

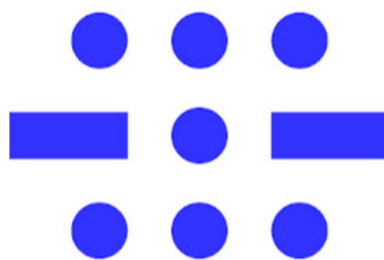
# Mesh QR Code Generator User Guide

Version 1.3

An asset for Unity

2019 © by  
Stefan Scholl

<https://www.stefan-scholl.net/>



Stefan Scholl Software Development

## Content

Intro.....	3
Usage.....	5
API.....	8
Properties.....	8
Methods.....	10
Tips.....	11
Dramatic appearance.....	11
3D printing and modeling.....	11
Using visual scripting tool “Bolt”.....	11
Changelog.....	13
Version 1.3.....	13
Version 1.2.....	13
Version 1.1.....	13

## Intro

Welcome to the wonderful world of generating QR codes in 3D.



There are many QR code generators out there. The Mesh QR Code Generator is different. Instead of getting a bitmap you get a 3D object (respectively multiple objects) to work with.

This gives you more creative opportunities than a flat and boring code. You can leave out parts of the code to fit your ideas because of the error correction feature of QR codes. Ideal for mazes.

Possible applications of 3D QR codes:

- Imagine a camera flying through a maze and then taking off to the sky, turning back to the maze it just left and you see a QR code. Take out your smartphone and scan the code.
- Hidden Easter egg hanging on a wall but only recognizable when you look at it at a certain angle.
- Ad with animated QR code at the end flying at you for dramatic effect. Standing still long enough for you to scan it with your phone.
- Your game gives you a secret code for a raffle. The QR code is animated so that people with cheap and slow phones don't have a chance to win the prize. ] :->

- Showing off the URL to your website. “Can you give me the link to your site?” – “Sure. Here’s a QR code on my phone for you to scan.” – “Why is it moving?” – “Because it’s linked to the gyroscope.”
- You enter some text (can be a URL) and a QR code maze gets generated. Surviving all the dangers in the maze you win all the fights and reach the exit. Now the camera shows the maze from above with the trail you left. You can save a screenshot and now have a unique QR code you had to fight for.
- A variation of the “build the pattern we show you here” games. Two main differences: 1. it’s lot more complicated and 2. you don’t need to be 100% perfect and complete because of the error correction. Show the pattern with a too difficult contrast for QR code scanners so it is only possible to scan it after you move blocks with a darker color at the right places.
- Just a display of a QR code maze in a shop window with animated monsters running around in it. Or something specific to the shop like cars and the QR code is the URL to the company’s website.

The possibilities are endless and more fun than bitmaps.

Let the component generate a QR code automatically or controlled by script.

Feel free to examine the six example scenes:

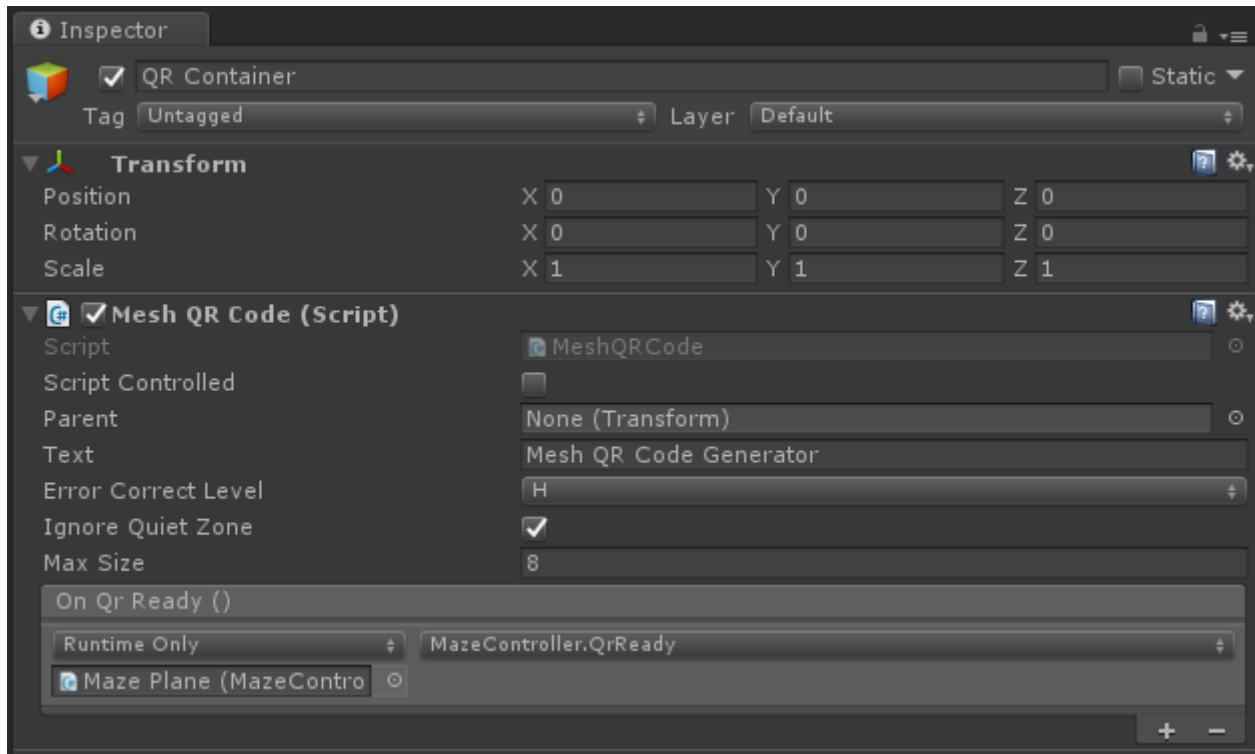
- 01-Simple-Example
- 02-What-is-this
- 03-Come-at-me
- 04-Skyline
- 05-Timepiece
- 06-A-Maze-thing

