

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**  
**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN MÔN PHÁT TRIỂN TRÒ CHƠI**

# **TÌM HIỂU CẤU TRÚC, CÁCH SỬ DỤNG CỦA 1 GAME ENGINE CHO GAME 3D.**

*Người hướng dẫn: Thầy Vũ Đình Hồng*

*Người thực hiện:*

**NGUYỄN KHÁNH MINH - 51900132**

**HOÀNG NHẬT TÂN - 51900206**

**NGUYỄN XUÂN THANH NHÀN - 51800455**

**NGUYỄN HOÀNG HƯNG - 51800777**

**MẠC TRUNG TÍN - 51801039**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020**

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**  
**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN MÔN PHÁT TRIỂN TRÒ CHƠI**

# **TÌM HIỂU CẤU TRÚC, CÁCH SỬ DỤNG CỦA 1 GAME ENGINE CHO GAME 3D.**

*Người hướng dẫn: Thầy Vũ Đình Hồng*

*Người thực hiện:*

**NGUYỄN KHÁNH MINH – 51900132**

**HOÀNG NHẬT TÂN – 51900206**

**NGUYỄN XUÂN THANH NHÀN - 51800455**

**NGUYỄN HOÀNG HƯNG - 51800777**

**MẠC TRUNG TÍN - 51801039**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020**

## **LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn thầy Vũ Đình Hồng. Thầy là giảng viên dạy chúng em môn Phát triển trò chơi. Thầy đã hướng dẫn và giúp đỡ chúng em hoàn thành bài tập lớn này này.

## **ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm bài tập lớn của riêng chúng tôi và được sự hướng dẫn của thầy Vũ Đình Hồng;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 30 tháng 11 năm 2020*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Khánh Minh*

*Hoàng Nhật Tân*

*Nguyễn Xuân Thanh Nhân*

*Nguyễn Hoàng Hưng*

*Mạc Trung Tín*

## MỤC LỤC

LỜI CẢM ƠN .....	I
MỤC LỤC .....	1
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ .....	2
PHẦN 1 – GIỚI THIỆU ĐỀ TÀI .....	3
1.1 Trình bày về đề tài.....	3
1.2 Tại sao cần thực hiện đề tài .....	5
1.3 Yêu cầu của đề tài cần thực hiện.....	5
PHẦN 2 – NỘI DUNG TÌM HIỂU .....	6
2.1 Cấu trúc của 1 game engine cho game 3D .....	6
2.2 Cách sử dụng của 1 game engine cho game 3D .....	8
PHẦN 3 – DEMO .....	17
3.1 NỘI DUNG DEMO.....	17
3.2 HƯỚNG DẪN CHẠY DEMO .....	17
TÀI LIỆU THAM KHẢO .....	18

## **DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ**

### **DANH MỤC HÌNH**

Hình 1.....	7
Hình 2.....	8
Hình 3.....	10
Hình 4.....	11
Hình 5.....	12
Hình 6.....	13
Hình 7.....	14
Hình 8.....	15
Hình 9.....	16

### **DANH MỤC BẢNG**

## PHẦN 1 – GIỚI THIỆU ĐỀ TÀI

### 1.1 Trình bày về đề tài

#### *a/ Game Engine là gì?*

Game engine là tập hợp mã dưới dạng thư viện và khuôn khổ.

Thư viện là **tập hợp mã** tập trung vào việc cung cấp giải pháp cho các vấn đề cụ thể. Bạn có thể đã nghe nói về các tệp “.dll” của chương trình — từ viết tắt đó là viết tắt của “thư viện được liên kết động” và đó chính xác là loại thư viện mà chúng ta đang nói đến. Nếu bạn có một vấn đề cần giải quyết hoặc nhiệm vụ cần hoàn thành, có một thư viện cho điều đó. Một trong những thư viện được sử dụng rộng rãi và phổ biến nhất hiện nay là **OpenGL** (Open Graphics Library). Như tên của nó, nó tập trung vào việc cung cấp một tiêu chuẩn đồ họa để sử dụng cho nhiều nền tảng khác nhau.

**Các khung công tác** là các bộ sưu tập mã cấp cao hơn các thư viện. Chúng thường kết hợp nhiều thư viện và các thành phần chi tiết khác để tấn công các vấn đề khác nhau cùng một lúc, hướng chúng đến việc tạo ra một trải nghiệm cụ thể. Đây là nơi chúng ta bắt đầu thấy động cơ proto xuất hiện từ quả trứng tục ngữ. Một khuôn khổ phổ biến là **.NET Framework của Microsoft**, cung cấp cho lập trình viên các công cụ kỹ thuật số để xử lý mọi thứ, từ truy vấn cơ sở dữ liệu nội dòng đến mã đa luồng, không đồng bộ.

#### *b/ Game Engine hoạt động như thế nào?*

Tất cả các thư viện và khuôn khổ này hợp tác với nhau và kết quả là trò chơi bạn nhìn thấy trên màn hình, nghe qua loa và thao tác với đầu vào của bạn. Mỗi thành phần của engine đều giao tiếp giữa phần cứng, hệ điều hành và chính engine để tạo ra một kết quả. Những kết quả này tạo ra một trò chơi từ Tether kỹ thuật số.

Có hàng trăm, nếu không phải hàng nghìn, thư viện và khuôn khổ đi vào các trò chơi yêu thích của bạn và công cụ của chúng – và đó là chưa nói gì đến các thành phần

được phát triển rõ ràng cho các trò chơi riêng lẻ. Đúng vậy: ngay cả một Engine cũng không thể bao phủ tất cả các căn cứ.

### *c/ Game Engine phổ biến*

**Unity** là một trong những công cụ trò chơi phổ biến ngày nay, đặc biệt là trong lĩnh vực độc lập. Trong số các nhà phát triển AAA, **Unreal Engine** phổ biến hơn, với độ trung thực hình ảnh cao hơn và quy trình làm việc mạnh mẽ hơn giúp tận dụng tối đa một nhóm phát triển lớn. Nếu bạn là fan của Bethesda, chắc chắn bạn đã từng gặp phải engine Gamebryo, cung cấp năng lượng cho các tựa game như Morrowind, Oblivion và Fallout 3. Source engine là đứa con của Valve, cung cấp năng lượng cho mọi thứ từ Half-Life 2 đến Portal 2 và Left 4 Dead.

