

TFG

SHEEPY HELP. CREACIÓN Y DESARROLLO DE UN VIDEOJUEGO 3D.

**Presentado por Javier Aranda Godoy
Tutor: Maribel Pleguezuelos**

**Facultat de Belles Arts de Sant Carles
Grado en Bellas Artes
Curso 2017-2018**



**UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA**



**UNIVERSITAT POLITÈCNICA DE VALÈNCIA
FACULTAT DE BELLES ARTS DE SANT CARLES**

RESUMEN

En esta memoria de trabajo de fin de grado de tipología práctica, expongo mi participación del desarrollo de un prototipo de un videojuego 3D, *Sheepy Help*, creado conjuntamente por Sergi Adrover y Néstor Lázaro. Esta participación abarca muchos ámbitos como: *concept art*, modelado 3D, *rigging*, animación 3D, programación, y la integración de los recursos realizados a un motor multiplataforma como Unity 3D. *Sheepy Help* es un videojuego de un solo jugador orientado a PC, dónde nuestra pequeña protagonista es una oveja que deberá encontrar a su granjero para llevarlo de vuelta a la granja, y así, retornar la felicidad en ésta.

También hablaré de la documentación realizada y de la industria del videojuego actualmente en Valencia para profundizar más en mi proyecto de futuro laboral.

Palabras clave: videojuego, 3D, interacción, escultura, animación.

ABSTRACT

In this memory practical typology of the Final Degree Project, i show my participation about the development of my 3D videogame prototype, *Sheepy Help*, made jointly by Sergi Adrover and Nestor Lazaro. This participation covers lots of fields like: concept art, 3D modeling, rigging, 3D animation, programming and the integration of the full filed resources to a multiplatform tool, like Unity 3D. *Sheepy Help* is a videogame for a single player orientated for PC, were out little main character is a sheep which will must find her farmer to make him come back to farm, and then, to return the happiness in her.

I also talk about the documentation done and the factory of videogames that exists nowadays in Valencia to deepen in my project of future labor.

Keywords: videogame, 3D, interaction, sculpture, animation.

AGRADECIMIENTOS

En primer lugar, agradecer a mi familia por todo el apoyo que me han dado y las facilidades que he recibido a lo largo de estos cuatro años.

A mis amigos y compañeros de trabajo por todo el esfuerzo realizado durante todo el proyecto. Sin ellos no habría sido posible.

A Luis Alisandra por ayudarnos a entender mejor la lógica de la programación y facilitarnos recursos para ello.

A mi tutora Maribel y a todos los profesores con los que me he encontrado durante el camino que han sido pacientes y me han intentado ayudarme.

Gracias.

ÍNDICE	Pág.
1. Introducción	5
2. Objetivos y metodología	6
2.1. Objetivos	6
2.2. Metodología	6
3. Contexto de la industria del videojuego y referentes	7
3.1. Concepto de videojuego	7
3.2. La industria del videojuego en España	9
3.3. Entrevista a Carlos Escutia (CEO Frame Over)	10
3.4. Referentes en los videojuegos	14
4. Proceso de trabajo	17
4.1. <i>Briefing</i>	17
4.1.1. Historia	18
4.1.2. Personajes	18
4.1.3. Escenarios	18
4.1.4. Estética general	19
4.2. <i>Concept Art</i>	19
4.3. Modelado 3D	21
4.3.1. Granjero Fray	21
4.3.2. <i>Props</i>	22
4.4. Texturizado y Mapeado	23
4.5. <i>Rigging y Skinning</i>	24
4.6. Animación	27
4.7. Creación del primer nivel	28
4.8. Programación	30
5. Exportaciones e importaciones a Unity3D	32
5.1. Importación de modelado 3D y textura a Unity3D	32
5.2. Importación de animaciones a modelado en Unity3D	32
6. Resultados finales	34
7. Conclusiones	38
8. Bibliografía	39
8.1. Material Online	40
8.2. Cursos Online adquiridos	40

1. INTRODUCCIÓN

Debido a la complejidad que conlleva el Trabajo Final de Grado que aquí se presenta, ha sido necesaria la colaboración y participación activa de Sergi Adrover y Néstor Lázaro, ambos alumnos de la Facultad de Bellas Artes de San Carlos, Universitat Politècnica de València (UPV). En este sentido, y tras años trabajando conjuntamente en diversos proyectos de aula, durante el Grado, se consideró acertado seguir manteniendo esta dinámica de trabajo-colaborativo, no sólo para fortalecer las habilidades y conocimientos adquiridos durante los años de estudio en dicha facultad, sino también para potenciar los resultados en el proyecto (TFG). Es decir, se pretendía conseguir un proyecto más complejo y acabado que permitiera fomentar nuestro aprendizaje más allá de nuestras capacidades individuales.

Por tanto, dicho Trabajo Final de Grado se debe entender, fundamentalmente, como un acercamiento a los procesos y principios que conlleva la creación de un videojuego. Entender estos procesos en el desarrollo de un videojuego es importante, ya que en ellos se establece la planificación y la estructura que determinarán el conjunto de actividades necesarias para su correcta finalización; en este caso una demo para un videojuego 3D denominado, *Sheepy Help*.

Se tuvo que organizar y acordar la partición del trabajo en cada una de sus fases según el potencial de cada uno de los miembros. De este modo Néstor Lázaro se encargaría de una parte importante de programación y Sergi Adrover y el que aquí suscribe, de la estética visual, del diseño y desarrollo de escenarios, de las animaciones para la demo *Sheepy Help*, y una parte de programación esencial. (*Concept art*, modelado 3D, mapeado, *rigging*, *skinning*, animaciones y su importación a Unity 3D).

Sheepy Help, lo podemos definir como un videojuego realizado con Unity (motor de videojuego multiplataforma, disponible como plataforma de desarrollo para Microsoft Windows, OS X, Linux), para un jugador, en el que una oveja tendría como objetivo rescatar a su granjero para devolver la normalidad a la granja. Su estética general es *cartoon low poly*. Para ello, tendremos un escenario lleno de obstáculos con los que habrá que lidiar para llegar hasta el cuidador.

2. OBJETIVOS Y METODOLOGÍA

2.1. OBJETIVOS

Los objetivos principales de este trabajo son:

Diseñar, elaborar e integrar un conjunto de modelados 3D para un primer nivel de un videojuego 3D, adecuándolos para un motor de juegos como Unity3D, para conseguir una calidad gráfica visual general, encontrando la estética deseada.

Crear un personaje 3D y adecuarlo para poder utilizarlo en cualquier tipo de animación o videojuego con un esqueleto completo e intuitivo.

Integrar los conocimientos adquiridos durante la formación en el Grado de Bellas Artes para obtener un resultado final más firme.

Participar en las diversas disciplinas que contiene la acción de crear un videojuego para conocerlas más a fondo y desarrollar una buena base para un cercano futuro profesional.

2.2. METODOLOGÍA

A partir de una investigación del contexto, la industria del videojuego y los referentes, los integrantes del equipo nos reunimos para marcar unas pautas a la hora de realizar este trabajo donde cada uno tenía unas tareas asignadas. Se acordó que el videojuego iba a ser 3D, *low poly*, runner, y con colores muy distintivos para diferenciar sus profundidades y las partes que lo componían.

A continuación, se marcó el proceso de trabajo donde iban a diferenciarse en diversas partes:

El *concept art* de los personajes, se realizó digitalmente a partir de un *briefing* y se juntó el proceso de trabajo de todo el equipo.

El modelado 3D se realizó con Maya y Zbrush. En esta parte se tuvo que valorar la cantidad de polígonos que íbamos a utilizar para nuestros personajes principales ya que determinarían el estilo del videojuego y la cantidad de recursos que ahorraríamos para que el juego fuera lo más fluido posible.

El texturizado y mapeado se realizó con Zbrush, y se realizó para desplegar las coordenadas UV, donde se decidió por donde se iba a cortar el despliegue de la malla poligonal. En este punto, se pintó el personaje del granjero para poder extraerle el mapa de textura que le correspondía.

El *rigging* y el *skinning* se creó en el software de Maya. En este apartado se crearon los huesos y los controladores del personaje Fray para que pudiera conseguir una movilidad suavizada, y pudiera animarse de forma intuitiva y completa adhiriéndolo a su malla poligonal.

Más tarde, se animó un caminado básico con personalidad para darle movilidad a nuestro personaje, para añadirlo a nuestro videojuego 3D, *Sheepy Help*.

Finalmente, en el proceso de trabajo, se llevó a cabo la creación del primer nivel y la programación de éste mismo. Estos dos apartados se realizaron con la colaboración de Sergi Adrover Morant, integrante del equipo del videojuego. Aquí, se integraron todos los elementos 3D conseguidos para crear una estética general y funcional. Se programaron el movimiento del personaje principal, la cámara, los *colliders* y otros aspectos como los sonidos.

3. CONTEXTO DE LA INDUSTRIA DEL VIDEOJUEGO Y REFERENTES

3.1. CONCEPTO DE VIDEOJUEGO

Antes de abordar el TFG en toda su magnitud, es necesario abordar muy brevemente los orígenes del videojuego, su definición y concepto; siendo necesario referirnos a Johan Huizinga. Según el citado autor; el juego es una acción libre y ejecutada como tal, ubicada fuera de la vida corriente, pero que, a pesar de todo, puede absorber por completo al jugador sin que haya ningún interés material. El juego, se realiza en un tiempo y espacio determinado, desarrollado en un orden y ceñido a unas reglas que puede presentarse como la lucha por algo, o una representación de algo. También, cobra inmediatamente una sólida forma cultural que fomenta la sociabilidad y puede llegar a ser repetida a lo largo del tiempo. Unos de los primeros juegos encontrados a lo largo de la historia que se han podido interpretar como tal, son el *Juego Real de Ur* (2600 a.C.), interpretado como un juego de mesa, junto con el antiguo juego egipcio llamado *Senet*, del que existen registros posiblemente anteriores al siglo XXXIII a.C.



Juego Real de Ur (2600 a.C.)



William Higinbotham: *Tennis for Two*, 1958

No obstante, no será hasta la llegada de la cultura electrónica cuando se dé un salto importante que posibilitará la aparición de toda una industria, la de los videojuegos.

Como sabemos, el origen de la llamada cultura electrónica es motivado (en su más remoto origen), por la Revolución Industrial acaecida en pleno siglo XIX y posteriormente por los avances tecnológicos fomentados en pleno siglo XX, donde ya se establecieron las primeras relaciones hombre-máquina. El 25 de enero de 1947, Thomas T. Goldsmith y Estle Ray Mann patentaron un sistema electrónico llamado *Dispositivo de Entretenimiento de Tubos de Rayos Catódicos* (CRT) basado en los radares utilizados en la Segunda Guerra Mundial, convirtiéndose en el primer "programa" realizado para verse en una pantalla. Este dispositivo es un simulador interactivo de misiles en el que debemos ajustar la curva y la velocidad de los misiles para impactar en unos objetivos virtuales. Su mecánica jugable lo acerca a lo que sería un videojuego, pero no se le puede considerar como tal porque no mostraba ningún gráfico en pantalla.

En 1952, Alexander Shafto "Shandy" Douglas presenta su tesis doctoral en matemáticas en la Universidad de Cambridge basada en la interactividad entre humanos y computadoras. Dicha tesis incluía una versión computarizada del tres en raya llamado *OXO*.

Es el primer juego de computadora en usar una pantalla gráfica digital, y en teoría el primer videojuego de la historia, sin embargo, hay algunas voces que contradicen este hecho, ya que, prefieren considerarlo como un programa gráfico por ordenador ya que no tenía vídeo en movimiento.

El primero videojuego considerado como tal, fue *Tennis for Two*. El 18 de octubre de 1958, un físico estadounidense llamado William Higinbotham que participó en el *Proyecto Manhattan* (nombre en clave para el desarrollo de la bomba atómica) durante la Segunda Guerra Mundial, lanzó un programa que simulaba un partido de tenis a través de un osciloscopio del Laboratorio Nacional de Brookhaven. Este videojuego usaba una perspectiva lateral en la que se podía observar una línea horizontal que representa el campo de juego, y otra pequeña vertical en el centro del campo que hacía de red.



