

Lightbox Web AR Model viewer

Author: Andro Mikulić

Project overview

Unity version	2022.3.0f1
GitHub repository	https://github.com/LightboxStudio/WebGL
Developed by	Andro Mikulić
Developer contact	andro.mikulic@toptal.com
Platform	WebGL
Deployed on	Digital Ocean

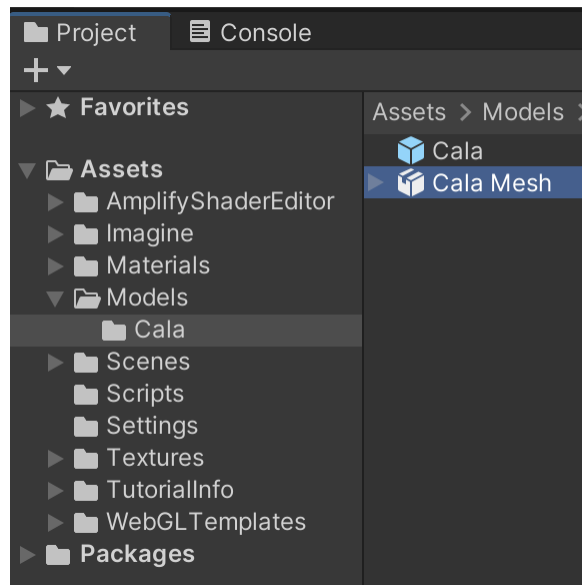
Adding a new model to the project

Step 1 – Adding the model file to the project

Estimated time: 1 minute

Add your model to the project. Currently, the models are located in the Models folder. It is recommended to put each model into its own folder.

During this step, only your model file is in the folder (in this case **Cala Mesh**)

