

# 前端合成二维码

使用插件完成，插件为 DrawQRCode.unitypackage

合成二维码直接使用 [DrawQRCode](#) 类里提供的方法即可，该类的方法有两种

## 1. 生成二维码的方法

[DrawCode\\_Color32](#): 根据二维码包含的信息以及宽高，对文本信息进行转码, 返回 [Color32](#) 数组，该方法需要三个参数：

"textForEncoding": 需要转换成二维码的字符串

"width": 二维码的宽，默认宽度 256

"height": 二维码的高，默认高度 256

该方法会返回一个 [Color32](#) 数组 供后续操作

[DrawCode\\_Texture2D](#): 根据二维码包含的信息以及宽高，对文本信息进行转码, 返回 [Texture2D](#) 对象，该方法需要三个参数：

"Lastresult": 需要转换成二维码的字符串

"width": 二维码的宽，默认宽度 256

"height": 二维码的高，默认高度 256

该方法会返回一个 [Texture2D](#) 对象 供后续操作

[MergedQrCade](#): 将字符串转换成二维码，并将二维码合并到一张底图上，返回合并好的 [Texture2D](#)，该方法需要六个参数：

"Lastresult": 需要转换成二维码的字符串

"bgPath": 背景图的路径

"width": 二维码的宽

"height": 二维码的高

"x": 二维码在底图上 x 的位置 (起始位置为底图的左下角)

"y": 二维码在底图上 y 的位置 (起始位置为底图的左下角)

该方法会返回一个 [Texture2D](#) 对象 供后续操作

## 2. 保存图片

[SaveImage](#): 保存图片到本地，该方法有三个重载

< 1 >

"image": 图片文件 [Texture2D](#)

"path": 保存的路径

"imageName": 图片名字

"type": 图片保存的类型，默认为 .jpg ( 0: jpg, 1: png)

< 2 >

"image": 图片文件 Color32 数组

"path": 保存的路径

"imageName": 图片名字

"width": 图片的宽度，默认为 256

"height": 图片的高度，默认为 256

"type": 图片保存的类型，默认为 .jpg ( 0: jpg, 1: png)

< 3 >

"image": 图片文件 Byte 数组

"path": 保存的路径

"imageName": 图片名字