



16
A

0
PANTERRE
LAB. CNC
LAB. FALCCHAMERIA
LAB. LASER
LAB. METALLI
LAB. TRAFORO

LaborA Virtual

Student Guide

LABO

English

Rendering Engines Overview **2**

Virtual Theatre Guide **3**

Introduction 3

Currently available service 4

Prerequisites 4

Setting up the project 5

Preparing the project for importing 5

 Disclaimer on scale of the model 6

 Unity 6

 Using .obj and .fbx models: 7

Hologram Table guide **9**

Introduction 9

Importing Software 9

Currently available service 9

 Model view 9

 Model Analysis 10

Prerequisites (Viewing Only) 10

 Preparing the project for importing 11

 Importing from .fbx 11

Prerequisites (deconstructable) 11

Making the model deconstructable 12

 Setting up the model for deconstruction 12

 Workaround in case of no hierarchy export 12

General Model Guidelines **13**

Common Issues 13

 Flipped Normals 13

Italiano

Panoramica sui motori di render **15**

Guida al Teatro Virtuale **16**

Introduzione 16

Servizio attualmente disponibile 16

Prerequisiti 17

Preparazione del progetto	17
Preparazione del progetto per l'importazione	18
Avviso sulla scala del modello	19
Uso di Unity	19
Importazione di modelli .obj e .fbx:	20
Guida al Tavolo Olografico	22
Introduzione	22
Software di Importazione	22
Servizi attualmente disponibili	23
Visione modelli	23
Analisi Modelli	23
Prerequisiti(Solo visualizzazione)	23
Preparazione del modello per l'importazione	24
Importare da .fbx	24
Prerequisiti(analizzabile)	24
Rendere il modello scomponibile	25
Preparare il modello per la scomposizione	25
Soluzione in caso di gerarchia mancante	25
Linee Guida Generali per la modellazione	27
Problemi Comuni	27
Normali invertite	27



General Guide to LaborA Virtual

This guide will describe the services offered to them by LaborA's Virtual Department as well as explain how to set up their project in order to make use of such services.

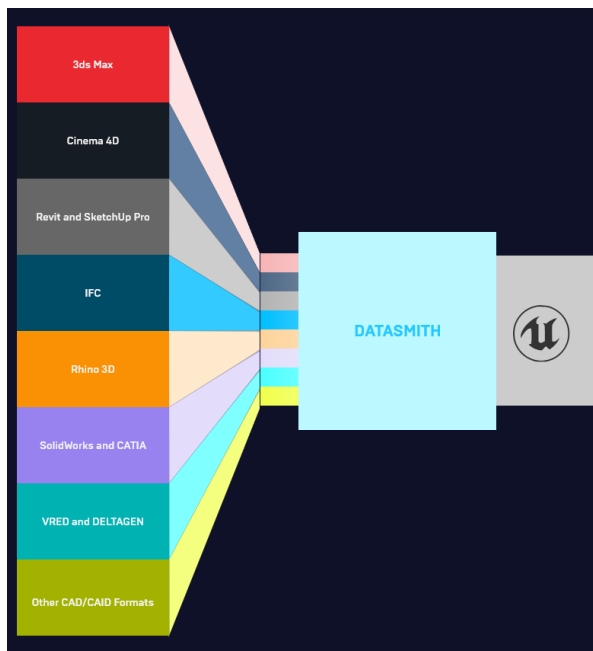
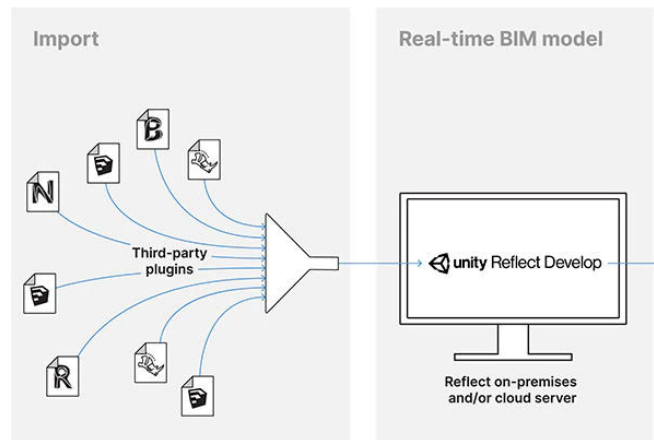
Rendering Engines Overview

LaborA Virtual leverages the power of real-time rendering engines in order to produce immersive environments and interactive architectural visualisation for students to use and experience.

In order to achieve this, the two main real-time engines are used: [Unreal Engine](#) and [Unity](#).

Natively, Unity and Unreal are made for producing video games and interactive experiences. In order to extend Unreal and Unity capabilities to visualisation of detailed architectural models two extensions of the two pieces of softwares are used:

- [Unity Reflect](#) is a dedicated importer which allows any user to import models from the main architectural modelling software such as Revit and Rhino. LaborA Virtual uses a copy of the importer to allow students to visualise their projects.



- [Datasmith](#) is Unreal Engine's custom file format supported by most modelling software. This format allows Unreal Engine to recreate any scene made by students into the engine natively, with little to none pre-processing to ensure compatibility. Datasmith also allows seamless transfer of entire scene data (including lighting and camera position) over to Unreal, speeding up the transition from legacy rendering engines to real-time ones.



Virtual Theatre Guide



Introduction

The Virtual Theatre is a 7m diameter cylinder capable of immersing the user in a virtual environment.

The Theatre is capable of displaying 360 degree images, videos as well as virtual interactive environments.

Through the use of its 6 projectors, the theatre displays the virtual scene on both its walls and its pavement, allowing the user to immerse themselves in a virtual environment of their choice with ease of use.

Virtual environments can currently be explored with both a keyboard and a virtualization platform, capable of keeping the user in place while transferring their movements in the virtual world, thus allowing anybody to actually walk inside the virtual space.

Currently available service

The theatre is capable of displaying architectural and design projects in a 360 environment with 1 to 1 scale.

