浙江大学

程 序 设 计 专 题

大 程 序 报 告

1. 学生姓名 ： 李向 学号： 3180106148

2. 学生姓名 ： 许昊 学号： 3180102084

3. 学生姓名 ： 苏敬宁 学号： 3180100157

2018~2019 春夏学期 2019 年 06 月 08 日

1、 题目描述和题目要求

Flappy Bird 游戏是一款鸟要尽量穿过越多的水管的游戏，游戏开始后不得暂停，鸟的移动由鼠标右键，键盘空格键以及键盘向上方向键决定，每按一次上升固定高度，否则则加速掉落。游戏目标让鸟通过越多的水管，每通过一个分数加一，一但碰到水管或地面游戏结束。游戏过程中会有状态栏显示当前得分。不同于最原始版的游戏，我们增加了难易度选项，三者的水管生成速度依次加快。其他功能有在设置中的引导指示，即最终的排行榜。在游戏中排名前十的玩家将会被列在排行榜中，输入姓名方式为姓名数入，最后添加一个'#'登入资料。排行榜的内容亦可以清空。

2、 需求分析

在开始制作游戏之前，我们无数次的挑战了正版的 Flappy Bird 游戏，经过数次讨论，列出了以下的需求，以进行分工。

在思路的部份列出以下四个大方向，包含图片素材(声音素材)、启动界面、游戏界面，及分数界面。在这些模块内，我们又继续细分列点，以大致掌握整体工作量。

图片素材中包含我们所有的角色(鸟、水管)及背景，然而最终金币级怪物功能未能实现。原先的设定是使金币随机分布，碰触到一个金币等同于经过一个水管的得分；而怪物原先的用途则是为了增加游戏难度，以上两功能都并非原版游戏拥有，而是当时我们自己添加的。

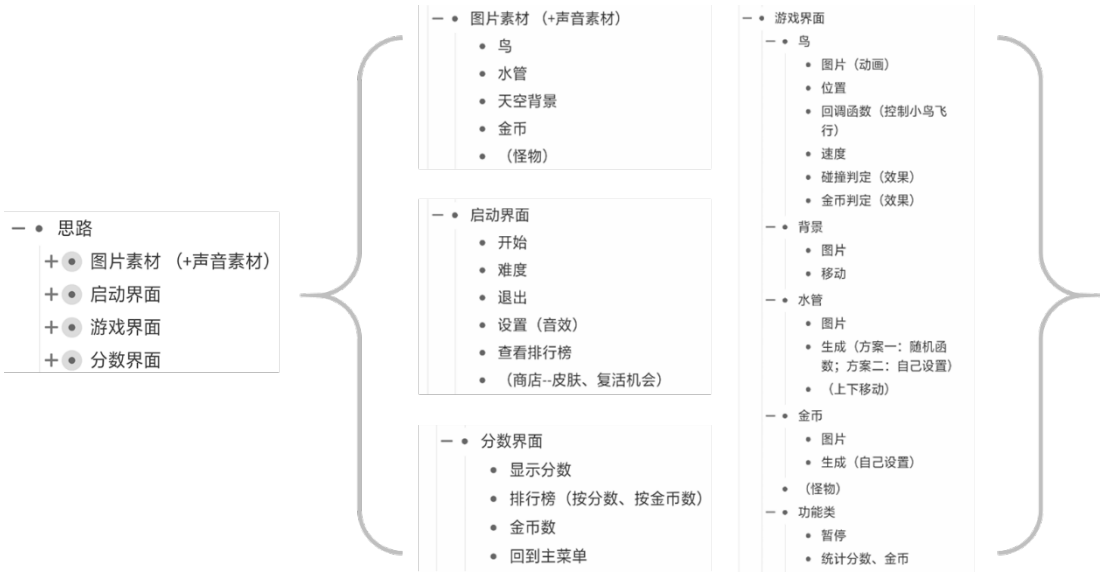
在启动界面的部份，除了参考原始界面，我们加入新的难易度功能，包含选

择难易度等等。我们希望可以透过难易度的不同，达到能够提高用户广度的功能。在讨论中的我们，还抱有极大的梦想，希望可以在添加商店的功能，让玩家可以自由选择鸟的皮肤，换上自己喜欢的颜色。

分数界面的部份则就希望能做出用链表实现的排行榜界面，包含玩家姓名分数及名次，比较没有太多的花样。

最后游戏界面的部份是一开始我们就知道将会最困难的部份，不只是界面的样式，还有许多功能函数的需求。我们罗列了以下，在对象方面包含绘制的函数、鸟的移动、背景图片(水管)的移动及随机生成，还有原先设有的金币及怪物的移动等等；在游戏进行方面则有碰撞的判断函数、控制小鸟飞行的回调函数、控制速度的计时器等等；功能方面更设置随时统计分数的状态栏以及最终显示玩家排名的排行榜等功能。

下图为初次讨论列出的大致需求分析。



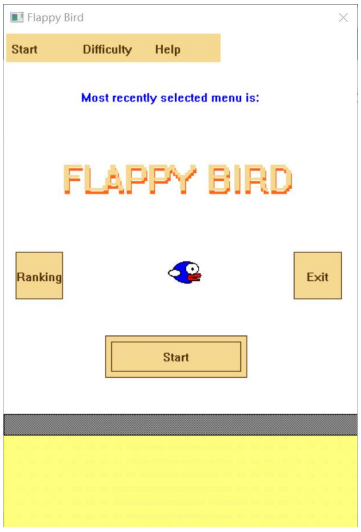
下图为游戏部份更细节的函数列表及初步分工。此笔记为第二次小组会议所讨论内容。

3、总体设计

在总体的设计上，我们选择参照原版的游戏，进行编写修正，再发展新功能。

3.1 功能模块设计

- **开始界面模块**



1. 菜单栏：开始、难度及帮助栏位，开始分为快速开始和结束，快速开始会直接展开难度为 EASY 的游戏，退出则关闭界面；难度栏可选择易中难三种难度分别开始游戏。
2. 按钮：Start、Ranking、Exit、Instruction。按下 Start 后会出现易中难三个选项开始游戏，点击 Ranking 后跳转至排行榜界面，点击 Exit 后则会退出，点击 Instruction 进入游戏说明界面。
3. 状态信息栏：页面上方显示最近一次选择的菜单按钮；
4. 实现过程：采用全局 bool 变量，当发生点击按钮事件时相应全局变量发生改变，再通过 if 判断从而实现页面之间的跳转，状态信息栏利用变量 selectedLable，在点击菜单按钮之后将相应的按钮上的字符串赋给 selectedLable，再将其输出。

