**SDKTools使用说明**

|  |  |  |  |
| --- | --- | --- | --- |
| **文件属性** | | | |
| 作者 | v\_jiaxxli | 制定日期 | **2019-04-09** |
| 审核 |  | 审核日期 |  |
|  |  |  |  |
| **版权宣告** | | | |
| **内部资料，未经授权，任何人不得擅自传播、复制本文档的部分或全部内容。** | | | |

腾讯科技（深圳）有限公司

 修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 修订版本 | 修改内容 | 作者 |
| 2019/04/09 | V1.0 | Created | v\_jiaxxli |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目 录

[1、SDKTools整体介绍 4](#_Toc14368)

[1\_1、安装编译库 4](#_Toc28706)

[2、整体流程 5](#_Toc7946)

[2\_1、新建，导入项目 5](#_Toc20471)

[2\_1\_1、新建项目流程演示（尽量使用1280×720分辨率图片配置）： 5](#_Toc31981)

[2\_1\_2、导入项目流程演示： 7](#_Toc3269)

[2-2、选择文件类型 9](#_Toc22336)

[2\_2\_1、选择文件类型流程演示： 9](#_Toc3896)

[3、配置项介绍 11](#_Toc20678)

[3\_1、添加场景 11](#_Toc167)

[3\_1\_1、添加场景流程演示： 11](#_Toc16536)

[3\_1\_2、添加task流程演示： 13](#_Toc19066)

[3\_1\_3、task文件参数： 14](#_Toc26210)

[3\_1\_4、refer文件 18](#_Toc25072)

[添加参考任务流程演示： 18](#_Toc19877)

[3\_2、添加UI 22](#_Toc19159)

[3\_2\_1、添加UI识别项流程演示： 22](#_Toc16386)

[3\_2\_2、UI配置参数： 23](#_Toc20781)

[3\_3、添加动作配置 23](#_Toc26325)

[3\_3\_1、动作配置参数： 23](#_Toc14548)

[3\_3\_2、添加动作流程演示： 24](#_Toc32423)

[3\_4、添加地图路线 26](#_Toc23470)

[3\_4\_1、添加地图路线流程演示： 26](#_Toc21318)

[4、 GameReg调试 31](#_Toc12773)

[4\_1、GameReg调试流程 31](#_Toc12674)

[5、注意事项 32](#_Toc23852)

[6、附 33](#_Toc9179)

# 1、SDKTools整体介绍

SDKTools主要用于生成AI服务所需要的配置文件，按照json格式保存。

它是用python编写的，用QT作为图形界面。

**1\_1、安装编译库**

**Linux或windows：**

**cd <path of SDKTool>**

**virtualenv env //创建虚拟环境**

**source env/bin/activate**

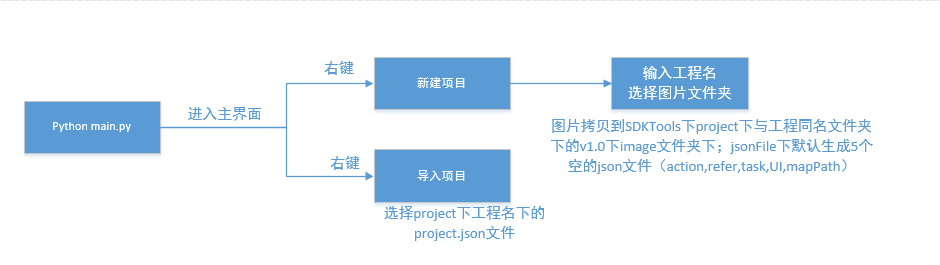
**pip install -r requirements.txt**

**python main.py**

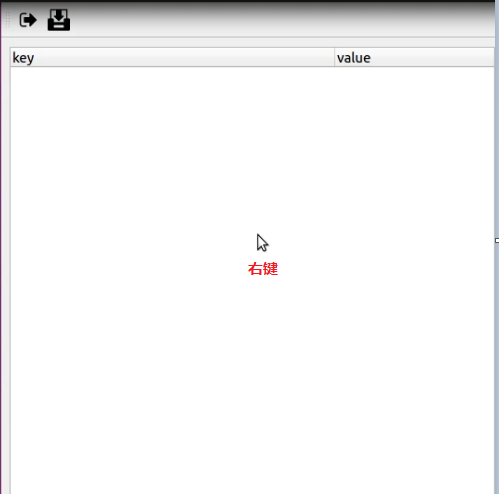
# 2、整体流程

Cfg/SDKTool.ini中的[debug][flag]配置项决定是否需要调试，如果是windows环境，一定要置为False

**2\_1、新建，导入项目**



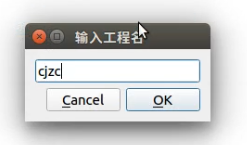
**2\_1\_1、新建项目流程演示（尽量使用1280×720分辨率图片配置）：**



