

DOMUM

Trabajo final

Taller de Diseño Multimedial V
Facultad de Bellas Artes | UNLP

González, Esteban José
Legajo: 70745/3

Descripción breve del proyecto

VDomum es un servicio destinado principalmente para agentes inmobiliarios con el fin de generar un recorrido 3D interactivo de un inmueble para que posteriormente los usuarios puedan explorarlo en su totalidad, interactuando con el entorno y obteniendo información durante el recorrido.

Busco ofrecer una nueva posibilidad para que los potenciales usuarios puedan conocer una propiedad que deseen alquilar o comprar sin la necesidad de moverse de su casa.

El recorrido está pensado principalmente para ser embebido en páginas web, utilizando la tecnología WebGL, ya que entendemos que las plataformas web son el entorno en donde más se mueven las inmobiliarias.

Sumado al soporte HTML5/WebGL, el recorrido es multiplataforma; soportando además PC, Mac, Linux y VR.

El proceso consta de los siguientes pasos:

- Recibimos los planos del inmueble
- **Realización de un trabajo de campo que consiste en:**
 - *(En el caso de no disponer de los planos)* Sacar medidas del lugar.
 - Registro fotográfico y de video para entender la espacialidad.
 - Registro fotográfico de alta calidad para generar las texturas que componen el inmueble (paredes, maderas, azulejos, etc).
- **Trabajo de software:**
 - Traducción del plano a una imagen bitmap, utilizando patrones preestablecidos para la generación del inmueble
 - Modelado de los objetos mobiliarios
 - Texturizado
 - Trabajo de iluminación
 - Reutilización de scripts en C# para la interacción del usuario con el entorno
 - Exportación y entrega al cliente

Punto de partida, investigación y antecedentes

El proyecto tiene como punto de partida la detección de la necesidad de que tanto potenciales inquilinos como compradores de un inmueble puedan recorrer el mismo sin la necesidad de moverse de su casa en una primera instancia.

Esto contempla desde grandes inmobiliarias que buscan explorar nuevas metodologías para el muestreo de sus inmuebles, así como rental managers que utilizan plataformas de short-term rentals como Airbnb, Homeaway o Expedia; los cuales buscan mejorar la administración de sus tiempos y potenciar, al igual que los agentes inmobiliarios, a brindar un panorama claro de los inmuebles que ofrecen.

De ahí surge el trabajo de investigación, focalizado en las tecnologías que actualmente existen en el mercado y detectar cuál es el diferencial que VDomum puede ofrecer para ser un servicio rentable.

Dentro de las tecnologías existentes, destaco y analizo dos:

Recorridos virtuales 360°

Estos recorridos poseen una alta fidelidad que parte de la esquematización del espacio del inmueble por intermedio de cámaras 360° que toman fotografías desde diferentes puntos de la locación para luego ser ensambladas todas en conjunto y finalmente, generar la interacción “point & click”.

La fortaleza de esta tecnología es su rendimiento, ya que no requiere de una buena computadora para ser utilizados por su óptimo rendimiento, teniendo en cuenta que ya es una tecnología que se utiliza desde hace mucho tiempo.

El principal limitante radica en que la interacción es extremadamente acotada, ya que hay puntos prefijados donde el usuario puede ubicarse. La iluminación y la percepción del mismo se limita a las fotografías sacadas en ese preciso momento y se podrían tomar como otra forma de ver una galería de imágenes o un video recorrido del lugar, algo que en la actualidad no se puede percibir como un producto diferencial.

Ejemplos de uso en la actualidad

[Kuula](#) - Servicio ofrecido para la creación de estos recorridos.

[ZonaProp](#) - Algunos inmuebles de esta plataforma disponen de estos recorridos, integrados en la publicación de las propiedades.

[Visita 360°](#) - Servicio ofrecido para la creación de los mismos.

Exceptuando ZonaProp, todos los casos anteriores se plantean como un servicio ofrecido, generalmente destinado a inmobiliarias que buscan explotar estas tecnologías para mejorar la experiencia de sus clientes.