As such, students can submit their projects in order to have them being displayed onto the theatre and immerse themselves in their projects, experiencing them to full scale.

The users will be able to choose to use either the virtualization platform or a keyboard to move around the virtual space.



Prerequisites

In order to recreate the virtual environment in real time, two different game engines are used to project the scene onto the theatre. Currently, LaborA makes use of both Unity and Unreal Engine. The students will thus need to know how to use at least one of the following pieces of software in order to be able to create a scene usable by the theatre. Knowledge of one of the two rendering engines is required by the students to the end of creating a physically usable 3d scene of the project:

- Supported Modelling Software
 - Rhino
 - 3ds Max
 - Revit and Sketchup
 - Blender
 - Maya
 - ZBrush
- Used Rendering Engines
 - Unity (Version 2020.1 and later)
 - Unreal Engine (Version 4.26 and later, 5.0 not yet supported)



Setting up the project

Currently, we plan to support the most common model formats used in architecture and design in order to allow ease of mind when preparing a project to view in the Theatre.



Students will thus be able to choose which rendering engine to use between Unreal Engine and Unity. Depending on the software choice, different steps will be required in order to set up the project.

Preparing the project for importing

First and Foremost : Check the [General Model Guidelines](#) to ensure your model has been made correctly and is ready for real-time rendering.

Given that the project should be physically coherent in order to be viewed by the camera, the project needs to be structured in a way which is able to discern between walls, floors and any architectural elements which the camera should not go through. The roof can be ignored since the camera for the theatre won't move vertically and thus won't need to collide with elements that are over it.

In the picture shown above, all walls, the floor and the reception desk have been made such that they can then be individually selected into Unity. Given that this is a generic guide for multiple software, it is best to check how to produce this type of assets through the documentation of the software being used.

For example: In the case of Autodesk Revit, this can be easily done by exporting the hierarchy of the project when creating an .fbx model, since Unity would be able to import such hierarchy and select each individual mesh separately from the others. The same cannot be said for Rhino, which does not have a hierarchy structure when modelling and thus is only able to export groups and layers.

To sum everything up here's steps to check that the model is ready to be imported into Unity:

- Ideally each element in the project should be exported in a way that it is discernible from the others (for example Unity should be able to select individually a chair, a table, a wall etc...).
- In case this is not possible, the best makeshift solution would be to export 3 different models: visual-only models (such as roofs and far away objects), walls (any vertical elements through which the camera should not be able to go) and floors. In this case, follow this brief overview of what should be set as visual-only:
 - All Chairs, Small tables and objects shorter than 50cm should be exported into the visual-only set.
 - If windows are lower than 2m, they should be exported into the visual-only set, otherwise they should go into the walls set.
 - Any object taller than 50cm should be exported into the model set containing all the walls..
 - All exported model sets should have the pivot set to the same coordinate, in order to align the resulting model correctly into Unity.



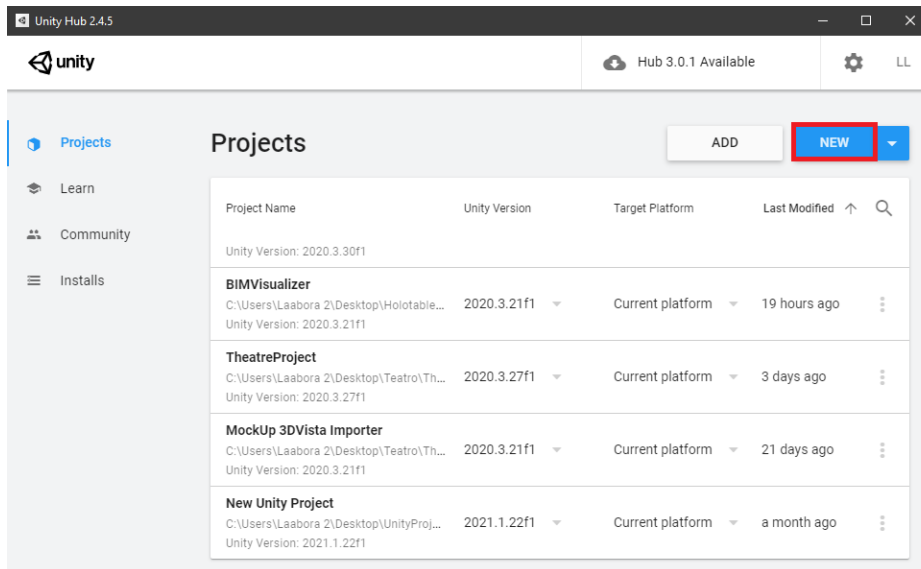
Disclaimer on scale of the model

Unity uses 1m scale for the environment and Unreal Engine uses 1cm scale for the environment. It is thus suggested to align the model's scale to the one used by the rendering software before exporting.

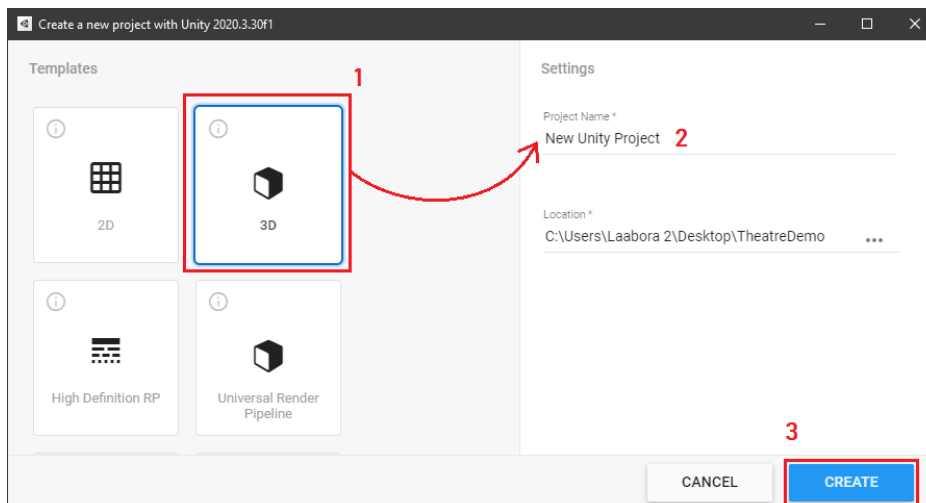
Unity

First off, when using Unity, the first thing to do is setting up a new Unity Project. After installing Unity and making sure the version is 2020.1 or later, the only thing to be done is the following:

- Open Unity Hub
- Click on the New button



- Choose the 3D template, give a name to the new project and then proceed to click "Create"



Once this is done your project should open up automatically, showing an empty scene with a light and a camera.

