歌曲测试内容说明：

1. 谱面关卡（data\stage\android\au\xxxx.aue）
2. 音乐原文件（Res\Music）
3. 动作谱面文件（Res\Animations\Controller）
4. 动作文件（Res\Animations）

为了便于测试，编写脚本实现自动检测。脚本执行步骤说明：

脚本执行路径：[\\172.16.1.25\publish\Music\_Check](\\\\172.16.1.25\\publish\\Music_Check)

1. 将音乐表中的关卡表复制粘帖至song.txt文件中，并保存成功（第一行titile也需复制，脚本中会自动删减表头的）
2. 执行checkmusic.bat，执行过程中自动更新四个资源文件夹；如果出现文件缺失，窗口内会显示缺失的文件名并终止运行；脚本一旦执行，就会顺序检查上面提及的四个测试内容；实现原理下面会一一解释说明

**一、谱面关卡文件检查**：

data\stage\ 是歌曲对应的谱面关卡文件，其中android目录是安卓设备加载的关卡，ios是ios设备加载的关卡。具体文件命令规则：/模式文件夹名称/song+歌曲id+.+模式缩写+难度缩写，如：stage\android\au\song0521.aue，代表的是歌曲id：521，歌曲难度：简单，歌曲模式：劲舞

|  |  |  |  |
| --- | --- | --- | --- |
| 模式id | 模式名称 | 对应文件夹 | 模式缩写 |
| 1 | 经典模式 | taigu | tg |
| 2 | 恋舞模式 | tradition | td |
| 3 | 泡泡模式 | osu | os |
| 4 | 劲舞模式 | au | au |
| 11 | 超级经典 | taigu | sg |
| 12 | 超级恋舞 | tradition | sd |
| 13 | 超级泡泡 | osu | so |
| 14 | 超级劲舞 | au | su |
| 21 | 心动模式 | heartbeats | hb |
| 5 | 星光模式 | rhythm | rh |
|  |  |  |  |
| 难度 | 难度说明 | 难度缩写 |  |
| 1 | 简单 | e |  |
| 2 | 普通 | n |  |
| 3 | 困难 | h |  |

脚本实现流程：

1、加载song.txt并进行操作，转换格式至log-song-stage.txt：

获取歌曲id，格式为3位数的补0，4位数的不操作，然后统一转换格式为song+歌曲id+“.”

获取模式id，转换为缩写格式，并标注所在文件夹目录

获取难度level，转换为缩写格式

最后格式：tradition/song0314.tdn

2、根据log-song-stage.txt中的内容，逐行在stage进行查找文件，判断是否存在，如果文件找不到则在执行窗口输出错误信息，信息中包含错误的关卡文件名称

1. **音乐原文件检查**：

Res/Music是歌曲对应的原文件，分别有三种格式后缀：.smp .sog

.smp :真机上播放的格式（安卓和ios都是使用smp的文件）

.sog :unity上播放的格式，unity客户端使用的文件后缀

脚本实现流程：

1、加载song.txt并进行操作，转换格式至log-song-smp.txt：

获取歌曲id，格式为3位数的补0，4位数的不操作，然后统一转换格式为song+歌曲id + .smp

最后格式：song0314.smp

2、根据log-song-smp.txt中的内容，逐行在music进行查找文件，判断是否存在，如果文件找不到则在执行窗口输出错误信息，信息中包含错误的关卡文件名称

1. **动作谱面文件及动作文件检查：**

Animations\Controller\songxxxx\Dance.txt 是歌曲对应的谱面文件，其中Motion:A0002150000 行代表的是歌曲谱面中需要播放的动作文件

 1、检查动作谱面Dance.txt脚本实现流程：

1、加载song.txt并进行操作，转换格式至log-song-controller.txt：

获取歌曲id，格式为3位数的补0，4位数的不操作，然后统一转换格式为song+歌曲id+/+Dance.txt

最后格式：song0314/Dance.txt

2、根据log-song-controller.txt中的内容，逐行在Animations\Controller进行查找文件，判断是否存在，如果文件找不到则在执行窗口输出错误信息，信息中包含错误的关卡文件名称

2、检查动作文件.assetbundle脚本实现流程：

1、   在Dance.txt中找到包含Motion:的行，并截取Motion:后面的字符输出到templog.txt

2、  统一格式，全部转换成大写字符，输出到findresult.txt

3、  因为动作名的后缀都是.assetbundle，所以在findresult.txt的基础上统一增加后缀，输出到newresult.txt，这么做的目的是将找到的动作变成文件全名称，便于下一步查找文件是否存在

最后格式：A000215.assetbundle

4、  根据newresult.txt中的内容，逐行在Animations进行查找文件，判断是否存在，如果文件找不到则在执行窗口输出错误信息，信息中包含错误的动作名称

5、  因为安卓和IOS的Controller是一样的，只是动作因为平台不一样，所以.assetbundle分别对应的是两个svn路径，因此IOS也使用安卓目录下Controller里的相关txt文件进行检查

**四、AllCombo最高分计算统计：**

StageTester工具路径：svn://172.16.1.237/Tools/StageTester/Bin/Debug

1. 打开StageTester.exe
2. 路径选择data\stage\中的每个模式对应的文件夹，会一一生成每首歌对应模式和难度的最高分
3. 输出的excel文件在对应的exe执行程序目录中，生成的文件名为：AllMaxScore.xls
4. Excel中第一列为歌曲id、第二列为模式id、第三列为难度、第四列为分数；其中第三列中，11、12、13分别表示为超级模式-简单、超级模式-普通、超级模式-困难