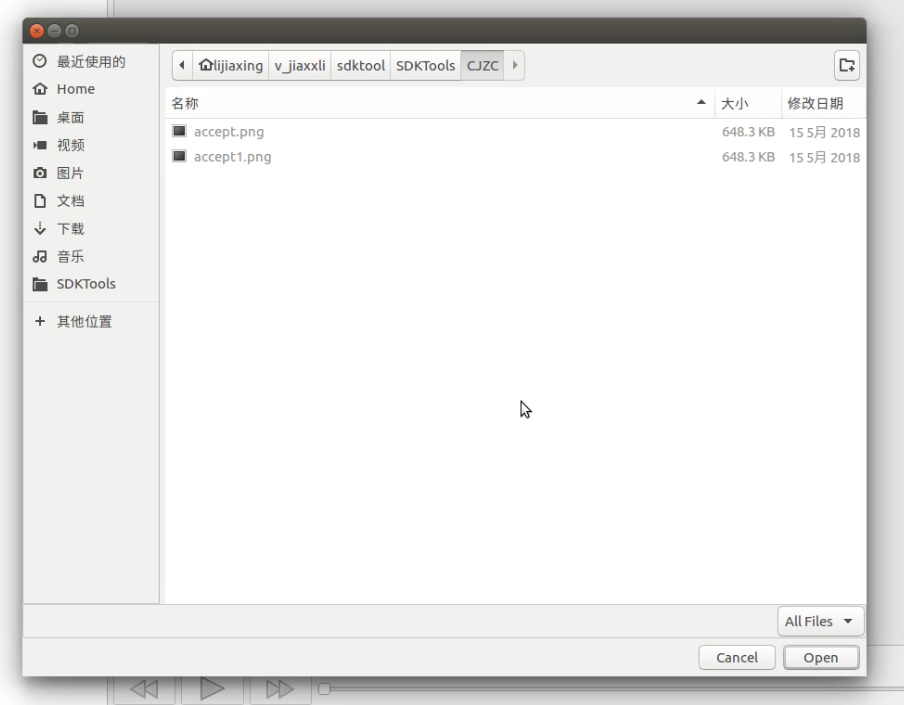
**2-1-1-1屏幕右键**



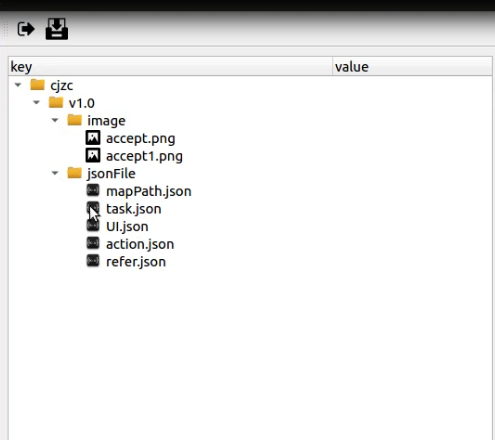
**2-1-1-2新建项目**



**2-1-1-3输入工程名**



**2-1-1-4选择装有图片的文件夹**

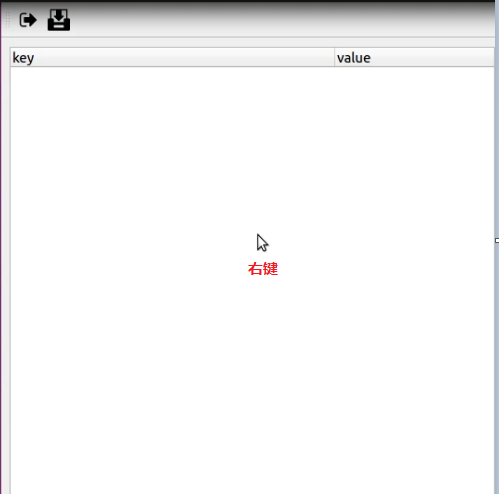


**2-1-1-5生成文件演示**

选择装有图片的文件夹时，会把所选择文件夹下所有图片或子文件另存到SDKTools/project/cjzc（工程名）/v1.0（版本号）/image/下。

生成文件保存在SDKTools/project/cjzc（工程名）/v1.0（版本号）/jsonFile/下。

**2\_1\_2、导入项目流程演示：**

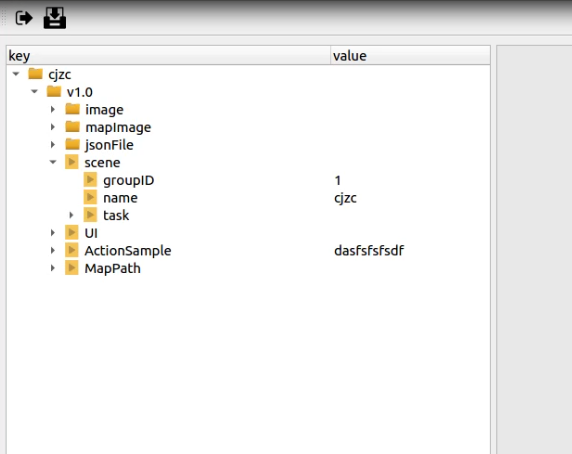
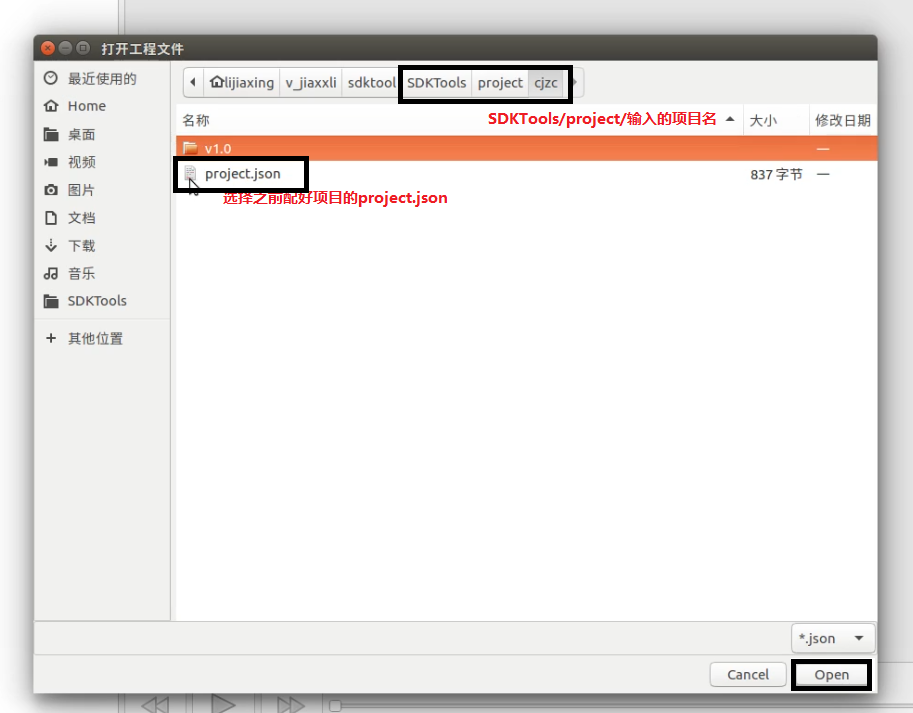


