Forge 2023

Introduction to Unity

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

Please read the following paragraphs if you are not having any experience with Unity. They will establish the most important concepts of this program. We provide positions and rotations for everything that needs to be placed in your exhibition. These values can be entered via the “Transform”-Component in the Inspector at the top right.

If you want to have a look at the final scene or want to have a digital version of this handout you will find them under the folders “Assets/_CompletedExhibit” and “Assets/_Handout” respectively.

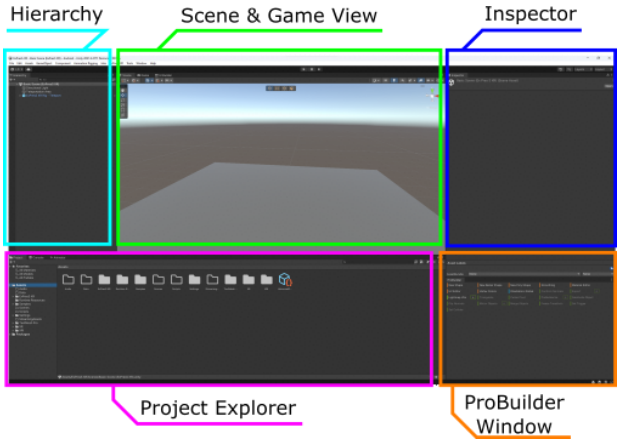
Preliminary Information

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.
  The top-most Component in the Inspector is the Objects Transform that specifies the position, rotation, and scale of the GameObject. We will be using this to position the objects in our room.
- **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" at the top of the Unity-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.

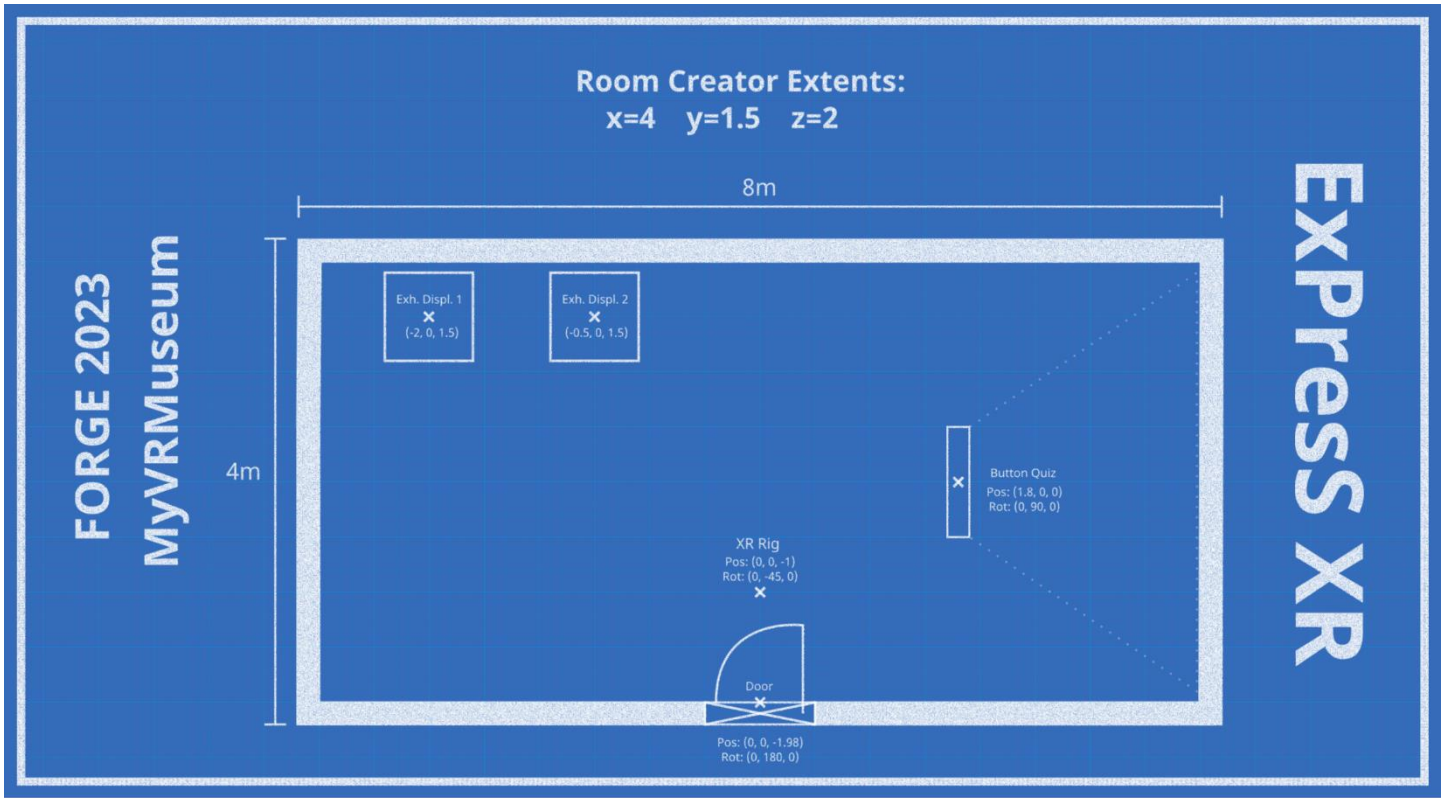
We will not explain every action as there is plenty and we only need a few. The necessary ones will be explained when they are needed.

