaciones adicionales, por ejemplo para desplazamiento lateral ("strafe" en inglés). Revisa las configuraciones de la animación, quizás tengas que activar el "Loop Time", el "Loop Pose" y las opciones de "Bake into Pose" para que no se desplace.

Existen Blend Tree de 2 dimensiones. Ahora que tienes control de movimiento lateral y hacia adelante, ¿te atreves a gestionarlo todo con un solo Blend Tree de 2 parámetros? El primer parámetro gestiona la posición en X dentro del eje de coordenadas, y el segundo la posición en Y.

Este sería el aspecto de ese Blend Tree.