**2-1-2-1屏幕右键**



**2-1-2-2导入项目**

**2-1-2-3打开project.json**

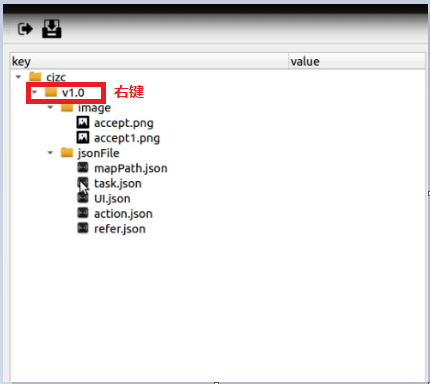


**2-1-2-4导入项目演示**

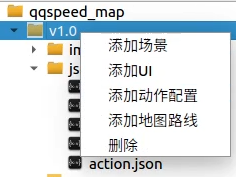
**2-2、选择文件类型**

**2\_2\_1、选择文件类型流程演示：**

在版本号上右键



**2-2-1-1版本号右键**



**2-2-1-2配置对应文件**

**添加场景**：配置GameReg的识别任务，

包括task（普通识别任务），refer（多分辨率识别任务处理）

**添加UI**：配置UI操作

**添加动作配置**：配置模仿学习动作录制区域

**添加地图路线**：配置需要跑图的游戏，跑图路线

（详见第三章）

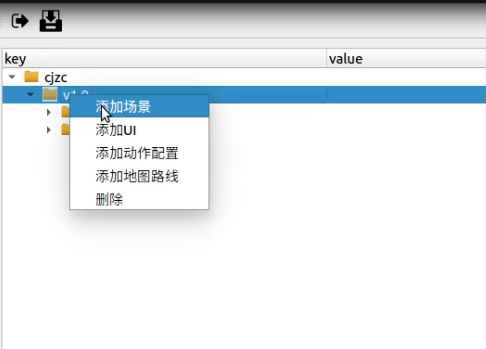
# 3、配置项介绍

**注：按顺序填写配置项；想要保存到json文件里，一定要先保存（ctrl+s；保存按钮）再关闭。**

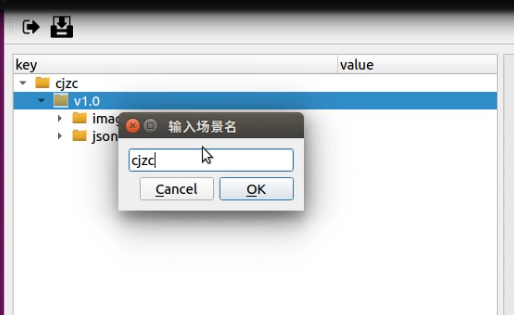
**3\_1、添加场景**

点击添加场景会让你输入一个场景名，输入好之后，就会生成一个场景名为你刚刚输入值的scene，默认会生成三项groupID：1；name：你输入的scene名；一个task。尽量使用1280×720分辨率的图片进行配置。

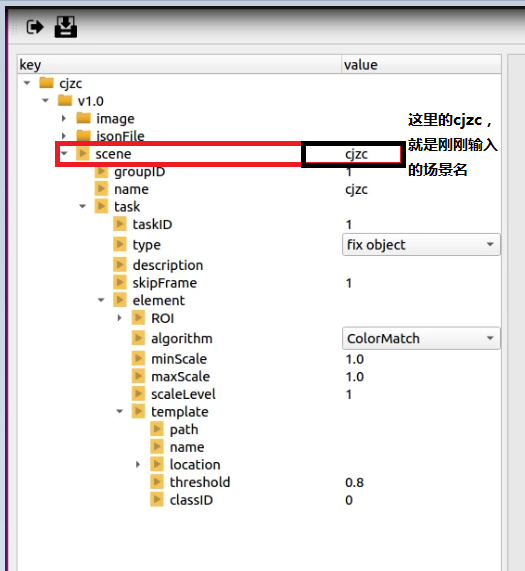
**3\_1\_1、添加场景流程演示：**



**3-1-1-1添加场景**



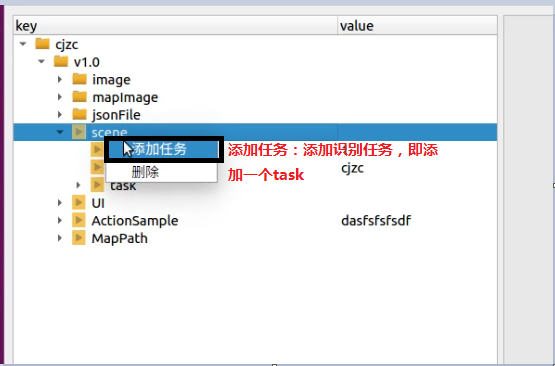
**3-1-1-2输入场景名**



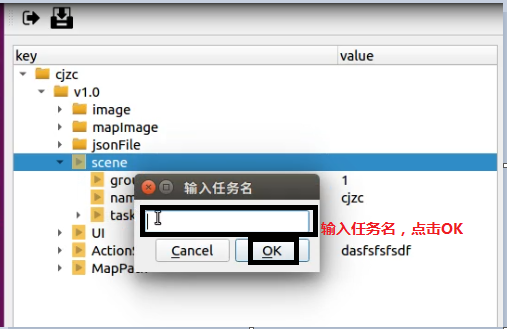
**3-1-1-3生成文件演示**

**3\_1\_2、添加task流程演示：**

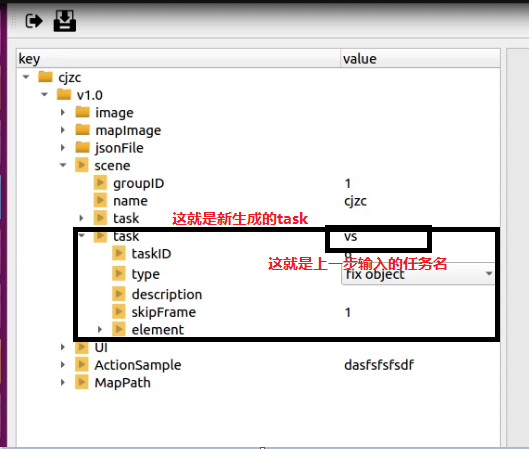
scene里面主要就是配置每个task，每个task就是一项单独的识别任务。你可以先修改默认生成的task，需要添加一个task的话，在scene上右键，点击添加任务。点击没有配置过的task，会先让你选择一张图片，然后你可以调整参数，配置过的task，会将你配好的都显示出来。