- **结束页面模块**

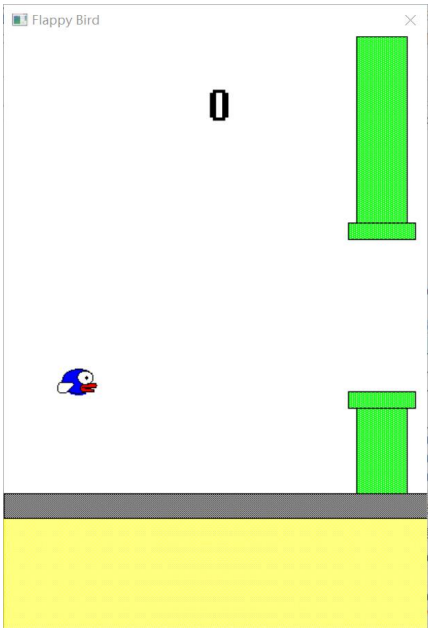
1. 按钮：Restart、Ranking、Exit。按下 Restart 后会初始化变量并重新回到开始菜单；点击 Ranking 后跳转至排行榜界面，点击 Exit 后则会退出。

2. 姓名输入：输入姓名后加上'#'登入姓名，前十名将记录在排行榜上。
3. 得分显示：显示游戏得分。
4. 实现过程：采用全局 bool 变量，当发生点击按钮事件时相应全局变量发生改变，再通过 if 判断从而实现页面之间的跳转。姓名输入使用 textbox 控件实现读取。得分显示利用 score 全局变量，并用 itoa（）函数将数字转换成字符后，用 DrawTextString 函数输出到结束界面上。

- **游戏主体——小鸟模块**

1. 小鸟图片绘制：绘制翅膀向上，翅膀在中间，翅膀向下三个状态的图片。并利用图片制作翅膀扇动的动画。
2. 小鸟运动控制：不操作时小鸟加速下落，进行指定操作时小鸟上升固定的距离，从而控制其通过水管空隙。另外，小鸟飞出界面上边界时，使其自由下落到界面内。
3. 小鸟碰撞控制：见游戏主体——小鸟模块
4. 实现过程：图片绘制利用 libgraphics 库内的绘图函数进行绘制；画的制作利用计时器函数 startTimer，通过计时器控制每隔一定时间绘制不同振翅状态的小鸟实现动画效果；自由下落通过设定小鸟下落的速度每次回调时都增加实现，每次点击上升一定距离通过点击之后将小鸟的速度赋值为一个固定值实现。

- **游戏主体——水管模块**



1. 静态水管绘制：以水管体的左边缘位置、下水管的高度为参数绘制静态水管图案，而水管的宽度、水管口的大小、上下水管间空档的大小为定值。
2. 水管的生成与移动：首先，我们定义了一个数组 Pipe 来存放每个水管的两个参数 cx、cy。画面中始终有三个水管，所以我们定义了三个 Pipe 数组来存放他们的位置参数。其中 cy 由随机函数获得，以保证水管高度的随机性。水管的生成与移动需要与定时器结合。每刷新一次，都将 cx 减少一定的值，再调用静态水管绘制函数，以达到水管左移的动画效果。当画面中第一个水管的 cx 减小到一定的值（即刚好从画面中消失）时，将原画面中第二、三个水管的参数值分别赋给第一、二个水管的参数，

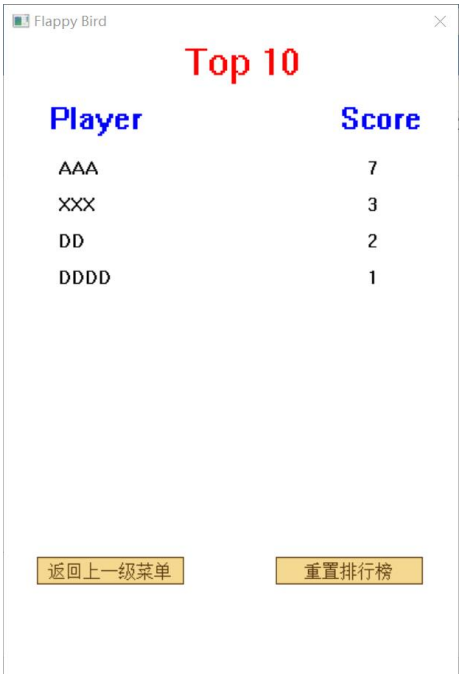
这样原二、三个水管便被认作为第一、二个水管。最后再为第三个水管赋新值。如此便可得到水管不断生成并移动的效果。

3. 碰撞的判断：首先通过比较小鸟的纵坐标与天空、地面的高度，来判断小鸟是否触地或飞出画面。其次判断小鸟是否与水管发生碰撞。小鸟通过水管大致可分为两种情形。其一，小鸟完全处于水管间。此时只需比较小鸟的纵坐标与上下水管头的高度即可；其二，小鸟部分处于水管之间。此时我们将小鸟近似为圆形，并比较圆心到水管顶角的距离是否大于圆的半径。如此便可大致实现对碰撞的判断。