Archviz for Unreal Engine 4

Una tecnología en auge, que destaca por su nivel de realismo, partiendo del trabajo de texturas, el modelado, la iluminación e interacción que se asemeja mucho al de la vida real. La principal fortaleza que lo desprende de los recorridos 360° es que el nivel de interacción es mucho mayor, el espacio dispone de una iluminación dinámica y en la mayoría de los casos, el usuario puede recorrer todo el inmueble sin limitantes.

El principal problema es el rendimiento y en las plataformas para las cuales está pensado. Sólo soporta plataformas como Windows o Mac y su extensión a VR (Oculus) y requieren de tarjetas gráficas muy potentes y costosas. Este problema se enfatiza aún más entendiendo el contexto de uso de estos recorridos, donde el campo de acción principal de los usuarios a los que apuntan se da en sitios web, y en este aspecto Archviz se vuelve incompatible por los requerimientos que demanda.

Ejemplos de uso en la actualidad

[Evermotion](#) [Substance3D](#) [Trinity3D](#)- Sitios web donde se venden este tipos de recorridos con fines exploratorios. No está pensado como un servicio ya que no tienen la misma finalidad que un recorrido pensado para un cliente en particular, sino más bien para otros desarrolladores e investigadores de esta tecnología.

[Unreal Engine marketplace](#) - Con una finalidad similar a la anterior. Es un marketplace donde uno puede descargar los proyectos de estos recorridos para utilizarlos en el motor gráfico Unreal Engine (en el cual se desarrollan).

Análisis de lo investigado

Basándome en la investigación previa, decidí analizar las ventajas y desventajas que poseen cada uno de los productos y servicios que existen hoy en día con mayor detalle, y ver de qué manera puedo ofrecer un servicio diferencial, enfocado en el mercado objetivo que me propuse, explotando las oportunidades que se me presenten.

Recorridos 360°

Ventajas

1. Excelente rendimiento
2. Multiplataforma
3. Bajo costo de producción (con una inversión inicial moderada)
4. Accesible en términos de costos para muchos clientes.

Desventajas

1. Interacción pobre
2. Sólo se limita a lo visual
3. Baja proyección por ser un producto “acotado” en cuanto a las posibilidades que se les ofrece a los usuarios a la hora del recorrido.
4. Al utilizar recursos accesibles por cualquier persona para su creación, la demanda se achica ya que la oferta aumenta.
5. Entorno estático.

Archviz for Unreal Engine 4

Ventajas

1. Mayor inmersión.
2. Alto nivel de realismo.
3. Entorno dinámico.
4. Capacidad de reutilización de componentes para otros recorridos.
5. Motor gráfico gratuito.

Desventajas

1. Plataformas acotadas y sólo en entornos offline.
2. Bajo rendimiento ya que exige una buena placa de video, procesador, ram entre otros componentes.
3. Resigna interacción más “humana” para brindar mayor realismo en el apartado gráfico.
4. Requiere de personal capacitado para su realización.
5. Actualmente está enfocado para fines de investigación y exploración, sin tener un mercado definido activo.
6. Demanda mucho tiempo para su elaboración.

Hallazgos y puntos a trabajar

En relación a las ventajas y desventajas de cada una de las tecnologías existentes, comencé a hacer una bajada de los puntos a mejorar para que el resultado sea superador en términos del producto final que se le entregará al cliente y la relación costo/beneficio de su producción.

Para esto en principio traduje las desventajas de la competencia para transformarlas en potenciales ventajas para mi servicio y luego, basándome en una Matriz FODA, entender qué aspectos de mi servicio posee debilidades que en un futuro puedo fortalecer.

1. Ecosistema web y ámbitos multiplataforma

El primer aspecto a considerar es el contexto en el cual se manejan mis potenciales clientes. Entendí que un factor decisivo es que el producto final sea compatible dentro de un ecosistema web, sin dejar de lado la posibilidad de poder ser utilizado en otras plataformas.

2. Rendimiento sin resignar aspectos gráficos

Otro de los ejes centrales de mi servicio era que al tener como condicionante el ecosistema web, el producto debía tener un rendimiento acorde para que un browser (Chrome, Mozilla) lo pueda ejecutar, sin dejar de lado el nivel de realismo deseado para no quedar en desventaja con Archviz.

