



Escuela
Politécnica
Superior

Entornos virtuales inmersivos para la detección y tratamiento de problemas de salud mental



Grado en Ingeniería Multimedia

Trabajo Fin de Grado

Autor:

Josué Pérez Villar

Tutor/es:

Mireia Luisa Sempere Tortosa

Septiembre 2017



Universitat d'Alacant
Universidad de Alicante

Justificación y Objetivos

Mi principal motivación para llevar a cabo este proyecto es la utilidad que pudiera tener en un futuro. Al tratarse de un trabajo que combina realidad virtual con psicología clínica creo que tiene un gran potencial, tanto en mi currículum personal como en todas aquellas personas que puedan llegar a beneficiarse del mismo.

Así mismo, también vi en este proyecto la posibilidad de demostrar y mejorar mis habilidades adquiridas durante la carrera, tanto a nivel de programación como a nivel de diseño y modelado tridimensional. Por otro lado, además ha supuesto un gran esfuerzo en tareas de documentación, las cuales me han hecho adquirir una experiencia que no poseía en este ámbito.

De todo lo mencionado se desprende la gran complejidad que posee el tema a desarrollar y lo mucho que me ha aportado como ingeniero multimedia. Y doy gracias por ello.

Por último, desglosaré los distintos objetivos de este trabajo de fin de grado:

- Investigar y comprender la realidad virtual.
- Investigar y comprender el tratamiento de fobias.
- Investigar y comprender el uso de la realidad virtual en el tratamiento de fobias.
- Modelar y texturizar un entorno geométrico tridimensional fiel a un aula de colegio.
- Crear un entorno virtual inmersivo lo suficientemente personalizable como para hacer viable su uso en psicología clínica.
- Incluir modelado de personajes en el entorno.
- Hacer que el entorno virtual sea usable mediante dispositivos de realidad virtual.

Agradecimientos

Estoy escribiendo este texto cuando "solamente" le he dedicado 70 horas al proyecto. Se podría decir que apenas he empezado a darle forma y ya tengo a mucha gente a la que agradecer su apoyo.

En primer lugar quiero darle las gracias a todos los profesores que, para bien o para mal, me han influenciado a lo largo de todos estos años, desde primaria hasta la universidad. Aunque lo cierto es que hay algunos a los que les haré una mención especial.

A Trini y a Isa, sois las mejores profesoras de primaria que he podido tener, vuestro entusiasmo siempre me inspiró. Pero en especial a Isa, con la que he seguido en contacto toda mi vida y sin la cual esto habría sido más complicado, pues fue ella la que me permitió acudir a una clase real de mi colegio (CP Teixereta de Ibi) para poder tomar todas las medidas necesarias para el proyecto.

A Laura, profesora de filosofía en el instituto y, sobretodo, amiga. Gracias por todas las veces que hemos quedado y por todos tus consejos. Tu cercanía siempre ha sido muy importante para mí.

A Carlos, Miguel Ángel, Fran y Mireia, profesores de la universidad. Aunque todos y cada uno de los profesores me han ayudado en todo el proceso de aprendizaje, creo que vosotros merecéis una mención especial por vuestra paciencia y apoyo durante estos años.

También quiero darle las gracias a mis amigos y compañeros de la vida. Estos años he tenido la suerte de rodearme de grandes personas con las que he podido contar siempre. A mis amigos de siempre, a mis amigos y compañeros de baile, a mis amigos y compañeros de teatro, a mis amigos y compañeros de la universidad, solo puedo deciros una cosa: gracias.

De entre estos últimos me gustaría destacar a Andrés y Clara, por todo este tiempo haciendo locuras y permitiéndome ser yo mismo. A Tino, por su paciencia infinita y por

su ayuda incondicional. Al resto de Crying Games, por acompañarme en uno de los periodos más intensos de mi vida y por haber sobrevivido a ello juntos. A Andrea, por ofrecerse a leer este documento y por sacarme fallos de redacción. Y, por último, a Ainoa, por escuchar mis frustraciones con este trabajo y por darme los ánimos que necesitaba.

Por otro lado, también me gustaría agradecerle a mi prima Mado por ayudarme a ver el lado más humano de mi trabajo. Su preparación profesional le hizo perfecta para indicarme aquellos detalles sobre psicología que más le llamaron la atención en mi texto.

Y, por último, a mi familia. Un pilar fundamental en todo este tiempo estudiando. Quiero darle las gracias a Sara, mi hermana, por toda la paciencia que ha tenido conmigo y por soportarme tan bien en mis años de adolescencia (y en los tuyos). Y a mis padres, que, con su sacrificio y esfuerzo aún cuando económicamente en casa no estábamos tan bien, han logrado que haya podido estudiar todos estos años y han conseguido que el año que viene pueda seguir estudiando un máster lejos de casa.

A todos y cada uno de los mencionados (y a todos los que se me olvida mencionar), gracias por estar ahí.

Dedicatoria

A todos los que dudaron... Como yo.

Citas

**"La realidad no me entusiasma,
pero sigue siendo el único lugar
donde se come decentemente"**

- Groucho Marx.

Índices

Índice de contenidos

Justificación y Objetivos	3
Agradecimientos	5
Dedicatoria	7
Citas	9
Índices	11
Índice de contenidos	11
Índice de figuras.....	19
Códigos QR.....	19
Ilustraciones.....	19
Índice de tablas.....	23
1. Introducción	27
2. Marco teórico.....	30
2.1. Realidad Virtual y Realidad Aumentada	30
2.1.1. Definición	30
2.1.2. Elementos de un sistema de Realidad Virtual.....	33
2.1.2.1. Dispositivos de entrada	34
2.1.2.2. Dispositivos de salida.....	35
2.1.2.3. Ordenador	35
2.1.3. Primeros sistemas	36
2.2. Realidad Virtual en Psicología Clínica. Ciberterapia.....	37
2.2.1. Trastornos de ansiedad	38
2.2.2. Imagen corporal.....	39
2.2.3. Adicciones	40

2.2.4. Dolor agudo	42
2.2.5. Trastornos psicóticos	43
2.3. Conceptos clave de la Realidad Virtual para su uso como tratamiento.....	44
2.3.1. Presencia.....	44
2.3.1.1. Dispositivos que propician la sensación de Presencia	45
2.3.1.1.1. Dispositivos visuales.....	45
2.3.1.1.2. Dispositivos sonoros	47
2.3.1.1.3. Dispositivos táctiles.....	47
2.3.1.1.4. Dispositivos olfativos.....	48
2.3.2. Interacción.....	48
2.3.2.1. Dispositivos que propician la Interacción.....	50
2.3.2.1.1. Dispositivos de control	50
2.3.2.1.2. Dispositivos de localización (trackers)	50
2.4. Primeros usos de la Realidad Virtual en Psicología clínica.....	51
2.4.1. 1982	51
2.4.2. 1992.....	52
2.4.3. 1993	52
2.4.4. 1994.....	53
2.4.5. 1995	54
2.4.6. 1998.....	54
2.5. Estado actual de la Realidad Virtual en el campo de la salud mental.....	54
2.5.1. Ventajas	55
2.5.1.1. Cuestiones económicas.....	56
2.5.1.2. Cuestiones de tiempo	56
2.5.1.3. Presentación de situaciones en cualquier momento.....	56
2.5.1.4. Repeticiones.....	57

2.5.1.5. Simulación de situaciones del mundo real.....	57
2.5.1.6. Experiencias similares a la vida real.....	58
2.5.1.7. Seguridad del entorno virtual.....	58
2.5.1.8. Control sobre las escenas presentes.....	59
2.5.1.9. Confidencialidad.....	60
2.5.1.10. Cuestiones estadísticas	60
2.5.2. Desventajas.....	60
2.5.2.1. Incompatibilidad.....	61
2.5.2.2. Posible "enfermedad del simulador"	61
2.5.2.3. Coste económico.....	62
2.6. Tecnologías necesarias para la realización del entorno	63
2.6.1. Creación de los modelos tridimensionales	63
2.6.1.1. Conceptos matemáticos	64
2.6.1.2. Modelado 3D	65
2.6.1.3. Texturizado.....	68
2.6.1.3.1. Mapeado UV.....	68
2.6.1.3.2. Baking.....	70
2.6.1.3.3. Pintado de texturas.....	71
2.6.1.4. Materiales	72
2.6.2. Unity	73
2.6.2.1. Características y ventajas generales.....	73
2.6.2.2. Características y ventajas del Editor de Unity	75
2.6.2.2.1. Configuración básica.....	75
2.6.2.2.2. Desarrollando la aplicación	76
2.6.2.3. Características y ventajas del uso de Scripts.....	80
2.6.3. Dispositivos visuales de Realidad Virtual	81

2.6.3.1. Clasificación	81
2.6.3.2. Características y especificaciones	84
2.6.3.2.1. Calidad de la imagen	84
2.6.3.2.2. Sensores.....	85
2.6.3.2.3. Efecto rejilla.....	85
2.6.3.3. Otros datos de interés	86
2.6.3.3.1. Seguridad para los ojos	86
2.6.3.3.2. Edad mínima recomendada	87
2.6.3.3.3. Preparativos para el uso del dispositivo.....	87
2.6.3.4. Samsung Gear VR.....	88
2.6.3.4.1. Ventajas	89
2.6.3.4.2. Desventajas	89
3. Objetivos	91
4. Metodología.....	93
4.1. Documentación.....	94
4.2. Creación de los elementos del entorno virtual.....	95
4.2.1. Modelado	96
4.2.2. Texturizado	98
4.2.3. Personajes.....	101
4.3. Creación del entorno virtual.....	105
4.3.1. Editor de Unity	105
4.3.2. Interfaz de usuario.....	109
4.3.3. Código	112
4.3.3.1. Colocación de elementos	112
4.3.3.2. Adaptado del código para los parámetros ideados.....	112
4.3.3.3. Navegación por el menú	113

4.3.3.4. Sistema de carga/guardado	115
4.3.3.4.1. Escritura en el fichero de carga	115
4.3.3.4.2. Escritura en el fichero de guardado	116
4.3.3.4.3. Lectura del fichero de carga.....	117
4.3.3.4.4. Lectura del fichero de guardado	117
4.3.4. Samsung Gear VR.....	119
4.4. Testeo y solución de errores	120
5. Cuerpo del trabajo.....	123
5.1. Resultados de la documentación.....	123
5.1.1. Toma de medidas y fotos de referencia	124
5.1.2. Diseño de las funcionalidades y del menú de personalización.....	126
5.1.2.1. Familiaridad	127
5.1.2.1.1. Profesorado	128
5.1.2.1.2. Paciente	128
5.1.2.1.3. Resto de alumnos	129
5.1.2.2. Fobias de carácter social	129
5.1.2.2.1. Distribución de las mesas	129
5.1.2.2.2. Elementos diferenciadores.....	130
5.1.2.3. Claustrofobia	131
5.1.2.4. Cuestiones sobre el género y orden.....	132
5.1.2.5. Repetición de sesiones	133
5.2. Resultados de la creación de los elementos del entorno	133
5.2.1. Modelados tridimensionales.....	133
5.2.1.1. Mesa del alumno	134
5.2.1.2. Silla del alumno.....	136
5.2.1.3. Estructura de la clase	138

5.2.1.4. Otros modelados	140
5.2.2. Mapas UV y texturas	142
5.2.3. Personajes	145
5.3. Resultados de la creación del entorno	149
5.3.1. Creación del proyecto y configuración inicial	149
5.3.2. Importación y colocación manual de los modelos tridimensionales	151
5.3.3. Interfaz de usuario.....	158
5.3.4. Scripts	162
5.3.4.1. AlumnosController	162
5.3.4.2. ApplicationController	164
5.3.4.3. ButtonController	166
5.3.4.4. CameraController	167
5.3.4.5. CargadorSesiones.....	168
5.3.4.6. ClaseCreator	169
5.3.4.7. Guardado.....	170
5.3.4.8. ImagenesController	173
5.3.4.9. LectorCreador	174
5.3.4.10. MenuController.....	175
5.3.4.11. MesasCreator.....	178
5.3.4.12. MobiliarioController	181
5.3.4.13. OpcionController	182
5.3.4.14. PacienteController.....	183
5.3.4.15. TeacherController	184
5.4. Últimas mejoras del entorno.....	185
6. Conclusiones.....	192
6.1. Posibles mejoras	193

7. Bibliografía y referencias	195
Anexo 1: Enlaces de descarga de la aplicación para móviles (sin RV) y ordenador	199
Enlace para dispositivos móvil Android	199
Enlace para ordenadores	199
Anexo 2: Video y capturas de la aplicación en funcionamiento.....	200
Vídeo	200
Capturas.....	200

Índice de figuras

Códigos QR

Código QR 1. Unity UI Tutorial - How to make a scrollable list.....	111
Código QR 2. Enlace a la imagen de fondo de la aplicación.....	158
Código QR 3. Enlace para descargar la apk de la versión móvil sin realidad virtual.	199
Código QR 4. Enlace para descargar un comprimido con la versión de ordenador.....	199
Código QR 5. Enlace al video demostrativo del entorno en ejecución.	200

Ilustraciones

Ilustración 1. Ejemplo de Realidad Aumentada.	32
Ilustración 2. Ejemplo de dispositivos de entrada.....	34
Ilustración 3. Ejemplo de dispositivos de salida.....	35
Ilustración 4. Realidad virtual para tratar adicciones.	41
Ilustración 5. Uso de realidad virtual para pacientes con quemaduras.....	43
Ilustración 6. CAVE.	46
Ilustración 7. Guantes de Realidad virtual.....	47
Ilustración 8. Diferencia entre vértice, segmento y cara.....	65
Ilustración 9. Escáner 3D.	66
Ilustración 10. Ejemplo de esculpido 3D.	67
Ilustración 11. Ejemplo de mapeado UV.....	69
Ilustración 12. Ejemplo de pintado de texturas.	71
Ilustración 13. Logo de Unity.....	74
Ilustración 14. Ventana "bulb settings" de Unity.....	76
Ilustración 15. Ejemplo de GameObject y sus componentes.	78
Ilustración 16. Dispositivo monocular.....	82
Ilustración 17. Ejemplo de dispositivo "encadenado".	83
Ilustración 18. Samsung Gear VR.	88
Ilustración 19. Modelado de una silla de alumnos.....	97
Ilustración 20. Modelado de muebles y sus objetos.	98
Ilustración 21. Modelado de un borrador junto con su mapeado UV.....	99
Ilustración 22. Interfaz del programa Substance Painter.	100

Ilustración 23. Materiales por defecto del programa Substance Painter.....	100
Ilustración 24. Interfaz del programa MakeHuman.....	102
Ilustración 25. Modelado de alumna junto con las opciones de geometría disponibles para el pelo.....	104
Ilustración 26. Configuración de los materiales en Unity.....	106
Ilustración 27. Anotaciones de las posiciones y rotación de los elementos del entorno....	107
Ilustración 28. Anotación de las posiciones y rotación de mesas y sillas para el caso de distribución individual.....	108
Ilustración 29. Jerarquía de la aplicación, sus grupos y sus GameObjects vacíos.....	109
Ilustración 30. Modelados del entorno que se usaron como fondo para el menú.	110
Ilustración 31. Pantalla principal del menú de la aplicación.	111
Ilustración 32. Pantalla principal del menú de la aplicación con información sobre la distribución de mesas.	114
Ilustración 33. Pantalla de carga de la aplicación con botones a modo de ejemplo.....	118
Ilustración 34. Captura de SDK Manager.	119
Ilustración 35. Objetos y muebles antes del cambio del número de polígonos.....	121
Ilustración 36. Objetos y muebles después del cambio del número de polígonos.....	122
Ilustración 37. Fotografía real de una clase de primaria.	124
Ilustración 38. Bocetos y medidas de la pizarra y el borrador.....	125
Ilustración 39. Boceto y medidas de una silla del alumnado.....	125
Ilustración 40. Bocetos y medidas de la silla del profesorado y de los percheros.....	126
Ilustración 41. Distribuciones en forma de U e individuales de las mesas del entorno. ...	130
Ilustración 42. Fotografía de mesa del alumno.....	134
Ilustración 43. Boceto y medidas de una silla del alumnado.....	135
Ilustración 44. Modelado de mesa del alumnado con la rejilla por separado.	136
Ilustración 45. Fotografía de una silla del alumnado.	136
Ilustración 46. Vistas superior e inferior del modelado del asiento de las sillas del alumnado.....	137
Ilustración 47. Modelado compacto de la clase.	138
Ilustración 48. Modelado de la clase por piezas.	139
Ilustración 49. Modelado del borrador.....	140

Ilustración 50. Modelado de la lámpara.....	140
Ilustración 51. Modelado del mobiliario con material escolar dentro.	141
Ilustración 52. Modelado de la silla del profesorado.	141
Ilustración 53. Mapeado UV de la parte de madera de un borrador junto con su posterior pintado de texturas.	142
Ilustración 54. Mapeado UV del asiento de la silla del profesorado junto con su posterior pintado de texturas.	143
Ilustración 55. Fotografía de un aula de colegio real.....	144
Ilustración 56. Mapeado UV de la punta de un lápiz junto con su posterior pintado de texturas.	145
Ilustración 57. Modelado de la profesora junto con las opciones de materiales para simular piel.....	146
Ilustración 58. Modelado del profesor junto con las opciones de poses.	147
Ilustración 59. Modelados de los alumnos.....	148
Ilustración 60. Captura de todos los modelados de personajes visualizados en Unity.	149
Ilustración 62. Aspect ratio de la pestaña Game de Unity.....	150
Ilustración 61. Player settings del proyecto.	150
Ilustración 63. Organización de las carpetas del proyecto.....	151
Ilustración 64. Capturas del primer montaje del entorno aún sin la iluminación.....	152
Ilustración 65. Anotaciones de las mesas para distribución individual.....	153
Ilustración 66. Anotaciones de las mesas para distribución en parejas.	153
Ilustración 67. Anotaciones de las mesas para distribución en tríos.....	154
Ilustración 68. Anotaciones de las mesas para distribución en forma de U.	154
Ilustración 69. Anotaciones de las mesas para distribución en grupos.	155
Ilustración 70. Jerarquía existente entre los GameObjects de la aplicación.	157
Ilustración 71. Escena en Unity con luces y fondo.....	157
Ilustración 72. Modelados del entorno que se usaron como fondo para el menú.	159
Ilustración 73. Primera pantalla del menú.	159
Ilustración 74. Pantalla principal de personalización del menú.....	160
Ilustración 75. Pantalla secundaria de personalización del menú.....	160
Ilustración 76. Pantalla de carga del menú.	161

Ilustración 77. Pantalla de guardado del menú.	161
Ilustración 78. Código para realizar fundidos en negro.	178
Ilustración 79. Ejemplo de Destroys en el código.	186
Ilustración 80. Ejemplo de eliminado de GameObjects vacíos.	186
Ilustración 81. Mesa del alumnado antes del cambio del número de polígonos.	187
Ilustración 82. Mesa del alumnado después del cambio del número de polígonos.	188
Ilustración 83. Perchero antes del cambio del número de polígonos.	188
Ilustración 84. Perchero después del cambio del número de polígonos.	189
Ilustración 85. Silla del alumnado antes (izquierda) y después (derecha) del cambio del número de polígonos.	189
Ilustración 86. Objetos y muebles antes del cambio del número de polígonos.	190
Ilustración 87. Objetos y muebles después del cambio del número de polígonos.	190
Ilustración 88. Icono de la aplicación.	191
Ilustración 89. Vista general de la clase generada con los valores por defecto del menú.	200
Ilustración 90. Casillero y estanterías con sus objetos por defecto.	201
Ilustración 91. Pizarras, puerta y papelera del entorno.	201
Ilustración 92. Otra vista de la clase generada con los valores por defecto del menú.	202
Ilustración 93. Ejemplo 1 de pantalla principal con algunos valores ya seleccionados.	202
Ilustración 94. Ejemplo 2 de pantalla principal con algunos valores ya seleccionados.	203
Ilustración 95. Clase generada en base a los valores seleccionados en los ejemplos anteriores.	203
Ilustración 96. Ejemplo de pantalla de guardado.	204
Ilustración 97. Ejemplo de pantalla de carga mostrando la sesión guardada en el ejemplo anterior.	204

Índice de tablas

Tabla 1. Relación de tareas principales y sus tiempos estimados	93
Tabla 2. Información general AlumnosController.....	162
Tabla 3. Variables AlumnosController	163
Tabla 4. Funciones AlumnosController.....	164
Tabla 5. Información general ApplicationController.....	164
Tabla 6. Variables ApplicationController	165
Tabla 7. Funciones ApplicationController.....	166
Tabla 8. Información general ButtonController.....	166
Tabla 9. Funciones ButtonController.....	166
Tabla 10. Información general CameraController	167
Tabla 11. Variables CameraController.....	167
Tabla 12. Funciones CameraController	167
Tabla 13. Información general CargadorSesiones	168
Tabla 14. Variables CargadorSesiones.....	168
Tabla 15. Funciones CargadorSesiones	169
Tabla 16. Información general ClaseCreator	169
Tabla 17. Variables ClaseCreator	170
Tabla 18. Funciones ClaseCreator.....	170
Tabla 19. Información general Guardado	171
Tabla 20. Variables Guardado.....	172
Tabla 21. Funciones Guardado.....	172
Tabla 22. Información general ImagenesController	173
Tabla 23. Variables ImagenesController.....	173
Tabla 24. Funciones ImagenesController	174
Tabla 25. Información general LectorCreador	174
Tabla 26. Funciones LectorCreador	174
Tabla 27. Información general MenuController	175
Tabla 28. Variables MenuController.....	176
Tabla 29. Funciones MenuController	177
Tabla 30. Información general MesasCreator	178

Tabla 31. Variables MesasCreator.....	179
Tabla 32. Funciones MesasCreator	180
Tabla 33. Información general MobiliarioController.....	181
Tabla 34. Variables MobiliarioController.....	181
Tabla 35. Funciones MobiliarioController.....	182
Tabla 36. Información general OpcionController.....	182
Tabla 37. Variables OpcionController	183
Tabla 38. Funciones OpcionController.....	183
Tabla 39. Información general PacienteController	183
Tabla 40. Variables PacienteController.....	184
Tabla 41. Funciones PacienteController	184
Tabla 42. Información general TeacherController.....	184
Tabla 43. Variables TeacherController	185
Tabla 44. Funciones TeacherController.....	185
Tabla 45. Relación de tareas principales y el tiempo invertido en cada una.....	193

1. Introducción

El auge de la informática y de las nuevas tecnologías es un hecho y podríamos afirmar, que están mejorando nuestra calidad de vida (Cristina Botella, 2007, pág. 33). Cada año se logra aumentar la velocidad de los procesadores, la capacidad de las memorias y los recursos gráficos, que hace años requerían equipos muy costosos, hoy están al alcance de cualquiera. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 1)

Las tecnologías de la información llegan ya a prácticamente todos los ámbitos de nuestras vidas y, como es de esperar, también han alcanzado muchos campos científicos, siendo el de la psicología clínica uno de ellos. (Cristina Botella, 2007, pág. 33)

En este último campo, ya se han desarrollado sistemas que ayudan a los profesionales de la salud en sus complejas tareas de detectar y tratar problemas de índole mental y en lo que se plantea la utilización de muy distintas herramientas para dar solución a diversas patologías. Una de las tecnologías que más frutos está dando es la realidad virtual y el uso de entornos virtuales inmersivos. (Cristina Botella, 2007, pág. 33)

Aunque parezca una herramienta muy novedosa, estamos acostumbrados a encontrarnos con entornos inmersivos en nuestra realidad cotidiana. El teatro, el cine, la televisión o incluso la lectura son ambientes o actividades que disminuyen la sensación de presencia en el entorno real y nos permiten sentirnos en otra parte. Pero lo cierto es que la realidad virtual posee un distintivo muy poderoso con respecto a estos "sistemas inmersivos": la interacción. El sujeto ya no se limita a observar, sino que sus acciones pueden afectar el entorno de diferentes maneras. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 1)

Dicho esto, queda evidenciado que el gran atractivo que la realidad virtual ofrece al campo de la psicología clínica son estas dos características: la sensación de presencia y la interactividad. Gracias a estas, se puede tratar a un paciente con realidad virtual exponiéndole de forma controlada a situaciones que pueden provocarle la aparición de ansiedad, teniendo el especialista en todo momento el control sobre la simulación.

En general, se realizan un determinado número de sesiones de realidad virtual en las que se sitúa al paciente, gradualmente, en situaciones de mayor riesgo de ansiedad. Así, gracias a la sensación de presencia y a la interactividad, el paciente logra habituarse de forma progresiva hasta conseguir ser capaz de enfrentarse a dichas situaciones en su vida real. (Maldonado, La realidad virtual es útil, 2010, pág. 1)

Como se puede deducir de lo anteriormente explicado, el uso creciente de las tecnologías de la información ha permitido estimular un cambio en la manera de suministrar los servicios de salud y bienestar social. La realidad virtual ha sido un elemento clave en todo este proceso de cambio, proliferando su uso en diferentes áreas de la psicología clínica y la medicina. Pero como todo cambio, conlleva un tiempo. (Ornedo, 2009, pág. 2)

Las aplicaciones tecnológicas en el ámbito de la salud requieren al menos de 15 años para alcanzar un uso generalizado tras haberse demostrado su eficacia en laboratorio. En el caso de la realidad virtual en psicología, donde los primeros experimentos con cierto rigor científico en laboratorio se realizaron en el 1992, es ahora cuando se empieza a detectar cierto interés a nivel científico y social. Su uso aún no es masivo en hospitales y centros de salud, pero con el continuo abaratamiento de la tecnología durante los años venideros crecerá de forma exponencial su utilización. (Maldonado, La realidad virtual es útil, 2010, pág. 1)

En conclusión, a medida que avanzan los años, los desarrollos tecnológicos incrementan. Los ordenadores llegarán a ser todavía más pequeños, más rápidos y más baratos. Los programas de software serán más sofisticados e inteligentes. Las tecnologías de la información y la comunicación se adentrarán cada vez más en nuestra realidad cotidiana y nos resultarán imprescindibles en la nueva vida moderna. (Cristina Botella, 2007, pág. 34)

Sin lugar a dudas, la interacción hombre-máquina seguirá avanzando, cambiando. Aparecerá en todas las áreas de nuestra vida y la psicología clínica será una de las grandes beneficiadas. (Cristina Botella, 2007, pág. 24)

Queda, por lo tanto, más que evidenciada la relevancia y vigencia actual del tema a tratar en este proyecto: los entornos virtuales inmersivos para la detección y tratamiento de problemas de salud mental.

El propósito de este proyecto es, como acabo de adelantar, la creación de un entorno virtual concreto (un aula de colegio) con el fin de utilizarlo como herramienta para tratar problemas de salud mental en niños, más concretamente: fobias. Con la realización del sistema de realidad virtual pretendo crear una herramienta útil, con alto grado de personalización que permitiera a un especialista realizar un tratamiento flexible sobre pacientes de pocos años de edad.

Para lograr la creación del sistema de realidad virtual descrito ha sido necesario combinar varias tecnologías de la información, así como una ardua tarea de documentación que me permitiera aplicarlas de forma correcta y objetiva.

En el ámbito del modelado tridimensional, en este trabajo de fin de grado se han realizado 27 modelos, entre los que se incluyen la estructura misma de la clase así como de todo el mobiliario necesario para representarla. Con el objetivo de crear una clase de colegio lo más fiel a la realidad posible, también se ha realizado el mapeado y texturizado de forma personalizada de cada uno de los elementos del entorno. Además, se han diseñado 4 personajes, dos de ellos con 5 variaciones distintas, logrando así poder poblar el entorno con una gran variedad de personas.

Por último, en la parte de programación, cabría destacar el alto grado de complejidad que requería el menú de la aplicación, pues debía contar con las suficientes variables como para lograr que el terapeuta pudiera realizar tratamientos flexibles. Así mismo, también se ha programado un sencillo sistema de cargado y guardado de sesiones para facilitar el trabajo de repetición dentro de un tratamiento.

2. Marco teórico

2.1. Realidad Virtual y Realidad Aumentada

2.1.1. Definición

Si se tuviera que definir la realidad virtual en un único párrafo una definición más o menos correcta sería esta:

"La Realidad Virtual es la tecnología informática que genera entornos virtuales tridimensionales con los que el sujeto interactúa en tiempo real, logrando causarle una sensación de inmersión semejante a la presencia en el mundo real." (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 2)

Pero no se puede definir algo tan extenso y complejo como la realidad virtual en un único párrafo sin que se pierdan ciertos matices; su alcance y sus posibilidades son tan extensas que necesitaríamos hojas y hojas para enumerarlas. Dado que el tema del trabajo de fin de grado no es ese, me limitaré a simplemente a desglosar y completar la definición que he dado anteriormente.

La realidad virtual se puede definir, pues, como la forma más avanzada de relación entre el ordenador y la persona, permitiendo al usuario interactuar con el ordenador y sumergirse en un entorno generado artificialmente. Esta tecnología se basa en la generación interactiva multisensorial de estímulos con el objetivo de mantener la sensación completa de inmersión en un mundo real, permitiendo además que el usuario sienta que puede interactuar y participar en el entorno sintético (Ornedo, 2009, pág. 3). Cabe destacar que tales entornos no tienen por qué limitarse a la reproducción más o menos fiel del mundo real, sino que se pueden construir entornos sintéticos mediante los que se manipulan determinadas propiedades de manera que se obtienen resultados improbables o imposibles en la realidad y no por ello causan menos sensación de

presencia. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 2)

Por lo tanto, se deduce que el principal objetivo de la realidad virtual es la generación del sentido de presencia en el usuario, es decir, la experiencia subjetiva de estar en un lugar, incluso cuando físicamente se está localizado en otro (Ornedo, 2009, pág. 3). Los usuarios ya no son simplemente observadores externos de imágenes, sino que son participantes activos en el mundo virtual que genera y gestiona un ordenador. (Torres, 2012, pág. 81)

De esto último se desprende una idea vital para entender esta tecnología y es que la realidad virtual implica el uso de ordenadores tanto para su desarrollo como para su aplicación. (Torres, 2012, pág. 81)

Es por esto que algunos autores han subrayado la importancia de incluir en la definición de realidad virtual los dos factores que, según su interpretación, serían centrales para poder llegar a comprender qué es la realidad virtual: la tecnología empleada y la persona. En efecto, ambos elementos serían necesarios para que se produzca una experiencia virtual completa. Es decir, para que la persona llegue a percibir los estímulos virtuales como reales y, por lo tanto, para que se genere en ella una sensación de presencia, es necesario que el ordenador o el software encargado de generar el mundo virtual y de producir la estimulación de los sentidos funcione correctamente y a la mayor velocidad posible, siendo lo más indicado que pueda funcionar en tiempo real. (Torres, 2012, págs. 82-83) (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 3)

Actualmente, el campo ha crecido de forma exponencial y se ha producido una enorme expansión de la tecnología, aunque hoy en día sigue sin estar al alcance de todo el mundo. Pero, como ha ocurrido con los ordenadores de sobremesa, en la medida en que los desarrollos tecnológicos avancen y los costes disminuyan, la realidad virtual estará disponible en todos los aspectos de nuestra vida: en casa, en el trabajo, en el hospital, etc. Dentro de unos años la realidad virtual estará tan presente en nuestra realidad cotidiana

que no seremos capaces de imaginarnos un mundo moderno sin ella. (Cristina Botella, 2007, pág. 33)

Pero la realidad virtual no es la única tecnología de gran utilidad en la psicología clínica: la realidad aumentada también ha demostrado ser muy eficaz en tratamientos de fobias y, por lo tanto, también debe ser incluida en este trabajo.

La realidad aumentada, por el contrario, es la mezcla de información generada por ordenador con el mundo real. En una definición clásica, la realidad aumentada es un tipo de ambiente virtual en el cual el usuario no se sumerge completamente en un mundo virtual, sino en una mezcla de éste con el mundo real, de tal forma que, para el usuario, aparezcan los objetos virtuales y reales coexistiendo en el mismo espacio. (Ornedo, 2009, pág. 3)



Ilustración 1. Ejemplo de Realidad Aumentada.

Fuente: <https://upload.wikimedia.org/wikipedia/commons/5/5f/Augmented-reality.jpg>

El aspecto central de la realidad aumentada es que los elementos virtuales que se superponen en el mundo real proporcionan información adicional y relevante a la imagen final que está visualizando el usuario del sistema con el objetivo de ayudarlo. En resumen, la premisa central de esta tecnología es que la información adicional que aparece en el

mundo real y a la que tiene acceso el usuario sean de utilidad para él. (Cristina Botella, 2007, pág. 33)

Viendo ambas definiciones se hace evidente que existen ciertas diferencias entre la realidad virtual y la realidad aumentada, siendo la más importante el grado de inmersión del usuario en el sistema. Mientras que un sistema de realidad virtual envuelve completamente al usuario (la visión y otros canales perceptivos, como la audición, el olfato y el tacto, están controlados totalmente por el ordenador), en uno de realidad aumentada lo que se hace es completar la información del mundo real sin llegar a suprimirlo (el usuario percibe un mundo real especial enriquecido por la información adicional existente en los elementos virtuales que proporciona el sistema). En conclusión, mientras que en la realidad virtual el usuario se siente presente en el entorno virtual, en la realidad aumentada el usuario se sigue sintiendo presente en el mundo real. (Cristina Botella, 2007, pág. 3)

2.1.2. Elementos de un sistema de Realidad Virtual

Una vez explicadas brevemente las dos tecnologías, sus objetivos y sus diferencias, llega la hora de enumerar y definir los elementos necesarios para que tales tecnologías puedan llegar a cumplir sus propósitos. Tales elementos se pueden separar entre: dispositivos de entrada y dispositivos de salida.

Como su propio nombre indica, mientras que los dispositivos de entrada se encargan de capturar la información proveniente del usuario y del mundo real para "entrarla" en el ordenador, los dispositivos de salida lo que hacen es transmitirle al usuario la información que se genera y "sale" del ordenador.

Además de tales dispositivos, otro elemento clave en la realidad virtual es el ordenador, sin el cual sería imposible generar el entorno.

2.1.2.1. Dispositivos de entrada



Ilustración 2. Ejemplo de dispositivos de entrada.

Fuente: https://upload.wikimedia.org/wikipedia/commons/7/7f/Oculus_Touch_Controller.jpg

Como ya se ha mencionado anteriormente, los dispositivos de entrada son los que permiten que el entorno virtual reciba los datos del usuario. Esta clase de dispositivos se pueden dividir en dos grandes ramas: los dispositivos de control y los dispositivos de localización o trackers. Mientras que los primeros se encargan de transmitir las órdenes del usuario al sistema, los segundos son lo que utiliza el sistema para obtener información del entorno, por ejemplo, la posición 3D del usuario. (Ornedo, 2009, pág. 4)

2.1.2.2. Dispositivos de salida



Ilustración 3. Ejemplo de dispositivos de salida.

Fuente: http://i.vimeocdn.com/video/566624591_1280x720.jpg

En contraposición a los dispositivos de entrada están los de salida, que proporcionan al usuario información sobre el mundo y provocan en él determinadas sensaciones que pueden ser visuales, auditivas o táctiles. Se utilizan en realidad virtual dispositivos gráficos o visuales como los cascos de realidad virtual, o *Head mounted displays*, y los sistemas de proyección de una o varias pantallas. También se usan dispositivos de sonido como altavoces y auriculares y/o dispositivos de realimentación táctil y de fuerza como los guantes de realidad virtual. (Ornedo, 2009, pág. 5)

2.1.2.3. Ordenador

Poco se puede decir del ordenador que nadie conozca. En un sistema de realidad virtual el ordenador es la pieza clave que soporta el sistema y que realiza tareas de simulación y control de entrada/salida de datos (Ornedo, 2009, pág. 5). Además, el

ordenador no sólo cumple funciones de simulación y gestión, sino que también son los encargados de dar soporte al desarrollo de las distintas aplicaciones de realidad virtual y realidad aumentada.

2.1.3. Primeros sistemas

Para entender completamente una tecnología no sólo hay que saber su funcionamiento, sus elementos y sus objetivos, sino que es de vital importancia conocer sus orígenes.

Cuando se buscan en la historia antecedentes de la realidad virtual actual, se suele retroceder al año 1963, cuando Sutherland puso de manifiesto en su tesis doctoral "Sketchpad: a man-machine graphical communication system" que era posible utilizar ordenadores para la realización de gráficos interactivos. Poco después, en 1968, publicó "A head-mounted three dimensional display", con el que sentaría las bases que guiarían el desarrollo de los actuales cascos de realidad virtual (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 4).

Los primeros sistemas de RV se diseñaron para grandes industrias, siendo el propósito fundamental de las cuales crear escenarios que simularan determinadas situaciones en las que el personal especializado pudiera entrenarse (Cristina Botella, 2007, pág. 33). Por lo tanto, no es de extrañar que los primeros cascos de realidad virtual, que utilizaban sistemas que proyectaban ante los ojos la señal que recibían desde cámaras, fueran desarrollados en los sesenta por la compañía *Bell Helicopter*. Ésta ideó un casco conectado a una cámara de visión nocturna que permitía al piloto aterrizar en la oscuridad. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 5)

En 1966, Ivan Sutherland convirtió el sistema de visión remota antes descrito en un sistema de realidad virtual reemplazando las imágenes que provenían de la cámara por imágenes generadas por un ordenador. Este primer escenario virtual consistía en un sencillo cubículo de cuatro paredes con los puntos cardinales anotados sobre ellas. El

sujeto podía "entrar" en la habitación, mover la cabeza y ver las diferentes direcciones. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 5)

Las aportaciones de Sutherland sentaron las bases para el desarrollo de la informática gráfica, hasta llegar a la actual realidad virtual. Aunque hay que destacar que esta expresión no fue propuesta hasta la década de los ochenta, cuando Jaron Lanier, fundador de VPL Research, Inc., la primera compañía que vendió gafas y guantes de realidad virtual, la acuñó. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 5)

2.2. Realidad Virtual en Psicología Clínica. Ciberterapia.

La ciberterapia o tratamiento con realidad virtual se puede definir, de una forma más o menos técnica, como "el uso de ordenadores como herramientas para posibilitar y/o mejorar el suministro de servicios terapéuticos". (Ornedo, 2009, pág. 6)

En el mundo de la psicología clínica, la realidad virtual se suele emplear sobre todo para el tratamiento de fobias y de ansiedad (Ornedo, 2009, pág. 6), pero eso no impide que su uso en otro tipo de trastornos, como la deformación de la imagen corporal, las adicciones o el autismo, sea igual de eficaz.

Centrándonos en el mundo de las fobias, desde principios de los noventa, cuando Hodges et al. lograron tratar con éxito a pacientes acrofóbicos con realidad virtual, se ha propuesto a la realidad virtual como un nuevo medio de terapia. Desde entonces, se ha demostrado que el uso de esta tecnología es tan eficaz como la terapia 'in vivo' para el tratamiento de acrofobia, aracnofobia, agorafobia, claustrofobia y miedo a volar, entre otras. (Ornedo, 2009, pág. 6)

Curiosamente, uno de los resultados obtenidos en el tratamiento de la claustrofobia consiste en que los logros terapéuticos conseguidos por algunos pacientes se generalizaban a otras situaciones que no habían sido específicamente tratadas y que

pertenecían a otro problema que padecían, como, por ejemplo, la agorafobia. (Cristina Botella, 2007, pág. 35)

Otra de las aplicaciones de la ciberterapia en psicología clínica es el tratamiento del estrés postraumático, ya sea para veteranos de guerra o para víctimas de la violencia de género. (Ornedo, 2009, pág. 6)

En conclusión, la ciberterapia ha estimulado un cambio en la manera de suministrar los servicios de la salud y bienestar social, dando lugar a un proceso global distribuido en el cual la comunicación y colaboración de usuarios geográficamente dispersos (pacientes y clínicos) juegan un papel clave.

A continuación se explicarán brevemente las principales aplicaciones de la realidad virtual, pero, antes de hacerlo, es necesario mencionar que la realidad virtual también está demostrando ser altamente eficaz en otro tipo de trastornos como lo son el déficit de atención, el autismo o el acoso escolar.

2.2.1. Trastornos de ansiedad

Tradicionalmente para tratar los trastornos de ansiedad se usaba la técnica de la exposición, ya fuera en vivo o mediante la imaginación, la cual consiste en afrontar de forma gradual y sistemática las situaciones o actividades que los pacientes temen y evitan como consecuencia de un problema de ansiedad. (Cristina Botella, 2007, pág. 34)

Pese a los buenos resultados de esta técnica, cerca del 25% de los pacientes rechaza la exposición o abandona el tratamiento (Cristina Botella, 2007, pág. 34), ya sea por el temor a la propia exposición o por la falta de abstracción suficiente como para que el uso de la imaginación consiguiera el efecto deseado.

Por ello, el uso de realidad virtual se hizo esencial, pues permite simular la realidad de forma que el usuario se encuentra en un espacio tridimensional en el que puede interactuar con los elementos virtuales de una forma similar a como interactúa con

los elementos reales. Es más, el hecho de que las situaciones no sean «reales» puede favorecer que los pacientes acepten mejor la exposición. (Cristina Botella, 2007, pág. 34)

Con estos datos se hace evidente que las aplicaciones desarrolladas en éste ámbito se hayan focalizado en conseguir sumergir a la persona en un ambiente de realidad virtual similar al de la vida real en donde ésta es expuesta de manera gradual y controlada a aquellos estímulos o situaciones que desencadenan las respuestas fóbicas de evitación y ansiedad. (Torres, 2012, pág. 99)

En consecuencia, con el uso de estas aplicaciones se han conseguido asombrosos resultados, donde la exposición ha reducido la intensidad e incluso ha eliminado los estados emocionales y fisiológicos asociados como consecuencia de la habituación de la persona a dichas situaciones amenazantes evitadas hasta el momento. (Torres, 2012, pág. 99)

2.2.2. Imagen corporal

Otra área de aplicación de la realidad virtual es el uso de esta herramienta para el tratamiento de los trastornos alimentarios y, más específicamente, para la deformación de la imagen corporal. Esta tecnología constituye una herramienta que, por sus características, se ha mostrado particularmente útil a la hora de trabajar la distorsión de la imagen corporal que habitualmente presentan las personas que sufren de anorexia o bulimia nerviosa. (Cristina Botella, 2007, pág. 35)

En primer lugar, una gran aportación de esta herramienta es la capacidad de concretar y mostrar al paciente y al terapeuta este estado mental difícilmente objetivable. Mediante esta tecnología es posible representar una figura corporal en tres dimensiones en un entorno inmersivo y la persona siente que «está ahí» moldeando su propio cuerpo y enfrentándose a él. Además, las aplicaciones de realidad virtual desarrolladas permiten evaluar tanto el cuerpo en su totalidad como solo alguna zona del mismo. (Cristina Botella, 2007, pág. 35)

La gran ventaja del uso de realidad virtual para curar este trastorno es que los pacientes se muestran menos reacios a afrontar sus miedos en un ambiente virtual seguro, pues ven al ordenador como un juez imparcial y aceptan en mayor medida sus alteraciones en la imagen corporal. (Cristina Botella, 2007, págs. 35-36)

Llegados a este punto, cabe decir que la imagen corporal se ha evaluado recientemente comparando la exposición a dichos estímulos en tres condiciones experimentales: un mundo virtual, la vida real y a través de fotografías en personas diagnosticadas de anorexia y bulimia nerviosa comparado con un grupo de personas sin trastorno. (Torres, 2012, pág. 147)

Al analizar los resultados se observó que tanto el grupo de anorexia como el de bulimia experimentó niveles de ansiedad significativamente más altos que los experimentados por el grupo control, además también se mostraron significativamente más ansiosos cuando fueron expuestos a comida real y virtual comparado con la exposición a través de fotografías. (Torres, 2012, págs. 147-148)

Por tanto, se puede concluir que para personas con diagnóstico de anorexia o bulimia la exposición a estímulos virtuales es igual de efectivo que la exposición a esos mismos estímulos en la vida real, así como más efectivo que la exposición a través de fotografías. (Torres, 2012, pág. 148)

2.2.3. Adicciones

Un campo emergente en la psicología clínica es el uso de realidad virtual para tratar adicciones. De hecho existen ya algunos equipos de investigación que están diseñando herramientas virtuales para la evaluación y tratamiento de trastornos relacionados con el uso de sustancias como la nicotina o la heroína, aunque también para el tratamiento de una adicción no tóxica, como lo es el juego patológico. (Cristina Botella, 2007, pág. 36)

Un procedimiento clásico para estudiar y evaluar los comportamientos de tipo adictivo es el relacionado con la investigación mediante exposición a estímulos y contextos de consumo, evaluando las respuestas que esta desencadena. Dicha exposición se hacía mediante vídeos, fotos o imaginación, pero los resultados no eran muy prometedores tanto por la falta de estandarización de los procedimientos tradicionales como por la baja generalización de las conductas observadas en terapia con respecto a lo que hacen estas personas cuando se encuentran inmersas en un contexto real de consumo. (Torres, 2012, pág. 133)

Por lo tanto, era evidente que el uso de la realidad virtual era capaz de lograr algo que con los medios tradicionales no se estaba consiguiendo. Ya se han comenzado a crear procedimientos de realidad virtual donde se sumerge al paciente en entornos generados computacionalmente que representan diversos objetos y personas relacionadas con el consumo de drogas, como puede ser la presentación de las mismas sustancias evaluadas, objetos o “parafernalia” relacionada con su consumo, lugares de ocio asociados a estos comportamientos, etc. (Torres, 2012, pág. 133)

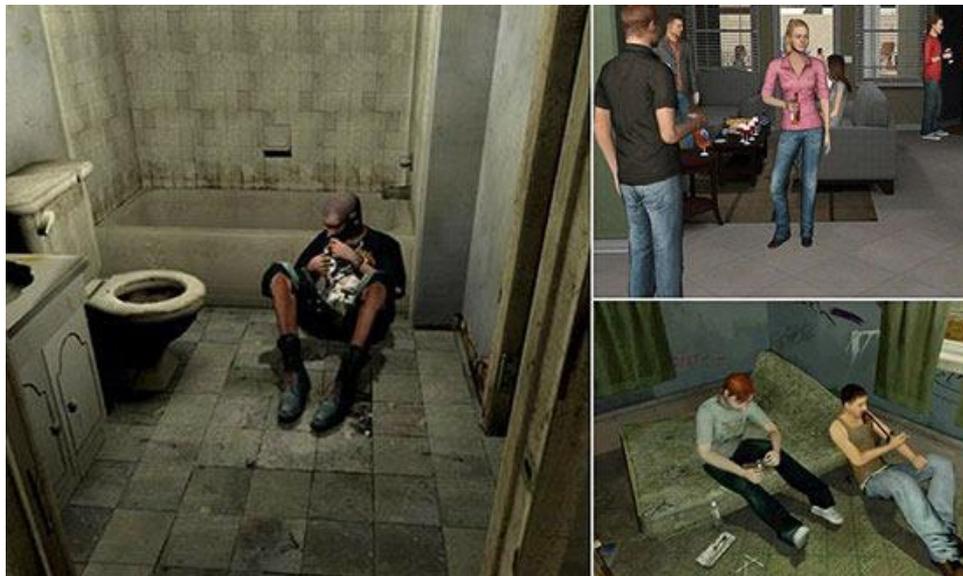


Ilustración 4. Realidad virtual para tratar adicciones.

