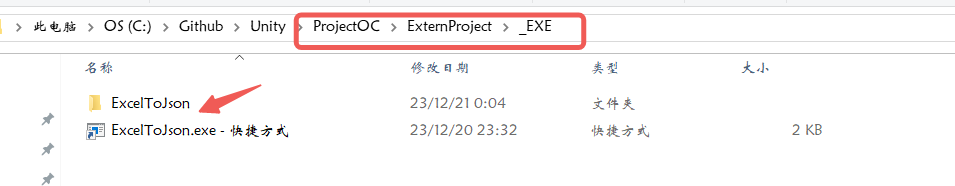
### 策划

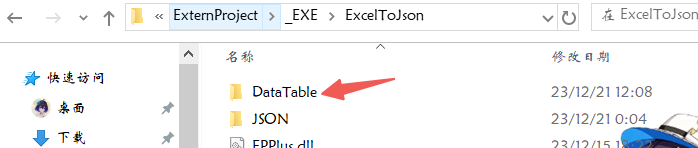
更改为DataToBinary.EXE

#### Exe可执行文件位置



#### Excel表格放置位置

建议放置于



也可以放置于这个目录下的相对路径位置，只要与对应的程序协商好，可以找到对应文件即可

#### 转换更新

新增表格或者表格结构有更新，记得先通知程序侧更新转换程序

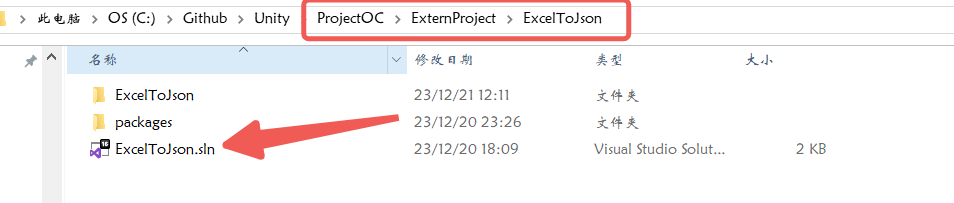
执行ExcelToJson.exe 即可

默认是生成的JSON文件路径直接放置于对应的unity JSON资产目录，然后打包一下AB包即可使用

若有问题，找到对应程序 battle (为什么路径没写对啊!!!(doge)

### 二、程序

#### 1. 程序项目路径



#### 新增表格转换

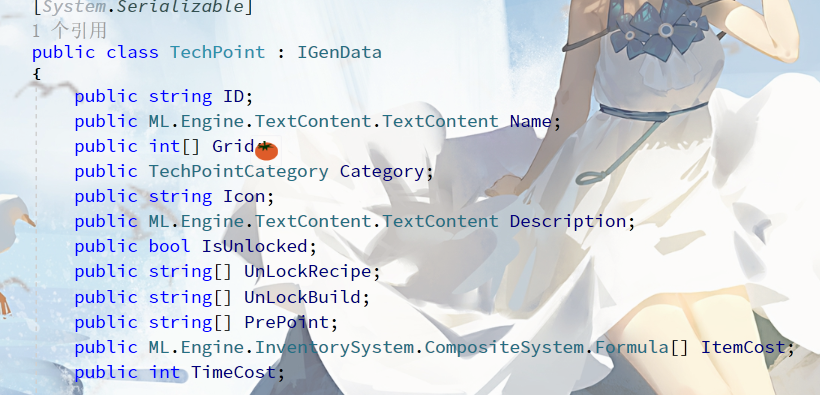
#### 注意！！！！！！不然二进制文件找不到引用

对应的转换结构体的命名空间必须和UNITY里面的结构体的命名空间一模一样，

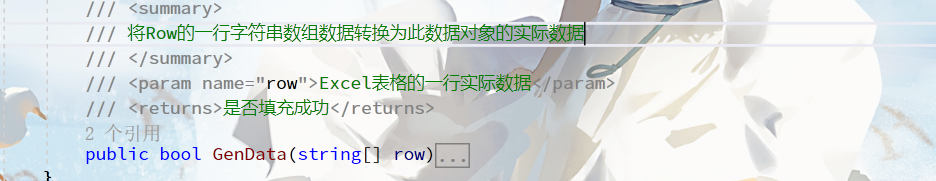
类型、名称也必须一模一样

如果是类的内部类，那么也必须用同样名称的内部类包裹起来，详情可见TPPanel结构体

##### ① 加入对应的数据转换结构体



##### ② 实现接口 “IGenData”



        /// <summary>

        /// 将Row的一行字符串数组数据转换为此数据对象的实际数据

        /// </summary>

        /// <param name="row">Excel表格的一行实际数据</param>

        /// <returns>是否填充成功</returns>

        public bool GenData(string[] row)

        {

            if (row[0] == null || row[0] == "")

            {

                return false;

            }

            // 0 -> ID

            this.ID = row[0];

            // 1 -> Category

            this.Category = (ProjectOC.TechTree.TechPointCategory)Enum.Parse(typeof(ProjectOC.TechTree.TechPointCategory), row[1]);

            // 2 -> Name

            this.Name = new ML.Engine.TextContent.TextContent();

            this.Name.Chinese = row[2];

            this.Name.English = row[2];