- **游戏主体——背景模块**

1. 背景绘制：绘制游戏背景，只需要一个黄色的矩形和一个灰色的矩形。
2. 实现过程：利用 libgraphics 库内的绘图函数进行绘制即可。

- **排行榜模块**



1. 界面绘制：需要一个 TOP10 在最上面，然后左边是 Player，右边是 Score，最后将前十名的玩家数据依次显示，并实现一个返回之前界面的 button 和一个清除现有排行榜数据的 button；
2. 数据存取与处理：需要将当前玩家的姓名，分数数据存入文本文件中，再读取文本文件，并将排序好的内容依次显示在图形界面上。
3. 实现过程：界面的绘制使用 DrawTextString 函数即可，返回上级菜单的 button 利用 simpleGUI 的内容完成，清除现有排行榜数据的 button 利用 simpleGUI 的内容并以'w'的方式打开文本文件即可清除文件内容；用户的数据存取可定义 Player 结构进行保存，并利用与文件操作相关的函数进行读入，写入操作，数据的排序利用链表储存数据并进行排序，最后的显示也只需要用 DrawTextString 函数即可。

- **实时计分显示模块**

1. 需要在游戏界面上显示实时分数。
2. 实现过程：利用全局变量 score 存储分数，输出时利用 itoa 函数将数字转换为字符后就可以用 DrawTextString 函数进行输出了

3.2 数据结构设计

使用结构组织玩家的信息：

```
typedef struct {    //定义玩家结构
char name[100];
int score;
}Player;
```

使用链表处理排行榜玩家的信息

```
struct Toptenplayer {    //构造排行榜上玩家信息的链表
    char name[100];
    int score;
    struct Toptenplayer *next;
};
```

使用文本文件“ranking.txt”存储排行榜上的玩家信息

3.3 函数功能描述

void DrawMenu()

- 功能描述：游戏点开的首界面的具体绘制函数，若游戏结束时选择重新开始亦会使用这个函数。功能包含绘制界面，按键选择难易度、游戏开始、积分榜以及引导，状态信息栏等。
- 参数描述： 无
- 返回值描述：无
- 重要局部变量定义与用途描述：

A. 菜单栏的坐标、高度、宽度

```
    double fH = GetFontHeight();
    double h = fH*2;
    double x = WindowWidth/2.8;
    double y = WindowHeight/2-2*h;
    double w = WindowWidth/5;
    double wlist = TextStringWidth(menuListDifficulty[3])*1.2;
    double xindent = WindowHeight/20;
```

B. 用于选择菜单栏，记录所选对象，进而判断后进行下一步操作

```
    int selection;
```

C. 菜单栏中栏目与其子菜单栏内容及显示选择菜单栏时的操作状态

```
static char * menuListStart[] = {"Start", "Quick Start | Ctrl-Q", "Exit | Ctrl-E"};
static char * menuListDifficulty[] = {"Difficulty", "Easy | Ctrl-O", "Medium | Ctrl-M", "Hard | Ctrl-H"};
static char * menuListHelp[] = {"Help", "Instruction | Ctrl-I"};
static char * selectedLabel = NULL;
```

- 函数算法描述：使用 drawButton, drawMenuList 等 simpleGUI 功能。通过改变全局布尔变量：

  IsGameOn,IsGameOver,IsIntroducrtionOn,IsRankingOn,InputOver

  实现页面间的跳转。

- 其他描述：在开始和届数页面的部份使用 simple GUI，参照老师的范例进行研读及使用。我们将整个开始菜单编写成一个函数，drawMenu 每当要重新开始游戏前，就叫出这个函数。其中在 drawMenu 中运用到的还有 drawButton 及 drawMenuList 这两个现有的函数, 调用了 imgui.c 及 imgui.h。另外, 为了让画面看起来更加协调, 我们修改了 graphics.c 里面的现有颜色，自己使用了重新调配的颜色。整张画面的中间大大的 Flappy Bird 字样也使用了新调配的颜色，并且用两层字的打印营造阴影的效果，加强画面感

}

void DrawGameEnd()

- 功能描述：游戏结束界面的具体绘制函数，包含输入玩家姓名，重新开始，积分榜等的绘制。
- 参数描述： 无
- 返回值描述：无
- 重要局部变量定义与用途描述：

A. 菜单栏的坐标、高度、宽度

```
double fH = GetFontHeight();
double h = fH*2;
```

```
double x = WindowWidth/2.8;
double y = WindowHeight/2-2*h;
double w = WindowWidth/5;
```

B. 玩家姓名输入

```
static char memo[30]="\0";
```

- 函数算法描述：使用 drawButton, drawMenuList 等 simpleGUI 功能。通过改变全局布尔变量:

  IsGameOn,IsGameOver,IsIntroducrtionOn,IsRankingOn,InputOver

  实现页面间的跳转。姓名的输入使用了 simpleGUI 中的 textbox 控件。

- 其他描述: 结束页面的部份和开始页面相差不大, 一旦游戏判定 GAMEOVER 就会叫出这个函数，主要是移动了开始界面的按键位置，并加上了 GAMEOVER 字样、分数及姓名输入。姓名输入部份也参照了 imgui.c 里面的函数进行使用，再稍稍修改了颜色。

}

void Main()

- 功能描述： 设置窗口大小，设置窗口标题，初始化图形界面，注册所用的各回调函数，打开定时器，绘制前三根水管。
- 参数描述：无
- 返回值描述：无
- 重要局部变量定义与用途描述：局部 double 型变量 d 用于随机取得水管参数 cy 的值。D 作为中间值获得 cy 的小数部分。
- 函数算法描述：分别注册了字符、键盘、鼠标、定时器回调函数。共开启四个定时器, 分别对应菜单界面与三种难度下的游戏界面。绘制前三根水管时, 由于我们使用的随机函数 rand () 只可获得整数, 所以共使用两次 rand (),

一次作为 cy 的整数部分，一次强制转换为 double 型后处理为 cy 的小数部分。

void DrawBackground()

- 功能描述： 绘制游戏过程中的背景
- 参数描述：无
- 返回值描述：无
- 重要局部变量定义与用途描述：无
- 函数算法描述：简单的 libgraphics 图形库应用

void DrawBird(int state, double bx, double by)

- 功能描述： 绘制游戏过程中小鸟的三种状态
- 参数描述： state 是小鸟翅膀的状态，分为 WINGUP，WINGMID，WINGDOWN 三种，bx，by 分别为小鸟的横纵坐标。

```
#define   WINGUP          1
#define   WINGMID         2
#define   WINGDOWN        3
```

