

Universidad Carlos III de Madrid
Escuela Politécnica Superior
Ingeniería en Informática



Proyecto Fin de Carrera

Desarrollo de un Entorno Marciano en
USARSim para Evaluación de Robots Móviles

Autor: D.^a Claudia Brito Pacheco

Tutores: Prof. D. Alberto Valero Gómez y Cated^{co} D. Luis
Moreno Lorente

Fecha: Octubre 2011



Título: Desarrollo de un Entorno Marciano en USARSim para Evaluación de Robots Móviles

Asunto: Memoria del Proyecto Fin de Carrera de Ingeniería en Informática

Autor: D. ^a Claudia Brito Pacheco

Tutores: Prof. D. Alberto Valero Gómez y Cated^{co} D. Luis Moreno Lorente

Universidad Carlos III de Madrid

Campus de Leganés



PROYECTO FIN DE CARRERA

Departamento de Ingeniería de Sistemas y Automática
Universidad Carlos III de Madrid

Título: Desarrollo de un Entorno Marciano en USARSim
para Evaluación de Robots Móviles

Autor: D. ^a Claudia Brito Pacheco

Tutores: Prof. D. Alberto Valero Gómez y Cated^{co} D. Luis Moreno Lorente

La lectura y defensa del presente Proyecto Fin de Carrera se realizó el día 24 de Octubre de 2011 bajo el tribunal:

- **Presidente:** Juan González Gómez
- **Secretario:** César Arismendi Gutiérrez
- **Vocal:** Beatriz López Boada

Habiendo obtenido la calificación de:

Presidente

Secretario

Vocal



Agradecimientos

A mis padres, José y Conchi, gracias por haberme dado una formación académica y humana, por haber sido los mejores compañeros de viaje, por haber estado en todos y cada uno de los momentos importantes de mi vida, por aconsejarme y sobre todo por demostrarme día a día vuestro amor incondicional.

A mi hermano Adrián, compañero de estudios y pronto de profesión, por tener un gran talento para la informática y por ser una de las mejores personas que conozco. Gracias por estar siempre ahí dispuesto a echarme una mano.

A mi hermano Fabio, el peque de la casa, gracias por tu sentido del humor en momentos de mucho estrés y por tu inocencia, haces que el mundo parezca un lugar mejor.

A mi abuela Antonia, una mujer increíble, trabajadora, luchadora y adelantada a su tiempo. Gracias por quererme tanto, por preocuparte por mí cuando estaba lejos de casa y por ayudarme cuando más lo necesitaba.

A mi tía Mari, la persona más aventurera que conozco, gracias por enseñarme que el trabajo bien hecho siempre es una gran satisfacción personal, que sin sacrificio y sudor no se consiguen logros en la vida.

Por último a mis tutores por haber despertado mi interés por la robótica, por haber creído en mí y sobre todo por haberme facilitado el trabajo en este tiempo.



Resumen

Este documento contiene la memoria del Proyecto Fin de Carrera que consiste en el desarrollo de un entorno marciano en USARSim para el estudio del movimiento de un vehículo robótico de exploración espacial (rover) que tendrá como objetivo explorar la superficie de Marte.

Para abordar esta cuestión se ha realizado un trabajo de investigación sobre la creación de entornos con la herramienta Unreal Level Editor que utiliza USARSim y sobre el propio manejo del simulador.

Con los conocimientos adquiridos se ha procedido a la construcción del entorno marciano conforme a los requisitos inicialmente establecidos. Una vez creado, se ha simulado el entorno y el rover en USARSim y se han llevado a cabo diversas pruebas en el simulador para verificar que la solución aportada es válida para resolver el problema que se plantea.



ÍNDICE DE CONTENIDOS

1.	Introducción al problema a resolver	19
1.1	Problema general	19
1.1.1	La exploración del espacio	19
1.1.1.1	Breve historia	19
1.1.1.2	Situación actual de la exploración espacial.....	24
1.1.2	La exploración robótica del espacio	30
1.1.2.1	Breve historia	30
1.2	Particularización del problema dentro del marco general	32
1.2.1	Otros simuladores	34
1.2.2	Otros modos de resolver el problema (sin simulador).....	35
1.2.2.1	FMARS (Flashline Mars Artic Research Station)	35
1.2.2.2	MDRS (Mars Desert Research Station)	36
1.2.2.3	Euro-MARS (European Mars Analogue Research Station).....	37
1.2.2.4	MESCH (Mars Environmental Simulation Chamber)	38
1.3	Organización del documento	39
2.	Resolución del problema	40
2.1	Metodología.....	40
2.1.1	Unreal Tournament.....	40
2.1.1.1	Unreal Tournament 2004 (UT2004)	40
2.1.2	Unreal Engine	41
2.1.2.1	Unreal Engine 1 (UE1) – 1998.....	42
2.1.2.2	Unreal Engine 2 (UE2) – 2002.....	42



2.1.2.3	Unreal Engine 3 (UE3) – 2006.....	43
2.1.3	Unreal Editor (UnrealEd)	43
2.1.3.1	Unreal Level Editor (también denominado UT2004 Editor)	44
2.1.4	USARSim	82
2.1.4.1	Características	83
2.1.4.2	Arquitectura.....	83
2.1.4.3	Componentes.....	85
2.1.4.4	Gamebots.....	86
2.1.4.5	Controladores	90
2.2	Descripción de la solución.....	91
2.2.1	Creación del entorno marciano.....	91
2.2.1.1	Creación de un mapa para un mundo marciano	92
2.2.1.2	Creación de la atmósfera marciana	100
2.2.1.3	Iluminación del mapa	105
2.2.1.4	Creación del terreno marciano	113
2.2.1.5	Construcción de rampas	130
2.2.1.6	Creación del cielo marciano.....	133
2.2.2	Simulación del entorno marciano	152
2.2.2.1	Starting Poses	152
2.2.2.2	Modo MultiView.....	153
2.2.2.3	Incorporación del robot al entorno.....	156
2.2.2.4	Control del robot	159
2.3	Resultados obtenidos	161
3.	Conclusiones.....	163



3.1	Objetivos conseguidos.....	165
3.2	Cronograma	166
3.3.	Presupuesto.....	170
3.3.1	Costes de personal	170
3.3.2	Costes de equipos	170
3.3.3	Costes de software	171
3.3.4	Costes de material fungible	171
3.3.5	Coste total.....	172
4.	Líneas futuras de trabajo.....	173
	Anexos.....	174
A.	Simulación en USARSim	174
A.1	Compilar USARSim.....	174
A.2	Cómo arrancar USARSim.....	175
B.	Glosario de términos.....	180
	Referencias	185
	Acrónimos	186



ÍNDICE DE ILUSTRACIONES

Ilustración 1 Lanzamiento de un V-2 en 1943	19
Ilustración 2 Sputnik 1	20
Ilustración 3 Sonda espacial Luna 2	21
Ilustración 4 Apolo 17, viaje final a la Luna	21
Ilustración 5 Primer lanzamiento del transbordador espacial en 1981	22
Ilustración 6 La estación espacial Mir el 12 de junio de 1998	22
Ilustración 7 La estación espacial internacional fotografiada el 30 de mayo de 2011 ..	23
Ilustración 8 Lanzamientos comerciales frente a no comerciales en EEUU	27
Ilustración 9 Estimación de lanzamientos comerciales vs. no comerciales en EEUU ..	27
Ilustración 10 Evolución de los lanzamientos espaciales comerciales	28
Ilustración 11 Lanzamientos espaciales comerciales por país	28
Ilustración 12 Lanzamientos espaciales comerciales estimados por país	29
Ilustración 13 Vista de Flashline Mars Artic Research Station en Julio de 2009	36
Ilustración 14 Mars Desert Research Station: hábitat, observatorio e invernadero.....	36
Ilustración 15 Euro-MARS: diseño del hábitat	37
Ilustración 16 MESCH (Cámara de simulación de entorno marciano).....	38
Ilustración 17 Interfaz del editor de nivel.....	45
Ilustración 18 Barra de menú.....	45
Ilustración 19 Menú archivo.....	45
Ilustración 20 Menú edición.....	46
Ilustración 21 Ventana de búsqueda de actores.....	48
Ilustración 22 Opciones de "Select Surfaces"	48



Ilustración 23 Menú ver.....	50
Ilustración 24 Actor Class Browser.....	53
Ilustración 25 Ventana del explorador de animaciones.....	53
Ilustración 26 Ventana del explorador de grupos.....	54
Ilustración 27 Ventana del explorador maestro.....	54
Ilustración 28 Ventana del explorador de Mesh.....	55
Ilustración 29 Ventana del explorador de música.....	55
Ilustración 30 Ventana del explorador de prefabricados.....	56
Ilustración 31 Editor de código.....	56
Ilustración 32 Ventana del explorador de sonidos.....	57
Ilustración 33 Ventana del explorador de Static Mesh.....	57
Ilustración 34 Ventana del explorador de texturas.....	58
Ilustración 35 Propiedades de un actor de tipo ZoneInfo.....	58
Ilustración 36 Propiedades de una superficie.....	58
Ilustración 37 Propiedades del nivel.....	59
Ilustración 38 Opciones avanzadas del editor.....	59
Ilustración 39 Opciones relativas a los viewports.....	59
Ilustración 40 Configuración de los viewports.....	59
Ilustración 41 Menú pincel.....	60
Ilustración 42 Menú construir.....	61
Ilustración 43 Menú herramientas.....	62
Ilustración 44 Barra de herramientas.....	63
Ilustración 45 Grupo “Modos de edición” de la caja de herramientas.....	66
Ilustración 46 Grupo “Brush Clipping” de la caja de herramientas.....	66



Ilustración 47 Grupo “Brushbuilders” de la caja de herramientas	66
Ilustración 48 Grupo “Brush Operations” de la caja de herramientas.....	66
Ilustración 49 Herramienta para la edición del terreno	69
Ilustración 50 Herramientas para la manipulación del terreno.....	70
Ilustración 51 Opciones de la herramienta Terrain Editing.....	71
Ilustración 52 Soft Selection (Terrain Editing)	72
Ilustración 53 Terrenos (Terrain Editing).....	72
Ilustración 54 Capas (Terrain Editing)	73
Ilustración 55 Decoradores (Terrain Editing).....	73
Ilustración 56 Generador de terreno	74
Ilustración 57 Grupo Brush Clipping	74
Ilustración 58 Marcadores de comienzo y fin de la “línea de clipping”	75
Ilustración 59 Grupo Brushbuilders	75
Ilustración 60 Ventana emergente para añadir pinceles especiales	77
Ilustración 61 Ventana emergente para definir una nueva "StaticMesh"	77
Ilustración 62 Tipos de volúmenes.....	78
Ilustración 63 UnrealEd Viewport.....	78
Ilustración 64 Top Viewport.....	79
Ilustración 65 Front Viewport	79
Ilustración 66 Side Viewport.....	79
Ilustración 67 3D Viewport	80
Ilustración 68 Barra de estado	80
Ilustración 69 Arquitectura de USARSim	84
Ilustración 70 Iridium GUI.....	90



Ilustración 71 Mapa vacío	92
Ilustración 72 Propiedades del Cube Builder	93
Ilustración 73 Brush con forma de cubo.....	93
Ilustración 74 Buscador de texturas.....	94
Ilustración 75 Paquete que contiene la textura para la superficie del cubo.....	95
Ilustración 76 Grupo de texturas "SolidColours" dentro del paquete UCGeneric	95
Ilustración 77 Textura elegida para la superficie del cubo	96
Ilustración 78 Cubo con la textura elegida	97
Ilustración 79 Cubo amarillo que delimita el mundo marciano	97
Ilustración 80 Seleccionar LimitationVolume.....	98
Ilustración 81 LimitationVolume	99
Ilustración 82 LimitationVolume alineado con el mapa	99
Ilustración 83 Actor "ZoneInfo"	101
Ilustración 84 Añadir al mapa un actor de tipo "ZoneInfo"	101
Ilustración 85 Seleccionar ZoneInfo Properties para acceder a sus propiedades	102
Ilustración 86 Propiedades de "ZoneInfo"	102
Ilustración 87 DrawScale a 20 para que el icono se perciba mayor	103
Ilustración 88 Propiedad "ZoneInfo" del actor "ZoneInfo"	104
Ilustración 89 Propiedad "ZoneLight" del actor "ZoneInfo"	105
Ilustración 90 Actor "Sunlight"	106
Ilustración 91 Añadir al mapa un actor de tipo "Sunlight"	107
Ilustración 92 Seleccionar Sunlight Properties para acceder a sus propiedades	107
Ilustración 93 Propiedad "LightColor" para el primer actor "Sunlight"	108
Ilustración 94 Propiedad "Lighting" para el primer actor "Sunlight"	109



Ilustración 95 Propiedad "LightColor" para el segundo actor "Sunlight"	110
Ilustración 96 Propiedad "Lighting" para el segundo actor "Sunlight"	110
Ilustración 97 Propiedad "LightColor" para los actores "Sunlight" tercero y cuarto ...	111
Ilustración 98 Propiedad "Lighting" para el tercer actor "Sunlight"	111
Ilustración 99 Propiedad "Lighting" para el cuarto actor "Sunlight"	112
Ilustración 100 Beagle cráter fotografiado por el rover Opportunity	113
Ilustración 101 Endeavour cráter fotografiado por el rover Opportunity	114
Ilustración 102 Volcanes en el engrosamiento de Tharsis (Mars Global Surveyor) ...	114
Ilustración 103 Melas Chasma - Valles Marineris (Mars Odyssey)	115
Ilustración 104 Terreno marciano fotografiado por el Viking 2	115
Ilustración 105 Editor de terreno	116
Ilustración 106 Crear un nuevo terreno	116
Ilustración 107 Configurar las opciones para la creación de un nuevo terreno.....	117
Ilustración 108 Mapa en el que se ha añadido un actor de tipo "TerrainInfo"	117
Ilustración 109 Mover el terreno a su origen.....	118
Ilustración 110 Propiedades del actor TerrainInfo	118
Ilustración 111 Buscador de texturas.....	119
Ilustración 112 Elegir el paquete AlleriaTerrain.utx	119
Ilustración 113 Textura elegida para la capa base del terreno.....	120
Ilustración 114 Definir las capas que tendrá el terreno	120
Ilustración 115 Crear una nueva capa para el terreno	121
Ilustración 116 "AlphaFill" para la primera capa del terreno	121
Ilustración 117 "ColorFill" para la primera capa del terreno	122
Ilustración 118 Configuración elegida para la primera capa del terreno.....	122



Ilustración 119 Elegir el paquete AbaddonTerrain.utx	123
Ilustración 120 Textura elegida para la segunda capa del terreno.....	123
Ilustración 121 Definir las capas que tendrá el terreno	124
Ilustración 122 Crear una nueva capa para el terreno	124
Ilustración 123 “AlphaFill” para la segunda capa del terreno.....	125
Ilustración 124 “ColorFill” para la segunda capa del terreno	125
Ilustración 125 Configuración elegida para la segunda capa del terreno	126
Ilustración 126 Terreno marciano desde la vista en 3D	126
Ilustración 127 Configuración elegida para "TerrainScale" y TerrainSectorSize"	127
Ilustración 128 Buscador de Static Meshes.....	127
Ilustración 129 Seleccionar el paquete SC_Volcano_M.usx.....	128
Ilustración 130 Seleccionar el "Static Mesh" Mountain1	128
Ilustración 131 "Static Mesh" que representa a un conjunto de montañas.....	129
Ilustración 132 Propiedades del "Static Mesh"	129
Ilustración 133 Propiedades del CubeBuilder	130
Ilustración 134 Seleccionar los vértices	131
Ilustración 135 Brush que servirá de molde para crear la rampa	131
Ilustración 136 Buscador de texturas.....	132
Ilustración 137 Textura elegida para la rampa	132
Ilustración 138 Buscador de texturas.....	134
Ilustración 139 Importar fichero para utilizar las texturas dentro del editor	134
Ilustración 140 Se selecciona la textura que se quiere importar.....	135
Ilustración 141 Importar la textura al mapa actual	136
Ilustración 142 Las cinco texturas ya están disponibles en el editor.....	136



Ilustración 143 Propiedades del CubeBuilder	137
Ilustración 144 Brush rojo para la construcción del cielo marciano	137
Ilustración 145 Buscador de texturas.....	138
Ilustración 146 Textura mars1_back* del paquete “myLevel”	138
Ilustración 147 3D viewport - Brush rojo después de haber hecho "Subtract"	139
Ilustración 148 Textura aplicada a la cara trasera del cubo.....	139
Ilustración 149 Textura aplicada a la cara delantera del cubo.....	139
Ilustración 150 Textura aplicada a la cara izquierda del cubo	140
Ilustración 151 Textura aplicada a la cara derecha del cubo.....	140
Ilustración 152 Textura aplicada a la cara superior del cubo	140
Ilustración 153 Textura mars1_front* del paquete “myLevel”	140
Ilustración 154 Aplicar textura mars1_front	141
Ilustración 155 Textura mars1_left* del paquete “myLevel”	141
Ilustración 156 Aplicar textura mars1_left.....	142
Ilustración 157 Textura mars1_right*	142
Ilustración 158 Aplicar textura mars1_right.....	143
Ilustración 159 Textura mars1_top*.....	143
Ilustración 160 Aplicar textura mars1_top	144
Ilustración 161 Composición de texturas para la creación del cielo marciano	145
Ilustración 162 Actor SkyZoneInfo.....	146
Ilustración 163 Añadir al cielo un actor de tipo SkyZoneInfo	146
Ilustración 164 SkyBox con SkyZoneInfo	147
Ilustración 165 Propiedades de SkyZoneInfo.....	147
Ilustración 166 Propiedades ZoneInfo y ZoneLight de SkyZoneInfo.....	148



Ilustración 167 Seleccionar las superficies.....	149
Ilustración 168 Seleccionar aquellas superficies pertenecientes al mismo brush	149
Ilustración 169 Acceder a las propiedades de las cinco caras del cubo	150
Ilustración 170 Propiedades de las caras del cielo - Habilitar la opción "Unlit".....	150
Ilustración 171 Habilitar las opciones "Unlit" y "Fake Backdrop"	151
Ilustración 172 Seleccionar NavigationPoint > SmallNavigationPoint > PlayerStart .	152
Ilustración 173 Añadir un actor de tipo "PlayerStart"	153
Ilustración 174 Icono usado para representar un actor de tipo "PlayerStart"	153
Ilustración 175 Abrir el paquete USARBot.u.....	154
Ilustración 176 Seleccionar Pawn > MultiView	154
Ilustración 177 Añadir al mapa un actor de tipo "MultiView".....	155
Ilustración 178 Objeto "MultiView"	155
Ilustración 179 "MultiView" ubicado en el techo del mapa.....	155
Ilustración 180 Comando para añadir un robot al entorno	156
Ilustración 181 Sistema de coordenadas.....	156
Ilustración 182 Conversión entre UU y m.....	156
Ilustración 183 Pitch, Yaw y Roll	157
Ilustración 184 Conversión entre UU y radianes-gradus	157
Ilustración 185 Iridium - Añadir P2AT al entorno marciano	157
Ilustración 186 P2AT añadido al entorno marciano.....	158
Ilustración 187 Fichero de configuración USARBot.ini [USARBot.P2AT].....	159
Ilustración 188 Comando para controlar el robot.....	159
Ilustración 189 Mover P2AT hacia delante a velocidad 1 rad/s.....	160
Ilustración 190 Girar P2AT a izquierda a 5 rad/s.....	160



Ilustración 191 Formas volcánicas (Static Mesh)	161
Ilustración 192 Rampas (Brush Clipped)	161
Ilustración 193 Cráter (deformación del terreno)	162
Ilustración 194 Montañas (deformación del terreno)	162
Ilustración 195 Atmósfera marciana.....	162
Ilustración 196 Principales hitos del proyecto.....	166
Ilustración 197 Diagrama de Gantt con los principales hitos del proyecto	169
Ilustración 198 Compilar USARSim.....	174
Ilustración 199 Comando para arrancar el servidor.....	175
Ilustración 200 Arrancar el servidor (modo manual)	175
Ilustración 201 Arrancar el servidor (modo automatizado).....	176
Ilustración 202 Comando para arrancar el cliente	176
Ilustración 203 Arrancar el cliente (modo manual).....	177
Ilustración 204 Carga del mapa en USARSim	177
Ilustración 205 Comando para arrancar el simulador.....	178
Ilustración 206 IridiumUI- Interfaz de control	178
Ilustración 207 Conectarse al servidor USARSim	179



ÍNDICE DE TABLAS

Tabla 1 Funciones del menú “Archivo”	46
Tabla 2 Funciones del menú “Edición”	47
Tabla 3 Funciones de "Select Surfaces"	49
Tabla 4 Funciones del menú "Ver"	52
Tabla 5 Funciones del menú "Pincel"	61
Tabla 6 Funciones del menú "Construir"	62
Tabla 7 Funciones del menú "Herramientas"	63
Tabla 8 Funciones de la barra de herramientas	65
Tabla 9 Funciones de la Caja de Herramientas	68
Tabla 10 Funciones para la manipulación del terreno	71
Tabla 11 Funciones del grupo “Brush Clipping” de la caja de herramientas.....	75
Tabla 12 Funciones del modo Brush Operations	77
Tabla 13 Funciones de la barra de estado	81
Tabla 14 Protocolo de comunicación (mensajes)	88
Tabla 15 Protocolo de comunicación (comandos)	89
Tabla 16 Valores de Z para cada robot.....	165
Tabla 17 Total de horas para cada tarea	168
Tabla 18 Costes de personal	170
Tabla 19 Costes de equipos	170
Tabla 20 Costes de software	171
Tabla 21 Costes de material fungible	171
Tabla 22 Coste total del proyecto	172



1. INTRODUCCIÓN AL PROBLEMA A RESOLVER

1.1 Problema general

1.1.1 La exploración del espacio

1.1.1.1 Breve historia

La crónica de la exploración espacial se puede dividir en cuatro grandes etapas: inicial, heroica, de consolidación y de comercialización.

Etapa inicial – Los orígenes de la exploración espacial

Este período va desde el redescubrimiento en 1903 de la ecuación del cohete por “El Padre de la Cosmonáutica” Konstantin Tsiolkovsky hasta los V-2 nazis desarrollados a comienzos de la Segunda Guerra Mundial. Algunas personas consideran a los V-2 como el primer vehículo espacial. Sin embargo, esto no es cierto debido a que son incapaces de permanecer en el espacio o entrar en órbita. Se consideran más bien los primeros SRBM¹ y los antecesores inmediatos del cohete espacial.



Ilustración 1 Lanzamiento de un V-2 en 1943²

¹ Para más información consultar http://en.wikipedia.org/wiki/Short-range_ballistic_missile

² Fuente: Deutsches Bundesarchiv (German Federal Archive)



Etapa heroica – Los héroes de la exploración espacial

Se inicia en 1957 con el lanzamiento por parte de la Unión Soviética del Sputnik 1, las primeras naves espaciales que alcanzaron la superficie lunar en 1959 (Luna 2 y Luna 3) y el vuelo de Yuri Gagarin en 1961 a bordo del Vostok 1 (duró 1 hora y 48 minutos). Finaliza aproximadamente con las misiones interplanetarias a Venus iniciadas en 1965 y con los vuelos tripulados a la Luna (Apolo 11 a Apolo 17) que tuvieron lugar entre 1969 y 1972. De estos últimos merece la pena destacar dos:

- Apolo 11: misión liderada por los astronautas Amstrong y Aldrin; tuvo una gran repercusión social en la época, ya que fue la primera vez que el hombre logró pisar la superficie lunar.
- Apolo 17: fue la última misión tripulada a la Luna en la que los astronautas Ron Evans y Eugene Cernan permanecieron durante tres días en su superficie.



Ilustración 2 Sputnik 1³

³ Fuente: <http://www.nationalmuseum.af.mil>

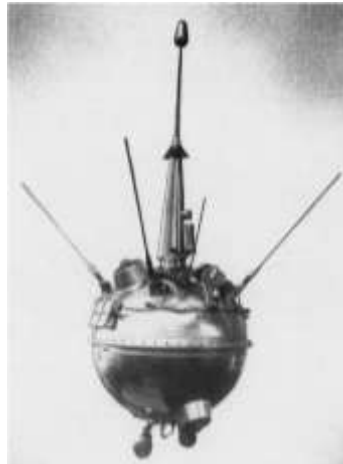


Ilustración 3 Sonda espacial Luna 2 ⁴



Ilustración 4 Apolo 17, viaje final a la Luna ⁵

Etapa de consolidación – Las grandes inversiones en la exploración espacial

Esta fase supuso una gran apuesta por parte de los EEUU: el transbordador espacial. Se comenzó a fabricar en los años 70 y voló por primera vez en 1981. Este mismo año 2011, después de treinta en activo y tras 135 misiones, se procedió a su retirada sin que hubiera logrado ni uno sólo de sus objetivos iniciales y tras haber dejado un balance de catorce astronautas muertos y dos transbordadores destruidos. La URSS, por su parte,

⁴ Fuente: <http://nssdc.gsfc.nasa.gov/database/MasterCatalog?sc=1959-014A>

⁵ Fuente: <http://www.hq.nasa.gov/alsj/a17/ap17-KSC-72PC-589.jpg>



apostó por un amplio programa de estaciones espaciales entre 1971 y 1986 (Salyut 1 - Salyut 7) que finalizaría con la mítica Mir (1986-2001).



Ilustración 5 Primer lanzamiento del transbordador espacial en 1981 ⁶



Ilustración 6 La estación espacial Mir el 12 de junio de 1998 ⁷

⁶ Fuente: <http://dayton.hq.nasa.gov/IMAGES/LARGE/GPN-2000-000650.jpg>

⁷ Fuente: <http://spaceflight.nasa.gov/gallery/images/shuttle/sts-91/hires/91727051.jpg>



Etapa de comercialización – La rentabilidad económica de la exploración espacial

Por primera vez en toda su historia los objetivos se dirigen hacia la rentabilidad económica. La caída de la URSS en el año 1991 y las nuevas políticas económicas en Occidente llevan a una reducción del gasto público y a la liberalización de la economía. Esto trae consigo el aplazamiento y/o la supresión de grandes programas espaciales (regreso a la Luna, viajes tripulados a Marte y a Venus, inicio de la exploración de las lunas de Júpiter y de Saturno o la industrialización del espacio) en beneficio de otros con una rentabilidad más inmediata o fruto de la cooperación internacional. Un ejemplo de esto último es la Estación Espacial Internacional (ISS) ⁸ recientemente concluida.

Pero, en estos tiempos, ¿en qué punto se encuentra exactamente la exploración espacial? ¿Hacia dónde avanza? ¿A qué nuevos retos se enfrenta? En los siguientes apartados se dará respuesta a estas cuestiones.

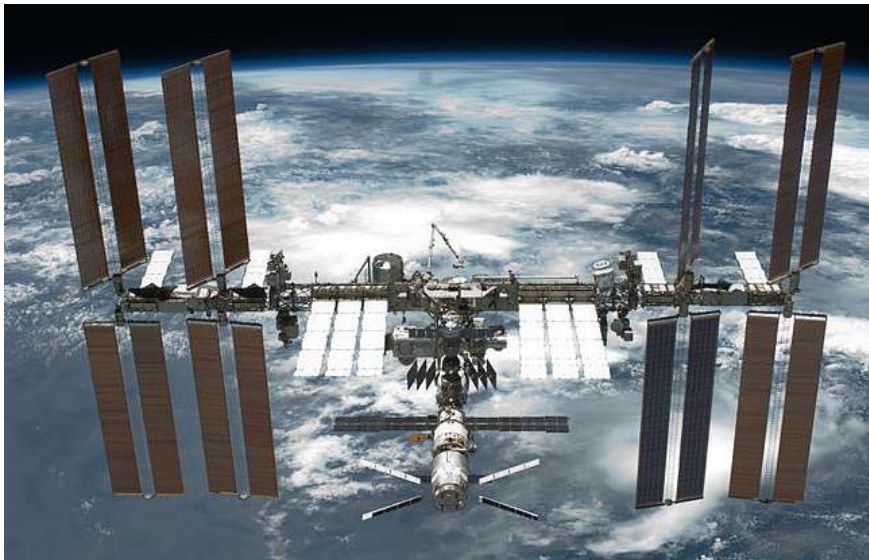


Ilustración 7 La estación espacial internacional fotografiada el 30 de mayo de 2011 ⁹

⁸ Para más información consultar <http://es.wikipedia.org/wiki/ISS>

⁹ Fuente: <http://www.spaceflight.nasa.gov/gallery/images/shuttle/sts-134/html/s134e010137.html>



1.1.1.2 Situación actual de la exploración espacial

Atrás quedaron los años dorados de los vuelos tripulados al espacio. Pese a los numerosos esfuerzos que ha venido realizando la comunidad científica internacional a lo largo de todo este tiempo aún no se han conseguido superar las barreras biológicas y tecnológicas que impiden dar un paso adelante en la exploración espacial con seres humanos. Esta presenta, hoy por hoy, numerosos riesgos que no vale la pena correr.

Los tres grandes problemas a los que se enfrenta la exploración espacial con humanos

Las principales dificultades que existen en el presente para enviar seres humanos a explorar el espacio son de tres tipos: biológicas, mecánicas y económicas. Las primeras han sido el principal obstáculo para los viajes interplanetarios en el espacio debido a que resulta difícil luchar contra las propias condiciones del cuerpo humano para la gravedad, el aire y el agua. Las mecánicas están relacionadas con la tecnología que hace posible la vida en el espacio. Las últimas, por su parte, son determinantes porque condicionan directamente las dos anteriores.

Biológicas

Actualmente se sabe de los efectos negativos a largo plazo que tiene para la salud de los astronautas la vida en el espacio. Durante las misiones espaciales pueden contraer diferentes enfermedades relacionadas con la ingravidez y la exposición a las radiaciones ionizantes.

- Pérdida de calcio en el cuerpo. Esto tiene consecuencias fatales para el sistema nervioso y óseo (síndrome de Allison o desmineralización de huesos).
- También el corazón (su tamaño y función cardíaca), los pulmones y los músculos (atrofia muscular) se ven afectados por la gravedad.
- Muerte celular, tumores, cataratas y alteraciones en los genes debidas a las radiaciones.



- Fatiga, somnolencia, dolor de cabeza, mareo y en ocasiones pérdida de conciencia a causa de la falta de oxígeno.
- Visión borrosa a consecuencia de los cambios bruscos de velocidad dentro de la nave durante el despegue, el aterrizaje o mientras se mueve.

Mecánicas

El entorno del espacio requiere del desarrollo de un sistema de apoyo a la vida que sea capaz de reciclar fundamentalmente aire y agua en un “entorno cerrado” por un período suficientemente largo para soportar el viaje a otro planeta. El hecho de que la Estación Espacial Internacional tenga que ser reabastecida con aire y agua dan cuenta de este problema.

- La tecnología que hace posible la vida en el espacio no es todavía capaz de soportar a una tripulación de astronautas durante un período de un año (el tiempo mínimo requerido para un viaje a Marte).

Económicas

Los programas espaciales centrados en exploradores humanos son muy caros debido a las dos limitaciones anteriores: biológicas y mecánicas.

- Se necesitan una gran cantidad de recursos para preservar la vida humana en el espacio: comida, agua, aire, sistemas de control ambiental, instalaciones sanitarias, equipos de seguridad, etc. Todos estos elementos añaden un coste enorme a cualquier misión espacial.

Enviar un hombre al espacio, hoy en día, le cuesta al gobierno americano unos 65 millones de dólares. ¹⁰En estos tiempos de grandes penurias económicas, este factor ha sido el más decisivo a la hora de llevar al presidente de los EEUU, Barack Obama, a

¹⁰ Fuente consultada: http://www.elmundo.es/america/2010/04/15/estados_unidos/1271359408.html



cancelar futuras misiones tripuladas como “Constellation”, que pretendía enviar al hombre de vuelta a la Luna en 2020. También el periódico El Correo de Andalucía en su edición digital del 16 de abril de 2010¹¹ se hacía eco de las palabras del mandatario: “Para mediados de la década de 2030 creo que podremos enviar misiones tripuladas a la órbita de Marte y conseguir que vuelvan sanas y salvas a la Tierra”.

Aún con todo, la exploración espacial se perfila en el presente y también de cara a un futuro como una disciplina de bastante utilidad dentro de la astronáutica en la cual están participando un gran número de países (Estados Unidos, Rusia, Europa, China, Japón e India principalmente). El presupuesto que estos destinan a los programas espaciales se reparte fundamentalmente entre tres sectores: militar, científico y comercial.

Lanzamientos estatales/militares

Suponen el 50% del total, en mayor medida en EEUU que en Rusia.¹²

Lanzamientos científicos

En los últimos años apenas superan el 15% de los lanzamientos totales.

Lanzamientos comerciales

Las cifras proporcionadas por la Administración Federal para la Aviación (FAA)¹³ de los EEUU demuestran que este sector de la exploración espacial está en auge en Rusia y en Europa (Ver ilustración 10). Actualmente, entre el 26% y el 33% de los lanzamientos espaciales que se realizan en Estados Unidos son comerciales (Ver ilustraciones 8 y 9). Pero sin duda alguna quién manda en este sector es Rusia, que acapara entre el 44% y el 65% del total de los lanzamientos seguida por Europa en segundo lugar (Ver

¹¹ Fuente consultada: <http://www.elcorreoweb.es/actualidad/091095/obama/nasa>

¹² Fuente consultada: <http://www.foxnews.com/scitech/2011/05/09/military-launches-new-missile-warning-space/>

¹³ Para más información consultar http://es.wikipedia.org/wiki/Federal_Aviation_Administration



ilustraciones 11 y 12). La ex potencia soviética ofrece gran fiabilidad a bajo coste lo cual la ha situado a la cabeza del sector.

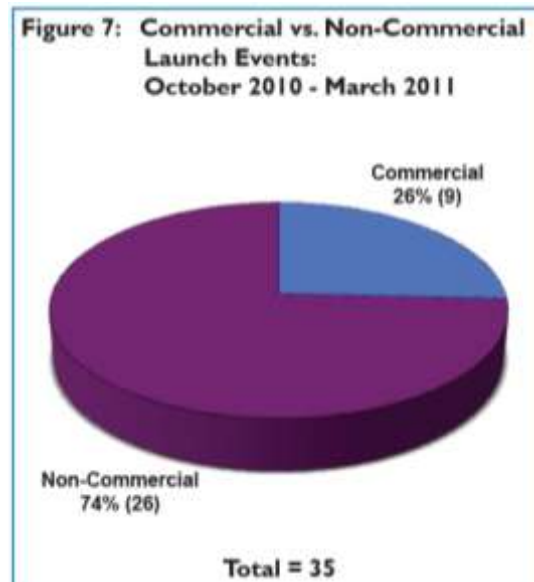


Ilustración 8 Lanzamientos comerciales frente a no comerciales en EEUU ¹⁴

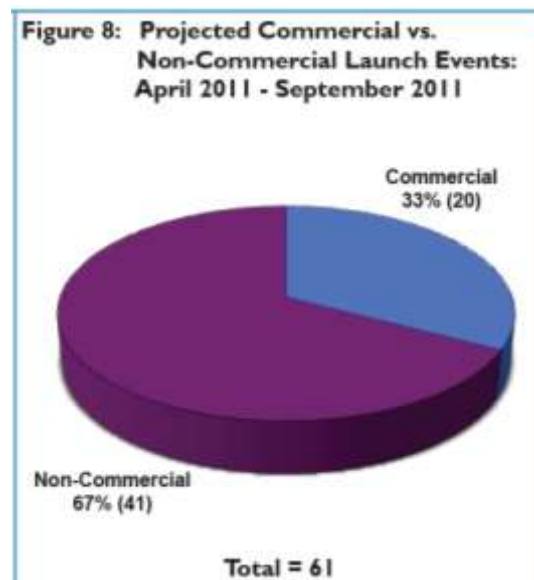


Ilustración 9 Estimación de lanzamientos comerciales vs. no comerciales en EEUU ¹⁵

¹⁴ Fuente:

http://www.faa.gov/about/office_org/headquarters_offices/ast/media/Semi%20Annual%20Launch%20Report%20April%2018%202011.pdf

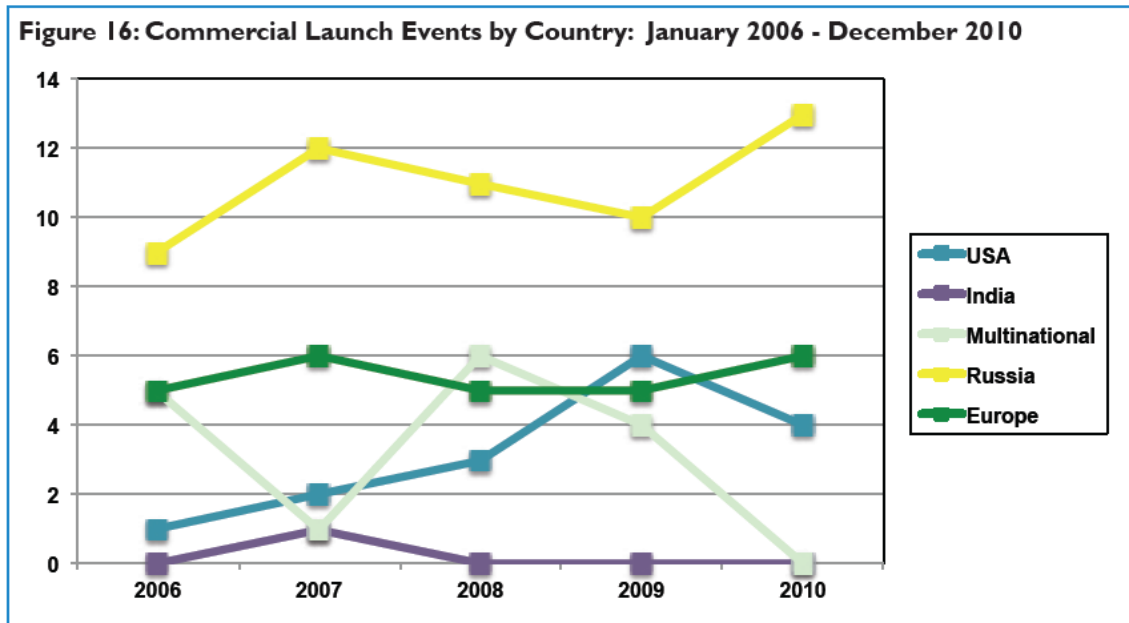


Ilustración 10 Evolución de los lanzamientos espaciales comerciales ¹⁶

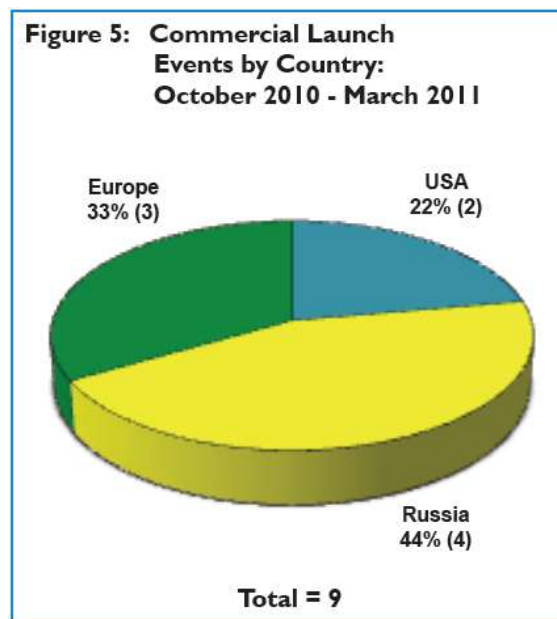


Ilustración 11 Lanzamientos espaciales comerciales por país ¹⁷

¹⁵ Fuente: Ídem

¹⁶ Fuente:

http://www.faa.gov/about/office_org/headquarters_offices/ast/media/Semi%20Annual%20Launch%20Report%20April%2018%202011.pdf

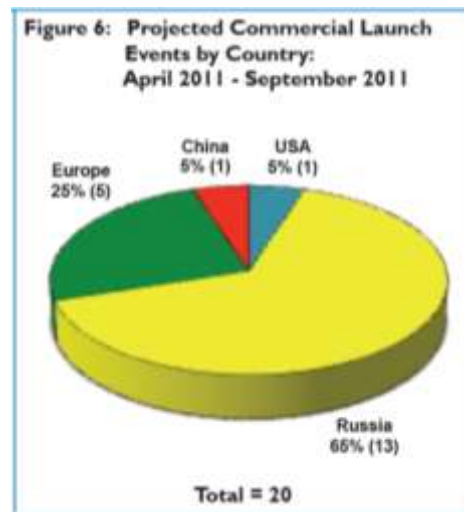


Ilustración 12 Lanzamientos espaciales comerciales estimados por país ¹⁸

Con un presupuesto que se ve recortado año tras año y sin nueva tecnología que permita alargar y hacer más seguras las estancias del hombre en el espacio, ¿cuál será el futuro inmediato de la exploración espacial?

En el año 1996, ubicada dentro del programa Discovery de la NASA que promueve el envío de naves de bajo coste y de lanzamientos frecuentes bajo la política “más barato, más rápido y mejor”, tiene lugar la misión Mars Pathfinder (Sojourner). Esta permitió a los científicos analizar la atmósfera, el clima, la geología, la composición de las rocas y el suelo marciano.

Los acontecimientos que vienen sucediéndose desde entonces como parte del programa de la NASA Mars Exploration (año 2004 Spirit y Opportunity y, más recientemente, el que tendrá lugar este mismo año, entre el 25 de Noviembre y el 18 de Diciembre, Mars Science Laboratory) parecen apuntar a que serán los vehículos robóticos de exploración, comúnmente denominados rovers, los que tomen el relevo de los astronautas.

¹⁷ Fuente: Ídem

¹⁸ Fuente:

http://www.faa.gov/about/office_org/headquarters_offices/ast/media/Semi%20Annual%20Launch%20Report%20April%2018%202011.pdf



1.1.2 La exploración robótica del espacio

1.1.2.1 Breve historia

Durante los años 60 el programa tripulado Apolo acaparaba toda la atención pero no era el único proyecto de exploración espacial en marcha. A principios de 1962 la NASA lanzó sondas robóticas que sobrevolaron los planetas Venus, Mercurio y Marte y enviaron de vuelta a la Tierra fotografías de ellos. Más tarde, en los años 70 naves robóticas espaciales más sofisticadas aterrizaron en Marte o fueron enviadas a sobrevolar los planetas exteriores (Júpiter, Saturno, Urano y Neptuno). A estas misiones se les dieron nombres heroicos: Mariner, Viking y Voyager. Fue en los años 90 cuando la NASA perdió tres naves robóticas espaciales en su viaje a Marte. Tan sólo una consiguió llegar a la superficie marciana. A comienzos del año 2000 la NASA envió dos nuevas misiones a Marte. La primera se denominó Mars Odyssey. Estaba provista de una sonda que entró en órbita alrededor del planeta en 2001. Dos años más tarde se le unió la sonda de la Agencia Espacial Europea (ESA)¹⁹ Mars Express. Desafortunadamente, una sonda espacial de esta misión se perdió en su camino a la superficie.

Inicialmente las misiones de exploración robótica del espacio funcionaban en su mayoría como grandes cámaras que estaban en el aire. Estos instrumentos han evolucionado con el paso del tiempo para incluir imágenes por radar más sofisticadas, reproductores de imágenes multiespectrales, espectrómetros, detectores de onda de gravedad, un gran número de sensores ubicados de antemano y, más recientemente, rovers.

A diferencia de la exploración humana del espacio, la robótica no supone ningún riesgo para el hombre, evita las pérdidas humanas y ante todo es mucho más barata.

¹⁹ Página web oficial: [http:// www.esa.int/](http://www.esa.int/)



Las ventajas de la exploración espacial con robots

- Existen muchos entornos en los que no es seguro o rentable enviar a seres humanos.
- Es mucho más barato construir robots y enviarlos al espacio para realizar tareas de exploración.
- Los robots tienen menos necesidades (alimento, agua y oxígeno) que las personas.
- Las máquinas pueden construirse de tal forma que soporten condiciones inhóspitas (temperaturas inferiores a los $-105\text{ }^{\circ}\text{C}$, altas presiones, lluvias ácidas).
- Los robots no se ven afectados por la radiación espacial como los astronautas
- Los robots pueden llegar a sitios que son inaccesibles para las personas
- Se tardan muchos años en viajar a los planetas exteriores. Con la tecnología actual esto es demasiado tiempo para un viaje con humanos.
- Los robots constituyen una valiosa herramienta que aporta información que ayudará a científicos y a astronautas a preparar mejor los futuros viajes al espacio.

En definitiva, la exploración espacial con humanos es mucho más cara, inconveniente y peligrosa que la realizada con robots. Por ello no es de extrañar que en estos últimos años se hayan dado grandes pasos en esta dirección.

La exploración espacial de otros planetas se realizará en un futuro inmediato principalmente con rovers y ordenadores que serán controlados por humanos desde la Tierra. Sin embargo, incluso para las máquinas esta tarea resulta difícil.

En abril de 2009, el rover Spirit quedó atrapado en un banco de arena en Marte cuando se desplazaba por unas capas de suelo aparentemente duras que cubrían una zona arenosa. Mientras tanto en la Tierra, los científicos utilizaban una réplica del vehículo robótico de exploración en la denominada sala de arena del JPL (Jet Propulsion



Laboratory²⁰). En esta sala de pruebas se simuló la situación real del vehículo y se ensayaron posibles maniobras de salida, antes de mandarle las órdenes necesarias. Pero los graves problemas de movilidad que presentaba y la falta de tracción de sus ruedas hicieron que después de varios meses de intentos fallidos tratando de liberarlo la NASA optara finalmente por redefinir la misión.²¹

El ejemplo anterior sirve para ilustrar el hecho de que antes de llevar un rover a Marte o a cualquier otro planeta es muy importante llevar a cabo numerosas pruebas en la Tierra: primero con un simulador, contemplando todas las situaciones posibles, y una vez haya funcionado en este, el siguiente paso es construir una réplica del mismo y realizar más pruebas hasta que todo esté listo para enviarlo al espacio.

1.2 Particularización del problema dentro del marco general

Como ya se ha visto anteriormente toda misión espacial lleva consigo una complejidad técnica y un coste muy elevado. Por este motivo antes de enviar cualquier vehículo robótico de exploración al espacio primero hay que realizar numerosas pruebas en la Tierra. Para llevar a cabo esta importante tarea hace falta reproducir un entorno que cumpla con las mismas condiciones que se dan en el planeta de destino.

Este proyecto consiste en la creación y simulación de un entorno marciano para un rover que tendrá como objetivo explorar la superficie del planeta Marte. A la hora de crear el entorno se han tenido muy en cuenta las siguientes cuestiones: características del terreno marciano y peculiaridades de la atmósfera marciana.

²⁰ Página web oficial: <http://www.jpl.nasa.gov/>

²¹ Fuente consultada:

http://www.elpais.com/articulo/futuro/NASA/disena/futuros/vehiculos/recurreran/Luna/elpepusocfut/20090729elpepifut_1/Tes



La simulación de entornos marcianos es útil para resolver el problema de la exploración robótica (tratado en el punto 1.1.2) ya que permite:

- Llevar a cabo numerosos experimentos en un entorno muy similar al real sin que ello suponga ningún riesgo para el robot (daños/desperfectos que lo dejen inservible/inutilizable).
- Recrear de manera simultánea diferentes tipos de situaciones (tormentas de arena, desniveles en el terreno) y estudiar a posteriori los efectos que tienen en el comportamiento del robot.
- Desarrollar modelos mucho más precisos y robustos (cuántas más pruebas se realicen en el entorno simulado mejores serán los diseños y por tanto los resultados finales).
- Tener una visión global del comportamiento del robot en el entorno que ayuda a mejorar aquellos aspectos en los que aún falla.
- Ahorrar mucho tiempo, esfuerzo y dinero (a la vez que se van desarrollando nuevos componentes se van probando, no hay que esperar a verlos terminados para poder realizar las pruebas pertinentes).

La recreación del entorno puede hacerse con ayuda de una herramienta de simulación. Para el presente trabajo se ha utilizado el simulador USARSim porque ofrece las siguientes ventajas:

- Incluye robots simulados, a diferencia de otros simuladores (OpenRAVE)
- Es el que utiliza el NIST
- Es el simulador que se usa en la RoboCup, la competición de robótica más importante del mundo.



- Permite realizar cambios en el entorno en muy poco tiempo y con mucha facilidad: añadir/quitar elementos (cráteres, volcanes, valles, rocas), modificar la posición inicial del robot, jugar con la iluminación para recrear el día y la noche, añadir efectos del tipo tormentas de arena, etc.
- Permite reproducir los movimientos del robot en la superficie marciana con una gran precisión, evitándose de este modo situaciones como la que tuvo lugar en el año 2009 con el rover Spirit.
- Permite tener múltiples robots desplegados en diferentes posiciones del entorno y estudiar cómo interaccionan y colaboran entre ellos para realizar una tarea común.

Para conocer más detalles sobre el simulador ir al punto 2.1.4 del presente documento.

