

电子科技大学 计算机 学院

实 验 报 告

课程名称 C++程序设计

实验题目 精灵游戏

难度级别 4

提交时间 2020. 12. 26

姓 名 韩博宇

学 号 2019040708023

1 软件说明

1.1 游戏使用说明

1. 打开方式：打开 SpriteGame 文件夹中的 SpriteGame.exe 文件或通过 1.2 源码编译说明中的步骤打开。

2. 基本功能：

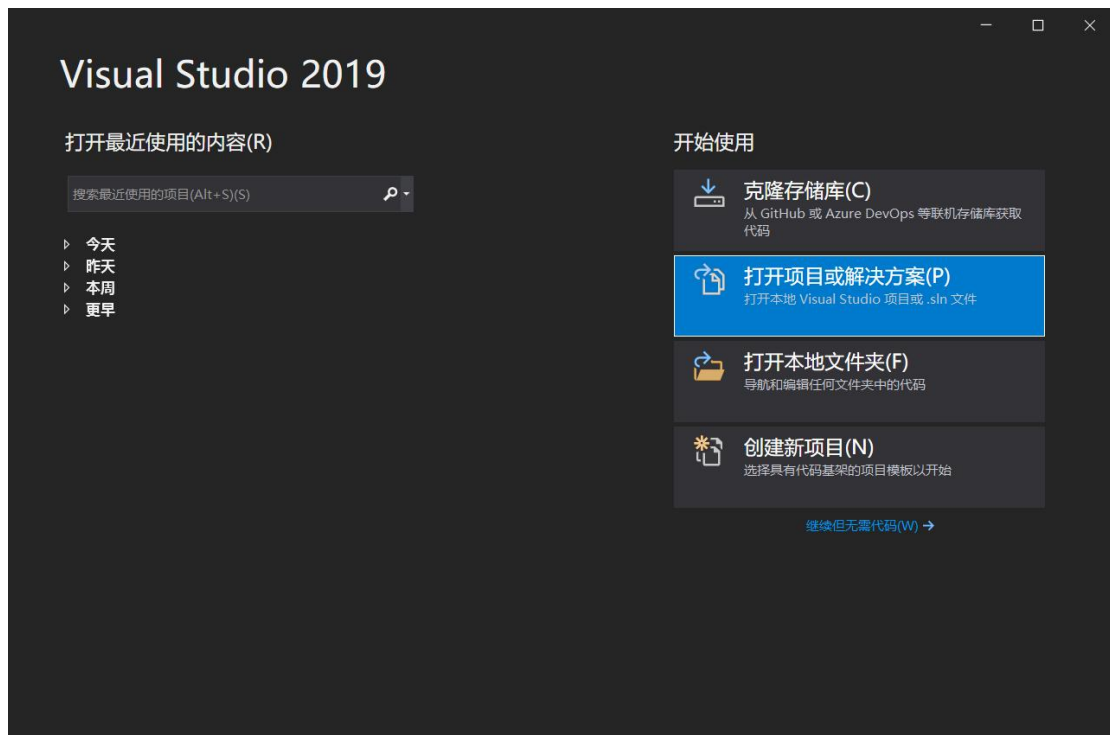
- ①显示游戏规则界面，通过按任意键开始游戏；
- ②在进行游戏时，可通过按空格键暂停/继续游戏；
- ③老鼠碰到墙壁会反弹，但不可以穿过墙壁，在小距离内碰到猫会反向逃跑；
- ④鸭子可以穿过墙壁，并在较大距离内碰到猫会更加智能的逃跑；
- ⑤狗可以穿过墙壁，并在较大距离内碰到猫会追猫；
- ⑥爱心可以穿过墙壁，并在较大距离内碰到猫会更加智能的逃跑；
- ⑦每抓到一只老鼠得 1 分，每抓到一直鸭子得 5 分；被狗抓到生命值减 1，吃到爱心生命值加 1；猫初始有 3 条生命值；
- ⑧分数达到 100 分游戏胜利，生命值用尽游戏失败。

