

基于AR Engine 1.0 case制作

For Android

工程师 于清
增强现实实验室 (AR Lab)



内容目录：

1. AR case简介
2. AR case的结构
3. AR case制作工具
4. AR case制作流程
5. scene.json字段讲解
6. AR scene开发技巧

什么是AR case?

AR case简介:



顾名思义，AR case就是指一个AR案例，能够在特定场景下触发增强现实效果，生成一种逼真的视、听、力、触和动等感觉的虚拟环境，通过各种传感设备使用户“沉浸”到该环境中，实现用户和环境的直接自然交互。

AR case的结构

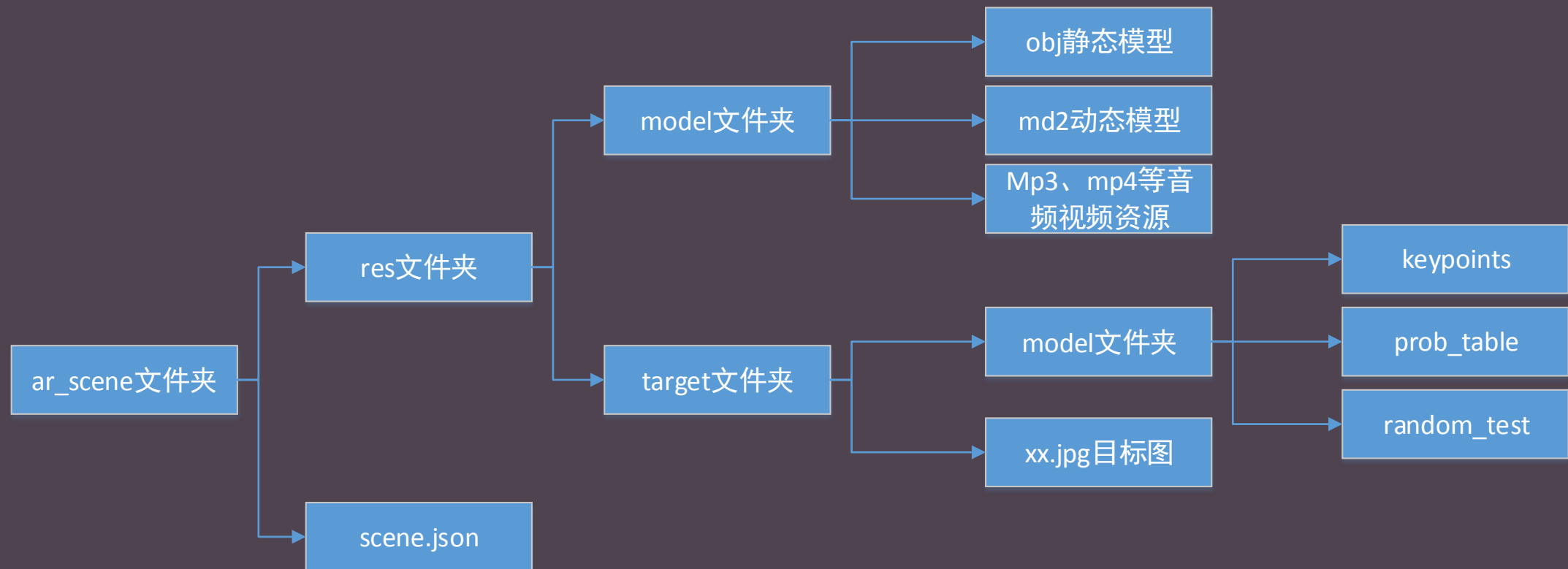
Android AR case中包含的文件：

Model: 模型，包括静态模型（.obj 3D模型、.jpg .png平面模型）和动态模型（.md2 3D模型）

Target: 与触发图相关的训练文件

Json: 设置AR场景中模型位置、大小、方向和动作的脚本

详细的目录结构：



制作AR case需要用到的工具

制作AR case需要用到的工具