Fuente: http://consalud.es/saludigital/admin/imagenes/realidad_virtual_adicciones_15042016.jpg

La finalidad de dicha exposición es evaluar la capacidad de los entornos virtuales de generar respuestas de *craving* (experiencia subjetiva de deseo intenso de consumir, o de necesidad imperiosa, de autoadministrarse una determinada sustancia adictiva (Romero, 2013)) con el fin de analizar su posible utilización como estrategias de exposición en el marco de las intervenciones que utilizan este tipo de tecnologías como parte del tratamiento de adicciones. (Torres, 2012, pág. 133)

Para el caso concreto de la adicción al juego se dispone de un escenario que representa una calle desde donde el paciente puede acceder a distintas situaciones relacionadas con el juego (un casino y un bar). Estos escenarios sirven para confrontar al paciente mediante exposición a los estímulos que generan el impulso de jugar y aprender, y practicar destrezas para superarlo. (Cristina Botella, 2007, pág. 36)

En última instancia, lo que se perseguiría sería utilizar dicha exposición con realidad virtual para conseguir disminuir las respuestas de *craving* a través de procesos de extinción, cambio de las cogniciones y, en definitiva, otras técnicas que se han mostrado eficaces para el tratamiento de estos problemas, habiéndose desarrollado dichos procedimientos principalmente desde una perspectiva cognitivo-conductual. (Torres, 2012, pág. 134)

2.2.4. Dolor agudo

Otro campo emergente es el uso de la realidad virtual en la psicología de la salud. Por todos es sabido que la realidad virtual constituye un potente distractor, debido al alto grado de inmersión y a la posibilidad de hacer que el usuario centre toda su atención en el mundo virtual. (Cristina Botella, 2007, pág. 36)



Ilustración 5. Uso de realidad virtual para pacientes con quemaduras.

Fuente: http://twimsgs.com/informationweek/galleries/automated/658/07-309IW500_Lehigh_Burn_Unit_7802_590_full.jpg

Este poder de distracción se ha comenzado a usar satisfactoriamente como táctica distractora en el manejo del dolor agudo en quemados asociado a procedimientos médicos como las curas y la fisioterapia. (Cristina Botella, 2007, pág. 36)

2.2.5. Trastornos psicóticos

Otro campo, aunque menos extenso, es el de la utilización de aplicaciones de realidad virtual para tratar síntomas psicóticos. Dichas aplicaciones se han focalizado en dos puntos vitales del tratamiento; por un lado en la simulación y el estudio de las experiencias (o síntomas psicóticos) que estas personas tienen, como pueden ser alucinaciones visuales o auditivas y, por otro lado, en la rehabilitación y evaluación de las habilidades cognitivas y sociales. (Torres, 2012, pág. 157)

2.3. Conceptos clave de la Realidad Virtual para su uso como tratamiento

A lo largo de todo el trabajo se está viendo la gran utilidad y repercusión que está teniendo la realidad virtual como tratamiento de problemas de salud mental pero, ¿a qué propiedades de la realidad virtual se debe eso?

Esta pregunta tiene una respuesta bien sencilla: la presencia y la interacción son las dos propiedades fundamentales de los sistemas de realidad virtual que posibilitan su uso en psicología clínica. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 2)

2.3.1. Presencia

Se denomina presencia a la sensación de estar en el entorno virtual. Los sujetos que pasan por entornos de realidad virtual no tienen la sensación de observar estos desde fuera, sino que sienten formar parte de ellos (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 2). Este fenómeno también es conocido como inmersión sensorial, pues supone la desconexión del mundo real y la inmersión en el mundo virtual generado por computador. (desconocido, pág. 5)

Hay diferentes factores que contribuyen a incrementar la sensación de presencia en un entorno virtual. Uno de ellos es el hardware que se utilice. Aquellos equipos que limitan la entrada de estímulos del ambiente real y potencian la correspondiente al entorno virtual disminuyen la sensación de presencia en el mundo real e incrementan la presencia en el entorno virtual (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 3). También hay que destacar que de los diferentes órganos de los sentidos, la vista es el que nos proporciona una mayor cantidad de información y, por lo tanto, las aplicaciones de realidad virtual tienden a centrarse en dicho sentido. (desconocido, pág. 5)

Dicho de otra forma: el fin último de los dispositivos de salida de una aplicación de realidad virtual es crear el efecto de que los objetos del entorno virtual existen con independencia del hardware que lo está generando. Sin realidad virtual, los usuarios son capaces de distinguir que los objetos están siendo proyectados sobre la superficie de una pantalla. Sin embargo, la utilización de realidad virtual crea en el usuario la sensación de que los objetos se encuentran a diferentes distancia en el espacio que le envuelve, como si realmente estuviesen a su alrededor. (desconocido, pág. 6)

Por otro lado, la presencia también depende de variables motoras. Si el sujeto tiene posibilidades de desplazarse y de interactuar con los objetos del entorno, su sensación de presencia será mayor que si se limita a observar lo que ocurre. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 3)

Aunque la variable que influye en mayor medida en la sensación de presencia es la velocidad de respuesta del sistema; una velocidad de respuesta elevada, o de manera ideal una respuesta en tiempo real, es determinante para que la interacción lleve a un aumento de la presencia. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 3)

Por último, creo conveniente destacar que los dispositivos que influyen en mayor medida en la sensación de presencia son los dispositivos de salida, los cuales definiré a continuación.

2.3.1.1. Dispositivos que propician la sensación de Presencia

2.3.1.1.1. Dispositivos visuales

Quizás el componente más conocido de este tipo sean los cascos de realidad virtual, los cuales cubren por completo el campo visual del sujeto con información que proviene del entorno virtual. En el caso de este dispositivo, cuanto mayor sea el ángulo visual, mayor será la experiencia inmersiva del observador. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 3)

La clave de esta clase de dispositivos es la visión estereoscópica, la cual se basa en proporcionar dos imágenes con ligeras diferencias de perspectiva del mundo virtual, una por cada ojo, de forma que el sentido de la visión humano pueda deducir la profundidad de los objetos a partir de las imágenes. La visión estereoscópica es un elemento vital en los sistemas de realidad virtual, pues es el único medio para hacer que los objetos tengan una fuerte presencia espacial. (desconocido, pág. 6)

Otros dispositivos de salida son los proyectores, los cuales pueden llegar a ser más inmersivos que los cascos, puesto que elimina la incomodidad de llevar un peso en la cabeza. Un ejemplo de este tipo de dispositivos es el CAVE: una habitación en la que paredes y suelo son pantallas sobre las que se proyectan partes del entorno virtual, haciendo así que el usuario quede rodeado por proyecciones. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, págs. 3-4)



Ilustración 6. CAVE.

Fuente: https://upload.wikimedia.org/wikipedia/commons/6/6d/CAVE_Crayoland.jpg

2.3.1.1.2. Dispositivos sonoros

Los dispositivos sonoros también influyen enormemente en la sensación de presencia, pues proporcionan al observador señales que le llevan a experimentar de manera precisa la localización de fuentes de sonido en el espacio. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 4)

2.3.1.1.3. Dispositivos táctiles

Este tipo de dispositivos son menos comunes, pero ya existen guantes especiales que consiguen transmitir al usuario sensaciones táctiles y, por lo tanto, aumentan su sensación de presencia en el entorno virtual. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 4)



Ilustración 7. Guantes de Realidad virtual.

Fuente: <https://www.cosasdejuegos.es/wp-content/uploads/2016/01/Juegos-Realidad-Virtual-guantes.jpg>

2.3.1.1.4. Dispositivos olfativos

Estos son sin duda alguna los dispositivos de uso menos extendido, pero si se quiere conseguir una sensación completa de presencia son vitales. Este tipo de dispositivos se basan en el uso de mezcladores químicos que pueden transmitir olores. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 4)

2.3.2. Interacción

La interacción es otro elemento esencial en un sistema de realidad virtual inmersivo, pues, como su propio nombre indica, permite al usuario actuar sobre el entorno. Sin embargo, antes de definirla como una característica de la realidad virtual, hablaré sobre dicho término tanto en su significado más tradicional como en el significado hombre-máquina habitual.

En una definición más clásica del término, se podría decir que la interacción es la acción, relación o influencia recíproca entre dos o más personas o cosas (Oxford Living Dictionaries). Esta definición es vital para entender la interactividad dentro de un sistema de realidad virtual, pues de ella se desprende la idea de que la interacción implica que toda acción tenga su reacción y, por lo tanto, en el ámbito de la realidad virtual debe cumplirse este factor.

Antes de llegar a definir la interactividad dentro de la realidad virtual, debo pasar por la tradicional forma de relación hombre-máquina, donde hasta ahora también ha existido interacción. En esta, el usuario que quiere hacer una acción tiene que comunicársela de forma explícita al ordenador, por lo que el usuario tiene que realizar el esfuerzo de recordar y/o buscar el comando o elemento gráfico asociado a la acción que quiere realizar. Además, debe hacer llegar al ordenador esta orden, fundamentalmente mediante el tacto a través de los dispositivos de interacción clásica: el teclado y el ratón. (desconocido, pág. 4)

De todas formas, debo indicar que incluso en ese ámbito se están haciendo avances sustanciales que minimizan el entorno tecnológico como son la miniaturización de periféricos, las tecnologías sin cables, el reconocimiento de voz y el empleo de herramientas de *software* que sustituyen al *hardware*, entre otros. (Lapuente)

Si nos centramos por fin en los sistemas de realidad virtual, la interacción se basa en el diálogo entre los usuarios y el entorno virtual, donde el hardware y el software son elementos vitales para conseguir que el usuario consiga un control total para con la representación virtual. De esta idea, se hace evidente que el "control" es una de las claves para definir la interactividad, pues, más allá de la simple navegación, se trata de que el usuario tenga verdadero control sobre determinadas acciones dentro del entorno. (Lapuente)

Así mismo, para que un sistema de realidad virtual genere interacción real, es esencial que el diseño, tanto del hardware o dispositivos como de los programas y aplicaciones, haga posible que la interfaz tenga capacidad suficiente para implicar al usuario en el espacio virtual generado. (Lapuente)

El buen diseño en la realidad virtual permite, en contraposición a lo expresado para la interacción clásica hombre-máquina, que el sistema capture la voluntad del usuario implícita en sus movimientos naturales. Un claro caso que ejemplifica la eficacia del diseño es que en un sistema de realidad virtual la cámara se actualiza en función de los movimientos de la cabeza del usuario, mientras que en la interacción clásica el giro de cámara se realizaría mediante movimientos del ratón. (desconocido, pág. 5)

Otra variable que influye en el ámbito de la interactividad, al igual que en la presencia, es el tiempo de respuesta. Ésta es sin duda una de las más importantes y sin duda alguna los futuros progresos en la realidad virtual se realizarán con la pretensión de una interacción instantánea para mejorar la experiencia sensorial del usuario. (Lapuente)

Por lo tanto, y a modo de conclusión, la interacción, al implicar tantos elementos (usuario, hardware, software, espacio geométrico virtual...), es una de las características más complicadas de generar, en lo que respecta a diseño y estrategia, de una aplicación de realidad virtual (Wikipedia).

Por último, me gustaría dar un apunte, al igual que he hecho con la sensación de presencia, indicando que los dispositivos que más favorecen la interactividad son los de entrada, los cuales definiré brevemente a continuación.

2.3.2.1. Dispositivos que propician la Interacción

2.3.2.1.1. Dispositivos de control

Dentro de los dispositivos que propician la interacción están los de control, los cuales son los encargados de transmitir las órdenes del usuario al sistema y, por lo tanto, son los que permiten una interacción más directa del usuario con el entorno virtual. En este tipo de dispositivos se encuentran los de sobremesa, como los teclados, los ratones 2D (trackballs), tabletas digitalizadoras, los joysticks convencionales y los dispositivos 3D (spaceballs), entre otros. (Ornedo, 2009, pág. 4)

2.3.2.1.2. Dispositivos de localización (trackers)

Por otra parte, también hay dispositivos de localización o trackers. Son los que utiliza el sistema para obtener información del entorno, por ejemplo, la posición 3D del usuario. Existen localizadores de diferentes tipos: electromagnéticos, mecánicos, ópticos, ultrasónicos, inerciales y los híbridos que mezclan diferentes tecnologías para aumentar la precisión. Otros localizadores más sofisticados son los guantes de datos, los cuales son muy utilizados para la comunicación de gestos además de la posición, y los eye-tracking que consiste en un sistema de localización basado en la detección de la posición de la pupila para localizar dónde se está mirando. (Ornedo, 2009, pág. 4)

2.4. Primeros usos de la Realidad Virtual en Psicología clínica

Una vez definida la realidad virtual, así como sus primeros sistemas, sus usos en psicología clínica y las características que posibilitan su uso como tratamiento, llega el momento de explicar brevemente la historia de la realidad virtual en el ámbito de la psicología clínica.

Lo cierto es que resulta complicado localizar en el tiempo, y con precisión, la aportaciones pioneras sobre sistemas de realidad virtual y realidad aumentada, la cual también contemplaré como tratamiento, en el campo de la salud mental (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 6). Las primeras aplicaciones aparecieron en Estados Unidos y se extendieron rápidamente a Inglaterra, Alemania, Francia y España (Maldonado, La realidad virtual es útil, 2010, pág. 2).

Para finalizar, y antes de pasar a enumerar los casos que mayor importancia han tenido, veo adecuado destacar que el primer grupo que publicó estudios aquí en España fue el que coordina la doctora Cristina Botella en la Universidad de Castellón (Maldonado, La realidad virtual es útil, 2010, pág. 2). Así mismo, también creo conveniente indicar que he decidido enumerar aquellos casos que he creído más relevantes desde el primer experimento que he encontrado (1982) hasta el año 2000, cuando ya se empezó a extender poco a poco su uso.

2.4.1. 1982

El primer estudio del que se tiene constancia es el de Schneider, que en 1982 realizó un pionero procedimiento mediante el cual trataba a pacientes con acrofobia (fobia a las alturas) con unas rudimentarias gafas que alteraban la percepción de profundidad con el objetivo de magnificar la altura percibida por el paciente durante procedimientos de desensibilización en vivo. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 6) (Torres, 2012, pág. 98)

Dichas gafas no se podrían considerar como un sistema de realidad virtual como tal, pero sin embargo sí que podría considerarse realidad aumentada, dado que modifican la realidad, sin llegar a sustituirla, añadiendo información que el paciente percibe.

2.4.2. 1992

En 1992, se llevó a cabo el primer experimento con realidad virtual propiamente dicha. En noviembre de ese mismo año, North, M.M. y North, S.M. trataron a una paciente de 32 años de edad con fobia a volar. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 8)

Para el tratamiento se realizaron ocho sesiones de media hora durante las cuales se exponía a la paciente a un entorno virtual que representaba una ciudad vista desde el aire. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 8)

Este experimento es vital en la historia de la realidad virtual como tratamiento de problemas de salud mental, puesto que se demostró que el paciente llegó a alcanzar unos niveles de ansiedad de valor de cero. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 8)

2.4.3. 1993

El experimento que a continuación expondré tiene el defecto de que hay muy pocos documentos que abalen su eficacia o validez (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 7). Pese a ello, el procedimiento resultó ser pionero en el uso de la realidad virtual en el campo de la psicología clínica y por ello le hago una mención especial.

El 1993, Kijima y Hirose, investigadores japoneses, describieron la utilización de un entorno virtual que simulaba un juego de arena para el tratamiento de autismo en niños. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 7)

2.4.4. 1994

En 1994, North y North, llevaron a cabo un estudio en el que se destacaba la relevancia de usar sistemas de realidad virtual como un paso intermedio entre la desensibilización sistemática por imaginación y en vivo en el contexto fóbico real. (Torres, 2012, pág. 100)

Entre sus estudios, destaca el desarrollo de un simulador de helicóptero, en el que mediante diversos dispositivos (*Head Mounted Display* junto con un *Head Tracking Device*), sonidos de motor de helicóptero, representaciones virtuales de aeropuerto, ciudad, ríos y lagos; y una cabina con vibración, lograban tratar a pacientes con miedo a volar. (Torres, 2012, pág. 101)

Según sus resultados obtenidos, el sistema fue capaz de generar altos niveles de ansiedad, los cuales fueron descendiendo durante la exposición, y una reducción significativa de los síntomas. Así mismo, también se notó un incremento en la habilidad para manejar situaciones fóbicas similares en la vida real y se observó que los resultados que se mantuvieron a largo plazo. (Torres, 2012, pág. 101)

Por otra parte, también me gustaría comentar que existe un apunte de prensa (RJ, 1994), de los mismos autores, en el que hablan de otro tratamiento de acrofobia realizado en 1994. En este caso, usaron un entorno virtual inmersivo que simulaba un profundo barranco que 32 pacientes debían atravesar pasando por un puente colgante y una tabla de madera. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 7)

2.4.5. 1995

Para este año solamente mencionaré un experimento muy concreto debido a que no trata el problema de fobias, sino que se centre en otro problema de salud mental.

En 1995, Strickland informó en que gracias al uso de realidad virtual, había logrado que dos niños autistas fueran capaces de reconocer e identificar objetos familiares dentro del entorno virtual. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 7)

2.4.6. 1998

El estudio publicado en 1998 tiene especial relevancia en el sentido de que fue el primer estudio español en tratar el caso. El equipo dirigido por Cristina Botella, en la Universidad de Castellón, presentó datos de eficacia en el tratamiento de la claustrofobia (fobia a los espacios cerrados) mediante el uso de un mundo virtual diseñado específicamente para dicha condición psicológica. Este sistema de realidad virtual incluía diferentes entornos tridimensionales: una habitación grande con ventanas y puertas interactivas, una habitación sin ventanas en la que se podía reducir el tamaño así como bloquear la salida, y un ascensor en el que también se podía reducir el tamaño y generar situaciones específicas como una avería. (Cristina Botella, 2007, págs. 34-35)

2.5. Estado actual de la Realidad Virtual en el campo de la salud mental

Como se ha podido observar, el campo de la realidad virtual dentro de la psicología clínica ha ido avanzando con los años hasta llegar al estado actual, donde cada vez tiene un uso más extendido y tiene unos resultados bien contrastados.

Aunque la mayoría de sistemas virtuales se desarrollen con la intención de tratar fobias, también se está avanzando en otras enfermedades, como los trastornos alimentarios o la rehabilitación psíquica y psicomotora. (Maldonado, Aplicaciones de la realidad virtual en psicología clínica, 2002, pág. 6)

Existen numerosos estudios en el que se comparan los resultados obtenidos mediante exposición en vivo (tratamiento tradicional), exposición por realidad virtual y un grupo de control en lista de espera con seguimientos que suelen abarcar el año completo. En dichos ensayos clínicos, se demuestra que la exposición virtual es igualmente eficaz que la exposición en vivo para el caso concreto de las fobias. (Cristina Botella, 2007, pág. 35)

Dado que ya es una realidad el uso de los entornos virtuales inmersivos para el tratamiento de problemas de salud mental, se conoce con claridad las distintas ventajas y desventajas que tiene su uso. Por ello, las enumeraré a continuación.

2.5.1. Ventajas

En la exposición en vivo tradicional, el tratamiento de fobias consiste en exponer al paciente, de forma controlada y gradual, a situaciones fóbicas que le generen niveles de ansiedad para, poco a poco, ir reduciendo estos niveles además de enseñarle mecanismos para el manejo de la ansiedad. (ms psicólogos)

Ahora, supongamos que hay un paciente con miedo a volar. Si se le tratara estrictamente como la forma tradicional sugiere, habría que llevar al paciente en todas las sesiones a un aeropuerto y, en algunas de esas sesiones, habría que subirlo a un avión (Maldonado, La realidad virtual es útil, 2010, pág. 1). Durante esas sesiones se debería proceder de forma gradual, primero sin llegar a subirse a un avión y después subiendo en aviones de apariencia más segura hasta los aviones más inestables.

Pero ese tratamiento, como bien se puede observar, tiene una serie de desventajas que la realidad virtual podría suplir.

2.5.1.1. Cuestiones económicas

En el caso concreto que abarcamos, el miedo a volar supondría un gasto económico demasiado alto, pues al gasto medio de una sesión con el psicólogo habría que sumarle el transporte hasta el aeropuerto así como los billetes de paciente y psicólogo (Maldonado, La realidad virtual es útil, 2010, pág. 1).

Si se usara realidad virtual, el paciente podría ir al aeropuerto y subirse en aviones sin salir de la consulta y, por lo tanto, sin tener que pagar esas enormes cantidades de dinero.

2.5.1.2. Cuestiones de tiempo

Dicho desplazamiento constante al aeropuerto junto con el hecho de subirse a un avión, no sólo supone un gasto económico, sino que también conllevan una gran inversión de tiempo, tanto por el psicólogo como por el paciente.

Con el uso de los entornos virtuales, el psicólogo tendría total control sobre el tiempo de la exposición, pudiendo así ajustarlo a las necesidades del paciente como al tiempo establecido para sus sesiones.

2.5.1.3. Presentación de situaciones en cualquier momento

De la ventaja anterior se desprende otra muy importante: el psicólogo puede graduar las situaciones según las necesidades del paciente sin necesidad de esperar a que éstas sucedan en la vida real. Es decir, no habría que esperar a que el avión despegara, sino que en consulta éste puede despegar siempre que el especialista así lo desee.

Así, este hecho permitiría exponer al paciente a situaciones clínicamente significativas tantas veces como fuese necesario, superándose así el inconveniente de la

exposición en vivo clásica en la que es necesario esperar a que determinado evento se produzca para poder exponer a la persona a dicho contexto. (Torres, 2012, pág. 94)

2.5.1.4. Repeticiones

Como bien se puede apreciar en la ventaja anterior, exponer al paciente a situaciones en vivo conlleva otro riesgo para su tratamiento: el psicólogo no tendría forma de controlar que siempre que se exponga al paciente a una situación, esta ocurra en las mismas condiciones. (Cristina Botella, 2007, pág. 34)

La exposición virtual permite, por lo tanto, repetir la misma tarea de exposición una y otra vez sin cambiar sus parámetros. De este modo, el psicólogo podría exponer al paciente, por ejemplo, a una situación concreta de despegue en la que siempre será de día y no habrá viento. (Cristina Botella, 2007, pág. 34)

2.5.1.5. Simulación de situaciones del mundo real

Todas las ventajas ya descritas no tendrían sentido si los entornos virtuales no pudieran simular realmente situaciones del mundo real. Mediante los grandes progresos realizados en materia de rendimiento y diseño tridimensional, los actuales sistemas de realidad virtual son capaces de simular situaciones, escenas y entornos con un grado de realismo y fidelidad muy alto. Esto permite exponer al paciente a un ambiente similar al que se encontraría en la vida real. (Torres, 2012, pág. 93)

Esta característica supone una gran ayuda en el tratamiento de aquellos pacientes con dificultades para introducirse en ejercicios de exposición por imaginación. En este sentido, los resultados obtenidos por realidad virtual se considerarían de mayor validez al lograr exponer al paciente a entornos con un alto grado de realismo y similitud al entorno real. (Torres, 2012, págs. 93-94)

2.5.1.6. Experiencias similares a la vida real

Dado el alto grado de realismo que han adquirido las simulaciones, no es de extrañar que los pacientes también sientan experiencias similares a las que acontecerían en el mundo real.

Hay estudios que ponen de manifiesto la gran similitud entre las reacciones que el paciente tiene frente a un entorno virtual y frente a la vida real. Dicho de otra forma, la persona que recibe el tratamiento experimenta las mismas reacciones y experiencias emocionales que las que se producen en la vida real. (Torres, 2012, pág. 92)

De esta forma, el psicólogo puede evaluar de forma precisa como el paciente siente, piensa y actúa de manera similar a como lo haría al verse expuesto a la misma situación en la vida real. Al tratarse de reacciones semejantes, las respuestas dadas ante sistemas de realidad virtual serían válidas, tanto desde el punto de vista del tratamiento, como de evaluación e investigación. (Torres, 2012, pág. 92)

2.5.1.7. Seguridad del entorno virtual

Volviendo al caso del paciente con miedo a volar, si se le tratara estrictamente como la exposición en vivo tradicional sugiere, el psicólogo no podría asegurar la integridad de su paciente, dado que existen innumerables elementos ajenos a su control. Por ejemplo, en el recorrido al aeropuerto podría ocurrir un accidente que pusiera en riesgo la salud tanto del paciente como del psicólogo.

Por lo tanto, los entornos virtuales se presentan como un ambiente seguro donde el paciente no es expuesto a los peligros y riesgos reales que podría causarle una exposición en el mundo real. En este sentido, el paciente que recibe el tratamiento virtual puede experimentar las emociones y pensamientos con la seguridad de que aquello que realmente le atemoriza no puede causarle ningún daño. (Torres, 2012, págs. 92-93)

Este hecho, lleva a que muchos pacientes se sientan más relajados y perciban la terapia como un lugar seguro y protegido, llegando a ser de gran ayuda en el caso de las personas que no están dispuestas a exponerse a ciertas situaciones en la vida real. (Torres, 2012, pág. 93)

Otro caso en el que resulta muy útil esta ventaja, es en el tratamiento de adicciones a sustancias ilegales, pues se puede exponer al paciente a un situación en que le ofrecen una determinada droga sin que exista un riesgo real. (Torres, 2012, pág. 93)

2.5.1.8. Control sobre las escenas presentes

Como he demostrado, la exposición en vivo tiene una serie de problemas en cuanto a repetir el mismo evento, así como de asegurar la integridad del paciente. Volviendo al ejemplo del pánico a volar, supongamos que en el segundo vuelo que coge el paciente se producen turbulencias. El paciente seguramente no quiera seguir con su tratamiento. (Maldonado, La realidad virtual es útil, 2010, pág. 1)

Una ventaja a tener en cuenta en la terapia por realidad virtual es la posibilidad de controlar los estímulos y eventos del mundo virtual en su totalidad. El especialista puede elegir que parámetros representar en el entorno así como la intensidad de los mismos, logrando así adecuar enormemente el entorno a cada paciente. (Torres, 2012, pág. 94)

Esto a su vez supone otra ventaja en la evaluación, pues permite al psicólogo medir con mayor precisión que estímulos y eventos concretos son los que más alteran al paciente. Estos datos posibilitan llevar un registro de datos más homogéneo y objetivo y, en definitiva, hacen posible que el especialista pueda realizar un tratamiento más flexible y adaptado. (Torres, 2012, pág. 95)

2.5.1.9. Confidencialidad

Si tratáramos al paciente con miedo a volar mediante exposición en vivo, los datos obtenidos así como las reacciones y sentimientos del paciente quedarían expuestos a toda persona que pasara por el lugar. El simple hecho de lo tener que salir de la consulta para realizar el tratamiento también favorece aspectos éticos como la protección de la intimidad, ya que no es necesario que el paciente manifieste sus síntomas en lugares públicos. (Cristina Botella, 2007, pág. 34) (Torres, 2012, pág. 95)

Este trato más confidencial que ofrecen los sistemas de realidad virtual también propicia una mayor seguridad del paciente y conlleva a que se cohiba menos a la hora de expresar sus pensamientos, sentimientos y reacciones.

2.5.1.10. Cuestiones estadísticas

Por último, el tratamiento mediante realidad virtual también tiene ventajas en cuestiones estadísticas de la medición. Me parece importante destacar que, tanto a nivel del tratamiento como en la investigación científica, los datos obtenidos con el uso de entornos virtuales inmersivos gozan de más validez y fiabilidad. (Torres, 2012, pág. 95)

La flexibilidad de los tratamientos, como he comentado más arriba, favorece la individualización del tratamiento y la evaluación, además de facilitar que sean replicados de forma más sencilla los estudios llevados a cabo por otros investigadores. (Torres, 2012, pág. 96)

2.5.2. Desventajas

A pesar de las enormes ventajas que posee el uso de entornos virtuales inmersivos para el tratamiento de problemas de salud mental, existen también una serie de inconvenientes y desventajas que también deben ser explicadas.

Debo decir que las limitaciones que expondré a continuación no suponen, en principio, un factor a tener en cuenta que limite o ponga en duda la efectividad de la utilización de la realidad virtual en el campo de la psicología clínica.

Además, las desventajas de este tipo de tratamientos radican principalmente en que la realidad virtual aún no es una forma de proceder lo suficientemente extendida y a la disparidad existente entre los distintos entornos que se utilizan, por lo que en un futuro posiblemente terminen desapareciendo dichas desventajas.

2.5.2.1. Incompatibilidad

Como bien se ha observado a lo largo de todo el documento, la mayoría de los sistemas de realidad virtual existentes hasta la fecha se centran en solventar un problema específico. (Cristina Botella, 2007, pág. 36)

Este hecho supone un problema a la hora de aplicar estos tratamientos en la práctica rutinaria, ya que el terapeuta necesitaría tener a su disposición distintos ambientes virtuales para tratar las diferentes situaciones de ansiedad. Así mismo, tampoco existe una convencionalidad en la que todos los ambientes hagan uso de los mismos dispositivos, así que al psicólogo, además, necesitaría diferente hardware para cada tratamiento. (Cristina Botella, 2007, pág. 36)

2.5.2.2. Posible "enfermedad del simulador"

Esta desventaja sólo se ha observado en casos concretos, pero he creído conveniente definirla en esta lista. Algunas investigaciones pusieron de manifiesto que, a pesar de las increíbles ventajas de este tratamiento, algunos pacientes presentaban una serie de efectos secundarios a los que acuñaron "enfermedad del simulador".

Dichos efectos, que consistirían en problemas oculares, desorientación, problemas de equilibrio y náuseas, se ponían de manifiesto tanto en el momento de la exposición como después de la misma. Algunas personas incluso los padecían con solo 10 minutos de exposición. (Torres, 2012, pág. 96)

No obstante, no llega a estar clara la relación entre la exposición por realidad virtual y dichos síntomas, ya que existen gran cantidad de estudios en los que el tratamiento no habría producido efecto secundario alguno. (Torres, 2012, pág. 96)

Por último, añadir que con el constante avance de la tecnología y con la mejora paulatina de su calidad, seguramente los casos de pacientes con la enfermedad del simulador se reduzcan enormemente.

2.5.2.3. Coste económico

En el apartado de Ventajas, he comentado el aspecto económico como una factor favorable para los pacientes, pero, paradójicamente, el factor económico supone, a su vez, una desventaja para el uso de esta tecnología.

El alto coste del desarrollo de entornos virtuales, así como de la adquisición de los mismos y de los dispositivos necesarios, hace que algunos profesionales no se interesen todavía en su uso como terapia. (Torres, 2012, pág. 97)

Según el proyecto europeo VEPSY (Telemedicine and Portable Virtual Environments for Clinical Psychology), el coste que conlleva diseñar, desarrollar, aplicar y evaluar un sistema virtual inmersivo tradicional estaría entre los 150.000 y los 200.000 euros. Por este motivo, algunos investigadores han propuesto desarrollar los entornos virtuales para su uso en sistemas de realidad virtual de sobremesa comerciales, los cuales supondrían un menor gasto de adquisición para los terapeutas interesados. (Torres, 2012, pág. 97)

2.6. Tecnologías necesarias para la realización del entorno

Tras describir brevemente la historia y el estado de la Realidad Virtual y la Realidad Aumentada, así como su uso en psicología clínica, debo explicar cuáles son las tecnologías que me permitirán crear un entorno virtual inmersivo para la detección y tratamiento de problemas de salud mental.

Como se ha podido entrever a lo largo del marco teórico, para la creación de los sistemas de realidad virtual se necesitan: unos modelos geométricos tridimensionales para representar en mundo virtual y sus objetos; un código que permita generar dicho entorno así como que favorezca la personalización del mismo y la interacción con el usuario; y, por último, un dispositivo que permita la correcta visualización del espacio virtual. Por lo tanto, en el presente apartado se dará una breve explicación del modelado 3D, del entorno de trabajo Unity y de los dispositivos visuales de realidad virtual, respectivamente.

2.6.1. Creación de los modelos tridimensionales

En términos generales, aplicables a distintas disciplinas, se pueden definir los modelos como una simplificación de un objeto para su posterior estudio y/o representación. Del mismo modo que los modelos matemáticos simplifican los fenómenos físicos, los modelos geométricos tridimensionales definen información sobre la geometría y orientación de un determinado objeto en el espacio. (Centro de excelencia de software libre)

Por lo tanto, se puede afirmar que el modelado 3D es el proceso de crear modelos virtuales que puedan ser manejados computacionalmente. Esta técnica es comparable a esculpir modelos de yeso en el mundo real, pero en este caso se utiliza el ordenador y los programas de computo 3D para generarlos virtualmente. (Portal de información CAD/CAM/CAE)

En líneas generales, la construcción de estos elementos se realiza en diferentes etapas, partiendo desde las primitivas geométricas básicas hasta terminar con su texturizado, pasando, si procede, por el *rigging* y la animación. (Centro de excelencia de software libre)

El resultante de dicho proceso puede ser visualizado como una imagen bidimensional mediante el renderizado 3D o bien utilizarlo en una simulación por computadora de fenómenos físicos. Dados los avances actuales, también existe la posibilidad de crear físicamente el modelo usando dispositivos de impresión 3D. (Wikipedia)

A continuación se hará una breve descripción de los distintos pasos del modelado 3D y texturizado que han sido necesarios para la realización de este proyecto, aunque antes se explicarán una serie de conceptos matemáticos indispensables para comprender correctamente esta rama de la computación.

2.6.1.1. Conceptos matemáticos

El primer concepto a definir es el de sistema de coordenadas, las cuales son un conjunto de valores que permiten definir la posición de cualquier punto del espacio geométrico en relación a un punto de referencia u origen. (Cobián, pág. 1)

Dicho sistema de coordenadas se aplica a todos los elementos, como, por ejemplo, a los vértices. Éstos, son el componente básico en la construcción de los modelos, pues gracias a ellos se forman el resto de figuras del modelado tridimensional. Los vértices, además de tener unas coordenadas espaciales, también poseen otras de textura, que son indispensables para poder aplicar de forma correcta la textura a los modelos de los que forman parte.

El siguiente elemento de importancia es el segmento, que es aquella parte de una línea que queda entre dos vértices. Este concepto es necesario para entender el de polígono, el cual se define como una figura plana y cerrada formada por tres o más

segmentos que se cruzan (Cobián, pág. 1). Los polígonos, o caras, son la base de todo modelado pues son ellos los que, en gran medida, indican la calidad de la malla de un objeto tridimensional virtual.

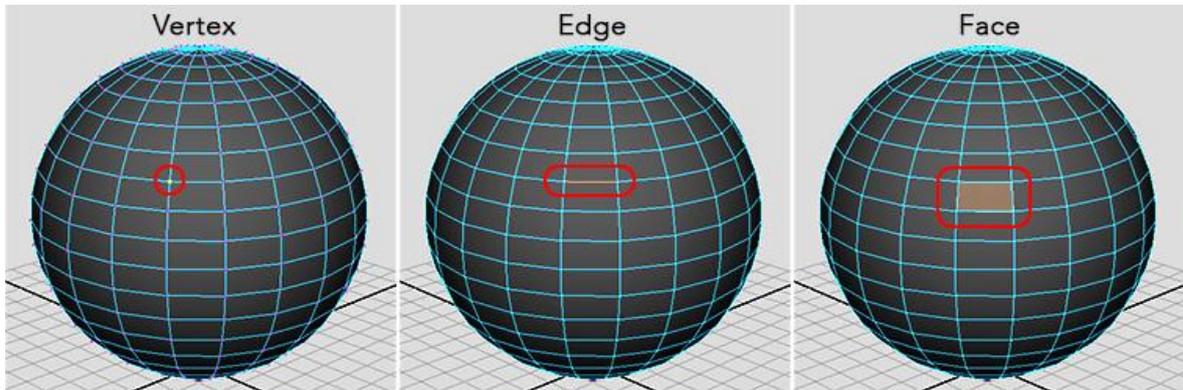


Ilustración 8. Diferencia entre vértice, segmento y cara.

Fuente: http://blog.hitchingsdesign.com/wp-content/uploads/2013/07/3D_mesh_components.jpg

Otro concepto vital es el de vector, que es todo segmento dirigido en el espacio. Por lo tanto, se componen de una serie de características: un punto de origen, un módulo que indica su longitud, una dirección que marca su orientación en el espacio, y un sentido, que se suele indicar con una flecha. Dentro de los estos, destaca el vector normal, pues, al tratarse de un vector perpendicular a un punto concreto de un modelado dado, indica hacia qué lado miran los polígonos de un objeto virtual y, por lo tanto, aportan información importante a las tareas de iluminación y renderizado, entre otras. (Cobián, pág. 1)

2.6.1.2. Modelado 3D

El proceso de modelado se puede hacer, principalmente, de dos formas bien diferenciadas: automática y manualmente. En el primer caso, definiendo a grandes rasgos, se hace uso de escáneres 3D que analizan un objeto o escena para reunir datos de su forma

y que posteriormente se utilizarán para construir el modelo geométrico tridimensional. (Wikipedia)

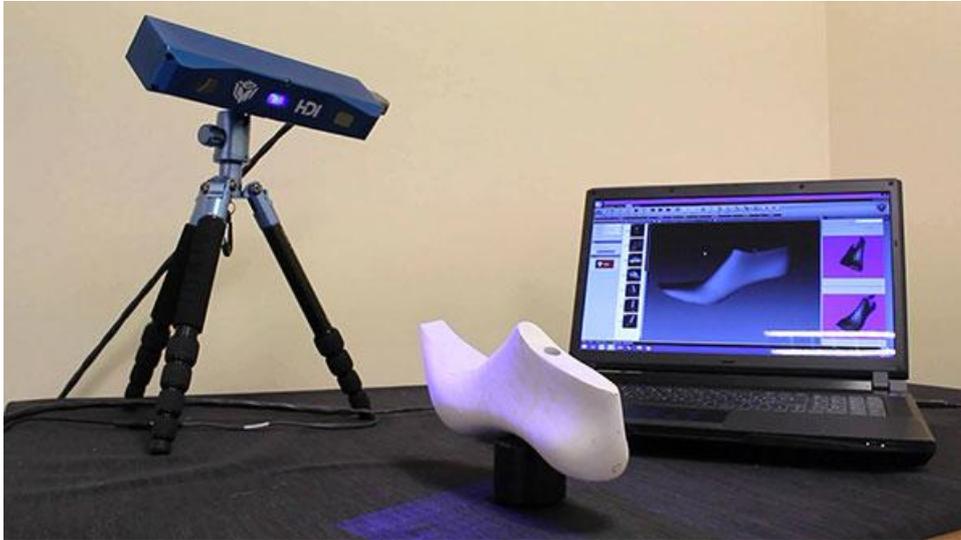


Ilustración 9. Escáner 3D.

Fuente: <http://www.aniwaa.com/wp-content/uploads/2014/11/3D-scanner-LMI-Technologies-HDI-120-in-use.jpg>

Dado que en este trabajo se ha realizado el modelado de forma manual, este método se explicará, en principio, más detalladamente. Lo primero que hay que indicar es que la creación manual de modelos también se puede dividir, a grosso modo, en tres grandes ramas: el modelado poligonal, el modelado por curvas y el modelado por esculpido (Wikipedia). Antes de pasar a describir las tres formas, debo indicar que no son las únicas, pues también están las superficies de subdivisión y el *hard surface modelling*, entre otros.

En el caso del modelado poligonal se suele empezar a partir de geometrías planas simples (planos, triángulos, vértices...) o bien de cuerpos geométricos básicos (esferas, cubos o cilindros), a los cuales se les aplican deformaciones y transformaciones hasta que se obtiene la malla poligonal deseada. Esta forma de proceder es la más extendida, ya que los resultados son muy flexibles y los ordenadores pueden renderizarlos con relativa rapidez. (Wikipedia)

Si nos centramos en el modelado de curvas, hay que decir que se basa en la utilización de curvas para generar los objetos tridimensionales, siendo dos las principales formas de usarlas: con carriles y por revolución. Mientras que en la primera una curva indica el recorrido que debe seguir otra que realiza la función de perfil de extrusión, en la segunda lo que se hace es revolucionar una curva en relación a un eje para conseguir, principalmente, un modelo de forma cilíndrica. (Wikipedia)

En el caso del esculpido, hay que destacar que, aunque todavía es un método relativamente nuevo, se ha vuelto muy popular en la industria (Wikipedia). La escultura digital se basa en el uso de softwares que permiten empujar, estirar, deformar y suavizar (entre otros) un objeto virtual que se comporta como si estuviera hecho con arcilla real. (Wikipedia)



Ilustración 10. Ejemplo de esculpido 3D.

Fuente: https://formlabs.com/media/upload/_thumbs/stl_griffin.jpg.895x0_q80_crop-smart.jpg

Tanto si se trabaja manual como automáticamente, el modelado obtenido mediante este proceso se puede entender como un objeto conceptual en tres dimensiones que se puede definir de dos formas distintas: técnica y visualmente. (Diccionario de informática y tecnología)

Desde el punto de vista técnico, es un conjunto de algoritmos matemáticos que describen un modelo tridimensional dentro de un mundo generado por ordenador. Visualmente, se puede decir que es una representación esquemática visible a través de un

grupo de objetos, elementos y propiedades que, una vez procesados, se convertirán en una imagen renderizada, una animación digital o una impresión 3D. (Diccionario de informática y tecnología)

2.6.1.3. Texturizado

Cuando se modela un objeto, a su superficie se le asigna por defecto un color uniforme y liso, pero gracias al uso de texturas se puede conseguir que adquiera el nivel de realismo necesario. (Cobián, pág. 4) Es por ello que la fase de texturizado es tan importante como la del modelado, pues no sólo permite añadir color a un modelo tridimensional, sino que también puede simular diferentes materiales y dar mayor nivel de detalle a determinadas formas de la geometría. (Ruiz, 2011, pág. 16)

Como en cualquier desarrollo, para poder llegar a unas texturas de calidad, hay que seguir un proceso de trabajo que incluye desarrollar los modelos, "cocinarlos o hornearlos" y pintarles las texturas encima. La correcta ejecución de todos estos pasos es de vital relevancia para obtener el resultado deseado y, por lo tanto, se definirán de forma breve y concisa a continuación.

Por último creo conveniente destacar que aunque el texturizado es un proceso clave dentro del modelado 3D, este no tiene una ubicación específica en el proceso de creación de un objeto tridimensional, sino que cada profesional lo realiza cuando encuentra que es más conveniente (Orozco). Aún así, por norma general, se suele realizar, como en mi caso, justo después de terminar el modelado y desarrollarlo.

2.6.1.3.1. Mapeado UV

Al proceso de desarrollar un modelo 3D se le conoce como mapeado UV, siendo estas letras una referencia a las dos coordenadas U y V que tendrá el mapa de textura resultante. (Wikipedia)

El mapeado UV se considera una parte indispensable en el texturizado de toda figura geoméricamente compleja, pues con él se busca poder desenvolver una malla tridimensional y plasmarla en una imagen bidimensional con la menor distorsión posible.