3. Niveles de interacción

Si bien priorizar el rendimiento indefectiblemente hace que el apartado gráfico deba disminuir en parte, si lograba generar una interacción más lograda (que no sólo se limite a un recorrido en sí, sino que también el usuario pueda interactuar con el entorno), esto iba a aportar valor al producto final.

4. Agilizar el proceso de producción

Entendí desde el momento cero que este era un desafío grande, ya que para poder llevarlo a cabo se requiere de la exploración de varios campos del conocimiento que abarcan: Programación, modelado 3D, texturizado, trabajos de iluminación y todo el trabajo de campo previo a la ejecución.

Algunos de estos aspectos se pueden automatizar, otros tercerizar; siempre y cuando la relación costo/beneficio sea adecuada.

5. Aprovechar el auge, la investigación y la exploración como motor para fortalecer en un futuro el producto

Estas tecnologías están mutando constantemente, y cosas que antes eran impensadas hoy se pueden realizar. El desarrollar este servicio y poder tangibilizarlo en algo concreto permite no sólo perfeccionarlo por cuenta propia, sino también poder generar un interés que permita obtener inversiones, conseguir gente capacitada que se suma al proyecto y pueda hacer crecer aún más el servicio, y por qué no; en un futuro desarrollar una herramienta pensada para que cualquier persona pueda utilizar.

Definición

Ecosistema web y ámbitos multiplataforma: HTML5 y WebGL 2.0

Este fue el disparador que me llevó a utilizar cómo base del proyecto el motor gráfico gratuito Unity3D.



El mayor diferencial con su competidor Unreal Engine 4, es que Unity dispone de exportación para WebGL 2.0, que es la tecnología actual que se utiliza para correr aplicaciones en un entorno web, con lo cual mi proyecto base debía comenzar por ahí.

Rendimiento sin resignar aspectos gráficos:

- Unity3D cómo motor gráfico
- Post Processing Effect
- Baked illumination
- PBR textures

Unity3D destaca por el nivel de rendimiento de las aplicaciones que genera, ya que la lógica del motor gráfico opera de manera distinta a su directo competidor Unreal 4, que posee una calidad gráfica superior, resignando parte del rendimiento.

Realizando una investigación me topé con una tecnología recientemente incorporada llamada **Post Processing Effect**, un script (código en lenguaje C#) que se vincula a la cámara que simula la visión del usuario y permite regular aspectos gráficos cómo el Antialiasing, ambient occlusion, motion blur, color grading, eye adaption, entre otros que sirven para aumentar el realismo.



Con este script, Unity se posiciona a un nivel muy similar a Unreal que ya tiene integrado en su motor base estas características, pero la diferencia es que las posibilidades de configuración que dispone Unity son ampliamente superadoras, con lo cual la relación de rendimiento - calidad final se vuelve mucho más maleable, contemplando los diferentes dispositivos y entornos en donde se va a ejecutar el recorrido planteado, para que el recorrido sea lo más accesible posible.