1.2.1 Otros simuladores

Actualmente existe una gran variedad de simuladores en el mercado que resultarán más o menos adecuados según lo que se quiera hacer. Entre los más conocidos se encuentran:

- OpenRAVE: proporciona un entorno para el desarrollo, testeo y despliegue de algoritmos de planificación de movimiento en aplicaciones de robótica para la vida real. Se centra en la parte de simulación y en el análisis de la información geométrica y cinemática relacionada con la planificación del movimiento.²²
- robotSuite: permite simular robots en un entorno 3D realista que incluye simulación física de gran precisión proporcionada por Ageia PhysX. También facilita enormemente el diseño y las pruebas de los algoritmos de control del robot. Todos los entornos y robots son configurables. Asimismo, está diseñado para ser usado en cualquier industria.²³

²² Página web oficial: <http://openrave.programmingvision.com/en/main/index.html>

²³ Página web oficial: <http://www.cogmation.com/index.html>



- Gazebo: simulador multi-robot 3D para entornos de exterior que forma parte del proyecto Player/Stage. Se trata de software libre distribuido bajo la licencia pública GNU. Es capaz de simular una población de robots, sensores y objetos así como de generar interacciones físicas entre objetos creíbles.²⁴
- Webots: software para la simulación de robots móviles utilizado en numerosas universidades y centros de investigación alrededor del mundo. Permite al usuario diseñar configuraciones del robot complejas en un entorno compartido. También dispone de una gran variedad de sensores simulados y de actuadores con los que se puede equipar a los robots.²⁵

1.2.2 Otros modos de resolver el problema (sin simulador)

A continuación se presentan los trabajos más importantes que se han realizado o que continúan desarrollándose actualmente en este sentido.

1.2.2.1 FMARS (Flashline Mars Artic Research Station)

En el verano de 2000 la “Mars Society” instaló esta primera estación (de las cuatro programadas) para la investigación de la exploración de la superficie marciana. Las zonas en las que están ubicadas se eligieron por sus similitudes físicas con Marte. En concreto, esta primera se situó en la ladera norte del cráter Haughton en la isla de Devon (archipiélago ártico canadiense).

Las investigaciones científicas realizadas en el año 2009 se centraron en las tareas de localización y muestreo de un depósito de “gypsum” —un mineral de sulfato de calcio hidrogenado en cuya composición hay un 20% de agua y que se encuentra en abundancia en muchas regiones de Marte— situado en las montañas Gemini (cráter Haughton).

²⁴ Página web oficial: <http://playerstage.sourceforge.net/gazebo/gazebo.html>

²⁵ Página web oficial: <http://www.cyberbotics.com/>



Ilustración 13 Vista de Flashline Mars Arctic Research Station en Julio de 2009 ²⁶

1.2.2.2 MDRS (*Mars Desert Research Station*)

Es el segundo de los cuatro hábitats simulados para la exploración de la superficie de Marte dirigido por la “Mars Society”. La estación de investigación está situada en una zona desértica al sur de Utah (EEUU).

Las principales tareas de investigación que se llevan a cabo están relacionadas con el estudio de organismos capaces de vivir en entornos extremos. Otros experimentos incluyen un estudio del efecto que tienen las actividades extra vehiculares, EVA²⁷, en el ritmo cardíaco y en la presión sanguínea de los miembros de la tripulación.



Ilustración 14 Mars Desert Research Station: hábitat, observatorio e invernadero

²⁶ Autor: <http://commons.wikimedia.org/wiki/User:Brianshiro>

²⁷ Para más información consultar http://en.wikipedia.org/wiki/Extra-vehicular_activity



1.2.2.3 Euro-MARS (*European Mars Analogue Research Station*)

A principios del año 2002 con ayuda del programa científico que le da su nombre (Euro-MARS), en el que participaron países como Reino Unido, Francia, Italia, Bélgica y España, se logró financiar y construir esta tercera estación. Sin embargo no sería hasta el año 2007 cuando darían comienzo las operaciones de campo.

Está ubicada cerca del volcán Krafla Rift, a 15 kilómetros de Myvatn (Islandia). Las características de origen volcánico que presenta esta región, similares a las existentes en la superficie marciana, fueron decisivas a la hora de elegirla.

Su propósito es permitir a los científicos e ingenieros conducir una exploración geológica y biológica en las mismas condiciones que se dan en Marte con los objetivos siguientes:

- Desarrollar tácticas de campo basadas en dichas exploraciones
- Probar las características de diseño del hábitat y las herramientas
- Evaluar los protocolos de selección de la tripulación

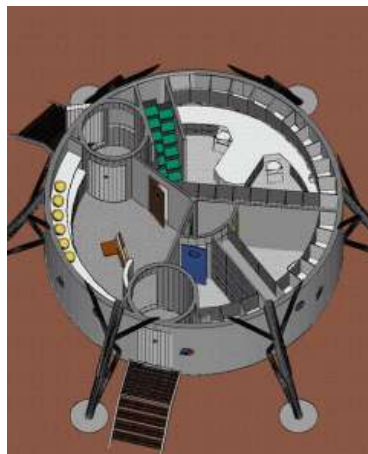


Ilustración 15 Euro-MARS: diseño del hábitat ²⁸

²⁸ Fuente: <http://euomars.net/>



1.2.2.4 *MESCH (Mars Environmental Simulation Chamber)*

Esta cámara de simulación de entorno marciano desarrollada por un equipo interdisciplinar de la facultad de Ciencias Naturales de la Universidad de Aarhus (Dinamarca) permite recrear las condiciones sobre e inmediatamente debajo de la superficie de Marte.

Los experimentos realizados —de los que se hacía eco la revista especializada *Astrobiology* en el año 2008— sugerían que alguna forma de vida podría existir debajo de la superficie marciana.

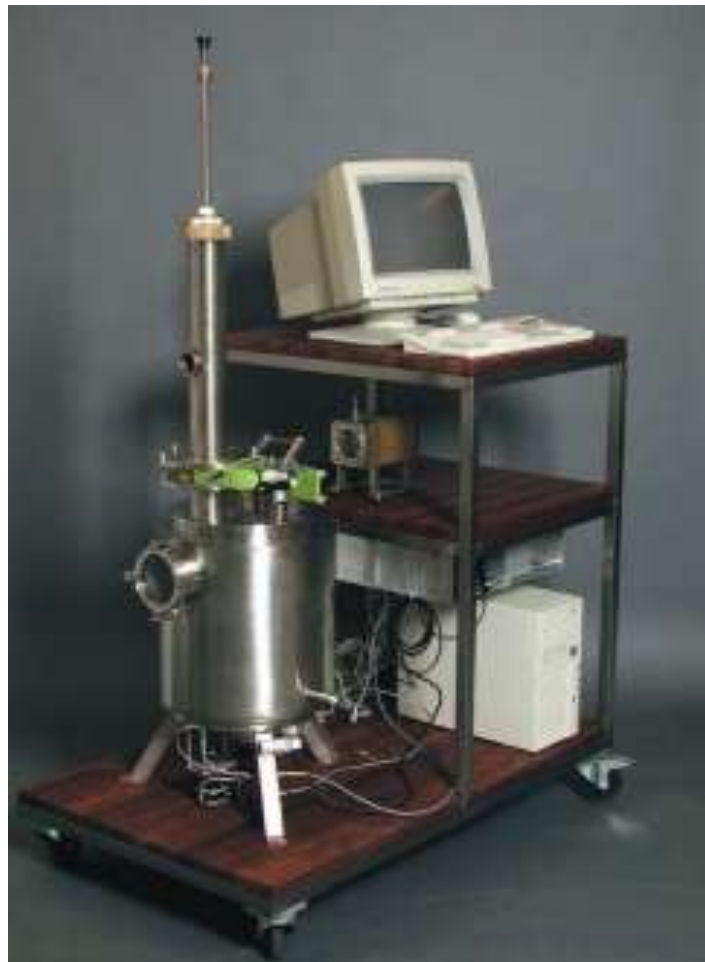


Ilustración 16 MESCH (Cámara de simulación de entorno marciano)



1.3 Organización del documento

En esta sección se realiza una breve introducción al resto de las cuestiones que van a ser abordadas en esta segunda parte de la memoria.

En primer lugar se hablará de las herramientas que se han utilizado para la creación del entorno marciano y para la simulación del mismo y del rover. Asimismo se muestran los conocimientos que han sido necesarios adquirir para poder acometer estas dos tareas con éxito.

En segundo término, se describe fundamentalmente qué es lo que se ha hecho en este proyecto y lo que es más importante cómo se ha procedido. Se explicará paso a paso cómo se han ido completando las distintas etapas del proyecto hasta llegar a la solución final.

El último apartado de este bloque muestra la solución final aportada y cómo esta resuelve lo tratado en el punto 1.2.

En el penúltimo apartado de la memoria se recapitula todo el trabajo realizado, se enumeran los objetivos que se han alcanzado, se detalla el tiempo que ha llevado realizar el proyecto y se presenta una ficha técnica en la que se recogen entre otras cosas el coste de las herramientas usadas y el número de horas y precio/hora de un ingeniero para realizar este trabajo.

Finalmente se presentan algunas sugerencias como posibles mejoras de cara a un futuro.



2. RESOLUCIÓN DEL PROBLEMA

2.1 Metodología

En este apartado se describen las principales herramientas usadas con motivo de la creación de un entorno marciano para el simulador USARSim²⁹. Este se basa en el motor del juego Unreal Tournament³⁰, Unreal Engine (de ahora en adelante UE), que a su vez utiliza Unreal Editor (comúnmente UnrealEd) como editor para la manipulación de contenido (geometría, actores, volúmenes...) en Unreal.

Resulta conveniente seguir este mismo esquema lógico, de modo que se comenzará esta sección hablando en primer lugar de Unreal Tournament y de UE en términos generales, para seguidamente abordar en mayor profundidad UnrealEd. En última instancia se explicará el simulador, prestando especial atención a aquellas cuestiones directamente relacionadas con el proyecto.

2.1.1 *Unreal Tournament*

Videjuego desarrollado por Digital Extremes³¹, fue publicado en Noviembre de 1999 por GT Interactive³² (hoy en día Atari, Inc.) para la plataforma PC. Posteriormente fue relanzado para PlayStation® 2, Linux, Dreamcast y Mac. Es el primero de la serie Unreal Tournament y el segundo de la serie Unreal. Cabe mencionar que el mismo año de su publicación ('99) ganó un premio al mejor videojuego del año (GOTY).

2.1.1.1 *Unreal Tournament 2004 (UT2004)*

Es el tercero de la saga Unreal Tournament y fue lanzado en Marzo de 2004. Incluye UnrealEd 3 y utiliza UE2 como versión del motor. Estas características han sido

²⁹ Página web oficial: <http://usarsim.sourceforge.net/>

³⁰ Página web oficial: <http://www.unrealtournament.com/>

³¹ Página web oficial: <http://www.digitalextremes.com/>

³² Para más información consultar http://es.wikipedia.org/wiki/GT_Interactive



determinantes, como se verá en el siguiente apartado, a la hora de elegirlo como herramienta principal de trabajo para el proyecto.

2.1.2 *Unreal Engine*

Motor de juegos desarrollado por la compañía Epic Games³³ en el año 1998 para el FPS (shooter³⁴ en primera persona) Unreal. Aunque inicialmente fue creado para este tipo de videojuegos, posteriormente se ha empleado con bastante éxito en una gran variedad de géneros, entre los que destacan los de rol: stealth³⁵, MMORPG³⁶ y RPG³⁷. Está escrito en el lenguaje de programación C++ y es compatible con múltiples plataformas como PC (Microsoft Windows, GNU/Linux), Apple Macintosh (Mac OS, Mac OS X) y la mayoría de consolas (Dreamcast, GameCube, Wii, Xbox, Xbox 360®, PlayStation® 2 y PlayStation® 3). Algunos ejemplos de juegos que lo utilizan son: *Unreal Tournament*, *Deus Ex*, *Turok*, *Tom Clancy's Rainbow Six: Vegas*, *America's Army*, *Red Steel*, *Gears of War*, *BioShock*, *BioShock 2*, *Star Wars Republic Commando* o *Batman: Arkham Asylum*.

Actualmente existen tres versiones: Unreal Engine 1, Unreal Engine 2 y Unreal Engine 3. De entre todas ellas, se ha elegido la segunda por los motivos que se detallan a continuación:

- La primera generación se encuentra en desuso al estar obsoleta
- La tercera generación presenta problemas con las colisiones y con la capacidad de conducción de los robots en USARSim.
- La segunda generación es la más utilizada actualmente por USARSim

³³ Página web oficial: <http://www.epicgames.com/>

³⁴ Para más información consultar http://es.wikipedia.org/wiki/Videojuego_de_disparos

³⁵ Para más información consultar http://en.wikipedia.org/wiki/Stealth_game

³⁶ Para más información consultar http://en.wikipedia.org/wiki/Massively_multiplayer_online_role-playing_game

³⁷ Para más información consultar http://en.wikipedia.org/wiki/Role-playing_game



Una cuarta generación, Unreal Engine 4, está todavía en proceso de producción y no estará lista hasta 2012-2018.³⁸

2.1.2.1 *Unreal Engine 1 (UE1) – 1998*

Este motor se usó en los videojuegos Unreal y Unreal Tournament.

Características:

- Generación de imágenes a partir de un modelo
- Detección de colisiones
- IA (Inteligencia Artificial)
- Visibilidad
- Redes
- Manipulación de archivos de sistema

2.1.2.2 *Unreal Engine 2 (UE2) – 2002*

Este motor hizo su debut con el videojuego America's Army.

Características:

- Reescritura íntegra del código del núcleo y del motor de procesamiento de gráficos
- Nueva versión del editor (UnrealEd 3) que incluye un editor de sistema de partículas.
- SDK (Software Development Kit) de Karma physics
- Soporte para PlayStation® 2, GameCube y Xbox
- Mejora del rendimiento
- Físicas para vehículos
- Soporte 64 bits en Unreal Tournament 2004

³⁸ Fuente consultada: <http://arstechnica.com/gaming/news/2008/07/epic-games-unreal-engine-4-ready-in-2012.ars>



2.1.2.3 *Unreal Engine 3 (UE3) – 2006*

Este motor se usó en los videojuegos Gears of War y Unreal Tournament 3 aunque también se ha empleado en otros sectores: simulación de construcciones, simuladores de conducción, pre visualización de películas y generación de terrenos.

Características:

- Plataformas de desarrollo: PC (con soporte DirectX9/10/11), Xbox 360®, iOS, PlayStation® 3 y Android.
- HDRR
- Normal mapping
- Sombras dinámicas
- Sustitución de Karma por PhysX de NVIDIA
- Animaciones faciales mediante FaceFX

2.1.3 *Unreal Editor (UnrealEd)*

Paquete de herramientas que permite trabajar con contenido en Unreal Engine. Fundamentalmente, se utiliza como editor para la creación de niveles de un videojuego, pero también incluye editores y buscadores para importar y manipular contenido.

Con cada versión del motor aparece también una nueva del editor. Además, cada juego basado en esta plataforma tiene su propia versión del editor.

- *UnrealEd 1.0*
 - Primera versión del editor
 - Escrito en Visual Basic
 - Versión del motor: UE1

- *UnrealEd 2.0*
 - 100% reescrito en C++
 - Versión del motor: UE1



- *UnrealEd 3.0*
 - Basado en el anterior
 - Añade nuevas características y mejoras
 - Incorpora nuevas herramientas para trabajar con terrenos y *Static Mesh*
 - Versión del motor: UE2

A lo largo de las siguientes páginas del documento se hablará en detalle de este último, por ser el que se incluye con el paquete de software Unreal Tournament 2004.

UnrealEd 3 se compone de muchas herramientas, algunas más complejas que otras. Para la creación del entorno marciano se ha utilizado fundamentalmente el editor de nivel.

2.1.3.1 Unreal Level Editor (también denominado UT2004 Editor)

Es la herramienta de edición principal en UnrealEd 3, mediante la cual se crean los mundos o niveles. Los niveles, tal y como se explicará más adelante, se pueden construir de dos modos diferentes, a través de BSP (Binary Space Partitioning) o bien mediante *Static Mesh*. Ambas técnicas pueden emplearse conjuntamente en la creación del nivel.

Interfaz

La ventana del editor de nivel se divide en cinco partes:

1. Barra de menú
2. Barra de herramientas
3. Caja de herramientas
4. Ventanas gráficas
5. Barra de estado

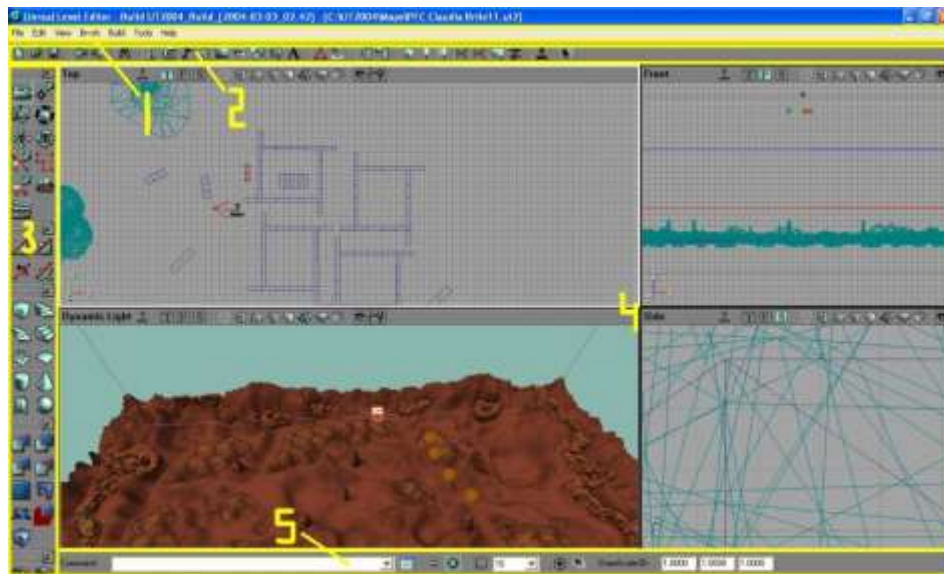


Ilustración 17 Interfaz del editor de nivel

1. Barra de menú



Ilustración 18 Barra de menú

File - Menú Archivo

Proporciona acceso a la creación de un nuevo mapa, a su apertura y almacenamiento, así como a la importación de una parte o de su totalidad.

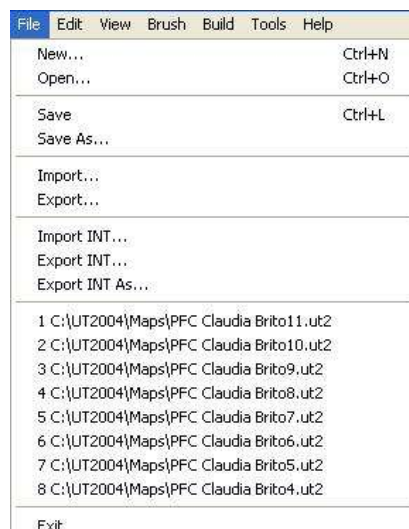


Ilustración 19 Menú archivo



Función	Descripción
New...	Crea un nuevo mapa en el que es posible construir un mundo/nivel de dos modos: aditivo o sustractivo.
Open...	Abre un mapa existente previamente almacenado en disco
Save	Guarda el mapa
Save As...	El sistema solicita que guarde el mapa con un nombre de fichero diferente
Import...	Abre un mapa con extensión .t3d. Este formato es un fichero de texto de Unreal que almacena información acerca de actores tipo mapa.
Export...	Almacena el mapa completo como un fichero t3d
Import INT	Importa modelos creados con otros programas (Maya, Blender) al UnrealEd 3
Exit	Cierra Unreal Level Editor

Tabla 1 Funciones del menú “Archivo”

Edit - Menú Edición

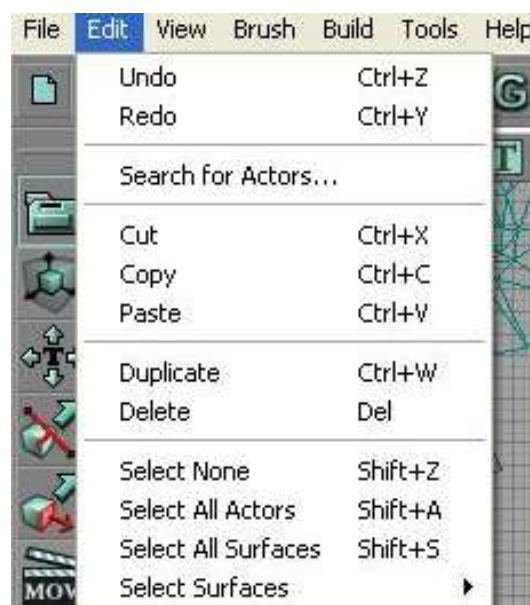


Ilustración 20 Menú edición



Función	Descripción
Undo	Deshace la última acción de la pila de transacciones
Redo	Rehace la última acción deshecha en la pila de transacciones
Search for Actors...	Abre la ventana de búsqueda de actores. Esta herramienta es extremadamente útil ya que permite definir filtros para realizar búsquedas precisas.
Cut	Coloca los elementos seleccionados en el portapapeles y los elimina del mapa
Copy	Coloca los elementos seleccionados en el portapapeles
Paste	Coloca el contenido del portapapeles en el mapa
Duplicate	Crea copias de los elementos seleccionados
Delete	Elimina los elementos seleccionados
Select None	Deselecciona todos los elementos actualmente seleccionados
Select All Actors	Selecciona todos los actores que se encuentran actualmente en el mapa
Select All Surfaces	Selecciona todas las superficies del mapa actual
Select Surfaces	Desplegable que contiene a su vez las opciones que se muestran en la ilustración número 22.

Tabla 2 Funciones del menú “Edición”

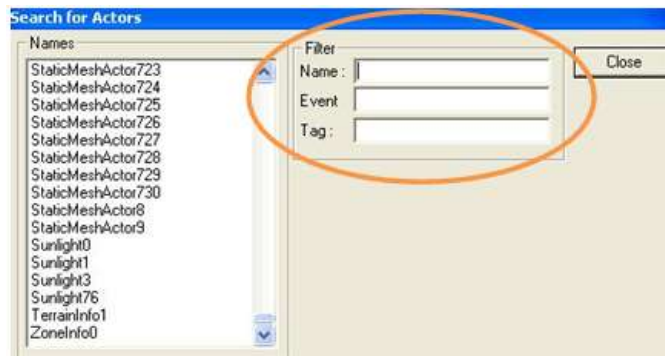


Ilustración 21 Ventana de búsqueda de actores

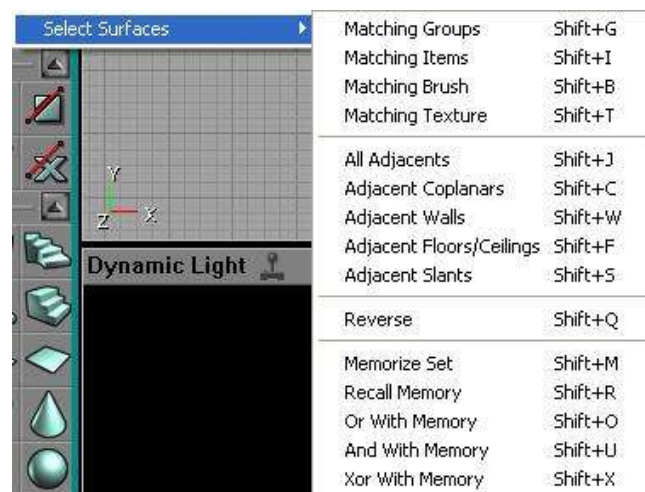


Ilustración 22 Opciones de "Select Surfaces"

Función	Descripción
Matching Groups	Esta opción no puede usarse debido a que da problemas en UnrealEd 3
Matching Items	Selecciona aquellas superficies con ítems que concuerdan con los nombres de los ítems de las superficies seleccionadas.
Matching Brush	Selecciona las superficies que pertenecen al mismo <i>Brush</i>
Matching Texture	Selecciona aquellas superficies que tienen la misma textura
All Adjacents	Selecciona todas las superficies que son adyacentes



Función	Descripción
Adjacent Coplanars	Selecciona superficies coplanarias adyacentes
Adjacent Walls	Selecciona las paredes que son adyacentes
Adjacent Floors/Ceilings	Selecciona los suelos/techos que son adyacentes
Adjacent Slants	Selecciona todas las superficies adyacentes sobre el mismo plano
Reverse	Invierte el conjunto de superficies seleccionadas
Memorize Set	Memoriza las superficies seleccionadas
Recall Memory	Recuerda la selección de superficies memorizadas
Or With Memory	Selecciona las superficies de la intersección de las superficies seleccionadas y de las que están almacenadas en memoria.
And With Memory	Selecciona la unión de las superficies seleccionadas y de las memorizadas.
Xor With Memory	Realiza el "OR exclusivo" de las superficies seleccionadas y de las memorizadas.

Tabla 3 Funciones de "Select Surfaces"



View - Menú Ver

Principalmente proporciona acceso a los nueve buscadores y a las ventanas de propiedades.

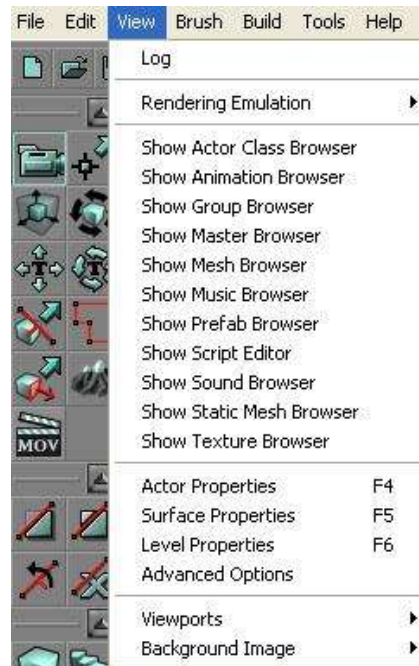


Ilustración 23 Menú ver

Función	Descripción
Log	Ventana que muestra la salida de algunos comandos
Rendering Emulation	Muestra cómo se vería el entorno según el tipo de procesamiento de los gráficos: ninguno, GeForce1, GeForce2 o Xbox.
Show Actor Class Browser	Abre la ventana del explorador en la que se muestran todas las clases de actores que se pueden añadir al mapa. Por defecto, está acoplada en el Master Browser.
Show Animation Browser	Abre la ventana del explorador en la que se muestran todas las animaciones. Algunas pueden importarse para una Skeletal Mesh. Por defecto, está acoplada en el Master Browser.



Función	Descripción
Show Group Browser	Abre la ventana del explorador en la que es posible ver, seleccionar y modificar grupos de cosas que se hayan definido. Por defecto, está acoplada en el Master Browser.
Show Master Browser	Abre la ventana del explorador maestro en la que se encuentran acoplados el resto. La imagen número 29 ilustra este hecho.
Show Mesh Browser	Abre la ventana del explorador en la que se puede pre visualizar diversos modelos animados (no incluye <i>Skeletal Mesh</i>). Por defecto, está acoplada en el Master Browser.
Show Music Browser	Abre la ventana del explorador en la que es posible pre visualizar y añadir música al nivel. Por defecto, está acoplada en el Master Browser.
Show Prefab Browser	Abre la ventana del explorador que permite crear y cargar actores prefabricados: pinceles <i>CSG</i> , texturas, luces, <i>Static Mesh</i> , etc. Por defecto, está acoplada en el Master Browser.
Show Script Editor	Permite escribir y editar el código para el juego
Show Sound Browser	Abre la ventana del explorador en la que es posible pre visualizar y añadir sonidos. Por defecto, está acoplada en el Master Browser.
Show Static Mesh Browser	Abre la ventana del explorador en la que se muestran todas las <i>Static Mesh</i> , agrupadas por paquetes, que se pueden insertar en el nivel. Por defecto, está acoplada en el Master Browser.
Show Texture Browser	Abre la ventana del explorador en la que se muestran todas las texturas, agrupadas por paquetes, que se pueden asignar a la geometría en el mundo. Por defecto, está acoplada en el Master Browser.



Función	Descripción
Actor properties	Abre la ventana flotante en la que se listan todas las propiedades del/de los actor/es seleccionado/s actualmente.
Surface properties	Abre la ventana flotante que permite habilitar/deshabilitar las propiedades, ver las estadísticas y alinear las texturas de las superficies actualmente seleccionadas.
Level properties	Abre la ventana de propiedades del nivel
Advanced Options	Abre la ventana con las opciones avanzadas del editor
Viewports	Mediante esta opción se puede establecer su configuración, definir un nuevo <i>viewport</i> o cerrarlos todos.
Background Image	Coloca una imagen detrás de los <i>viewports</i>

Tabla 4 Funciones del menú "Ver"

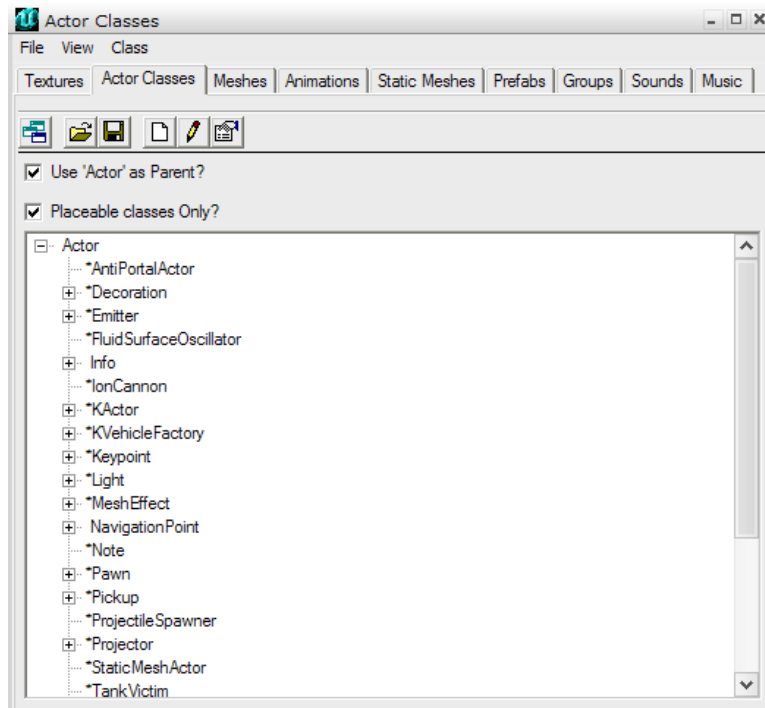


Ilustración 24 Actor Class Browser

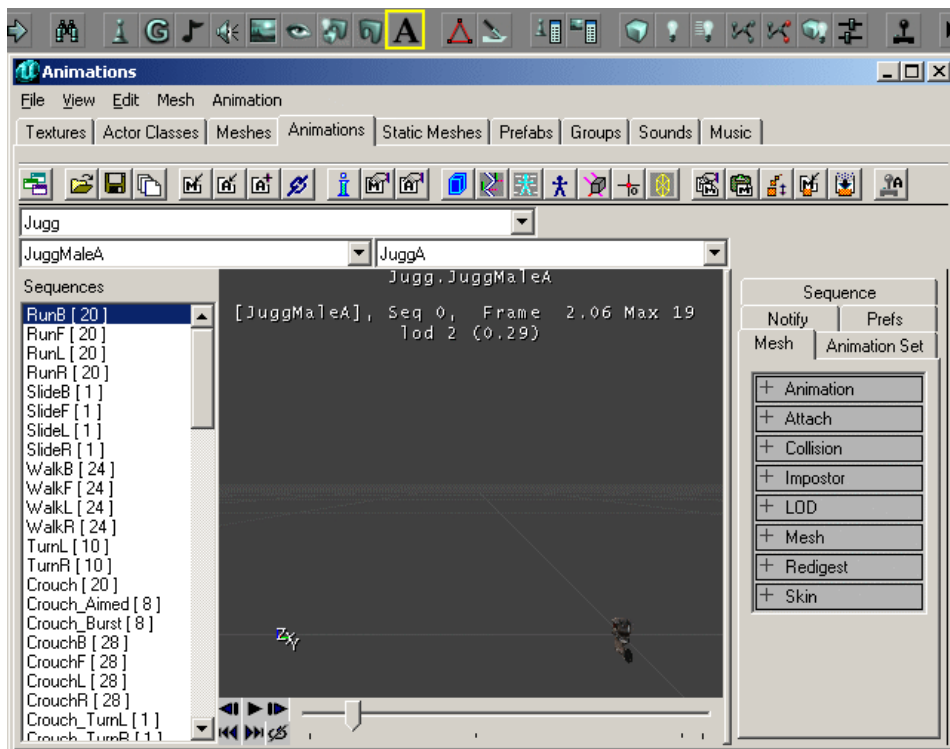


Ilustración 25 Ventana del explorador de animaciones

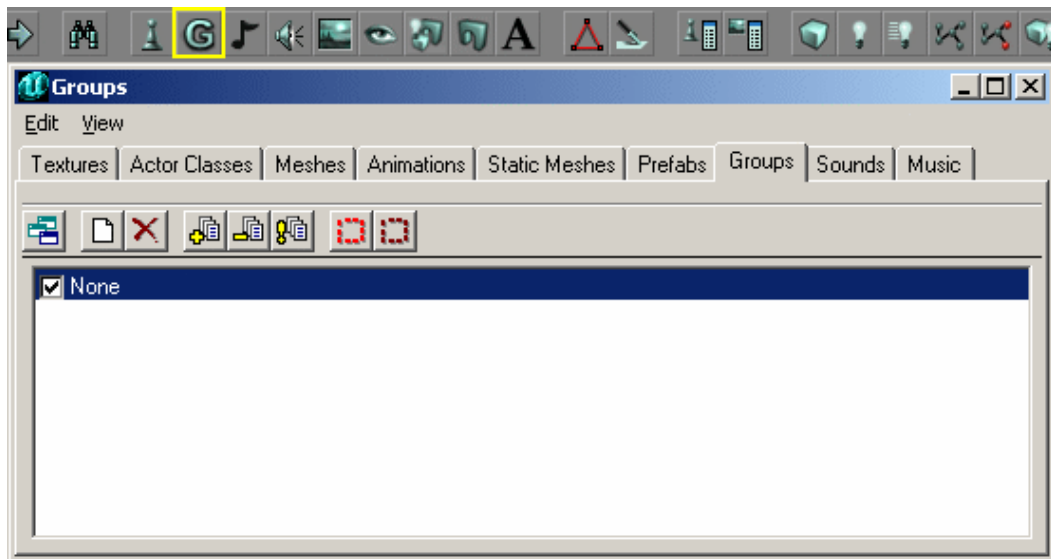


Ilustración 26 Ventana del explorador de grupos

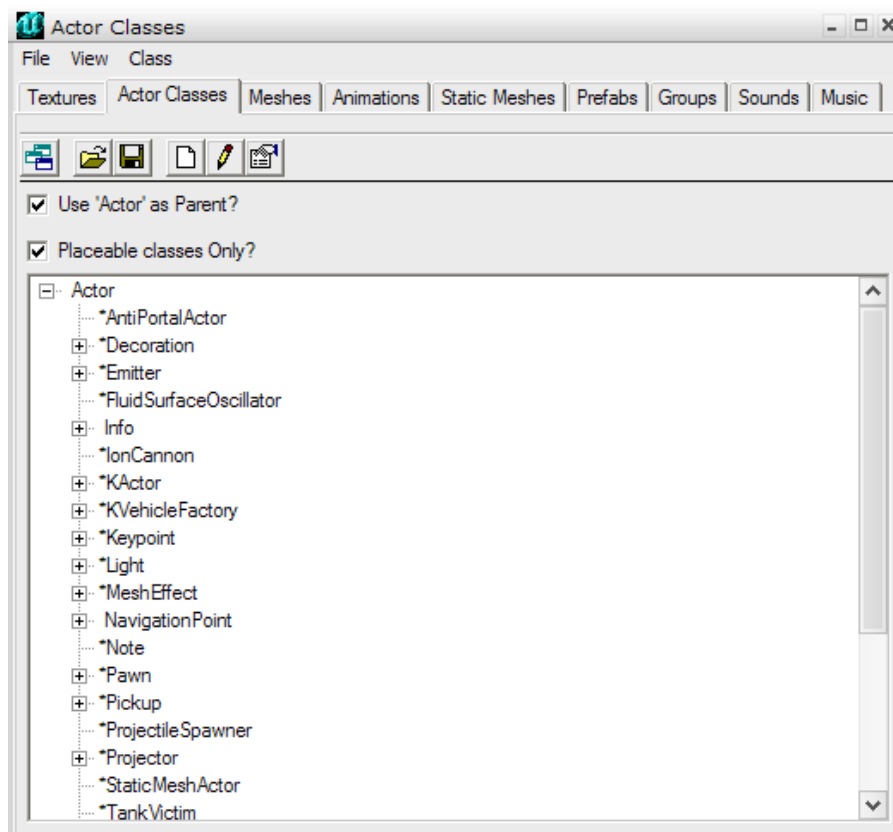


Ilustración 27 Ventana del explorador maestro

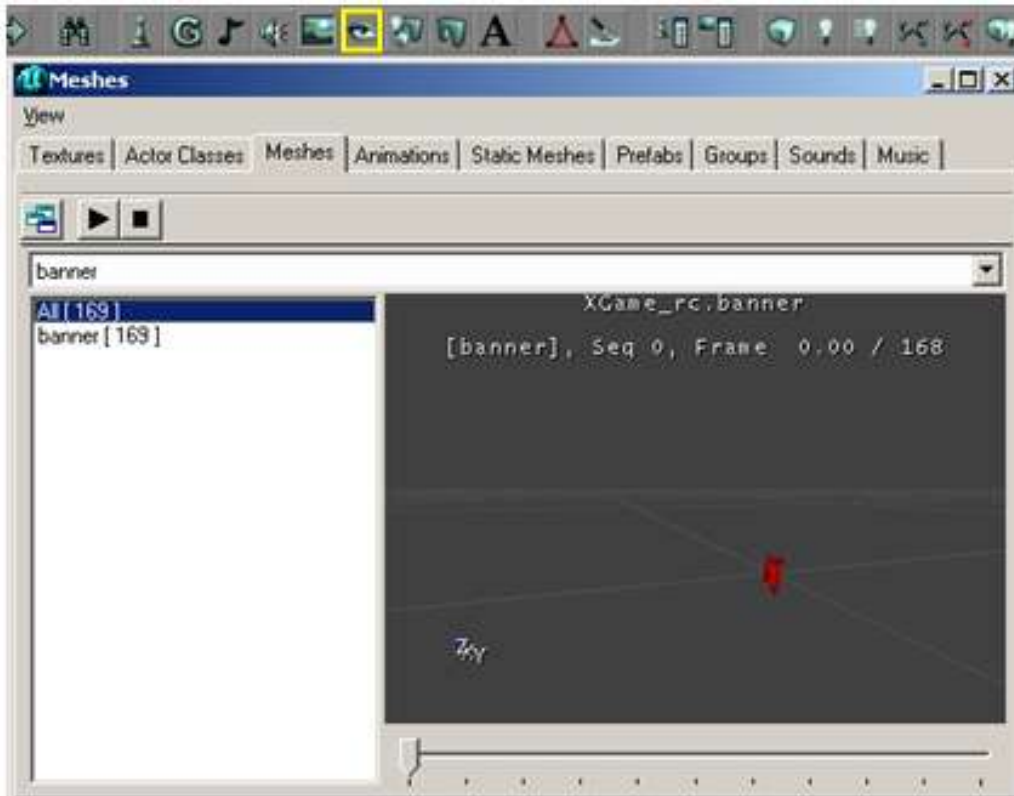


Ilustración 28 Ventana del explorador de Mesh

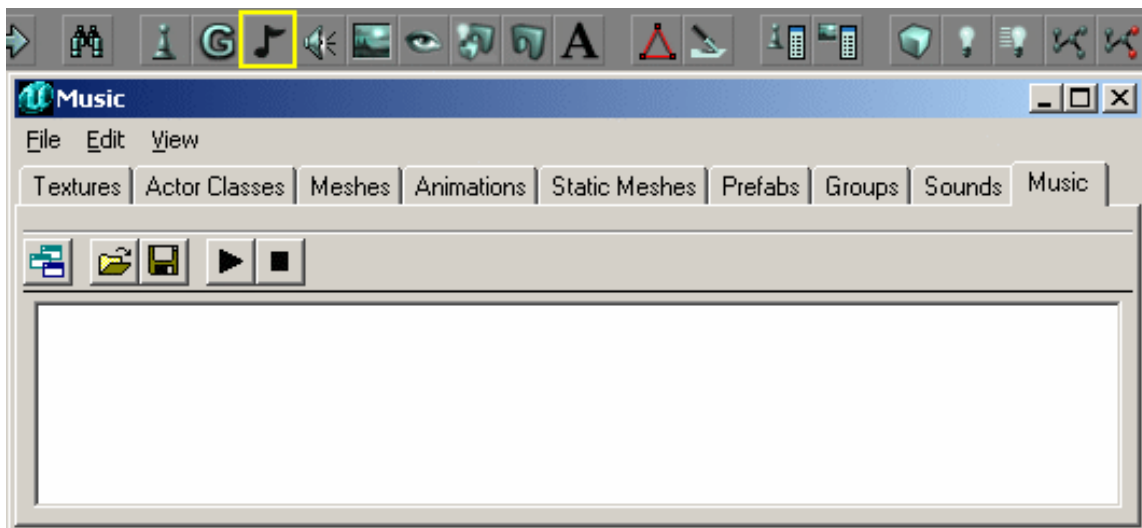


Ilustración 29 Ventana del explorador de música

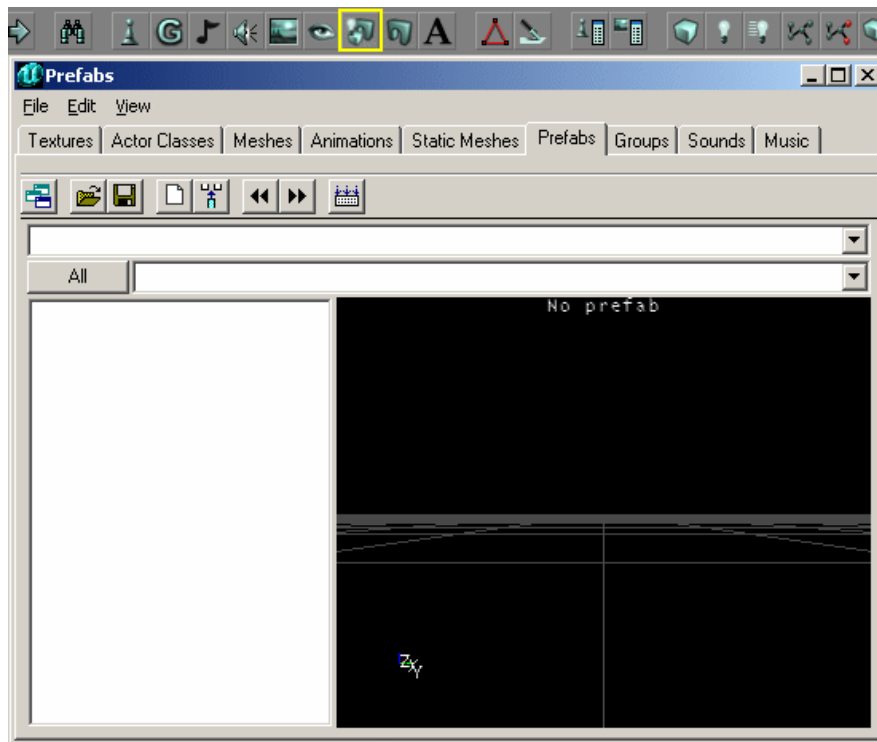


Ilustración 30 Ventana del explorador de prefabricados

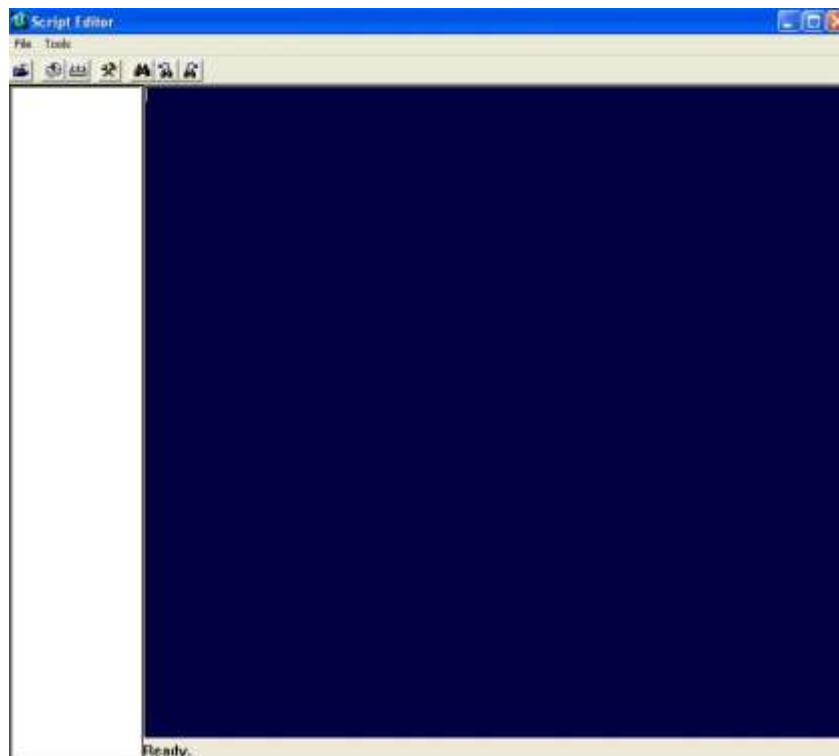


Ilustración 31 Editor de código

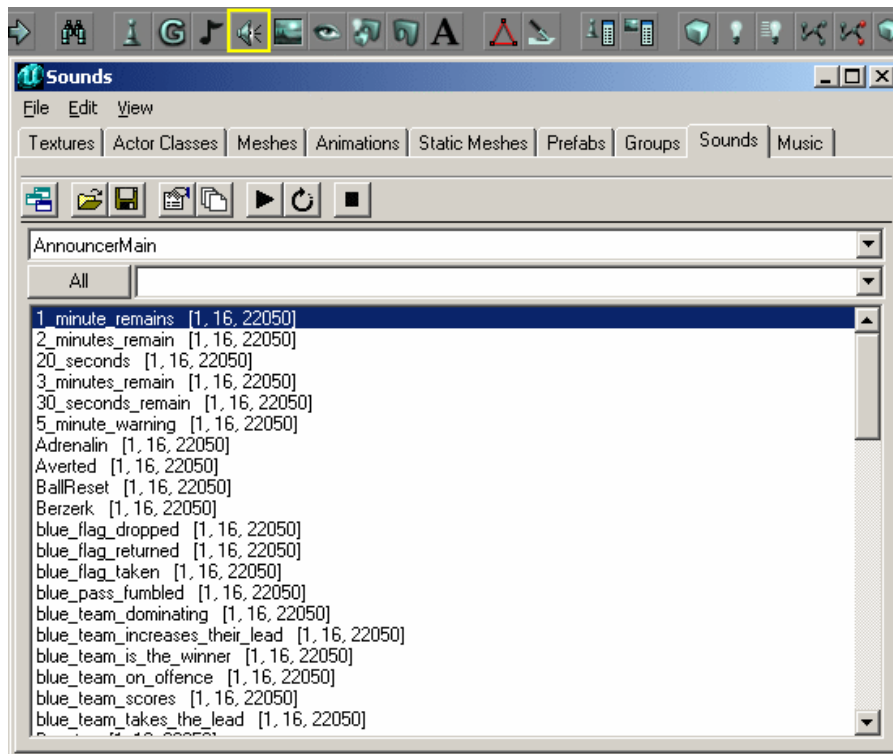


Ilustración 32 Ventana del explorador de sonidos

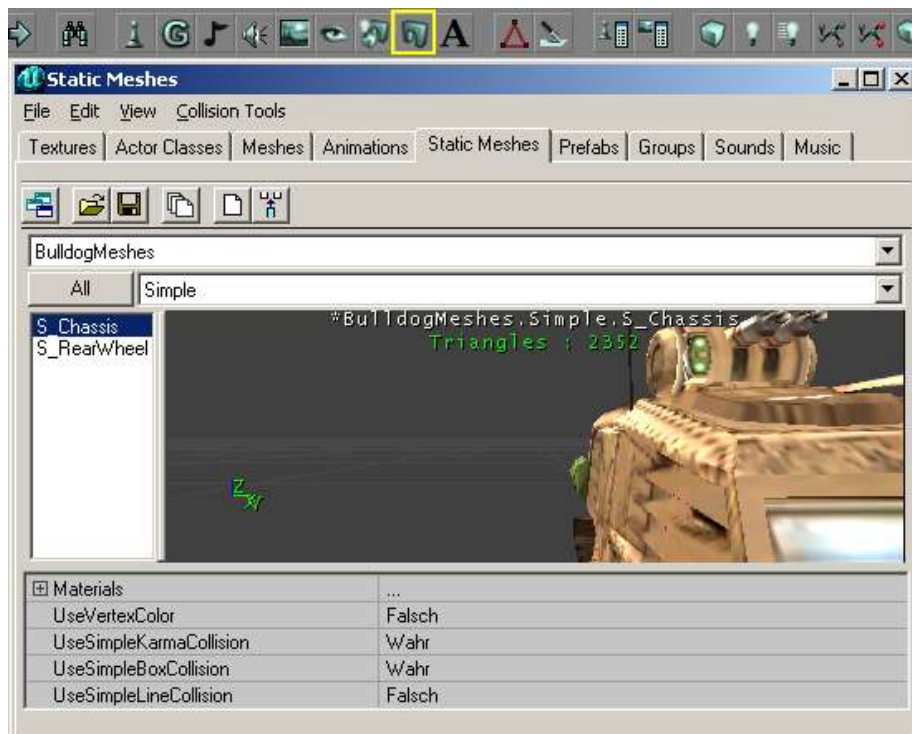


Ilustración 33 Ventana del explorador de Static Mesh

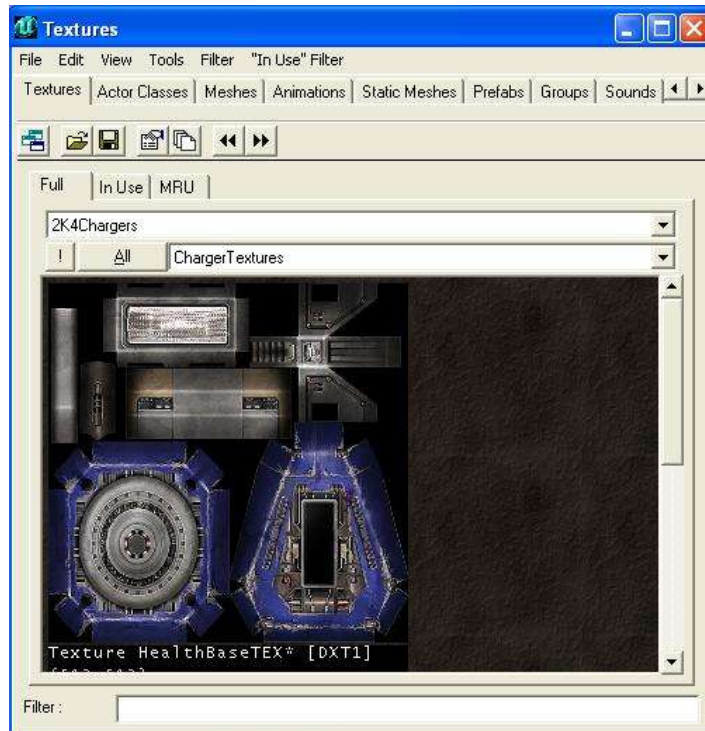


Ilustración 34 Ventana del explorador de texturas

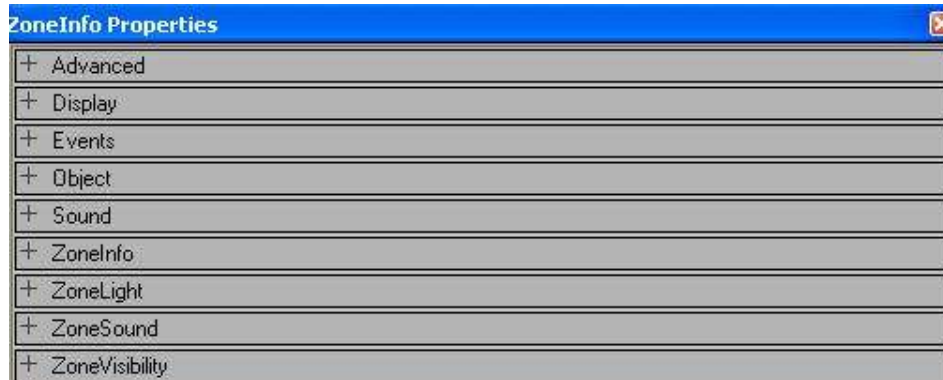


Ilustración 35 Propiedades de un actor de tipo ZoneInfo



Ilustración 36 Propiedades de una superficie



Ilustración 37 Propiedades del nivel

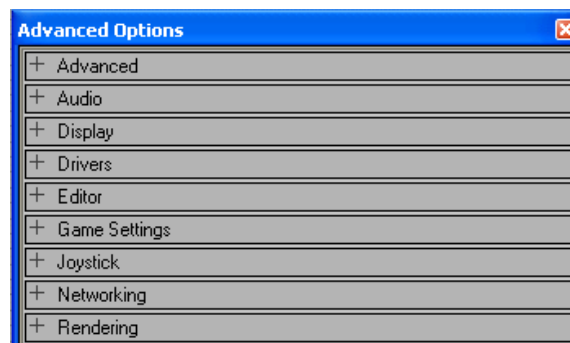


Ilustración 38 Opciones avanzadas del editor



Ilustración 39 Opciones relativas a los viewports

La configuración que se ha elegido para los viewports es la que viene por defecto.

