

Practical Session 1

Introduction to Unity

Start Unity Hub and open the Project “MySummerSchoolMuseum” we already created for you.

Preliminary Information

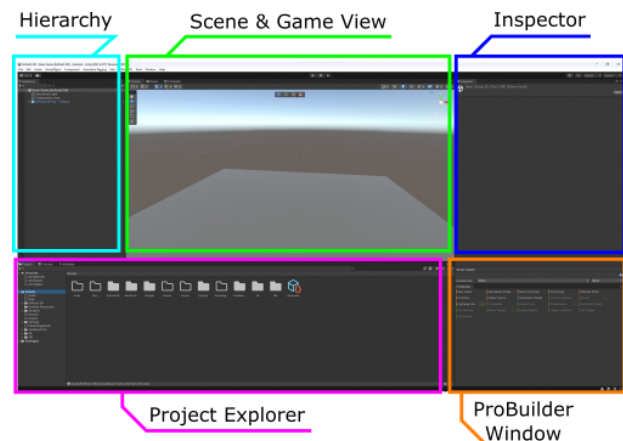
First, we need to establish some Unity concepts:

- **GameObject:** The building blocks of a Unity Game. They do not implement any behavior but rather provide a Transform (= Position, Rotation and Scale) and a place to add multiple Components.
- **Components:** These are used to add behavior to GameObjects. This may be movement, collision detection, player controls and much more.
- **Scene:** A Scene contains every GameObject in your game for a given situation, like a level in a classical video game or your virtual exhibition.
- **Prefab:** GameObjects you created within your Scene can be saved outside of it as so called Prefab. This allows you to use them in various scenes and even multiple times in the same scene. Prefabs will be highlighted blue.
- **Material:** A Material describes the visual surface properties of a 3D-object like it's color, it's bumps and how well it reflects light.

Unity Window

Let's look at the Unity Editor:

- **Bottom Left:** Here you will find the Project Explorer. This is used to browse all Files, Objects, Materials... within your project.
- **Top Left:** The Hierarchy displays the structure of all GameObjects within the Scene. Some GameObjects are indented as they are children of the parent GameObject. Using the small triangle to the right of the parent, all its children can be hidden or unfolded in the Hierarchy. This is more than a logical structure: When the parent is moved, rotated or scaled, the effect will also apply to all its children. You can compare this behavior to folders on your computer. When you move it somewhere, all its contents will be moved too.
Right-Clicking in the Hierarchy opens a menu for adding new GameObjects. For us the most important sub-menu is “ExPresS XR > ...”.
- **Top Right:** The Inspector lets you manage the Components of currently selected GameObject and change their values. Individual Components - but also the whole GameObjects - can be de-/activated with the checkboxes in their headers. Once deactivated, they do not have any effect in your application.



- **Top Middle:** The Scene View renders the Viewport of your game. You can move through the viewport using *W*, *A*, *S*, *D*. Moving the mouse while holding the right mouse button rotates your camera.
This is also the place to move, scale or rotate your GameObjects. You can change your tool either in the top-right of the Viewport or using the buttons *W*, *E*, *R* for moving, rotating or scaling respectively.
Another useful feature is focusing the camera on the GameObject that is selected in the Hierarchy by pressing *F*.

ProBuilder

ProBuilder lets you create and edit meshes within Unity. To access its functionality, the ProBuilder-Window must be opened by selecting “Tools > ProBuilder > ProBuilder Window”. We advise docking the window to the bottom-right by dragging the ProBuilder-Tab at the top of the newly opened window and dragging it to the **bottom right**.

Upon opening the ProBuilder-Window four symbols will appear at the top of the Scene View Window. These will allow you to change your selection and editing mode to normal GameObjects, Vertices, Edges or Planes. The later three will only work for special ProBuilder-Meshes and you will not be able to move, rotate or scale other objects. For that, “Object Selection” must be enabled.



Using the ProBuilder-Window you can perform more complex actions like creating or mirroring Objects or add new edges or vertices. These operations are context-sensitive as they change with your Selection Mode. When following the tutorials, make sure that you have the correct selection mode enabled.

We will not explain every action as there is plenty and we only need a few. The necessary ones will be explained when they are needed.