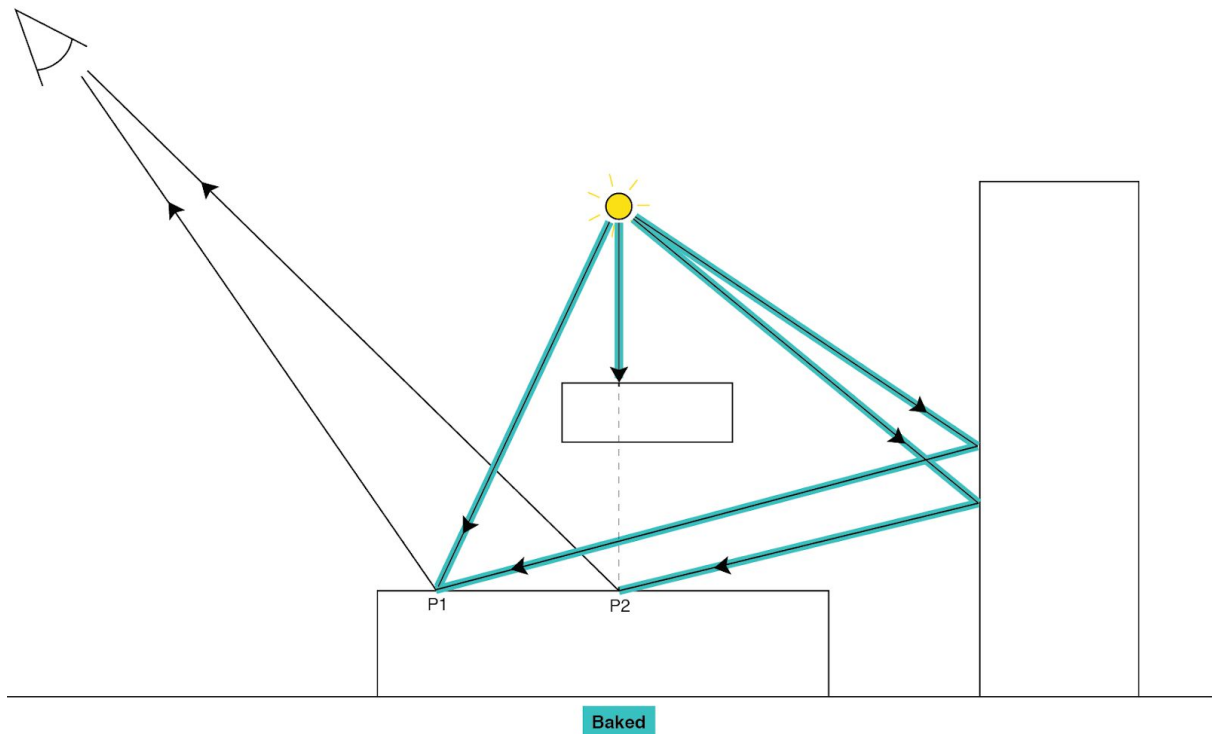
Baked Illumination & Reflection Probes

Otro aspecto importante es la iluminación y renderizado del entorno.

Hasta hace no mucho tiempo, el renderizado siempre era en tiempo real con lo cual era indispensable que el dispositivo que corra la aplicación sea potente, ya que la tarjeta gráfica y el procesador estaban constantemente renderizando el entorno en tiempo de ejecución.

La **iluminación Baked** lo que permite es que uno pueda definir qué objetos del entorno son estáticos y cuáles no y por intermedio del **Global Illumination** (la solar) y los **reflection probes** (que se encargan de obtener y almacenar la información reflectiva en un espacio circundante en relación a los objetos estáticos y sus respectivas propiedades reflectantes), el motor gráfico renderiza por única vez (no en tiempo de ejecución) todo el entorno, con la posibilidad de aumentar y disminuir la calidad final del renderizado, y sólo renderiza en tiempo real aquellos objetos y fuentes de iluminación que uno definió como **Real-time lighting** en el proyecto.

De esta manera, el rendimiento mejora ampliamente sin perder absolutamente nada de realismo, ya que la mayoría de los objetos que componen un inmueble son estáticos.

