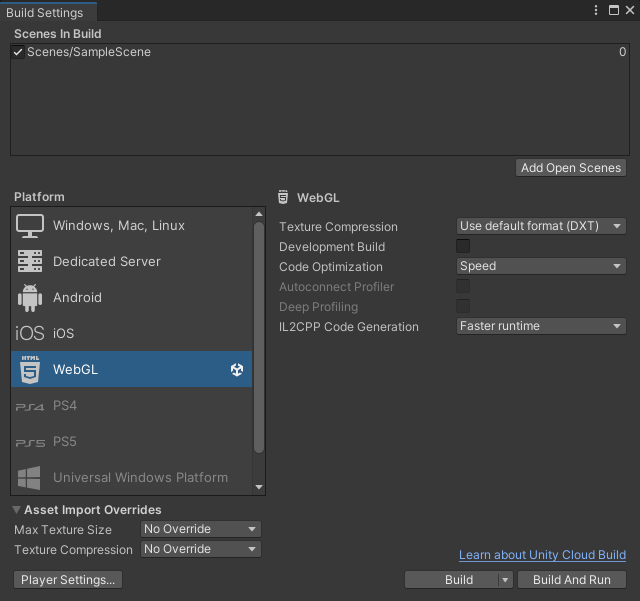
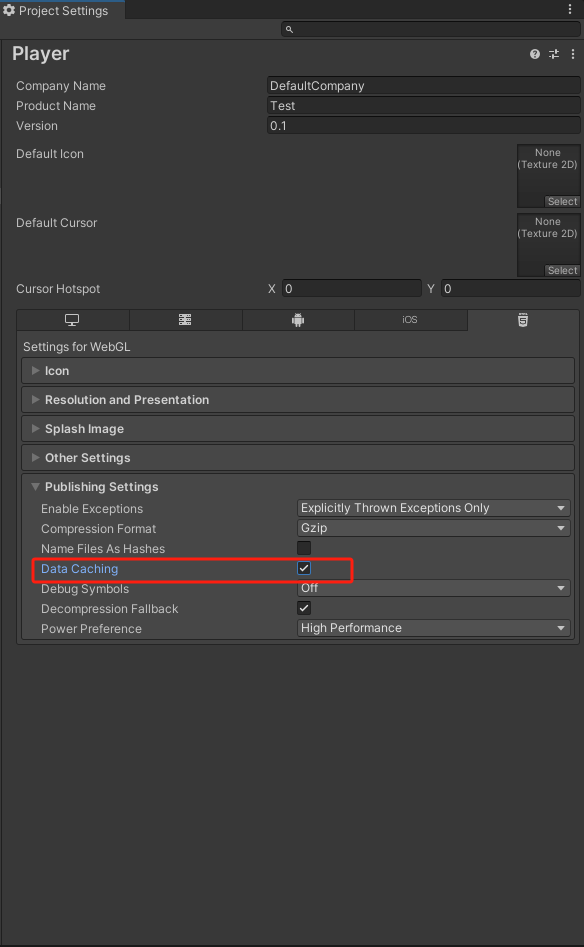
# Unity WebGL

## 基础设置

* 1. 设置导出平台为WebGL



* 1. 在Player Settings -> Publishing Settings 中勾选 Data Caching 和 Decompression Fallback 选项。下述链接是官方解释：Data Caching 简单来说就是缓存数据到本地下次打开不需要下载直接可以进入游戏 Decompression Fallback 出现无法解析gz文件、web服务器配置错误等需勾选该选项 Unity - 手册：WebGL Player 设置 (unity3d.com) <https://docs.unity3d.com/2021.3/Documentation/Manual/class-PlayerSettingsWebGL.html>