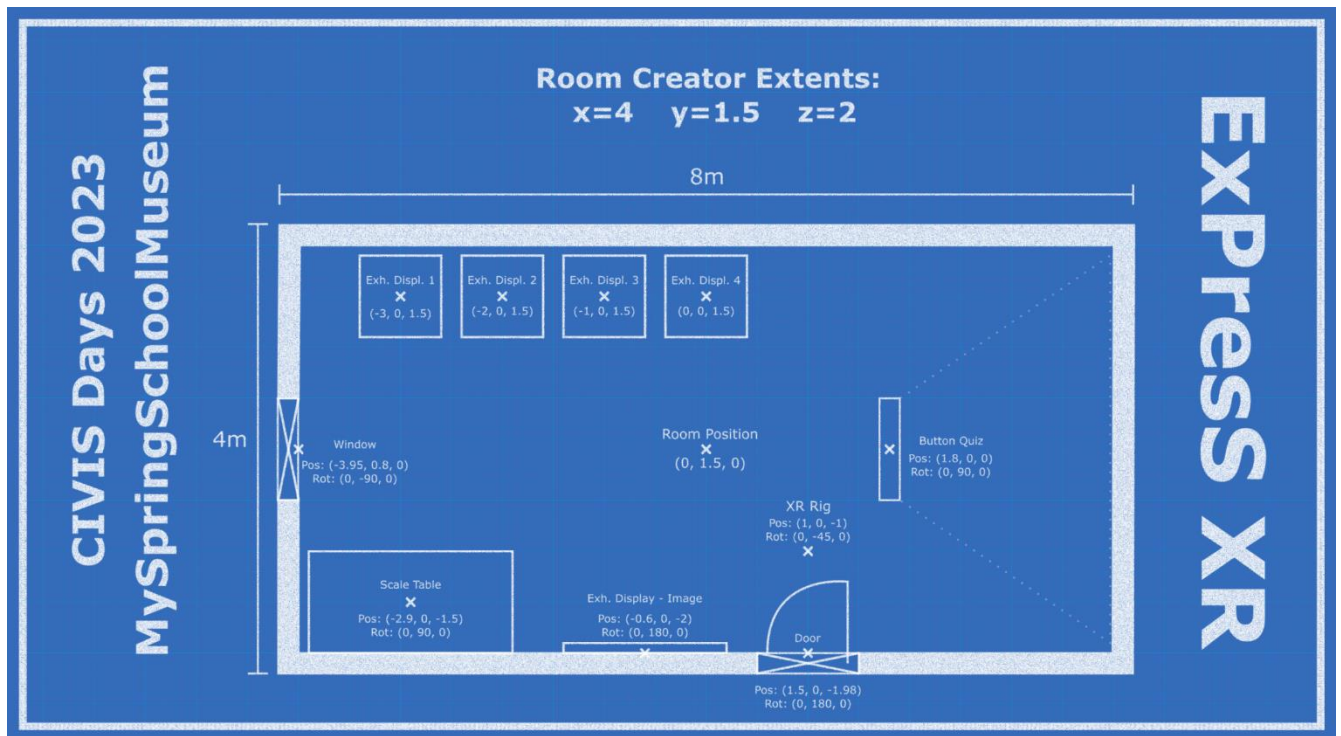
ExPresS XR

As we are now familiar with the Unity Editor, we can look at ExPresS XR and its tools. We will use it to create your VR exhibition. You can use the blueprint below as a guide but feel free to change the layout as you please.

As a visual guide we created YouTube-Tutorials for some of the steps. You can find them using the Links and QR-Codes.

Making a Scene

The very first thing we need is to create a new Scene. Go to “File > New Scene” and select “Basic (URP)” in the next Dialog. When pressing the “Create”-Button, you will usually be asked if you want to Save the current Scene, in which case we will click “No”. In the next dialog, we need to select a name and save-location for our scene. Save it under “Assets/Scenes/” and name it “MySpringSchoolMuseum”.



Create a Room

Tutorial-Video: <https://www.youtube.com/watch?v=MdBKWcyNF4w>



Before you can display your exhibits, we need a location for our virtual museum. In our case, this will be a single room that we can create using ExPresS XR's RoomCreator. You will find it under at the top under "ExPresS XR > Rooms... > Open Room Creator". Here you can specify the position and extents of your room. The latter describes the distance from the center of the room to the walls. The actual size of the room is double this value. Leave the position at **(0, 0, 0)**. Enter **X=4**, **Y=1.5** and **Z=2** as values for the room's **Extents**, to create a room that is 8 meters wide, 3 m high and 4 m deep.

Make sure have "Add Teleportation Area" is **enabled**, "Wall Mode" set to **Separate Floor** and "Room Preset" to **Exhibition**. Click "Create" and close the Room Creator.

You will find your room being generated. It is already configured so that we will only see the walls when looking from the inside. This makes editing easier as this will be done while viewing the room from the outside.