- 返回值描述：无
- 重要局部变量定义与用途描述：无
- 函数算法描述：简单的 libgraphics 图形库应用

void DrawPipe(double PipeLocation, double PipeHeight)

- 功能描述： 绘制静态水管
- 参数描述：PipeLocation 为水管体左边缘的横坐标，PipeHeight 为下水管的高度
- 返回值描述：无
- 重要局部变量定义与用途描述：无
- 函数算法描述：使用 MovePen（）、DrawLine（）、StartFilledRegion（）等函数，分别进行上下水管的填色与描边。

void DrawRanking()

- 功能描述： 绘制排行榜
- 参数描述：无
- 返回值描述：无
- 重要局部变量定义与用途描述：fH 为字体高度，h，w 分别为 button 控件的高和宽。

```
double fH = GetFontHeight();
double h = fH*2;
double w = WindowWidth/5;
```

- 函数算法描述：简单的 simpleGUI 应用。

void PrintRanking()

- 功能描述： 从文件中读取排行榜信息并显示

- 参数描述：无
- 返回值描述：无
- 重要局部变量定义与用途描述：fp 为文件指针用来处理存储排行榜信息的文本文件；s[20][30]为字符数组用来保存从文件中读入的信息；i，j 为循环变量。

```
FILE *fp;
char s[20][30];
int i,j;
```

- 函数算法描述：利用文件处理函数以及简单的循环来读取文件以及显示文件内的内容。

void DrawIntroduction()

- 功能描述： 展示游戏说明界面
- 参数描述：无
- 返回值描述：无
- 重要局部变量定义与用途描述：fH 为字体高度，x，y，h，w 分别为 Label 的横纵坐标以及高和宽。

```
double fH = GetFontHeight();
    double h = fH*2;
    double x = 0.0;
    double y = WindowHeight-2*h;
    double w = WindowWidth/5;
```

double m = TextStringWidth("(一)游戏说明 ");

- 函数算法描述：简单的 simpleGUI 应用。

void DrawScore()

- 功能描述： 实时显示分数
- 参数描述：无
- 返回值描述：无
- 重要局部变量定义与用途描述：无
- 函数算法描述：利用全局变量 str，储存 score 的值转换后的字符，再用 DrawTextString 进行输出。

void displaymenu()

- 功能描述：显示开始界面，并实现开始界面，排行榜，游戏说明之间的切换
- 参数描述：无
- 返回值描述：无
- 重要局部变量定义与用途描述：无
- 函数算法描述：首先利用 DisplayClear 清屏，然后根据 IsIntroducrtionOn 和 IsRankingOn 的不同值来决定执行 DrawMenu，DrawRanking，DrawIntroduction，中的哪一个从而实现开始界面，排行榜，游戏说明之间的切换。

void displaygame()

- 功能描述： 显示游戏界面，实现小鸟动画，死亡判断以及分数记录
- 参数描述：无
- 返回值描述：无
- 重要局部变量定义与用途描述：无
- 函数算法描述：首先利用 DisplayClear 清屏，然后调用 DrawBackground，DrawBird 等实现游戏界面的绘制，其中 DrawBird(RandomInteger(1,4),bx,by)利用随机函数实现小鸟三张图片的切换，从而在定时器的间隔执行下显示出扇动翅膀的效果。

void displaygameend()

- 功能描述： 显示游戏结束界面，并实现排行榜和结束界面之间的切换
- 参数描述：无
- 返回值描述：无
- 重要局部变量定义与用途描述：无
- 函数算法描述：首先利用 DisplayClear 清屏，根据 IsRankingOn 的值判断状态，分别调用 DrawRanking()或 DrawGameEnd()实现排行榜和结束界面之间的切换。

void CharEventProcess(char ch)

- 功能描述： 字符事件响应函数，获取字符输入，刷新显示。
- 参数描述：无
- 返回值描述：无

- 重要局部变量定义与用途描述：无
- 函数算法描述：无

void KeyboardEventProcess(int key, int event)

- 功能描述： 键盘事件响应函数，获取键盘，刷新显示。
- 参数描述：无
- 返回值描述：无
- 重要局部变量定义与用途描述：无
- 函数算法描述：检测到用户按下空格键或上方向键时使小鸟的速度 v=-0.17，以达到使小鸟向上飞一定距离的目的。

void MouseEventProcess(int x, int y, int button, int event)

- 功能描述： 鼠标事件响应函数，获取鼠标，刷新显示
- 参数描述：无
- 返回值描述：无
- 重要局部变量定义与用途描述：无
- 函数算法描述：检测到用户按下鼠标左键时使小鸟的速度 v=-0.17，以达到使小鸟向上飞一定距离的目的。

void TimerEventProcess(int timerID)

- 功能描述： 定时器响应函数
- 参数描述：timerID 为各定时器的 ID

- 返回值描述：无
- 重要局部变量定义与用途描述：无
- 函数算法描述：通过判断 TimerID 与判断一系列判断游戏状态的布尔变量，来调用显示不同界面的函数。

void readfile()

- 功能描述：读取文件内容，并将其加入链表中等待排序，之后进行排序，然后将排序好的数据存入文本文件中。（调用了 sortlinkedlist()和 putfile()）
- 参数描述：无
- 返回值描述：无
- 重要局部变量定义与用途描述：fp 为文件指针，用于文件处理，s[20][30]为字符数组，用于储存文件中读取的信息，i，j 是循环变量，

```
FILE *fp;                          //定义文件指针
char s[20][30];                    //定义读取文件信息后作为容器的数组
int i, j;                          //定义循环变量
```

- 函数算法描述：首先进行文件读取流程，然后利用简单的循环将读取的数据存储到数组中去，再利用循环将数组中的数据存储到链表中去，再调用 sortlinkedlist()和 putfile()对链表数据进行排序后，将链表内的数据输入的到文本文件中去。

void sortlinkedlist()