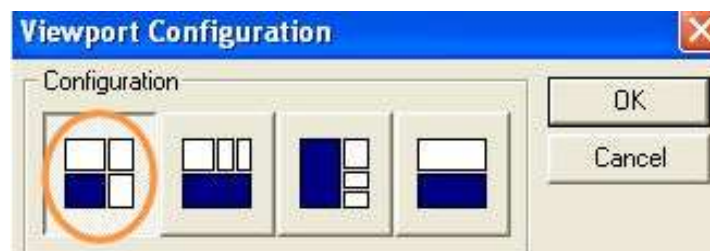


Ilustración 40 Configuración de los viewports



Brush - Menú Píncel

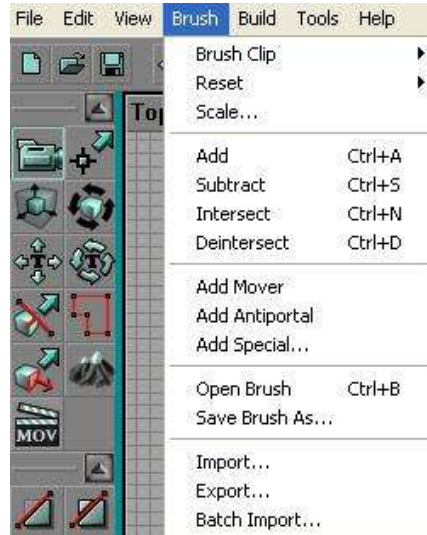


Ilustración 41 Menú píncel

Función	Descripción
Brush Clip	Explicado en los “Modos de Edición” de la caja de herramientas
Reset	Establece a su valor inicial las siguientes propiedades para el píncel actual: mover a origen, pivote, rotación, escalado y todo.
Scale...	Escala el píncel seleccionado
Add	Explicado en el grupo Brush Operations de la caja de herramientas
Subtract	Explicado en el grupo Brush Operations de la caja de herramientas
Intersect	Explicado en el grupo Brush Operations de la caja de herramientas
Deintersect	Explicado en el grupo Brush Operations de la caja de herramientas
Add Mover	Explicado en el grupo Brush Operations de la caja de herramientas



Función	Descripción
Add Antiportal	Explicado en el grupo Brush Operations de la caja de herramientas
Add Special...	Explicado en el grupo Brush Operations de la caja de herramientas
Open Brush	Permite que un pincel sea leído desde fichero
Save Brush As...	Guarda a fichero el pincel actualmente seleccionado
Import...	Importa un pincel desde un fichero
Export...	Exporta un pincel a un fichero
Batch Import...	Importa un conjunto de pinceles de una sola vez

Tabla 5 Funciones del menú "Pincel"

Build - Menú Construir

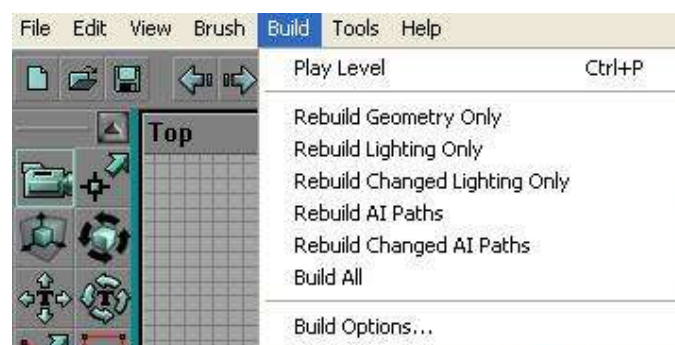


Ilustración 42 Menú construir



Función	Descripción
Play Level	Ejecuta el nivel en tiempo real como si fuera el usuario final
Rebuild Geometry Only	Reconstruye únicamente la geometría, no tiene en cuenta la iluminación ni los BSP cuts.
Rebuild Lighting Only	Reconstruye únicamente la iluminación
Rebuild Changed Lighting Only	Reconstruye únicamente la iluminación que ha sufrido cambios
Rebuild AI Paths	Reconstruye los caminos de inteligencia artificial
Rebuild Changed AI Paths	Reconstruye los caminos de inteligencia artificial que han sufrido cambios.
Build All	Reconstruye toda la geometría, la iluminación y los caminos de inteligencia artificial del nivel, según las opciones de construcción que se han establecido en "Build Options".
Build Options...	Establece las opciones de construcción del nivel para la opción anterior.

Tabla 6 Funciones del menú "Construir"

Tools - Menú Herramientas

Facilita el acceso a herramientas adicionales como el editor de formas 2D.

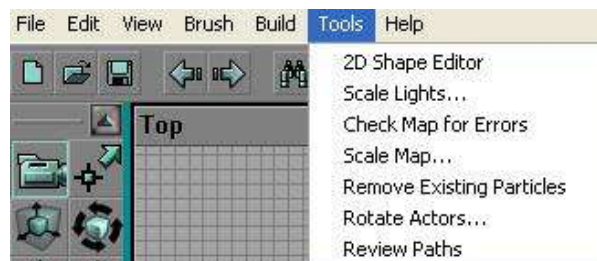


Ilustración 43 Menú herramientas



Función	Descripción
2D Shape Editor	Herramienta que permite crear formas en dos dimensiones y que puede utilizarse en lugar de los “ <i>Builder Brush</i> ” para construir pinceles.
Scale Lights...	Añade lo que se introduce en la caja como valor de LightBrightness para las luces seleccionadas, en valor o en tanto por ciento.
Check Map for Errors	Busca posibles errores en el mapa y abre una ventana mostrando los que hubiese.
Scale Map...	Herramienta que además de escalar los pinceles comprueba que las luces, disparadores (triggers), etc. estén situados en las posiciones adecuadas. Esto significa que podemos aumentar o disminuir el mapa completo sin tener mayores dificultades.
Remove Existing Particles	Permite eliminar todas las partículas que existen en la pantalla de una sola vez.
Rotate Actors...	Explicado en los modos de edición de la caja de herramientas
Review Paths	Comprueba que todos los caminos del nivel sean alcanzables















Tabla 7 Funciones del menú "Herramientas"

2. Barra de herramientas



Ilustración 44 Barra de herramientas



Función	Icono	Descripción
New Map		Explicado en la tabla 1 (Funciones del menú “Archivo”)
Open Map		Explicado en la tabla 1 (Funciones del menú “Archivo”)
Save Map		Explicado en la tabla 1 (Funciones del menú “Archivo”)
Undo		Explicado en la tabla 2 (Funciones del menú “Edición”)
Redo		Explicado en la tabla 2 (Funciones del menú “Edición”)
Search for Actors		Explicado en la tabla 2 (Funciones del menú “Edición”)
Actor Class Browser		Explicado en la tabla 4 (Funciones del menú “Ver”)
Group Browser		Explicado en la tabla 4 (Funciones del menú “Ver”)
Music Browser		Explicado en la tabla 4 (Funciones del menú “Ver”)
Sound Browser		Explicado en la tabla 4 (Funciones del menú “Ver”)
Texture Browser		Explicado en la tabla 4 (Funciones del menú “Ver”)
Mesh Browser		Explicado en la tabla 4 (Funciones del menú “Ver”)
Prefab Browser		Explicado en la tabla 4 (Funciones del menú “Ver”)
Static Mesh Browser		Explicado en la tabla 4 (Funciones del menú “Ver”)

















Animation Browser		Explicado en la tabla 4 (Funciones del menú “Ver”)
2D Shape Editor		Explicado en la tabla 7 (Funciones del menú “Herramientas”)
UnrealScript Editor		Explicado en la tabla 4 (Funciones del menú “Ver”)
Actor Properties		Explicado en la tabla 4 (Funciones del menú “Ver”)
Surface Properties		Explicado en la tabla 4 (Funciones del menú “Ver”)
Build Geometry		Explicado en la tabla 6 (Funciones del menú “Construir”)
Build Lighting		Explicado en la tabla 6 (Funciones del menú “Construir”)
Build Changed Lighting		Explicado en la tabla 6 (Funciones del menú “Construir”)
Build Paths		Explicado en la tabla 6 (Funciones del menú “Construir”)
Build Changed Paths		Explicado en la tabla 6 (Funciones del menú “Construir”)
Build All		Explicado en la tabla 6 (Funciones del menú “Construir”)
Build Options		Explicado en la tabla 6 (Funciones del menú “Construir”)
Play Map		Explicado en la tabla 6 (Funciones del menú “Construir”)
Context Sensitive Help		Ayuda contextual

Tabla 8 Funciones de la barra de herramientas



3. Caja de herramientas

A partir de UnrealEd 2 se divide en grupos que se pueden ocultar/hacer visibles. Los más importantes son:

- Modos de edición



Ilustración 45 Grupo “Modos de edición” de la caja de herramientas

- Brush Clipping



Ilustración 46 Grupo “Brush Clipping” de la caja de herramientas

- Brushbuilders



Ilustración 47 Grupo “Brushbuilders” de la caja de herramientas

- Brush Operations



Ilustración 48 Grupo “Brush Operations” de la caja de herramientas



Modos de edición

Son herramientas en el sentido clásico dado que alteran el comportamiento del puntero del ratón.

Función	Icono	Descripción
Camera Movement		Modo por defecto de la caja de herramientas. Permite mover la vista alrededor de la rejilla, y también a los actores, dependiendo de los botones del ratón y de las combinaciones de teclas que se usen.
Vertex Editing		Permite modificar la forma de un pincel mediante el desplazamiento de sus vértices.
Actor (Brush) Scaling		Modo de edición para escalar horizontal o verticalmente un pincel previamente seleccionado.
Actor Rotate		Modo de edición de la caja de herramientas que permite rotar pinceles o actores de tipo meshes.
Texture Pan		Mueve una textura a la izquierda, a la derecha, arriba o abajo.
Texture Rotate		Este modo permite girar una textura fácilmente cualquier ángulo.
Brush Clipping		Herramienta para cambiar la forma de los pinceles. Ofrece dos posibilidades: Clip y Split, que se verán en el apartado correspondiente a este grupo.
Freehand Polygon Drawing		Modo alternativo al editor de formas 2D. Permite crear formas en dos dimensiones irregulares con el fin de extraerlas en objetos que pueden ser sustraídos o añadidos






Función	Icono	Descripción
		posteriormente.
Face Drag		Mediante este modo es posible alargar un pincel arrastrando sus bordes. Esta herramienta no funciona en UnrealEd 3.
Terrain Editing		Aparece con la versión 3 del editor y sirve para manipular el terreno.
Matinee		Es una herramienta usada en Unreal Tournament que permite programar escenas de cámara complejas: level fly by, secuencias programadas, cinematográficas, etc.

Tabla 9 Funciones de la Caja de Herramientas



La herramienta Terrain Editing es una de las más complejas, por ello se ha dividido su interfaz en seis partes:

- 1 Herramientas
- 2 Opciones
- 3 Soft Selection
- 4 Terrenos
- 5 Capas
- 6 Decoradores

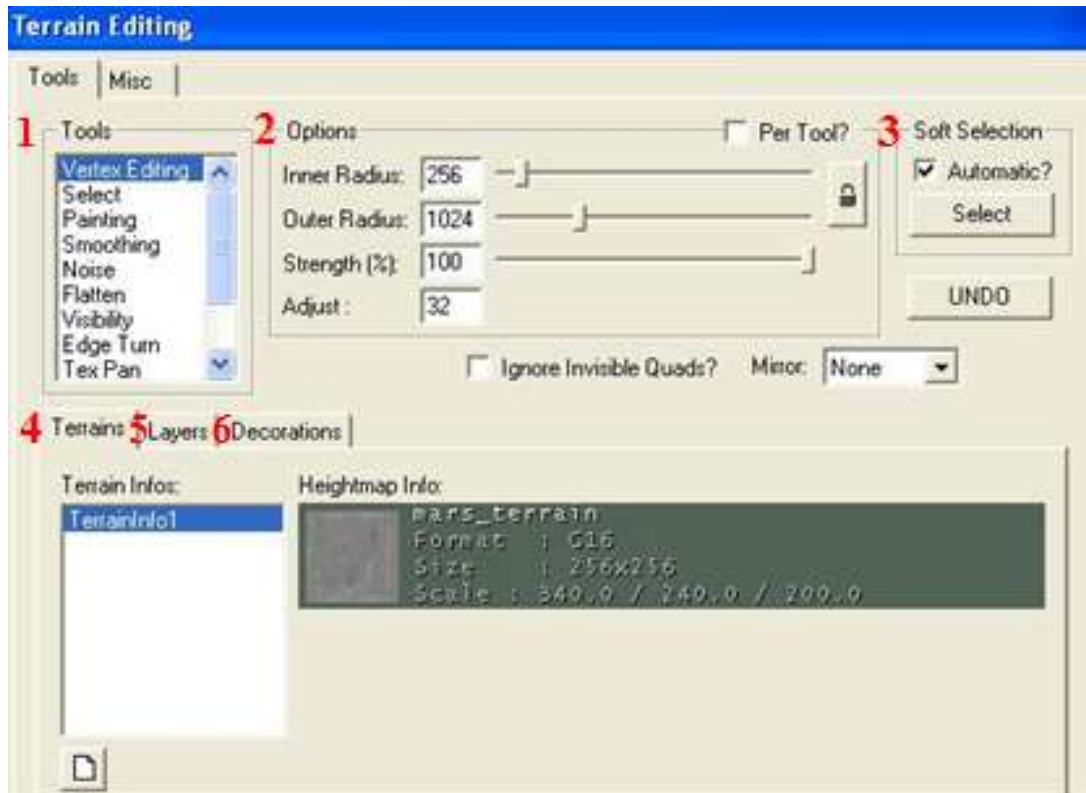


Ilustración 49 Herramienta para la edición del terreno

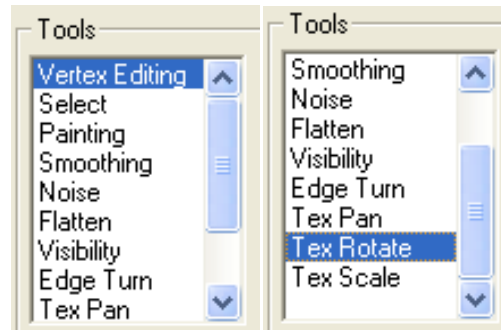
*Herramientas para la edición del terreno*

Ilustración 50 Herramientas para la manipulación del terreno

En total son once pero se hará mayor hincapié en aquellas que se han usado para la construcción del terreno del mapa.

Función	Descripción
<i>Vertex Editing</i>	Permite seleccionar los vértices del terreno que se quieren desplazar hacia arriba o hacia abajo.
<i>Select</i>	Permite seleccionar un área del terreno que será generada de manera aleatoria.
<i>Painting</i>	Sirve para levantar y bajar el terreno mediante pinceladas tipo “brush” en la vista 3D. Se ha utilizado principalmente en la construcción de las montañas y de los valles. (Ver apartado 2.2.1.4 Creación del terreno marciano, para ver cómo se ha configurado).
<i>Smoothing</i>	Se deshace de los bordes recortados y de los picos en el terreno. Tan sólo se ve afectada por el radio exterior.
<i>Noise</i>	Esta herramienta funciona de forma similar a la de generación del terreno. Realiza cambios aleatorios en el terreno (sólo afecta al radio exterior).



Función	Descripción
<i>Flatten</i>	Toma como referencia la altura del vértice que está en el centro de la herramienta de selección y hace que todos los demás vértices en su radio tengan la misma altura.
Visibility	Esta herramienta se utiliza para hacer invisibles determinadas partes del terreno.
Edge Turn	Sirve para girar el borde diagonal en los espacios de la rejilla dentro del radio exterior.
Tex Pan	Permite mover la textura aplicada a la capa del terreno seleccionada.
Tex Rotate	Permite girar la textura aplicada a la capa del terreno seleccionada.
Tex Scale	Permite escalar la textura aplicada a la capa del terreno seleccionada.

Tabla 10 Funciones para la manipulación del terreno

Opciones

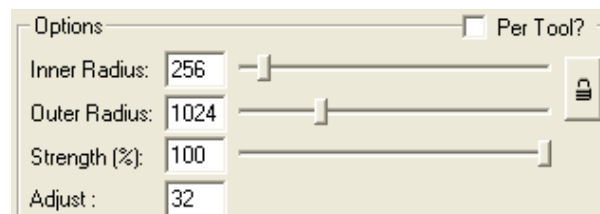


Ilustración 51 Opciones de la herramienta Terrain Editing

- Inner Radius: Radio interior
- Outer Radius: Radio exterior
- Strength (%): Fuerza aplicada en tanto por ciento



- Per Tool?: Habilitando esta opción se puede tener una configuración distinta para cada herramienta. Si no se marca se utiliza la misma configuración para todas.

Los vértices seleccionados en el radio interior se ven afectados por la totalidad de la opción “Strength”, mientras que aquellos que están situados entre el radio interior y el exterior se ven afectados cada vez menos conforme se va avanzando hacia el radio exterior.

Soft Selection

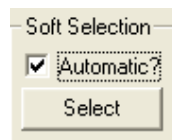


Ilustración 52 Soft Selection (Terrain Editing)

Teniendo habilitada la opción Automatic se puede seleccionar más de un vértice de una sola vez.

Terrenos

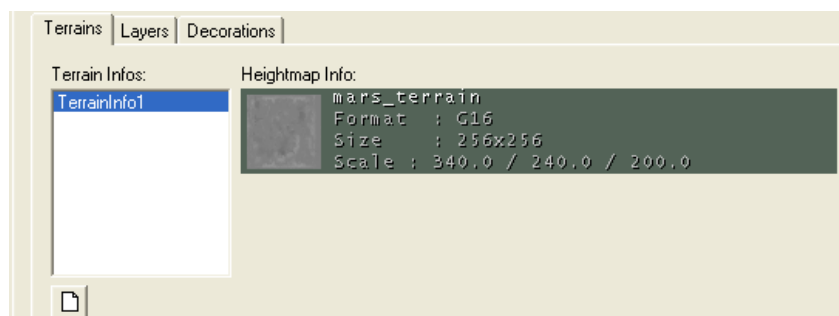



Ilustración 53 Terrenos (Terrain Editing)

Muestra una lista con todos los actores de tipo “TerrainInfo” que hay en el mapa. A la derecha se muestra el “Heightmap” para el “TerrainInfo” actualmente seleccionado. En la esquina izquierda inferior se encuentra el botón que permite añadir nuevos “TerrainInfo” .



Capas

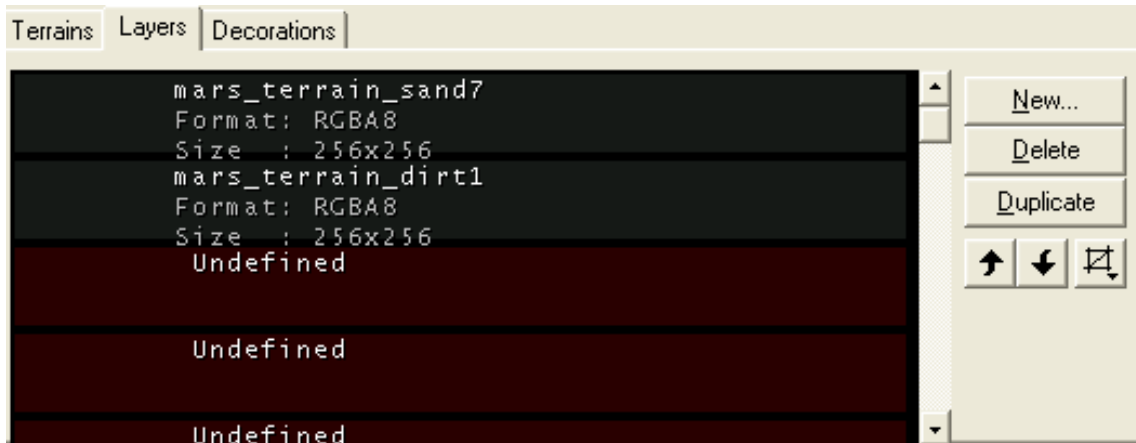


Ilustración 54 Capas (Terrain Editing)

Muestra la lista de capas para el terreno actualmente seleccionado en la pestaña “Terrains”. Las capas contienen información acerca de las texturas que están siendo usadas en el terreno. La lista es jerárquica, mostrándose en primer lugar la capa base.

Decoradores

Sirven para añadir características al terreno. Un ejemplo de decorador sería la hierba. Dado que el terreno marciano no presenta en su superficie vegetación alguna, no se ha considerado oportuno añadir alguno.



Ilustración 55 Decoradores (Terrain Editing)



Generador de terreno

- Steps: Controla el número de pasos que el generador realiza sobre el terreno. Más pasos significan más montañas.
- Strength: Controla la magnitud de las montañas que se están generando. Cuanto más alto sea este valor mayor cantidad de cimas escarpadas habrá que no dejan espacio para caminar.
- Use Entire Heightmap?: Habilitar esta opción si se quiere que la superficie generada ocupe todo el espacio del terreno.

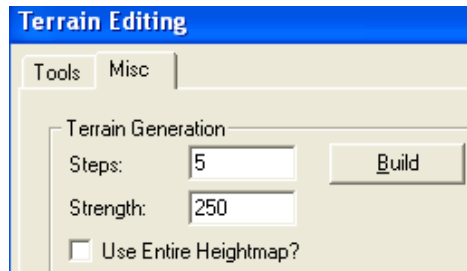


Ilustración 56 Generador de terreno

Brush Clipping



Ilustración 57 Grupo Brush Clipping

Función	Descripción
Clip Selected Brushes	Descarta la parte de un pincel que se inclina de un lado de un plano.
Split Selected Brushes	Divide un pincel en dos nuevos pinceles a lo largo de un plano.



Función	Descripción
Flip Clipping Normal	Voltea la línea perpendicular a la “línea de clipping” que indica qué lado del pincel será cortado.
Delete Clipping Markers	Elimina los marcadores que indican dónde comienza (1) y dónde termina (2) la “línea de clipping”.

Tabla 11 Funciones del grupo “Brush Clipping” de la caja de herramientas

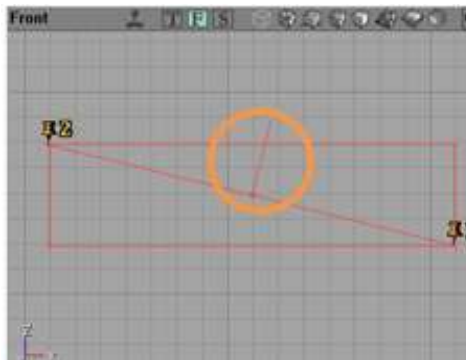


Ilustración 58 Marcadores de comienzo y fin de la “línea de clipping”

Brushbuilders

Los constructores de pinceles crean nuevas formas a partir del constructor de pincel de alambre rojo.



Ilustración 59 Grupo Brushbuilders

1 **Cube** (Cubo)





2 **Curved Staircase** (Escalera Curva)



- 3 **Spiral Staircase** (Escalera en Espiral)
- 4 **Linear Staircase** (Escalera Lineal)
- 5 **BSP Based Terrain** (BSP Basado en el Terreno)
- 6 **Sheet** (Lámina, plancha)
- 7 **Cylinder** (Cilindro)
- 8 **Cone** (Cono)
- 9 **Volumetric** (Volumétrico)
- 10 **Sphere** (Esfera)

Brush Operations

Estos comandos permiten modificar la arquitectura del mapa:

Función	Icono	Descripción
Add		Crea un nuevo pincel aditivo (additive brush) de color azul a partir del actual constructor de pincel de color rojo.
Subtract		Crea un nuevo pincel sustractivo (subtractive brush) de color amarillo a partir del actual constructor de pincel de color rojo.
Intersect		La operación de intersección corta la porción del constructor de pincel rojo que está en el espacio sustractivo. Esta operación debe realizarse con anterioridad a la de “Subtract”.
Deintersect		Elimina la porción del constructor de pincel rojo que está en el espacio aditivo. Esta operación debe realizarse con anterioridad a la de “Add”.




Función	Icono	Descripción
Add Special Brush		Ventana emergente en la interfaz del Unreal Level Editor que permite al constructor del mapa añadir pinceles especiales.
Add Static Mesh		Ventana emergente para definir una nueva <i>Static Mesh</i> .
Add Mover Brush		Permite añadir un actor de tipo “ <i>Static Mesh</i> ” que puede moverse entre posiciones clave predefinidas y rotaciones. Un ejemplo sería la apertura y el cierre de una puerta.
Add Antiportal		Sirve para añadir un plano de oclusión que evita el procesamiento de las cosas que están ubicadas detrás de él.
Volume		Permite añadir un volumen, una región del espacio para la que se ha definido uno de los comportamientos que se muestran en la ilustración número 63.

Tabla 12 Funciones del modo Brush Operations

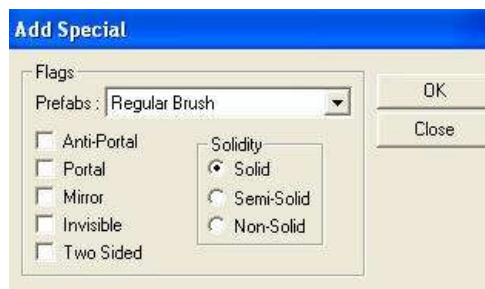


Ilustración 60 Ventana emergente para añadir pinceles especiales

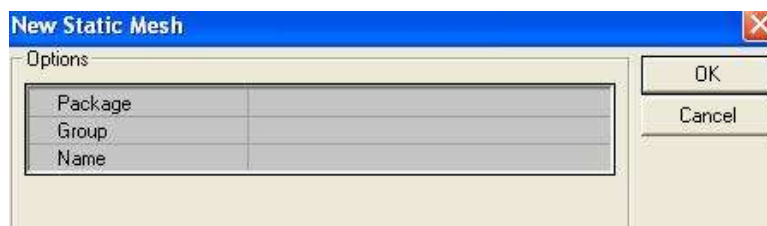


Ilustración 61 Ventana emergente para definir una nueva "StaticMesh"



BlockingVolume
Volume
LadderVolume
PhysicsVolume
DefaultPhysicsVolume
PressureVolume
WaterVolume
SnipingVolume
ConvoyPhysicsVolume
LavaVolume
xFallingVolume
IonCannonKillVolume
LimitationVolume
HitScanBlockingVolume
ASCriticalObjectiveVolume
LeavingBattleFieldVolume

Ilustración 62 Tipos de volúmenes

4 Ventanas gráficas

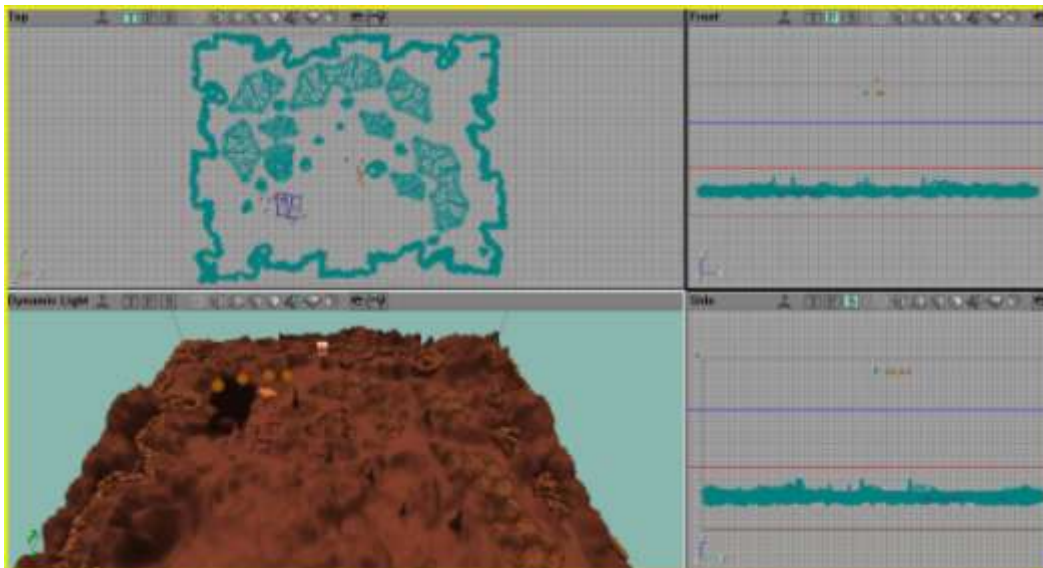


Ilustración 63 UnrealEd Viewport

Constituyen la parte principal del interfaz del Unreal Level Editor ya que muestran diferentes vistas del mapa que actualmente está cargado.

En la configuración por defecto del editor el espacio de trabajo aparece dividido en cuatro viewports o ventanas gráficas. Tres de ellas muestran las vistas ortogonales o en 2D del mapa:



- Top (Arriba)

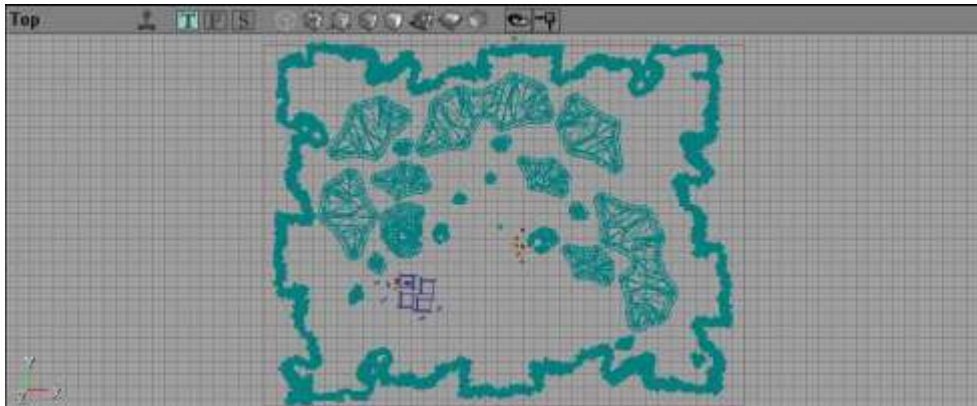


Ilustración 64 Top Viewport

- Front (Frente)

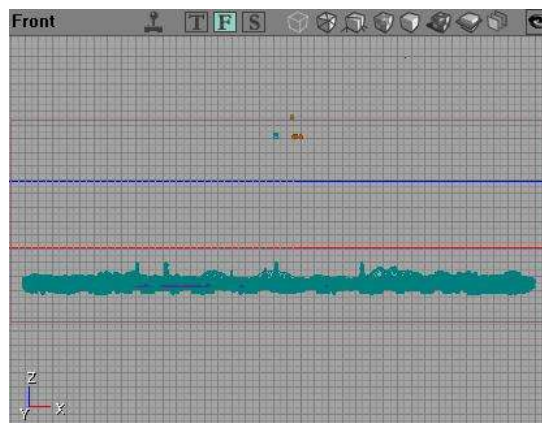


Ilustración 65 Front Viewport

- Side (Lado)

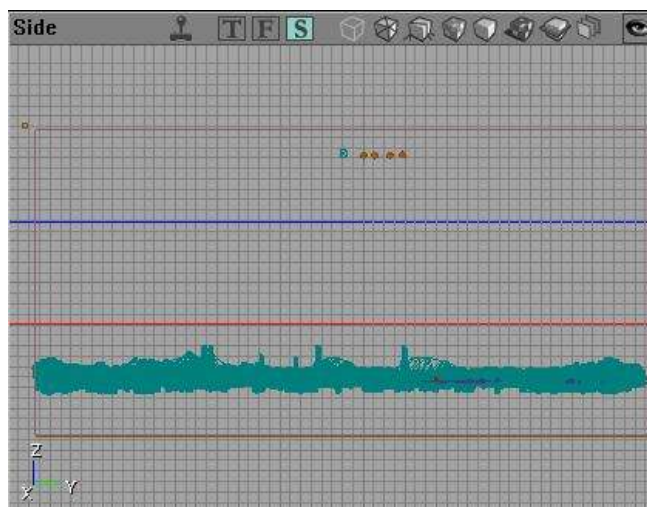


Ilustración 66 Side Viewport

Una cuarta vista muestra el mapa tal y cómo lo vería el rover desde dentro.

- 3D (Tres dimensiones)

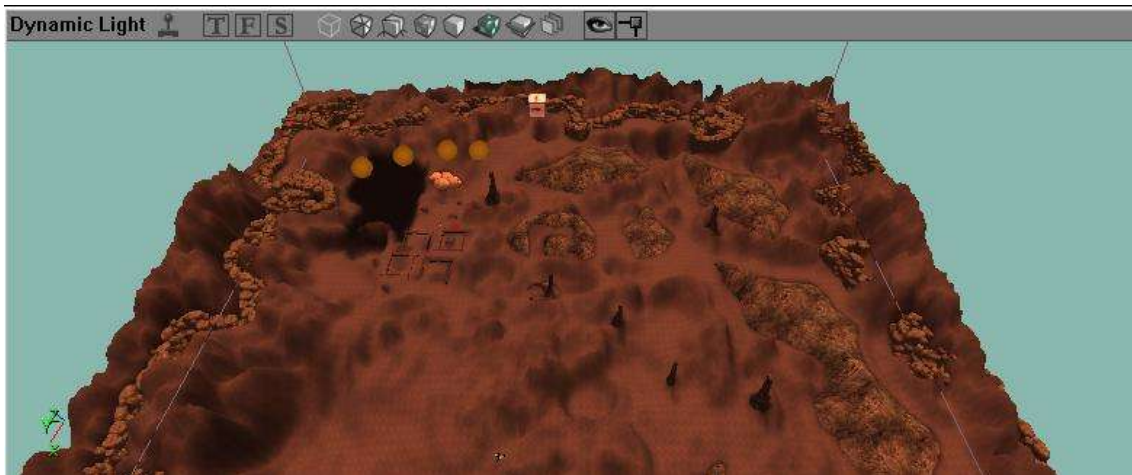


Ilustración 67 3D Viewport

5 Barra de estado

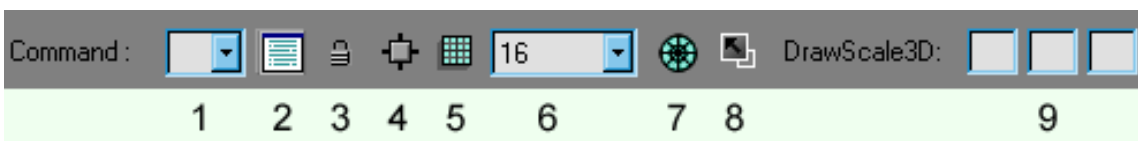


Ilustración 68 Barra de estado

Función	Nº	Descripción
Consola	1	La mayoría de comandos son equivalentes a acciones que se realizan en el resto del interfaz pero a algunas de las características del editor sólo puede accederse desde aquí.
Ventana de log	2	Explicado en la tabla 4 (Funciones del menú “Ver”)
Bloquea las selecciones	3	
Snap to vertex toggle	4	Ajuste para conmutador de vértice



Función	Nº	Descripción
Snap to grid toggle	5	Ajuste para conmutador de rejilla
Tamaño de la rejilla	6	Conviene fijar este valor a 16.
Snap to Rotation Grid toggle	7	Ajuste para conmutador de rejilla de rotación
	8	Maximiza el viewport activo
DrawScale3D	9	Muestra el valor de la propiedad Display > DrawScale3D para el/los actor/es seleccionado/s actualmente.

Tabla 13 Funciones de la barra de estado



2.1.4 *USARSim*

Simulador de código abierto³⁹ de alta fidelidad, inicialmente destinado a la recreación de entornos y robots de búsqueda y rescate urbano (USAR)⁴⁰. Su objetivo es servir de herramienta de investigación para el estudio de la interacción humano-robot (HRI)⁴¹ y la coordinación multirobot. Fue creado originalmente por Jijun Wang, estudiante de doctorado de la universidad de Pittsburgh. Posteriormente, el NIST⁴² lo desarrolló en forma de una guía de simulación (Reference Test Facility for Autonomous Mobile Robots)⁴³ e incluyó además un entorno normalizado de desastres. Este consiste en tres escenarios (Yellow, Orange y Red) de dificultad progresiva. La función principal de USARSim es la simulación del entorno interactivo, de los robots y sus sensores.

La interacción humano-robot se lleva a cabo mediante:

- El procesamiento preciso de los elementos del interfaz de usuario, en concreto de la cámara de vídeo.
- La representación exacta de la automatización y del comportamiento del robot
- La representación del entorno remoto

Desde su lanzamiento inicial, se ha ido ampliando para soportar diversos entornos así como robots de carretera, el desafío urbano DARPA⁴⁴, robots futbolistas, submarinos, humanoides y helicópteros.

³⁹ Para más información consultar http://es.wikipedia.org/wiki/C%C3%B3digo_abierto

⁴⁰ Para más información consultar http://en.wikipedia.org/wiki/Urban_search_and_rescue

⁴¹ Para más información consultar http://en.wikipedia.org/wiki/Human_robot_interaction

⁴² Página web oficial: <http://www.nist.gov/>

⁴³ Página web oficial: <http://www.isd.mel.nist.gov/projects/USAR/>

⁴⁴ Página web oficial: <http://www.darpa.mil/>



La versión de USARSim que se ha usado en este proyecto (usarsim-2004 Final) incluye varios modelos del entorno, modelos de robots experimentales y comerciales y modelos de los sensores.

2.1.4.1 *Características*

Las particularidades del simulador son:

- Arquitectura Cliente-Servidor que permite la conexión y el control de los robots desde equipos remotos.
- El simulador se basa, como ya se comentó en el punto 2.1, en el motor del juego Unreal Tournament 2004.
- Soporte íntegro para la simulación multirobot
- Permite simular cualquier tipo de robot
- El robot puede tener motores y sensores
- Los movimientos del robot y las interacciones están bajo el control del motor físico integrado en Unreal Engine (Karma).
- Es posible crear entornos complejos sin demasiado esfuerzo
- Procesamiento 3D, gracias a UT2004
- Fácil de usar, ya que su control se basa en cadenas de texto comprensibles por los humanos. Cada cadena que se envíe a USARSim debe terminar en “\r\n”.

2.1.4.2 *Arquitectura*

Se divide en dos partes: cliente y servidor. Del lado cliente se encuentran:

- Clientes Unreal encargados del procesamiento del entorno simulado
- Aplicaciones a nivel de usuario que realizan el control de los robots en el simulador.

Todos los clientes intercambian datos con el servidor a través de la red.



El servidor se encarga de:

- Almacenar el estado de todos los objetos en el simulador
- Responder a las peticiones de los clientes modificando el estado de dichos objetos.
- Enviar los datos de vuelta a los clientes y a los controladores del lado usuario

Está constituido por:

- Gamebots⁴⁵: protocolo que hace posible la comunicación entre el servidor y el cliente.
- Motor Unreal: se encarga de la simulación del entorno aunque no lleva a cabo su procesamiento.
- Modelos: incluye modelos de los robots, de los sensores, de las víctimas, etc.
- Mapa

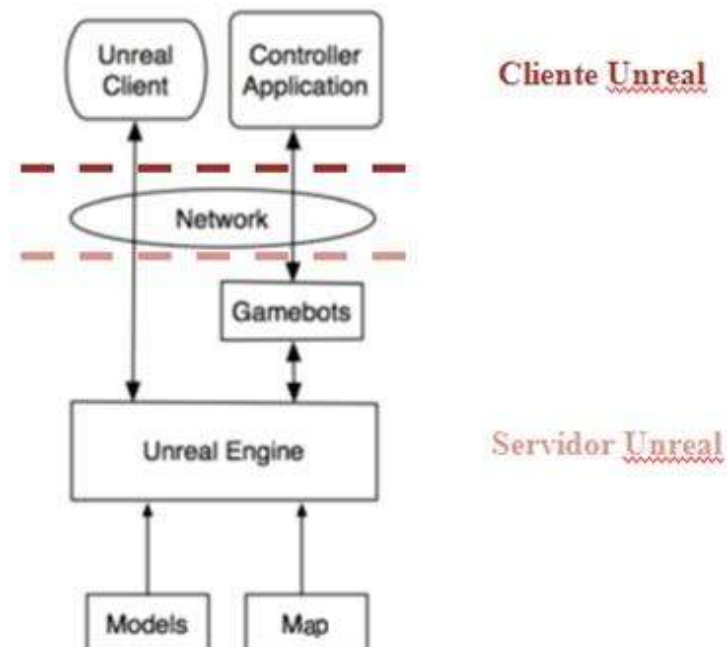


Ilustración 69 Arquitectura de USARSim

⁴⁵ <http://www.planetunreal.com/gamebots/>



2.1.4.3 Componentes

Como ya se comentó al principio del apartado 2.1.4 USARSim se encarga fundamentalmente de la simulación del entorno interactivo, de los robots y sus sensores.

Para ello consta de los siguientes componentes:

- Simulación del entorno
 - Entorno USAR
 - Modelos geométricos
 - Simulación de obstáculos
 - Simulación de la iluminación
 - Simulación de efectos especiales
 - Simulación de víctimas
- Simulación de sensores
 - Sensores propioceptivos
 - Estado de la batería y del foco
 - Sensores de estimación de posición
 - Sensores de localización, giro y velocidad
 - Sensores de percepción
 - Sonar, laser...
- Simulación de robots
 - Motor físico para cuerpos rígidos de Karma incluido en UT2004
 - Robot mecánico simulado a partir del modelo del robot
 - Incluye 8 robots de tierra: P2AT, P2DX, ATRV-Jr, Zerg, Tarantula, Talon, TeleMax y Soryu.
- Simulación de comunicaciones
 - Servidor de comunicaciones inalámbricas
 - Mensajes entre robots



2.1.4.4 *Gamebots*

El protocolo de comunicación usado por el motor Unreal es propietario. Esto dificulta enormemente el acceso a Unreal Tournament desde otras aplicaciones. Para solventar este problema un equipo de investigadores desarrolló Gamebots⁴⁶. Básicamente funciona del siguiente modo: abre un socket TCP/IP en el motor Unreal e intercambia datos con el exterior. USARSim lo habilita para comunicarse con el controlador.

Todos los datos (mensajes y comandos) de Gamebots siguen el formato:

tipo_dato {segmento1} {segmento2}...

tipo_dato: especifica el tipo del dato. Son caracteres en mayúsculas como INIT, DRIVE, etc.

segmento: lista de pares nombre/valor. El nombre y el valor van separados por un espacio. Por ejemplo, "Location x,y,z".

Un mensaje o un comando se componen de un tipo de dato y de múltiples segmentos. Todos los comandos se terminan con "\r\n".

Mensajes

Hay cinco tipos de mensajes:

- De estado: reportan información acerca del estado del robot
- De sensor: contienen los datos de los sensores
- De geometría: información relativa a la geometría del robot o sensor
- De configuración: información relativa a la configuración del robot o sensor
- De respuesta: estado de la respuesta a ciertos comandos

⁴⁶ <http://www.planetunreal.com/gamebots/>



Nombre	Parámetros	Función
STA	<i>Type</i> : describe el tipo del vehículo	Estado del robot
	<i>Time</i> : tiempo UT en segundos	
	<i>FrontSteer</i> : ángulo de las ruedas delanteras en radianes.	
	<i>RearSteer</i> : ángulo de las ruedas traseras en radianes	
	<i>SternPlaneAngle</i> : ángulo del plano de popa (vehículos de tipo náutico) en radianes.	
	<i>RudderAngle</i> : ángulo del plano de timón (vehículos de tipo náutico) en radianes.	
	<i>LightToggle</i> : habilita/deshabilita la luz	
	<i>LightIntensity</i> : intensidad del foco de luz	
SEN	<i>Battery</i> : duración de la batería en segundos	Información de sensores
	<i>Type</i> : tipo del sensor	
	<i>Name</i> : nombre del sensor	
	<i>Tick</i> : contador de tick	
	<i>Location</i> : posición estimada en metros	
	<i>Orientation</i> : orientación estimada en radianes	
	<i>Latitude</i> : grado, minuto y descripción cardinal	
	<i>Longitude</i> : grado, minuto y descripción cardinal	
	<i>Fix</i> : indica si se ha obtenido o no una posición	
	<i>Satellites</i> : número de satélites rastreados	
<i>Touch</i> : indica si el robot está tocando algo		
<i>Resolution</i> : resolución del sensor en radianes		
<i>FOV</i> : campo de visión del sensor en radianes		
<i>Range</i> : rango de valores		



Nombre	Parámetros	Función
GEO	<i>Type</i> : tipo del robot	Geometría del robot
	<i>Name</i> : nombre del robot	
	<i>Dimensions</i> : largo, ancho y alto del robot en metros	
	<i>COG</i> : posición del centro de gravedad en metros	
	<i>WheelRadius</i> : radio de las ruedas del robot en metros	
	<i>WheelSeparation</i> : separación de las ruedas en metros	
	<i>WheelBase</i> : base de la rueda en metros	
CONF	<i>Type</i> : tipo del robot	Configuración del robot
	<i>Name</i> : nombre del robot	
	<i>SteeringType</i> : puede ser “Skid”, “Ackerman” o “Aerial”	
	<i>Mass</i> : masa del robot en kilogramos	
	<i>MaxSpeed</i> : velocidad máxima de giro de las ruedas del robot en radianes/s.	
	<i>MaxTorque</i> : par máximo de torsión del robot en UU	
	<i>MaxFrontSteer</i> : ángulo máximo de giro de las ruedas delanteras del robot en radianes.	
<i>MaxRearSteer</i> : ángulo máximo de giro de las ruedas traseras del robot en radianes.		
RES	<i>Time</i> : “timestamp” en el mundo virtual cuando se envió el mensaje en segundos.	Respuesta genérica
	<i>Type</i> : tipo del sensor	
	<i>Name</i> : nombre del sensor	
	<i>Status</i> : si el comando se ejecutó satisfactoriamente o no.	