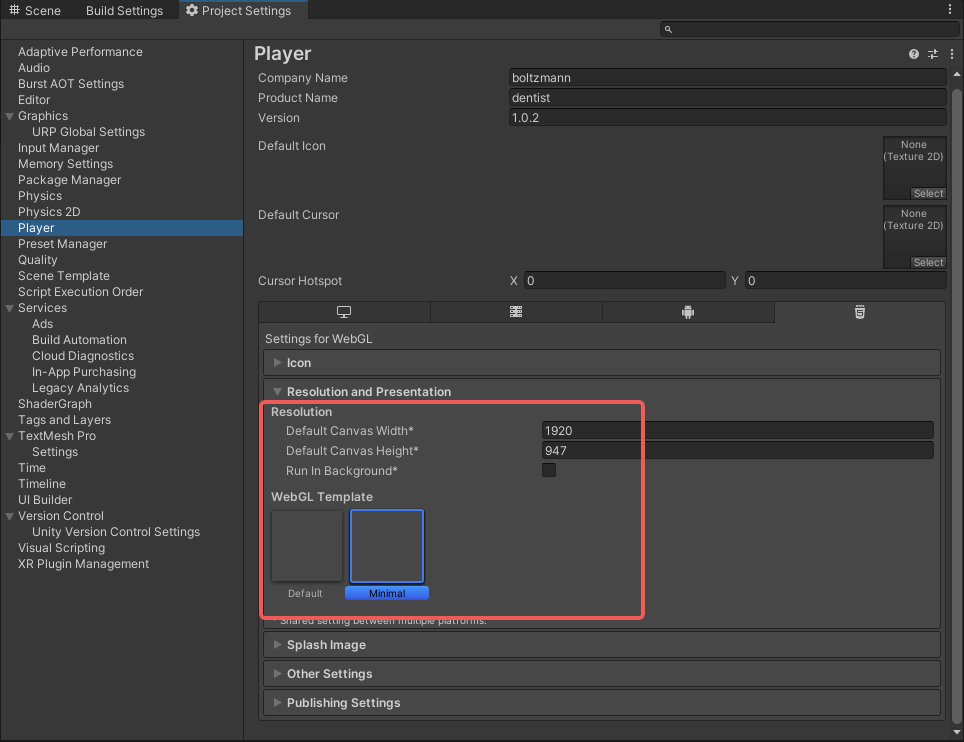


## 编写和导出

* 1. 场景搭建
  2. 编写代码
     1. 注意：方法必须是public，否则会访问不到



## UnityWebGL发布设置



## Unity WebGL如何自适应网页

* 方式一：能自适应网页，不会拉伸画面，但是如果画面不能达到要求时，会出现空白区



<body>

<script>

function resizeGame() {

var gameContainer = document.getElementById("unity-canvas");

var widthToHeight = 1920 / 947;

var newWidth = window.innerWidth;

var newHeight = window.innerHeight;

var newWidthToHeight = newWidth / newHeight;

if (newWidthToHeight > widthToHeight) {

newWidth = newHeight \* widthToHeight;

} else {

newHeight = newWidth / widthToHeight;

}

gameContainer.style.width = newWidth + 'px';

gameContainer.style.height = newHeight + 'px';

console.log( gameContainer.style.width);

}

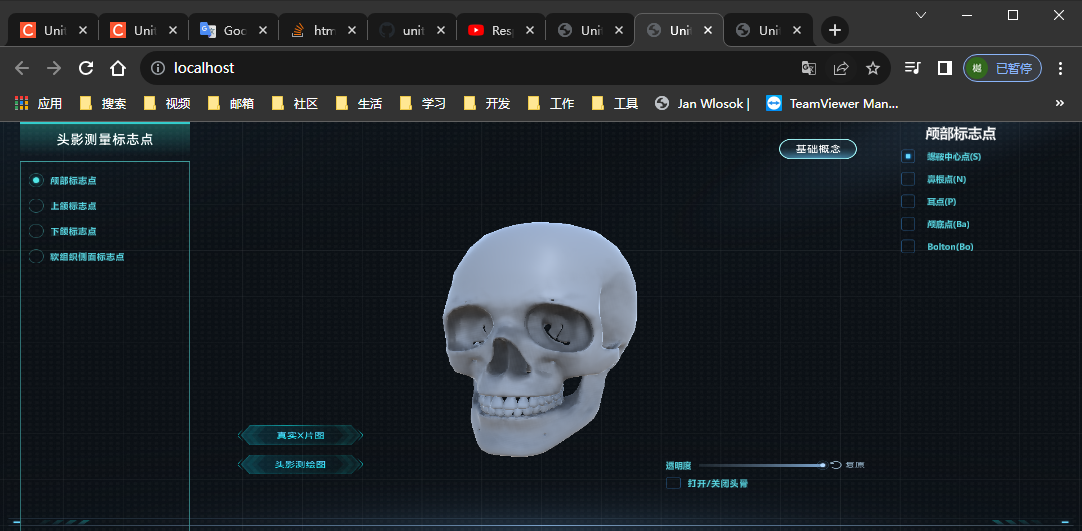
window.addEventListener('resize', resizeGame, false);