Comercialización del producto ODYSSEY de Magnavox

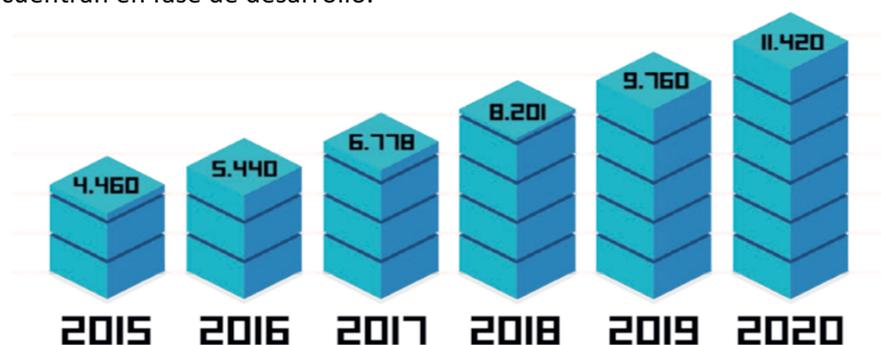
En 1973 llegó la Crisis del Petróleo y subió el paro, la inflación y el déficit público. El gobierno americano tuvo la solución de aplicar políticas de oferta, hacia el desarrollo de nuevas actividades y de nuevas tecnologías. En consecuencia, se lanzó la primera consola de la historia: ODYSSEY de Magnavox, diseñada por Ralph Baer, el padre de los videojuegos. A partir de este momento, la industria del videojuego se abrió paso para transformarse en una ventana abierta para la tecnología y el crecimiento de la misma para comercializar sus

productos enfocados a una sociedad lúdica. También, se crearán nuevos espacios dedicados exclusivamente para fomentar el consumo de videojuegos y sus novedades, llegando más tarde, a diferentes países del resto del mundo como España.

3.2. LA INDUSTRIA DEL VIDEOJUEGO EN ESPAÑA

El videojuego es un sector en constante crecimiento que proporciona muchas oportunidades de negocio para cualquier empresa, administración pública o inversor privado, según va creciendo la tecnología y las nuevas herramientas para poder realizar un buen desarrollo de ellos. Actualmente existen en España 450 empresas y estudios legalmente constituidos, sumándose otros 130 a la espera de constituirse. Las comunidades autónomas con el carácter más fuerte en este sector en España son: Comunidad Valenciana, Madrid, y en el punto más alto Cataluña, con una distribución territorial de empresas y estudios de videojuegos de un 27,7%.

Solo un 20% de las empresas y estudios activos a día de hoy existía hace 10 años. En estos últimos 5 años se ha creado el 52% de las empresas actuales, en consecuencia, se han creado una amplia base de microempresas que facturan menos de 2 millones de euros y representan el 87% de la masa empresarial. Algunas de ellas ni si quiera han empezado a facturar, ya que sus proyectos se encuentran en fase de desarrollo.

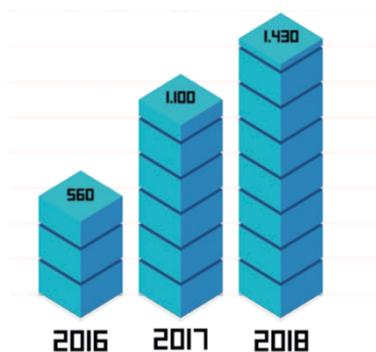


Evolución prevista del empleo en el sector (número de empleados)

La facturación de la industria española de videojuegos alcanzó en 2016 los 617 millones de euros, más de un 20% que en 2015. Las previsiones estiman que en 2020 se podría alcanzar los 1.440 millones de euros de facturación si la industria sigue en el mismo desarrollo. Así, el empleo directo creció un 20% en 2016, sumando 5.440 profesionales y su estimación en el 2020 son de 11.420 empleos directos.

El empleo de este sector está liderado por jóvenes de menos de 30 años que han cursado estudios superiores, siendo una gran ventaja para este gran negocio

que al parecer no se congela, sino todo lo contrario, sigue en constante crecimiento y avance. Sin embargo, la presencia de la mujer en la industria sí que está estancada. Sólo un 17% del total de los empleos directos son representados por la mujer.



Evolución de juegos publicados por empresas españolas (estimación)

Otro de los factores más importantes a mejorar en la industria, es la falta de financiación. el 91% del capital de las empresas de videojuegos procede únicamente de los socios fundadores. La venta digital es el modelo de negocio que más facturación genera de media, junto con los modelos de negocio *free to play*, que engloba a los juegos y aplicaciones móviles que los jugadores y usuarios pueden descargar y jugar sin necesidad de pagar, su monetización consta de la publicidad y de realizar compras dentro del juego.

La gran mayoría de empresas españolas desarrolla y crea juegos para ordenador y dispositivos móviles, teniendo de media total 2 juegos en desarrollo por empresa anualmente. En 2016 en España habían más de 900 juegos en procesos de desarrollo. Por otra parte, se ha apreciado un incremento de estudios que desarrollan juegos para consolas como *PlayStation 4* e *Xbox One*. También, el 40% de las empresas españolas ha trabajado y desarrollado proyectos para plataformas de Realidad Virtual (VR), y más de la mitad pretende hacerlo en un futuro, ya que, es un mercado muy prometedor, pero aún no se ha sabido explotar lo suficiente para que sea un negocio viable.

En definitiva y como hemos observado, la industria española del videojuego está en auge. Este incipiente crecimiento ha provocado el resurgimiento de un interés y de una necesidad personal que se ha manifestado en el trabajo que aquí presento. Interés por entender los mecanismos que lleva implícito dicha industria y los ecosistemas que la conforman. Es obvio, que crear un videojuego en un mercado en auge y tan competitivo es una tarea difícil, y entender los mercados lo era aún más, por lo que consideramos necesario profundizar más en el tema y entrevistarnos con profesionales del sector, para que nos ayudaran a comprender los por menores de la industria para así abordar el proyecto.

3.3. ENTREVISTA A CARLOS ESCUTIA (CEO FRAME OVER)

Este apartado, cabe decir que se realizó junto a Sergi Adrover Morant, y Jorge López Mollá.

Después de realizar un estudio y una búsqueda de las empresas de videojuegos y animación que podíamos encontrar en Valencia, decidimos ponernos en contacto con varias de ellas con la finalidad de concertar una cita para realizar una entrevista con las que más nos interesaban, y si cabía la posibilidad, ver el estudio por dentro. La finalidad de esta entrevista



Logotipo de Frame Over

principalmente consistía en sacarnos de muchas cuestiones tanto personales como profesionales y profundizar sobre el tema de la industria de los videojuegos y de sus estudios. Pudimos encontrar estudios de todo tipo, unos se centraban a fondo en los videojuegos, otros a cortometrajes de animación 3D, incluso al *stop motion*.

En este caso, Carlos Escutia, director ejecutivo del estudio Frame Over nos dio la oportunidad de concertar una entrevista.

La entrevista con Carlos Escutia (director ejecutivo de Frame Over) se realizó el 23 de marzo en Av. De Campanar, 39, València, a las 18:00h. La entrevista duró alrededor de 45 minutos, aunque el equipo estuvimos 1 hora y 30 minutos dentro del estudio de la escuela viendo las instalaciones, y cómo trabajaban los distintos profesionales junto a sus ordenadores. Le planteamos 14 preguntas a Carlos Escutia, pensando que esas preguntas eran las más adecuadas para profundizar y sacar conclusiones en el ámbito profesional de los videojuegos:

1. ¿Cómo se formó Frame Over?

“Frame Over nace a partir de esta escuela de animación, que es Primer Frame”. Con el tiempo, se vieron con la necesidad de formar Frame Over para aceptar ofertas de clientes para animaciones 3D, ya que, al principio solo se trataba de una escuela educativa, al mismo tiempo que los alumnos también buscaban trabajo relacionado con este sector: videojuegos, publicidad, producciones, etc. La escuela tiene 8 años y el estudio 5. Con proyectos propios para crecer como grupo y realizando services a clientes.

2. ¿Qué requisitos o conocimientos se necesitan para trabajar en una empresa como la vuestra?

“La gente que trabaja aquí tiene disciplinas bastante variadas”. Son una empresa mediana, oscilan entre los 15 profesionales. Hay diferentes facetas: la parte administrativa y el puesto medio. En el puesto medio se pueden incluir trabajadores como: modeladores, animadores, texturizadores, iluminadores y artistas de efectos especiales entre otros. La variedad es bastante amplia.

3. ¿Cómo está la industria del videojuego actualmente en Valencia?

*“Valencia es la tercera capital más grande en todo, y en este aspecto también”. En Valencia siempre ha habido mucha tradición artística donde se realizaban colaboraciones de series de animación como *Willy Fox* o *Los Pitufos*. Muchos artistas de *stop motion* también están aquí. En el ámbito del 3D, no hay tan buenas empresas como en Madrid o Barcelona, pero en Valencia es un buen*

momento, están en constante reproducción tanto en calidad como en estudios y hay muy buenas estadísticas.

4. ¿Qué tipos de proyectos habéis realizado? ¿Cuál ha sido el más interesante o el que más haya nutrido a vuestra empresa?



Frame Over: Título del videojuego
Maze of Gods

“Hacemos dos tipos de trabajo, *services* y *proyectos propios*”. Los servicios para los clientes pueden basarse en realizar el diseño de distintos personajes, los *concepts* de los mismos, diferentes animaciones de un solo personaje, efectos especiales, cortometrajes, posters para película, etc. En este factor, los más importantes pueden ser las colaboraciones con Motorola, Coca-Cola y la película *El Libro de la Vida*. Por la parte de proyectos propios, se basan en la creación de videojuegos, principalmente enfocan la empresa a este factor para crecer como empresa en esta industria, llevando a parte los servicios ajenos.

5. ¿En qué proyectos futuros os gustaría estar trabajando?

“Queremos luchar por hacer nuestros propios proyectos”. Como hemos comentado en la pregunta anterior, desean trabajar en el proyecto propio donde los diferentes artistas que participan se puedan expresar. Llevan dos años trabajando con su propio proyecto de videojuego: *Maze of Gods*.

6. ¿Aceptáis proyectos ajenos a Frame Over? ¿En qué os basáis a la hora de seleccionarlo?



Frame Over: Trabajando por capas el
videojuego *Maze of Gods*

“Principalmente son los *services*”. Buscar proyectos ajenos que les interese sobre todo económicamente, es la principal preocupación y que esté dentro de los márgenes de las cosas que saben hacer, ya que su principal motivación y sentimiento es llevar adelante su proyecto personal como empresa industrial. A la hora de escoger el *service* adecuado, repito, se basan en el beneficio económico y están abiertos a muchas empresas sin ningún inconveniente.

7. ¿Cómo tenéis distribuido el trabajo durante un proyecto?

“La organización a la hora de hacer un proyecto es uno de los factores más complejos”. Cuanto más tiempo trabajes en la preproducción, más fluido irá luego y menos cosas tendrás que repetir. No suele ser nada fácil organizarse rápido ya que cada *service* suele querer los resultados rápidamente. Afirma Carlos Escutia que está demostrado que hay que darle mucha importancia a la preparación de la idea, que conlleva al PNG, un archivo donde están todas las ideas, *concepts* e historias planteadas. Seguidamente la producción, que se basa en un calendario, unas fechas, un presupuesto y gente que sepas que van a terminar el proyecto. Previo a esto, obtener inversores y promotores de tu trabajo, teniendo claras las fechas de cuándo se va a lanzar.

8. ¿Qué programas utilizáis para desarrollar un proyecto?

*“Depende, nosotros los tocamos todos, depende sobre todo del cliente”. 3D Studio Max, Maya, Cinema 4D, Zbrush, Madbox, Adobe Photoshop, After Effect, cada uno para su finalidad sea de modelado, efectos especiales, *motion graphics*, entre otras. No todos los estudios obtienen el mismo material ni las mismas actualizaciones, en su escuela Primer Frame sí que disponen de éstas, y además están incorporando el programa Substance.*

9. ¿Cuánto tiempo se tarda en realizar un proyecto de un videojuego?

“Se puede llegar a tardar desde un mes de trabajo hasta dos años perfectamente”. Obtuvimos dos ejemplos para comparar y contestar la pregunta, los cortometrajes de animación y un videojuego. Un cortometraje de entre 7 y 10 minutos con una colaboración de 20 personas suelen tardar un año entero contando con la preproducción y la producción. En el caso de videojuegos, se puede hacer una aplicación móvil sencilla con un mes de trabajo, o se puede tardar incluso dos años para hacer un buen videojuego de alta escala.

