



Facultad de Ciencias de la Computación
Benemérita Universidad Autónoma de Puebla

ANÁLISIS DE TÉCNICAS INTERACTIVAS PARA LA NAVEGACIÓN EN 3D

Tesis para obtener el título de:

Ingeniería en Ciencias de la Computación

Presentada por:

Roberto Pérez Hernández

Asesor:

Dr. Juan Manuel González Calleros

Dra. Josefina Guerrero García

Contenido

Resumen.....	1
Capítulo 1. INTRODUCCIÓN	2
1.1 MOTIVACIÓN Y JUSTIFICACIÓN	2
1.2 OBJETIVOS	3
1.2.1 Objetivo general.....	3
1.2.2 Objetivos específicos (Que vas a hacer y para qué lo necesitas hacer)	3
1.3 MARCO CONCEPTUAL	3
1.4 MAPA DE LECTURA DE LA TESIS	4
Capítulo 2. ESTADO DEL ARTE	5
Capítulo 3. METODOLOGÍA.....	8
Capítulo 4. SOLUCIÓN PARA LA NAVEGACIÓN EN ENTORNO 3D.....	9
4.1 INTRODUCCIÓN	9
4.2 Navegación en entornos virtuales.....	9
4.2.1 Sistema base.....	9
4.2.2 Creación de texturas	9
4.3 Mapa	9
4.3.1 Segmentación del Mapa.....	10
4.3.2 Edición de las texturas.....	11
4.4 Configuración en Unity.....	25
4.5 Scripts.....	34
4.6 Texturas.....	40
Capítulo 5. CONCLUSIONES	50
BIBLIOGRAFÍA.....	54

Índice de Ilustraciones

Ilustración 1 Mapa CU	10
Ilustración 2 Photoshop CC	11
Ilustración 3 Instituto de Física	12
Ilustración 4 Primera vista en Photoshop	12
Ilustración 5 Convertir a Capa	13
Ilustración 6 Opciones creación de nueva capa	13
Ilustración 7 Redimensionar lienzo	14
Ilustración 8 Opciones tamaño de lienzo	15
Ilustración 9 Nuevo tamaño	16
Ilustración 10 Nuevo lienzo	16
Ilustración 11 Herramienta de selección rápida en la herramienta Photoshop®	17
Ilustración 12 Herramienta de Borrador	17
Ilustración 13 Herramientas Selección rápida	18
Ilustración 14 Contenido seleccionado	19
Ilustración 15 Contenido eliminado	19
Ilustración 16 Facultad editada	20
Ilustración 17 líneas extras.....	20
Ilustración 18 Herramienta bote de pintura	21
Ilustración 19 Color frontal	22
Ilustración 20 Selector de colores	22
Ilustración 21 Biblioteca de Colores.....	23
Ilustración 22 pantone 7611C	23
Ilustración 23 Edificios pintados.....	24
Ilustración 24 Guardar como.....	24
Ilustración 25 Unity 4.5.5	25
Ilustración 26 Abrir proyecto	26
Ilustración 27 Selección de Proyecto	26
Ilustración 28 Crear Nueva Cámara	27
Ilustración 29 Cámara	28
Ilustración 30 cambio de nombre	28
Ilustración 31 Herramienta Rotar	29
Ilustración 32 Cámara rotada.....	29
Ilustración 33 Herramienta desplazamiento.....	30
Ilustración 34 Cámara desplazada	30
Ilustración 35 Juego iniciado	31
Ilustración 36 Propiedades cámara.....	32
Ilustración 37 Pestaña Game	32
Ilustración 38 Cámara configurada	33
Ilustración 39 Guardar Script	36
Ilustración 40 Importar Script	36
Ilustración 41 Seleccionar Script	37

Ilustración 42 Adjuntar scripts	38
Ilustración 43 Cámara Adjunta.....	38
Ilustración 44 Objetivo fijado en el script	39
Ilustración 45 Scripts añadidos y configurados.....	40
Ilustración 46 Carpeta de texturas.....	41
Ilustración 47 Texturas importadas.....	41
Ilustración 48 Plano.....	42
Ilustración 49 Plano agregado al proyecto.....	42
Ilustración 50 Plano desplazado.....	43
Ilustración 51 Textura añadida al plano	44
Ilustración 52 Textura transparente.....	44
Ilustración 53 Textura ajustada	45
Ilustración 54 Cursor	46
Ilustración 55 Textura Adjunta.....	47
Ilustración 56 Agregar capa.....	48
Ilustración 57 Nueva capa	48
Ilustración 58 Cámara con y sin capa	49

Resumen

Los mapas son una herramienta útil para las personas en diferentes maneras. En un principio los mapas fueron iniciados con el propósito de conocer el mundo, pero hoy en día son una fuente importante de información, más específicamente hablando, se resalta su utilidad cuando se trata de hallar un lugar en particular, o familiarizarse con un entorno para hacer más fácil el recorrido real. Algunos mapas también cuentan con medidas métricas, lo que hace posible tomar medidas de distancia, la cual está relacionada con las medidas del mundo real.

Los mapas han pasados por diferentes etapas en cuanto a su creación, en un principio eran solamente trazos en la arena o la tierra, pero han ido evolucionando hasta llegar a mapas digitales realmente sofisticados.

La Benemérita Universidad Autónoma de Puebla cuenta con diferentes facultades, la mayoría de ellas ubicadas dentro de Ciudad universitaria (CU). En un principio, para las personas que no están familiarizadas con CU, y por primera vez buscan hallar una facultad en particular, o más aun, un edificio en concreto, tienen el problema de que al llegar, no saben hacía donde dirigirse. Al inicio de cuatrimestres, muchos estudiantes pasan por esto, no saben hacia dónde ir, e inclusive cuando saben dónde está la facultad, no saben con exactitud cuál es el edificio que buscan en particular, los que mayormente se ven afectados por esto son los alumnos de nuevo ingreso, si bien algunos conocen parcialmente CU, la mayoría siempre busca como llegar, y en algunas ocasiones, terminan en facultades completamente diferentes.

El proyecto consta de modelos en 3D a escala que representan CU de manera real, no tiene como objetivo únicamente a los estudiantes de la universidad, sino que también, a todas aquellas personas interesadas en conocer más acerca de CU, pudiendo así recorrerla de manera virtual, permitiéndoles hallar sitios de interés, o simplemente caminar entre sus edificios y recorrer las diferentes facultades.

Capítulo 1. INTRODUCCIÓN

1.1 MOTIVACIÓN Y JUSTIFICACIÓN

Wayfinding es definido como el proceso para determinar la estrategia, dirección y curso necesario para alcanzar un destino deseado (Volbracht, 1999). En los mundos virtuales existen problemas con el wayfinding, lo que significa que los usuarios no pueden encontrar su destino fácilmente. Siendo esto un inconveniente mayor de usabilidad que impacta la experiencia de uso en los mundos virtuales, ya que si no pueden encontrar su destino pues simplemente no podrán usar el mundo virtual (Minocha & Hardy, 2011). Cuando los mundos virtuales comenzaron a ser objeto de estudio, uno de los grandes problemas que se citaba sobre el wayfinding eran los asociados al poco realismo o simpleza de los modelos (van Dijk, op den Akker, & Nijholt, 2003). Sin embargo, con el desarrollo de la tecnología estos ya no son problemas, a menos que no se disponga de los recursos humanos o tecnológicos necesarios para hacer el mundo virtual con suficiente calidad para una apariencia realista.

No obstante que se logre crear un realismo en el mundo virtual, normalmente es difícil familiarizarse con el entorno virtual ya que es ajeno a nuestra realidad y no tenemos un modelo de referencia que nos sirva para situarnos en el mundo virtual de manera efectiva. Por ejemplo, imaginemos un campus universitario que visitamos por primera vez y nos piden llegar al edificio de rectoría. Es una tarea difícil ya que no conocemos ni la apariencia del edificio de rectoría, y mucho menos en dónde estamos con respecto al campus universitario, a dónde podemos ir, de dónde venimos, preguntas fundamentales que debe responder el usuario si existen elementos de apoyo al wayfinding.

En este contexto de mundos reales nos situamos con esta investigación. Particularmente, buscando estrategias efectivas de navegación de nuestro campus universitario virtual para apoyar a los visitantes y miembros de nuestra comunidad universitaria a identificar la ubicación de un destino deseado. Y en este escenario nos enfrentamos a tres grandes retos (Arthur & Passini, 1992) que son: i) toma de decisiones, ii) ejecución de la decisión, y iii) procesamiento de la información. Nuestro trabajo se desarrolla en el contexto de escenarios del mundo real con el que no hay familiaridad, partiendo del entendido de que los usuarios del sistema no conocen el campus universitario y requieren ayuda para llegar a su destino. Este tipo de escenario requiere de brindar ayuda al usuario en todo momento que sea requerido (Minocha & Hardy, 2011) es decir: i) para la toma de decisiones, ayudar al usuario a entender el mundo completo, cómo está organizado y dividido, identificar dónde están y a dónde quieren ir; ii) ejecución de la decisión, guía continua acompañándolo a su destino; iii) procesamiento de la información, hacer evidente que han llegado a su destino. Esta herramienta permitirá conocer el campus sin necesidad de estar recorriéndolo por completo, podrán los usuarios buscar algún edificio en particular sin necesidad de andar buscando presencialmente y llegar de una manera más directa, o simplemente conocer el campus.

Si bien tradicionalmente se han diseñado muchas técnicas de interacción para ayudar a navegar de manera adecuada los mundos virtuales, como un trazado de rutas o algún GPS, muchas veces no pueden ser adaptados a cualquier entorno 3D sin un objetivo específico. Nos centraremos en el uso de la metáfora de mini mapa como una herramienta de ayuda en la navegación de mundos

virtuales. Las personas que desean conocer ciudad universitaria BUAP son los que mayor beneficio tendrían, ya que permitiría desplazarse sabiendo donde se encuentran en todo momento. Este trabajo puede tener otros usos, como son la simulación de actividades en el campus, es decir, probar el funcionamiento de algún otro proyecto, simulándolo dentro del entorno virtual, como se comportaría o reaccionaría bajo ciertas circunstancias, el vuelo de un dron, la navegación de una silla de ruedas.

1.2 OBJETIVOS

1.2.1 Objetivo general

La finalidad es obtener un mapa 3D de un campus universitario que permita realizar búsquedas, así como permitir a los usuarios recorrer la universidad libremente.

Esta tesis propone una serie de modelos que representan un campus universitario, la Benemérita universidad Autónoma de Puebla, con un sistema de navegación mediante un mundo en miniatura añadido al sistema de modelos, con el propósito de dar una idea de en donde se encuentra y lo que está a su alrededor.

1.2.2 Objetivos específicos (Que vas a hacer y para qué lo necesitas hacer)

- Diseño del sistema de navegación 3D para recorridos virtuales
 - Identificación de técnicas de visualización de mapas para determinar la más adecuada a ser usada en esta tesis.
- Creación de texturas para la visualización del mundo en miniatura.
- Configuración del mundo en miniatura dentro del ambiente de motor de videojuegos.
- Contribuir con una herramienta de navegación, no solo útil para los estudiantes y profesores, sino que también para cualquier usuario interesado en conocer el campus universitario.
- Validar el uso del sistema de navegación con usuarios reales para garantizar que nuestra aplicación es adecuada.

1.3 MARCO CONCEPTUAL

Realidad virtual

La realidad virtual es un entorno de escenas y/u objetos que emulan la realidad, generado mediante tecnología informática, que crea en el usuario la sensación de estar inmerso en él y así mismo de poder explorar dicho entorno.

Navegación 3d

La navegación en 3d es una herramienta que permite al usuario explorar su entorno, mediante objetos conocidos como cámaras, las cuales permiten visualizar el entorno de la realidad virtual.

Recorridos Virtuales

Los recorridos virtuales son una manera innovadora, interactiva y fácil de conocer algún lugar o espacio en particular, ya que permite desplazarse y recorrer el entorno.

1.4 MAPA DE LECTURA DE LA TESIS

El Capítulo 2 ESTADO DEL ARTE define que son los mundos en miniatura. Posteriormente se analizan los métodos de navegación mediante la interacción con los mundos en miniatura. En base en lo anterior se proponen los criterios necesarios para el caso específico del desarrollo de un mundo virtual, a su vez, adjunto con un mundo en miniatura con enfoque a la navegación con un entorno virtual.

El Capítulo 3 METODOLOGÍA se expone la metodología de desarrollo. Se explica método para la creación de las texturas utilizadas en el mundo en miniatura, así como la configuración en unity para la vista en miniatura. Esta metodología permitirá obtener una vista de mundo en miniatura en Unity.

Posteriormente el Capítulo 4 SOLUCIÓN PARA presenta la solución propuesta. La solución es dividida en el mundo virtual y la representación del mundo en miniatura. Se especifican las etapas de desarrollo basadas en la metodología del Capítulo 3, desde la especificación del problema, la identificación de tareas, el modelado de tareas y su transformación a interfaces abstractas y concretas.

En el 0

CONCLUSIONES se presentan los avances obtenidos de la solución y se analizan aspectos de usabilidad y diseño; finalmente, se presenta el trabajo futuro.

Capítulo 2. ESTADO DEL ARTE

Hoy en día la navegación por medio de mapas se ha vuelto algo cotidiano para muchas personas, aunque no acaben de comprender los mapas por ser complicados o no saber leerlos adecuadamente. Una opción para facilitar la navegación son los mapas en 3D.

Los mapas en 3D permiten una visualización más detallada, al poder visualizar el entorno de manera cercana a la realidad, haciendo más fácil su comprensión.

Existen diferentes navegadores en 3D, el más conocido es Google Earth que permite desplazarse por todo el mundo y ver casi cualquier parte. Además existen unos cuantos más como Flyover que permiten la visualización en 3D para mirar el entorno de algún punto en específico, si bien su calidad es ligeramente superior a la de Google, no es tan conocido.

Así como existen diferentes navegadores, hay diferentes técnicas de navegación, cada una con su propio atractivo, aunque algunas de ellas aún están en desarrollo ya se pueden aplicar a algunos entornos.

Algunas herramientas son:

- La caminadora omnidireccional (Omnidirectional treadmill) es un dispositivo que permite desplazarse en cualquier dirección, es una de las herramientas para realidad virtual.
- La mano virtual (Virtual Hand), la cual imita la interacción en el mundo real, como si se estuviera interactuando directamente con el objeto.
- Gestos manuales es una herramienta que permite la navegación mediante gestos para desplazarse dentro del entorno.
- Mundos en miniatura, los cuales pueden ser aplicados dentro de un mundo virtual existente para funcionar como un minimapa, permitiendo observar su posición actual y su entorno.

Además de pensar en las técnicas de navegación hay que escoger tanto las herramientas de modelado y el motor de juegos que mejor se adapte al objetivo para su desarrollo.

Herramientas de modelado

El modelado es parte fundamental de los mundos en 3D, ya que sin esto sería un mundo vacío. A grandes rasgos, cuando hablamos de diseño 3D nos referimos tanto a la creación tridimensional de piezas, objetos, personajes o estructuras, empleado generalmente en ingeniería y arquitectura, o a la generación de imágenes en 3D relacionadas con el mundo multimedia y la animación 3D.

Para el modelado en 3D existen diferentes herramientas como por ejemplo Google SketchUp, Blender, 3D Max, Maya, etc. Cada software tiene sus ventajas y desventajas frente a los demás, pero la posibilidad de realizar un trabajo de calidad no depende de esto, sino de los conocimientos, la creatividad, y no tanto del software.

Google SketchUp

La mayoría de programas de modelado 3D requieren al menos una base de conocimientos de dibujo. Sin embargo, Google SketchUp está diseñado para que cualquier persona pueda usarlo ya que es una herramienta muy intuitiva, sobretodo en comparación con otros programas de dibujo 3D, además su versión básica es gratuita. Sin embargo, como todo software tiene desventajas, la mayor es que es limitado en comparación con otras herramientas de modelado, ya que carece soporte técnico y no tiene tampoco algunas herramientas 3D específicas de modelado, ni genera informes. Tampoco puede exportar a 3D Studio Max o Autocad.

Blender

Blender no solo es una herramienta de modelado, diseño y animación de gráficos 3D, sino que también incluye, entre otras cosas, un potente software para la elaboración de juegos y un editor de audio y vídeo. Una de las mayores ventajas de Blender es que se trata de software libre y multiplataforma.

Cuenta con diferentes características útiles para el modelado, como es la capacidad para una gran variedad de primitivas geométricas, edición de audio y sincronización de video, cuenta con un motor de juegos 3D integrado, permite el uso del lenguaje Python para automatizar o controlar varias tareas, además ofrece también herramientas de simulación avanzada tales como dinámicas de cuerpos rígidos, de cuerpos suaves y de fluidos, herramientas de animación de personajes, un sistema de simulación física avanzado, y un sistema de composición de materiales. Su mayor desventaja es su interfaz poco intuitiva y curva de aprendizaje muy lenta al principio.

3D Studio Max

3D Studio Max es uno de los programas de modelado y animación 3D más utilizados por los desarrolladores de videojuegos, en proyectos de animación como películas, anuncios de televisión o efectos especiales y en arquitectura.

Sus principales características son que cuenta con más de 100 herramientas de modelado, escultura y manipulación de objetos, también existen infinidad de plugins, muchos de ellos de pago, que facilitan la creación de nuestros proyectos, es compatible con multitud de formatos, permite la integración con Adobe Photoshop y podemos importar los archivos PSD como texturas. Es una herramienta potente y estable. En cuanto a desventajas destacan su alto precio, sus requisitos de hardware elevados y que no es multiplataforma (solo disponible para Windows).

Maya

Maya se caracteriza por su potencia y las posibilidades de expansión y personalización de su interfaz y herramientas. Ésta enfocado principalmente a desarrollo de juegos y películas, animación en general y creación de efectos.

La característica más importante de Maya es lo abierto que es al software de terceros, el cual puede cambiar completamente la apariencia de Maya. El mismo software se puede transformar debido a sus opciones altamente personalizables. Aparte de su potencia y flexibilidad, este aspecto solamente hizo que Maya fuera muy interesante para los grandes estudios que tienden a escribir mucho código personalizado para su producción. Como contras posee una interfaz poco intuitiva y hay software más estables que Maya.

Motores de Video juegos

Para crear los mundos en 3D no basta con el modelado, hay que tener donde integrar todos los componentes desarrollados, es decir un motor de videojuegos. Su función básica es servir de motor de renderizado ya sea 2D o 3D. La mayoría contienen un motor de físicas, un detector de colisiones, sonidos, animación, Scripting entre otras características. En general, es una herramienta que facilita la construcción de niveles y mecánicas de Juego, mediante la importación de Assets (objetos externos) como sonidos, animaciones, modelos y gráficos.

Unity 3D

Posiblemente uno de los motores gráficos más conocidos a día de hoy. Robusto, fácil de usar, potente, versátil tanto para un artista como para un programador, compatible con un montón de plataformas, innovador en el modo que afronta el desarrollo de un videojuego y sobre todo una gran comunidad de usuarios .

La mejor característica de Unity son su estabilidad y robustez, además de contar con versión gratuita que ofrece casi todo lo necesario para un gran desarrollo. La mayor desventaja es que para acceder a todas sus características avanzadas necesitas pagar por la licencia Pro.

Shiva 3D

Definido como el motor para videojuegos más compatible del mundo, es capaz de soportar desarrollos para Windows, MacOs, Linux, iPhone, Android, BlackBerry, Palm, Wii y iPad así que va servido de plataformas. Si bien no es del todo intuitivo al principio, al acostumbrarse a su uso parece ser muy cómoda y versátil. La mayor desventaja que posee es que su comunidad no es tan grande, así que no es fácil encontrar información de algún objetivo en particular.

Unreal Engine 4

Es un conjunto completo de herramientas de desarrollo de videojuegos hecho por desarrolladores, para desarrolladores. Tiene unas capacidades gráficas impresionantes, incluida la iluminación dinámica y un sistema de partículas que permite manejar un millón de partículas en una misma escena. Un sueño hecho realidad para los artistas 3D. Además de todo, es gratuito.

Como desventaja es que tiene requisitos altos y hay que compilar constantemente para notar los cambios hechos en el juego.