resizeGame();

 </script>

</body>

* 方式二：能自适应网页，网页全屏，但是会有拉伸



<head>

<style>

  #unity-canvas {

        position: fixed;

        top: 0;

        left: 0;

        width: 100%;

        height: 100%;

}

</style>

</head>

 <body>

  <script>

  function resizeGame() {

      var gameContainer = document.getElementById("unity-canvas");

      gameContainer.style.width = window.innerWidth + "px";

      gameContainer.style.height = window.innerHeight + "px";

    }

 window.addEventListener('resize', resizeGame, false);

      resizeGame();

 </script>

 </body>

## UnityWebGL不支持Task.Delay()

* 使用协程替换

## 示例网页

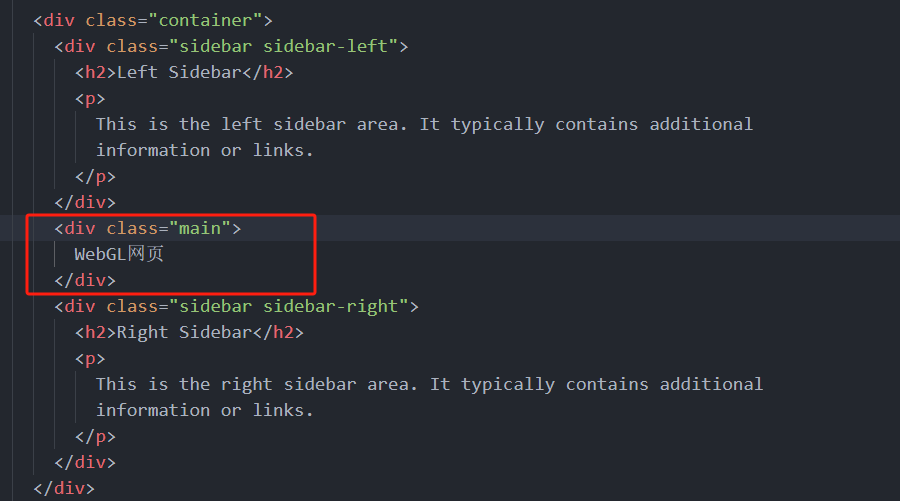


## 嵌入到webGL

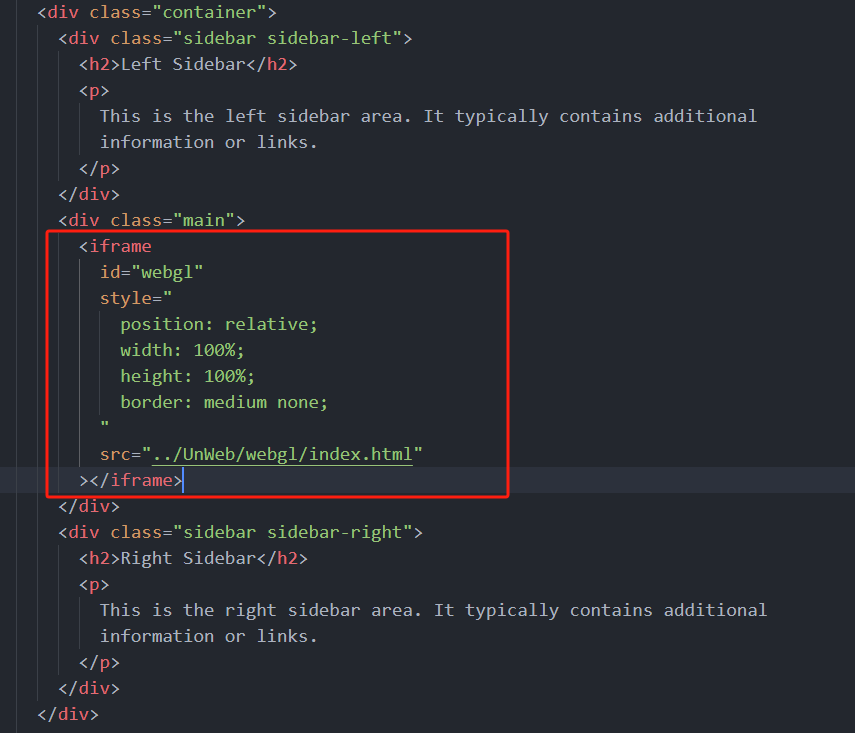
* 1. 创建文件夹将打包好的WebGL和网页放在同一文件夹下，如下图（这步的主要是为了好管理，也可以不用这样做）



* 1. 使用IDE打开网页，在代码中找到下图红框标记的位置

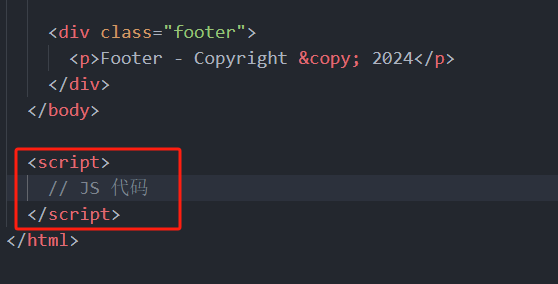