For more information about QR codes refer to the very informative Wikipedia page:

[https://en.wikipedia.org/wiki/QR\\_code](https://en.wikipedia.org/wiki/QR_code)

## Usage

Add the component “Mesh QR Code” to an existing GameObject. This can be an empty GameObject or any other one. This GameObject is the default parent for all generated GameObjects.



## Script Controlled

Check if you want to initiate the code generation by script. Without it the component uses all given parameters to generate a 3D QR code at start.

## Parent

The transform that is used as a container (folder) for all generated GameObjects by the component. Per default (= None) it is the current transform. The transform's position is the center of the QR code that gets generated on the X-Z plane. The scaling of the transform determines the size of the generated code. Rotation of the transform determines the code's plane in the world.

## Text

This is the data for the QR code that is used if the component isn't script controlled.

### Error Correction Level

QR codes can have 4 levels of error correction:

<b>L</b> (Low)	7% of codewords can be restored
<b>M</b> (Medium)	15% of codewords can be restored
<b>Q</b> (Quartile)	25% of codewords can be restored
<b>H</b> (High)	30% of codewords can be restored

This means you can remove parts of the QR code (or give them a lighter color) without destroying the data. It can still get scanned. Ideal for mazes because a QR code doesn't always have a viable path from one side to the other.

### Ignore Quiet Zone

The quiet zone is the area around the QR code. It has a width of 4 modules. A module is a bit of the QR code that can be on or off, light or dark. Mesh QR Code Generator ignores all light modules and only creates GameObjects for the dark modules. But it factors in the quiet zone when computing the scale factor to fit the given size. Unless you check "Ignore Quiet Zone".

### Max Size

The maximum size (Unity units, float) available for the generated QR code. Scales down if one module every unit doesn't fit the maximum size. Takes the **Ignore Quiet Zone** flag into account.

### Height Scale

The generated QR code can have a size between 21x21 and 177x177, excluding the quiet zone. Scaled down to Max Size the modules don't have a fixed height unless you set Height Scale to something other than 0. Any other value scales the height of the generated QR code.

### On Qr Ready

Callback that gets called after the QR code got generated.

**BEWARE:** When the component isn't script controlled its `Start()` method gets called early and possibly before the `Start()` method in your code. Don't rely on variables being initialized in your `Start()` method when using **On Qr Ready**! Alternative: poll the **IsReady** property.



## API

The component is in the namespace **MeshQRCodeGenerator**.

You need

```
using MeshQRCodeGenerator;
```

in your source code to access it.

## Properties

```
public bool ScriptControlled
```

Default is **false** and causes this component to generate the QR code at Start().

See [Script Controlled](#).

```
public Transform Parent
```

Parent folder for generated GameObjects. Defaults to current transform. See [Parent](#).

```
public string Text
```

The text used as data for the automatically generated QR code (**ScriptControlled** is false). See [Text](#).

```
public QRCodeGenerator.ECCLevel ErrorCorrectLevel
```

The error correction level used. See [Error Correction Level](#).

```
public bool IgnoreQuietZone
```

Don't take the quiet zone into account when scaling. See [Ignore Quiet Zone](#).

```
public float MaxSize
```

Maximum size (in Unity units) the QR code will have. Takes **IgnoreQuietZone** into account. Defaults to 10.0. See [Max Size](#).



public float **HeightScale**

Fixed scale of the generated QR codes height (y). Gets ignored when 0f. Defaults to 0f. See [Height Scale](#).

public UnityEvent **OnQrReady**

Callback that gets called when the QR code is generated. See [On Qr Ready](#).

public List<GameObject> **Modules**

List of all the used (active, dark) modules of the current QR code. You can manipulate them individually. The modules have a `localScale` of 1.0f for x, y, and z.

