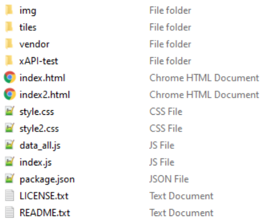
# 3D Low-cost Web-based Virtual Tour Software Framework

## Steps

## Step 1. Download Virtual Tour Framework

Step 1. Download Github repository https://github.com/Insa-Maria/Virtual_Tour by clicking on the green button “Code” and then “Download ZIP”

Step 2. Save “Virtual_Tour-master.zip” and “Extract all”

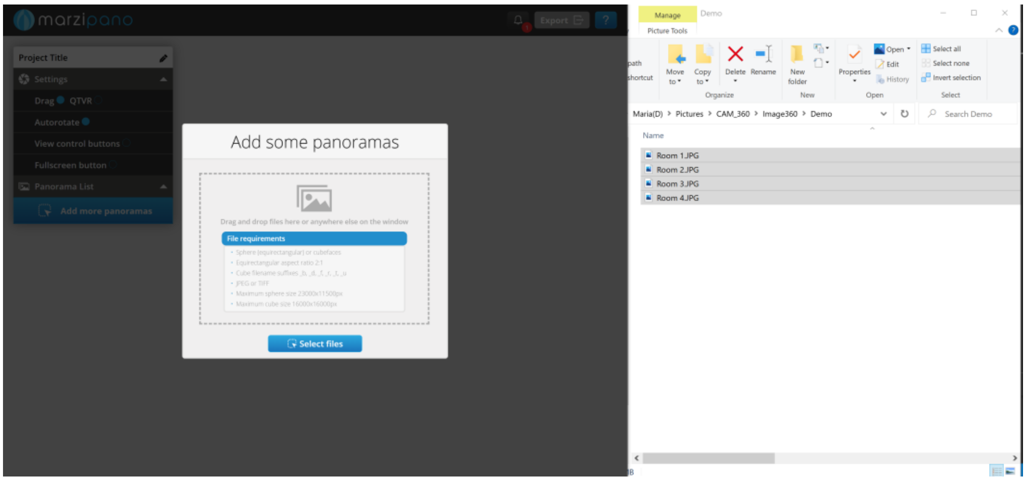


| | |
|---|---|
| img | File folder |
| tiles | File folder |
| vendor | File folder |
| xAPI-test | File folder |
| index.html | Chrome HTML Document |
| index2.html | Chrome HTML Document |
| style.css | CSS File |
| style2.css | CSS File |
| data_all.js | JS File |
| index.js | JS File |
| package.json | JSON File |
| LICENSE.txt | Text Document |
| README.txt | Text Document |

Step 3. For future reference, this folder will be called “Virtual_Tour_Framework” in this tutorial.

## Step 2. Create Virtual Tour with Marzipano Tool and 360 images

1. Navigate to https://www.marzipano.net/ (you can explore some Demo and Documentation
2. Click on “Marzipano Tool”, top left button
3. Click on “Start” and “Select files” to add the 360 images



Add some panoramas

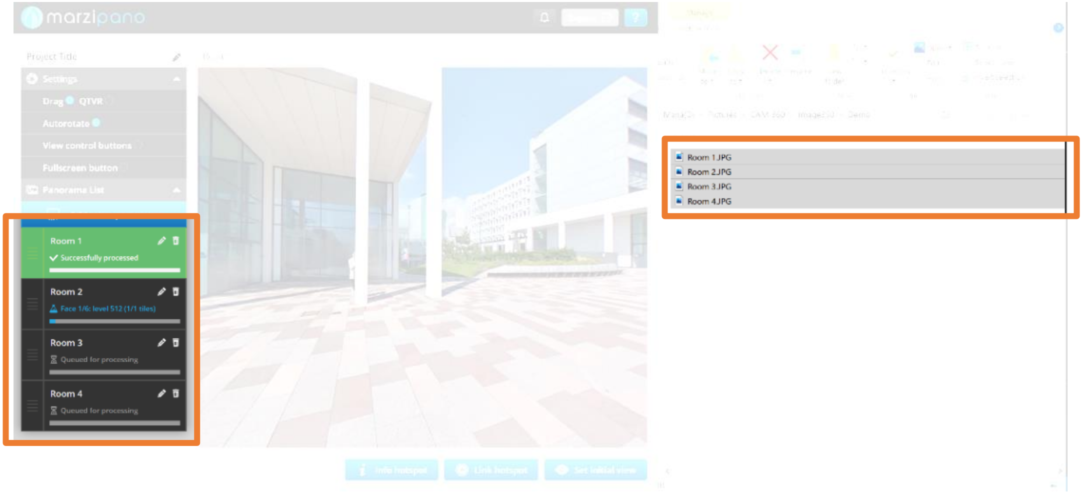
Drag and drop files here or anywhere else on the window

File requirements
- Sphere (equirectangular) or cubefaces
- Equirectangular aspect ratio 2:1
- Cube filename suffixes _b, _d, _f, _r, _t, _u
- JPEG or TIFF
- Maximum sphere size 23000x11500px
- Maximum cube size 16000x16000px

Maria(D) › Pictures › CAM_360 › Image360 › Demo

Room 1.JPG
Room 2.JPG
Room 3.JPG
Room 4.JPG

Note: By default, the file name is set as the 360image title in the web-based Virtual tour.



Room 1 ✓ Successfully processed
Room 2 Face 1/6: level 512 (1/1 tiles)
Room 3 Queued for processing
Room 4 Queued for processing

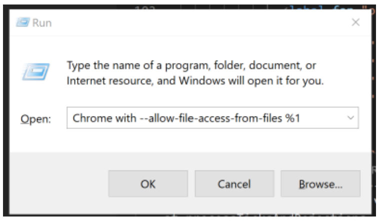
Room 1.JPG
Room 2.JPG
Room 3.JPG
Room 4.JPG

4. Set the “Project Title” and “Settings”

5. Export the project by clicking on the top left button. Save the .zip file (i.e. project-title.zip) and "Extract all”

6. Open the “project-title” folder and open the folder called “app-files”

7. To run the Marzipano Virtual Tour, right click on the “index.html” file and open with a browser (i.e. Google Chrome).
   Note: A black screen with the “info” button may appear and the menu bar may appear. To fix this error:
   (1) close ALL Google Chrome,
   (2) on your windows bar, search for “Run” app,
   (3) type “Chrome with --allow-file-access-from-files %1”



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: Chrome with --allow-file-access-from-files %1

