项目说明文档

面向对象程序设计期末项目

——2048

作 者 姓 名： 郭辉

学 号： 1652688

学院、 专业： 软件学院 软件工程

**一.简介**

**1.1开发工具**

（1）开发环境：visual studio2017 cocos2d-x

（2）开发语言：c++

**1.2项目简介**

《2048》是一款比较流行的数字游戏，最早于2014年3月20日发行。原版2048首先在GitHub上发布，原作者是Gabriele Cirulli，后被移植到各个平台。这款游戏是基于《1024》和《小3传奇》的玩法开发而成的新型数字游戏.游戏规则很简单，每次可以选择上下左右其中一个方向去滑动，每滑动一次，所有的数字方块都会往滑动的方向靠拢外，系统也会在空白的地方乱数出现一个数字方块，相同数字的方块在靠拢、相撞时会相加。系统给予的数字方块不是2就是4，玩家要想办法在这小小的16格范围中凑出"2048"这个数字方块。

**1.3功能简介及操作指南**

（1）打开可执行程序或者通过Visual Studio运行程序后，将进入加载界面（开始界面），两秒钟后将进入游戏的主界面。

（2）通过鼠标上下左右滑动或者键盘上下左右滑动控制卡片的滑动。每滑动一次，所有的数字方块都会往滑动的方向靠拢外，系统也会在空白的地方乱数出现一个数字方块，相同数字的方块在靠拢、相撞时会相加。

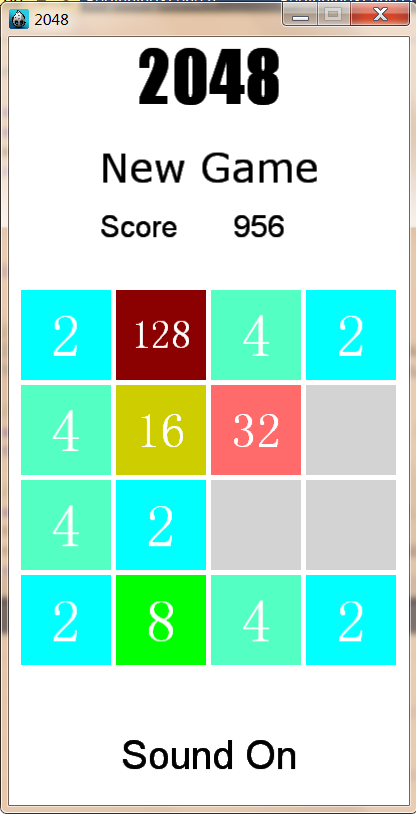
（3）在游戏的主界面中的Score显示当前的分数。

（4）下面Sound On和Sound Off键显示当前的声音状态，按键后则会转变为相反的状态。我们可以通过此按键来控制声音的开关。

（5）如果有一个卡片变为2048，则游戏胜利；如果在有一个卡片变为2048之前，所有的卡片堆满了整个4\*4的矩阵而且没有相邻的可以融合，则判定为游戏结束。

（6）游戏胜利和游戏结束的界面中，都会显示当前分数及本地的排行榜，在本地排前五名的分数可以上榜。

（7）游戏胜利和游戏结束的界面中，有一个Restart键，按此键后即会重新开始游戏；如果不想重新开始游戏，则关闭界面就可退出游戏。

** **

****

**二.设计**

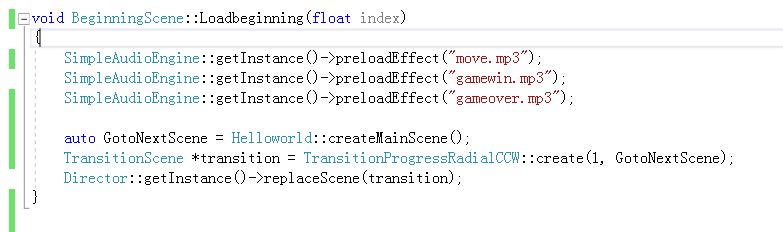
**2.1基础功能的实现**

**1.三个界面的定义**

本项目主要有三个界面：开始界面（加载界面）、游戏主界面和菜单界面（游戏胜利或游戏结束后的界面）。为了定义这三个界面，我定义了HelloWorldScene.h、BeginningScene.h、GoundOfMenu.h三个头文件和HelloWorldScene.cpp、BeginningScene.cpp、GoundOfMenu.cpp三个cpp文件（使用cocos2d-x创建项目时会自动生成AppDelegate.h和AppDelegate.cpp）。