Tabla 14 Protocolo de comunicación (mensajes)

Comandos

Los comandos más importantes que soporta son:

Nombre	Parámetros	Función
INIT	<i>ClassName</i> : tipo del robot	Añade un robot al entorno
	<i>Name</i> : nombre del robot	
	<i>Location</i> : posición inicial del robot en metros	
	<i>Rotation</i> : balanceo, viraje y cabeceo del robot en radianes.	
DRIVE	<i>Left</i> : velocidad de giro ruedas lado izquierdo	Controlar el robot
	<i>Right</i> : velocidad de giro ruedas lado derecho	
	<i>Normalized</i> : si los valores son normalizados o no	
	<i>Light</i> : apaga/enciende el foco	
SET	<i>Flip</i> : endereza al robot	Controlar articulación, cámara o pinza del robot
	<i>Type</i> : puede ser Joint, Camera, Gripper	
	<i>Name</i> : nombre de la articulación, cámara o pinza	
	<i>Opcode</i> : Código de operación	
GETGEO	<i>Params</i> : Valor del Opcode	Consultar la geometría
	<i>Type</i> : tipo del objeto	
GETCONF	<i>Name</i> : nombre del objeto (es opcional)	Consultar la configuración
	<i>Type</i> : tipo del objeto	

Tabla 15 Protocolo de comunicación (comandos)

Si se desea ampliar estos conocimientos consultar el manual de USARSim en http://robotics.mem.drexel.edu/mhsieh/Courses/MEM380I/resources/USARSim-manual_3.1.3.pdf.



2.1.4.5 Controladores

Con el paquete de USARSim se incluyen cinco controladores:

- Testing Control Interface (USAR_UI): “overflow” de la salida
- MOAST
- Pyro: está en desuso
- Player: funciona sobre Linux
- SimpleUI

El controlador que se ha usado en este proyecto no es ninguno de los anteriores. El que se ha elegido es para una versión superior de USARSim (UDK) y se denomina Iridium. Los motivos por los cuales se ha elegido son:

- Es el controlador usado actualmente por el NIST
- GUI es fácil de usar (incluye menús desplegables)
- UI sugiere automáticamente valores disponibles
- Sensores envían información en tiempo real a través de barras de información
- Mapa en tiempo real con datos reales sobre el terreno
- Doble clic sobre la ventana de log hace que se vuelva a ejecutar el comando
- Botón de parada de emergencia para parar el movimiento del robot



Ilustración 70 Iridium GUI



2.2 Descripción de la solución

Para resolver el problema inicialmente planteado y aportar una solución válida se ha dividido el proyecto básicamente en dos grandes tareas: creación del entorno marciano y simulación del entorno marciano. Cada una de estas se subdivide a su vez en subtareas.

2.2.1 Creación del entorno marciano

Como paso previo al desarrollo del entorno marciano fue preciso llevar a cabo dos tareas de investigación. Por un lado se realizó un estudio de la composición del planeta en cuanto a su atmósfera y terreno. Se buscó información en páginas web especializadas en esta materia e imágenes en la página web de la NASA (<http://www.nasa.gov>) —obtenidas en diferentes misiones enviadas al planeta a lo largo de estos años de exploración espacial. Por otro lado se estudió en profundidad la herramienta utilizada para la creación del mundo marciano, Unreal Level Editor, anteriormente explicada en el punto 2.1.3.1. Para ello se buscaron tutoriales y guías de usuario en Internet, entre los que destacan Unreal Wiki, UnrealED 3.0 y Unreal Editor User Guide (consultar Referencias). También se aprendieron conceptos fundamentales como *Viewport*, *Builder Brush*, *Subtractive BSP*, *Additive BSP*, *Volume*, *Sunlight*, *ZoneInfo*, *Terrain*, *PlayerStart* y *SkyBox* (entre otros) que fueron decisivos a la hora de obtener la solución final. Una vez aprendidos los conceptos necesarios y familiarizados con el manejo del programa se procedió al desarrollo del entorno tomando como referencia las imágenes de Marte y teniendo muy en cuenta los requisitos dados inicialmente.




2.2.1.1 Creación de un mapa para un mundo marciano

Requerimientos:

- Mapa de exteriores (outdoor map)
- Dimensiones 30000 x 60000 x 80000

Procedimiento:

Para crear el mapa (en el interfaz del Unreal Level Editor) se han seguido los siguientes pasos:

1º Clic izquierdo sobre New Map 

Aparece un nuevo mapa vacío.

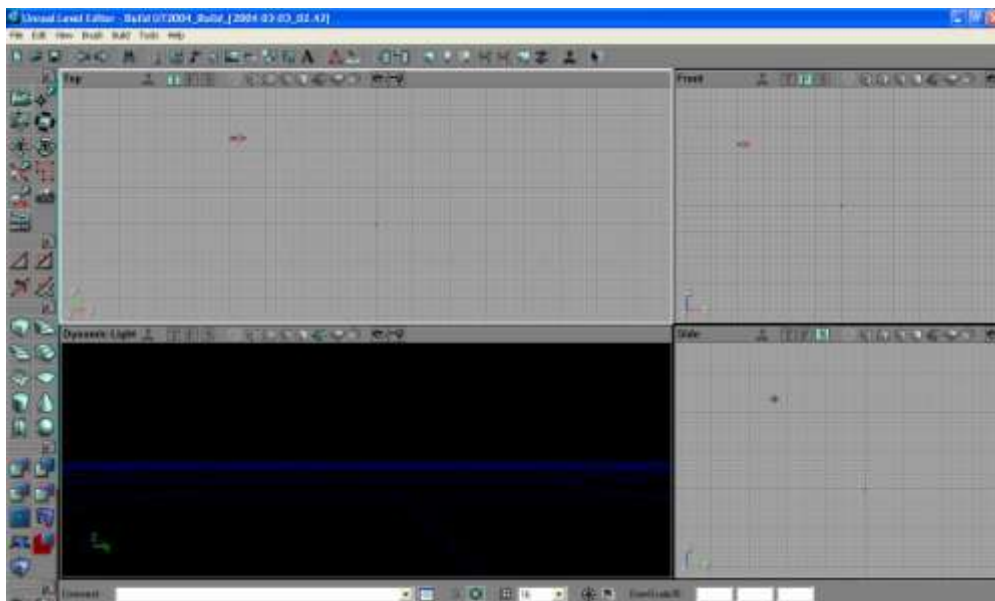



Ilustración 71 Mapa vacío

2º Clic derecho sobre “Cube” 

Aparece un cuadro de diálogo como el que se muestra más abajo con las propiedades del “Cube Builder”. Se han seleccionado los siguientes valores:

- Height 30.000
- Width 60.000



- Breadth 80.000

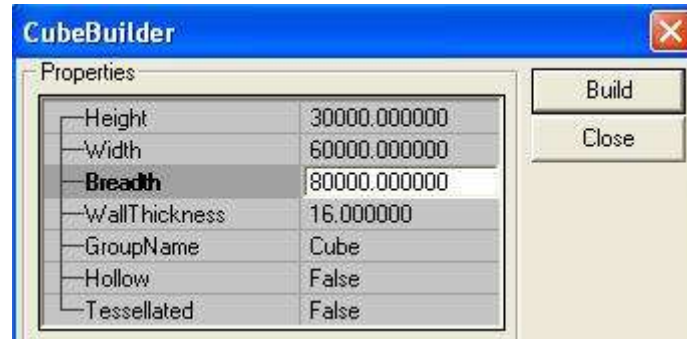


Ilustración 72 Propiedades del Cube Builder

Seguidamente se ha pulsado sobre “Build”.

Ahora aparece dibujado en los cuatro *viewports* (Top, Front, Side, 3D) el *brush* con forma de cubo de alambre rojo que nos servirá de molde para construir el mundo marciano.

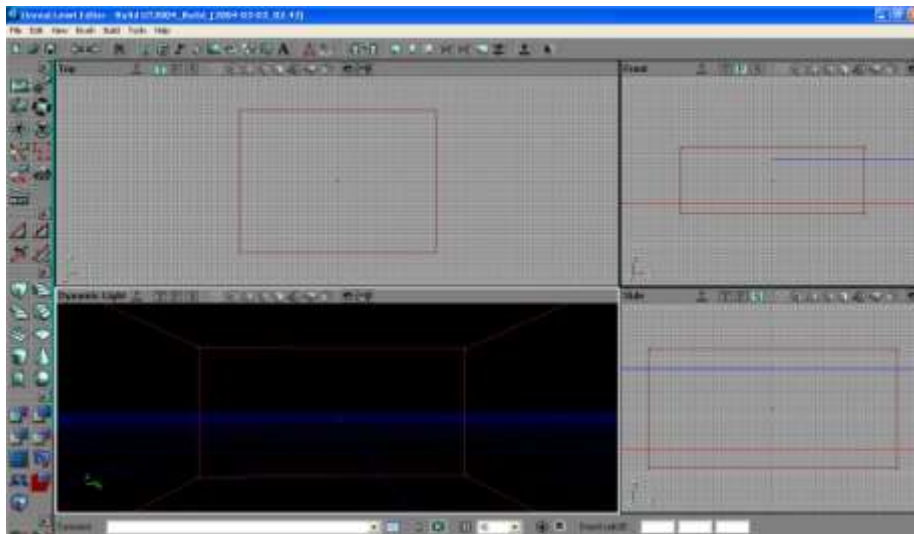



Ilustración 73 Brush con forma de cubo

El siguiente paso sería elegir una textura para la superficie del cubo. No importa demasiado cuál se elija aunque interesa que sea un color que destaque sobre el resto del entorno para facilitar la visualización del terreno y de los demás elementos del mapa.

3º Clic izquierdo sobre *Texture Browser* 

Aparece un cuadro de diálogo como el siguiente:

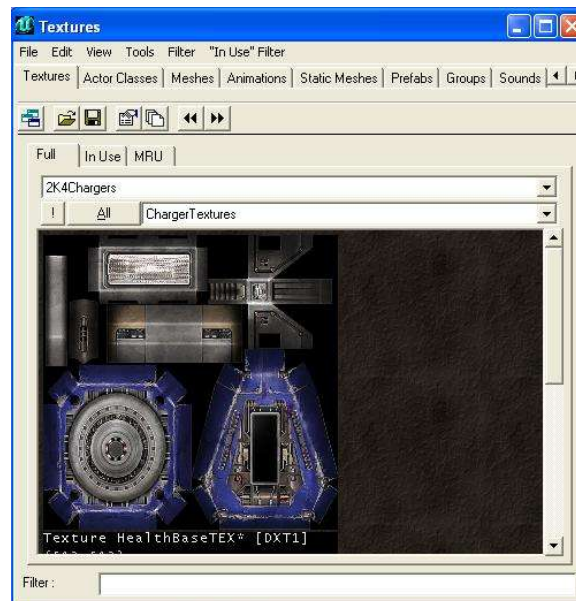



Ilustración 74 Buscador de texturas

4º Hacer clic en Open Package . Las texturas se encuentran agrupadas en paquetes y tienen la extensión .utx (unreal texture).

Aparece un cuadro de diálogo (ilustración número 75) donde se muestran todos los paquetes disponibles. Se ha seleccionado el paquete UCGeneric.utx.

A continuación pulsar sobre “Abrir”.

Aparece un cuadro de diálogo (ilustración número 76). En el desplegable que está abajo a la izquierda debajo de “UCGeneric” se ha elegido el grupo de texturas “SolidColours” (aparece sombreado en azul).

De entre todas las texturas de este grupo se ha elegido “Texture GreenBlueA* [DXT1]” (ilustración número 77).

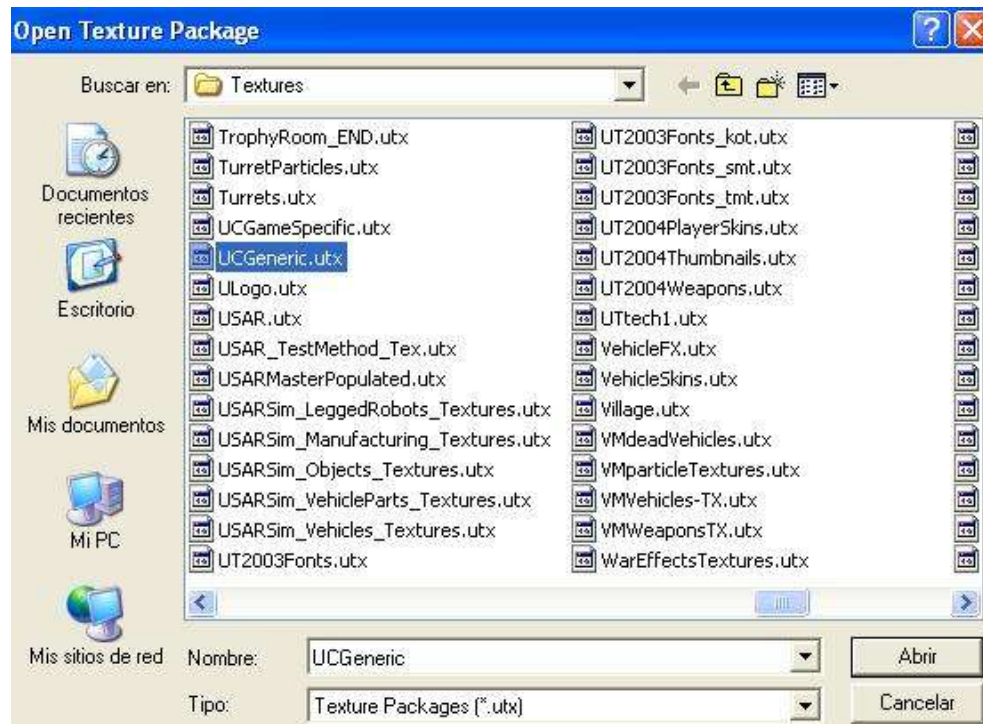


Ilustración 75 Paquete que contiene la textura para la superficie del cubo

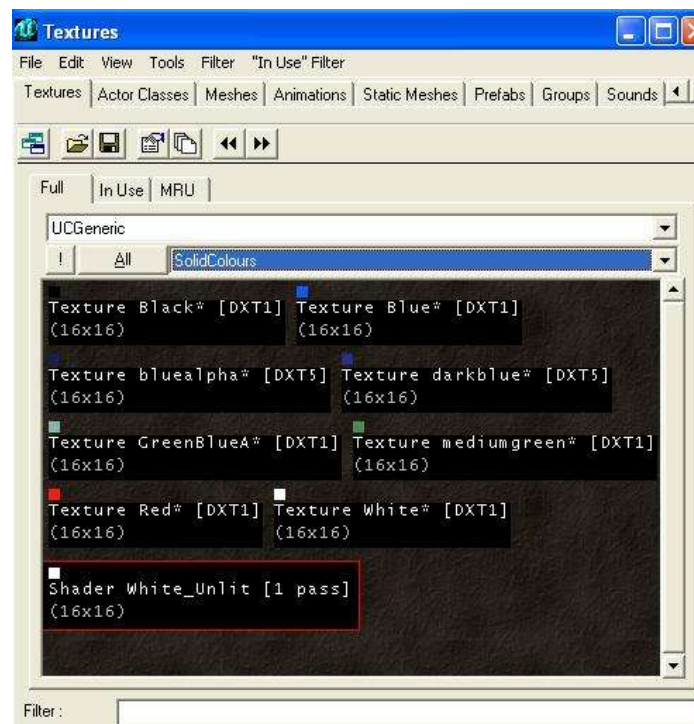


Ilustración 76 Grupo de texturas "SolidColours" dentro del paquete UCGeneric

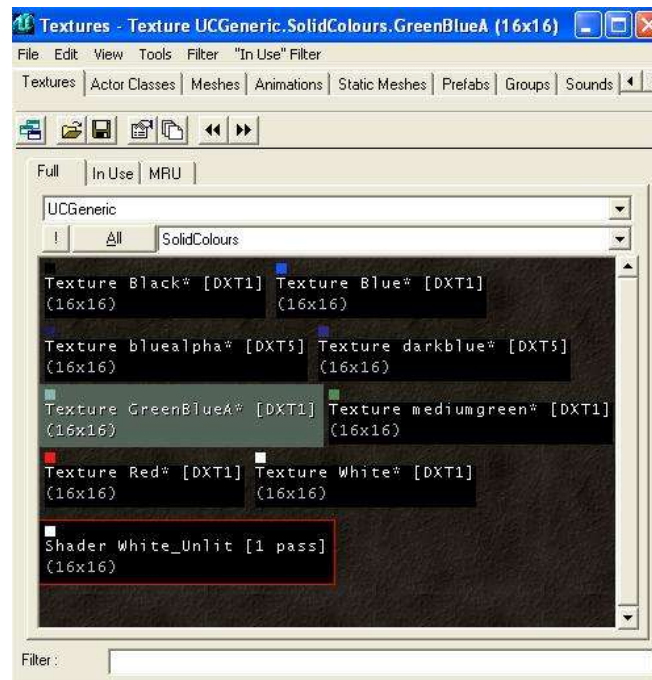


Ilustración 77 Textura elegida para la superficie del cubo

5º Posteriormente se ha minimizado el cuadro de diálogo y se ha hecho clic izquierdo en

Subtract .

Como se puede apreciar en la ilustración número 78, en el viewport situado abajo a la izquierda (3D) el cubo de alambre rojo aparece ahora con la textura que se ha elegido previamente.

6º El siguiente paso sería separar el molde usado para construir el mundo (cubo de alambre rojo). Para ello seleccionar en el Top viewport el cubo de alambre rojo (clic izquierdo) y manteniendo presionada la tecla Ctrl. arrastrar mediante movimiento del ratón el cubo hasta situarlo fuera de su posición original, por ejemplo a la izquierda. Hacer lo mismo en el Front viewport pero en este caso arrastrar el cubo hacia arriba. Después de haber hecho esto, en el Top viewport, Front viewport, Side viewport y 3D viewport se observa lo siguiente (ilustración número 79). Como se puede apreciar ahora aparecen dos cubos: uno de color rojo (molde empleado) y otro de color amarillo que es el espacio que ocupará el mundo.

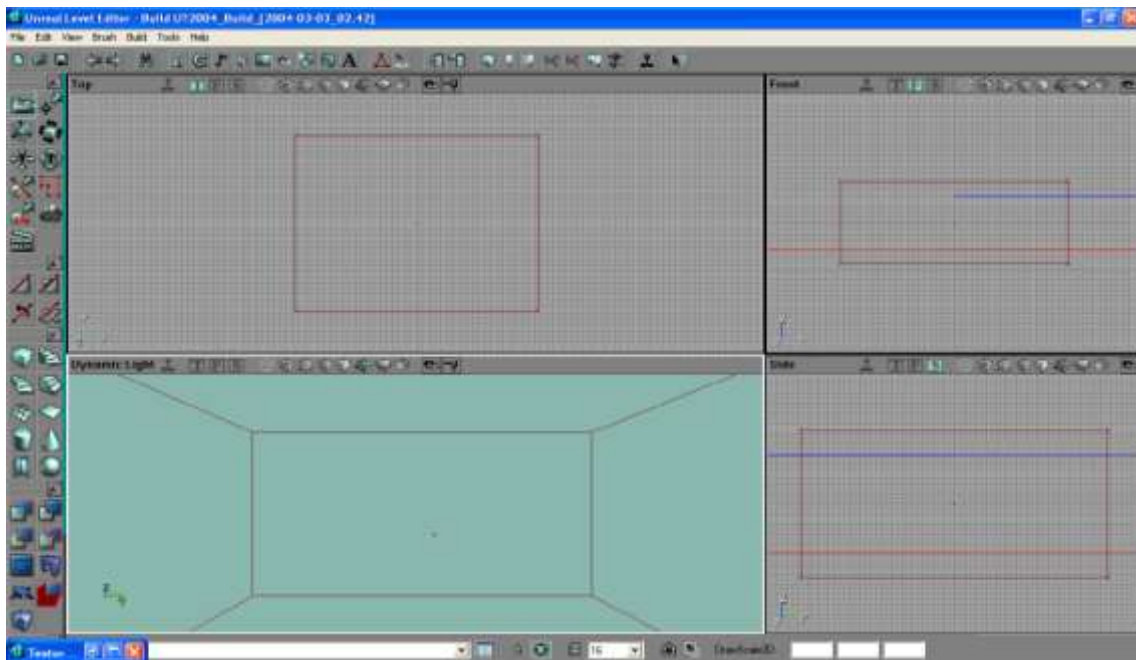


Ilustración 78 Cubo con la textura elegida

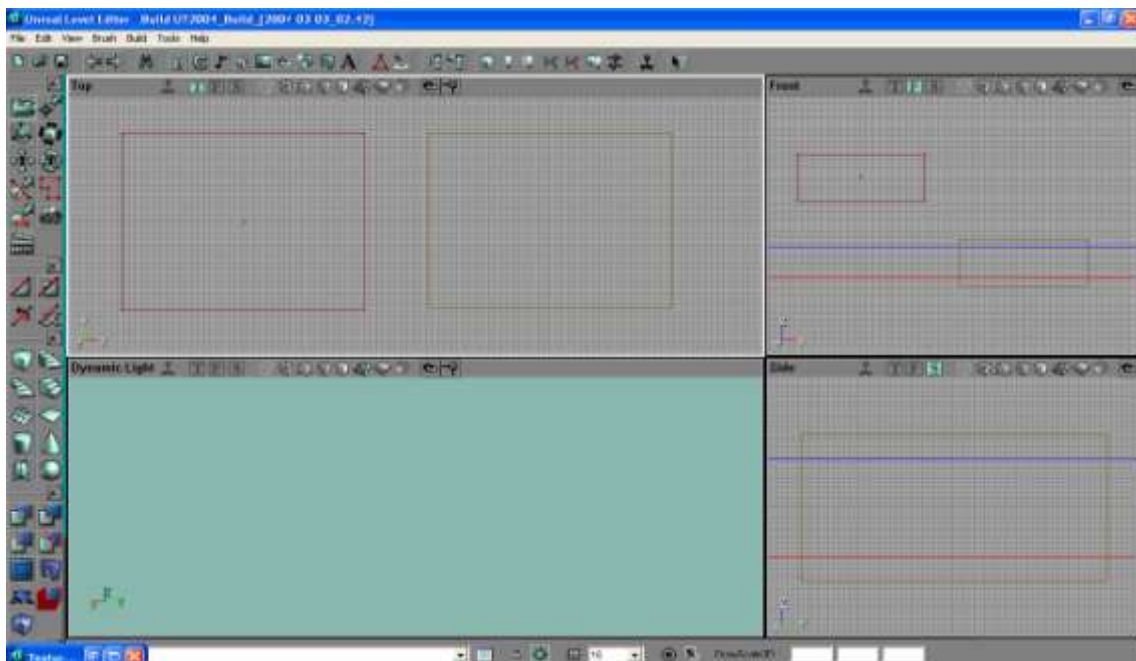



Ilustración 79 Cubo amarillo que delimita el mundo marciano

7º Una vez el mapa está creado hay que añadir un LimitationVolume. Estos se utilizan generalmente para trazar los límites de un mapa de exteriores como este.



En primer lugar se ha seleccionado el brush rojo (clic izquierdo) y a continuación se ha hecho clic izquierdo en Volume  y se ha elegido la opción LimitationVolume, tal y como se aprecia en la ilustración número 79.



BlockingVolume
Volume
LadderVolume
PhysicsVolume
DefaultPhysicsVolume
PressureVolume
WaterVolume
SnipingVolume
ConvoyPhysicsVolume
LavaVolume
xFallingVolume
IonCannonKillVolume
LimitationVolume
HitScanBlockingVolume
ASCriticalObjectiveVolume
LeavingBattleFieldVolume

Ilustración 80 Seleccionar LimitationVolume

8º Manteniendo la tecla Ctrl presionada al mismo tiempo que hacemos clic izquierdo sobre el brush rojo arrastrar el ratón hasta que el LimitationVolume (brush violeta) se haga visible. En la siguiente ilustración puede verse el resultado de esta acción: ahora en el Top y Front viewport además del brush rojo y amarillo aparece un tercer brush violeta, el correspondiente al LimitationVolume.

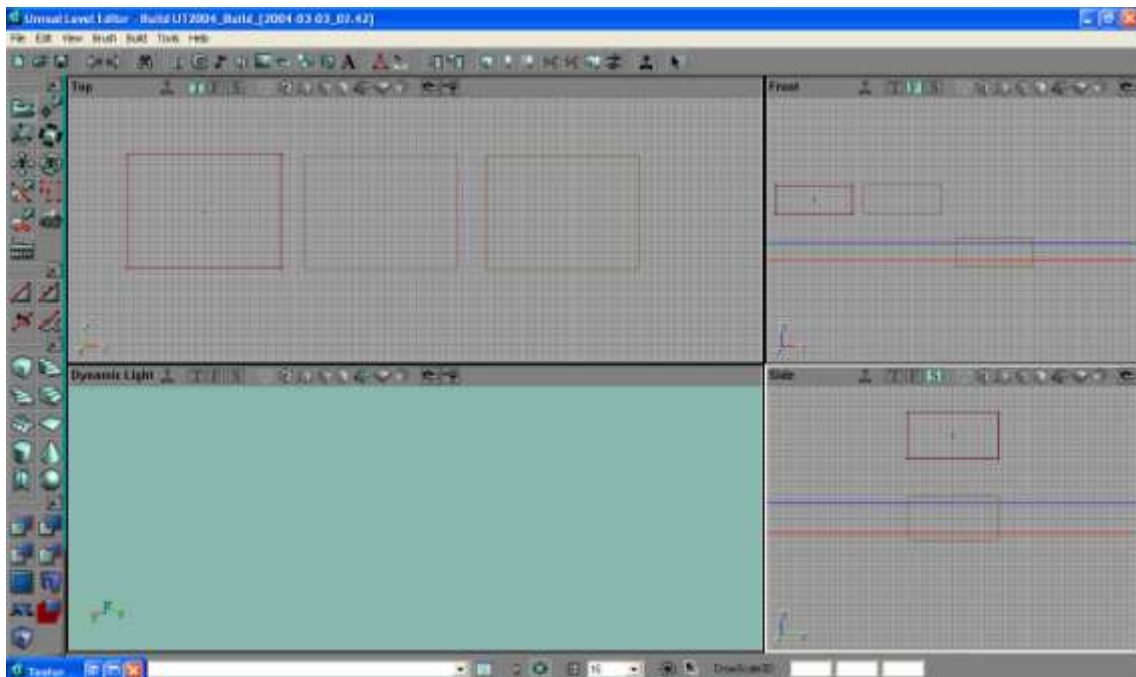


Ilustración 81 LimitationVolume

9º Alinear el LimitationVolume (brush violeta) con el mapa (brush amarillo) de tal forma que quede superpuesto tanto en el Top como en el Front viewport. (Ver ilustración siguiente)

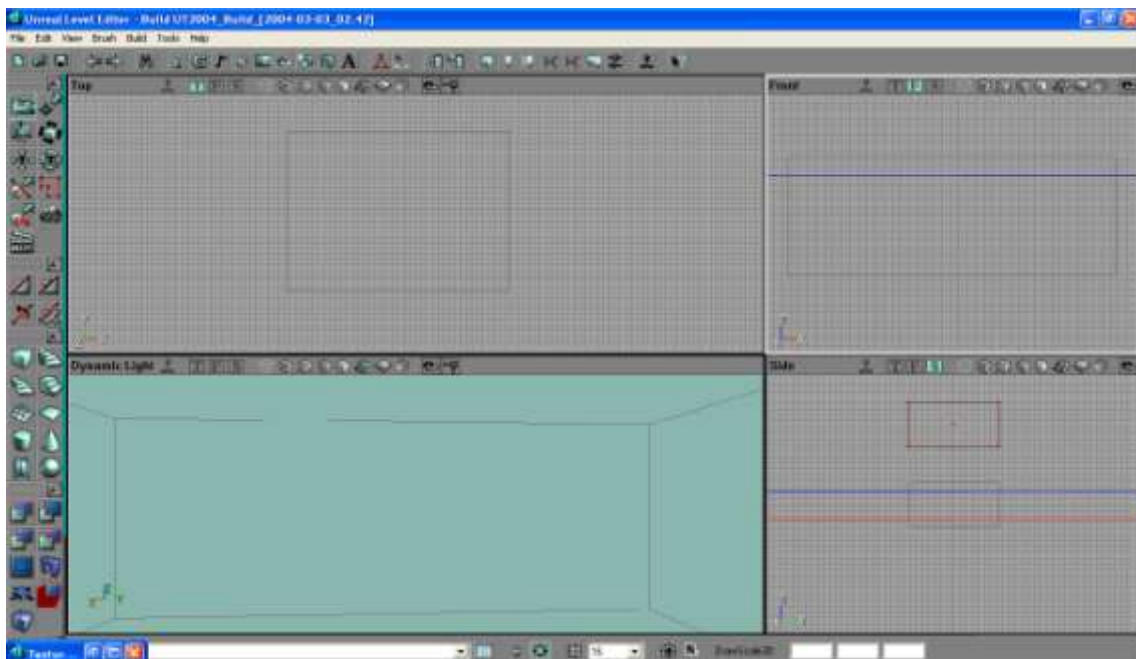


Ilustración 82 LimitationVolume alineado con el mapa



2.2.1.2 Creación de la atmósfera marciana

Requerimientos:

- Su composición es principalmente dióxido de carbono
- Presencia de partículas de polvo en suspensión
- Tonalidad marrón claro

Procedimiento:

Para conseguir una atmósfera similar a la de Marte se han definido varias zonas en el mapa como “*Distance Fog*”. Esto permite crear una ilusión óptica de niebla que hace que lo que está más alejado se perciba de forma difuminada y que conforme el rover va avanzando por el terreno los desniveles, montañas, cráteres y demás elementos dispuestos a lo largo de él se vayan haciendo visibles.

Para crear una zona de este tipo es necesario añadir en el mapa un actor de tipo ZoneInfo. En la interfaz del Unreal Level Editor se han realizado las siguientes acciones:

1º Clic izquierdo sobre Actor Class Browser 

Aparece un cuadro de diálogo en el que se selecciona Info > ZoneInfo y a continuación se minimiza el cuadro de diálogo.

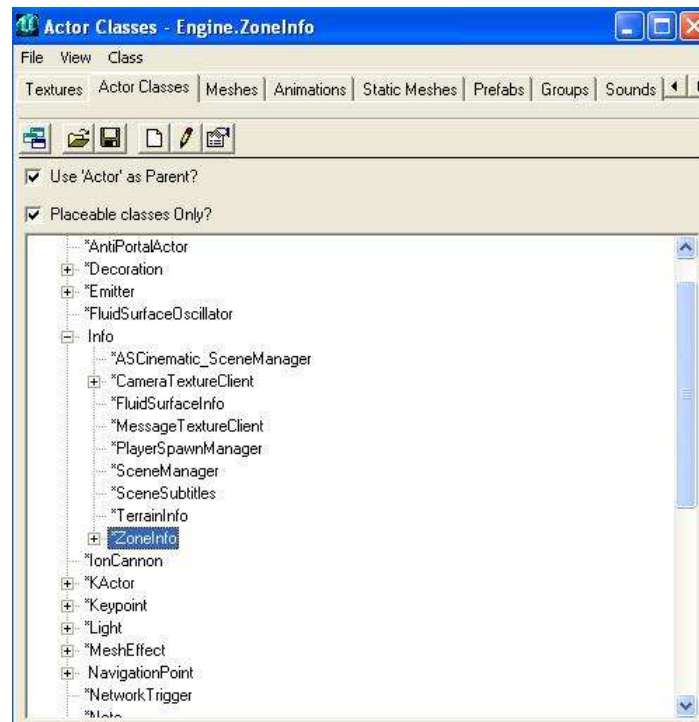


Ilustración 83 Actor "ZoneInfo"

2º En el Top viewport se ha hecho clic derecho sobre la zona central del mapa y se ha seleccionado "Add ZoneInfo Here".

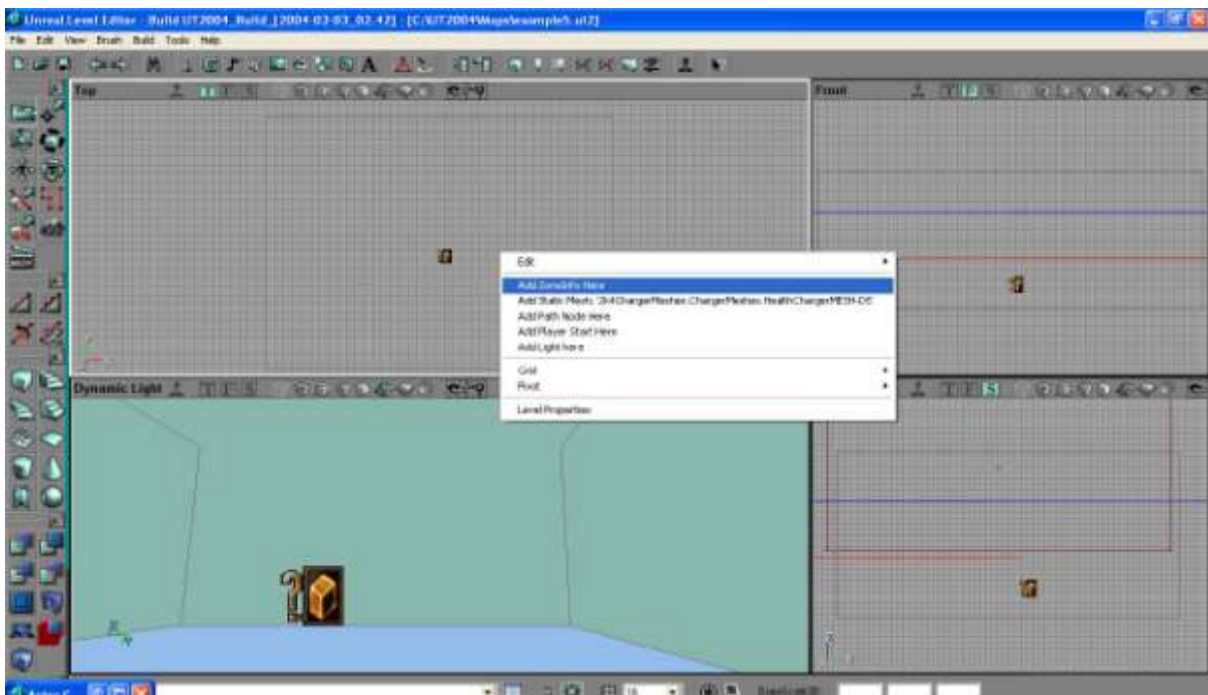


Ilustración 84 Añadir al mapa un actor de tipo "ZoneInfo"



A continuación se ha hecho clic derecho sobre el icono de “ZoneInfo” para acceder a sus propiedades.

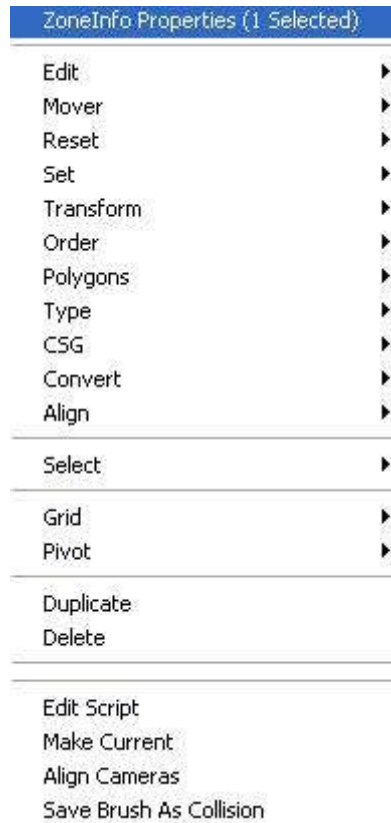


Ilustración 85 Seleccionar ZoneInfo Properties para acceder a sus propiedades

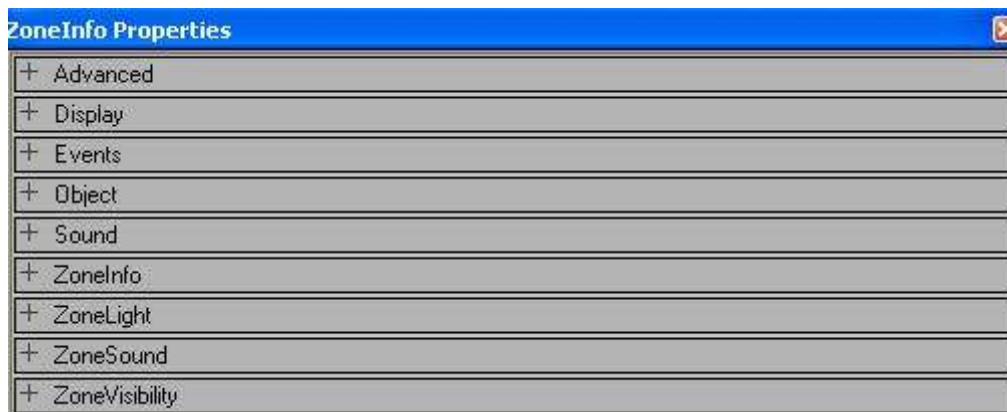


Ilustración 86 Propiedades de "ZoneInfo"

3º En el cuadro de diálogo que aparece arriba con las diferentes propiedades de “ZoneInfo”, se ha hecho clic izquierdo sobre “Display” para seleccionarla y que se



muestren todas las opciones. Se ha modificado la propiedad “DrawScale” a 20 para que en UnrealEd su icono se vea mayor.

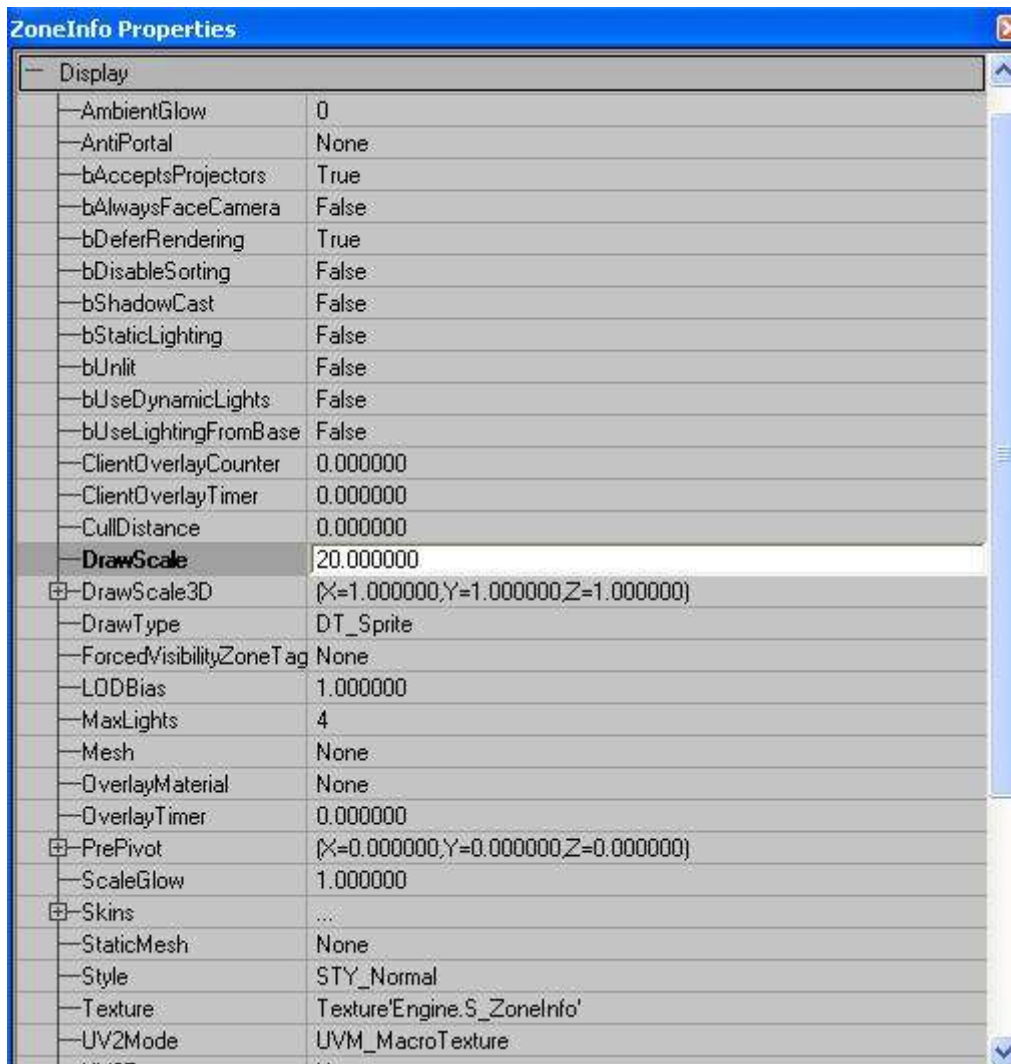


Ilustración 87 DrawScale a 20 para que el icono se perciba mayor

4º Seleccionar ahora la propiedad ZoneInfo (Información de la zona). Es muy importante que la opción “bDistanceFog” esté a True, de lo contrario no funcionará el “efecto niebla”.

- bClearToFogColor: permite que el rover vea más allá de la distancia en la que termina la niebla.
- bDistanceFog: habilita la distancia de la niebla para esta zona.



- bTerrainZone: se ha habilitado esta opción para indicar al motor que en esta zona hay terreno (más adelante se explicará cómo se ha creado) que se ve afectado por la distancia de niebla.
- LocationName: descripción de la zona

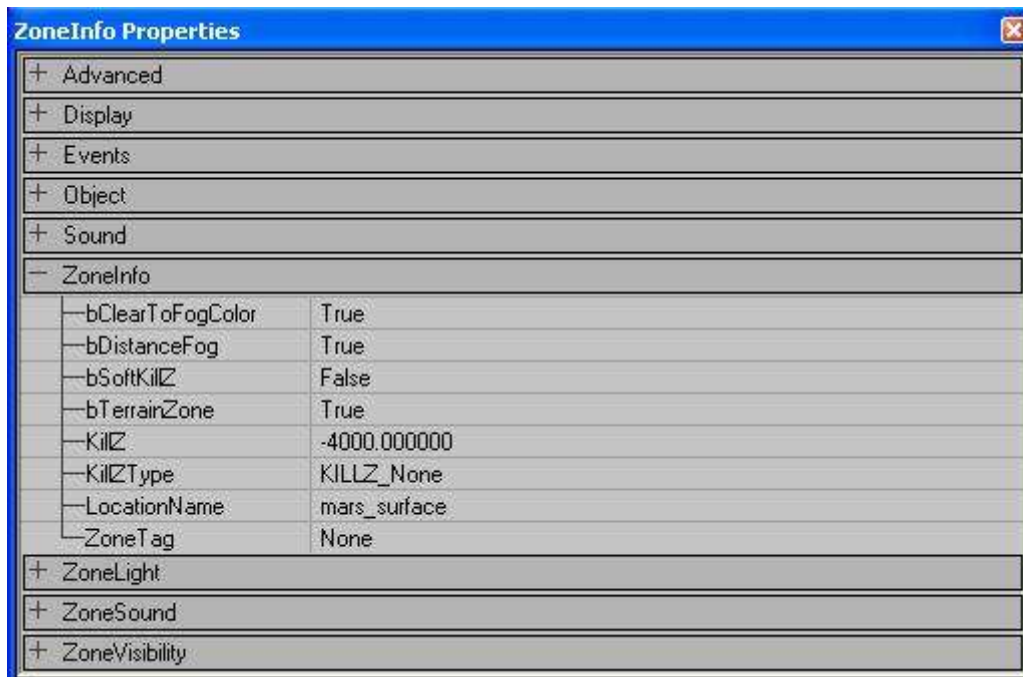


Ilustración 88 Propiedad "ZoneInfo" del actor "ZoneInfo"

5º Seleccionar la propiedad ZoneLight (Luz de la zona). Se han configurado las siguientes opciones:

- AmbientBrightness: cuánto de brillante se quiere que sea la luz. En nuestro caso no se quiere que sea muy brillante por lo que se ha elegido 40 (sobre 255).
- AmbientHue: permite seleccionar un color a través del espectro de color por defecto.
- AmbientSaturation: cantidad de luz blanca que se mezcla con el color seleccionado. Se ha elegido 125 (sobre 255).
- DistanceFogColor: color de la niebla. Se ha elegido un tono marrón rojizo similar al color de la atmósfera marciana.
 - Componente Roja: 83
 - Componente Verde: 27
 - Componente Azul: 11



- DistanceFogEnd: distancia desde el rover a partir de la cual la niebla se vuelve 100% opaca. De este modo se indica al motor que descarte aquellos objetos y aquellas zonas que se encuentren más allá de esta distancia lo que supone un incremento importante de la velocidad.
- DistanceFogStart: distancia desde el rover a partir de la cual la niebla se vuelve visible. Para conseguir un efecto natural se recomienda que esta distancia sea menor a 2.000⁴⁷

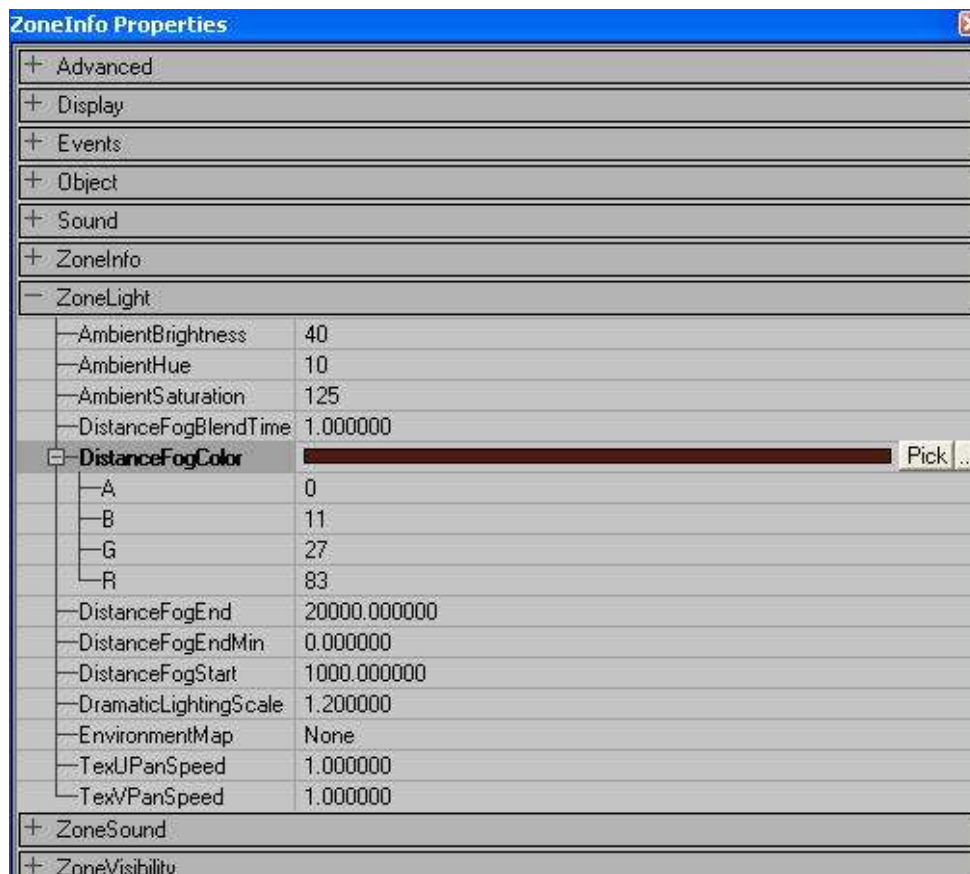


Ilustración 89 Propiedad "ZoneLight" del actor "ZoneInfo"

2.2.1.3 Iluminación del mapa

Requerimientos:

- Luz procedente de los rayos solares (tonalidades marrones y rojizas)

⁴⁷ Fuente consultada: http://wiki.beyondunreal.com/Legacy:Distance_Fog



Procedimiento:

Para crear un efecto de luz solar natural sobre el terreno es preciso añadir en el mapa un actor de tipo Sunlight.

Para ello en la interfaz del Unreal Level Editor se han seguido los siguientes pasos:

1º Clic izquierdo sobre Actor Class Browser 

Aparece un cuadro de diálogo en el que se selecciona Light > Sunlight y a continuación se minimiza el cuadro de diálogo.