10. ¿Qué puestos de trabajo podemos encontrar aquí?

“Cuanto más pequeño sea un estudio, más generalista ha de ser uno, más cosas ha de saber”. En la empresa y en la propia escuela se prepara a la gente desde varias disciplinas: modeladores, animadores, texturizadores, iluminadores, etc. La escuela Primer Frame va directamente unida a la empresa Frame Over.

11. ¿Cuál es el presupuesto medio que se invierte en un proyecto?

*“Hablando de proyectos propios, la inversión puede oscilar entre los 20.000 y 30.000 euros desarrollando un videojuego para el teléfono móvil, para uno de alta escala como *Maze of Gods*, unos 200.000”. Las primeras actividades y primeros proyectos son los más difíciles, hay que intentar que la inversión sea lo más eficiente posible, cubrir bien todos los aspectos y parámetros y no meter la pata.*

12. ¿Cuál es el sueldo medio de un ilustrador de concept, modelador 3D etc.? ¿Cómo se distribuye?

“Hay que tener en cuenta varios factores, el prestigio de la empresa, lo grande que sea a nivel industrial y la localidad”. La media de un trabajador en Frame Over oscila entre 1.500 euros, unos más y otros menos, depende de la persona y el nivel de profesionalidad.

13. ¿Qué consejo le darías a alguien que actualmente se está formando para poder ingresar en una empresa como la vuestra?

“Efectivamente, la industria actual demanda gente muy especialista, y de muchas carreras no se sale siendo un especialista”. Según Escutia, nunca tienes que parar y tienes que hacer algo, a partir que se termina la carrera, viene el momento complicado. A la empresa le suelen llegar entre 10 y 15 currículums a la semana y solo uno de ellos suele aparentar profesional, es necesario tener un recorrido demostrable. Lo más recomendable para gente que le interesa trabajar de esto, es muy importante tener una buena *reel*¹ y un proyecto final. Muy importante tener conocimiento de varias disciplinas y demostrar un buen trabajo en equipo, pero sobre todo mostrar un trabajo finalizado y no solo un currículum.

14. ¿Qué empresas en Valencia a parte de la vuestra, crees que son de gran prestigio?

“Actualmente en Valencia se hacen cosas muy chulas y no hay que dejar escapar ninguna oportunidad”. Cada vez mayor exigencia es lo que piden las grandes empresas. Los estudios que se pueden encontrar en Valencia son unos 20 que disponen de entre 5 y 10 componentes, pero también hay otras como Elite 3D que lo componen unos 40 o 50 trabajadores.

3.4. REFERENTES EN LOS VIDEOJUEGOS

Una vez contextualizado el marco actual en la industria de los videojuegos en Valencia era necesario abordar el proyecto tomando como referencia aquellos videojuegos que nos sirvieran como ejemplo para abordar el proyecto de TFG con toda su magnitud. Concretamente, citaré ahora aquellos que me han servido mucho para poder aportar la parte de creatividad necesaria en el videojuego *Sheepy Help*.

El primer referente a tener en cuenta es la saga de videojuegos de *Mario Bros*, creada por Shigeru Miyamoto. La mayoría de juegos de Mario han sido lanzados para el arcade o para consolas Nintendo, desde la *Nintendo Entertainment System* hasta la generación actual de videoconsolas, incluyendo un juego para móviles, *Super Mario Run*. Analistas sugieren que hacia 2012, la franquicia de Mario fue valorada en más de 10 mil millones de dólares.

¹ Currículo, portafolio o presentación en video realizado con la finalidad de promocionar el trabajo personal, proyectos o servicios.



New Super Mario Bros, 2006

He de indicar, además, que el juego en el cual puse toda mi atención, fue el *New Super Mario Bros*, lanzado para la Nintendo DS y sus variantes. Llegó al mercado en mayo de 2006 en EEUU y Japón, y en junio del mismo año, en Australia y Europa.

La estética del juego y los colores utilizados fueron determinantes para el proyecto que compartía, *Sheepy Help*. La estética es simple pero muy cuidada, avistando diferentes tipos de imágenes que engañan a la vista creyendo que estamos en un entorno 3D al 100%; y así, ahorrar la máxima cantidad de polígonos para que el videojuego vaya lo más fluido posible en la consola. Los colores del personaje principal son escasos y vistosos (contrastando con el fondo) para que el usuario/jugador no pierda la atención y pueda localizarlo rápidamente. Otro aspecto determinante en el juego que me sirvió de referencia fue la posición que tenía la cámara. La cámara podía registrar tanto el fondo (plano secundario) como todo lo que aparecía en un primer plano, sin tener que perder una buena visión en un videojuego de plataformas permitiendo predecir los siguientes movimientos en el juego; una cuestión muy importante que hace posible que sea fácilmente jugable.



Geometry Dash, 2013

Geometry Dash es un juego independiente de plataformas completamente gratuito. Fue creado por *RobTop* en 2013 para dispositivos móviles, y un año más tarde se lanzó a la plataforma de *Steam* (PC). Su personaje principal tiene forma de cubo y solo puedes saltar o dejarte caer (habiendo variantes), y el juego funciona únicamente con un comando, es decir, se puede jugar sólo con un dedo, cosa que lo hace muy cómodo de jugar a pesar de su dificultad. No se requiere completar un nivel para jugar al siguiente, aunque hay niveles que están bloqueados y deberán desbloquearse con un porcentaje de monedas superado. Generalmente la dificultad va aumentando según vas avanzando de niveles.

En este videojuego lo realmente interesante es la capacidad de atención que debe de emplear el jugador para intentar superar todos los obstáculos a la primera, ya que, el personaje principal está variando su posición sin parar, lo que hace que el juego sea divertido a la vez que permite potenciar ciertas habilidades y reflejos en los jugadores. Además, te permite variar de altura las plataformas del escenario con los saltos para “transportarte”, te lleva a otros lugares con sorprendentes colores ofreciendo una gran versatilidad estética que ameniza al jugador. De igual modo, la música empleada en este videojuego se sincroniza con él mismo, variando su tempo y melodía según el escenario que juegues y según los saltos y caídas que el personaje principal va realizando.



Crash Bandicoot, 1996

Otro referente es *Crash Bandicoot* de género de plataformas en tercera persona. Creado por Naughty Dog para la consola PlayStation lanzado oficialmente al público en 1996 en la Electronic Entertainment Expo. El protagonista del videojuego es Crash Bandicoot (como el título del juego indica). El jugador debe seguir un camino derrotando a enemigos, evitando hoyos y rompiendo cajas para detener los planes del doctor Neo Cortex. Su clasificación es PEGI 3, lo que hace que el juego sea jugable tanto como para personas de temprana edad, como personas más mayores.

El juego, aunque sea para un solo jugador, contiene niveles que tendrán que jugar con el personaje secundario Coco, la hermana de Crash. Este juego es interesante en tanto es definido y catalogado como muy ameno, alegre y divertido. Estas cualidades propias del juego son interesantes como objetivos a desarrollar para el proyecto de TFG *Sheepy Help*. Otro aspecto a tener en cuenta y menos afortunado en el juego es el uso de la cámara en algunos niveles: la cámara trasera del personaje resta visibilidad en cuanto al escenario y los obstáculos que tienes delante. En contraposición, la estética es sencilla y con pocos polígonos, con unos colores estudiados muy adecuados para el público al que se dirige, lo que hace que este juego acompañado de su música sea divertido y jugable, dejando un buen sabor de boca en cuanto a gráficos de la época. Actualmente, este juego está remasterizado para PlayStation4, y los gráficos han mejorado notablemente.



Temple Run, 2011

Uno de los referentes más notorios que se podrán apreciar en nuestro videojuego, *Sheepy Help*, es *Temple Run*. Es un videojuego interminable al estilo *runner* en 3D, catalogado como *endless run*^{*2}. Fue desarrollado y publicado por Imangi Studios. Fue lanzado en 2011 y está disponible en las plataformas de iOS, Android y Windows Phone 8. En el videojuego podemos apreciar la ubicación de la cámara que se sitúa en la parte trasera de nuestro personaje, así que nuestra visión siempre será la de ir hacia delante. La velocidad es *in crescendo* cuanto más rato estemos corriendo. En el escenario iremos cogiendo monedas que más tarde se podrán intercambiar en la tienda de objetos por ítems para nuestro personaje.

² En mi opinión, el estilo *endless run* hace que el jugador tenga cierto "pique" con su propio score o con otros jugadores y quiera vencerse así mismo una y otra vez, aunque esto hace que el juego no tenga un final determinado y te puedas cansar antes del juego, que otro videojuego separado por niveles donde los objetivos están marcados y estás deseando llegar al final del juego para saber cómo concluye la historia.

Por último, mencionar al videojuego *Tunic*, desarrollado por los estudios Finji. A pesar de que este título aún no está disponible para ser jugado, pudimos verlo en 2017 en una de las conferencias del e3³ de Xbox, donde nos enseñaron este juego lleno de acción y aventuras. Este juego *indie*, estará disponible en 2019 en las plataformas de *Steam* y de Xbox One X.

Este videojuego tan prometedor, posee de unos gráficos minimalistas muy trabajados con un estilo *low poly*. Sus gráficos fueron lo que más me llamó la atención, ya que tiene un encanto que es difícil que pasen desapercibidos. Sin embargo, solo pudimos ver un pequeño tráiler en el e3, y no nos contaron nada de la historia, solo de la parte jugable. Aunque podemos decir que, su personaje principal es un zorro con gran personalidad que tendrá que lidiar con una serie de acertijos en una civilización un tanto extraña.

4. PROCESO DE TRABAJO

4.1. BRIEFING

Antes de comenzar a dibujar y de proponer las primeras ideas sobre nuestro videojuego, debíamos de tener una historia principal y establecer cómo se realizaría el seguimiento de todo el proceso general (de todo lo que teníamos que realizar); de configurar un poco cómo serían los personajes y los escenarios que posteriormente se modelarían y se prepararían. Así, todos los integrantes de nuestro equipo nos pusimos de acuerdo en ciertos parámetros para tener unas reglas y normas a la hora de trabajar.

4.1.1. Historia

Nuestra historia empieza con uno de nuestros personajes principales, el viejo granjero Fray. Fray, vive en las afueras de *Rake Farm*, donde trabaja feliz la agricultura y la ganadería junto a su rebaño de ovejas. Un día como otro cualquiera, Fray sale del granero a buscar madera para las provisiones del siguiente invierno como todos los años para que sus ovejas estén en buena temperatura y no pasen frío. Después de muchos kilómetros recorridos buscando la mejor madera de *Rake Farm*, en una de las montañas que estaba repleta de árboles, uno de ellos se le cayó encima aplastándolo e imposibilitándole la movilidad.

³ *Electronic Entertainment Expo*, se realiza en Los Ángeles.

Los días pasaban en la granja mientras que el granjero continuaba atrapado en uno de los grandes árboles de las afueras de Rake Farm. Las ovejas que estaban en el granero tenían mucha hambre y estaban muy preocupadas por él. Ninguna de las ovejas sabía si volverían a verlo de nuevo. Así, la oveja más valiente e inteligente del granero, salió corriendo en busca del granjero Fray sin perder un ápice de tiempo, recorriendo diferentes paisajes llenos de dificultades y obstáculos peligrosos para rescatarle y devolver la normalidad a la granja.

4.1.2. Personajes

- Personaje principal. Sheepy

Sheepy es la oveja y el personaje principal de esta historia. Sus rasgos se caracterizan por ser un personaje bípedo, de estatura mediana, de lana esponjosa y de edad temprana/mediana. Su personalidad se distingue por ser valiente, ágil, veloz e inteligente. También es una oveja empática ya que arriesga su vida por salvar a las demás.

- Personaje secundario. Granjero Fray

Sus rasgos físicos se pueden considerar como una persona de edad media/avanzada, de estatura mediana/baja y muy campechano. Su personalidad es de: trabajador, bondadoso, empático y muy despistado.

4.1.3. Escenarios

- Escenario 1. Montaña con árboles

Aquí es donde Fray va a buscar la madera para sus ovejas, y donde le cae el árbol. Es un lugar frondoso; con muchos árboles.