**WARNING:** Don't destroy them! Set a module inactive with `SetActive(false)` instead.

**Be aware** that modules get reused when you generate a new QR code.

public float **ScaleFactor**

The factor that was used to scale the QR code down to fit the **MaxSize**.

public int **MatrixSize**

The size of the QR code in number of modules per column and row. Includes the quiet zone. Always an odd number.

public int **MatrixSizeNoQuietZone**

The size of the QR code in number of modules per column and row. Without the quiet zone. Always an odd number.

public bool **IsReady**

Indicates (with true) that the generated QR code is completed.

## Methods

`public void Clear()`

Clear all currently displayed modules. They remain in the object pool. The QR code vanishes.

`public void ClearAll()`

Destroy all modules and clear the object pool. QR code vanishes and all generated objects with it.

`public void Generate(string data)`

Generates a new QR code. If this component has already generated one, all the used modules get set inactive and reused for the new QR code.

You don't set the **Text** property if you want to generate the QR code by script.

## Tips

### Dramatic appearance

Set the Parent property to an inactive object. When the QR code is generated (but invisible) loop over Modules and make every module inactive. Now set the container (Parent) to active and randomly activate modules in Modules. After a few seconds activate all modules.

### 3D printing and modeling

With ProBuilder installed you can export generated QR codes in many different formats.

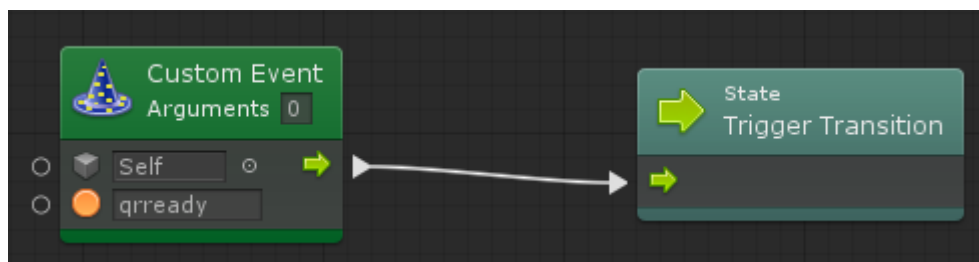
Make a scene with a base cube (X:10, Y:1, Z:10, pos 1,0,-1) as a bottom plate and position your container for the QR code in the middle (on surface, pos 5,1,-5) of it. Hit play.

Now go back to the Scene view, select the QR code container in the Hierarchy window, click "ProBuilderize" in the ProBuilder window, then shift-select the base, click "Merge Objects" (ProBuilder). Now you have one object.

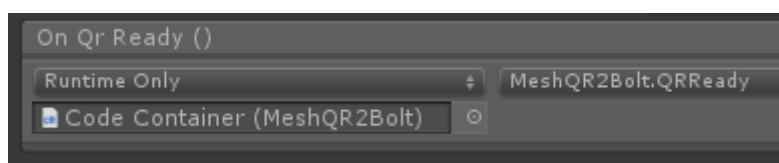
"Export" in the ProBuilder lets you export the whole thing in many different formats usable for 3D printing or modeling software.

### Using visual scripting tool "Bolt"

Mesh QR Code Generator can't call your Bolt graph directly. But it's possible to let a script trigger an event in your Bolt graph:



This example graph triggers a state transition when it receives the custom event "qready". Set the callback On QR Ready in Mesh QR Code Generator:



This small script does the work. It provides a callback and sends a custom event to your Bolt graph:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Bolt;

public class MeshQR2Bolt : MonoBehaviour
{
    public void QRReady()
    {
        CustomEvent.Trigger(gameObject, "qrready");
    }
}
```

## Changelog

### Version 1.3

- Small fixes in documentation.
- Tip about using callbacks together with visual scripting tool Bolt.
- Rotation, scaling, etc. from **Parent** won't get ignored anymore.
- New property **IsReady** to indicate that the generated QR code is complete.

### Version 1.2

- Documentation comments for API code.
- New property HeightScale to have a fixed height for the generated QR code regardless of the QR code's size.

### Version 1.1

- New methods Clear, to remove the currently displayed modules (QR code vanishes), and ClearAll, to remove all modules, including the ones in the object pool.
- Tips section in documentation.