汇编大作业文档——《DX-balls》游戏开发

潘乐怡 2020012374 软件02

孙骜 2020012367 软件02

徐霈然 2020012355 软件01

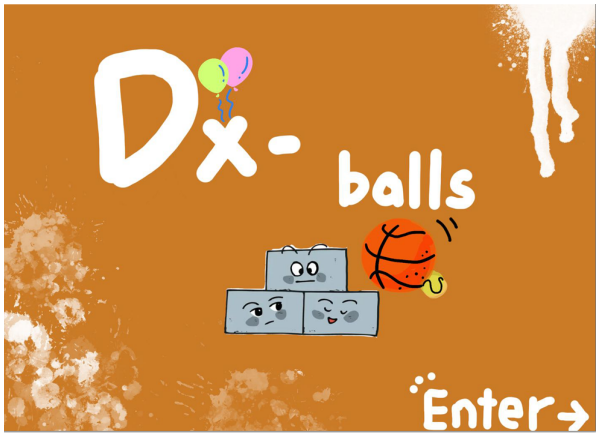
---

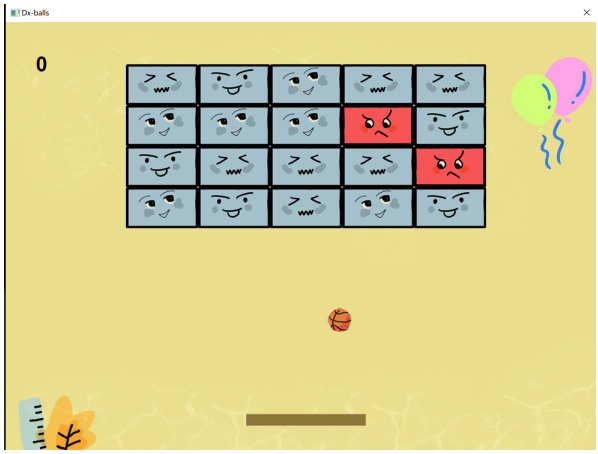
1. 游戏设计

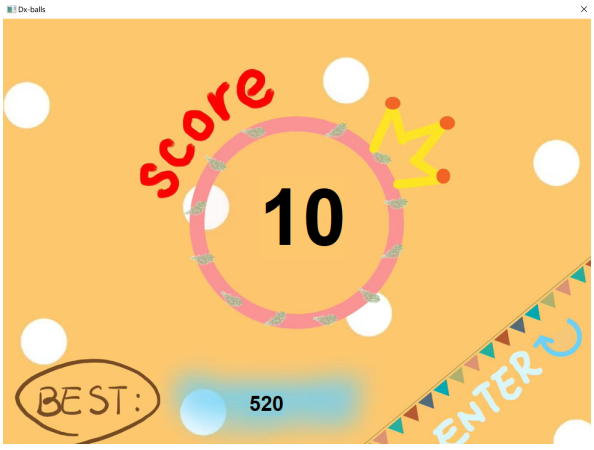
- 这款《DX-balls》小游戏是模仿经典的打砖块游戏进行开发的。整个游戏分为3个界面：（1）主页（2）游戏页面（3）积分界面，如下图所示。
- **主页**上展示了我们的游戏名称和主要的游戏元素：小球和表情各异的砖块。主页右下角的Enter指示玩家按下`Enter`键后可以进入游戏界面。主页采用橘色为底，生动的卡通图标配上轻快的背景音乐，以求对玩家有更大的吸引力。
- 玩家一进入**游戏界面**，就可以看到窗口最底端有一个挡板和一个小球，窗口的上端将会随机生成20个表情各异的砖块。红色底呈“愤怒”状的砖块是不可击碎的，青色砖块是能够被小球击碎的，击碎后可以+10分；且呈“痛苦”状表情的砖块在被击碎后还会释放隐藏技能，它有一定概率可以使小球运动速度变快，有一定概率使挡板变长一段时间，有一定概率使挡板速度变快，也有一定概率会使场上小球的数量+1。

  小球如果撞击到挡板，窗口的左、上、右边缘以及红色砖块，则会以弹性碰撞反弹；如果击中青色砖块，则青色砖块会碎掉，同时小球也会以弹性碰撞反弹。场上只要有一个小球从窗口下沿掉落，则游戏结束。若把所有的青色砖块打完，则游戏通关（自动结束），分数+200。

  游戏界面每隔一小段时间会在随机的x坐标上掉落一枚金币，如果用挡板接到金币则可以+10分。
- 游戏结束后会进到**积分页面**，积分页面会显示本局分数和历史最高分数。