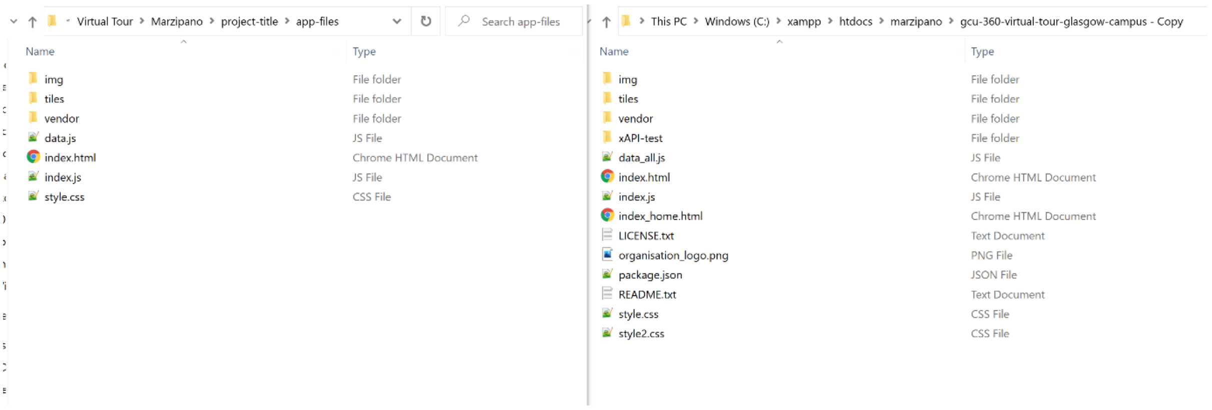
8. For future reference, this folder will be called “Marzipano_Virtual_Tour” in this tutorial.

## Step 3. Customise Virtual Tour to your Organisation

In this section, “Marzipano_Virtual_Tour” files are edited and copied to “Virtual_Tour_Framework” folder.

Open on two different windows:

1. “Organisation_Virtual_Tour” folder (downloaded from GitHub)
2. “Marzipano_Virtual_Tool” folder (exported from Marzipano Tool)

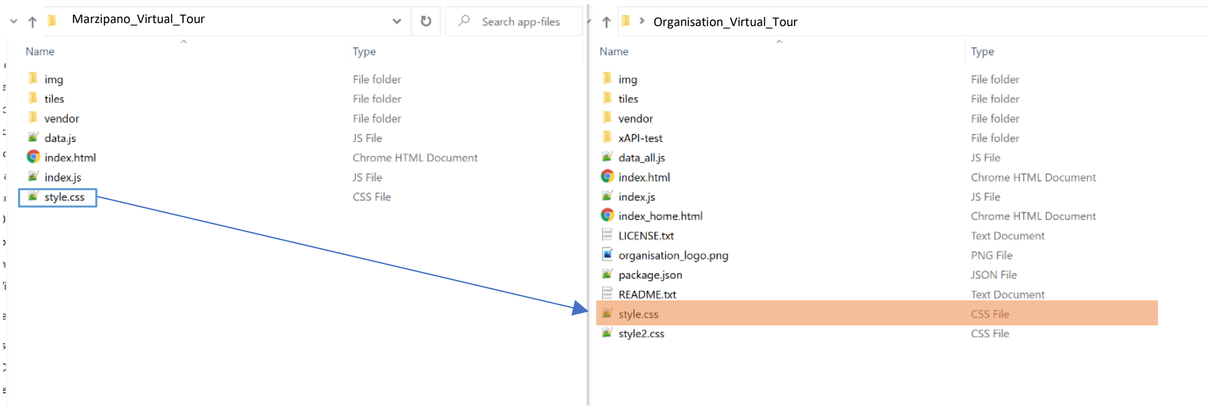

Virtual Tour › Marzipano › project-title › app-files
Search app-files
This PC › Windows (C:) › xampp › htdocs › marzipano › gcu-360-virtual-tour-glasgow-campus - Copy

| Name | Type |
|---|---|
| img | File folder |
| tiles | File folder |
| vendor | File folder |
| data.js | JS File |
| index.html | Chrome HTML Document |
| index.js | JS File |
| style.css | CSS File |

| Name | Type |
|---|---|
| img | File folder |
| tiles | File folder |
| vendor | File folder |
| xAPI-test | File folder |
| data_all.js | JS File |
| index.html | Chrome HTML Document |
| index.js | JS File |
| index_home.html | Chrome HTML Document |
| LICENSE.txt | Text Document |
| organisation_logo.png | PNG File |
| package.json | JSON File |
| README.txt | Text Document |
| style.css | CSS File |
| style2.css | CSS File |

### Step 3.1 Implement Web Design

1. Copy and Replace the “Marzipano_Virtual_Tool/style.css” file to “Organisation_Virtual_Tour” folder.

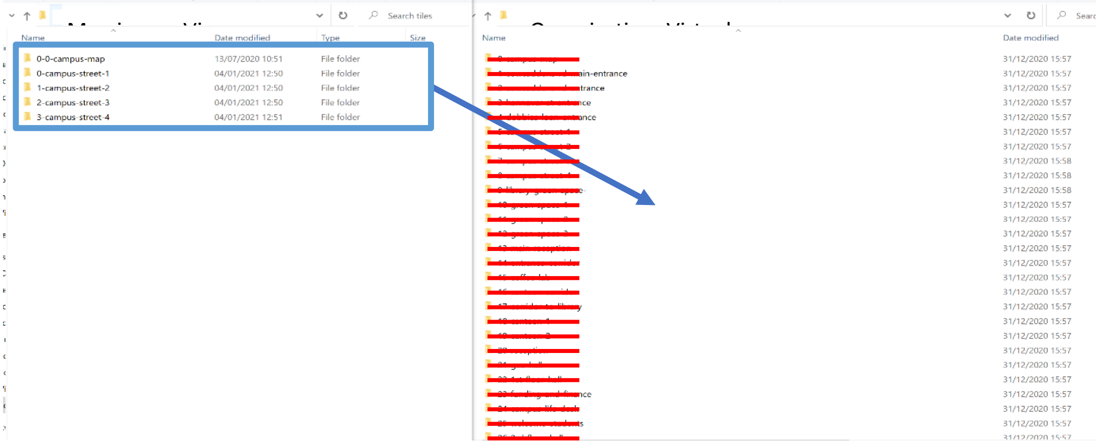


| Name | Type |
|---|---|
| img | File folder |
| tiles | File folder |
| vendor | File folder |
| data.js | JS File |
| index.html | Chrome HTML Document |
| index.js | JS File |
| style.css | CSS File |

| Name | Type |
|---|---|
| img | File folder |
| tiles | File folder |
| vendor | File folder |
| xAPI-test | File folder |
| data_all.js | JS File |
| index.html | Chrome HTML Document |
| index.js | JS File |
| index_home.html | Chrome HTML Document |
| LICENSE.txt | Text Document |
| organisation_logo.png | PNG File |
| package.json | JSON File |
| README.txt | Text Document |
| style.css | CSS File |
| style2.css | CSS File |

## Step 3.2 Integrate 360 images

1. Open “Organisation_Virtual_Tourttiles” and delete all folders. Open “Marzipano_Virtual_Tool/tiles” , copy-paste the folder to Open “Organisation_Virtual_Tour/tiles”



| Name | Date modified | Type | Size |
|---|---|---|---|
| 0-0-campus-map | 13/07/2020 10:51 | File folder | |
| 0-campus-street-1 | 04/01/2021 12:50 | File folder | |
| 1-campus-street-2 | 04/01/2021 12:50 | File folder | |
| 2-campus-street-3 | 04/01/2021 12:50 | File folder | |
| 3-campus-street-4 | 04/01/2021 12:51 | File folder | |

## Step 3.3 Integrate MaplinHotspots and VideoHotspots

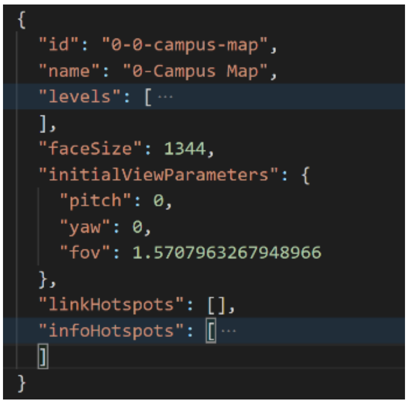
1. Duplicate your “Marzipano_Virtual_Tool/data.js” file and save it somewhere in your desktop for future reference.