Introduction to 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 a 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. At the top of the Unity Editor Window 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 “MyVRExhibition”.

Create a Room

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

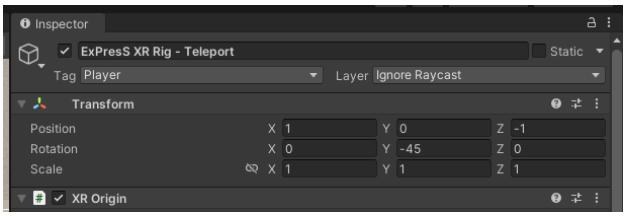
Before you can display your exhibits, we need a location for our virtual museum. In our case, this will be a single room that we can create using ExPresS XR’s RoomCreator. You will find it under at the top under "ExPresS XR > Rooms... > Open Room Creator". Here you can specify the position and extents 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”.



Set its position via the Transform Component in the Inspector at the top right to **(0, 0, -1)** and its rotation to **(0, -45, 0)**.

To let users only interact with objects directly, the “Interaction Options” must be changed. Select the XR Rig from the hierarchy on the left. Then on the right find the section for the “ExPresS XR”-Component 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**.

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

Tutorial-Video: https://youtu.be/MdBKWcyNF4w?t=369

Your room seems a bit empty and narrow, so let’s a door. We need a 3D-model for it, which is already included in the project under “Assets/MyRoom/”. Add it to the room by dragging it from the Project Explorer into the Scene View. Position the door at **(0, 0, 1.98)** with a rotation of **(0, 180, 0)**.

Adding windows and hallways would be also possible but will be skipped for time’s sake.

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” and set its “position” to **(-2.5, 0, 1.5)**.
Add second one at **(-1, 0, 1.5)**.

Enable ”Toggle Info”, so that the additional information is being toggled. Otherwise, the information would disappear automatically after a few seconds.

Be careful when clicking on the Displays in the Scene View as this will usually select sub-objects of the Displays but **not** the full Display. Changing their positions might mess up the visuals of the Exhibition Displays. Better select it from the Hierarchy on the left to be sure.

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 one of the two Exhibition Displays. Later you will learn how to create your own exhibits and add the remaining exhibit.

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.

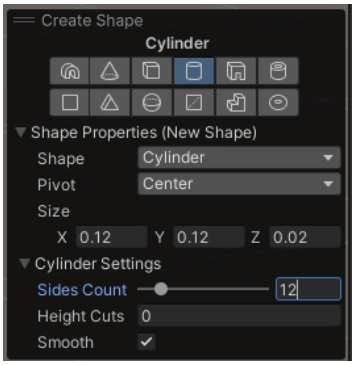
This completes the first display.

Creating own Exhibits

Of course, the goal is to also add the scanned object to the exhibition. We added only the results of a 3D-scan at "Assets/MyScan/". Nowadays these scans can be created using a smartphone or can be downloaded from Sketchfab and such.

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 this cylinder.

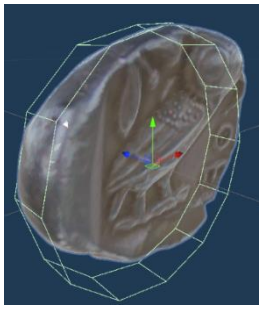


We will first add the rough shape, its rotation and exact size can be changed later. Using ProBuilder's "New Shape" to add a cylinder. To make it rounder expand the Shape Properties in the dialog on the bottom right of the Scene View. Select the Cylinder shape, set the "Side Count" to **12** and the "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 **ScalableGrabInteractable**-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.

Find and **enable** the "Dynamic Attach" option in the Section "ScalableGrabInteractable" of the Inspector. This lets you grab the coin on its surface rather than in a fixed point at its center.

Save the prefab by going to the "Assets/MyExhibits/"-folder in your "Project Explorer" and dragging your "My Scan Prefab" into it. Delete the Object in the Scene View, so we can display it in another exhibition display.

Select the prefab from the "Project Explorer" and drag it into the "Displayed Prefab"-Slot of the second, still empty Exhibition Display. The "Label" should be set to **Greek Coin** and the "Info Text" to:

This coin was made in Athens, which can be derived from the owl symbol and the letters AΘE. This abbreviation was short for "of the Athenians".
The face shows Athena, goddess of Wisdom, Warfare and Handicraft and patron of Athens.