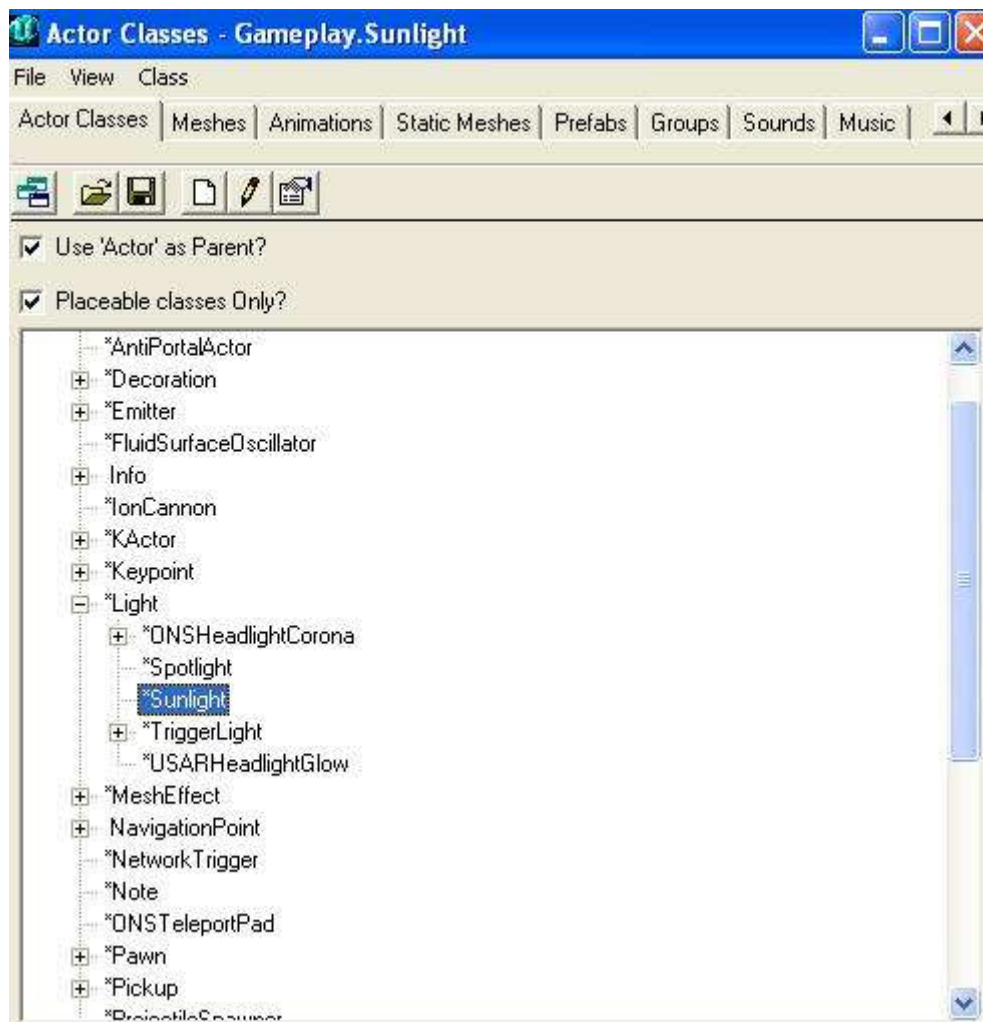


Ilustración 90 Actor "Sunlight"

2º En el Top viewport se ha hecho clic derecho sobre la zona central del mapa y se ha seleccionado "Add Sunlight Here". Este paso se ha repetido cuatro veces para añadir



cuatro actores de este tipo. El resultado final puede observarse en la siguiente ilustración.

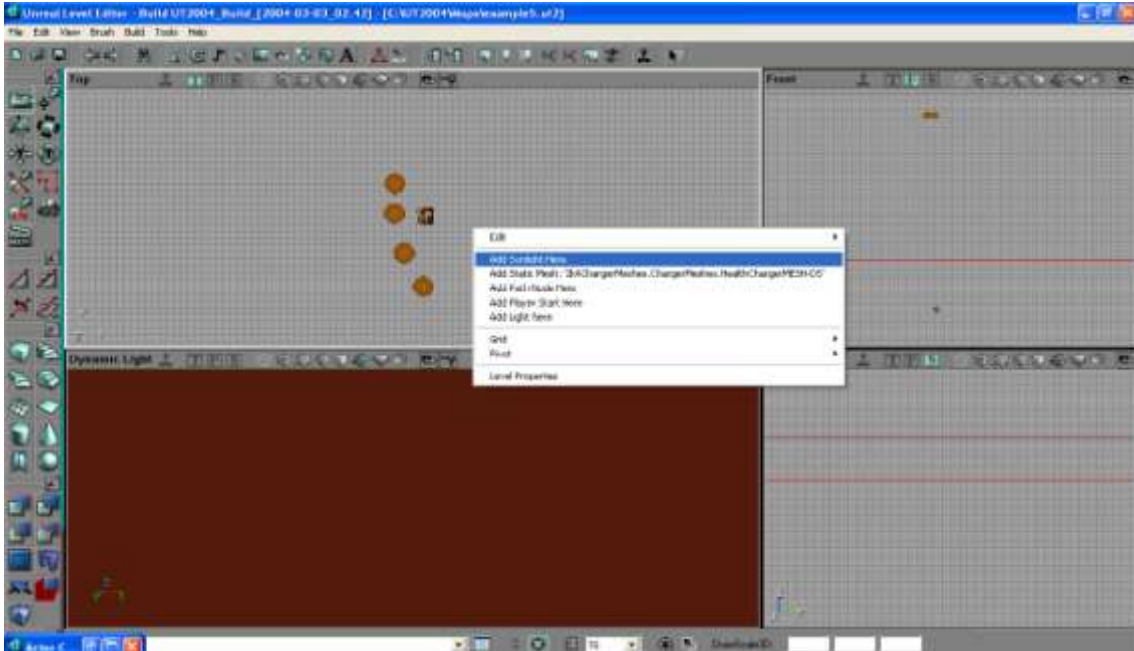


Ilustración 91 Añadir al mapa un actor de tipo "Sunlight"

Los actores se han dispuesto de la forma que se aprecia en el Top viewport en la imagen superior. En el Front viewport (vista de frente) se puede observar cómo se han colocado en la parte superior del mapa, cerca del techo.

A continuación se ha hecho clic derecho sobre el primer icono de "Sunlight" (de arriba abajo tal y como aparecen en la ilustración anterior) para acceder a sus propiedades.



Ilustración 92 Seleccionar Sunlight Properties para acceder a sus propiedades



Las propiedades de este actor que han sido necesarias configurar se detallan a continuación:

3º En el cuadro de diálogo que aparece abajo con las diferentes propiedades de “Sunlight”, se ha hecho clic izquierdo sobre “LightColor” y se ha elegido la siguiente configuración:

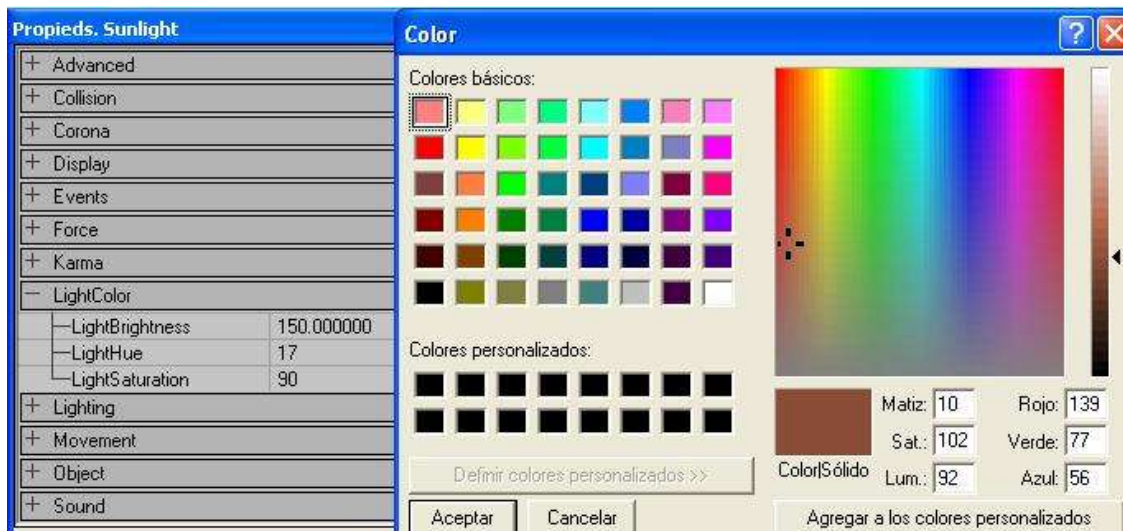


Ilustración 93 Propiedad "LightColor" para el primer actor "Sunlight"

4º En el cuadro de diálogo que aparece abajo con las diferentes propiedades de “Sunlight”, se ha hecho clic izquierdo sobre “Lighting” y únicamente se ha habilitado la propiedad bSpecialLit para indicar que esta luz solamente afectará a superficies concretas que hayan sido definidas como Special Lit (más adelante se profundizará en este aspecto).

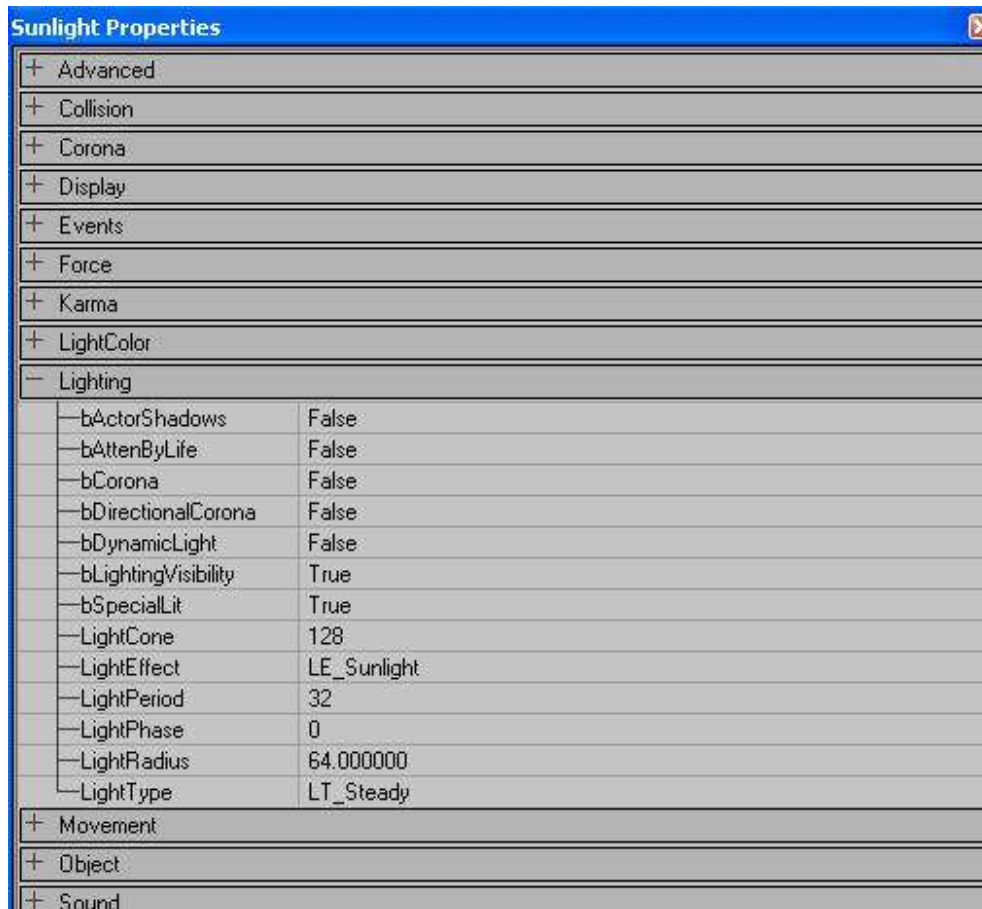


Ilustración 94 Propiedad "Lighting" para el primer actor "Sunlight"

A continuación se ha hecho clic derecho sobre el segundo icono de “Sunlight” (de arriba abajo tal y como aparecen en la ilustración 91) para acceder a sus propiedades.

5º En el cuadro de diálogo que aparece abajo con las diferentes propiedades de “Sunlight”, se ha hecho clic izquierdo sobre “LightColor” y se ha elegido la siguiente configuración:

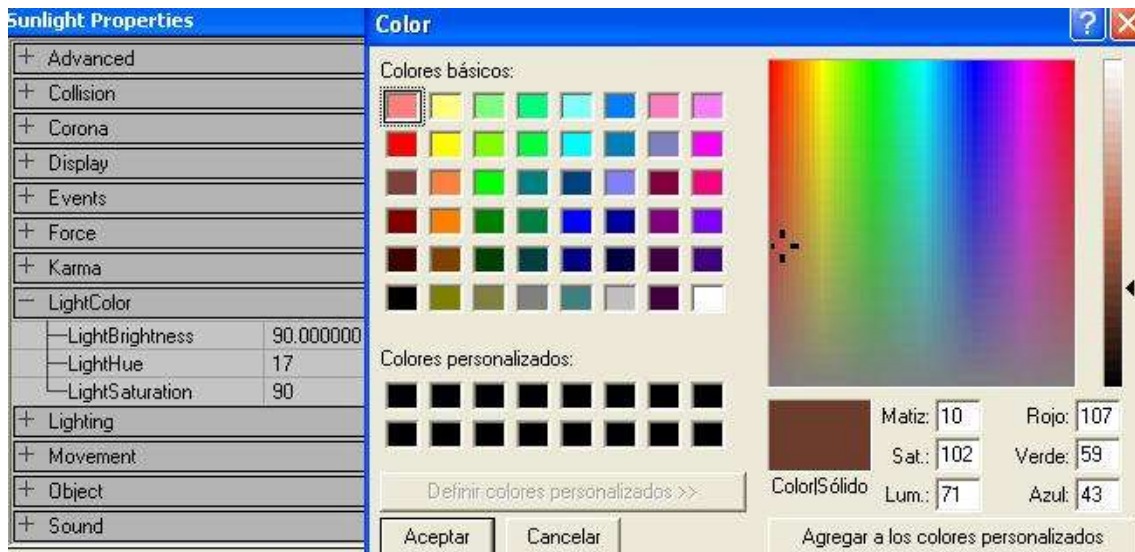


Ilustración 95 Propiedad "LightColor" para el segundo actor "Sunlight"

6º En el cuadro de diálogo que aparece abajo con las diferentes propiedades de "Sunlight", se ha hecho clic izquierdo sobre "Lighting" y se ha dejado la configuración que viene por defecto y que se muestra a continuación:

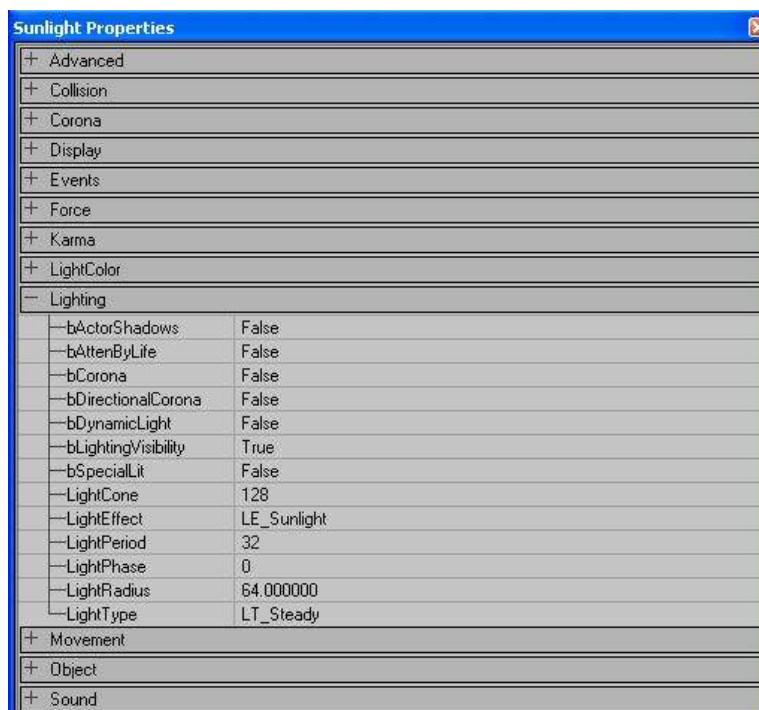


Ilustración 96 Propiedad "Lighting" para el segundo actor "Sunlight"

A continuación se ha hecho clic derecho sobre el tercer icono de "Sunlight" (de arriba abajo tal y como aparecen en la ilustración 91) para acceder a sus propiedades.

7º En el cuadro de diálogo que aparece abajo con las diferentes propiedades de “Sunlight”, se ha hecho clic izquierdo sobre “LightColor” y se ha elegido la siguiente configuración:

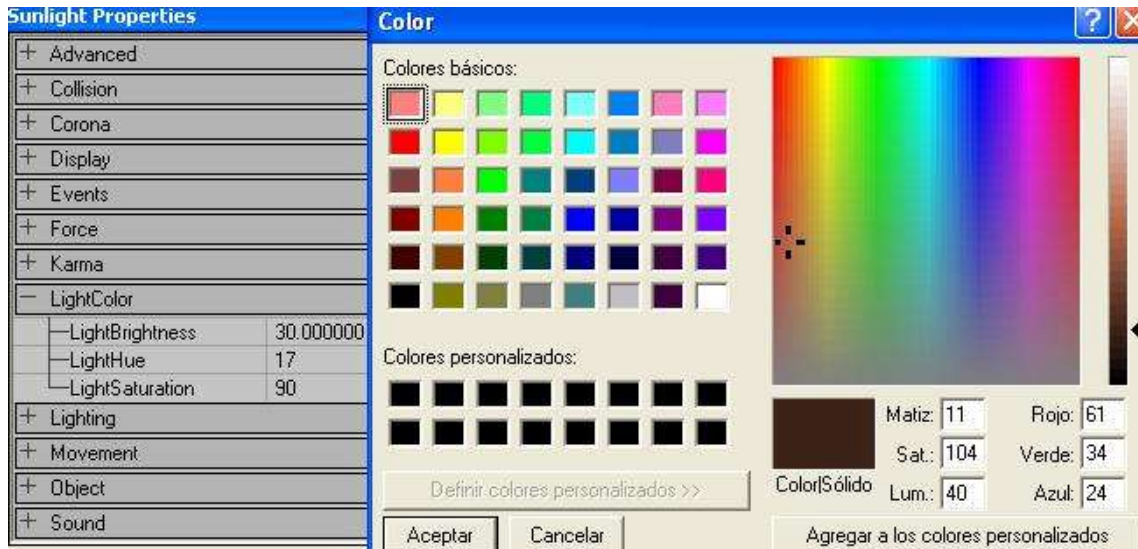


Ilustración 97 Propiedad "LightColor" para los actores "Sunlight" tercero y cuarto

8º En el cuadro de diálogo que aparece abajo con las diferentes propiedades de “Sunlight”, se ha hecho clic izquierdo sobre “Lighting” y se ha dejado la configuración que viene por defecto y que se muestra a continuación:

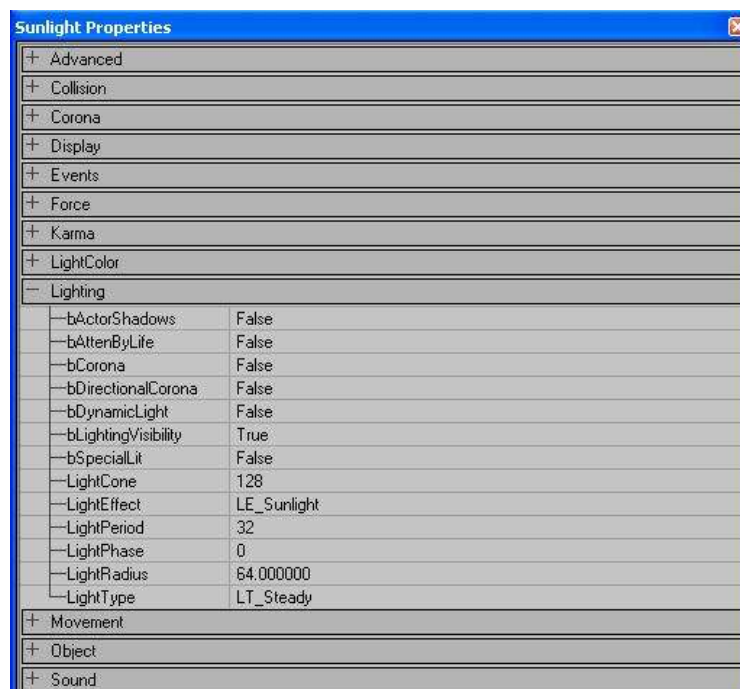


Ilustración 98 Propiedad "Lighting" para el tercer actor "Sunlight"



Seguidamente se ha hecho clic derecho sobre el cuarto icono de “Sunlight” (de arriba abajo tal y como aparecen en la ilustración 91) para acceder a sus propiedades.

9º Repetir el paso 5º, puesto que para el cuarto actor de tipo “Sunlight” se ha elegido la misma configuración que para el tercero.

10º En el cuadro de diálogo que aparece abajo con las diferentes propiedades de “Sunlight”, se ha hecho clic izquierdo sobre “Lighting” y se ha dejado la configuración que viene por defecto y que se muestra a continuación:

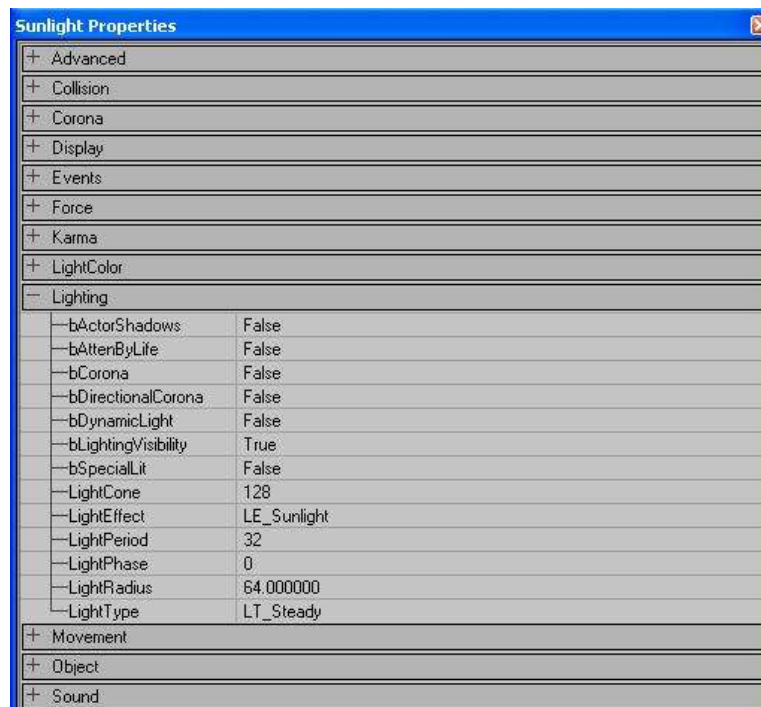


Ilustración 99 Propiedad "Lighting" para el cuarto actor "Sunlight"



2.2.1.4 Creación del terreno marciano

Requerimientos:

- Presencia de numerosos cráteres en su mitad sur.
- En su mitad norte hay importante actividad volcánica. En el engrosamiento de Tharsis destaca un volcán basáltico de 25 km de alto denominado Olympus Mons, considerado el mayor volcán conocido del sistema solar.
- Al este de la región de Tharsis, en el ecuador del planeta, se encuentra Valles Marineris, el mayor sistema de cañones conocido del sistema solar.
- El terreno de Marte presenta regiones oscuras donde existen rocas similares al basalto terrestre.
- La tonalidad del terreno es rojiza.

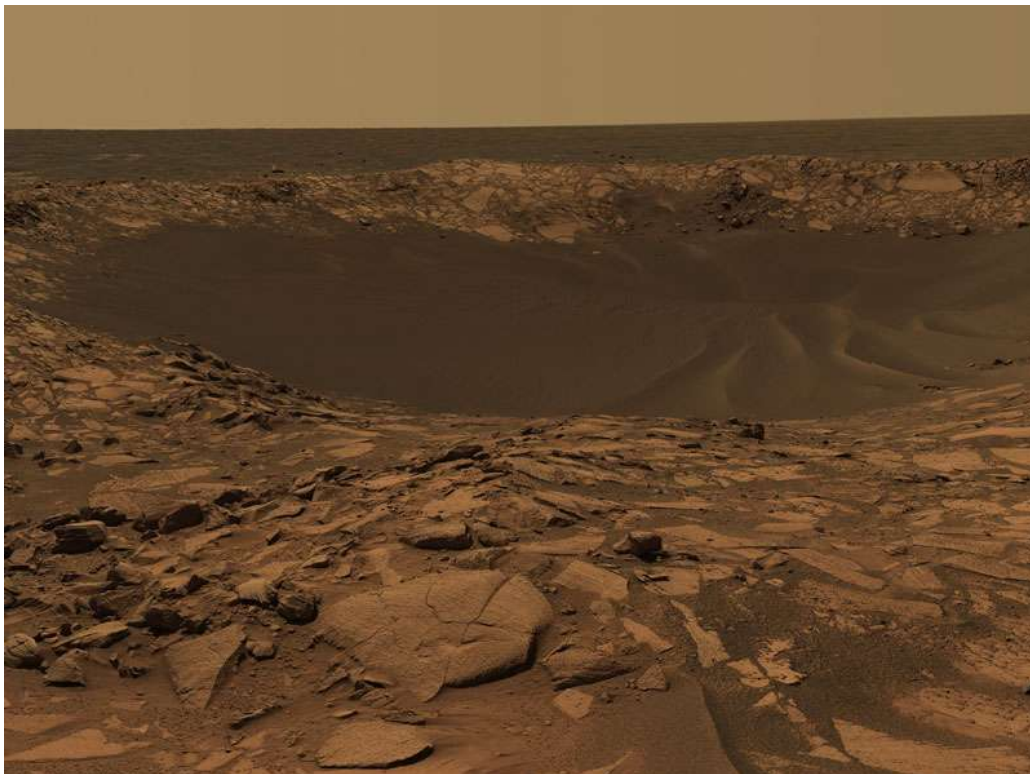


Ilustración 100 Beagle cráter fotografiado por el rover Opportunity ⁴⁸

⁴⁸ Fuente: NASA <http://apod.nasa.gov/apod/ap060919.html>



Ilustración 101 Endeavour cráter fotografiado por el rover Opportunity ⁴⁹

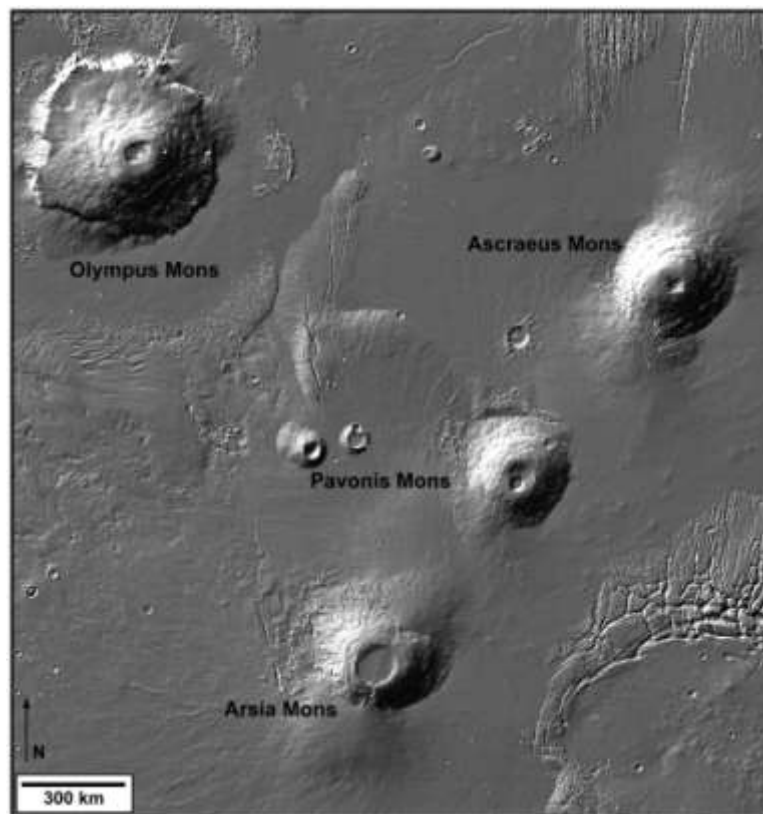


Ilustración 102 Volcanes en el engrosamiento de Tharsis (Mars Global Surveyor) ⁵⁰

⁴⁹ Fuente: NASA http://photojournal.jpl.nasa.gov/jpegMod/PIA14508_modest.jpg

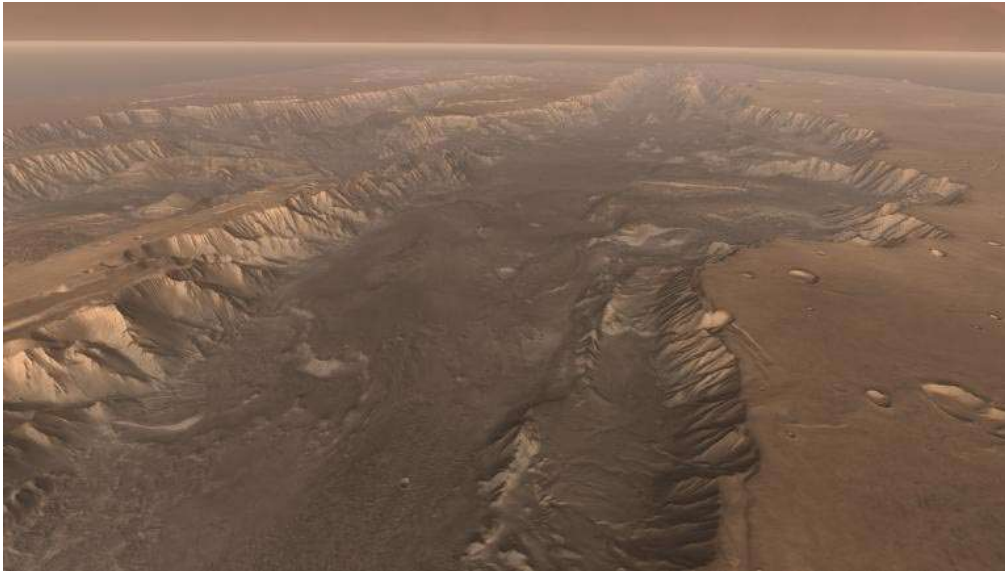


Ilustración 103 Melas Chasma - Valles Marineris (Mars Odyssey) ⁵¹

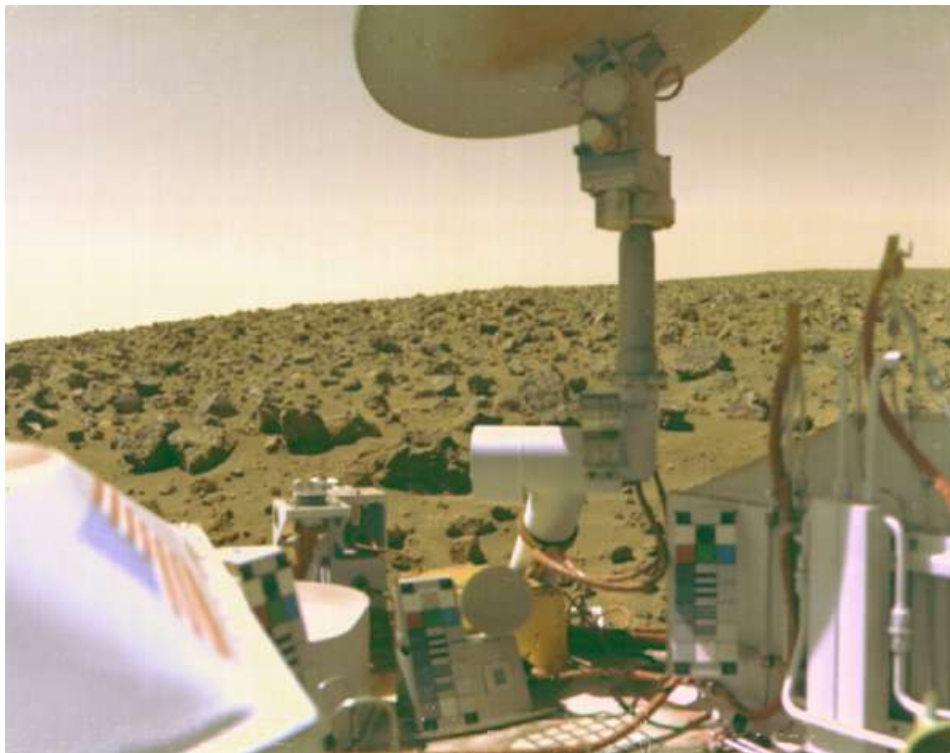


Ilustración 104 Terreno marciano fotografiado por el Viking 2 ⁵²


⁵⁰ Fuente: NASA http://www.nasa.gov/centers/goddard/news/topstory/2007/mars_volcanoes.html

⁵¹ Fuente: NASA http://www.nasa.gov/mission_pages/odyssey/odyssey-20060313.html



Procedimiento:

Para crear el terreno se ha procedido del siguiente modo:

1º Clic izquierdo sobre el botón Terrain Editing  que aparece en la caja de herramientas situada en la zona izquierda del interfaz de Unreal Level Editor.

Aparece una nueva ventana correspondiente al editor de terreno.

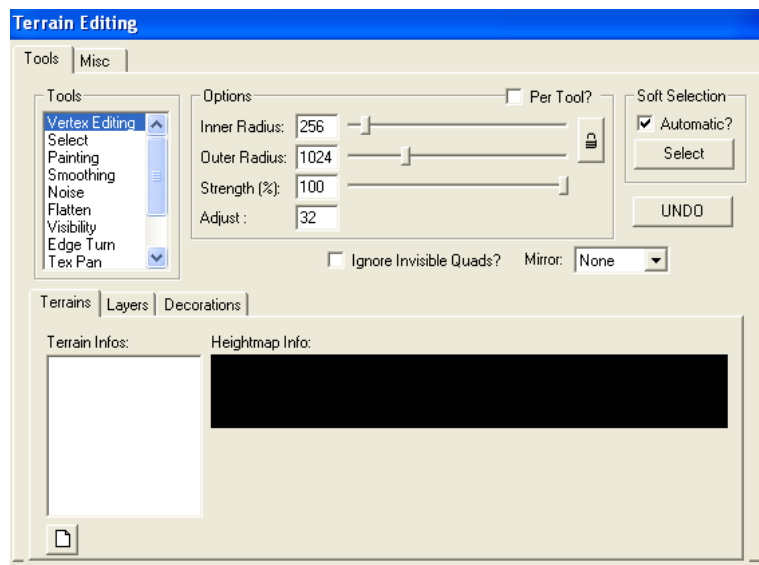


Ilustración 105 Editor de terreno

2º Clic izquierdo sobre el botón “New Terrain”  que aparece en la esquina inferior en la ilustración anterior.

Aparece un cuadro de diálogo con las siguientes opciones:

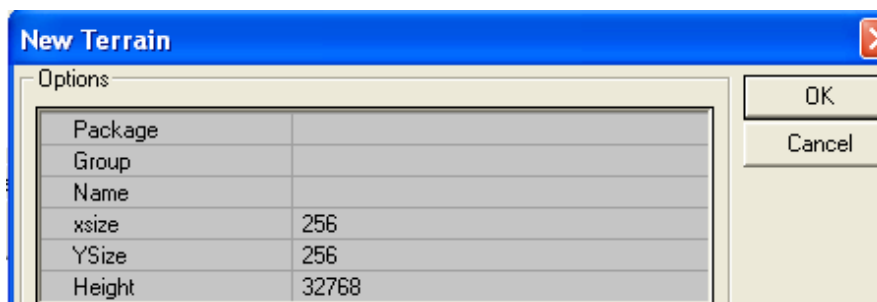


Ilustración 106 Crear un nuevo terreno

⁵² Fuente: NASA <http://grin.hq.nasa.gov/IMAGES/SMALL/GPN-2000-000426.jpg>



- Package: Nombre del paquete en el que se guardará la información relativa al terreno. Elegir myLevel para que se almacene en aquel correspondiente al mapa actual.
- Name: Nombre del terreno. Conviene elegir un nombre representativo como por ejemplo mars_terrain.
- xsize: Número de filas de la rejilla que ocupará el terreno
- YSize: Número de columnas de la rejilla que ocupará el terreno

Estos números tienen que ser siempre potencias de dos, de lo contrario falla el editor.

Dejar los valores que vienen por defecto.

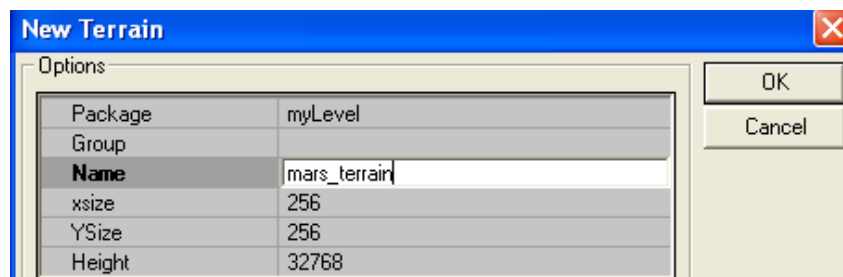



Ilustración 107 Configurar las opciones para la creación de un nuevo terreno

A continuación pulsar sobre el botón “OK”. Esto hará que se añada al mapa un actor de tipo “TerrainInfo” .

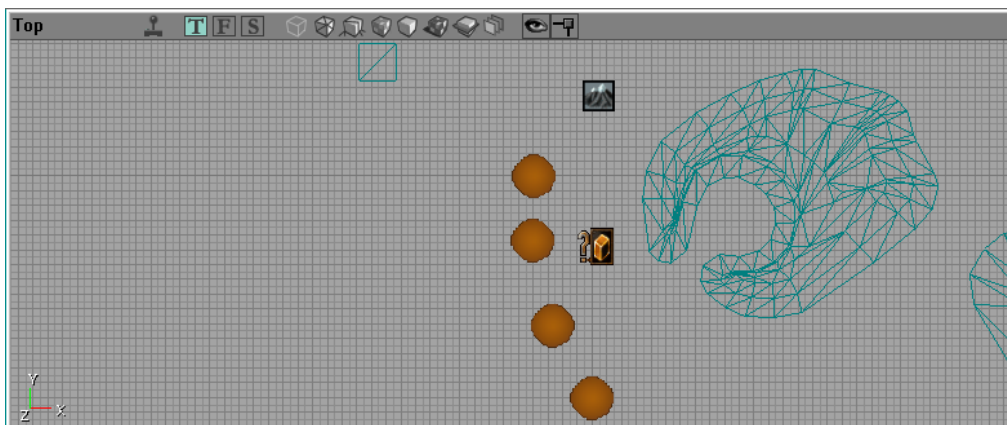


Ilustración 108 Mapa en el que se ha añadido un actor de tipo "TerrainInfo"

3º Alinear el terreno a la rejilla del mundo. Esto facilita la tarea de alinear *Static Mesh* a la rejilla del terreno posteriormente.



Ilustración 109 Mover el terreno a su origen

Si se quiere mover el terreno a cualquier otra posición del mapa pero se quiere que siga estando alineado a la rejilla se debe realizar lo siguiente:

4º Clic derecho para abrir las propiedades de “TerrainInfo” y en Advanced establecer la opción bEdShouldSnap a True.

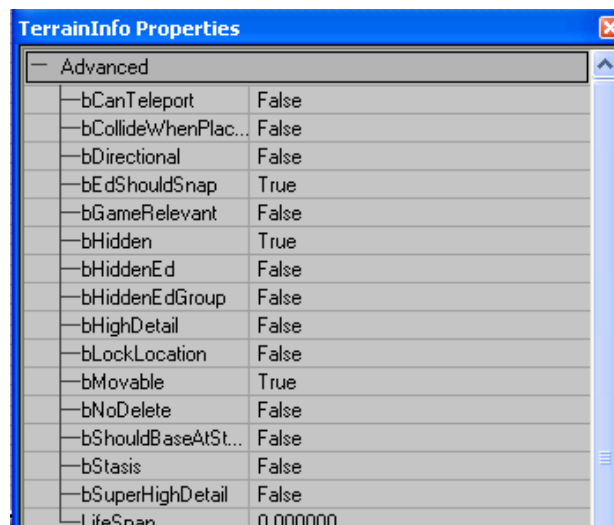



Ilustración 110 Propiedades del actor TerrainInfo

Para que el terreno se haga visible en el 3D viewport es preciso definir las capas y elegir las texturas que conformarán su aspecto. Los pasos que hay que seguir se detallan seguidamente:

1º Clic izquierdo sobre “Texture Browser” .

Aparece un cuadro de diálogo como el siguiente:

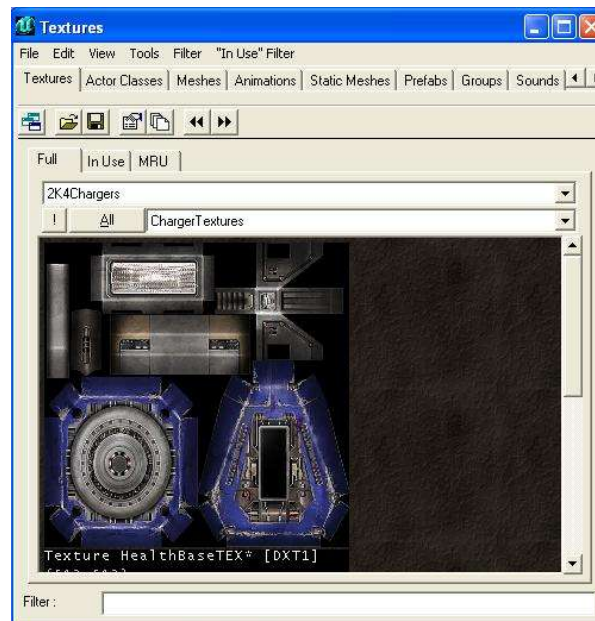



Ilustración 111 Buscador de texturas

2º Clic en Open Package . Seleccionar el paquete AlleriaTerrain.utx.

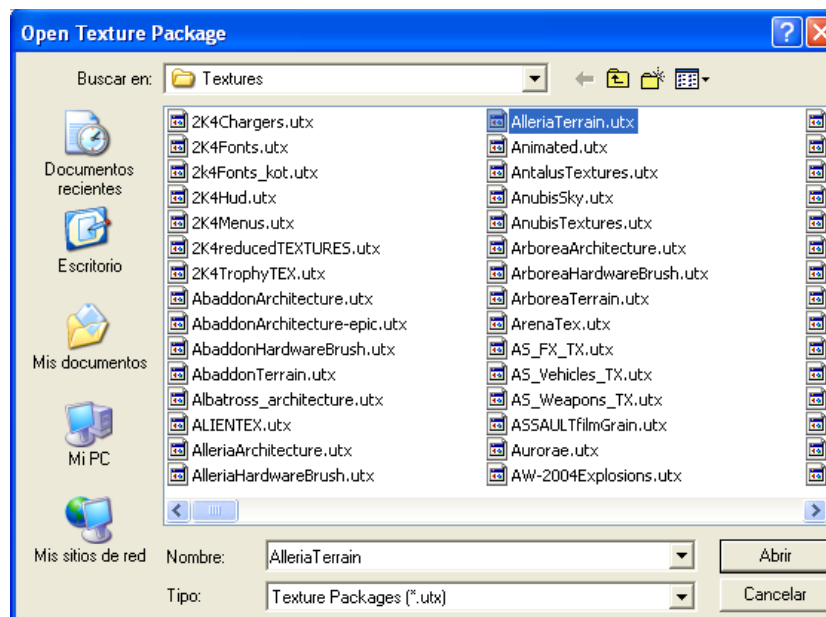


Ilustración 112 Elegir el paquete AlleriaTerrain.utx

3º Elegir el grupo de texturas “ground” y dentro de este seleccionar la textura sand02AL. Se ha escogido esta para la primera capa porque se aproxima bastante a las imágenes del terreno marciano que se presentaron al comienzo de esta sección.

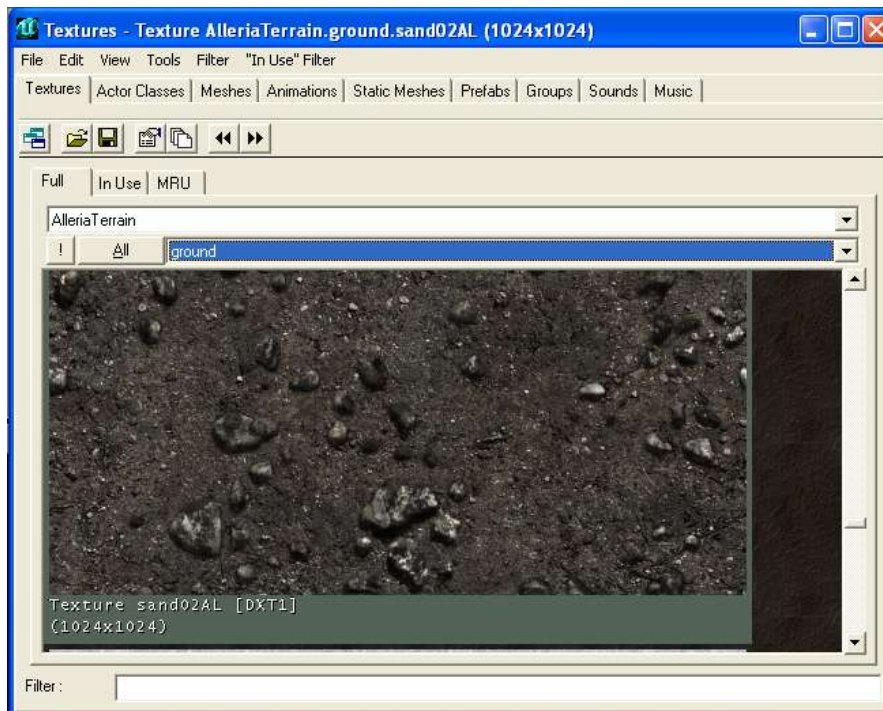


Ilustración 113 Textura elegida para la capa base del terreno

4º Una vez minimizado el buscador de texturas, situarse en la pestaña “Layers” del editor de terreno.

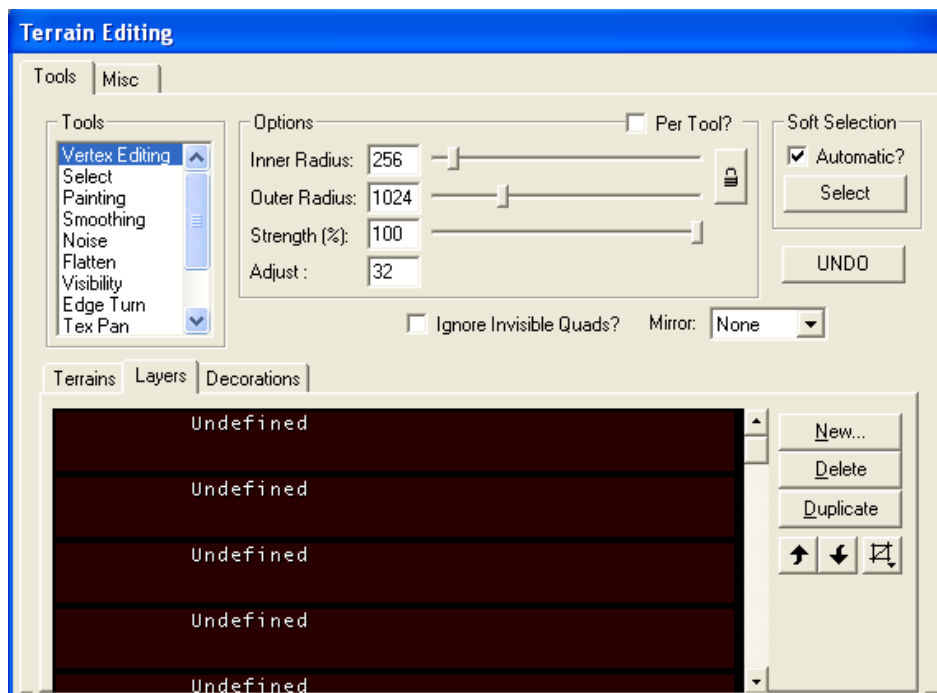


Ilustración 114 Definir las capas que tendrá el terreno



5º Clic sobre la primera capa que aparece como “Undefined” y pulsar sobre el botón “New”. Aparece un cuadro de diálogo como el que se muestra más abajo con las opciones para crear dicha capa:

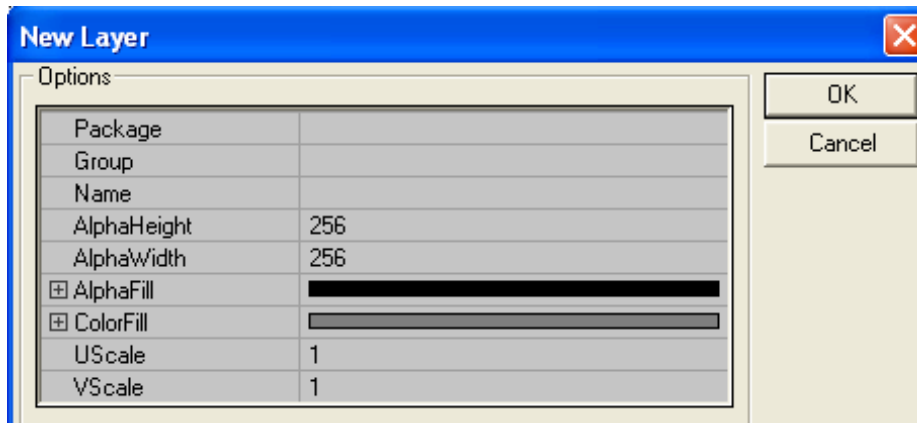


Ilustración 115 Crear una nueva capa para el terreno

- Package: Dado que la información del terreno se guardó en el paquete myLevel almacenar la información correspondiente a esta capa también en este.
- Name: Elegir como nombre de esta capa mars_terrain_sand_7.
- AlphaFill: Fija el canal alfa de la capa, es decir, su matiz y opacidad. Un valor completamente negro indica que esta capa es invisible, mientras que un valor completamente blanco indica que es totalmente visible. Dado que se trata de la primera capa del terreno (capa base) interesa que sea vea bastante opaca. Para ello se ha elegido la siguiente configuración:

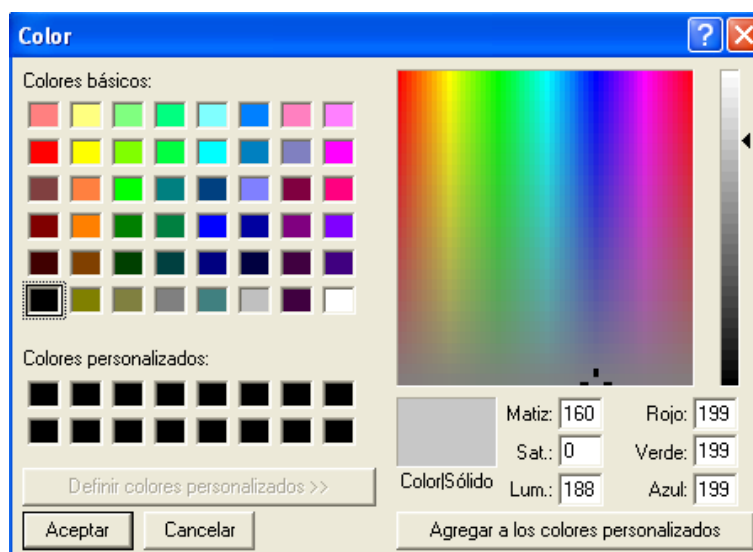


Ilustración 116 “AlphaFill” para la primera capa del terreno

- ColorFill: Color usado para dibujar. Si se elige un valor totalmente blanco (R = 255, G = 255, B = 255) significa que se dibuja todo en su color por defecto. Como la tonalidad de la textura elegida es muy oscura en este caso conviene aclararla.