2. Open “Marzipano_Virtual_Tour/data.js” file using an editor note (i.e. Visual Code).

   The data.js collect all data relevant to 360 images (called scene, according to marzipano terminology) added to the Virtual Tour with the “info” and “link” hotspots. It is coded the following way:

   - id
   - name
   - levels
   - faceSize
   - initialViewParameters
   - linkHotspots
   - infoHotspots

   The proposed system introduces two more elements: “maplinkHotspots” and “videoHotspots”.



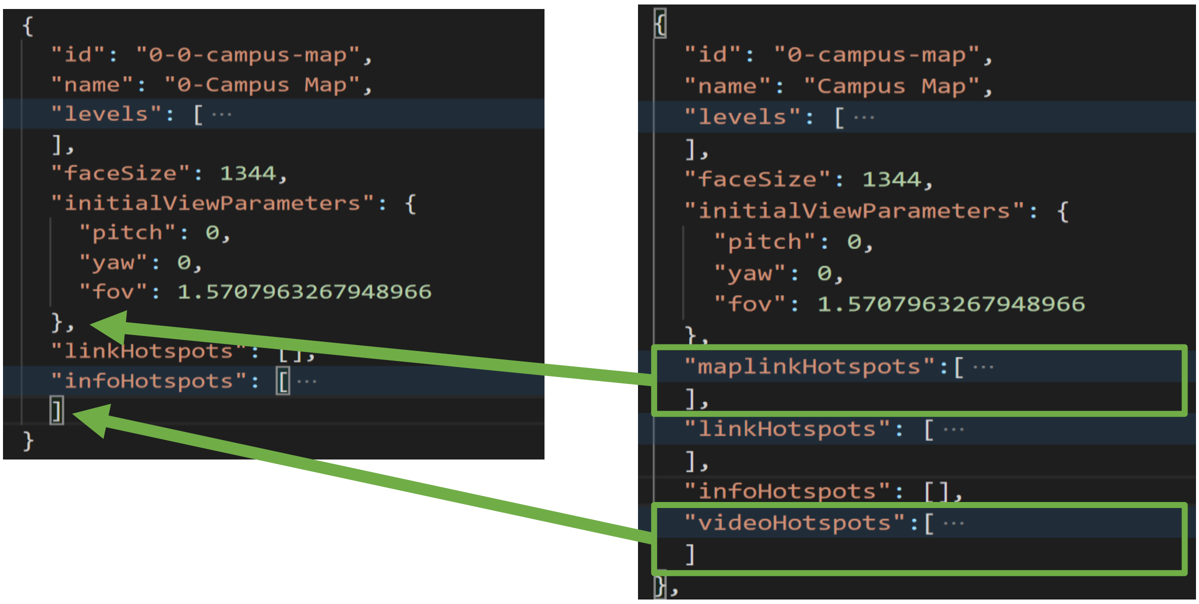
```
{
  "id": "0-0-campus-map",
  "name": "0-Campus Map",
  "levels": [ ⋯
  ],
  "faceSize": 1344,
  "initialViewParameters": {
    "pitch": 0,
    "yaw": 0,
    "fov": 1.5707963267948966
  },
  "linkHotspots": [],
  "infoHotspots": [ ⋯
  ]
}
```

3. To the “Marzipano_Virtual_Tour/data.js”, add the bellow code to <u>ALL scenes</u> (which start with “id”) after “infoHotspot”: [ … ] :

```
,
"maplikHotspots": [],
"videoHotspots": []
```

See an example bellow image:

Marzipano_Virtual_Tour/data.js        Organisation_Virtual_Tour/data.js



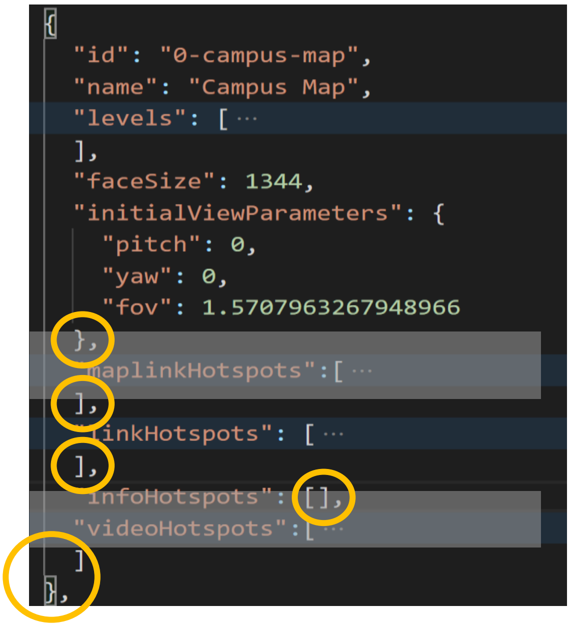
```
{
  "id": "0-0-campus-map",
  "name": "0-Campus Map",
  "levels": [ ···
  ],
  "faceSize": 1344,
  "initialViewParameters": {
    "pitch": 0,
    "yaw": 0,
    "fov": 1.5707963267948966
  },
  "linkHotspots": [],
  "infoHotspots": [ ···
  ]
}
```

```
{
  "id": "0-campus-map",
  "name": "Campus Map",
  "levels": [ ···
  ],
  "faceSize": 1344,
  "initialViewParameters": {
    "pitch": 0,
    "yaw": 0,
    "fov": 1.5707963267948966
  },
  "maplinkHotspots":[ ···
  ],
  "linkHotspots": [ ···
  ],
  "infoHotspots": [],
  "videoHotspots":[ ···
  ]
},
```

*Note: Make sure you do not mis any “,” or “() [] {}”, otherwise it will result on an error. For example, the last element “videoHotspots” do not require a “,” at the end.*

*Note: Make sure you added the code to ALL scenes, otherwise it will result on an error.*