3. 使用方法：通过按键盘的上下左右键控制猫移动抓精灵，根据“基本功能”中的规则判定游戏输赢。

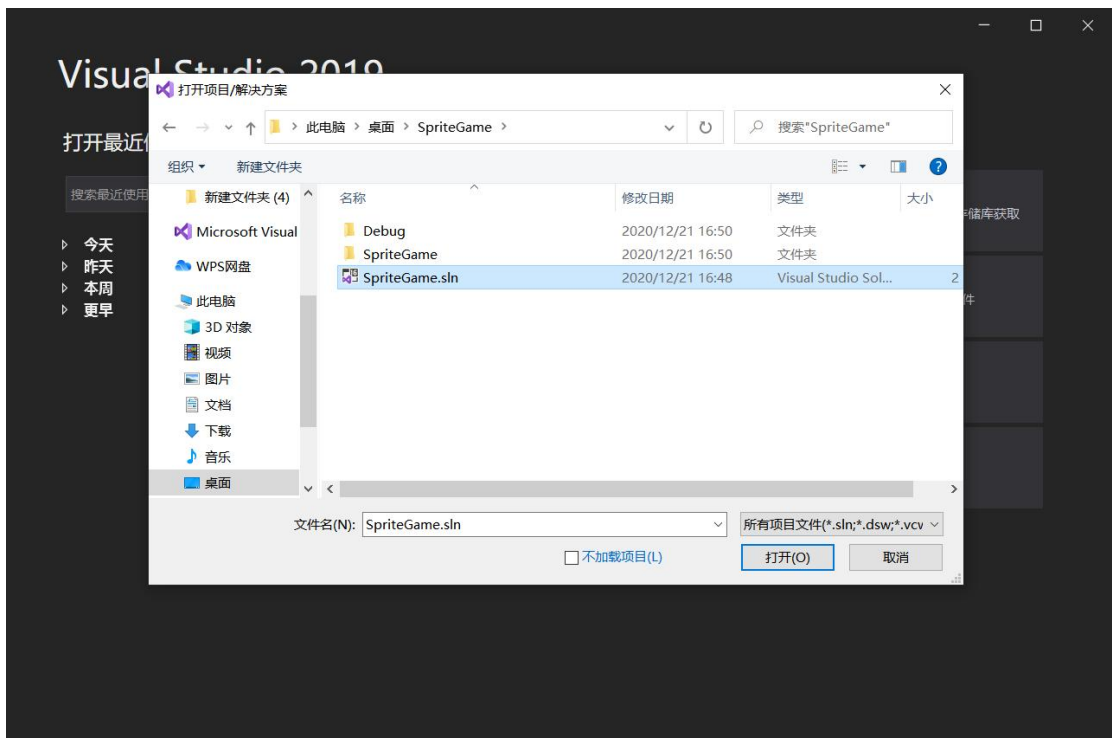
1.2 源码编译说明

（一）使用图形化编译器，下面以 Visual Studio 2019 为例说明编译过程。

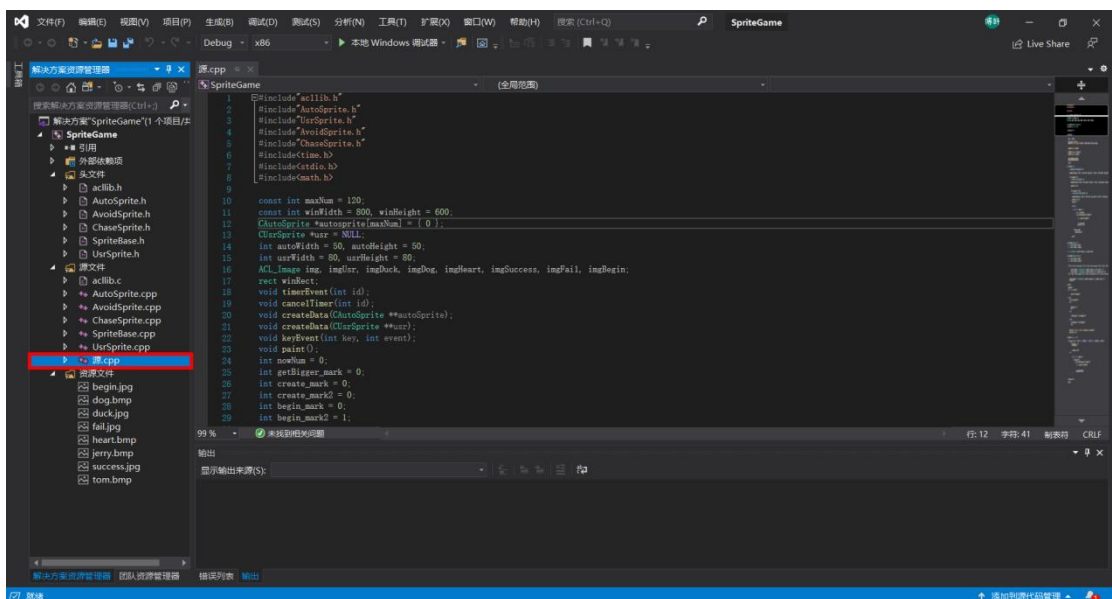
1. 打开 Visual Studio 2019，选择打开项目或解决方案，如下图：