Đối với Unity, công cụ phát triển chính là Unity Editor và nó được trang bị để nhập và làm việc với các nội dung từ nhiều định dạng tệp tiêu chuẩn. Tất nhiên, Unreal Editor dành cho Unreal Engine và hoạt động theo cách tương tự. Một số công cụ như Gamebryo và Nguồn không có trình chỉnh sửa toàn diện, họ chọn cung cấp các công cụ độc quyền chỉ dành cho thiết kế cấp độ – Bộ công cụ xây dựng và Trình chỉnh sửa búa van, tương ứng. Trong mọi trường hợp, bạn có thể sẽ tạo nội dung từ bên ngoài, sau đó chuyển đổi và nhập nội dung bằng SDK. Tuy nhiên, các trình chỉnh sửa có thể mở rộng hiện đại cho các công cụ như Unity và Unreal đang thấy một sự thúc đẩy hướng tới việc cho phép tạo một số nội dung nhất định ngay trong trình chỉnh sửa.

Các tính năng như **Đồ thị Shader** của Unity và **Trình chỉnh sửa Niagara** của Unreal – cho phép tạo các trình tạo bóng và hiệu ứng hình ảnh tương ứng – có nghĩa là số lượng các chương trình khác nhau bạn cần trong quy trình phát triển của mình đang giảm dần theo thời gian. Nhưng một số thiết kế đặc biệt và được thiết kế tốt đến mức không thể thay thế chúng bằng một biến thể tổng quát.



## **1.2 Tại sao cần thực hiện đề tài**

- Các game thường có quy mô lớn
- Game engine sử dụng để phát triển nhiều game khác nhau
- Sử dụng game engine có thể phát triển game nhanh hơn
- Chi phí phát triển game thấp hơn
- Cấu trúc tốt hơn → Dễ quản lý hơn

## **1.3 Yêu cầu của đề tài cần thực hiện**

- Tìm hiểu cấu trúc, cách sử dụng của 1 game engine cho game 3D.

## PHẦN 2 – NỘI DUNG TÌM HIỂU

### 2.1 Cấu trúc của 1 game engine cho game 3D

#### *a/ Unreal Engine là gì?*

Unreal Engine là một công cụ được tạo ra bởi các chuyên gia nhằm đơn giản hóa việc làm game nhưng vẫn giữ được tính chuyên nghiệp, hiệu suất và đầy đủ tính năng, UE thậm chí còn không cần code, để làm game bạn chỉ cần có một sự kiên nhẫn và táy máy để tìm hiểu thay vì phải có một lượng kiến thức chuyên sâu. Bên cạnh đó, Unreal Engine cũng hỗ trợ hai ngôn ngữ lập trình chính là C++ và Python cho những chuyên gia muốn đi sâu vào nó.

Nhà phát triển cũng rất chú ý đến trải nghiệm của người dùng, cũng để nhằm bắt kịp những xu hướng mới nhất, dẫn đầu về công nghệ, các phiên bản được cập nhật định kỳ nhằm tăng thêm tính năng, cải thiện hiệu suất cho người dùng.