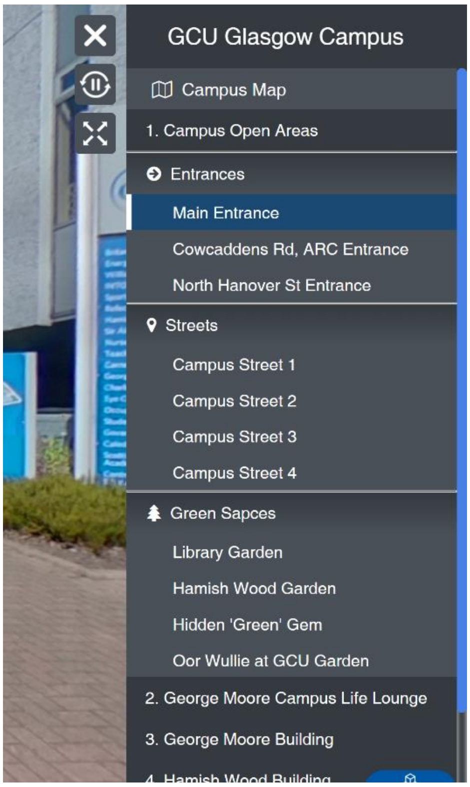
Organisation Virtual Tour/data.js



```
{
  "id": "0-campus-map",
  "name": "Campus Map",
  "levels": [ ···
  ],
  "faceSize": 1344,
  "initialViewParameters": {
    "pitch": 0,
    "yaw": 0,
    "fov": 1.5707963267948966
  },
  "maplinkHotspots":[ ···
  ],
  "linkHotspots": [ ···
  ],
  "infoHotspots": [],
  "videoHotspots":[ ···
  ]
},
```

4. Copy and Replace the “Marzipano_Virtual_Tool/data.js” file with “Organisation_Virtual_Tour/data.js”.
5. We will come back to this later, after completing Step 4

## Step 3.4 Integrate Accessible Sub-Grouped Menu

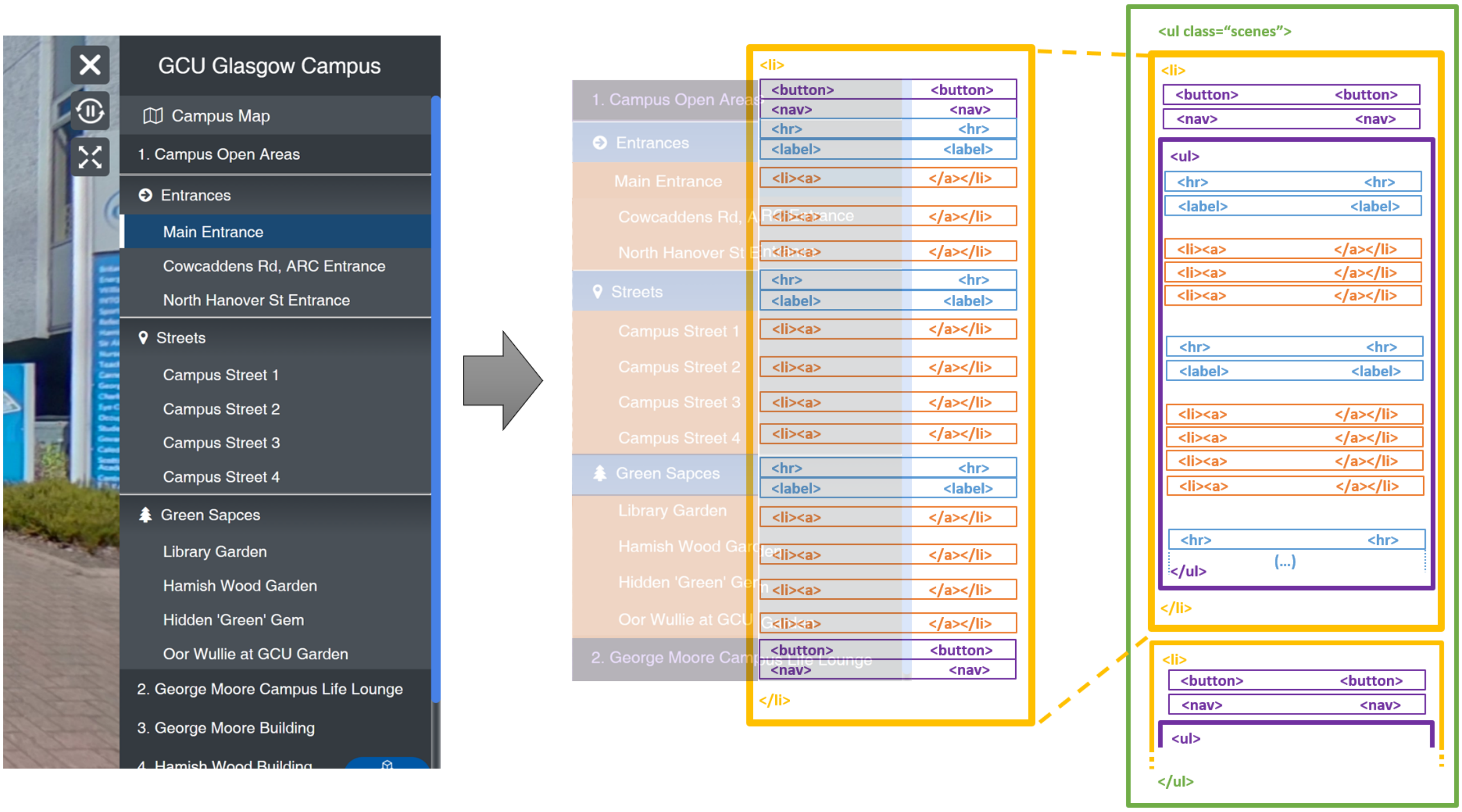


### BREVE INTRODUCTION

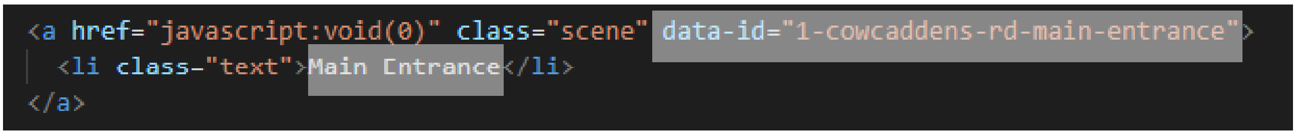
The vertical menu is created with the combination of the three files:

1. index.html (provides content),
2. index.js (provides the functionality),
3. style.css (provide the style (design))

The code structure is the following:



The resulting vertical menu follows the bellow structure, where each 360 image is coded with the following elements:



```
<a href="javascript:void(0)" class="scene" data-id="1-cowcaddens-rd-main-entrance">
  <li class="text">Main Entrance</li>
</a>
```