- 功能描述：对链表内容进行排序

- 参数描述：无
- 返回值描述：无
- 重要局部变量定义与用途描述：cur 和 curnext;分别为当前节点指针与下一个节点的指针，i 为循环变量。

  struct Toptenplayer *cur, *curnext;　　//当前节点指针与下一个节点的指针

  int i;
- 函数算法描述：利用插入排序法，遍历链表，找到最新玩家分数应处的位置，然后进行节点的插入从而实现分数数据的排序。

void putfile()

- 功能描述：将链表内容存储到文件中
- 参数描述：无
- 返回值描述：无

  重要局部变量定义与用途描述：f1 为文件指针用于文件读取关闭等相关操作，Top 作为排序过的链表的头结点，通过遍历它所指向的链表将其中的数据存储到文本文件当中去，num[10]数组用于存储分数（数字）转换而成的字符，i，j 为循环变量。

  FILE *fl;

  struct Toptenplayer *Top;

  char num[10];

int i, j;

- 函数算法描述：首先打开文本文件，然后利用 Top 作为排序过的链表的头结点，通过遍历它所指向的链表将其中的数据存储到文本文件当中去，从而实现将链表内容存储到文件中。

void initfile()

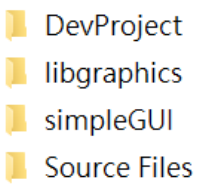
- 功能描述： 清空文件内容
- 参数描述：无
- 返回值描述：无
- 重要局部变量定义与用途描述：f2;是文件指针，用于文件相关操作。

  FILE *f2;
- 函数算法描述：f2 = fopen("ranking.txt", "w")，利用“w”方式打开已有的 ranking.txt 文件会直接清空文件内容，直接实现了目标。

3.4 程序文件结构

1）文件函数结构

首先，在大文件价下包含的是 DevProject，其中放的是我们的主函数，命名为 Our Flappy Bird。

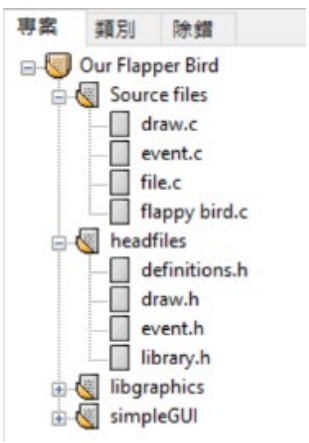


而在这在大文件下又分为四个子文件：

(1) Sourcefiles

i. draw.c

void DrawBird(int state, double bx, double by)
void DrawPipe(double PipeLocation, double



```
PipeHeight)
void DrawRanking()
void printsort()
void DrawIntroduction()
void DrawScore()
void DrawMenu()
void DrawGameEnd()
```

ii. event.c

```
void displaymenu()
void displaygame()
void displaygameend()
void CharEventProcess(char ch)
void KeyboardEventProcess(int key, int event)
void MouseEventProcess(int x, int y, int button, int event)
void TimerEventProcess(int timerID)
```

iii. file.c

```
void readfile()
void sortlinkedlist()
void putfile()
void initfile()
```

iv. flappy bird.c

```
void Main()
```

(2) Headfiles

i. definition.h /*此头文件定义一些全局用到的宏、变量、结构等*/

```
#define WindowWidth 5.0          //窗口宽
#define WindowHeight 7.0         //窗口高
#define BlankHeight 1.8          //上下水管之间空档的大小
#define GroundHeight 1.6         //地面高度
#define BlankHeight 1.8          //上下水管之间空档的大小
#define GroundHeight 1.6         //地面高度
#define Pipe_Speed 0.12

//定时器 ID 选择:
#define SHOW_GAME_EASY 1
#define SHOW_GAME_MEDIUM 2
#define SHOW_GAME_HARD 3
#define SHOW_MENU 4

//翅膀状态:
#define    WINGUP              1
```

```
#define    WINGMID          2
#define    WINGDOWN         3

//难度:
#define    Easy             1
#define    Medium           2
#define    Hard             3

static int enable_move_pipe=1;          //允许移动

//动画间隔时间（可用于调整游戏难度）:
static int timer_interval_easy=35;
static int timer_interval_medium=32;
static int timer_interval_hard=25;

static int timer_interval_menu=50;   //开始界面刷新间隔时间（防止界面闪烁）
static int       show_more_buttons1 = 0;
static int       show_more_buttons2 = 0;
static int       show_more_buttons3 = 0;

typedef struct{   //定义水管结构
    double cx;
    double cy;
} Pipe;
Pipe Pipe_1, Pipe_2, Pipe_3;

typedef struct {   //定义玩家结构
    char name[100];
    int score;
}Player;

struct Toptenplayer {   //构造排行榜上玩家信息的链表
    char name[100];
    int score;
    struct Toptenplayer *next;
};
```

ii. draw.h /*此头文件声明一些绘制界面过程中用到的函数*/

```
void DrawBird(int state, double bx, double by);
void DrawPipe(double PipeLocation, double PipeHeight);
void DrawMenu(void);
void DrawGameEnd(void);
void DrawBackground(void);
void DrawIntroduction(void);
```

```
void DrawScore(void);
void DrawRanking(void);
```

iii. event.h /*此头文件声明事件处理中需要用到的函数*/

```
void KeyboardEventProcess(int key, int event);
void MouseEventProcess(int x, int y, int button, int event);
void TimerEventProcess(int timerID);
void CharEventProcess(char c);
void DisplayClear(void);       //清屏函数
void startTimer(int id, int timeinterval);     //计时器启动函数
```

iv. library.h /*此头文件在于简化主函数使之不显冗杂*/

```
    #include "graphics.h"
    #include "extgraph.h"
    #include "genlib.h"
    #include "simpio.h"
    #include "conio.h"
    #include "imgui.h"
    #include "random.h"
    #include <stdio.h>
    #include <stdlib.h>
    #include <stddef.h>
    #include <math.h>
    #include <windows.h>
    #include <olectl.h>
    #include <mmsystem.h>
    #include <wingdi.h>
    #include <ole2.h>
    #include <ocidl.h>
#include <winuser.h>
```