**3-1-2-1添加任务**



**3-1-2-2输入任务名**



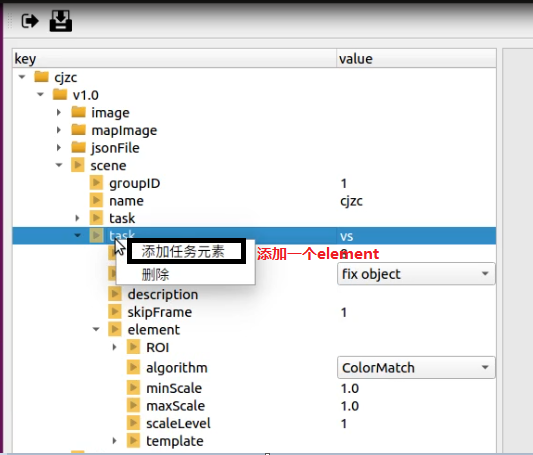
**3-1-2-3新生成task演示**

**3\_1\_3、task文件参数：**

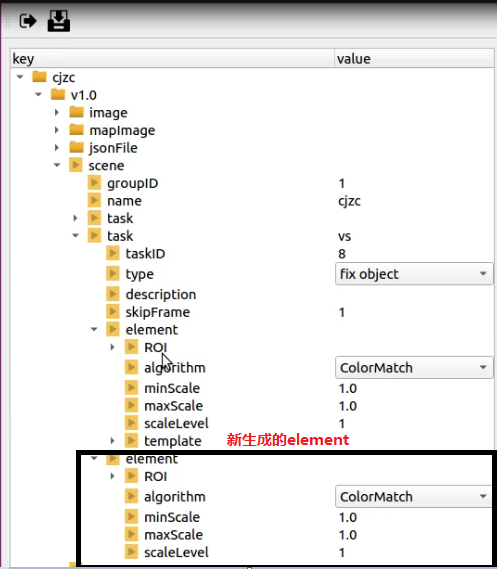
|  |  |
| --- | --- |
| **Key** | **描述** |
| taskID | 默认从1开始往上叠加；可以自己点上去修改，不可以重复 |
| type | 识别类型：  fix object:固定物体检测  pixel:基于像素的检测  stuck:卡住检测  deform object:形变物体检测  number:数字检测 |
| description | 描述，可不填，建议填 |
| skipFrame | 默认为1，可修改，你希望隔多少帧处理一次 |
| element | 识别项，根据type的选择，会生成不同的element |

**添加element流程演示：**

一个task默认会生成一个element，一个task可以有多个element，需要多个element时，可以在想要添加element的task上右键，添加任务元素。



**3-1-3-1添加任务元素**



**3-1-3-2新生成element演示**

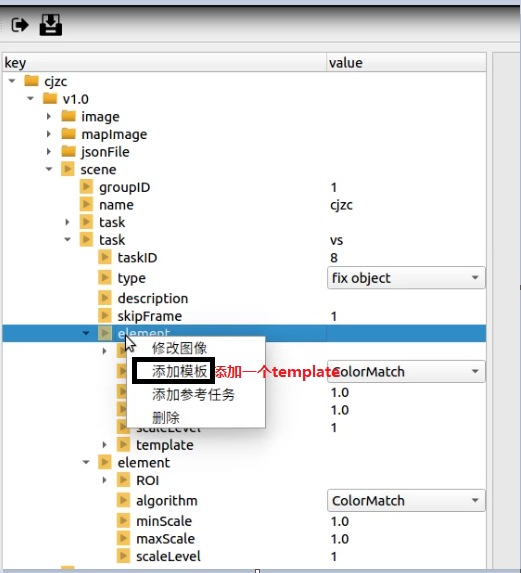
element：识别项，会有如下这些参数（根据type的选择，会自动生成对应的参数）

|  |  |
| --- | --- |
| **Key** | **描述** |
| ROI | 游戏识别区域，在右图上按住鼠标左键拖一个矩形出来，释放鼠标左键，会在ROI中记录矩形左上角的x，y值和宽w，高h（单位都是px） |
| algorithm | 算法，会跟据上面选择，默认给 |
| minScale | 最小缩放比例，因为可能要做多分辨率处理，可能需要将模板进行放缩，默认是1.0，不进行处理 |
| maxScale | 最大放大比例,默认是1.0，不进行处理 |
| scaleLevel | 将模板在minScale-maxScale中进行，分多少级别进行，默认为1 |
| template | 模板图像配置，首先先选择一张图片，然后画出模板区域，调整阈值，阈值默认为0.8，阈值越高越难匹配。可以添加多个模板，可以在element上右键添加模板图像。注：task的type等于number时，默认会生成10个模板，需要自己添加0-9的图片 |
| condition | 每个通道的像素值范围，如："R < 100, G > 100, 90 < B < 130”，表示过滤Red通道灰度值小于100且 Green通道灰度值大于100且Blue通道灰度值在90和130之间的所有像素值 |
| filterSize | 形态学处理的滤波器大小。(整型) |
| maxPointNum | 返回最大像素点个数 |
| intervalTime | 计时器，隔多久检测一次（s） |
| threshold | 阈值，0-1之间，默认0.8，越大越难识别到 |
| cfgPath | 网络的cfg文件，选择一个路径 |
| weightPath | 权重文件路径，选择一个路径 |
| namePath | 名字文件路径，选择一个路径 |
| maskPath | 做了掩膜的图片路径 |