- Escenario 2. Granero

Es el lugar donde se encuentran las ovejas y donde empieza la historia de nuestra protagonista Sheepy. Será un lugar que nos transmita melancolía, pero a su vez esperanza. Su diseño es rústico. Aquí podremos acceder a la selección de niveles del juego a través de la interacción con la puerta.

- Escenario 3. Escenario jugable

Este escenario es donde se desarrollará la acción: la aventura y la capacidad de poder controlar a la oveja para intentar llegar hasta el granjero, al final del

juego. En este primer nivel tendremos obstáculos que saltar y que agachar, y combinaciones que deberemos de estar pendientes para hacerlas correctamente a la primera. Deberá de pasar por 4 fases para completar el primer nivel:

- La primera es la zona del “tutorial” donde habrá un par de obstáculos donde empezaremos a dominar los controles y adaptarse a las físicas y al estilo de juego.
- La segunda es la parte subterránea, donde nos encontraremos a un nivel del suelo inferior, para aparentar que estamos en una especie de cavernas o de cueva.
- La tercera parte es la zona aérea. Aquí apreciaremos la ausencia del suelo y nos dará la sensación de estar jugando por las alturas habiendo muchas posibilidades de caerte al infinito.
- Y la cuarta y última parte, es la zona de los objetos punzantes. Dará la sensación de estar en una parte final del primer nivel habiendo una dificultad superior. Al final de este tramo nos encontraremos con el final del nivel intuitivamente.

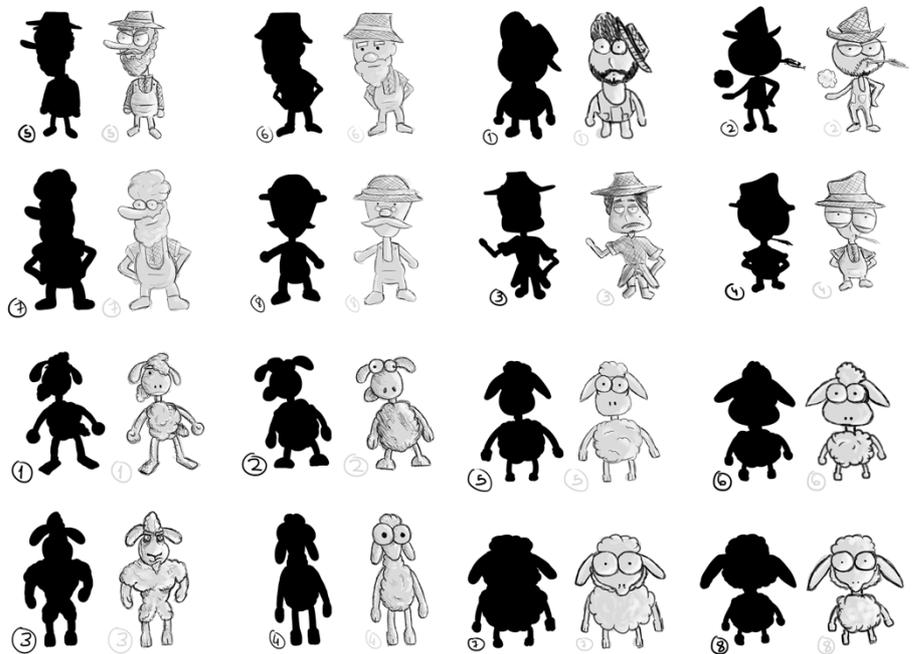
4.1.4. Estética general

La estética general de nuestro videojuego está basada en el uso de colores vivos e intensos, acompañados de una buena iluminación que permitiera ver los obstáculos inmediatamente como en *New Super Mario Bros*. También, nos pareció buena idea que lo que hubiera dentro de los escenarios fuera de bajo poligonaje acercándonos a la estética *low poly*, y así, podríamos ahorrar polígonos a la hora de procesar el juego y que fuera más fluido. Nuestra intención era que todos los componentes que aparecieran en la pantalla tuvieran una relación entre sí, sin que hubiera alguno que llegase a influenciar negativamente en relación a lo visual.

4.2. CONCEPT ART

En este apartado donde participamos los tres componentes del grupo, Sergi Adrover y Néstor Lázaro, mostraré las primeras ideas dibujadas de los dos personajes principales de nuestro videojuego. Para empezar, primero realicé una serie de *thumbnails*⁴. Así, si nosotros reconocemos la figura con una mancha y es funcional, podremos dibujar a partir de ella y empezar con el *concept art*.

⁴ Los *thumbnails* son una serie de imágenes que se usan para el reconocimiento de una figura a través de una mancha.



Concept arts y thumbnails de los dos personajes principales: Sheepy y Fray

Hicimos tanto variaciones del granjero como de la oveja para empezar a buscar una estética común, y al final de todas las que hicimos tuvimos que elegir uno de cada personaje. Hice varias pruebas de color para buscar el adecuado y que tuvieran relación los dos personajes entre sí, y que no distaran de un buen resultado final. Así, junté los dos personajes y los coloqué uno al lado del otro para buscar un resultado final dentro del *concept art*.



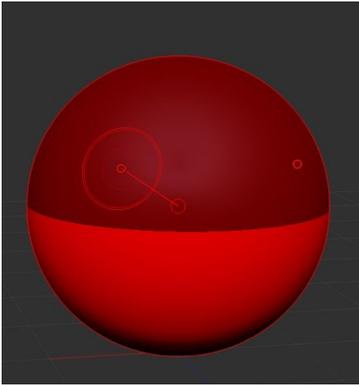
Pruebas de color de los dos personajes elegidos.

Concept art final de los personajes principales Sheepy y Fray

Una vez finalizados los modelos principales para empezarlos a modelar, realicé dos vistas del personaje a modelar: vista frontal y vista de perfil.

4.3. MODELADO 3D

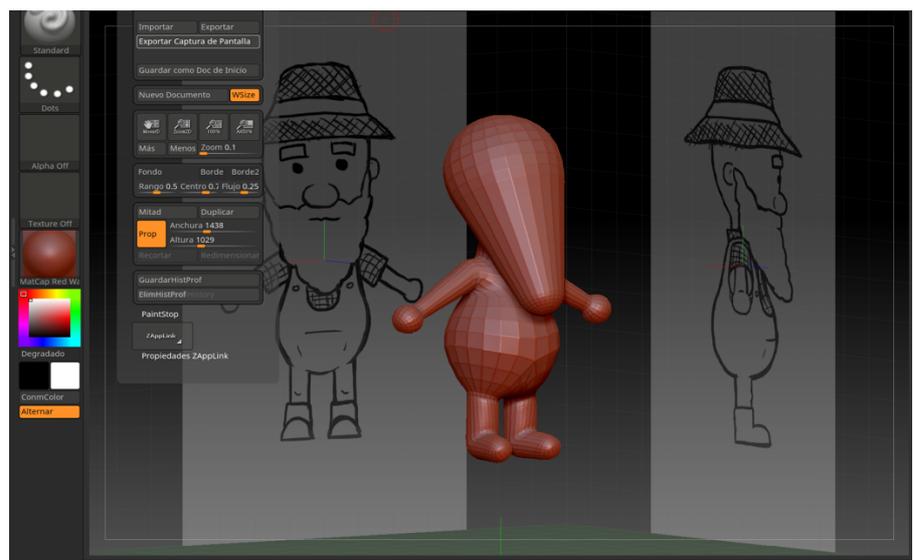
4.3.1. Granjero Fray



Zsphere de Zbrush

El modelado 3D del granjero lo realizamos con el programa de Zbrush. Lo primero que hicimos fue crear una *ZSphere* como base. Las *ZSpheres* son una forma de crear mallas base a partir de esferas de forma rápida para nuestras creaciones. Seguidamente, importamos nuestra vista frontal y perfil para dejarlas como referencia para no pasarnos de dimensiones y que el personaje tenga una coherencia. Colocamos en una posición centrada del personaje nuestra *ZSphere* para poder ir creando más a partir de la primera hasta completar una aproximación a lo que iba a ser la geometría base del granjero. Fue muy importante trabajar siempre con la simetría activada, ya que nos facilitaría mucho la creación de una buena tipología, y posteriormente, para hacer el *rigging*.

Cuando conseguimos acabar el acercamiento a la geometría base con las *ZSpheres*, creamos un *Adaptative Skin*, lo que nos permitió crear por completo una geometría 3D hecha por polígonos y lista para empezar a modelarla. A partir de aquí, empezamos a esculpir digitalmente con la tableta gráfica para facilitar el trabajo, fijándonos muy bien en nuestras referencias mencionadas anteriormente.



Geometría realizada con *Zspheres* con las vistas frontal y perfil.

Aunque utilizamos todo tipo de herramientas para resolver nuestro granjero, una de las herramientas más utilizadas fue el pincel *Move*, ya que nos permitió



Realización de gorro y botas con la herramienta *Extract*

mover de posición los polígonos pudiendo agrandar formas y rebajar las sobrantes. Otra de las herramientas que nos ayudó mucho, fue el pincel *Clay Buildup*, el cual añadía masa a la geometría para aumentar la forma, y más tarde poder suavizarla para que quedara homogénea a la base. También, el pincel *Inflat* nos facilitó la realización de la barba del granjero, para darle un toque esponjoso.

Le insertamos los ojos independientemente con dos esferas 3D, y moviéndolas y escalándolas hasta las hendiduras que marcamos para situarle los ojos. Las cejas fueron hechas por este mismo método.

El gorro y las botas, las realizamos a partir de seleccionar la herramienta de máscara y pintando las zonas donde queríamos que se situaran. A raíz de esto, utilizamos la herramienta *Extract* en el apartado *Subtool*, y le damos el grosor y el suavizado que queramos. Con esta herramienta conseguimos que nos cree un objeto nuevo a partir de la máscara que habíamos pintado sin perder la geometría anterior. Al crearse como objeto independiente, podemos modelarlo y modificar lo que hayamos creado a nuestro gusto.



Aproximación sin smooth al resultado final. Pintado directamente a la malla *low poly*

Una vez acabado el modelado, tuvimos que bajarle la cantidad de polígonos a nuestra geometría con la opción *ZRemesher* y bajarle su carga poligonal para que tuviera menos densidad. El problema fue qué al bajarle la carga poligonal, perdía un 50% de detalle, así que tuvimos que duplicar anteriormente el modelado para utilizar la herramienta *Project* para proyectar la malla que está en baja resolución en la que está en alta resolución y así, tener una geometría menos densa, pero con más detalle. También, fue muy importante fijarse en la retopología de la malla ya que nos informaba de la organización y el orden que llevaban los polígonos de nuestro personaje, y darnos información de donde había más carga poligonal y donde había menos, y de si realmente el personaje era simétrico.

Finamente, le asignamos un material como *FastShader* y pintamos a Fray con herramientas de color del Zbrush, en la malla de alto poligonaje para poder extraer el color posteriormente.

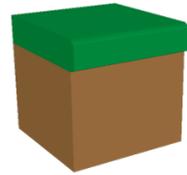
4.3.2. Props

Modelamos también los bloques de las plataformas donde el personaje iba a correr por encima de ellos, y un ovillo de lana. Estos dos, fueron realizados con *Maya3D*.

El ovillo fue realizado mediante la unión de esferas y los bloques mediante cubos y extrusiones de los mismos, pintados por caras y exportados directamente en formato FBX.



Ovillo de lana realizado en Maya



Bloque de tierra con césped hecho en Maya

4.4. TEXTURIZADO Y MAPEADO

Este apartado fue muy importante ya que el programa de *Zbrush* no exporta el color que está pintado encima del modelado, y hay que exportarlo manualmente para poder aplicarlo más tarde al modelo como si fuera una imagen independiente a él. Para ello, tuvimos que pintar el modelado del granjero subiéndole el nivel de subdivisiones de la malla poligonal, para que se pudiera pintar con el detalle máximo posible ya que *Zbrush* funciona con un pintado por vértices, es decir, pinta el vértice y pixela el color de lo que tiene alrededor de ese vértice. Así, con un incremento de subdivisiones de la malla poligonal, habría muchos más vértices y más juntos, lo cual nos interesaba para sacarle el mayor partido al color de *Zbrush* y que el granjero pudiera verse bien a una buena distancia sin que se pixelara el color.

