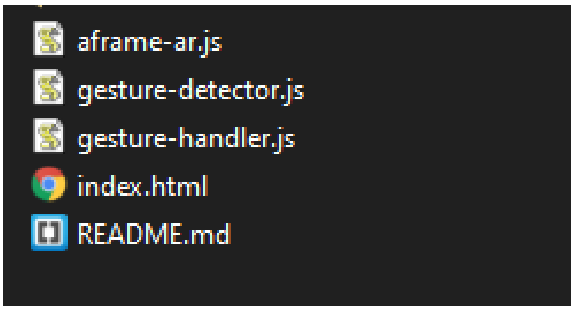
How to create Augmented Reality experience

Files/Frameworks

- Step 1: Create a folder
- Step 2: Create a index.html file
- Step 3: Include aframe-ar.js



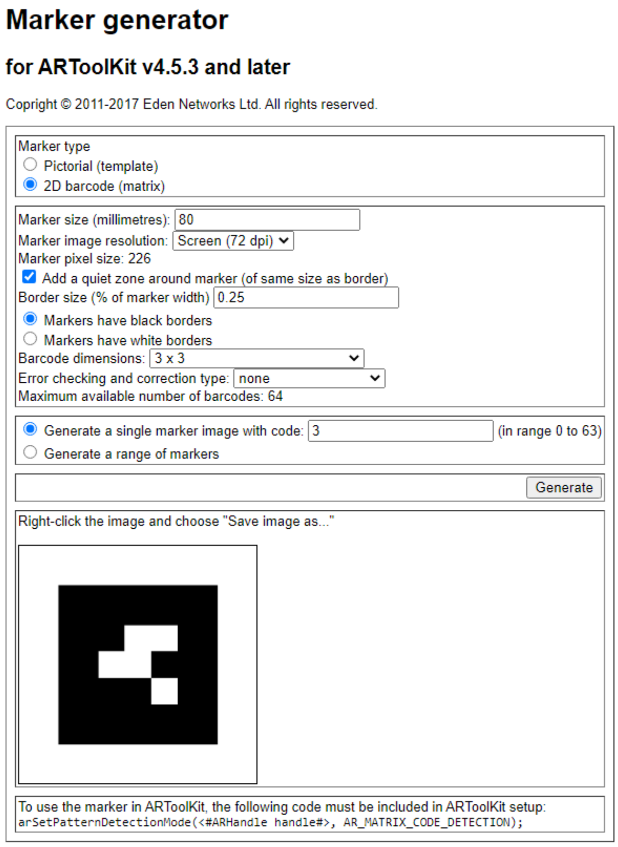
aframe-ar.js
gesture-detector.js
gesture-handler.js
index.html
README.md

Gestures

- Step 4: include gesture-detector.js
- Step 5: include gesture-handler.js

Generate a marker

- Step 1: Select 2D barcode (matrix)
- Step 2: Add a quiet zone around the marker
- Step 3: Select markers have black borders
- Step 4: Make sure barcode dimensions is 3x3
- Step 5: Select generate a single image with code value and chose any number from 0 to 63
- Step 6: Click Generate
- Step 7: Right click on the barcode and save as
- Step 8: Include detectionMode and matrixCodeType in the scene to tell AR.js to recognise barcode markers



Marker generator

for ARToolKit v4.5.3 and later

Copyright © 2011-2017 Eden Networks Ltd. All rights reserved.

Marker type
○ Pictorial (template)
● 2D barcode (matrix)

Marker size (millimetres): 80
Marker image resolution: Screen (72 dpi)
Marker pixel size: 226
☑ Add a quiet zone around marker (of same size as border)
Border size (% of marker width) 0.25
● Markers have black borders
○ Markers have white borders
Barcode dimensions: 3 x 3
Error checking and correction type: none
Maximum available number of barcodes: 64

● Generate a single marker image with code: 3 (in range 0 to 63)
○ Generate a range of markers

Right-click the image and choose "Save image as..."