**添加template流程演示：**

当type为fix object时，固定物体检测使用模板匹配的方法，默认会生成一个模板（template），当type为number时，数值检测也是使用模板匹配的方法，默认会生成十个模板（0-9）

当需要多个模板时，可以在需要添加添加模板（template）的识别项（element）上右键添加模板



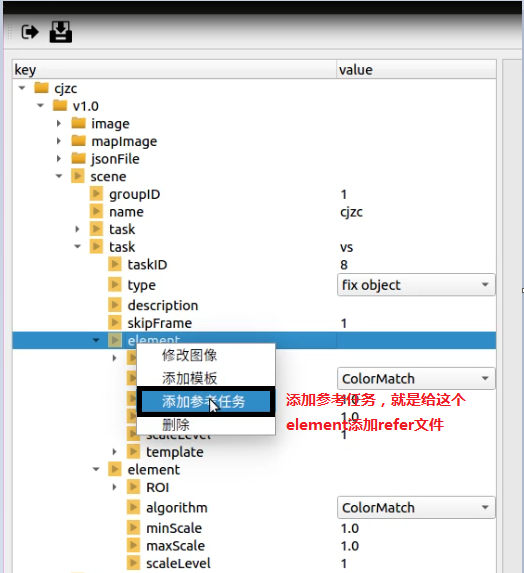
**3-1-3-3添加模板**

**3\_1\_4、refer文件**

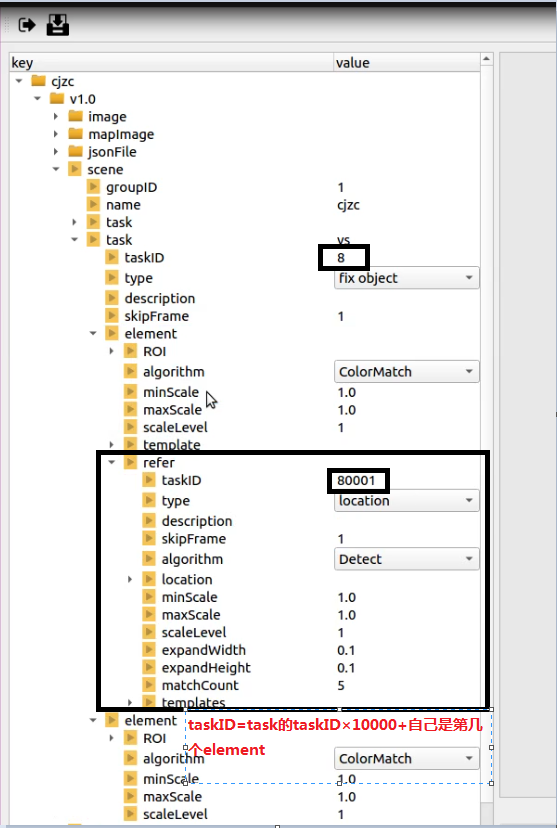
refer文件是用来做多分辨率下task文件每个element的ROI区域的推测。

**添加参考任务流程演示：**

可以在你想要添加refer的element上右键添加参考任务，就会在element的下面生成refer文件。



**3-1-4-1添加参考任务**



**3-1-4-2生成refer文件演示**

refer文件参数：

|  |  |
| --- | --- |
| **Key** | **描述** |
| taskID | 默认等于task\*1000+是第几个element |
| type | location：多分辨率情况下推测检测区域和尺度变化  Bloodlengthreg：多分辨率情况下血条长度识别 |
| description | 描述，可不填，建议填上 |
| skipFrame | 默认为1，可修改，你希望隔多少帧处理一次 |
| algorithm | type：location  Detect：静态  Infer：动态  type：Bloodlengthreg  TempplateMatch |
| location | 匹配位置，表示为(x, y, w, h) |
| minScale | 最小尺度，默认为1.0 |
| maxScale | 最大尺度，默认为1.0 |
| scaleLevel | 尺度等级，默认为1 |
| expandWidth | 扩展宽度，默认为0.1 |
| expandHeight | 扩展高度，默认为0.1 |
| matchCount | 匹配次数，默认为5 |
| condition | 每个通道的像素值范围 |
| templates | 模板：  模板路径：path  匹配阈值：threshold |

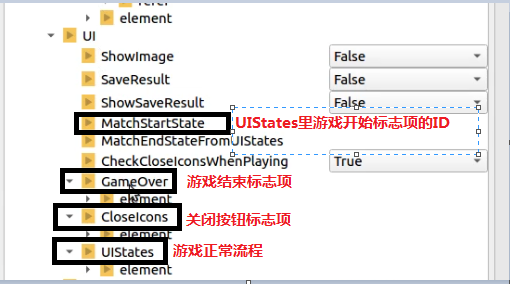
**3\_2、添加UI**

|  |  |
| --- | --- |
| **Key** | **描述** |
| ShowImage | 展示UI窗口 |
| SaveResult | 保存匹配图片 |
| ShowSaveResult | 在窗口中显示匹配结果 |
| MatchStartState | 配置一个id  UIStates里面点击游戏开始的UI（UIStates里标志游戏开始的id） |
| MatchEndStateFromUIState | 现在没有用 |
| CheckCloseIconsWhenPlaying | 是否在游戏中检查CloseIcons（true：检查，false不检查） |
| GameOver | 用于配置游戏结束的画面，如胜利，失败 |
| CloseIcons | 用于配置多次弹出某些重复的图标，比如某些广告界面，很多里面内容都不一致，但上面的关闭图标都是一样的，就可以把关闭图标配成CloseIcons，CloseIcons会在全图去寻找关闭图标 |
| UIStates | 用来配置进入游戏的流程，如挑选模式，开始游戏 |