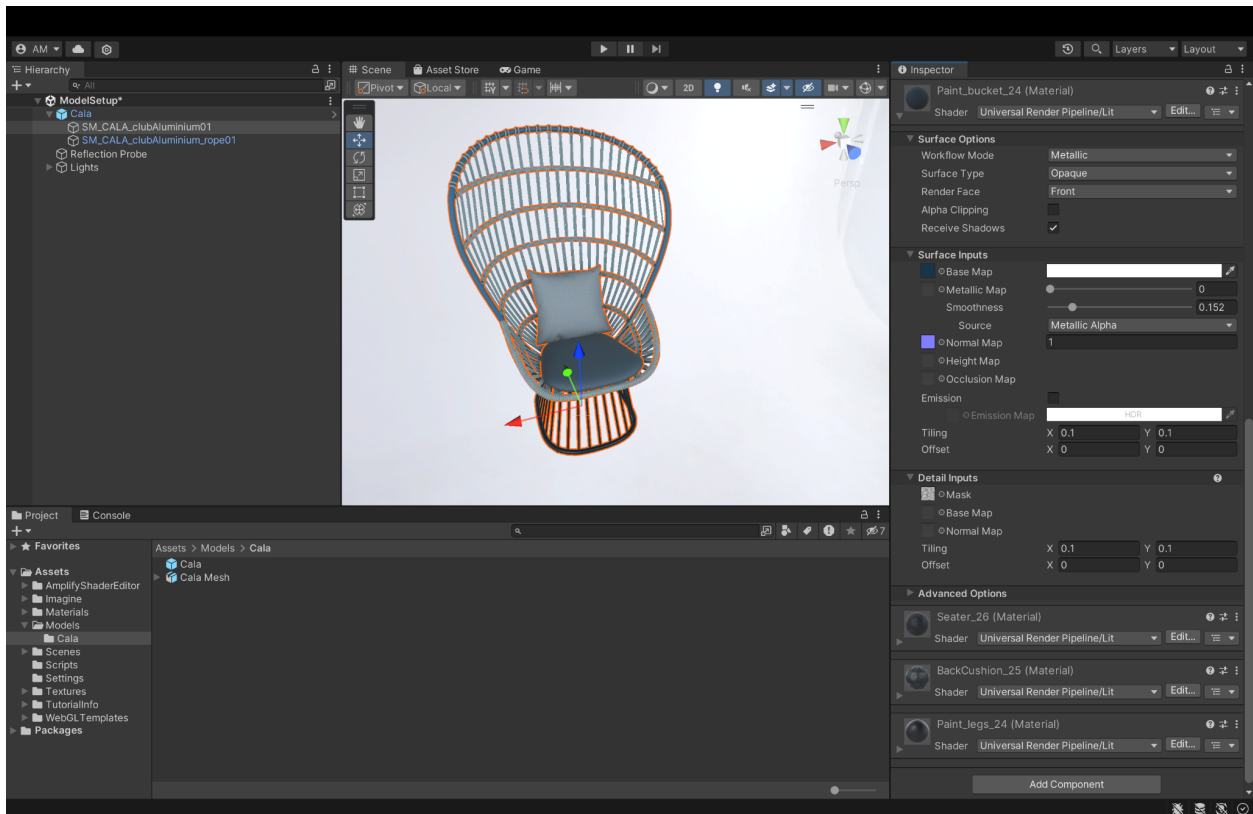
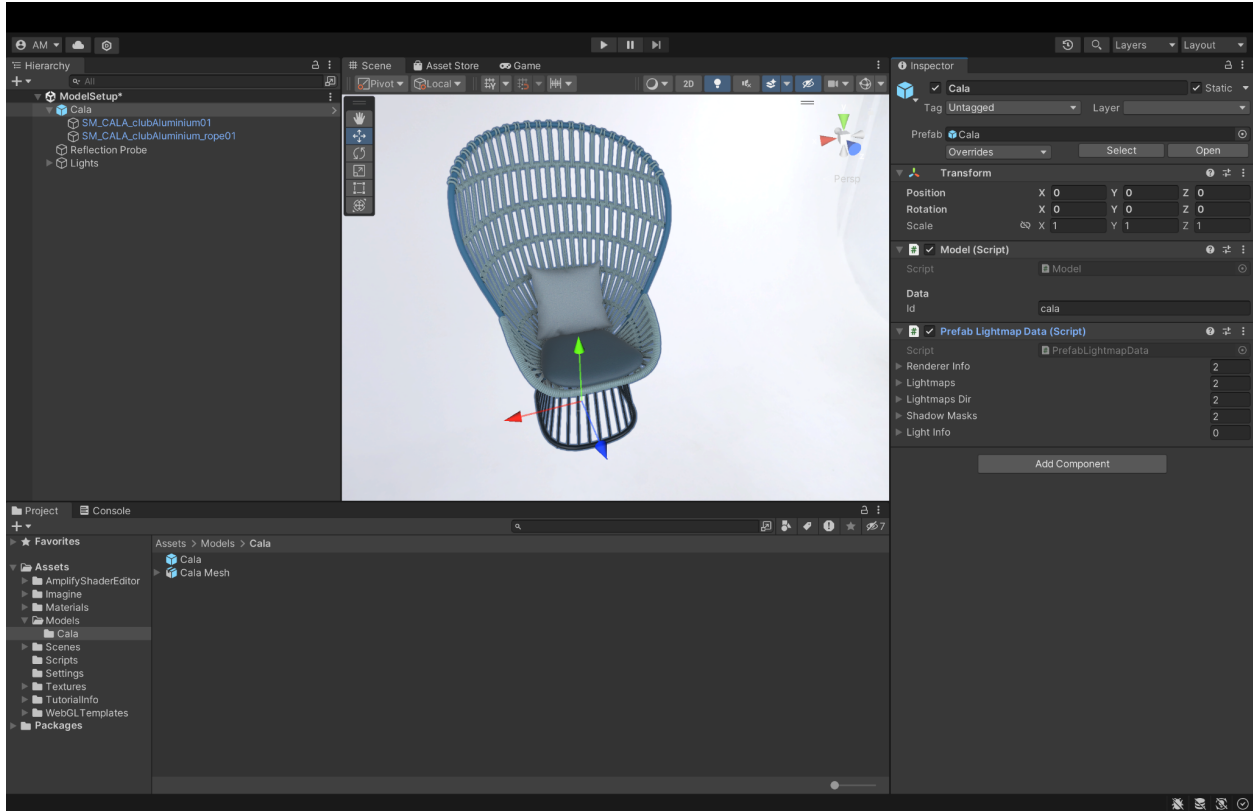


Step 2 – Preparing the model for usage

Estimated time: 5–10 minutes.

1. Open the **Model Setup** scene located in Assets/Scenes/.
2. Place your model into the scene.
3. Drag and drop it back into the folder of the model to create a prefab.
4. Extract all of the materials from the model and put them in the Materials folder. Organize them however you like.
5. Attach all of the textures to the materials that need to be added. Keep in mind: **do not use amplify shaders**. Use the **Universal Render Pipeline/Lit shader** for all materials. Apply these materials to the model as you would to any other Unity model in a scene.
6. Attach the Model script to the root of the prefab. In the ID field enter a unique ID of the model. You will need to use this ID in the configurator as well.
7. Bake the lighting.
8. From the menu go to Assets -> Bake Prefab Lightmaps.
9. Save all changes to the prefab. Save the scene.