objSerilized.jar序列化工具

该工具可将obj文件序列化生成objser文件，加快引擎加载和渲染静态模型的速度。

面向开发者的预览工具APK

在安卓开发测试机上安装该APK，将开发好的AR case导入到该应用文件夹中，打开应用即可预览该AR case效果。

AR JSON调试工具

提供丰富的json字段查询功能，并根据编辑的内容生成json格式，还提供了json文件的格式校验功能。



制作AR case流程

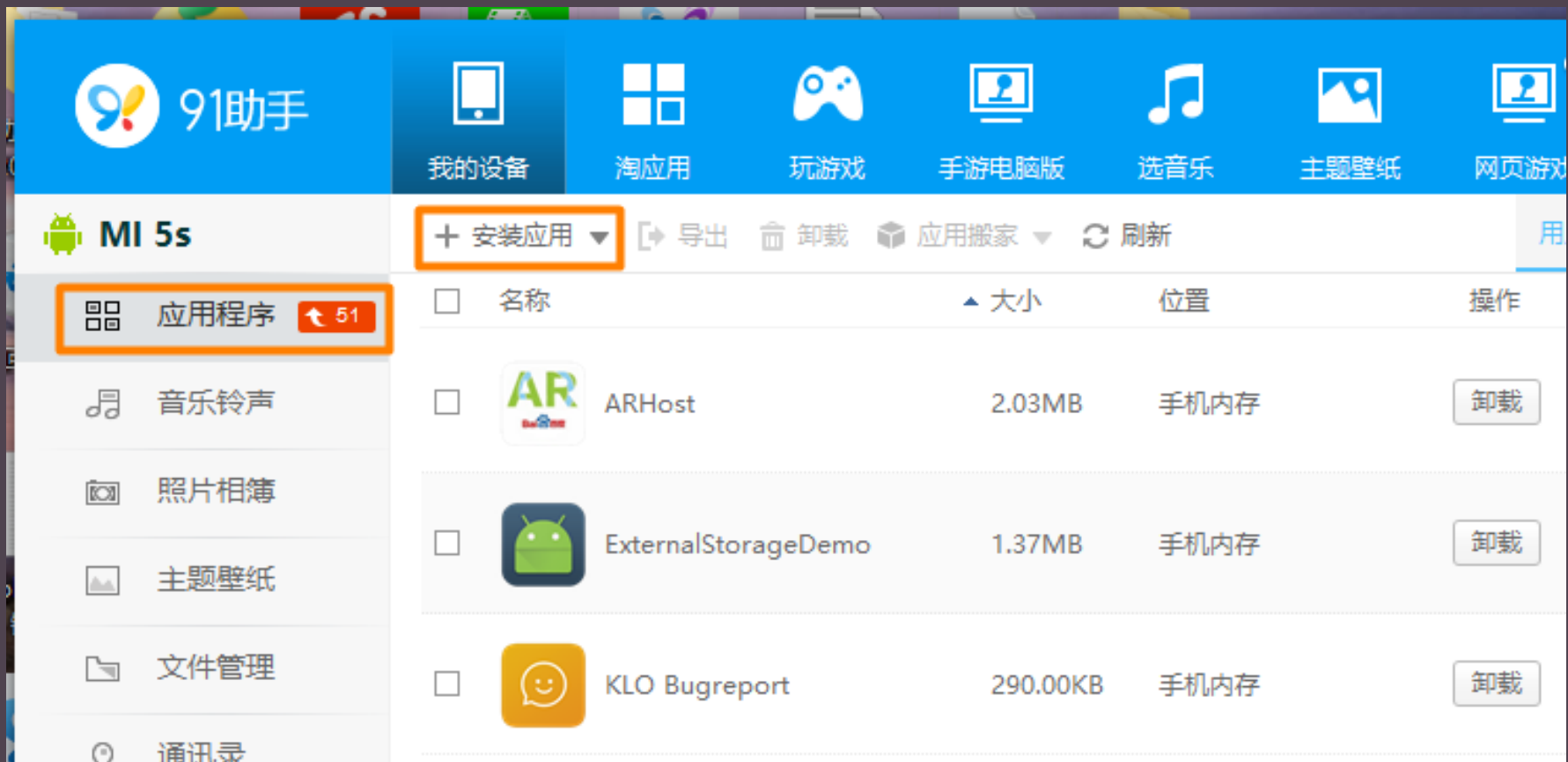
制作AR case流程:

1. 更改mtl文件中的贴图路径
2. obj 文件序列化
3. 将模型文件等资源导入case资源文件夹中
4. 将target训练文件导入case目标文件夹中
5. 制作scene.json 脚本
6. 将case工程导入到预览工具APP中
7. 使用预览工具预览

面向开发者的预览工具APK 使用方法

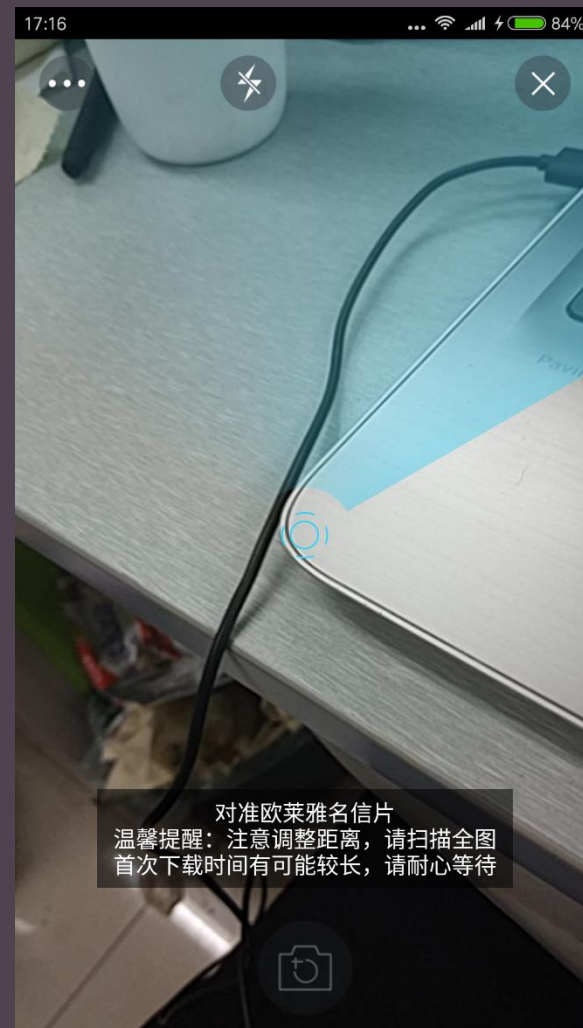
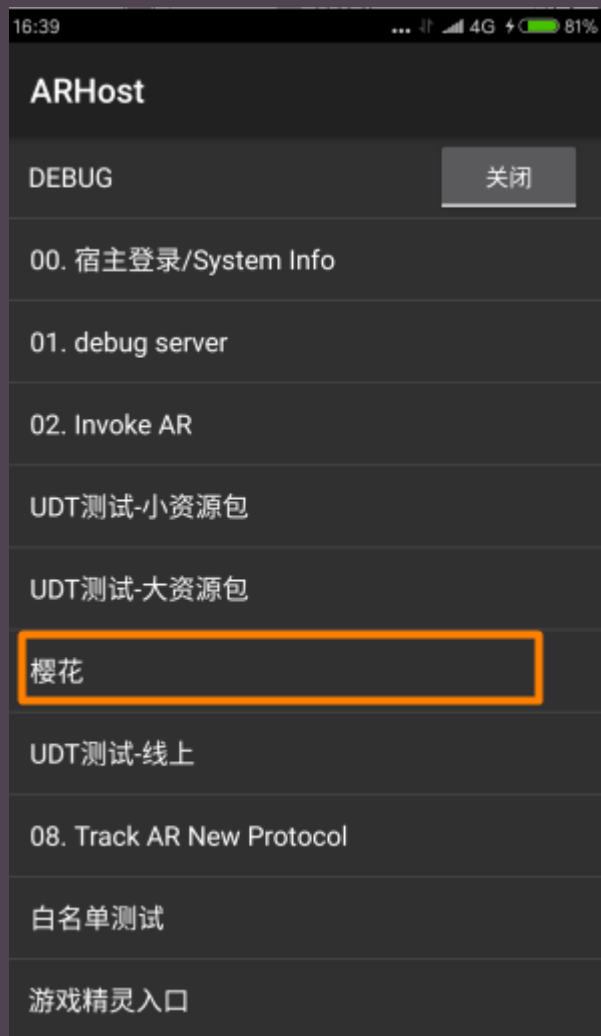
STEP 1:

将arhost-release.apk安装到安卓开发测试机上（推荐使用91助手）

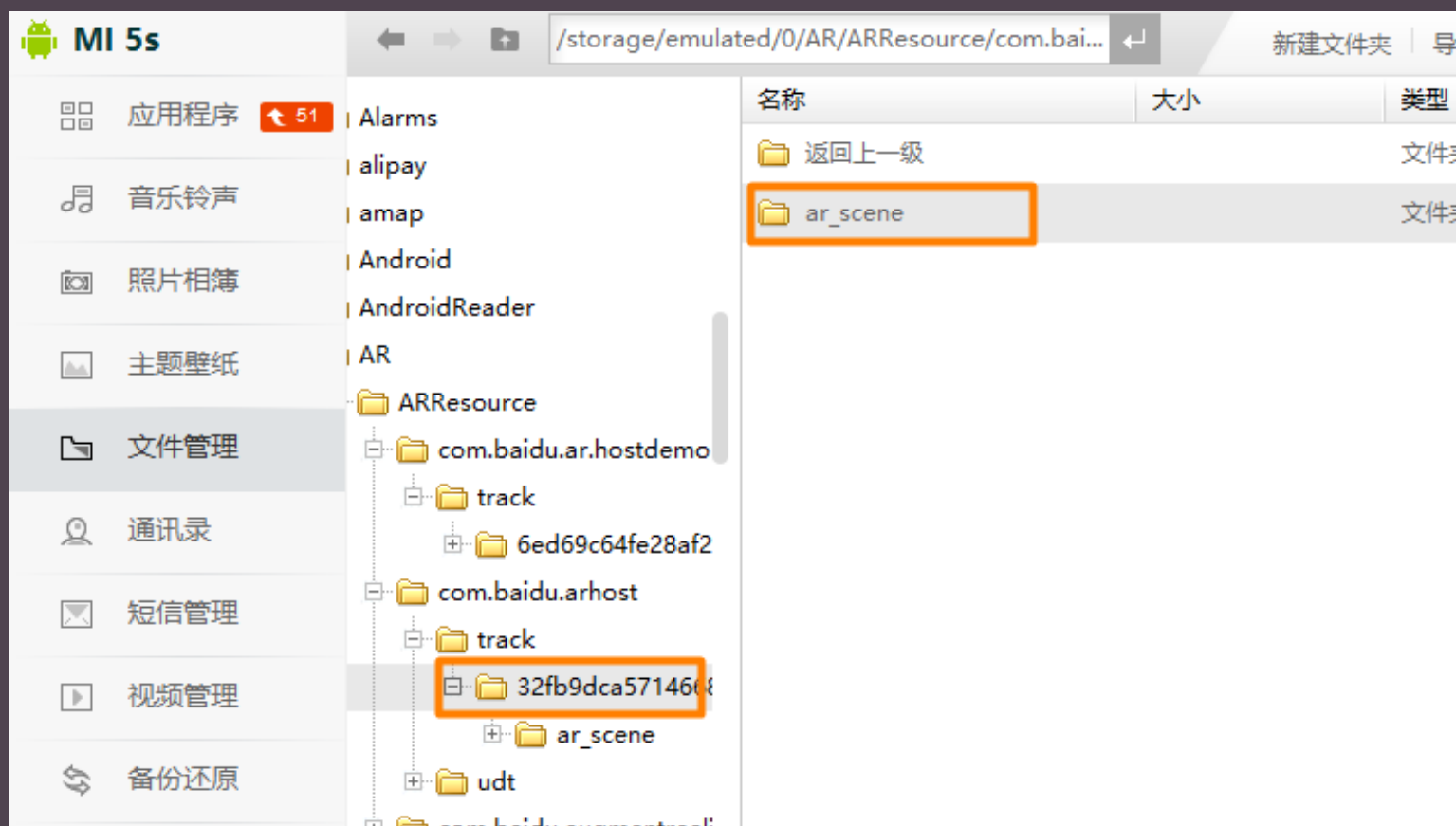


STEP 2:

打开安装好的APP（应用名ARHost），点击List中“樱花”这项，此时APP会从服务器上下载AR case包，出现下图，然后返回



STEP 3:
使用91助手（或者其他可以与手机互传文件的软件），找到手机SD卡中路径AR/ARResource/com.baidu.arhost/track，其中有一个名字以32开头的文件夹，用重新开发的case包替换其中的ar_scene文件夹(注意文件夹名字需要相同)



STEP 4:

替换完成后，重新点击APP List中“樱花”这项，即可预览开发者自己开发的case。

（注：首次使用预览APP时需执行步骤1、步骤2，之后只需重复执行步骤3、步骤4替换case包即可实现预览）

obj序列化化工具使用方法

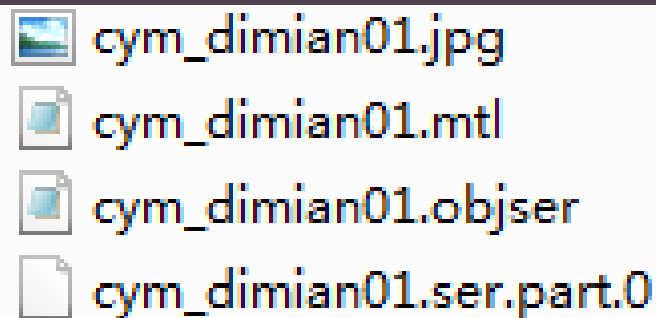
为什么要对obj模型文件序列化?

对obj模型文件序列化可提高引擎对obj静态模型的加载和渲染速度，减少用户等待时间，从而改善AR case用户交互时体验。

静态模型文件(obj)的组成:

在case中引用静态模型文件一般由四部分组成:

1. jpg文件，静态模型的贴图
2. mtl文件，材质库文件，描述的是物体的材质信息
3. objser文件，obj经序列化工具转换而来
4. ser.part.0文件，obj经序列化工具转换而来



更改mtl文件中贴图的相对路径：

用记事本打开mtl文件可看到其中对应贴图地址为绝对路径，将其修改为相对路径，并将该mtl文件和贴图一并放在case包中的资源文件中，如右图所示。

```
newmtl Material_34
 Ns 10.0000
Ni 1.5000
d 1.0000
Tr 0.0000
Tf 1.0000 1.0000 1.0000
illum 2
Ka 0.5880 0.5880 0.5880
Kd 0.5880 0.5880 0.5880
Ks 0.0000 0.0000 0.0000
Ke 0.0000 0.0000 0.0000
map_Ka C:\Users\Administrator\Desktop\shennongjia\bofang.png
map_Kd C:\Users\Administrator\Desktop\shennongjia\bofang.png
map_d C:\Users\Administrator\Desktop\shennongjia\bofang.png
```

```
newmtl Material_34
 Ns 10.0000
Ni 1.5000
d 1.0000
Tr 0.0000
Tf 1.0000 1.0000 1.0000
illum 2
Ka 0.5880 0.5880 0.5880
Kd 0.5880 0.5880 0.5880
Ks 0.0000 0.0000 0.0000
Ke 0.0000 0.0000 0.0000
map_Ka bofang.png
map_Kd bofang.png
map_d bofang.png
```

obj序列化工具使用方法

STEP 1:

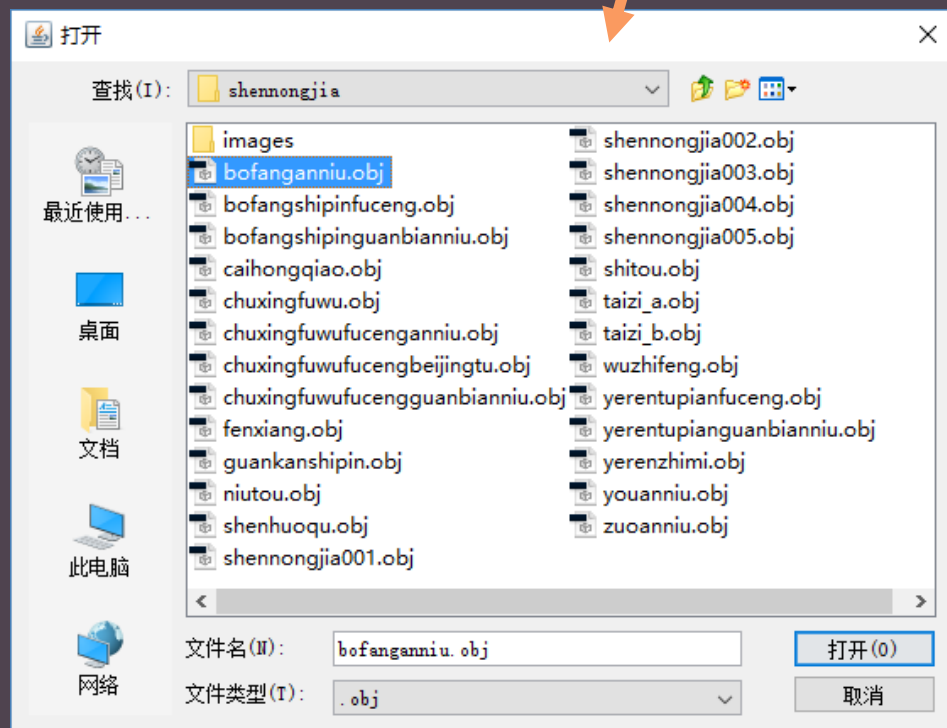
运行jar包需要配置java运行环境（可百度搜索“java运行环境”根据网上教程安装）