(3) Libgraphics 与 simpleGUI
此二资料夹为老师提供的库。

2）多文件构成机制

使用 library.h 简化主函数使之不显冗杂,使用 definition.h 定义一些全局用到的宏、变量、结构等，使用 draw.h 声明一些绘制界面过程中用到的函数，使用 event.h 声明事件处理中需要用到的函数。
采用多个.c 文件，flappybird.c 是主源文件，另外组织了 event.c，draw.c，file.c，分别放置事件处理相关函数，绘制界面相关函数，以及文件处理相关函数，便于修改。

4.部署与运行

4.1 编译安装运行说明

1.下载压缩包，解压

2. 进入 DevProject 目录，最好使用最新版 Dev-C++5.8.3 打开 Our Flappy Bird 工程文件，编译运行后即可体验游戏（若无最新版 Dev-C++5.8.3，可打开 http://www.cad.zju.edu.cn/home/xgliu/clang/2018/自行下载安装）。

3.具体操作指南和目录介绍见解压后的“readme.txt”文件。

4.2 典型测试情况

（选取测试阶段典型的案例，说明如何设计测试数据，发现和定位错误的，测试结果可以含有屏幕截图。）

4.2.1 小鸟绘制。设置三个参数分别为 WINGMID、1、3.5，绘制小鸟图案。使用多组数据后确认小鸟绘制函数无问题。

4.2.2 水管绘制。采用多种长宽比，最后确定水管体宽 0.6，水管头宽 0.8。设置两个参数为 1、3，绘制单个水管图案。使用多组数据后确认水管绘制函数无问题。

4.2.3 随机获取水管参数 cy。取 cy=RandomReal（1.9，4.9），发现水管畸形。改变 RanDomReal 函数中的值，发现仍然畸形，而令 cy 等于一个常数例如 3，水管正常。由此推测问题出在 RandomReal（）函数没有取得我们想要的值。使用控制台与 printf 函数，输出多组 RandomReal（）函数取得的值，发现始终为一个九位数的定值，因此得出结论：问题就在于 RandomReal 函数。于是改用 rand（）函数。

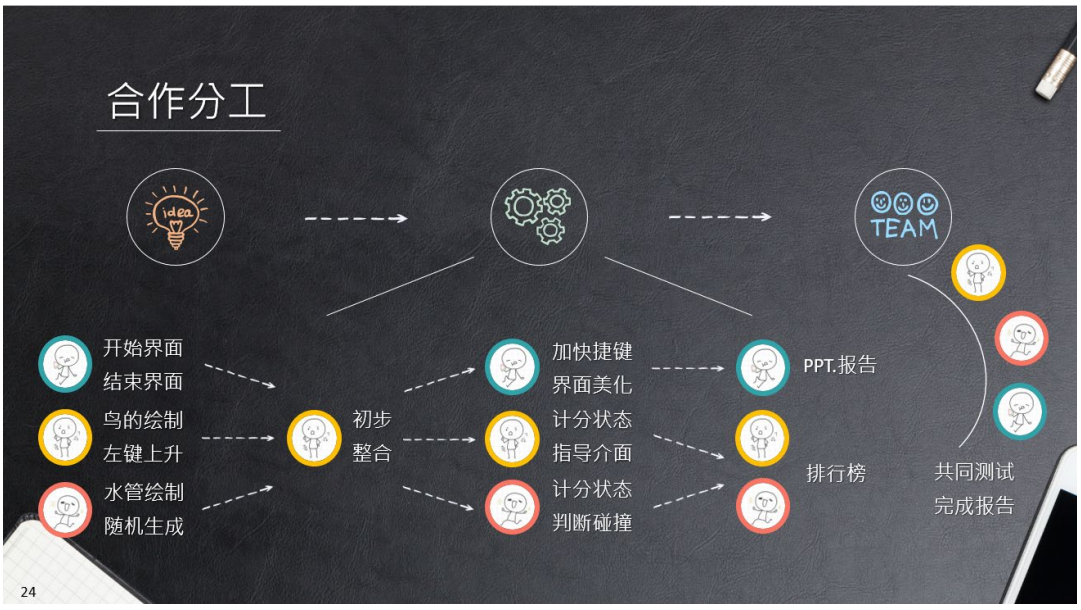
4.2.4 水管的持续生成。起初设擦除第一根水管的条件为 cx==-0.6，发现只能生成前三跟水管。于是设计生成水管的条件为 cx==-0.6，发现无水管生成。由此推测问题在于判断条件上。将判断条件改为 cx<=-0.6，发现有水管生成。问题解决。

4.2.5 画面闪烁。添加多个定时器后画面开始闪烁，由此推断问题在于画面刷新频率过高。将时间间隔调整为 35 以后，画面不再闪烁。因此推断正确，问题解决。

4.2.6 文件的读写。起初排行榜中始终无内容显示，为找出问题是在于文件的读还是写，我们在 txt 文档中输入内容再返回到游戏中，发现排行榜中有内容显示。由此断定问题在于文件的写入函数。

5.组内分工

5.1 组内分工情况



蓝圈小人：苏敬宁

黄圈小人：李向

红圈小人：许昊

5.2 个人实践过程中遇到的难点及解决方案

（分人员，多个点加（1）（3）…的标号）

李向：

（1） 难点：StartFilledRegion（）函数似乎只能填充不同深度的黑色，而我想要把小鸟的身体填充为彩色。

解决方案：进入 graphics.c 文件中查看 StartFilledRegion（）函数的完整定义，发现可以利用 SetPenColor（）函数实现填充颜色的变更。

（2） 难点：如何让小鸟动起来

解决方案：绘制三张不同状态的小鸟图片，利用定时器函数每隔一段时间绘制不同状态的小鸟并擦除之前的图片从而实现动画效果。

（3） 难点：连续点击鼠标左键小鸟上升的会非常快，极其影响游戏体验。

解决方案：之前连续点击鼠标左键小鸟上升的会非常快是因为鼠标回调函数中设定了每点击一次鼠标左键小鸟向上的速度都会增加一定的值，研究后改为每点击一次鼠标左键小鸟向上的速度都变为一个固定值，这样就实现了每次点击上升固定的距离，从而极大提升了游戏体验。