Below you will find 2 screenshots of a working example that is the result of the steps above.

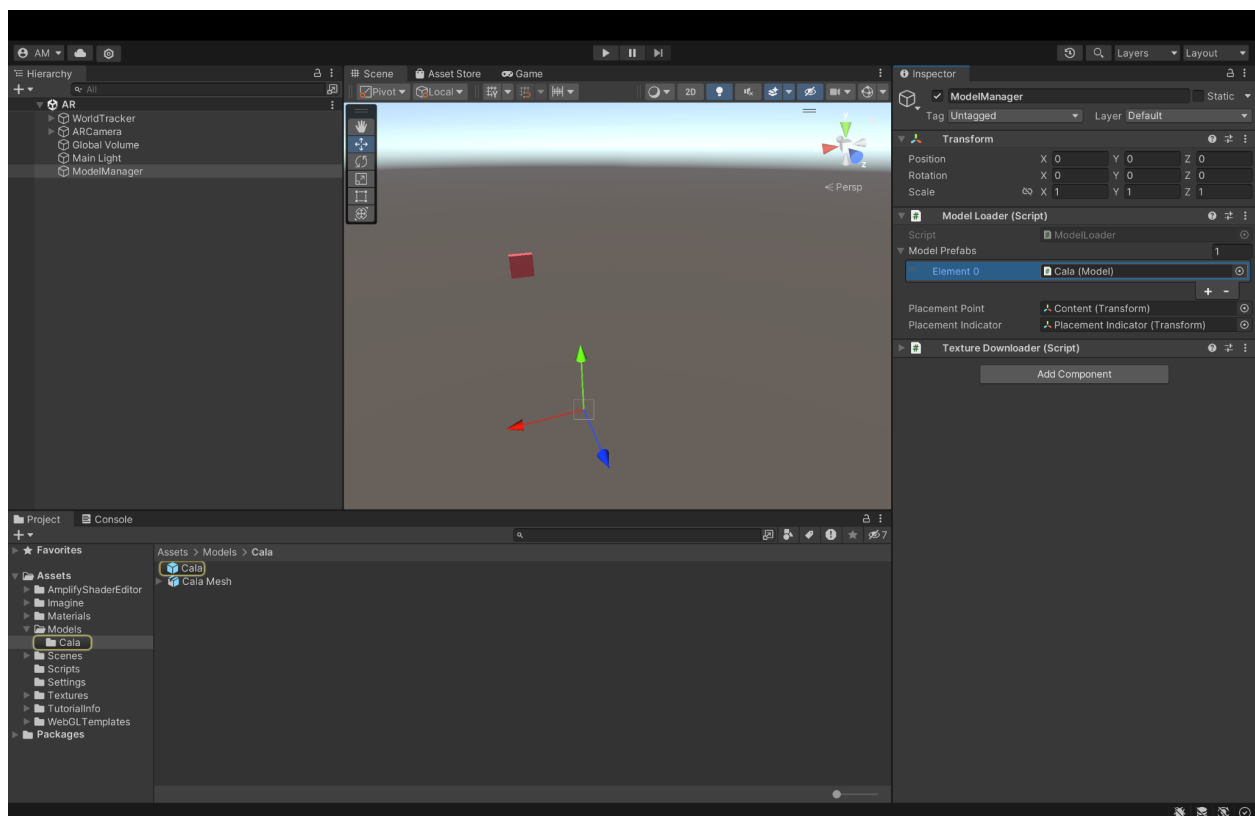


Step 3 – Update the AR scene

Estimated time: 1 minute.

1. Open the AR scene located in Assets/Scenes/.
2. Click on the Model Manager gameobject and expand the Mode Manager component
3. Add a new element to the Model Prefabs list.
4. Save the scene.