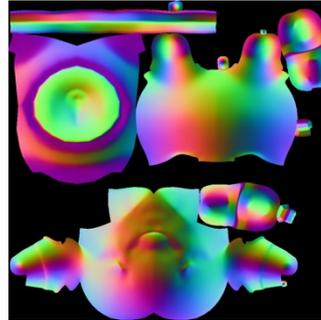
El mapeado de texturas dicta cómo se sitúa la textura sobre el objeto al momento de proyectarse, y es un método para añadir detalles, colores o texturas exteriores a un modelo 3D.

Una vez pintado el modelado de alto poligonaje, creamos un nuevo *Texture Map* a partir de *Polygroups* para que nos separara las diferentes partes del granjero y no se mezclaran entre ellas y para que el propio programa no se confunda para insertarla al modelo. El mapeado de UV es el procedimiento de asignarle una coordenada para cada vértice del objeto que más adelante se va a interponer. Exportamos el mapa de UV's a 1024 de resolución, ya que no necesitábamos más resolución para nuestro videojuego, y así, ahorraríamos recursos para que fuera más fluido y no se sobrecargara. Con esto, se crearía un mapa de UV's. Este procedimiento se podría haber hecho manualmente si *Zbrush* nos hubiera hecho el mapa de UV's erróneamente, pero no nos hizo falta

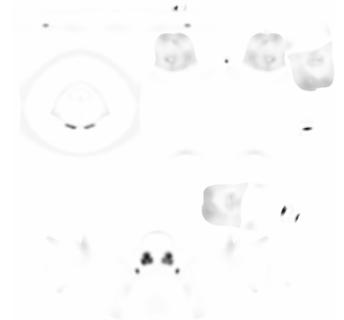
ya que después de mucho trabajo conseguimos resolverlo sin problemas. A partir de aquí tendríamos el modelado 3D del granjero en *low poly* y su textura de una resolución media/alta para poder importarlo en Unity3D o Maya3D posteriormente.



Texture Map de Fray el Granjero, realizado en la malla *high poly* y aplicada al *low poly*



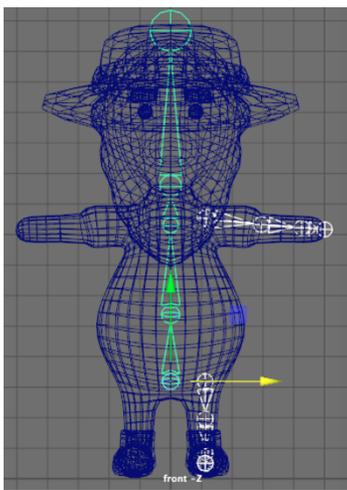
Normal Map de Fray el Granjero



Ambient Occlusion Map de Fray el Granjero

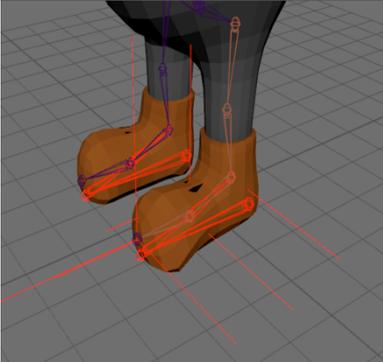
4.5. RIGGING Y SKINING

El *rigging* es el proceso de crear un sistema de controles digitales a un modelo 3D para que pueda ser animado fácilmente y eficientemente. A pesar de ser un proceso largo, y que haya programas que te resuelvan un *rig* completo, hemos querido hacer un *rig* manual para aprender y poder controlar del todo a nuestro personaje modelado. Para que un *rigging* sea eficiente, debe de ser lo suficientemente intuitivo para que el animador no tenga problemas. El Rigger es la persona que trabaja y se encarga exclusivamente en hacer *riggs*, así que queríamos experimentar cómo es su trabajo y darle vida a nuestro personaje.



Creando los joints del personaje

Lo primero que hicimos fue crear e incorporar los *joints* (huesos) a la geometría del granjero para ubicarle las flexiones que tendría nuestro personaje. Fue importante tener conocimientos previos de Maya3D para poder manejarnos en esta interfaz y que nos resultara menos dificultoso. En este apartado la herramienta *Wireframe on shaded* nos permitió ver la geometría poligonal del granjero para ubicarle bien los *joints*. A medida que íbamos creándolos era muy importante ir nombrándolos y emparentándolos adecuadamente para la organización del mismo. Solo hizo falta crear los joints de la mitad del cuerpo, ya que al ser una geometría simétrica podíamos aplicarle un *Mirror* (espejo), por eso era tan importante que al realizar el modelado 3D en Zbrush, utilizáramos la simetría. Una vez finalizado los huesos de todo el cuerpo, le añadimos unos adicionales a las plantas de los talones con otro color,

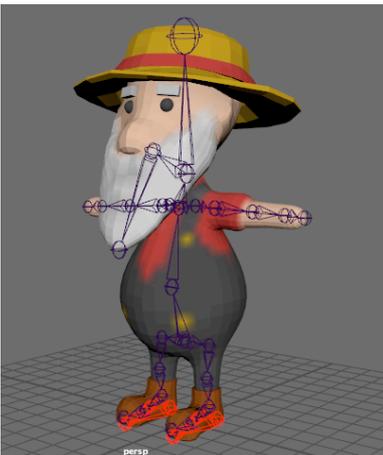


Creando el IK

que nos servirían para plantar al personaje sobre el terreno, así, cuando nosotros quisiéramos mover su centro de gravedad, los pies no se moverían.

Más tarde, configuramos el *rig* de los pies por medio de *IK (Inverse Kinematics)*, es decir, que el pie se mueve de abajo hacia arriba y no al contrario. Así, podríamos mover nuestros pies seguidos por las rodillas de una forma natural. Añadimos unos atributos al pie para que se pudiera rotar en el eje Z el talón, el empeine y la punta del pie.

Una vez terminados los *joints* por completo, creamos todas las curvas que nos servirían como controladores para el *rig*. Es decir, las curvas que realmente va a seleccionar el animador para poder controlar al personaje y que sea lo más intuitivo posible. Tuvimos que adherir las formas de las curvas que nosotros vimos adecuadas a los *joints* que le pertenecían, aunque las formas de las curvas no iban a modificar el *rig*, tan solo es para que el animador pueda identificar rápidamente cada uno de los controladores y sea más intuitivo. También, cambiamos los colores de los controladores para diferenciar la parte izquierda de la derecha.

Personaje con los *joints* creados

Creamos los controladores de las rodillas (igual que el de los codos y los hombros) fuera de la geometría para que fueran más visibles y poder manejar las direcciones de las mismas. Los brazos, al igual que las plantas de los pies, también les atribuimos el *IK Handle* para que al mover la muñeca pueda moverse hasta el hombro. En los brazos tuvimos que emparentar los controladores de forma correcta para que tuvieran un movimiento lógico según el controlador que moviéramos. Por ejemplo, al rotar el controlador del codo (no el de la dirección del codo), el de la muñeca también debería de moverse. También, le añadimos un modo *FK* y lo configuramos para que el *rig* fuera lo más completo posible y desaparecieran los controladores que no nos hicieran falta en este modo. El modo *FK* es la forma por defecto que se comportan los huesos.



Creando los controladores del cuerpo con la textura en alta resolución

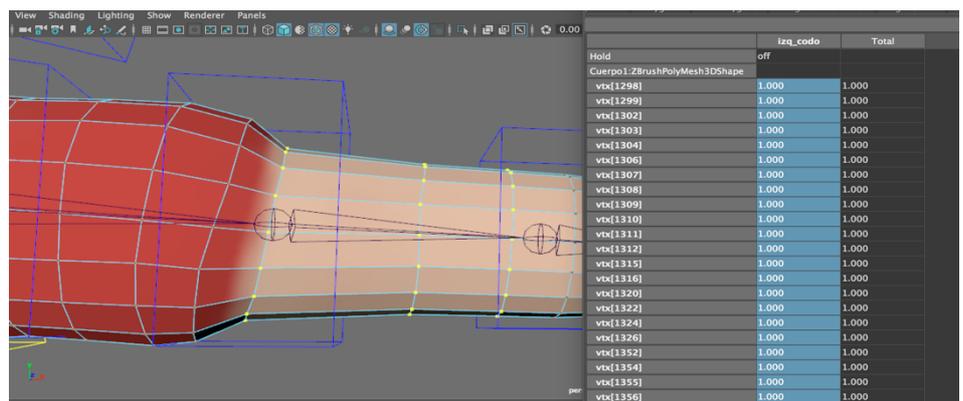


Resultado de todos los controladores creados

Continuamos con crear los controladores restantes como el de la cabeza, el pecho, la columna y el de la cintura y los parentamos con sus respectivos huesos y demás controladores para que cuando movamos un controlador base, el resto le siga. También, creamos y configuramos un controlador global que lo situamos en la base de los pies, que nos permitiría mover a todos los controladores y consigo, a todo el personaje. Al controlador global le dimos forma de gota para tener presente la dirección a la que está mirando el granjero.

Llamamos *skinning* al proceso por el cual se asigna el *rigging* a una malla. En esencia el *skinning* es una relación de parentesco entre la malla y el hueso donde el que manda es este último. A continuación, hicimos todo este largo proceso.

Una vez terminados y configurados todos los controladores, hicimos *Blend Shapes* correctivos, que nos ayudaron a corregir malas deformaciones al rotar ciertas partes del cuerpo. Una vez realizado el *Blend Shape*, seleccionamos al personaje y al hueso raíz y realizamos un *Bind Skin*. Esta opción nos dice que, por cada vértice del personaje, cuantas influencias de *joints* va a tener, y la marcamos en 1. Ahora, si nosotros movemos un controlador, ya podemos ver que la malla se mueve acorde al controlador que estamos moviendo y el rig empieza a funcionar, esta técnica se llama *Binding*. Aunque tuvimos que arreglar esas deformaciones con una técnica que se llama pesado, logramos que el personaje pudiera moverse por primera vez. El pesado tuvimos que hacerlo con todo el cuerpo ya que había que arreglar todas las malas deformaciones que nos encontrábamos. Con el pesado le decíamos a Maya3D qué hueso mueve a qué geometría.

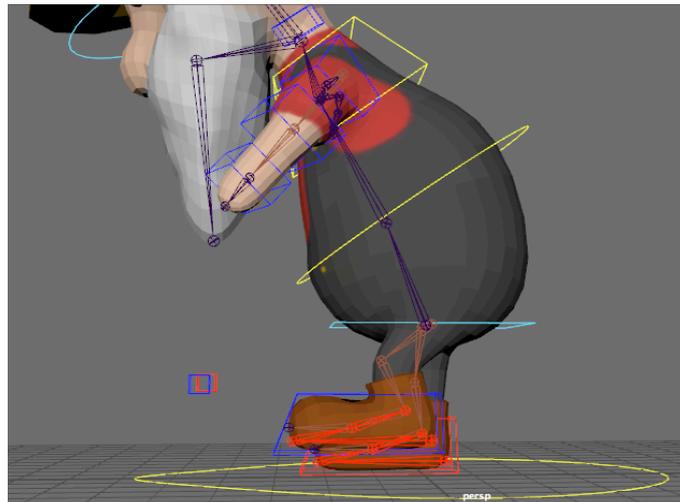


Realizando el pesado al brazo

Una vez pesado toda la geometría y cerciorarnos de qué todos los controles funcionaban correctamente, aplicamos la herramienta *Delta Mush*, una de las herramientas más innovadoras que ha incorporado Maya3D en los últimos años. Esta herramienta nos permitió suavizar todas las deformaciones que podíamos tener al mover cualquier controlador desde la pose inicial del personaje.

Así, el movimiento del granjero sería mucho más suavizado y más natural. Finalmente, arreglamos las pequeñas malas deformaciones desde el *Component Editor* para añadirle los valores a los vértices que hicieran falta.

Una vez acabado de hacer el *skinning* y de arreglar las malas deformaciones que se nos habían presentado, podíamos empezar a animar sin problemas y decir que teníamos un *rigging* completo.



Personaje preparado para ser animado finalmente

4.6. ANIMACIÓN

Una vez hecho el *rig* completo del personaje del granjero, ya podíamos empezar a darle movilidad a Fray y empezar la animación que le corresponde. En este caso, realizamos un caminado con personalidad para la escena donde el granjero va caminando tranquilamente y un árbol le cae encima continuando con la siguiente escena.