2. 开发环境 & 运行说明

2.1 开发环境

- VS2012
- Irvine32
- MASM32

2.2 运行说明

（如果不想运行源码，则可以跳过这一段，直接运行exe文件夹里的.exe文件，注意开声音，有bgm）

- 新建一个解决方案和项目（或复制一个已有的项目）
- 将src文件夹里的所有内容复制到项目文件夹下（不是解决方案）
- 将 `mygame.inc` 中第15行masm路径修改为自己的路径
- 重新生成该项目并点击绿色三角运行

3. 功能点及其实现原理

3.1 整体架构

- 由于游戏是窗口程序，因此从整体上讲，我们采用汇编中的窗口类来进行开发
- 程序主体是 `WndProc` 函数
    - 当窗口创建时，调用 `startGame` 函数，`startGame` 函数主要分为三个部分，即：创建逻辑模块、创建绘画模块，以及播放背景音乐 `PlaySound`
    - 逻辑模块主要通过 `while` 来实现，我们设置了一个变量 `game_status` 来记录当前的游戏状态：当 `game_status=0` 即游戏处于主页时，循环执行Sleep函数；当 `game_status=1` 即游戏处于游戏页面时，首先生成20个随机砖块；接着，用while函数，每隔15ms计算新的小球位置、挡板速度、挡板位置、金币位置并保存，以及判断游戏是否结束。
    - 绘画模块每隔10ms将已经更新过位置的小球、挡板、金币等绘制在窗口上。

      绘画模块共分为三步：（1）加载图片：图片资源在.rc 文件中通过路径定义，并在 loadGameImage 函数中加载到变量中。（2）不停循环把整个页面设置成无效区域，InvalidateRect 会把 WM_PAINT 信号加入队列，等到 WM_PAINT 信号处于队列首位的时候调用updateScene函数，进行所有图片和物体的绘制。（3）位图绘制：SelectObject 函数把位图选择到兼容 DC 中，然后调用 Transparent 函 数把图片白色的部分变透明，进行显示。同一种 struct 不同种类型的物体进行分类绘制。

3.2 场景的切换和背景音乐播放

- 场景切换

  场景切换采用Enter键键盘响应配合`game_status`变量实现。

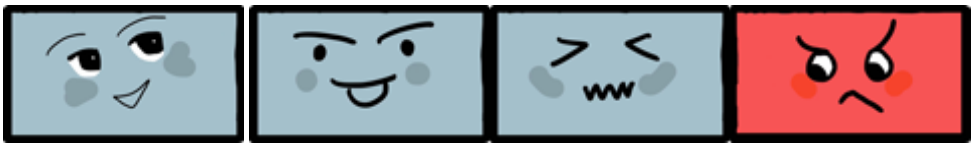
  - 当页面为主页时，即`game_status=0`时，按下Enter键，`game_status`将会切为1
  - 当页面为游戏界面时，即`game_status=1`时，按下Enter键，`game_status`将会切为2
  - 当页面为积分界面时，即`game_status=2`时，按下Enter键，`game_status`将会切为0
- 背景音乐播放

  采用`playSound`函数来播放背景音乐，SND_LOOP表示循环播放

```
invoke PlaySound, 155, hInstance, SND_RESOURCE or SND_ASYNC or SND_LOOP
```

  注：参数155为音乐路径的DEFINE编码

3.3 砖块的随机生成



- 我们的任务是：生成20个砖块，每个砖块从4个砖块类型中随机挑选出一个进行生成。
- 我们采用的随机生成算法：用`crf_time`函数获取当前时间作为随机种子生成第一个随机数，再用线性同余法生成一系列0-99的随机数。`getRandInEdx`函数如下：

```
getRandInEdx proc uses eax ebx,
    tempMod:DWORD
    ;线性同余法: a[n+1] = (a[n] * b + c) % p
    mov eax, randint
    mul rand_b
    add eax, rand_c
    mov edx, 0
    div rand_p
    mov randint, edx
    mov eax, randint
    mov edx, 0
    div tempMod
    ret
getRandInEdx endp
```

3.4 挡板的运动

- 响应按键消息
  1. 在`WndProc`函数中监听按键消息（`WM_KEYDOWN`），调用响应函数`processKeyDown`，将保存在`wParam`中的按键类型（`VK_LEFT`或`VK_RIGHT`）作为参数传给响应函数
  2. 在响应函数`processKeyDown`根据按键类型将全局标志变量`key_left`或`key_right`置1
  3. 在`WndProc`函数中监听松开按键消息（`WM_KEYUP`），调用响应函数`processKeyUp`，将保存在`wParam`中的按键类型（`VK_LEFT`或`VK_RIGHT`）作为参数传给响应函数
  4. 在响应函数`processKeyUp`根据按键类型将全局标志变量`key_left`或`key_right`清0
- 更新挡板速度

  在`logicThread`中循环调用更新挡板速度的函数，函数中根据全局标志变量`key_left`或`key_right`给挡板速度赋值（`key_left == 1`，挡板速度为负；`key_right == 1`，挡板速度为正；`key_left == key_right == 0`，挡板速度为0

- 更新挡板位置

  在`logicThread`中循环调用更新挡板位置的函数，将挡板的水平坐标加上速度的值（若到达窗口边界，不动）

3.5 小球的运动及其与边界、砖块的碰撞

- 定义小球结构体：

```
;球对象
ball struct
    pos             box <>              ;位置信息
    speed           point <>            ;速度
    exist           SDWORD 0            ;0表示不存在
ball ends
```

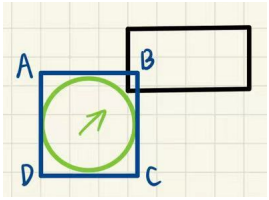
  其中pos包含：`pos.top`，`pos.bottom`，`pos.left`，`pos.right`；speed包含`speed.x`，`speed.y`；

- 在`initBall`函数中设定小球的初始速度和初始位置，每隔一个循环周期，使`pos.left += speed.x`，`pos.right += speed.x`，`pos.top += speed.y`，`pos.bottom += speed.y`来更新小球位置
- 小球与边界的碰撞：与上边界的碰撞只需要判断`pos.top`是否小于0，如果小于0则`neg speed.y`即可，与左边界和右边界的碰撞类似处理即可。
- 小球与砖块的碰撞：小球与砖块的碰撞是诸多功能点中较难实现的一个，实现思路如下：

  小球运动的方向分为四种：右上、右下、左上、左下，我们以右上为例分析可能发生的碰撞情况：

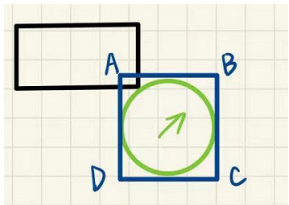
  1. 小球的右上角撞击到砖块

     我们画出小球的外接正方形，则第一种碰撞情况为B点在砖块内部

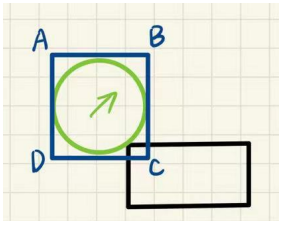
  2. 小球的左上角撞击到砖块

     第二种碰撞情况为A点在砖块内部

  3. 小球的右下角撞击到砖块

     第二种碰撞情况为C点在砖块内部

因此，对于小球速度为向右上的情况，我们需要逐个验证小球的左上角、右上角和右下角三个点是否在任一砖块内部，如果在，则认为它与该砖块发生了碰撞。此外，我们还需要判断小球到底撞击了砖块的哪个面，这个问题可以通过计算小球外界正方形在砖块内部的顶点与砖块两边的距离，比较即可。

3.6 金币的定时掉落 & 挡板接住金币得分

- 随机生成金币掉落

  1. 在`logicThread`中循环调用生成金币的函数，函数中首先将当前`game_counter`（相当于游戏的计时器）值与`last_coin`（上一次生成金币时`game_counter`值）相减，大于一定间隔才继续，否则直接返回
  2. 遍历金币的数组，找到一`exist == 0`的槽位回收利用初始化新的金币。调用随机数函数随机生成金币横坐标，纵坐标为0（屏幕顶端），金币速度向下，更新`last_coin`为当前`game_counter`值
  3. 在`logicThread`中循环调用更新金币位置的函数，将挡板的纵坐标加上速度的值模拟掉落，金币纵坐标大于窗口下界时将`exist`属性置为0

- 金币与挡板碰撞得分

  在更新金币位置的函数中判断条件`coin.exist == 1 && coin.pos.bottom >= pad.pos.top && coin.pos.right > pad.pos.left && coin.pos.left < pad.pos.right`（表示金币**还在场上**且与板子接触）为真则`exist`置0，加分。