(4) 难点：想要在游戏界面上输出实时分数，存取实时分数的变量 score 是 int 型，没有合适的函数可用于直接将其显示在界面上，且无法调整大小和颜色。

解决方案：在助教老师的点拨下，使用 itoa（）函数将数字转换为字符之后，就可以用 SetPointSize () 调整字体大小，并用 DrawTextString（）函数直接在游戏界面上显示出来了。

(5) 难点：最后的小组讨论中我们提出想要实现一个清空已有排行榜数据的功能，但是觉得很难没有想出很好的办法。

解决方案：重新翻阅了一下大一上的 C 语言教材，突然想到用如果用“w”方式打开已有的文本文件会先清空原有的文本内容，于是十分简洁地实现了这个功能。

许昊：

(1) 对大作业感到无从下手。解决方案：首先小组讨论，将大问题肢解，并理出清晰的思路，定制具体的目标。其次通过老师提供的 ppt、democode 等学习回调函数、simpleGUI 等的具体用法。最后，从最基本的绘制图案做起，由浅入深。

(2) 难以做到水管的不断生成。解决方案：首先定义一个数组 Pipe 来存放每个水管的两个参数 cx、cy。画面中始终有三个水管，所以我们定义了三个 Pipe 数组来存放他们的位置参数。其中 cy 由随机函数获得，以保证水管高度的随机性。水管的生成与移动需要与定时器结合。每刷新一次，都将

cx 减少一定的值，再调用静态水管绘制函数，以达到水管左移的动画效果。当画面中第一个水管的 cx 减小到一定的值（即刚好从画面中消失）时，将原画面中第二、三个水管的参数值分别赋给第一、二个水管的参数，这样原二、三个水管便被认作为第一、二个水管。最后再为第三个水管赋新值。如此便可得到水管不断生成并移动的效果。

（3） random 库中随机获取实数的 RandomReal（）函数使用异常。解决方案：我们可以用的随机函数只有生成的 rand（）。所以我共使用两次 rand（），一次取得的数据作模除后作为 cy 的整数部分，一次强制转换为 double 型后再加上适当的值作为 cy 的小数部分。这样即可取得符合范围的随机 double 值。

（4） 由粗心导致的各种拼写、语句顺序错落等错误。解决方案：利用添加断点功能，找出错误所在；将指定语句改为注释，以准确定位错误所在；合理使用控制台，用 printf（）函数输出某变量的值，以判断代码是否编写正确。

苏敬甯：

(1) 问题：按钮在编写完成后点下，清屏却不完全。

解决方案：除了通过 display 函数，进行清屏，和重新绘制以外，调换绘制的顺序。

(2) 问题：按钮的位置在不同电脑看会有些许误差。

解决方案：并没有找到太好的解决方式，但每次有误差都作出修正，现在几乎达到一个即使偏差也不会相差太明显的平衡。

(3) 问题：画面及颜色偏单调。

解决方案：多次修改界面，完善功能，增加调配新的颜色。

(4) 问题：电脑编码不同，档案互传出现乱码。

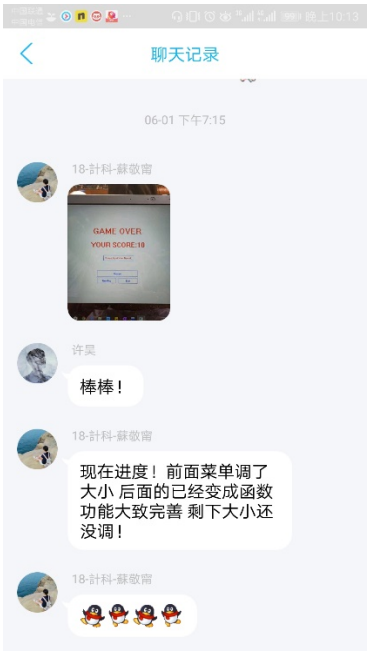
解决方案：尽量打英文或是截图看注解。

6.合作纪要

小组会议记录见另一个 word 文档“26 组-小组讨论记录表”。

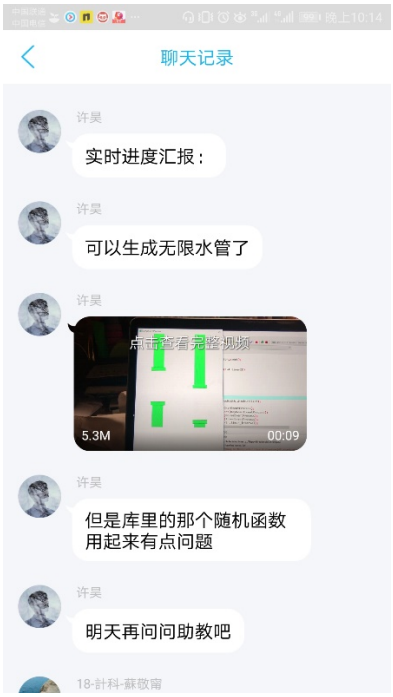
附截图：


聊天记录
我试试把你的包含进我的工程里面去
18-計科-蘇敬甯
豪
06-01 上午10:52
李向
那个按钮是不是太大了。。
李向
我之前的界面很小的
李向
My Flapper Bird.zip
2.05MB


聊天记录
06-01 下午7:15
18-計科-蘇敬甯
GAME OVER
YOUR SCORE:10
许昊
棒棒！
18-計科-蘇敬甯
现在进度！前面菜单调了大小 后面的已经变成函数功能大致完善 剩下大小还没调！
18-計科-蘇敬甯


聊天记录
哦对了问一下
许昊
startfilledregion括号里的数字是用来干嘛的呀
李向
从零到一
李向
拉割灰度
李向
1是纯黑
李向
0是纯白
许昊
好哒


聊天记录
太厉害了！
星期一 下午8:01
许昊
那我接着做水管的生成和移动啦
星期一 下午8:33
李向
吼的
18-計科-蘇敬甯
那那那我接着先做甚么呢
18-計科-蘇敬甯
还是我先大致理一个ppt出来