To use the marker in ARToolKit, the following code must be included in ARToolKit setup:

```
arSetPatternDetectionMode(<#ARHandle handle#>, AR_MATRIX_CODE_DETECTION);
```

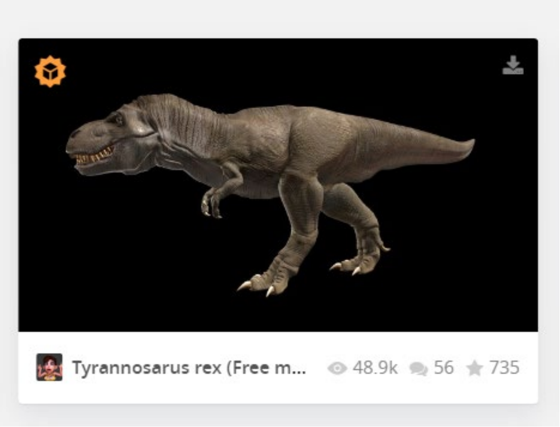
Source: https://au.gmented.com/app/marker/marker.php

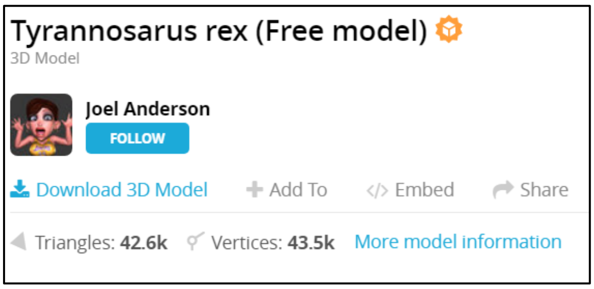
Downloading 3D models

You could either create your own 3D model or download one from Sketchfab.com for free.

- Step 1: Search for a 3D model, for example (Trex)
- Step 2: Download 3D model
- Step 3: Download the (glTF) format
- Step 4: Make sure to download 3D model that has less triangles as this can have an impact on the download speed
- Step 5: Extract to the folder that you created in the beginning. It as to be in the same directory as your index.html file for it to work

Sketchfab



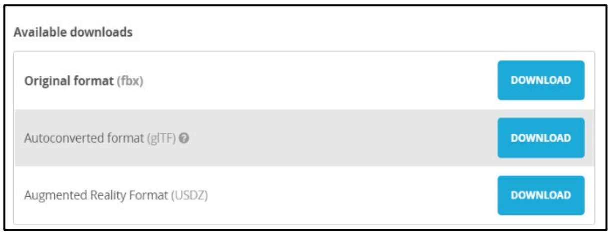


Tyrannosarus rex (Free model)

3D Model

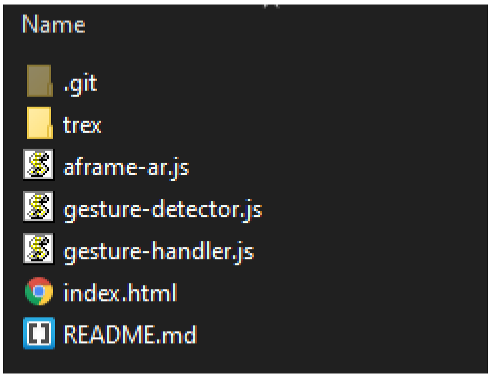
Joel Anderson

Triangles: 42.6k Vertices: 43.5k More model information



Available downloads

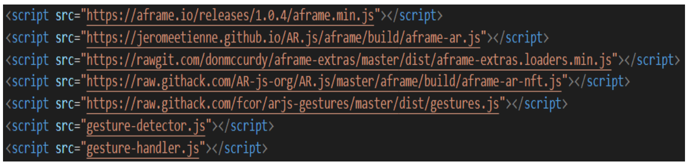
Original format (fbx)	DOWNLOAD
Autoconverted format (glTF)	DOWNLOAD
Augmented Reality Format (USDZ)	DOWNLOAD



```
Name
.git
trex
aframe-ar.js
gesture-detector.js
gesture-handler.js
index.html
README.md
```