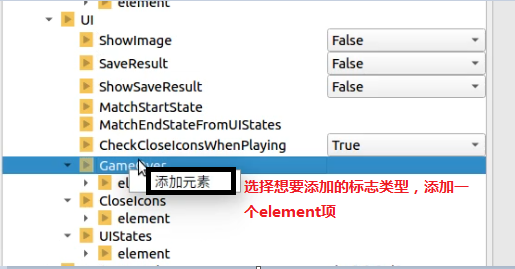
GameOver,CloseIcons，UIStates都是配置element，每个element就是一个动作。

**3\_2\_1、添加UI识别项流程演示：**

element的添加是先确定需要配置的element类型，比如要配置一个GameOver的element，先将光标移到GameOver上，然后右键，添加元素，就添加了一个GameOver的element。



**3-2-1-1添加UI后生成文件**



**3-2-1-2添加元素**

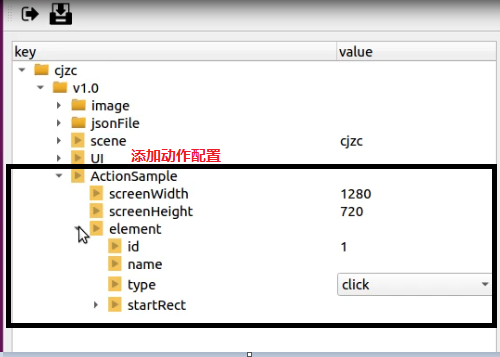
|  |  |
| --- | --- |
| **Key** | **描述** |
| id | 用作标志，最好不要重复 |
| action\_type | 动作类型分为click点击动作，drag滑动动作 |
| desc | 描述，可不填 |
| imgpath | 用作模板的图像；选择一张图像，会在右侧的框中显示出来 |
| ROI | 模板图像的标志区域；在右侧的图像上左键拉一个框然后释放，ROI会记录左上角点的x，y，区域的宽w，高h |
| shift | 搜索区域等于ROI值加上shift值；默认值为20 |
| action | 点击动作，滑动动作；点击：在右边的图像上左键点一个点，滑动：在右边图像上左键从A点滑到B点释放，会记录下x，y值；当游戏中ROI区域匹配上时，就会做这个操作 |
| template | 是否使用模板匹配，0不使用模板匹配，1使用模板匹配 |
| keypoints | 特征点个数，默认为100 |

**3\_2\_2、UI配置参数：**

**3\_3、添加动作配置**

**3\_3\_1、动作配置参数：**

动作配置文件用于模仿学习录制样本，可以在版本上右键，添加动作配置。



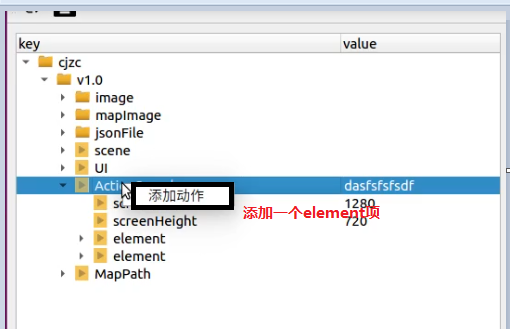
**3-3-1-1添加动作配置生成项演示**

|  |  |
| --- | --- |
| **Key** | **描述** |
| screenWidth | 图像宽度 |
| screenHeight | 图像高度 |
| element | 每个动作配置 |

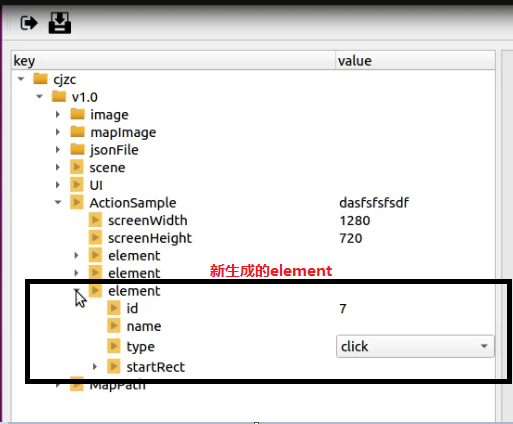
会生成以下配置项：

**3\_3\_2、添加动作流程演示：**

添加一个动作就是添加一个element项，在ActionSample上右键，添加动作。



**3-3-2-1添加动作**



**3-3-2-1生成element演示**

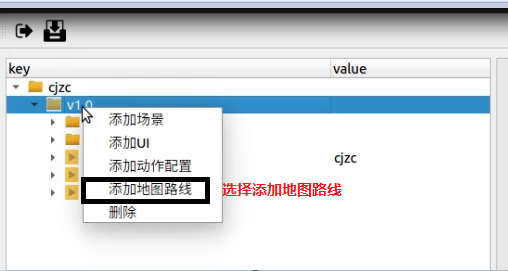
element的参数有：

|  |  |
| --- | --- |
| **Key** | **描述** |
| id |  |
| name |  |
| type | 动作类型有5种：  none：在某个区域没有动作  down：触点按下去的动作，在某个区域内点下去  up：触点弹起，释放某个区域点下去的触点  click：down+up一次点击操作  swipe：滑动，从某个区域滑到另一个区域，只有swipe有endRect |
| startRect | 在这个区域内上面定义的动作发生，会进行记录。  区域定义为x，y，w，h左上角的横坐标和纵坐标与宽高 |
| endRect | 只有swipe动作会有  会记录从startRect到endRect的滑动动作，通常用于调视角 |