            // 3 -> Icon

            this.Icon = row[3];

            // 4 -> 描述

            this.Description = new ML.Engine.TextContent.TextContent();

            this.Description.Chinese = row[4];

            this.Description.English = row[4];

            // 5 -> Grid

            var sgrid = row[5].Split(',');

            this.Grid = new int[2] { int.Parse(sgrid[0]), int.Parse(sgrid[1]) };

            // 6 -> IsUnlocked

            this.IsUnlocked = row[6] != "0";

            // 7 -> 配方

            this.UnLockRecipe = row[7].Split(',').Where(x => !string.IsNullOrEmpty(x)).ToArray();

            // 8 -> 建筑物

            this.UnLockBuild = row[8].Split(',').Where(x => !string.IsNullOrEmpty(x)).ToArray();

            // 9 -> 前置科技点

            this.PrePoint = row[9].Split(',').Where(x => !string.IsNullOrEmpty(x)).ToArray();

            // 10 -> ItemCost

            this.ItemCost = Program.ParseFormula(row[10]).ToArray();

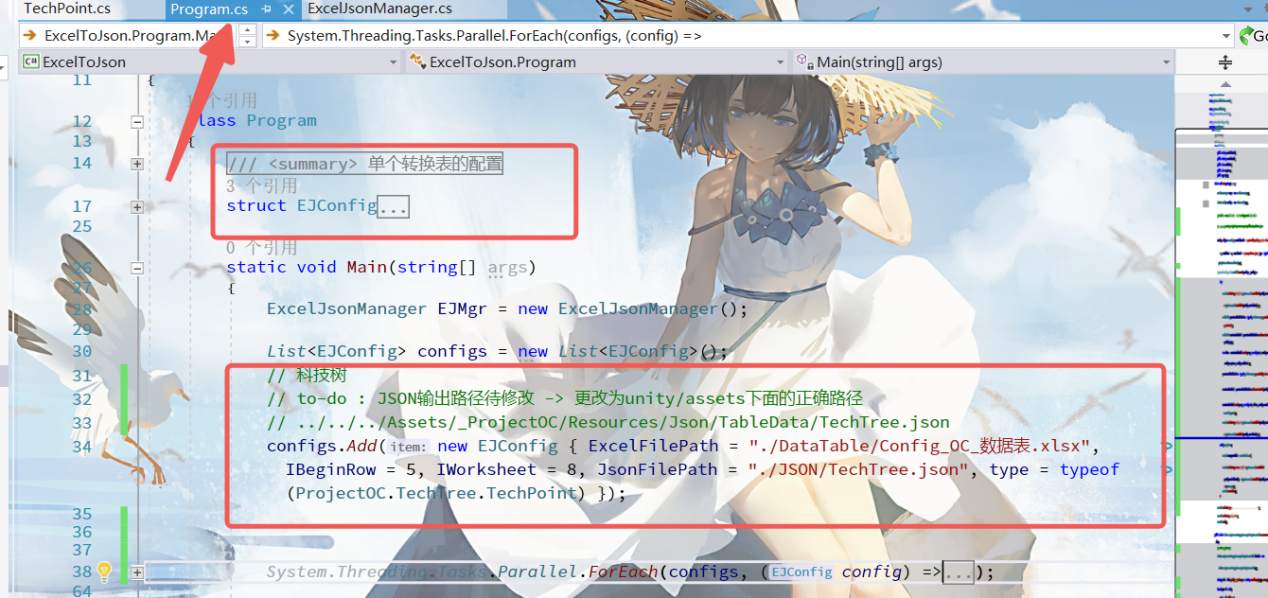
            // 11 -> TimeCost

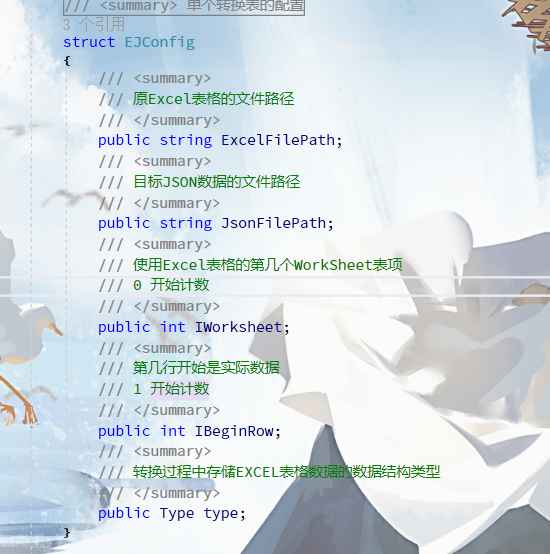
            this.TimeCost = int.Parse(row[11]);

            return true;

        }

##### ③ 加入表格配置信息





这里的JSON输出路径应是可以放置于unity项目内的实际json资产路径，直接覆盖，不需要手动更新覆盖

目前科技树这里的to-do是尚未确定好科技树的EXCEL表格配置，为避免数据不合适，暂时没有使用实际的路径，使用临时路径测试功能无误

上面是EXCEL转二进制，图片是之前的转JSON，但是操作流程不变

JSON转二进制一样，位于下面的JSON块代码