Para realizar esta animación, nos basamos en los 8 fotogramas que contiene una animación tradicional fluida de un caminado básico. En este caso, nosotros utilizaremos el programa Maya3D para realizar la animación, y contaremos con 24 fotogramas. La animación pasa por diferentes estados:

- **Contacto:** los dos pies están en contacto con el suelo mientras una pierna está más estirada que la otra, pero las dos en tensión.
- **Amortiguación:** en este estado la cadera está en la posición más baja del caminado, y las dos piernas están flexionadas y el peso cae sobre una pierna.
- **Pose de paso:** aquí el peso del cuerpo recae sobre una pierna dejando la otra en el aire. La cadera está subiendo para alcanzar el siguiente estado.



Animando al personaje realizándole un caminado

- **Extremo alto:** este es el último estado por donde pasa el caminado donde la cadera se encuentra en su punto más alto y un pie está en tensión, mientras que el otro está a punto de caer en contacto con el suelo para repetir todas las fases de nuevo.

La animación se realizó sin mover su controlador global, ya que así obtendríamos una animación que se realizaría en el sitio sin modificar su translación. Con esto conseguíamos tener una buena animación para llevarla a Unity3D y modificarla a nuestro gusto directamente en ese programa, ya que si llegamos a modificar su translación nos habría imposibilitado realizar una animación en *loop* a nuestro gusto.



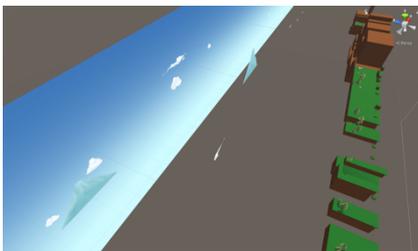
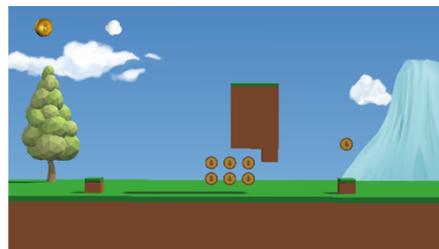
Fray el Granjero realizando el caminado

4.7. Creación del primer nivel

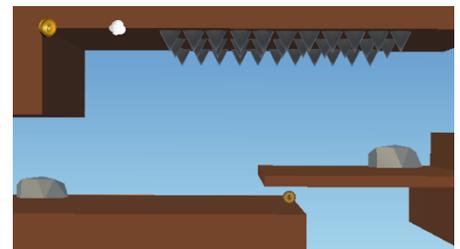
Este apartado se creó conjuntamente con la colaboración de Sergi Adrover. En primer lugar, tuvimos que organizarnos para separar las diferentes partes que iba a contener el escenario jugable, y decidimos separarlas en cuatro partes para conseguir una mayor jugabilidad y que no resultara monótono:

- **Introducción:** fue la primera parte donde debía contener obstáculos sencillos de rebasar para adaptarse a los controles del videojuego.
- **Subterráneo:** en esta parte decidimos añadir una parte subterránea con obstáculos menos intuitivos que la introducción cambiando visualmente el lugar donde nos encontrábamos.

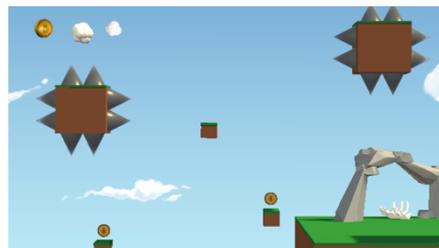
- **Aéreos:** esta zona la reservamos para unos obstáculos más difíciles con posibilidades de caer al vacío y poco intuitivos para que el jugador tuviera la agilidad de superarlos.
- **Final:** en este apartado nos decantamos por construir una zona elaborada mayormente por pinchos y finalizarla con un subterráneo decorado y cristales brillantes.

Efecto *Parallax*

Introducción



Subterráneo



Aéreos



Final

El escenario jugable fue un conjunto de prefabricados⁵ que realizamos manualmente y de *assets*⁶ descargados gratuitamente directamente de Asset Store, realizando combinaciones entre ellos para que quedara visualmente agradable buscando siempre la estética acordada. Fue importante buscar el contraste con nuestro personaje protagonista Sheepy para que destacara y poder visualizarla con facilidad.

El fondo del escenario lo realizamos utilizando el efecto *Parallax*, donde parece que el fondo se mueve a una velocidad distinta que lo que tenemos delante. Logramos este efecto al colocar imágenes en formato PNG a distintas profundidades, y activando una cámara ortográfica. Así, conseguimos añadirle más profundidad a nuestro videojuego 3D.

Finalmente, situamos prefabricados de monedas y de ovillos en distintas ubicaciones para marcar las distintas fases del primer nivel, y añadimos sonidos gratuitos sin licencia a lo que creímos conveniente.

⁵ Tipo de Asset que le permite almacenar un objeto *GameObject* completamente con componentes y propiedades para poderse reutilizar las veces que sea.

⁶ Representación de cualquier ítem que puede ser utilizado en el proyecto.

4.8. Programación

Para finalizar todo el proceso de trabajo, Sergi Adrover y yo tuvimos que programar ciertas partes del escenario jugable para que fuera funcional y poder jugar finalmente. Tuvimos que entender la lógica de la programación en C# de forma básica previamente para programar estas partes:

- Cámara:

```
No selection
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Camera : MonoBehaviour {
6     [SerializeField]
7     Transform target;
8     Vector3 offset;
9
10    // Use this for initialization
11    void Start () {
12        offset = target.transform.position - transform.position;
13    }
14
15    // Update is called once per frame
16    void Update () {
17        Vector3 pos = target.transform.position - offset;
18        this.transform.position = Vector3.Lerp(this.transform.position,pos,1.5f);
19    }
20 }
21
```

- Ovillo:

```
No selection
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Ovillo : MonoBehaviour {
6
7     public AudioSource fuenteSonido;
8     public AudioClip MyAudio;
9
10    void OnTriggerEnter(Collider Other){
11        fuenteSonido.PlayOneShot (MyAudio, 0.9f);
12        actividad2.Score += 1;
13        Destroy (gameObject, 0.1f);
14    }
15
16 }
17
```

- Sheepy:

```

No selection
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Player : MonoBehaviour {
6     private Rigidbody rb;
7     [SerializeField]
8     private float speed;
9     private Animator anim;
10    public bool ActivaSalto = true;
11
12    // Use this for initialization
13    void Start () {
14        anim = GetComponent<Animator> ();
15        rb = this.GetComponent<Rigidbody> ();
16        rb.velocity = new Vector3 (speed, 0f, 0f);
17    }
18
19
20    // Update is called once per frame
21    void Update ()
22    {
23        if (Input.GetKey (KeyCode.X) && ActivaSalto == true) {
24            rb.AddForce (0f, 12.5f, 0f, ForceMode.Impulse);
25            anim.Play ("JumpingUp");
26            ActivaSalto = false;
27        }
28    }
29    if (Input.GetKeyDown ("z")) {
30        anim.Play ("Croach");
31    }
32 }
33
34
35    void OnCollisionEnter(Collision other)
36    {
37        ActivaSalto = true;
38    }
39 }
40 |

```

- Collider⁷ Sheepy:

```

No selection
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class ColliderSheepy : MonoBehaviour {
6     public BoxCollider player;
7
8     // Use this for initialization
9     void Start () {
10        player = player.GetComponent<BoxCollider> ();
11    }
12
13
14    // Update is called once per frame
15    void Update ()
16    {
17        if (Input.GetKeyDown ("z")) {
18            player.size = new Vector3 (0.02899357f, 0.01791253f, 0.01749538f);
19            player.center = new Vector3 (0.0007780308f, 0.008529937f, -0.001612005f);
20        }
21        if (Input.GetKeyUp ("z")) {
22            player.size = new Vector3 (0.02899357f, 0.04390647f, 0.01749538f);
23            player.center = new Vector3 (0.0007780308f, 0.0215269f, -0.001612005f);
24        }
25    }
26 }
27

```

⁷ Los collider definen la forma de un objeto para los propósitos de colisiones físicas. Es invisible y no necesita tener la forma exacta de la malla del objeto.

5. EXPORTACIONES E IMPORTACIONES A UNITY3D.

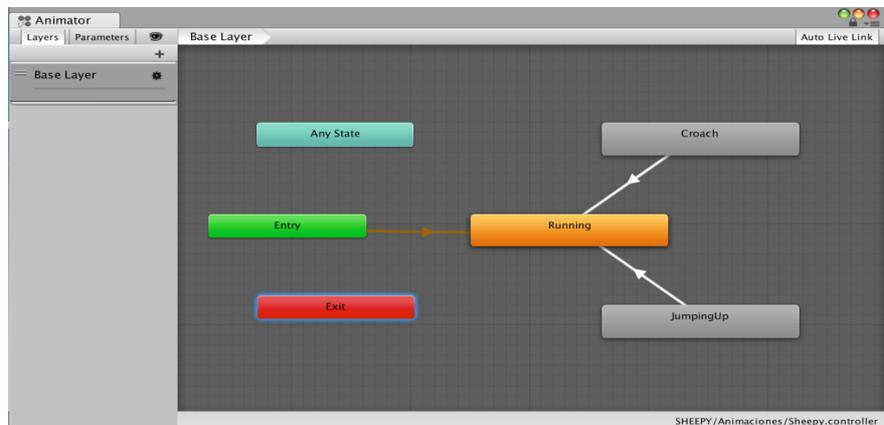
5.1. Importación de modelado 3D y textura a Unity3D

El granjero que modelamos anteriormente, tuvimos que exportarlo desde Zbrush3D a un formato FBX. Los archivos FBX permiten importar y exportar objetos 3D, objetos 2D con grosores, luces, cámaras y materiales de una aplicación a otra. Así, lo exportamos en FBX y lo enviamos a Unity3D directamente, con el inconveniente de que el modelado 3D que habíamos realizado y pintado manualmente, al importarlo al programa de Unity3D, aparecía sin colores ni texturas ya que Zbrush3D no extrae el color directamente a otros programas y tuvimos que hacerlo manualmente como comentamos en el apartado de “Texturizado y mapeado”. También, cabe decir que Unity3D nos aplicó un *Smooth* (suavizado) a la malla al importar el modelado.

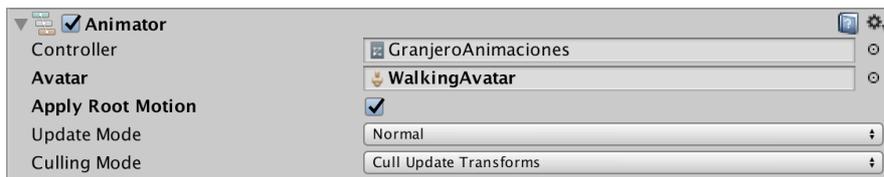
Realizamos varias pruebas ya que por defecto Zbrush3D volteaba el mapa de UV's y el color se nos presentaba de forma aleatoria en el modelado. Una vez seguros de que la textura estaba perfectamente adecuada para el granjero, lo único que tuvimos que crear fue un material en Unity3D y asignarle el mapa de UV's extraído de Zbrush3D. Con este procedimiento el granjero obtuvo el mapa de color creado anteriormente a partir de un modelado *high poly* para llevarlo a un modelado *low poly*. A partir de aquí, Unity3D facilitó unas herramientas para poder variar el material como el metalizado o el brillo.

5.2. Importación de animaciones a modelado Unity3D

Una vez tenemos completada la animación desde un programa externo, la exportamos y la importamos a Unity3D. Por otra parte, para conseguir que nuestro modelado tuviera la movilidad de la animación que habíamos creado, tuvimos que agregarle el componente *Animator* al modelado. El componente *Animator* es utilizado para asignar una animación a un *GameObject* en su escena. El componente *Animator* requiere una referencia a un *Animator Controller* que define qué clips de animación utilizar, y controla cuándo y cómo mezclar y hacer una transición entre estos. Aunque el *Animator Controller* nos sirviese para marcar transiciones entre animación y animación, más tarde tuvimos que programarlas para controlarlas de forma más precisa.



