题目：贪吃蛇游戏的设计与实现（难度：*****）

描述：贪吃蛇游戏是一款特别流行的小游戏，深受人们喜爱，已经出现手机、平板、电脑等多个不同平台的版本。贪吃蛇游戏规则比较简单，具体为：一条蛇出现在封闭的空间中，同时此空间里会随机出现一个食物，通过键盘的上下左右方向键来控制蛇的前进方向。蛇头撞到食物，则食物消失，表示被蛇吃掉了。蛇身增加一节，累计得分，接着又出现食物，等待蛇来吃。如果蛇在前进过程中，撞到墙或自己的身体，那么游戏结束。

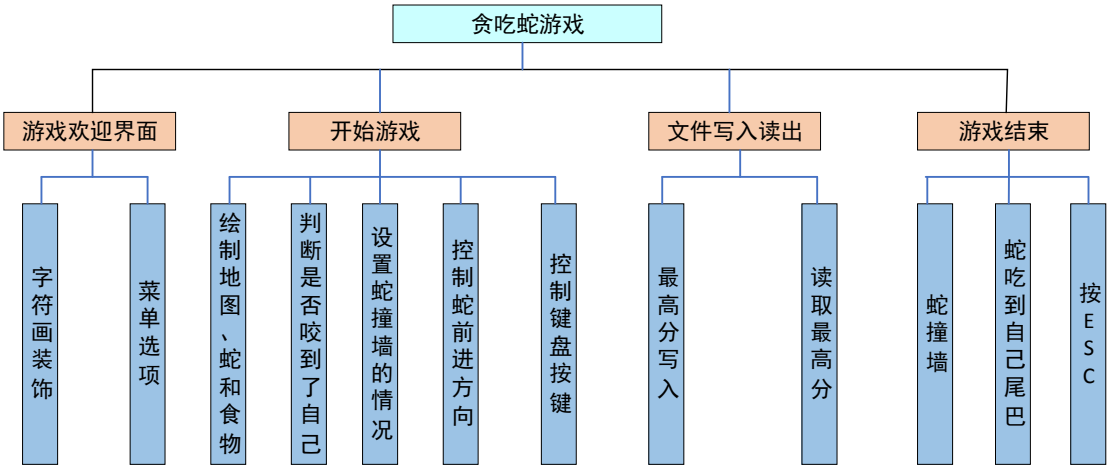
可使用 Visual stdio 来开发本游戏。

基本要求：

1. 设计字符画装饰；
2. 绘制游戏地图；
3. 从文件中读取最高分；
4. 绘制贪吃蛇；
5. 设计不按键时，蛇自动前进；
6. 设计键盘按键控制蛇的前进方向。

功能需求：

系统功能结构：贪吃蛇游戏共分为 4 个界面，分别是游戏欢迎界面、游戏主窗体、游戏说明界面和游戏结束界面。具体如下图所示：



1. 游戏欢迎界面

1.1. 蛇的字符画

在程序中通过设置控制台文字的颜色和获得控制台坐标，绘制技巧：打印时从上至下，从左至右，算好空行和空格的数量，根据自己喜好，自定搭配颜色。

Void printsnake();//main 函数入口，绘制字符蛇

1.2. 字符蛇下方的菜单选择

自行设计选择菜单，如：‘1’表示进入游戏，‘2’表示显示最高分，‘3’表示退出游戏等，根据需要自行设计菜单。
Void welcometogame();//main 函数入口

2. 游戏主窗体设计与实现

在游戏主窗体中可以玩贪吃蛇的游戏。需要完成两部分：游戏地图，及得分信息和按键小信息。设计思路：首先绘制游戏地图，然后打印得分信息和按键小提示，最后添加蛇和食物。

2.1. 创建游戏地图

可通过双层 for 循环打印空心方块的形式来获得地图。

Void createMap();

提示：在欢迎界面按‘1’之后，进入游戏主窗体，显示游戏地图，需修改 welcometogame（）中的代码，加入 createMap()的调用。

2.2. 绘制得分和小提示

最高得分可从文件中读取出来，所以首先应该调用读取文件函数读出最高分，打印输出就可以了。

Void scoreandtips();

该函数应该在按键控制的函数中调用，每次按键之后蛇如果吃到食物，“得分”的分数才会随时变化。

2.3. 从文件中读取游戏最高分

在游戏主窗体中显示了最高分，这个最高分需要从文件中读取。

Void File_out();

在 main()函数中添加调用 File_out 方法的语句。

2.4. 绘制游戏中的蛇身体

可用一定数量的特殊符号来 绘制蛇身，比如 5 个黄色的五角星。

提示：在设计蛇身体时，首先定义蛇尾，设置蛇尾的初始位置，坐标为（24,5），然后设置蛇头，蛇头坐标为（24+2*I,5），i 的值在 1~4，i 等于 1 时，蛇头坐标为(26,5),i=2 时,坐标(26,5)位置的蛇头变成蛇身,蛇头的坐标变成(26,5)，如此设计蛇头至蛇尾的初始位置。I 为循环变量。

Void initsnake();