From there on out the importing process for the scene will change depending on the model format.

Using .obj and .fbx models:

When the file to import is either an .obj or .fbx, the importing process is pretty simple.

- First of all, drag and drop the model into Unity's "Project" tab. This will add the model to the list of available assets for the project in the form of a Prefab.
- After that, drag and drop from the Prefab representing the model into the scene, by dragging it over the "Scene" tab inside the project.

The model is now imported into the scene but it isn't physically coherent yet. In order to make the walls and pavement solid such that the camera won't clip into the project while viewing it, colliders need to be added to the parts of the project that need to be made "solid".

In order to do this, select all meshes representing walls, floors and all objects the camera is not supposed to go through and add a "Mesh Collider" component. This can be done easily by clicking on the wanted mesh and then selecting "Add Component" in the "Inspector" tab. From there, a menu will open up containing the "Mesh Collider" component.

Disclaimer: make sure that when selecting the mesh, the object shown in the inspector is made up of a "Mesh Renderer" and "Mesh Filter" component, otherwise the Mesh Collider won't be able to compute collisions properly.



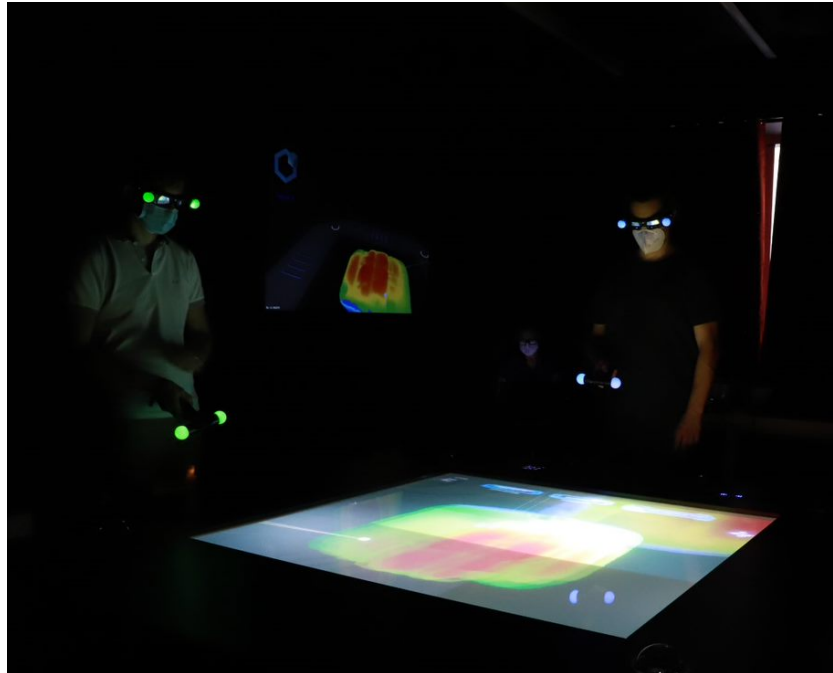
Hologram Table guide

Introduction

The Euclidean Hologram Table is a visualisation table capable of rendering a 3D view that can be seen from two different perspectives, thus allowing two to four users when using the table.

Users will be able to visualise mesh models as well as point clouds and data on this table through the use of dedicated glasses which will allow them to see everything in 3D. The table can also be

interacted with by two different users thanks to dedicated remote controlling wands, thus allowing anybody to move, rotate and scale the 3D visualisation being shown.



Importing Software

In order to quickly import and visualise models on the Hologram Table, a set of pieces of software can be used. These pieces of software will be used by the staff at the laboratory and won't be available to students and thus, in case of just model visualisation, the students will be in charge of submitting the model in one of the supported formats. The staff will then import the file of the model through UStream, Unlimited Detail Stream, a proprietary software capable of importing a variety of formats into Euclidean Presenter, thus allowing any point cloud or model to be viewed onto the Hologram Table.

Currently available service

Model view

The table currently offers a tool which is capable of displaying almost any kind of 3D object through the use of its dedicated software, Euclidean Presenter. Students can



thus submit their models and plan a viewing session where they can explore their creation directly, through the use of the dedicated controls.

Model Analysis

One more thing that can be done with the hologram table is being able to assemble and disassemble a model made up of different components, allowing students to inspect single sections of a model such as just the floor, furniture or walls. This service, though, requires one more degree of content curation before submitting the project for viewing.

Prerequisites (Viewing Only)

Given that the table supports most 3D model formats. Any software that can export models to the formats in the following table can be used by the student to produce a model to be viewed through the holotable.

Supported file format	
Supported Mesh Model formats	OBJ
	FBX (<i>only data on the first layer is recognised</i>)
	PM (<i>Euclidean's optimised mesh format – exporting to this will allow you to load mesh models faster next time</i>)
	DAE
	DXF
	PLY
	STL
Supported Point Cloud formats	UDS
	LAS
	LAZ

Given that, here is a list of pieces of software suggested to make the model to be viewed:

- Autodesk Revit and Sketchup
- 3ds Max
- Rhino
- Blender
- Maya



Preparing the project for importing

First and Foremost : Check the [General Model Guidelines](#) to ensure your model has been made correctly and is ready for real-time rendering.

As long as the model is in a supported format, the students shouldn't need to make any changes in order to view the project on the holotable. The only exception will be exporting to .fbx, which requires some caution.