3.7 触发技能（挡板变长、小球速度变快、挡板速度变快、多个小球）

- 挡板变长 & 持续时效

  设置一全局变量`longpad`，触发技能时置1，为1时挡板宽度增加、挡板右端点坐标增加、绘制函数`paintCoin`中加载长度变长的位图；持续时效：类似于`last_coin`，设置一全局变量`skill_timer`记录最近一次触发技能时`game_counter`的值，在`logic_thread`中每增加`game_counter`时判断`game_counter - skill_timer`大于时效间隔**且**`long_timer == 1`，则`long_timer`置0且还原挡板宽度

- 小球速度变快

  小球`x`，`y`速度分量的绝对值分别+1即可

- 挡板速度变快

  将`pad_speed` + 1即可

- 多个小球

  为了实现多球，我们需要改变存储小球的数据结构，由一个球对象转变为一个存储球对象的数组。在更新小球位置、判断碰撞等函数中需要遍历小球数组中的每个小球。

3.8 显示得分 & 保存历史最高纪录

- 显示得分页面

  采用我们手绘的插画作为背景图片，通过设置字体大小、颜色，字体背景颜色，文本框位置等参数，使其与背景图片中预留的位置相匹配，并且背景颜色不突兀。该页面会显示本局游戏所得分数、游戏历史最高分数，以及按“Enter”键重新开始游戏的提示文字。

- 保存历史最高纪录

  在一局游戏结束后（小球掉出屏幕或者所有可击碎砖块全部击碎后），会首先进入历史记录检测函数。该函数会读取保存历史最高分数的文件`highest_scores.txt`,在该文件中保存了历史最高分数。函数读取该分数后，会与当前分数进行比较，如果当前分数高于最高分数，则将当前分数写回文件，并覆盖掉原历史最高分数。该过程结束后，才会进入得分显示页面，保证显示的分数是正确的。

4. 难点和创新点

4.1 难点

- 显示图片

  由于之前的小作业中没有涉及GUI的相关知识，因此对于我们来说，如何搭建窗口程序的架构、如何显示图片是一个难点。首先，汇编语言只能加载.bmp格式的文件，且不能直接另存为.bmp，而是要用画图工具打开，再另存。此外，为了让图片更加美观，我们需要把图片的白边抠掉，但是用PS软件处理后的图片经画图3D工具处理之后透明的部分就显色了；后来我们找到的解决办法是，用画图软件打开（而不是画图3D），再保存就可以了。

- 随机生成各种砖块

  汇编语言不像其它高级语言，有比较成熟的生成随机数的函数；对于汇编语言，我们需要自己编写随机数生成算法。我们借鉴了比较经典的线性同余法来生成随机数，并用`crt_time`函数获取当前时间作为第一个随机种子，避免"伪随机"的出现。

- 小球与砖块的碰撞

  实现小球与砖块的丝滑碰撞挑战还是比较大的，它需要设计者充分考虑各种碰撞的可能性。最开始，我们对于向右上运动的小球只去判断它外界正方形右上顶点是否与在某个砖块内，导致出现了bug。后来经过仔细分析，向右上运动的小球需要判断它外界正方形右上、左上、右下三个点与砖块的碰撞才算全面。因此，小球与砖块的碰撞需要考虑4*3=12种情况，较为复杂。

4.2 创新点

- 主页背景、游戏界面、积分界面背景以及各表情砖块、小球、挡板**全部手绘**，游戏主视觉与背景音乐贴合，营造轻松快乐的氛围
- 相比传统的打砖块游戏，我们的游戏增加了**不同的砖块类型**，并增加了**技能触发**，使得游戏元素更加丰富，变数更多

5. 与中期相比的进展

- 挡板的运动（键盘响应）
- 小球的运动及其与边界、砖块的碰撞
- 金币的定时掉落 & 挡板接住金币得分
- 触发技能
  - 小球速度变快
  - 挡板变长
  - 挡板速度变快
  - 多个小球
- 积分界面以及历史最高分数

6. 小组分工

组员	分工
潘乐怡	设计整体代码结构； 主页背景、游戏页面背景、砖块、小球、挡板的绘制； 背景音乐嵌入； 小球与边界及砖块碰撞； 多个小球技能； 撰写报告； 录制视频；
孙骜	设计整体代码结构； 编写随机生成算法，随机生成20个砖块并贴图； 积分页面背景绘制 显示得分与保存历史最高纪录 撰写报告；
徐霈然	设计整体代码结构； `Enter`键跳转响应； 小球、挡板贴图； 挡板的运动功能； 触发小球速度变快、挡板变长技能； 金币的定时掉落 & 挡板接住金币得分； 撰写报告；