

预设组件导出工具

一、开启工具

进入预设模式，勾选根节点最右方选项即开启工具

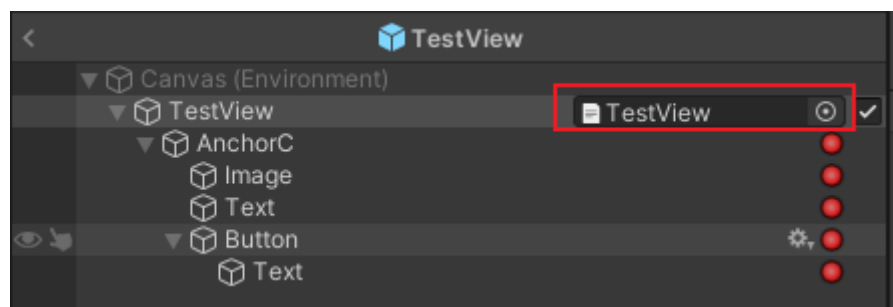
如下图



二、界面及操作介绍

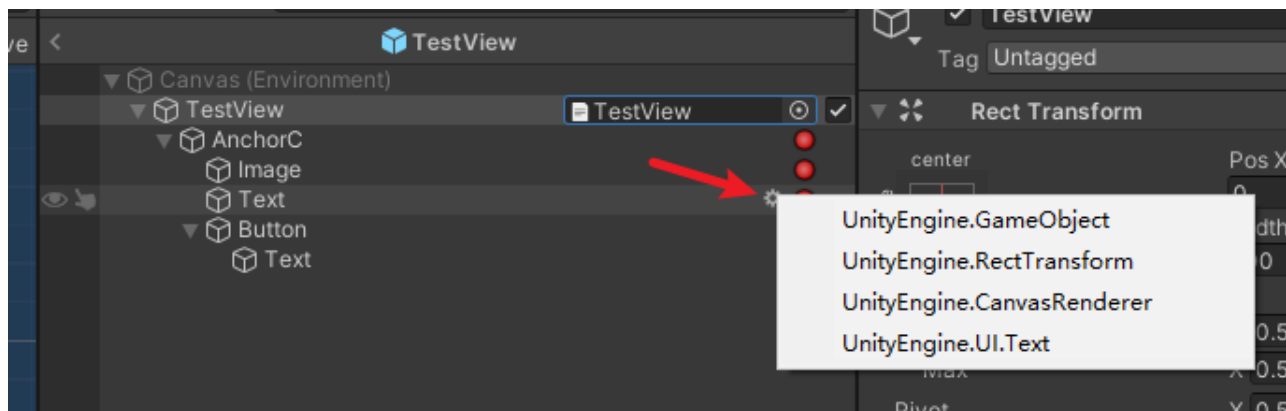
1.绑定lua文件

如下图所示，红色框内为预设对应的Lua文件，可直接将对应的Lua文件拖拽至框内

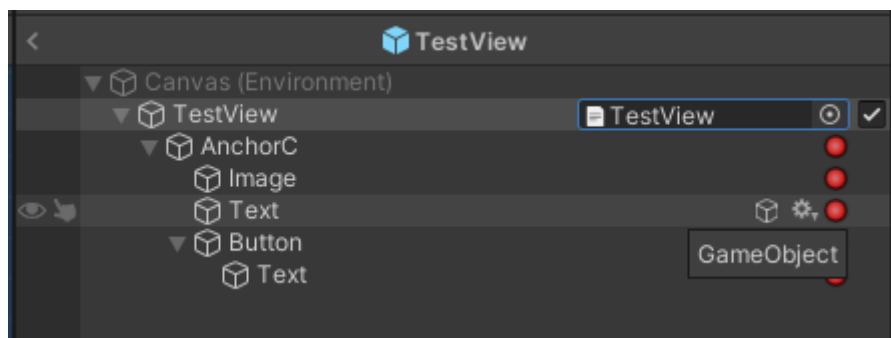


2.设置组件为导出

鼠标当前所在对象行上会出现箭头所指的齿轮按钮，点击后会出现下拉框，点击下拉框中组件，即会设置该组件为导出项，

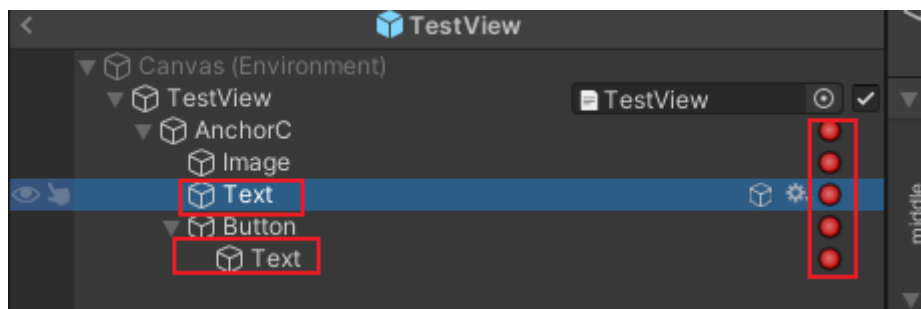


