2016-2017学年夏季小学期c语言程序设计实验实验报告

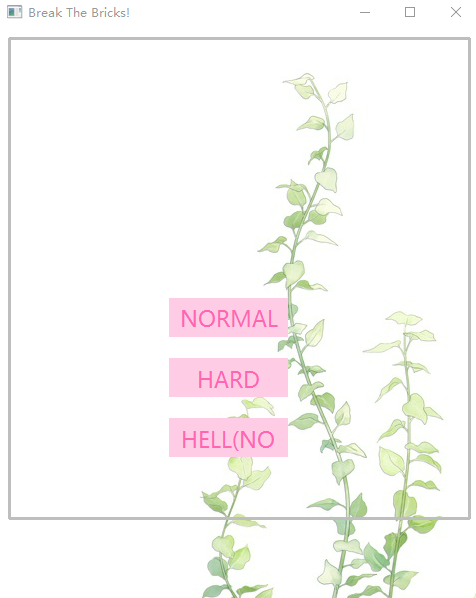
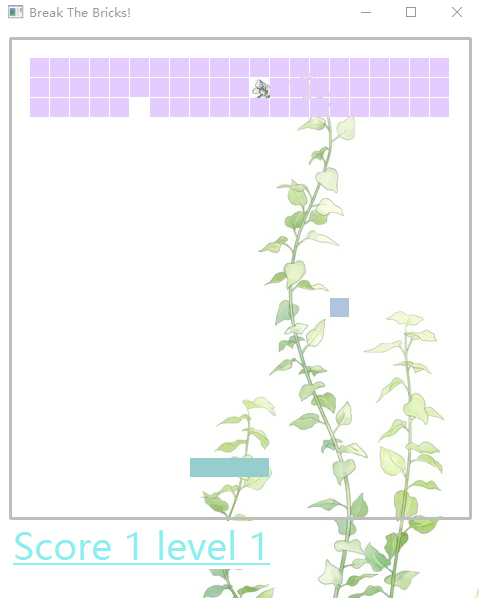
1. **项目标题**

打砖块

1. **实验效果**

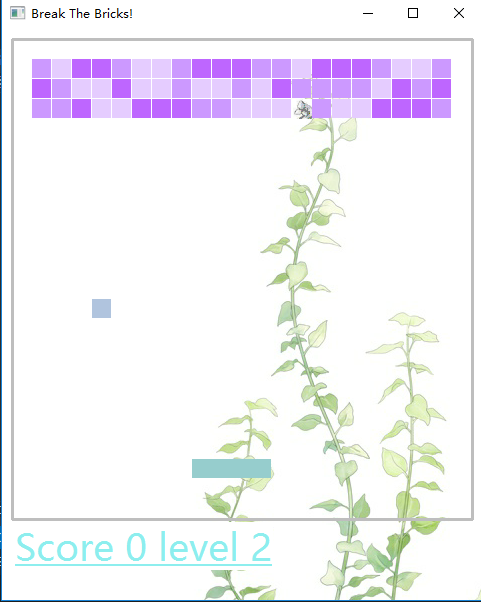
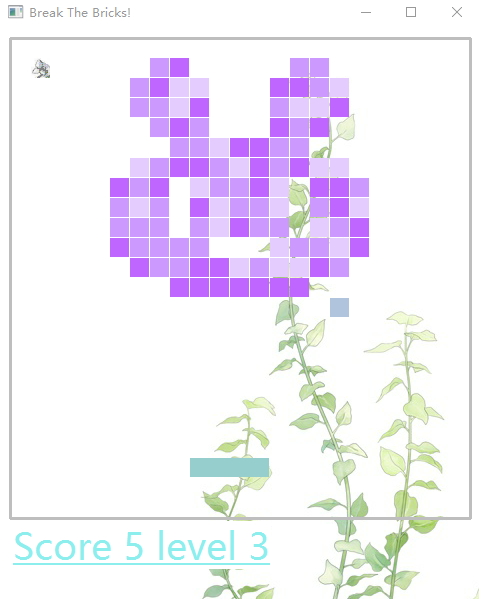
下述即为游戏最终完成后的效果。

主菜单界面，按下1、2、3可以进行关卡的选择。

第一关：最简单的一关，所有砖块可以一遍就打掉，速度也比较慢。在所有砖块中间有一个特别的砖块，上面有花的图案，只要打到这个砖块，玩家就可以直接获得胜利。

第二关：砖块出现了分类，共有三种砖块，按颜色深浅来区分，颜色最深的砖块需要打三次才能打掉，每打到一次颜色就浅一个层次。颜色第二深的砖块则需要打两次。同样地，只要打到随机生成的花纹砖块就能直接通关。