The ExPresS XR Rig

We need a way to see and interact with our world. With the "ExPresS XR Rig" we have a configurable all-in-one solution for this. As we want to use teleportation as movement, we can use one of the presets that ExPresS XR provides. It can be added to your Scene by right-clicking in the Hierarchy and selecting "ExPresS XR > XR Rig > Teleport".

To let users only interact with objects directly, the “Interaction Options” must be changed. Select the XR Rig and click on the dropdown next to “Interaction Options”. First select **None** to deselect everything and then select **Direct** to enable grabbing and(!) **Choose Teleport Forward**.

Set its position via the Inspector to (1, 0, -1) and its rotation to (0, -45, 0).

Note: Unity might have automatically added a “MainCamera”-GameObject to your Scene, which we **must** remove. Select it in the Hierarchy and press the *Delete*-Button to remove it.

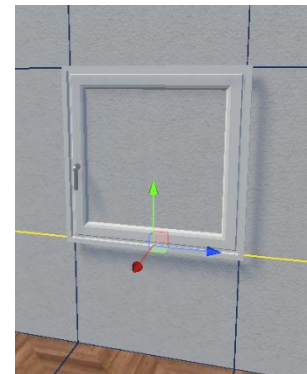
Add Room Features

Your room seems a bit empty and narrow, so let’s add some windows and a door. For those we need some models which are already included in the project under “Assets/MyRoom/”. Add a door and one window to the room by dragging them from the Project Explorer into the Scene View. Position the door at (0, 0, 1.98) with a rotation of (0, 180, 0) and the window at (-3.95, 0.8, 0) with a rotation of (0, -90, 0).

Next, we need a way to cut holes in the wall for the windows. The door will not be functional, so we skip this step for it.

Enter ProBuilder’s “Edge Selection”-Mode and select the top or bottom edge of a wall where you want your window to be. Click “Insert Edge Loop” from the ProBuilder window to insert edges perpendicular to your selected edge. Drag the edge so that it intersects with the rim of the window. You do not need to be super precise here, it just should be inside the frame.

Proceed the same way to add another edge loop for the other vertical rim of the window. You will notice that the top/bottom edge is now split, so you need to select the one that intersects with the other side of the rim.



We can do the exact same thing to insert horizontal slices by selecting the left/right edges.

Once four edge loops were inserted for your window, switch to “Face Selection”. Select the face that blocks the sight of the window and press the *Delete*-Button to remove it. Switch to “Object Selection”.

To make the outside prettier, add a Skybox. Drag and drop the “WheatFieldFloor.mat” at “Assets/MyRoom/” onto the background of your Scene.

Add Exhibition Displays

Tutorial-Video: <https://www.youtube.com/watch?v=Qp2mLUXbSGM>

Let’s add some displays for displaying exhibits to the room. This is done by placing Exhibition Displays, ExPresS XR’s fully customizable object displays.

They can either display Objects or Images whilst allowing users to grab and inspect them. More context can be provided in the form of text, video, images and audio which will be automatically presented by the click of a button next to the display.



Add your first display by right-clicking in the Hierarchy and selecting “ExPresS XR > Presentation > Exhibition Display - Object Small”. Add 3 more and an “Exhibition Display – Image” for displaying images. Distribute all display in the room according to the positions from the room blueprint.

Adding Exhibits

Now that the Displays are placed, you can begin filling them with information. The display will automatically react to your edits and will do it's best to display the information appropriately.

For now, we will be filling only two of the five Exhibition Displays. Later you will learn how to create your own exhibits from real world objects and add the remaining exhibits.

Find the *RomanCoin* prefab at “Assets/MyExhibits/” and drag it into the “Displayed Prefab”-property of the left-most “Exhibition Display - Object Small”. The coin will now be displayed on the display.

Enable “Spin Object” to automatically spin the coin in the display and enter **Roman Coin** as “Label”.

Lastly, you can copy and paste this text as “Info Text” (P.S. Use *Ctrl + C* to copy and *Ctrl + V* to paste):

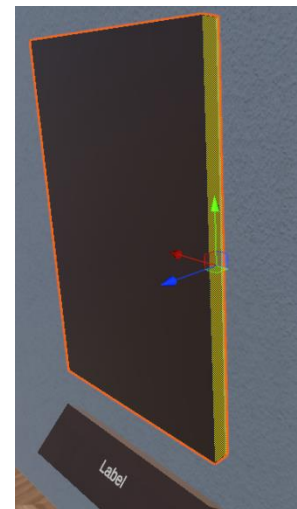
Roman coins we more than just a mean of payment.