Importing from .fbx

In case the student wants to export their model to the .fbx format, the entire model will need to be exported to the first layer of the .fbx data definition. This is due to the limitations of the importing software. This issue can be circumvented by exporting to .obj or any other supported format.

Prerequisites (deconstructable)

In order to export a deconstructable model, the hologram table will leverage Unity's capabilities instead of just Euclidean Presenter. Thus, the students will need to know how to use at least one of the following pieces of software in order to be able to create a usable scene.

- Supported Modelling Software
 - 3ds Max
 - Revit and Sketchup
 - Blender
 - Maya
 - ZBrush
- Used Rendering Engine
 - Unity (Version 2020.1 and later) with [Reflect](#)

Given that a real-time rendering engine is used to make the deconstructable preview of the model, some curation will be needed in order to actually make the model usable by the table. Ultimately, as long as a hierarchy of the different sections of the model can be imported into Unity, the model should be able to be imported and used properly.



Making the model deconstructable

First and Foremost : Check the [General Model Guidelines](#) to ensure your model has been made correctly and is ready for real-time rendering.

As stated before, students will also be able to view their projects down to the single components, being able to take apart single walls and pieces of furniture for inspection.

Not all models are supported for this kind of analysis, though.

Setting up the model for deconstruction

In order to make sure that a model can be disassembled once imported into Unity, please make sure that the model's hierarchy is exported properly by your modelling software of choice. In most cases, exporting to .fbx should be enough to keep the hierarchy but there are exceptions (as in the case of Rhino, which exports one single mesh with no hierarchy).

A detailed set of guidelines is listed on the Unity website, explaining the workflow for each piece of software supported. The guidelines can be found [here](#).

Workaround in case of no hierarchy export

In case the modelling software does not allow you to export the entire hierarchy to the .fbx file, there is a possible workaround which requires a bit more work.

A possible solution is exporting sections of the model, according to which parts of it the student wants to disassemble from the entire structure. Thus the student can do the following in order to still have a model which can be deconstructed:

- For each desired section:
 - Select the parts of the model belonging to the section
 - Export only the selected parts
- Make sure the **pivot stays constant during the export process**.

Once this process is done, the user will end up with **a number of .fbx files equal to the sections they want to deconstruct**. Only thing left now is import all the files into Unity, selecting the Pivot Mode in the upper part of the editor and then setting all the models' location to the same coordinates for all the models. This should reassemble the sections into the original model.



General Model Guidelines

The last two guides have illustrated how a user can make their model viewable through either the theatre and the table. While the illustrated guidelines are enough for just viewing the model, it is often good practice to curate the project as much as possible in order to make it usable by the rendering engines.

Common Issues

Flipped Normals

One of the most common issues that arise when modelling a building or a prop comes from the incorrect generation of normals from the modelling software. This is due to the software not being able to compute normals properly when it comes to very complex geometries and it is thus up to the user to fix the normals. The issue can **be missed by users given that most modelling software renders both faces**. This is not true for real time rendering engines, as they only render faces when the normal's direction is opposite to the camera, to save performance.

Please note : Incorrect normals will lead the engine to not being able to compute lighting properly, thus breaking the final image when viewing the model in either the theatre or the hologram table.

Given this, here is a list of quick guides that can help students find incorrect normals and fix them, the links have been organised by software used.

- Rhino3D : [Guide from Funbie Studios](#)
- 3ds Max : [Official Autodesk Guide on managing normals](#)
- Sketchup : [Forum post detailing multiple solutions](#)
- Blender : [Youtube guide explaining how to view and fix normal direction](#)
- Maya : [Quick guide by Pursuant Media](#)
- ZBrush : [Quick guide by Michael Pavlovich](#)



Guida Studenti a LaborA Virtual

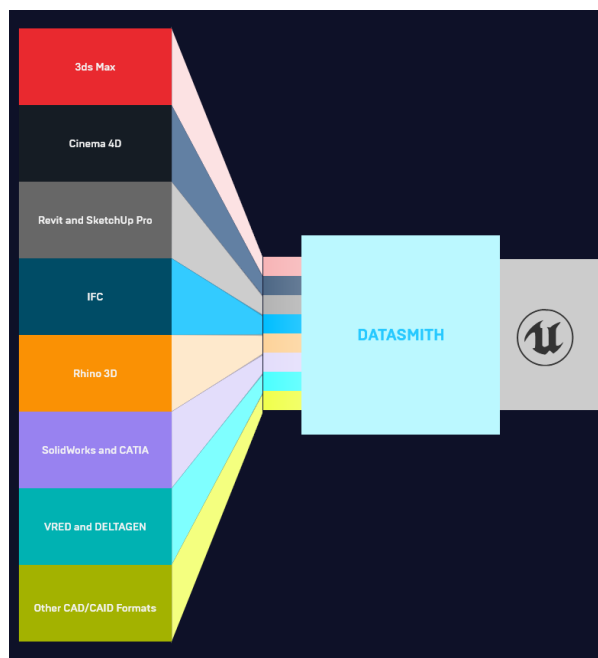
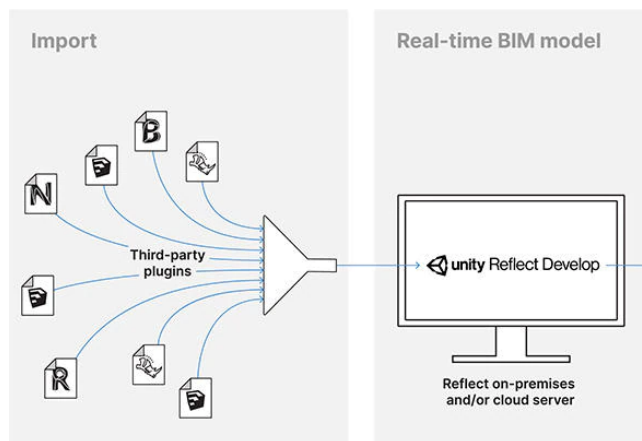
Questa guida illustrerà agli studenti i servizi offerti dal dipartimento Virtual di LaborA e spiegherà come preparare i propri progetti al fine di fare uso di questi servizi.