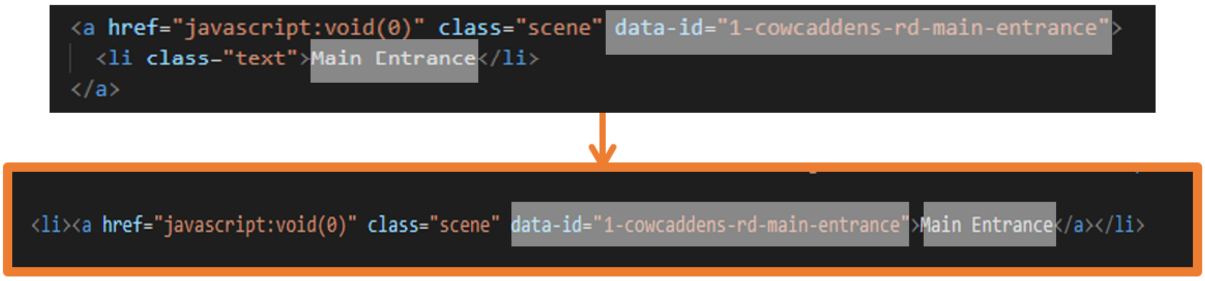
The important parts are:

- data-id: each 360 image id on the data.js file created (i.e. 1-cowcaddens-rd-main-entrance)
- 360_room_name: room name that appears on the vertical menu, which references each 360 image (i.e. Main Entrance)

STEPS

1. Open “Organisation_Virtual_Tour/index.html” file using an editor note (i.e. Visual Code).

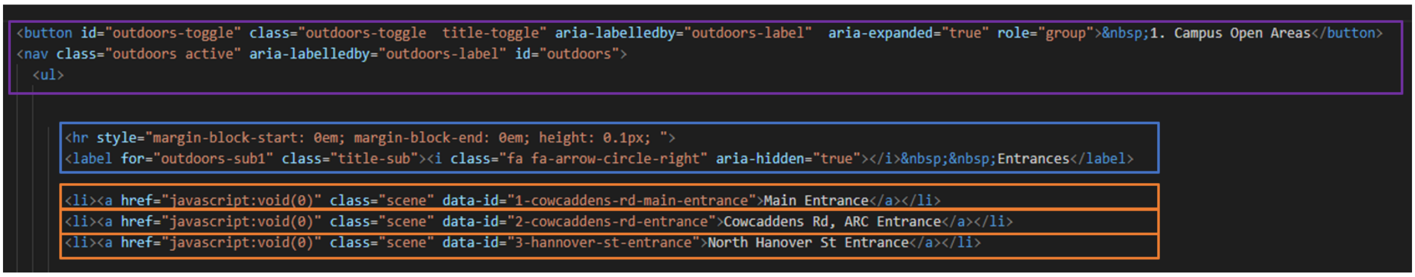
2. Re-write the code lines from <a><li></li></a> to the following way <li><a></a></li>



```
<a href="javascript:void(0)" class="scene" data-id="1-cowcaddens-rd-main-entrance">
  <li class="text">Main Entrance</li>
</a>
```

```
<li><a href="javascript:void(0)" class="scene" data-id="1-cowcaddens-rd-main-entrance">Main Entrance</a></li>
```

3. Keeping in mind the code structure (described in the introduction of this section “Step 3.4”), group the scenes by groups adding the code line (squared in **blue** and **purple**) to obtain a header:



```
<button id="outdoors-toggle" class="outdoors-toggle  title-toggle" aria-labelledby="outdoors-label"  aria-expanded="true" role="group">&nbsp;1. Campus Open Areas</button>
<nav class="outdoors active" aria-labelledby="outdoors-label" id="outdoors">
  <ul>

    <hr style="margin-block-start: 0em; margin-block-end: 0em; height: 0.1px; ">
    <label for="outdoors-sub1" class="title-sub"><i class="fa fa-arrow-circle-right" aria-hidden="true"></i>&nbsp;&nbsp;Entrances</label>

    <li><a href="javascript:void(0)" class="scene" data-id="1-cowcaddens-rd-main-entrance">Main Entrance</a></li>
    <li><a href="javascript:void(0)" class="scene" data-id="2-cowcaddens-rd-entrance">Cowcaddens Rd, ARC Entrance</a></li>
    <li><a href="javascript:void(0)" class="scene" data-id="3-hannover-st-entrance">North Hanover St Entrance</a></li>
```

## Step 4. Learning Record Stores (LRS) Set Up

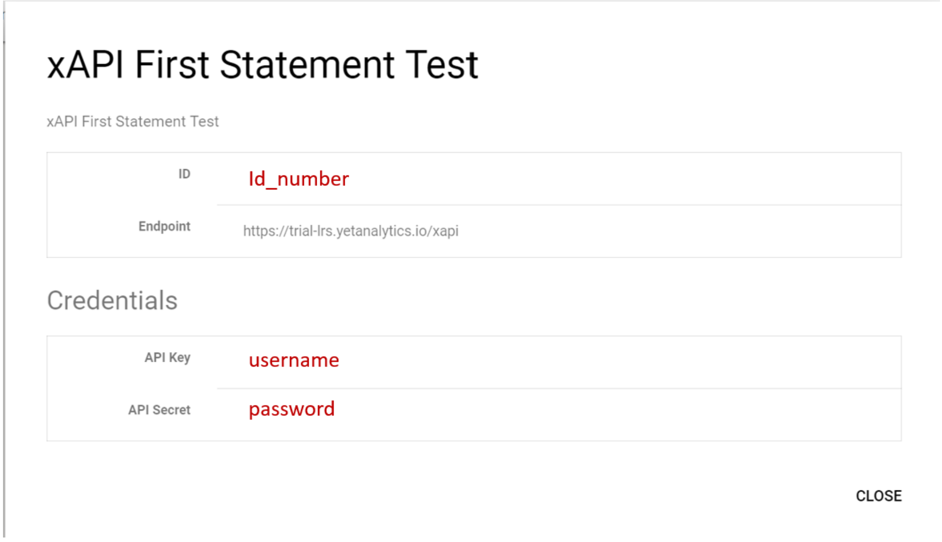
This section describes few steps to create and set up the LRS account with your Virtual Tour (created with Marzipano and the software framework proposed).

### Step 4.1 Create Learning Record Store account

Follow STEP 2 from this tutorial to set up a Learning Record Store (LRS) that can receive data from outside sources. You will:

1. Get access to an LRS (this tutorial uses YetAnalytics)
2. Set up the LRS for xAPI statements

As a result, you obtain the following credentials that you will need on the next step.



xAPI First Statement Test

xAPI First Statement Test

| | |
|---|---|
| ID | Id_number |
| Endpoint | https://trial-lrs.yetanalytics.io/xapi |

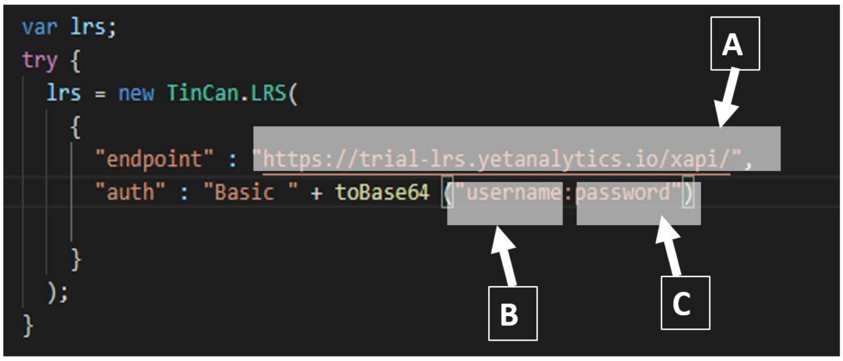
Credentials

| | |
|---|---|
| API Key | username |
| API Secret | password |

## Step 4.2 Connect your website to the LRS and send statement

1. Open “Organisation_Virtual_Tour/index.js” file using an editor note (i.e. Visual Code).

2. Modify the following code line with your “xAPI LRS” Credentials obtained in “Step 4.1”:



```
var lrs;
try {
  lrs = new TinCan.LRS(
    {
      "endpoint" : "https://trial-lrs.yetanalytics.io/xapi/",
      "auth" : "Basic " + toBase64 ("username:password")

    }
  );
}
```

A) Endpoint = The URL from YetAnalytics (mine was: https://trial-lrs.yetanalytics.io/xapi).
Note: make sure you keep the backslash!!!