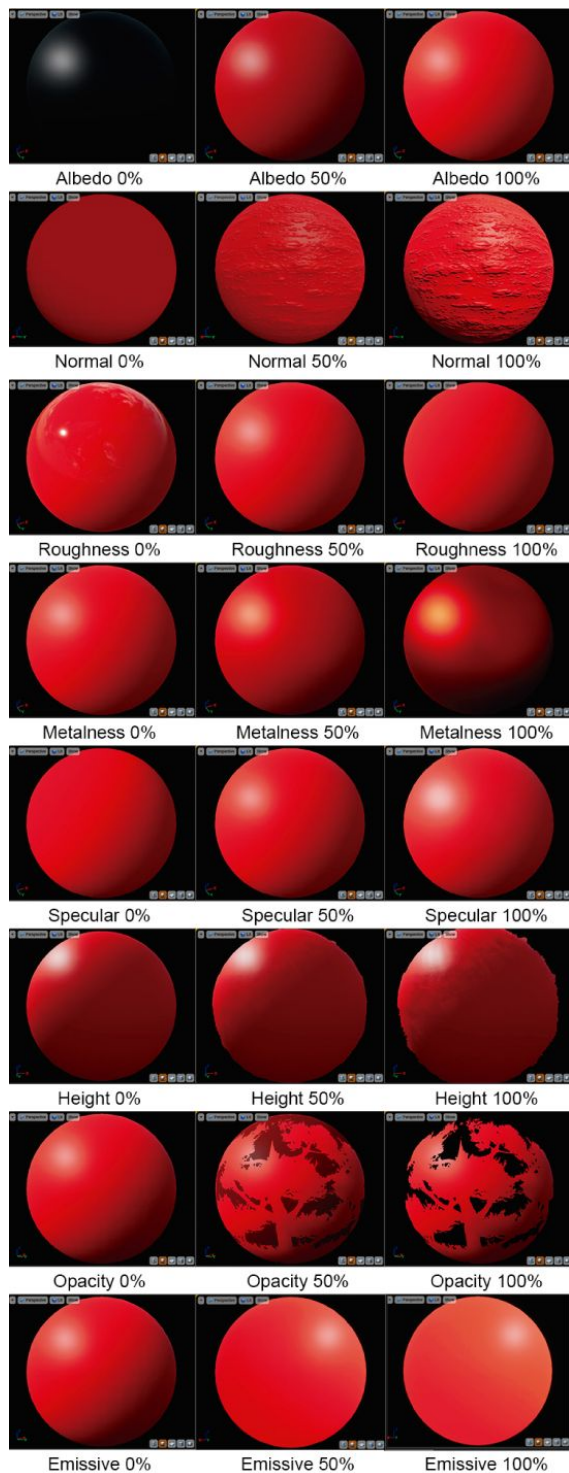


La información lumínica Baked (indicada en celeste) se guarda al ser renderizada en texturas que se aplican a los objetos estáticos.

Physically based rendering (PBR Textures)

PBR es una técnica que se utiliza para generar resultados realistas, basado en las propiedades físicas de los materiales y la luz. De esta manera, se pueden realizar cambios **dinámicos** de iluminación y que los materiales se muestren tal cual son.

A diferencia de las texturas convencionales, las texturas PBR se conforman de diversas texturas que en su conjunto forman la textura final, considerando aspectos que se dan en la vida real de cómo percibimos los objetos, cómo la **reflectividad** y su relación con la **difusión** (*mayor reflectividad, menor difusión y viceversa*), la **conductividad** y los **aislantes** y la **rugosidad**, entre otras.



Niveles de interacción: El entorno como parte de la experiencia

Ya mencioné la interacción acotada de los recorridos 360°, y en el caso de Archviz; muchos de los recorridos están pensados para ser interactuados como una cámara cinematográfica (similar a ver un render en un programa de modelado 3D), ya que complejizar la interacción del usuario hace que el rendimiento disminuya.

En principio planteo un recorrido más inmersivo, en donde el usuario no sólo recorre el inmueble, sino que también tiene la posibilidad de:

- Interactuar con interruptores para ver la iluminación de los diferentes espacios.
- Interacción con puertas y ventanas.
- Visualización de las especificaciones del mobiliario con el que se cuenta, con la posibilidad de redireccionar mediante un link a una ficha del producto para comprarlo si el usuario quisiera.
- Dependiendo de la complejidad de los modelos 3D, se puede interactuar con heladeras, alacenas, etc.
- Utilizar el recorrido para publicitar el inmueble u partners asociados para poder monetizar el producto.

Todas estas interacciones se plantean dentro de scripts ya creados, en donde uno debe simplemente arrastrar ese código a los objetos y configurar las variables dependiendo de la interacción que se desee. Todo de manera visual, sin necesidad de programar.

Agilizar el proceso de producción:

- Plano real a imagen de bitmaps
- Generador de inmueble
- Modularización

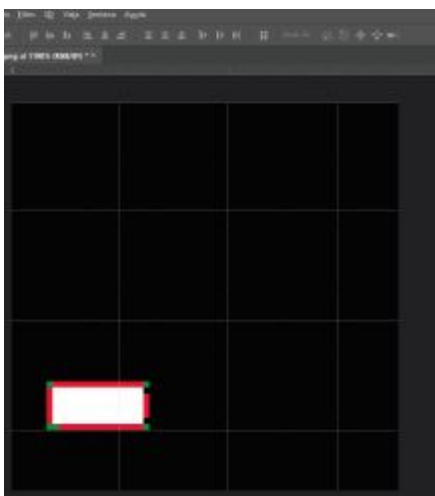
Plano real a imagen de bitmaps, modularización y la generación base del inmueble.