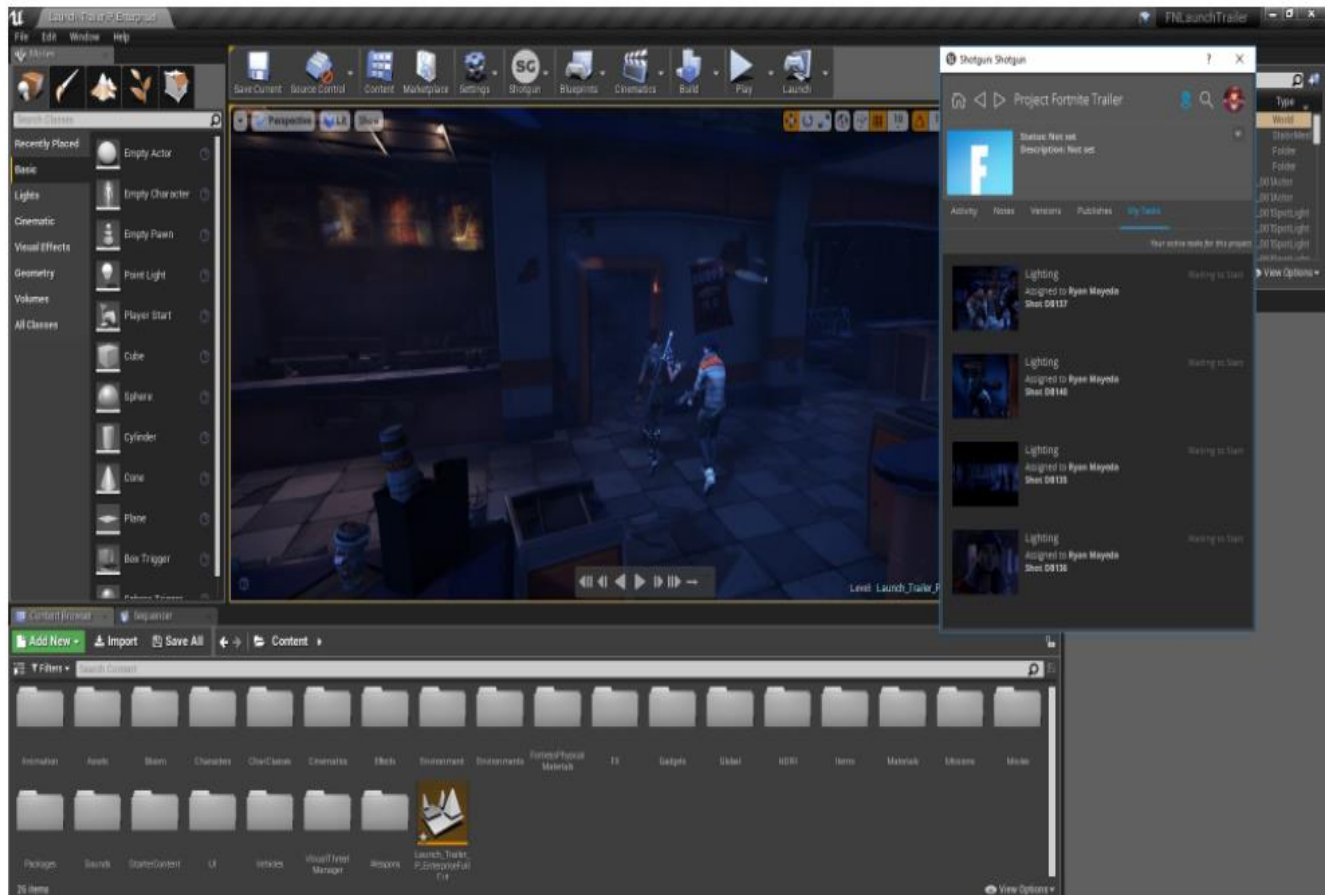
#### *b/ Tính năng của Unreal Engine*

Unreal Engine mang lại trải nghiệm đỉnh cao, mượt mà và dễ hiểu, bằng hình thức kéo thả, chỉnh sửa các vật thể, UE mang lại sự liền mạch và logic trong trải nghiệm làm game. Ngoài ra, còn có nhiều tính năng như:

- **(Mobile)** Từ những tựa Game 2D đơn giản cho đến những tựa Game 3D với đồ họa đẹp. Unreal Engine đưa đến cho bạn đủ sức mạnh để lập trình Game trên các hệ máy iOS và Android.
- **(Blueprints)** Blueprints Visual Scripting cho phép bạn nhanh chóng thử nghiệm và xuất ra một tựa Game hoàn chỉnh một cách nhanh chóng mà không yêu cầu phải biết lập trình.
- **(Tools)** Trình chỉnh sửa Unreal là một bộ công cụ đầy đủ để giúp bạn xây dựng hoàn hảo mọi khía cạnh của Project. Unreal Engine 4 còn được tích hợp những tính năng nâng cao

nếu: physically-based rendering, UI, level building, animation, visual effects, physics, networking, và asset management.

- **(Source Code)** Mọi nhà phát triển Game sử dụng Unreal Engine đều được sử dụng trình chỉnh sửa mã nguồn C++ (C++ Editor Source Code). Unreal Engine 4 cung cấp cho bạn toàn bộ sức mạnh của ngôn ngữ lập trình Unreal Engine 4 để bạn có thể tùy chỉnh và gỡ lỗi cho tựa Game của mình.
- **(Marketplace)** Bạn có thể tăng tốc tiến độ lập trình và phát triển một Game bằng những dự án mẫu và s, art, audio, blueprint logic sẵn có thậm chí là những dòng code C++ đã được viết trước.

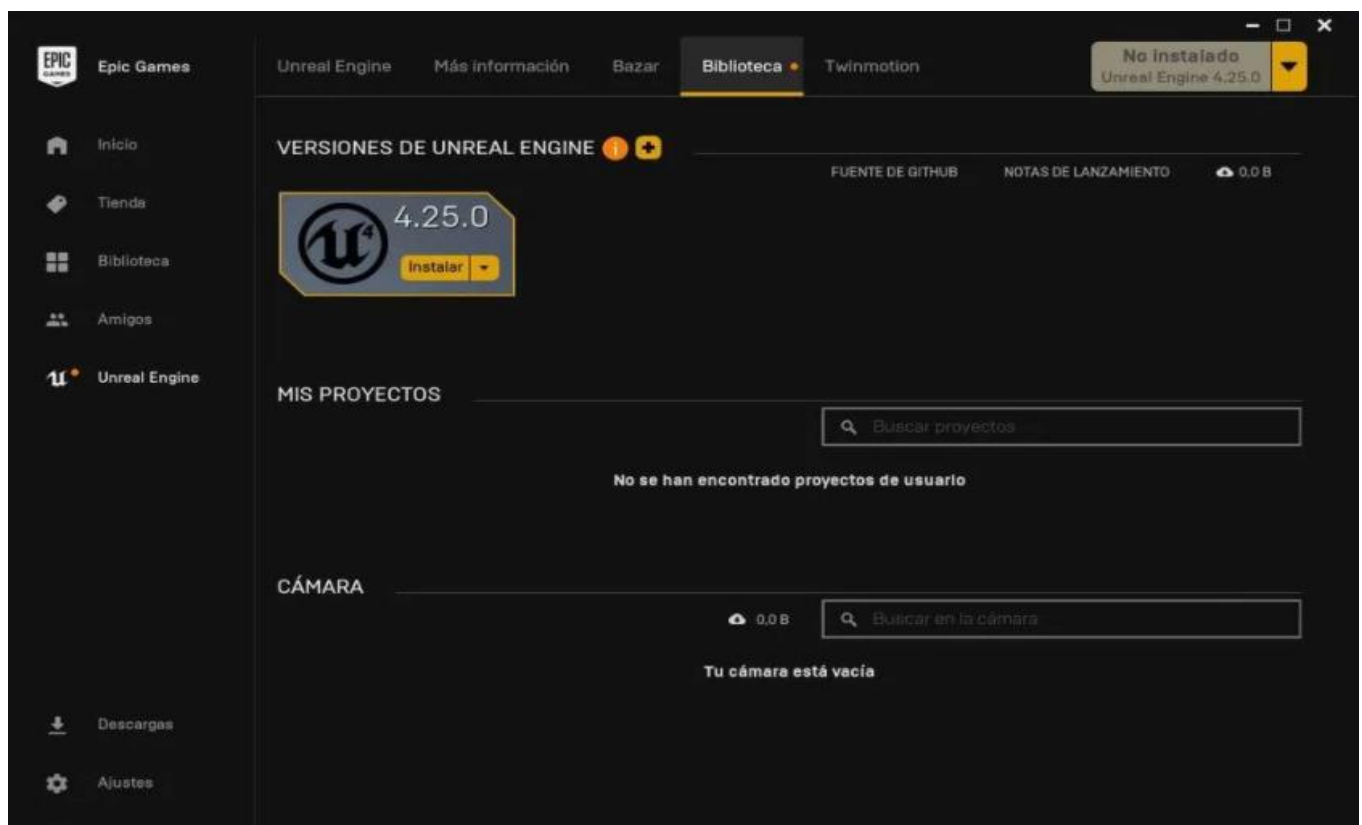


## 2.2 Cách sử dụng của 1 game engine cho game 3D

### *a/ Download Unreal Engine*

Cách tốt nhất để tải xuống công cụ này và để có thể kiểm soát tất cả các dự án của chúng tôi và luôn có phiên bản mới nhất, là thực hiện từ **Trình khởi động Epic Games**. Và cũng phải có một tài khoản người dùng để có thể đăng nhập và truy cập vào tất cả các tài nguyên được cung cấp bởi nền tảng này.