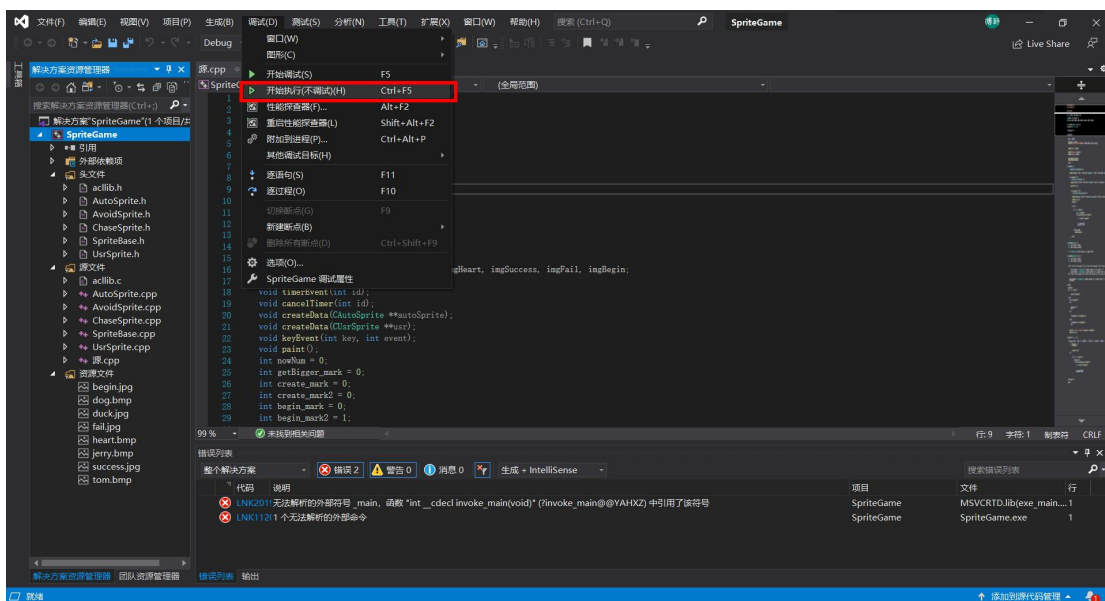
2. 找到文件夹中的 SpriteGame.sln 选择打开，如下图：



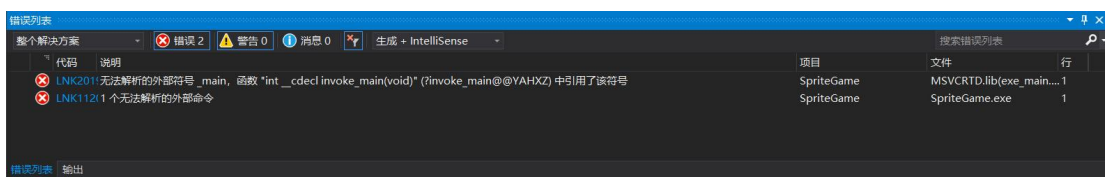
3. 选择左侧源文件目录下的 源.cpp 打开，如下图：



4. 选择 调试->开始执行，即可打开游戏，如下图：

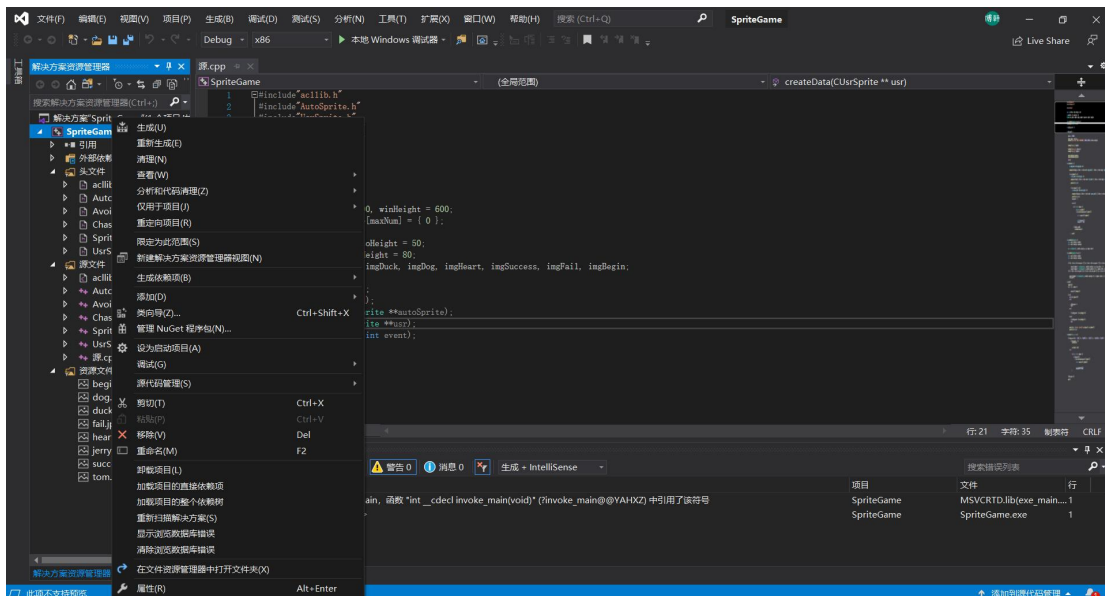


常见报错解决方法：
若开始执行后出现如下报错：

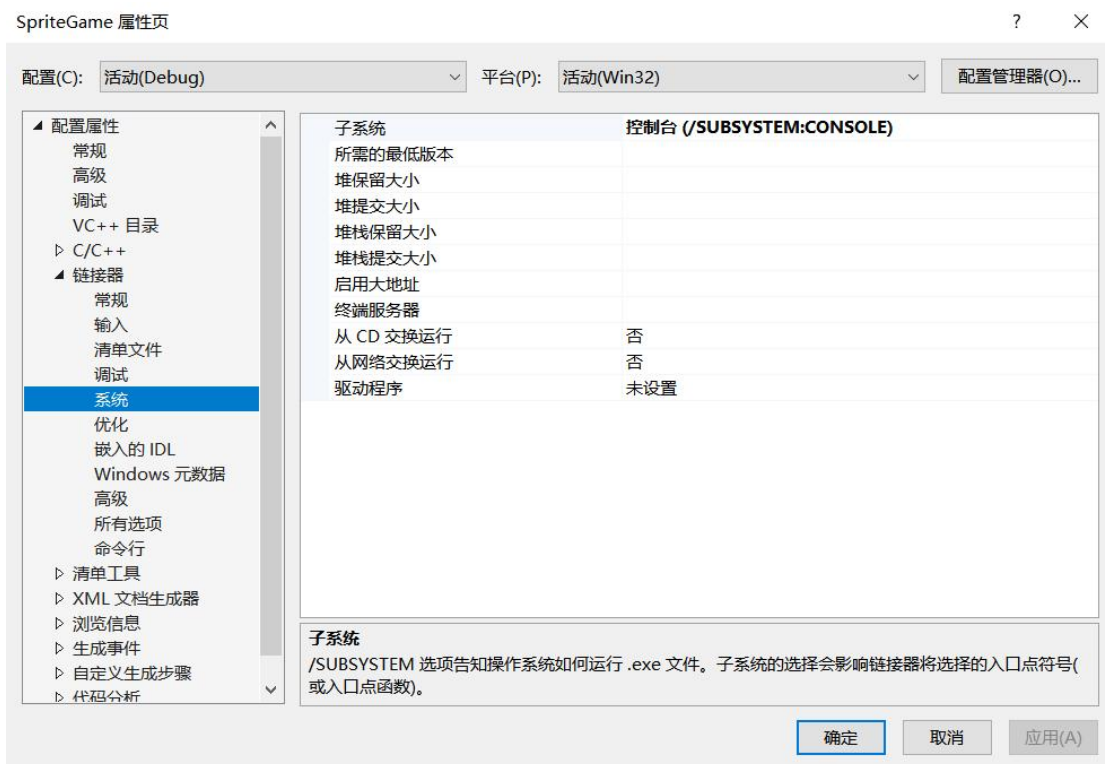


解决方法如下：

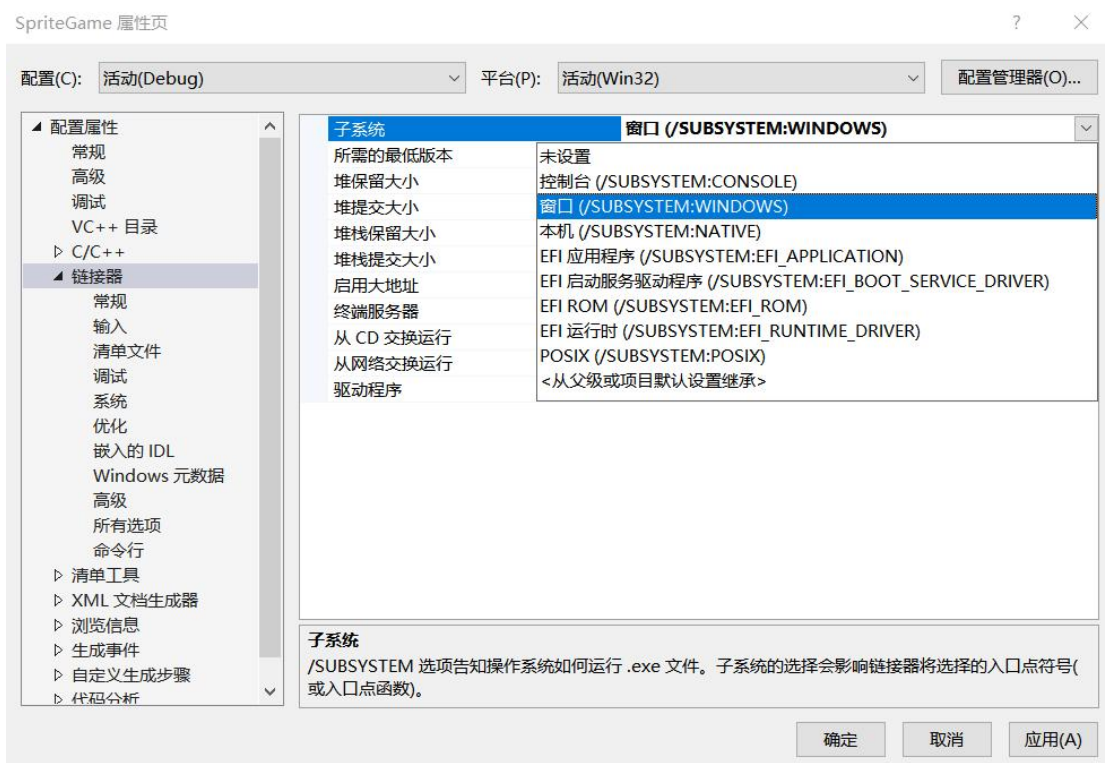
1. 右键点击左侧资源管理器中的 SpriteGame ， 选择 属性 打开，如下图：



2. 选择 配置属性->链接器->系统，如下图：



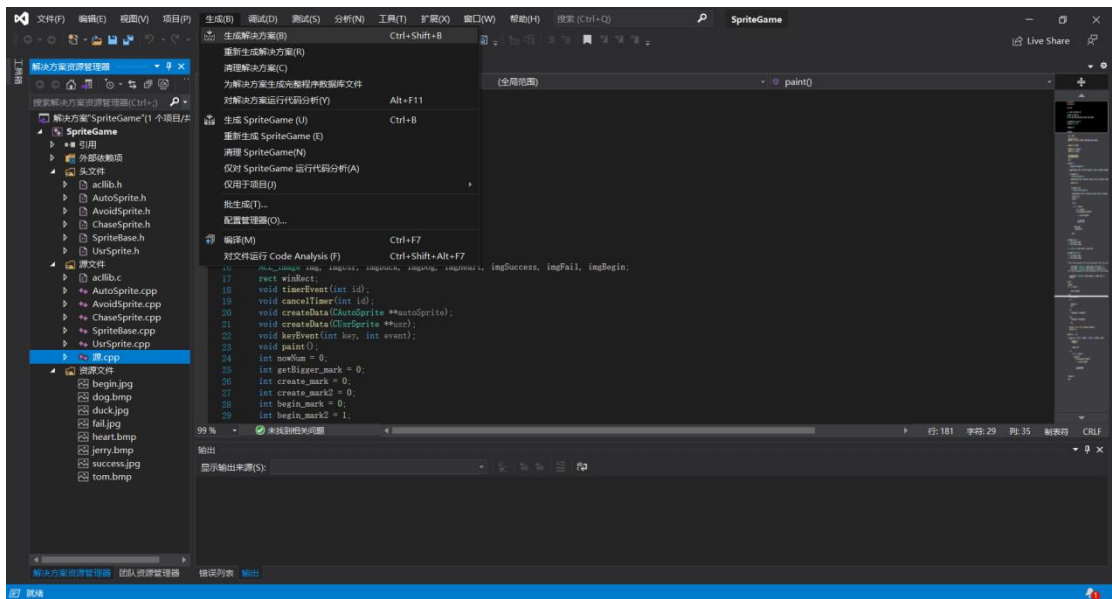
3. 将 子系统 中 控制台 (/SUBSYSTEM:CONSOLE) 改为窗口 (/SUBSYSTEM:WINDOWS)，点击确定，如下图：



5. 按正常步骤重新执行即可

(二) 生成可执行文件方法

1. 在进入下图所示页面后，点击工具栏上方的生成按钮



2. 点击生成 SpriteGame



3. 精灵游戏生成的 SpriteGame.exe 文件在 Debug 文件夹中，此时若直接打开此文件，则会产生以下报错提示：



4. 产生此报错的原因是 acllib 库加载图片时图片必须与可执行文件在同一个文件夹中，此时将 Debug 中的 SpriteGame.exe 文件拷贝到与图片同一个文件夹。

名称	修改日期	类型	大小
ChaseSprite	2020/12/20 14:29	C++ Source File	2 KB
ChaseSprite	2020/12/20 14:29	C Header File	1 KB
dog	2016/5/24 19:16	BMP 图片文件	199 KB
duck	2020/12/20 11:19	JPG 图片文件	40 KB
fail	2020/12/20 16:02	JPG 图片文件	26 KB
heart	2016/5/19 18:41	BMP 图片文件	532 KB
jerry	2016/5/19 18:10	BMP 图片文件	91 KB
SpriteBase	2020/12/19 21:35	C++ Source File	1 KB
SpriteBase	2018/8/1 11:24	C Header File	1 KB
SpriteGame	2020/12/22 19:35	应用程序	93 KB
SpriteGame.vcxproj	2020/12/22 17:14	VC++ Project	8 KB
SpriteGame.vcxproj.filters	2020/12/21 16:50	VC++ Project Fil...	3 KB
SpriteGame.vcxproj.user	2020/12/21 16:48	Per-User Project...	1 KB
success	2020/12/20 15:53	JPG 图片文件	26 KB
tom	2016/5/19 18:31	BMP 图片文件	583 KB
UsrSprite	2020/12/20 14:42	C++ Source File	3 KB
UsrSprite	2020/12/20 14:29	C Header File	1 KB
源	2020/12/21 0:29	C++ Source File	7 KB