Panoramica sui motori di render

LaborA Virtual sfrutta le potenzialità di motori di render in tempo reale al fine di produrre ambienti immersivi e visualizzazioni architettoniche interattive che gli studenti possono usare e provare direttamente. A questo fine, i due motori di render real-time più usati vengono supportati: [Unreal Engine](#) e [Unity](#).

Nativamente, i due motori sono fatti per produrre videogiochi ed esperienze interattive. Per poter dunque estendere le capacità dei software alla visualizzazione di modelli architettonici, viene fatto uso di due estensioni:

- [Unity Reflect](#) è un software di importing dedicato che permette a qualunque utente di importare modelli dai principali software di modellazione tra cui: Revit e Rhino. LaborA Virtual ha accesso ad una copia dell' importer per poter permettere agli studenti di visualizzare i loro progetti.



- [Datasmith](#) è un formato di file proprietario definito da Unreal Engine ed esso è supportato dai maggiori software di modellazione. Questo formato permette al motore grafico di ricreare qualunque scena creata dagli studenti all'interno di Unreal Engine in maniera nativa, con minimo lavoro di pre processo al fine di assicurare compatibilità. Datasmith permette anche di trasferire interi dati di scena (incluse luci e posizioni della telecamera) ad Unreal, velocizzando la transizione dai motori di render statici a quelli real-time.



Guida al Teatro Virtuale



Introduzione

Il Teatro Virtuale è un cilindro con diametro di 7 metri capace di immergere l'utente in un ambiente virtuale.

Il Teatro può mostrare immagini e video a 360 gradi e può anche riprodurre ambienti virtuali interattivi.

Tramite l'uso dei suoi 6 proiettori, il teatro riproduce la scena virtual sia sui muri interni che sul pavimento, permettendo all'utente di immergersi completamente in un ambiente virtuale di sua scelta con facilità di accesso.

Gli ambienti virtuali possono essere esplorati, al momento, tramite l'utilizzo di una tastiera oppure di una piattaforma virtualizzante dedicata, capace di tenere l'utente fermo sul posto e di trasferire i passi dell'utente nel mondo virtuali, permettendo dunque di effettivamente camminare all'interno dello spazio esplorato.

Servizio attualmente disponibile

Come descritto in precedenza, il teatro può riprodurre progetti di design e di architettura in scala 1 a 1 a 360 gradi.

Detto ciò, gli studenti possono inviare i loro progetti affinché questi vengano mostrati nel teatro, rendendo dunque possibile immergersi e percepire questi progetti nella loro intera grandezza.

Agli studenti è data la possibilità di scegliere se usare la tastiera o la piattaforma virtualizzante come mezzo per muoversi all'interno dello spazio virtuale.



Prerequisiti

Al fine di visualizzare gli ambienti virtuali in modo interattivo e in tempo reale, il teatro può fare uso sia di Unity che di Unreal Engine.

Agli studenti è dunque richiesta almeno una conoscenza minima di uno di questi due motori grafici ed una conoscenza efficace di uno dei software di modellazione elencati di seguito.

- Software di Modellazione Supportati
 - Rhino (con limitazioni)
 - 3ds Max
 - Revit e Sketchup
 - Blender
 - Maya
 - ZBrush
- Motori grafici usati nel teatro
 - Unity (Versione 2020.1 o più recente)
 - Unreal Engine (Versione 4.26 o più recente, 5.0 non supportata)



Preparazione del progetto

Al momento, è in piano il supporto dei maggiori formati file per modelli usati in architettura e design, per permettere serenità d'animo durante la preparazione del progetto da visualizzare nel teatro.



Gli studenti, dunque, potranno scegliere quale motore grafico usare tra Unreal Engine e Unity. In base alla scelta del software, è richiesta una serie di accorgimenti differente al fine di avere un progetto visualizzabile nel teatro.

Preparazione del progetto per l'importazione

Prima di tutto : Siete pregati di leggere le [Linee Guida Generali](#) per assicurarvi che il modello sia stato fatto correttamente e che sia pronto per il render in tempo reale. Dato che il progetto dovrà essere fisicamente coerente al fine di poter essere visualizzato dalla telecamera senza che essa attraversi gli oggetti, il progetto dovrà essere ristrutturato in modo da poter distinguere tra muri, pavimenti e ogni elemento del progetto che la camera non può attraversare. Il tetto può essere ignorato in quanto la telecamera non si muoverà verticalmente ma rimarrà a terra per simulare un utente di altezza circa 1.7 metri.

Nell'immagine mostrata precedentemente, tutti i muri, il pavimento ed il banco reception sono stati importati in Unity in modo tale che potessero essere selezionati individualmente.

Questa è una guida generica per diversi software di modellazione, dunque si suggerisce di consultare la documentazione del software di modellazione scelto al fine di poter esportare il proprio modello architettonico con la gerarchia degli oggetti inclusa nel file.

Esempio: Nel caso di Autodesk Revit, questo può essere fatto facilmente esportando la gerarchia del progetto quando viene esportato in .fbx, in quanto Unity può estrarre le informazioni di gerarchia dal file e dunque separare le mesh di ogni oggetto.

Rhino, invece, non supporta l'esportazione di una gerarchia e questo porta all'unificazione delle mesh di tutti gli oggetti nel progetto, perdendo dunque le informazioni di gerarchia.