Trong menu bên trái, chúng ta sẽ phải chọn Unreal Engine bạn có thể tải miễn phí phiên bản mới nhất về máy tính.



Sau khi tải xuống và cài đặt Unreal Engine, chúng ta có thể mở nó từ trình khởi chạy Epic Games. Và bây giờ chúng ta chỉ có thể bắt đầu biến trí tưởng tượng của mình thành hiện thực.

### ***b/ Cần biết gì để sử dụng Unreal Engine***

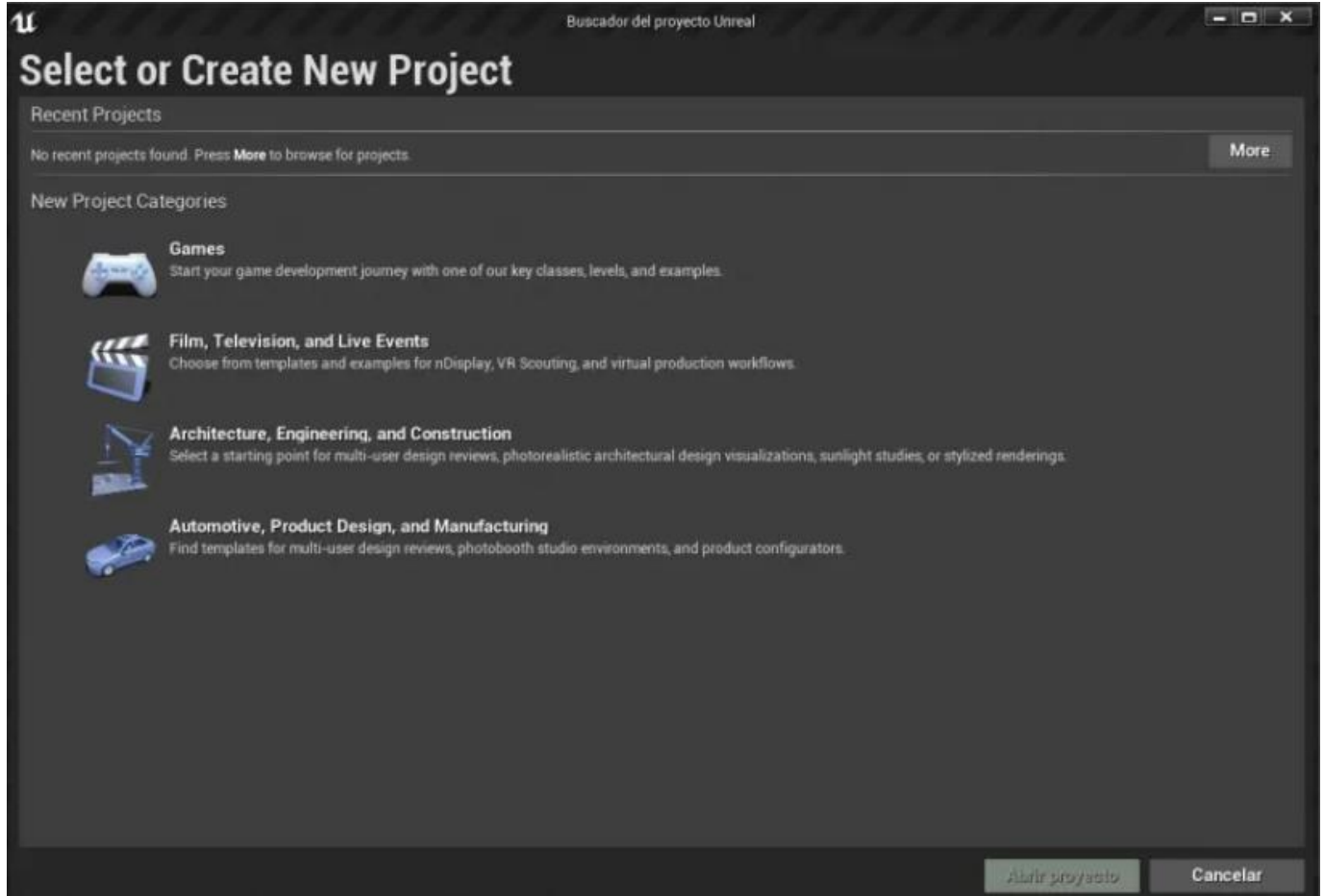
Điều đầu tiên chúng ta cần, rất quan trọng, là thời gian. Đường cong học tập về UE khá dốc. Do đó, chúng ta sẽ phải dành nhiều thời gian để làm quen với giao diện của nó và tất cả các chức năng mà nó cung cấp cho chúng ta nếu chúng ta thực sự muốn bắt đầu lập trình trò chơi điện tử với engine này. Nếu có thể, chúng tôi khuyên bạn nên bắt đầu với một khóa học Unreal Engine, ngay cả khi chúng ta đã có kiến thức trước đó về các ngôn ngữ hoặc công cụ khác, để tự làm quen với nó. Tài liệu là một yếu tố mà chúng ta luôn phải có trong tay.

Ngoài ra, mặc dù chúng ta có thể làm rất nhiều thứ từ giao diện của chương trình, nhưng có những thứ mà chúng ta chắc chắn phải lập trình bằng tay. Và ngôn ngữ duy nhất mà chúng tôi có thể sử dụng cho mục đích này là C ++ , vì trình thông dịch của nó không hỗ trợ các ngôn ngữ khác. Nếu chúng ta không biết cách lập trình bằng C ++ về lâu dài sẽ gặp vấn đề.