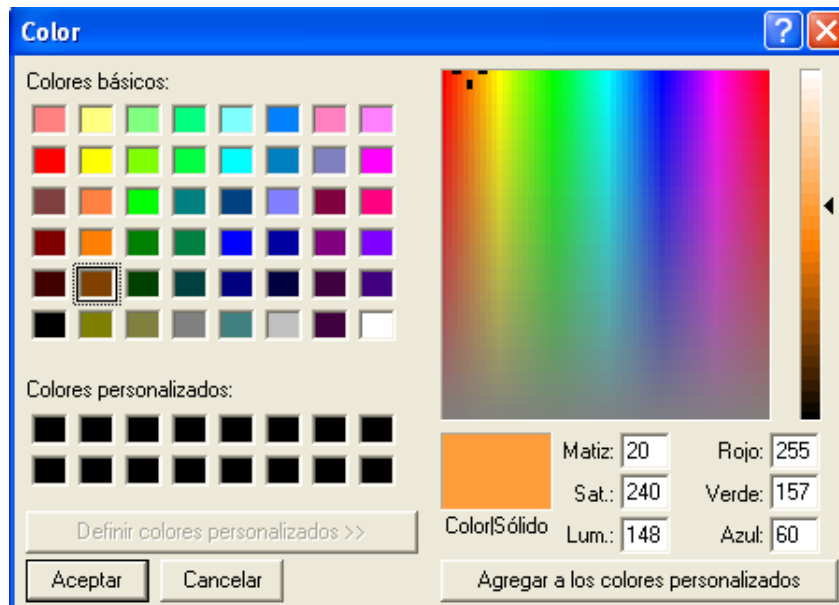


Ilustración 117 “ColorFill” para la primera capa del terreno

6º Una vez establecida la configuración pulsar sobre el botón “OK” para que se cree la nueva capa.

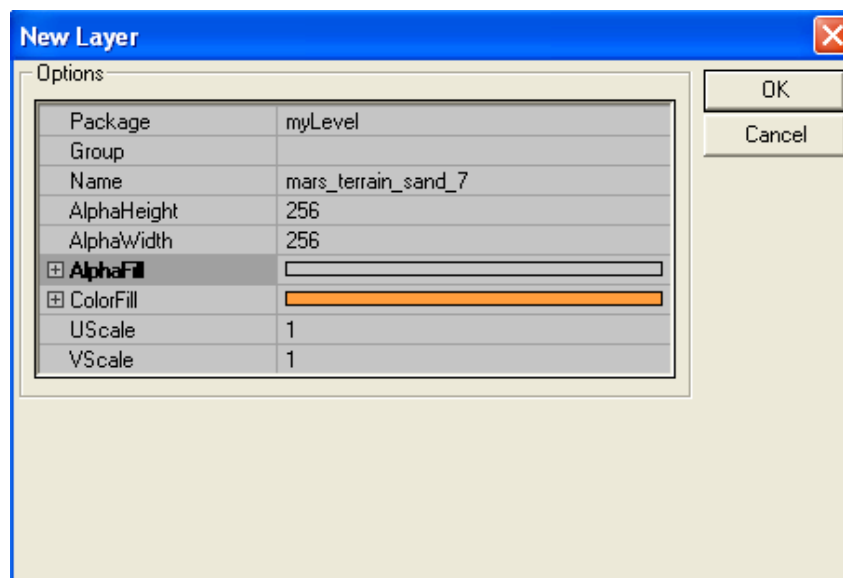



Ilustración 118 Configuración elegida para la primera capa del terreno



7º Maximizar el buscador de texturas y hacer clic en Open Package . Seleccionar el paquete AbaddonTerrain.utx.

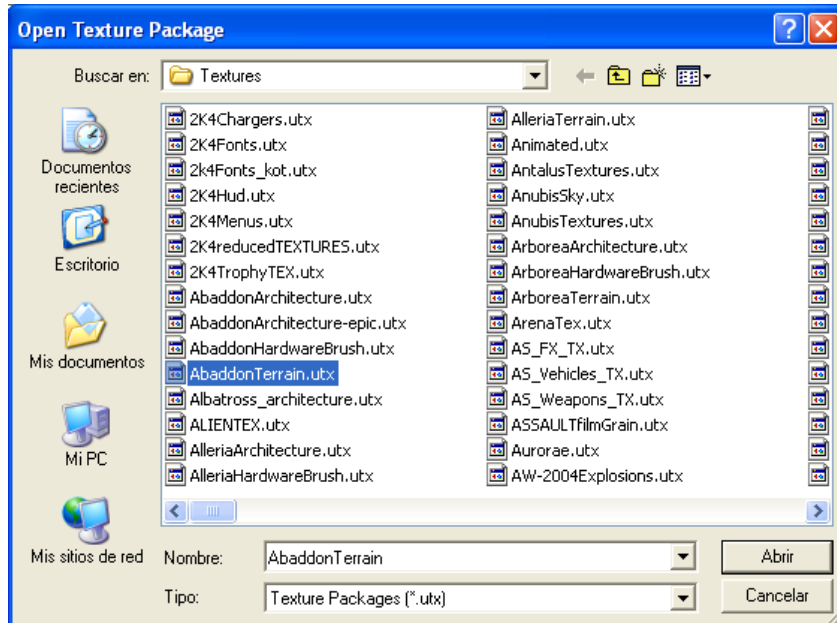


Ilustración 119 Elegir el paquete AbaddonTerrain.utx

8º Elegir el grupo de texturas “ground” y dentro de este seleccionar la textura dirt02GO. Se ha escogido esta para la segunda capa porque se aproxima bastante a las imágenes del terreno marciano que se presentaron al comienzo de esta sección.

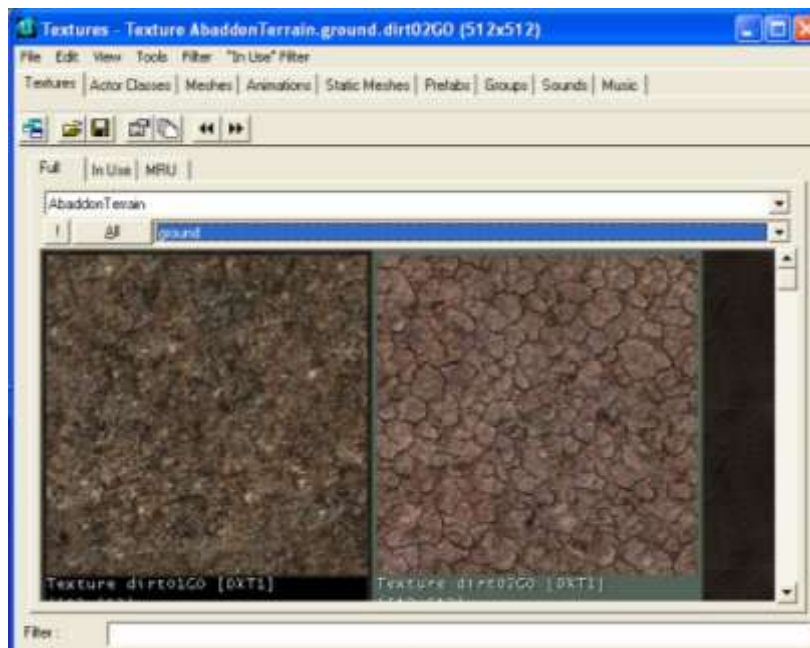


Ilustración 120 Textura elegida para la segunda capa del terreno



9º Una vez minimizado el buscador de texturas, situarse de nuevo en la pestaña “Layers” del editor de terreno.

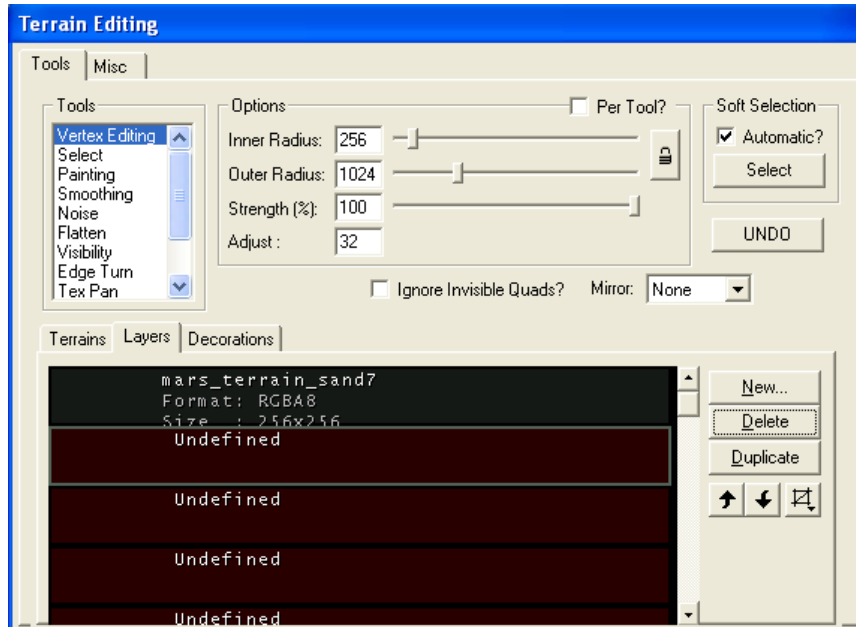


Ilustración 121 Definir las capas que tendrá el terreno

10º Clic sobre la segunda capa que aparece como “Undefined” y pulsar sobre el botón “New”. Aparece un cuadro de diálogo como el que se muestra más abajo con las opciones para crear dicha capa:

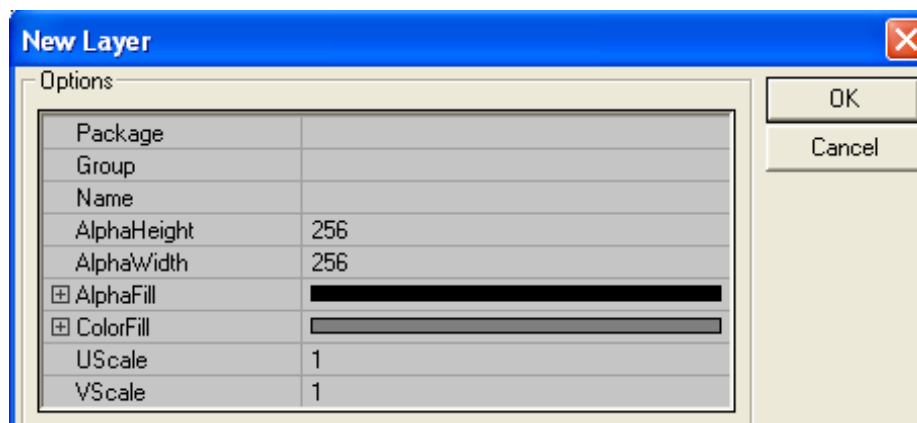


Ilustración 122 Crear una nueva capa para el terreno

- Package: Dado que el terreno se guardó en el paquete myLevel almacenar esta segunda capa también en este.
- Name: Elegir como nombre de esta capa mars_terrain_dirt_1.

- AlphaFill: Como ya se comentó, esta opción fija el canal alfa de la capa, es decir, su matiz y opacidad. Dado que se trata de la segunda capa del terreno interesa que sea un poco menos visible que la primera, de modo que se aprecie levemente. Para ello se ha elegido la siguiente configuración:

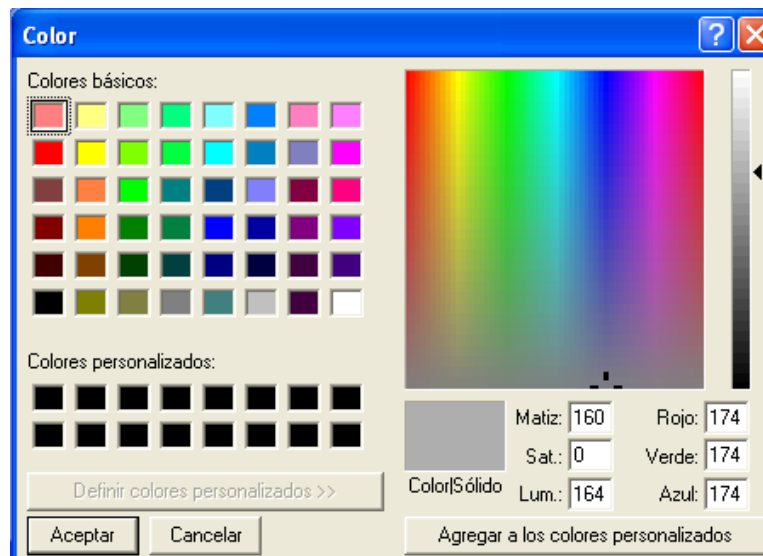


Ilustración 123 “AlphaFill” para la segunda capa del terreno

- ColorFill: Color usado para dibujar. Si se elige un valor totalmente blanco (R = 255, G = 255, B = 255) significa que se dibuja todo en su color por defecto.

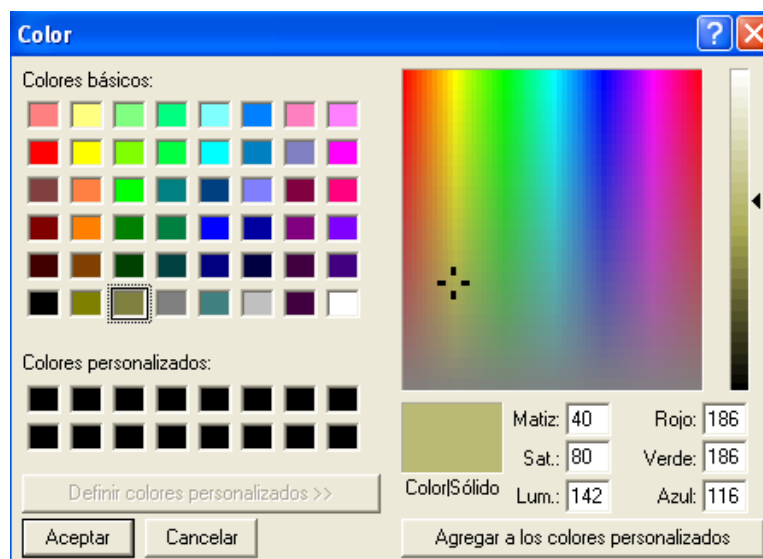


Ilustración 124 “ColorFill” para la segunda capa del terreno



11º Una vez establecida la configuración pulsar sobre el botón “OK” para que se cree la nueva capa.

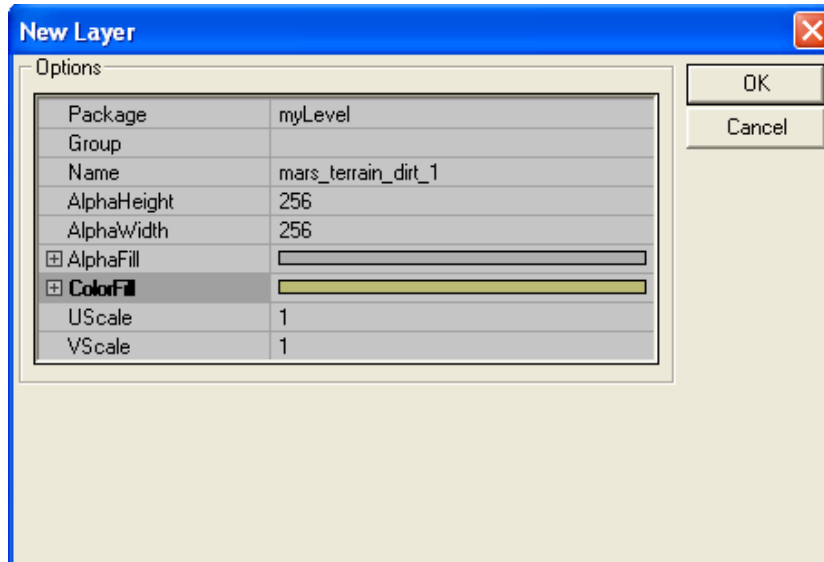


Ilustración 125 Configuración elegida para la segunda capa del terreno

12º Después de haberse creado las dos capas, este es el aspecto que presenta el terreno en la vista 3D.



Ilustración 126 Terreno marciano desde la vista en 3D

13º Finalmente regresar al mapa y en el Top viewport hacer clic derecho sobre el icono correspondiente al actor “TerrainInfo” para acceder a sus propiedades. En el desplegable correspondiente a “TerrainInfo” seleccionar la siguiente configuración para “TerrainScale” y “TerrainSectorSize”.

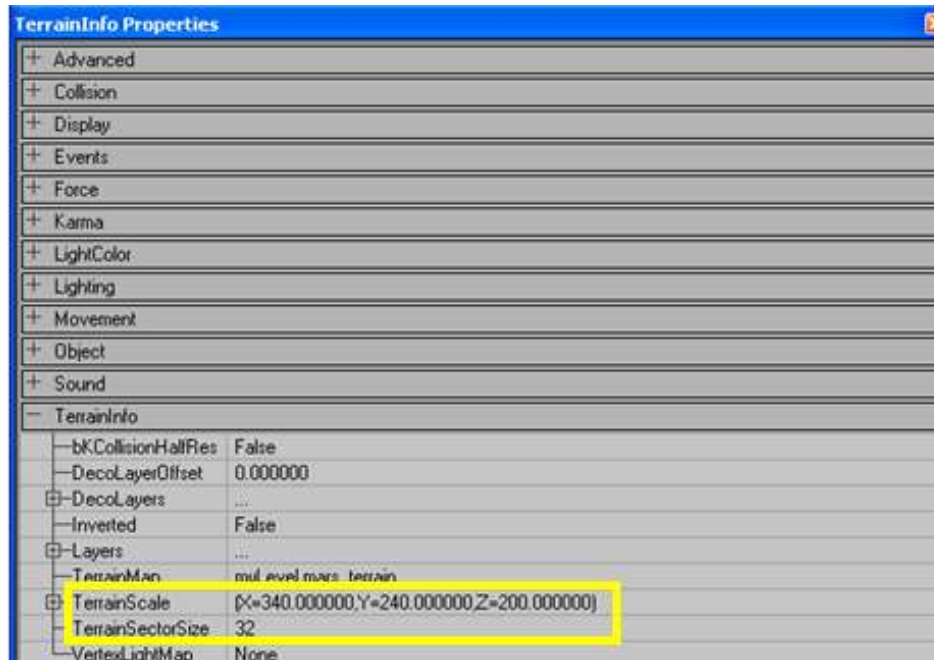


Ilustración 127 Configuración elegida para "TerrainScale" y TerrainSectorSize"

Para añadir algunas montañas al terreno marciano se utilizaron “*Static Mesh*”. A continuación se explica cómo se procedió:

1º Clic izquierdo sobre “Static Mesh Browser” .

Aparece una ventana como la siguiente:

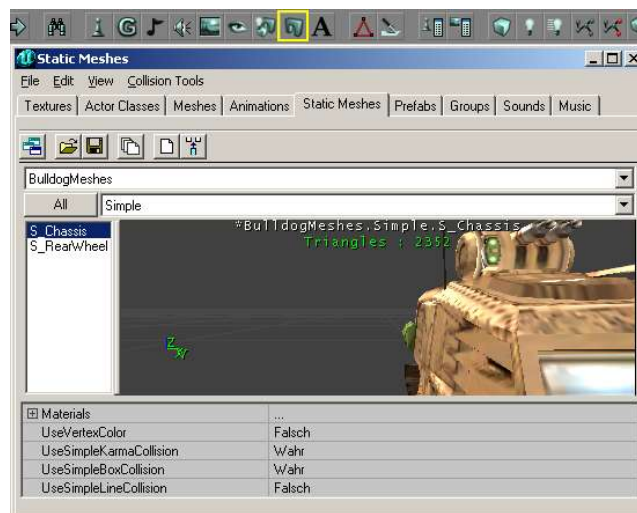



Ilustración 128 Buscador de Static Meshes

2º Clic en Open Package . Seleccionar el paquete SC_Volcano_M.usx.

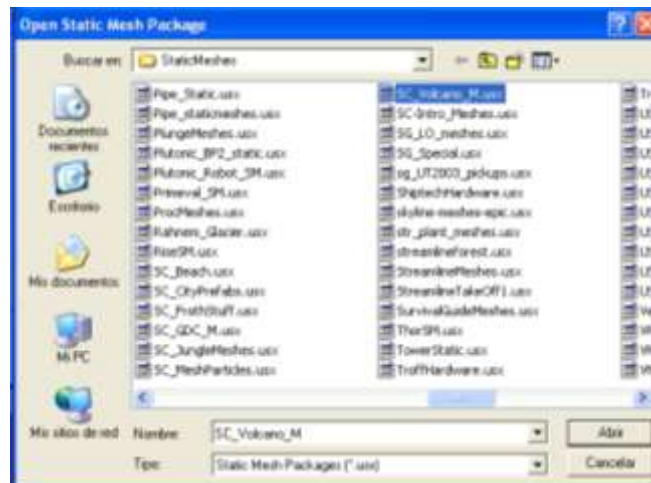


Ilustración 129 Seleccionar el paquete SC_Volcano_M.usx

3° En la caja que aparece a la izquierda recuadrada en amarillo seleccionar Mountain1.

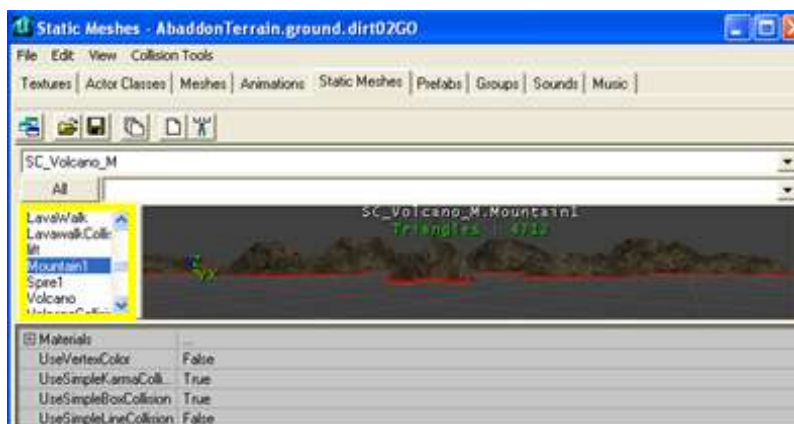



Ilustración 130 Seleccionar el "Static Mesh" Mountain1

4° Clic izquierdo sobre el botón “Insert Static Mesh Into Level”  para añadir el “Static Mesh” al mapa. El resultado de esta acción se puede observar en la siguiente imagen:

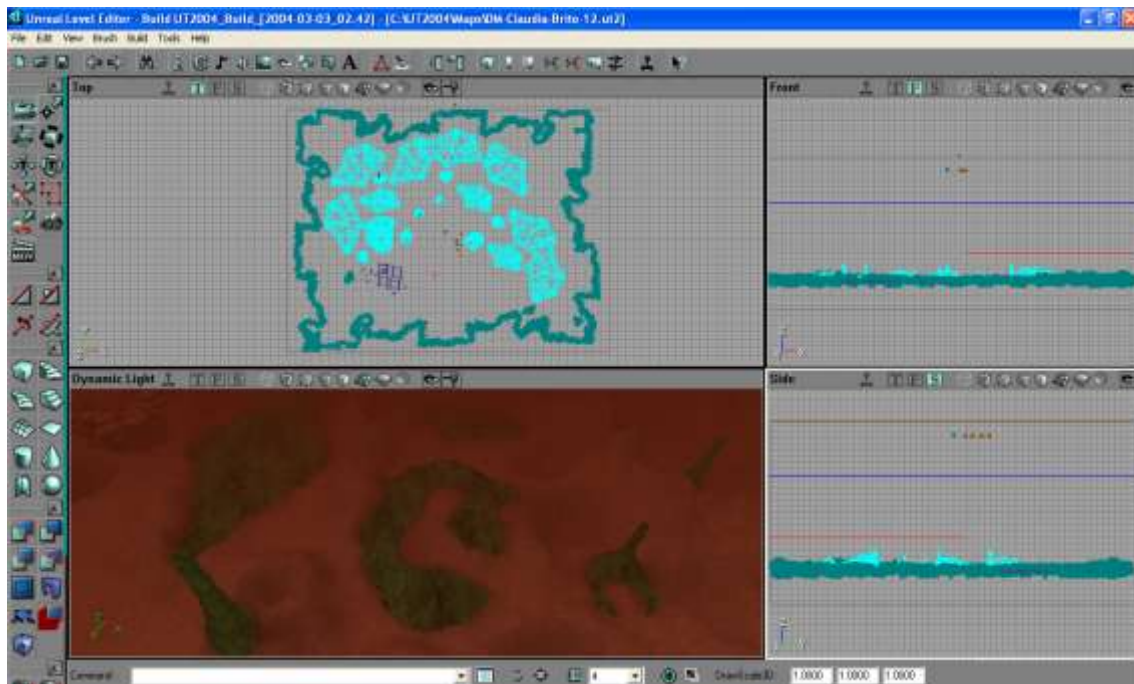


Ilustración 131 "Static Mesh" que representa a un conjunto de montañas

5° Para que las montañas reciban los haces de luz procedentes de los rayos solares es preciso acceder a las propiedades del "Static Mesh" haciendo clic derecho sobre su icono. Una vez abierta la ventana de propiedades se seleccionará el desplegable Lighting y se establecerá la opción bSpecialLit a True.

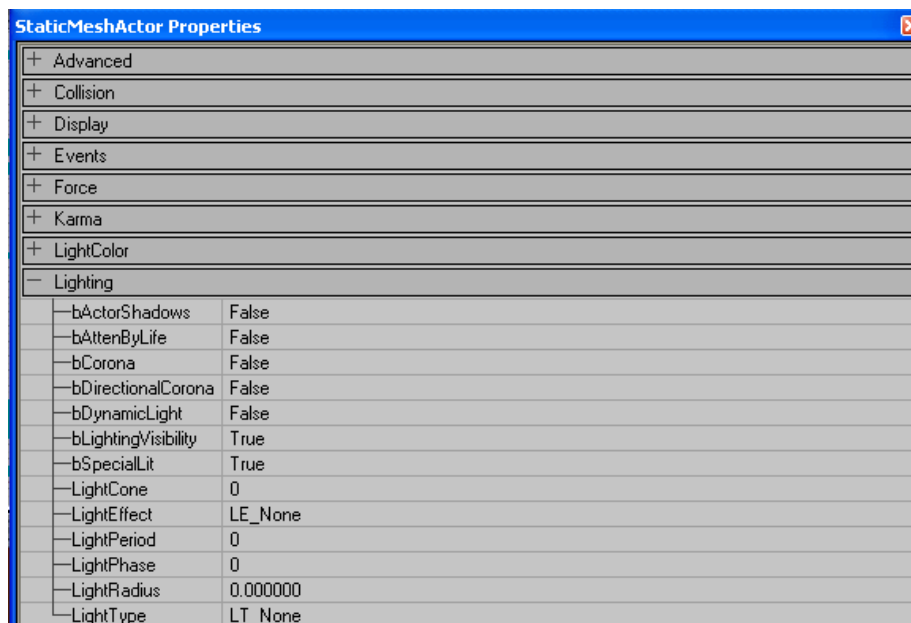


Ilustración 132 Propiedades del "Static Mesh"



2.2.1.5 Construcción de rampas

Con el objetivo de comprobar que el rover es capaz de ascender/descender por terreno que presenta distintas inclinaciones se han introducido en el mapa diversas rampas que cuentan con diferentes pendientes.

Para la realización de las rampas se ha utilizado una técnica denominada “*Brush Clipped*” y se ha procedido del siguiente modo:

1º Clic derecho sobre “Cube”



Aparece un cuadro de diálogo como el que se muestra más abajo con las propiedades del “Cube Builder”. Establecer los siguientes valores:

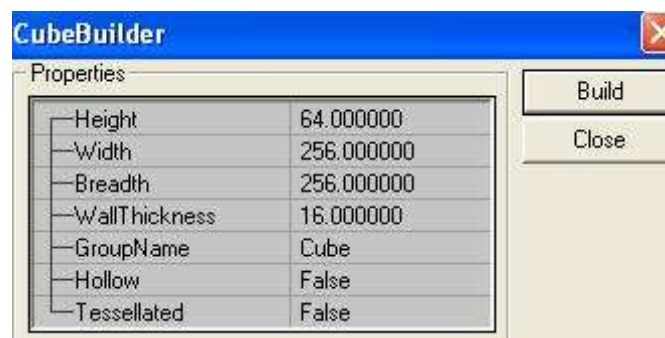


Ilustración 133 Propiedades del CubeBuilder

Seleccionar la herramienta “Brush Clipping”



A continuación, en el Front viewport, se selecciona primero el vértice derecho inferior y se presiona la tecla Ctrl al mismo tiempo que se hace clic izquierdo sobre él. Repetir el mismo proceso para el vértice izquierdo superior. Esto hará que aparezcan dos iconos en forma de marcador que indican dónde comienza (1) y dónde termina (2) la “línea de clipping”. La línea perpendicular que aparece en medio de los dos marcadores indica qué lado del brush será cortado, en este caso, el lado superior.

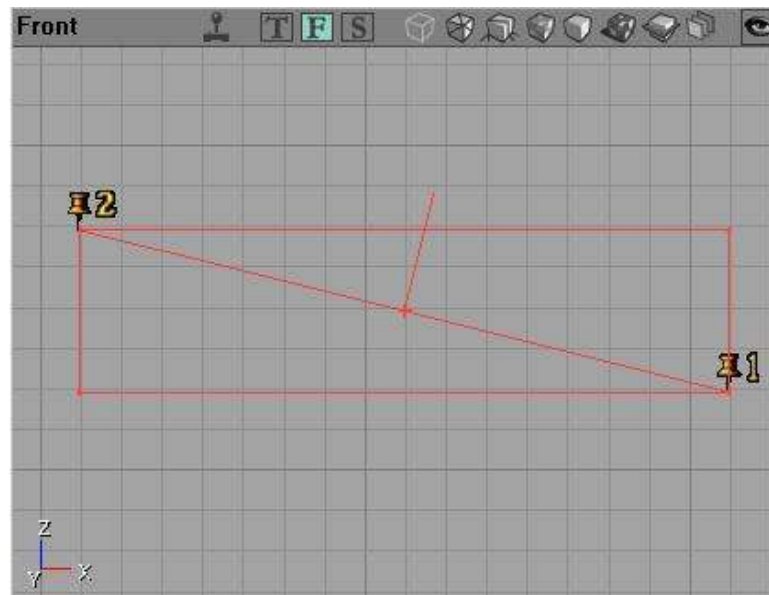



Ilustración 134 Seleccionar los vértices

Una vez se han seleccionado los vértices y la línea perpendicular está apuntando hacia el lado que se quiere cortar se selecciona el botón “Clip Selected Brushes” .

Con esto se consigue suprimir definitivamente la mitad superior, quedando la rampa como se muestra más abajo.

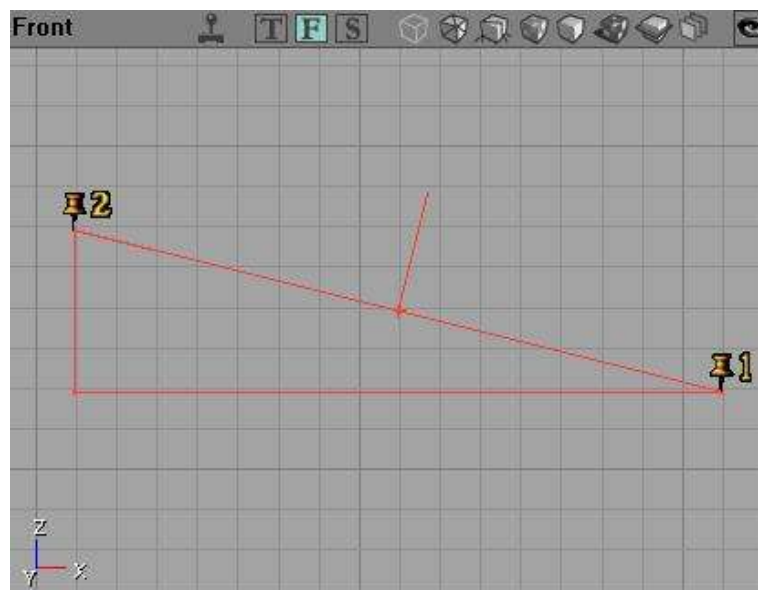



Ilustración 135 Brush que servirá de molde para crear la rampa



A continuación se elige una textura para la rampa, en este caso se ha elegido una que está dentro del paquete UTtech1.

1º Clic izquierdo sobre “Texture Browser” 

Aparece un cuadro de diálogo como el siguiente:

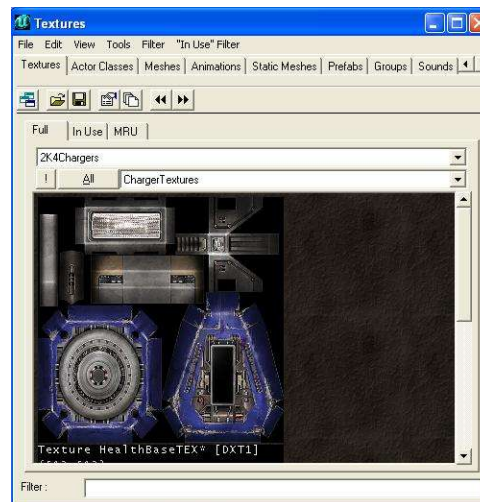



Ilustración 136 Buscador de texturas

2º Después se hace clic en “Open Package”  , se elige el paquete “UTTech1” y en el desplegable que aparece abajo recuadrado en rojo se selecciona el grupo “Floor”. Dentro de este grupo se ha escogido la textura hashlow-cont*.

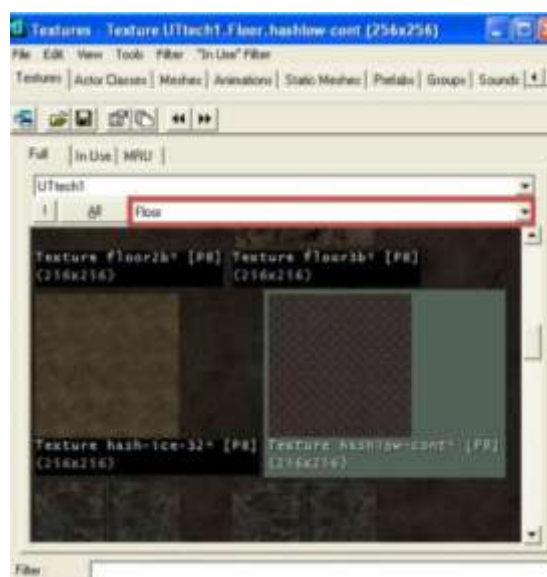


Ilustración 137 Textura elegida para la rampa



2.2.1.6 Creación del cielo marciano

Requerimientos:


- Color rojizo-anaranjado

Procedimiento:

Para su construcción se ha empleado una técnica denominada “SkyBox” que permite crear un cielo que se percibe como lejano en el mapa. Es muy habitual usarla en mapas de exteriores, como el que se ha diseñado en este proyecto, debido a que le otorga un mayor realismo. Aunque se puede utilizar casi cualquier forma para construirlo, normalmente se emplea una caja porque de este modo resulta más sencillo alinear las texturas usadas en sus seis caras.

Las texturas empleadas son cinco imágenes JPG de dimensiones 512 x 512 que se han obtenido de la siguiente URL <http://www.redsorceress.com/skybox.html>, en concreto se ha elegido el grupo denominado Mars 1. Para poder utilizarlas en el Unreal Level Editor se han tenido que convertir previamente mediante la herramienta GIMP 2.6 a formato BMP de 24 bits (R8 G8 B8) con el que trabaja el editor.

Una vez completado este paso se han importado cada una de las texturas para poder usarlas dentro del Unreal Level Editor. Seguidamente se explica cómo se ha procedido:

1º Clic izquierdo sobre “Texture Browser” 

Aparece un cuadro de diálogo como el siguiente:

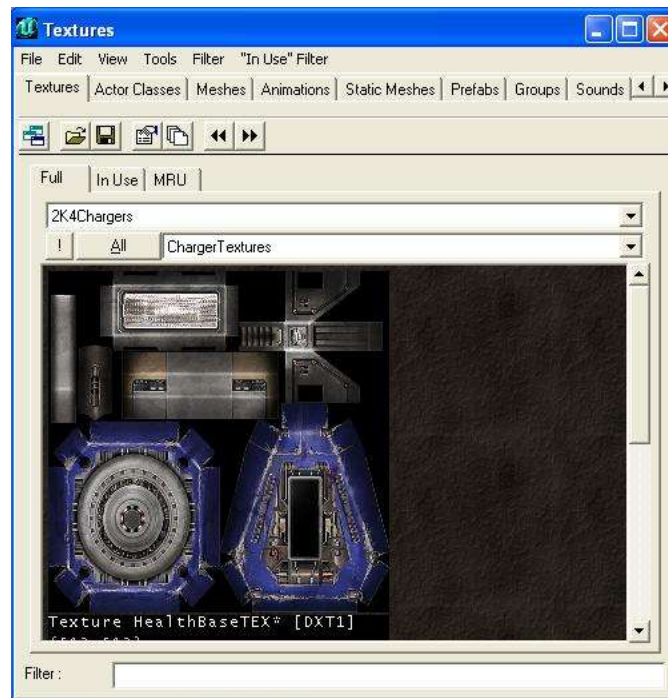


Ilustración 138 Buscador de texturas

2º Ir a File > Import

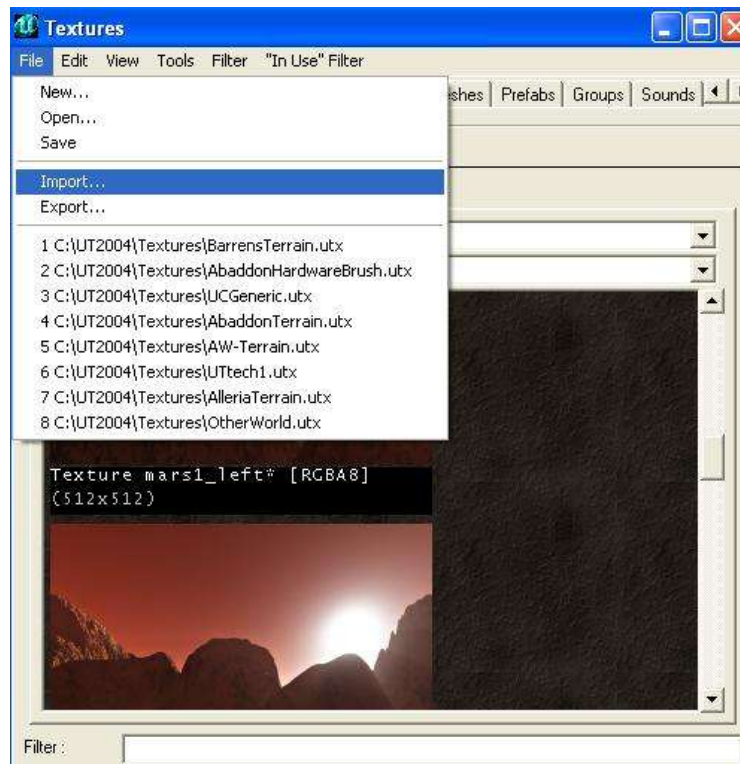


Ilustración 139 Importar fichero para utilizar las texturas dentro del editor



3º Aparece un nuevo cuadro de diálogo donde se busca la carpeta en la que se encuentran las texturas ya convertidas a formato BMP. Se selecciona la textura deseada y se le da a “Abrir”.

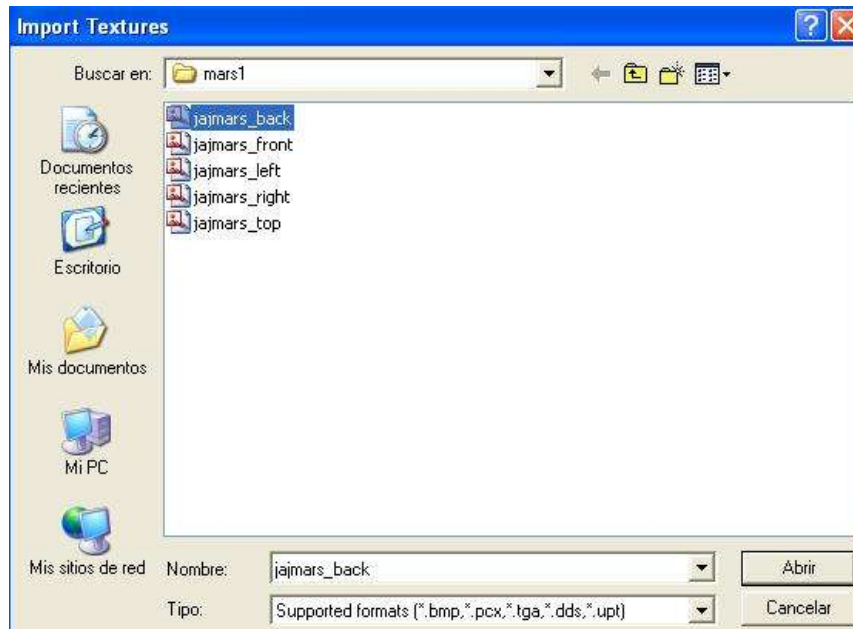


Ilustración 140 Se selecciona la textura que se quiere importar

4º A continuación se muestra el cuadro de diálogo que aparece y en el que se han configurado las siguientes opciones:

- **Package:** nombre del paquete en el que se almacenará la textura. En este caso se ha elegido el paquete myLevel para que la textura sea almacenada en el mapa actual.
- **Name:** nombre de la textura
- **Generate MipMaps:** Marcar esta opción para que se genere también la versión reducida de la textura. Esto es útil cuando se tiene una textura grande y el área de la pantalla a la que se va a aplicar es pequeña para evitar que se produzcan destellos.

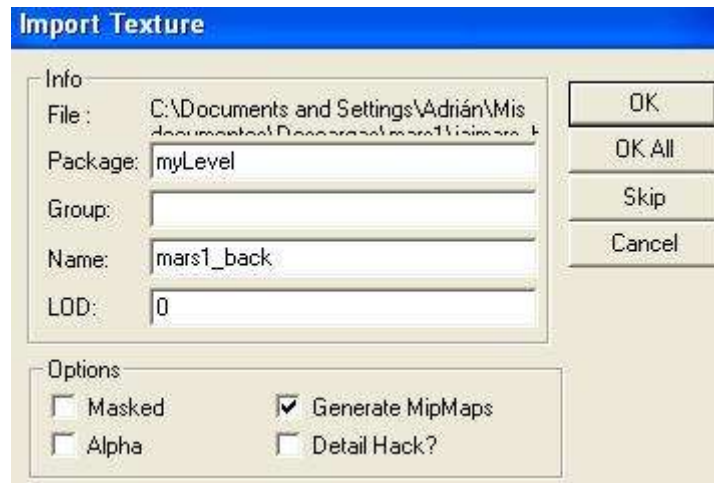


Ilustración 141 Importar la textura al mapa actual

5º Finalmente pulsar sobre “Ok”.

Después de haber repetido los pasos del 2º al 5º para cada una de las cinco texturas que se emplean para construir el cielo, este es el resultado que se ha obtenido:

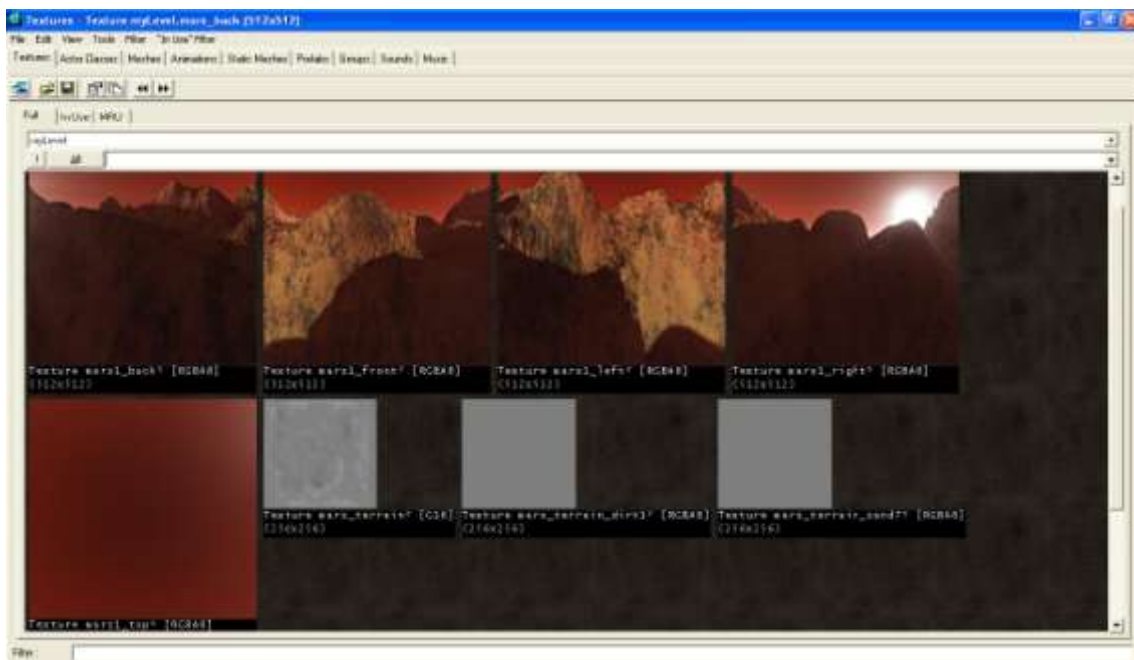



Ilustración 142 Las cinco texturas ya están disponibles en el editor

Una vez se encuentran almacenadas dentro del paquete “myLevel” ya están listas para ser usadas. Ahora el siguiente paso es construir la caja o cubo sobre la que se aplicarán dichas texturas. Para ello se ha procedido tal y como se detalla a continuación:



1º Clic derecho sobre “Cube” 

Aparece un cuadro de diálogo como el que se muestra más abajo con las propiedades del “Cube Builder”.

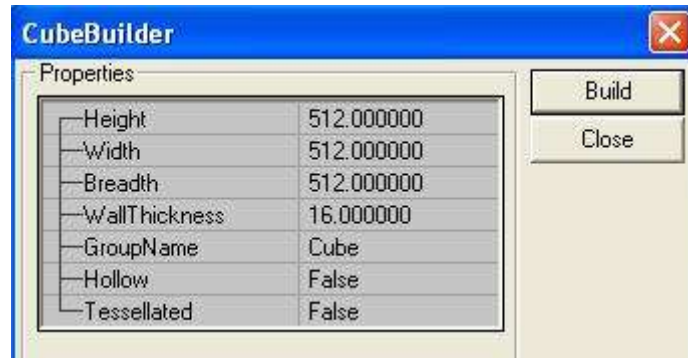


Ilustración 143 Propiedades del CubeBuilder

Se han seleccionado estos valores para el cubo precisamente porque las texturas elegidas tienen unas dimensiones de 512 x 512 de modo que encajan perfectamente en las caras del cubo y no es necesario alinearlas.