Para realizar esta tarea se suele recurrir a software especializado, aunque siempre es recomendable aprender los fundamentos del desplegado de mapas UV antes de pasar a emplear un software que haga el proceso prácticamente solo, pues así se logrará entender que hace el programa y tener mayor control sobre este y sus automatismos. (Orozco)

Una forma habitual de proceder es indicarle al software de desplegado qué partes de la malla son juntas, con el fin de que al desenvolverlo realice los cortes en esa zona evitando así distorsiones en las áreas más conflictivas de la malla. (Wikipedia)

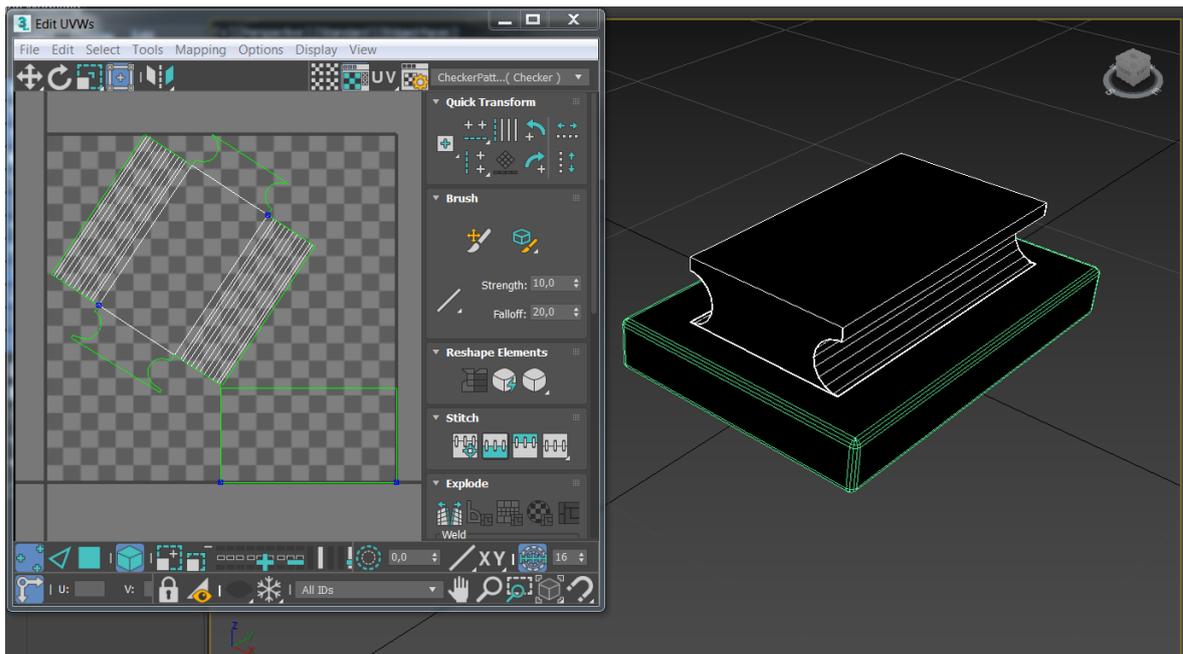


Ilustración 11. Ejemplo de mapeado UV.

Fuente: propia

Por lo tanto, se puede definir el mapa UV como una imagen bidimensional que puede asimilar al envoltorio con el que se cubren los objetos y que se asocia a un material para lograr aumentar así su fidelidad al objeto original. Es importante indicar que no solo

se usan para añadir color al modelo, sino que además pueden simular otras características como la rugosidad o el relieve sin necesidad de alterar la geometría del objeto tridimensional. (Ruiz, 2011, pág. 18)

2.6.1.3.2. Baking

El cocinado o *bake* es una parte del proceso que no siempre se aplica a la hora de texturizar un modelo tridimensional. Esto se debe a que el *bake* tiene como objetivo crear una malla simple que tenga los detalles de una geometría compleja sin necesidad de consumir una gran cantidad de recursos del procesador y, por lo tanto, no es necesario realizarlo si el modelo 3D final no debe tener una malla compleja. (Animat 3D, 2009)

Para realizarlo se suele proceder realizando un mapeado UV de una malla compleja ya modelada de alta resolución y capturando sus normales. El mapa de normales resultante se guarda y entonces se desenvuelve de forma similar el modelo de baja resolución al que se le quiere hacer el *bake*. (wiki de Blender)

El siguiente paso es aplicarle el mapeado de normales del objeto complejo a la malla de baja resolución, que tiene unas UV desplegadas de forma similar. Si se ha realizado correctamente, ésta lucirá similar a la de alta resolución, pero contendrá un número mucho menor de caras o polígonos. (wiki de Blender)

El cocinado de texturas tiene, como se puede deducir de lo anteriormente explicado, una serie de ventajas a tener en cuenta, como lo son: la reducción de los tiempos de procesamiento, la reducción de la cantidad de polígonos en escena y el facilitado del pintado de texturas. (wiki de Blender)

Por otra parte, también tiene la desventaja de que fuerza a que los objetos deban tener unas coordenadas de mapeo UV. Así mismo, el cocinado implica que haya que dedicarle un tiempo de trabajo a desplegar, capturar, pintar y aplicar las texturas a otros materiales (wiki de Blender). En relación a esto último, es importante destacar que en el

mercado actual existe software específico capaz de realizar el *bake* de un modelo casi de forma automática, por lo que el tiempo que habría que dedicarle al proceso se reduciría.

2.6.1.3.3. Pintado de texturas

Una vez que se ha hecho el mapeado UV, hay varias formas de proceder para su pintado, pudiendo dividirse principalmente en dos: mediante programas tradicionales de edición de imagen o bien haciendo uso de software especializado en esta tarea.

La primera forma consiste, como se puede deducir, en la utilización de programas como Photoshop para pintar directamente sobre el mapeado. Sin embargo, esta no suele ser la forma más habitual de proceder.

Si nos centramos por el contrario en la segunda forma, nos encontramos con programas que permiten colorear, no sólo sobre el mapa UV que representa el modelo desenvuelto, sino que también sobre el modelo en sí mismo, de forma tridimensional.

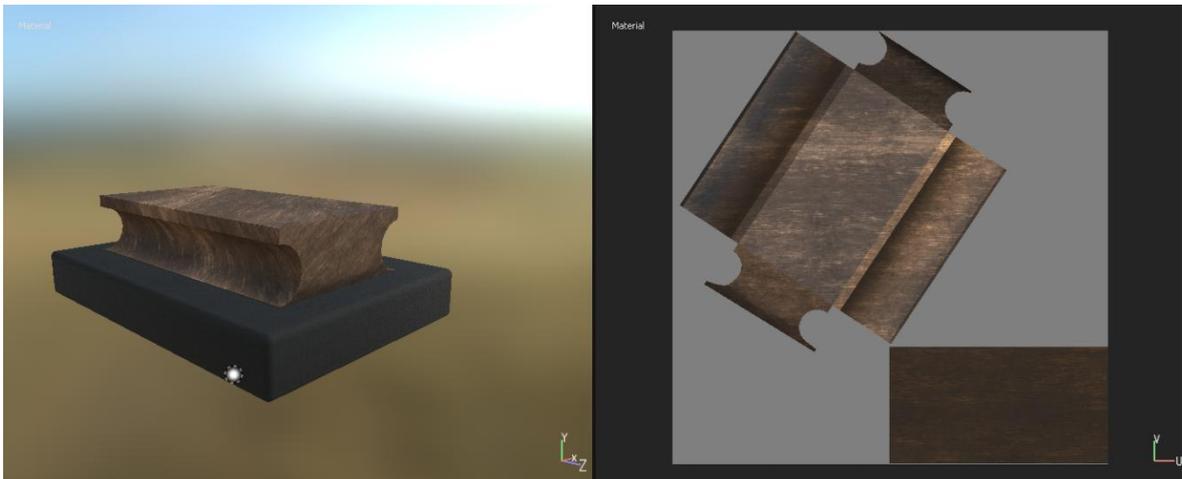


Ilustración 12. Ejemplo de pintado de texturas.

Fuente: propia

Además, al igual que con el software convencional de edición de imagen, los programas especializados en el pintado de textura dan la opción de trabajar por capas, lo que facilita en gran medida las tareas de incluir más de un tipo de textura dentro del mismo modelo.

Cabe destacar, que hay artistas que recomiendan que, una vez usado el programa especializado en pintado, se mejore la textura con software de edición de imagen, es decir, se puede aprovechar la potencia de estos programas para perfilar y detallar todas aquellas partes de la textura que hayan podido quedar peor. (Orozco)

2.6.1.4. Materiales

Una vez se han obtenido los modelos tridimensionales y sus respectivas texturas y mapas de normales, llega el momento de aplicar los materiales. Técnicamente hablando se puede afirmar que estos últimos realizan principalmente la función de contenedores de los *shaders* y texturas que se pueden asignar a los objetos. (Geig, 2013)

Los sistemas de materiales son vitales en el modelado 3D, ya que permiten simular los diferentes materiales de la vida real en un objeto tridimensional generado computacionalmente. (Wikipedia)

Dado que su principal función es la de almacenar y aplicar texturas, los materiales también se encargan de gestionar parámetros avanzados del renderizado (como la reflexión o la especularidad) así como algunas propiedades físicas de comportamiento (por ejemplo: la densidad o la fricción). (Wikipedia)

Por lo tanto, no es de extrañar que también posea una gran cantidad de variables relacionadas con el texturizado en sí mismo. De estas cabe destacar el color difuso o albedo, el mapa de normales y el *metallic-roughness*; recibiendo todos estos parámetros como entrada una imagen.

En el caso del albedo, al asignarle una textura al material, los colores de la imagen sustituirán el color difuso del material en cuestión (Autodesk Navisworks 2016, 2016). Si

además se le asigna un mapa de normales obtenido por mapeado UV se logra dotar de una gran realismo al modelo 3D. Por último, el parámetro *metallic* recibe una imagen en la que, siguiendo los colores blancos y negros, dota la zona correspondiente de la textura de más o menos brillo, respectivamente.

2.6.2. Unity

Una vez descrito el flujo de trabajo para la creación de modelados 3D, junto con su posterior cocinado y asignado de texturas y materiales, el siguiente paso lógico es explicar el entorno de trabajo que posibilitará la creación del entorno virtual inmersivo para la detección y tratamiento de problemas de salud mental.

En el caso concreto de este trabajo se ha optado por elegir el motor de videojuegos multiplataforma de Unity. Esta decisión se basa en una serie de ventajas que se tuvieron en cuenta y que a continuación se enumerarán.

2.6.2.1. Características y ventajas generales

A pesar de que originalmente fue construido exclusivamente para funcionar y generar proyectos en los equipos de la plataforma Mac, ya está disponible como plataforma de desarrollo para Microsoft Windows, OS X y Linux. Lo cual constituye una importante ventaja a la hora de elegirlo como motor de desarrollo. (Wikipedia)

Otros de sus puntos a favor es que puede usarse junto con una larga serie de programas de creación de contenidos (Blender, 3ds Max, Maya, ZBrush, Adobe Photoshop, Adobe Fireworks y Allegorithmic Substance, entre otros), pues actualiza automáticamente los cambios realizados a los objetos creados con estos productos en todas las instancias de ese archivo durante todo el proyecto sin necesidad de volver a importar manualmente. (Wikipedia)



Ilustración 13. Logo de Unity.

Fuente: <https://unity3d.com/profiles/unity3d/themes/unity/images/assets/favicons/ogimg.jpg>

Por otra parte, es importante destacar que tiene soporte de compilación con diferentes tipos de plataformas como, por ejemplo, web (WebGL), dispositivos móviles (iOS, Android, Windows Phone), Smart TVs (tvOS, Samsung Smart TV), videoconsolas (PlayStation, Xbox, Nintendo) y dispositivos de realidad virtual (Oculus Rift, Samsung Gear VR), entre otros. (Wikipedia)

Como última ventaja a resaltar, decir que Unity cuenta con tres versiones: Unity Pro, Unity Plus y Unity Personal. Mientras que las dos primeras son de pago, la tercera es totalmente gratuita y, aun así, permite publicar los videojuegos y las aplicaciones realizadas con el motor, aunque con algunas limitaciones (la restricción de compilar con un splash screen con el logo y la leyenda "Made with Unity", así como un tope de ingresos de \$100 mil dólares). (Wikipedia)

Tras todas estas ventajas, también debo destacar que Unity posee una interfaz amigable e intuitiva y que tanto su editor como su sistema de scripts suponen unos aliados muy importantes a la hora de realizar cualquier aplicación con el programa y, por lo tanto, se describirán de forma más extensa a continuación.

2.6.2.2. Características y ventajas del Editor de Unity

A lo largo de este apartado se destacarán aquellas características del editor de Unity que o bien he considerado de mayor relevancia o bien me han resultado de mayor utilidad a lo largo de la creación del entorno virtual inmersivo.

Los parámetros a resaltar se dividen en aquellos que ayudan en tareas de configuración, ya sea del entorno como de la aplicación en sí, y en aquellos que asisten a lo largo del desarrollo de un proyecto.

2.6.2.2.1. Configuración básica

Una de las primeras características que se pueden observar a simple vista nada más entrar al editor de Unity es que permite elegir si el proyecto se realizará en 2D o en 3D. Esta herramienta resulta de gran utilidad, pues posibilita realizar cambios de última hora ya que da la opción de cambiar de modo en cualquier momento a lo largo del desarrollo de la aplicación. (Unity Documentation)

Así mismo, en las preferencias generales de Unity se encuentra una opción muy atractiva a la hora de crear un entorno virtual con modelados tridimensionales de elaboración propia: el *auto refresh*. La gran utilidad de esta opción, que viene seleccionada por defecto, radica en que si se detectara cualquier detalle a cambiar en un elemento del proyecto, solamente con editar el archivo original bastaría para que se modificaran también todas las instancias del mismo dentro de la aplicación. (Unity Documentation)

Otras de las herramientas de gran utilidad son las que contempla el menú de opciones de "construcción" (*build*) del proyecto. En esta ventana se puede elegir desde que escenas y en qué orden se cargan en el proyecto como la plataforma para la que se exportará, pasando por decidir si el programa resultante soportará la realidad virtual o no. (Unity Documentation)

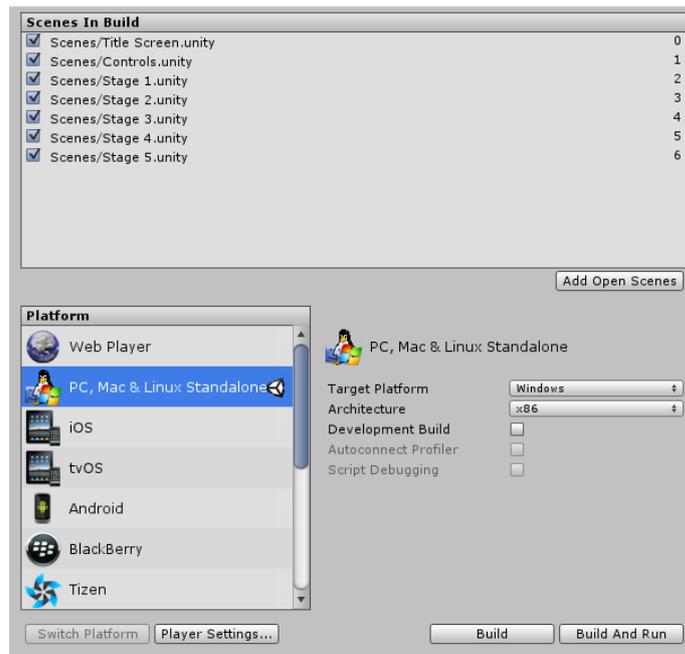


Ilustración 14. Ventana "build settings" de Unity.

Fuente: <https://docs.unity3d.com/uploads/Main/BuildSettings.png>

En la ventana anterior también se puede acceder a los ajustes del jugador, donde se encuentran parámetros como la resolución, el tamaño de pantalla por defecto o el icono de la aplicación, entre otros. (Unity Documentation)

Por último, destacar la posibilidad que ofrece Unity de elegir el orden de ejecución de los Scripts. El motor del juego carga los scripts de forma arbitraria por defecto, por ello, se da la opción al desarrollador de que elija cuales se ejecutarán primero y cuales lo harán a continuación. (Unity Documentation)

2.6.2.2.2. Desarrollando la aplicación

Pasando ya al desarrollo de la aplicación, el primer elemento a destacar es la gran usabilidad que presenta el entorno de trabajo y su versatilidad. El editor permite recolocar cualquier elemento de la interfaz al gusto del desarrollador, logrando ofrecer así un menú con infinitas opciones de organización. Entre estos apartados móviles se encuentran:

- *Scene*: pantalla en la que se trabaja de forma directa con los elementos de la aplicación dentro del entorno virtual.
- *Game*: pantalla que muestra cómo se verían los modelos de la escena con la aplicación en ejecución.
- *Hierarchy*: pestaña en la que se puede comprobar la relación que existe entre los elementos del entorno virtual.
- *Project*: pestaña que muestra los elementos que contiene la carpeta *Assets* de la aplicación, permitiendo, además, poder navegar y reorganizar la misma.
- *Console*: consola donde se muestran los mensajes de error y los *prints* de la aplicación, entre otros.
- *Asset Store*: tienda donde se pueden descargar, ya sea de forma gratuita o de pago, distintos activos para el entorno, como, por ejemplo, modelados, materiales, animaciones, etc.
- *Inspector*: este último apartado es el que más utilidad tiene para los desarrolladores, pues es el que permite observar y modificar todos los atributos de cualquier elemento seleccionado en la escena.

Una vez explicados brevemente los elementos del entorno, es el turno de hablar del resto de ventajas que el motor ofrece durante el desarrollo de cualquier aplicación. Entre estas ventajas destaca el gran acierto de Unity de implementar un sistema basado en *GameObjects*, los cuales, aunque no logren nada por sí mismos, funcionan como contenedores para Componentes. (Unity Documentation)

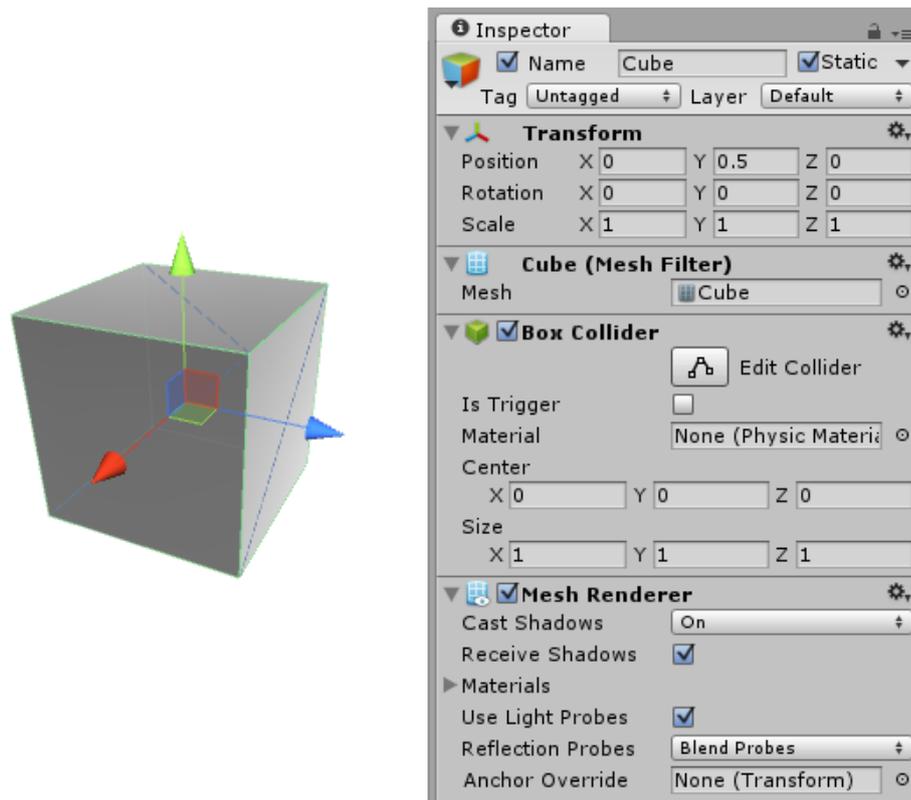


Ilustración 15. Ejemplo de GameObject y sus componentes.

Fuente: <https://docs.unity3d.com/uploads/Main/GameObjectCubeExample.png>

Dichos componentes, que implementan la verdadera practicidad del sistema, pueden agregar funcionalidades tan distintas como lo son las mallas del objeto, sus materiales, sus físicas, si ilumina o no, los scripts que hace un programador y su posición, orientación y escala, entre muchos otros. (Unity Documentation)

Los *GameObjects*, además de contener componentes, pueden ser etiquetados mediante el uso de *tags*, las cuales permiten identificar y vincular a uno o más objetos. El claro propósito de esta función es facilitar ciertas tareas de scripting en las que se hace necesario acceder a todo los elementos de una misma clase o bien saber qué tipo de *GameObject* es el que se recibe por parámetro. (Unity Documentation)

Así mismo, existe una opción dentro de los *GameObject* que permite identificar si dicho objeto se ve afectado o no por las físicas del juego. La función *Static*, que así es como se la identifica, resulta de gran utilidad a la hora de que la aplicación identifique, por ejemplo, que elementos se pueden mover y cuáles no. (Unity Documentation)

Otra función de utilidad es la de poder almacenar un elemento como "prefabricado". Cuando existe un *GameObject* que se va a repetir y usar con frecuencia y con unas componentes y características concretas, el desarrollador tiene la opción de guardarlo como plantilla, de modo que cada vez que vaya a necesitarlo, no tiene más que clonarlo. (Unity Documentation)

Del mismo modo, esta forma de trabajar mediante prefabricados hace posible que si durante el desarrollo de la aplicación se hace necesario un cambio en dicho elemento, cualquier edición hecha a la plantilla será inmediatamente reflejada en todas las instancias producidas por ella. (Unity Documentation)

Llegando casi al final de esta lista de utilidades ventajosas de Unity, se encuentra el elemento Canvas, el cual es un lienzo donde se colocan todos los elementos de la interfaz de usuario que incluya el desarrollador. Cabe destacar que este objeto es mostrado como un rectángulo en la Vista de Escena, lo cual facilita en gran medida posicionar los elementos de la interfaz sin necesitar tener una Vista de Juego todo el tiempo. (Unity Documentation)

Por último, es de vital importancia destacar el modo "Play" de Unity, que supone una herramienta de desarrollo increíblemente poderosa para la edición iterativa rápida mientras la aplicación ese encuentra en ejecución. Al pulsar Play, Unity lanza el programa, permitiendo así previsualizar como se verá en la compilación final específica de la plataforma. (Unity)

Así mismo, también permite pausar la ejecución para modificar los valores, activos, secuencias de comandos y otras propiedades convenientes pudiendo ver a continuación los resultados. El modo "Play" es, por lo tanto, un gran aliado a la hora de depurar el código de un programa. (Unity)

2.6.2.3. Características y ventajas del uso de Scripts

A todas las ventajas del motor en sí y de su entorno de trabajo, habría que añadir la posibilidad que ofrece Unity a los desarrolladores de escribir scripts y posteriormente asignarlos a *GameObjects* con el fin de dotarlos de cualquier funcionalidad programable deseada.

Este hecho resulta de gran utilidad cuando, a pesar de su versatilidad, los componentes que se pueden aplicar a un elemento se antojan insuficientes si queremos crear una mecánica de juego más personalizada o compleja. (Academia Android, 2015)

Es importante aclarar que un script no es más que un archivo de texto plano, formado por una serie de procedimientos y órdenes que proporcionan una funcionalidad específica y que mediante componentes de tipo Script, Unity puede coordinar el código con los modelos. Al tratarse de una pieza de código, esta puede asignarse a uno o varios *GameObjects* de la escena posibilitando así un abanico inmenso de variantes para el desarrollo de la aplicación. (Academia Android, 2015)

Unity soporta, principalmente, tres lenguajes de programación de forma nativa: C#, UnityScript (una "versión" particular de Javascript) y Boo. De estos es importante destacar que, en relación al motor, C# es el de uso más extendido, mientras que Boo ha sido discontinuado a efectos prácticos en las nuevas versiones de Unity, debido al pequeño porcentaje de usuarios que lo utilizaban. (Academia Android, 2015)

Por otra parte, cabe destacar la facilidad que ofrece el motor a la hora de crear y asignar Scripts a los elementos de la escena. Este proceso consiste básicamente en seleccionar el botón "Add Component", situado al final de la pestaña del inspector del *GameObject* seleccionado, y pulsar sobre la opción New Script. Tras esto, solamente se deberá indicar el nombre del script y el lenguaje de programación en el que se desea programar. (Academia Android, 2015)

Al crear un Script, este lo hará automáticamente con dos métodos por defecto: *Start()*, que se invoca justo antes del inicio del juego, y *Update()*, el cual es llamado en cada vuelta de ejecución de la aplicación. (Academia Android, 2015)

A parte de estas funciones, Unity tiene todo un largo catálogo de métodos predefinidos a las que el desarrollo puede acceder. Entre estos se encuentran algunos tan útiles como aquellos que permiten lanzar operaciones asíncronas o los que detectan desde eventos de ratón y teclado hasta colisiones con otros objetos.

Como última característica a comentar de los scripts, estaría el hecho de que a las variables locales que se definen como públicas se les puede asignar un valor desde el propio editor de Unity. Esto unido a la posibilidad de declarar variables de todo tipo (int, bool, transform, button, gameobject...) hace que desde el entorno de trabajo se pueda controlar y pasar cualquier tipo de elemento a los scripts para su posterior tratado.

2.6.3. Dispositivos visuales de Realidad Virtual

El último paso para disfrutar de una experiencia virtual inmersiva, una vez que ya se cuenta con los modelos tridimensionales creados, texturizados y ubicados en el entorno, es, lógicamente, un dispositivo que permita su correcta visualización.

Dado que ya se ha explicado su utilidad [anteriormente](#), en este apartado se explicarán, a grandes rasgos, los distintos tipos de cascos de realidad virtual, las características generales a tener en cuenta, otros datos de interés y, posteriormente, se detallará más concienzudamente el hardware elegido para la realización de este trabajo de fin de grado: Samsung Gear VR.

2.6.3.1. Clasificación

La primera clasificación viene dada según reproduzcan las imágenes sobre un ojo o dos. Aunque lo cierto es que esta clasificación se convierte en irrelevante a la hora de aplicarse a los dispositivos de realidad virtual. (Wikipedia)

Para que un casco sea realmente considerado de realidad virtual, debe tratarse obligatoriamente de un hardware binocular, es decir, para que la visión estereoscópica se produzca las imágenes que reproduce el dispositivo deben hacerlo sobre los dos ojos. (Wikipedia)

Si por el contrario fuera un dispositivo monocular y las imágenes solo se reprodujeran sobre un ojo, no podrían realizar la visión estereoscópica y, por lo tanto, sólo podrían usarse para la realidad aumentada. (Wikipedia)

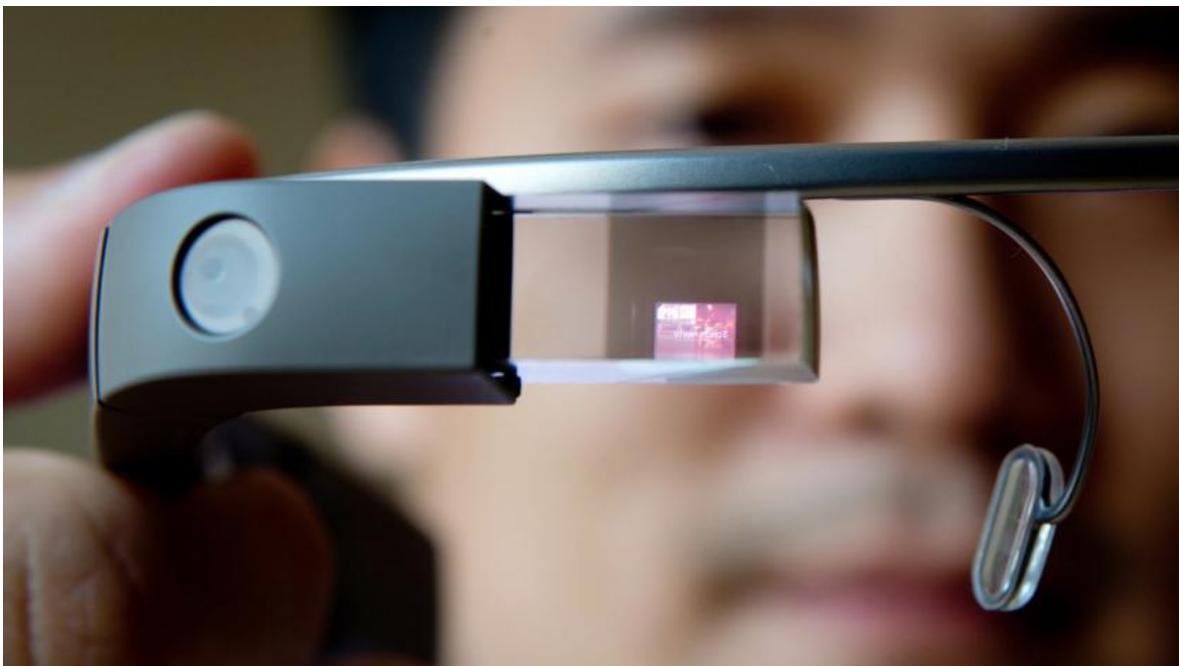


Ilustración 16. Dispositivo monocular.

Fuente: <http://www.objetivored.com/wp-content/uploads/2016/04/Gafas-Realidad-Aumentada-Google-Glass.png>

El siguiente método de clasificación tendría que ver, pues, sobre si los cascos se tratan realmente de realidad virtual y, en consecuencia, hacen que el usuario reciba únicamente información visual generada por computador, o bien son gafas de realidad aumentada y, por lo tanto, se limitan a complementar la información del mundo real con elementos virtuales. (Wikipedia)

Por último, los cascos de realidad virtual se pueden clasificar según su operatividad, pudiendo distinguirse entre móviles, sin procesador y autónomas. Mientras que los primeros son unas carcasas que no cuentan con pantalla propia que dependen de un teléfono móvil que pueden albergar en su interior, los dos últimos sí que poseen su propia pantalla. Por lo tanto, la diferencia entre estos dos reside en su procesador. (Wikipedia) (Navas, 2016)

Los cascos de realidad virtual sin procesador, también llamados dispositivos encadenados, dependen de un hardware externo para producir y generar el entorno que van a visualizar. De esta definición se desprende el hecho de que este tipo de dispositivo, por lo tanto, necesitará estar conectado por cables a su procesador. (Wikipedia) (Navas, 2016)



Ilustración 17. Ejemplo de dispositivo "encadenado".

Fuente: https://i.blogs.es/210757/vive-vr-wires/450_1000.png

Por descarte, los cascos de realidad virtual autónomos se tratan, pues, de aquellos dispositivos que no necesitan de ningún equipo externo para funcionar, posibilitando con ello un alto nivel de operatividad del hardware. (Wikipedia) (Navas, 2016)

Como ya se ha adelantado antes, el hardware elegido para visualizar el entorno virtual inmersivo para la detección y tratamiento de problemas de salud mental son las

Samsung Gear VR, las cuales, según lo explicado en este sub-apartado, se pueden catalogar como gafas de realidad virtual móvil binoculares.

2.6.3.2. Características y especificaciones

Al igual que con los ordenadores, los cascos de realidad virtual cuentan con una serie de características y especificaciones técnicas que sirven para aclarar su nivel de procesamiento o la calidad de la imagen proyectada, entre otros. Es por ello que en este apartado se dividirán según a que aspecto afectan.

2.6.3.2.1. Calidad de la imagen

El primer concepto clave de esta tecnologías es, como acabo de adelantar, la resolución de la pantalla, pues de ella depende la definición de la imagen percibida por el usuario. (Wikipedia)

Otra característica que interviene en la calidad de imagen es la de campo de visión. Los seres humanos tenemos, en general, un campo de visión horizontal de unos 180° a 220°. Sin embargo, hay que aclarar que esta es monocular y, por lo tanto, sólo es percibida por uno de los dos ojos. Si hablamos del campo de visión percibido por ambos ojos, este es de unos 114° y por este motivo, la mayoría de los cascos de realidad virtual funcionan con un campo de visión de entre 110° y 120°. (VR Lens Lab)

También en relación con la imagen que percibe el usuario se encuentra el refresco de pantalla, que mide el número de fotogramas mostradas por segundo. Por norma general, a partir de 60Hz se considera un buen ratio, aunque existen dispositivos de realidad virtual que llegan a alcanzar entre 90Hz y 120Hz. (Wikipedia)

Obviamente, otra característica vital de estos dispositivos la visión estereoscópica, la cual ya ha sido mencionada en repetidas ocasiones a lo largo de este documento y que,

recuerdo, consiste en mostrar una imagen ligeramente diferente a cada ojo permitiendo así visualizar el entorno en tres dimensiones (Wikipedia)

2.6.3.2.2. Sensores

Las siguientes especificaciones que se explicarán se encuentran más relacionadas con el apartado de funcionalidad de los distintos sensores que se encuentran o no instalados en los cascos de realidad virtual.

La latencia de seguimiento es un parámetro indispensable que mide el tiempo que transcurre entre que el usuario mueve su cabeza hasta que la imagen reproducida se reajusta a la nueva posición. Si la latencia fuera demasiado alta, podría llegar a producir mareos en los usuarios del sistema. (Wikipedia)

Del párrafo anterior se puede deducir que los cascos de realidad virtual cuentan con sensores que miden tanto la orientación (sensor interno) como la posición (sensor externo) de la cabeza del usuario. Estos permiten que la aplicación se pueda actualizar en base a los movimientos de la persona que lleva puesto el hardware. (Wikipedia)

Por último, destacar que algunos modelos avanzados de cascos incluyen unos sensores infrarrojos dentro del dispositivo que captan el movimiento de los ojos y, por lo tanto, permite realizar un seguimiento ocular del usuario. La utilidad de esta característica radica en que se logra replicar los movimientos de los ojos en los del avatar virtual, o provocar reacciones de otros personajes según la manera en la que se les mira. (Wikipedia)

2.6.3.2.3. Efecto rejilla

Para finalizar, cabe destacar que cuando los cascos de realidad virtual no cuentan con una tecnología suficientemente avanzada puede llegar a aparecer el efecto rejilla. Este

se produce en pantallas cuando las líneas que separan los píxeles de la misma se vuelven visibles en la imagen proyectada. (Wikipedia)

De todas formas, los expertos de la industria esperan que las pantallas pasen de ofrecer una calidad HD a prestaciones de pantalla 4K o incluso 8K en los próximos años, por lo que el efecto rejilla pasaría a formar parte del pasado. (Ohannessian, 2015)

2.6.3.3. Otros datos de interés

Ya con todas las características y especificaciones generales correctamente explicadas, llega la hora de comentar algunos datos de especial interés y relevancia para la población general.

2.6.3.3.1. Seguridad para los ojos

Tradicionalmente siempre ha estado la creencia que estar expuesto a una pantalla durante mucho tiempo daña la vista y es por ello que no extraña que a mucha gente le imponga respeto colocarse unos cascos de realidad virtual donde la pantalla queda a unos escasos centímetros de los ojos. (Pascual, 2016)

Es cierto que los primeros prototipos producían mareos en los sujetos, pero cuando se logró aumentar la tasa de refresco de pantallas estos desaparecieron. (Pascual, 2016)

La tecnología de realidad virtual se desarrolló durante más de cuatro años, en los que se probó y analizó intensivamente durante miles de horas, además de superar todos los controles de seguridad de la industria. (Pascual, 2016)

Las pantallas creadas para la realidad virtual utilizan tecnología OLED que emite una luz propia de baja intensidad y, además, las lentes situadas entre los ojos y la pantalla

actúan de escudo protector. Estos hechos las convierten en complemente seguras para la vista. (Pascual, 2016)

De todas formas, los propios fabricantes recomiendan descansar un rato cada media hora y no conducir o realizar tareas de riesgo hasta pasados unos minutos después de la utilización de las gafas. Esto es debido a que por la propia tecnología y las imágenes 3D que se usan, puede provocar incomodidad después de un uso prolongado, pero no es algo a lo que temer. (Pascual, 2016)

2.6.3.3.2. Edad mínima recomendada

La realidad virtual, como se ha demostrado a lo largo del documento, es una tecnología que implica al usuario dentro del entorno, con el cual puede interactuar. De esa afirmación se desprende la idea de que la persona que haga uso de la tecnología debe de poder visualizar correctamente las pantallas así como disponer de un control sobre el propio cuerpo para poder así relacionarse con el sistema.

Si nos centramos en la vista, hay que decir que la visión humana no está completamente formada para enfocar correctamente hasta los 7 años de edad. Sin embargo la realidad virtual sube a los 12 o 13 años por razones de madurez, pues exige taparse los ojos con unas pantallas y en algunos casos también andar y mover los brazos y los pies, lo que podría dar lugar a caídas y tropiezos. (Pascual, 2016)

2.6.3.3.3. Preparativos para el uso del dispositivo

Para finalizar esta sección, creo importante remarcar el hecho de que cuando se hace uso de un dispositivo de realidad virtual, este, por lo general, tapa completamente la visión del usuario. Por lo tanto, es importante que, antes de usar el sistema, se despejen las mesas y los objetos del suelo para evitar tirar cosas o tropiezos. (Pascual, 2016)

Los cascos de realidad virtual sin procesador requieren además de cables para conectarse con el ordenador o la consola, así que hay que despejar el camino entre el dispositivo y las gafas, teniendo en cuenta que un movimiento brusco puede hacer que los cables se enreden y tiren alguna cosa. También es recomendable evitar el uso de alfombras en la habitación. (Pascual, 2016)

Por último, si las gafas se usan sentado conviene utilizar una silla giratoria, que facilitará realizar giros de 360 grados. También es recomendable una mesa para dejar los mandos, en caso de que se haga uso de ellos, sin necesidad de quitarse las gafas. (Pascual, 2016)

2.6.3.4. Samsung Gear VR

Las gafas Samsung Gear VR, que fueron desarrolladas por Oculus por medio de una colaboración entre las empresas, son los cascos de realidad virtual que propuso la empresa surcoreana Samsung y que se clasifican dentro de las gafas de realidad virtual móvil binoculares. (Navas, 2016)



Ilustración 18. Samsung Gear VR.

Fuente: [http://images.samsung.com/is/image/samsung/in-gear-vr-r323-sm-r323nbkainu-000000003-1-perspective-black?\\$PD_GALLERY_JPG\\$](http://images.samsung.com/is/image/samsung/in-gear-vr-r323-sm-r323nbkainu-000000003-1-perspective-black?$PD_GALLERY_JPG$)

Aunque a simple vista parezca que solamente se encargan de sujetar el móvil en su interior, este modelo elegante y sólido tiene unas prestaciones que van más allá de su acabado final. (Pastor, 2016)

2.6.3.4.1. Ventajas

Este cómodo hardware, que posee un acabado muy bueno que privilegia el confort, cuenta con un campo de visión de 96º y un buen sistema de seguimiento de cabeza gracias a una serie de sensores internos específicamente dirigidos a esa tarea. (Pastor, 2016) (Navas, 2016)

Otra ventaja a tener en cuenta es la baja latencia que presenta el sistema, lo que provoca una mayor sensación de inmersión, así como que también haya menos propensión a potenciales mareos por experiencias prolongadas con este dispositivo. (Pastor, 2016)

En la parte derecha del hardware se encuentran una serie de controles que facilitan en gran medida la navegación por los menús de las distintas aplicaciones que se ejecutan en los cascos y, por lo tanto, ofrece una usabilidad mucho mejor. (Pastor, 2016) (Navas, 2016)

Por último, destacar que el simple hecho de que los cascos hayan sido desarrollados por Oculus y Samsung ya supone, en sí mismo, una ventaja, ya que cuenta con todo su apoyo en forma de contenidos. Y, todo ello, por menos de 100€. (Pastor, 2016)

2.6.3.4.2. Desventajas

La principal desventaja de estos cascos de realidad virtual móvil radica justamente en eso, en que sólo pueden funcionar con un dispositivo móvil insertado en su interior. En consonancia con este factor, se encuentra el hecho de que el hardware se diseñó para que

solo funcionara con los últimos modelos de los smartphones de gama alta de Samsung, los cuales pueden llegar a costar más de 800€. (Pastor, 2016)

3. Objetivos

Ya con el marco teórico y el estado del arte explicados, llega el momento de definir el objetivo general del proyecto, así como de los distintos sub-objetivos del mismo.

Como el propio nombre del trabajo sugiere, mi objetivo general es crear un entorno virtual inmersivo que pueda utilizarse para la detección y tratamiento de problemas de salud mental. Si lo concreto más, el objetivo podría definirse como la creación de un entorno geométrico tridimensional de un aula de colegio de primaria con el suficiente grado de personalización como para convertirlo en una posible herramienta para tratar casos de fobia en niños.

De este objetivo general, se desprenden una serie de objetivos secundarios, como lo son la investigación y comprensión de: las tecnologías de realidad virtual y aumentada, el tratamiento de fobias en psicología clínica; y el uso de sistemas de realidad virtual dentro del campo de la salud mental.

Otro objetivo secundario de gran importancia es el de crear un entorno geométrico tridimensional de una clase de colegio lo más fiel posible a la realidad. Éste además se puede desglosar en objetivos más pequeños: medir una clase de colegio real, visionar tutoriales de modelado, modelar los elementos por separado, texturizado de los elementos por separado y generar un código que permita colocar dichos elementos en base a unos parámetros.

Los parámetros mencionados en el párrafo anterior hacen referencia a otro objetivo secundario: crear un entorno virtual inmersivo lo suficientemente personalizable como para hacer viable su uso en psicología clínica. Es decir, mi intención es crear un sistema de realidad virtual que incluya un menú mediante el cual el terapeuta pueda elegir, entre otros parámetros, la disposición de las mesas o la existencia o no de ventanas en el aula.

Además, tengo como objetivo secundario, incluir modelado de profesor y alumnos dentro del entorno, para facilitar así que los usuarios/pacientes puedan percibir el espacio geométrico tridimensional como real.

El último objetivo secundario consiste en hacer posible que el entorno virtual se pueda visualizar mediante dispositivos de realidad virtual. Para ello usaré Unity junto con las Samsung Gear VR.

Como se puede apreciar, cada uno de los objetivos me exige unas competencias bien diferenciadas pero que, en conjunto, me permitirán crear un entorno virtual inmersivo que pueda servir realmente como una herramienta en el tratamiento de problemas de salud mental.

4. Metodología

Al igual que con el marco teórico, en el apartado de metodología se ha optado por separarlo todo en secciones y sub-secciones para facilitar su organización y la comprensión a la hora de leerlo. Así mismo, al dividirlo se puede entender mejor todo el trabajo realizado así como el orden que se ha seguido a la hora de llevarlo a cabo.

Antes de pasar a describir todos y cada uno de los trabajos realizadas, me gustaría aclarar que la forma de proceder ha sido mediante la asignación de tareas y tiempos, es decir, se dividió todo el proyecto en partes y a cada una de estas se le estimó un tiempo de desarrollo y una fecha límite.

¿Por qué un tiempo de desarrollo y una fecha límite? De esta forma se sentía más libertad temporal a la hora de afrontar una tarea, puesto que sólo con cumplir una de las dos limitaciones de tiempo significa que el trabajo avanzaba a buen ritmo. Así mismo, dado que en multitud de ocasiones se terminaba una tarea en una fecha distinta a la límite, el resto se modificaban para ajustarse a la nueva temporalidad. Debido a esto, en la siguiente tabla solo se contemplará el tiempo estimado.

Tarea principales	Tiempo estimado
Documentación	25 horas
Creación de los elementos del entorno	65 horas
Creación del entorno	110 horas
Realización de la memoria	100 horas
Horas totales:	300 horas

Tabla 1. Relación de tareas principales y sus tiempos estimados

Evidentemente no siempre fue posible cumplir los tiempos, pero se han intentando respetar lo máximo posible y, finalmente, se puede afirmar que se le ha dedicado al desarrollo de este trabajo el tiempo que se exigía.

También me gustaría aclarar que para este proyecto no se han usado repositorios ni control de versiones propiamente dichos. Al tratarse de un trabajo de una sola persona se tornaba imposible que otro "miembro del equipo" pudiera afectar el código.

De todas formas, sí que se han ido realizando copias de seguridad, tanto de los modelados como de los archivos de Unity, incluso de esta misma documentación, a medida que se producían avances significativos. Dichas copias de seguridad, además, se almacenaban en tres sitios diferentes (en mi ordenador habitual, en un disco duro externo y en la nube), con el fin de asegurar que nunca se perdía nada de información.

Otro dato que debo comentar es el público objetivo final de la aplicación que se ha obtenido. Dado que se trata de una herramienta que posibilita el uso de realidad virtual para el tratamiento de fobias, es evidente que la población a la que le puede interesar la misma son los psicólogos y terapeutas.

Así mismo, al tratarse de un entorno que simula un aula de un colegio de primaria, se puede afirmar también que otro sector de la población que puede beneficiarse de este trabajo de fin de grado son los niños y niñas en edad escolar que padezcan fobias tan serias como para que les pueda llegar a afectar en el ambiente del colegio.

Una vez aclarado todo esto, ya puedo proceder a explicar la metodología de trabajo que se ha seguido para la creación de un entorno virtual inmersivo para la detección y tratamiento de problemas de salud mental.

4.1. Documentación

Este trabajo de fin de grado tiene la gran virtud de combinar dos aspectos tan diferenciados dentro del ámbito académico como lo son la psicología clínica y la realidad virtual. Pero ese hecho también tenía un inconveniente: mientras que sí que se tenían nociones básicas de la realidad virtual, me encontraba completamente indefenso en el tema de la psicología clínica, sobre todo teniendo en cuenta de que el tratado de

desordenes mentales mediante entornos virtuales inmersivos es una rama relativamente nueva dentro del campo.

Por ello, la primera tarea que se realizó en el desarrollo de este proyecto fue la de documentación sobre el tema. Se buscó información sobre la psicología clínica en general, sobre el caso concreto del tratamiento de fobias de forma tradicional, sobre el uso de realidad virtual en las mismas y sobre sus ventajas y desventajas en relación a las metodologías convencionales.

El principal método de obtención de dicha información consistió en la búsqueda por Internet de documentos, libros, tratados, estudios, artículos y entrevistas que trataran del tema en cuestión y en su posterior lectura y estudio.

