

Spatial Minery Simulation

Proyecto VR

Acta de relación de contenidos y reparto de tareas

Miembros del equipo

Sebastián José Díaz Rodríguez

Ernesto Echeverría González

Germán Andrés Pescador Barreto

Antonio Jesús González Luis

Este documento representa el eje organizativo del grupo de desarrollo que ha llevado a cabo el proyecto de realidad virtual Spatial Minery Simulation y por la presente su composición de tareas, distribución de estas y los problemas derivados del proyecto.

Desarrollo de la simulación

Recopilación de información

1.1. Aprendizaje Unity3D en la web oficial.

Encargados: Grupo

1.1.1. Creación de escenas.

La enseñanza guiada durante el curso en otras asignaturas y la gran cantidad de tutoriales, guías y material multimedia dispuesto tanto oficialmente como por otros usuarios nos permitió comprender rápidamente la construcción de las escenas y las interacciones entre los objetos.

1.1.2. Scripting.

Debido al framework de desarrollo, MonoDevelop, y las bases de Unity en C# los integrantes del grupo tuvimos que profundizar en un nuevo lenguaje prácticamente desconocido. El punto positivo de esto es que, acompañando a las posteriores clases en otra asignatura, pudimos dar grandes saltos en el desarrollo mediante una enseñanza medianamente guiada.

1.2. Familiarización con realidad virtual.

La gran parte de la fase de aprendizaje se basó en comprender los artefactos de la Realidad Virtual y su posible aplicación al proyecto. Guiados por las clases de Interfaces Inteligentes, pudimos construir una buena base de desarrollo en Realidad Virtual con las Google Cardboard.

1.3. Aprendizaje SketchUP en la web oficial.

1.3.1. Modelado de figuras.

Encargados: Sebastián y Ernesto

A través de la herramienta SketchUp se aprendió a desarrollar estructuras. Sin embargo, tras invertir tiempo en la obtención de estos conocimientos, nos dimos cuenta de que el objetivo del software no era tan compatible como el de otras herramientas, lo que nos incitó a no aplicarlo en el proyecto, sino extender nuestros conocimientos de modelado mediante Blender.

Modelado

2.1. Creación y animación del personaje del usuario.

Encargados: Ernesto

El personaje del usuario es una astronauta perteneciente a un proyecto llamado “Robot Lab”, publicado por el equipo de Unity en su plataforma Asset Store. Incluía una animación de estado de reposo, pero se decidió retirarla debido a su intrusión con la inmersión de la realidad virtual al mover esta la cabeza del personaje.

2.2. Creación y animación del robot recolector.

Encargados: Antonio, Ernesto, Sebastián

El robot recolector se extrajo del mismo proyecto que el personaje principal. Venía con una animación de las ruedas que fue incluida en la simulación.

2.3. Creación de las migas para marcar el trayecto.

Encargados: Antonio

Las migas son una esfera simple sobre la que se ha incluido una textura plana.

2.4. Creación del mineral a buscar.

Encargados: Sebastián

El mineral a buscar fue creado en Blender enteramente.

2.5. Creación de la nave nodriza.

Encargados: Grupo

La nave nodriza se descargó como asset externo a través de la Asset Store. El nombre de la publicación es “Sci-Fi Scout Drone”.

2.6. Creación del dron vigilante.

Encargados: Grupo

El dron vigilante se obtuvo en la Asset Store. El nombre de la publicación es “Space Droid”.

2.7. Creación del modelo de superficie utilizado en la escena.

Encargados: Germán

La creación del escenario comenzó con la elevación de las montañas de los extremos en una plataforma de 250x250. Para ello se utilizaron distintos pinceles de volumen con opacidades y tamaños variables para que no hubiese dos montículos iguales. Posteriormente, se elevaron las colinas centrales que servirían de obstáculos para los recolectores.