Capítulo 3. METODOLOGÍA

El desarrollo del mundo en miniatura puede ser hecho de diferentes formas, dependiendo del objetivo final. El entorno sobre el que se presenta la solución es un entorno en 3D de CU, que ya es en sí un mundo en miniatura, un mundo en miniatura que contiene otro.

La metodología que se propone para desarrollar un mundo en miniatura para la navegación en un ambiente 3D contiene los siguientes componentes:

- 1) Mapa. Un mapa de la realidad a representar, el cual brindará la base para construir el entorno virtual, ya que el objetivo principal es mostrar lo que existe en la actualidad entre más reciente sea, mayor será la fidelidad al mundo real.
- 2) Modelos 3D. Los modelos 3D representan el mundo virtual basado en el entorno real sobre el cual se desea navegar.
- 3) Texturas. Las texturas, para este caso específico, forman la parte principal del mundo en miniatura, ya que estas brindan la perspectiva que facilitará la navegación en el entorno 3D.
- 4) Software. El software está implicado en la creación de las texturas, así como también en la creación de los modelos 3D y el entorno virtual.

Capítulo 4. SOLUCIÓN PARA LA NAVEGACIÓN EN ENTORNO 3D

4.1 INTRODUCCIÓN

Este capítulo presenta la solución propuesta para navegación en el entorno virtual de CU. La solución consta de un mundo en miniatura que sirve como mapa para la asistencia en la orientación de la ubicación en el entorno virtual de ciudad universitaria. Se especifica el desarrollo de la investigación y de cada componente.

4.2 Navegación en entornos virtuales

Este trabajo propone un mundo en miniatura que funciona como un mapa para ayudar a la navegación dentro del entorno virtual en 3D. Además se investigaron otras opciones las cuales también se presenta como una referencia, que bien podrían ser añadidas al sistema más adelante.

4.2.1 Sistema base

El sistema de navegación

Una vez dentro del sistema se podrá recorrer el entorno virtual de CU de manera libre, pudiendo activar o desactivar el mapa en el momento que se desee.

4.2.2 Creación de texturas

Las texturas representan una parte fundamental de este mundo en miniatura, estas son las responsables de mostrar la información de interés, es decir la ubicación donde se encuentra el usuario y lo que le rodea.

4.3 Mapa

Todas las texturas, en conjunto, son partes de un mapa general que abarca todo CU (Ilustración 1).

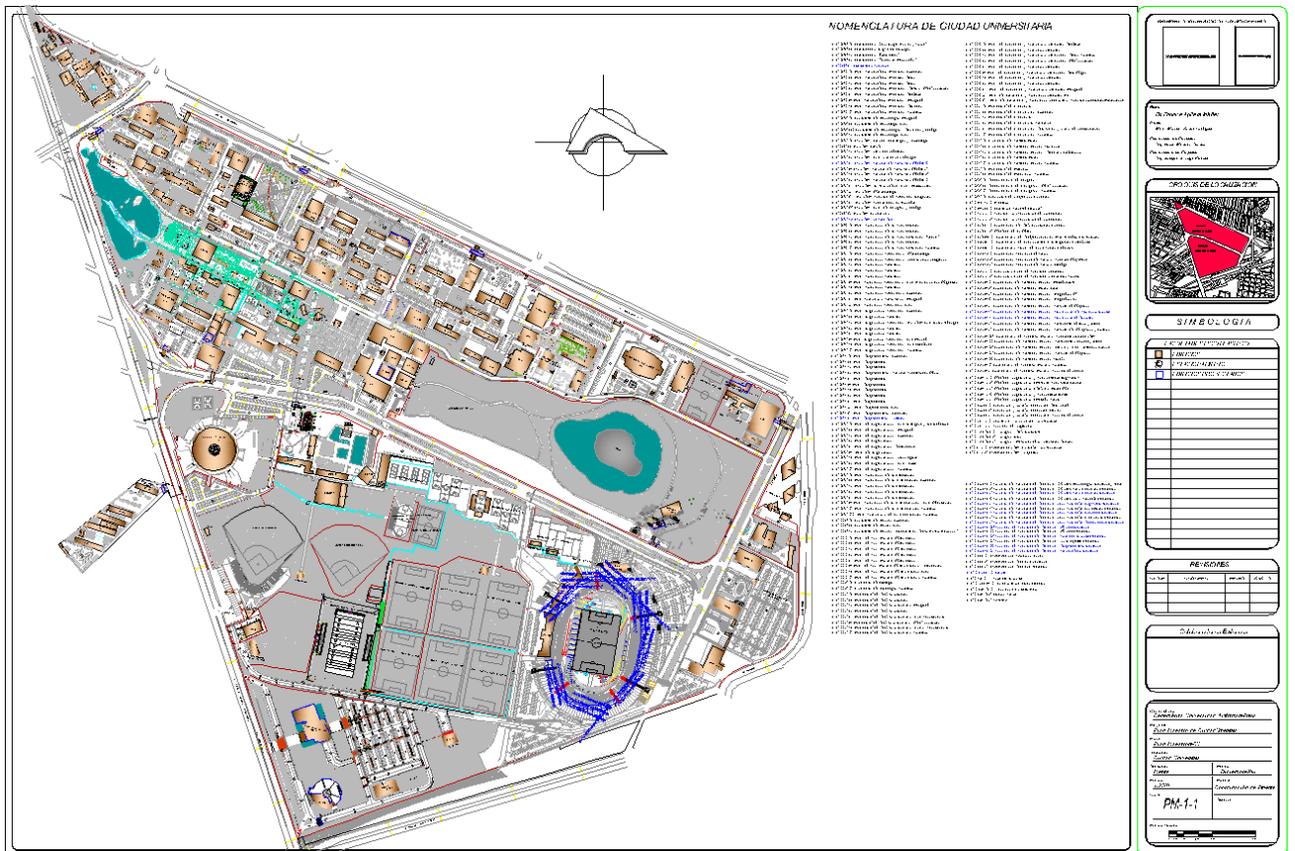


Ilustración 1 Mapa CU

El mapa general de CU es demasiado grande como para usarlo por sí mismo como una textura, si bien es posible crearla como tal, la calidad no sería la apropiada.

4.3.1 Segmentación del Mapa

El mapa está en formato .dwg, el cuál es un formato de archivo informático de dibujo computarizado, utilizado principalmente por el programa AutoCAD.

Como tal el programa utilizado para poder pasar el mapa a versiones más pequeñas para que puedas ser apreciada, sin perder la calidad, es AutoCAD 2015.

En un principio, al abrir el archivo en el programa nos muestra el dibujo de CU, el cual tiene un tamaño realmente grande, es posible simplemente tomar capturas de pantalla, sin embargo esto provocaría que perdiera calidad.

Para obtener las texturas es necesario segmentar el mapa en imágenes más pequeñas, sin perder la calidad para que se pueda apreciar desde el mundo en miniatura. En este caso se desea, no solo segmentar el mapa, sino que también se desea exportar hacia un formato diferente, ya que el formato .dwg no es reconocido por la plataforma de unity.

En este caso se eligió como formato para las texturas el .PNG ya que permite la transparencia, lo cual nos es de gran utilidad, porque así podemos mostrar únicamente lo que nos interesa sin ocultar ninguna parte del ambiente 3D.

AutoCAD no cuenta como tal con una función de exportación hacia PNG por lo que hay que aplicar un proceso un tanto diferente para exportar.

4.3.2 Edición de las texturas

Con las facultades segmentadas en PNG se procede a editarlas para tener las texturas requeridas para unity, esto sin alterar el formato.

Para la edición se utiliza Photoshop, en este caso Adobe Photoshop CC (Ilustración 2)

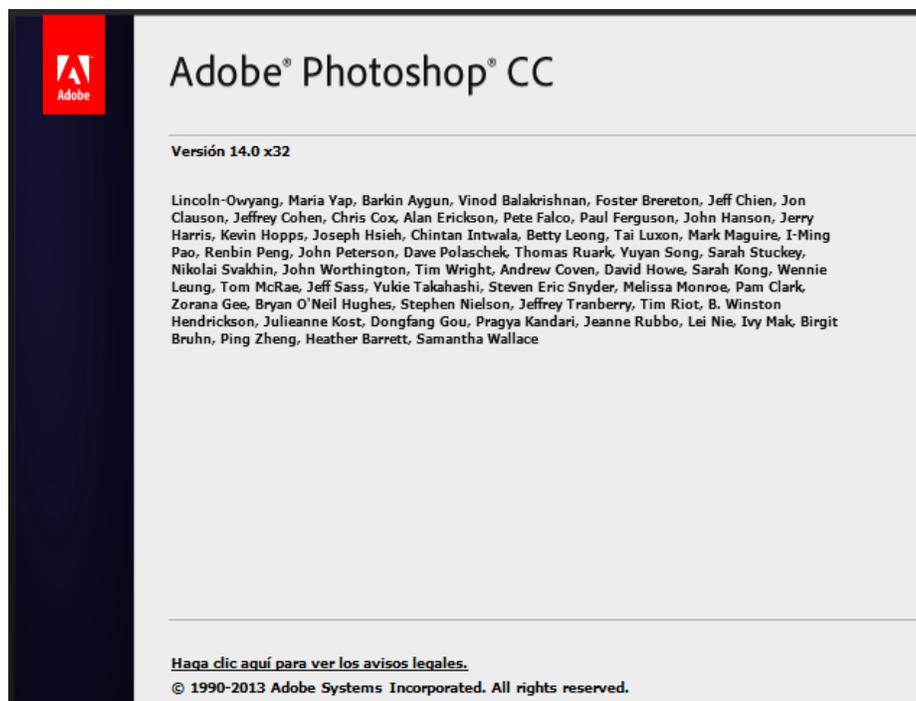


Ilustración 2 Photoshop CC

Primero que nada se debe seleccionar el archivo a editar, en este caso se tomara como ejemplo el Instituto de Física (Ilustración 3).

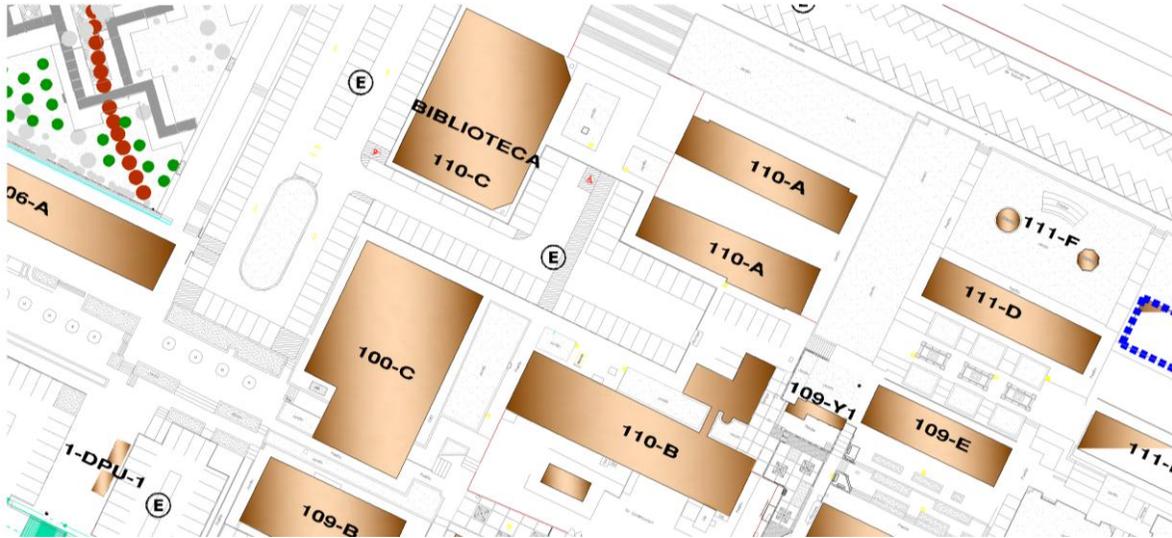


Ilustración 3 Instituto de Física

Una vez abierto en photoshop podremos apreciar que la imagen es demasiado grande (Ilustración 4), esto se debe a la exportación por default de AutoCAD, sin embargo es lo recomendable para no perder calidad, aun cuando parece que hay demasiado espacio en blanco la imagen es realmente grande en sí.

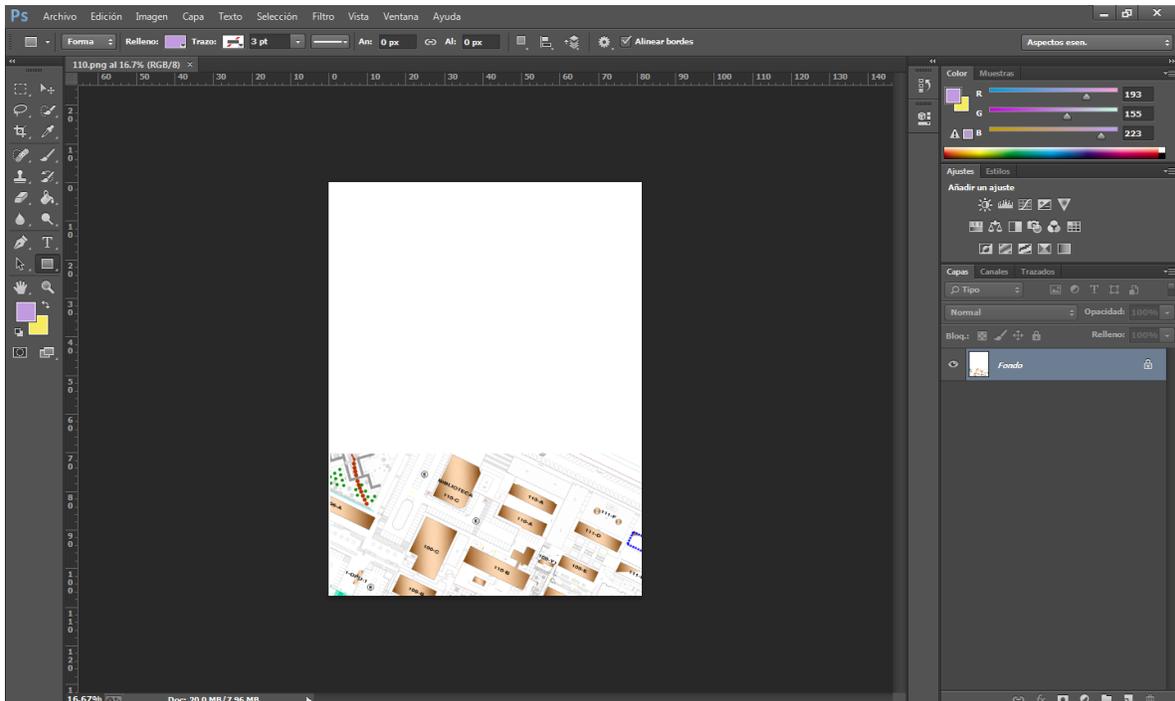


Ilustración 4 Primera vista en Photoshop

Lo primero es convertir la imagen en una capa editable. Para esto, de lado derecho se encuentra la pestaña de capa, en esa misma se pueden observar las capas con las que cuenta el archivo en ese momento, se da click derecho sobre la capa de "Fondo" y aparecerán una serie de opciones, de las cuales seleccionamos capa a partir de fondo (Ilustración 5), esto convertirá la capa de Fondo en una capa editable.

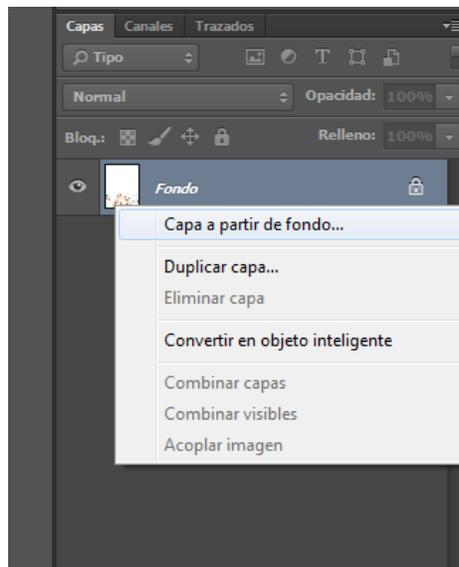


Ilustración 5 Convertir a Capa

Una vez hecho esto aparecerá una ventana emergente pidiendo dar el nombre a la capa y otras opciones (Ilustración 6), se dejan todas por default y el nombre de la capa se puede elegir al gusto, en este caso se dejara por default en capa 0.

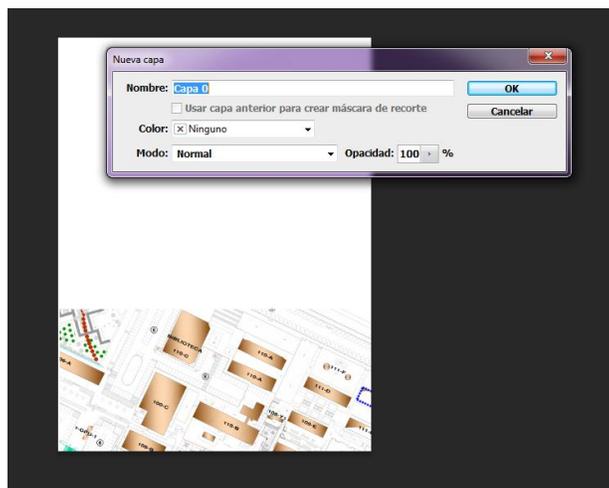


Ilustración 6 Opciones creación de nueva capa

Ahora con la nueva capa creada, hay que redimensionar el lienzo para quitar el sobrante, es decir, eliminar la parte blanca que no contiene nada de interés para la textura. Para esto hay que

redimensionar el lienzo. Esta opción se encuentra en el menú imagen -> Tamaño de lienzo (Ilustración 7).

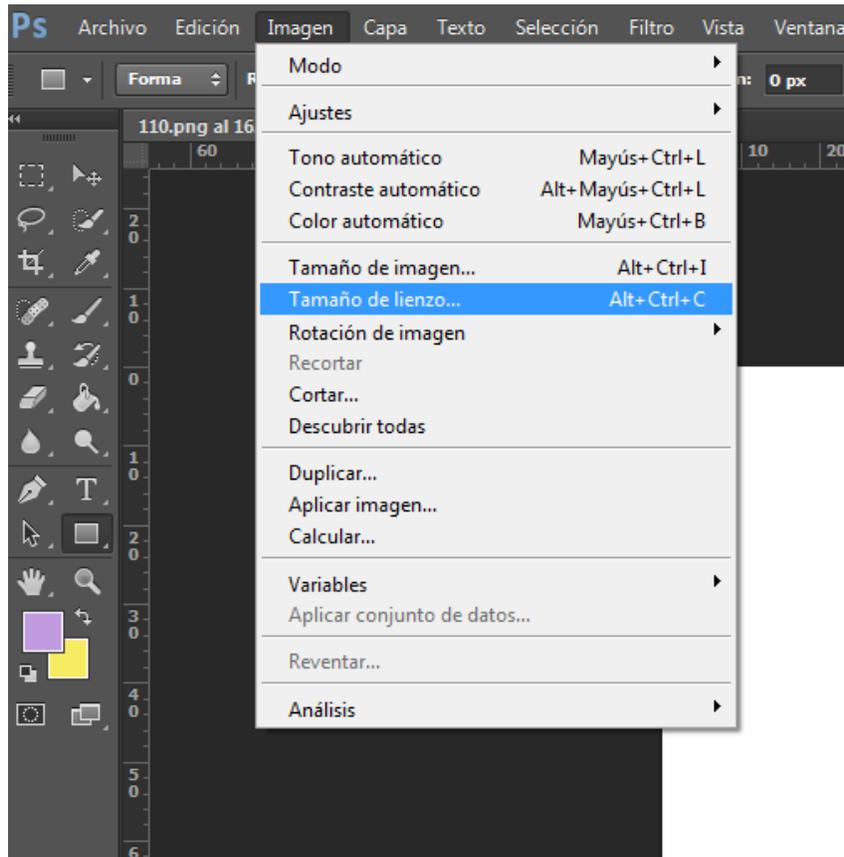


Ilustración 7 Redimensionar lienzo

Es preciso distinguir entre cambiar el tamaño de la imagen y cambiar el tamaño del lienzo.