STEP 2:

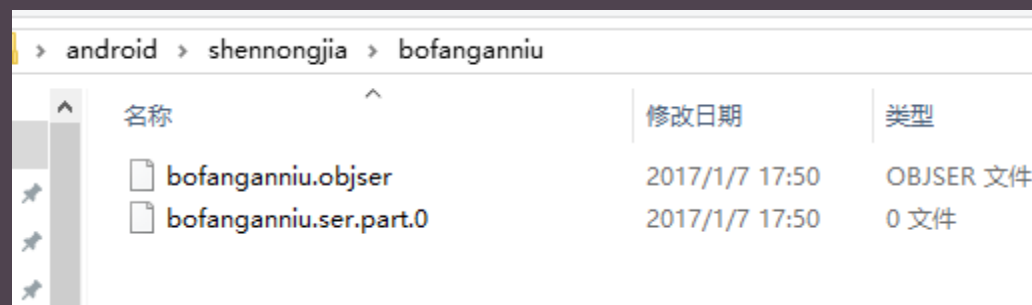
配置好java环境后直接双击运行jar包即可（若无法运行，可在cmd命令行中进入jdk安装路径中，然后运行，需根据jar包存放位置自行调整路径）

```
cd C:\Program Files\Java\jdk1.8.0_51\bin  
java -jar C:\Users\Administrator\Desktop\objSerialized.jar
```

STEP 3:
选择需要转换的obj文件



STEP 4:
转换完成后会在原文件夹中生成一个与已转换obj同名的文件夹,其中有两个文件, 将其复制到case的资源文件夹中即可。



scene.json中的结构

内容目录:

1. target 字段
2. sceneList 字段
3. objList字段中的静态类型
4. objList字段中的动态类型
5. objList字段中的随屏平面类型
6. instructions字段：指令集
7. instructions字段：单指令

target字段：与触发图相关

其中

model: 触发图训练文件的路径

hint: 扫描时提示

tooFarHint:触发后距离太远时提示

tooNearHint:触发后距离太近时提示

farThreahold: 产生距离太远提示的距离阈值

nearThreahold:产生距离太近提示的距离阈值

defaultTransform: 失去触发物后，模型（整体）的位置、大小、角度。去掉该字段后，失去触发物后不显示模型，重新回到扫描界面。

showImmediately：为true时无需触发物，立即显示模型

```
{
  "target": {
    "model": "res/target/model",
    "hint": "对准安定门图片\n温馨提醒:注意调整距离,请扫描全图",
    "tooFarHint": "距离太远, 请将镜头靠近目标",
    "tooNearHint": "距离太近, 请将镜头远离目标",
    "showImmediately": "true",
    "farThreshold": "1400",
    "nearThreshold": "750",
    "originImage": {
      "sizeWidth": "1248",
      "sizeHeight": "1248"
    },
    "featureImage": {
      "sizeWidth": "1024",
      "sizeHeight": "1024",
      "resource": "res/target/image_target.png",
      "position": "512,512,0",
      "targetWidth": "500",
      "targetHeight": "373"
    },
    "defaultTransform": {
      "position": "-490,460,1100",
      "scale": "0.9,0.9,0.9",
      "rotation": "270,90,75"
    }
  },
  "sceneList": [
    {
      "name": "andingmen",
      "cameraList": [
        {
          "cameraId": "001",
          "cameraType": "camera_2d_ui"
        },
        {
          "cameraId": "002",
          "cameraType": "camera_target_fix"
        }
      ]
    }
  ]
}
```

sceneList字段：与场景相关

其中

name:该场景名称

zFar: 最远处视距

zNear:最近出视距

cameraList:相机列表，主要设置随屏和跟随触发图两种状态

- cameraId:相机id
- cameraType:相机类型，camera_2d_ui表示随屏，camera_target_fix表示跟随触发图

objList:场景中的模型列表

```
        "targetHeight": "373"
    },
    "defaultTransform": {
        "position": "-490,460,1100",
        "scale": "0.9,0.9,0.9",
        "rotation": "270,90,75"
    }
},
"sceneList": [
    {
        "name": "andingmen",
        "zFar": "1200",
        "zNear": "500",
        "cameraList": [
            {
                "cameraId": "001",
                "cameraType": "camera_2d_ui"
            },
            {
                "cameraId": "002",
                "cameraType": "camera_target_fix"
            }
        ],
        "objList": [
            {
                "comment": "地面",
                "id": "adm_dimian",
                "type": "static",
                "cameraId": "002",
                // "position": "624.0,535.0,0",
                "position": "624.0,734.0,-535",
                "scale": "0.3,0.4,0.4",
                "rotation": "180,0,0",
                "texture": {},
                "lights": [
```

objList字段中的模型种类-静态类型

其中

comment:该模型名称

id:该模型的id,*此项比较重要，动作指令控制都是基于该id进行控制。

cameraId:使用相机的Id，是在cameraList中声明的，此处调用表明该模型是随屏还是随触发图

type:模型类型，static表示静态

position:在场景中的位置

scale:大小

rotation:角度

texture:纹理

isNeedDownSample:是否降低图片质量，设为false时模型清晰度会提高

lights:灯光

staticRes:静态资源

- type:资源类型，objser

- resource:资源路径

instructions:指令列表，可设置显示隐藏、设置点击事件、设置背景音乐、设置位移、缩放、旋转等刚体动画

```
{
  "comment": "安定门城楼",
  "id": "adm_tower",
  "cameraId": "002",
  "type": "static",
  "position": "624.0,734.0,0.0",
  "scale": "0.4,0.4,0.4",
  "rotation": "180,0,0",
  "texture": {},
  "isNeedDownSample": "false",
  "lights": [
    {
      "type": "directional",
      "direction": "1,0,0",
      "power": "0.5"
    },
    {
      "type": "directional",
      "direction": "-1,0,0",
      "power": "0.5"
    }
  ],
  "staticRes": {
    "type": "objser",
    "resource": "res/model/adm_chengmen.objser"
  }
},
```

objList字段中的模型种类-动态类型

其中

comment:该模型名称

id:该模型的id,*此项比较重要，动作指令控制都是基于该id进行控制。

cameraId:使用相机的Id，是在cameraList中声明的，此处调用表明该模型是随屏还是随触发图

type:模型类型，dynamic表示动态

position:在场景中的位置

scale:大小

rotation:角度

texture:纹理

isNeedDownSample:设为false时模型清晰度会提高

lights:灯光

dynamicRes:静态资源

- type:资源类型，md2
- resource:资源路径

instructions:指令列表，可设置播放md2自身动画，设置显示隐藏、设置点击事件、设置背景音乐、设置位移、缩放、旋转等刚体动画

```
{
  "comment": "士兵9",
  "id": "adm_shibing9",
  "cameraId": "002",
  "type": "dynamic",
  "position": "159.0,734.0,-645.0",
  "scale": "0.4,0.4,0.4",
  "rotation": "180,90,0",
  "texture": {},
  "lights": [
    {
      "type": "directional",
      "direction": "1,0,0",
      "power": "0.5"
    },
    {
      "type": "directional",
      "direction": "-1,0,0",
      "power": "0.5"
    }
  ],
  "dynamicRes": {
    "type": "md2",
    "resource": "res/model/namaoshibing.md2",
  },
  "instructions": [
    {
      "target": "adm_shibing9",
      "forwardLogic": "WaitForward",
      "backwardLogic": "CanBeCancled",
      "instructType": "InstructionSet",
      "instructCategory": "StartInstructSet",
      "instructId": "obi1",
    }
  ]
}
```

objList字段中的模型种类-随屏平面类型

其中

planePosition:2D 随屏目标物在屏幕的位置

- Percent:目标物屏占比widthAndHeight宽高
- screenOritation:屏幕方向 true 表示横屏 false 表示竖屏

alignParentLeft居左

- alignParentTop居上
- alignParentRight居右
- alignParentBottom居下
- centerVertical垂直居中
- centerHorizontal水平居中
- marginLeft左边距
- marginTop上边距
- marginRight右边距
- marginBottom下边距

texture:

- type:类型, image表示图片
- imageList:图片路径

transparent:透明

```
{
  "comment": "右下角按钮",
  "id": "adm_urlbutton",
  "type": "plane",
  "cameraId": "001",
  "cameraType": "camera_2d_ui",
  "rotation": "0,180,180",
  "planePosition":
    {
      "widthAndHeight": "105,105",
      "percent": "12",
      "screenOritation": "true",
      "normalizeWidth": "true",
      "alignParentRight": "true",
      "alignParentBottom": "true",
      "marginRight": "44",
      "marginBottom": "65"
    },
  "texture": {
    "type": "image",
    "imageList": [
      "res/model/urlbutton.png"
    ]
  },
  "attribute": {
    "sided": "single",
    "transparent": "true"
  },
  "lights": [
    {
      "type": "directional",
      "direction": "0, 0, -1",
      "power": "1"
    },
    {
      "type": "directional"
```

instructions字段：指令集

其中

target:执行指令的目标

forwardLogic:前向逻辑

backwardLogic:后向逻辑

instructType:指令类型

- InstructionSet代表指令集

instructCategory:指令策略

- 指令集中的StartInstructSet代表立刻执行指令，无需触发

- 指令集中的ClickInstructSet代表单击触发执行指令

instructId:指令ID,无特殊作用

param:指令集参数，一般将单指令放在param列表中

```
"instructions": [
  {
    "target": "tree_00001",
    "forwardLogic": "WaitForward",
    "backwardLogic": "CanBeCanceled",
    "instructType": "InstructionSet",
    "instructCategory": "StartInstructSet",
    "instructId": "obj1",
    "param": [
      {
        "target": "tree_00001",
        "forwardLogic": "WaitForward",
        "backwardLogic": "CanBeCanceled",
        "instructType": "AtomInstruction",
        "instructCategory": "PlayMusicInstruct",
        "instructId": "anniu_playBgm"
```

```
"instructions": [
  {
    "target": "guankanshipin",
    "forwardLogic": "WaitForward",
    "backwardLogic": "CanBeCanceled",
    "instructType": "InstructionSet",
    "instructCategory": "ClickInstructSet",
    "instructId": "id",
    "param": [
      {
        "target": "bofanganniu",
        "forwardLogic": "WaitForward",
        "backwardLogic": "CanBeCanceled",
        "instructId": "",
        "instructType": "AtomInstruction",
        "instructCategory": "AnimationInstruct",
        "param": {
          "time": "3000"
```

instructions字段：单指令-播放md2自身动画

其中

target:执行指令的目标

forwardLogic:前向逻辑

backwardLogic:后向逻辑

instructType:指令类型

- AtomInstruction代表单个指令

instructCategory:指令策略

- 单个指令中的PlayMd2AnimInstruct代表播放md2自身动画

instructId:指令ID,无特殊作用

playType:播放类型，repeat代表循环播放

```
{
  "target": "bear_1",
  "forwardLogic": "WaitForward",
  "backwardLogic": "CanBeCancled",
  "instructType": "AtomInstruction",
  "instructCategory": "PlayMd2AnimInstruct",
  "instructId": "instructIdPlayMd2-3",
  "param": {
    "playType": "repeat",
    "md2AnimName": ""
  }
}
```


instructions字段：单指令-缩放动画

其中

target:执行指令的目标

forwardLogic:前向逻辑

backwardLogic:后向逻辑

instructType:指令类型

- AtomInstruction代表单个指令

instructCategory:指令策略

- 单个指令中的AnimationInstruct代表动画指令

instructId:指令ID,无特殊作用

type:动画类型，scale代表缩放动画

duration和length：持续时间，一般设为相同值，单位为毫秒

delay:延时多少毫秒执行

updateRate:更新率，一般设为10

Interpolator:插值器

fromScale:起始大小，不填默认为动画开始时模型的大小，也可按toScale的格式填写，从精确大小开始动画。

toScale:终止大小

- animDataType:动画数据类型，absolute代表绝对值

- absoluteNum:值大小

```
{
  "target": "chuxingfuwuanniu_2",
  "forwardLogic": "WaitForward",
  "backwardLogic": "CanBeCancled",
  "instructId": "",
  "instructType": "AtomInstruction",
  "instructCategory": "AnimationInstruct",
  "param": {
    "type": "scale",
    "duration": "100",
    "length": "100",
    "start": "0",
    "delay": "0",
    "updateRate": "10",
    "direction": "1",
    "interpolator": {
      "type": "Linear"
    },
    "fromScale": {
    },
    "toScale": {
      "animDataType": "absolute",
      "absoluteNum": "0,0,0"
    }
  }
},
```


instructions字段：单指令-位移动画

其中

target:执行指令的目标

forwardLogic:前向逻辑

backwardLogic:后向逻辑

instructType:指令类型

- AtomInstruction代表单个指令

instructCategory:指令策略

- 单个指令中的AnimationInstruct代表动画指令

instructId:指令ID,无特殊作用

type:动画类型，translate代表位移动画

duration和length：持续市场，一般设为相同值，单位为毫秒

delay:延时多少毫秒执行

repeatCount:重复次数，4代表重复4次，共播放5次

repeatMode:重复模式，restart代表重新开始

updateRate:更新率，一般设为10

Interpolator:插值器

fromPosition:起始位置，不填默认为动画开始时模型的位置，也可按toScale的格式填写，从精确大小开始动画。

toPosition:终止位置

- animDataType:动画数据类型，absolute代表绝对值

- absoluteNum:值大小

```
{
  "target": "adm_shibing1",
  "forwardLogic": "WaitForward",
  "backwardLogic": "CanBeCancled",
  "instructType": "AtomInstruction",
  "instructCategory": "AnimationInstruct",
  "instructId": "anim_jiangjun",
  "param": {
    "type": "translate",
    "duration": "30000",
    "length": "30000",
    "start": "0",
    "delay": "0",
    "repeatCount": "4",
    "repeatMode": "restart",
    "updateRate": "10",
    "direction": "1",
    "interpolator": {
      "type": "Linear"
    },
    "fromPosition": {},
    "toPosition": {
      "animDataType": "absolute",
      "absoluteNum": "609.0,734.0,-645.0"
    }
  }
}
```

instructions字段：单指令-姿态动画

其中

instructType:指令类型

- AtomInstruction代表单个指令

instructCategory:指令策略

- 单个指令中的AnimationInstruct代表动画指令

instructId:指令ID,无特殊作用

type:动画类型，posture代表姿态动画

fromPosition:起始位置，不填默认为动画开始时模型的位置，也可按toScale的格式填写，从精确大小开始动画。

toPosition:终止位置

- animDataType:动画数据类型，absolute代表绝对值
- absoluteNum:值大小

fromScale:起始大小

toScale:终止大小

fromRotate:起始角度

toRotate:终止角度

```
{
  "target": "adm_kangqishibing2",
  "forwardLogic": "WaitForward",
  "backwardLogic": "CanBeCancled",
  "instructType": "AtomInstruction",
  "instructCategory": "AnimationInstruct",
  "instructId": "2",
  "param": {
    "type": "posture",
    "duration": "1000",
    "length": "1000",
    "start": "0",
    "delay": "0",
    "updateRate": "10",
    "direction": "1",
    "interpolator": {
      "type": "Linear"
    },
    "fromPosition": {
    },
    "toPosition": {
      "animDataType": "absolute",
      "absoluteNum": "640.0,734.0,-644.0"
    },
    "fromScale": {
    },
    "toScale": {
      "animDataType": "absolute",
      "absoluteNum": "0.3,0.4,0.4"
    },
    "fromRotate": {
      "animDataType": "absolute",
      "absoluteNum": "190,180,0"
    },
    "toRotate": {
      "animDataType": "absolute",
      "absoluteNum": "190,90,0"
    }
  }
},
```