Các nền tảng tương thích có thể phát triển trò chơi là:

- Window.
- Hệ điều hành MacOS.
- Linux.
- HTML5
- Xbox một.
- PS4.
- Android.
- iOS.
- VR.

Khi chúng ta đi tạo một dự án mới, chúng ta có thể chọn loại dự án, tải một mẫu và định cấu hình các tham số dự án theo nền tảng nơi nó sẽ được thực thi.



Ngoài vai trò là công cụ phát triển trò chơi điện tử, Unreal Engine còn đóng một vai trò quan trọng ở các thị trường khác. Ví dụ, nó được sử dụng rộng rãi để thực hiện mô phỏng và kết xuất trong **kiến trúc**, cho **kỹ thuật** công việc, in **y học** cho các thử nghiệm hoạt động và thậm chí trong thực tế ảo. Thậm chí, UE thường được sử dụng trong rạp chiếu phim để thực hiện tất cả các loại hình động và hiệu ứng đặc biệt.

### *c/ Hướng dẫn về Unreal Engine*

#### **\* Khối 3d (Mesh):**

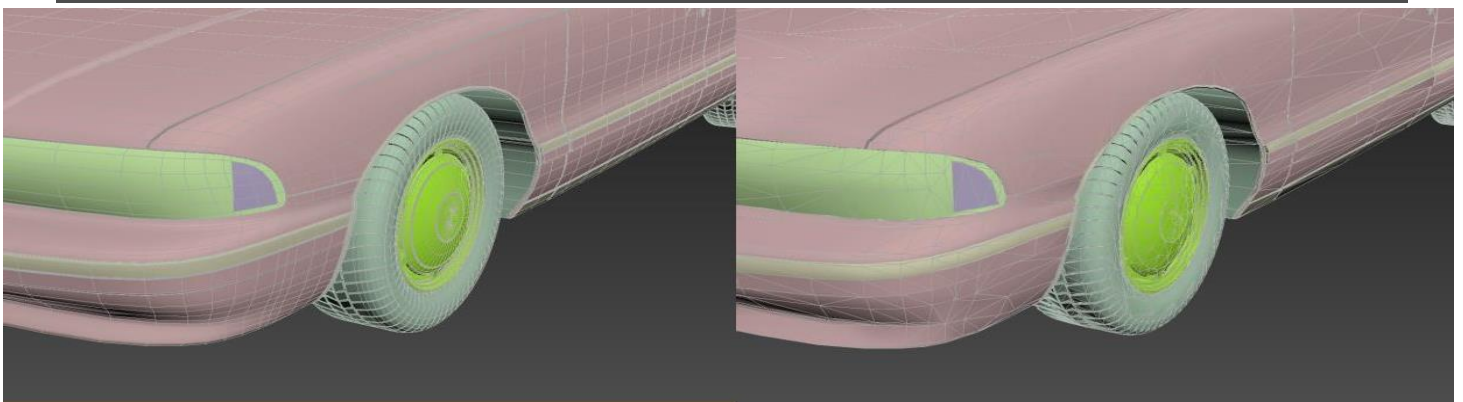
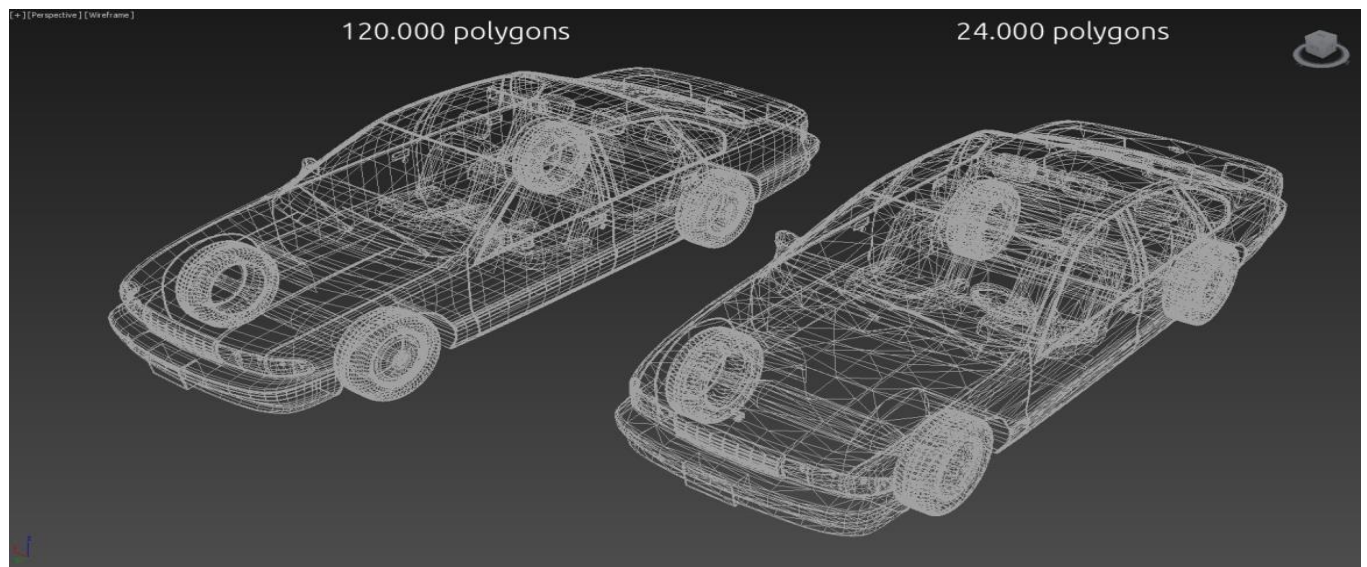
Đầu tiên, nên nhớ mặc định unit trong Unreal là 1cm, vì thế phải chỉnh unit trong 3ds max về cm cho khớp. Tổng số lượng polygon (polycount) (trong 3ds Max nhấn phím 7) nên cân nhắc, nên nhớ càng ít thì framerate (trong UE4 bấm phím ~ rồi nhập “stat fps” để kiểm tra) sẽ càng mượt nhưng bề mặt sẽ gồ ghề hơn và ngược lại, cần tùy chỉnh số polygon để cân bằng giữa framerate và đồ họa, tùy vào mục đích bạn muốn dùng làm gì – trong game (frame rate quan trọng hơn) hay archviz (chất lượng đồ họa quan trọng hơn).