El lienzo es el espacio de trabajo que proporciona el archivo de Photoshop. Este espacio de trabajo puede estar o no totalmente ocupados por la imagen. Al aumentar el tamaño de la imagen por supuesto que se aumentara el tamaño del lienzo. Sin embargo, no ocurre así a la inversa. Cuando se aumenta el tamaño del lienzo se aumenta el área de trabajo pero no la imagen. Por lo tanto, el resultado que se obtiene es un área de trabajo vacía alrededor de la imagen.

Una vez seleccionada la opción aparecerá una ventana emergente con las opciones para el tamaño de lienzo (Ilustración 8). En esta ventana te presenta el tamaño actual de la imagen, así mismo cuenta con las opciones para el tamaño nuevo. Tanto la anchura como la altura están en centímetros, pero se pueden cambiar por porcentaje, pixeles, pulgadas, entre otras. La medida se mantendrá en centímetros

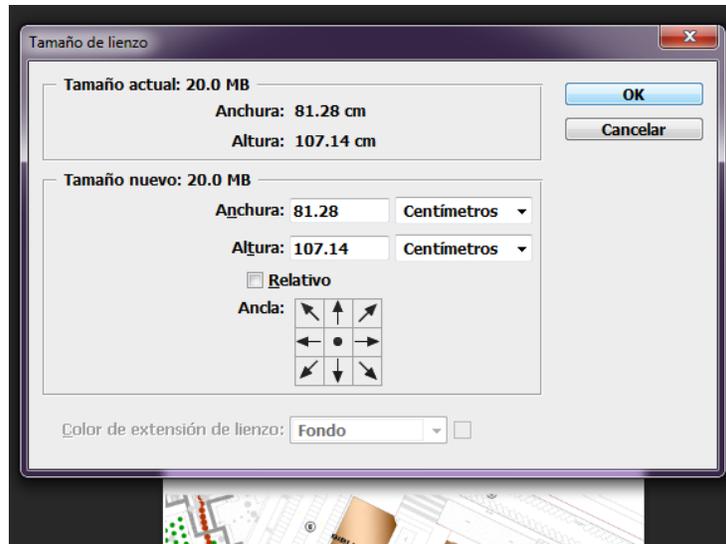


Ilustración 8 Opciones tamaño de lienzo

Para este caso específicamente, se reducirá la altura en 70, dando como resultado 37.14, la anchura no es necesario modificarla. También se debe tomar en cuenta la opción de ancla, ya que de otra manera la imagen se perderá completa o parcialmente, dependiendo de cómo se maneje (Ilustración 9).

El ancla permite indicar la posición de la imagen en el nuevo lienzo. Por ejemplo, si queremos que la imagen aparezca junto a la esquina superior del nuevo lienzo, se debe seleccionar el cuadro de la parte superior izquierda. Para este caso queremos que la imagen se conserve entera sin tener partes del lienzo en blanco, por lo que tendremos que situar el ancla en el cuadro central inferior, ya que de otra manera la imagen se perdería.

Una vez hecho esto, presionamos ok. Con esto aparecerá una ventana emergente diciendo que el tamaño de lienzo es más pequeño que el actual y que se recortará la imagen, simplemente presionamos continuar.

La imagen no se verá afectada debido al ancla que establecimos previamente.

Ahora ya tendremos la imagen con el contenido que nos interesa (Ilustración 10), sin embargo aún hay varios cambios que hacerle antes que podamos usarla como textura en unity.

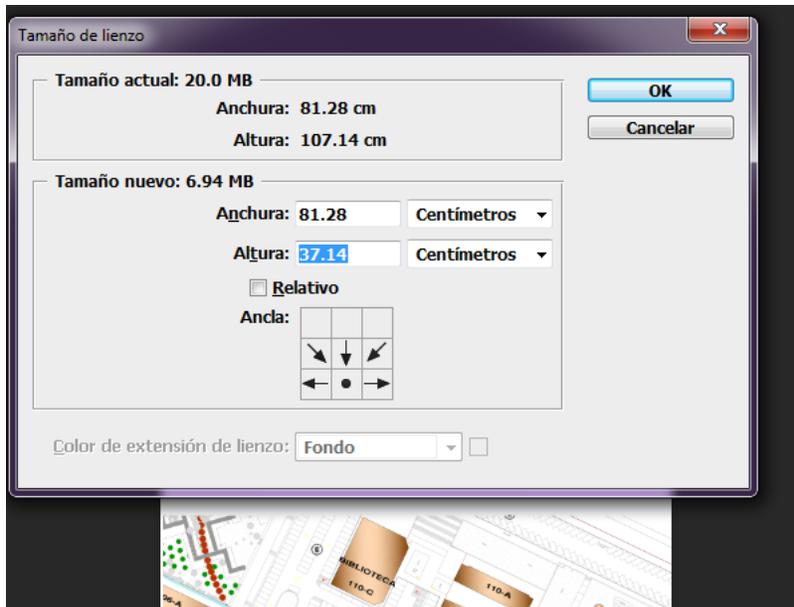


Ilustración 9 Nuevo tamaño

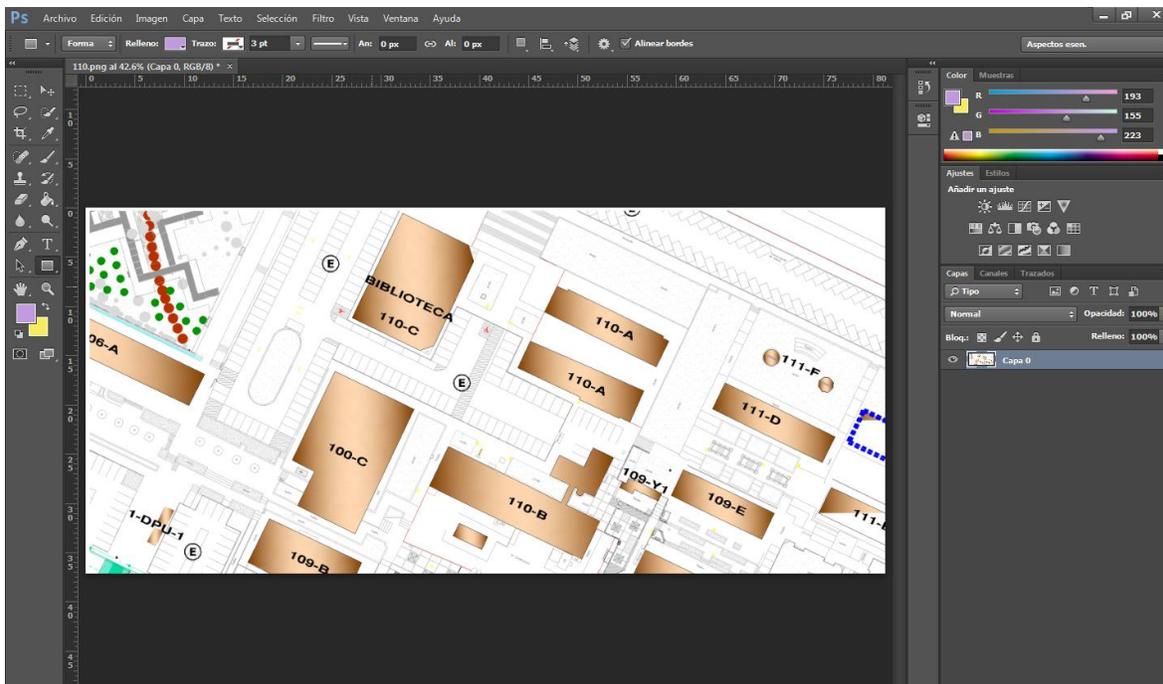


Ilustración 10 Nuevo lienzo

Ahora que contamos con el nuevo lienzo tenemos que eliminar los sobrantes y dejar las partes que nos interesa, es decir dejar los edificios 110 solamente y eliminar el resto.

Existen varias maneras para eliminar el resto, pero en este caso usaremos dos herramientas en concreto, la herramienta de selección rápida (Ilustración 11) y el borrador (Ilustración 12).

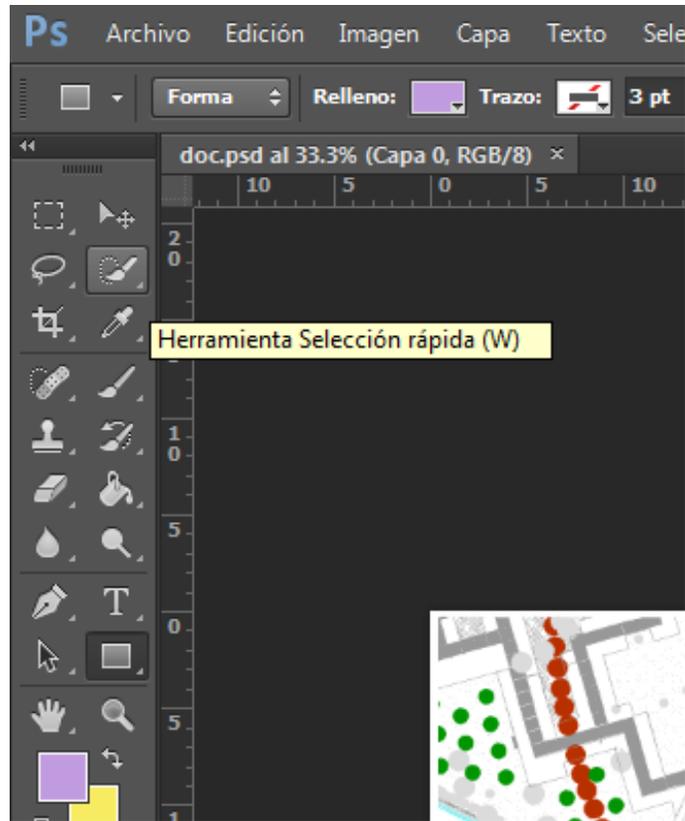


Ilustración 11 Herramienta de selección rápida en la herramienta Photoshop®

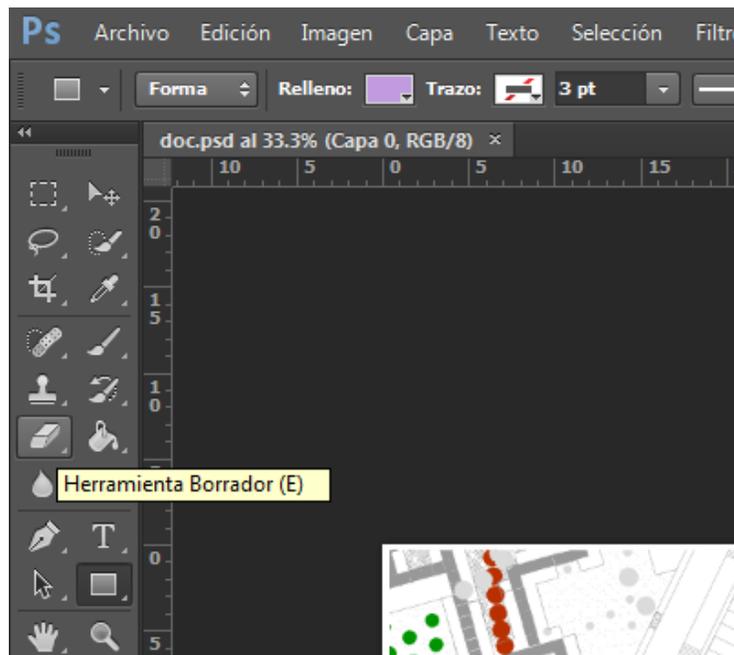


Ilustración 12 Herramienta de Borrador

La herramienta de Selección Rápida de Photoshop es una de las más interesantes. Con ella podemos realizar selecciones complejas en Photoshop de forma cómoda y sencilla. Usar la herramienta de Selección rápida es similar a pintar, pero el resultado final es una selección.

Para acceder a ella debemos hacer click sobre su icono en la barra de herramientas. La selección rápida se encuentra en el mismo conjunto que la varita mágica. Su atajo de teclado es la letra **W**.

La herramienta de selección rápida tiene una barra de opciones, como casi todas las herramientas de Photoshop (Ilustración 13).

1. Botones de operación: Con el primero definimos una nueva selección, con el segundo añade lo seleccionado a una selección ya existente y si activamos el tercero restaremos la nueva selección a lo que ya teníamos.

2. Pincel El pincel determinará el comportamiento de la herramienta de selección rápida. Su tamaño y forma serán fundamentales para aprovechar todas las posibilidades.

3. Muestrear todas las capas: Si tildamos esta opción se seleccionará lo elegido en todas las capas.

4. Mejorar automáticamente: Suaviza los bordes y mejora la precisión del marco.

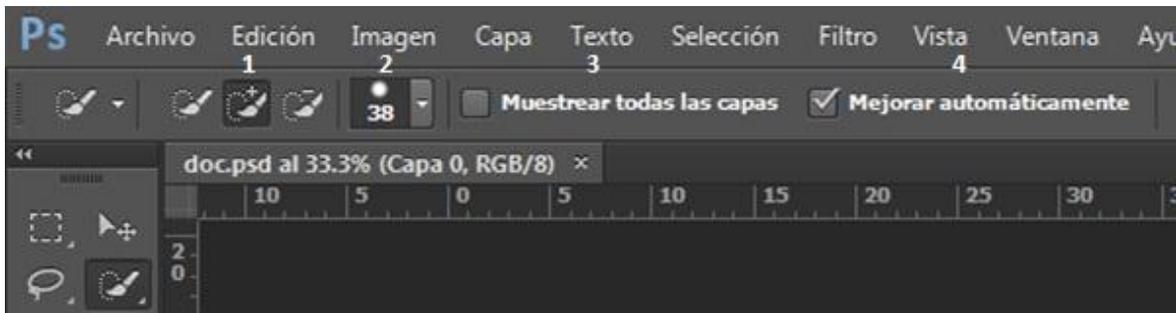


Ilustración 13 Herramientas Selección rápida

Una vez que tenemos seleccionada la herramienta de selección rápida y ya hemos explicado un poco su función y herramientas que posee (es recomendable tener seleccionada la opción de mejorar automáticamente para una mayor precisión) procedemos a seleccionar el contenido alrededor de los edificios para eliminarlo (Ilustración 14), una vez seleccionado presionaremos la tecla “supr” (suprimir), esto eliminara el contenido seleccionado (Ilustración 15). Cabe mencionar que si no se convierte el fondo a capa como se mencionó antes, no será posible eliminar el contenido seleccionado.

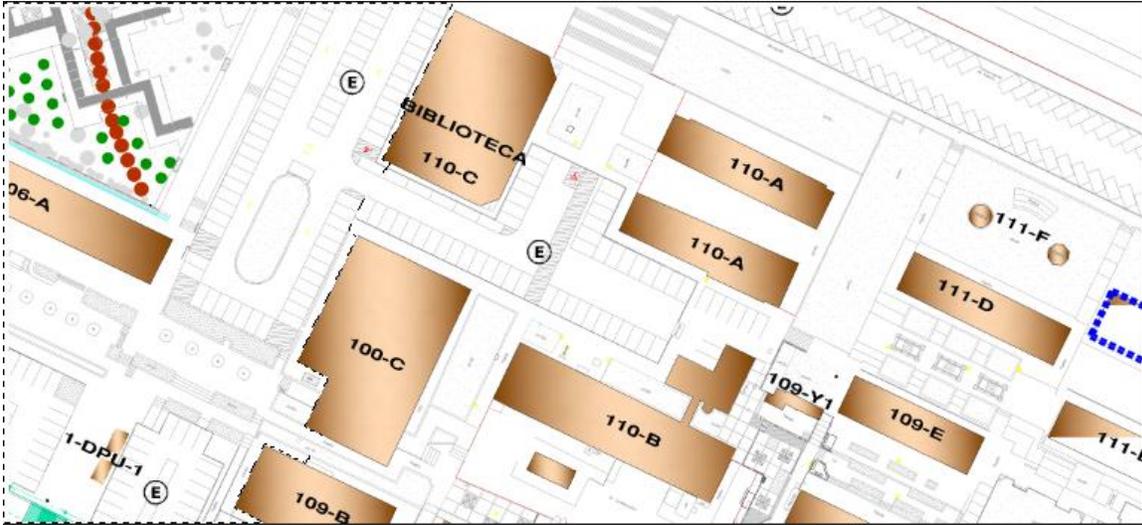


Ilustración 14 Contenido seleccionado

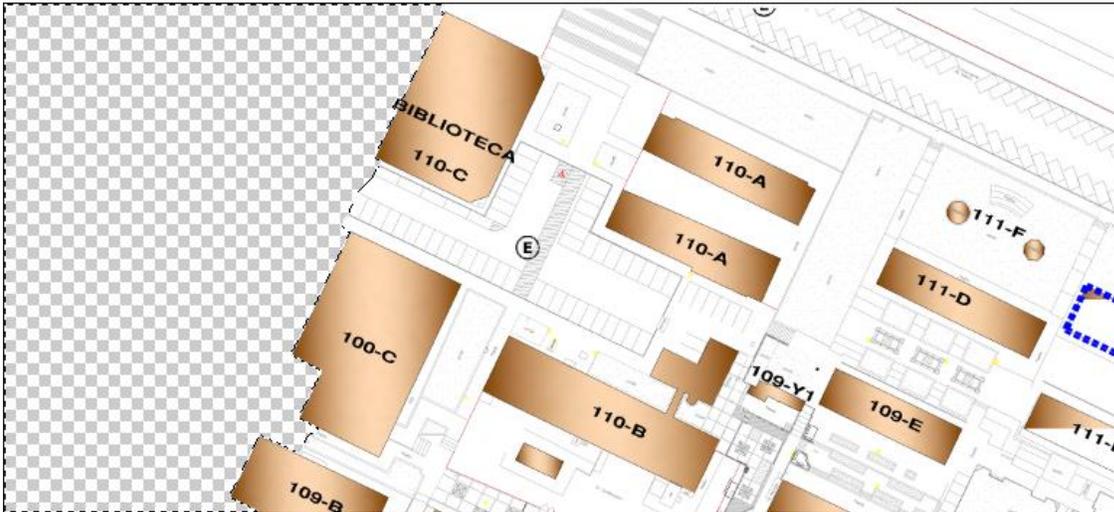


Ilustración 15 Contenido eliminado

Se procederá de la misma manera con el resto de la imagen, siempre teniendo cuidado de no seleccionar por error ninguna porción de los edificio. Hay que tener paciencia con esto, ya que muchas veces, aun cuando pasamos cerca de un edificio, este puede seleccionarse, lo más prudente sería ir por pequeñas secciones hasta haber quitado lo sobrante por completo (ilustración 16).

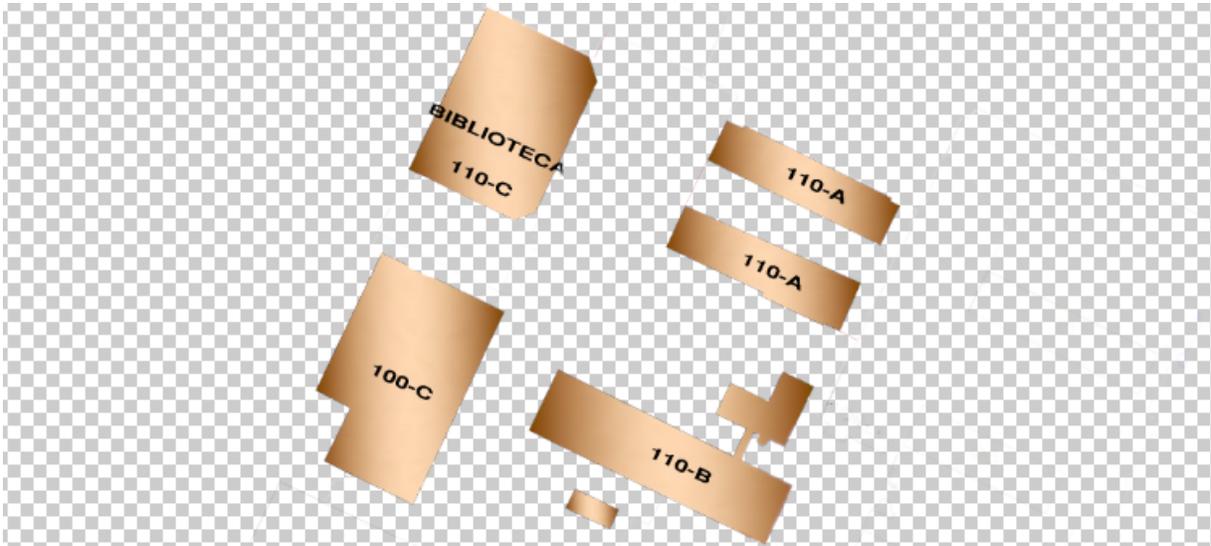


Ilustración 16 Facultad editada

A pesar de que parezca que todo el fondo ha sido eliminado, usualmente quedan ciertas líneas que no se pueden apreciar a menos que se agrande la imagen, y aun así hay ocasiones en que no se notan hasta que ya están como textura en unity (Ilustración 17).