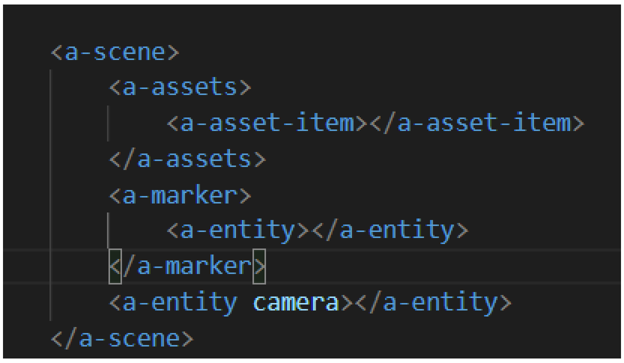
Setting up the scene

Put all necessary scripts for augmented reality to work correctly.



```html
<script src="https://aframe.io/releases/1.0.4/aframe.min.js"></script>
<script src="https://jeromeetienne.github.io/AR.js/aframe/build/aframe-ar.js"></script>
<script src="https://rawgit.com/donmccurdy/aframe-extras/master/dist/aframe-extras.loaders.min.js"></script>
<script src="https://raw.githack.com/AR-js-org/AR.js/master/aframe/build/aframe-ar-nft.js"></script>
<script src="https://raw.githack.com/fcor/arjs-gestures/master/dist/gestures.js"></script>
<script src="gesture-detector.js"></script>
<script src="gesture-handler.js"></script>
```

- Step 1: Create a scene
- Step 2: include assets inside the scene
- Step 3: include asset item inside the scene. This is where you call the scene.gltf
- Step 4: include a marker inside the scene. This where you chose the type of marker you want to use
- Step 5: include entity to scale and position 3D content as well as adding gestures
- Step 6: include entity camera inside the scene to support multiple markers

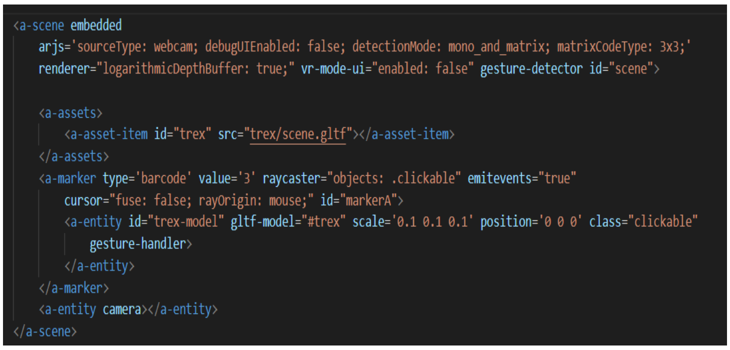


```html
<a-scene>
    <a-assets>
        <a-asset-item></a-asset-item>
    </a-assets>
    <a-marker>
        <a-entity></a-entity>
    </a-marker>
    <a-entity camera></a-entity>
</a-scene>
```

Implementing 3D model

For this part we need to go back to the scene we created in the index.html file

- Step 1: Name asset item id to (trex)
- Step 2: Add the scene.glf to the asset item and direct it to the trex folder as this is where the scene is located
- Step 3: Change marker type to (barcode) and add the code value chosen in the marker generator
- Step 4: Name entity id to (trex-model) and name the gltf-model to (#trex)
- Step 5: Add a scale of (0.1 0.1 0.1) and a position of (0 0 0) to center 3D model on the QR code
- Step 6: Add gesture handler