5. 此时打开 SpriteGame.exe 文件则可启动游戏

2 系统设计

1. 界面实现

采用 `acllib` 库函数创建界面、显示图片文字等；

2. 精灵移动效果实现

为了达到精灵移动的视觉效果，主要采用在比较短的时间（40ms）内，不断更新图像在屏幕上的坐标并显示的方法来实现；

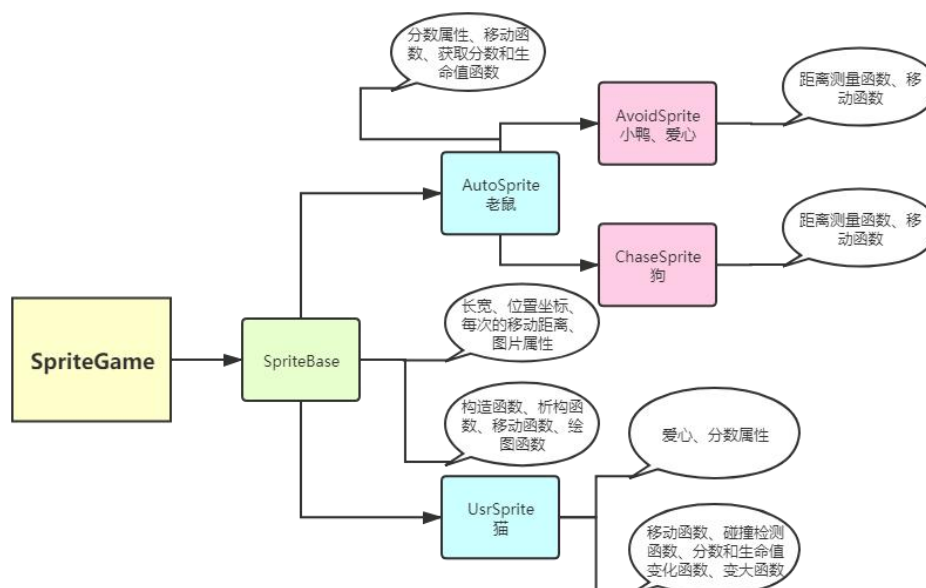
3. 精灵的实现

在程序设计之中，主体对象主要有猫、老鼠、小鸭、狗和爱心。它们有许多公共属性以及相似操作，因此应用 C++ 之中的类来实现，并采用封装、继承、多态的思想来设计对象。

在实现精灵时，主要有以下几个方面任务：

- ①进入游戏后，在屏幕内某一位置随机生成猫图像。
- ②每隔 1s 在屏幕内的任意位置随机生成一只老鼠或一只小鸭并更改已有图像的坐标，更新显示，以达到视觉上图像移动的效果。
- ③当分数达到 20 分后会随机生成狗，当分数达到 25 分后会随机生成爱心并更改已有图像的坐标，更新显示，以达到视觉上图像移动的效果。
- ④生成的精灵按照要求随机移动。老鼠近距离碰到猫或者碰到墙移动会反向；小鸭和爱心需要检测它们与猫的距离是否小于给定距离，若小于给定距离则控制移动方向，使其逃离猫；狗需要检测它们与猫的距离是否小于给定距离，若小于给定距离则控制移动方向，使其追逐猫。
- ⑤老鼠和猫不能穿墙，小鸭、狗和爱心可以穿墙。
- ⑥实时检测键盘方向键的输入情况并且根据输入情况来控制猫的移动。
- ⑦在游戏进行的过程中，实时检测猫和其他物体是否碰撞，完成碰撞检测后更新分数和生命值属性并把与猫碰撞的精灵清除。
- ⑧当分数达到 20 和 50 时，猫会变大。
- ⑨在按键检测过程中实时检测是否空格键被按下，若被按下则暂停/继续游戏。

程序设计时各个类之间的关系和实现的功能如下图：



3 程序实现

1. 运用下面函数创建窗口：

```
void initWindow(const char title[], int left, int top, int width, int height);
```

2. 窗口创建完毕后，运用下面函数将图片导入：

```
void loadImage(const char *pImageFileName, ACL_Image *pImage);
```

3. 进入程序后，通过 createData(CUsrcSprite **usr) 函数在随机的某个位置生成猫的图像，猫的原始长宽在源程序中给定并且可以更改。

4. 用定时器事件触发函数 timerEvent(int id)，每隔一定较短的时间更改老鼠、小鸭、爱心、狗的坐标值并更新显示，以达到随机移动的效果，借用定时器事件触发函数 timerEvent(int id)，每隔 1s 产生新的老鼠、小鸭、爱心、狗并显示。

5. 老鼠、小鸭、爱心、小狗的距离检测函数以及移动函数设计：

①老鼠移动函数 move(rect r1) 设计：

在老鼠生成时，给定两个数字 dx 和 dy 作为每次老鼠随机移动的步长，每隔较短的相同时间老鼠的 x 和 y 坐标分别加上 dx 和 dy，更新老鼠图像位置，达到随机移动效果。dx 和 dy 为经过试验反复测试后得到的较为合适的数字。

②距离检测函数 mynear(rect ur) 设计：

基类函数通过结构体的设计以及 getRect() 函数，实现在类外访问对象的 x 和 y 坐标值。距离检测函数即计算两个对象之间的坐标间的距离并与给定值比较，判断小鸭、爱心和狗是否处于猫的一定范围内。

③小鸭、爱心和狗移动函数 move(rect r1) 设计：

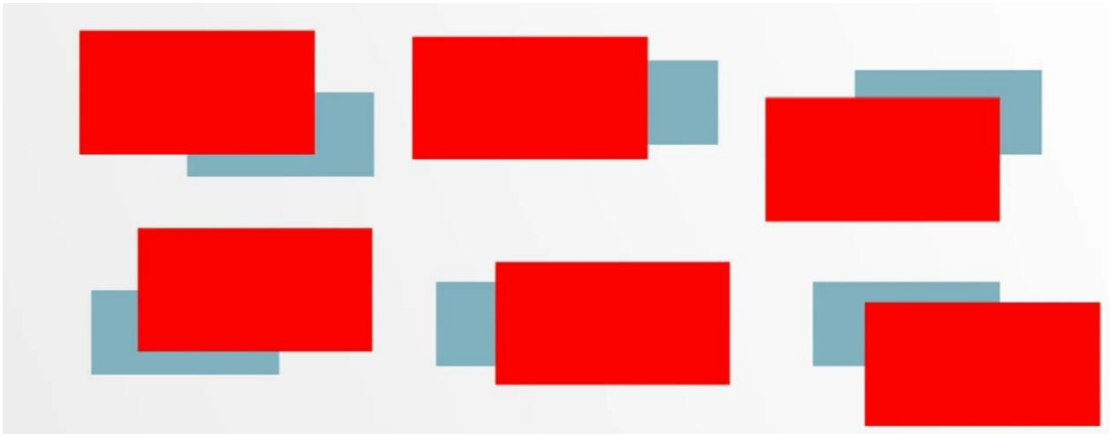
在小鸭、爱心、狗生成时，给定两个数字 dx 和 dy 作为每次小鸭、爱心和狗随机移动的步长，每隔较短的相同时间小鸭、爱心和狗的 x 和 y 坐标分别加上 dx 和 dy，更新小鸭、爱心和狗图像位置，达到随机移动效果。每次小鸭、爱心和狗移动前，都检测此三者与猫的距离，若距离大于给定距离则按照上述步长实现随机移动；若距离小于给定距离，则比较小鸭、爱心和狗的 (x, y) 与猫的 (x, y) 之前的大小关系，对于小鸭和爱心来说，需要远离猫，若 $x/y < tom.x/tom.y$ ，则说明爱心在猫的左侧/下侧，则小鸭和爱心应该向左侧/下侧运动，即 $x-=dx/y-=dy$ ，若 $x>tom.x$ 则相反， $x+=dx/y+=dy$ 。而对于狗需要追逐猫，则上述操作与小鸭和爱心相反即可。

④猫的移动函数 move(int key) 设计：

调用 acllib 库中有键盘操作的函数以及查看相应的虚拟键码，则可以检测对应按键是否被按下，当对应按键被按下后，更改猫的 x 和 y 值以及更新显示，同时进行碰撞检测。

⑤碰撞检测 collision(rect r2)：

利用 rect r2，传入带碰撞检测的物体（老鼠、小鸭、爱心和狗）的坐标以及图片长宽。下图中蓝色方格代表老鼠、小鸭、爱心和狗图形，红色方格代表猫图形，则二者各种可能的碰撞部分情况如下图所示：



根据猫和精灵的位置运用相应的判断语句实现。

当判断发生碰撞后，做出相应的清除该图像、更新显示、分数与爱心值更新并显示即可。

6. 在每次更新显示时，反复判断 `scores` 的值，将猫的大小加以调整。

7. 更新显示利用 `paint()` 函数，在产生新的图片和需要移动图片时，都需要更新显示，更新显示的过程中，为了防止残影的出现，先通过 `clearDevice()` 来清除原有屏幕上的图像，然后依次遍历对象数组，若对应对象数组中的项不为空则代表对象存在调用显示函数 `drawSprite()` 来显示该对象。

8. 根据分数，调用猫的显示函数来显示猫。而 `drawSprite()` 函数中调用了 `putImageScale(img, x, y, width, height)` 显示函数，并且由于对象的显示函数功能相同，仅仅是图片大小的区别，所以将 `drawSprite()` 函数放入基类之中，其他类公有继承该函数即可。

9. 要想实现游戏进入页面以及按任意键进入游戏的功能，首先先制作好游戏进入页面的图片，在 `Setup()` 函数中导入该图片。引入标记变量 `begin_mark`，起始值赋值为 0，在 `timerEvent()` 函数内部，判断 `begin_mark` 的值，若 `begin_mark` 值为 0 则显示游戏开始的界面。在按键检测的 `keyEvent(int key, int event)` 函数中不断检测是否有按键按下，若有按键按下则 `begin_mark` 值赋值为 1。这实现了游戏进入页面以及按任意键进入游戏的功能。

10. 想要实现在游戏中暂停和继续游戏的功能，定义标记 `begin_mark2` 初值为 1 并在按键检测中不断检测是否空格键被按下，如果空格被按下，`begin_mark2` 乘 -1。同时限定当 `begin_mark2` 为 1 时猫才能执行键盘移动，并规定当 `begin_mark2` 为 1 时开始计时，`begin_mark2` 为 -1 时停止计时。这实现了暂停/继续游戏的功能。