Channel trong 3ds hay trong UE4 là các kênh thể hiện bề mặt vật liệu trong khối, channel 1 trong 3ds max tương ứng với channel 0 trong UE4, tương tự channel 2 trong 3ds max tương ứng với channel 1 trong UE4. Channel 1 là kênh chứa thông tin của mesh về texture, về map, về độ reflex, refract... dùng để render hình tĩnh chắc ai render cũng biết. Channel 2 là kênh chứa thông tin của mesh về shadow/light. Ánh sáng tác động lên mesh gây ra bóng đổ như thế nào, channel 2 là nơi ghi nhận thông tin đó để thể hiện ra, nói dễ hiểu thì thiếu channel 2 thì không có chiều sâu, không có khối 3d, bạn chỉ có phiên bản như cartoon cho các vật thể mà thôi.



### \* Tinh chỉnh polygon (Polygon optimization):

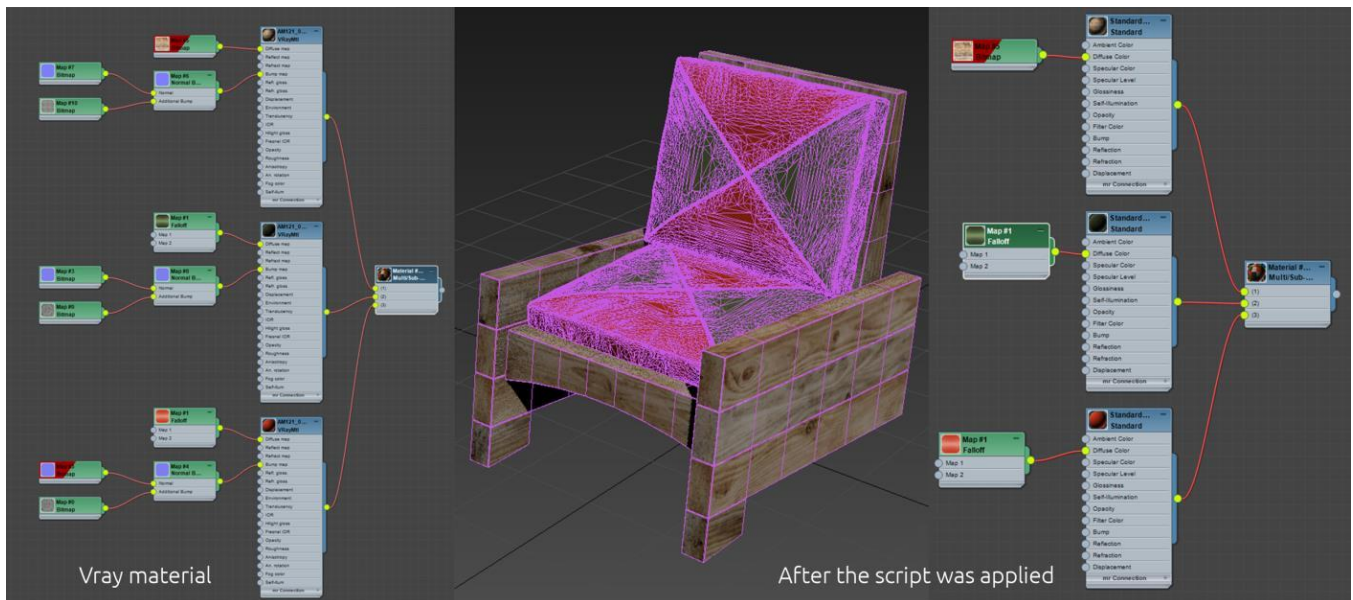
Giảm thiểu số polygon là bước quan trọng cho UE4. Ai đã từng có kinh nghiệm render, nhất là các cảnh vi mô cần đặc tả hay đại cảnh với số lượng polygon cỡ 7 8 chữ số chắc biết rõ vai trò của bước này. Trong 3ds Max có 3 cách để hiệu chỉnh là: Multires, Optimize, ProOptimizer. Bước này khá mệt, nhất là đưng mấy món phức tạp cong lượn hay gồ ghề như xe hơi hay cột gothic. Gặp trường hợp này bạn nên tách riêng từng mesh, vật càng phức tạp càng nên tách riêng ra. Đừng quên tick vào option “Protect UV” khi chỉnh.





### \* Chuyển từ Vray:

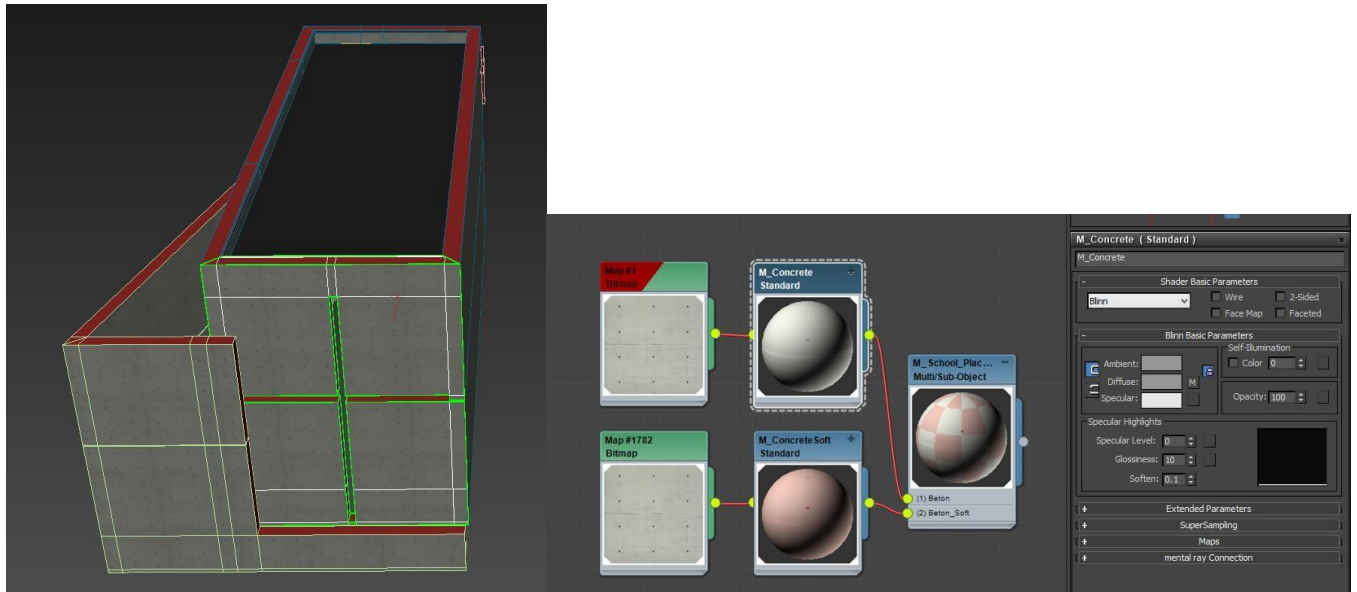
Chắc hầu hết ai render cũng dùng vray hay công cụ render trung gian chứ không dùng Mental ray mặc định của 3ds Max. Nếu bạn muốn xài lại vật liệu của Vray cho UE4, rất tiếc câu trả lời là không được vì UE4 không có plugin materialconverter như Corona hay Thea. Các chữa cháy chỉ là bạn mở bảng vật liệu Vray lên, convert hết sang standard material với các texture còn nguyên đó, sau đó chỉnh nhưng không nên đụng đến map UV. Bạn thấy script này chỉ có thể chuyển map diffuse, bump hay normal thì đã mất. Nên ngoài diffuse thì các map khác bạn phải import vào UE4 sau thủ công.