* 1. 将div标签中的内容替换为下述代码，如果你的WebGL项目已经部署在服务器上，则可将iframe标签中的src链接改为服务器地址即可。（关于iframe可参考HTML Iframe (w3school.com.cn) <https://www.w3school.com.cn/html/html_iframe.asp）保存后使用Firefox浏览器打开网页，已经部署在服务器上的项目则可以使用任意支持WebGL的浏览器打开。>



## 网页调用Unity方法

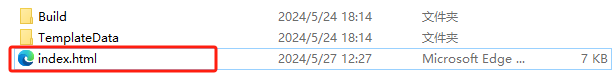
* 1. 在代码中找到下图红框标记的位置



* 1. 将script标签中的内容替换为下述代码（代码比较简单且都有注释笔者就不做过多解释）
     1. 注意：methodName 必须和Unity中的方法名一致否则会找不到方法



* 1. 完成上述步骤后，使用IDE打开WebGL网页（下图中的index.html文件）



* 1. 找到script标签，在该标签下找到初始化加载器的代码。并将红框中的代码添加进去。





* 1. 创建给Unity发送消息的代码，这里我们简单解释一下代码中用到的SendMessage 方法它允许你从 JavaScript 代码中调用 Unity 场景中游戏对象的特定方法。
     1. 注意：SendMessage 方法有三个参数：
     2. 目标对象名称：这是场景中游戏对象的名称，你想要调用的方法就定义在这个对象上。
     3. 方法名称：这是你想要调用的方法的名称，它应该是目标对象上脚本中的一个公共方法。
     4. 参数：这是你想要传递给方法的参数。在 Unity 的 SendMessage 中，这个参数只能是字符串类型。如果你需要传递更复杂的数据，你可能需要使用其它机制。
  2. 下述方法笔者已经做了简单的封装，你只需要更改目标对象名称即可。如需更详细的解释可参考Unity - 手动：从 JavaScript 调用 Unity C# 脚本函数 (unity3d.com) <https://docs.unity3d.com/2021.3/Documentation/Manual/web-interacting-browser-unity-to-js.html>