**2.开始界面（加载界面）**

BeginningScene.h和BeginningScene.cpp的主要内容有：创建界面及初始化、设置界面背景、预加载音频和转入下一界面。其中，预加载音频和转入下一界面的功能是通过回调函数Loadbeginning来实现的，主要代码如下：



**3. 菜单界面（游戏胜利或游戏结束后的界面）**

此界面的主要功能为显示状态（游戏胜利还是游戏结束）、显示当前分数、显示本地排行榜、创建Restart按钮并使之被按后转入游戏主界面。这些功能在HelloWorldScene.cpp中实现，将在下面详细说明。

**4. AppDelegate.cpp**

在这里设置窗口，分辨率适配，并使开始界面作为初始场景。

# 3.上下左右手势的识别（鼠标）

# 在HelloWorldScene.h头文件和HelloWorldScene.cpp中添加事件监听回调和滑向上下左右的函数，并在HelloWorldScene.cpp中具体实现。我设立了onTouchBegan()函数和onTouchEnded()函数，在onTouchBegan()函数中触摸开始 ，获取开始触摸点，在onTouchBegan()函数中触摸结束，获取结束触摸点。因为这两个触摸点都有横纵坐标，判断X轴和Y轴的移动距离，如果X轴的绝对值大，则向左右滑动，如果Y轴的绝对值大，则向上下滑动。然后再通过这个移动的向量（偏移量）的正负来判断是左还是右（上还是下）。

主要代码如下：



# 4. 创建卡片类

# 新建头文件CardSprite.h和CardSprite.cpp文件，主要内容包括初始化（数值，宽度，高度，卡片X轴，卡片Y轴）获取数字、设置数字等等。在初始化时，设置卡片的背景，并默认所有卡片都为0（除了两个自动生成的卡片之外）。在游戏过程中，如果卡片中的数字大于0，则显示数字；如果不大于0，则不显示数字。我们还可以根据不同的数字显示不同的颜色与大小。

# 

# 

# 

# 

# 5.把卡片添加到游戏中

HelloWorldScene.h中声明创建卡片函数（卡片矩阵）void createCardMatrix(Size cardsize);并在HelloWorldScene.cpp中具体实现。将在init函数中调用。代码将在其他地方展示。

# 6.上下左右的识别（键盘）

在HelloWorldScene.h头文件和HelloWorldScene.cpp中添加键盘事件监听回调函数，并在HelloWorldScene.cpp中具体实现。我设置了一个onKeyReleased函数，键盘的输入为上下左右依次对应滑动的上下左右，并在操作后判定输赢及自动生成卡片。

# 

# 

# 

# 6.游戏滑动上下左右相加的逻辑

在HelloWorldScene.h头文件中设置数组CardSprite \*CardNumberMatrix[4][4]代表卡片的矩阵，并在HelloWorldScene.cpp 中把数字添加到数组中。当我们滑动的时候，以向左滑动为例，从下到上从左到右遍历，如果右边的数等于左边的数，那么左边的数乘以2，然后把右边的数设置为0，也就变为了空。以此类推。上下右也是同样的道理。代码将在其他地方展示。