修改 welcometogame()方法的代码，加入 initsnake()方法的调用，在欢迎界面按数字‘1’之后，进入游戏主窗体，显示游戏地图和黄色的贪吃蛇。

2.5. 创建并随机出现食物

在绘制蛇身之后，开始绘制食物。在本游戏中，食物是随机出现的，但是这个随机也是有限制的。食物智能出现在地图网格内，不能出现在网格线上，同时食物也不能和蛇身重合。同时满足这些条件后，食物使用某种颜色的符号来表示。

Void createfood();

提示：修改 welcometogame()方法中代码，加入 createfood()方法的调用，在开始界面选择数字键‘1’智慧，创建游戏地图，初始化蛇身体并显示食物。

3. 游戏逻辑

导致游戏失败的因素有两个：咬到自己和撞到墙。所以需要判断和解决的问题有以下几点，分别为：

（1） 判断蛇是否咬到了自己；

（2） 判断蛇是否撞到了墙；

（3） 不按键时，设置蛇的前进方向；

（4） 通过键盘按键控制蛇的前进方向。

提示：在设置通过按键控制蛇的前进方向时，使用系统接口函数 GetAsyncKeyState()方法。该方法用于确定，用户是否按下了键盘上的一个按键，可以实现监听键盘按键，并作出对应操作。

函数原型为：short GetAsyncKeyState(int nVirtKey)

参数 nVirtKey 为虚拟键盘值常量，常用的虚拟键盘值常量可以自行查询到。

3.1 判断蛇是否咬到自己

当蛇头和蛇身围成了一个圈的时候，就说明蛇咬到了自己。

提示：在代码中的表示咬到自己,只要蛇头的坐标值和蛇身上的任意坐标值重合，那么就判定为蛇咬到了自己。

Int biteself();

3.2 判断蛇是否撞到墙

当蛇头触碰到游戏边框也就是墙的时候，就是蛇撞到了墙。游戏中，墙的长宽已经在创建地图函数 createMap（）中设定好，长的坐标范围 0~x， 宽的坐标范围是 0~y,蛇头（head）的 x 坐标为 0 或者 x,y 坐标为 0 或者 y 时，水明蛇头坐标与游戏地图边界坐标重合，则判定为蛇撞到了墙。一旦蛇撞到了墙，则游戏失败，进入失败界面。

Void cantcrosswall();

3.3 设置蛇加速前进

在两种情况下蛇会加速前进,分别为蛇吃到了食物和 F1 键,加速时,时间间隔 sleeptime 减为 10，得分比提速前多 2 分。如果时间间隔 sleeptime 为 320,那么每次吃到食物的得分会变为 2.防止减到 1 之后再加回来有错。

Void speedup();

3.4 设置蛇减速前进

当按 F2 按键时，蛇会减速前进。减速时，时间间隔 sleeptime 加 30，得分比减速前少 2 分，如果时间间隔 sleeptime 为 350,那么每次吃到食物的得分分为 1，如果不设置，那么蛇吃到食物的得分会变成 0，得保证蛇吃到食物得分，所以设置加分为 1.

Void speeddown();

3.5 不按键时设置蛇的前进方向

贪吃蛇游戏有一个不同于一般游戏的最大特点，就是在不进行按键操作的时候，蛇是会一直移动的。也就是会按照原本的前进方向一直前进。

蛇在上、下、左、右 4 个方向上移动的时候，会遇到两种情况，一种是在前进的道路上

有食物，另一种是没有食物，如果吃到了食物，那么蛇会自动加速；如果没有吃到食物，蛇会继续前进。

Void snakemove();

3.6 通过键盘按键控制蛇的方向

当蛇在原本的方向上前进时，可以通过方向键来控制蛇的前进方向。蛇头只能转向左右，不能转向与前进方向相反的方向。比如，原本蛇是向上前进的，这时可以按左右方向键，但是不能按向下的方向键。

Void keyboardControl();

需要在 main()函数中添加调用 keyboardControl（）方法的语句。

4. 游戏失败界面

以下 3 种情况会进入失败界面，分别为：1）蛇头撞到地图边界；

2）蛇头触碰到自己身体，也就是咬到自己；3）游戏时按 ESC 键。

在失败界面中记录了最高分，如果没有超过最高分，那么失败界面上会显示“继续努力吧~你离最高分还差：”,后面显示距离最高分还差多少分；如果得分高于最高分，会在失败界面上显示“创纪录啦！最高分被你刷新啦，真棒!!!”

在界面下方，设置分支选项，1：重玩，2：退出游戏。输入各自的序号即可实现各自的功能。但是如果输入的数字不是 1 或者 2,那么提示输入错误，重新选择。

如果玩出了最高分，最高分会被写进文件中，假如名称为 save.txt，把原先的得分替换掉。Save.txt 文件被创建在源代码的目录中。

提示：向文件中写入数据的步骤为：首先使用 fopen()方法打开文件，如果要打开的文件不存在，那么创建此文件，然后通过 fprintf()方法把数据写入文件中，最后使用 fclose（）方法关闭文件。

5. 游戏说明界面

在游戏欢迎界面输入数字‘2’,即可进入到游戏说明界面，在此界面中显示了游戏的详细说明。

Void explation();

上述为贪吃蛇游戏的 5 个模块。可根据模块化逐步添加功能，直到最终实现该功能。