Riassumendo, ecco i passi da seguire per verificare che il modello sia pronto per l'importazione in Unity:

- Idealmente, ogni elemento nel progetto dovrebbe essere esportato in modo che esso sia distinguibile dagli altri (dunque Unity dovrà poter selezionare il modello di una sedia separatamente dal muro, da un tavolo ecc...)
- Qualora questo non fosse possibile, la soluzione temporanea è quella di esportare 3 set di modelli differenti: un set al fine della sola visualizzazione (per esempio oggetti lontani o il soffitto), pareti (quindi elementi verticali non attraversabili dalla telecamera) e pavimentazione (elementi orizzontali non attraversabili.) In questo caso, siete pregati di seguire questa breve panoramica di cosa può essere inserito nel set di sola visualizzazione:
 - Tutte le sedie, piccoli tavoli e generalmente oggetti più bassi di 50 cm dovranno essere esportati nel set di sola visualizzazione.



- Se le finestre sono ad un'altezza minore di 2m, dovranno essere inserite nel set delle pareti, altrimenti potranno essere inserite nel set di sola visualizzazione.
- Ogni oggetto più alto di 50 cm o ad un'elevazione maggiore di 50 cm dovrà essere esportato nel modello contenente tutti gli elementi verticali non attraversabili.
- Tutti i modelli esportati dovranno avere il loro pivot alla stessa coordinata, al fine di poterli riallineare all'interno di Unity.

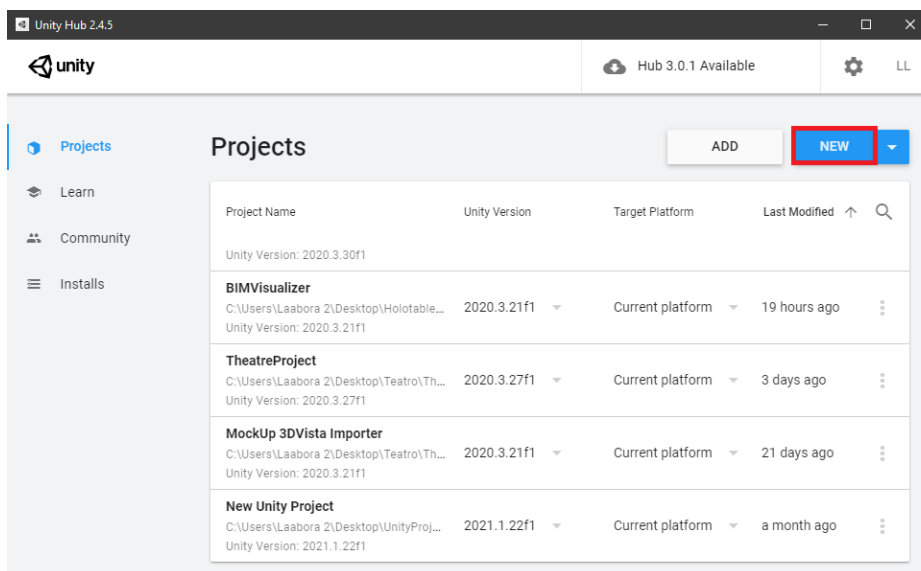
Avviso sulla scala del modello

Unity utilizza una scala di 1 metro per gli ambienti ed Unreal utilizza invece una scala di 1 centimetro. Si suggerisce dunque di allineare la scala dei modelli a quella del motore di rendering scelto. Si ricorda, inoltre, che le dimensioni del modello e la sua scala sono parametri differenti. Al momento dell'esportazione suggeriamo di impostare 1 come la scala reale su tutti gli assi (per esempio un oggetto 50cm x 50cm x 50cm non dovrà avere una scala di 0.5x0.5x0.5 ma di 1x1x1 con dimensioni 0.5mx0.5mx0.5m). Per eventuali chiarimenti si suggerisce di consultare la documentazione del proprio software.

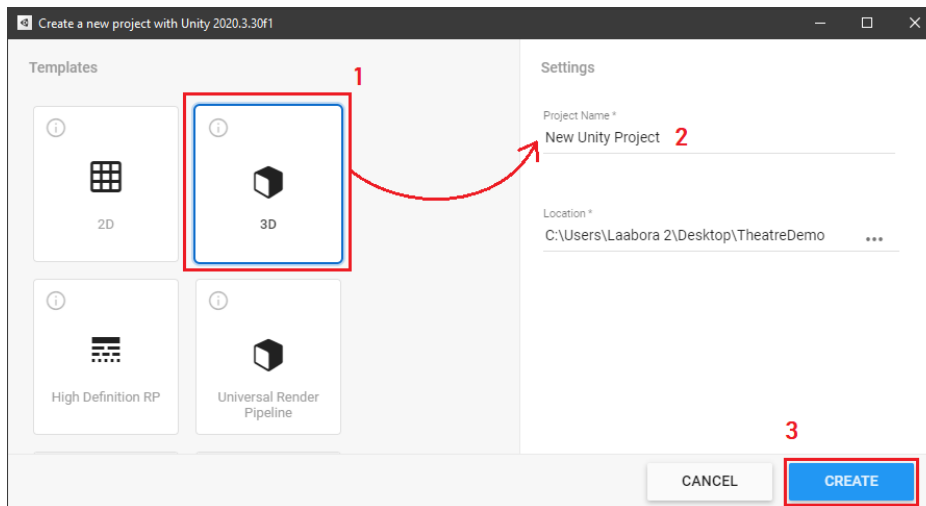
Uso di Unity

Innanzitutto, quando viene usato Unity, la prima cosa da fare è lanciare un nuovo Progetto Unity. Dopo aver installato e dopo essersi assicurati di aver installato una versione uguale o più recente della 2020.1, l'unica cosa da fare è la seguente:

- Aprire Unity Hub
- Cliccare sul tasto New



- Selezionare il template “3D”, dare un nome al nuovo progetto ed infine cliccare su “Create”.



Una volta fatto, il progetto dovrebbe aprirsi automaticamente, mostrando una scena vuota con una luce direzionale ed una telecamera.

Da qui in poi il processo di importazione cambierà in base al formato del modello.

Importazione di modelli .obj e .fbx:

Quando il file da importare è un .obj o .fbx, il processo di importing è nativamente supportato da Unity.