下图所示为选择GameObject，在齿轮前方会生成对应组件的图标，鼠标放置上方时会出现该组件名Tips，点击该图标会复制组件引用名

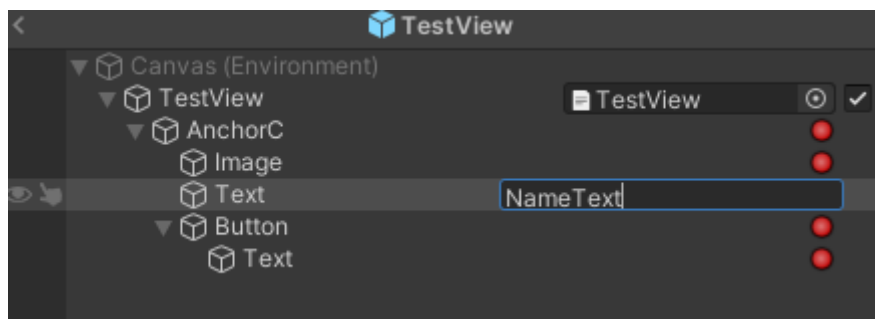


3.别名

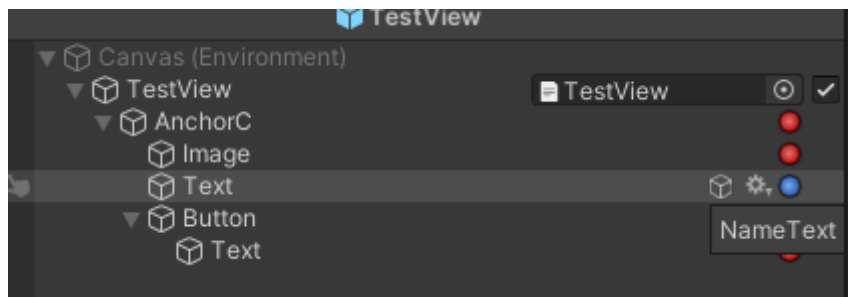
下图中出现了两个"Text"，直接导出会出现重复命名，需要设置别名



点击右侧的红色圆点，会弹出别名设置框，输入别名后按回车或者鼠标点击其他处即设置成功



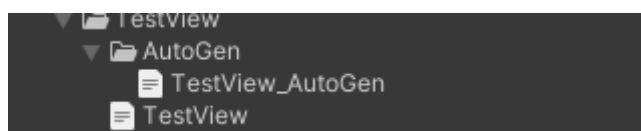
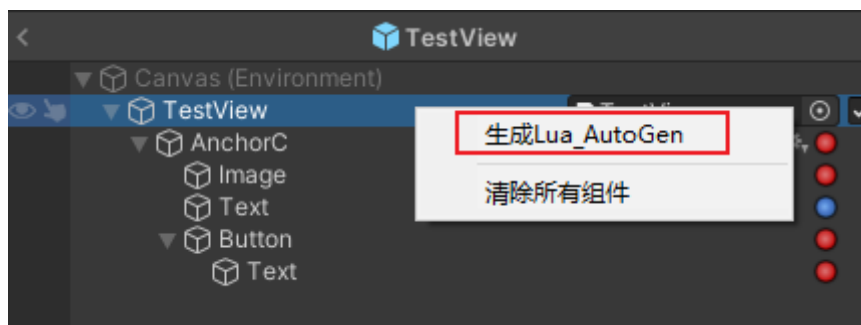
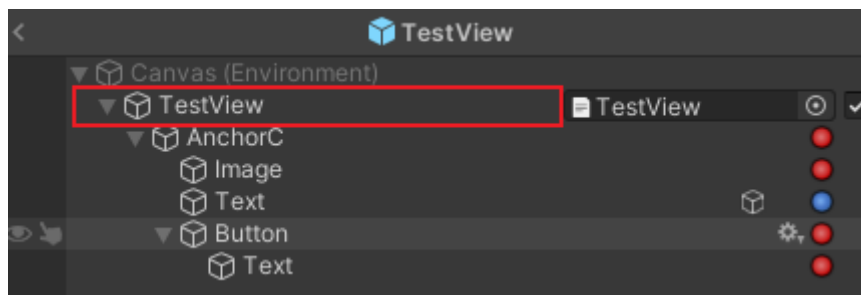
有别名的对象右侧的红色圆点会变成蓝色圆点，鼠标放至上方会出现别名Tips