Con el mapa ya maquetado, se procedió a la nivelación de las elevaciones y su posterior lijado para mejorar la inmersión del usuario. Teniendo el modelo del mapa perfecto, se pasó al pintado del mismo. Tras analizar texturas de todo tipo de rocas, se utilizaron 3 diferentes, una de aspecto rocoso llena con capas para las elevaciones, una imitación de ceniza volcánica para el suelo, y una ligeramente granulada para suavizar la transición entre ambas.

Teniendo en cuenta que la simulación transcurre en un planeta extraterrestre, se decidió utilizar una textura para el cielo con características completamente diferentes a las de la Tierra. Una atmósfera rojiza, con un satélite muy cercano y su estrella acercándose al horizonte.

Montaje

3.1. Montaje de la escena inicial.

Encargados: Grupo

La colocación de los elementos en la escena no tardó mucho en realizarse tras la implementación de estos por separado y su posterior fusión en un proyecto.

3.2. Estudio de estados y medidas de decisión de los droides.

Encargados: Ernesto

En el caso de los robots recolectores, debatimos mucho tiempo sobre las medidas para su implementación. Finalmente se optó por la implementación de agentes reactivos mediante estados de ejecución, lo que simplificó el proyecto.

Scripting

4.1. Script de movimiento del personaje del usuario y de robots recolectores.

Encargados: Ernesto

El script de movimiento del personaje es un script bastante simple que lo traslada a través de espacio en los ejes X y Z según se pulsen las flechas de dirección o conjuntos de teclas/botones similares.

El de los robots recolectores por otra parte no responde a la pulsación de ninguna tecla, simplemente se les van asignando direcciones aleatorias dependiendo de si chocan con obstáculos o cuánto tiempo llevan desde el último cambio de dirección.

5.1. Script de cálculo de dirección bot-nave nodriza.

Encargados: Ernesto

Tras el conocimiento de las funciones `lookAt()` y `moveForward()` proporcionadas por Unity3D, esto parecía una tarea sencilla, pero tras la implementación del algoritmo A* para evitar el choque en un bucle, todo este cálculo fue cambiado.

5.2. Script de recogida de recurso.

Encargados: Ernesto y Antonio

En el momento en el que un robot recolector entra en contacto con una mena, la cantidad de recurso en la misma disminuye, y el robot pasa a estado 3, debiendo volver mediante el camino determinado con el algoritmo A* hasta la nave para depositar el recurso encontrado, tras lo que volverá a por más. Cuando la cantidad de recurso en una mena alcanza 0, la mena desaparece.

5.3. Script de recogida de las migas.

Encargados: Antonio

De acuerdo a la implementación final, las migas ya no son recogidas, ya que no son parte del mundo y partes del recurso, sino marcadores gestionados por los recolectores para ayudarse los unos a otros.

5.4. Script generación y colocación de migas.

Encargados: Antonio

Tras encontrar un recurso por primera vez, el robot recolector que haya realizado el hallazgo regresará hacia la nave nodriza, dejando atrás migas que señalan a las migas anteriores, permitiendo la navegación entre migas desde la nave hasta los recursos.

Las migas son eliminadas mediante la llegada a 0 de su coste, una variable que aumenta con cada trayecto recurso-nave y disminuye en el trayecto inverso.

6.1. Script de choque

Se desestimó la implementación de un script de choque y se optó por utilizar colliders y rigid bodies en todos los elementos de la simulación.

6.2. Script de reconocimiento de voz

Encargados: Germán y Sebastián

Para el reconocimiento de voz en dispositivo Android se hizo uso de un paquete de pago de internet, dada la dificultad que el uso de esta característica presentaba en Unity en el marco del desarrollo de aplicaciones móviles. Con el uso de este 'plugin' se creó un plugin capaz de escuchar al usuario y cotejar sus palabras con una serie de strings predefinidos que desencadenan una serie de acciones efectuadas por un dron centinela que sobrevuela la zona siguiendo las instrucciones del operador.

Documentación

7.1. Documentación de los agentes.

Encargados: Sebastián

Por cuestiones de tiempo, la documentación de los agentes quedó como una definición escueta de su funcionamiento, de por sí entendible tras su uso.

7.2. Documentación de los scripts

Encargados: Ernesto

Los scripts utilizados en el proyecto fueron completamente documentados.