### \* UV Mapping:

UV mapping dùng ở channel 1 nhé. Ta chỉ nên chỉnh UV cho các mesh quan trọng như tường hay sàn, mấy cái mảng mà khi xuất ra ta muốn vị trí texture (ví dụ: tiling) từ 3ds max vào UE4 sẽ khớp y như thế. Còn các mesh ít quan trọng hơn bạn để mặc định UVW là 100,100,100cm. Mình đề nghị nên UVmap hết tất cả vật thể luôn cho tiện vì chẳng biết lúc này cần và không cần UV, phòng trường hợp lúc cần lại phải quay lại 3ds UV mất thời gian.

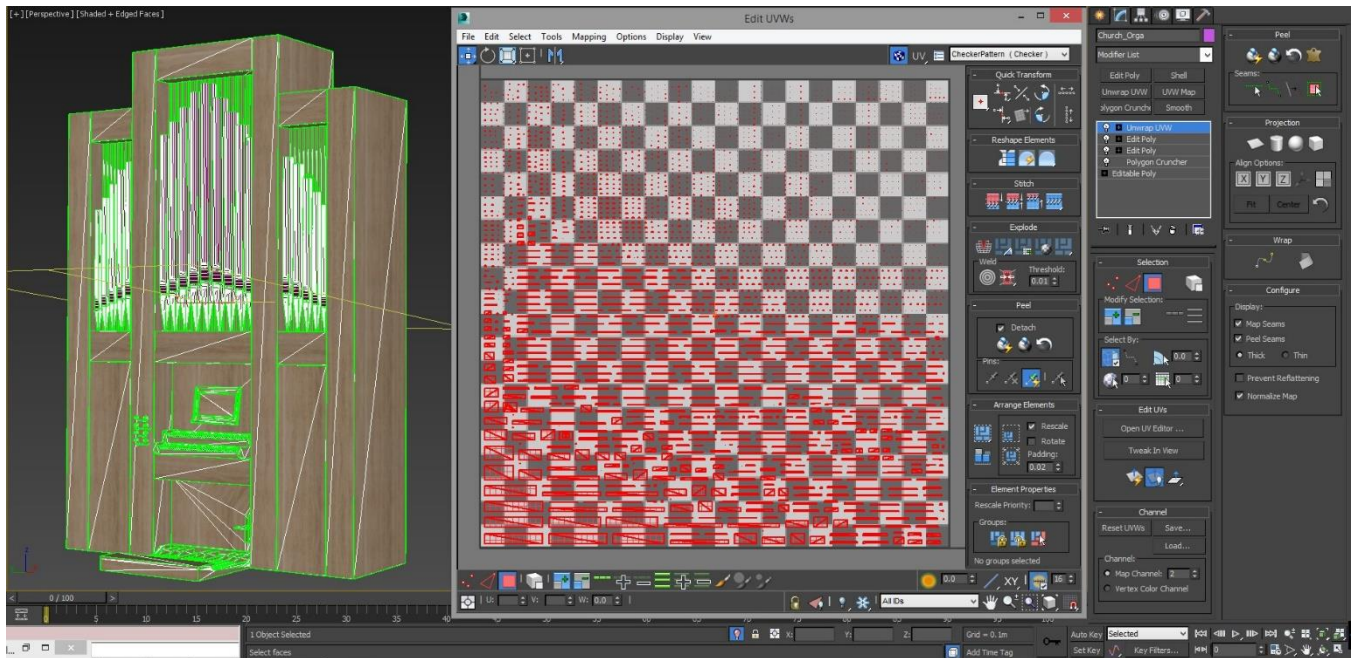
Vật thể có nhiều vật liệu, tốt nhất là tạo mat Multi/Sub Object cho nó (bên Revit xuất sang 3ds max rất hay dùng mat này) Các khối trước lúc xuất sang 3ds Max nên để mặc định là standard material. Lúc xuất sang UE4 sẽ tự động hiểu và apply (chỉ) diffuse map vào mesh cho bạn.



### \* Light Mapping:

Light Mapping dùng ở channel 2 nhé. Công đoạn này, theo kinh nghiệm của bản thân và nhận xét của các UE4 mem khác, là công đoạn khó nhai nhất. Không chỉ vì nó rất rất rất tốn thời gian, mà ngay cả khi bạn đã nghĩ là đã làm xong, chuyển sang dùng UE4 thì sau khi light baking (hiểu nôm na là render thời gian thực) vài lần vẫn xuất hiện rất rất nhiều lỗi, và rồi bạn phải mở 3ds max và chỉnh lại map rồi update lại. Các mesh càng phức tạp, mapping càng khó, chưa kể khi bạn xài phần mềm thứ 3 như revit xuất sang 3ds max thì khối đã nát nay càng nát hơn.