聊天记录
许昊
实时进度汇报：
许昊
可以生成无限水管了
许昊
点击查看完整视频
5.3M
00:09
许昊
但是库里的那个随机函数用起来有点问题
许昊
明天再问问助教吧
18-計科-蘇敬甯


聊天记录
李向
还有我发现
李向
只要刷新够慢
李向
屏幕就不闪
星期三 上午11:05
李向
我把定时器间隔调成35之后就不闪了
李向
点击查看完整视频


聊天记录
嗯嗯
李向
对于我们这个比例这个排版蛮好
18-計科-蘇敬甯
18-計科-蘇敬甯
有点这样的感觉
18-計科-蘇敬甯
我本来想说等功能都好了之后把鸟放在中间那块空白


聊天记录
快捷键加开始，退出，查看排行榜，鸟变换颜色，音效开关
李向
这五个
李向
鸟变换颜色你只要把manulist写好
18-計科-蘇敬甯
都是用menulist 麻
李向
具体变换我来写
李向
嗯
18-計科-蘇敬甯

7.总结

7.1 程序亮点或创新之处

a. 通过随机函数使小鸟扇动翅膀

b. 玩家可设置游戏难度（速度发生改变）

c. 以#键结束输入 ID

d. ID 输入后输入框显示输入完成

e. 排行榜可重置

7.2 应用知识点

a. 外部函数引用

b. 四大回调函数（人机交互）

c. 循环结构

d. 分支结构

e. 字符串操作

f. 结构

g. 链表

h. 多文件组合

i. 文件操作

7.3 心得体会

（总结团队在项目过程中的体会和心得，经验和教训，以及团队所有成员的自我评价。每位成员自我评价字数在 200-400 字之间。）

总体描述：小组的所有成员都积极地参与到本次大作业之中，认真负责地完成了自己的那份任务，并最终做出了我们都还满意的，属于我们的趣味小游戏“Flappy Bird”，这份成就感是我们每一位小组成员永远的财富。同时在作业完成过程中，我们增进了彼此之间的理解和感情，也发现了自己的不足之处，在今后的日子里我们一定会好好努力，改正不足，创造更多!

李向：在这次大作业的完成过程中，我最大的一个体会就是，痛并快乐着。痛是因为经常为了一些莫名其妙的 BUG 而改代码改到深夜，冥思苦想，身体是遭受了痛苦的，然而，这一切所谓的痛苦都在我们最终完善了所有功能，并且自己玩着自己写的游戏时烟消云散了，这最后获得的极大的成就感使我们快乐着，这次大作业使得我更加坚信选择计科是我做过的最正确的选择之

一。此外我还要感谢所有小组成员的辛勤付出，作为组长我总是给他们安排各种各样的任务，但他们都能够完美的完成，小组讨论也从来没有人缺席，讨论的气氛也很活跃，那些个深夜大家互相鼓劲的时刻，我一定会牢记在心。另外也要感谢助教的点拨，帮我们解决了不少困难。教训就是不要犯拖延症，我们这次大概是在 DDL 的十天前才真正开始的，其实时间上还是有点紧的，要是早一点开始，很多事情就能够做的从容许多。

对自己的评价的话，我觉得自己作为我们这个大作业的组长还是很负责的，我负责组织组员们讨论，负责在进度停缓的时刻督促他们，我自己也负责了小鸟的相关函数以及排行榜相关函数的编写，最终能够按时完成大作业，让每位组员都享受到成就感，我觉得我这个组长做的还不赖。同时在做大作业的过程中我也发现了自己以前在链表，文件等基础操作上面掌握的还很一般，所有很是需要进一步学习。

许昊：本次大作业是我第一次初步体验了团队开发程序的全过程。从中我掌握了打开文件、使用回调函数等许多专业知识，了解了开发一个大程序时的思路历程与步骤，学习了团队合作的要点与技巧，最重要的是体验到了团队合作的精神与程序开发的魅力。虽然代码报错、屡改屡败的情况让人崩溃，虽然时常深夜爆肝让人疲惫，但这重重阻力，每次都会被取得的成就冲刷干净。哪怕只是小小的进展，也足以使我的成就感爆棚一整天。更让我感到庆幸的，是能够结识两位优秀的组员。在此衷心感谢两位组员为我带来的陪伴与成长。

自我评价：首先对自己能够解决部分问题表示肯定。但是我在完成大作

业的过程中不够积极，时常拖延，需要组员的提醒才会增加对大作业的重视。除此，我们组中组长的工作量比我大出很多，一来是因为我能力不足，部分我不能解决的问题需要组长协助；二来我没有积极帮组长分担任务。这是我在团队合作中所欠缺的品质。

苏敬宁：一开始开会的时侯，我们三人有些面面相觑，面对这项大作业我们感觉无从下。陆陆续续讨论了两三次，各自询问了很多资讯，我们才慢慢开步上正轨，列出一些思路图进行分工。

制作过程中，其实我们都不是本来就会的人，而是慢慢的去学会，我们分头去研究不同部份，其中我主要负责的是界面按键选单等。我想整个过程中我最喜欢的部份就是调配版面整体营造的颜色，我在调色盘上搜索，也运用网络上的配色工具，试图找到最和谐的颜色，然后再进行 graphics.c 里面的颜色修改，完成最后的画面。

这次的大作业中，我除了学会自学写代码，更学会去读别人的代码。从他们的代码我也看到自己的弱点，因为我没有他们那样勇敢尝试的心，我应该要更勇敢不怕挫折，再多帮他们分担一些负担。

总而言之，我非常感谢我的组长和另一位组员，他们优秀又善良，还记得一开始见面的时候他们一直谦虚的说自己不是大佬，但他们在这短短的时间内学会了那么多，做出了这么多，实在让我感到无比敬佩！他们俩因为住在一栋楼，讨论起来比较方便，所以最难的部份主要都是他们完成的，真的辛苦他们了！我在这次的任务中负责的部份都没有他们困难，但我也都非常尽力的做到更好，PPT 也尽量做得更好，也更早完成，不拖到他们的进度。我喜欢询问他们的意见，因为希望做出来的东西真的属于我们三个！

8、参考文献和资料

（列出参考的书籍、论文、网站的信息和地址等。）