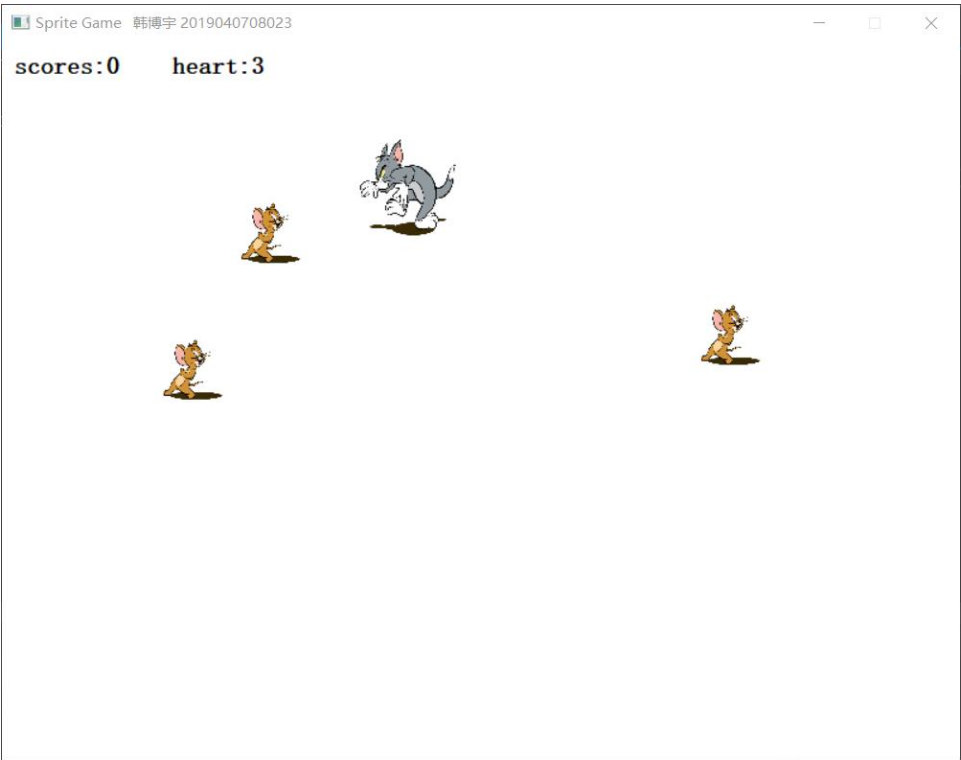
11. 在定时器响应函数中实时检测 `scores` 值与 `heart` 值是否达到游戏成功和失败的要求，若游戏成功或失败则显示之前在 `Setup()` 函数中导入的关于游戏成功和失败的图片。