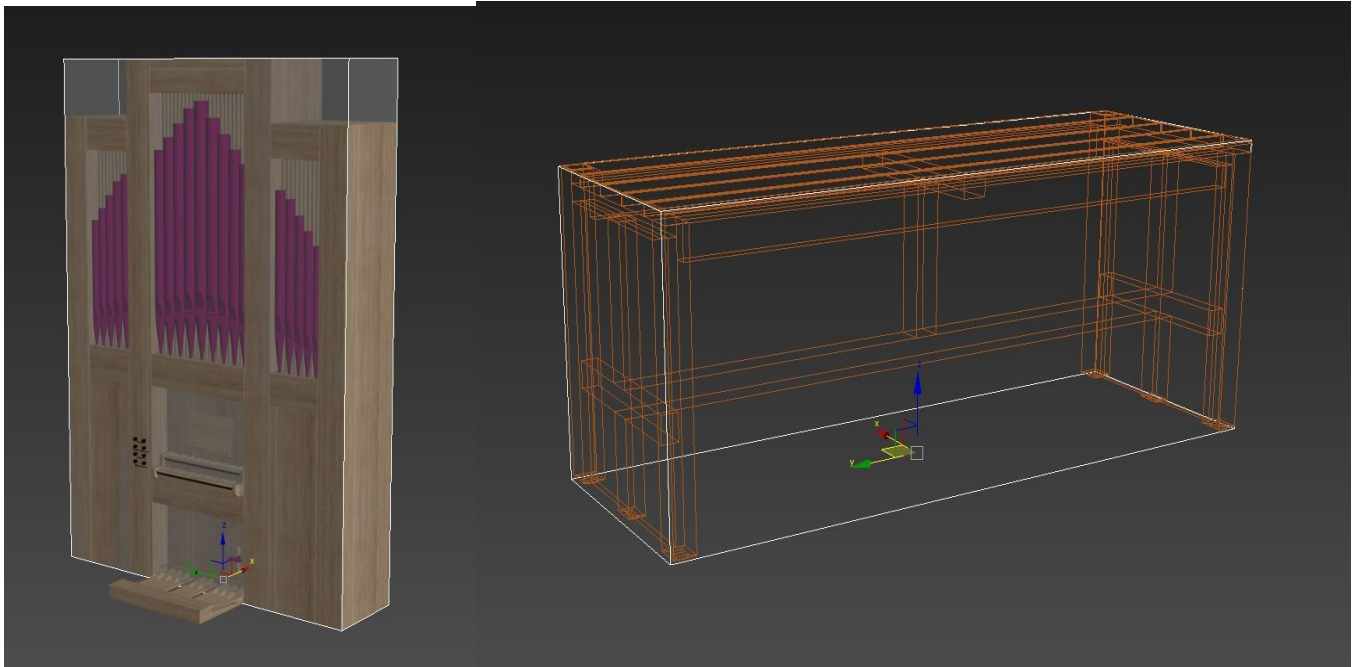
Ở channel 2, mỗi mesh tạo 1 bản uv mapping như trong hình, trong đó mỗi polygon tương ứng một unit. Để unwrapping nhanh hơn bạn cứ sử dụng lệnh unwrapping mặc định: Unwrap UVW modifier > default flatten mapping. Steamroller là 1 plugin giúp bạn tự động unwrapping hàng loạt mesh nhanh và gọn. Đây không phải là phương pháp tốt nhất để unwrap, đối với các khối hình hộp không bị cắt xẻ thì còn chấp nhận và làm rất nhanh, còn lại thì đòi hỏi một quá trình và phương pháp unwrap tỉ mỉ và phức tạp hơn.



### \* Collision (khối va chạm):

Chơi game chắc ai cũng biết đã là khối kiến trúc thì phải tương tác dc, không nhất thiết phải di chuyển dc nhưng ít nhất phải tạo cảm giác là bị cản lại khi muốn vượt qua. Ví dụ đơn giản là chơi CS, tất nhiên các bức tường, sàn, mái phải là vật tương tác được, để bạn có thể đứng trên đó, bị cản lại hay đạn bắn vào không xuyên qua được. Nếu không có collision, bạn sẽ có cảm giác như con ma, đi xuyên qua tất cả. Collision có thể hiểu là một khối vô hình, là vật mà tạo cho bạn cảm giác mesh là vật có thể va chạm (collide) tiếp xúc được.

Unreal có nhiều cách tạo ra collision dựa trên chính hình dáng kích thước mesh đó, nhưng khối càng phức tạp càng chiếm nhiều tài nguyên bộ nhớ nên làm giảm framerate. Tốt nhất là bạn tạo các khối collision đơn giản thôi, như hình hộp cho hầu hết các vật thể là được. Trước khi xuất khối thành collision từ 3ds max, bạn phải định pivot ở cùng vị trí với mesh và đặt phía trước tên nó “UCX\_” để Unreal tự hiểu đó là Collision, ví dụ: collision của 1 cái bàn có tên là “objectA” là một hình hộp có cùng pivot có tên là “UCX\_objectA”.



## **PHẦN 3 – DEMO**

### **3.1 Nội dung demo**

### **3.2 Hướng dẫn chạy demo**

## TÀI LIỆU THAM KHẢO

- [https://users.soict.hust.edu.vn/trungtt/uploads/slides/UNITY1 Game Engine.pdf](https://users.soict.hust.edu.vn/trungtt/uploads/slides/UNITY1_Game_Engine.pdf)
- <https://codelearn.io/sharing/unreal-engine-cong-cu-ho-tro-lam-game>
- <https://itigic.com/vi/unreal-engine-how-to-create-video-games/>
- [https://kupdf.net/download/huong-dan-tong-quat-ve-unreal-engine-4-cho-dien-hoa-kien-truc\\_5998db8fdc0d60b473300d1d\\_pdf](https://kupdf.net/download/huong-dan-tong-quat-ve-unreal-engine-4-cho-dien-hoa-kien-truc_5998db8fdc0d60b473300d1d_pdf)