- Prima di tutto, trascinare il file del modello nella finestra di Unity, nel pannello chiamato “Project”. Questo aggiungerà il modello alla lista di asset disponibili a Unity sotto forma di [Prefab](#).
- Dopo ciò, trascinare e lasciare il Prefab che rappresenta il modello nella scena, trascinando il file sul pannello “Scene” che mostra la luce e la telecamera iniziali.

Il modello ora importato non è ancora coerente fisicamente, ciò significa che la telecamera potrà attraversare le pareti e tutti gli oggetti creando artefatti visivi. Per correggere questo fenomeno, al prefab del modello dovranno essere aggiunti dei cosiddetti “[Collider](#)” che renderanno gli oggetti “solidi” secondo Unity.

Per poter fare questo, selezionare tutti i modelli e le mesh che rappresentano pareti, pavimenti ed oggetti solidi ed aggiungerci il componente “Mesh Collider”.

Questo può essere fatto cliccando la mesh scelta nel pannello “Hierarchy” e cliccando “Add Component” nel pannello “Inspector”. Da qui si potrà scegliere “Mesh Collider” o cercandolo nella barra di ricerca o scorrendo la lista di possibili componenti.

Avviso: assicurarsi, quando si seleziona la mesh, che l’oggetto selezionato contenga i componenti “[Mesh Filter](#)” e “[Mesh Renderer](#)”, altrimenti il Mesh Collider non potrà calcolare le collisioni adeguatamente.



Guida al Tavolo Olografico

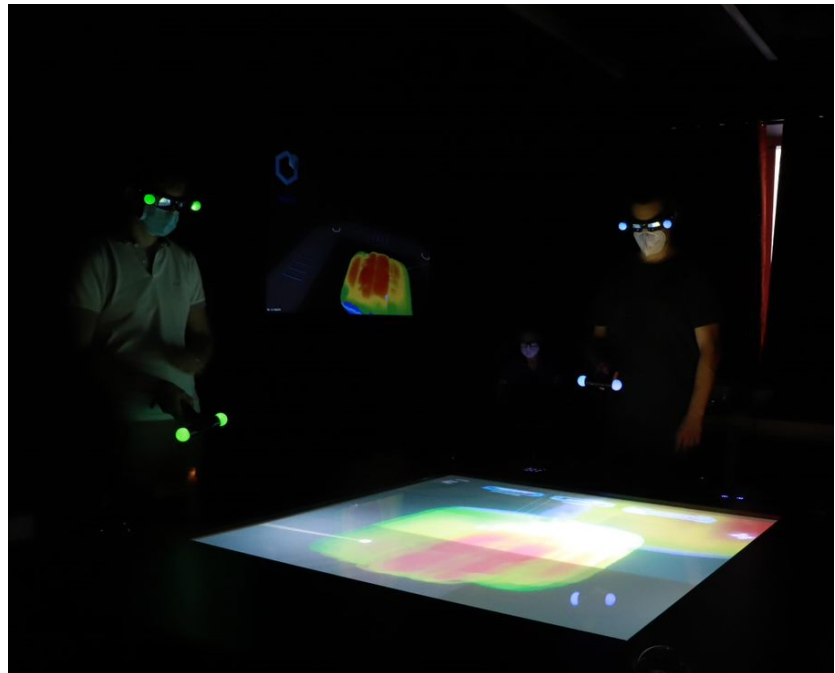
Introduzione

Il Tavolo Olografico

Euclidean è un tavolo di visualizzazione capace di renderizzare una vista 3D che può essere osservata da due prospettive differenti, permettendo a fino a 4 utenti l'utilizzo simultaneo del tavolo.

Gli utenti saranno capaci di visualizzare modelli mesh, nuvole di punti e dati tramite l'utilizzo di occhiali dedicati, che permettono di osservare il tutto in tre dimensioni. Due utenti

potranno anche interagire simultaneamente con il tavolo grazie a dei telecomandi dedicati, permettendo a chiunque di muovere, ruotare e ridimensionare la visualizzazione 3D mostrata.



Software di Importazione

Per importare rapidamente e visualizzare i modelli sul tavolo ologrammi, un insieme di programmi può essere usato.

Questi software verranno usati dallo staff al laboratorio, dunque non sono disponibili agli studenti.

Gli studenti, quindi, si dovranno occupare di inviare il modello in uno dei formati supportati. In seguito, il laboratorio si occuperà di importare i modelli tramite UStream, Unlimited detail Stream, un software proprietario capace di importare una varietà di format all'interno di Euclidean Presenter, il visualizzatore di modelli del tavolo olografico.



Servizi attualmente disponibili

Visione modelli

Il tavolo al momento offre uno strumento capace di mostrare quasi ogni tipo di oggetto 3D tramite l'uso del software dedicato, Euclidean Presenter. Gli studenti possono dunque inviare i loro modelli e pianificare una sessione in cui possono esplorare in maniera diretta i loro modelli 3D tramite il tavolo olografico.

Analisi Modelli

Un'altra capacità del tavolo ologrammi è quella di permettere agli studenti di interagire con i loro modelli, dunque permettendo di assemblare e scomporre un modello fatto di componenti diversi. Questo permetterà l'ispezione di sottosezioni del modello quali pavimentazione, mobili e pareti. Questo servizio, però, richiede della cura in più al fine di rendere il modello inviato interagibile con il tavolo.

Prerequisiti(Solo visualizzazione)

Dato che il tavolo supporta la maggior parte dei formati per modelli 3D, qualunque software che può esportare modelli in uno dei formati elencati di seguito può essere usato dagli studenti per creare un modello visualizzabile tramite il tavolo.

Supported file format	
Supported Mesh Model formats	OBJ
	FBX (<i>only data on the first layer is recognised</i>)
	PM (<i>Euclidean's optimised mesh format – exporting to this will allow you to load mesh models faster next time</i>)
	DAE
	DXF
	PLY
	STL
Supported Point Cloud formats	UDS
	LAS
	LAZ

Detto ciò, ecco una lista di suggerimenti per quanto riguarda i software di modellazione utilizzabili:

- Autodesk Revit and Sketchup



- 3ds Max
- Rhino
- Blender
- Maya

Preparazione del modello per l'importazione

Innanzitutto : Consultare le [Linee Guida Generali](#) per assicurarsi che il modello sia stato creato adeguatamente e sia pronto per il render real-time.