Seguidamente se ha pulsado sobre “Build”

Ahora aparece dibujado en los cuatro viewports (Top, Front, Side, 3D) el brush con forma de cubo de alambre rojo que servirá de molde para construir el cielo marciano. Dado que se trata del cielo conviene ubicarlo, tanto en el Top como en el Front viewport, justo en la zona central y encima del brush violeta que representa el mapa, tal y como se muestra en la ilustración siguiente:

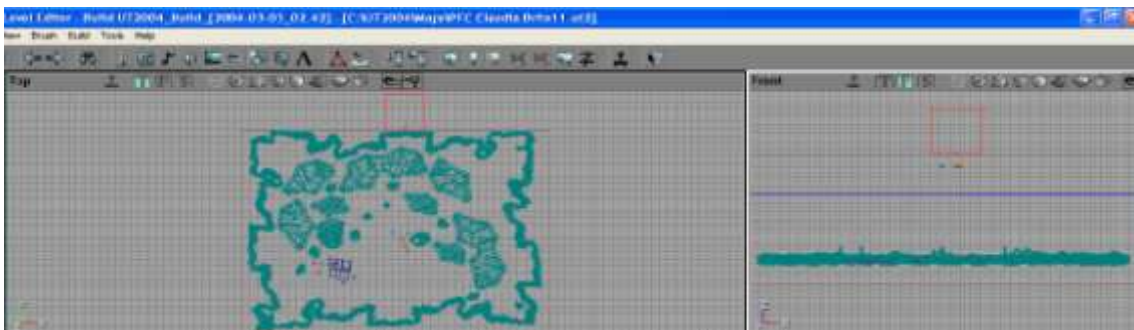


Ilustración 144 Brush rojo para la construcción del cielo marciano

2º A continuación hacer clic izquierdo sobre “Texture Browser” 



Aparece un cuadro de diálogo como el siguiente:

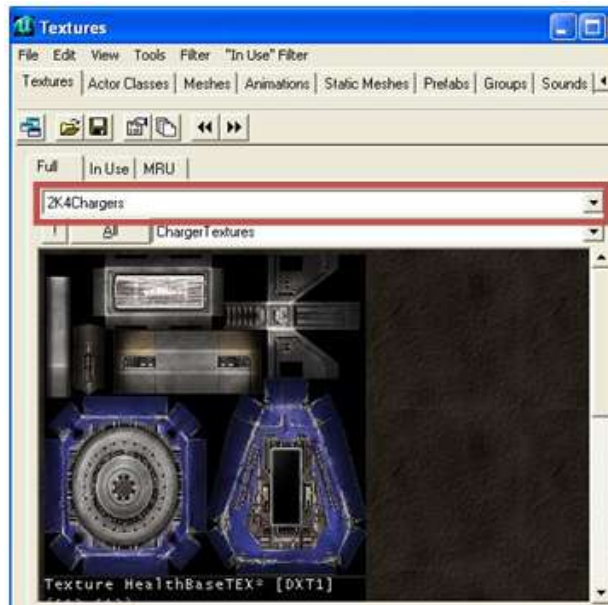


Ilustración 145 Buscador de texturas

Seleccionar en el desplegable que aparece recuadrado en rojo en la ilustración anterior el paquete myLevel y dentro de este elegir la textura mars1_back, que se empleará para la cara trasera del cubo.

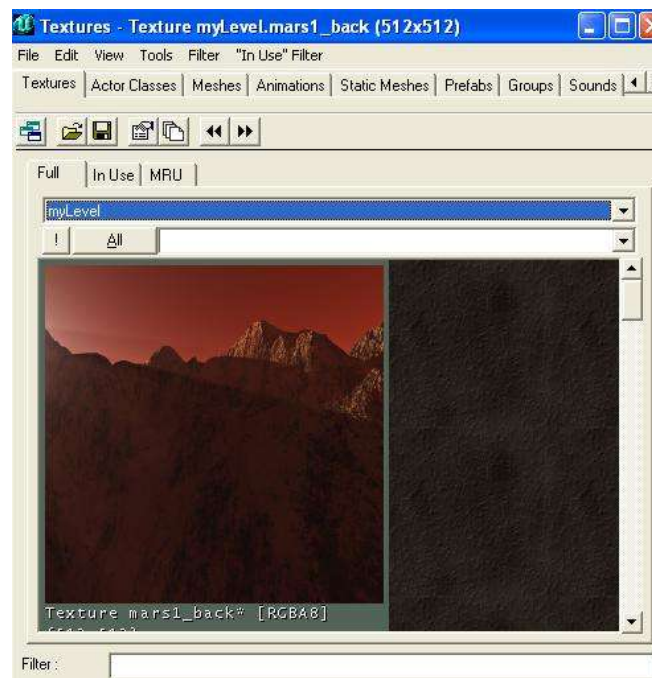


Ilustración 146 Textura mars1_back* del paquete "myLevel"



3º Posteriormente se ha minimizado el cuadro de diálogo y se ha hecho clic izquierdo en

Subtract 

Ahora todo el cubo de alambre rojo aparece con la textura que se ha elegido previamente. Para ilustrar este paso se muestra cómo se ve en el 3D viewport:

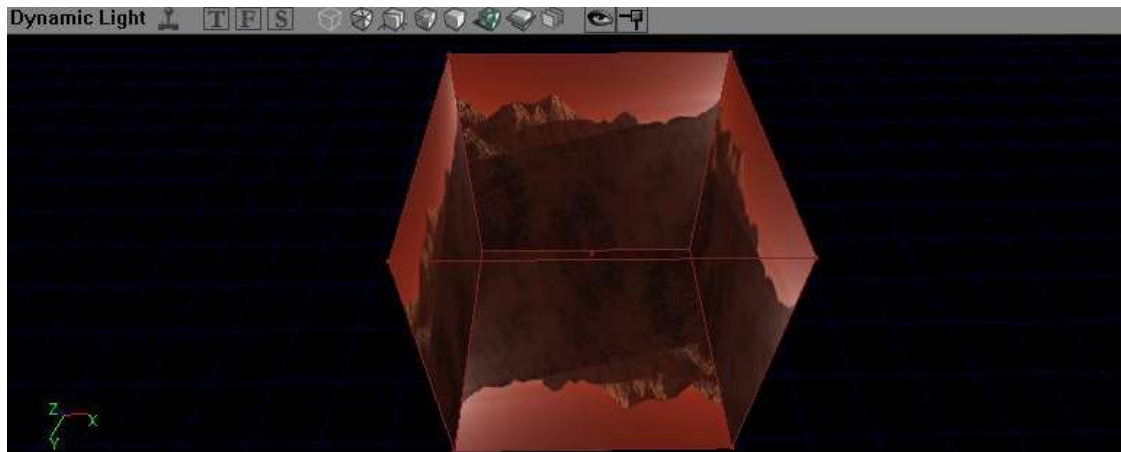


Ilustración 147 3D viewport - Brush rojo después de haber hecho "Subtract"

Pero esto no es lo que se pretende, sino más bien aplicar a cada cara del cubo una textura diferente para crear la composición del cielo, tal y como se detalla en las ilustraciones siguientes:

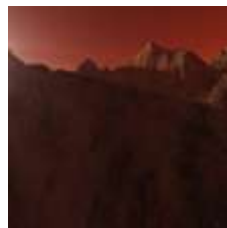


Ilustración 148 Textura aplicada a la cara trasera del cubo



Ilustración 149 Textura aplicada a la cara delantera del cubo



Ilustración 150 Textura aplicada a la cara izquierda del cubo



Ilustración 151 Textura aplicada a la cara derecha del cubo

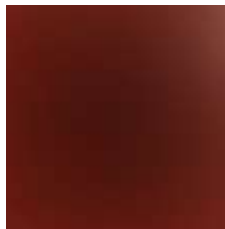


Ilustración 152 Textura aplicada a la cara superior del cubo

A continuación maximizamos el cuadro de diálogo correspondiente al buscador de texturas y seleccionamos ahora mars1_front para la cara delantera del cubo.

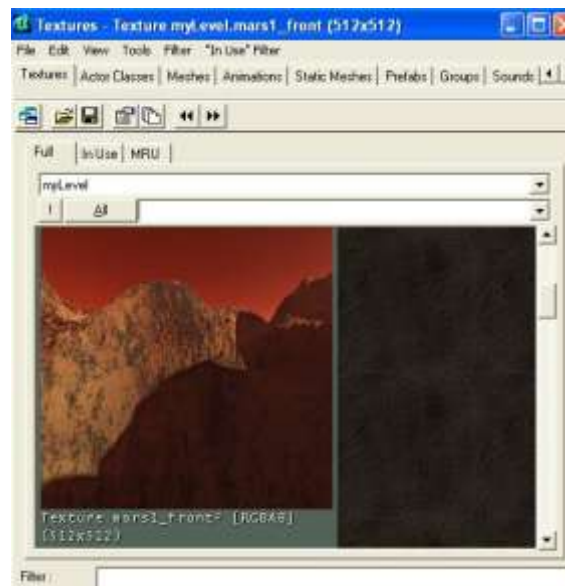


Ilustración 153 Textura mars1_front* del paquete "myLevel"



Una vez seleccionada la textura se minimiza el cuadro de diálogo y se hace clic derecho sobre la cara delantera del cubo para acceder a las opciones. Seleccionamos la opción “Apply Texture: myLevel.mars1_front” para que la textura previamente elegida se aplique.

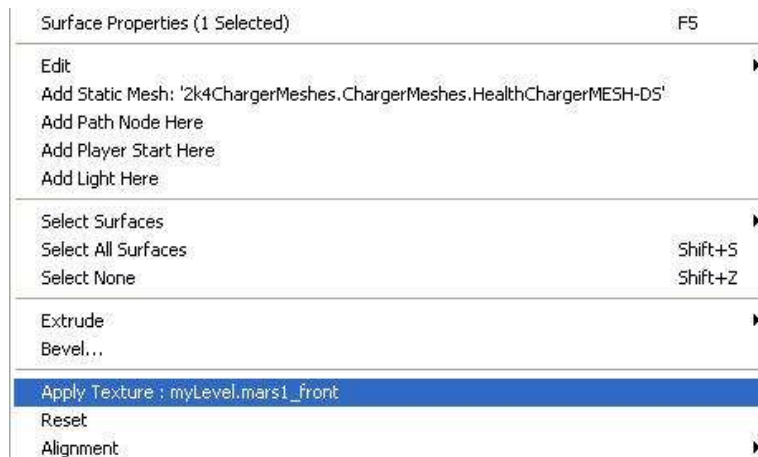


Ilustración 154 Aplicar textura mars1_front

Posteriormente volvemos a maximizar el cuadro de diálogo correspondiente al buscador de texturas y ahora seleccionamos mars1_left para la cara izquierda del cubo.

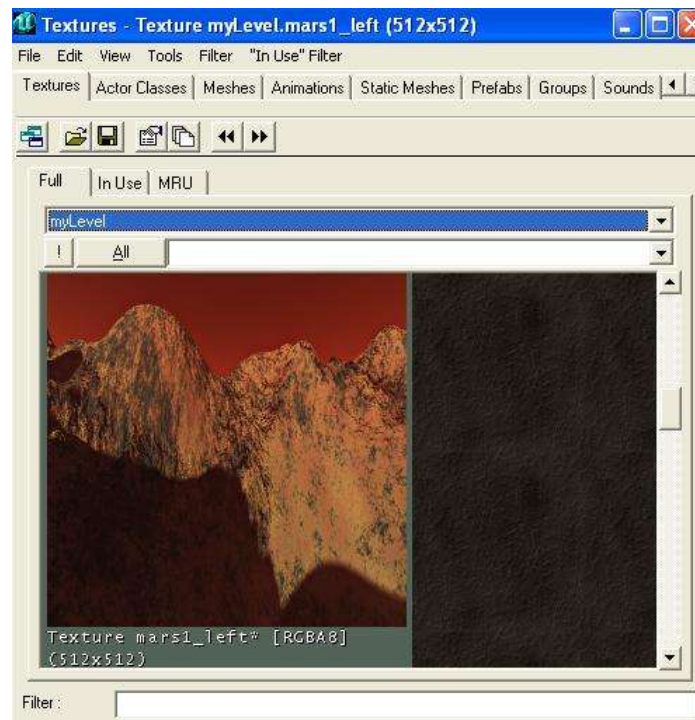


Ilustración 155 Textura mars1_left* del paquete “myLevel”



Una vez seleccionada la textura se minimiza el cuadro de diálogo y se hace clic derecho sobre la cara izquierda del cubo para acceder a las opciones. Seleccionamos la opción “Apply Texture: myLevel.mars1_left” para que la textura previamente elegida se aplique.

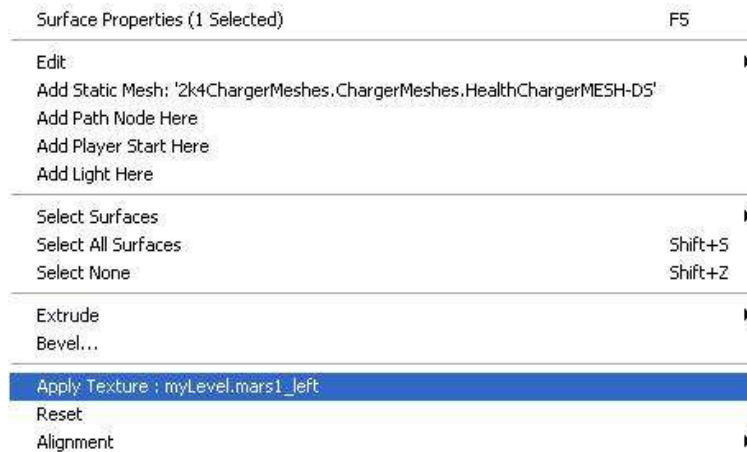


Ilustración 156 Aplicar textura mars1_left

Una vez más volvemos a maximizar el cuadro de diálogo correspondiente al buscador de texturas y ahora seleccionamos mars1_right para la cara derecha del cubo.

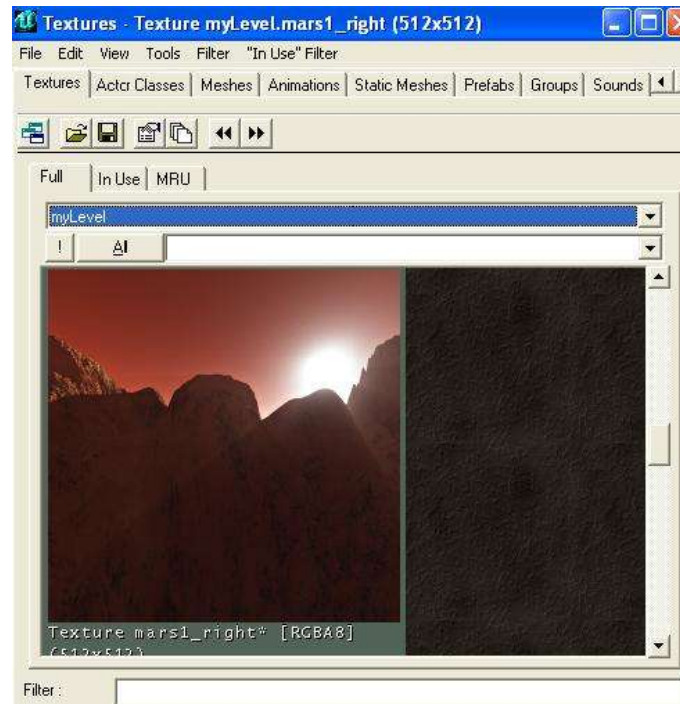


Ilustración 157 Textura mars1_right*



Una vez seleccionada la textura se minimiza el cuadro de diálogo y se hace clic derecho sobre la cara derecha del cubo para acceder a las opciones. Seleccionamos la opción “Apply Texture: myLevel.mars1_right” para que la textura previamente elegida se aplique.

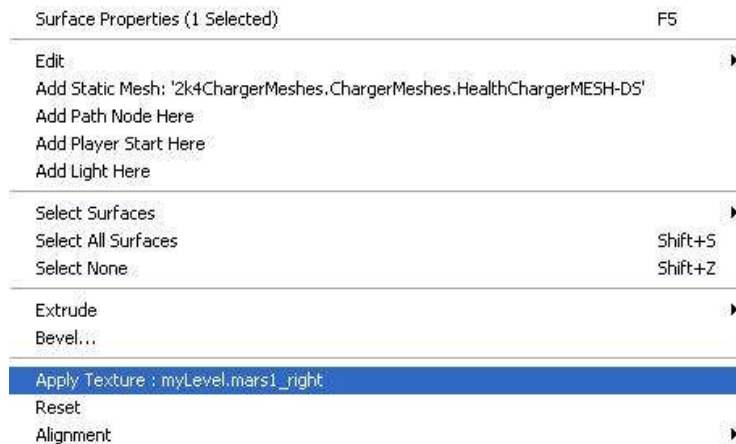


Ilustración 158 Aplicar textura mars1_right

Por última vez volvemos a maximizar el cuadro de diálogo correspondiente al buscador de texturas y ahora seleccionamos mars1_top para la cara superior del cubo.

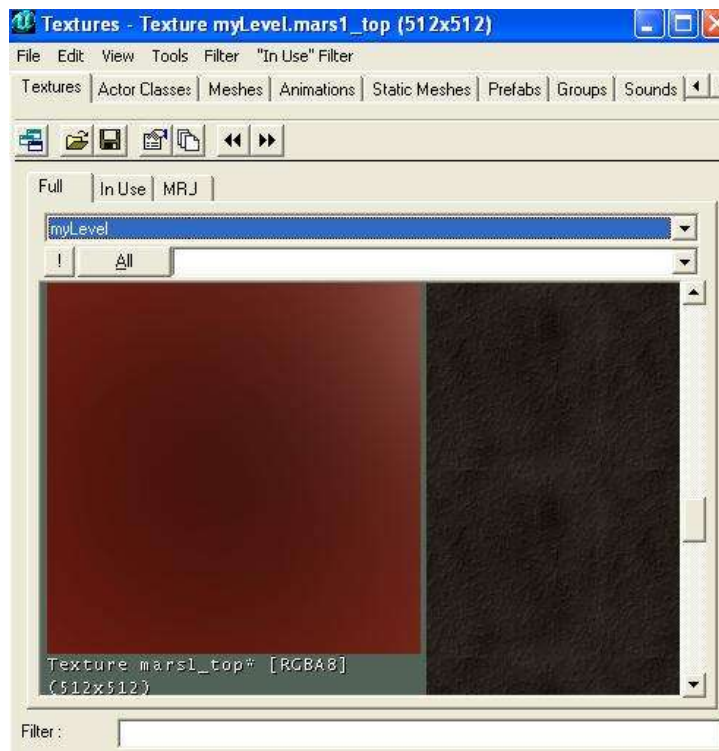


Ilustración 159 Textura mars1_top*



Una vez seleccionada la textura se minimiza el cuadro de diálogo y se hace clic derecho sobre la cara superior del cubo para acceder a las opciones. Seleccionamos la opción “Apply Texture: myLevel.mars1_top” para que la textura previamente elegida se aplique.

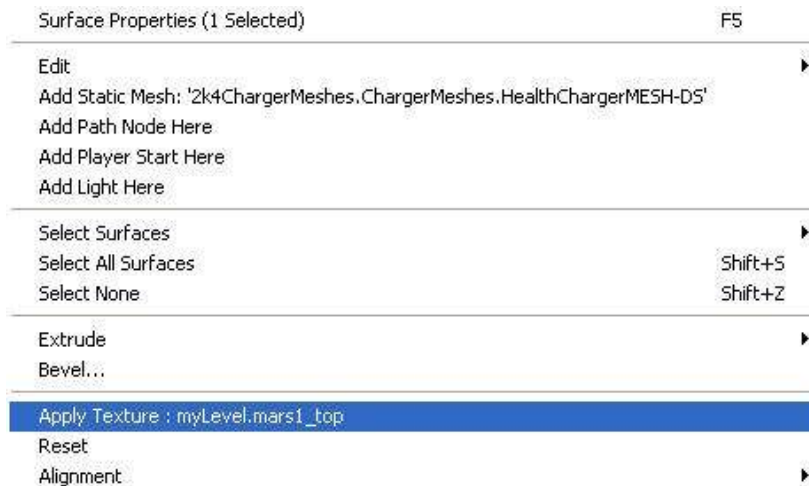


Ilustración 160 Aplicar textura mars1_top

4º El siguiente paso sería separar el molde usado para construir el cielo (cubo de alambre rojo). Para ello seleccionar en el Top viewport el cubo de alambre rojo (clic izquierdo) y manteniendo presionada la tecla Ctrl arrastrar mediante movimiento del ratón el cubo hasta situarlo fuera de su posición original, por ejemplo a la izquierda. Hacer lo mismo en el Front viewport pero en este caso arrastrar el cubo hacia arriba. Después de haber hecho esto, en el Top viewport, Front viewport, Side viewport y 3D viewport ahora aparecen dos cubos: uno de color rojo (molde empleado) y otro de color amarillo que se corresponde al espacio reservado para el cielo.

El resultado correspondiente a estos dos últimos pasos es el que se muestra seguidamente:

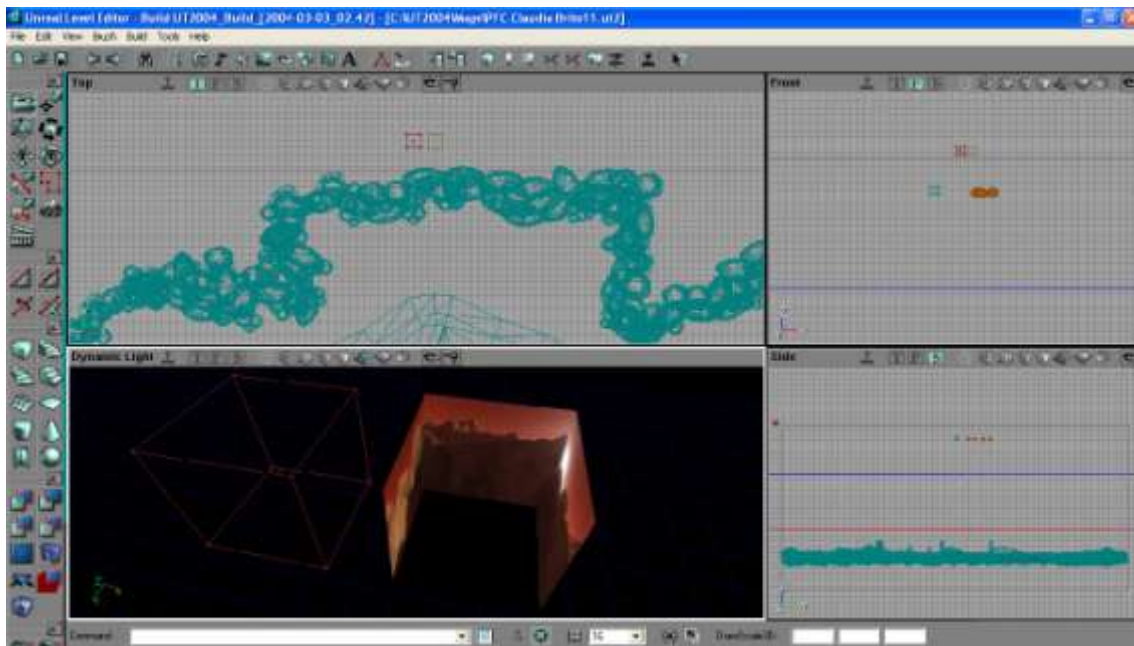



Ilustración 161 Composición de texturas para la creación del cielo marciano

Después de que se hayan aplicado las texturas a las caras del cubo y la composición para el cielo esté terminada es preciso añadir en el centro del cubo un actor de tipo “SkyZoneInfo”. Este actúa como una cámara que muestra lo que ve en cualquier superficie que tenga habilitado el flag “*Fake Backdrop*”.

5º Para crear un actor de este tipo, se ha hecho clic izquierdo sobre “Actor Class Browser” 

Aparece un cuadro de diálogo en el que se selecciona Info > ZoneInfo > SkyZoneInfo y a continuación se minimiza.

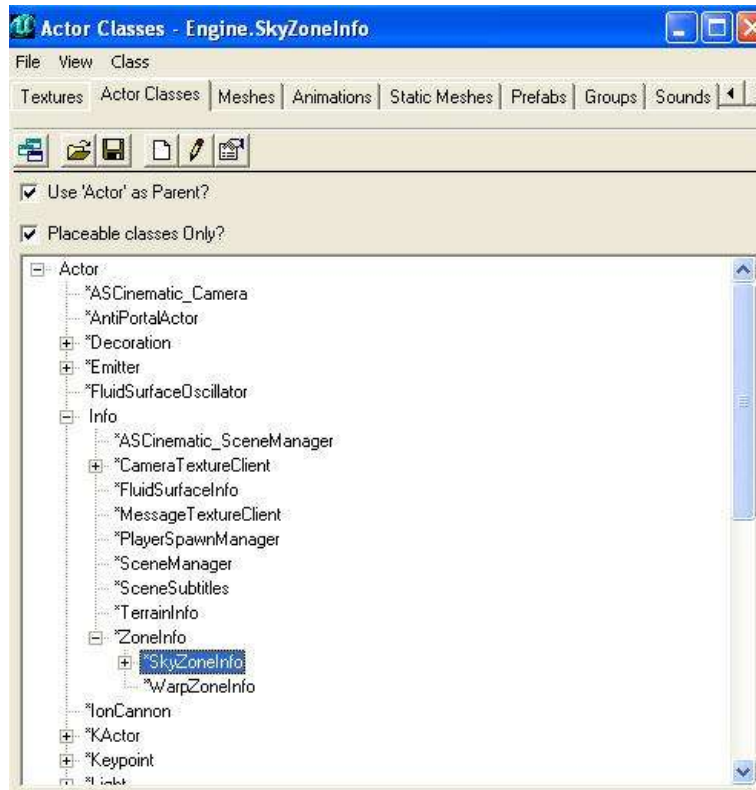


Ilustración 162 Actor SkyZoneInfo

En el Top viewport se hace clic derecho sobre la zona central del cielo y se selecciona “Add SkyZoneInfo Here”.

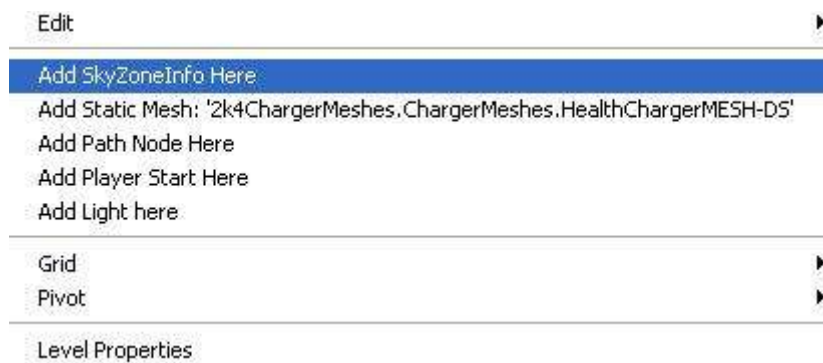


Ilustración 163 Añadir al cielo un actor de tipo SkyZoneInfo

Inmediatamente se añade en el centro del cielo, como puede observarse seguidamente:

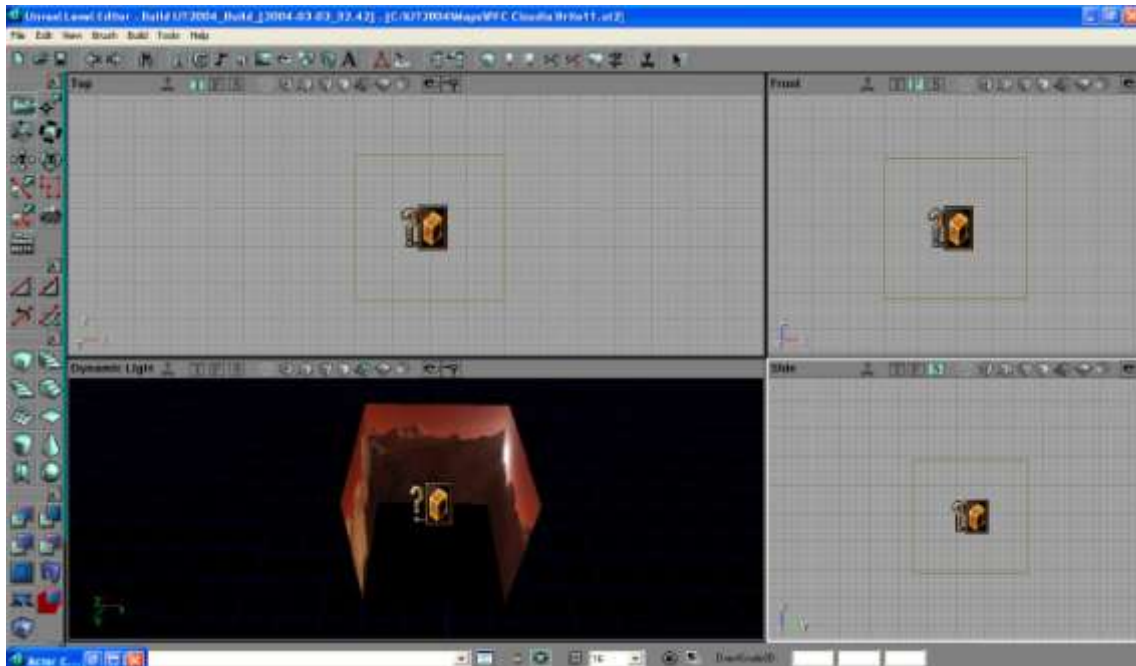


Ilustración 164 SkyBox con SkyZoneInfo

A continuación se hizo clic derecho sobre el icono y se seleccionó “SkyZoneInfo Properties”.

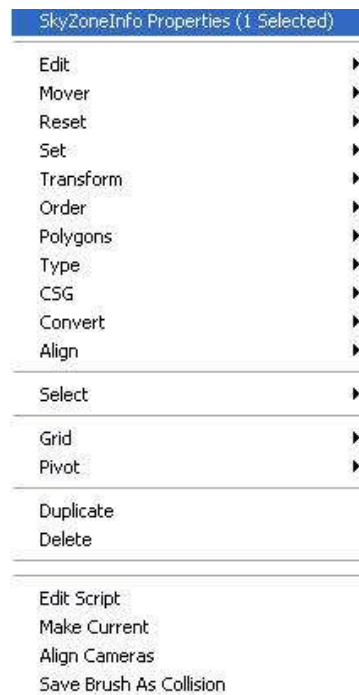


Ilustración 165 Propiedades de SkyZoneInfo



Para completar esta parte hubo que configurar las siguientes propiedades de SkyZoneInfo:

Grupo ZoneInfo

- bDistanceFog: se habilitó la distancia de niebla. Este concepto se explica en el punto 2.2.1.2.

Grupo ZoneLight

- AmbientBrightness: explicado en el punto 2.2.1.2
- AmbientHue: explicado en el punto 2.2.1.2
- AmbientSaturation: explicado en el punto 2.2.1.2
- DistanceFogColor: explicado en el punto 2.2.1.2
- DistanceFogEnd: explicado en el punto 2.2.1.2
- DistanceFogStart: explicado en el punto 2.2.1.2

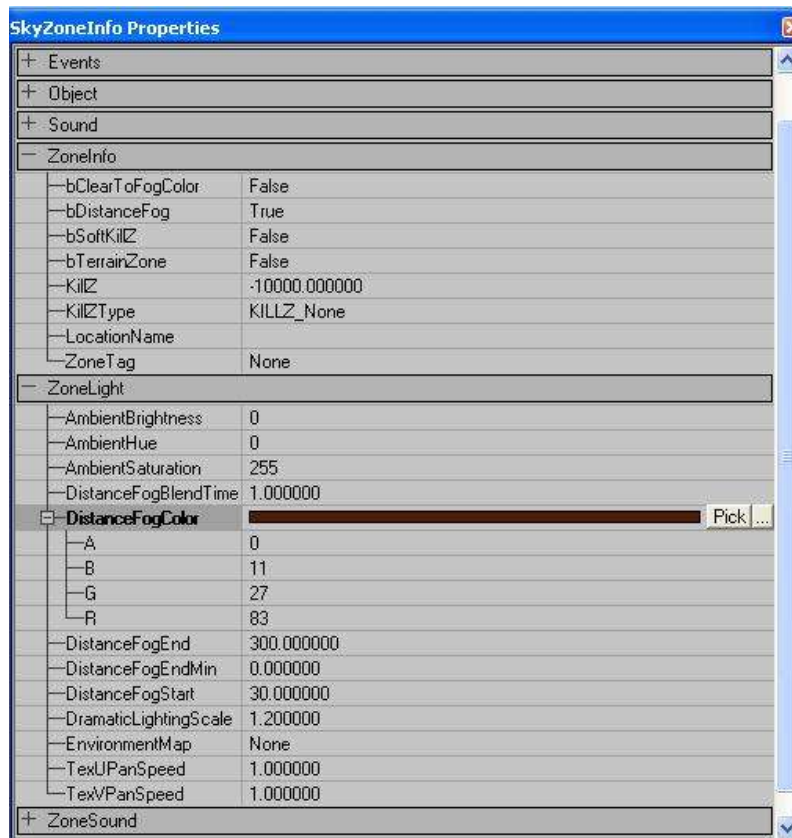



Ilustración 166 Propiedades ZoneInfo y ZoneLight de SkyZoneInfo



Ahora sólo queda realizar un último paso sin el cual no es posible ver adecuadamente las texturas cuando se le da al botón “Build All” 

6º Lo primero ha sido seleccionar todas las caras del cubo excepto la cara inferior. Para ello se ha hecho clic derecho en una de ellas y posteriormente se ha elegido la opción “Select Surfaces” > “Matching Brush”.

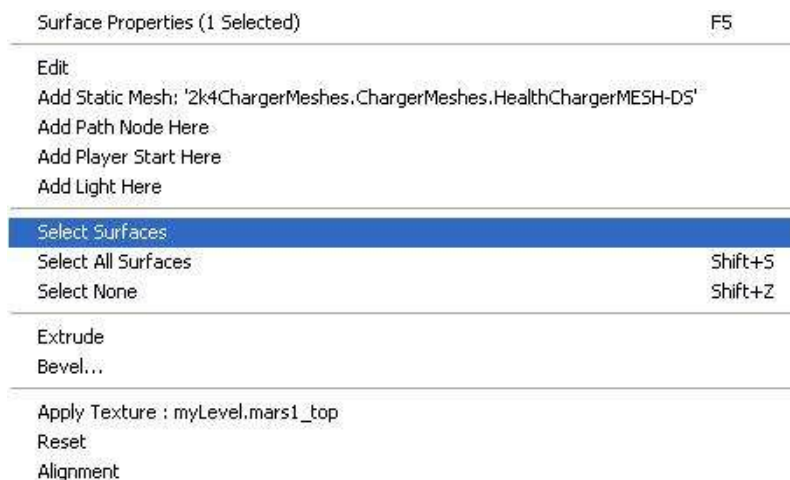


Ilustración 167 Seleccionar las superficies

Matching Groups	Shift+G
Matching Items	Shift+I
Matching Brush	Shift+B
Matching Texture	Shift+T
<hr/>	
All Adjacents	Shift+J
Adjacent Coplanars	Shift+C
Adjacent Walls	Shift+W
Adjacent Floors/Ceilings	Shift+F
Adjacent Slants	Shift+S
<hr/>	
Reverse	Shift+Q
<hr/>	
Memorize Set	Shift+M
Recall Memory	Shift+R
Or With Memory	Shift+O
And With Memory	Shift+U
Xor With Memory	Shift+X


Ilustración 168 Seleccionar aquellas superficies pertenecientes al mismo brush



Con esto se consigue seleccionar las seis caras del cubo. Para deseleccionar la cara inferior tan sólo hay que mantener presionada la tecla Ctrl mientras se hace clic izquierdo sobre ella. Ahora que ya están seleccionados únicamente los cuatro laterales y el techo del cubo se hace clic derecho para acceder a sus propiedades:



Ilustración 169 Acceder a las propiedades de las cinco caras del cubo

7º En el cuadro de diálogo de propiedades se marca el flag “Unlit”. Esta opción sirve para indicar que las caras del cubo no se verán afectadas por el proceso de iluminación (al hacer clic sobre “Build All” ) de modo que se verán igual tanto en la vista “Dynamic Light” como dentro del juego, mostrándose en su máxima luminosidad.

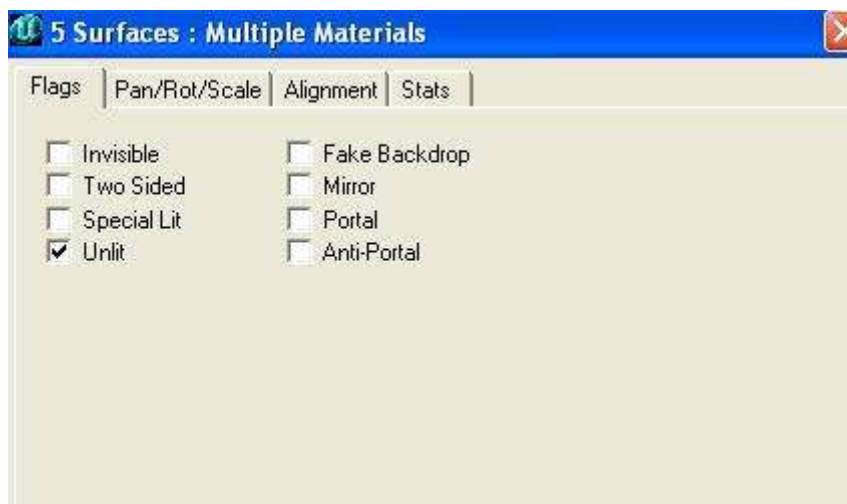


Ilustración 170 Propiedades de las caras del cielo - Habilitar la opción "Unlit"



Finalmente, se han seleccionado las paredes y el techo del mapa y se han habilitado las opciones “Unlit” y “Fake Backdrop”. El modo de proceder ha sido el mismo que en los pasos 6º y 7º, ya explicados anteriormente para seleccionar el flag “Unlit”. En este caso además se ha marcado la opción “Fake Backdrop” para que el cielo, visto desde la cámara del actor “SkyZoneInfo”, se vea reflejado/pintado en las superficies del mapa que interesan.



Ilustración 171 Habilitar las opciones "Unlit" y "Fake Backdrop"




2.2.2 Simulación del entorno marciano

Para cumplir con los objetivos de esta segunda parte del proyecto se ha utilizado el simulador USARSim. En el anexo A: Simulación en USARSim se detallan los pasos necesarios para arrancar el simulador e Iridium UI.

2.2.2.1 Starting Poses

Antes de añadir un robot al entorno es preciso definir sus posiciones iniciales. Para ello hay que añadir objetos de tipo “PlayerStart” al mapa. Se podrían utilizar también “NavigationPoint”, no obstante es preferible usar los primeros porque permiten ver y ajustar la orientación en el editor.

A continuación se explican los pasos necesarios para añadir un “PlayerStart” en Unreal Level Editor.

1º Clic izquierdo sobre “Actor Class Browser” .

2º Abrir el paquete Engine.u y seleccionar NavigationPoint > SmallNavigationPoint > PlayerStart.

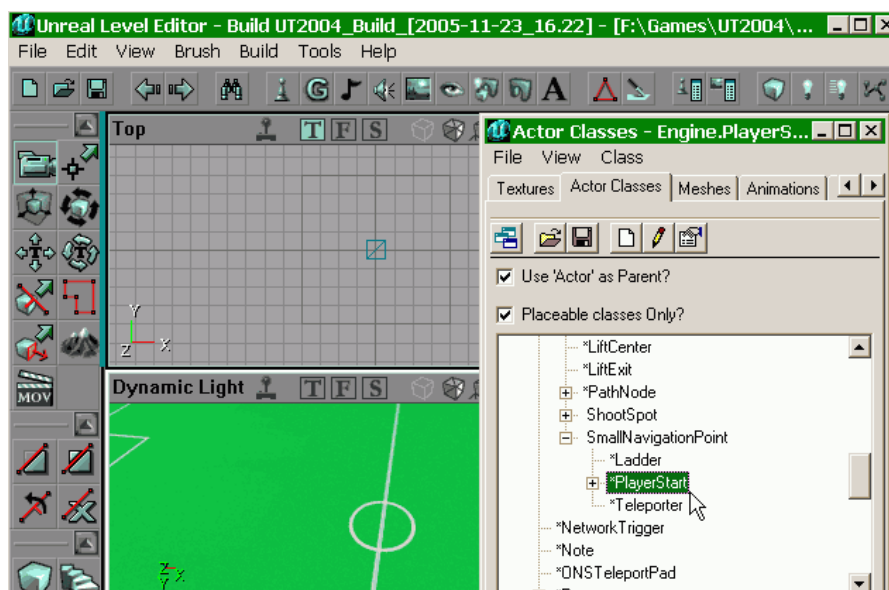


Ilustración 172 Seleccionar NavigationPoint > SmallNavigationPoint > PlayerStart

3º Clic derecho sobre cualquier lugar del mapa y seleccionar “Add PlayerStart Here”.

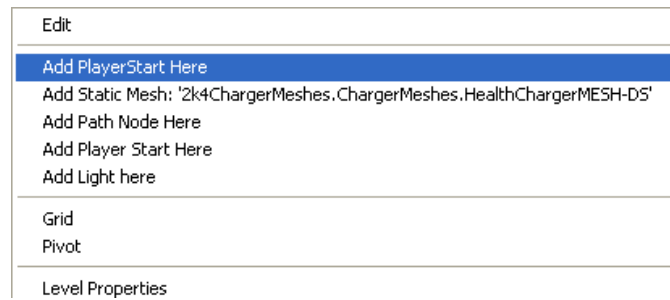


Ilustración 173 Añadir un actor de tipo "PlayerStart"

Un icono como el siguiente aparece ahora en el mapa.

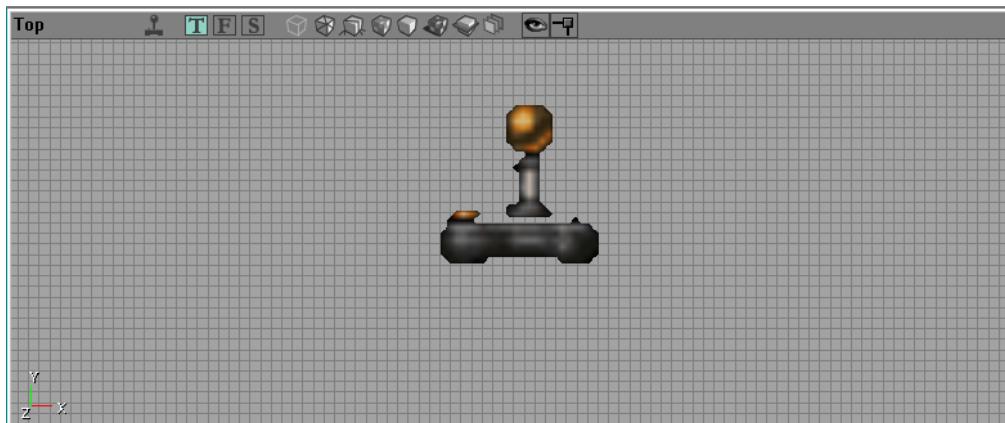


Ilustración 174 Icono usado para representar un actor de tipo "PlayerStart"

2.2.2.2 *Modo MultiView*

El cliente USARSim sólo permite ver a través de la/s cámara/s en un instante dado un robot. Para solventar esta dificultad se desarrolló una extensión denominada MultiView que permite capturar la vista de la cámara de muchos robots al mismo tiempo.

Para habilitarla hay que añadir un objeto especial desde Unreal Level Editor. A continuación se enumeran los pasos:

1º Clic izquierdo sobre “Actor Class Browser” . A continuación se abre una ventana del explorador en la que aparecen los distintos actores.

2º Abrir el paquete USARBot.u (ilustración 175) y seleccionar Pawn > MultiView (ilustración 176).

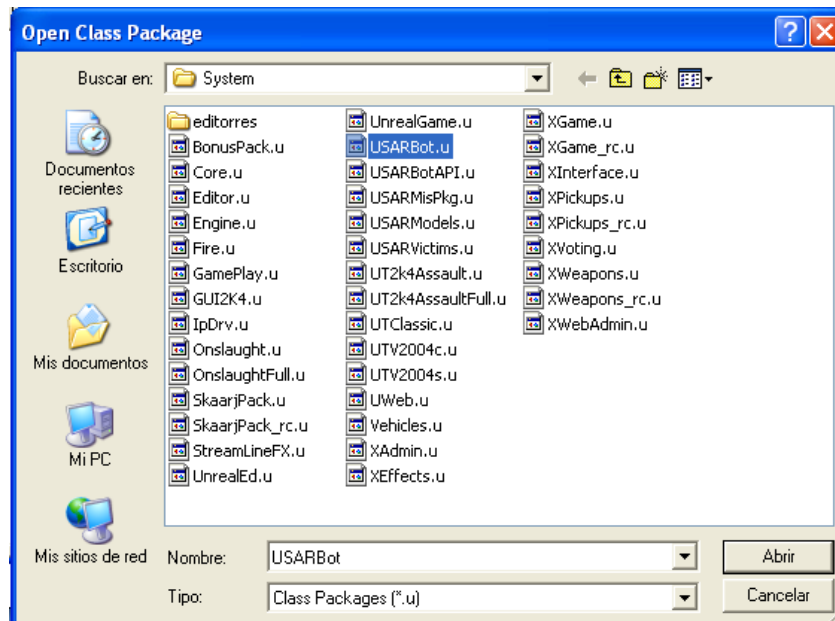


Ilustración 175 Abrir el paquete USARBot.u

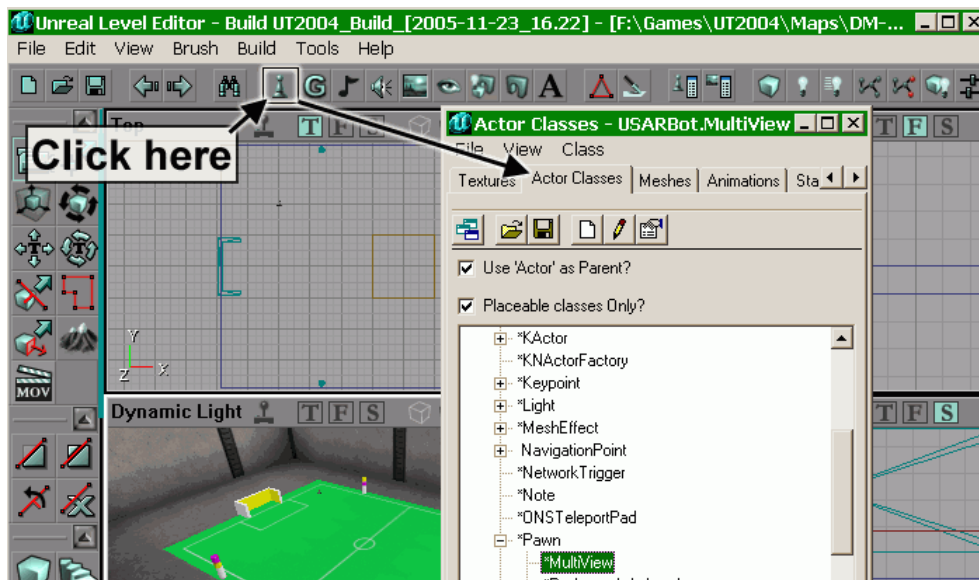


Ilustración 176 Seleccionar Pawn > MultiView

3º Clic derecho sobre cualquier lugar del mapa y seleccionar “Add MultiView Here”.

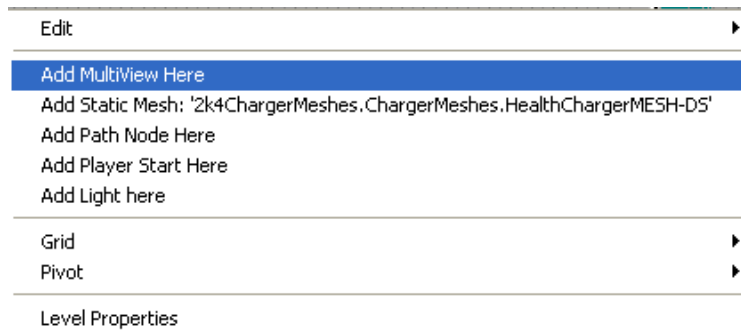


Ilustración 177 Añadir al mapa un actor de tipo "MultiView"

Una caja como la siguiente debería aparecer ahora en el mapa:

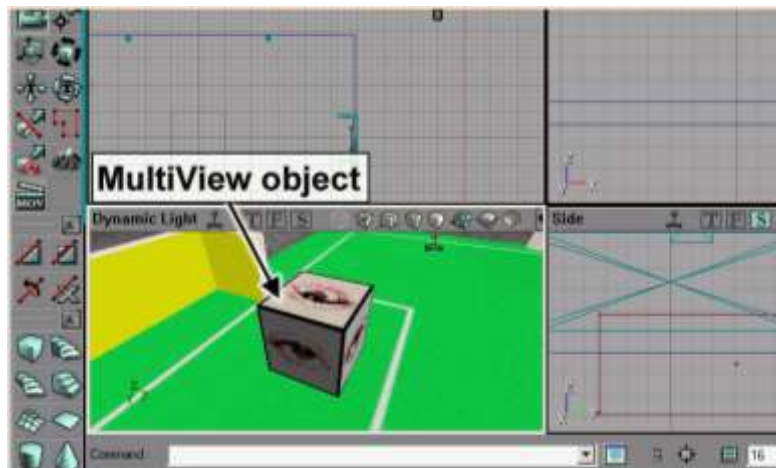


Ilustración 178 Objeto "MultiView"

4º En el Front viewport situar la caja cerca del techo del mapa para ocultarla de las cámaras del robot.

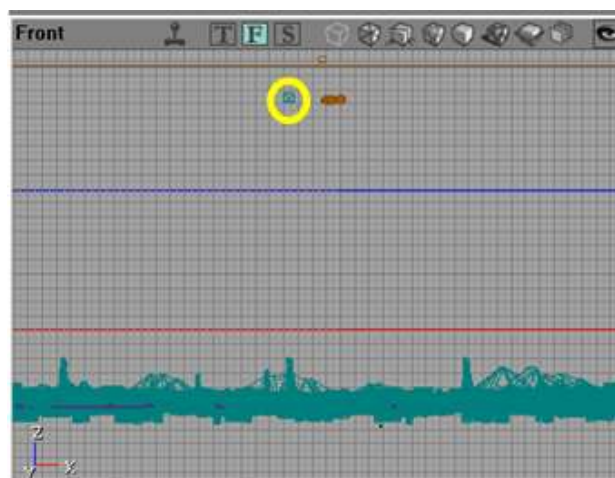


Ilustración 179 "MultiView" ubicado en el techo del mapa



2.2.2.3 Incorporación del robot al entorno

Al tratarse de un modelo dinámico el robot se añade en tiempo de ejecución. Una vez conectado Iridium al servidor (ver Anexo A) ya es posible enviar comandos textuales pero aún no se puede leer los sensores hasta que primero no se incluya el robot.