Animator Controller de Sheepy



Component Animator de Fray

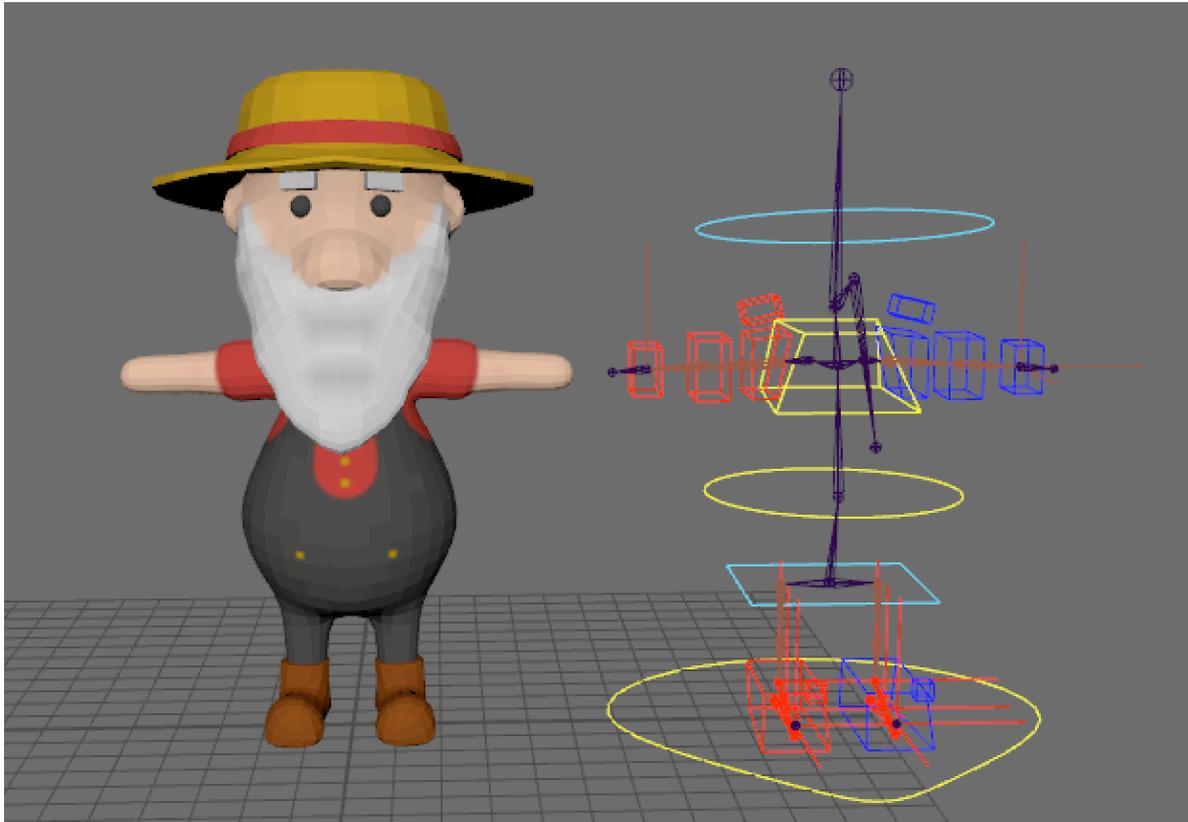
El Avatar también tuvo que ser asignado al Animator Controller, ya que son personajes humanoides caracterizados por ser *GameObjects*.

Este proceso fue realizado para el modelado de Fray, y el modelado de Sheepy para que tuvieran las animaciones integradas.

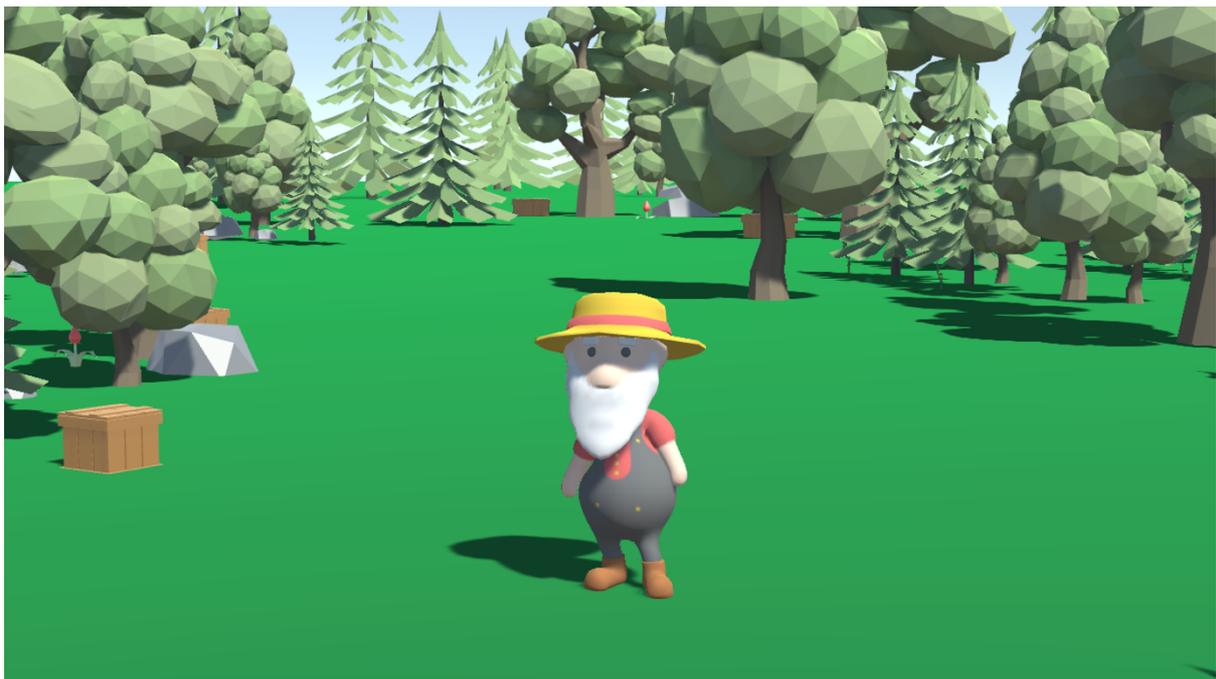
6. RESULTADOS FINALES



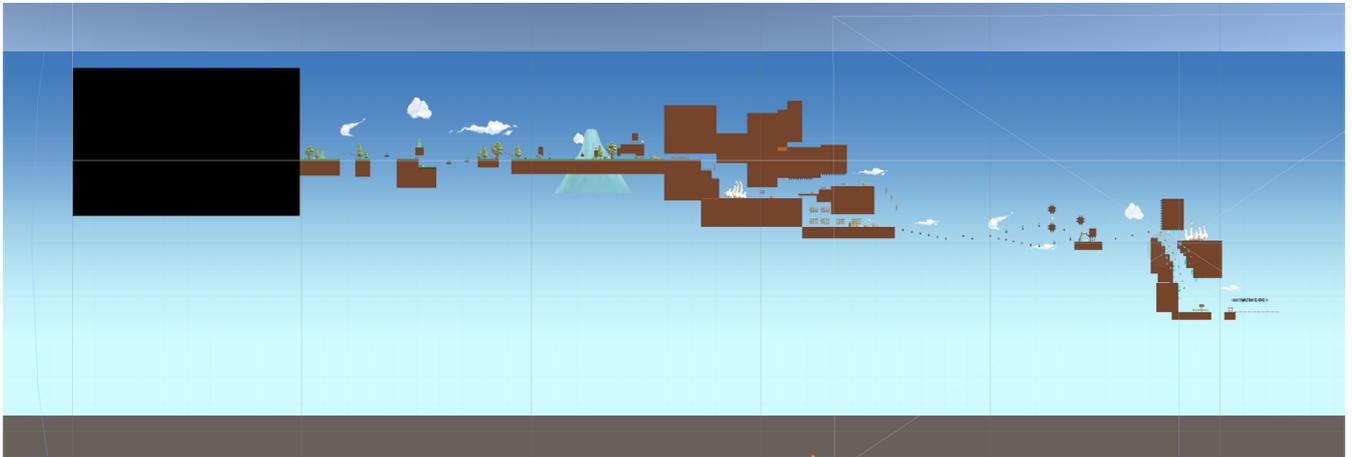
Modelado 3D de Fray el Granjero, con *smooth* y con Texture Map



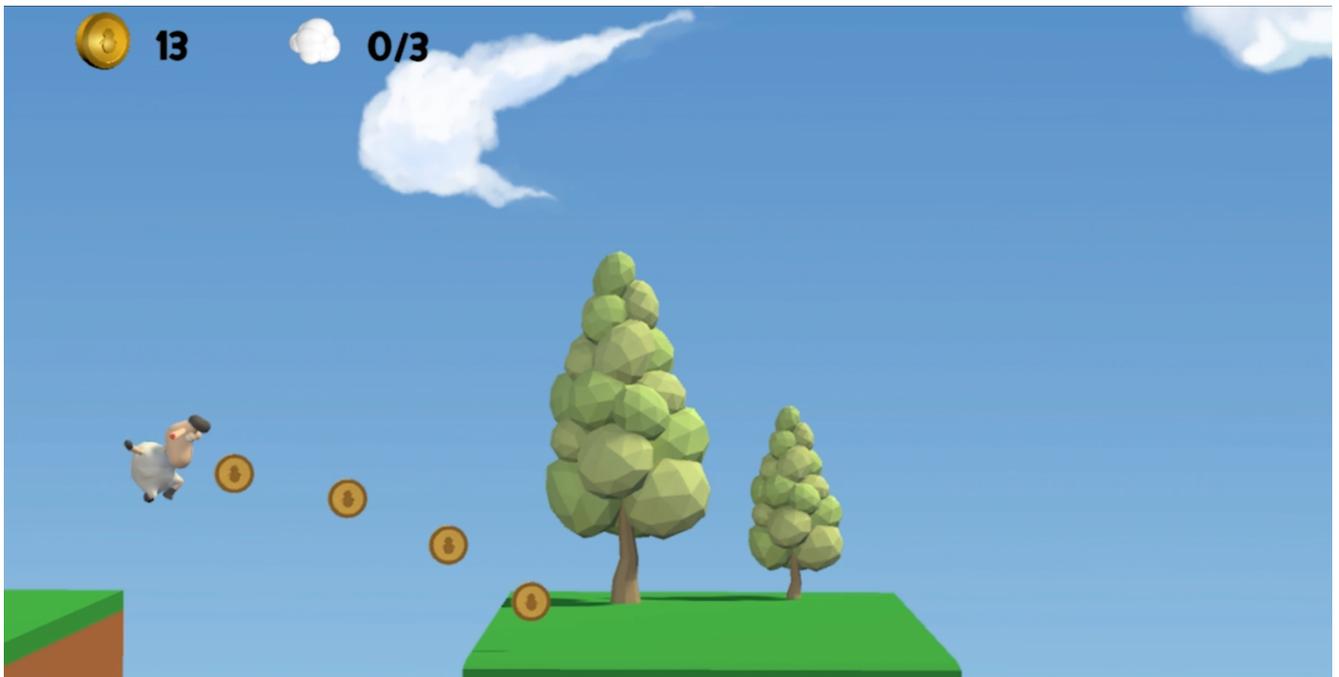
Modelado 3D de Fray el Granjero junto con sus huesos y sus controladores, distanciados para que pueda entenderse mejor el proceso.