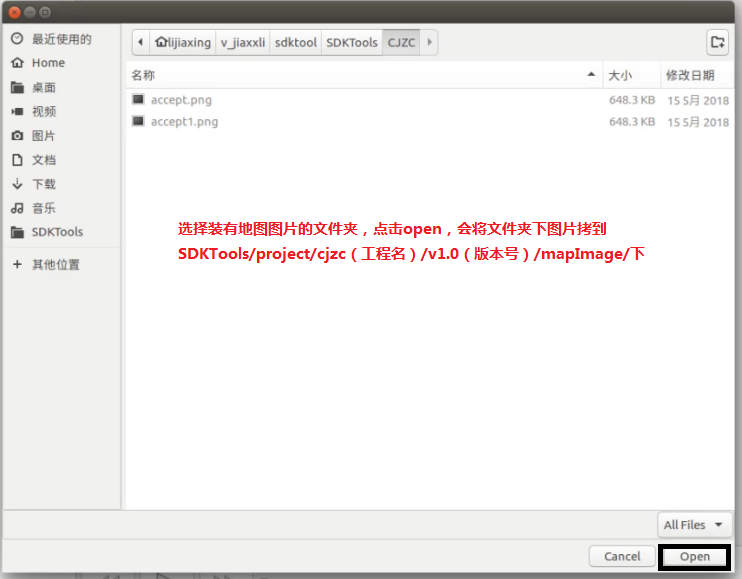
**3\_4、添加地图路线**

在版本号上右键添加地图路线，然后点击MapPath，选择装有地图图片的文件夹，会将文件夹拷贝到SDKTools/project/cjzc（工程名）/v1.0（版本号）/mapImage/文件夹下。