Distribución del trabajo

Durante todo el período de desarrollo se ha intentado hacer un uso equilibrado del tiempo entre los integrantes del proyecto, derivando tareas según se concluían las anteriores y buscando la equidad en la distribución de trabajo. Sin embargo ha sido imposible evitar distribuir determinados aspectos del proyecto a una única persona (debido a esto se acompaña de una etiqueta a cada tarea), lo que podríamos reflejar como concentraciones de trabajo por parte de los integrantes.

Las tareas de aprendizaje y recopilación de información fueron llevadas a cabo por todos los miembros, ya sea compartiendo recursos, enseñándonos entre nosotros o investigando librerías y guías para la implementación de la simulación. Si bien el modelado fue sobre todo llevado por Ernesto Echeverría González y Sebastián José Díaz Rodríguez, el montaje de la escena fue llevado a cabo en grupo, tanto para evitar problemas de importación o fallos de referencia como para ponernos de acuerdo en los aspectos que iban surgiendo según avanzaba el proyecto.

Respecto a la distribución de la programación, si bien se distribuyeron las funcionalidades y nos explicamos al resto así como documentamos el código sobre lo que íbamos haciendo, existe la especialización de cada miembro en ciertos apartados de la simulación, reflejado a través de las etiquetas que acompañan las tareas. La documentación del proyecto y los otros ficheros de gestión del mismo fueron repartidos entre Germán Andrés Pescador Barreto, Ernesto Echeverría González y Sebastián José Díaz Rodríguez en su mayoría.

La comunicación en el grupo, si bien no fue correctamente llevada en un principio debido a la inacción existente en este a finales de diciembre, se vio mermada y para enero mejorada gracias a las continuas reuniones destinadas a la resolución de los problemas que iban surgiendo en el problema, detallados a continuación, además de la puesta en conjunto de lo que íbamos desarrollando de forma particular. Las principales reuniones se dieron con la problemática de la licencia de colaboración de Unity y el reconocimiento de voz.

Problemas durante el desarrollo

En un principio, el proyecto no presentaba mayores dificultades que los plazos. Sin embargo, durante la fase de desarrollo, más específicamente en el modelado, surgieron problemas en cuanto al apartado visual del proyecto. Unity 3D no permite la animación directa sobre humanoides y no dispone de documentación que indique cómo crear clips para los modelos musculares. La búsqueda de una solución, añadida a la carga lectiva que supusieron las clases al principio del proyecto, ralentizó el desarrollo del proyecto levemente en comparación a los problemas siguientes.

Tras la formación del equipo, acercándose a la finalización de noviembre, dos de los cuatro miembros redujeron su comunicación a niveles mínimos, lo que ocasionó un desarrollo prácticamente congelado del proyecto. La fase de aprendizaje previa al conflicto de horarios fue un punto de apoyo a la hora de remontar en lo que respecta a fechas y tras la primera semana de enero pudimos contar con el equipo al completo para trabajar al unísono.

Por si fuera poco, la plataforma Unity, base del proyecto, canceló la suscripción de colaboración al equipo de desarrollo, lo que nos impidió compartir fácilmente los avances entre nosotros en épocas ajetreadas. La finalización del proyecto se logró gracias a las continuas reuniones de todos los miembros del grupo para poner puesta en común de lo desarrollado y avanzar en el desarrollo de la simulación.

Respecto a la funcionalidad de Realidad Virtual, intentamos implementar características específicas de unas gafas virtuales, las Google Cardboard, pero la gestión de sus elementos particulares supuso problemas de compatibilidad con el proyecto y tuvimos que optar por la opción básica y funcional ofrecida por Unity.

El último de los problemas fue el reconocimiento de comandos por voz. Debido al uso de librerías específicas para Windows, tuvimos que remodelar todo el reconocimiento de voz para hacerlo efectivo. Para ello tuvimos que comprar el asset de reconocimiento de voz propio de android para el entorno Unity, lo que nos supuso un golpe imprevisto en el desarrollo.