Create your own Quiz!

Tutorial-Video: https://www.youtube.com/watch?v=-k2wBBZ9a1w

One of the most common elements of knowledge validation in exhibitions is the use of quizzes. Sadly, younger visitors often dislike this way of getting tested since it reminds them of exams. To provide more modern types of quizzes and thereby hopefully motivate the audience to participate, ExPresS XR contains a quiz builder.

To create your own quiz, activate "ExPresS XR > Button Quiz Setup" at the top of the Unity window.

In the first dialog, an existing quiz can be edited. As we do not want to edit but rather create a new quiz, the "Quiz Config" must be set to **None**. Continue by clicking the "Next"-Button on the bottom right of the dialog.

Next enter the following settings:

- "Quiz Mode": **Single Choice**
- "Question Ordering": **Random**
- "Number of Answers": **Two**
- "Question Type": **Text**
- "Answer Type": **Text**
- "Feedback Mode": **Always Correct**
- "Feedback Type": **Show Answers**

Proceed by clicking "Next".

Skip the third and fourth steps by clicking "Next" again, as we already created our room.

Create and set up the interactive buttons by clicking "Create Buttons" and proceed by clicking "Next" in the fifth dialog.

In the sixth dialog, create the quit displays for visualizing the questions by clicking "Create Questioning Displays" and continue with "Next".

In the seventh dialog, you can add the textual question to your quiz. Since we set the number of answers to two, it is only possible to provide two answers per question. To mark an answer as correct, check the checkbox next to it. Add and remove questions using the "+"- and "-"-buttons. Add these three questions or formulate your own:

No.	Question	Answer 1	Answer 2	Correct?
1	Which emperor loved his horse so much that he wanted to make it one of his consuls?	Gaius Julius Caesar	Caligula	2
2	Why did wealthy romans always have a feather at their dining table?	To throw up	To sign contracts	1
3	Which law was put in place by Gaius Julius Caesar?	Prohibition of being taller than him	Mandatory military service	1

Proceed by clicking "Setup".

In the eighth dialog, you can save your quiz to load and edit later. Replace "Config.asset" with "MyQuiz.asset" and click on "Save Config". Skip the ninth dialog by clicking "Finish".

You can add a simple table by adding a cube. Right-click on the "Button Quiz"-GameObject in the Hierarchy and select "3D Object > Cube". Move it to **(0, 0.48, 0)** and set its "Scale" to **(1.2, 1.1, 0.3)**.

Move the "Button Quiz"-GameObject to the "Position" **(1.8, 0, 0)** with a "Rotation" of **(0, 90, 0)**.

Continue by moving the Questioning Displays a bit up and closer to the wall and the "Quiz Buttons"-GameObject (not all buttons individually!) a bit up so that they'll sit nicely on the table.

Data Gathering

Tutorial-Video: https://www.youtube.com/watch?v=IHNjMLe5WTA

Another benefit of a digital exhibition is the possibility of extracting almost any kind of value from the application. This data can be used by researchers to gain more insight into the behavior of their subjects. This can include precise values of a person's movement, eye tracking values, or, in our case, how visitors answer the questions of our button quiz.

Such a seemingly simple task does, however, require rather intricate programming for both selecting and saving the values. This is why ExPresS XR's Data Gatherer provides an easy and efficient way for extracting such data in the form of CSV tables without the need for coding.

Create a Data Gatherer by right-clicking into the hierarchy and selecting "ExPresS XR/Data Gatherer".

We want to store the data locally, so make sure that the "Export Data Type" is configured as such, and the "Local Export Path" is set to "Data/MyQuizData.csv" .

Next, unfold the "Data Bindings" list near the bottom of the inspector and add one entry by pressing the +. Unfold the new entry, then drag the ButtonQuiz-GameObject from the hierarchy on the left onto the "Target Object". Press in the "Value To Save" drop-down menu to find and select the entry "Button Quiz/GetFullQuizCsvValues()". You will notice that the "Export Column Name" will be auto-completed, as this function will already extract a multitude of values.

This already completes the data selection, but what is left is specifying when the values should be saved. We could periodically export values or each time a specific controller button was pressed. In the case of a button quiz, this will fill the CSV with duplicate values. To export values only once an answer is given, we can utilize Unity's EventSystem, which inspired the Data Gatherers data selection. Select the "Button Quiz"-GameObject on the left and find the "On Answer Given()"-Event. Add an entry via the +, then drag and drop the "Data Gatherer"-GameObject onto the slot where it says "None (Object)". Finalize your setup by selecting "Data Gatherer/ExportNewCsvLine()" in the drop-down menu.

(Optional) 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-left 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-left in the Build Settings window. Enter your name(s) as "Company Name" and **My VR 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 *MyVRMuseum.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