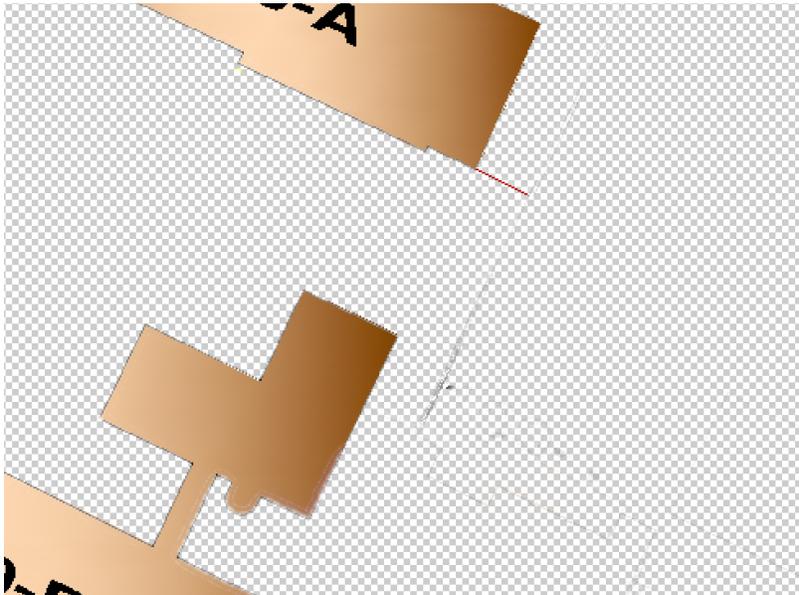


Ilustración 17 líneas extras

Aquí es donde entra la herramienta de borrador, basta con seleccionar la herramienta y pasar por encima de las líneas con el botón izquierdo del mouse presionado para eliminarlas, se puede ajustar el tamaño de del borrador para una mejor precisión. Es recomendable ampliar el zoom hasta un 200% para apreciar mejor los sobrantes.

Una vez que ya tenemos los edificios solamente queda el pintar los edificios de algún color para identificar la facultad a la que pertenecen, cabe mencionar que cada facultad tendrá un color diferente. Utilizaremos la herramienta de bote de pintura (ilustración 18) para hacer el relleno de color a los edificios.

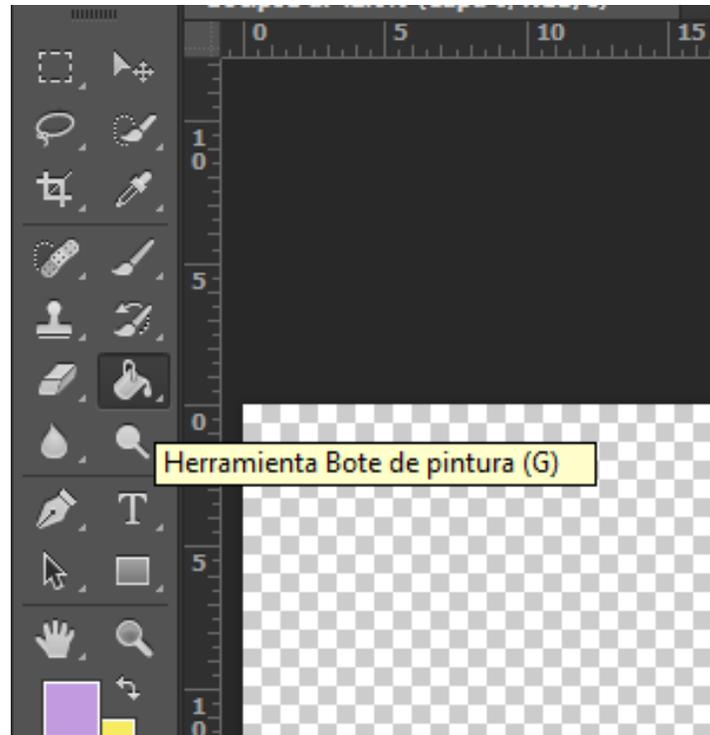


Ilustración 18 Herramienta bote de pintura

Con la herramienta de bote de pintura seleccionada procederemos a configurar el color frontal (ilustración 19), debemos hacer click izquierdo para que se abra una ventana emergente con el selector de colores (ilustración 2) del cual podremos elegir cualquier color.

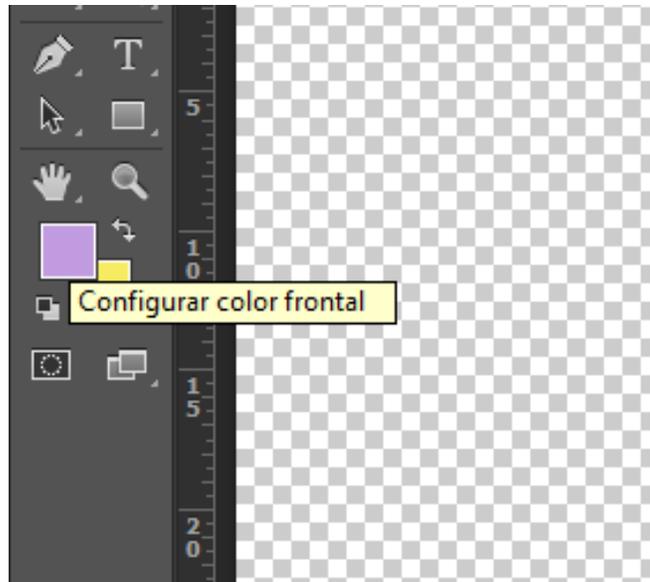


Ilustración 19 Color frontal

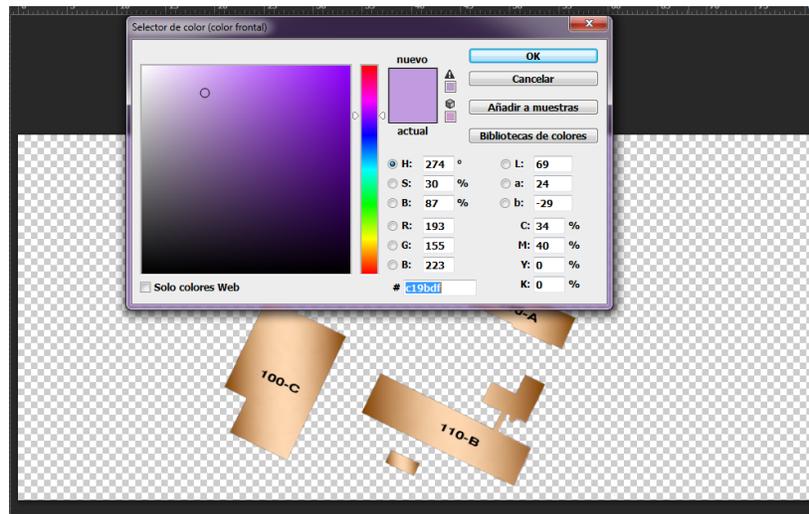


Ilustración 20 Selector de colores

Si bien escoger un color en particular, o alguno que nos agrade visualmente es complicado con el selector de color, tenemos una opción más, la cual aplicaremos para estas texturas. En la ventana de selector de color debemos presionar el botón de biblioteca de colores, lo cual nos abrirá una nueva ventana, de la cual podremos elegir entre una gran variedad de colores ya definidos (Ilustración 21).

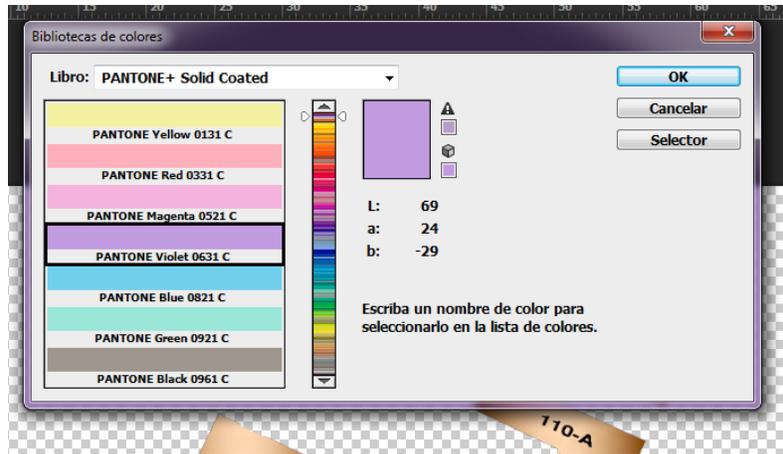


Ilustración 21 Biblioteca de Colores

Para esta facultad en particular se tomará el color pantone 7611C (Ilustración 22), sin embargo cualquier color puede ser usado, siempre teniendo en cuenta que deben de cambiar para cada facultad.



Ilustración 22 pantone 7611C

Una vez elegido el color se deberá hacer click izquierdo con la herramienta de bote de pintura seleccionada dentro de los edificios para pintarlas de dicho color, teniendo cuidado de no pintar por error el nombre del edificio. Será necesario hacer click varias veces en diferentes puntos para que se pinte por completo cada edificio (ilustración 23).

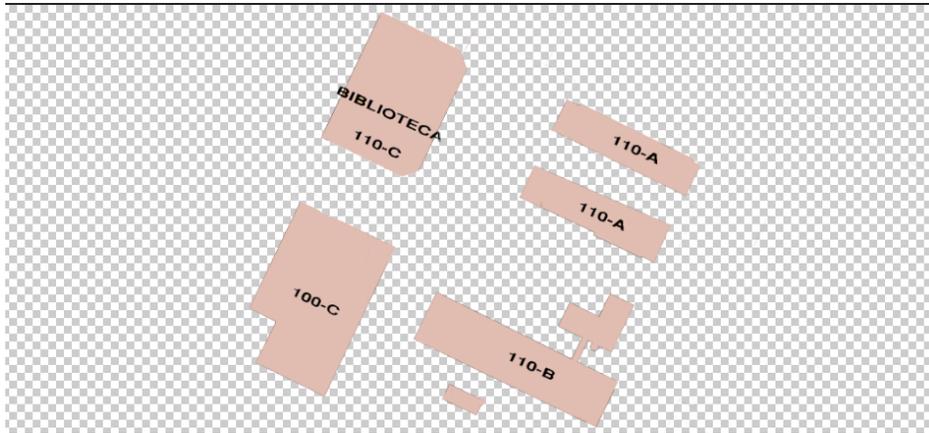


Ilustración 23 Edificios pintados

Una vez terminado de rellenar del color de la elección, solo basta guardar los cambios para tener la textura para unity. En la barra de herramientas -> Archivo -> Guardar como, se abrirá una ventana emergente, ahí se debe seleccionar la ubicación, el nombre del archivo (en este caso 110, haciendo referencia al grupo de edificios) y la extensión que será PNG (Ilustración 24). Debe ser en formato png, ya que este respeta la transparencia a diferencia del jpeg.

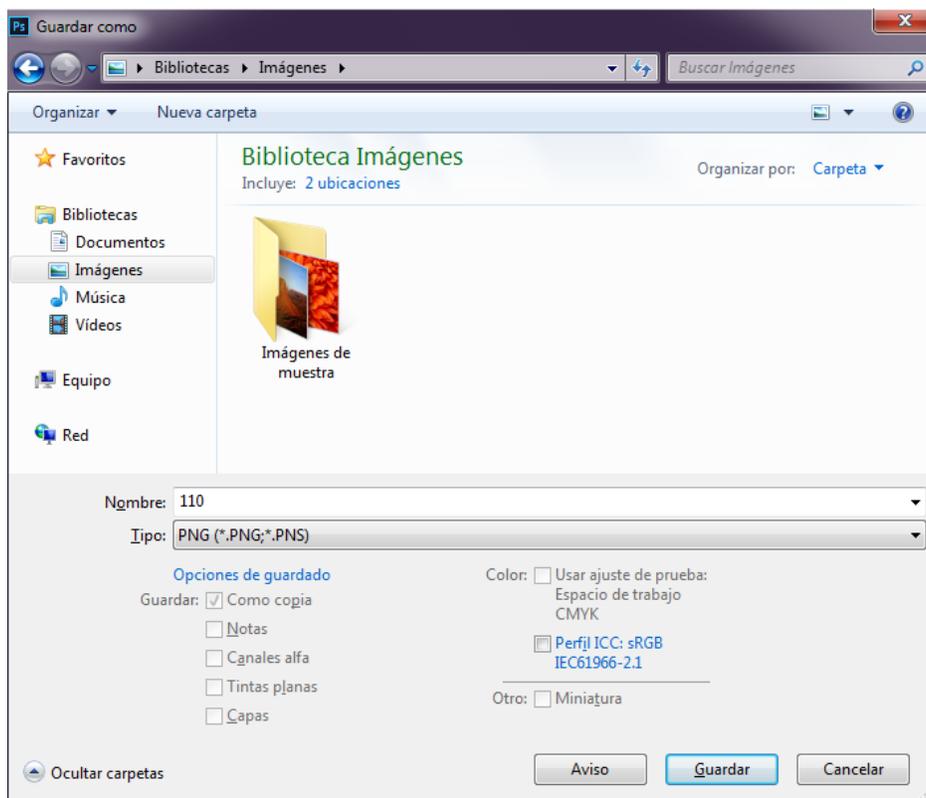


Ilustración 24 Guardar como

Se aplica el mismo proceso para cada una de las facultades, para contar con las texturas necesarias.

4.4 Configuración en Unity

Unity es un motor de videojuego multiplataforma creado por Unity Technologies, unity está disponible como plataforma de desarrollo para Microsoft Windows y OS X, y permite crear juegos para Windows, OS X, Linux, Xbox 360, PlayStation 3, PlayStatio vita, Wii, Wii u, Ipad, Iphone, Android y Windows Phone.

Para la creación del nuevo mundo en miniatura o el minimapa se hará sobre unity en la versión la versión 4.5.5 (Ilustración 25).

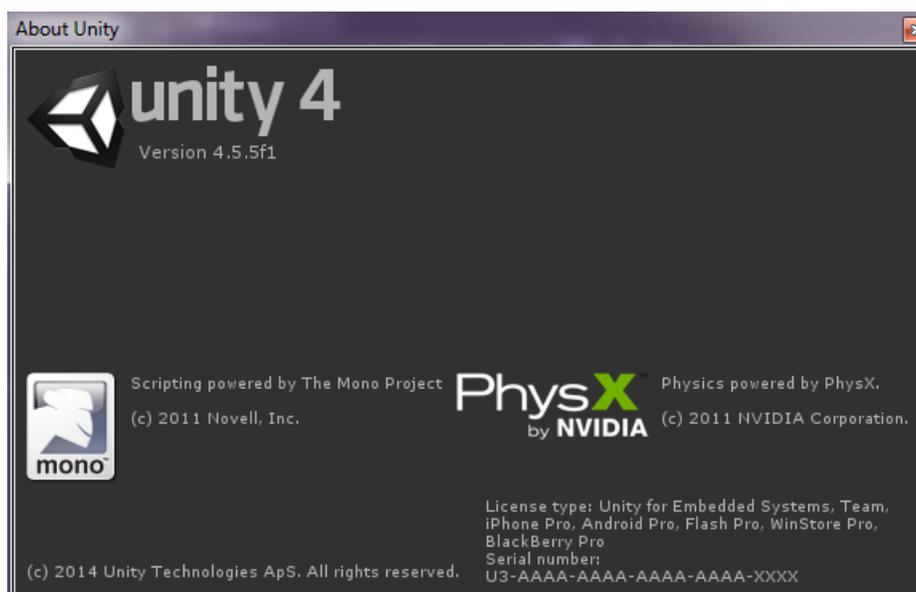


Ilustración 25 Unity 4.5.5

El mapa se añadirá a un proyecto ya existente de CU, esto con el fin de ayudar a la orientación. Lo primero es abrir el proyecto sobre el cual se incluirá el mapa. Este método para el mapa se puede insertar en cualquier momento y ajustarse a cualquier mundo en miniatura.

Lo primero es abrir el proyecto sobre el que queremos trabajar, en el menú File -> Open Project (Ilustración 26). Se abrirá una nueva ventana para elegir el proyecto, en caso de haber trabajado recientemente con el aparecerá directamente en esa ventana (ilustración 27), caso contrario se debe seleccionar la opción de abrir otro y seleccionar el proyecto.

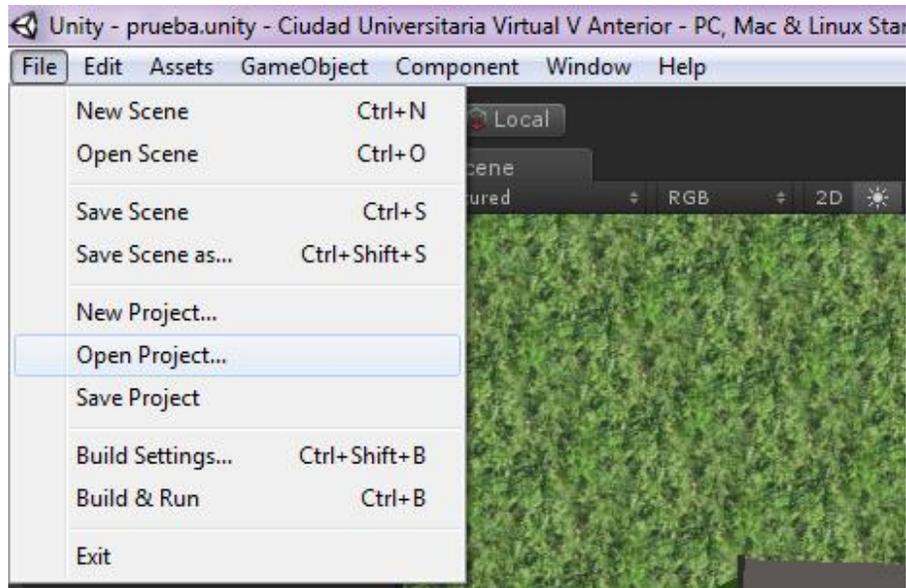


Ilustración 26 Abrir proyecto

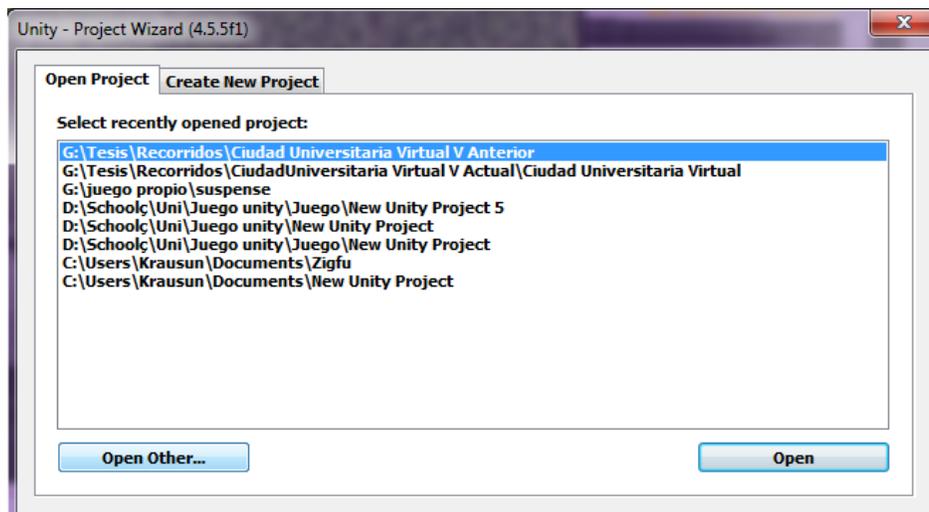


Ilustración 27 Selección de Proyecto

Una vez que tengamos el proyecto abierto lo primero que debemos agregar es una nueva cámara, al cual brindara la vista para el mini mapa. La cámara se encuentra en game Object -> Create Other -> Camera (Ilustración 28).