设置好别名后，导出时会优先使用别名

4.导出

在根节点空白处点击鼠标右键，弹出选项，选择"生成Lua_AutoGen"，即会在绑定的Lua文件同层目录生成名为"AutoGen"的子目录，并在该目录生成"XXX_AutoGen.txt"lua文件



三、在lua中导入AutoGen

直接复制AutoGen中这段代码

```

1  -----
2  -- 此文件为自动生成 请勿手动修改 --
3  -----
4
5  --[[拷贝以下代码在初始化方法内部调用
6  require("script/MainView/MiniGame/AutoGen/MiniGameView_AutoGen")
7  self:InitComponents()
8  ]]
9  local MiniGameView = require("script/MainView/MiniGame/MiniGameView");
10
11 function MiniGameView:InitComponents()
12     self.widgetTable = {
13         ["Touch"] = "TouchNode",
14         ["MiniGameEnter1"] = "AnchorC/GameNode/MiniGameEnter1",
15         ["MiniGameEnter2"] = "AnchorC/GameNode/MiniGameEnter2",
16         ["MiniGameEnter3"] = "AnchorC/GameNode/MiniGameEnter3",
17         ["CloseButton"] = "AnchorC/CloseButton",
18         ["TitleNode"] = "AnchorC/TitleNode",
19         ["ItemNode"] = "AnchorC/ItemNode",
20         ["HelpButton"] = "AnchorC/HelpButton",
21     };
22
23     self.componentTable = {
24         ["CountIconImage"] = {"AnchorC/ItemNode/CountIcon", "Image"},
25         ["CountText"] = {"AnchorC/ItemNode/CountText", "Text"},
26     };
27
28     ParseUIWidget(self.gameObject, self.widgetTable, false);
29     ParseUIWidget(self.gameObject, self.componentTable, true);
30 end
31
32 return MiniGameView;

```

必须在self.gameObject赋值后调用这段代码

```

-- 只会初始化一次，之后会缓存起来，需要每次打开刷新的逻辑不要放到这个函数。
function MiniGameView:OnInit(name, BaseUIView)
    self.modName = name;
    self.viewManager = BaseUIView;
    self.gameObject = BaseUIView.gameObject;

    require("script/MainView/MiniGame/AutoGen/MiniGameView_AutoGen")
    self:InitComponents()

    self.game1 = self:CreateGraphic(MiniGameEntrance, self.widgetTable.MiniGameEnter1, 1) -- 扭蛋
    self.game2 = self:CreateGraphic(MiniGameEntrance, self.widgetTable.MiniGameEnter2, 2) -- 九宫格
    self.game3 = self:CreateGraphic(MiniGameEntrance, self.widgetTable.MiniGameEnter3, 3) -- 弹珠

    self:BindUI();
    self:AddBtns();
end

```

四、注意事项

- GameObject的层级变化后需要重新导出
- 导出的GameObject的命名为该节点名，导出的其他组件，则为节点名+组件名，如果节点名已经以组件名结尾，则直接使用节点名，如一个名为Count的Text，导出的GameObject名为"Count"，导出的Text名为"CountText"，以此类推，如CountAnimator、CountRectTransform等。推荐以主要组件命名结尾，如前面的Count命名为CountText，保持显示的名字和导出的名字尽量一致。
- 别名尽量少用，如不同Button中的Text，最好直接改名，别名是给嵌套预设用的，如一个预设中有多个相同预设的实例，常见情况为滚动框中的Item