Using letters and symbols, Romans had their own “language” for coins. The depictions told the people how an emperor shall be perceived and what their achievements were.

This completes the first display, let's continue with the second one:

Select the “Exhibition Display – Image” and add the *Minting Image*-Prefab from “Assets/MyExhibits/”. The image will be placed automatically in the frame.

As you can see, it is too wide for its frame. We could simply use another model for the frame, but for our case we can edit it in Unity. Change to „Face Selection“ at the top of the Scene View and click on the left or right side of the picture frame. Use the red arrow of the Move Tool (Press: W) to move the face horizontal, making the frame wider. Try to hit match the dimensions of the image. Change back to „Object Selection“ and move the frame to fit the image. If you are not satisfied, switch back to „Face Selection“ and change the dimensions of the frame again until you are satisfied.



Make sure that “Spin Object” is disabled. Then finalize the second display by setting the text of the “Label” as **Minting Ancient Coins** and adding the following “Info Text”:

Roman coins were created from pre-shaped disks made from metals like, gold, copper or bronze.

These "mints" were then clamped between two dies and struck with a hammer. This served the purpose of stamping the coin's face from the die onto the coin.

After striking the coin, another worker added details like the curls of hair or lettering, to make sure the coin was perfect.

Image courtesy of: <https://www.thecollector.com/ancient-roman-coins-how-were-they-made/>

Practical Session 2

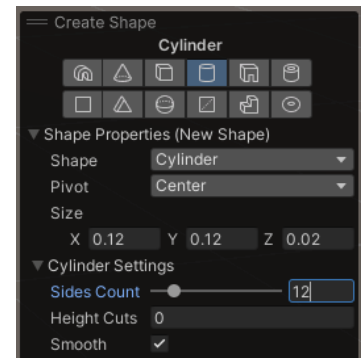
Creating own Exhibits

Of course, the goal is to also add the scanned object to the exhibition. We added the results of an earlier scan at "Assets/MyScan/".

Create a new empty GameObject by right-clicking in the hierarchy and choosing "Create empty". Make sure it is positioned at position **(0, 0, 0)** which can be changed through the Inspector on the right side. Also name it "My Scan Prefab". Select the "MyScanModel" and drag it on-top of the empty GameObject. Rename the model to "Mesh" and set the "Scale"-property in the Inspector to **(0.005, 0.005, 0.005)** to scale the model to a more appropriate size.

Next, we need collisions. This could be generated from the mesh itself. This is too expensive and prone to errors, as our model has a huge amount of complexity whilst being rather small. A better way is to use a simpler shape, like a cylinder. Just compare about 30,000 faces from our model over 36 faces of a cylinder.

We will first add the rough shape, its rotation and exact size can be changed later. Using ProBuilder's "New Shape" to add a cylinder. To make it rounder expand the Shape Properties in the dialog on the bottom right of the Scene View. Select the Cylinder shape, set the "Side Count" to **12** and the "Pivot" to **Center**. Set the size of your shape to be **(0.12, 0.12, 0.02)**. Then hold shift and press the left mouse button in the Scene View to add the cylinder.

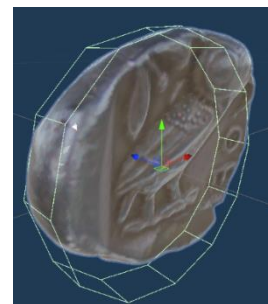


Rename the cylinder to "Collision" and drag it in the Hierarchy over the "My Scan Prefab" object to make it a child.

Change the Cylinder's position to **(0, 0, 0)** and its rotation to **(90, 0, 0)** to roughly cover the model.

Find the cylinder's "Mesh Renderer"-Component in the Inspector. Disable it by unchecking the checkbox next to "Mesh Renderer".

This component could be added to your Exhibition Display, but you would not be able to grab it. Add a **RigidBody**- and a **XRGrabInteractable**-Component to "My Scan Prefab" by using the "Add Component"-Button at the bottom of the inspector. These two components let you grab and throw your object.



Save the prefab by going to the "Models" folder in your "Project Explorer" and dragging your "My Scan Prefab" into it. Delete the Object in the Scene View, so we can fill the exhibition displays in the next step.

Fill the Exhibition Displays

We went ahead and prepared some information for your scan and the remaining exhibits in Table 1.

All Prefabs are stored under "Assets/MyExhibits/". Their additional information is placed in the sub-folders named after the Object.

Choose some information and objects and drag the prefabs into the field for „Displayed Object“ of the Exhibition Displays. The "Object-Name" can be entered as "Label"-Property of the Exhibition

Display. Proceed with the info text if you decide to use it. Images or audios can be adding by locating them in the Project Explorer and drag them in the respective slots.

To see and format your information find the “Info Text Canvas”-child of your Exhibition Display and enable it via the Inspector. Make sure to disable it when you are done.

Object-Name	Info-Text
My Scan Prefab (Greek Coin)	This coin was made in Athens, which can be derived from the owl symbol and the letters ΑΘΕ. This abbreviation was short for “of the Athenians”. The face shows Athena, goddess of Wisdom, Warfare and Handicraft and patron of Athens.
Roman Coin	This gold coin shows a roman emperor named wearing a laurel wreath. The bust depicts “Antonius Pius” who reigned the Roman Empire between 138 and 161 AD.
Another Roman Coin	Besides the name and bust of the emperor, the face has what seems to be a random sequence of letters. The legend is a list of abbreviations, each standing for one title of the emperor. Some examples are “CEAS” for Caesar or “AVG” for Augustus – both meaning emperor. The allegories and divinities on the reverse are used to indicate the state of the Roman Empire and or a trait of the emperor. Some examples are peace, wealth, luck or intelligence.
Denarius	This is a roman Denarius. The front shows the head of Rome, wearing a helmet, whilst looking to the right. The face also has a “X” – or rather the Roman numeral for 10 – imprinted in the lower right. This indicated that the coin was worth 10 acres, the less valuable bronze coin of that area. The back shows a chariot as a symbol of victory.
Flat Greek Coin	This suspicious coin looks deceptively real, but it is just an image projected on a cylinder. It cannot recreate the surface or reflections of a real coin. 3D-Scans on the other hand recreate the surface. This adds depth to the model, making it a great opportunity for handling ancient coins without deterioration or a risk of destruction.
Onager	The Onager was a roman catapult used roughly between the 4th and 6th century. This siege weapon got its power via torsion. It was generated by placing the arm between multiple ropes. Twisting the ends of the of ropes applied a force to the arm and the frame. When the arm is released, the stored energy is suddenly converted to kinetic energy, propelling the arm and its payload forward.
Mughal Coin Die	This coin die produces the obverse for coins of Bahadur Shah II, Mughal Emperor 1837-1857. He was the last Mughal Emperor of India.

An Exit

Let’s add something special to exit the game. ExPresS XR comes with an “Exit Game Interactable”, a reactive sphere that will close the game when being grabbed. Instance it by right-clicking in the Hierarchy and select “ExPresS XR > Interaction > Exit Game Interactable”. Place it over the doorknob of the door you placed earlier.

To mark the door as an exit, place an Exit Sign over it. One can be instantiated by right-clicking in the Hierarchy and selecting “ExPresS XR > Misc > Exit Sign”.

Building the Game

Tutorial-Video: <https://www.youtube.com/watch?v=t1F34SB1d1A>



To play your exhibition without Unity and for you to take them home, we will need to build it. This will create the app we can play outside of Unity.

Head to "File > Build Settings..." at the top-right of the Unity window. Depending if you want to use it on a headset for Android (Quest) or an PC (Valve Index, HTC Vive) we need to select the build platform. Your project should be configured to build for Quest 2 which is indicated with a Unity-Symbol next to "Android". If you need to change your platform, select it in the list on the left, press "Switch Platform" at the bottom and wait until it is done.

Uncheck the "ExPresS XR/Scenes/General Export Scene" and press the "Add Open Scenes"-Button. This will build the game using with your current exhibition.

Open the Player Settings with the button at the bottom-right in the Build Settings window. Enter your name(s) as "Company Name" and **My Spring School Museum** as "Product Name". Both will be the displayed for your app.

Go back to the Build Settings and hit "Build", Select the "Builds/" folder within your project and enter a name for the app. This will finally start the build process and after a few minutes we should be almost ready to play the application.

- **Windows:** Start SteamVR if it is not running and start your build application.
- **Android:** We need to transfer the application to your Oculus using the Meta Quest Developer Hub. Connect your Quest via USB and allow the connection within the headset. In the Developer Hub, you should find your Oculus under the Devices-Section. Simply drag and drop the *MySpringSchoolMuseum.apk* on to the right side of the Developer Hub window. To start your VR-exhibit, go to your app library and select the exhibition under the section "Unknown Sources".

Congratulations, you've created your very first VR Exhibition!

VR Controls