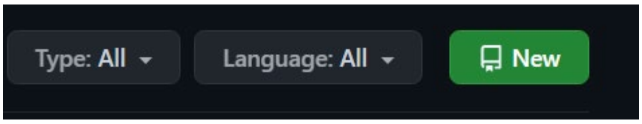
```html
<a-scene embedded
    arjs='sourceType: webcam; debugUIEnabled: false; detectionMode: mono_and_matrix; matrixCodeType: 3x3;'
    renderer="logarithmicDepthBuffer: true;" vr-mode-ui="enabled: false" gesture-detector id="scene">

    <a-assets>
        <a-asset-item id="trex" src="trex/scene.gltf"></a-asset-item>
    </a-assets>
    <a-marker type='barcode' value='3' raycaster="objects: .clickable" emitevents="true"
        cursor="fuse: false; rayOrigin: mouse;" id="markerA">
        <a-entity id="trex-model" gltf-model="#trex" scale='0.1 0.1 0.1' position='0 0 0' class="clickable"
            gesture-handler>
        </a-entity>
    </a-marker>
    <a-entity camera></a-entity>
</a-scene>
```

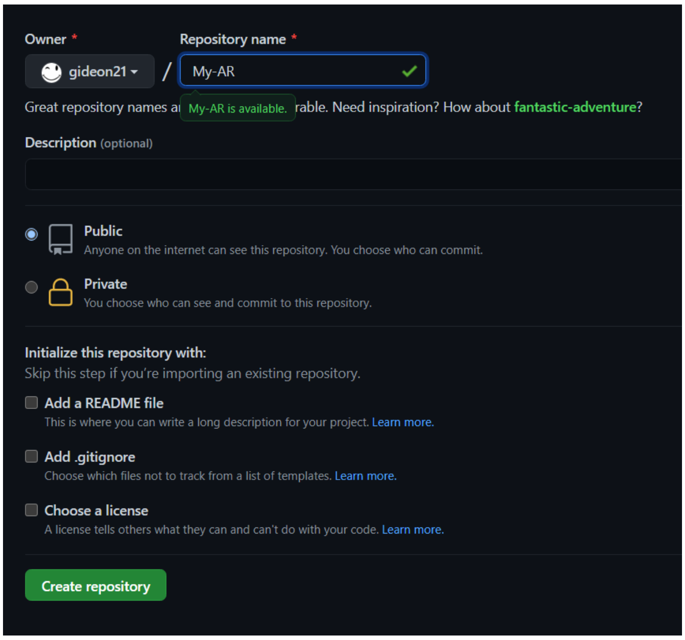
Source code:
https://github.com/gideon21/gideon21.github.io/blob/master/index.html

GitHub repository

We need to create a repository to store and run your AR content. Open the Command Prompt as administrator and follow the steps below.

- Step 1: git init
- Step 2: git add . to add all files
- Step 3: git commit -m “new commit”
- Step 4: git branch -m main
- Step 5: git remote add origin and insert HTTPS link
- Step 6: git push and insert HTTPS link
- Step 7: Refresh the GitHub page and the files will appear




Owner *

gideon21

Repository name *

My-AR ✓

Great repository names ar My-AR is available. rable. Need inspiration? How about fantastic-adventure?

Description (optional)

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

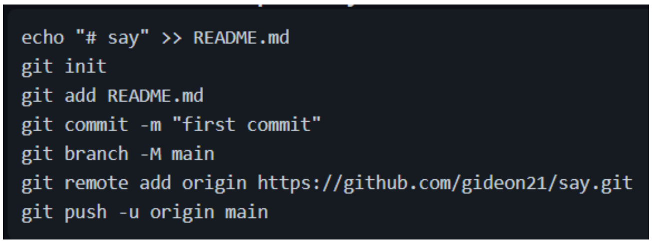
Initialize this repository with:
Skip this step if you're importing an existing repository.

□ Add a README file
This is where you can write a long description for your project. Learn more.

□ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

□ Choose a license
A license tells others what they can and can't do with your code. Learn more.