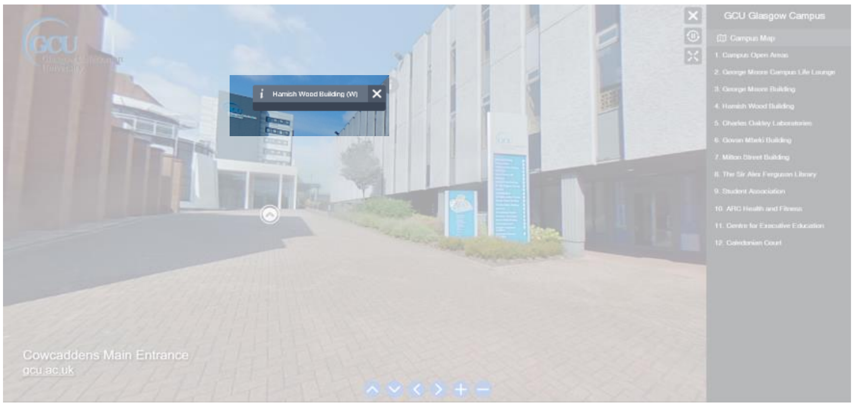
B) API Key = username

C) API Key Secret = Password
Note: before the colon, you’ll do the “API Key” and after the colon is the “Key Secret”

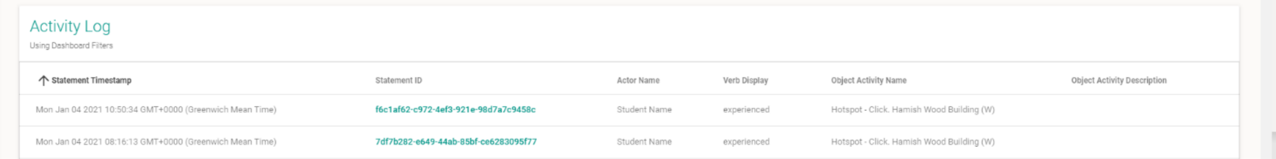
## Step 4.3 Run a Test

Now that you’ve created the tied the site to your LRS, you’ll need to trigger an event that will send that statement to your LRS

1. Run the “Organisation_Virtual_Tour/index.html”, right click on the “index.html” file and open with a browser (i.e. Google Chrome).
2. Navigate to a 360 image and click on an “Info Hotspot” element (i.e. Hamish Wood Building). See example on the image bellow:



3. Open up YetAnalytics, scroll down and you should see some analytics



Activity Log
Using Dashboard Filters

| ↑ Statement Timestamp | Statement ID | Actor Name | Verb Display | Object Activity Name | Object Activity Description |
|---|---|---|---|---|---|
| Mon Jan 04 2021 10:50:34 GMT+0000 (Greenwich Mean Time) | f6c1af62-c972-4ef3-921e-98d7a7c9458c | Student Name | experienced | Hotspot - Click. Hamish Wood Building (W) | |
| Mon Jan 04 2021 08:16:13 GMT+0000 (Greenwich Mean Time) | 7df7b282-e649-44ab-85bf-ce6283095f77 | Student Name | experienced | Hotspot - Click. Hamish Wood Building (W) | |

## Step 5. Host your Virtual Tour using Amazon S3 (AWS S3)

Note: There are many way to host your website, but this method worked for this project during the testing stage to share our work and receive feedback.

The bellow link provides an overview of how to use “Amazon S3” service to host a static website.

Check this other link to get a step-by-step tutorial on how to set Amazon S3 to web hosting. It describes how to:

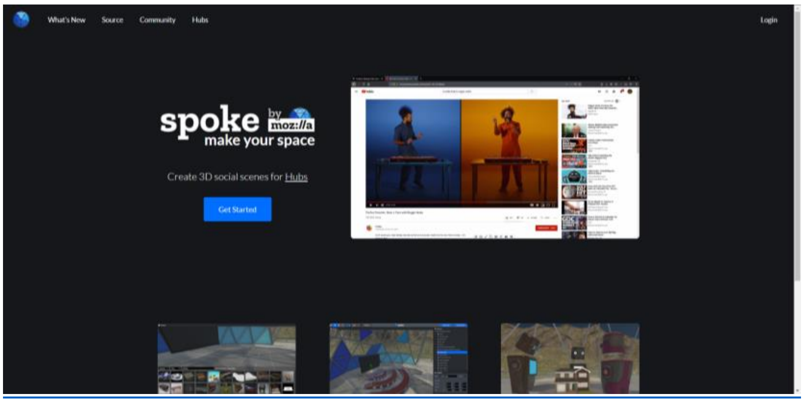
1. configure a bucket for website hosting,
2. upload a sample index document,
3. test the website using the Amazon S3 website endpoint for the bucket.

This other link shows how you can use your own domain, such as example.com, instead of the Amazon S3 bucket website endpoint, and serve content from an Amazon S3 bucket configured as a website.

## Step 6. Create 3D Environment with Mozilla Hub and Add link to

### Step 6.1 Create 3D environment and Link

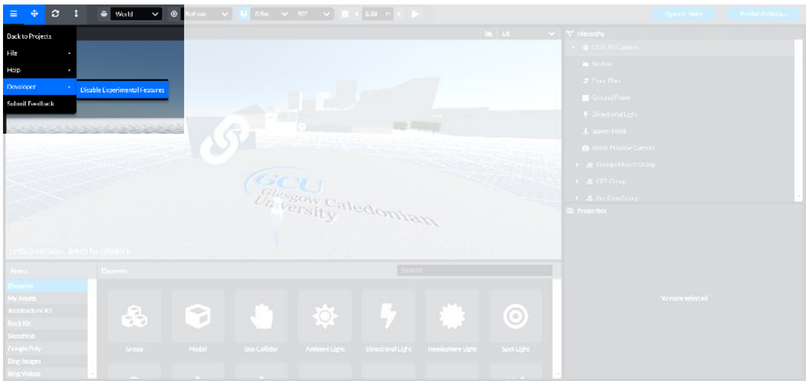
1. Navigate to https://hubs.mozilla.com/spoke