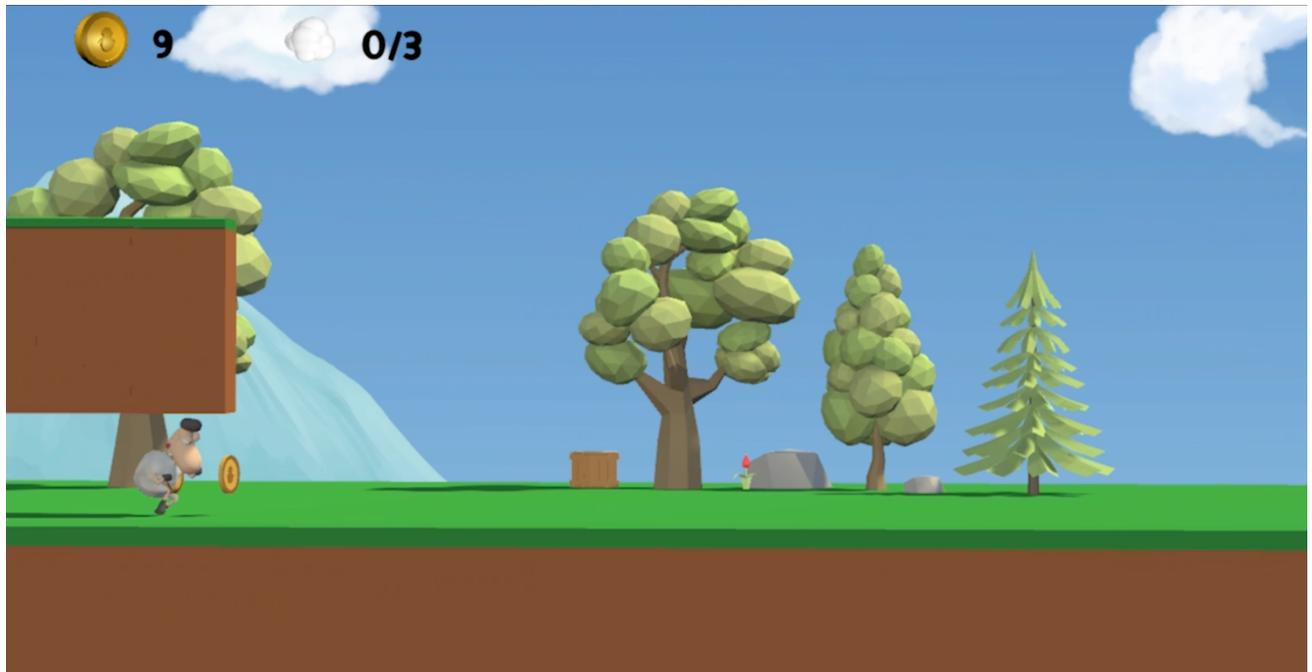
Escenario de la escena correspondiente al modelado 3D de Fray el Granjero, integrado con más elementos en Unity con pose del personaje



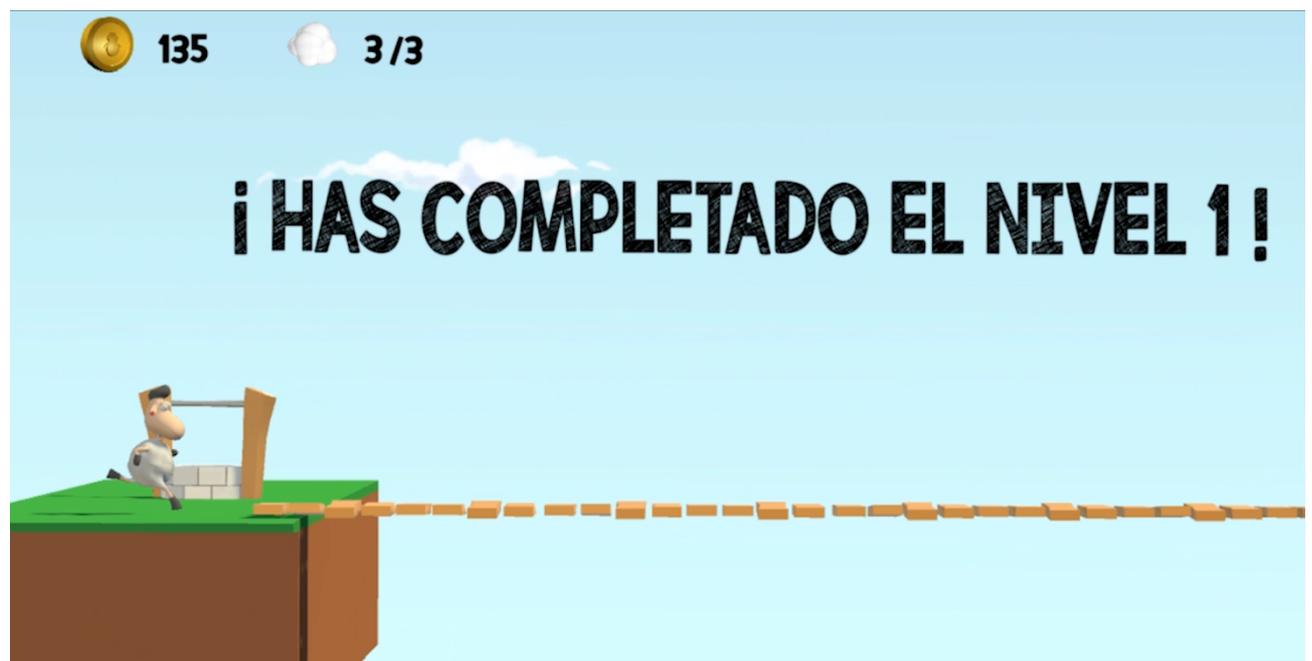
Diseño del primer nivel del videojuego en una vista 2D con todos los elementos que aparecen



Fotograma extraído del videojuego *Sheepy Help* donde el personaje tiene activada la acción de salto, cuando estamos *in game*



Fotograma extraído del videojuego *Sheepy Help*, donde el personaje tiene activada la acción de agacharse, cuando estamos *in game*



Fotograma extraído del videojuego *Sheepy Help*, donde el jugador ha conseguido pasarse el nivel, cuando estamos *in game*

7. CONCLUSIÓN

Después de haber realizado este Trabajo Final de Grado; este proyecto de videojuego denominado *Sheepy Help*, me gustaría indicar (muy brevemente), que dicho proyecto ha contribuido en la consolidación de todo lo aprendido hasta el momento y, por tanto, en mi futura formación personal.

Fue todo un reto abarcar todas y cada una de las fases que están implícitas en el desarrollo de un videojuego, hemos de recordar que estos proyectos suelen solventarse por un grupo de trabajadores de mayor número que el nuestro, muy especializados en sus diferentes áreas, demarcadas por departamentos.

También es importante destacar, que se han integrado y desarrollado muchos de los conocimientos adquiridos durante el Grado. También, se ha necesitado de un aprendizaje extra fuera del Grado; como la adquisición de cursos *online* y de una importante búsqueda de documentación sobre el sector y la materia a tratar para obtener el mejor resultado posible, siendo totalmente satisfactoria y muy grata la búsqueda que consolidase los resultados del aprendizaje.

Por otra parte, muchos han sido los problemas que han ido surgiendo por falta de experiencia en los diferentes ámbitos y fases que conllevaba el proyecto, que han provocado un leve retraso en los tiempos, aunque se han solventado de la mejor manera posible.

La resolución de los problemas y la consecución final del proyecto ha sido un logro que ha superado nuestras expectativas iniciales. La adecuación de la parte gráfica y su optimización dentro del videojuego ha sido, personalmente, todo un reto alcanzado: el poligonaje para el personaje Fray el Granjero ha alcanzado tan solo los 4.000 polígonos y se ha realizado un *rig* completo e intuitivo además de personal. Fue muy grato el hecho de finalizar el *rig* y poder realizarle cualquier tipo de animación al personaje incluyéndolo en cualquier tipo de proyecto de animación o videojuego.

Este proyecto me ha reforzado y preparado para un futuro en el ámbito de los videojuegos, ya que he tenido la oportunidad de trabajar en grupo en las diferentes fases que conlleva el desarrollo y creación de un videojuego. Cabe señalar que con todo lo aprendido durante estos años y su consolidación en este TFG me ha permitido comprender y entender mejor el sector, permitiéndome incluso establecer un estudio de caso, realizado a *Frame Over*, que ha solventado algunas dudas sobre el funcionamiento, actualmente, de toda una industria en pleno apogeo.

8. BIBLIOGRAFÍA

- CAILLOIS, R. *Los juegos y los hombres. La máscara y el vértigo*. Ed. Fondo de cultura económica, 1967. Descargable en pdf [Consultado el 29/04/2018]:
https://poliformat.upv.es/access/content/group/GRA_13869_2017/TEXTOS/caillois-r-los-juegos-y-los-hombres.pdf
- DESARROLLO ESPAÑOL DE VIDEOJUEGOS. *Libro Blanco del Desarrollo Español de Videojuegos 2017*. Descargable en pdf [Consultado el 26/02/2018]:
<http://www.dev.org.es/images/stories/docs/resumen%20ejecutivo%20-%20libro%20blanco%20dev%202017.pdf>
- ERKKI HUHTAMO. *Máquinas de diversión, máquinas de problemas*. Artnodes. Revista de arte, ciencia y tecnología. Universitat Oberta de Catalunya. 2007. Descargable en pdf [Consultado el 08/02/2018]:
<http://www.upv.es/laboluz/mie1/text/huhtamo.pdf>
- EXPOSICIÓN MUVIM. *Del tilt al byte*. 2014. Descargable en pdf. [Consultado el 25/03/2018]:
https://poliformat.upv.es/access/content/group/GRA_13869_2017/TEXTOS/Del-tilt-al-byte.pdf
- FLANAGAN, MARY. *Critical Play*. MIT Press, 2009. Descargable en pdf [Consultado el 07/04/2018]
- GRETHE, M, y CLARKE.A. "Videogames and Art", Ed. Intellect books. Universidad de Chicago, 2007. Descargable en pdf [Consultado el 12/05/2018]
- IVANA VERÓNICA RIVERO. *El juego desde los jugadores. Huellas en Huizinga y Caillois*. Universidad Nacional de Río Cuarto 2015. Descargable en pdf [Consultado el 19/02/2018]:
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=6&ved=0ahUKEwjchd_ttp7cAhUyyYUKHRRaBuoQFghSMAU&url=https%3A%2F%2Fwww.raco.cat%2Findex.php%2FEnrahonar%2Farticle%2Fdownload%2F306827%2F396810&usg=AOvVaw3ZWSLiU1yegwwpetXLWzfn
- JOHAN HUIZINGA. *Homo Ludens*. El libro de bolsillo. Historia. Alianza Editorial/Emecé Editores. 1972. Descargable en pdf. [Consultado el 10/2/2018]:
<http://zeitgenoessischeaesthetik.de/wp-content/uploads/2013/07/johan-huizinga-homo-ludens-español.pdf>

8.1. Material Online

- <https://sites.google.com/site/elcajondelosvideojuegos/home/4-oxo-tic-tac-toe> : [Consultado el 22/02/2018]
- <http://www.dev.org.es/images/stories/docs/resumen%20ejecutivo%20-%20libro%20blanco%20dev%202017.pdf> : [Consultado el 26/02/2018]
- <http://zeitgenoessischeaesthetik.de/wp-content/uploads/2013/07/johan-huizinga-homo-ludens-espa%C3%B1ol.pdf> : [Consultado el 25/03/2018]
- http://es.geometry-dash.wikia.com/wiki/Geometry_Dash_SubZero?li_source=LI&li_medium=wikia-footer-wiki-rec : [Consultado el 30/03/2018]
- <http://es.ign.com/temple-run-android/73401/review/analisis-de-temple-run-iphone> : [Consultado el 30/03/2018]
- <https://onewindows.es/2018/06/tunic-uno-de-los-exclusivos-sorpresa-de-xbox-one-e3-2018/> : [Consultado el 31/03/2018]
- <http://docs.autodesk.com/CIV3D/2012/ESP/filesAUG/WS1a9193826455f5ff603f1a63123ab7a861f-7426.htm> : [Consultado el 09/04/2018]
- https://www.youtube.com/watch?v=m9N_sIBRWvY : [Consultado el 10/04/2018]
- <https://docs.unity3d.com/es/current/Manual/class-Animator.html> : [Consultado el 17/04/2018]
- <https://www.e3expo.com> : [Consultado el 19/05/2018]

8.2. Cursos Online adquiridos

- Udemy: *Rigging para personajes en Autodesk Maya*. Edgar Edrey, Instructor profesional de 3D y VFX certificado por Autodesk. Curso de pago. [Consultado des del 10/04/2018]
- Udemy: *Master en Programaci3n de Videojuegos con Unity 2018*. Mariano Rivas, Desarrollador de videojuegos Certificado por Unity, profesor en la U-Tad, Programador y desarrollador de videojuegos con Unreal Engine. Curso de pago. [Consultado des del 14/05/2018]
- YouTube: *Modelar un Pokemon Fusion*. Tutoriales Kames. Curso totalmente gratuito. [Consultado des del 20/03/2018]