第三关：最难的一关，小球的速度是前两关的两倍，用二维数组设计了一个兔子的图案，兔子的下巴距离木板非常近，这也是本关的一个难点——打到兔子下巴时很可能来不及接住小球。和前两关一样使用了带有花纹的砖块。

游戏共计四个界面，每个界面有不同的背景音乐，小球在撞击到木板时和打到砖块时都有各自的音效，最后游戏结束时也设置了提示音，在后续的详细说明中会展示如何插入这些音效。

1. **实现方法**

游戏一共使用了两个.h文件和四个.c文件。下图即为文件一览。



我制作这个游戏的第一步是绘制窗口。游戏的窗口沿用了贪吃蛇的设计，自行调整了一下适合打砖块游戏的长宽。绘制窗口的函数写在ui.c当中。

下表即为所用的绘制函数。

|  |  |  |
| --- | --- | --- |
| 用途 | 函数 | 截图 |
| 画边界 | MoveTo()  LineTo() |  |
| 画砖 | Rectangle() |  |
| 画木板 |  |
| 画球 |  |
| 画花 | StretchBit()  LoadImage() |  |
| 画背景 | BitBlt()  loadImage() |  |

最早绘制的是窗口界面，然后是木板，接着是砖和小球。花和背景是在游戏能运行了之后加上的。想尽量将贪吃蛇和飞机大战里用到的函数都用一遍，也为了游戏的最终美观程度，才加上了花和背景。

绘制的函数写好之后，沿用贪吃蛇的CreateGame函数进行了游戏的初始化传参，设定了游戏界面的大小、球和木板及砖块的位置、木板的长度。

|  |  |  |
| --- | --- | --- |
| 对象 | 函数 | 截图 |
| 边界 | GetBoundary()  SetBoundary() |  |
| 砖 | GetBrickPointer()  CreateBricks()  DestroyBricks()  GetBrick()  SetBrickType() |  |
| 木板 | CreateBoard()  SetBoardDirection()  GetBoard() |  |
| 球 | CreateBall()  GetBallPosition()  BallMove()  GetBallDirection()  SetDirction()  isBallBreakBrick() |  |

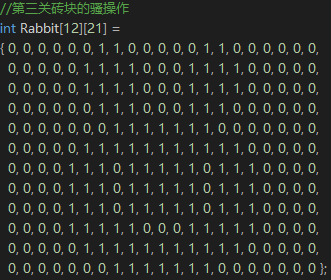
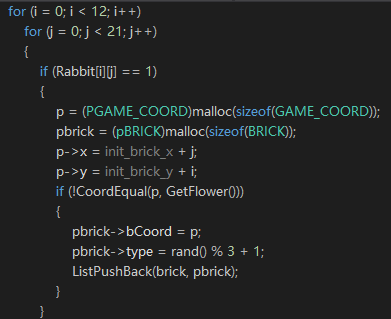
创建砖块在后期一共有三个函数，根据玩家选择的关卡来创建不同难度的砖块，砖块的状态有三种，分别是NORMAL、HARD、DIAMOND。木板的方向只有左右。球的状态一共有8种，分别是移动、死亡、打到左、右、上边界、打到木板、打到砖块、打到通关的花。球的方向一共有6种，分别是左上、左下、右上、右下、直上、直下。根据球的状态不同，判定是否改变方向及如何改变方向。

写好以上函数之后，调试OnTimer函数使得小球的位置可以进行更新，游戏也就可以正常运行起来了。通过OnkeyDown函数操控木板移动以接到小球，如果没有接到，则游戏结束，关闭计时器。

在游戏基本完成之后，我开始改进游戏体验，第一步是添加了背景音乐。通过阅读网上资料之后添加了头文件，并学会了在正确的地方新建线程插入音效。最终音乐的效果是，主菜单页和三个游戏界面都有不同的音乐，小球在击中木板、砖块及死亡时有不同的音效。

添加完音效之后我研究了一下主菜单界面，最终给游戏设计了一个GameState的全局变量，用以控制砖块生成等。

自己比较喜欢的一个设计就是第三关的兔子。一开始是想在CreateBricks直接手动把需要画的砖块加进去，后来使用了一个由0,1组成的二维数组，比较简单地画出了那只兔子。（1表示要画砖块的地方）

到此这个游戏就做完了，最后要展示之前加上了一个暂停功能，其实就是一个控制计时器的小操作，为了能更好地展现自己的游戏画面。

1. **总结**

个人对这次小学期的作业效果十分满意，首先，完成度比较高，超出了我的预期效果；其次，画面非常清新好看，符合个人的审美。

本次作业让我学到了很多东西，尤其是有关Windows API及GDI的部分，同时也让我了解到了游戏初始策划的重要性。在一开始设计游戏时我并没有考虑到多种类型的砖块，导致在后期才将砖块的结构改成了结构体，去存储其属性，这样的工程量是非常大的。以后如果还要写一个这样的大型项目，一定要把最终的效果明确好，不要中途进行特别大的修改。