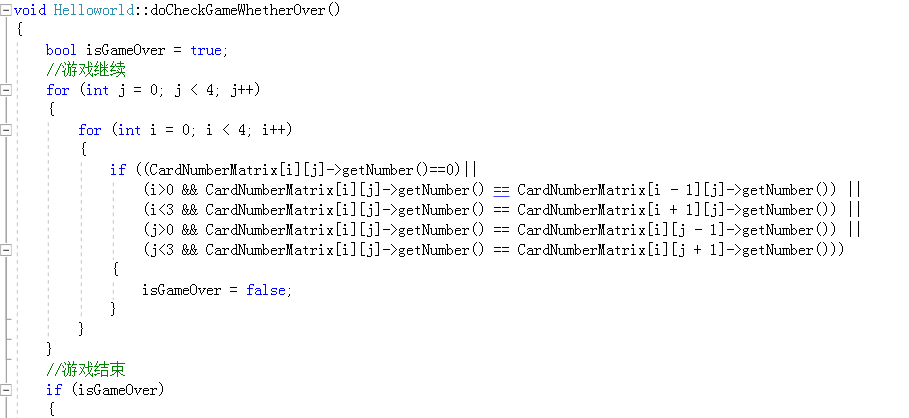
# 7.游戏自动生成随机卡片

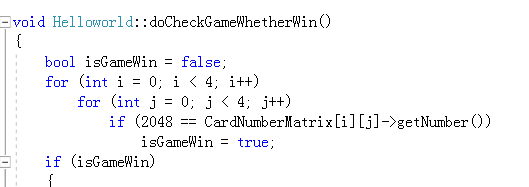
# 首先默认所有地方是不能添加的，然后遍历所有位置，如果这些位置上没有数字（即数值为0），则可以添加。然后在随机位置生成卡片，有10%的几率生成4（剩下90%的几率）生成2 。为了使生成有动画效果，我使用了这样一条语句auto action = Sequence::createWithTwoActions(ScaleTo::create(0, 0), ScaleTo::create(0.3f, 1))使该卡片在0.3秒内从小缩放到大。

# 8.判断游戏是否能够继续进行

如果游戏不能继续进行，则分为两种情况：游戏胜利或者游戏结束。如果有一个卡片变为2048，则游戏胜利；如果在有一个卡片变为2048之前，所有的卡片堆满了整个4\*4的矩阵而且没有相邻的可以融合，则判定为游戏结束。判断游戏胜利，只需遍历所有卡片看是否有卡片的数值为2048；判断游戏没有结束，则需要五个条件：还有空卡片、还可以向右滑、还可以向左滑、还可以向上滑、还可以向下滑。

代码：





# 9.添加游戏分数

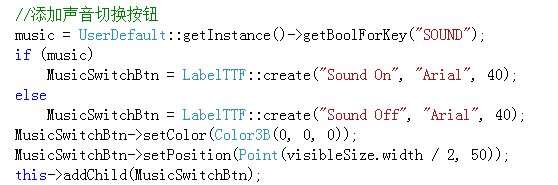
# 在HelloWorldScene.h头文件中定义当前分数的变量currentscore，在卡片合并的时候，分数要相应地改变。可以在上下左右滑动的函数中来完成这一功能。要先判断卡片是否合并，如果合并的话则加上所有合并了的卡片的数值。

**2.2其他功能的实现**

**1.添加声音开关按钮并加载声音**

添加按钮之后，将声音开始写作Sound On，将声音关时写作Sound Off，通过触摸的监听事件，在onTouchBegan中实现按键后状态和显示的转变。

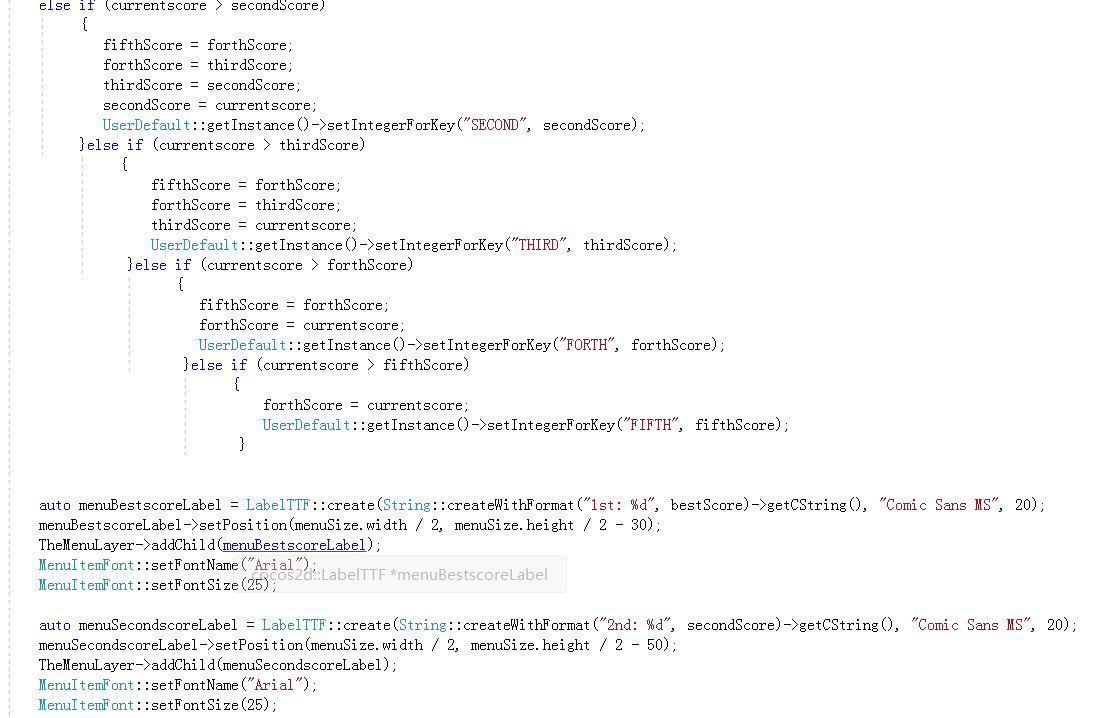
声音首先在BeginningScene.cpp中加载，当卡片融合时播放“move.mp3”,当游戏胜利时播放“gamewin.mp3”，在游戏结束时播放“gameover.mp3”。