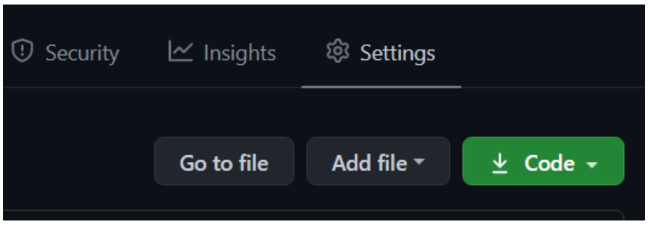
Create repository

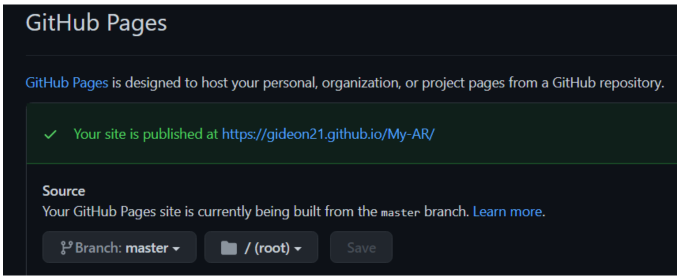

```
echo "# say" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/gideon21/say.git
git push -u origin main
```

GitHub pages

The content needs to be hosted on browser before you can see. To do this we need to create a page using GitHub pages which is designed to host project pages from GitHub repository.

- Step 1: Go to settings and scroll down till you find GitHub pages
- Step 2: Change the branch to master if it wasn’t already
- Step 3: A link will be giving to you to access your site




GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at https://gideon21.github.io/My-AR/

Source
Your GitHub Pages site is currently being built from the master branch. Learn more.

Branch: master / (root) Save

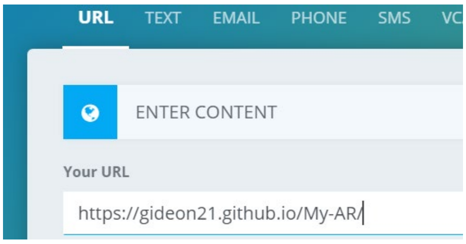
Overlay 3D content on QR code

This is the final step in getting the 3D model to appear when you scan the QR Code with a device and before we can do this, we need to generate a QR Code from QR Code monkey

- Step 1: In the enter content section put your git pages URL
- Step 2: In the set color section leave the single color as black and background as white
- Step 3: In the add logo image section upload the generated marker
- Step 4: increase the quality of the QR Code to high so that the QR Code can be tracked better
- Step 5: Click create QR Code
- Step 6: Download QR Code and use a code scanner to scan QR Code

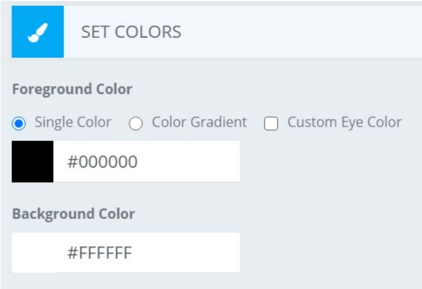
QRCODEMONKEY
THE 100% FREE QR CODE GENERATOR

Source: https://www.qrcode-monkey.com/



Your URL

https://gideon21.github.io/My-AR/



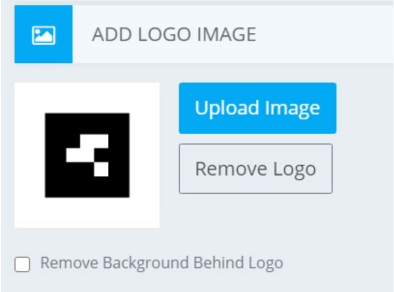
Foreground Color

● Single Color ○ Color Gradient □ Custom Eye Color

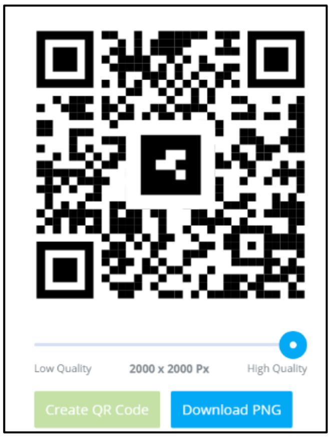
#000000

Background Color

#FFFFFF



□ Remove Background Behind Logo



2000 x 2000 Px

GitHub source code: https://github.com/gideon21/My-AR