La tarea de documentación tenía asignada una gran cantidad de tiempo debido a que, dada la complejidad del tema a desarrollar, se consideró preferible invertirlo en este cometido para que las decisiones tomadas a continuación fueran, en principio, sobre unos argumentos sólidos.

Creo conveniente indicar que durante todo el proceso de búsqueda de documentos, se fue realizando de forma paralela la parte correspondiente al marco teórico y estado del arte, con el fin de agilizar posteriormente la redacción de este documento.

El resultado de todo este proceso fue la adquisición suficiente de información como para diseñar un entorno virtual inmersivo con unas opciones de personalización suficientes para que fuera considerado una herramienta eficaz y flexible.

4.2. Creación de los elementos del entorno virtual

Una vez que se tuvo el entorno planificado, tanto a nivel visual como a nivel de funcionalidades, dio comienzo la segunda gran tarea de este trabajo de fin de grado: la creación de todos los elementos del entorno virtual. Dentro de este bloque, se incluye el modelado 3D, el mapeado UV, el texturizado y la asignación de materiales de todos y cada uno de los modelos tridimensionales.

4.2.1. Modelado

Como se acaba de adelantar, el primer paso fue el modelado de los objetos del aula del colegio, tanto de aquellos que suponían elementos estructurales de la clase como de aquellos que servirían para rellenar el entorno.

Para llevar a cabo esta tarea se optó por la utilización del programa 3ds Max 2017, debido a que ya se conocía el entorno y, por lo tanto, no habría que dedicar un tiempo al aprendizaje. Así mismo, se eligió esta potente herramienta por la calidad de los modelados que se pueden obtener con su uso y porque, además, se puede acceder a ella de forma gratuita.

Cabe destacar que para el modelado de los elementos de la clase se llevaron a cabo diversas técnicas de modelado, aunque la gran mayoría de los mismos se realizaron aplicando deformaciones y transformaciones sobre cuerpos geométricos básicos, como los cubos y los cilindros.

Dado que el entorno virtual no contaría con un número excesivo de elementos de mallas complejas, se consideró que sería conveniente intentar conseguir un alto grado de realismo en todos los modelados, prestando especial atención a todos los que incluían superficies curvas en sus diseños. Esta decisión, tendría consecuencias negativas en las últimas fases, pero se comentarán posteriormente.

Me gustaría comentar el hecho de que hubo ciertos objetos que presentaron una especial dificultad a la hora de modelarlos y que, por lo tanto, llevaron más tiempo del esperado. Entre estos destacaría la rejilla que tienen todas las mesas de los alumnos debajo, debido a la complejidad del diseño, y la forma del asiento de las sillas de los alumnos, pues casi todas las líneas de su geometría eran curvas.

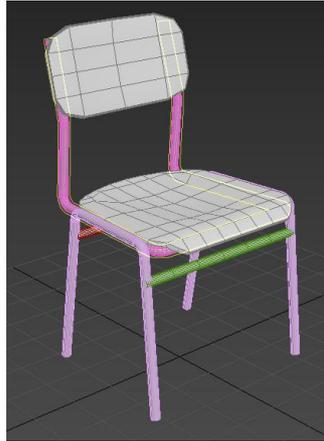


Ilustración 19. Modelado de una silla de alumnos.

Fuente: propia

Una vez superadas las dificultades, se pudo terminar esta fase de la creación de los elementos del sistema. Creo importante destacar el hecho de que en total se realizaron veintisiete modelados distintos que contemplaban desde la estructura de la clase en sí misma hasta cada uno de los objetos que posteriormente se utilizarían para terminar de completar el entorno.

Por último, se decidió que aquellos elementos que se usarían para rellenar las el mobiliario y las mesas lo harían siempre en el misma posición, por lo que para agilizar el proceso de scripting se juntaron dicho objetos en un archivo separado donde se fueron copiando y organizando hasta conseguir las organizaciones deseadas. Cuando estas se obtuvieron se exportaron y se les asignó un nombre que les identificara como objetos de relleno, así como para qué mueble serían usados.

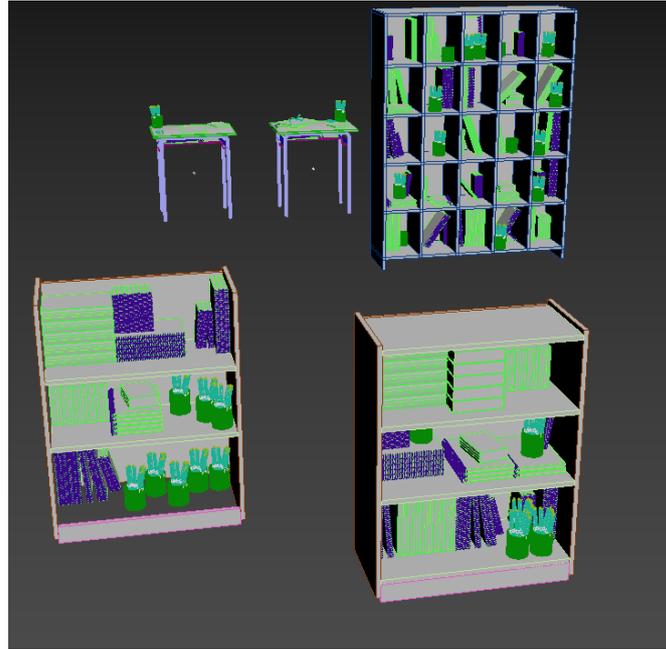


Ilustración 20. Modelado de muebles y sus objetos.

Fuente: propia

4.2.2. Texturizado

Ya con todos los modelados con el nivel de realismo deseado, se pasó a la tarea de desenvolverlos para su posterior texturizado.

He de admitir que en este ámbito aún no había realizado ningún trabajo y, por lo tanto, el tiempo que estimé para la misma resultó excesivamente ajustado para el mismo. Pero tras unos días de bloqueo, finalmente se pudo empezar la tarea con buen ritmo.

Para el mapeado UV se usó el modificador *Unwrapping UV* que el software 3ds Max trae incorporado de serie. Por experiencia propia, comentaré que aunque el modificador realiza la tarea de forma automática, siempre se obtienen mejores resultados si se le indica manualmente al programa donde debe realizar los cortes para desenvolver el modelado.

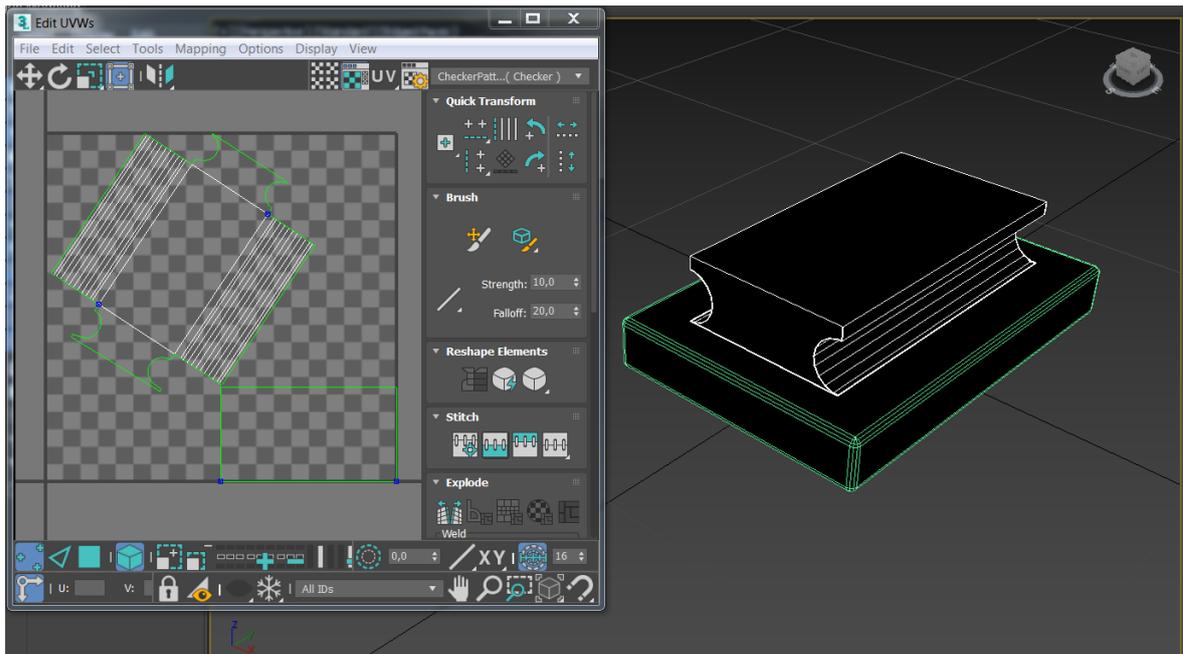


Ilustración 21. Modelado de un borrador junto con su mapeado UV.

Fuente: propia

Una vez se llegó a esa conclusión no hubo más que abrir de uno en uno todos los modelados que se habían realizado haciendo especial hincapié en las juntas y en aquellas zonas más conflictivas de la malla. Cabe destacar que en aquellos modelados que contenían partes de distintos materiales, se optó por separarlos en trozos y hacer el mapeado de cada uno de esos trozos por separado.

Cuando todos los elementos estuvieron desenvueltos, se les aplicó un material estándar a cada una de las partes en las que estaban divididos, en caso de que lo estuvieran. Hay que decir que a cada material se le dio un nombre identificativo que indicaba con claridad a qué parte de qué modelo se le estaba aplicando, con el fin de agilizar posteriormente las tareas de organizado y asignación de los mismos.

Al terminar de aplicar los materiales, se exportaron todos los objetos en formato .fbx para su posterior importación en el programa Substance Painter, el cual fue el elegido para las tareas de pintado de texturas.

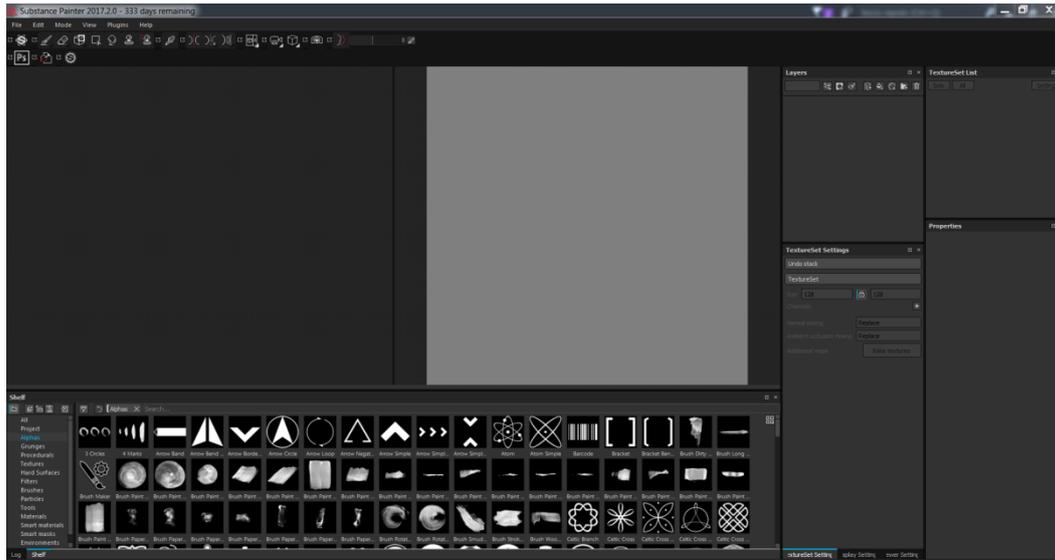


Ilustración 22. Interfaz del programa Substance Painter.

Fuente: propia

Este software especializado fue el escogido diverso a varias razones de peso. Entre estas, se encuentra el hecho de que permite pintar texturas, además de sobre el mapa UV, sobre la propia malla. Otro factor que se tuvo en cuenta es que, al igual que Photoshop, permite trabajar en capas, a las que se les puede dar distintos efectos para conseguir el acabado deseado. También es importante destacar el hecho de que posee una gran variedad de materiales y pinceles por defecto que le dotan de gran versatilidad.

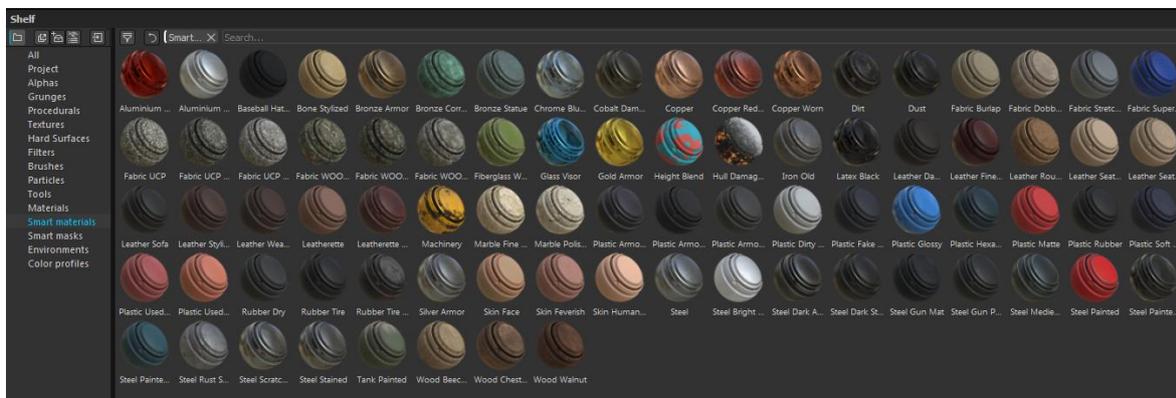


Ilustración 23. Materiales por defecto del programa Substance Painter.

Fuente: propia

Todos estos factores sumados al hecho de que se puede conseguir de forma gratuita y a la facilidad de su aprendizaje, le convirtieron en el programa ideal para el pintado de las texturas.

Por norma general, con los materiales por defecto del programa, aunque casi siempre con alguna pequeña modificación, fue más que suficiente para lograr el grado de realismo que se buscaba en las texturas.

Este paso era vital para que el acabado final de la clase tuviera el suficiente grado de realismo como para garantizar la total inmersión y sensación de presencia de los pacientes con los que se usara el entorno.

También me gustaría destacar el hecho de que, aunque no se haya realizado un cocinado como tal en este proyecto, sí que se aplicaba el *bake* a aquellas texturas que pertenecían a mallas bastante complejas, con la intención de que, a la hora de pintarlo, se pudiera apreciar mejor el alto nivel de detalle de los modelados.

Tras el pintado de todas las texturas de cada modelado, se debía exportar el resultado. Otra de las ventajas de Substance Painter, y que anteriormente no se ha citado, es el hecho de que permite seleccionar para qué programa se desea exportar, generando así los mapas de texturas necesarios para cada caso.

En la casuística concreta de este trabajo, bastaba con seleccionar que los importados se usarían en Unity 5 para que el mismo programa generase de forma automática el albedo, el mapa de normales y el *metallic-roughness*, los cuales son vitales para la correcta visualización de las texturas en dicho motor.

4.2.3. Personajes

Este apartado, aunque se encuentre seguidamente después del texturizado, debo admitir que no fue en esa posición cuando se llevó a cabo. Debido a la gran complejidad del modelado orgánico de personajes humanos, se tomó la decisión de posponer esta tarea hasta que se encontrara una alternativa viable.

Dicho momento no llegó hasta momentos antes de iniciar el proceso de scripting en Unity. Con esto pretendo decir que hubo una gran cantidad de tiempo entre el modelado del resto de elementos y de los personajes y que, por ese motivo, aunque los resultados sean los buscados, no se le pudo dedicar tanto tiempo como el deseado.

El software específico que posibilitó que esta tarea se llevara en un periodo de tiempo relativamente reducido es MakeHuman, un programa especializado en el modelado de personajes humanoides.

Se eligió este programa debido a que se especializaba únicamente en el aspecto que más convenía para el proyecto y la gran facilidad que presenta en un uso. Prácticamente se puede diseñar un personaje en apenas media hora y sin tener ningún tipo de conocimiento previo sobre la interfaz y utilidades del mismo.

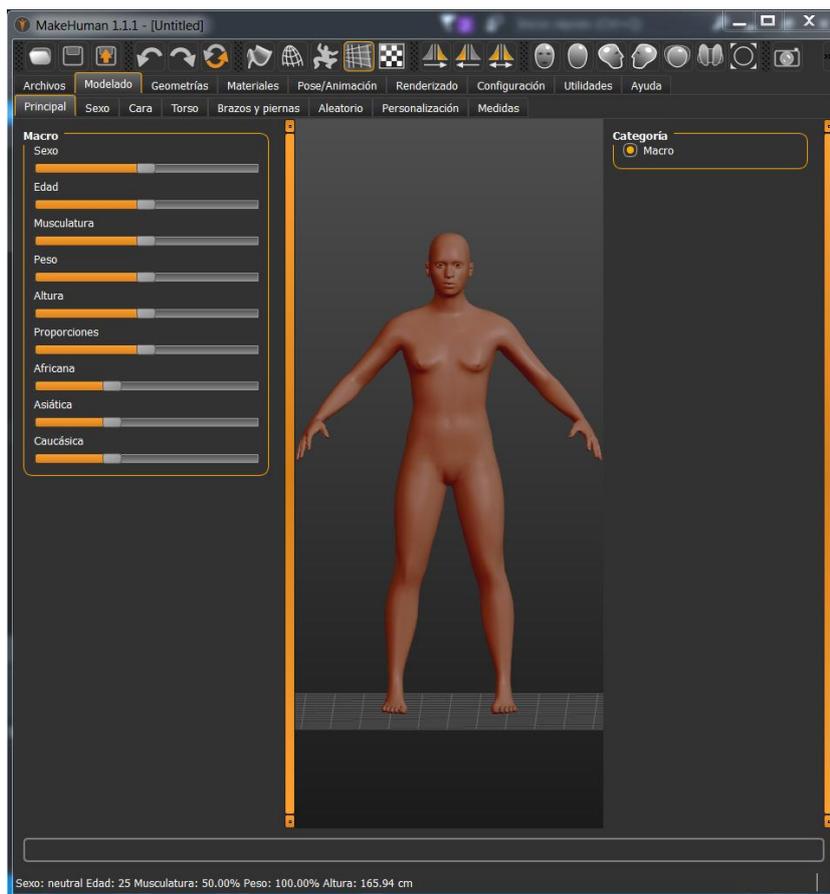


Ilustración 24. Interfaz del programa MakeHuman.

Fuente: propia

MakeHuman permite crear un modelo humano prácticamente desde cero y con una gran cantidad de intuitivos parámetros al alcance. Entre estos se encuentra desde la altura o complexión del modelad hasta la selección de edad que aparente y el género, pasando por la gran cantidad de variables que posee para poder modificar todas y cada una de las partes del cuerpo.

Además, al tratarse de un software libre gratuito, es relativamente sencillo descargar e incluir nuevas geometrías y variables en el editor del programa. Para ello no hay más que acudir a la misma página web del programa, descargarse los distintos ficheros necesarios para el complemento que se desee añadir, y después guardarlos en la carpeta indicada del programa.

Como dato curioso, indicar que para agilizar este proceso de descarga de añadidos, se precisó de la instalación del navegador Opera, el cual posee la opción directa de guardar el contenido de un enlace haciendo click derecho sobre el mismo.

Centrándonos ya en el modelado de los personajes para el entorno virtual inmersivo de la clase de colegio, informar de que para ello se realizaron un total de cuatro modelos: un niño, una niña, un hombre adulto y una mujer adulta. De estos, a los dos primeros se le realizaron distintas variaciones hasta lograr obtener finalmente cinco modelos distintos de niños, cinco de niñas, uno del profesor y un último de la profesora, haciendo un total de doce.

Creo importante comentar el hecho de que se diseñó una nomenclatura específica para los modelos humanos con la intención de que fuera fácilmente ampliable y catalogable la generación de nuevas mallas humanoides. Aunque, finalmente, dadas las limitaciones de tiempo, y del software, no se pudo implementar correctamente.

Con esto último me refiero a que dentro del programa, a pesar de la gran utilidad y versatilidad del mismo, había ciertas geometrías y texturas que no terminaban de causar el grado de realismo que se perseguía, por lo que se tuvieron que desechar ideas que en un principio estaban plasmadas en la nomenclatura.

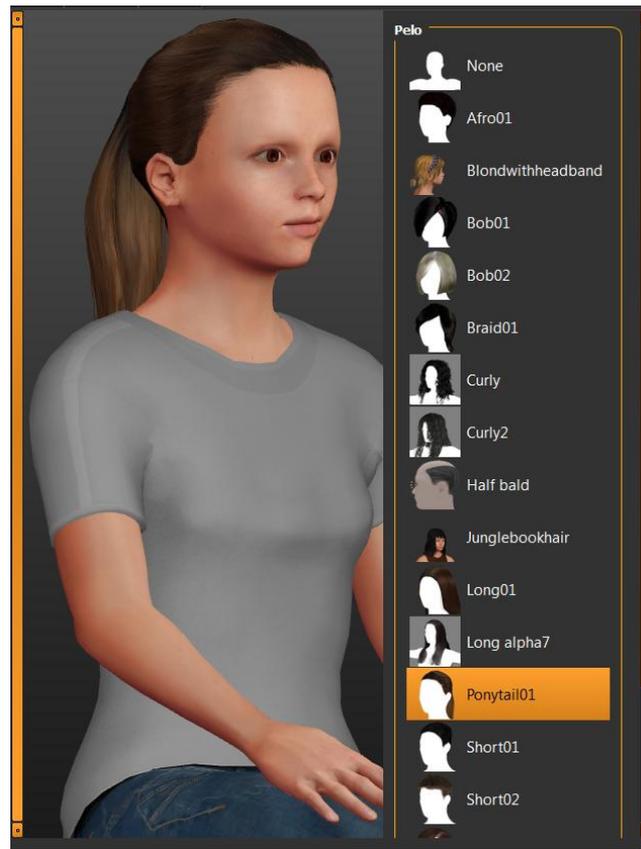


Ilustración 25. Modelado de alumna junto con las opciones de geometría disponibles para el pelo.

Fuente: propia

Otro hecho relevante que conllevó a la eliminación de la nomenclatura fue que al comienzo del modelado de los personajes se consideró que estos pudieran sentarse o bien con las piernas cruzadas o bien rectos, con los dos pies en el suelo. Esta idea fue eliminada debido a que a la hora de exportar los modelos, estos poseían ligeras variaciones en la posición que dificultaban la tarea de scripting para su creación y colocado, por lo que se optó por colocarlos a todos con ambos pies en el suelo.

Aún con estos inconvenientes de última hora, el software demostró su utilidad en el proceso de creación de personajes y supuso una ayuda indispensable para la correcta finalización a tiempo de este trabajo de fin de grado.

4.3. Creación del entorno virtual

La tercera y última gran fase del desarrollo del entorno virtual inmersivo para la detección y tratamiento de problemas de salud mental es la relativa a la generación automática, en base a unos parámetros, del mismo.

Para ello se ha contado con el motor de videojuegos multiplataforma Unity, que, a pesar de que esté pensado para el desarrollo de juegos, cuenta con las herramientas necesarias para realizar cualquier tipo de aplicación de realidad virtual.

Por último, me gustaría indicar que a lo largo de este apartado se mencionan opciones del menú y decisiones tomadas en relación al entorno que se encuentran explicadas posteriormente en el apartado [5.1. Resultados de la documentación](#). Pero, con la intención de que resulten aclaratorias, se han incluido imágenes del producto final para facilitar el entendimiento de cada una de las partes explicadas.

4.3.1. Editor de Unity

Tras crear y configurar el proyecto de Unity, se importaron todos los modelados creados junto con sus texturas. Una vez estuvo todo importado, se recorrieron todos los materiales añadiendo a sus casillas de albedo, normal map y *metallic-roughness*, los mapas correspondientes generados durante el proceso de texturizado.

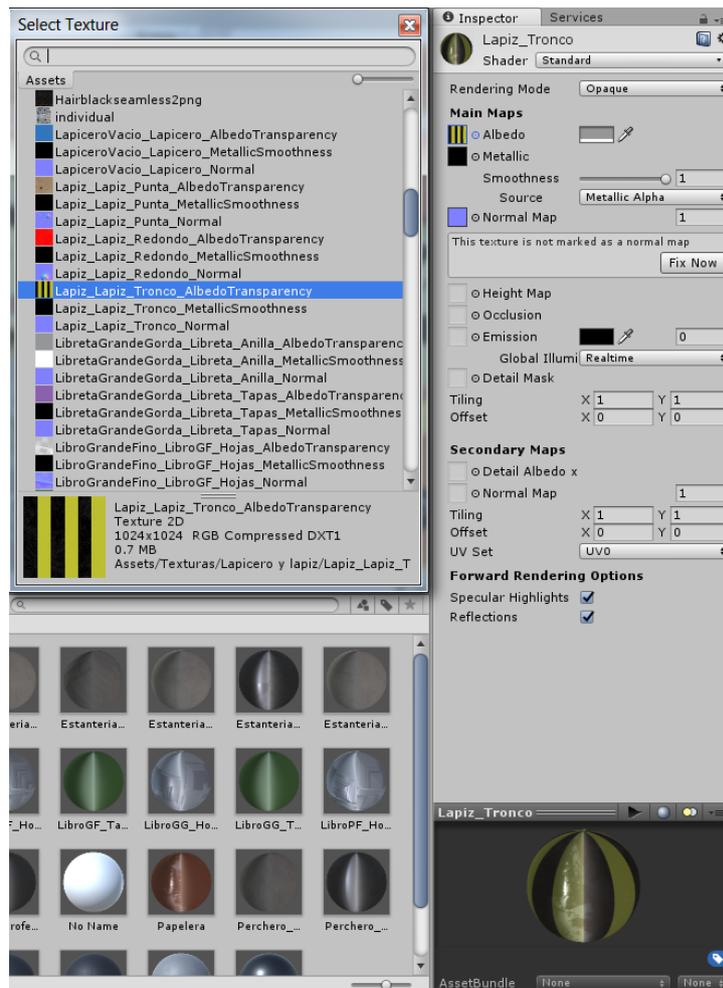


Ilustración 26. Configuración de los materiales en Unity.

Fuente: propia

Cuando estuvieron todos los materiales correctamente ajustados, se fueron colocando los distintos elementos del entorno, de uno en uno y de forma aproximada, en el sitio que les correspondía dentro del entorno.

Este primer paso inicial de colocado, resultó vital para comprobar que los tamaños, así como sus texturas, eran los adecuados para su correcta visualización. Gracias al buen trabajo de modelado y texturizado, no hubo que hacer ningún cambio en ninguno de los objetos tridimensionales y, por lo tanto, fue un tiempo que se pudo invertir en otros aspectos.

Una vez realizada esta primera comprobación rápida, se procedió a la colocación de los modelos, esta vez de forma muy precisa, en el entorno. Para este caso, se hizo un

uso intensivo de la vista ortográfica, la cual elimina la perspectiva de la escena, dado que con ella es más sencillo ajustar los límites de los objetos y su posición dentro del entorno.

Al llegar unas posiciones que se consideraron adecuadas para el entorno que se perseguía, se procedió a la anotación de las coordenadas de posición y de la información de la rotación de todos y cada uno de los elementos de la escena. Apuntando también la relación que existía entre algunos de estos objetos, como, por ejemplo, la relación entre la posición de una silla y su mesa correspondiente.

MODELADO	POSICION			ROTACION			
	x	y	z	x	y	z	
Clase	0	-0,065	0	0	0	0	
Techo	0	3	-0,285	0	0	0	
Ventanas	$\pm 0,515$ $\pm 3,05$	0,985	-3,167	0	0/180	0	
Puerta	3,81	0	3,167	0	0	0	
Pizarra	4,38	0,6	1	0	-90	0	
Borrador	"-0,08"	"	"	0	0	0	
Papeleca	$\pm 4,2$	0	2,9	-90	0	0	
Pizarra digital	4,38	0,75	1,5	-90	0	-90	
Mesa y obj prof	3	0	-2	0	90	0	
Escala y=1,2	Silla profes*	" +0,54"	0	" +0,12"	0	-90	0
Estanterias	3,98	0	-2,937	0	0	0	
profesor	3,14	0	-2,937	0	0	0	
Estanterias	1,35	0/1,08	2,95	0	180	0	
	-2,6	0/1,08	2,95	0	180	0	
	-3,45	0/1,08	2,95	0	180	0	
	-4,2	0/1,08	-2,2	0	90	0	
Casillero	-4,2	0	-1	-90	0	90	
Percheros	-0,9/0,15	1	3,105	0	180	0	
	-4,335	1	0,75/180	0	90	0	
Profesor	* -0,05	0,048	*	0	*	0	
Profesora	* -0,05	0,018	*	0	*	0	

Ilustración 27. Anotaciones de las posiciones y rotación de los elementos del entorno.

Fuente: propia

Esta tarea, que resultó más larga y ardua de lo esperado, sirvió para agilizar posteriormente la tarea de scripting, pues, cuando llegó el momento de programar el código de colocación de los elementos, ya se disponía de sus posiciones y de la relación entre las mismas. Cabe decir que precisamente en el caso de las distintas distribuciones de las mesas de los alumnos fue cuando demostró su gran utilidad.

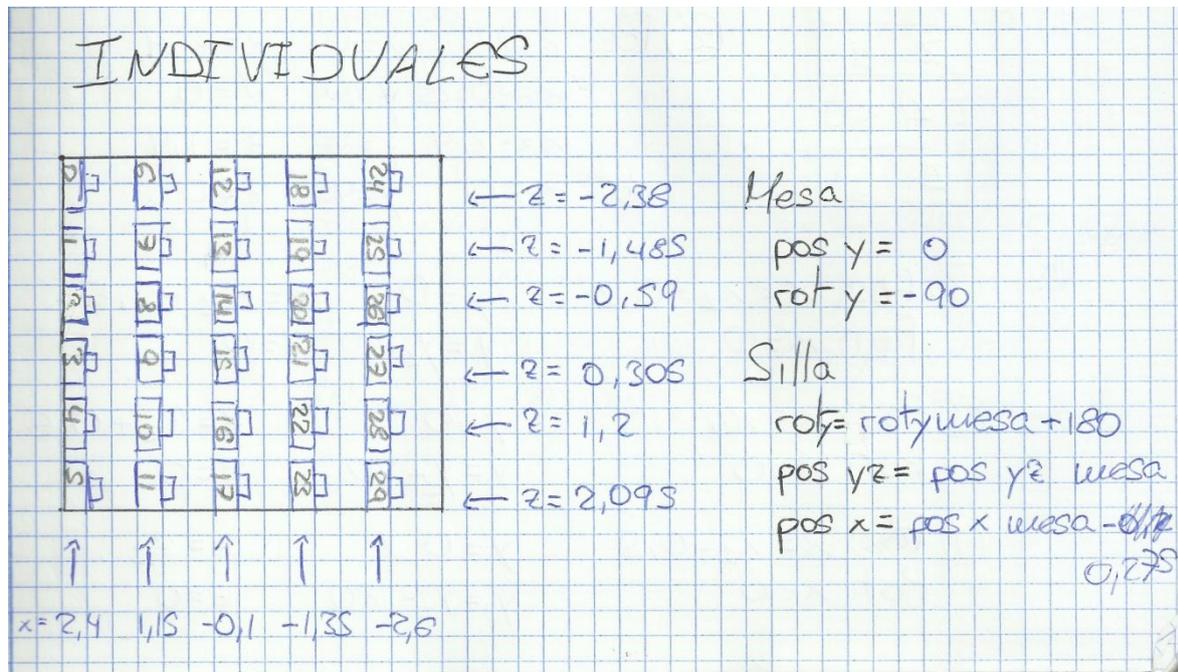


Ilustración 28. Anotación de las posiciones y rotación de mesas y sillas para el caso de distribución individual.

Fuente: propia

El siguiente paso, fue el de crear una jerarquía útil y eficaz entre los distintos elementos de la escena. Para ello se fueron agrupando los modelos según su funcionalidad dentro de la clase. Así mismo, también se tuvieron en cuenta los parámetros de personalización que se iban a implementar, con el objeto de hacer más sencilla la tarea de programación del código del entorno.

Otra de las tareas que se realizaron con el editor fue la de crear un gran número *GameObjects* vacíos que servirían como "padres" de los distintos elementos que se instanciarían justo antes de la ejecución del programa y después de que el terapeuta seleccionara los parámetros de la sesión en curso.

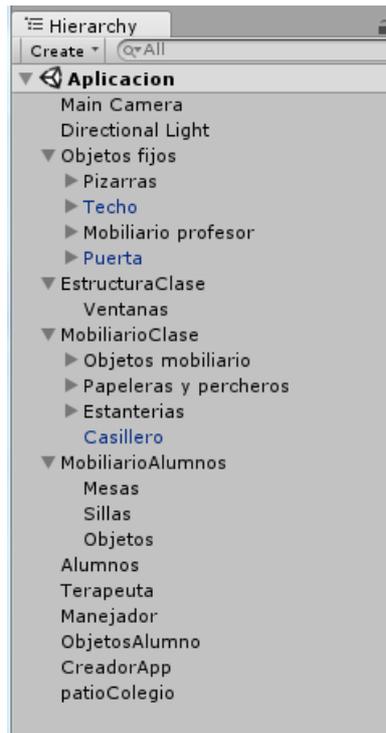


Ilustración 29. Jerarquía de la aplicación, sus grupos y sus GameObjects vacíos.

Fuente: propia

Así mismo, también se crearon las luces que iluminarían el entorno: una exterior a la clase que simularía la luz solar y otras tantas interiores, una por cada lámpara, con el fin lograr un aula con una iluminación lógica y natural.

Por último, se colocó una imagen de gran tamaño en el lado que ocupaban las ventanas y que, por lo tanto, funcionaría de fondo del entorno. Dicha imagen consistía en un patio de colegio, facilitando así que los pacientes puedan reconocer el entorno como si de un recinto escolar real se tratara.

4.3.2. Interfaz de usuario

Antes de comenzar a describir cómo se llevó a cabo la tarea de crear el interfaz de usuario, querría aclarar que este paso en realidad se realizó conjuntamente con la

programación del código que permitía la navegación, pero que, por cuestiones de continuidad dentro del informe, he decidido colocarlo justo antes del scripting.

Lo primero que se hizo para crear la interfaz del usuario fue decidir cuantas pantallas tendría el menú inicial de la aplicación y cómo se plantearían estas. Para darle una mayor continuidad a todo el proceso, se creyó conveniente usar los propios modelos como fondo de dichas pantallas, lo cual también significó que no era necesario crear ningún elemento más para el entorno.

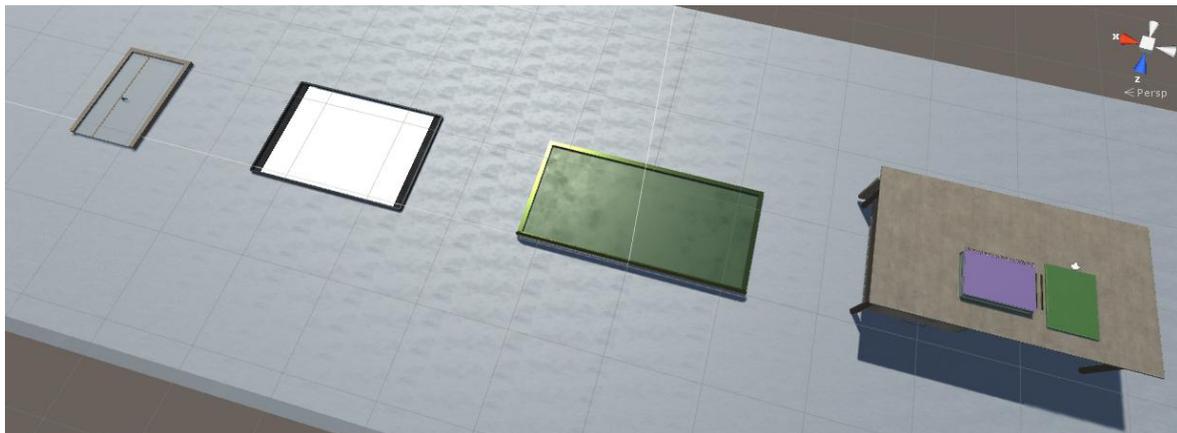


Ilustración 30. Modelados del entorno que se usaron como fondo para el menú.

Fuente: propia

Una vez se asignaron los fondos a cada pantalla, se procedió a la organización de los distintos botones y funcionalidades dentro de estas, siempre con la pretensión de llegar a un diseño de interfaz que fuera sencillo e intuitivo.

Tras realizar esta tarea, se crearon los distintos elementos y se colocaron dentro del canvas de Unity, preparados para recibir sus funcionalidades a través de los distintos scripts que se realizarían.

Cabe destacar que este paso presentó una serie de dificultades provocadas por el hecho de contar con distintas pantallas dentro del menú. El hecho de plantear las distintas pantallas en una única escena de Unity, con la intención de facilitar el scripting, resultó ser un quebradero de cabeza en el editor.

La solución que se encontró al respecto fue la de trabajar siempre desde el modo de vista *Game*, que muestra cómo se vería la aplicación en ejecución, y colocar los distintos elementos en su sitio correspondiente dentro de cada pantalla. Éstas, por lo tanto, simplemente suponen un cambio de posición de la cámara para cambiar de una a otra.

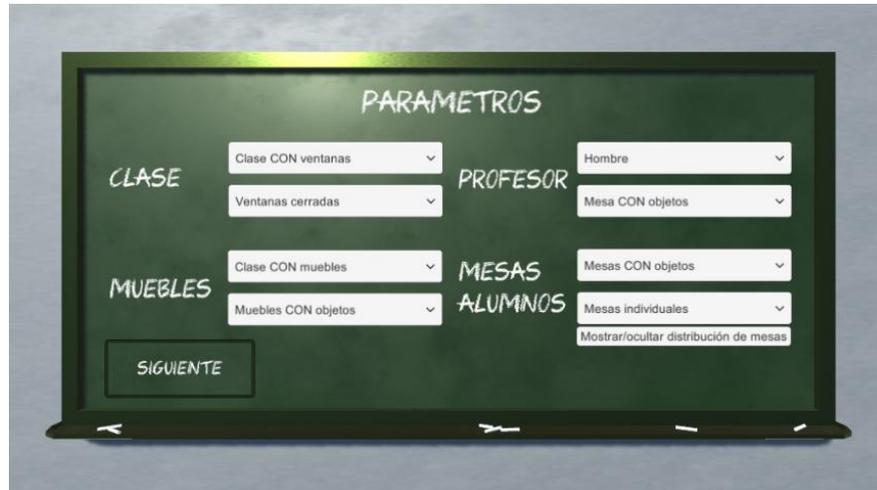


Ilustración 31. Pantalla principal del menú de la aplicación.

Fuente: propia

Otra gran dificultad fue la de crear una interfaz de carga de sesiones donde se pudiera crear y ubicar correctamente un botón para cada sesión guardada. A pesar de los intentos que se realizaron, finalmente se acudió a un tutorial de Youtube para poder llevarlo a cabo.



Código QR 1. Unity UI Tutorial - How to make a scrollable list.

Fuente: <https://www.youtube.com/watch?v=IUun2xW6FJ4>

4.3.3. Código

Este apartado se puede considerar como el más importante dentro del desarrollo de este proyecto, ya que mientras que el resto de tareas se centraban en crear los elementos del entorno, o bien en facilitar su creación, esta supone realmente la funcionalidad de toda la aplicación.

Dada la gran relevancia del código dentro de este entorno virtual inmersivo para la detección y tratamiento de problemas de salud mental, se ha optado por dividir el apartado en cuatro secciones según a que fuera dedicado el scripting.

4.3.3.1. Colocación de elementos

El primer objetivo que se persiguió fue el de crear un código que generara de forma automática el entorno con todos sus elementos dentro del mismo y perfectamente ubicados en la posición asignada.

En general esta tarea resultó sencilla, a excepción de la generación de mesas y sillas pues, al poder estar las mesas colocadas en distintas distribuciones y orientaciones, había que controlar que la posición de las sillas tuviera en cuenta dichas orientaciones para saber en qué eje se debía desplazar para ubicarse en el lado correcto.

Cabe destacar que se realizaron scripts distintos para cada sección del entorno, mientras que uno controlaba la generación de los muebles, otro lo hacía de las mesas y otro se encargaba de cargar las paredes y ventanas.

4.3.3.2. Adaptado del código para los parámetros ideados

Cuando se consiguió llegar a un entorno con todos sus objetos tridimensionales colocados perfectamente y de forma automática, se prosiguió editando los scripts para

incluir en ellos variables que permitieran llevar a cabo los deseos del terapeuta para con la clase virtual.

Para facilitar esta tarea, se creó un script separado al que se le iban añadiendo todas las llamadas al resto de clases, con el fin de que solamente hiciera falta modificar ese código para controlar todos los parámetros del entorno.

En este proceso hubo ciertas dificultades debido a problemas de diseño del código o a falta de comprensión de algunas opciones preestablecidas por Unity. Entre estas dificultades destacaría todas aquellas que tienen que ver con los arrays.

Muchos de los elementos del entorno, al tratarse de un mismo elemento repetido, se almacenaron en vectores, pero, había ocasiones, en los que, por necesidades del entorno, ciertas posiciones del array debían quedarse vacías. Para dichos casos, primeramente se decidió que se les asignaría un valor de "null", lo cual funcionó al principio.

Después, resultó que si otro método necesitaba crear un objeto en una posición almacenada como "null", éste generaba errores. Al final, se optó por crear *GameObjects* vacíos en aquellas posiciones conflictivas, pues no provocaba ninguna dificultad después, y, una vez que se hiciera el uso correspondiente al vector, se recorría el mismo para eliminarlos.

A excepción de esta pequeña dificultad, el adaptado del código para que fuera capaz de llevar a cabo las opciones de personalización elegidas por el terapeuta resultó más sencillo de lo esperado, en parte debido a la gran sencillez que aporta el propio motor de videojuegos Unity.

4.3.3.3. Navegación por el menú

Cuando ya estuvo todo el entorno preparado para recibir las órdenes, se pasó a programar la navegación principal por el menú. Para ello se hizo uso de los botones y de

la asignación de etiquetas a los mismos con el fin de que fuera más sencillo reconocer su función en base a las mismas.

Una vez se consiguió que cada botón llevara a la pantalla indicada, se procedió a la creación de un script que se encargara de recoger y guardar todos los parámetros que se eligieran en los distintos apartados del menú.

También se añadieron métodos que hacían posible que ciertos apartados de las pantallas se mostraran u ocultaran en base a las decisiones del terapeuta, como, por ejemplo, quitar el seleccionable de si se quiere o no objetos encima de los muebles si antes se ha elegido que dicho mobiliario no aparecería en el entorno virtual.

Estas últimas funciones que se añadieron también resultaron ser las más problemáticas, sobretodo la que se implementó para que el terapeuta pudiera ver una imagen de qué quería decir cada distribución de mesas. Tras unas horas de bloqueo, finalmente se logró a la solución de que fueran las propias imágenes, y no el seleccionable de distribución, las que controlaran su propia aparición y actualización.

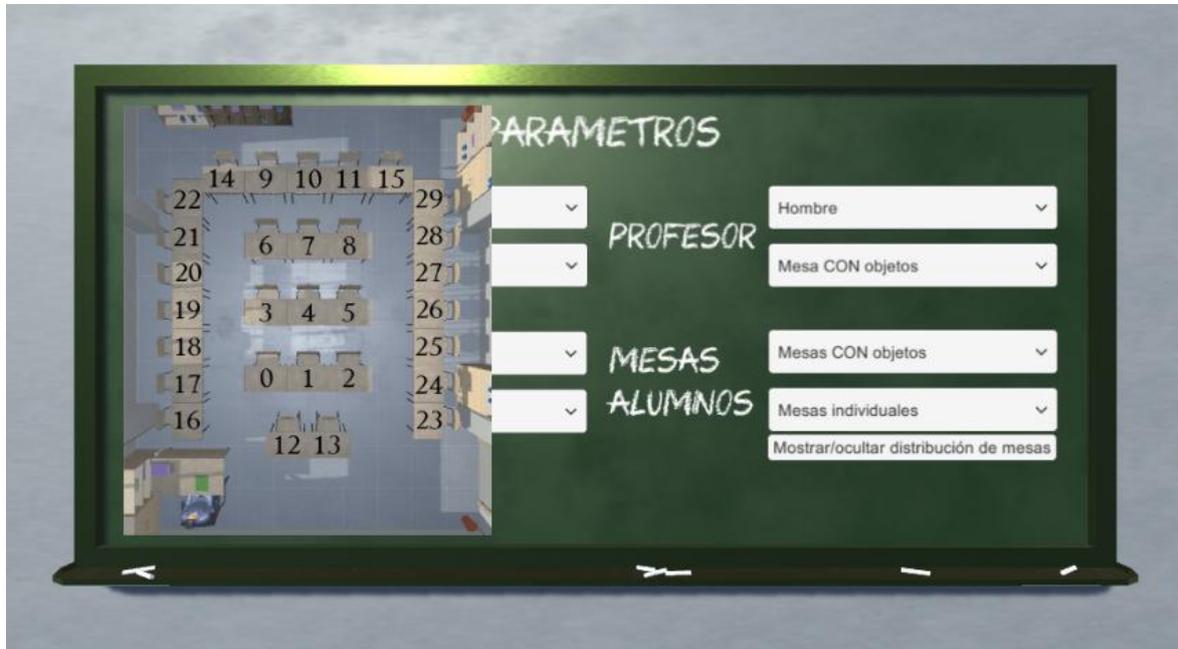


Ilustración 32. Pantalla principal del menú de la aplicación con información sobre la distribución de mesas.

Fuente: propia

4.3.3.4. Sistema de carga/guardado

El sistema de carga y guardado que se planteó para el entorno se basaba, principalmente, en la utilización de dos ficheros bien diferenciados: guardado, que se encargaría de almacenar toda la información de las sesiones que el terapeuta quisiera conservar, y carga, el cual sería leído siempre para obtener los parámetros a plasmar en el entorno virtual inmersivo generado.

Por lo tanto, se dividió esta tarea en cuatro apartados funcionales bien diferenciados, con el objetivo de conseguir el proceso de programación resultara más sencillo y menos abrumador. Estos son, pues, la escritura en el fichero de carga, la escritura en el fichero de guardado, la lectura del fichero de carga y la lectura del fichero de guardado.

4.3.3.4.1. Escritura en el fichero de carga

El primer paso de la escritura del fichero de carga fue decidir cómo se querían guardar los parámetros de la sesión. Tras varios bocetos, se decidió que se almacenarían en cuatro líneas distintas, separando cada parámetro entre sí con una barra baja (_), donde cada una tendría los datos de un apartado distinto dentro de la aplicación.

- 1ª línea: datos sobre la estructura de la clase y su mobiliario.
- 2ª línea: datos sobre los alumnos de la clase.
- 3ª línea: datos sobre el paciente.
- 4ª línea: datos sobre el profesor.

Una vez se tuvo la organización clara, se creó un script que se encargara de almacenar todos los parámetros seleccionables del menú, además de contener el método de escritura del fichero de carga.

Tras unas horas de intensa búsqueda de información y comprensión de las variables y funciones que permiten la creación, y posterior escritura, de un fichero se logró, finalmente, completar la tarea en marcha.

El método final funcionaba de la siguiente manera: primero se comprobaba si ya existía un fichero con el nombre de "Carga.txt" y, en el caso de que existiera, se borraba; a continuación, pues, simplemente había que escribir todas las líneas descritas anteriormente.

4.3.3.4.2. Escritura en el fichero de guardado

Dado que ya se conocía la forma de crear y escribir ficheros, esta tarea resultó, en comparación, más sencilla que la anterior. Pero, aunque en esencia se tratara de almacenar los mismos datos, hubo que añadir unas líneas de información vitales para poder identificar y separar las distintas sesiones guardadas por el terapeuta. Estas se añadieron al principio de las descritas en el apartado anterior y son:

- Línea 1: El separador de sesiones (-).
- Línea 2: El nombre de la sesión.

Para escribir el fichero de guardado, el primer paso es comprobar si ya está creado, pues, al contrario que con la escritura del de carga, en caso de que no exista, se crea. después, el funcionamiento es el mismo, se van añadiendo una a una todas las líneas con la información correspondiente.

Por último, cabe destacar que se diseñó el sistema de modo que, al darle al botón de "Guardar", el programa automáticamente realizara la escritura tanto del fichero de guardado como del de carga.

4.3.3.4.3. Lectura del fichero de carga

Mientras que los dos apartados anteriores se realizan en el lado del menú, este caso concreto se lleva a cabo ya dentro de la aplicación, siendo lo primero que se realiza en la creación del entorno.

Del mismo modo que con la escritura de un fichero, el primer paso fue el de investigar y comprender las variables y funciones necesarias para abrir, y posteriormente leer, un fichero ya creado.

Tras las primeras horas dedicadas a esta acción, se procedió a crear un método que se encargara precisamente de eso, de abrir y leer el fichero de carga, el cual, como se ha descrito anteriormente, contiene toda la información de las opciones de personalización que el terapeuta haya seleccionado en el menú.

Dicho método, además, se encarga de comunicarle los parámetros que va leyendo al script encargado de manejar la creación de la clase. Una vez que ha acabado la lectura del fichero de carga, se procede a activar dicha clase para iniciar la ejecución del entorno virtual inmersivo para la detección y tratamiento de problemas de salud mental.

4.3.3.4.4. Lectura del fichero de guardado

La última función que se implementó fue la de lectura del fichero de guardado y es la que realmente genera la verdadera utilidad de todo el sistema de carga y guardado de parámetros. Sin este paso, el terapeuta no sería capaz de acceder a las sesiones almacenadas.

En este caso se diseñó un método que leyera todas las líneas del fichero de guardado y que cada vez que se encontrara con un separador de sesiones (-) leyera la siguiente, que es donde se encontraba el nombre de sesión que el terapeuta le asignó.

Una vez leído el título, el programa crea un botón al que se le asigna el título, el cual muestra como texto del botón, y un número que indica la posición de la sesión dentro del fichero. Gracias a asignarle estas variables a cada botón, el script puede detectar a que sesión se refiere el terapeuta cada vez que pulsa uno.

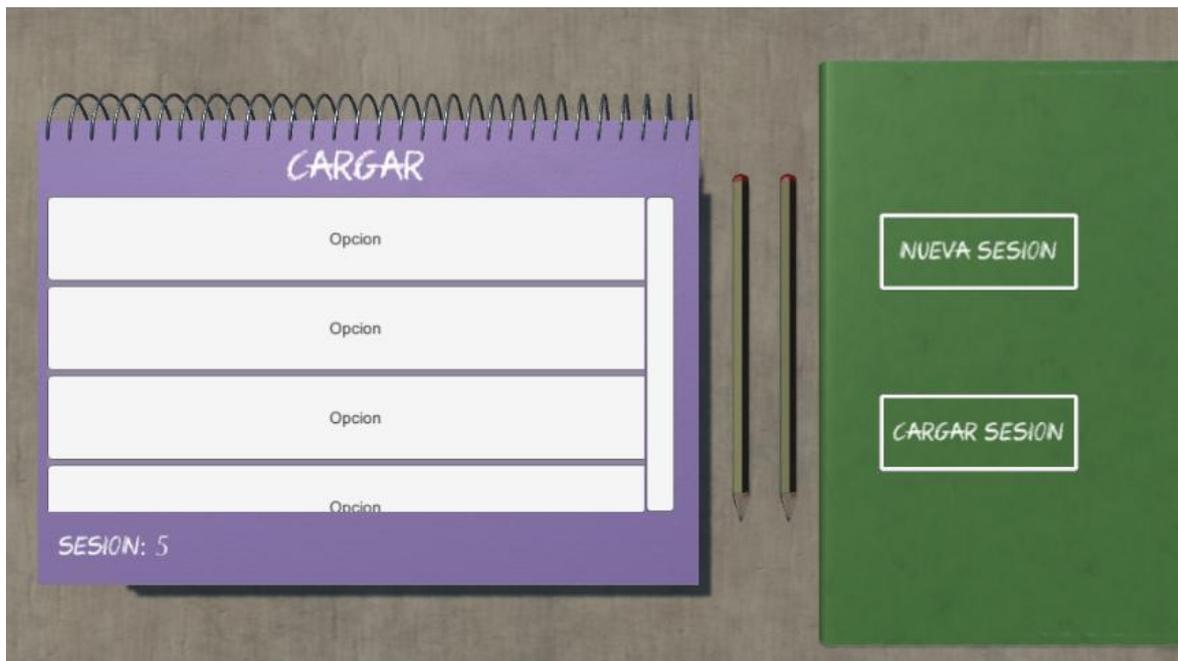


Ilustración 33. Pantalla de carga de la aplicación con botones a modo de ejemplo.

Fuente: propia

Cuando el psicólogo ya ha seleccionado la sesión correspondiente y pulsa el botón de cargar sesión, el propio script de lectura tiene otro método que detecta cuál ha seleccionado y escribe los datos de la misma en el fichero de carga, logrando así que, al iniciarse la aplicación, esta lo haga con los parámetros de la sesión que se ha cargado.

4.3.4. Samsung Gear VR

Una vez que estuvo creado todo el entorno y programado para su correcta visualización y funcionalidad, se preparó el proyecto para que pudiera ser ejecutado con gafas de realidad virtual

El primer paso, fue instalar todos los programas y ficheros de Android que requiere Unity para poder exportar sus programar para que puedan ser ejecutados en un smartphone. Entre estos, destaca el *SDK Manager*, mediante el cual es posible descargar y actualizar el resto de herramientas necesarias.

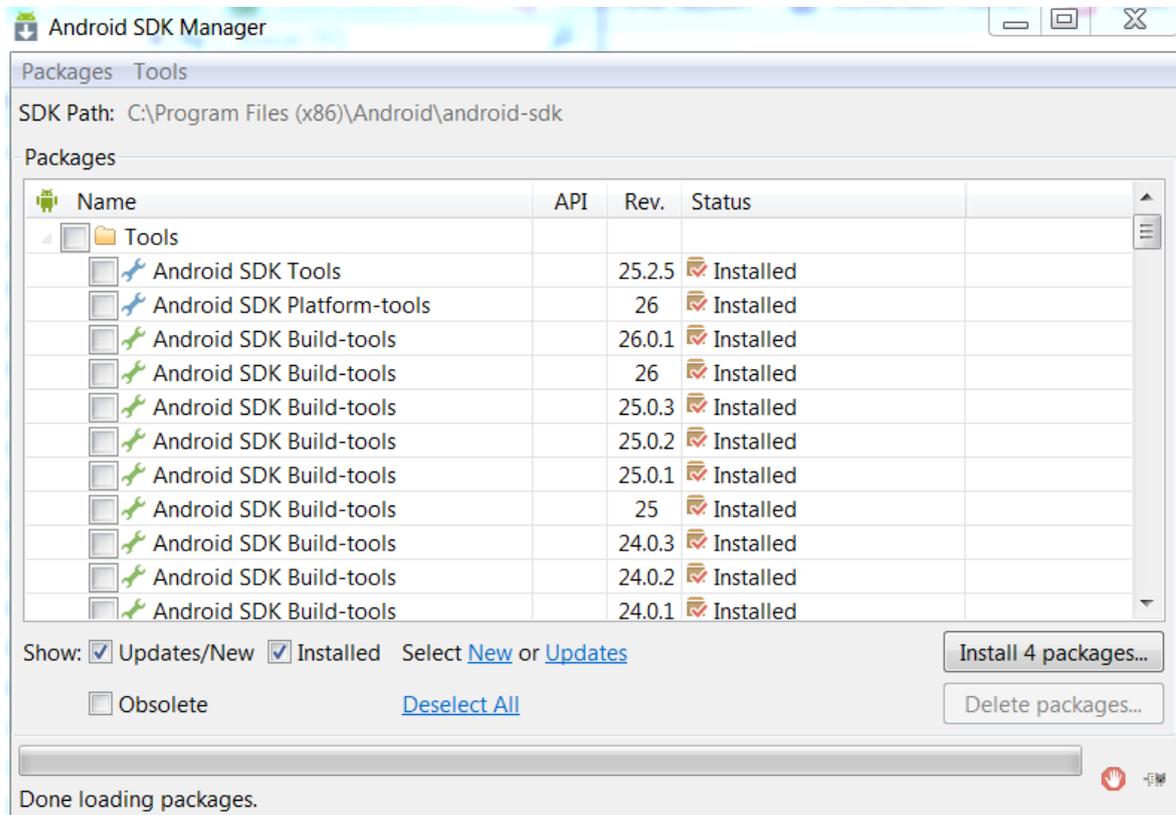


Ilustración 34. Captura de SDK Manager.

Fuente: propia

Ya con todo lo necesario instalado y en funcionamiento, se exportó el proyecto para que pudiera ser ejecutado en un smartphone Android, habilitando también la realidad virtual en el programa.

Al finalizar este paso, se pudieron apreciar una serie de errores que se explicarán y tratarán en el siguiente apartado.

4.4. Testeo y solución de errores

El último gran paso en el desarrollo de este trabajo de fin de grado fue el de probar el funcionamiento de la aplicación y mejorar aquellos aspectos que así lo requerían. Aunque se realizaron arreglos de distinta dificultad, todos ellos se podrían resumir con un único objetivo: mejorar el rendimiento final de la aplicación.

Es importante recalcar esta última idea, ya que fue al momento de ejecutar el entorno en un dispositivo móvil cuando se pudo apreciar que la gestión de los recursos en algunos aspectos de la aplicación había sido nefasta.

Uno de los detalles más importantes de la realidad virtual para posibilitar su uso en tratamientos de problemas de salud mental es que la respuesta del sistema sea lo más próxima posible al tiempo real, pero el entorno que se había desarrollado no cumplía con dicho requisito.

Por ello, el primer paso fue revisar el código que generaba el aula escolar con el fin de encontrar y corregir todos los errores que se encontraran. En esta fase, destaca la inclusión de llamadas "Destroy" y la eliminación de *GameObjects* vacíos que se creaban durante la ejecución de la aplicación.

Se incluyeron "Destroys" en todos aquellos métodos que creaban *GameObjects* y los almacenaban. Este tipo de llamadas permite eliminar por completo un elemento, por lo que se optó por realizarlas nada más inicializarse las variables de almacenamiento, con el objetivo de suprimir cualquier resto que hubiera podido quedarse en memoria de la ejecución anterior de la aplicación.

Así mismo, ciertos métodos requerían de la creación de distintos *GameObjects* vacíos para su posterior tratamiento, por lo que se arreglaron dichas funciones para, una vez finalizado el manejo de la información, se eliminaran los mismos.

Con estos dos cambios breves, se consiguió, en principio, mejorar el rendimiento interno de la aplicación, pero no fue suficiente.

Tras investigar la situación, se comprobó que la mayoría de smartphones soportan en ejecución la visualización y tratamiento de entre cincuenta mil y cien mil vértices, pero el entorno que se había generado contaba, en el peor de los casos, con más de diez millones de estos.

Es por ello que el siguiente paso fue revisar aquellos modelados cuyo grado de precisión fuera elevado para intentar reducir en el mayor grado posible el número de vértices y polígonos en los mismos.

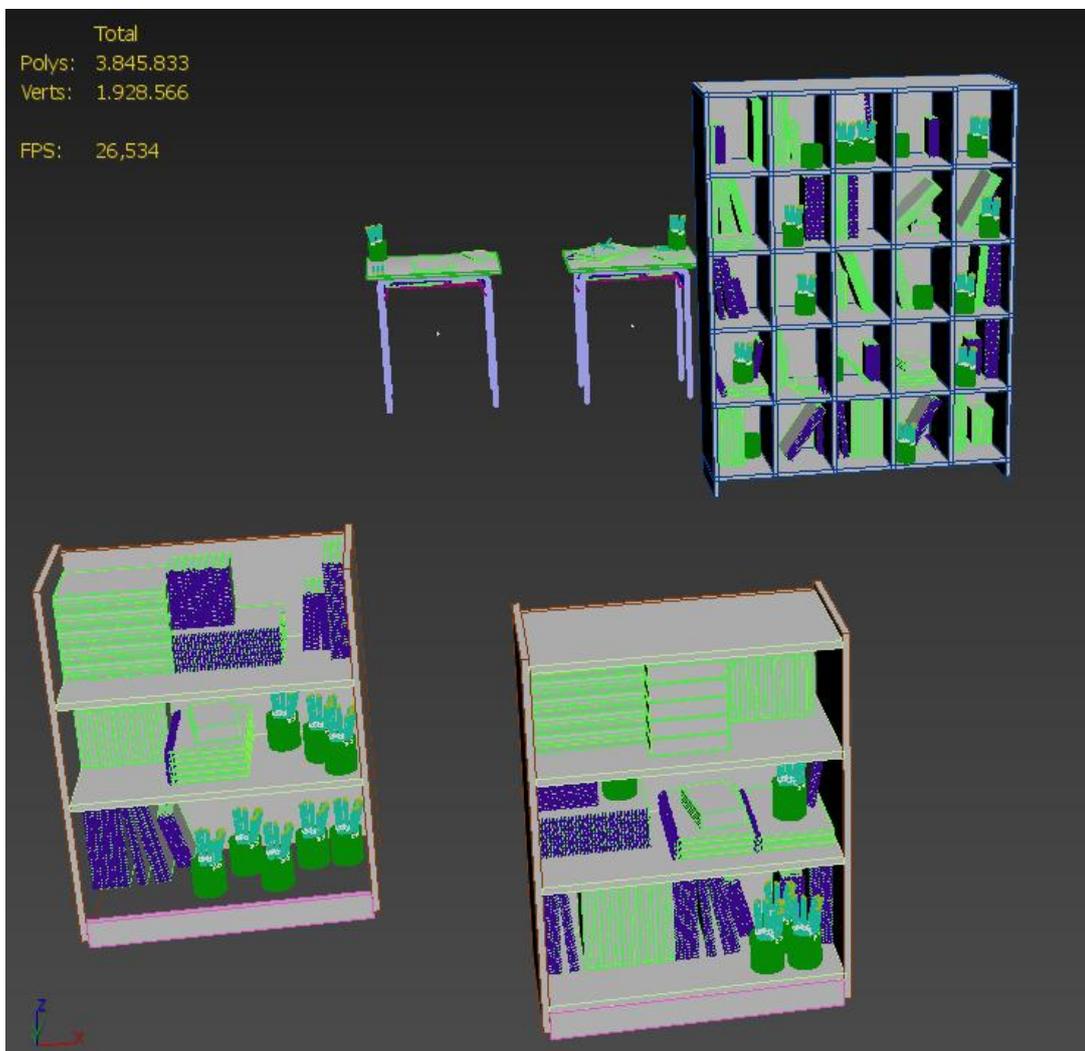


Ilustración 35. Objetos y muebles antes del cambio del número de polígonos.

Fuente: propia

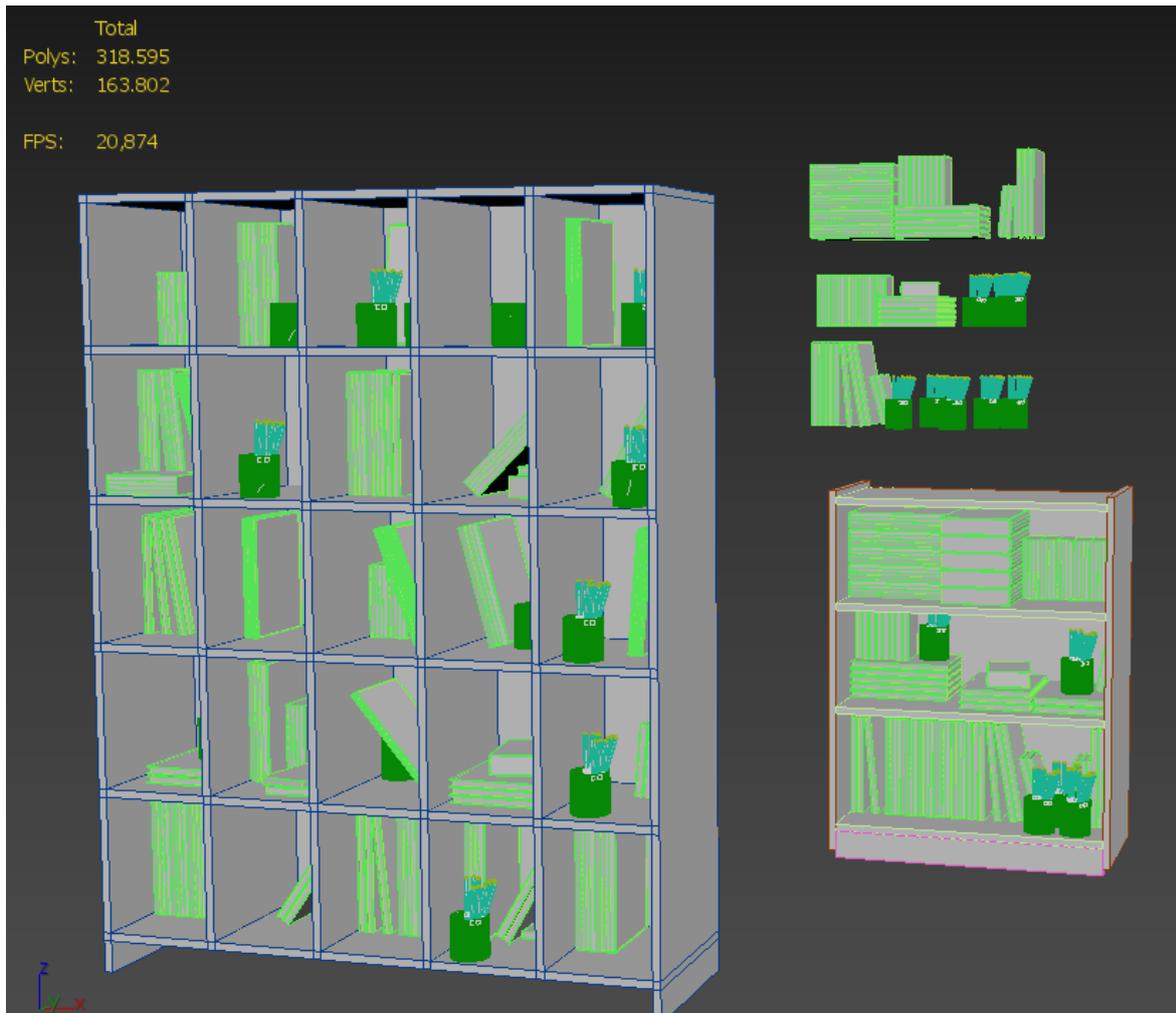


Ilustración 36. Objetos y muebles después del cambio del número de polígonos.

Fuente: propia

Al finalizar este proceso, se consiguió que la aplicación, en sus peores condiciones, contara con una escena con algo menos de siete millones de polígonos. A pesar de que seguía siendo un número excesivo, el rendimiento de la aplicación mejoró en gran medida, logrando ser ejecutada en tiempo real en aquellos entornos favorables.

Por último, se observó que el sistema de carga y guardado había quedado inutilizado durante el proceso de exportado de la aplicación a dispositivos móviles. Para arreglarlo, no hubo más que cambiar las rutas de los distintos ficheros necesarios, por lo que este error se podría decir que fue el más sencillo de solucionar.

5. Cuerpo del trabajo

Una vez explicadas las tres partes principales de la metodología seguida para el desarrollo de este trabajo de fin de grado, llega el momento de exponer y analizar los resultados obtenidos en cada una de esas partes.

Antes de pasar exponer el grueso de este documento, me gustaría destacar que a lo largo de este apartado se explicará el porqué de algunas de las decisiones tomadas en relación al número personajes dentro del entorno y porqué ciertos parámetros se aplican a unos modelados pero a otros no.

5.1. Resultados de la documentación

Al acabar la tarea de documentación, fui consciente de la gran envergadura que este trabajo tenía y que, si se quería terminar con un producto final acabado y funcional, se debía contar con una planificación eficiente. Así que lo que se hizo fue delimitar y aclarar cómo debía ser el entorno y qué opciones debía otorgarle al usuario final.

Junto con mi tutora, se decidió que el entorno virtual inmersivo a desarrollar sería un aula de un colegio de primaria y que, por lo tanto, todo el trabajo debía ir orientado a ese fin.

Por lo tanto, se puede decir que la consecuencia lógica, o resultado, de la tarea de documentación fue el diseño de una clase de colegio lo más fiel posible a la realidad. Además, dado que la trata de fobias se basa en la incorporación progresiva de aquello que provoca el miedo a los pacientes, también se debía plantear un menú en la aplicación que permitiera al terapeuta realizar un tratamiento eficaz y flexible.

5.1.1. Toma de medidas y fotos de referencia

Al tratarse de un aula de colegio, se decidió que la mejor fuente fiable para realizar el entorno era acudir a uno. Por eso, me puse en contacto con una profesora del colegio público en el que estudié de niño y, tras explicarle el objetivo del trabajo de fin de grado, dio permiso de que se acudiera al mismo a medir y fotografiar los objetos de una de sus clases.

Se tomaron todas las medidas y fotografías que se creyeron conveniente de los distintos elementos de la clase. Entre estos, había algunos que poseían formas especialmente complejas, como las mesas de los alumnos o la silla del profesor, y que, por lo tanto, llevaron más tiempo de acotación del esperado.



Ilustración 37. Fotografía real de una clase de primaria.

Fuente: propia

Entre todas las anotaciones que se tomaron, también se incluyeron las medidas generales de la clase, la puerta y las ventanas, con la intención de que el entorno virtual resultante fuera lo más fiel posible a la realidad. Por lo tanto, no es de extrañar, que en todo el proyecto no se hayan contemplado tareas de diseño del entorno, puesto que el diseño del mismo fue copiado del propio mundo real.

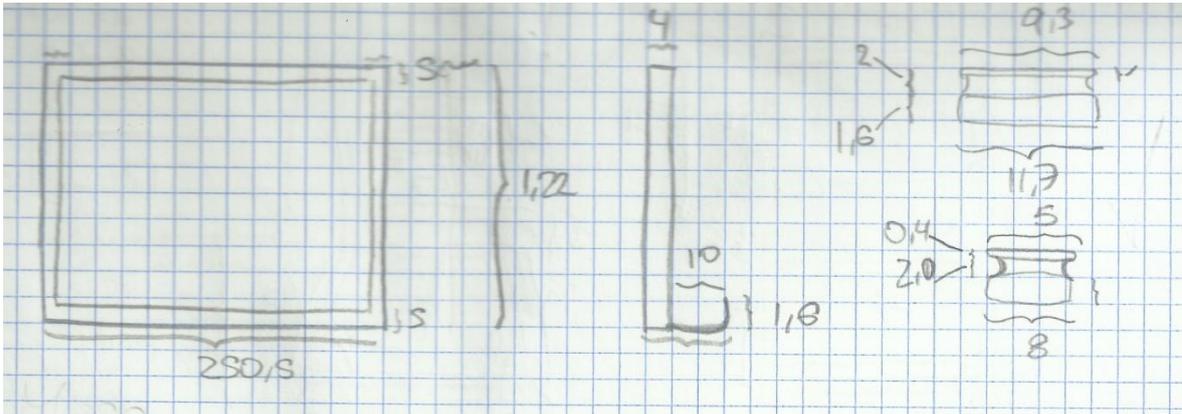


Ilustración 38. Bocetos y medidas de la pizarra y el borrador.

Fuente: propia

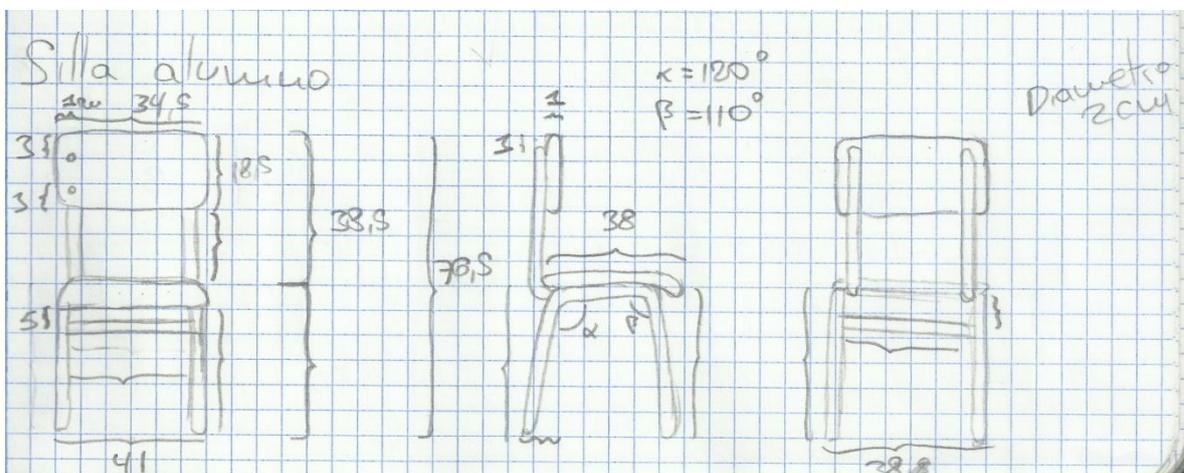


Ilustración 39. Boceto y medidas de una silla del alumnado.

Fuente: propia

Por ello, se investigaron que tratamientos de fobias se podían contemplar con los elementos y el tiempo que se disponía, así como que factores ayudarían a que los pacientes sintieran el entorno como familiar y que, por lo tanto, pudieran identificarlo como su colegio del mundo real.

Antes de pasar a describir todas las funcionalidades que se han implementado y la explicación de porqué se ha decidido que así fueran, quiero aclarar que a pesar de todo el trabajo de documentación sigo siendo un inexperto en el tema del tratamiento de fobias en niños. Es por ello que pido perdón y comprensión en el caso de que algo de lo que se va a explicar carezca del rigor científico necesario o bien no se corresponda con la realidad, ya que algunos de los parámetros que se decidieron son fruto de la deducción.

5.1.2.1. Familiaridad

En este sentido, se decidió que el terapeuta debía tener opciones suficientes como para poder recrear en el entorno virtual inmersivo una clase lo más parecida posible al aula del paciente.

Para lograr este cometido, se diseñaron parámetros que afectaban a tres grandes aspectos dentro del sistema y que condicionaban la familiaridad del ambiente: el profesor/a, el/la paciente y el resto de alumnos.

Antes de pasar a describirlos, aclarar que también se decidió que todos los alumnos, tanto chicos como chicas, vistieran de forma parecida, con el propósito de homogeneizar la clase y que el entorno pudiera ser aplicado también a instituciones escolares donde los alumnos acuden en uniforme.

5.1.2.1.1. Profesorado

Se creyó que tenía especial relevancia el hecho de que el profesor pudiera ser tanto hombre como mujer, pues de este modo el paciente podría relacionarlo con su tutor real. Esta simple función, además, permite que el terapeuta pueda considerarse a él mismo como el profesor.

Con esto último se pretende ofrecer la opción de que el paciente pueda percibir que el psicólogo se encuentra junto con él mismo dentro del entorno en caso de que fuera necesario para algún tratamiento.

También me gustaría aclarar porque en el resto de personajes se puede elegir el color de piel pero en el profesorado no. La principal motivación de esta decisión radica en que en el caso de niños, las posibles diferencias corporales asociadas al color no son apreciables, mientras que en el caso de los adultos sí que saltan más a la vista. Con eso pretendo afirmar que mientras que con los alumnos bastaría con modificar un único modelado, con el profesor se tendrían que hacer tres modelos distintos por sexo, lo cual se consideró algo inasequible dado el tiempo de 300 horas que se le asigna al trabajo de fin de grado.

5.1.2.1.2. Paciente

En el caso del paciente, y como se acaba de adelantar, se decidió no limitar la personalización de su avatar digital al sexo que mostrara, sino que también se daría la opción de elegir el color de piel del mismo. En relación a esto último, aclarar que desde un principio se estableció que el terapeuta solamente pudiera elegir entre tres tonos: claro, moreno y oscuro.

El objetivo principal de estas dos opciones es que si el paciente decidiera mirarse a sí mismo pudiera reconocer una forma corporal y un color de piel parecidos al suyo propio.

5.1.2.1.3. Resto de alumnos

Al igual que con el avatar del paciente, se creyó conveniente que el terapeuta pudiera controlar el sexo del resto de alumnos así como que colores de piel aparecerían en la clase. Mediante estos parámetros el psicólogo puede elegir el porcentaje de niños y niñas, además de si quiere que haya alumnos de las tres tonalidades de tez (clara, morena y oscura), solo de dos o sólo de una.

Al ofrecer estas opciones de personalización, se logra que en el entorno se pueda representar tanto colegios mixtos como aquellos donde se separa por sexos. Del mismo modo, también se contemplan muchos ámbitos de población y, por lo tanto, se puede ajustar la aplicación para que sea más fiel a la multiculturalidad o no del colegio del paciente.

5.1.2.2. Fobias de carácter social

Al investigar sobre los posibles usos de los elementos del entorno, se dedujo que varios de los elementos se podían usar para focalizarlos al tratamiento de fobias de carácter social. Esta clase de trastornos tienden a hacer que quienes las padecen se sientan constantemente observados o excluidos de la sociedad.

Por lo tanto, se decidió implementar parámetros que afectaran a estos aspectos y que pudieran provocar emociones relacionadas con su fobia en el paciente. Estas opciones se podrían dividir en dos: la distribución de las mesas y los elementos diferenciadores.

5.1.2.2.1. Distribución de las mesas

Este parámetro es vital en el caso concreto de las fobias sociales pues se consideró que dependiendo de la distribución de las mesas y de la posición del alumno dentro de

clase podría provocarse en el paciente sensaciones tanto de exclusión como de sentirse observado.

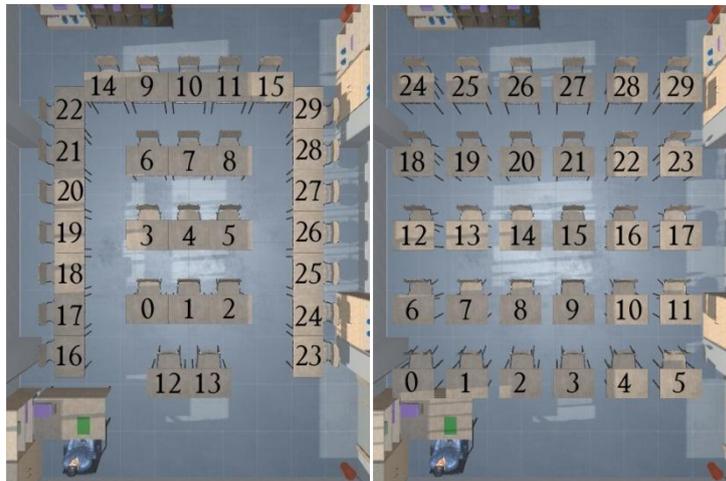


Ilustración 41. Distribuciones en forma de U e individuales de las mesas del entorno.

Fuente: propia

Es por ello que se planificó que el terapeuta pudiera elegir entre cinco distribuciones bien diferenciadas (individuales, en parejas, en tríos, en grupos y en forma de U), además de poder decidir la mesa del paciente de entre las 30 mesas que se disponen en la clase.

Creo conveniente destacar el hecho de que solo con estos dos factores de personalización ya se pueden hacer un total de 150 combinaciones posibles, por lo que las posibilidades que tiene el psicólogo de hacer un tratamiento adaptado son considerablemente altas.

5.1.2.2.2. Elementos diferenciadores

En el caso de que las opciones de distribución no fueran suficientes para crear un sentimiento de observación/exclusión, se decidió aportar a la personalización de la clase más parámetros que permitieran diferenciar de alguna manera al paciente de entre el resto de alumnos.

Los primeros parámetros a comentar son los de familiaridad, los cuales ya se han descrito anteriormente. Con todas las opciones que éstos ofrecen sobre color de piel y el género, tanto del paciente como de alumnos y profesores, se puede configurar el entorno, por ejemplo, para que el paciente sea el único varón en una clase donde todo son mujeres y/o para que su color de piel sea el único del entorno.

También se decidió incorporar parámetros relacionados con los objetos de los alumnos y del profesor, es decir, el terapeuta podría decidir si sobre las mesas de los presentes hay o no material escolar y, en el caso concreto del alumno, el orden del mismo. Por lo tanto, si se creara un entorno donde nadie tiene objetos sobre la mesa excepto el paciente, esto podría evocarle emociones relacionadas con su fobia de carácter social.

Por último, se planificó la opción de que se pudiera elegir el número de alumnos dentro de la clase y que el orden de creación de los mismos debía quedar claro en el menú. Por ello, el terapeuta es capaz de ver sobre las mesas un número que indica dicho orden y, por lo tanto, sumado al hecho de que también puede elegir la posición del paciente, le posibilita colocar al paciente, por ejemplo, en solitario si así lo deseara.

Cabe destacar que si el terapeuta decidiera que, por ejemplo, la clase solamente contará con 5 alumnos y además coloca al paciente entre las 5 primeras mesas en llenarse, el paciente estará incluido entre esos 5 alumnos. Si por el contrario lo coloca en cualquier otra mesa, el entorno virtual inmersivo resultante contaría con 6 alumnos (el paciente y los 5 que eligió).

5.1.2.3. Claustrofobia

Dado que en entorno virtual inmersivo elegido iba a consistir en ambiente cerrado, se hizo evidente que se podía aprovechar ese factor para el tratamiento de claustrofobia. Es por ello, que al terapeuta, durante el proceso de personalización de la clase, se le da la opción de decidir si el aula tendrá o no huecos para las ventanas. Además, en el caso de que sí las habilite, podrá escoger como de abiertas las quiere.

Otros parámetros que se idearon con este propósito es el de elegir si en la clase, además de las pizarras y las mesas, aparece o no más mobiliario (casillero, estanterías, papeleras y percheros). Por otra parte, se creyó conveniente que en el caso de que sí que se quisieran esos muebles, el terapeuta pueda escoger si sobre ellos hay material escolar o no, por si se le antojaba necesario para el tratamiento de algún paciente.

5.1.2.4. Cuestiones sobre el género y orden

Por último, me gustaría comentar el hecho de que al adaptar el entorno para tratar las fobias anteriormente, se consiguió, de forma casi involuntaria, que algunos de sus parámetros pudieran ser usados para otros tratamientos o trastornos.

En primer lugar, destacar el hecho de que cuando el programa se refiere al profesorado lo hace como "sexos", mientras que con el alumnado y paciente lo hace como "géneros". Esto es debido a que se consideró apropiado que el entorno virtual inmersivo también pudiera usarse para aclarar el género de un paciente que se considere transgénero. A pesar de que la identidad de género no se tuvo en cuenta durante el proceso de documentación, se dedujo que el hecho de, por ejemplo, colocar a un paciente biológicamente chico pero con género femenino en una clase donde solamente hay alumnas podría arrojar datos muy interesantes.

Por último el hecho de que se pueda decidir sobre todo el mobiliario y sobre si hay material escolar o no sobre el mismo, así como el orden de los objetos sobre la mesa del paciente, se consideró que el entorno también tendría posibilidades para tratar a aquellos pacientes que pudieran tener cierta obsesión por el orden de las cosas.

5.1.2.5. Repetición de sesiones

Por lo que se pudo investigar, las repeticiones suponen una de las grandes ventajas del uso de realidad virtual como tratamiento de problemas de salud mental y, por ello, se decidió que se implementaría un sistema de guardado y cargado de sesiones.

Dicho sistema se basaría en darle al terapeuta la opción de guardar la configuración del entorno si así lo deseara, pudiendo asignarle como nombre la fobia, el nombre del paciente y el número de la sesión, para posteriormente recuperarla con los mismos parámetros.

5.2. Resultados de la creación de los elementos del entorno

Mientras que los resultados de la documentación fueron las medidas exactas de los elementos del entorno y la definición de las funcionalidades con las que cuenta la aplicación, los resultados de esta etapa del desarrollo son los distintos modelos tridimensionales que le dan vida a la clase del colegio.

5.2.1. Modelados tridimensionales

Como ya se ha comentado en el apartado de metodología, en esta tarea se usó el software 3ds Max 2017 para el modelado de todos los elementos del entorno debido a la gran potencia y versatilidad que representa.

Dado que la gran mayoría de los modelados se realizaron de forma similar, en este apartado sólo se indagará más en aquellos que, por su forma de proceder o por su dificultad, destacaron entre el resto. Casualmente, los elementos que cumplen dichos requisitos son la propia estructura de la clase y los que están más relacionados con el alumnado: su mesa y su silla.

5.2.1.1. Mesa del alumno



Ilustración 42. Fotografía de mesa del alumno.

Fuente: http://www.soyvisual.org/sites/default/files/styles/twitter_card/public/images/photos/cole_0021-p.jpg?itok=888B6fKB

Como se puede observar en la imagen, ni el tablero ni las patas de la mesa suponen ningún desafío a la hora de modelarlos, pero, en cambio, el diseño de la rejilla inferior sí que puede llegar a ser una dificultad. Por lo tanto, se explicará de qué forma se procedió en su creación.

Cabe destacar que, dado que todos los elementos que componen el entramado de la rejilla son cilindros en distintos ángulos de doblado, el proceso que se va a explicar a continuación se hará en líneas generales, sin referirse a ninguno de sus componentes en concreto, ya que todos se realizaron de la misma manera.

La creación de la rejilla se basó, pues, en la generación de cilindros a los que se les aplicaba el modificador *Bend*, que permite doblar cualquier modelado sobre cualquiera de los tres ejes de coordenadas en base al ángulo que se le proporciona.

La longitud y diámetro de dichos cilindros fue sacada de las medidas que se tomaron en un inicio de una mesa de colegio real y, por lo tanto, en ese sentido el modelado se simplificó.

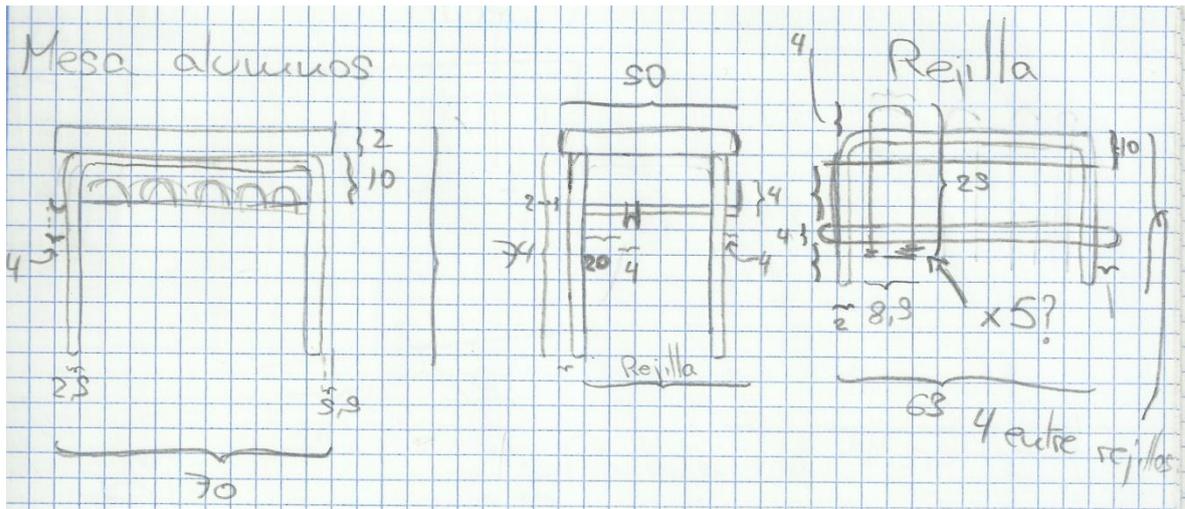


Ilustración 43. Boceto y medidas de una silla del alumnado.

Fuente: propia

En la mayoría de los casos, no era posible generar la varilla a partir de un único cilindro con una única modificación de tipo *Bend*, por lo que para modelar los elementos se hizo uso también de la herramienta de composición de objetos *Proboolean*. Ésta, permite realizar operaciones entre distintos modelados, las cuales incluyen: intersección (genera un modelado a partir del espacio donde coinciden las distintas mallas seleccionadas), substracción (se le borra a la primera malla el espacio donde coincide con el resto de mallas seleccionadas) y unión (genera un modelado juntando las distintas mallas seleccionadas), entre otras. Por lo tanto, la operación que más utilidad demostró en la generación de la rejilla fue la de "unión".

Tras horas generando cilindros, doblándolos y uniéndolos de forma concisa, se logró crear el modelado de la mesa junto con su rejilla.

Antes de pasar al siguiente modelado, me gustaría comentar el hecho de que la creación de la mesa supuso sobre todo un reto a la paciencia, más allá de la dificultad que conllevaba el tener que doblar tantos cilindros en el ángulo correcto.

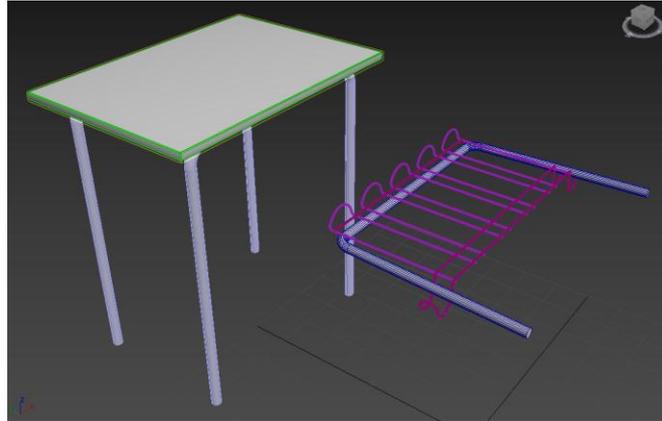


Ilustración 44. Modelado de mesa del alumnado con la rejilla por separado.

Fuente: propia

5.2.1.2. Silla del alumno

El siguiente modelado que destacó por su dificultad fue la silla de los alumnos, más concretamente su asiento. Esto es debido a que la geometría del mismo a penas contempla líneas rectas y que posee un diseño curvado y ergonómico que dificulta su generación por modelado manual.



Ilustración 45. Fotografía de una silla del alumnado.

Fuente: propia

Dado que el resto de la estructura de la silla se modeló de la forma explicada en el apartado anterior, solamente se desarrollará en este el generado del asiento.

En primer lugar, se creó un plano al que se convirtió en polígono editable. Al hacerlo, el programa 3ds Max da completa libertad para modificar su geometría moviendo, editando y añadiendo vértices, segmentos y caras.

Una vez que se logró una forma parecida a la original, se eliminó la mitad del plano y se le aplicó el modificador *Mirror*, el cual duplica el modelado de forma inversa, generando así un efecto espejo. La utilidad de este radica en que puede generar modelados simétricos con solo activarlo e indicarle el eje que debe tomar como eje de simetría.

Ya con el plano preparado, se le aplicó el modificador *Shell* que permitió dotar de tres dimensiones al objeto que hasta entonces había sido bidimensional. Al pasar a tener un altura, se pudieron observar algunos detalles a mejorar de la malla, pero dado que era un polígono editable se pudieron corregir con facilidad.

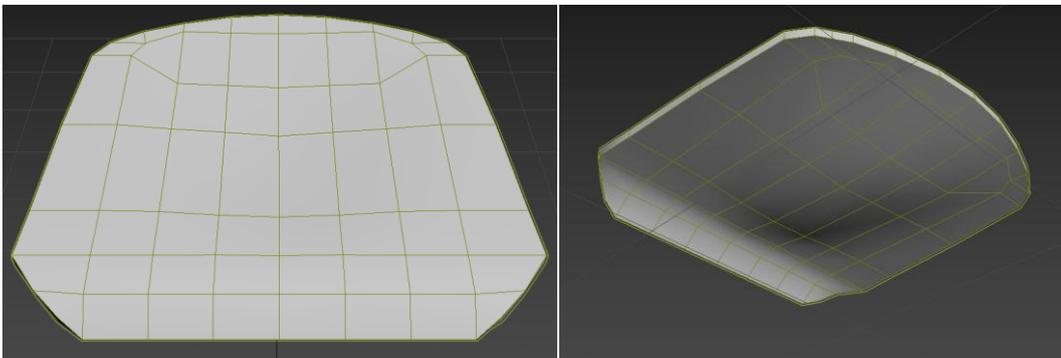


Ilustración 46. Vistas superior e inferior del modelado del asiento de las sillas del alumnado.

Fuente: propia

5.2.1.3. Estructura de la clase

La estructura de la clase en sí no supuso ningún problema al principio, sino que fue una vez terminado cuando mostró una serie de inconvenientes.

Inicialmente se planteó el modelado de la clase como un único objeto tridimensional (a excepción del techo, que el cual se hizo juntando baldosas y lámparas) en el que tanto las paredes como el suelo formaran parte de un todo.

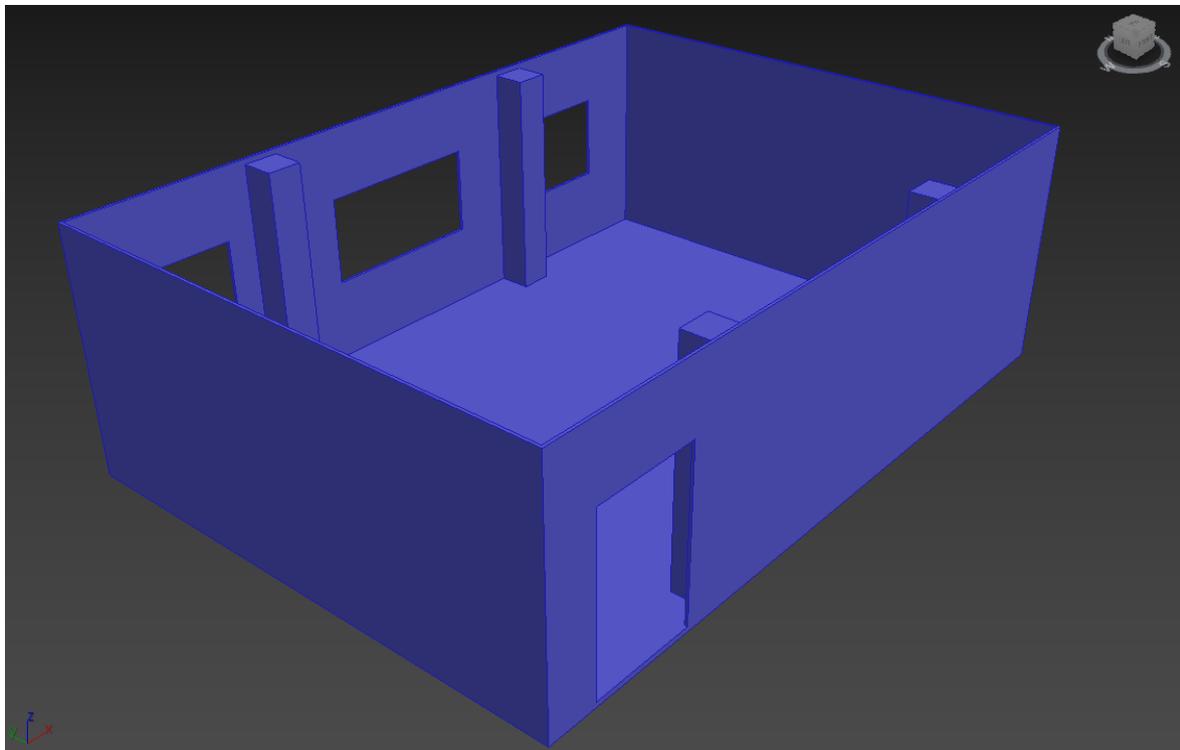


Ilustración 47. Modelado compacto de la clase.

Fuente: propia

Para su creación, se partió de un cubo con las medidas de la clase original (6.30 m de ancho, 8.80 m de largo y 2.8 m de altura) al que se le fueron retirando las partes sobrantes haciendo uso de la operación de substracción del modificador *ProBoolean*.

El resultado fue el esperado pero cuando hubo que desenvolver el modelado para generar el mapa UV, empezaron a surgir problemas. Al parecer al crearlo substrayendo partes, algunas de las aristas que en principio harían falta no se generaban y, por lo tanto, el modelado tenía una geometría que el *UV Unwrapping* no entendía.

Es por ello que se decidió que se volvería a crear el objeto tridimensional pero, en esta ocasión, se separarían las columnas, el suelo y las paredes, con el objetivo de facilitar el desenvuelto.

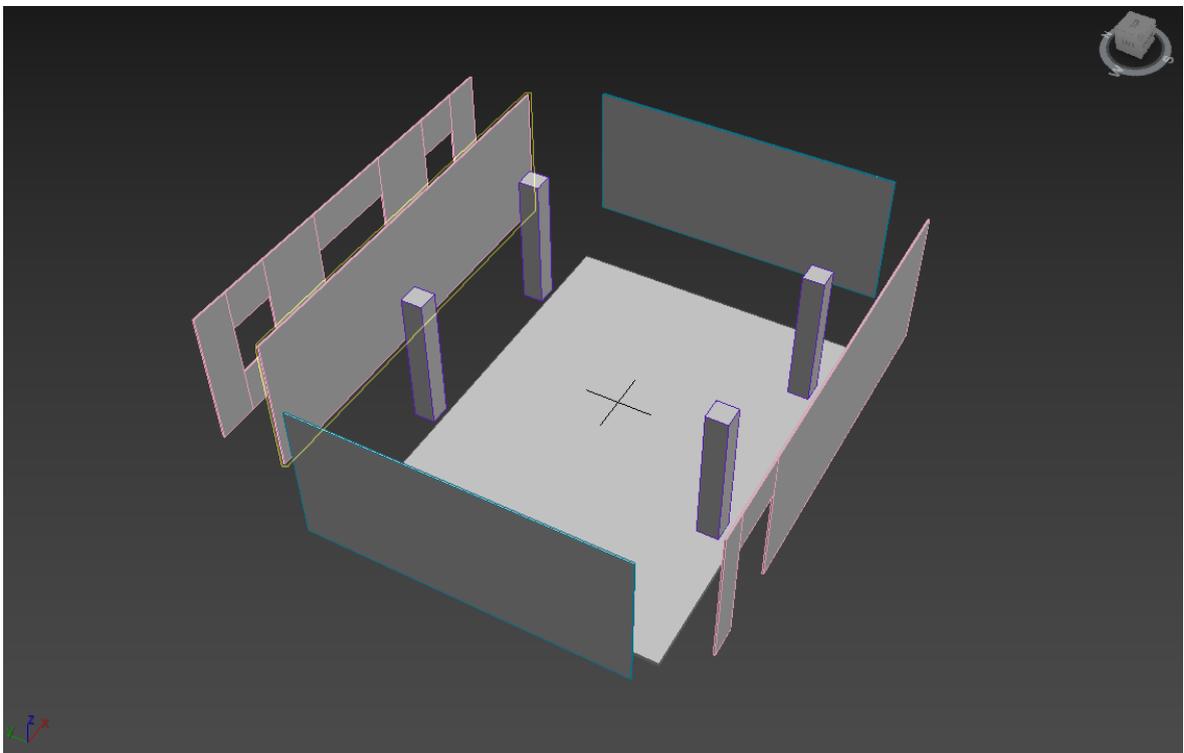


Ilustración 48. Modelado de la clase por piezas.

Fuente: propia

Cabe destacar que las paredes que presentaban aberturas, como las ventanas o la puerta, se crearon igualmente haciendo uso de la operación de substracción del modificador *ProBoolean*.

5.2.1.4. Otros modelados

Para finalizar este sub-apartado, se colocarán algunas capturas del resto de los modelados realizados para el entorno.

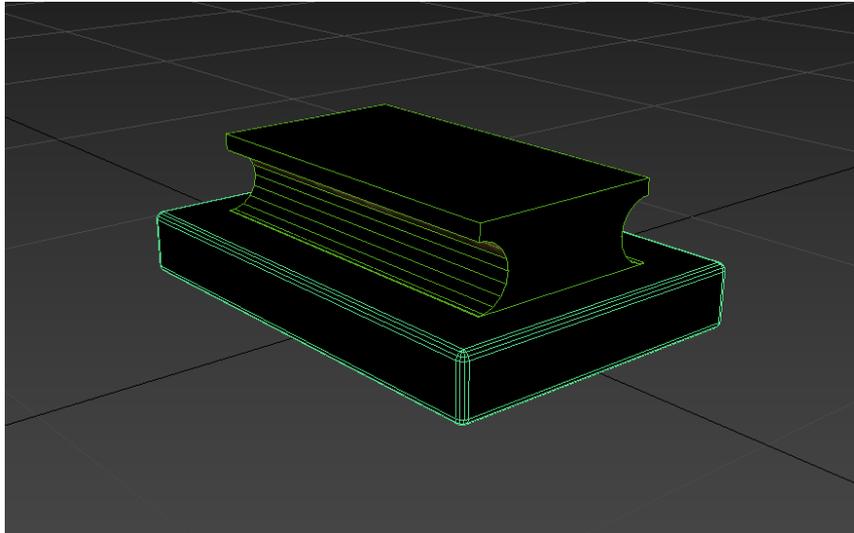


Ilustración 49. Modelado del borrador.

Fuente: propia

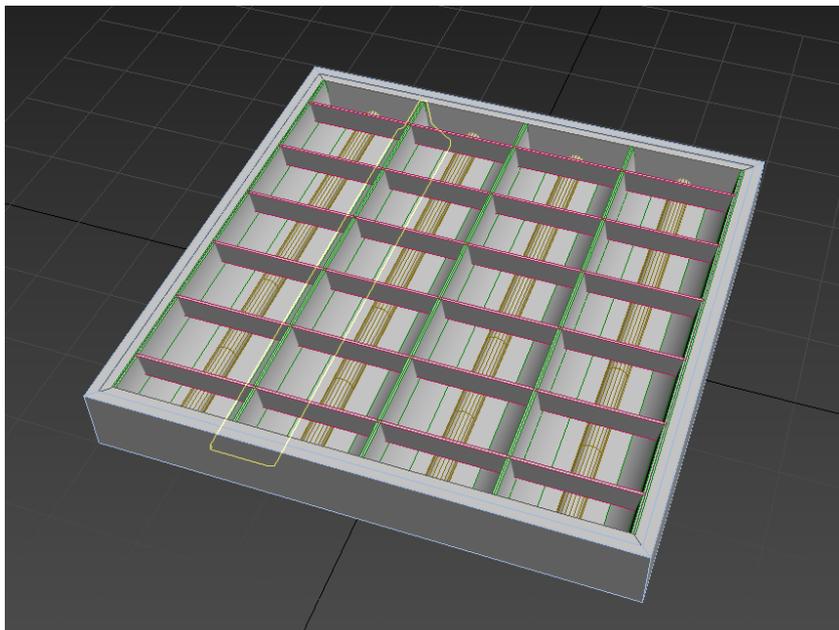


Ilustración 50. Modelado de la lámpara.

Fuente: propia

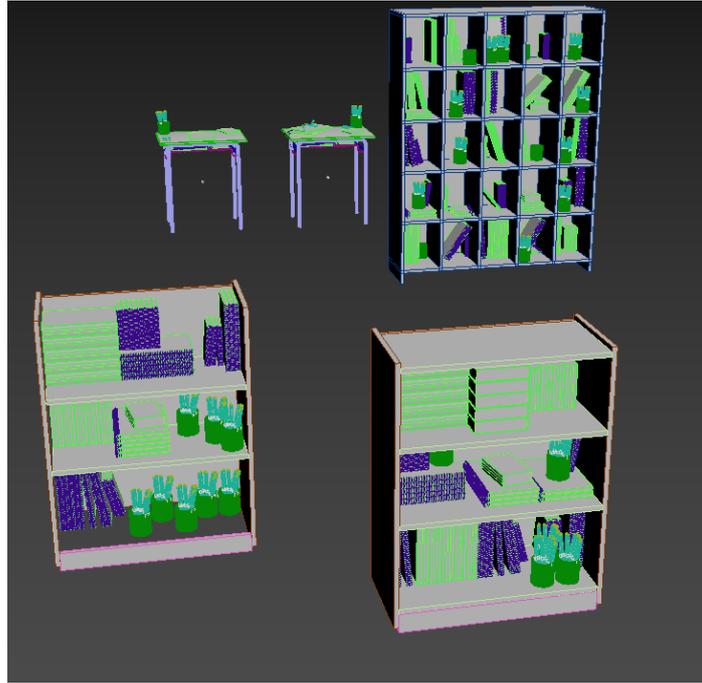


Ilustración 51. Modelado del mobiliario con material escolar dentro.

Fuente: propia

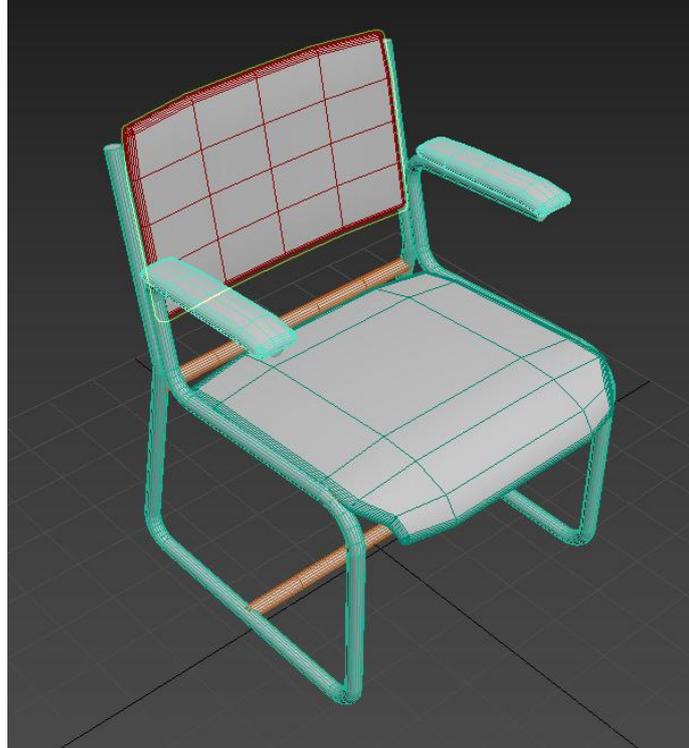


Ilustración 52. Modelado de la silla del profesorado.

Fuente: propia

5.2.2. Mapas UV y texturas

En el caso del mapeado UV y del pintado de texturas no hubieron casos especiales a destacar, por lo que se pasará directamente a presentar algunos de los resultados obtenidos y a comentar algunos aspectos a destacar del proceso.

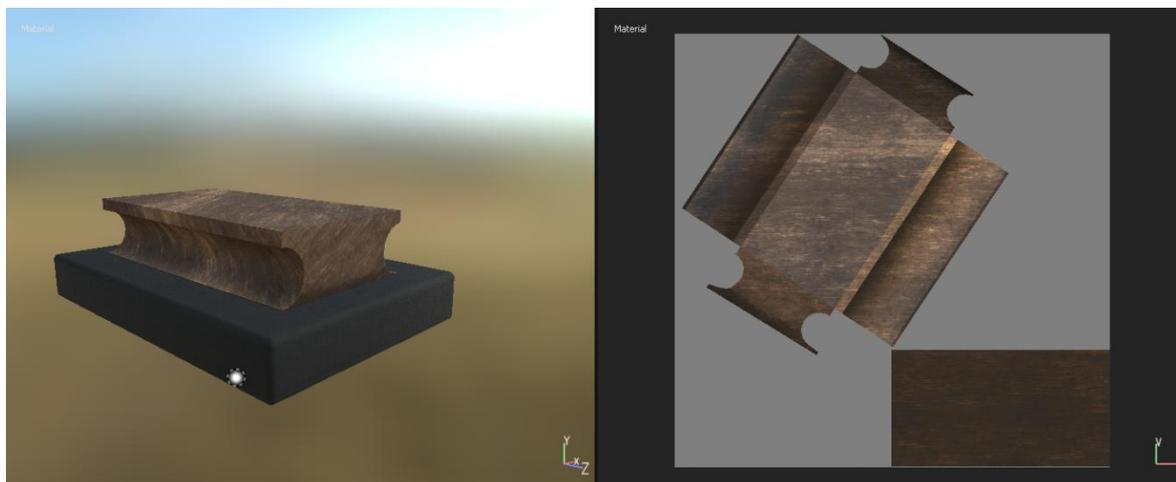
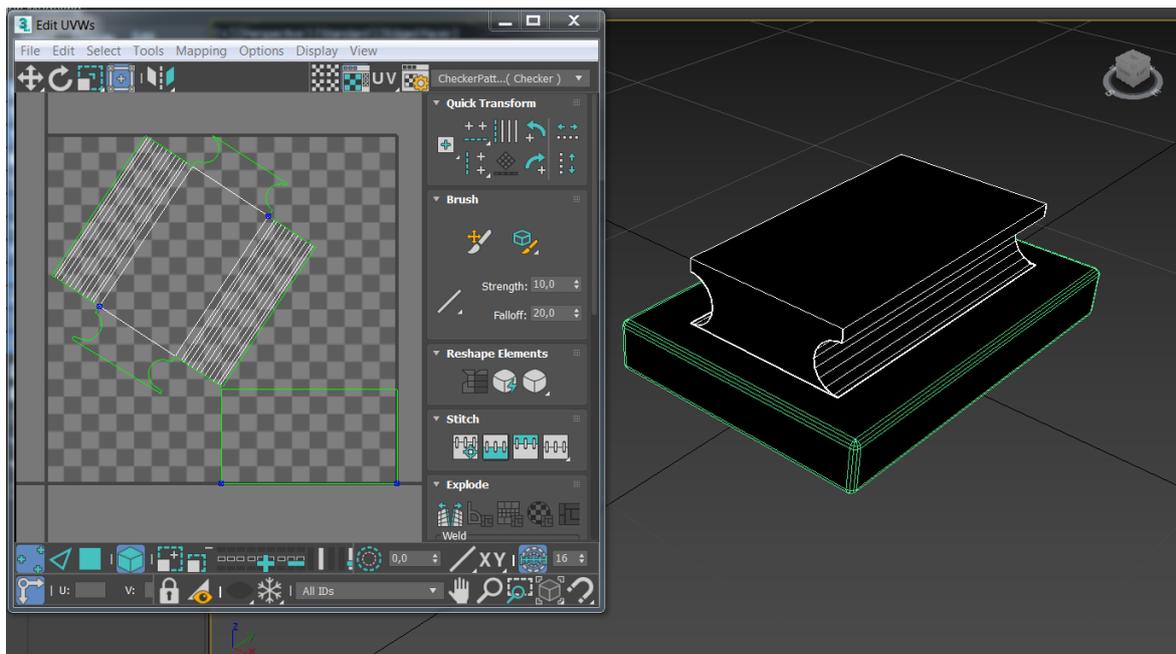


Ilustración 53. Mapeado UV de la parte de madera de un borrador junto con su posterior pintado de texturas.

Fuente: propia

Me gustaría comentar un detalle del proceso de mapeado: las islas, que son cada una de las partes que se quedan por separado dentro del mapa UV (todas las caras superiores juntas y la cara inferior por su parte). Es importante destacar el hecho de que en este trabajo se ha intentado, en la medida de lo posible, generar siempre el menor número de estas, pues cada una de ellas puede generar problemas en la continuidad de la textura resultante.

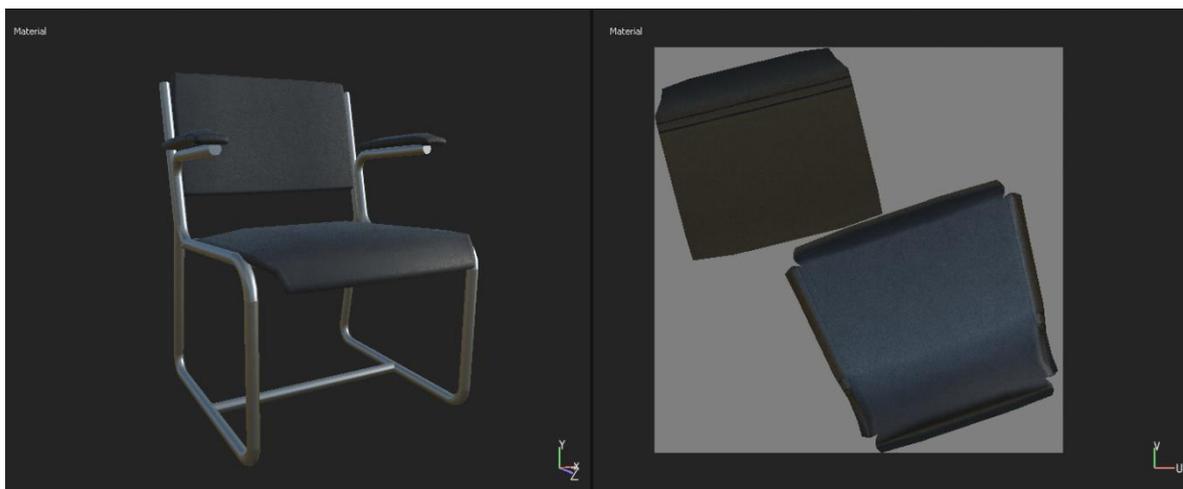
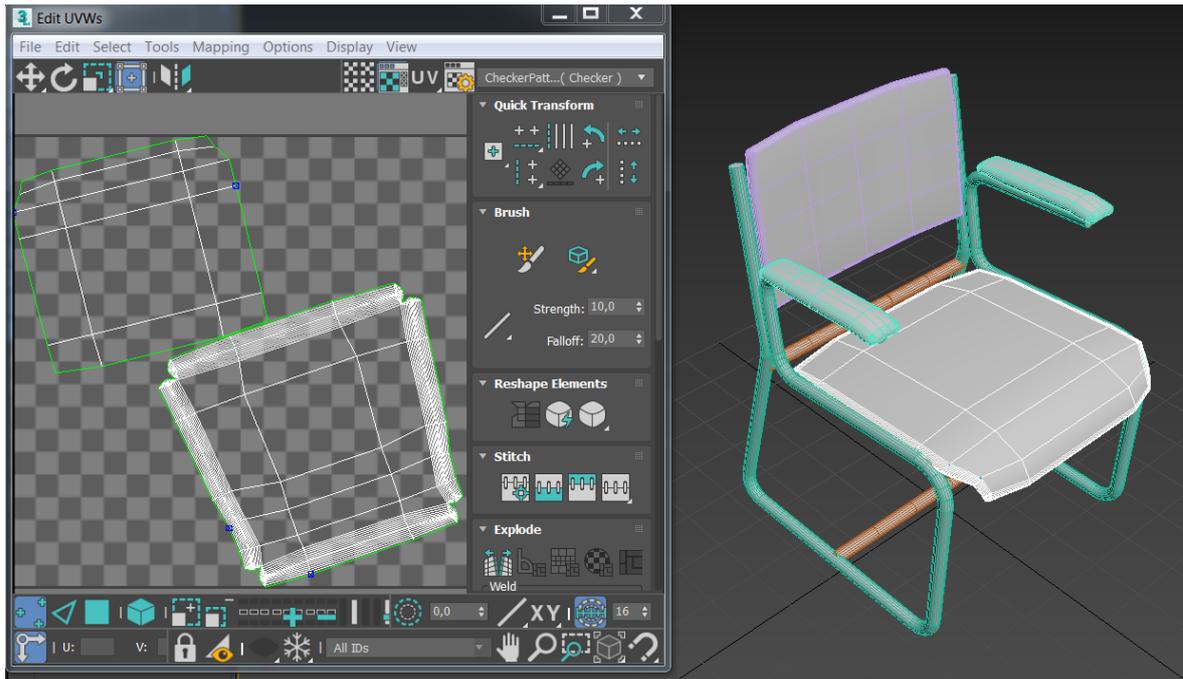


Ilustración 54. Mapeado UV del asiento de la silla del profesorado junto con su posterior pintado de texturas.

Fuente: propia

Por último, he de decir que, a pesar de que en la clase fotografiada la gran mayoría de los elementos de la misma presentan tonalidades verdes, se ha optado por evitarlas a la hora de realizar el texturizado.



Ilustración 55. Fotografía de un aula de colegio real.

Fuente: propia

Esto es debido a que, tras indagar en Internet buscando clase de colegios, muchos de ellos presentaban las maderas y los metales de su mobiliario en su color natural. Por ello, se creyó conveniente adoptar esa filosofía dejando las texturas con formas y colores que recordaran a los naturales.

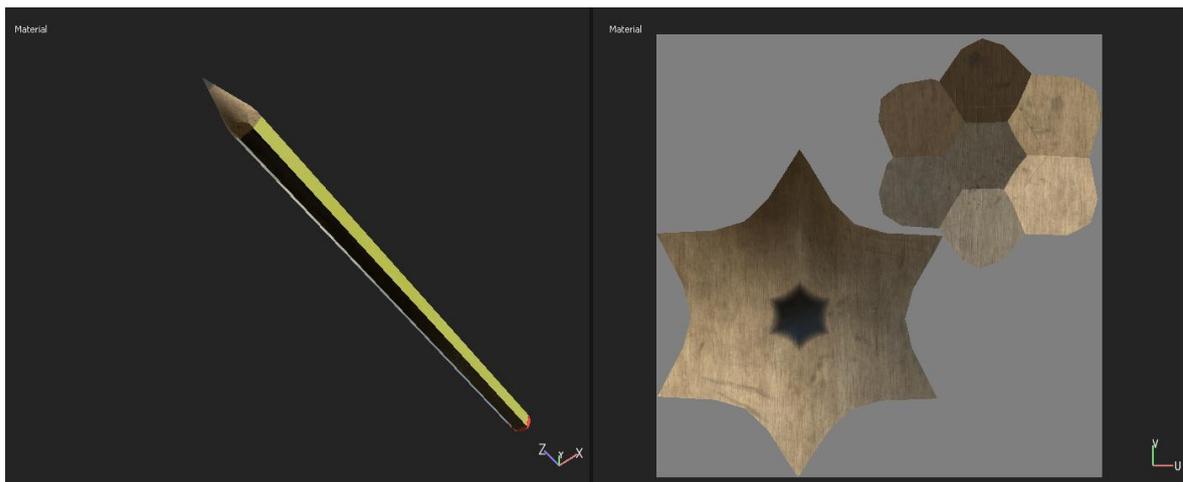
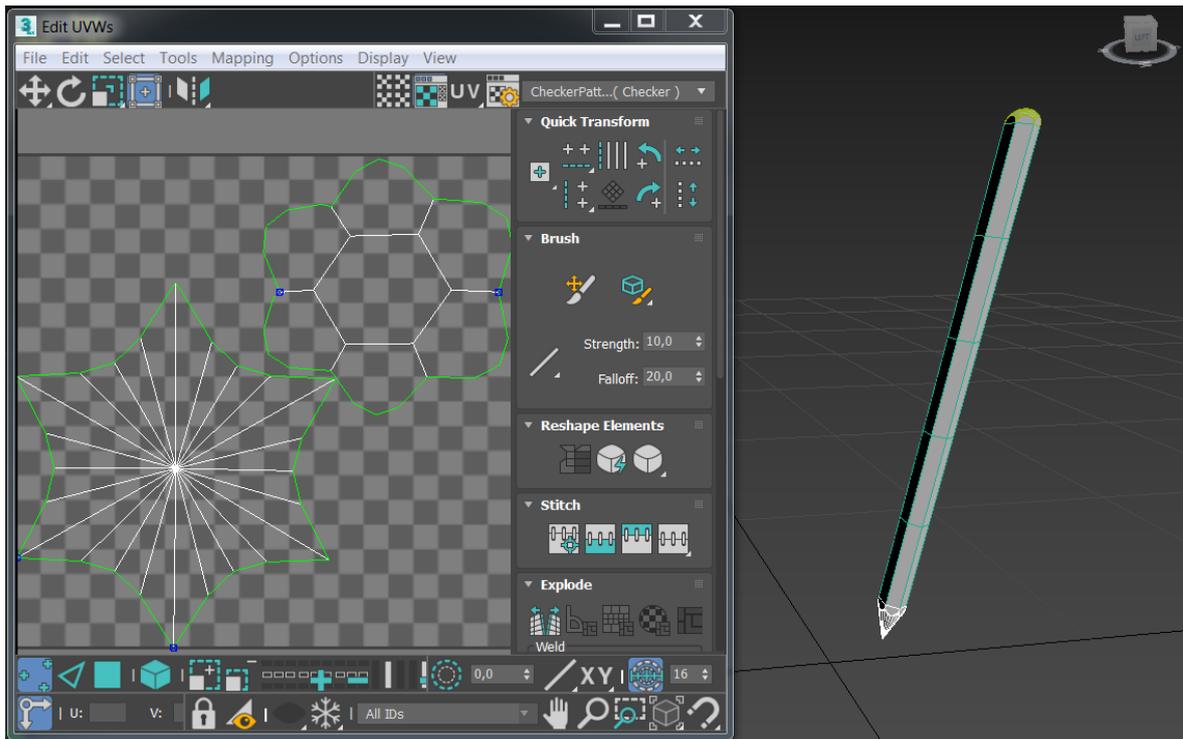


Ilustración 56. Mapeado UV de la punta de un lápiz junto con su posterior pintado de texturas.

Fuente: propia

5.2.3. Personajes

Al igual que en el sub-apartado anterior, se expondrán los distintos modelados de personajes que se realizaron comentando únicamente aquellos aspectos que se consideren de interés o utilidad.

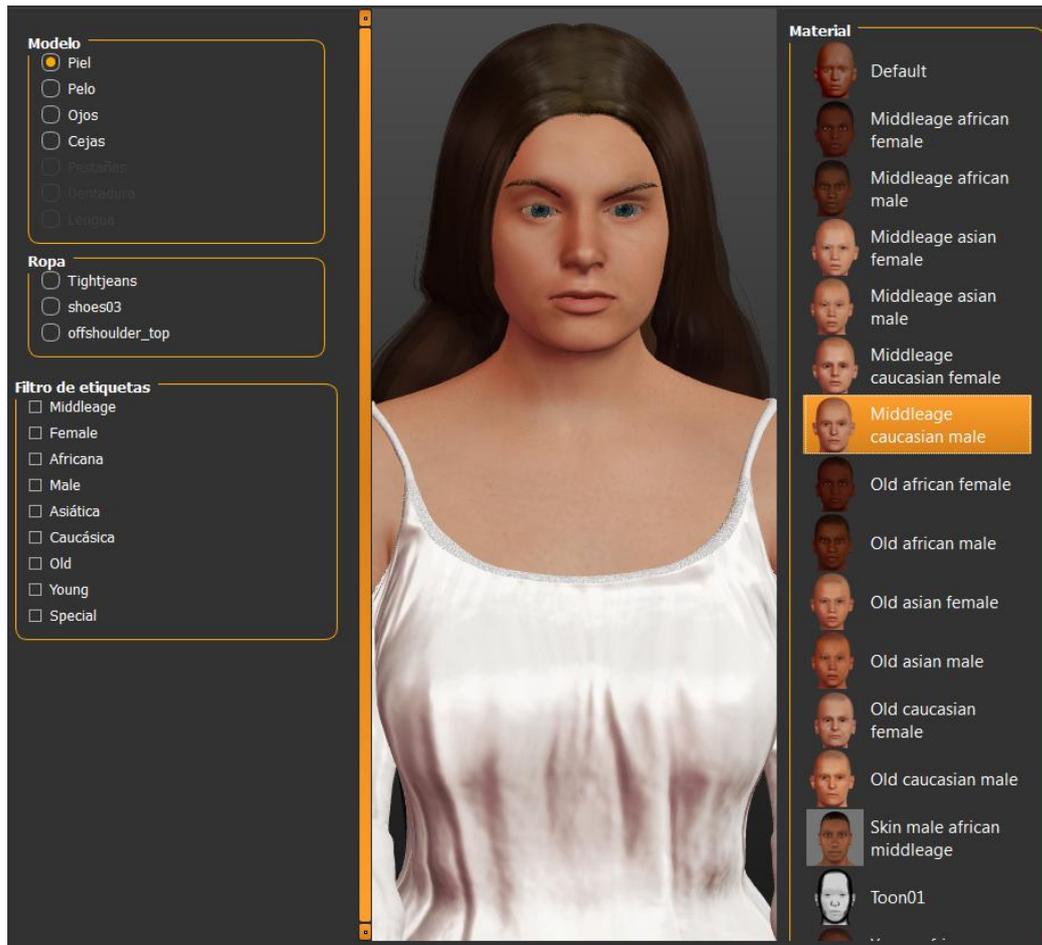


Ilustración 57. Modelado de la profesora junto con las opciones de materiales para simular piel.

Fuente: propia

En MakeHuman, el programa utilizado para el modelado de personajes, además de las variables de edad que se podían regular durante el modelado, existen distintos materiales con los que se puede simular la piel de hombre y mujeres, de varios colores de piel y tres rangos de edad diferenciados (niños, adultos y ancianos). Gracias a ello la tarea del simulado de edad se facilita enormemente.



Ilustración 58. Modelado del profesor junto con las opciones de poses.

Fuente: propia

Así mismo, el software también incluye distintas opciones a elegir de poses para los modelados que se realizan. Este apartado resultó especialmente útil a la hora de comprobar si las dimensiones de los distintos elementos del personaje eran las correctas, pues jugando con las poses se podía observar, por ejemplo, si los brazos eran demasiado largos.

A pesar de su utilidad, como ya se comentó en el [apartado correspondiente de la metodología](#), todos los personajes presentan la misma pose debido a ligeras diferencias de ubicación del modelado que dificultaban después las tareas de scripting.



Ilustración 59. Modelados de los alumnos.

Fuente: propia

Como último detalle a comentar está el hecho de que para el modelado de los alumnos en un primer momento se optó por que ambos fueran exactamente iguales y que el único elemento diferenciable fuera su peinado y ropa, con el objetivo de no sexualizar a los alumnos del entorno.

Aunque en un principio la idea se llevó a cabo, cuando se importaron en Unity y se colocaron en sus respectivos lugares se observó que este hecho le quitaba realismo a la clase, ya que todos los alumnos parecían clones (y lo eran). De modo, que se decidió rehacer el modelo para darle rasgos más masculinos (se remarcó la mandíbula, principalmente) y hacer uno completamente nuevo para las alumnas (este se hizo unos centímetros más bajo, con la mandíbula menos marcada y con algo más de cadera).

Tras realizar estos cambios se exportaron de nuevo los modelos y se reimportaron en el motor de videojuegos, donde se observó que en esta ocasión generaban una mayor impresión de realidad.



Ilustración 60. Captura de todos los modelados de personajes visualizados en Unity.

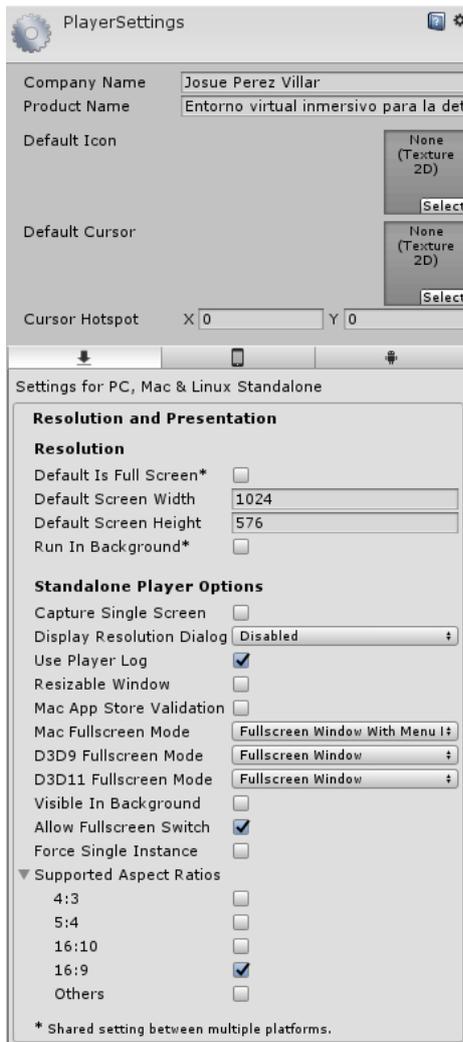
Fuente: propia

5.3. Resultados de la creación del entorno

Finalmente, llegamos a los resultados obtenidos durante la generación mediante Unity del entorno virtual inmersivo del que trata este trabajo de fin de grado: un aula de un colegio escolar que se pueda usar como tratamiento para fobias en niños.

5.3.1. Creación del proyecto y configuración inicial

Una vez llegados a este punto, el primer paso fue crear un proyecto nuevo en Unity. Dado que la aplicación a realizar era un entorno tridimensional para su visionado mediante gafas de realidad virtual, se seleccionó el modo 3D en la creación del proyecto.



Nada más crearse, se acudió a la ventana de *Build Settings*, a través de la cual se puede acceder a los ajustes del jugador, donde se configura todo lo relacionado con la resolución que tendrá la aplicación en ejecución.

Como se puede observar en la imagen lateral, se estableció por defecto la altura y el ancho de una pantalla de ordenador estándar con un aspecto de 16:9, el cual también poseen numerosos smartphones.

Así mismo, para asegurar ese ratio de pantalla, se limitó el programa para que solamente aceptara el aspecto 16:9 y se desactivó la opción que permitía que el usuario eligiera el tamaño de pantalla.

Ilustración 61. Player settings del proyecto.

Fuente: propia

Una vez estuvo configurada la resolución de la aplicación, se modificó el *aspect ratio* que muestra la pantalla de ejecución de Unity, con el fin de ajustarla lo máximo posible al aspecto final del entorno virtual inmersivo.

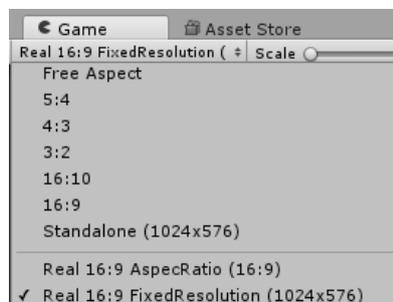


Ilustración 62. Aspect ratio de la pestaña Game de Unity.

Fuente: propia

Por último, con el fin de adelantar posteriores tareas de organización, se crearon y ordenaron las distintas carpetas con las que contaría la carpeta *Assets* del proyecto.

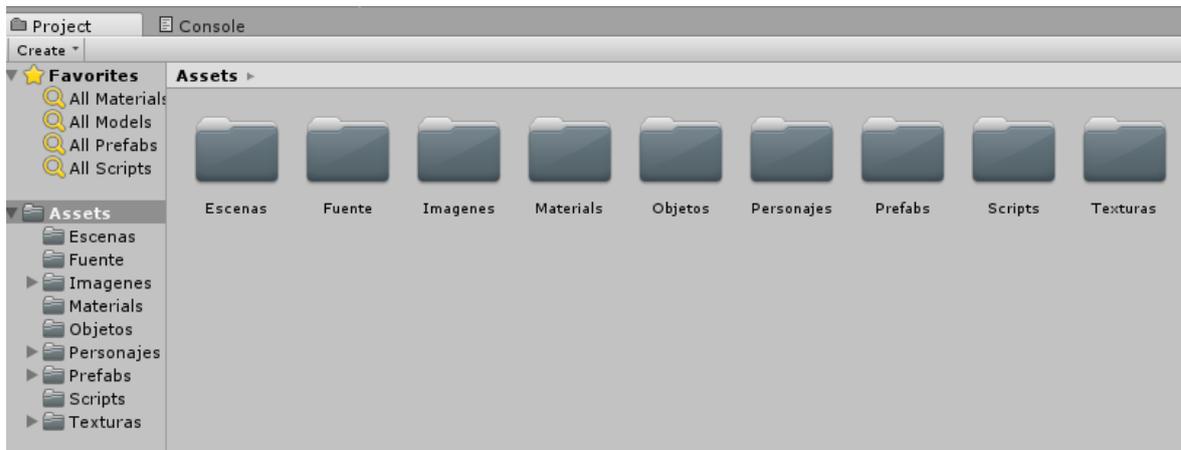


Ilustración 63. Organización de las carpetas del proyecto.

Fuente: propia

Como se puede observar en la imagen, se crearon carpetas para las escenas de las que constaría el proyecto, la fuente de los textos del menú, las distintas imágenes que se usarían en el entorno, los materiales de los objetos, los objetos en sí mismos, las texturas de estos, los modelos de personajes, los posibles *prefabs* que se hicieran y los scripts.

5.3.2. Importación y colocación manual de los modelos tridimensionales

A continuación se importaron todos los elementos a la escena y se le asignó los mapas de textura correspondientes a los distintos materiales. Tras esto, se hizo una primera prueba colocándolos en sus posiciones aproximadas, ya que lo que se pretendía no era crear el entorno directamente, sino comprobar que las dimensiones y texturas de los objetos a ajustaban al aspecto final deseado.

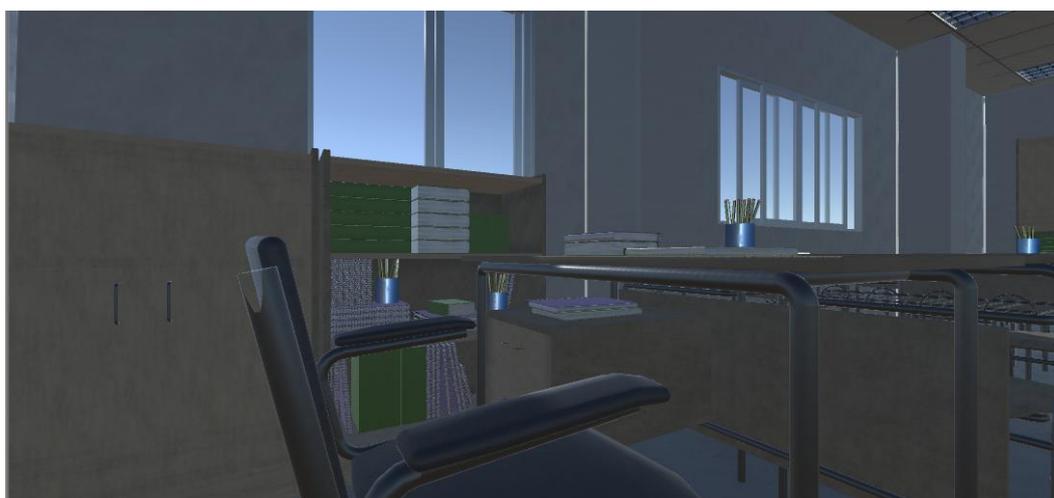


Ilustración 64. Capturas del primer montaje del entorno aún sin la iluminación.

Fuente: propia

Como se puede observar en las imágenes, no hubo que realizar cambio alguno en ninguno de los elementos, por lo que se pudo proceder a su correcta colocación, esta vez con más precisión y esmero.

Posteriormente, se anotaron las [coordenadas de posición y los valores de rotación](#) de todos los elementos. Esta tarea resultó sencilla pero ocupó más tiempo del deseado debido a que se debía apuntar la posición de las mesas y las sillas para cada una de las posibles distribuciones del aula.

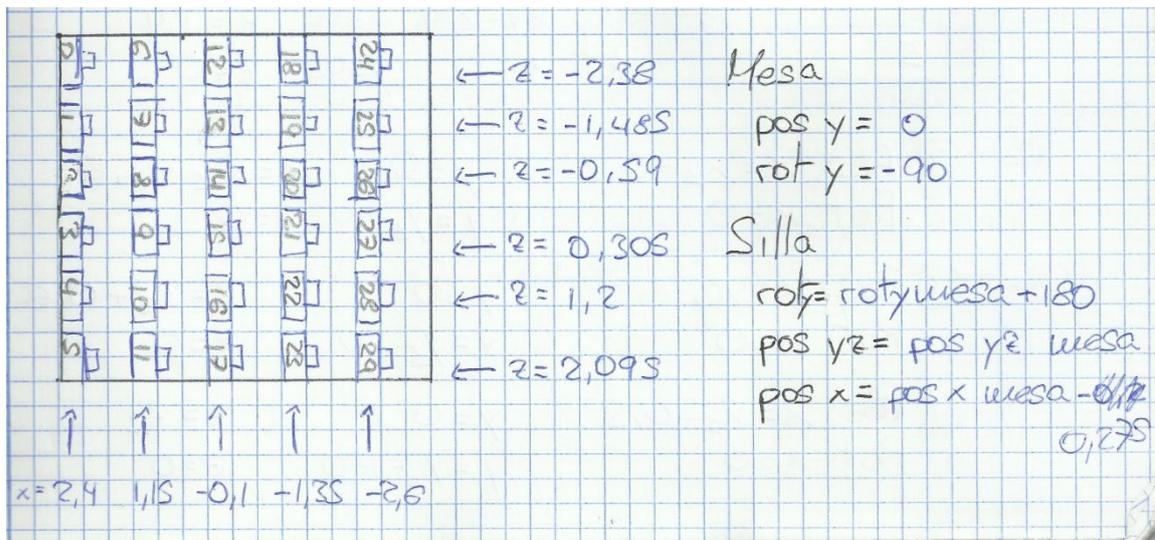


Ilustración 65. Anotaciones de las mesas para distribución individual.

Fuente: propia

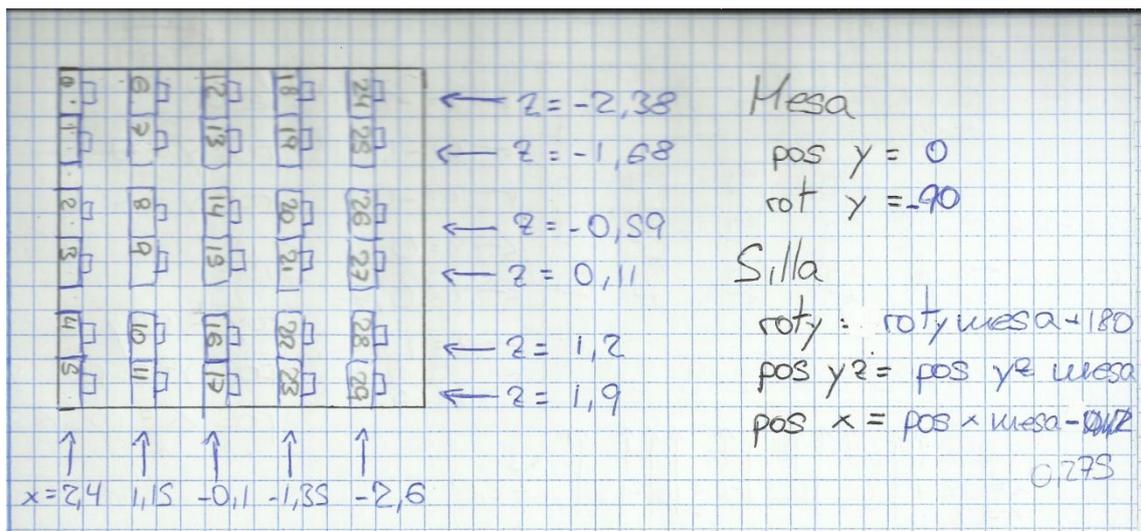


Ilustración 66. Anotaciones de las mesas para distribución en parejas.

Fuente: propia

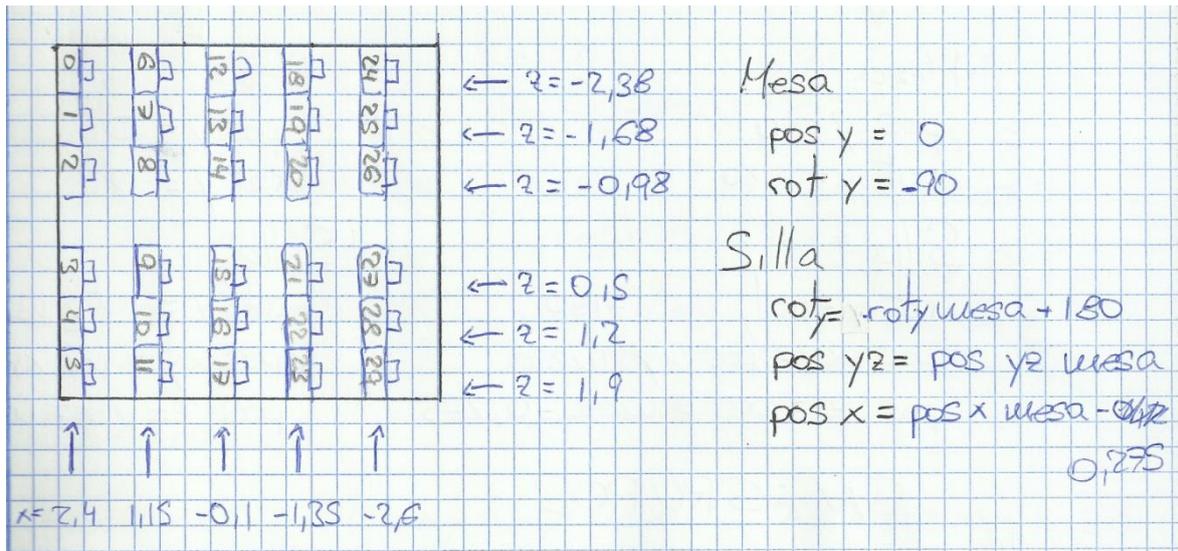


Ilustración 67. Anotaciones de las mesas para distribución en tríos.

Fuente: propia

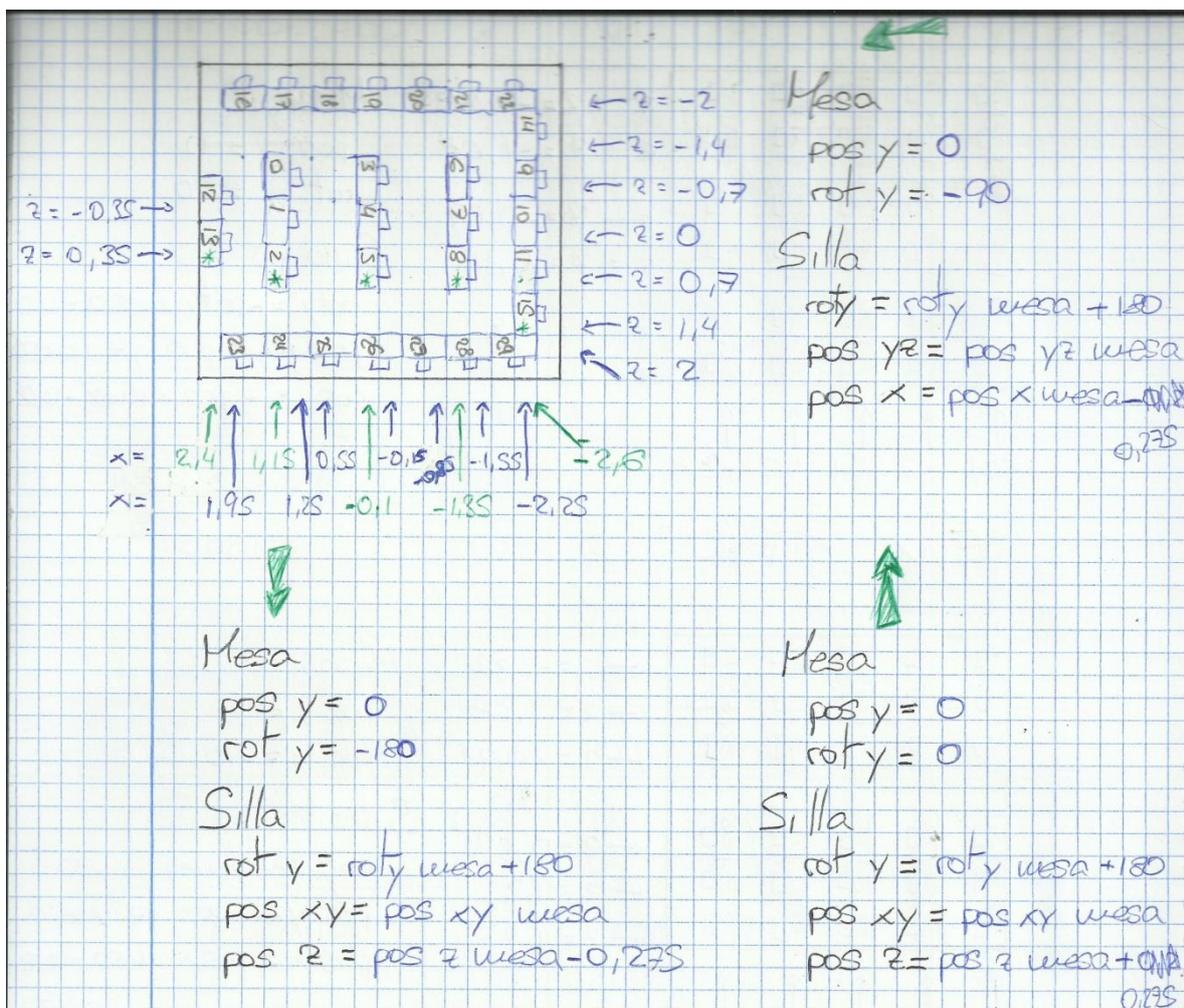


Ilustración 68. Anotaciones de las mesas para distribución en forma de U.

Fuente: propia

A

2	5
4	1
3	6

Pos: $x = 1,9 \quad y = 0 \quad z = -1,4$
 $x = -0,25 \quad y = 0 \quad z = 0$
 $x = -2,4 \quad y = 0 \quad z = 1,4$
 Rot: $x = 0 \quad y = 90 \quad z = 0$

② Pos: $x = x_2 + 0,2 \quad y = 0 \quad z = z_2 - 0,6$
 Rot: $x = 0 \quad y = y_2 + 90 \quad z = 0$

③ Pos: $x = x_2 \quad y = 0 \quad z = z_2 + 0,6$
 Rot: $x = 0 \quad y = y_2 - 90 \quad z = 0$

④ Pos: $x = x_2 - 0,5 \quad y = 0 \quad z = z_2$
 Rot: $x = 0 \quad y = y_2 - 180 \quad z = 0$

⑤ Pos: $x = x_2 - 0,6 \quad y = 0 \quad z = z_2 - 0,6$
 Rot: $x = 0 \quad y = y_2 + 90 \quad z = 0$

⑥ Pos: $x = x_2 \quad y = 0 \quad z = z_2 + 0,6$
 Rot: $x = 0 \quad y = y_2 - 90 \quad z = 0$

B

3	2
6	5

① Pos: $x = 1,4 \quad y = 0 \quad z = 1,5$
 $x = -2,4 \quad y = 0 \quad z = -2$
 Rot: $x = 0 \quad y = 180 \quad z = 0$

② Pos: $x = x_2 - 0,6 \quad y = 0 \quad z = z_2 - 0,1$
 Rot: $x = 0 \quad y = y_2 + 90 \quad z = 0$

③ Pos: $x = x_2 + 0,6 \quad y = 0 \quad z = z_2 - 0,1$
 Rot: $x = 0 \quad y = y_2 - 90 \quad z = 0$

④ Pos: $x = x_2 \quad y = 0 \quad z = z_2 + 0,5$
 Rot: $x = 0 \quad y = y_2 - 180 \quad z = 0$

⑤ Pos: $x = x_2 - 0,6 \quad y = 0 \quad z = z_2 + 0,6$
 Rot: $x = 0 \quad y = y_2 + 90 \quad z = 0$

⑥ Pos: $x = x_2 + 0,6 \quad y = 0 \quad z = z_2 + 0,6$
 Rot: $x = 0 \quad y = y_2 - 90 \quad z = 0$

SILLAS

[A2, A5, B1] Pos: $x = x_{\text{mesa}} \quad z = z_{\text{mesa}} - 0,275$
 Rot: $x = x_{\text{mesa}} \quad y = y_{\text{mesa}} + 180$

[A1, B3, B6] Pos: $x = x_{\text{mesa}} + 0,275 \quad z = z_{\text{mesa}}$
 Rot: $x = x_{\text{mesa}} \quad y = y_{\text{mesa}} + 180$

[A3, A6, B4] Pos: $x = x_{\text{mesa}} \quad z = z_{\text{mesa}} + 0,275$
 Rot: $x = x_{\text{mesa}} \quad y = y_{\text{mesa}} + 180$

[A4, B2, B5] Pos: $x = x_{\text{mesa}} - 0,275 \quad z = z_{\text{mesa}}$
 Rot: $x = x_{\text{mesa}} \quad y = y_{\text{mesa}} + 180$

Ilustración 69. Anotaciones de las mesas para distribución en grupos.

Fuente: propia

Para el caso concreto de la distribución en grupos no se anotaron las posiciones absolutas, como en el resto, sino que se apuntaron las coordenadas de posición de las mesas en relación a una única mesa central en el grupo.

El siguiente paso, fue el de crear una jerarquía útil y eficaz entre los distintos elementos de la escena. Para comenzar se optó por agrupar aquellos objetos de la clase que fueran a estar siempre en una posición fija y que no tendrían cambios: las pizarras, el borrador, el techo, la puerta y la mesa y la silla del profesor.

A continuación se asociaron aquellos que pertenecían a la estructura de la clase, siendo estos los distintos modelos de clase, así como las tres opciones de ventanas preestablecidas. Por otro lado, también se agrupó el mobiliario de la clase y sus objetos, es decir, se juntaron en un conjunto las estanterías, el casillero, los percheros, las papeleras y todo el material escolar que se utiliza para rellenarlos.

Seguidamente, se creó una agrupación separada sólo para las mesas, las sillas y el material escolar de los alumnos, con el fin de facilitar así su gestión en las posteriores tareas de programación y scripting.

Así mismo, se crearon distintos *GameObjects* vacíos con la intención de que estos realizaran la función de contener los distintos scripts de gestión de la aplicación, además de tareas de parentesco para con los distintos elementos del entorno que se crearían, logrando así que estos últimos se agruparan de forma automática bajo un único padre.

Tras todos estos cambios, así quedó la jerarquía de la aplicación:

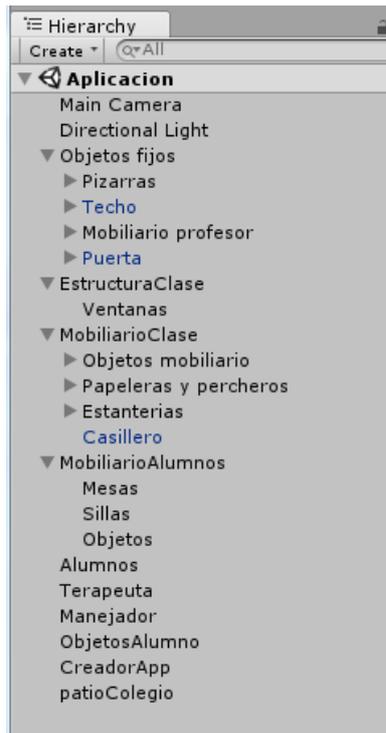


Ilustración 70. Jerarquía existente entre los GameObjects de la aplicación.

Fuente: propia

Para acabar, se crearon las luces que iluminarían el entorno (un *Point light* con sus valores por defecto para cada una de las doce lámparas y una *Directional light* exterior a la clase para simular la luz solar). Así mismo también se colocó una gran imagen que realizaría la función de fondo.



Ilustración 71. Escena en Unity con luces y fondo.

Fuente: propia



Código QR 2. Enlace a la imagen de fondo de la aplicación.

Fuente: http://www.sanestanislaolao.net/files/images//FICHERO385.JPG?KeepThis=true&TB_iframe=true

5.3.3. Interfaz de usuario

Para la interfaz de usuario se optó por reutilizar algunos de los modelados con los que ya se contaba para crear los fondos de las distintas pantallas de los menús, siempre intentando que tuvieran un sentido estético y/o que estuvieran relacionados de alguna manera con la función de la pantalla en la que se estuviera. Teniendo en cuenta estas premisas, la asignación resultante fue la siguiente:

- Puerta: Primera pantalla del menú. Da inicio a la aplicación y, por lo tanto, supone la entrada al entorno.
- Libreta, libro y lápices: Pantalla de cargado. Se eligieron estos modelados debido a que de normal se acude a la agenda para recordar (o cargar) algo que se hizo con anterioridad.

- Pizarra: Pantalla principal. Dado que de normal el profesor escribe lo que desea que realicen los alumnos en ella, se consideró que también se podría usar para que el terapeuta escribiera lo que desea que realice el entorno.
- Pizarra digital: Pantalla de guardado. Por puras razones estéticas.

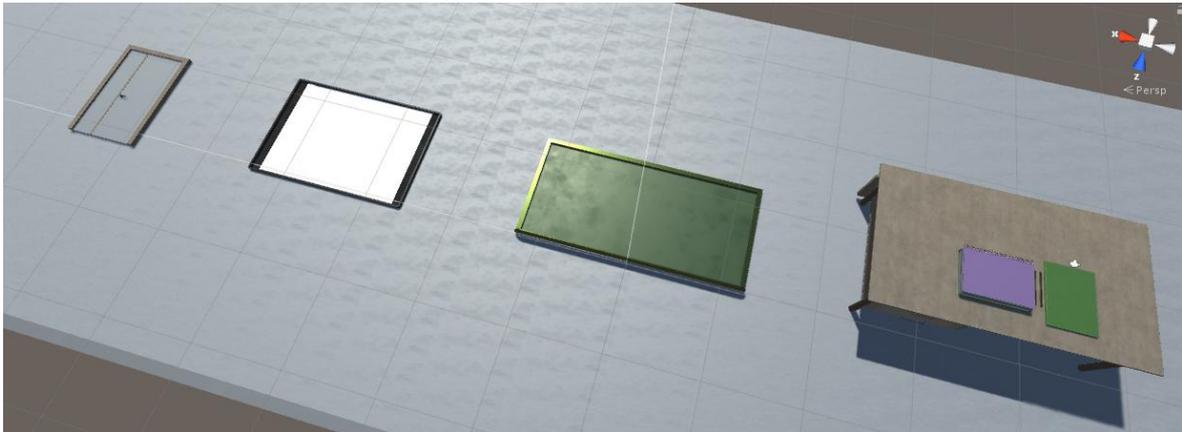


Ilustración 72. Modelados del entorno que se usaron como fondo para el menú.

Fuente: propia

Una vez estuvieron asignados los fondos a cada pantalla, se procedió a diseñar las mismas. El principal objetivo que se tenía para con la organización del menú era que este fuera lo más sencillo e intuitivo posible, lo cual, se considera que se ha logrado.



Ilustración 73. Primera pantalla del menú.

Fuente: propia

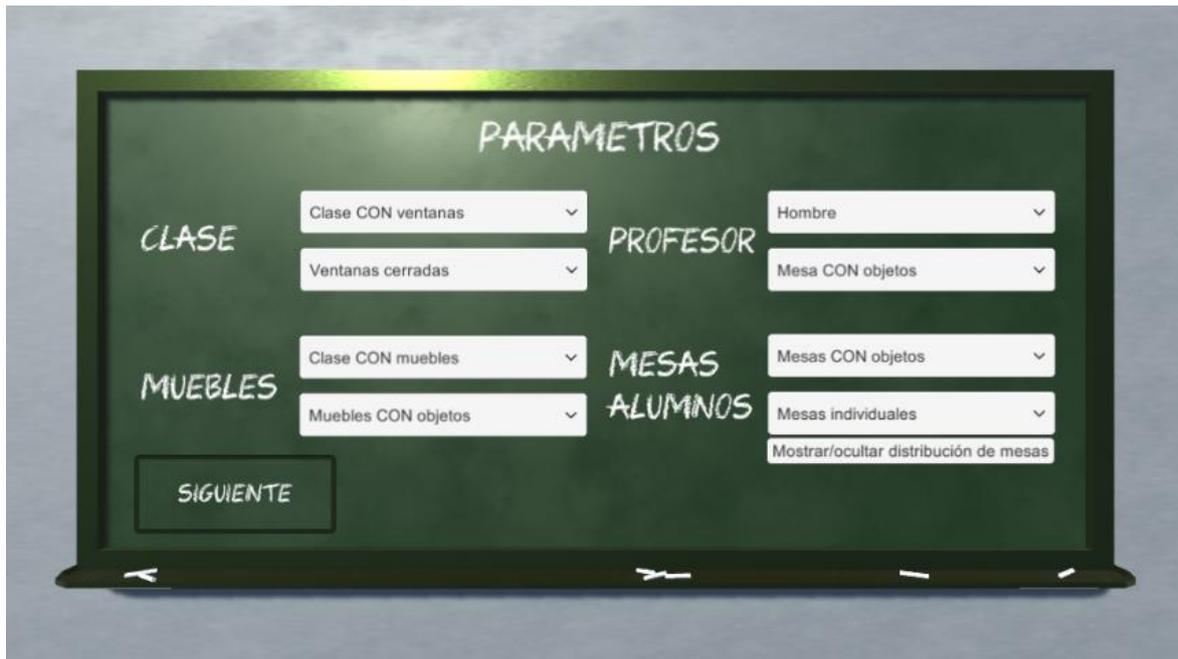


Ilustración 74. Pantalla principal de personalización del menú.

Fuente: propia



Ilustración 75. Pantalla secundaria de personalización del menú.

Fuente: propia

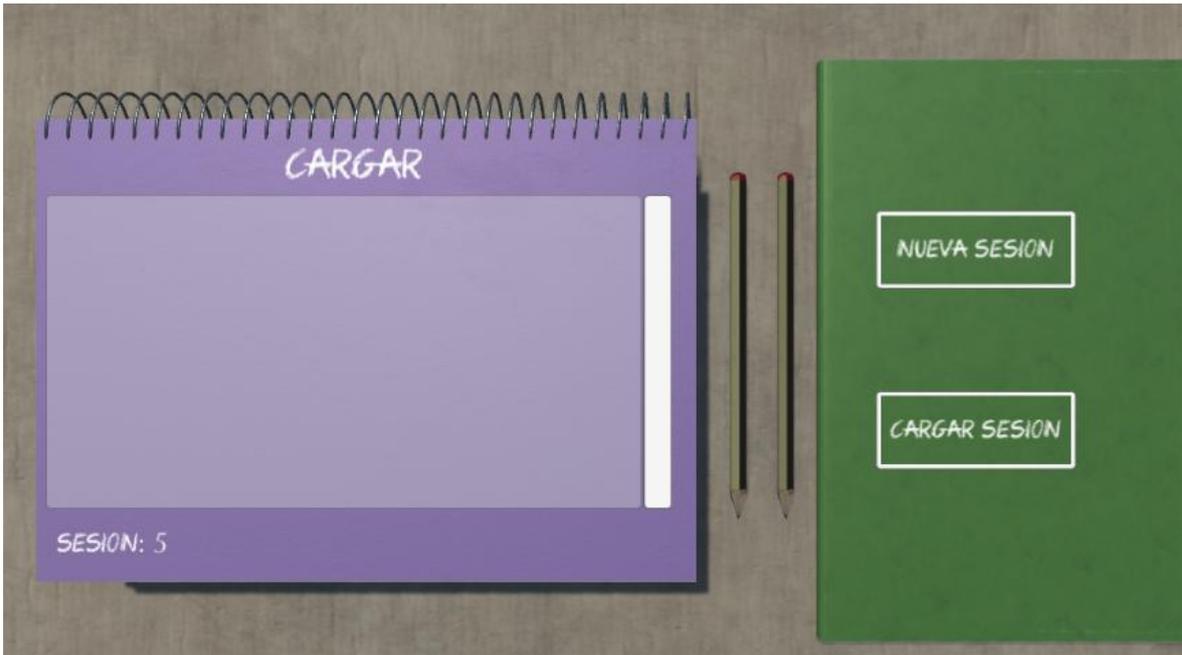


Ilustración 76. Pantalla de carga del menú.

Fuente: propia



Ilustración 77. Pantalla de guardado del menú.

Fuente: propia

5.3.4. Scripts

Para el planteamiento de este apartado se ha elegido que el código se explicará mediante el uso de tablas, aunque igualmente se incluirá un texto descriptivo comentando brevemente el funcionamiento de cada script. Además, debo aclarar que su explicación se ha ordenado de forma alfabética.

5.3.4.1. *AlumnosController*

Nombre del Script	AlumnosController
<i>GameObject</i> que lo contiene	Alumnos (<i>GameObject</i> vacío)
Escena donde actúa	Aplicación

Tabla 2. Información general *AlumnosController*

La principal función de este script es la creación y posterior gestión de los alumnos del entorno. Así mismo, también realiza la importante tarea de crear el avatar que representará al paciente.

Variable	Tipo	Descripción
alumnos	Transform[]	Array donde se almacenan todos los alumnos que se crean
padreAlumnos	GameObject	GameObject que actúa como padre de los modelados de los alumnos
chico1	Transform	Modelado de alumno de piel clara
chico2	Transform	Modelado de alumno de piel clara
chico3	Transform	Modelado de alumno de piel bronceada
chico4	Transform	Modelado de alumno de piel bronceada
chico5	Transform	Modelado de alumno de piel morena

chica1	Transform	Modelado de alumna de piel clara
chica2	Transform	Modelado de alumna de piel clara
chica3	Transform	Modelado de alumna de piel bronceada
chica4	Transform	Modelado de alumna de piel bronceada
chica5	Transform	Modelado de alumna de piel morena
maxAlum	int	Número de alumnos en la clase
genero	int	Porcentaje de chicos en clase
claro	bool	Indica si en el entorno habrá alumnos de piel clara
bronce	bool	Indica si en el entorno habrá alumnos de piel bronceada
moreno	bool	Indica si en el entorno habrá alumnos de piel morena
chicos	Transform[]	Array donde se almacenan los posibles modelos de alumnos
chicas	Transform[]	Array donde se almacenan los posibles modelos de alumnas

Tabla 3. Variables AlumnosController

Función	Descripción
crearVectores	Inicializa todos los vectores y almacena los modelados de alumnos y alumnas en su vector correspondiente
getVecPiel	Crea un vector de enteros, que indican las posiciones que se pueden usar de los vectores de alumnos, en base a los valores de los booleanos que indican el color de piel
crearAlumnos	Crea los alumnos correspondientes teniendo en cuenta los valores de género, máximo de alumnos y color de piel. Los almacena en el array de alumnos y les asigna el padre.
setGenero	Actualiza el valor de genero
setPiel	Actualiza los valores de las tres variables de color de piel
setMaxAlum	Actualiza el valor de maxAlum

crearPaciente	Crea el modelado del paciente en la posición indicada y con los valores que recibe por parámetro para su género y su piel. Además se le asigna el script "PacienteController" y se coloca la cámara principal en la posición de su cabeza
---------------	---

Tabla 4. Funciones AlumnosController

5.3.4.2. AppController

Nombre del Script	AppController
GameObject que lo contiene	Manejador (GameObject vacío)
Escena donde actúa	Aplicación

Tabla 5. Información general AppController

El AppController es quizás el script más importante una vez que se pasa el menú y se entra en la aplicación, pues es este el encargado de asignar los valores elegidos por el terapeuta al resto de elementos de la clase. Además, también se encarga de ordenar la creación del entorno, por lo que se podría decir que sin este, el entorno virtual inmersivo no sería posible.

Variable	Tipo	Descripción
tipoClase	bool	Indica si la clase tendrá o ni ventanas
tipoVentanas	int	Indica como de abiertas estarán las ventanas
muebles	bool	Indica si en la clase habrá o no muebles
objetosMuebles	bool	Indica si los muebles tendrán o no objetos dentro
objetosMesas	bool	Indica si las mesas de los alumnos tendrán o no objetos encima
dist	int	Indica la distribución de las mesas

maxAlum	int	Indica el número de alumnos en la clase
genero	int	Indica el porcentaje de chicos en clase
claro	bool	Indica si en la clase habrá alumnos de piel clara
bronce	bool	Indica si en la clase habrá alumnos de piel bronceada
moreno	bool	Indica si en la clase habrá alumnos de piel morena
posPaciente	int	Indica la posición del paciente
generoPaciente	bool	Indica el género del paciente
pielPaciente	int	Indica el color de piel del paciente
objetosPaciente	bool	Indica si sobre la mesa del paciente habrá o no objetos
ordenPaciente	int	Indica como de ordenados están los objetos sobre la mesa del paciente
sexoProfesor	bool	Indica el sexo del profesor
objetosProfesor	bool	Indica si sobre la mesa del profesor habrá o no objetos

Tabla 6. Variables ApplicationController

Función	Descripción
setVariablesClase	Actualiza todas las variables relacionadas con la clase y con los muebles
setVariablesAlumnos	Actualiza todas las variables relacionadas con los alumnos, sus mesas, sus sillas y su material escolar
setVariablesPaciente	Actualiza todas las variables relacionadas con el paciente
setVariablesProfesor	Actualiza todas las variables relacionadas con el profesor

crearEntorno	Le envía cada una de las variables a su script correspondiente y ordena la creación de todos los elementos que conforman el entorno
--------------	---

Tabla 7. Funciones AppController

5.3.4.3. ButtonController

Nombre del script	ButtonController
GameObject que lo contiene	Canvas
Escena donde actúa	Menú
Otros datos	Sin variables

Tabla 8. Información general ButtonController

Este script, por su parte, se encarga de la detección de los clicks en las distintas pantallas del menú y de llamar, posteriormente, a la clase correspondiente para el tratado de la información.

Función	Descripción
OnClick	Detecta qué botón es pulsado y le indica al script correspondiente qué debe hacer
afectarActivo	Muestra o oculta el objeto que se le pasa por parámetro
textoGenero	Le indica al script "MenuController" que cambie los textos que muestran el porcentaje de alumnos y alumnas
textoNumAlum	Le indica al script "MenuController" que cambie el texto que muestra el número de alumnos en el entorno
textoPosPac	Le indica al script "MenuController" que cambie el texto que muestra la posición del paciente en la clase

Tabla 9. Funciones ButtonController

5.3.4.4. CameraController

Nombre del Script	CameraController
GameObject que lo contiene	Main Camera
Escena donde actúa	Aplicación

Tabla 10. Información general CameraController

La función principal del script CameraController es la de permitir que la cámara del entorno se comporte como la cabeza del paciente en el mundo real, haciendo así que los movimientos del mismo queden plasmados en el entorno tridimensional.

Variable	Tipo	Descripción
velX	float	Indica la velocidad de rotación en el eje X
velY	float	Indica la velocidad de rotación en el eje Y
velMov	float	Indica la velocidad de desplazamiento
ejeX	float	Almacena el ángulo de rotación del eje X
ejeY	float	Almacena el ángulo de rotación del eje Y

Tabla 11. Variables CameraController

Función	Descripción
Update	Actualiza posición y rotación de la cámara en cada frame

Tabla 12. Funciones CameraController

5.3.4.5. CargadorSesiones

Nombre del Script	CargadorSesiones
<i>GameObject</i> que lo contiene	ListaOpciones (panel donde aparecen los botones con las distintas sesiones guardadas)
Escena donde actúa	Menú

Tabla 13. Información general CargadorSesiones

El CargadorSesiones, como su propio nombre indica, realiza la vital tarea de leer el fichero de Guardado y generar tantos botones como sesiones haya almacenadas en el fichero. La importancia de esta clase radica en que gracias a ella es posible que el terapeuta pueda repetir una sesión con los mismos valores exactos que guardo, lo cual es de gran relevancia dentro del tratamiento de fobias.

Variables	Tipo	Descripción
opcionActiva	int	Indica cual es la última sesión que ha pulsado el terapeuta
textoSesion	Text	Texto en el que se muestra el nombre de la última sesión que se haya pulsado
sesionElegida	bool	Indica si se ha pulsado ya o no sobre alguna sesión
gridOpciones	<i>GameObject</i>	<i>GameObject</i> sobre el que se crean los botones que indican las sesiones
botonOpciones	Button	Prefab con el modelo de botón que se crea para cada sesión
botones	Button[]	Array donde se almacenan los botones que se crean

Tabla 14. Variables CargadorSesiones

Función	Descripción
Start	Inicializa las variables, crea el vector de botones y llama a la función "cargarOpciones"
cargarOpciones	Lee el fichero de Guardado y crea un botón por cada sesión que lee
setOpcionActiva	Pone a true la variable sesionElegida, actualiza el valor de opcionActiva y el texto de textoSesion
cargarSesionGuardada	Abre el fichero de carga y escribe en él los datos de la sesión elegida para cargar. Después llama al script "MenuController" para lanzar la aplicación.

Tabla 15. Funciones CargadorSesiones

5.3.4.6. ClaseCreator

Nombre del Script	ClaseCreator
<i>GameObject</i> que lo contiene	EstructuraClase (<i>GameObject</i> vacío)
Escena donde actúa	Aplicación

Tabla 16. Información general ClaseCreator

Este script se encarga de la creación de la estructura de la clase, así como de la generación o no de ventanas en el entorno. Esta tarea, aunque parezca irrelevante, es la que permite que el espacio virtual del aula pueda ser usado de forma eficaz en el tratamiento de claustrofobia.

Función	Tipo	Descripción
clase	Transform	Transform sobre la que se crea la estructura de la clase
padreClase	<i>GameObject</i>	Padre de la clase que se crea

claseVentanas	Transform	Almacena el modelado de clase con huecos para las ventanas
claseCerrada	Transform	Almacena el modelado de clase sin huecos para las ventanas
ventanas	Transform[]	Array donde se almacenan las ventanas que se crean
padreVentanas	<i>GameObject</i>	Padre de las ventanas que se crean
abierta	Transform	Almacena el modelado de ventana abierta
medio	Transform	Almacena el modelado de ventana medio abierta
cerrada	Transform	Almacena el modelado de ventana cerrada
tipoClase	bool	Indica si la clase tiene o no ventanas
tipoVentanas	int	Indica como de abiertas estarán las ventanas en el entorno

Tabla 17. Variables *ClaseCreator*

Función	Descripción
crearClase	Crea la clase y las ventanas en base a las variables tipoClase y tipoVentanas
setTipoClase	Actualiza la variable tipoClase
setTipoVentanas	Actualiza la variable tipoVentanas

Tabla 18. Funciones *ClaseCreator*

5.3.4.7. Guardado

Nombre del Script	Guardado
<i>GameObject</i> que lo contiene	Memoria (<i>GameObject</i> vacío)

Escena donde actúa	Menú
Otros datos	Posee una función set para todas sus variables que se activa cada vez que una opción es modificada en el menú (excepto para las de tipo InputField)

Tabla 19. Información general Guardado

Si el ApplicationController era el script más importante de la aplicación, este es, sin duda alguna, el del menú. La clase Guardado es la encargada de almacenar todos los valores de personalización que seleccione el terapeuta con el fin de poder escribirlos posteriormente en los fichero de Carga y, si así ha sido seleccionado, de Guardado. Este script, por lo tanto, posibilita que el entorno sea tan personalizable y, por lo tanto, que se puedan realizar tratamientos flexibles mediante su uso.

Variable	Tipo	Descripción
tipoClase	bool	Indica si la clase tendrá o no ventanas
tipoVentanas	int	Indica como de abiertas estarán las ventanas
muebles	bool	Indica si en la clase habrá o no muebles
objetosMuebles	bool	Indica si los muebles tendrán o no objetos dentro
objetosMesas	bool	Indica si las mesas de los alumnos tendrán o no objetos encima
dist	int	Indica la distribución de las mesas
maxAlum	int	Indica el número de alumnos en la clase
genero	int	Indica el porcentaje de chicos en clase
claro	bool	Indica si en la clase habrá alumnos de piel clara

bronce	bool	Indica si en la clase habrá alumnos de piel bronceada
moreno	bool	Indica si en la clase habrá alumnos de piel morena
posPaciente	int	Indica la posición del paciente
generoPaciente	bool	Indica el género del paciente
pielPaciente	int	Indica el color de piel del paciente
objetosPaciente	bool	Indica si sobre la mesa del paciente habrá o no objetos
ordenPaciente	int	Indica como de ordenados están los objetos sobre la mesa del paciente
sexoProfesor	bool	Indica el sexo del profesor
objetosProfesor	bool	Indica si sobre la mesa del profesor habrá o no objetos
fobia	InputField	InputField donde se escribe el nombre de la fobia que se tratará en la sesión
nombre	InputField	InputField donde se escribe el nombre del paciente al que se trata
numSesion	InputField	InputField donde se escribe el número de la sesión para el paciente

Tabla 20. Variables Guardado

Función	Descripción
Start	Inicializa todas las variables a sus valores por defecto
guardar	Escribe los parámetros seleccionado en el fichero de guardado
crearFicheroCarga	Crea el fichero Carga y escribe en él los parámetros seleccionados

Tabla 21. Funciones Guardado

5.3.4.8. ImagenesController

Nombre del Script	ImagenesController
GameObject que lo contiene	ImagenesMuestras (<i>GameObject</i> que agrupa las cinco imágenes posibles, cada una con una distribución de mesas distinta)
Escena donde actúa	Menú

Tabla 22. Información general ImagenesController

ImagenesController se encarga de la activación y desactivación de las imágenes que muestran, si así se desea, las distintas distribuciones de mesas dentro del aula escolar. Así mismo, le facilita al terapeuta el seleccionado de la posición del alumno de forma objetiva, ya que las propias imágenes indican el orden de llenado de las mesas y, por lo tanto, las posiciones de la mesas en el vector.

Variable	Tipo	Descripción
img0	<i>GameObject</i>	Imagen de distribución de mesas individual
img1	<i>GameObject</i>	Imagen de distribución de mesas en parejas
img2	<i>GameObject</i>	Imagen de distribución de mesas en tríos
img3	<i>GameObject</i>	Imagen de distribución de mesas en grupos
img4	<i>GameObject</i>	Imagen de distribución de mesas en forma de U
drop	Dropdown	Desplegable con todas las opciones de distribución

Tabla 23. Variables ImagenesController

Función	Descripción
Update	Decide qué imagen se muestra en base al valor seleccionado en el Dropdown

Tabla 24. Funciones ImagenesController

5.3.4.9. LectorCreador

Nombre del script	LectorCreador
<i>GameObject</i> que lo contiene	CreadorApp (<i>GameObject</i> vacío)
Escena donde actúa	Aplicación
Otros datos	Sin variables. Se autodestruye al terminar su función

Tabla 25. Información general LectorCreador

Una vez que se han seleccionado los parámetros indicados en el menú y se da inicio a la aplicación, es cuando entra en acción este script. El LectorCreador es el encargado de leer el fichero de Carga y de comunicarle los valores que lee al ApplicationController. Por lo tanto, este código es el que permite que las opciones seleccionadas por el terapeuta se plasmen dentro de la aplicación.

Función	Descripción
cargarEntorno	Abre y lee el fichero de Carga. Conforme lo lee va pasándole todos los parámetros al script "AppController". Cuando termina de leerlo, le dice al "AppController" que cree el entorno y se autodestruye.

Tabla 26. Funciones LectorCreador

5.3.4.10. MenuController

Nombre del Script	MenuController
GameObject que lo contiene	Fondo (<i>GameObject</i> que contiene los elementos del fondo)
Escena donde actúa	Menú

Tabla 27. Información general MenuController

Mientras que el ButtonController es el encargado de detectar los clicks en el menú, el MenuController es quien realiza el tratamiento oportuno con la información que se genera. Entre sus funciones se encuentran el paso de pantallas, la actualización de textos y la ocultación de algunos seleccionables que dependen directamente de otros (por ejemplo: si se elige que la clase no tenga ventanas, se oculta de desplegable que da a elegir cómo de abiertas se quieren las mismas).

Por otra parte, es importante decir que este script también se encarga de darle un efecto de fundido al menú, cuando se entra y sale del mismo. Con este efecto aparentemente sencillo se logra evitar los cambios bruscos de escena que puedan llegar a desconcertar a los usuarios.

Variables	Tipo	Descripción
fadeOutTexture	Texture2D	Textura sobre la que hace la transparencia
fadeSpeed	float	Velocidad a la que se hace la transparencia
alpha	float	Transparencia de la textura
fadeDir	int	Dirección en la que se realiza la transparencia
elementosPantalla1	<i>GameObject</i>	Padre de todos los elementos de la interfaz para la primera pantalla

elementosPantallaPrincipal	<i>GameObject</i>	Padre de todos los elementos de la interfaz para la pantalla principal
elementosPantallaSecundaria	<i>GameObject</i>	Padre de todos los elementos de la interfaz para la pantalla secundaria
elementosPantallaCargar	<i>GameObject</i>	Padre de todos los elementos de la interfaz para la pantalla de carga
elementosPantallaGuardar	<i>GameObject</i>	Padre de todos los elementos de la interfaz para la pantalla de guardado
mainCam	Camera	Cámara principal del menú
pantallaPrincipal	Vector3	Posiciones de la cámara para el correcto visualizado de la pantalla principal y secundaria
pantalla1	Vector3	Posiciones de la cámara para el correcto visualizado de la primera pantalla
pantallaCargar	Vector3	Posiciones de la cámara para el correcto visualizado de la pantalla de carga
pantallaGuardar	Vector3	Posiciones de la cámara para el correcto visualizado de la pantalla de guardar
imagenes	<i>GameObject</i>	Padre de las imágenes que muestran la distribución de mesas
activo	bool	Indica si las imágenes están activas o no
nChicos	Text	Texto que muestra el porcentaje de chicos en el aula
nChicas	Text	Texto que muestra el porcentaje de chicas en el aula
nAlumnos	Text	Texto que muestra el número de alumnos en el aula
posPaciente	Text	Texto que muestra la posición del paciente en el aula

Tabla 28. Variables MenuController

Función	Descripción
Start	Inicializas los vectores de posiciones de la cámara, almacena la cámara y le indica que se ponga en el posición de la primera pantalla
mostrarOcultarImg	Si las imágenes están visibles las oculta y viceversa
cambiarTexto	Actualiza los textos que muestran el porcentaje de alumnos y alumnas
textoNumAlumnos	Actualiza el texto que muestra el número de alumnos
textoPosPaciente	Actualiza el texto que muestra la posición del paciente
cambiarPantalla	Cambia la posición de la cámara, activa los elementos de la pantalla actual y oculta los del resto
OnGui	Realiza los cambios en la opacidad de la textura para poder llevar a cabo los fundidos en negro
BeginFade	Actualiza la dirección de la transparencia (si hacia la opacidad total o hacia la ausencia de opacidad)
OnLevelWasLoaded	Cuando la escena se carga, ordena empezar el fundido en negro
cambiarEscena	Lanza funciones asíncronas para realizar un fundido en negro mientras se carga la siguiente escena
empezarApp	Ordena al script "Guardado" que genere el fichero de Carga y llama al método "cambiarEscena"
cargarSesion	Llama al método "cambiarEscena"

Tabla 29. Funciones MenuController

Dado que este script presenta una programación tan diferenciada del resto de clases, se pondrá a continuación el código que permite realizar los fundidos en negro para entrar y salir de la escena del menú. Cabe destacar que estos se pondrán como una imagen con el objetivo de respetar el esquema de colores del editor de textos y que, de ese modo, tenga una lectura más clara.

```

void OnGUI()
{
    //Se calcula alpha en base a la direccion, la velocidad y el tiempo
    alpha += fadeDir * fadeSpeed * Time.deltaTime;
    alpha = Mathf.Clamp01(alpha); //se le da un valor entre 0 y 1

    GUI.color = new Color(GUI.color.r, GUI.color.g, GUI.color.b, alpha); //se cambia el alpha del color
    GUI.depth = -1000; //se asigna un numero bajo para que sea lo ultimo en dibujarse y, por lo tanto, este encima de todo
    GUI.DrawTexture(new Rect(0, 0, Screen.width, Screen.height), fadeOutTexture); //se dibuja la textura en toda la pantalla
}

public float BeginFade(int direction)
{
    fadeDir = direction; //Se asigna un valor a la direccion
    return (fadeSpeed); //Se devuelve el tiempo para que sea mas facil calcular el Application.LoadLevel()
}

void OnLevelWasLoaded()
{
    BeginFade(-1);
}

IEnumerator cambiarEscena()
{
    float fadeTime = BeginFade(1);
    yield return new WaitForSeconds(fadeTime);
    //Carga la escena correspondiente
    SceneManager.LoadScene(1);
    //SceneManager.LoadScene("Aplicacion", LoadSceneMode.Single);

    BeginFade(-1);
}

public void empezarApp()
{
    GameObject.Find("Memoria").GetComponent<Guardado>().crearFicheroCarga();
    StartCoroutine(cambiarEscena());
}

```

Ilustración 78. Código para realizar fundidos en negro.

Fuente: propia

5.3.4.11. MesasCreator

Nombre del Script	MesasCreator
GameObject que lo contiene	MobiliarioAlumnos (GameObject que agrupa las mesas, sillas y objetos de los alumnos)
Escena donde actúa	Aplicación

Tabla 30. Información general MesasCreator

El script de MesasCreator tiene la vital tarea de crear, rotar y posicionar todas las mesas y sillas del aula, así como del material escolar que pueda haber esparcido sobre estas. Por lo tanto, esta clase es la encargada de llevar a cado una de las funcionalidades

más importantes y que más relevancia tiene dentro del entorno: la distribución de las mesas en el aula escolar.

Variable	Tipo	Descripción
mesas	Transform[]	Array que almacena todas las mesas que se crean
padreMesas	<i>GameObject</i>	Padre de todas las mesas que se crean
table	Transform	Modelado de la mesa del alumnado
sillas	Transform[]	Array que almacena todas las sillas que se crean
padreSillas	<i>GameObject</i>	Padre de todas las sillas que se crean
chair	Transform	Modelado de la silla del alumnado
objetos	Transform[]	Array que almacena todos los objetos que se crean
padreObjetos	<i>GameObject</i>	Padre de todos los objetos que se crean
orden	Transform	Modelado del material escolar ordenado del alumno
desorden1	Transform	Modelado del material escolar desordenado del alumno
desorden2	Transform	Modelado del material escolar muy desordenado del alumno
padreObjPaciente	<i>GameObject</i>	Padre del avatar del paciente
conObjetos	bool	Indica si sobre las mesas de los alumnos habrá o no material escolar
dist	int	Indica la distribución de las mesas
maxAlum	int	Indica el número de alumnos en el aula

Tabla 31. Variables MesasCreator

Función	Descripción
crearClase	Crea los vectores de sillas, mesas y alumnos y llama a los métodos "colocarMesas", "colocarObjetos" y "colocarSillas", en ese orden
colocarObjetos	En base al booleano conObjetos crea y coloca o no los objetos en 3 de cada 4 mesas, en la misma posición de la misma
colocarSillas	Crea y coloca las sillas en relación a la posición y rotación de las mesas
colocarMesas	Dependiendo del valor de dist, llama a un método o a otro según las mesas deban estar en una distribución o en otra
individuales	Crea y coloca las mesas en distribución individual
parejas	Crea y coloca las mesas en distribución en parejas
trios	Crea y coloca las mesas en distribución en tríos
grupos	Crea y coloca las mesas en distribución en grupos
formaU	Crea y coloca las mesas en distribución en forma de U
setObjetos	Actualiza el valor del booleano conObjetos
setDistribucion	Actualiza el valor del entero dist
setMaxAlum	Actualiza el valor del entero maxAlum
getPosSillas	Devuelve un array de Vector3 con las posiciones de todas las sillas
getRotSillas	Devuelve un array de Quaternions con la rotación de todas las sillas
setOrdenPaciente	Crea los objetos de la mesa del alumno, en caso de que sí deban crearse, y en el orden que recibe por parámetro

Tabla 32. Funciones MesasCreator

5.3.4.12. MobiliarioController

Nombre del Script	MobiliarioController
<i>GameObject</i> que lo contiene	MobiliarioClase (<i>GameObject</i> que agrupa las estanterías, casilleros, percheros, papeleras y material escolar del entorno)
Escena donde actúa	Aplicación

Tabla 33. Información general MobiliarioController

Este sencillo script se encarga, por su parte, de mostrar u ocultar el mobiliario de la clase (estantería, casillero, percheros y papeleras), el material escolar que se haya en este y los distintos objetos que hay sobre la mesa del profesor.

Variable	Tipo	Descripción
mobiliarioClase	<i>GameObject</i>	Agrupar el estanterías, casillero, papeleras y percheros de la clase
objetosMobiliario	<i>GameObject</i>	Agrupar el material escolar que está dentro de los muebles
ObjetosProfesor	<i>GameObject</i>	Material escolar que hay sobre la mesa del profesor
muebles	bool	Indica si en la clase habrá o no muebles
objetos	bool	Indica si en la clase habrá o no material escolar
profe	bool	Indica si sobre la mesa del profesor habrá o no material escolar

Tabla 34. Variables MobiliarioController

Función	Descripción
Update	Actualiza si el mobiliario, sus objetos y los objetos del profesor se ven o no
setMuebles	Actualiza el valor del booleano muebles
setObjetos	Actualiza el valor del booleano objetos
setObjetosProfesor	Actualiza el valor del booleano profe

Tabla 35. Funciones MobiliarioController

5.3.4.13. OpcionController

Nombre del Script	OpcionController
<i>GameObject</i> que lo contiene	prefab BotonOpciones (plantilla de los botones que crean al leer el fichero de Guardado)
Escena donde actúa	Menú

Tabla 36. Información general OpcionController

OpcionController es el encargado de almacenar la información básica de las sesiones que se cargan desde el fichero de guardado, es decir, este script se asigna a cada uno de los botones que se crean por cada sesión almacenando en él el nombre de esta y la posición de la misma dentro del fichero de guardado. Así mismo, cuando se clic encima de uno de dichos botones, OpcionController le manda al CargadorSesiones el nombre de la sesión y su posición, para que este pueda saber cuál ha sido la sesión elegida por el terapeuta para ser cargada.

Variable	Tipo	Descripción
posOpcion	int	Indica el orden de creación del botón que contiene al script, empezando por 0

sesion	string	Nombre de la sesión guardada que representa el botón
--------	--------	--

Tabla 37. Variables OpcionController

Función	Descripción
sendPosOpcion	Cuando el botón es pulsado, le envía al script "cargadorSesiones" su entero posOpcion
setSesion	Actualiza el valor del string sesion
getSesion	Devuelve el valor del string sesion
setPosOpcion	Actualiza el valor del entero posOpcion
getPosOpcion	Devuelve el valor del entero posOpcion

Tabla 38. Funciones OpcionController

5.3.4.14. PacienteController

Nombre del Script	PacienteController
<i>GameObject</i> que lo contiene	-
Escena donde actúa	Aplicación
Otros datos	La casilla " <i>GameObject</i> que lo contiene" se ha dejado vacía debido a que este script en concreto se asigna en el momento en que el script "AlumnosController" crea el modelado del avatar del paciente

Tabla 39. Información general PacienteController

Ha decir verdad, este script no presenta una funcionalidad asombrosa, ya que se dedica únicamente a decirle a MesasCreator que coloque el material escolar del paciente de una forma u otra. Sin embargo, la verdadera utilidad de PacienteController es la posible ampliación de las funcionalidades del avatar del paciente, es decir, la existencia de

esta clase facilita en gran medida cualquier tarea de añadido de código que se pudiera realizar en el futuro.

Variable	Tipo	Descripción
genero	bool	Indica el género del paciente
piel	int	Indica el color de piel del paciente
posPaciente	int	Indica la posición del paciente
orden	int	Indica como de ordenados está el material escolar sobre la mesa del paciente
conObjetos	bool	Indica si sobre la mesa del paciente habrán o no objetos

Tabla 40. Variables PacienteController

Función	Descripción
setDatos	Actualiza los valores de todas las variables
ordenMesa	Indica al script "MesasCreator" cuál es la mesa del paciente, si hay o no objetos y en qué orden están dichos objetos

Tabla 41. Funciones PacienteController

5.3.4.15. TeacherController

Nombre del Script	TeacherController
GameObject que lo contiene	Terapeuta (GameObject vacío)
Escena donde actúa	Aplicación

Tabla 42. Información general TeacherController

Por último, y al igual que el `PacienteController`, la verdadera utilidad de este script radica en cualquier posible ampliación de las funcionalidades del profesor. Pero si nos centramos en lo que realmente realiza, `TeacherController` se encarga de la generación del modelado del profesor.

Variable	Tipo	Descripción
teacher	Transform	Donde se almacena el modelado del profesor/a que se crea
padre	<i>GameObject</i>	Padre del profesor
hombre	Transform	Almacena el modelado del profesor
mujer	Transform	Almacena el modelado de la profesora
sexo	bool	Indica si el profesor será hombre o mujer

Tabla 43. Variables `TeacherController`

Función	Descripción
crearModelo	Crea el modelado del profesor en base al booleano sexo
setSexo	Actualiza el valor del booleano sexo

Tabla 44. Funciones `TeacherController`

5.4. Últimas mejoras del entorno

Como en todo gran proyecto, el último paso siempre es el testeo del mismo con el objetivo de localizar y corregir aquellos aspectos de la aplicación que así lo requieran y, en el caso de la que nos ocupa, fueron más de los que esperaba.

El visualizado en el dispositivo móvil resultó ser mucho peor de lo que convendría de una aplicación con el propósito de tratar fobias, las cuales exigen que el tiempo de respuesta del sistema sea casi inmediato.

Con el fin de mejorar el rendimiento del programa, se incluyeron llamadas "Destroy" en todos los métodos que implicaban la creación y almacenamiento de *GameObjects*. Estas se ubicaron después de la inicialización de las variables y antes de que se les asignara el valor real. El motivo de dicha posición dentro del código tiene como meta eliminar cualquier posible resto en memoria de la ejecución anterior.

```
public void crearClase()
{
    mesas = new Transform[30];
    sillas = new Transform[30];
    objetos = new Transform[30];

    for(int i = 0; i < 30; i++)
    {
        Destroy(mesas[i]);
        Destroy(sillas[i]);
        Destroy(objetos[i]);
    }

    colocarMesas();
    colocarObjetos();
    colocarSillas();
}
```

Ilustración 79. Ejemplo de Destroys en el código.

Fuente: propia

Por otra parte, se incluyó código que eliminaba *GameObjects* vacíos en aquellos métodos donde se generaban como parte del proceso por algún motivo concreto.

```
//Borra todos los gameobjects vacíos que se han creado durante el proceso
for(int j = 0; j < 30; j++)
{
    if(objetos[j].GetComponent<Component>().Length == 1)
    {
        Destroy(objetos[j].gameObject);
    }
}
```

Ilustración 80. Ejemplo de eliminado de *GameObjects* vacíos.

Fuente: propia

Estos cambios, aunque útiles, resultaron insuficientes para solucionar el problema de rendimiento, por lo que se optó por reducir el número de polígonos en escena, siempre intentando no comprometer el aspecto visual de la misma y, por lo tanto, el grado de realismo e inmersión que esta pudiera causar.

Los modelados sobre los que se realizaron más cambios son aquellos cuya malla incluía cilindros (lo cuales eran la gran mayoría) y anillas (las libretas) pues se consideró que en ellos sería más sencillo reducir el número de polígonos.

Antes de exponer mediante imágenes los cambios más notables, comentar que se intentó reducir el número de polígonos en los modelados de personajes, pero estos presentaban cambios en el acabado final demasiado grandes como para realizarlos. Por ello la reducción de vértices se centró en los elementos de la clase.

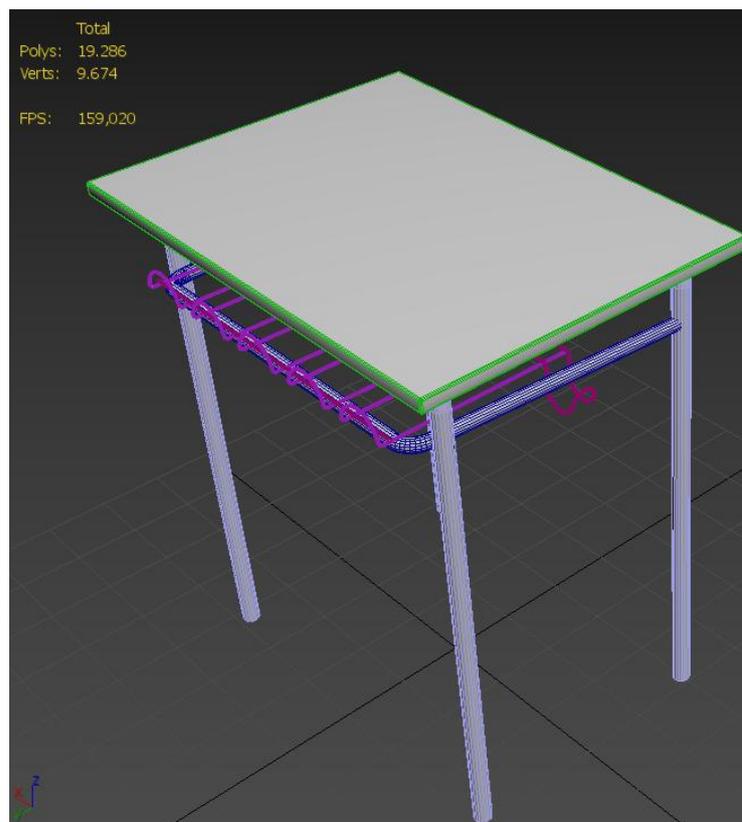


Ilustración 81. Mesa del alumnado antes del cambio del número de polígonos.

Fuente: propia

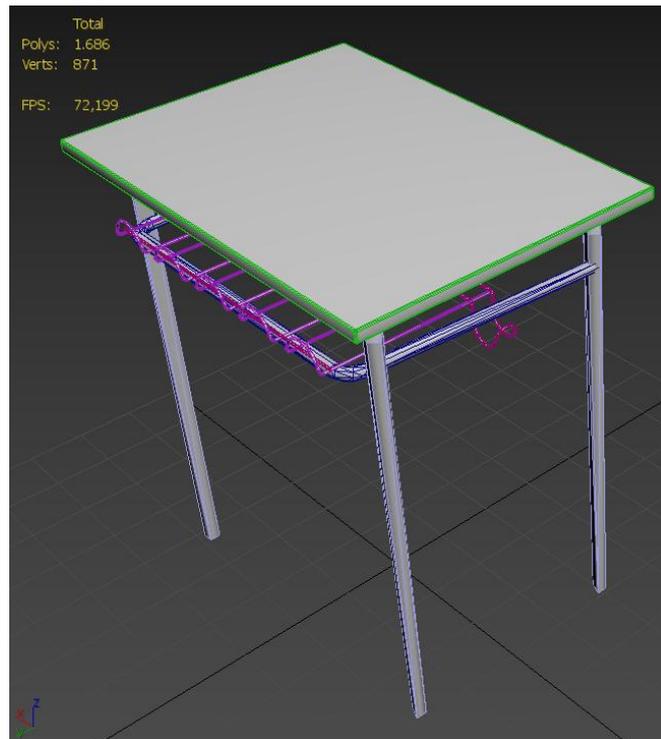


Ilustración 82. Mesa del alumnado después del cambio del número de polígonos.

Fuente: propia

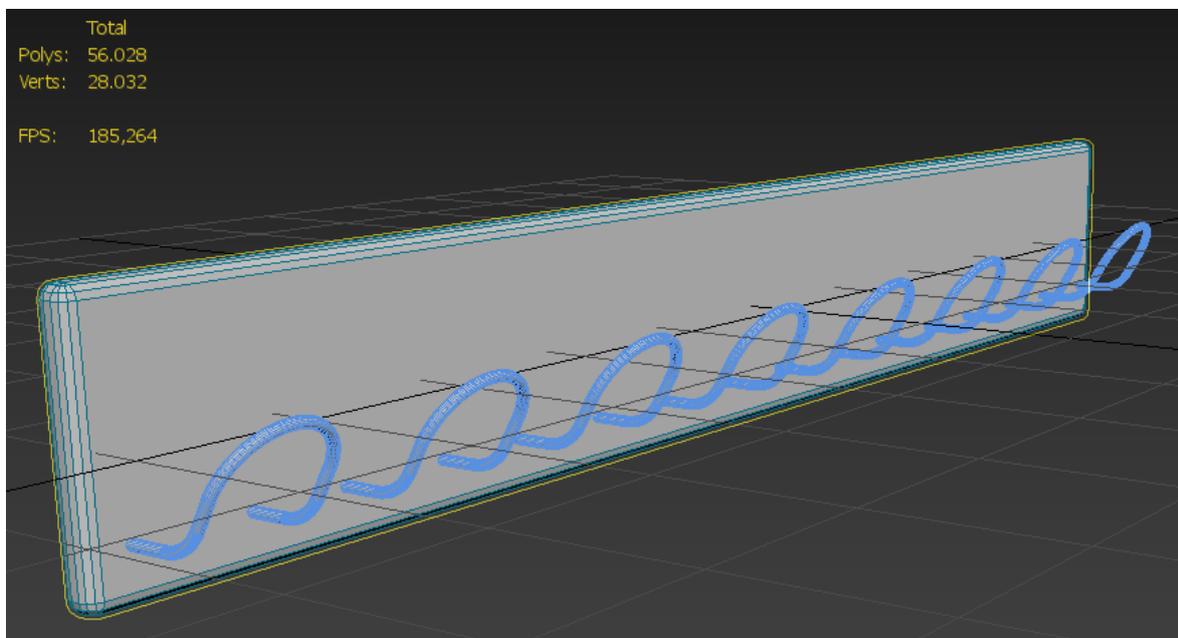


Ilustración 83. Perchero antes del cambio del número de polígonos.

Fuente: propia

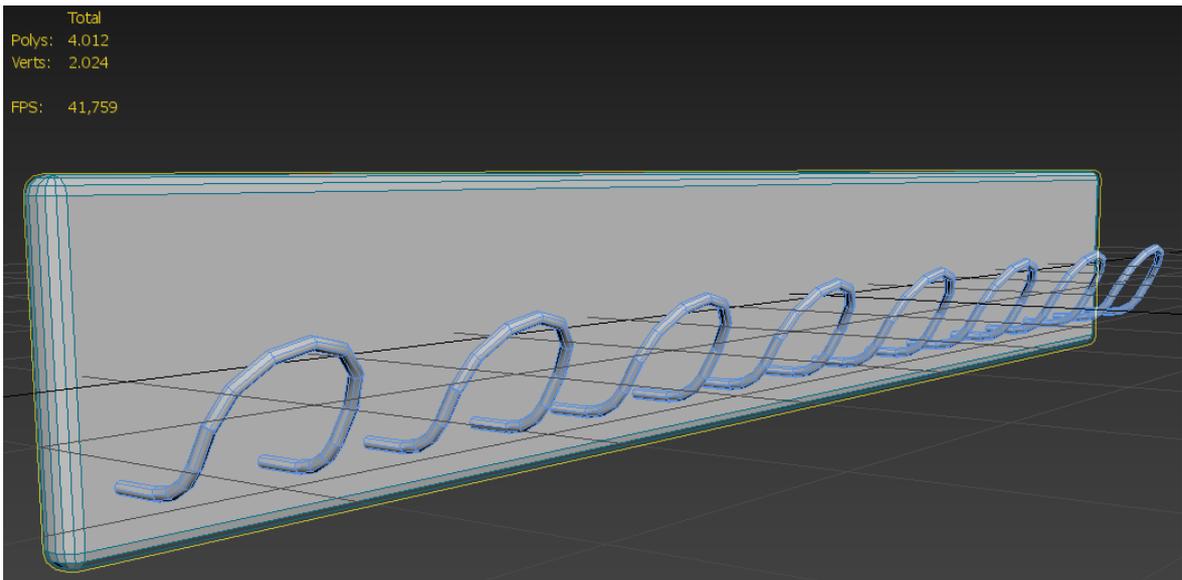


Ilustración 84. Perchero después del cambio del número de polígonos.

Fuente: propia

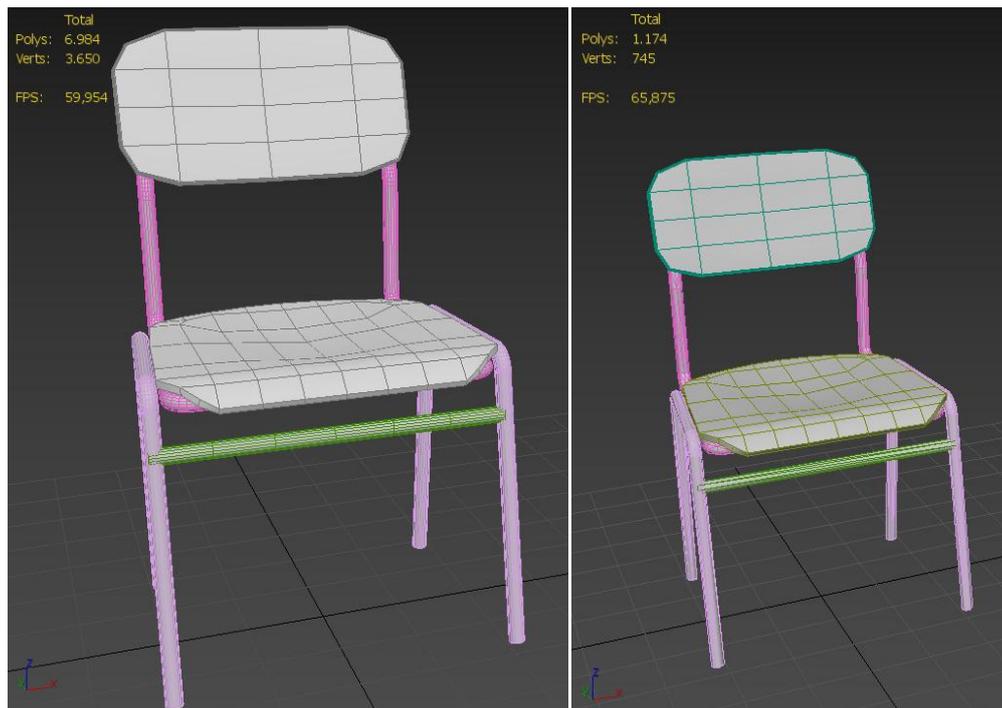


Ilustración 85. Silla del alumnado antes (izquierda) y después (derecha) del cambio del número de polígonos.

Fuente: propia

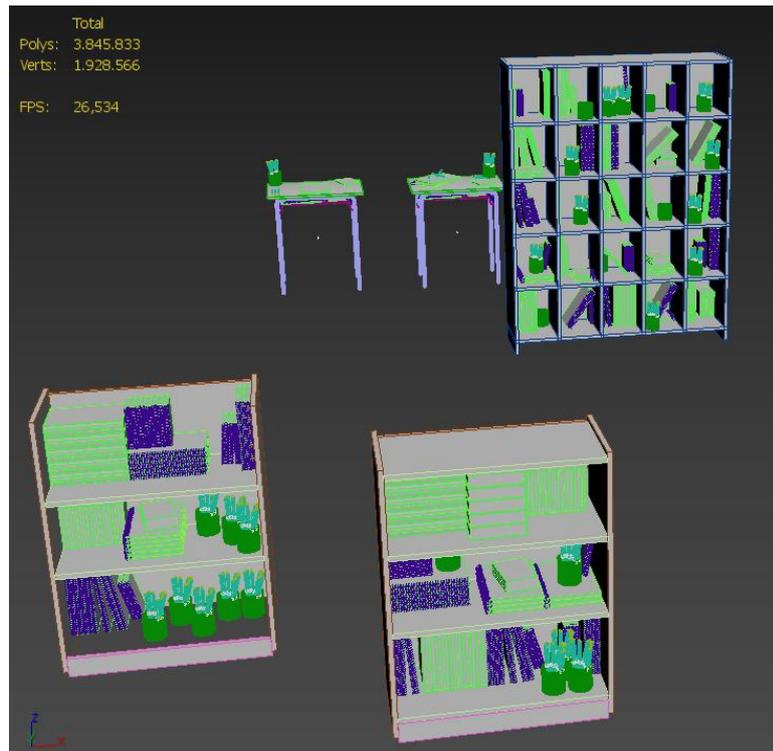


Ilustración 86. Objetos y muebles antes del cambio del número de polígonos.

Fuente: propia

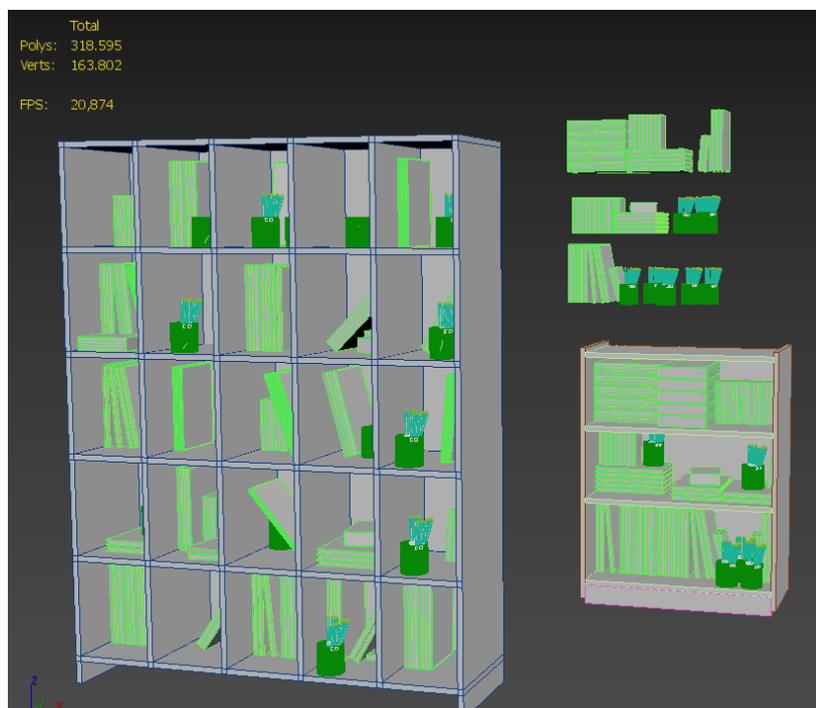


Ilustración 87. Objetos y muebles después del cambio del número de polígonos.

Fuente: propia

Con estos últimos cambios se logró reducir a la mitad el número de vértices en escena, mejorando considerablemente el rendimiento final de la aplicación, a pesar de que continuó siendo insuficiente.

Sin embargo, se consiguió que el entorno reaccionara de forma instantánea al movimiento de cabeza del paciente en aquellas configuraciones de la clase que permiten el menor número de elementos en escena, algo que en un primer momento se consideraba inalcanzable.

Para finalizar, se decidió crear un icono personalizado para la aplicación, el cual, aunque no supone una mejora en sí del sistema, supuso el broche final de este trabajo de fin de grado: la creación de un entorno virtual inmersivo para la detección y tratamiento de problemas de salud mental.



Ilustración 88. Icono de la aplicación.

Fuente: propia

6. Conclusiones

Llegados a este punto, se puede afirmar que se han cumplido todos los objetivos que se plantearon al inicio del desarrollo de este trabajo de fin de grado. Haciendo memoria, estos objetivos fueron:

1. Investigar y comprender la realidad virtual.
2. Investigar y comprender el tratamiento de fobias.
3. Investigar y comprender el uso de la realidad virtual en el tratamiento de fobias.
4. Modelar y texturizar un entorno geométrico tridimensional fiel a un aula de colegio.
5. Crear un entorno virtual inmersivo lo suficientemente personalizable como para hacer viable su uso en psicología clínica.
6. Incluir modelado de personajes en el entorno.
7. Hacer que el entorno virtual sea usable mediante dispositivos de realidad virtual.

Fue durante la tarea de documentación cuando se llevaron a cabo los tres primeros objetivos de este proyecto. Sin duda alguna, se logró encontrar y filtrar la suficiente información que permitió que las posteriores decisiones tomadas para con el entorno virtual inmersivo para la detección y tratamiento de problemas de salud mental fueran sobre fundamentos sólido.

Por otra parte, al finalizar la tarea de creación de los elementos del entorno se logró contar con los suficientes modelados tridimensionales como para generar un aula escolar. Por lo tanto, se puede afirmar que los objetivos 4 y 6 se han cumplido de forma satisfactoria.

Así mismo, mediante el uso de Unity, se ha logrado cumplir los objetivos 5 y 7, consiguiendo de esta manera que todos y cada uno de los mismos se hayan realizado y

que, por lo tanto, se ha logrado crear un entorno virtual inmersivo útil para el tratamiento de fobias concretas en niños en edad escolar.

Como conclusión, me gustaría añadir que, a pesar de las distintas dificultades que fueron surgiendo a lo largo del desarrollo, se ha logrado respetar en gran medida los tiempos estimados que se asignaron a cada tarea y, por lo tanto, se puede considerar que este proyecto ha recibido tanto como se esperaba.

Tareas	Tiempo invertido
Documentación	35 horas
Creación de los elementos del entorno	75 horas
Creación del entorno	115 horas
Realización de la memoria	100 horas
Horas totales	325 horas

Tabla 45. Relación de tareas principales y el tiempo invertido en cada una

A modo de cierre personal, quería añadir que este trabajo ha supuesto una montaña rusa de emociones y que, incluso así, ha supuesto una de las experiencias más enriquecedoras de mi vida, tanto a nivel personal como de estudios.

Por último, me gustaría decir que se tiene constancia de que el proyecto es mejorable y, por lo tanto, se enumerarán todas las mejoras que podrían hacer del entorno tridimensional creado una herramienta redonda.

6.1. Posibles mejoras

En primer lugar, y por razones más que evidentes, la primera mejora que en este sistema se debe considerar es en tema del rendimiento del mismo, pues es este, muy a mi pesar, el aspecto que peor se ha gestionado dentro del desarrollo del proyecto.

Por otro lado, una buena forma de enriquecer el entorno sería la de mejorar los modelados de personajes. Éstos, a pesar de que gozan de un nivel de realismo aceptable, podrían presentar un aspecto más orgánico que dotara de más inmersión al proyecto.

Así mismo, el añadido de animaciones en los personajes aumentaría enormemente el valor de la herramienta. Dichas animaciones se deberían aplicar no solamente al desplazamiento y movimiento de los modelados, sino que se podrían crear unas que se encargaran del movimiento facial, dotando así a los mismos de reacciones creíbles.

Con el añadido de animaciones, se podría programar el entorno para que permitiera que el paciente se pudiera mover por el entorno, aumentando así la interactividad del espacio tridimensional. Por otro lado, otra mejora derivada sería que los alumnos podrían reaccionar a ciertas acciones del paciente, logrando de esta manera un acabado mucho más realista.

Otra de las funcionalidades de gran interés sería la de añadir un mecanismo que permitiera que el terapeuta pudiera mover a voluntad las facciones del profesor. Con esto último se pretende dar al psicólogo la opción de, por ejemplo, hacer el modelado del profesorado moviera los labios cuando él quisiera hablar y, por lo tanto, generar en el paciente la idea de que el profesor se trata realmente del terapeuta que se encuentra también dentro del entorno.

Llegando al final, me gustaría comentar que otra potencial mejora sería la de añadir más opciones de personalización a la hora de crear el personaje del profesor por razones parecidas a las explicadas en el párrafo anterior.

Por último, señalar que la inclusión de algunos sonidos podría ser uno de los aspectos clave para mejorar el acabado final del entorno. Entre estos, se podría añadir conversaciones de fondo de algunos alumnos, ruidos del exterior o incluso el ruido de las manecillas de algún posible reloj. Otro sonido que podría resultar de extraordinaria versatilidad, entre otros, sería el de incluir distintas sirenas (policía, bomberos, alarma de incendios...) con el fin de poder evaluar la reacción del paciente ante situaciones de estrés elevado.

7. Bibliografía y referencias

1. *Academia Android*. (12 de noviembre de 2015). Recuperado el 29 de agosto de 2017, de Creación de Scripts en Unity: <https://academiaandroid.com/crear-scripts-en-unity/>
2. *Academia Android*. (2 de noviembre de 2015). Recuperado el 29 de agosto de 2017, de Scripts y lenguajes de programación en Unity: <https://academiaandroid.com/scripts-lenguajes-programacion-unity/>
3. *Animat 3D*. (30 de marzo de 2009). Recuperado el 28 de agosto de 2017, de Bake en Blender: <https://animat3d.wordpress.com/2009/03/30/bake-en-blender/>
4. *Autodesk Navisworks 2016*. (2016). Recuperado el 28 de agosto de 2017, de Asignación de una textura a un material: <http://help.autodesk.com/view/NAV/2016/ESP/?guid=GUID-CE17ADAD-7C79-4395-B657-7F309356D916>
5. *Centro de excelencia de software libre*. (s.f.). Recuperado el 28 de agosto de 2017, de Introducción al Modelado 3D: <http://www.esi.uclm.es/www/cglez/fundamentos3D/02.01.Introduccion.html>
6. Cobián, Á. B. (s.f.). *Modelado 3D*. Recuperado el 28 de agosto de 2017, de Conceptos fundamentales de la modelación 3D: <http://abc.mitreum.net/wp-content/uploads/clase2-parte1-teoria.pdf>
7. Cristina Botella, R. B.-P. (2007). La utilización de las nuevas tecnologías de la información y la comunicación en psicología clínica. *uocpapers* (4), 41.
8. desconocido, A. (s.f.). *www.academia.edu*. Recuperado el 27 de agosto de 2017, de http://www.academia.edu/4254583/Cap%C3%ADtulo_2_Introducci%C3%B3n_a_la_Realidad_Virtual
9. *Diccionario de informática y tecnología*. (s.f.). Recuperado el 28 de agosto de 2017, de Definición de Modelado en 3D: http://www.alegsa.com.ar/Dic/modelo_en_3d.php
10. Geig, M. (19 de diciembre de 2013). *informIT*. Recuperado el 28 de agosto de 2017, de Working with Models, Materials, and Textures in Unity Game Development: <http://www.informit.com/articles/article.aspx?p=2162089&seqNum=2>

11. Lapuente, M. J. *Hipertexto: El nuevo concepto de documento en la cultura de la imagen*. Tesis doctoral, Universidad Complutense de Madrid, Dpto. de Biblioteconomía y Documentación.
12. Maldonado, J. G. (2002). *Aplicaciones de la realidad virtual en psicología clínica*. Universidad de Barcelona, Departamento de Personalidad, Evaluación y Tratamientos Psicológicos, Barcelona.
13. Maldonado, J. G. (enero de 2010). La realidad virtual es útil. *L'entrevista* 26.
14. *ms psicólogos*. (s.f.). Recuperado el 27 de agosto de 2017, de terapia de exposición: <http://www.mspsicologos.com/tratamientos-psicologicos/terapia-cognitivo-conductual/terapia-de-exposicion/>
15. Navas, M. Á. (29 de octubre de 2016). *Profesional review*. Recuperado el 29 de agosto de 2017, de Gafas realidad virtual: todo lo que necesitas saber y los mejores modelos: <https://www.profesionalreview.com/2016/10/29/gafas-realidad-virtual/>
16. Ohannessian, K. (29 de septiembre de 2015). *IQ Intel*. Recuperado el 29 de agosto de 2017, de Cómo funciona la realidad virtual: <https://iq.intel.es/como-funciona-la-realidad-virtual/>
17. Ornedo, V. N. (2009). *La realidad virtual al servicio del bienestar social*. Instituto Interuniversitario de Investigación en Bioingeniería y Tecnología Orientada al Ser Humano, Departamento de Comunicaciones.
18. Orozco, S. (s.f.). *santiorozco.com*. Recuperado el 28 de agosto de 2017, de 3D para todos #7. El proceso de texturizado: <http://www.santiorozco.com/2010/01/3d-para-todos-7-el-proceso-de-texturizado/>
19. *Oxford Living Dictionaries*. (s.f.). Recuperado el 27 de agosto de 2017, de <https://es.oxforddictionaries.com/definicion/interaccion>
20. Pascual, J. A. (3 de abril de 2016). *Computerhoy.com*. Recuperado el 29 de agosto de 2017, de Realidad Virtual: 25 preguntas y respuestas que debes conocer: <http://computerhoy.com/noticias/zona-gaming/realidad-virtual-25-preguntas-respuestas-que-debes-conocer-42543>
21. Pastor, J. (18 de marzo de 2016). *Xataka*. Recuperado el 29 de agosto de 2017, de La guerra de la realidad virtual 2016 ya está aquí: comparativa a fondo de todas las

- opciones: <https://www.xataka.com/realidad-virtual-aumentada/la-guerra-de-la-realidad-virtual-2016-ya-esta-aqui-comparativa-a-fondo-de-todas-las-opciones>
22. *Portal de información CAD/CAM/CAE*. (s.f.). Recuperado el 28 de agosto de 2017, de Modelado 3D: <http://www.3dcadportal.com/modelado-3d.html>
 23. RJ, L. (4 de 1994). Virtual therapy of anxiety disorders. *Cyberedge Journal* .
 24. Romero, C. S. (2013). *Aplicación de estrategias didácticas en contextos desfavorecidos*. UNED.
 25. Ruiz, M. F. (2011). *Modelado, texturizado y ajuste de malla*. Universidad Carlos III, Área de Comunicación Audiovisual, Madrid.
 26. Torres, J. A. (2012). *Aplicaciones de la simulación tridimensional para la detección precoz de consumo de sustancias y violencia escolar en ámbitos educativos: desarrollo y validación de una herramienta informática para su detección*. Tesis doctoral, Universidad de Almería, Departamento de Personalidad, Evaluación y Tratamientos Psicológicos, Almería.
 27. *Unity*. (s.f.). Recuperado el 28 de agosto de 2017, de Editor: <https://unity3d.com/es/unity/editor>
 28. *Unity Documentation*. (s.f.). Recuperado el 28 de agosto de 2017, de Ajustes del modo 2D y 3D: <https://docs.unity3d.com/es/current/Manual/2DAnd3DModeSettings.html>
 29. *Unity Documentation*. (s.f.). Recuperado el 28 de agosto de 2017, de Preferencias: <https://docs.unity3d.com/es/current/Manual/Preferences.html>
 30. *Unity Documentation*. (s.f.). Recuperado el 28 de agosto de 2017, de Build Settings: <https://docs.unity3d.com/es/current/Manual/BuildSettings.html>
 31. *Unity Documentation*. (s.f.). Recuperado el 28 de agosto de 2017, de Ajustes del Orden de Ejecución de Scripts: <https://docs.unity3d.com/es/current/Manual/class-ScriptExecution.html>
 32. *Unity Documentation*. (s.f.). Recuperado el 28 de agosto de 2017, de GameObject: <https://docs.unity3d.com/es/current/Manual/class-GameObject.html>
 33. *Unity Documentation*. (s.f.). Recuperado el 28 de agosto de 2017, de Etiquetas(Tags): <https://docs.unity3d.com/es/current/Manual/Tags.html>

34. *Unity Documentation*. (s.f.). Recuperado el 28 de agosto de 2017, de GameObjects Estáticos: <https://docs.unity3d.com/es/current/Manual/StaticObjects.html>
35. *Unity Documentation*. (s.f.). Recuperado el 28 de agosto de 2017, de Prefabs: <https://docs.unity3d.com/es/current/Manual/Prefabs.html>
36. *Unity Documentation*. (s.f.). Recuperado el 29 de agosto de 2017, de Canvas: <https://docs.unity3d.com/es/current/Manual/UICanvas.html>
37. *VR Lens Lab*. (s.f.). Recuperado el 29 de agosto de 2017, de Field of View for Virtual Reality Headsets Explained: <https://vr-lens-lab.com/field-of-view-for-virtual-reality-headsets/>
38. *wiki de Blender*. (s.f.). Recuperado el 28 de agosto de 2017, de Captura del procesamiento: <https://wiki.blender.org/index.php/Doc:ES/2.6/Manual/Render/Bake>
39. *Wikipedia*. (s.f.). Recuperado el 27 de agosto de 2017, de Multimedia interactiva: https://es.wikipedia.org/wiki/Multimedia_interactiva
40. *Wikipedia*. (s.f.). Recuperado el 28 de agosto de 2017, de Modelado 3D: https://es.wikipedia.org/wiki/Modelado_3D
41. *Wikipedia*. (s.f.). Recuperado el 28 de agosto de 2017, de Escáner 3D: https://es.wikipedia.org/wiki/Esc%C3%A1ner_3D
42. *Wikipedia*. (s.f.). Recuperado el 28 de agosto de 2017, de UV mapping: https://en.wikipedia.org/wiki/UV_mapping
43. *Wikipedia*. (s.f.). Recuperado el 28 de agosto de 2017, de Digital sculpting: https://en.wikipedia.org/wiki/Digital_sculpting
44. *Wikipedia*. (s.f.). Recuperado el 28 de agosto de 2017, de Materials system: https://en.wikipedia.org/wiki/Materials_system
45. *Wikipedia*. (s.f.). Recuperado el 28 de agosto de 2017, de Unity (motor de juego): [https://es.wikipedia.org/wiki/Unity_\(motor_de_juego\)#Plataformas_objetivo](https://es.wikipedia.org/wiki/Unity_(motor_de_juego)#Plataformas_objetivo)
46. *Wikipedia*. (s.f.). Recuperado el 29 de agosto de 2017, de Cascos de realidad virtual: https://es.wikipedia.org/wiki/Casco_de_realidad_virtual

Anexo 1: Enlaces de descarga de la aplicación para móviles (sin RV) y ordenador

Enlace para dispositivos móvil Android



Código QR 3. Enlace para descargar la apk de la versión móvil sin realidad virtual.

Fuente: <https://mega.nz/#!s9VFwDbA!bPgJoPjZ35wvIvcyy5T7RR5Ie2f5ETknKKJbVT0Gcyc>

Enlace para ordenadores



Código QR 4. Enlace para descargar un comprimido con la versión de ordenador.

Fuente: <https://mega.nz/#!0w8gEZDa!ZSODtVeqiW3KQbjzJ9jyztKJhkZLV9Gv-ue4d4kLMFs>

Anexo 2: Video y capturas de la aplicación en funcionamiento.

Vídeo



Código QR 5. Enlace al video demostrativo del entorno en ejecución.

Fuente: <https://www.youtube.com/watch?v=7zk56PyQxJs&feature=youtu.be>

Capturas



Ilustración 89. Vista general de la clase generada con los valores por defecto del menú.

Fuente: propia



Ilustración 90. Casillero y estanterías con sus objetos por defecto.

Fuente: propia



Ilustración 91. Pizarras, puerta y papeleras del entorno.

Fuente: propia



Ilustración 92. Otra vista de la clase generada con los valores por defecto del menú.

Fuente: propia

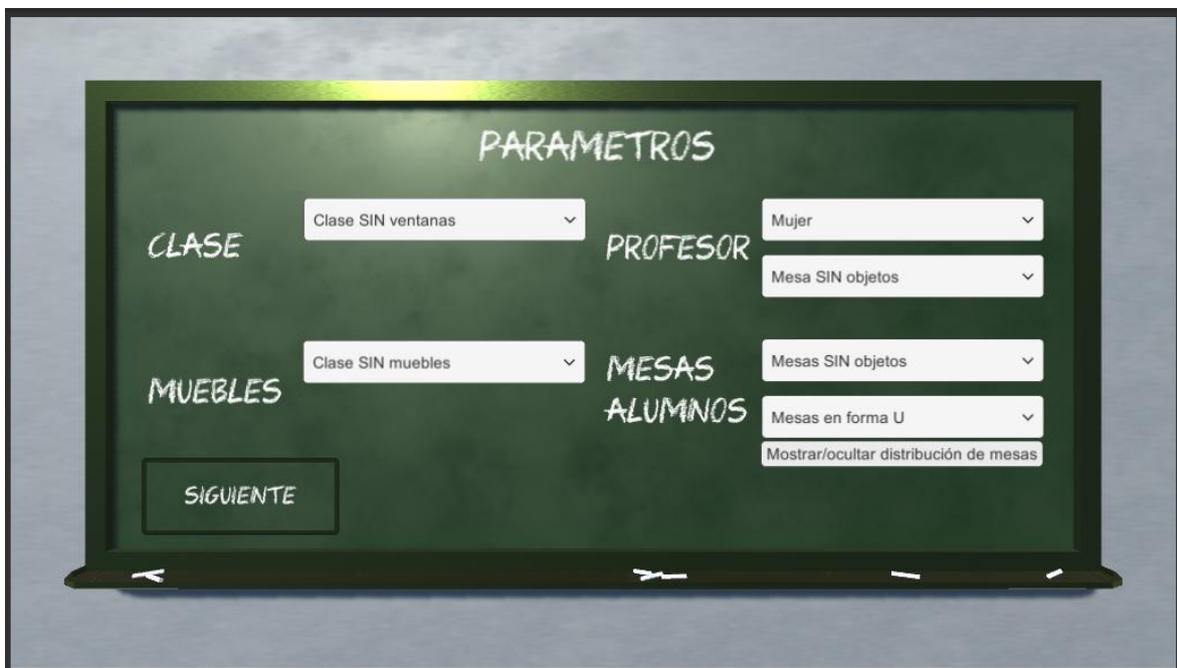


Ilustración 93. Ejemplo 1 de pantalla principal con algunos valores ya seleccionados.

Fuente: propia



Ilustración 94. Ejemplo 2 de pantalla principal con algunos valores ya seleccionados.

Fuente: propia



Ilustración 95. Clase generada en base a los valores seleccionados en los ejemplos anteriores.

Fuente: propia

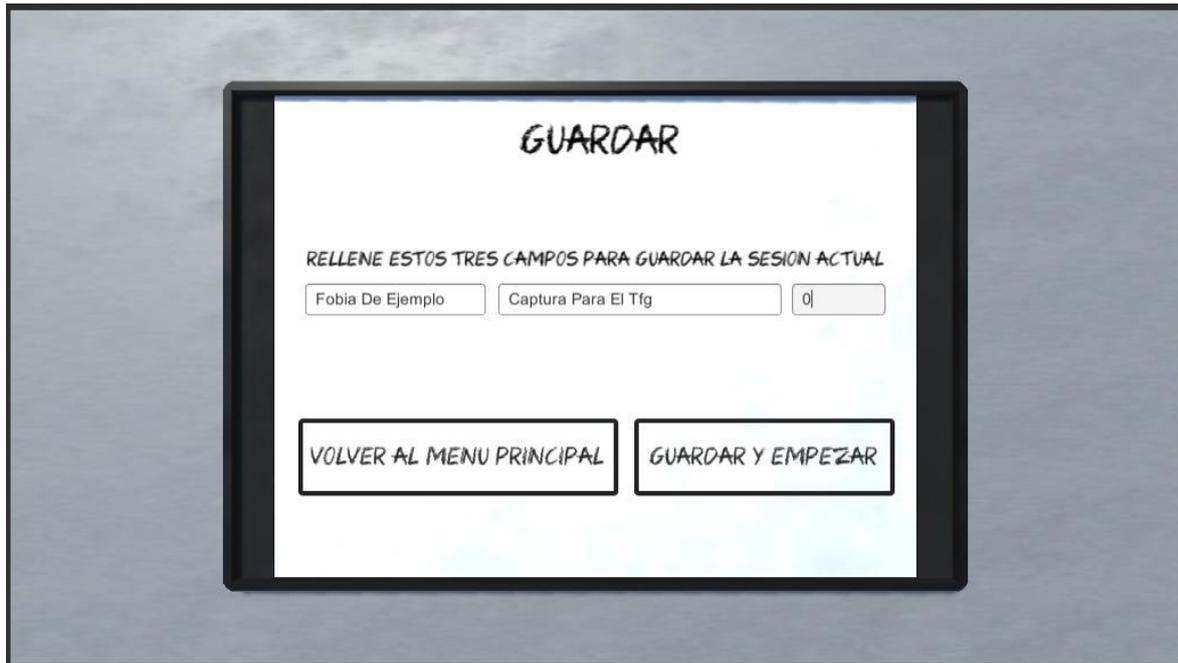


Ilustración 96. Ejemplo de pantalla de guardado.

Fuente: propia



Ilustración 97. Ejemplo de pantalla de carga mostrando la sesión guardada en el ejemplo anterior.

Fuente: propia