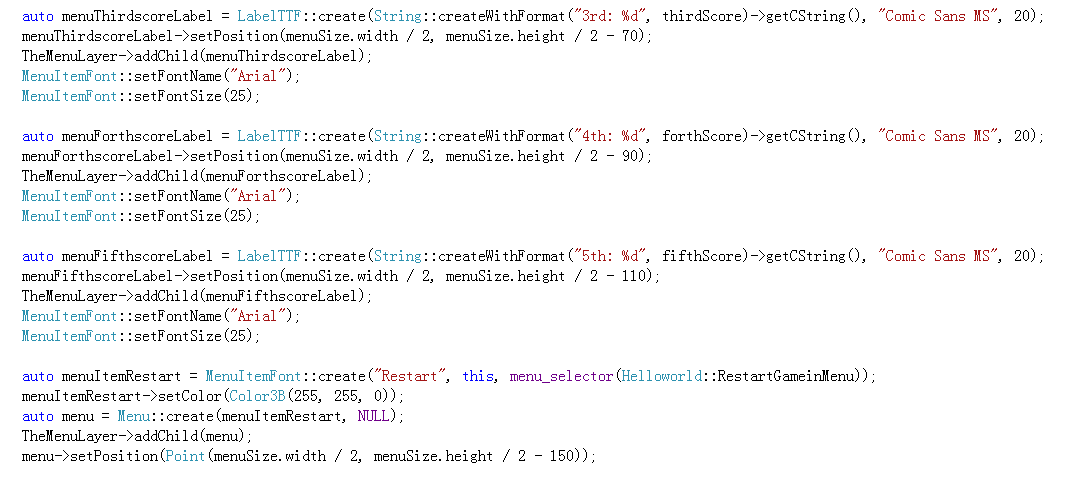




**2.本地的排行榜的实现**

游戏胜利和游戏结束的界面中，都会显示当前分数及本地的排行榜，在本地排前五名的分数可以上榜。在HelloWorldScene.h头文件中定义五个变量bestScore、 secondScore、thirdScore、forthScore、fifthScore并在HelloWorldScene.cpp判定游戏是否能够继续的两个函数中，将这五个值与当前分数进行比较，排序后取前五上榜。菜单层的代码如下：





**3.卡片动画的实现**

在HelloWorldScene.cpp文件中的创建卡片函数中，创建临时卡片矩阵，用于动画，每个动画卡片对应一个实际卡片的动画，这是个技巧,并且动画层在卡片层之上，所以后加入。平时把他们隐藏起来。当我们上下左右滑动时，以向左滑动为例，先计算出移动的步进间距，然后在动画层卡片实现定位、显现、移动、隐藏系列动画，每次都重新把动画卡片重新定位到实际对应的卡片位置，并设置相同的数字。这样就实现了卡片的移动。然后通过缩放等函数实现卡片的融合效果。