4 测试报告

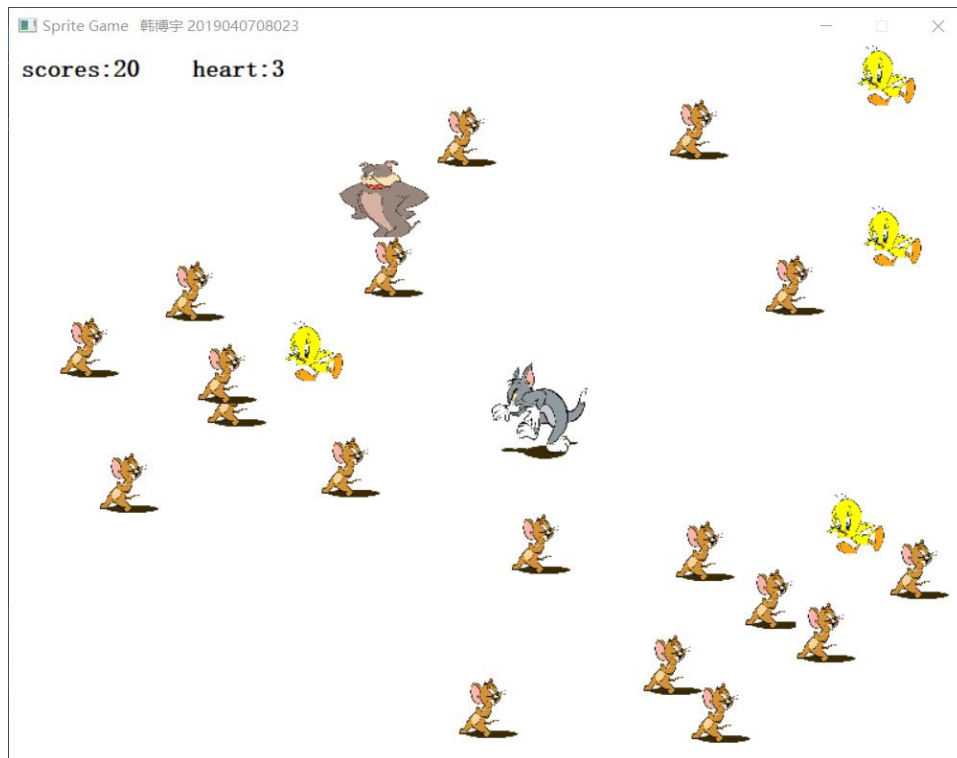
打开游戏界面



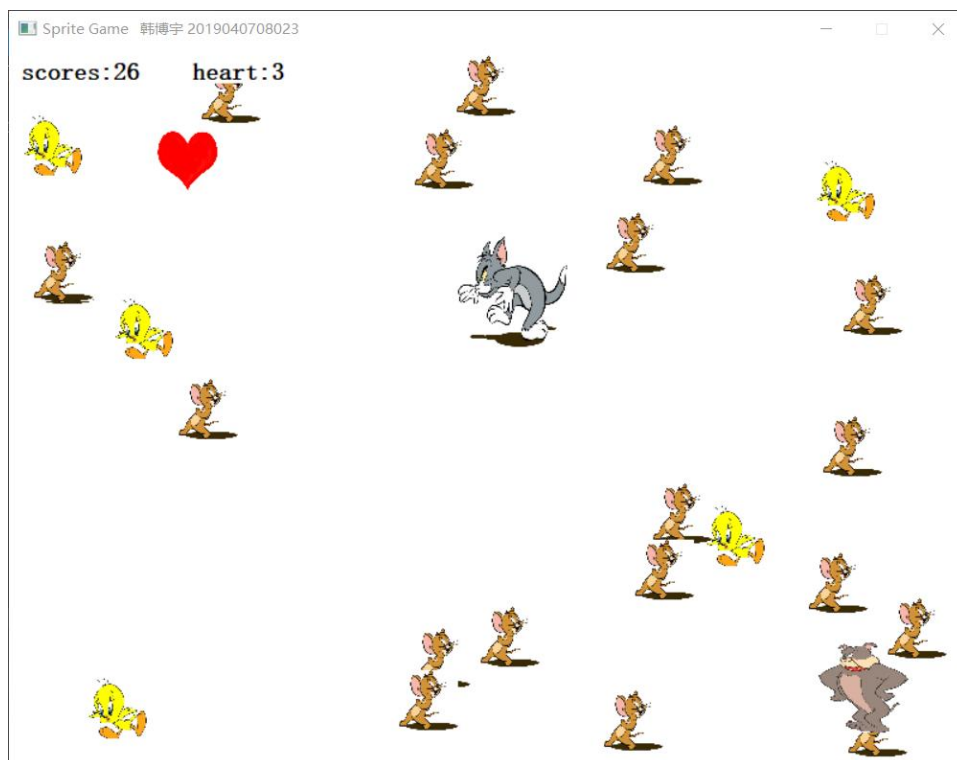
按下任意键后开始游戏



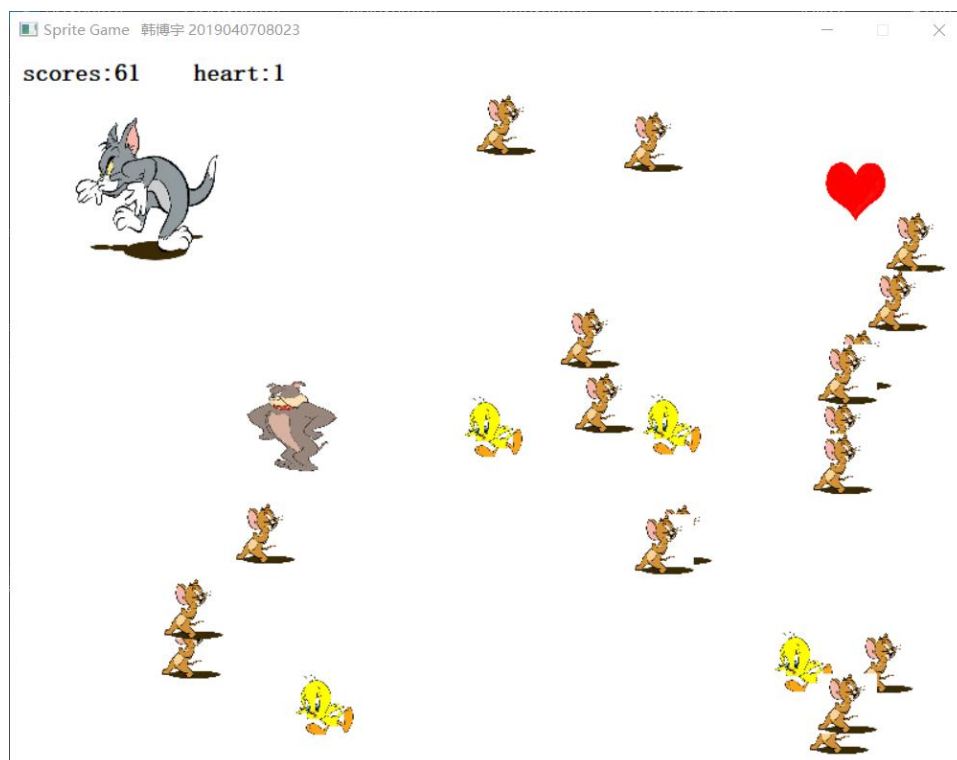
当积分达到 20 后，出现狗



当分数达到 25 后，出现爱心且猫变大



分数达到 50 后，猫再次变大



当分数达到 100 时，游戏成功



当生命值用尽时，游戏失败



按下空格可以暂停/继续，由于无法通过图片展示，此处没有说明

在游戏时，小鸭和爱心在离猫一定距离时会逃离猫，但并不是永远无法抓到；狗在离猫一定距离时会追逐猫，但并不是猫一定会被抓到。

抓老鼠技巧：老鼠不可以穿墙，将老鼠逼到墙边即可容易抓到。

抓小鸭和爱心技巧：由于小鸭和爱心的躲避函数的存在以及其穿墙功能，当小鸭和爱心在窗口边缘时，很难被抓到；但当其运动到窗口中央时，操控猫预判其躲避位置，有较大概率抓到

躲避狗追逐技巧：由于狗的跟踪函数以及狗运动方向的固定性的存在，应尽量避免猫往角落运动，当被小狗跟踪时尽量左右快速移动离开小狗的检测范围。

5 实验心得

1. 目前仍存在的问题：

当游戏结束后继续按上下左右键，显示界面会出现频闪。

2. 之前解决过的问题：

按下空格键后，猫可以通过键盘继续移动，之后通过引入 mark 标记变量，更改了移动条件解决了此问题。

老鼠被逼到墙边后会发生频闪，通过引入另一个 mark 标记变量并增加 if 语句的判断条件解决。

游戏难度系数难以控制，容易发生猫抓不到老鼠和小鸭，逃不了狗的追逐的情况。通过不断尝试，修改各个精灵的移动速度和产生时机解决。

3. 想要增加但未实现的功能：

想要增加一个双人对战精灵游戏，但是 `acllib` 库无法同时识别两个按键的输入，所以没有实现这一功能，考虑之后采用其他库，优化实现这一功能。

由于我是通过转专业进入计算机学院的，所以没有做过 C 语言版本的精灵游戏，这次实验我从头学起，最终顺利完成。通过此次试验，我初步掌握了 `acllib` 库的调用，进一步理解了 C++ 中类、封装、继承、派生的概念，对 C++ 面向对象的编程思想有了更深的领悟和思考。

在制作精灵游戏的过程中，我将理论运用到实际，实现了学以致用；在修复 bug 的过程中，我体会到理论知识的学习和实际操作是实现是有很大区别的，实际情况更加复杂多样，需要我们有严谨细致的编程思想。通过这次实验我收获颇丰，希望今后在学习过程中也能继续学以致用，在实战中成长。也非常感谢老师一个学期的悉心讲授。