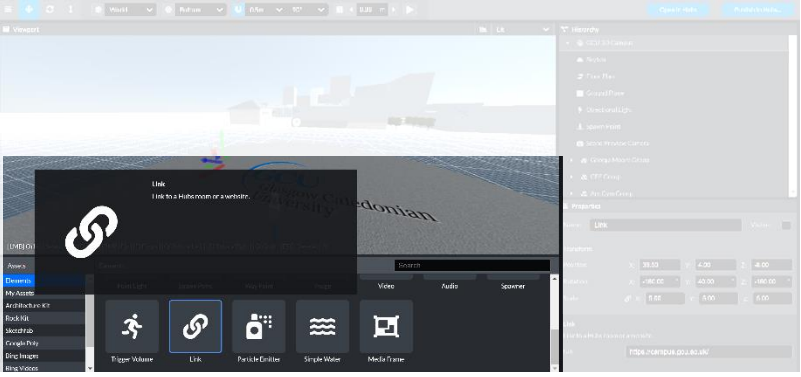
2. Click on “Get Started” and “New Project”

3. Select an environment or select the option “New Empty Project” to create a 3D environment from scratch.

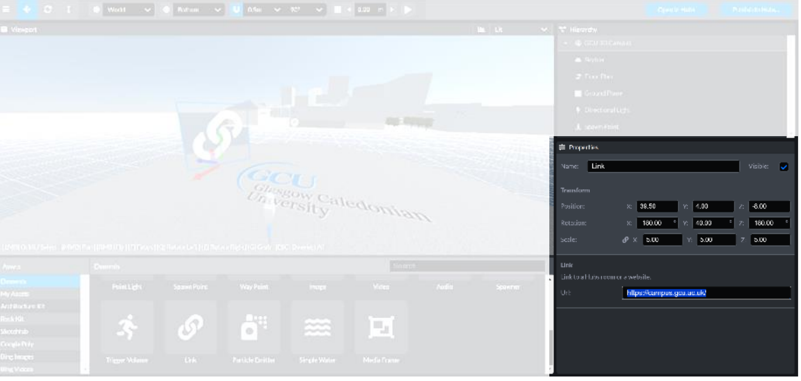
4. To add (embed) the web-based Virtual Tour, activate the “Experimental Features” clicking on the menu button (displayed as ≡) > “Developer” > ”Enable Experimental Features”



5. On the bottom navigation menu tagged “Assets”, click on “Elements” and click on “Link” to add it to the environment.



6. Set the position of the "Link" element and add the "Url" direction obtained after hosting the Virtual Tour (Step 4) (i.e. https://campus.gcu.ac.uk/).



Properties

Name: Link Visible: ☑

Transform

Position: X: 38.50 Y: 4.00 Z: -6.00

Rotation: X: 180.00 ° Y: 40.00 ° Z: 180.00 °

Scale: X: 5.00 Y: 5.00 Z: 5.00

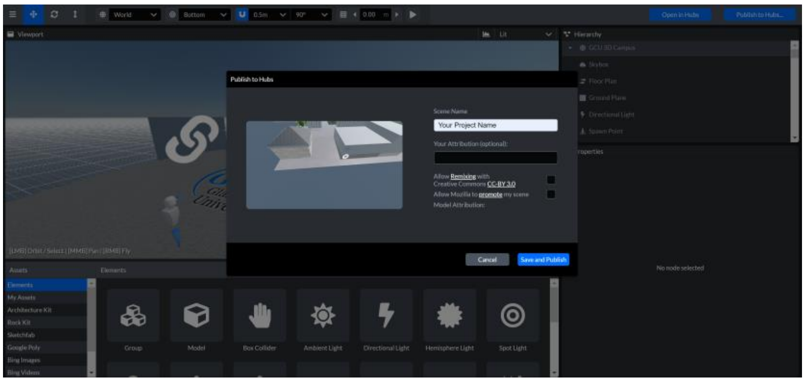
Link

Link to a Hubs room or a website.

Url: https://campus.gcu.ac.uk/

7. Click on the menu button (displayed as ≡) > "File" > "Save Project"

8. Click on the top left button "Publish to Hubs"



Publish to Hubs

Scene Name: Your Project Name

Your Attribution (optional):

Allow Remixing with Creative Commons CC-BY 3.0 □

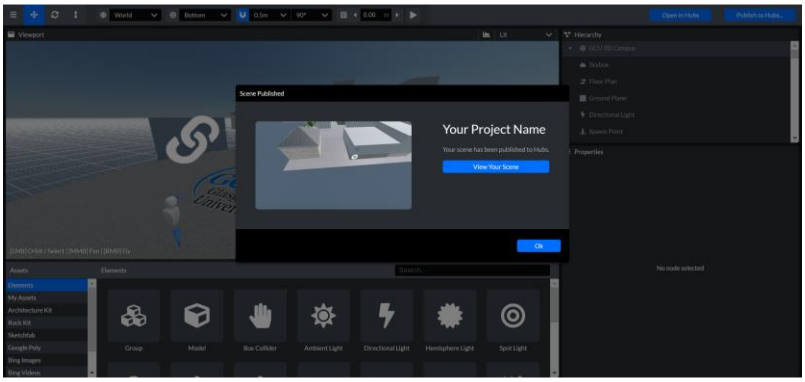
Allow Mozilla to promote my scene □

Model Attribution:

9. Set a "Scene Name" and click on "Save and Publish"

10. A "Performance Check" window appears and then click on "Publish Scene"

11. Once it has been published successfully, click on "View Your scene".



Scene Published

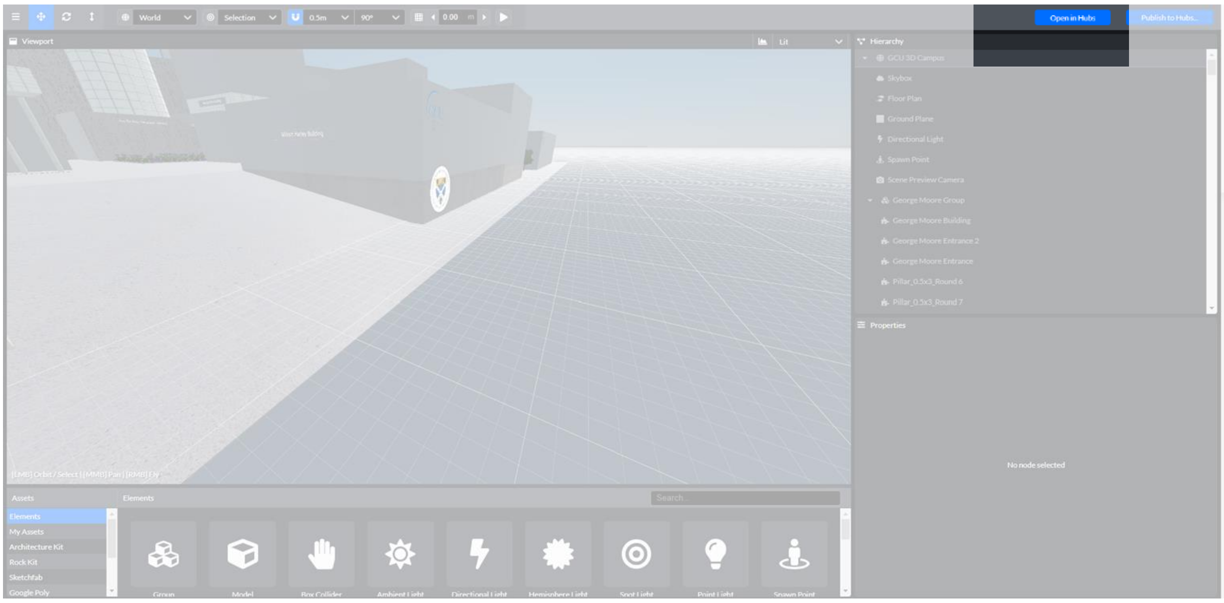
Your Project Name

Your scene has been published to Hubs.