Finché il modello è in un formato supportato, gli studenti non avranno bisogno di fare alcuna modifica per poter visualizzare il proprio progetto sull'holotable. L'unica eccezione è per il formato .fbx, che richiede un po' precauzione.

Importare da .fbx

In caso gli studenti vogliano esportare il loro progetto in formato .fbx, l'intero modello dovrà essere contenuto interamente nel primo livello del file .fbx. Questo è dovuto alle limitazioni del software di importazione. Questa limitazione può essere evitata esportando ad .obj il file.

Prerequisiti(analizzabile)

Per poter esportare un modello scomponibile, l'holotable sfrutterà le capacità di Unity invece di Euclidean Presenter. Gli studenti dovranno dunque avere una conoscenza minima del motore grafico oltre a conoscere un software di modellazione compatibile, elencato di seguito.

- Software di modellazione compatibili
 - 3ds Max
 - Revit e Sketchup
 - Blender
 - Maya
 - ZBrush
- Motore grafico usato
 - Unity (Versione 2020.1 o più recente) con [Reflect](#)

Dato che viene usato un motore grafico in tempo reale, saranno necessarie delle precauzioni per rendere il modello utilizzabile dal tavolo.

Fondamentalmente, finché la gerarchia delle differenti sezioni del modello può essere esportata in Unity, il modello potrà essere scomposto adeguatamente.



Rendere il modello scomponibile

Innanzitutto : Consultare le [Linee Guida Generali](#) per assicurarsi che il modello sia stato creato adeguatamente e sia pronto per il render real-time.

Come detto in precedenza, gli studenti potranno anche visionare i loro progetti fin nei singoli componenti, potendo smontare singole pareti e mobili per un sopralluogo.

Tuttavia, non tutti i modelli sono supportati per questo tipo di analisi.

Preparare il modello per la scomposizione

Per assicurarsi che un modello possa essere disassemblato una volta importato in Unity, gentilmente assicurarsi che la gerarchia del modello sia esportata correttamente dal software di modellazione scelto. Nella maggior parte dei casi, l'esportazione in .fbx dovrebbe essere sufficiente per mantenere la gerarchia, ma ci sono delle eccezioni (come nel caso di Rhino, che esporta una singola mesh senza gerarchia).

Una serie dettagliata di linee guida è elencata sul sito Web di Unity, che spiega il flusso di lavoro per ciascun software supportato. Le linee guida possono essere trovate [qui](#).

Soluzione in caso di gerarchia mancante

Nel caso in cui il software di modellazione non consenta di esportare l'intera gerarchia nel file .fbx, esiste una possibile soluzione alternativa che richiede un po' più di lavoro in più.

Una possibile soluzione è esportare sezioni del modello, in base a quali parti di esso verranno scomposte dallo studente. Lo studente, quindi, può fare quanto segue per avere comunque un modello che può essere decostruito:

- Per ogni sezione voluta:
 - Selezionare le parti del modello che vi appartengono
 - Esportare solo le parti selezionate
- Assicurarsi che il pivot **rimanga lo stesso durante ogni esportazione.**

Una volta terminato questo processo, l'utente si ritroverà con un numero di file .fbx pari alle sezioni che desidera disassemblare. L'unica cosa rimasta da fare è importare tutti i file in Unity, selezionando la modalità Pivot nella parte superiore dell'editor e quindi impostando la posizione di tutti i modelli sulle stesse coordinate per tutti i modelli. Questo dovrebbe ricostruire il modello originale dalle sezioni.



Linee Guida Generali per la modellazione

Le ultime due guide hanno illustrato come un utente può rendere visibile il proprio modello sia attraverso il teatro 360 che l'holotable. Sebbene le linee guida illustrate siano sufficienti per la semplice visualizzazione del modello, è spesso buona norma curare il più possibile il progetto in modo da renderlo utilizzabile dai motori di rendering.

Problemi Comuni

Normali invertite

Uno dei problemi più comuni che sorgono durante la modellazione di un edificio o di un oggetto di scena deriva dalla generazione errata di normali dal software di modellazione.

Ciò è dovuto al fatto che il software non è in grado di calcolare correttamente le normali quando si tratta di gestire geometrie molto complesse e spetta quindi all'utente correggere gli errori.

Gli utenti potrebbero **non accorgersi del problema dato che la maggior parte dei software di modellazione esegue il rendering di entrambe le facce di un poligono**. Questo non viene fatto dai motori grafici in tempo reale, poiché eseguono il rendering di un poligono solo quando la direzione della normale associata è opposta alla telecamera, per migliorare le prestazioni.

Nota bene: Normali errate porteranno il motore grafico a non essere in grado di calcolare correttamente la luce e le ombre per un oggetto, producendo così un'immagine finale distorta quando si visualizza il modello nel teatro o sul tavolo ologrammi.

Detto questo, ecco un elenco di guide rapide che possono aiutare gli studenti a trovare normali errate e correggerle, i collegamenti sono stati organizzati in base al software utilizzato.

- Rhino3D : [Guida da Funbie Studios](#)
- 3ds Max : [Guida ufficiale Autodesk sulla gestione delle normali](#)
- Sketchup : [Post su Forum Sketchup che descrive più soluzioni](#)
- Blender : [Guida Youtube su come visualizzare e correggere le normali](#)
- Maya : [Guida rapida da Pursuant Media](#)
- ZBrush : [Guida rapida di Michael Pavlovich](#)



Contatti

E-mail : labora@polimi.it

Telefono : 02 2399 5670

Direttore Scientifico

Cecilia Maria Bolognesi

cecilia.bolognesi@polimi.it

Responsabili Sezione Virtual

Simone Balin

simone.balin@polimi.it

Elio Sasso

elio.sasso@polimi.it