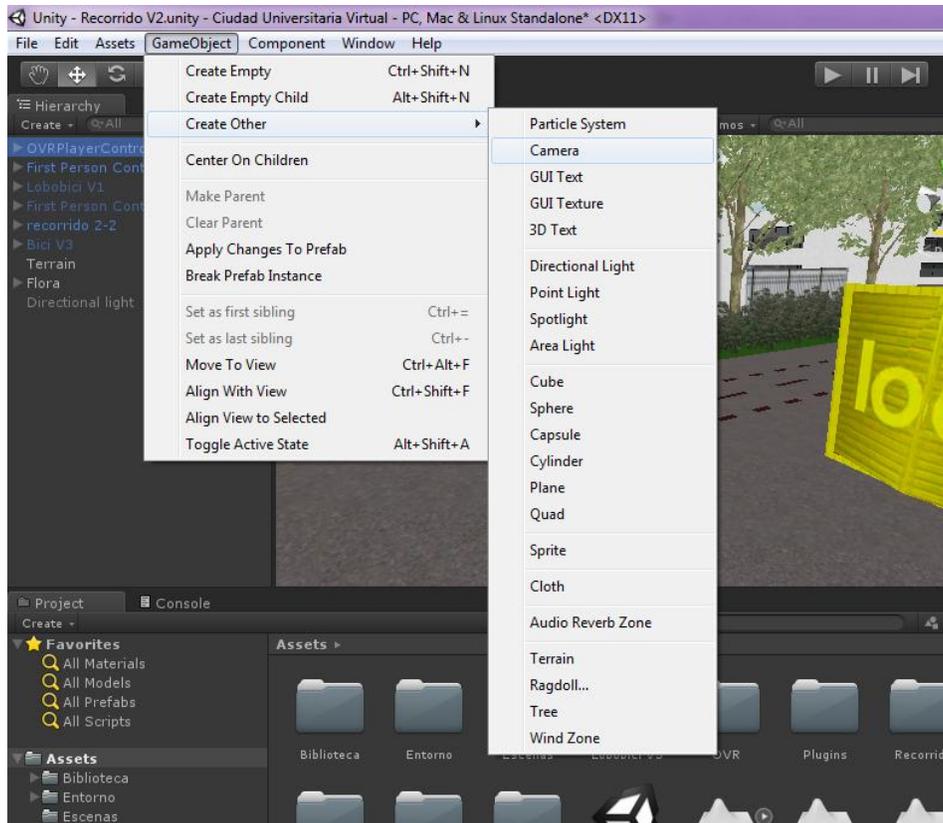


Ilustración 28 Crear Nueva Cámara

Una vez agregada la nueva cámara se mostrara de lado izquierdo (Ilustración 30), haciendo click nuevamente con la cámara seleccionada o presionando F2 se podrá modificar el nombre de la cámara, en este caso lo cambiaremos a minimapa (Ilustración 30) solo para tener más en claro de que es el componente, ya que en ocasiones, cuando más grande se hace el proyecto, si no se llevaba control de los componentes se tiende a confundirlos, sin embargo no es necesario.



Ilustración 29 Cámara

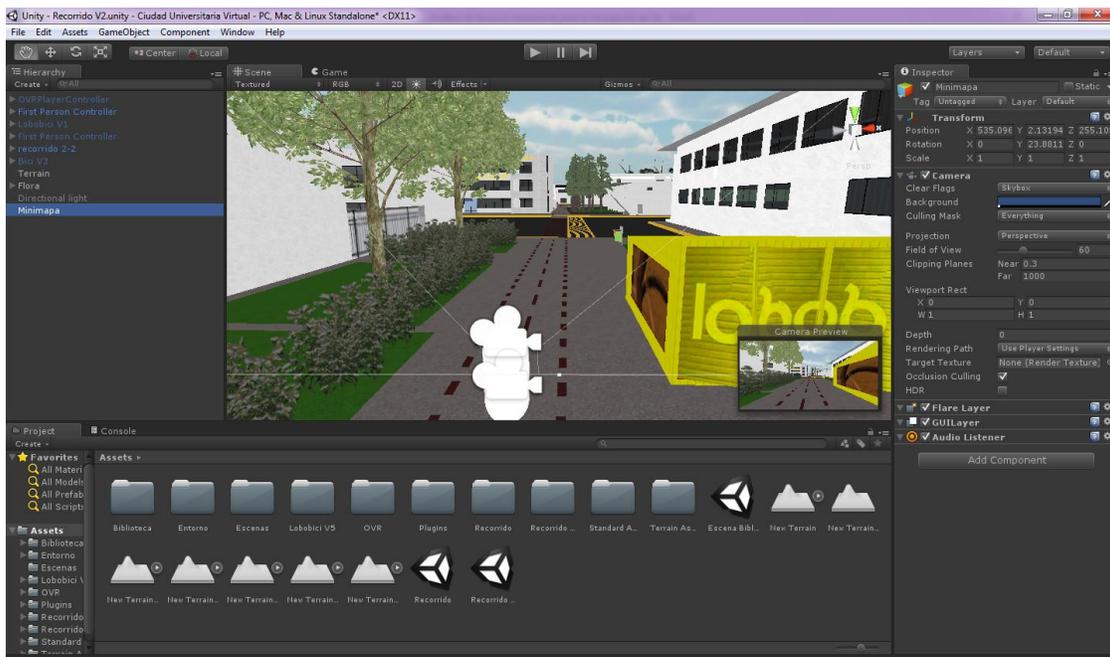


Ilustración 30 cambio de nombre

Con la cámara ya agregada debemos cambiar la perspectiva, nuestro objetivo es que vea el modelo desde arriba para que dé la impresión de un mapa.

La forma de cambiar la perspectiva es sencilla, con la cámara seleccionada solo tenemos que seleccionar en la barra de herramientas la herramienta de rotar  (ilustración 31). Una vez hecho esto se mostrarán 3 círculos alrededor de la cámara que representan cada coordenada (X, Y, Z) para poder hacer la rotación en la dirección que seleccionemos.

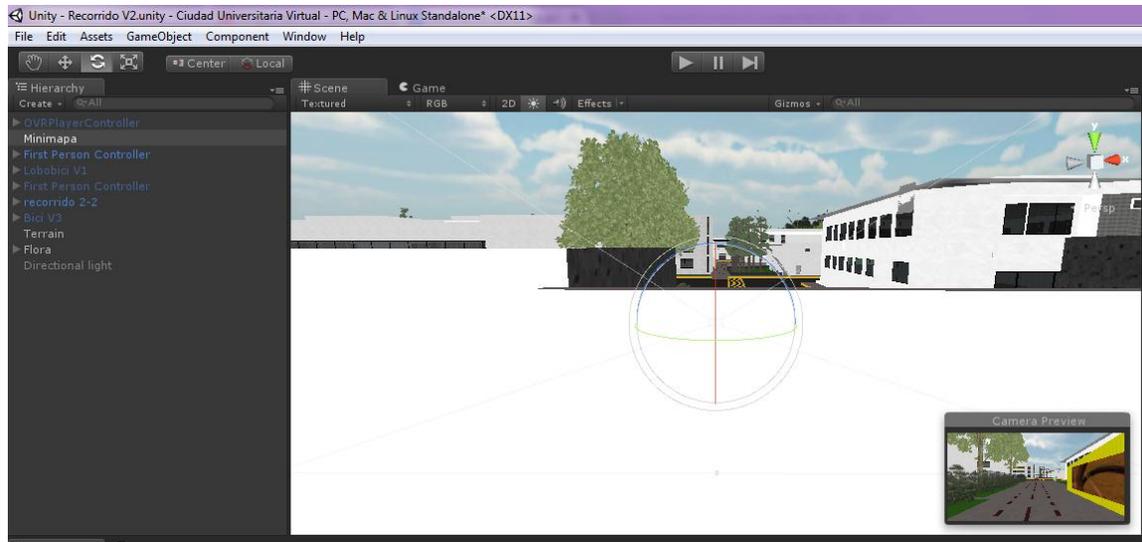


Ilustración 31 Herramienta Rotar

Haremos click izquierdo sobre la línea roja (Puede llegar a variar el color) y sin soltar moveremos hacia abajo, con esto empezara a rotar la vista de la cámara en el eje Y, rotaremos hasta que la cámara quede viendo al suelo (Ilustración 32). También es posible cambiar la configuración de lado derecho en el apartado Transform -> Rotation -> X a 90.

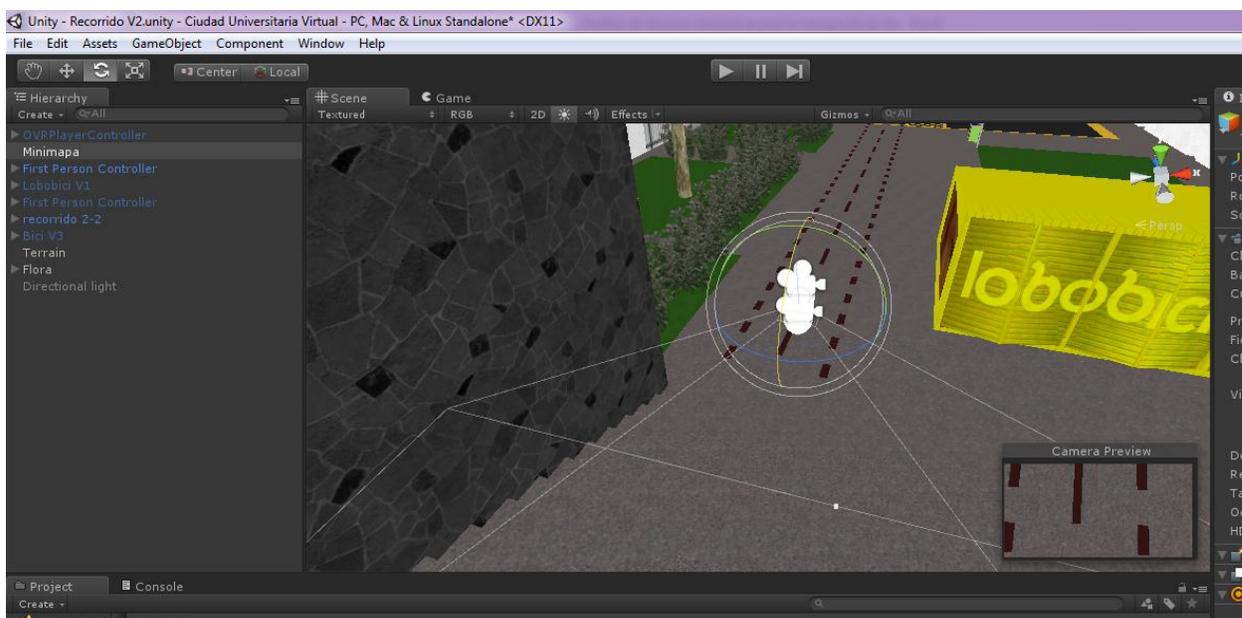


Ilustración 32 Cámara rotada

Ahora se debe desplazar la cámara hacia arriba para tener un panorama más amplio, para así poder mostrar más adelante las texturas que servirán como mapa. En la barra de herramientas debemos seleccionar la opción de desplazamiento . Con la cámara seleccionada al presionar el botón de desplazamiento aparecerán 3 direcciones en la cámara, cada una representa una coordenada (X, Y, Z) en la cual se puede desplazar el objeto (Ilustración 33). Click izquierdo sobre la línea azul (puede llegar a variar el color) y sin soltar se desplaza hacia arriba hasta una distancia que se aprecie adecuadamente la vista superior (Ilustración 34). También es posible modificar las propiedades de lado derecho, en este caso a 41.4025, pero varía dependiendo de la distancia que se necesite para cada proyecto.

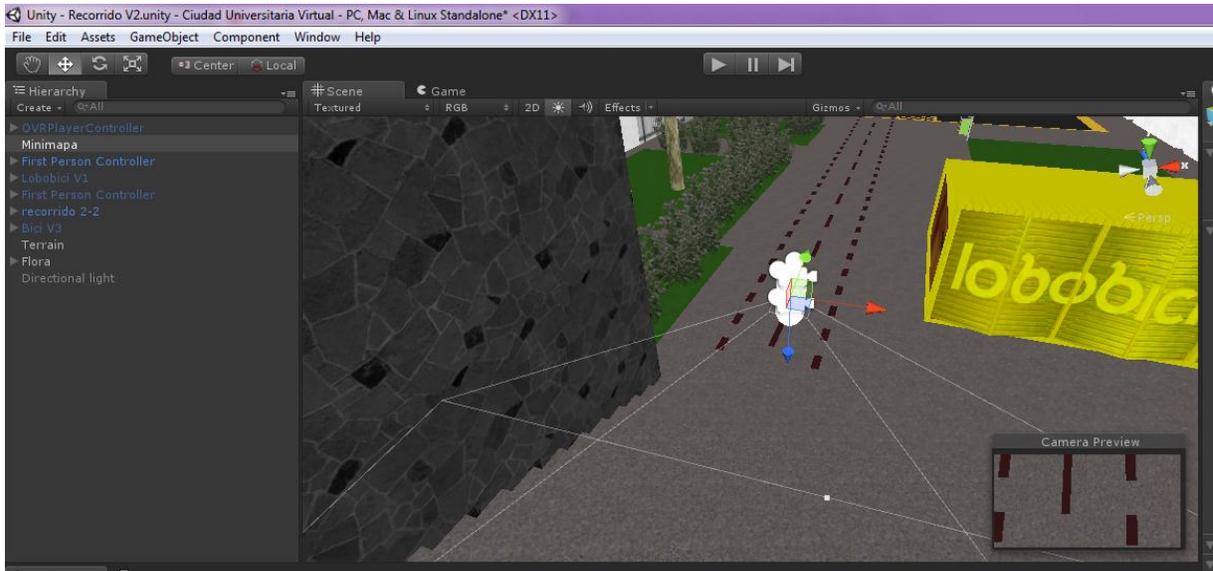


Ilustración 33 Herramienta desplazamiento

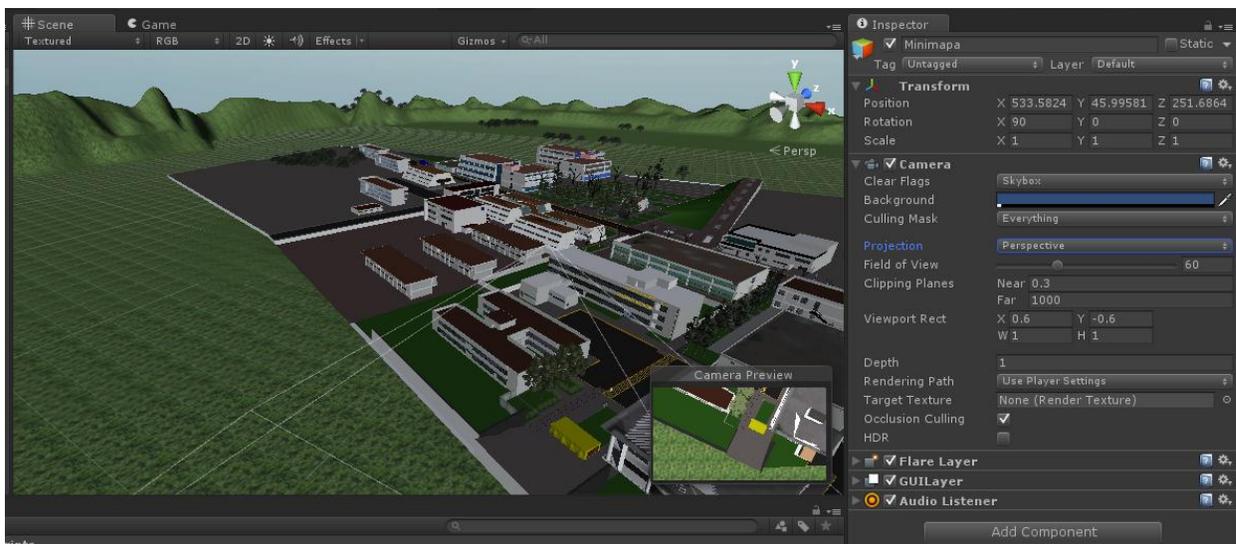


Ilustración 34 Cámara desplazada

Ahora si iniciamos el juego (▶) notaremos que no se ve la cámara que hemos agregado (ilustración 35), esto es porque aún queda algunas configuraciones adicionales para hacer visible la cámara.

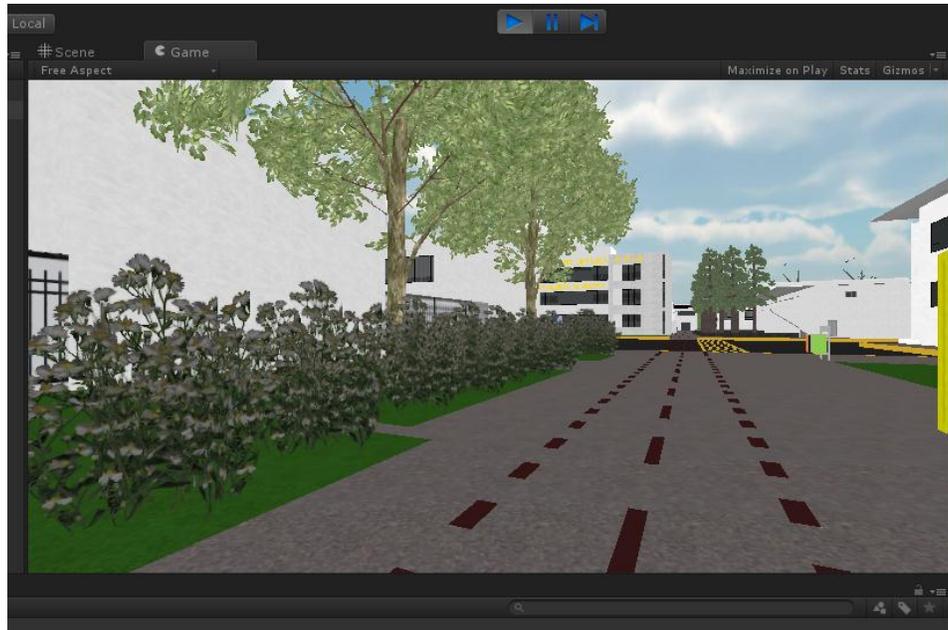


Ilustración 35 Juego iniciado

En las propiedades de la cámara (Ilustración 36) debemos modificar algunos datos en particular. Anteriormente modificamos en el apartado de Transformation -> Position -> Y. En el siguiente apartado de cámara (Camera) se modificarán los campos de viewport rect, near (puede o no cambiarse), depth, projection y size, para poder hacer visibles ambas cámaras al mismo tiempo. Primero debemos cambiar a la pestaña de Game para poder ver directamente los cambios, esto para que al modificar podamos adecuarlo completamente a lo que sea necesario. En donde se muestra el juego, encima se encuentran dos pestañas, una de Scene y otra de Game, debemos tener activa la de game para ver los cambios en el momento que se están haciendo, si no se encuentra en la pestaña de Game, basta con hacer click izquierdo sobre ella para pasar a esa pestaña (Ilustración 37).