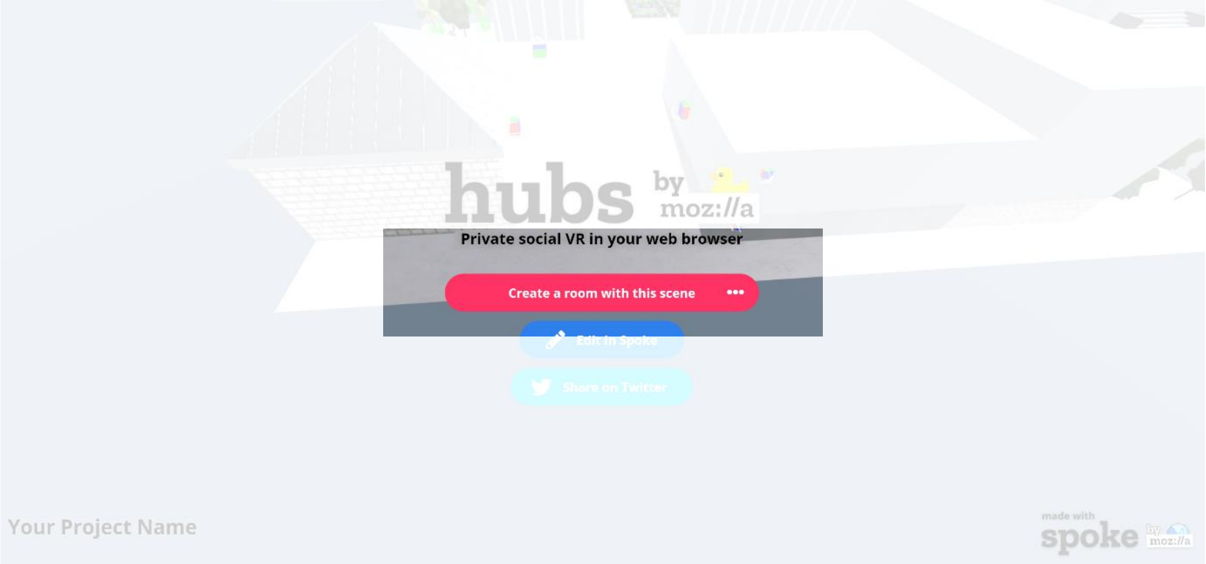
Note: you can also view your project clicking on the top left button "Open in Hubs", but you need first to save and publish it to see the last version you created.

## Step 6.2 Get Link to Allow Users Log In

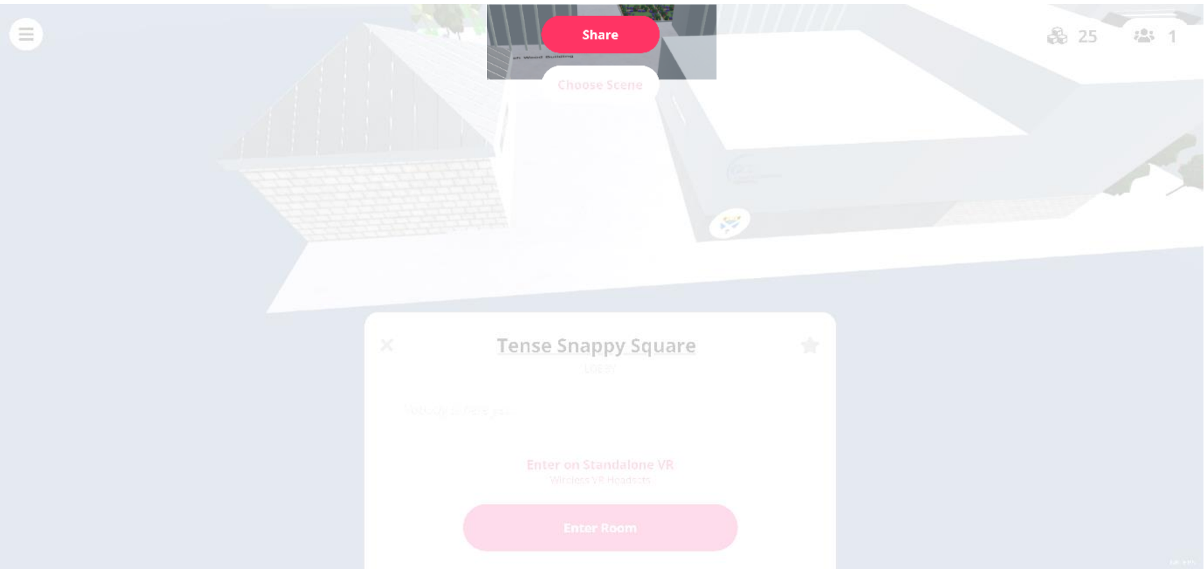
1. In Mozilla Hubs-Spoke, click on the top left button “Open in Hubs” to view your scene



2. Click on “Create a room with this scene”

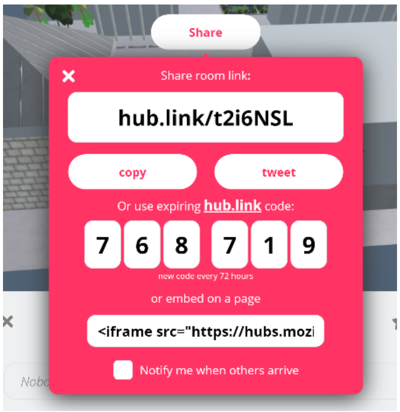

hubs by moz://a
Private social VR in your web browser
Create a room with this scene
Edit in Spoke
Share on Twitter
Your Project Name
made with spoke by moz://a

3. Click on the top button “Share”



4. Copy the text of the bottom box (starts with “<iframe sc….”).

   For example:
   <iframe src="https://hubs.mozilla.com/t2i6NSL/tense-snappy-square?embed_token=19774a93784d0919379e90d22684ceda" style="width: 1024px; height: 768px;" allow="microphone; camera; vr; speaker;"></iframe>



5. From the copied text, save for later the Mozilla Hub link (highlighted in yellow). This will be the link you share with other visitors.

   For example:
   https://hubs.mozilla.com/t2i6NSL/tense-snappy-square?embed_token=19774a93784d0919379e90d22684ceda