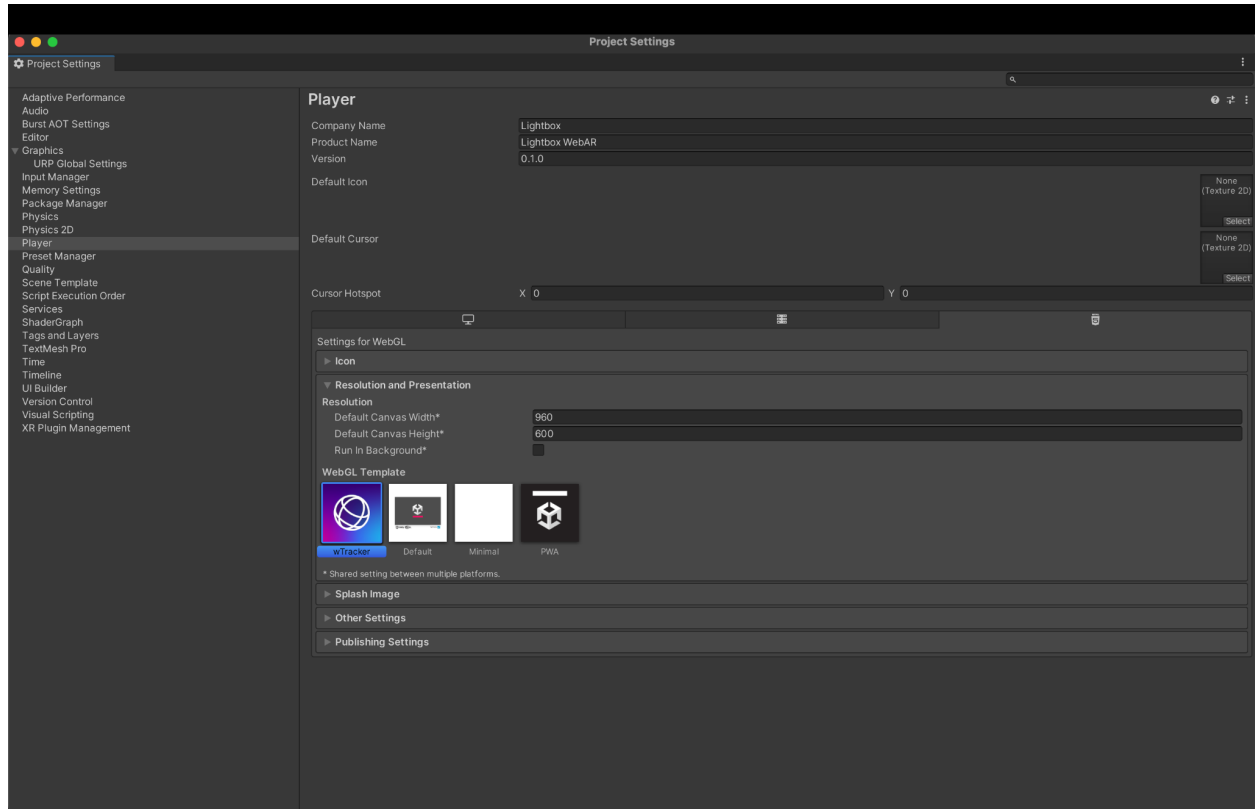
Screenshot of a working example



Step 4 – Build the new version of the project.

Estimated time: 30–60 minutes, depending on the computer being used to build.

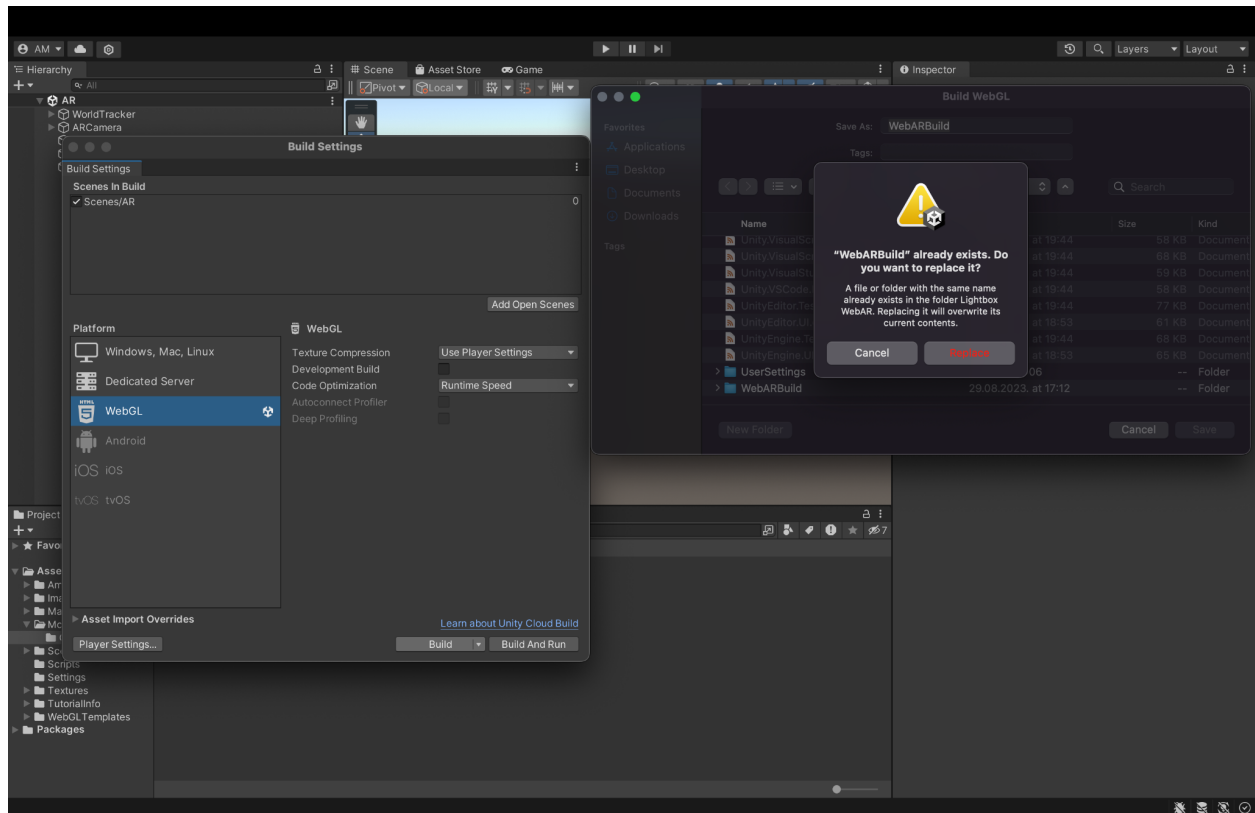
Make sure that you are building with the **wTracker** template.