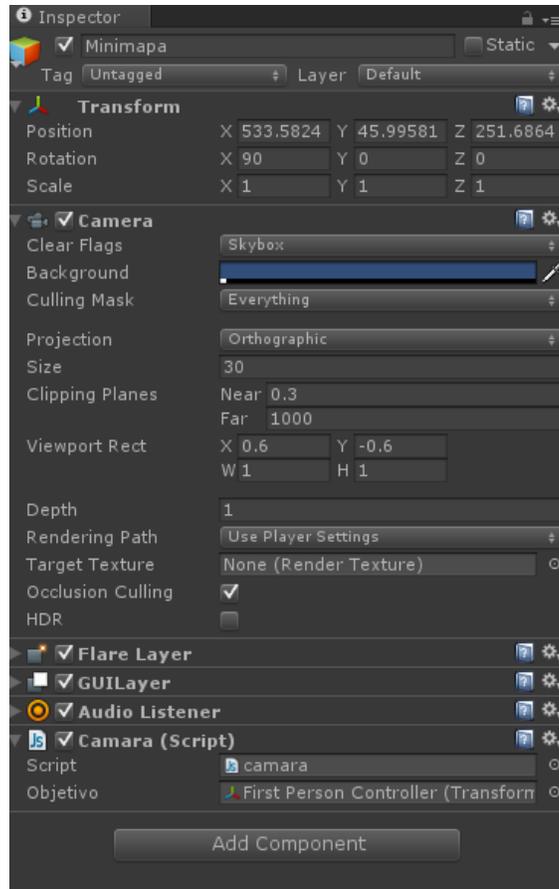


Ilustración 36 Propiedades cámara

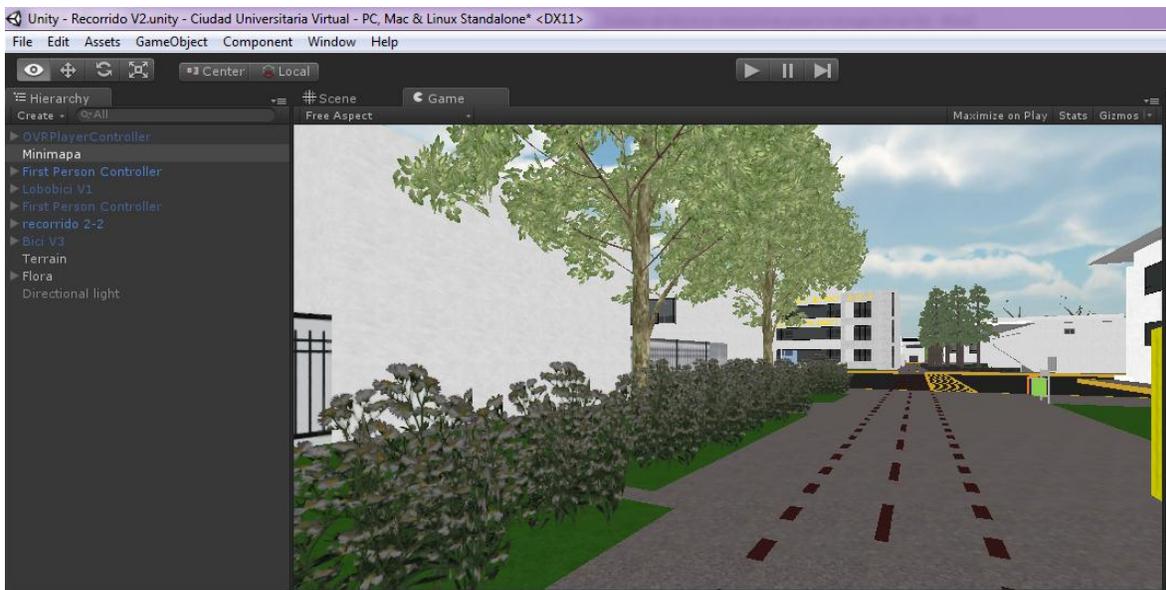


Ilustración 37 Pestaña Game

Cambiaremos los valores por default de Viewport rect, para esto posicionamos el cursor sobre la x, damos click izquierdo y arrastramos hacia un lado, en este caso la derecha, hasta que quede de un tamaño que se considere conveniente, dependiendo del proyecto. En este caso quedo en .6, se repite el proceso con y, igualmente en este caso se deja en .6, esto para que quede ubicado en la parte inferior derecha. También es posible solo cambiar el valor, si es que ya se tienen valores predefinidos. El campo near también se puede modificar según las necesidades del proyecto, o puede quedar tal como está, se puede cambiar del mismo modo que viewport rect, hasta que el mini mapa se vea lo mejor posible, en cuanto al campo depth, debe cambiarse a 1, para que así este por encima de la cámara principal y sea visible. En cuanto a projection cuenta con dos opciones, perspective y orthographic, por default aparece en perspective, se debe cambiar a orthographic, ya que puede presentar algunos problemas para nuestro mapa si lo dejamos en perspective. Una vez hecho esto, el campo que se encuentra bajo a projection cambiara a size, en este campo establecemos el valor a 30. Con esto terminamos las configuraciones básicas (Ilustración 38), aunque más adelante habrá que hacer unos cambios adicionales.

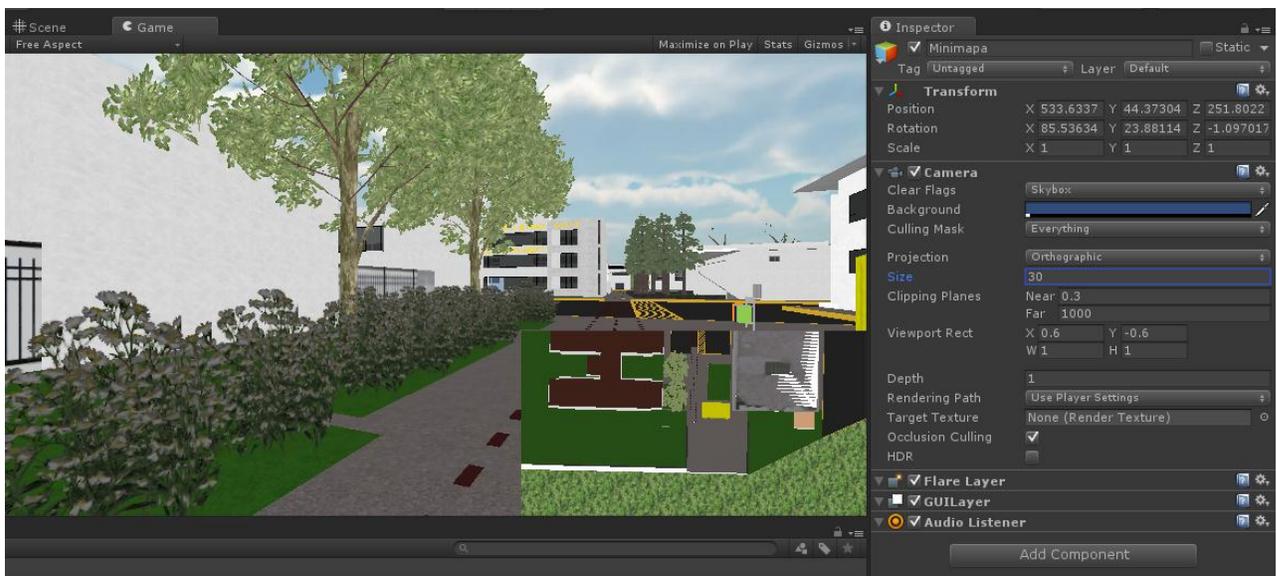


Ilustración 38 Cámara configurada

Si damos play para ver el juego, notaremos que la cámara ya no se pierde, sin embargo, esta no sigue al personaje como debería un minimapa. Para esto necesitamos agregar scripts para hacer que la cámara tenga como objetivo al personaje y se mueva junto con él, además de tener la opción de mostrar u ocultar el minimapa.

Se tiene que crear el script para que el mini mapa siga al personaje. Los scripts para unity pueden ser creados tanto C# como en JavaScript. Para este proyecto se usarán scripts hechos en JavaScript.

4.5 Scripts

Unity cuenta con un editor para trabajar con los scripts llamado Monodevelop, el cual puede ser utilizado para crear Scripts o modificar los ya existentes, sin embargo, no es la única manera de crearlos, ya es cuestión del usuario determinar el método para desarrollarlos.

En nuestro caso, ya que los scripts son pequeños, no es necesario contar con ninguna herramienta en particular, se pueden crear en un bloc de notas.

El primer script, el cual es un código para fijar la cámara sobre un objetivo, en este caso el personaje, se muestra a continuación:

```
var objetivo:Transform;

function LateUpdate (){

    transform.position = new
    Vector3(objetivo.position.x,objetivo.position.y,objetivo.position.z);
}
```

El segundo script, el cual es una manera de mostrar u ocultar el minimapa, según la elección del usuario, es el siguiente:

```
var cam1 : Camera;

function Start() {
    //activo y desactivo las camaras necesarias
    if(cam1.enabled==true)
        cam1.enabled = false;
    else
        cam1.enabled = true;
```

```
}
```

```
function Update() {
```

```
    if (Input.GetKeyDown(KeyCode.C)) {
```

```
        //al presionar la tecla c se activa la camara desactivada y se desactiva la otra camara
```

```
        cam1.enabled = !cam1.enabled;
```

```
    }
```

```
}
```

Si se desea cambiar la tecla con la que se muestra u oculta el mini mapa se debe reemplazar en la línea “if (Input.GetKeyDown(KeyCode.C)) {” la letra C, por ejemplo a M y quedaría así:

```
“if (Input.GetKeyDown(KeyCode.M)) {”
```

Se debe abrir un bloc de notas en blanco, escribir el código y guardar el archivo, el nombre no importa, pero es recomendable que sea uno que dé idea de su función, en este caso se nombrará cámara. Después de escribir el nombre se debe poner la extensión .js, para que el archivo sea un Script (Ilustración 39). La ubicación del archivo puede elegirse, no afecta la ubicación. Se debe hacer este proceso para ambos scripts.

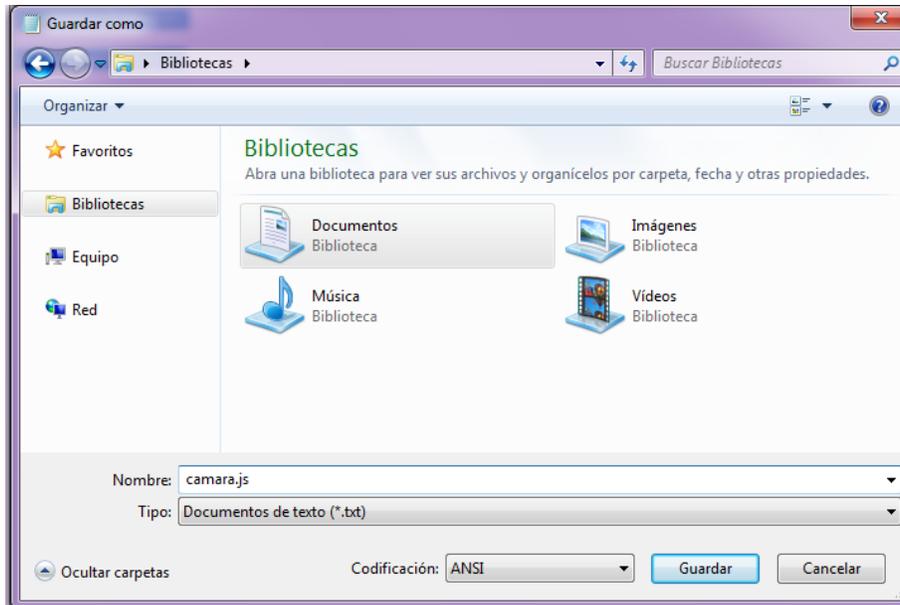


Ilustración 39 Guardar Script

Para llevar el script a unity se debe ir al menú Asstes -> Import New Asset. Con esto se abrirá una ventana emergente, solo hay que ir a la ubicación donde se guardó el script, seleccionarlo y dar click en el botón Import (Ilustración 40, 41). Ambos scripts deben importarse.

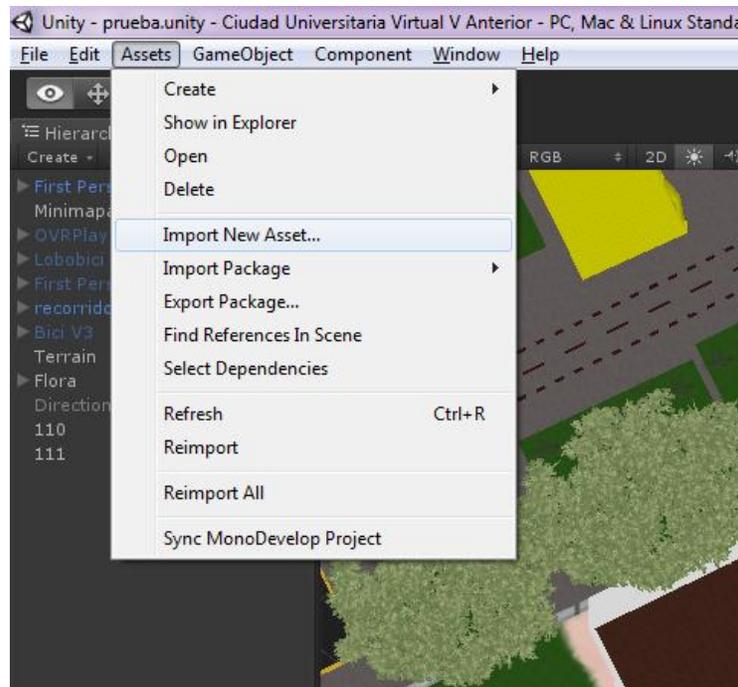


Ilustración 40 Importar Script

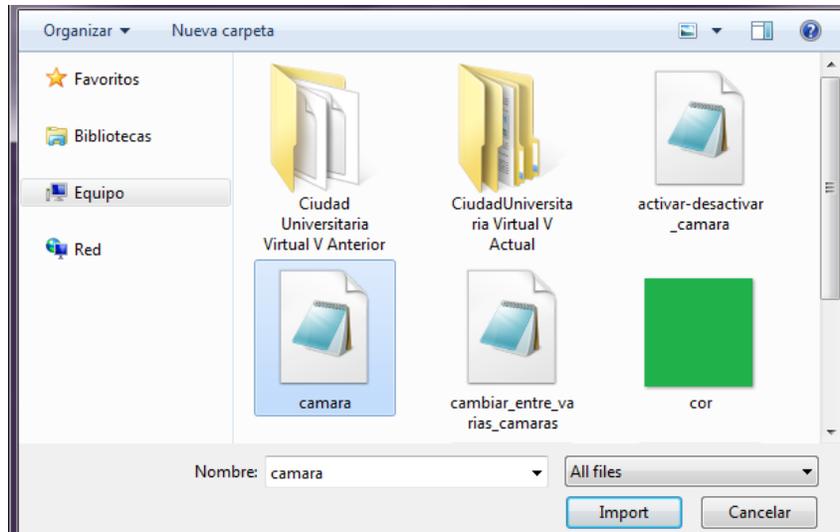


Ilustración 41 Seleccionar Script

Ahora que los scripts están dentro de nuestro proyecto de unity, se deben de adjuntar a la cámara, para que la modifiquen. Con la cámara seleccionada que se creó para el mini mapa (Ilustración 43), se da click izquierdo sobre el script y sin soltarlo se mueve el cursor hasta las propiedades, bajo el botón de Add Component y se suelta, esto hará que el script se adjunte a la cámara, añadiendo así funciones, en este caso el de hacer que la cámara siga al personaje. Aparecerá un nuevo apartado después de haber puesto el script, este apartado se llama cámara (Cambia dependiendo de cómo se llame el script) y tiene dos campos (los cambios pueden variar dependiendo de las funciones específicas del script). A pesar que el script ya está adjunto a la cámara, aun no sigue al personaje, esto se debe a los campos de este apartado. El primer campo es el de Script, el cual tiene el script que se acaba de añadir, este permanecerá sin cambios. El segundo es Objetivo, en este apartado debemos fijar al personaje, para que la cámara siga a este. Para esto se repite el mismo proceso que se hizo para añadir el script, solo que esta vez el objetivo a seleccionar es el personaje (que en este caso es el First Person Controller) y se soltara enfrente del apartado objetivo, en la caja de texto que tiene por default None (transform) (Ilustración 44). Esto hará que la cámara siga al personaje.

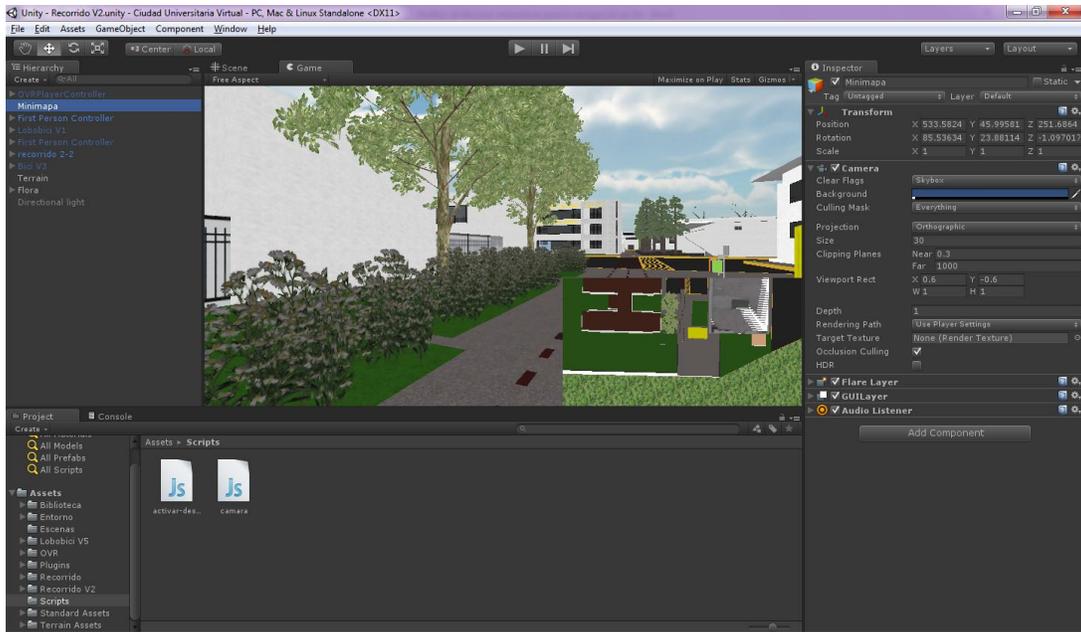


Ilustración 42 Adjuntar scripts

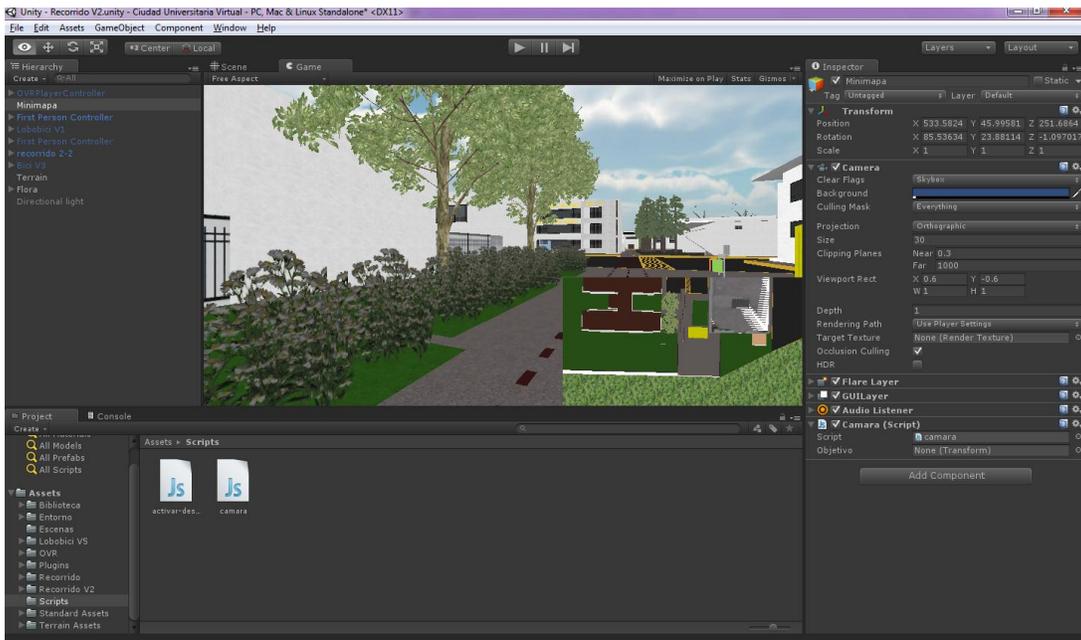


Ilustración 43 Cámara Adjunta

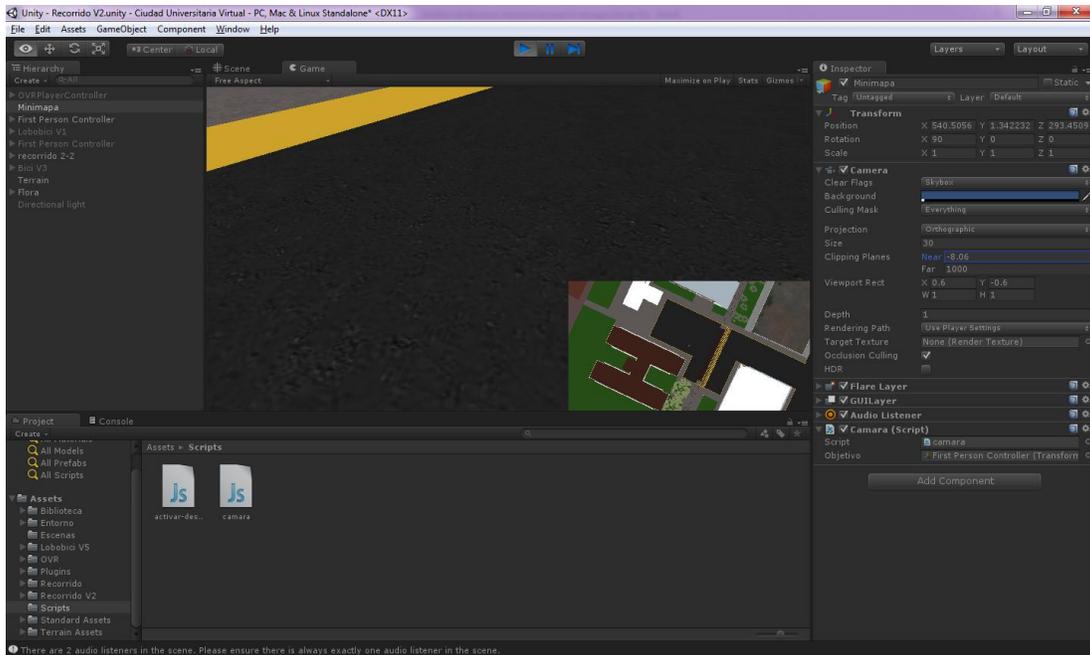


Ilustración 44 Objetivo fijado en el script

Para el siguiente Script se repite exactamente el mismo proceso para agregarlo. Este script tendrá también dos campos: Script y Cam 1. El primer campo, al igual que en el script anterior se quedará tal como está, mientras que con el segundo se seguirá el procedimiento anterior para el campo objetivo, solo que en este caso será con la cámara del minimapa, para que con esto se añada a la cámara la función de mostrarse u ocultarse al presionar la tecla C (o la tecla que se haya definido).