Para añadir un robot al entorno (como ya se vio en el punto 2.1.4.4) hay que utilizar el siguiente comando:

```
INIT {ClassName robot_class}{Name robot_name}{Location x,y,z}{Rotation p,y,r}
```

Ilustración 180 Comando para añadir un robot al entorno

- robot_class: tipo del robot que se desea añadir. Se encuentran en el paquete USARBot y puede ser uno de los siguientes:
 - P2AT, P2DX, ATRV-Jr, Talon, Tarantula, TeleMax...
- robot_name: nombre del robot que se desea añadir. Es opcional. Si no se especifica ningún nombre se le asignará Unnamed_Bot_n siendo n un número entero (comienza en 1).
- x,y,z: especifica la posición en el sistema de coordenadas en metros en la que se añadirá el robot.

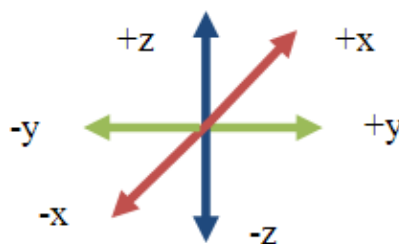


Ilustración 181 Sistema de coordenadas

Dado que las posiciones en Unreal se especifican en UU y USARSim trabaja en m existe la siguiente equivalencia.

$$250 \text{ UU} = 1 \text{ m}$$

Ilustración 182 Conversión entre UU y m



- p,y,r: especifica pitch, yaw y roll para el robot.

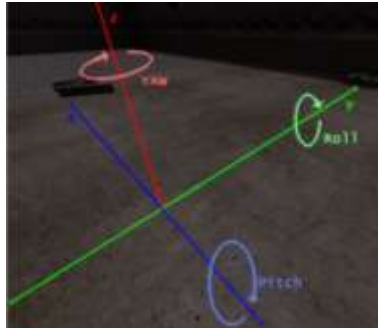


Ilustración 183 Pitch, Yaw y Roll

Dado que Unreal trabaja en UU y USARSim en radianes o grados dependiendo de si se trata de funciones trigonométricas o del campo de visión (FOV) existe la siguiente equivalencia.

$$32768 \text{ UU} = 3.1415 \text{ radianes} = 180 \text{ grados}$$

Ilustración 184 Conversión entre UU y radianes-grados

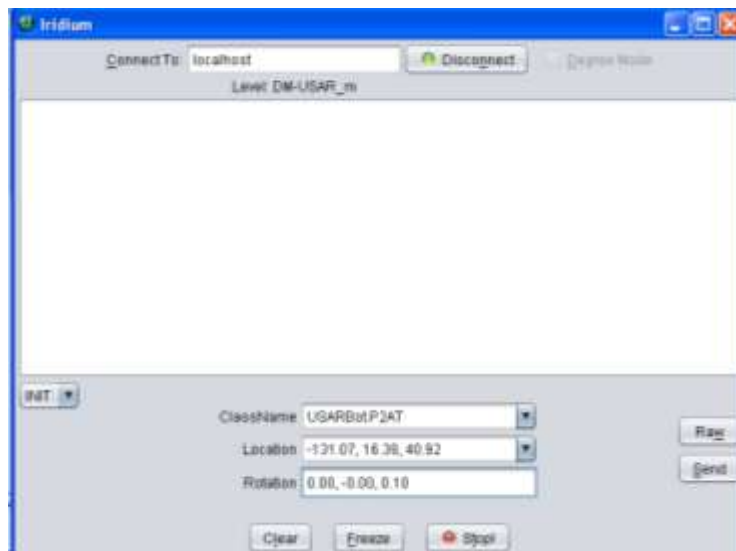



Ilustración 185 Iridium - Añadir P2AT al entorno marciano



Ilustración 186 P2AT añadido al entorno marciano

El robot que se ha añadido al entorno es un P2AT. En el directorio C:\UT2004\System

se puede editar el archivo de configuración USARBot.ini  en el que se define la configuración de cada robot. En la sección correspondiente al P2AT [USARBot.P2AT] es posible configurar los siguientes atributos:

- Peso del chasis en kilogramos
- Capacidad de carga útil en kilogramos
- Duración de la batería del robot en segundos
- Masa del chasis en kilogramos
- Sensores montados en el robot
 - Encoder sensor
 - Sonar sensor
 - Touch sensor
- Cámara del robot
- Headlight del robot

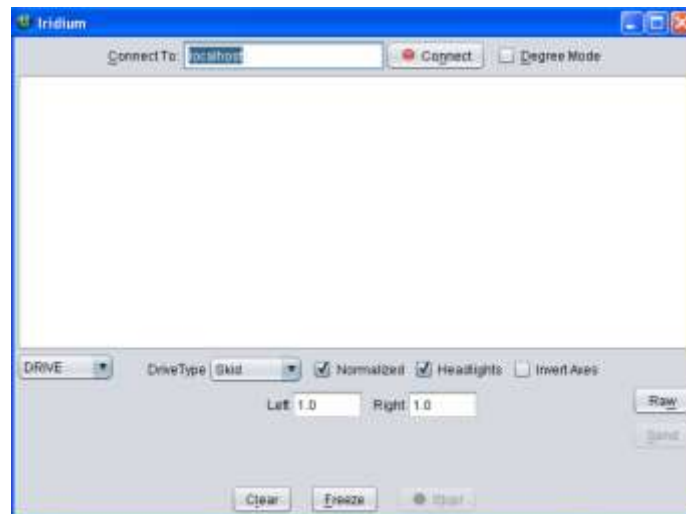


Ilustración 189 Mover P2AT hacia delante a velocidad 1 rad/s

Si ahora queremos que gire a la izquierda a una velocidad de 5 rad/s:

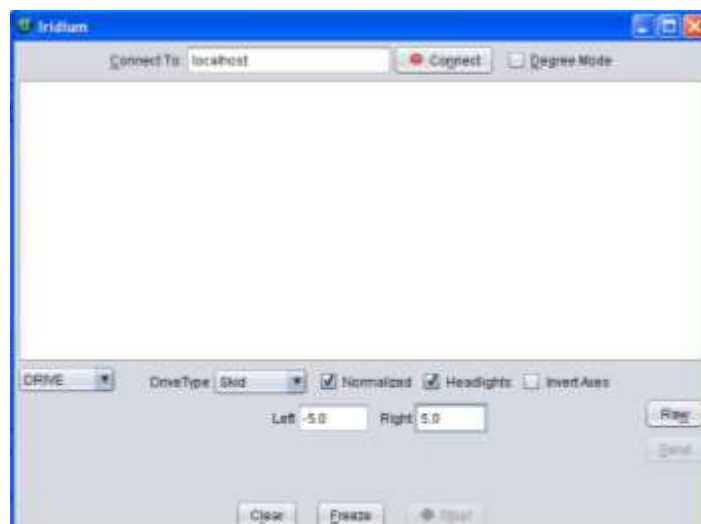


Ilustración 190 Girar P2AT a izquierda a 5 rad/s



2.3 Resultados obtenidos

La solución que aquí se propone es una de las posibles formas de resolver el problema. Consiste en un entorno virtual que imita las características de la superficie del planeta Marte: atmósfera, orografía y terreno principalmente.



Ilustración 191 Formas volcánicas (Static Mesh)

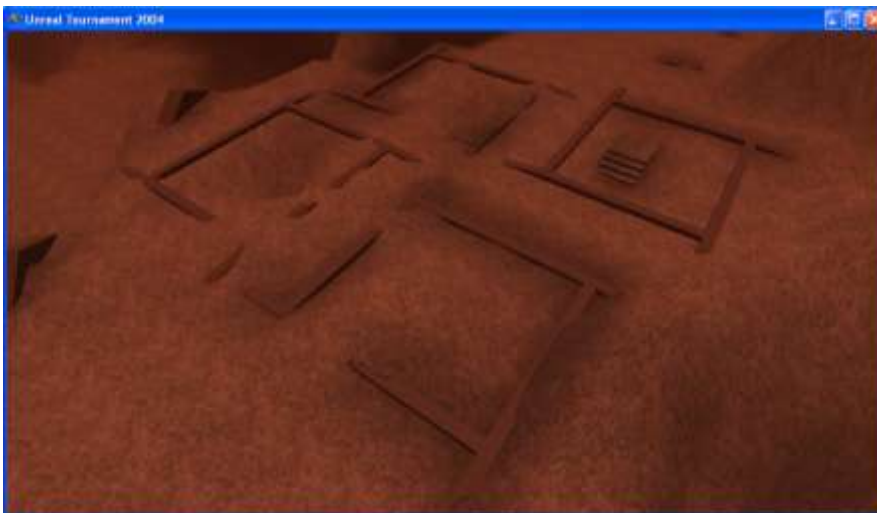


Ilustración 192 Rampas (Brush Clipped)

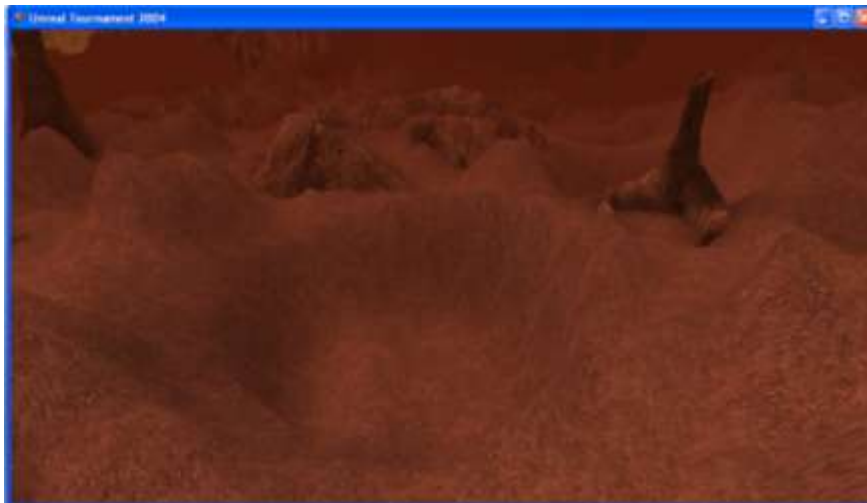


Ilustración 193 Cráter (deformación del terreno)

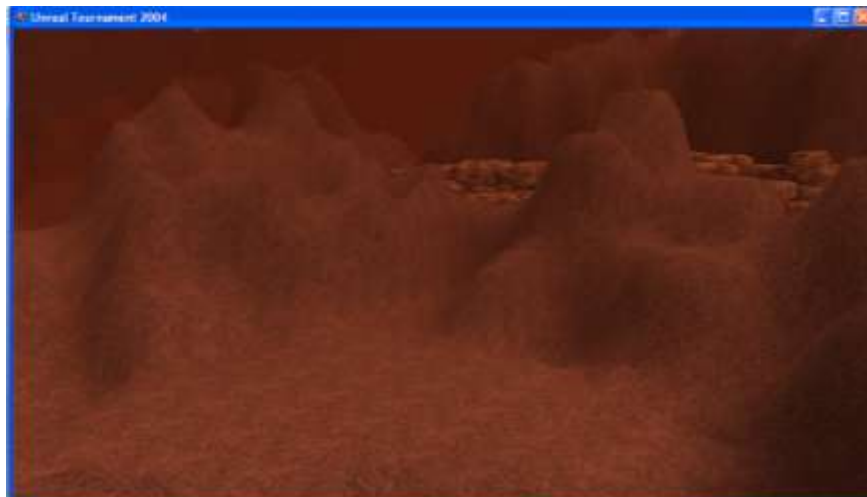


Ilustración 194 Montañas (deformación del terreno)



Ilustración 195 Atmósfera marciana



3. CONCLUSIONES

En este apartado se presenta un resumen del trabajo realizado en el que se recogen las ideas principales y más relevantes.

Actualmente la exploración humana del espacio conlleva una serie de riesgos que pueden clasificarse en biológicos, tecnológicos y económicos. La precaria situación financiera a nivel mundial ha sido uno de los factores determinantes que han llevado a la NASA a retrasar las misiones espaciales con seres humanos. Por su parte la exploración robótica del espacio ofrece numerosas ventajas y es mucho más barata. Sin embargo antes de enviar un robot al espacio hay que realizar pruebas en un simulador para evitar correr riesgos innecesarios. Para ello es necesario crear un entorno virtual similar al real en el que poder llevar a cabo las pruebas pertinentes.

El grueso del proyecto se concentra en la parte de desarrollo del entorno marciano. Para su creación se han consultado diversas fuentes bibliográficas con el fin de adquirir el conocimiento necesario tanto a nivel de aprendizaje de las herramientas usadas como en el ámbito de las características del entorno real. La labor de diseño se ha dividido en cinco fases —Creación de un mapa para un mundo marciano, Creación de la atmósfera marciana, Iluminación del mapa, Creación del terreno marciano, Construcción de rampas y Creación del cielo marciano— que han resultado en el entorno marciano propuesto. A continuación se enumeran los puntos más importantes de esta parte:

- El entorno marciano es un mapa de exteriores
- El mapa es un espacio substractivo (*Subtractive BSP*)
- El espacio substractivo representa aquella zona por la que puede moverse el robot.
- La iluminación elegida para el entorno simula la luz procedente de los rayos solares (Actor de tipo *Sunlight*).
- Efecto visual de polvo en suspensión en la atmósfera mediante *ZoneInfo*
- El terreno se ha modelado (deformado) conforme a las características reales de la superficie de Marte: cráteres, volcanes, valles, montañas, etc.



- Algunas montañas presentes en el entorno son modelos estáticos (*Static Mesh*)
- Las rampas son espacio aditivo (*Additive BSP*)
- Las rampas se han construido siguiendo la técnica denominada *Brush Clipped*
- El cielo se ha construido siguiendo la técnica denominada *Sky Box*

El punto que ha resultado más complejo de realizar ha sido la deformación del terreno. Como ya se comentó en el apartado 2.1.3.1 la herramienta de manipulación del terreno es una de las más complejas ya que requiere de un dominio preciso de las diferentes técnicas, para lo cual se necesita bastante tiempo y práctica.

La parte de simulación del entorno marciano se ha llevado a cabo en el simulador USARSim. Este incluye robots simulados lo cual ha facilitado enormemente el trabajo. Una vez arrancado el simulador en modo “Servidor-Cliente” se pasa a ejecutar la aplicación que actúa como controlador (ver Anexo A: Simulación en USARSim). Esta permite enviar comandos al servidor en forma de cadenas de texto y envía de vuelta las respuestas de este. Con estos comandos es posible añadir un robot al entorno y establecer un camino (waypoints) por el que se moverá el robot. La principal dificultad que se ha encontrado ha sido:

La posición en la que se añade el robot al entorno debe calcularse con mucha precisión porque de lo contrario se producirá un error del tipo `> DIE {Killer None} {DamageType Engine.fell}` debido a que el robot cae desde el cielo.

Para solucionar este problema hay que definir los valores de *z* exactos para el tipo de robot que se quiere añadir.

Robot	Posición sobre el suelo	Delta z
P2AT	-26 cm	-7 cm
P2DX	-20 cm	-1 cm



Robot	Posición sobre el suelo	Delta z
Zerg	-6 cm	+13 cm
Talon	-13 cm	+6 cm
ATRV-Jr	-39 cm	-20 cm

Tabla 16 Valores de Z para cada robot

En nuestro caso se ha añadido un P2AT por lo que debemos restar al valor de z original -0.07 m para obtener la posición exacta de z.

Si por ejemplo se quiere añadir el robot P2AT en la posición $x = -0.96$, $y = -1.86$, $z = -2.24$, la posición final será $x = -0.96$, $y = -1.86$, $z = (-2.24-0.07)$

3.1 Objetivos conseguidos

Al término del proyecto se han alcanzado los dos objetivos principales del mismo:

- ✓ Desarrollo del entorno marciano conforme a los requerimientos iniciales
- ✓ Simulación del entorno marciano en la herramienta USARSim



3.2 Cronograma

En este apartado se presenta un Diagrama de Gantt en el que se detallan en orden cronológico todas las tareas que se han ido realizando a lo largo del proyecto. Como se puede apreciar al término del mismo el porcentaje de completado de cada tarea es del 100%.

Los principales hitos del proyecto han sido:

- Desarrollo del entorno marciano
- Simulación del entorno marciano
- Elaboración de la memoria

En la ilustración inferior se muestra para cada tarea el nombre de la misma, su duración en días y la fecha de comienzo y de finalización.

	Nombre de tarea	Duración	Comienzo	Fin
1	DESARROLLO DEL ENTORNO MARCIAHO	15 días	jue 01/09/11	mié 21/09/11
2	Creación del mapa	3 días	jue 01/09/11	lun 05/09/11
3	Creación de la atmósfera	2 días	mar 06/09/11	mié 07/09/11
4	Iluminación del mapa	1 día	jue 08/09/11	jue 08/09/11
5	Creación del terreno	7 días	vie 09/09/11	lun 19/09/11
6	Construcción de rampas	1 día	mar 20/09/11	mar 20/09/11
7	Creación del cielo	1 día	mié 21/09/11	mié 21/09/11
8				
9	SIMULACIÓN DEL ENTORNO MARCIAHO	8 días	jue 22/09/11	lun 03/10/11
10	Definición de Starting Poses	1 día	jue 22/09/11	jue 22/09/11
11	Habilitar modo MultiView	1 día	vie 23/09/11	vie 23/09/11
12	Incorporación del robot al entorno	5 días	lun 26/09/11	vie 30/09/11
13	Control del robot	1 día	lun 03/10/11	lun 03/10/11
14				
15	ELABORACIÓN DE LA MEMORIA	17 días	jue 22/09/11	vie 14/10/11
16	Resolución del problema	6 días	jue 22/09/11	jue 29/09/11
17	Introducción al problema a resolver	3 días	vie 30/09/11	mar 04/10/11
18	Conclusiones	2 días	mié 05/10/11	jue 06/10/11
19	Líneas futuras de trabajo	1 día	vie 07/10/11	vie 07/10/11
20	Simulación en USARSim	2 días	lun 10/10/11	mar 11/10/11
21	Glosario de Términos	1 día	mié 12/10/11	mié 12/10/11
22	Acrónimos	1 día	jue 13/10/11	jue 13/10/11
23	Referencias	1 día	vie 14/10/11	vie 14/10/11

Ilustración 196 Principales hitos del proyecto



En la tabla siguiente se recoge el número total de horas dedicadas a los tres hitos principales del proyecto:

Tarea	Duración (días)	Dedicación (horas/día)	Total Horas
1. Desarrollo del entorno marciano			
1.1 Creación del mapa	3	8	24
1.2 Creación de la atmósfera	2	8	16
1.3 Iluminación del mapa	1	8	8
1.4 Creación del terreno	7	8	56
1.5 Construcción de rampas	1	8	8
1.6 Creación del cielo	1	8	8
Total			120
2. Simulación del entorno marciano			
2.1 Definición de Starting Poses	1	1	1
2.2 Habilitar modo MultiView	1	2	2
2.3 Incorporación del robot al entorno	5	8	40
2.4 Control del robot	1	4	4
Total			47



Tarea	Duración (días)	Dedicación (horas/día)	Total Horas
3. Elaboración de la memoria			
3.1 Introducción al problema a resolver	3	8	24
3.2 Resolución del problema	6	8	48
3.3 Conclusiones	2	4	8
3.4 Líneas futuras de trabajo	1	4	4
3.5 Simulación en USARSim	2	8	16
3.6 Glosario de Términos	1	4	4
3.7 Acrónimos	1	2	2
3.8 Referencias	1	1	1
Total			107

Tabla 17 Total de horas para cada tarea

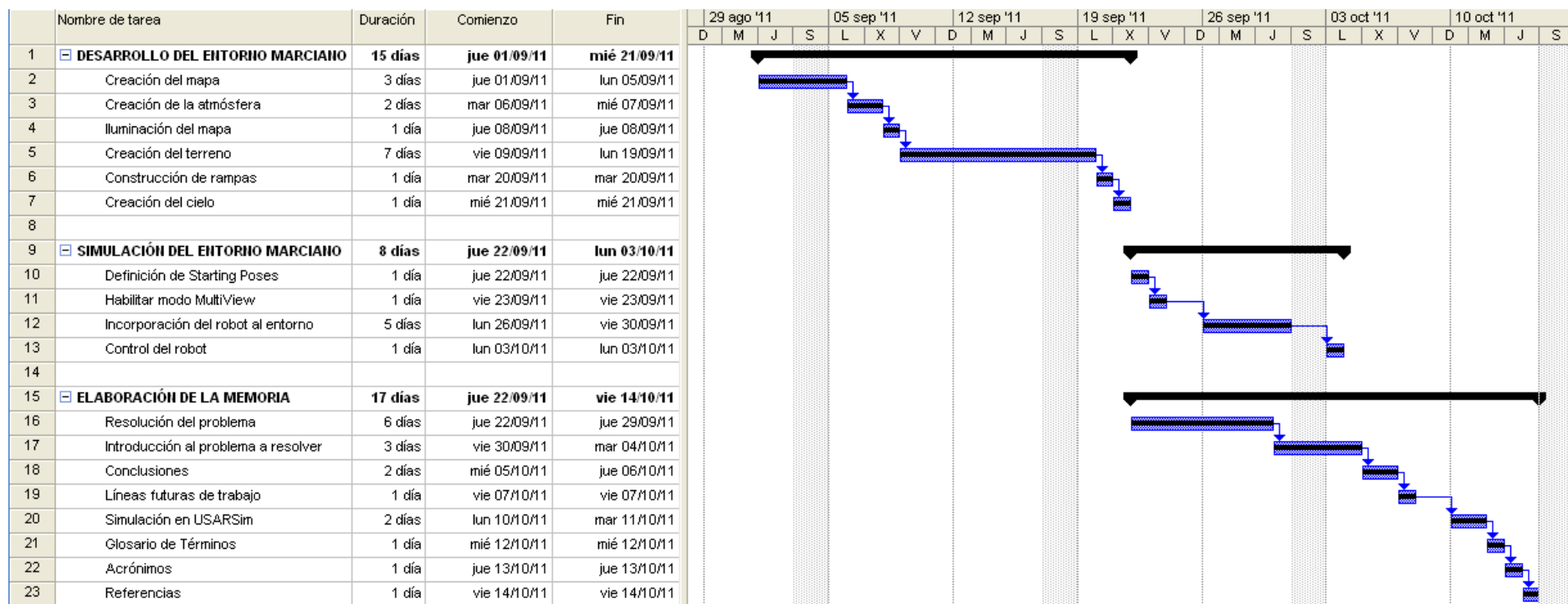


Ilustración 197 Diagrama de Gantt con los principales hitos del proyecto



3.3. Presupuesto

En esta sección se recoge el coste total del proyecto desglosado en: costes de personal, costes de equipos, costes de software y costes de material fungible.

3.3.1 Costes de personal

Para realizar los cálculos se ha considerado que el sueldo/hora de un ingeniero superior en informática es igual a 30€.

Tarea	Duración (horas)	Coste (€)
Desarrollo del entorno marciano	120	3600
Simulación del entorno marciano	47	1410
Elaboración de la memoria	107	3210
Total	274	8220

Tabla 18 Costes de personal

3.3.2 Costes de equipos

En la siguiente tabla se muestran los equipos adquiridos para la realización del proyecto.

Nombre	Precio unitario(€)	Unidades	Coste(€)
Portátil Fujitsu Amilo Pi 1536	1200	1	1200
Router Asus WL-600G	42,27	1	42,27
Impresora HP PhotoSmart C4480	194	1	194
Total			1436,27

Tabla 19 Costes de equipos



3.3.3 Costes de software

En la siguiente tabla se muestran los programas de pago necesarios para la realización del proyecto.

Concepto	Coste/licencia	Licencias	Coste(€)
Microsoft Windows XP *	0	1	0
Microsoft Office 2007	155	1	155
Total			155

Tabla 20 Costes de software

*La licencia de Windows XP se incluye con el portátil

El resto de los programas utilizados (Unreal Tournament 2004, USARSim para UT2004 y GIMP 2.6.11) son gratuitos al tratarse de software libre.

3.3.4 Costes de material fungible

Nombre	Precio unitario(€)	Unidades	Coste(€)
Memoria USB	6	1	6
Tóner impresora	28,45	2	56,90
Material de oficina	12	1	12
Total			74,90

Tabla 21 Costes de material fungible



3.3.5 *Coste total*

Tras haber calculado los costes correspondientes a personal, equipos, software y material fungible a continuación se presenta el presupuesto final para el proyecto:

Concepto	Coste(€)
Costes de personal	8220
Costes de equipos	1436,27
Costes de software	155
Costes de material fungible	74,90
	Subtotal 9886,17
IVA (16%)	1581,79
	Total 11467,96

Tabla 22 Coste total del proyecto



4. LÍNEAS FUTURAS DE TRABAJO

En este apartado se enumeran algunas consideraciones finales de cara a un futuro como posibles ampliaciones o mejoras.

Entorno marciano

- Añadir efectos especiales al mapa (como una tormenta de arena)
- Añadir nuevas capas al terreno para dotarle de mayor realismo
- Añadir nuevos caminos especiales (RoadPathNodes) por los que puedan transitar los vehículos en el mapa.

Simulación en USARSim

- Añadir varios robots de diferentes tipos y estudiar cómo se comportan en el entorno.
- Comunicar a los robots entre sí abriendo una conexión TCP/IP para cada uno
- Hacer pasar al robot por zonas estrechas y estudiar cómo actúa.



Anexos

A. Simulación en USARSim

En este anexo se describen los pasos necesarios para llevar a cabo la simulación. La versión del simulador que se ha utilizado puede descargarse de la siguiente URL: <http://sourceforge.net/projects/usarsim/files/usarsim-2004/Final/>.

A.1 Compilar USARSim

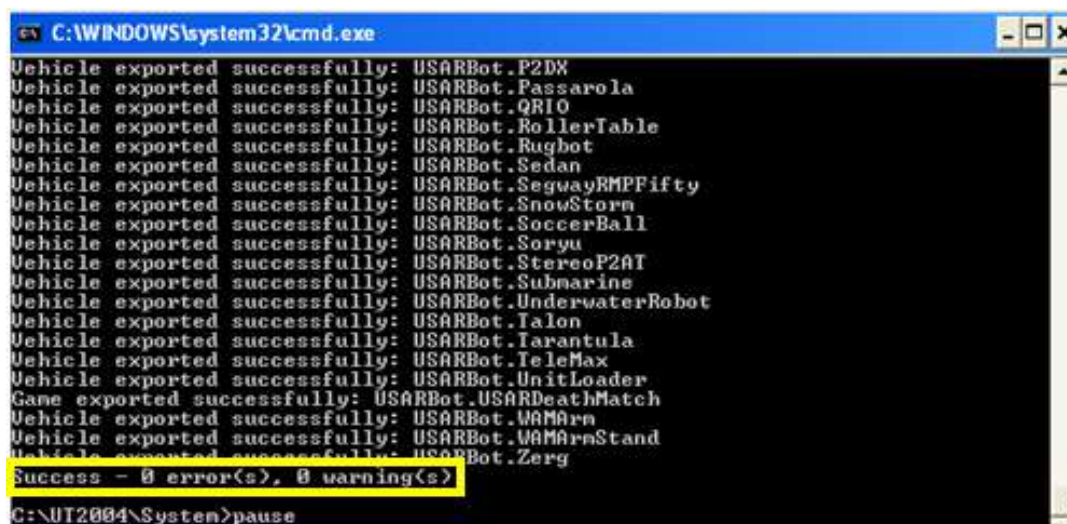
Una vez instalado USARSim en el mismo directorio de instalación de Unreal Tournament 2004 (C:\UT2004) proceder del siguiente modo:

1º Ir a C:\UT2004\System



2º Ejecutar el archivo make.bat 

Si todo ha ido bien, al finalizar aparece un mensaje como el que se muestra a continuación:



```
C:\WINDOWS\system32\cmd.exe
Vehicle exported successfully: USARBot.P2DX
Vehicle exported successfully: USARBot.Passarola
Vehicle exported successfully: USARBot.QRIO
Vehicle exported successfully: USARBot.RollerTable
Vehicle exported successfully: USARBot.Rugbot
Vehicle exported successfully: USARBot.Sedan
Vehicle exported successfully: USARBot.SegwayRMPFifty
Vehicle exported successfully: USARBot.SnowStorm
Vehicle exported successfully: USARBot.SoccerBall
Vehicle exported successfully: USARBot.Soryu
Vehicle exported successfully: USARBot.StereoP2AT
Vehicle exported successfully: USARBot.Submarine
Vehicle exported successfully: USARBot.UnderwaterRobot
Vehicle exported successfully: USARBot.Talon
Vehicle exported successfully: USARBot.Tarantula
Vehicle exported successfully: USARBot.TeleMax
Vehicle exported successfully: USARBot.UnitLoader
Game exported successfully: USARBot.USARDeathMatch
Vehicle exported successfully: USARBot.WAMARA
Vehicle exported successfully: USARBot.WAMARAStand
Vehicle exported successfully: USARBot.Zerg
Success - 0 error(s), 0 warning(s)
C:\UT2004\System>pause
```

Ilustración 198 Compilar USARSim



A.2 Cómo arrancar USARSim

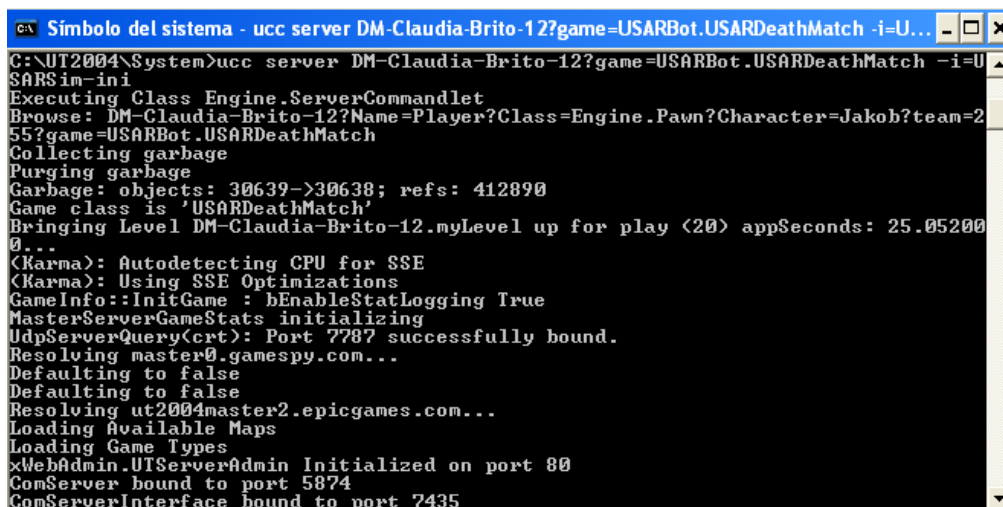
Para arrancar el simulador en modo “Servidor-Cliente” es necesario realizar los siguientes pasos:

1º Arrancar el servidor USARSim (modo manual)

- Abrir una consola MS-DOS
- Situarse en el directorio C:\UT2004\System
- Ejecutar el siguiente comando:

```
ucc server nombre_del_mapa?game=USARBot.USARDeathMatch -i=USARSim.ini
```

Ilustración 199 Comando para arrancar el servidor



```
C:\UT2004\System>ucc server DM-Claudia-Brito-12?game=USARBot.USARDeathMatch -i=USARSim.ini
Executing Class Engine.ServerCommandlet
Browse: DM-Claudia-Brito-12?Name=Player?Class=Engine.Pawn?Character=Jakob?team=255?game=USARBot.USARDeathMatch
Collecting garbage
Purging garbage
Garbage: objects: 30639->30638; refs: 412890
Game class is 'USARDeathMatch'
Bringing Level DM-Claudia-Brito-12.myLevel up for play <20> appSeconds: 25.052000...
(Karma): Autodetecting CPU for SSE
(Karma): Using SSE Optimizations
GameInfo::InitGame : bEnableStatLogging True
MasterServerGameStats initializing
UdpServerQuery(crt): Port 7787 successfully bound.
Resolving master0.gamespy.com...
Defaulting to false
Defaulting to false
Resolving ut2004master2.epicgames.com...
Loading Available Maps
Loading Game Types
xWebAdmin.UTServerAdmin Initialized on port 80
ComServer bound to port 5874
ComServerInterface bound to port 7435
```

Ilustración 200 Arrancar el servidor (modo manual)

Para arrancar el servidor de modo automatizado, una vez situados en el directorio



C:\UT2004\System ejecutar el archivo `usar_s.bat` `usar_s`. A continuación se abre una consola MS-DOS como la que se muestra más abajo en la que se ejecuta automáticamente el comando.



```
C:\WINDOWS\system32\cmd.exe
C:\UT2004\System>ucc server DM-Claudia-Brito-12?game=USARBot.USARDeathMatch?Time
Limit=0?GameStats=False -ini=USARSim.ini -log=usar_server.log
Executing Class Engine.ServerCommandlet
Browse: DM-Claudia-Brito-12?Name=Player?Class=Engine.Pawn?Character=Jakob?team=2
55?game=USARBot.USARDeathMatch?TimeLimit=0?GameStats=False
Collecting garbage
Purging garbage
Garbage: objects: 30639->30638; refs: 412890
Game class is 'USARDeathMatch'
Bringing Level DM-Claudia-Brito-12.myLevel up for play (20) appSeconds: 7.158000
...
(Karma): Autodetecting CPU for SSE
(Karma): Using SSE Optimizations
GameInfo::InitGame : bEnableStatLogging False
Defaulting to false
Defaulting to false
MasterServerUplink: DoUplink is False, not connecting to Epic master server
Loading Available Maps
Loading Game Types
xWebAdmin.UTServerAdmin Initialized on port 80
ComServer bound to port 5874
ComServerInterface bound to port 7435
```

Ilustración 201 Arrancar el servidor (modo automatizado)

De esta forma cuando se quiera cargar otro mapa no es necesario reescribir todo el comando completo. Simplemente hay que editar el archivo `usar_s.bat` y cambiar el nombre del mapa por el del nuevo.

2º Arrancar el cliente Unreal Tournament (modo manual)

- Abrir una consola MS-DOS
- Situarse en el directorio `C:\UT2004\System`
- Ejecutar el siguiente comando:

```
ut2004 dir_ip?spectatoronly=1?quickstart=true
```

Ilustración 202 Comando para arrancar el cliente

`dir_ip`: dirección IP del servidor. Si el servidor y el cliente están corriendo en la misma máquina `dir_ip` es `127.0.0.1`.

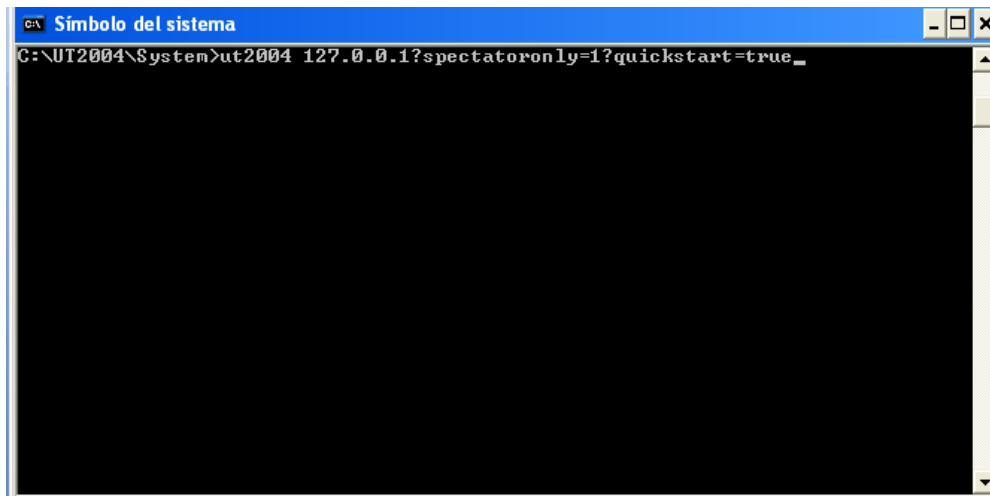


Ilustración 203 Arrancar el cliente (modo manual)

Una vez arrancado, el cliente UT se pondrá a escuchar conexiones TCP entrantes en el puerto 3000.

Para arrancar el cliente de modo automatizado, una vez situados en el directorio



C:\UT2004\System ejecutar el archivo `usar_c.bat usar_c`. Seguidamente aparece una ventana en la que se avisa al usuario de que se está cargando el mapa.

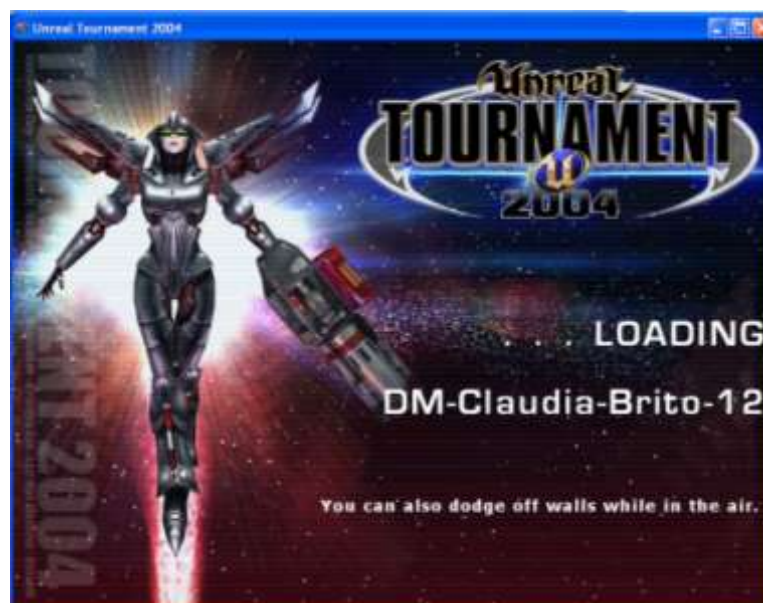


Ilustración 204 Carga del mapa en USARSim



Para arrancar el simulador en modo “sólo cliente” simplemente hay que introducir el siguiente comando desde la consola de MS-DOS:

```
ut2004 nombre_del_mapa?spectatoronly=1?quickstart=true -ini=usarsim.ini
```

Ilustración 205 Comando para arrancar el simulador


3º Arrancar el controlador

Descargar Iridium, el controlador de USARSim para UDK de la siguiente URL:

http://sourceforge.net/projects/usarsim/files/usarsim-UDK/USARSimFull_UDKV1.1.zip/download

Descomprimir en el directorio de UT2004 (C:\UT2004)

Para ejecutarlo hay que situarse primero en el directorio en el que se haya instalado. En este caso C:\UT2004\IridiumUI\bin y a continuación hacer doble clic sobre el

ejecutable.  Iridium
Executable Jar File
102 KB

Al hacerlo se abrirá una ventana como la siguiente:



Ilustración 206 IridiumUI- Interfaz de control



Para comenzar a trabajar con el interfaz lo primero que hay que hacer es conectarse al servidor. Para ello hacer clic sobre el botón “Connect”. Una vez conectados (el indicador se pone en verde) ya podemos enviar comandos al servidor (se habilita el botón “Send”).

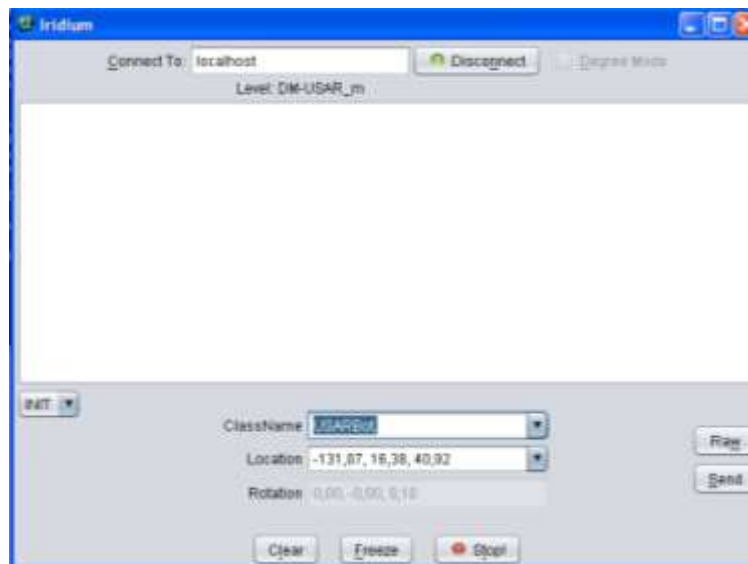


Ilustración 207 Conectarse al servidor USARSim



B. Glosario de términos

Actor

Objeto que se puede crear o situar en el mundo. Incluye: *PlayerStart*, *Static Mesh*, *ZoneInfo*, *Sound*, etc.

Additive BSP

Espacio *Binary Space Partitioning* añadido a un espacio BSP previamente sustraído.

Animation

Movimiento creado por un modelo 3D con huesos que permite moverlos en el juego.

Antiportal

Nuevo concepto que se introduce en UE2. Se trata de un volumen que oculta partes de la escena al motor de visualización, no permitiendo que dibuje lo que está más allá de él.

Artificial Intelligence

Término usado para describir el comportamiento y el aparente proceso mental de los personajes controlados por ordenador.

Binary Space Partitioning

Estructura de datos que se usa para organizar objetos en el ámbito del nivel. Geometría creada con el *Builder Brush*.

Brush

Cualquier geometría creada con las herramientas del grupo “Brush Builders” de la caja de herramientas situada en la parte izquierda del interfaz del Unreal Editor, incluyendo el pincel constructor rojo.

Brush Clipped

Técnica para construir rampas basadas en la herramienta del editor Brush Clipping.



Build

Acto de compilar el nivel de modo que pueda realizar todos los cálculos necesarios (iluminación, generación de caminos y resolución de la geometría).

Builder Brush

Pincel rojo que se emplea para sustraer o añadir volúmenes BSP al nivel.

Camera

En los viewports ortogonales (Top, Front y Side) se puede ver la cámara para la vista en perspectiva (3D) dibujada con forma de un ojo.

Constructive Solid Geometry

Término más adecuado (semánticamente) para designar un BSP. Se refiere a la geometría creada con el *Builder Brush*.

Distance Fog

Propiedad de una zona que permite descartar partes del nivel después de una cierta distancia mezclando geometría lejana con un color sólido. Esto provoca un efecto que se percibe como si la geometría estuviese desapareciendo en la atmósfera.

Fake Backdrop

Propiedad de una superficie que puede asignarse a las caras de un CSG para permitir que se vean como un *SkyBox*.

First Person Shooter

Tipo de videojuego que representa la escena en primera persona y cuya interacción principal con el mundo se realiza a través de una pistola.

Geometry

Cualquier cosa construida con triángulos en el nivel como *Static Mesh*, *BSP*, *Terrain*, etc.



Karma

Motor físico integrado en el motor Unreal.

Layer

Nivel jerárquico en el terreno que permite que las texturas sean pintadas y combinadas juntas.

Light

Actor que proyecta las luces en las superficies cercanas.

Map

Fichero que contiene todas las propiedades de los actores presentes así como el diseño de toda la escena creada.

Meshes

Modelo creado con un programa de terceros que usa animación de vértices. No es lo mismo que *Static Mesh*.

Mover

Tipo especial de *Static Mesh* que puede asignarse a fotogramas clave dentro del editor y luego ser asignado de nuevo para moverse a lo largo de un camino determinado por esos fotogramas clave cuando tiene lugar un evento.

myLevel

Paquete especial que no se puede guardar explícitamente pero que se salva cada vez que se almacena el nivel.

Package

Librería de ficheros de un tipo concreto (*Static Mesh* o *Texture*).

PlayerStart

Punto de partida del jugador o robot en el nivel



Prefab

Tipo de geometría (en desuso) usada para objetos que se sitúan alrededor del nivel

Properties

Las configuraciones para cualquier actor o textura.

Skeletal Mesh

Personaje animado creado en un programa de modelado de terceros usando un esqueleto y animaciones.

Skin

Textura/s aplicada/s a un *Static Mesh*, *Skeletal Mesh*, etc.

Sky Box

Zona del cielo que utiliza un cubo para crear la ilusión de un cielo lejano.

Sky Zone

Escena que se mueve con la vista del robot dejando atrás todo lo demás para crear un efecto de cielo/horizonte lejano.

Special Lit

Propiedad de una superficie que sólo se ve afectada por aquellas luces definidas como `bSpecialLit`.

Static Mesh

Modelo creado con otros programas (3 DS Max, Maya, Blender) que permanece inmóvil en el nivel cuando se añade al mapa.

Subtractive BSP

Espacio BSP que se ha sustraído del espacio aditivo del mundo. Representa espacio abierto en el nivel por el que el rover puede caminar.



Sunlight

Actor que simula una fuente de luz distante que emite rayos de luz paralelos del mismo modo que hace el Sol.

Surface

La textura visible de un BSP.

Surface Properties

Atributos que se pueden asignar a una superficie BSP.

Terrain

Tipo de geometría con forma de lámina teselada que puede esculpirse y deformarse mediante la herramienta “Terrain Editor” dentro del Unreal Editor.

Texture

Fichero tipo imagen que se puede asignar a la geometría en el mundo. A menudo se utiliza este término para referirse a los materiales.

Unlit

Propiedad de una textura/geometría que hace que la textura/geometría aparezca en su máximo brillo con independencia del número de luces que haya en la escena.

Viewport

Ventana gráfica en el Unreal Editor que muestra las vistas de frente y de lado, superior y en perspectiva del mapa actualmente cargado.

Volume

Nuevo concepto que se introduce en UT2003. Espacio 3D que puede afectar a diversas características del juego en tiempo real como al movimiento, la visibilidad, etc.

ZoneInfo

Actor que permite modificar las propiedades de la zona dentro de la cual reside.



Referencias

[1] Unreal Wiki

Consultas, tutoriales e ideas para el motor del videojuego Unreal

<http://wiki.beyondunreal.com/>

[2] UnrealED 3.0 Tutorials

Tutorial sobre el editor utilizado para el desarrollo del entorno marciano

<http://users.skynet.be/neimad/RHclan/tutorials/Architectonic-UnrealEd3.0Tutorials-PlanetUnreal.pdf>

[3] Unreal Editor User Guide

Guía de usuario del editor Unreal

<http://udn.epicgames.com/Three/UnrealEdUserGuide.html>

[4] Unreal Engine Tutorials

Tutoriales sobre UnrealEngine 1,2 y 3

<http://www.hourences.com/tutorials/>

[5] The Unreal Editor Tutorial Center

Tutorial sobre el editor Unreal

<http://www.freewebs.com/tuetc/index.htm>

[6] USARSim manual

Manual de USARSim para UT2004

http://usarsim.sourceforge.net/wiki/index.php/UT_2004_Manual

[7] How to create maps in USARSim

Cómo desarrollar mapas para USARSim

http://robotics.mem.drexel.edu/USAR/usar_maptut.pdf

[8] UnGlossary

Glosario de términos del UnrealEd

<http://udn.epicgames.com/Two/UnGlossary.html>



Acrónimos

AI	<i>Artificial Intelligence</i> : inteligencia artificial
BSP	<i>Binary Space Partitioning</i> : división del espacio binario
CSG	<i>Constructive Solid Geometry</i> : geometría constructiva de tipo sólido
DARPA	<i>Defense Advanced Research Projects Agency</i> : agencia de investigación de proyectos avanzados de defensa
EEUU	<i>Estados Unidos</i>
ESA	<i>European Space Agency</i> : agencia espacial europea
Euro-MARS	<i>European Mars Analogue Research Station</i> : estación para la investigación de Marte
EVA	<i>Extra-Vehicular Activity</i> : actividad extra vehicular
FAA	<i>Federal Aviation Administration</i> : administración federal para la aviación
FMARS	<i>Flashline Mars Artic Research Station</i> : estación para la investigación de Marte
FOV	<i>Field Of View</i> : campo de visión
FPS	<i>First Person Shooter</i> : videojuego de disparos en primera persona
GNU	<i>GNU's Not Unix</i> : gnu no es unix
GOTY	<i>Game Of The Year</i> : premio al videojuego del año
GUI	<i>Graphical User Interface</i> : Interfaz gráfica de usuario



HDRR	<i>High Dynamic Range Rendering</i>
HRI	<i>Human-Robot Interaction</i> : interacción humano-robot
IA	<i>Inteligencia Artificial</i>
ISS	<i>International Space Station</i> : estación espacial internacional
JPL	<i>Jet Propulsion Laboratory</i> : entidad responsable ante la NASA de la red DSN
MDRS	<i>Mars Desert Research Station</i> : estación para la investigación de Marte
MESCH	<i>Mars Environmental Simulation Chamber</i> : cámara de simulación de entorno marciano
MMORPG	<i>Massively Multiplayer Online Role-Playing Game</i>
NASA	<i>National Aeronautics and Space Administration</i> : administración nacional para la aeronáutica y el espacio.
NIST	<i>National Institute of Standards and Technology</i> : instituto nacional de estándares y tecnología.
PC	<i>Personal Computer</i> : ordenador personal
RPG	<i>Role-Playing Game</i>
SRBM	<i>Short-Range Ballistic Missile</i> : misil balístico de corto alcance
UDK	<i>Unreal Development Kit</i> : última versión del simulador USARSim
UE	<i>Unreal Engine</i> : motor del videojuego Unreal Tournament



- UI** *User Interface*: interfaz de usuario
- URSS** *Unión de Repúblicas Socialistas Soviéticas*
- USAR** *Urban Search And Rescue*: búsqueda y rescate urbano
- UT** *Unreal Tournament*: videojuego de disparos en primera persona
- UU** *Unreal Units*: unidad de medida en Unreal Tournament 2004