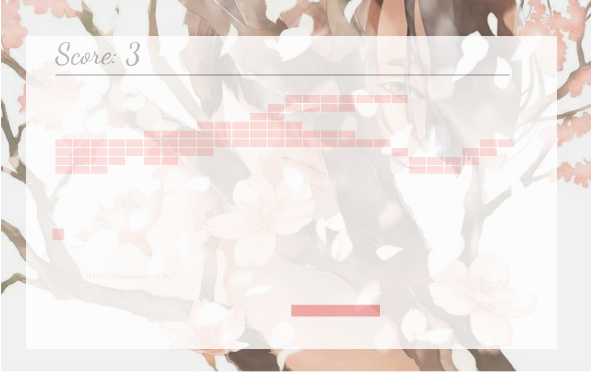
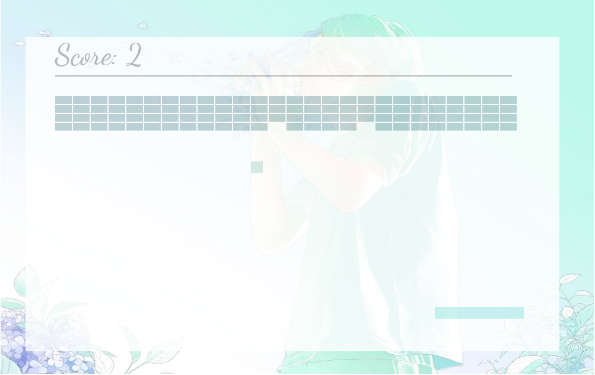
2016-2017学年夏季小学期JavaScript实验报告

1. **项目标题**

打砖块

1. **实验效果**

** **

****

后三张图的游戏难度是蓝<红<灰，难度递增体现在球的速度上。

1. **实现方法**

出于对将来大创项目需要用到js的考虑，我选择用javascript来写这个游戏。HTML和CSS的部分比较简单，主要就是div的分块、图片的插入、背景音乐的插入和砖块、木板、球的绘制。因为上述操作在HTML里都比较简单，在这里略过，下面将重点说明js的部分。

Js用到的函数比c要少很多，总共只写了六个，分别用于按id提取HTML内容、实现加载砖块、小球运动、判断是否击中砖块、判断如何改变方向及接收左右键的信号。

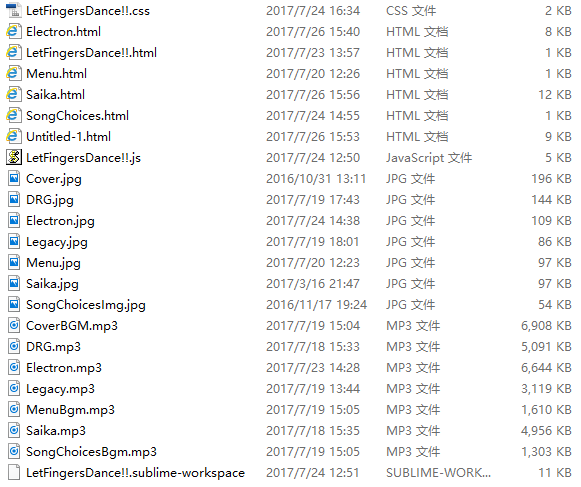
在绘制砖块时遇到了比较大的麻烦，因为此前没有接触过window.onload，不知道如何使用，所以研究了三四天，参考了网上的很多代码才最终实现，其中涉及到HTML的DOM树的问题，最终通过CreateElement和AppendChild实现。

之后出现问题的是计分的问题，发现虽然能显示分数的初始值0，但分数不会随着砖块被打掉而增加，本以为需要额外写一个函数，后来通过.innerHTML实现。

在砖块图案的绘制上，我觉得js不如c好用，可能是因为我没有掌握js中二维数组的正确用法，最后选择的是手动分块绘制，如上图中粉色的那一页，我是双重for循环中每一行只画需要的砖块。

最终没有实现菜单页面的制作，因为时间不够，同时因为我的几个页面写在不同的HTML文件中，也不知道如何由一个页面去控制其他页面，所以未能实现设置的功能。

1. **总结**

****

上图即为我本次实验用到的所有文件，感觉自己接触到的只是js当中非常小的一部分（甚至应不应该把这些关卡写在一个文件里以及怎么写），因为班里写js的同学非常少，所以我初期遇到的问题很多都没能在短时间内解决，现在得到的结果就是一个非常简单粗暴的产品（甚至没有像c一样设置属性不同的砖块）。由于js如今的应用范围极广，我今后也会继续学习如何使用js。