Con esto los scripts y sus configuraciones correspondientes añadidas a la cámara (Ilustración 45).



Ilustración 45 Scripts añadidos y configurados

4.6 Texturas

Ahora se deben añadir las texturas para que puedan verse en el minimapa, y así conocer la ubicación del personaje en ese momento.

Primero se deben añadir las texturas al proyecto, para esto, se debe ir a la ubicación de las texturas en el explorador de Windows (Ilustración 46), una vez ahí se debe hacer un proceso similar al que se utilizó para adjuntar el Script a la cámara, en este caso, se hará click izquierdo sobre la carpeta, y sin soltar se arrastrara la carpeta hacia unity, en la parte inferior izquierda, sobre la carpeta assets (puede colocarse incluso dentro de otra de las carpetas que se muestran ahí, es cuestión de cómo se desee organizar la información), una vez hecho esto comenzara a importar las texturas al proyecto, solo que habrá que esperar unos segundos (el tiempo varía dependiendo de cuantos archivos se importan) y se habrán agregado a unity (Ilustración 47). Como se tomó la carpeta completa de unity, todas las texturas se importan dentro de una carpeta con ese nombre, puede importarse directamente sin la carpeta, pero esto brinda un mejor control.

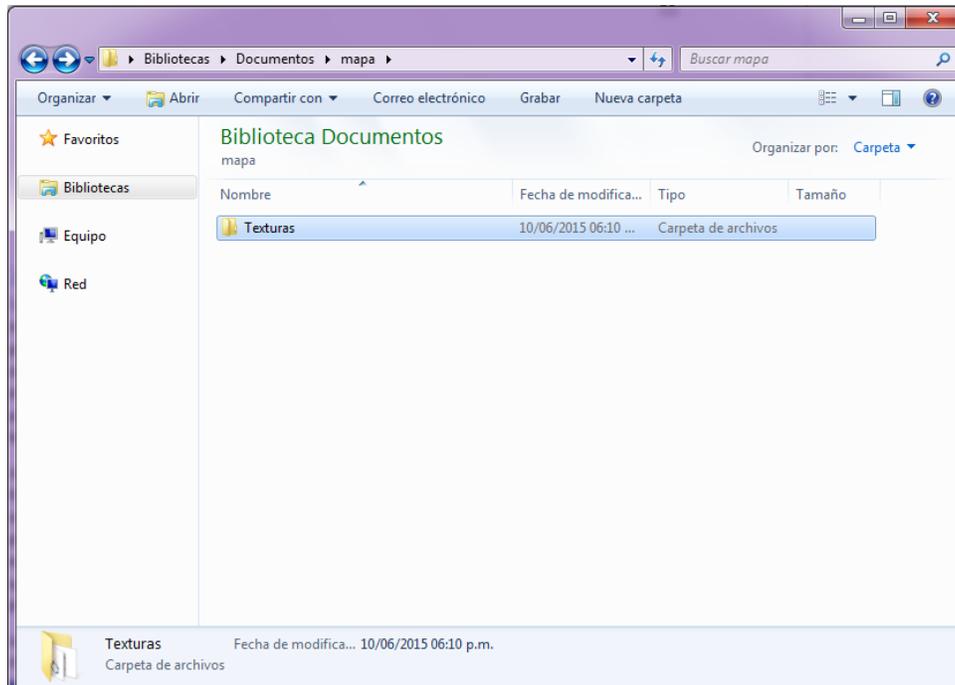


Ilustración 46 Carpeta de texturas

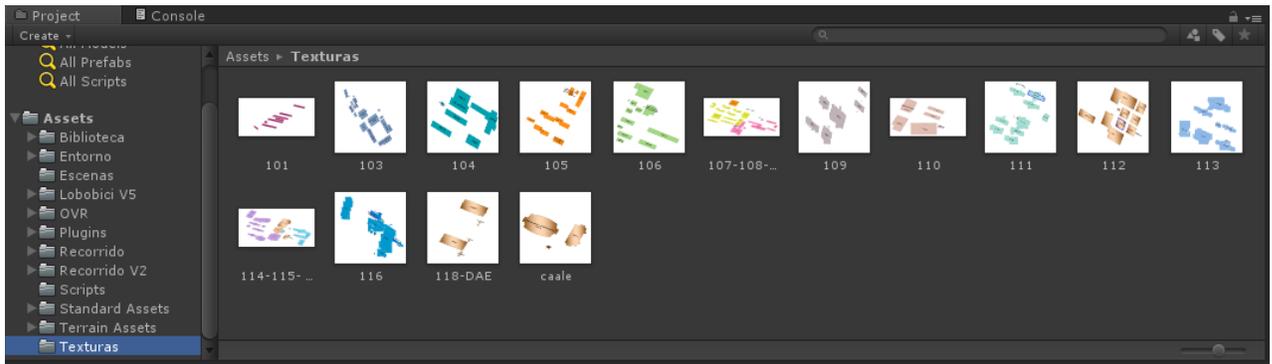


Ilustración 47 Texturas importadas

Ahora que se cuenta con las texturas dentro de unity se tienen que “fijar” para que sean visibles en el minimapa.

Para esto usaremos un objeto adicional que nos brinda unity: El plano (“Plane”), el cual se encuentra en Game object -> Create other -> Plane (Ilustración 48).

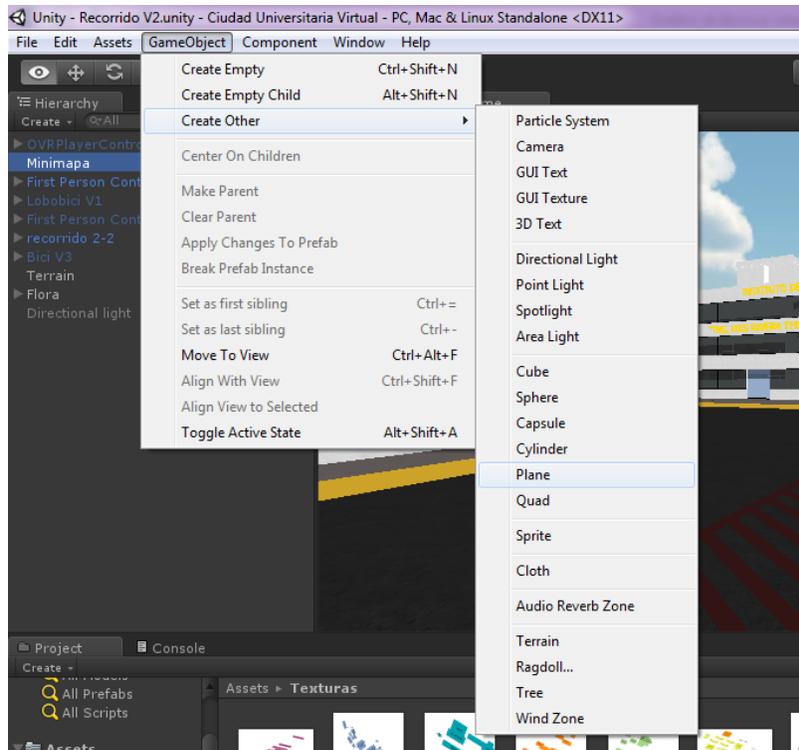


Ilustración 48 Plano

Ahora debemos de regresar a la pestaña de Scene para poder editar el plano que se acaba de agregar (Ilustración 49).

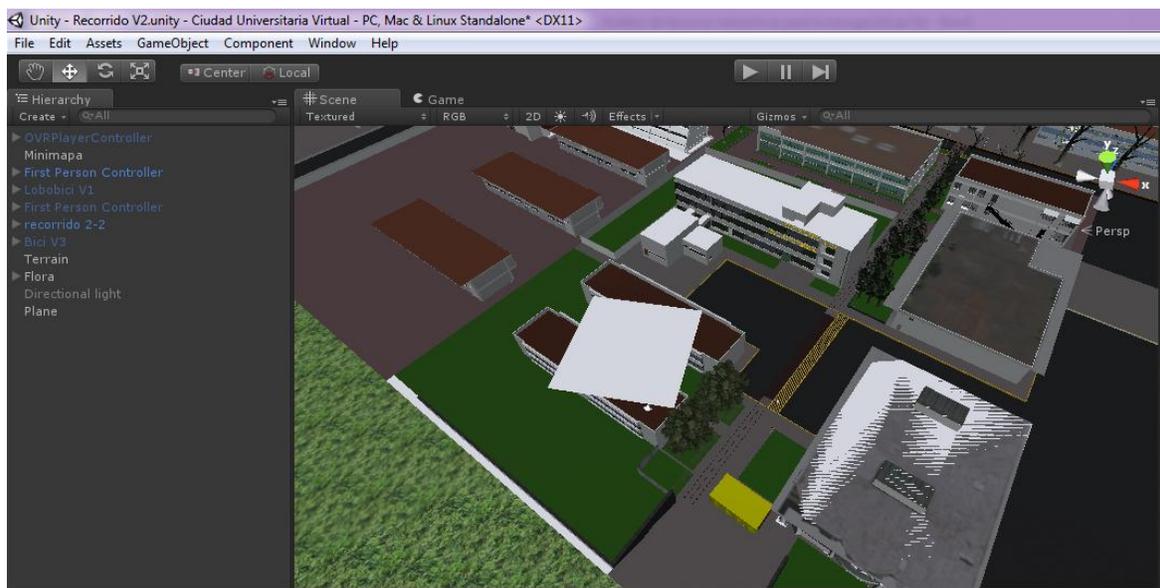


Ilustración 49 Plano agregado al proyecto

El plano recién agregado se puede dejar con ese nombre, pero es recomendable cambiarlo por alguno que identifique su función, en este caso servirá como base para la textura de dicha facultad

(110 - Instituto de Física). Con el plano seleccionado, se selecciona la herramienta de desplazamiento, y se desplaza casi a la misma altura del suelo o se modifican las propiedades directamente en el campo position -> Y, quedando en la posición Y alrededor de 16 (Ilustración 50).

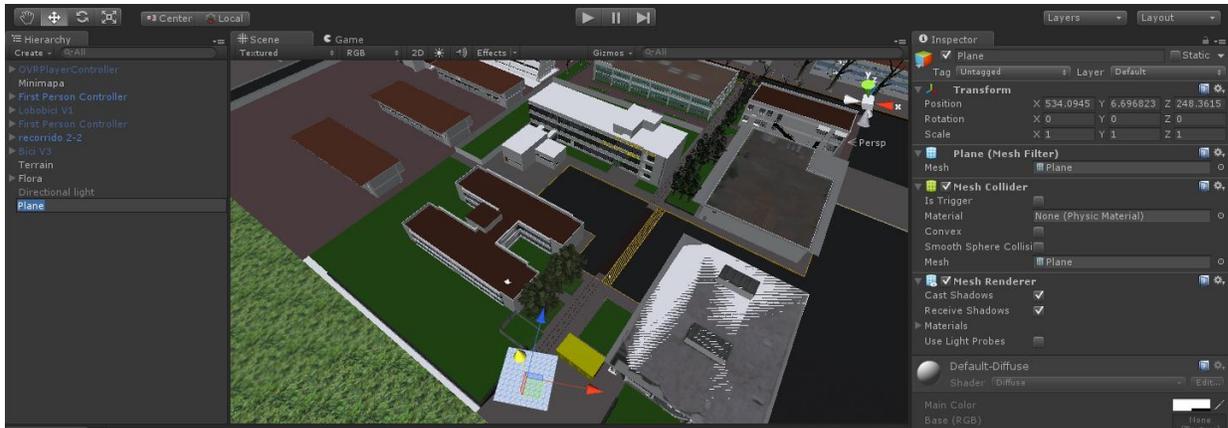


Ilustración 50 Plano desplazado

Una vez que se ha desplazado el plano, se debe transformar su tamaño, para que la textura se adapte al tamaño que se requiere, en este caso, que abarque los edificios que componen el Instituto de Física, para que correspondan cada uno con la textura.

Primero se adjuntara la textura que nos servirá como guía para reestructurar el plano y que quede acorde al tamaño requerido. Para esto, con el plano seleccionado, se debe ir a la carpeta que se agregó en unity de las texturas, seleccionamos la que corresponda a esa zona, en este caso la textura 110, y se adjuntara al plano. El proceso para adjunta es similar a los anteriores, con el plano seleccionado, se da click izquierdo sobre la textura, y sin soltar se arrastra hasta el plano. Esto hará que la textura aparezca en la superficie del plano (ilustración 51).

Ahora que la textura esta fija en la superficie del plano, se puede apreciar que los espacios que se están vacíos en la textura, unity lo rellena por default con color blanco, esto se debe a una propiedad del plano. Es necesario que nuestra textura solo muestre las áreas de interés, para esto, en las propiedades del plano, en la última sección, la cual es la textura que recién fue añadida, en la opción shader, se da click sobre la palabra Diffuse (que es la opción que pone por default), y debe ser cambiada por la opción Transparent -> Diffuse, esto hará que la textura se vuelva transparente (Ilustración 52).

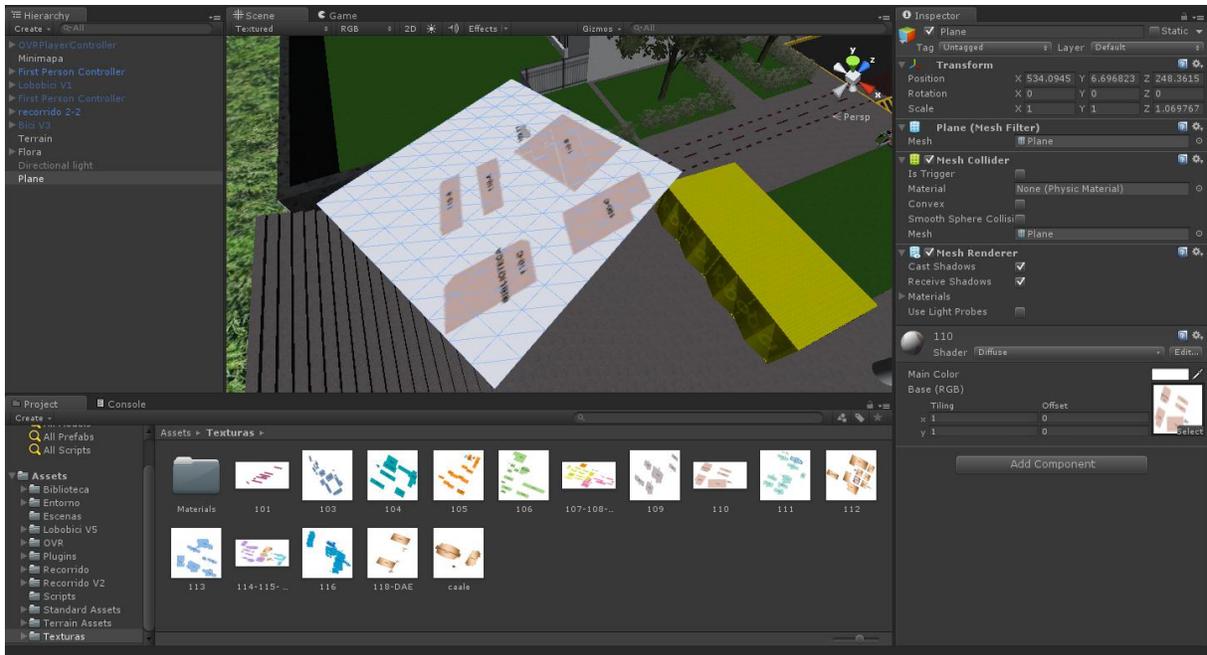


Ilustración 51 Textura añadida al plano

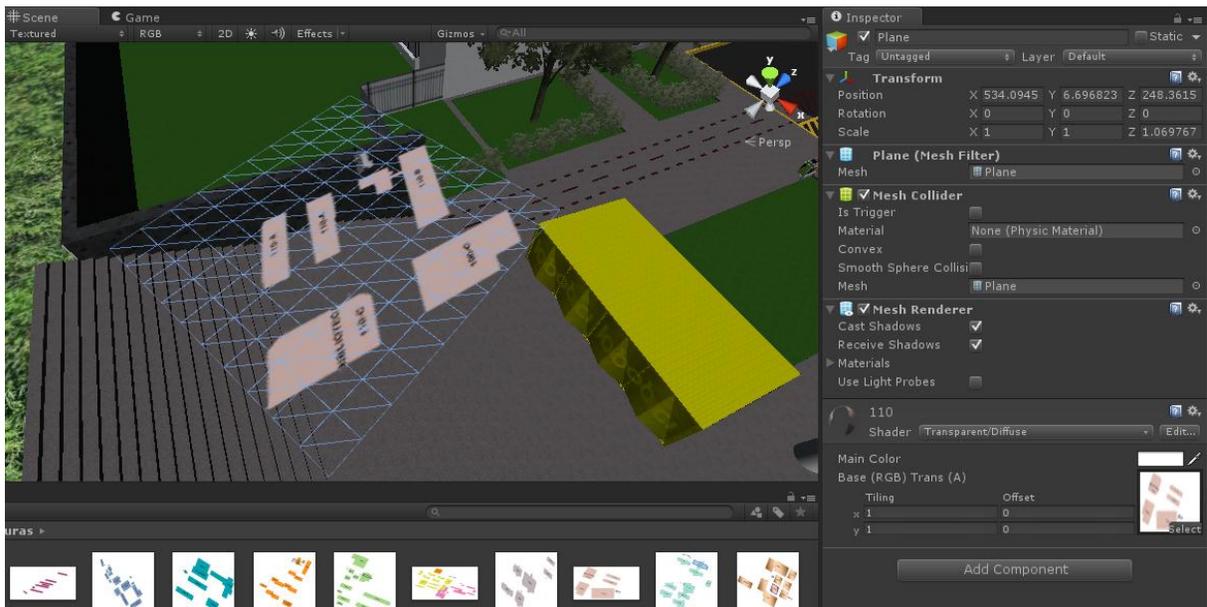


Ilustración 52 Textura transparente

Ahora que esta la textura y ya está transparente nos servirá como guía para adaptarlo al tamaño adecuado. Será necesario rotarlo, así como agrandar el plano para que corresponda al tamaño necesario. Para rotarlo se utiliza la herramienta que usamos para rotar la cámara, de manera que quede en la posición que corresponda con el mapa. Para aumentar su tamaño, en la barra de

herramientas se selecciona la herramienta de escalar (), al igual que para desplazar aparecerán 3 líneas que nos servirán para aumentar el tamaño del plano. En esta ocasión no es posible modificar directamente las propiedades del plano, a menos que se sepa con exactitud el tamaño necesario. Se deben usar estas líneas para aumentar el tamaño del plano. Se usaran estas opciones repetidamente, así como la de desplazamiento hasta ubicar la textura en la posición correcta (Ilustración 53). Aunque parezca que ya está en posición, al darle play para ver el mini mapa (no olvidando que el script oculta por default el mapa, y para visualizarlo hay que presionar la tecla 'c') puede que no se aprecie, por lo que habrá que modificar la posición en el eje Y hasta que la cámara del minimapa aprecie la textura.

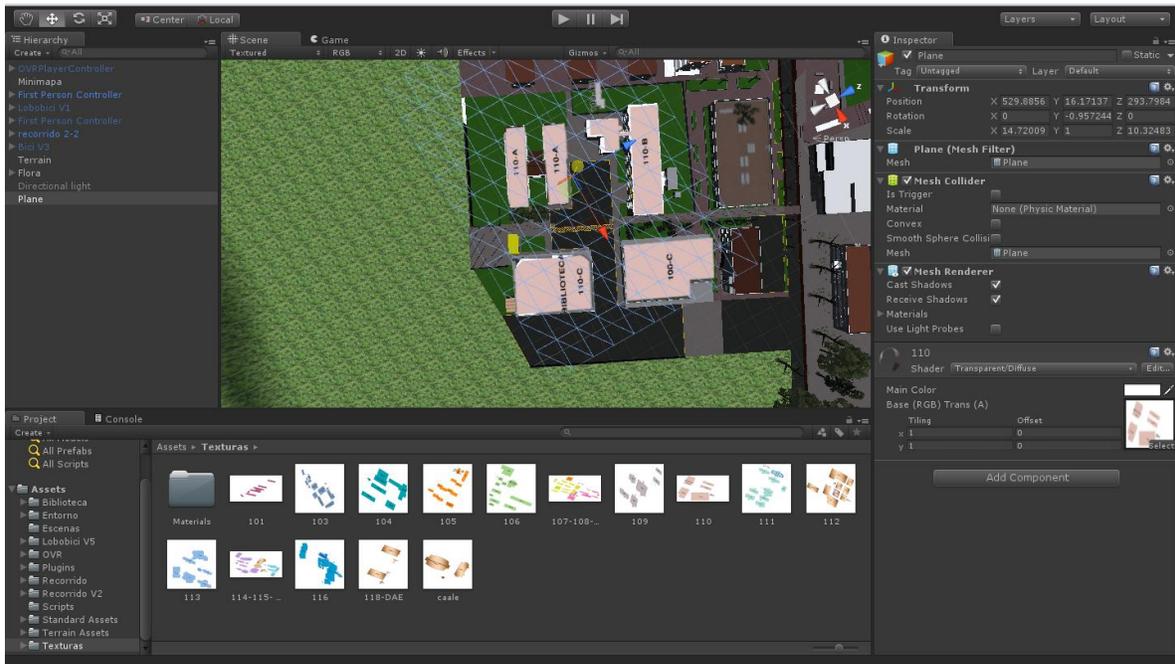


Ilustración 53 Textura ajustada

Solo queda quitar las colisiones para que el personaje no tope con el plano que contiene la textura, en las propiedades basta con desactivar la opción de Mesh Collider para quitar este efecto y el personaje pueda caminar como si no hubiera nada.

Si bien el mapa con esto ya se puede considerar terminado solo resta agregar un curso para saber con exactitud la posición del personaje.

El cursor puede ser cualquier imagen, esta nos permitirá ver en el minimapa el lugar donde está pasando el personaje. Existen diferentes imágenes que pueden funcionar como cursor, en este caso se usara un triángulo verde (Ilustración 54).

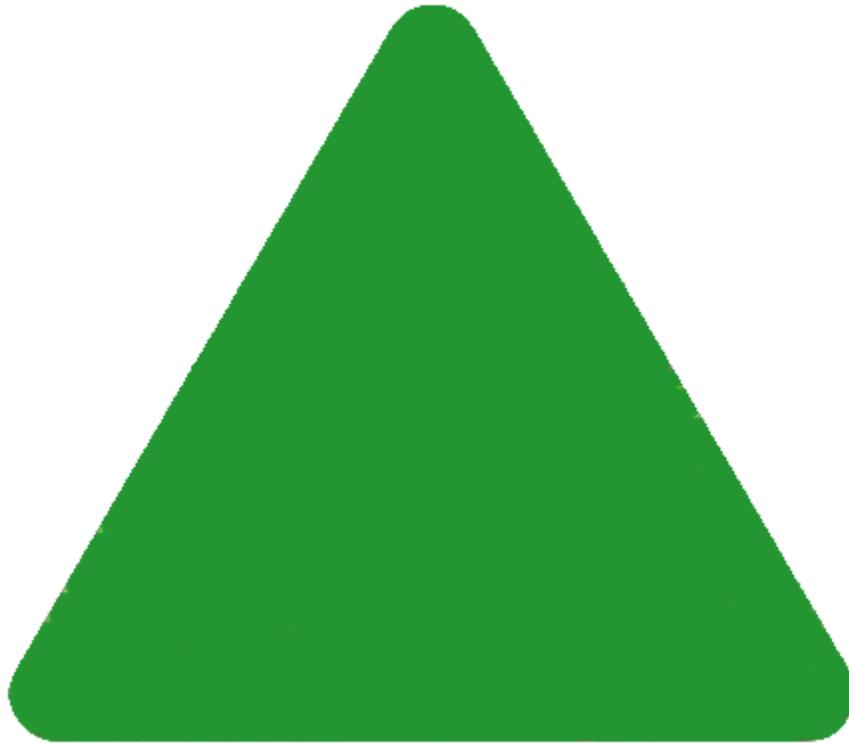


Ilustración 54 Cursor

La imagen que se haya elegido se debe importar a unity de la misma manera que las texturas, puede ser añadido en la carpeta raíz (Assets) o crear una para la textura. Una vez en unity se debe crear un plano como se hizo para añadir la textura de la facultad. Es recomendable renombrar el plano con algo que dé idea de para qué es, por ejemplo "Cursor". Se adjuntará la imagen al plano con el proceso anterior. Hecho esto se debe de posicionar en el exacto sitio donde se encuentra el personaje y cambiar la configuración para hacerla transparente como con las texturas anteriores, y ajustar la altura para que pueda verse con la cámara.

Ahora se debe ligar al personaje, para que se mueva con este, para esto basta con seguir el mismo proceso para adjuntar las texturas, solo que esta vez se hará con el plano, se debe arrastrar y soltar en el personaje (First person controller), esto hará que se adjunte como un componente del personaje (Ilustración 55).



Ilustración 55 Textura Adjunta

Una vez hecho esto el mini mapa ya está listo para ser usado, sin embargo, en ocasiones al adjuntar la textura, la cámara que está ligada al personaje también permite ver la textura añadida, mostrando parte del triángulo mientras el personaje camina (Puede o no presentarse, depende mucho de las configuraciones de cada proyecto). Para corregir esto se tienen que crear capas para las cámaras, es decir, las cámaras omitirán lo que corresponda a otra capa, en este caso, la textura que funciona como cursor.

Para hacer esto, primero se deben crear las capas. Se tiene que desplegar el contenido del personaje (First person controller) y seleccionar el cursor. En las propiedades de lado derecho se debe desplegar las opciones de capa y usar la opción add layer (Ilustración 56).

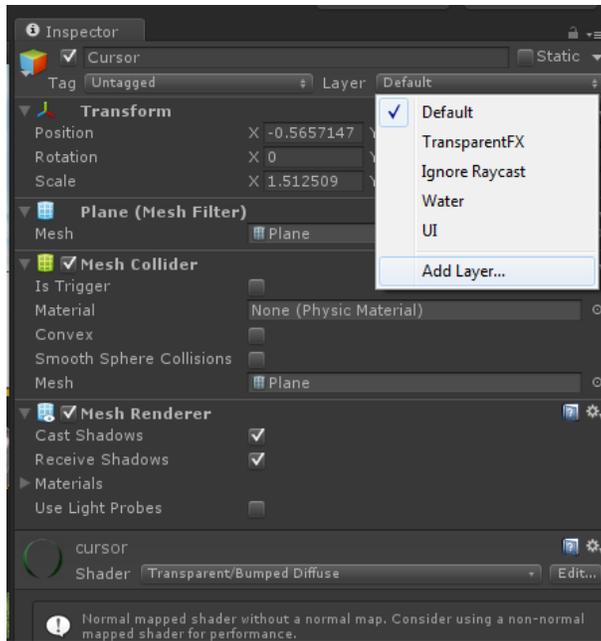


Ilustración 56 Agregar capa

Esto desplegará otra ventana con las capas enlistadas que ya existen, y el espacio para agregar nuevas, basta con escribir en alguna en blanco el nombre de la nueva capa que se desea tener, para este caso se nombró cursor (Ilustración 57), una vez hecho esto solo hay que hacer click fuera de esa ventana para cerrarla.

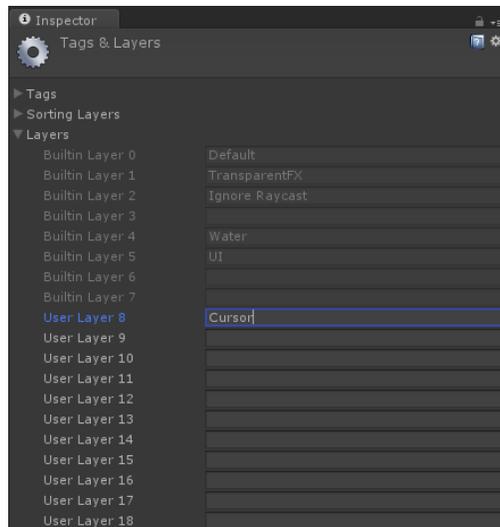


Ilustración 57 Nueva capa

Se deberá volver a seleccionar el cursor y nuevamente desplegar las opciones de Layer y seleccionar la nueva capa creada de cursor hará que la textura ya no sea visible (Ilustración 58).



Ilustración 58 Cámara con y sin capa

Capítulo 5. CONCLUSIONES

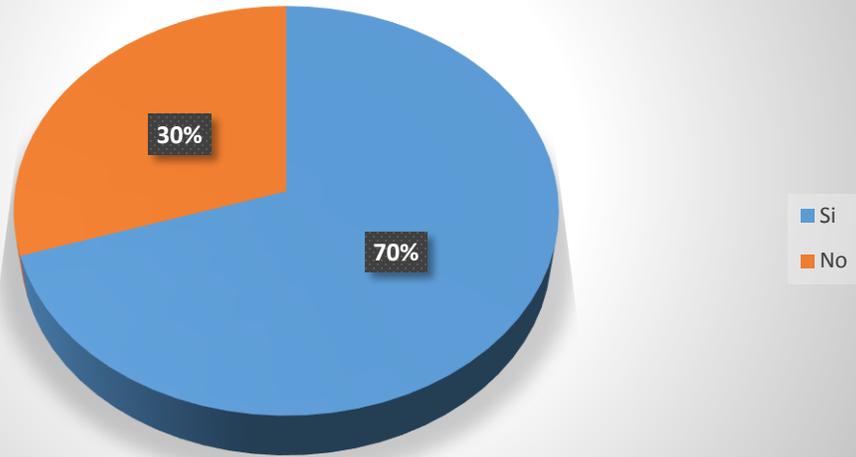
Obtener un mapa 3D en una serie de modelos que representan un campus universitario (en este caso de la Benemérita Universidad Autónoma de Puebla), con un sistema de navegación mediante un mundo en miniatura fue el objetivo general sobre el cual se trabajó. Se realizó un prototipo que muestra una sección concreta del mapa, pero también se encuentran las demás facultades disponibles para añadirse si fueran necesarias.

El objetivo con el que se diseñó el prototipo fue alcanzado satisfactoriamente, de la sección disponible para recorrer se pueden apreciar la ubicación actual del avatar, así como el entorno que le rodea, brindando así una ubicación parcial al usuario.

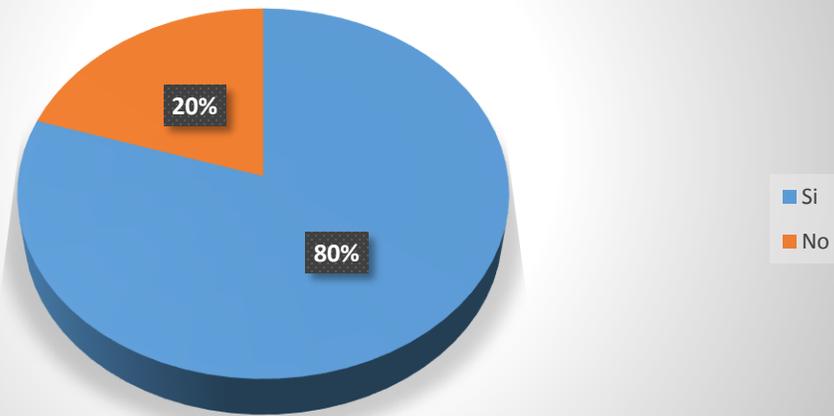
Se realizaron encuestas a un pequeño grupo de usuarios para recopilar información acerca de cuál era su opinión acerca del prototipo, tomando en cuenta la facilidad de uso, así como la utilidad y la opinión general del mismo. Las preguntas, para fines prácticos, en su mayoría están limitadas a preguntas que se puedan responder con sí o no, para así tener en claro la aprobación o desaprobación del mismo. A continuación se muestran los resultados de algunas preguntas.



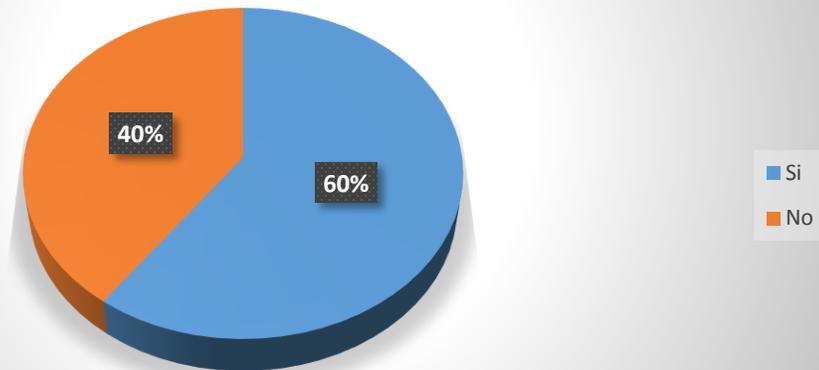
¿Te gusto la posibilidad de utilizar un mapa para navegar?



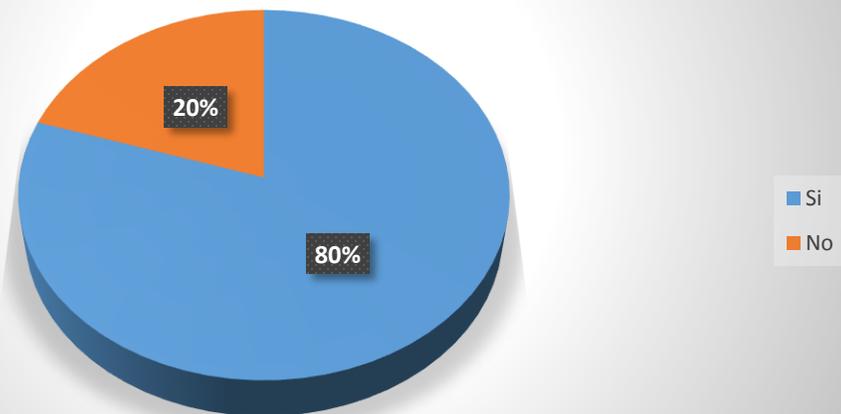
¿Te gusto la opción de poder mostrar/ocultar el mapa?



**¿Te gusto la precisión de la navegación?
(controles y sensibilidad de la camara)**



**¿Usarías esta aplicación para conocer la
universidad?**



Los resultados en general fueron buenos, mas también se debe tomar en cuenta que también existen detalles que se deben mejorar o cambiar para la facilidad del usuario, para que no importando el tipo de usuario, pueda hacer un uso correcto del prototipo, así como también le resulte de utilidad.

Además brindo nuevas opciones para ampliar el proyecto, como los son rutas para llegar de un punto a otro, la inclusión de movimiento automático de avatares predefinidos y automóviles, con el fin de que se sienta más vivo y brinde una experiencia más cercana a la realidad, e inclusive puntos de transporte, como por ejemplo las paradas del transporte interno escolar, que al ingresar a uno te lleve hasta su siguiente destino, con la opción de mostrar o no el transcurso de un punto a otro.

Tomando en cuenta lo anterior se puede concluir que se logró el objetivo inicial, más también abrió las puertas para implementar otras técnicas y complementos para la navegación en 3D que ayudaran a una mejor interacción con el usuario.

BIBLIOGRAFÍA

"Las mejores y más efectivas técnicas para hacer mapas conceptuales | eHow en Español", *eHow en Español*, 2012. [Online]. Available: http://www.ehowenespanol.com/mejores-mapas-3d-lista_311292/.

"iOS 9 - Maps - Apple", *Apple.com*, 2016. [Online]. Available: <https://www.apple.com/la/ios/maps/>.

"Omnidirectional treadmill", *Wikipedia*, 2016. [Online]. Available: http://en.wikipedia.org/wiki/Omnidirectional_treadmill.

Hale, K. S., & Stanney, K. M. (Eds.). (2002). *Handbook of virtual environments: Design, implementation, and applications*. CRC Press.

¹Schmalsteig, Dieter, L. Miguel Encarcacao, Zsolt Szalavari. "Using Transparent Props For Interaction with The Virtual Table.", In *Proceedings of the 1999 ACM Symposium on Interactive 3D Graphics*, 147-154, 1999.

Coquillart, S. and G. Wesche. "The Virtual Palette and the Virtual Remote Control Panel: A Device and Interaction Paradigm for the Responsive Workbench." *IEEE VR'99*, 213-217, 1999.

Stoakley, R., Conway, M. J., & Pausch, R. (1995, May). Virtual reality on a WIM: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 265-272). ACM Press/Addison-Wesley Publishing Co.

Ramírez, J. (2005, August). Guidelines for choosing VR devices from interaction techniques. In *Proceedings of the 5th WSEAS international conference on Multimedia, internet & video technologies* (pp. 46-51). World Scientific and Engineering Academy and Society (WSEAS).

Trueba, R., Andujar, C., & Argelaguet, F. (2009, January). Complexity and occlusion management for the world-in-miniature metaphor. In *Smart Graphics*(pp. 155-166). Springer Berlin Heidelberg

Chittaro, L., Ranon, R., & Ieronutti, L. (2009, November). 3D object arrangement for novice users: the effectiveness of combining a first-person and a map view. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology* (pp. 171-178). ACM..

Vallance, S., & Calder, P. (2001). Context in 3D planar navigation. *Australian Computer Science Communications*, 23(5), 93-99.