Once you check that, do the following:

1. Go to File -> Build Settings
2. Click on Build
3. Make sure the name is **WebARBuild**. This is extremely important for automatic deployment set up on Digital Ocean.
4. Click on Replace to overwrite the old folder.
5. Wait for the process to finish.

Working example screenshot below.



Step 5 – Publish and deploy changes.

Estimated time: 2 minutes + about 10 minutes for Digital Ocean to create a new build.

If you need help with using git, visit this link for additional information

<https://docs.github.com/en/get-started/using-git>

To publish new changes and have them automatically deployed:

1. Add all changes via git.
2. Commit them with any message you'd like.
3. Push changes to the main branch.
4. Wait for Digital Ocean to detect a new build and automatically deploy.

Once the deployment is done, you can see it live on Digital Ocean

The screenshot displays the DigitalOcean dashboard for a project named 'walrus-app'. The interface includes a left sidebar with navigation options like 'PROJECTS', 'MANAGE', and 'Apps'. The main content area shows the 'Overview' tab for the 'walrus-app', which is currently 'Available'. It also displays 'RECENT ACTIVITY' with three deployment events, each triggered by a push to the main branch of the 'LightboxStudio/WebGL/main' repository. The dashboard also features a search bar, a 'Create' button, and a 'My Team' section.

1 of 2 open incidents MongoDB Cluster Creation Failure To learn more, [check our status page.](#)

Search by resource name or public IP (Cmd+B)

Create ? ? My Team

← Apps

walrus-app

in WebGL / walrus-app-r2e6dondigitalocean.app

Create Actions

Overview Insights Activity Runtime Logs Console Settings

APP HEALTH CHECK

Available

INSIGHTS

Global CDN

webgl-web-ar-build

Insights >

HTTP ROUTES

ESTIMATED MONTHLY APP COST

RECENT ACTIVITY

Aug 29 2023

LIVE AndroMikulic's deployment went live

- Trigger: AndroMikulic pushed 5078cf2 to LightboxStudio/WebGL/main
- 06:50:42 PM • 1m 18s build

AndroMikulic's deployment went live

- Trigger: AndroMikulic pushed 9ff812e to LightboxStudio/WebGL/main
- 06:03:11 PM • 1m 4s build

AndroMikulic's deployment went live

- Trigger: AndroMikulic pushed 5037b02 to LightboxStudio/WebGL/main
- 05:40:10 PM • 1m 6s build

Step 6 – Update your Configurator project.

Estimated time: unknown

1. Set up a scene just like you have it set up for the Cala example.
2. Add the QRCodeManager prefab located in Assets/5–Prefabs.
3. Reference the model from the scene into the Model variable in the QR Code Manager component.
4. Enter the exact same model ID you previously entered in Step 2.6.
5. Publish changes how you did previously.

Working example screenshot:

