

纸牌程序设计

1. 实现需求

实现一个简单普通点击匹配以及回退功能

1. 需求

1. 需求1：手牌区翻牌替换

- 点击手牌区♥A，♥A会平移（简单MoveTo）到手牌区的顶部牌（♠4）并替换它作为新的顶部牌

2. 需求2：桌面牌和手牌区顶部牌匹配

- 点击桌面牌的♦3，卡牌会和手牌区顶部的♠4进行匹配【桌面牌区的牌只要和手牌区顶部牌点数差1就可以匹配，无花色要求】，点击的桌面牌（♦3）会平移到手牌区的顶部牌（♠4）并替换它作为新的手牌区的顶部牌

3. 需求3：回退功能

场景：点击♦3 -> 点击♥A -> 点击♠2 后；连续多次点击 **回退按钮**，各卡牌需要反着平移（简单MoveTo）到原位；直到无回退记录可回退；

2. 说明如下

1. 游戏采用cocos2d-x 3.17 + Visual Studio 2022 开发
2. 游戏窗口和分辨率设置为1080 * 1300，此大小下更为适配我的电脑（可调整）
3. 对于给定的关卡配置文件卡牌的位置信息也做了一定的修改来获得更好的显示效果

3. 程序结构介绍

1. configs文件夹

其内loaders/LevelConfigLoader相关文件实现一个初始从JSON文件加载关卡数据功能

在models/CardResConfig相关文件定义了卡牌的数据形式，如卡牌点数和花色对应的结构，和一些转换信息

2. controllers文件夹

在这里实现了当前游戏全局的控制器逻辑，导演类GameController也定义在此，负责场景的加载和初始化相关数据。还包括CardController，用来控制动画和卡牌的系列操作。

3. models文件夹

定义两个基础的结构：CardModel用于存储卡牌数据信息，不含视图；UndoModel用于存储用户执行的操作所产生的数据变动，供撤销使用

4. services文件夹

GameModelFromLevelGenerator用于定义游戏数据数组，提供一系列基础的数据操作；UndoService用于定义用户操作数据数组，提供一系列基础数据操作

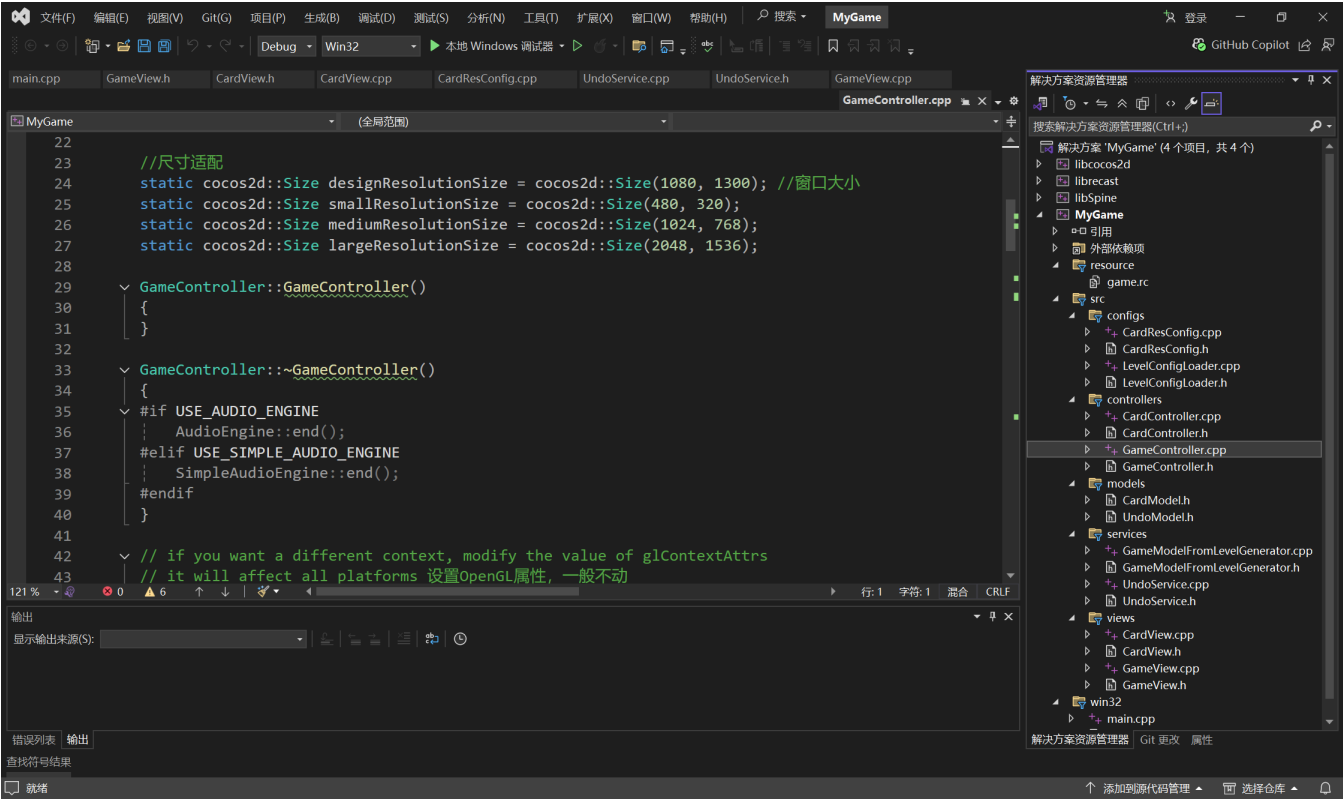
5. views文件夹

CardView用于定义卡牌UI布局，不包含卡牌具体数据；GameView用于指定游戏背景画布，卡牌展示在这个scene上

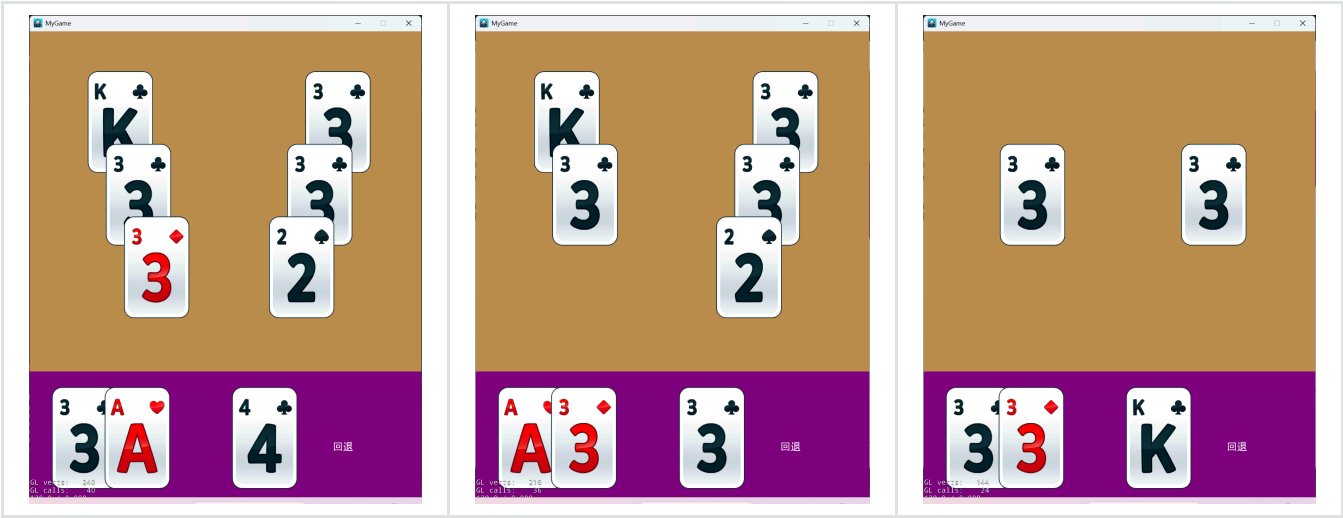
6. resources资源文件夹

添加了黑体字体文件和关卡配置levelConfig.json文件

4. 图片展示



运行结果图：



4. 其他

- 1. 使用了互斥访问的方法，避免多次点击造成数据访问错乱或丢失
- 2. 为回退按钮增加点击后颜色变化样式，便于观察点击效果
- 3. 手牌区为自定义摆放逻辑
- 4. 点击桌面卡牌时如有误则会左右摆动表示匹配失败
- 5. 可以同时多次点击任意元素，均不会出现错误

5. 添加新卡牌或新类型回退功能需要执行的操作：

1. 新增CardResConfig文件中枚举类型，相应的增加对于新名称文件命名string的map映射
2. 增加totalCardFaceNuner表示数字数量
3. 增加totalCardSuitNuner表示花色数量
4. 至此新增卡牌成功（注意只写了一套卡牌UI配置，如有变更需新增）

新类型回退功能：

1. 如果是完全不同的匹配逻辑，则需要增加Undo_Type枚举字段表示新的回退逻辑，然后再控制器里新增相关方法（如描述添加过程的函数和它对应的回退逻辑函数），在CardController的undoLastMove函数中添加case分支处理操作即可
2. 如果使用相同的匹配逻辑，不需要修改即可使用