instructions字段：单指令-播放音乐

其中

target:执行指令的目标

forwardLogic:前向逻辑

backwardLogic:后向逻辑

instructType:指令类型

- AtomInstruction代表单个指令

instructCategory:指令策略

- 单个指令中的PlayMusicInstruct代表播放音乐指令

instructId:指令ID,无特殊作用

resPath:要播放音乐的路径

```
{  
  "target": "MD4",  
  "forwardLogic": "WaitForward",  
  "backwardLogic": "CanBeCancled",  
  "instructType": "AtomInstruction",  
  "instructCategory": "PlayMusicInstruct",  
  "instructId": "anniu_playBgm",  
  "param": {  
    "resPath": "res/model/danding.mp3"  
  }  
}
```

instructions字段：单指令-改变可见状态

其中

target:执行指令的目标

forwardLogic:前向逻辑

backwardLogic:后向逻辑

instructType:指令类型

- AtomInstruction代表单个指令

instructCategory:指令策略

- 单个指令中的ChangePropertyInstruct代表改变属性指令

instructId:指令ID,无特殊作用

type:指令类型，change_visible代表改变可见状态指令

visibleType:可见类型，visible代表可见，invisible代表不可见

```
{
  "target": "btn_jinjing",
  "forwardLogic": "WaitForward",
  "backwardLogic": "CanBeCancled",
  "instructType": "AtomInstruction",
  "instructCategory": "ChangePropertyInstruct",
  "instructId": "",
  "param": {
    "type": "change_visible",
    "visibleType": "invisible"
  }
},
```

instructions字段：单指令-改变可点击状态

其中

target:执行指令的目标

forwardLogic:前向逻辑

backwardLogic:后向逻辑

instructType:指令类型

- AtomInstruction代表单个指令

instructCategory:指令策略

- 单个指令中的ChangePropertyInstruct代表改变属性指令

instructId:指令ID,无特殊作用

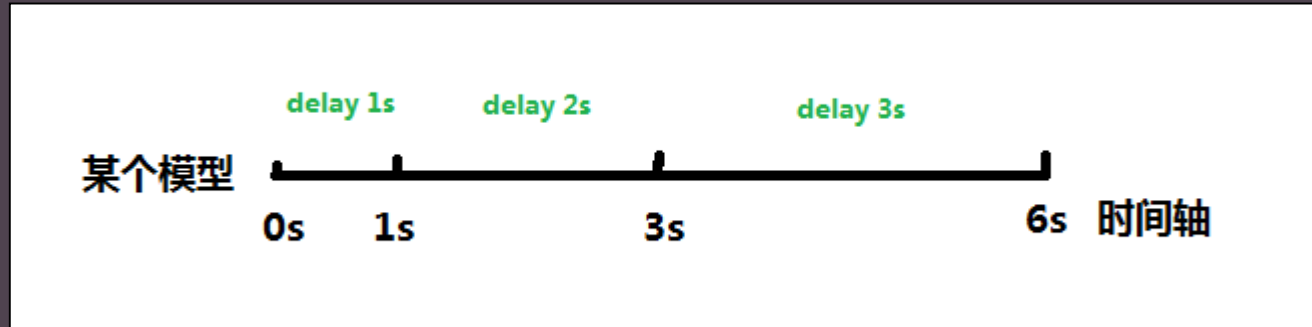
type:指令类型，change_clickable代表改变可点击状态指令

clickableType:可点击类型，clickable代表可点击，unclickable代表不可点击

```
{  
  "target": "chuxingfuwu",  
  "forwardLogic": "WaitForward",  
  "backwardLogic": "CanBeCancled",  
  "instructId": "",  
  "instructType": "AtomInstruction",  
  "instructCategory": "ChangePropertyInstruct",  
  "param": {  
    "type": "change_clickable",  
    "clickableType": "clickable"  
  }  
},
```

AR case制作技巧

- 1. scene中不同模型在空间位置中不符合客观逻辑的遮挡。
- 调整Json文件中不同模型的顺序，把需要放在前面的物体放在json文件结构的最后。
- 2. 关于delay的使用
- delay设置的时间是相对本模型上次出现时间进行触发,而不是针对时间轴原点



- 3. case中包含大量md2文件,并且触发后加载非常缓慢
- 在开发机运行该case , 将md2文件编译生成的md2idx文件从手机导出 , 放到工程包中打包
- 4. scene中某个或某些模型破碎
- 出现这种状况一般是模型问题,
 - ① 该模型太大,包含面数太多 , 导致引擎不完全渲染
 - ② 该模型包含四角面 , 引擎支持仅含三角面的模型的渲染需通知模型进行相应的调整
- 5. scene中某个模型出现闪烁破碎(静态模型)
- 调整zFar和zNear的值 , 一般是zFar调小 , zNear调大 , 缩小范围。

- 6. scene中某个模型出现闪烁破碎(动态模型)
- 出现这种状况的原因一般是渲染时法线计算错误，解决方法是

```
"dynamicRes": {  
    "type": "md2",  
    "md2Norm": "singlefile",  
    "md2NormFrame": "1",  
    "resource": "res/model/shou.md2"  
},
```

THANK YOU