El primer paso del proceso de creación fue ver de qué manera se podía automatizar el traspaso de la información de un plano de un inmueble a un modelo 3D, sin tener que diseñarlo a mano.

Lo que hice fue detectar diversos patrones y traducirlos a un medio que me permita generar la base del inmueble de forma automática.

Para esto cree un script que lo que hace es recorrer una imagen de bitmap (matriz de píxeles), leer pixel por pixel e interpretar esa información de color y traducirla a un objeto modular pre-cargado en un arreglo de objetos. Dependiendo del color de los píxeles aledaños, el script interpreta qué grado de rotación debe aplicar sobre cada módulo, para que el ensamblado final no tenga errores.

El script da la posibilidad de ajustar las escalas de cada pixel, para hacer coincidir la dimensión real del plano a cada uno de los píxeles.

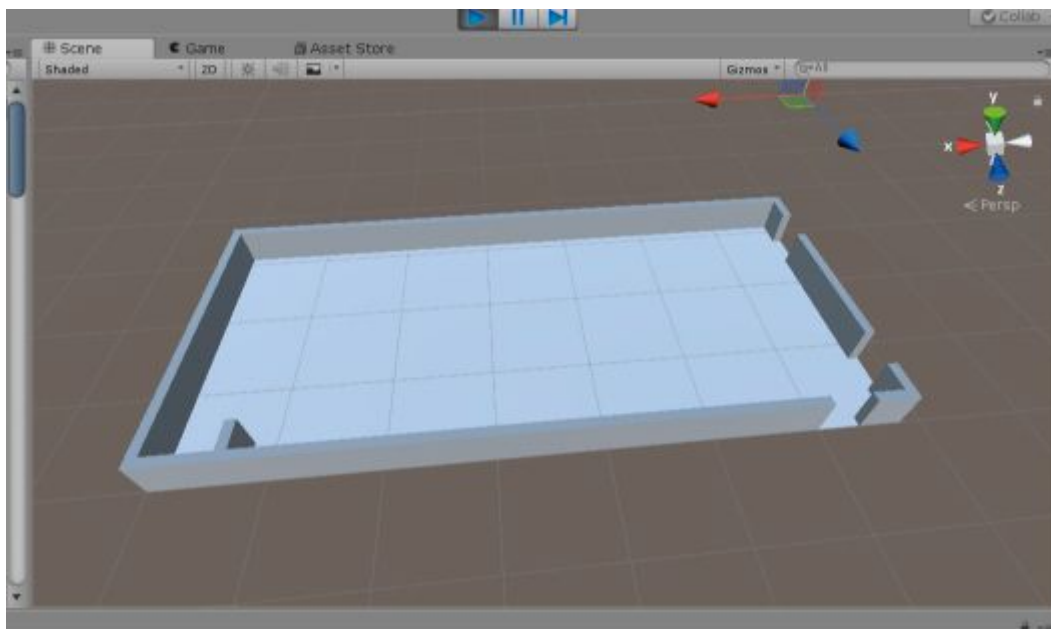


En este ejemplo simple, genero un plano rectangular básico, donde las paredes son representadas por el color rojo, las esquinas con el verde, el piso blanco y el negro es vacío (el script interpreta que no hay un objeto asociado en esas posiciones)

Así cómo utilizo estos patrones de color, también se puede utilizar el azul para las puertas, el cyan para las ventanas, el magenta para otro tipo de abertura, esquina o formato de pared. Las posibilidades son infinitas.

Cuando esta imagen está creada, desde Unity la cargo en el script y antes de ejecutarlo, añado los objetos modulares pre-modelados a un arreglo, para efectuar la asociación **color = objeto modular 3D**.

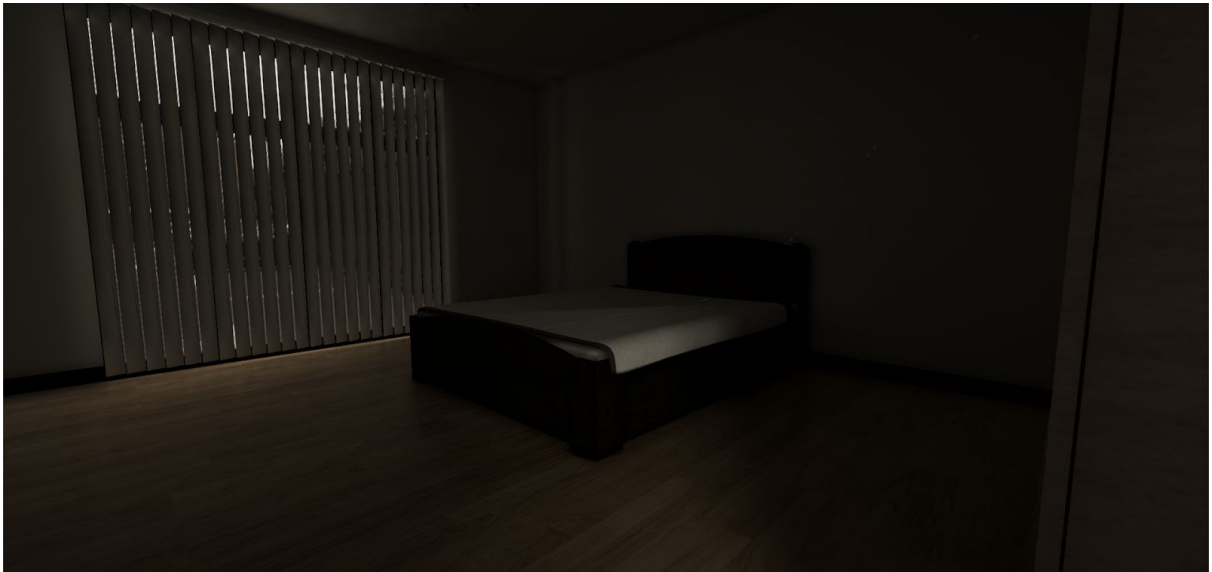
El resultado particular de esta imagen (considerando que los objetos modulares no tienen texturas aplicadas) es el siguiente



Este algoritmo agiliza mucho el proceso de creación de un inmueble, ya que si uno toma la totalidad del mismo como un conjunto de módulos, destina menos tiempo en el armado y diseño. Por otro lado, estos módulos son reutilizables para diversos proyectos, disminuyendo costos implicados al tiempo en producciones futuras.

Así como la estructura base parte de módulos, el mobiliario que se añade después está pensado con una lógica similar. Son paquetes que contienen no sólo el modelo y las texturas, sino que también tienen incluidos los script pertinentes que proveen las diferentes interacciones permitidas dependiendo del objeto en sí, al igual que sus fuentes de luces si es que las emiten.

El resultado del prototipo acabado, es el siguiente:





Aprovechar el auge, la investigación y la exploración como motor para fortalecer en un futuro el producto

El último aspecto a considerar es el alcance actual del proyecto y la proyección del mismo de cara al futuro.

Todo este trabajo fue llevado a cabo por una sola persona, entendiendo los recursos limitados por no disponer de una financiación y equipo de trabajo para dividir las tareas y agilizar los tiempos de producción.

¿Cuales son los aspectos a mejorar a futuro?

- Contar con una financiación para profundizar el desarrollo.
- Contar con un equipo de trabajo especializado en cada una de las diferentes áreas.
- Lograr que VDomum pueda ser un producto pensado para ser utilizado por personas que no tengan un enorme conocimiento sobre estas tecnologías.
- Seguir refinando aspectos del algoritmo central para generar estructuras mucho más complejas e incluir en esta automatización, los objetos mobiliarios.

- Lograr que el algoritmo pueda generar la estructura desde la lectura directa de un plano real, sin tener que generar una imagen bitmap

Links de interés

Post processing effect

<https://docs.unity3d.com/2018.3/Documentation/Manual/PostProcessing-Stack.html>

Reflection Probes

<https://docs.unity3d.com/es/2017.4/Manual/class-ReflectionProbe.html>

Real-time lighting

<https://docs.unity3d.com/es/2017.4/Manual/LightMode-Realtime.html>

Global Illumination

<https://docs.unity3d.com/es/530/Manual/GIIntro.html>

Baked Illumination

<https://docs.unity3d.com/es/2017.4/Manual/LightMode-Baked.html>

PBR Textures

<https://conceptartempire.com/texture-maps/>