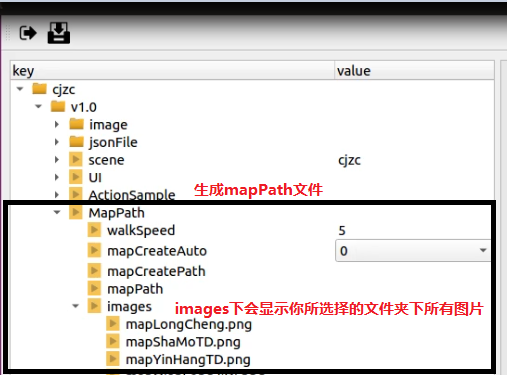
**3\_4\_1、添加地图路线流程演示：**



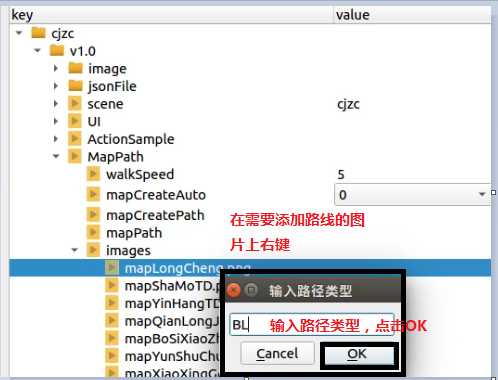
**3-4-1-1添加地图路线**



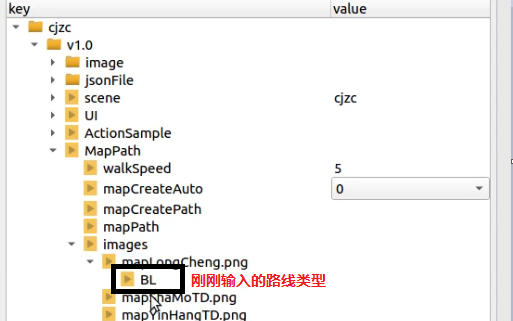
**3-4-1-2添加图片**



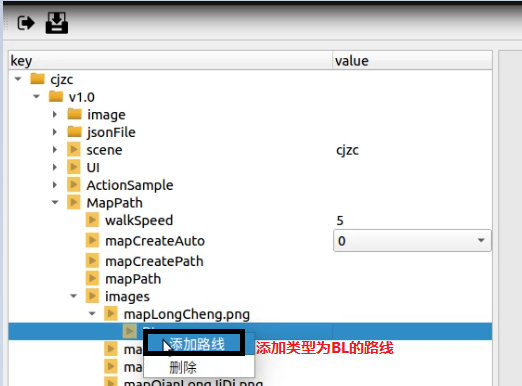
**3-4-1-3生成文件演示**



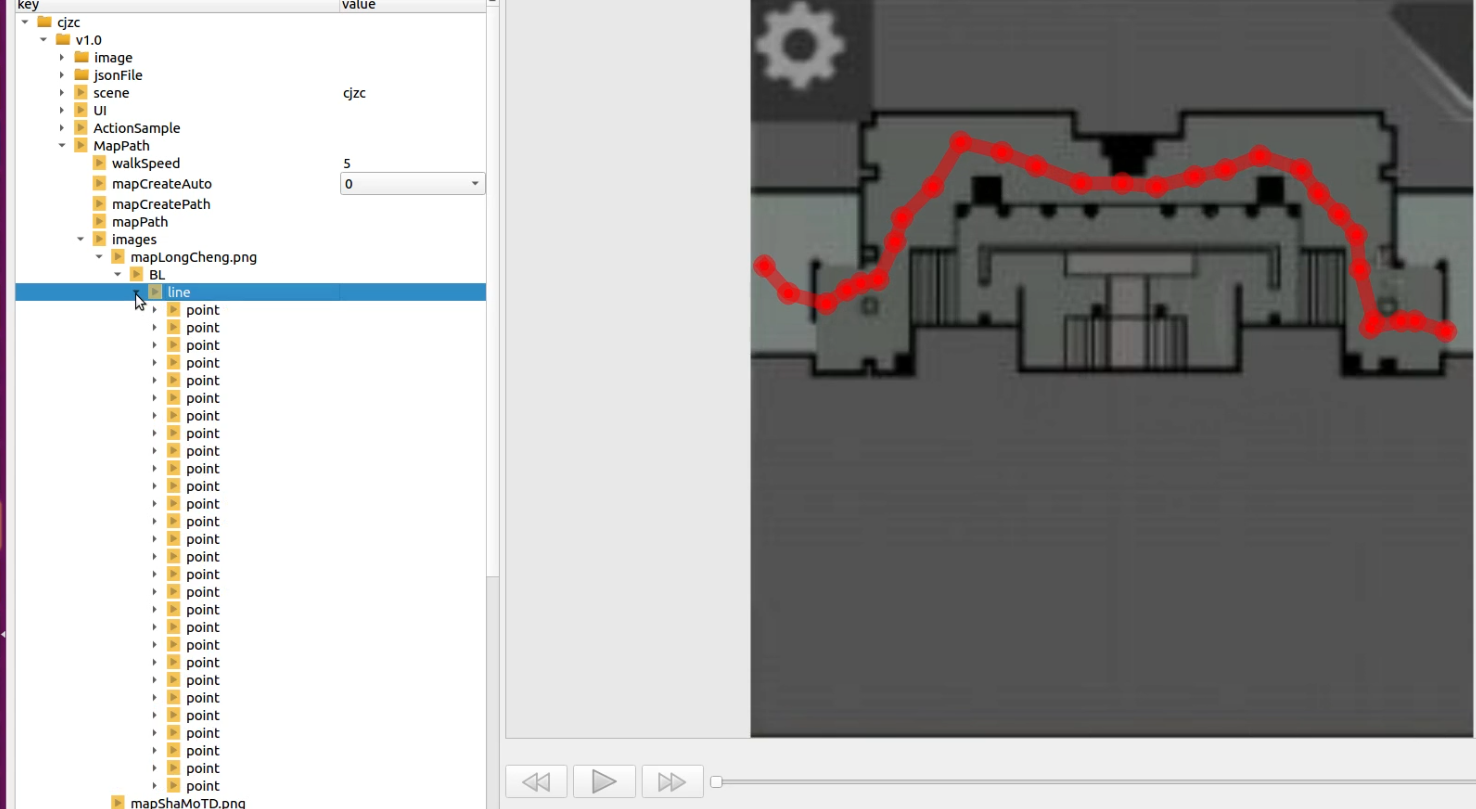
**3-4-1-4输入路径类型**



**3-4-1-5输入路径类型**



**3-4-1-5添加路线**



**3-4-1-6图片上添加具体路线**

**3\_4\_2、地图路线配置参数：**

|  |  |
| --- | --- |
| **Key** | **描述** |
| walkSpeed | 移动速度，默认为5 |
| mapCreateAuto | 是否自动采集地图 |
| mapCreatePath | 采集地图的存放位置 |
| mapPath | 地图位置 |
| images | 下面是你导入文件夹下所有图片的名字 |

Images下会显示你导入文件夹的所有图片的名字，在Images下你想要添加路径的图片上右键添加路线类型（比如CFM有警匪阵营，同样地图警匪出生点不同，走线是不同的），在路线类型上右键添加路线，会在右边显示出图片，可以在右边图片上标路线点，鼠标左键按压弹起就是一个点。一个类型可能会有多种走线方式，只需要在路线类型上右键添加路线即可。

# GameReg调试

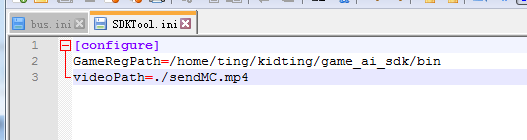
现在只针对task和refer文件

**4\_1、GameReg调试流程**

1、修改SDKTools/cfg/SDKTool.ini

GameRegPath：sdk的bin目录，这个参数填你的GameReg存在的地方

videoPath：基于某个视频做的图像识别效果，填写你想要识别任务的游戏场景视频

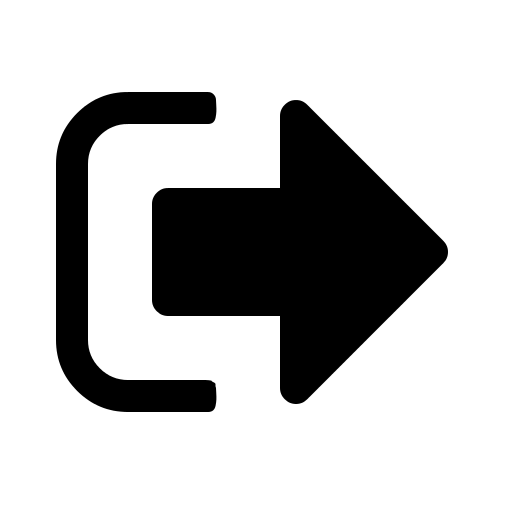


**4-1-1 SDKTool文件演示**

2、修改GameReg的识别类型

修改你的$AI\_SDK\_PATH下的cfg/task/gameReg\_mgr1.json的“Test”项

将Test下的“flag”改为true，“mode”改为SDKTool。如果没有$AI\_SDK\_PATH的环境变量，则修改sdk下cfg/task/gameReg\_mgr1.json。

3、点击test图标 ，进行调试，程序会将task文件和refer文件拷到$AI\_SDK\_PATH或者sdk下的cfg/task/下，并改名为task\_SDKTool.json和refer\_SDKTool.json

4、视频结果播放中，对某帧存在疑问，可以点击视频下方的暂停，恢复按钮。

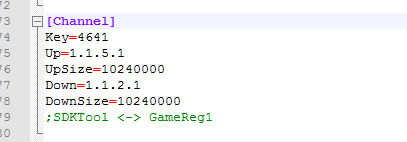
# 5、注意事项

1、配置好这些文件，需要在SDK下使用时，需要手动将这些cfg和data文件拷到相应的路径下，cfg中图片路径可能需要修改。

2、按顺序填写配置项；想要保存到json文件里，一定要先保存（ctrl+s；保存按钮）再关闭。

3、SDKTools保存的项目文件全都存在SDKTools/project/下，请注意不要误删。

4、使用GameReg调试时，注意SDKTools的bus.ini与GameReg的bus.ini通信端口一致



**5-1 bus.ini演示**

# 6、附

1、SDKTools的代码实现由kidtingzhu（朱以汀）完成，初期文档编辑由v\_jiaxxli（李佳兴）完成，如有问题请联系他们，谢谢。

2、配置文件的demo及解释，请见附件。

