

贪吃蛇

Gluttonous Snake

刘安民 2023.05.25

讲点什么

01

任务分解

带你看一遍作业要求

02

主题设计

设计一个好玩的贪吃蛇游戏

03

结构设计

使用OOP思想进行结构划分

04

注意事项

你猜要注意什么

俗话说
干饭人 干饭魂
ojer 都用大铁盆
身在曹营心在汉
一到上课想干饭
天气炎热 依然不能阻挡 ojer 干饭的决心
干字洋溢的热情
是对饭的最大尊重



01

任务分解

基础项

入门版

方向键控制；食物位置随机，数量随机，吃光则再次随机产生；当蛇撞到边界或者头身相撞时，游戏结束。

进阶版

蛇挂掉后，**尸身变成边界**，再随机产生新的食物和蛇，游戏继续。直到剩余空间不足以生成新的蛇和食物为止。

高级版

蛇挂掉后，**尸身变成食物**，再随机产生新的食物和蛇，游戏继续。直到撞墙次数 >5 ，或剩余空间不足以生成新的蛇和食物为止。

历史

将游戏结果**记录在文件中**，编码格式自定，要求能解析并输出到屏幕。至少保存**版本、用户名与得分**，且程序可以灵活**增删改查**（改查仅限针对用户名）。

实时UI

1. 蛇的长度
2. 剩余的生命数（对于部分模式）
3. 当前的分数，制定不同加分逻辑
4. 历史最高分，若当前最高则同步
5. 游戏时间

菜单

加分项

更复杂的游戏规则

软墙壁 & 硬墙壁 加速区 & 减速区
体能规则 大型地图 不同的食物

更丰富的游戏功能

加载历史游戏 RPG剧情模式

人机对战（AI蛇） 双人对战 联机对战

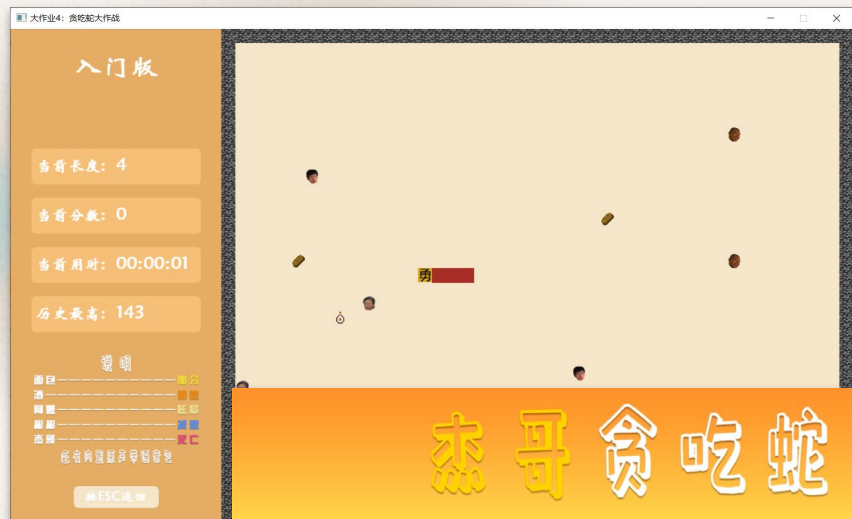
更美观的游戏界面

Easyx图形化界面

详见另两份PPT以及互联网

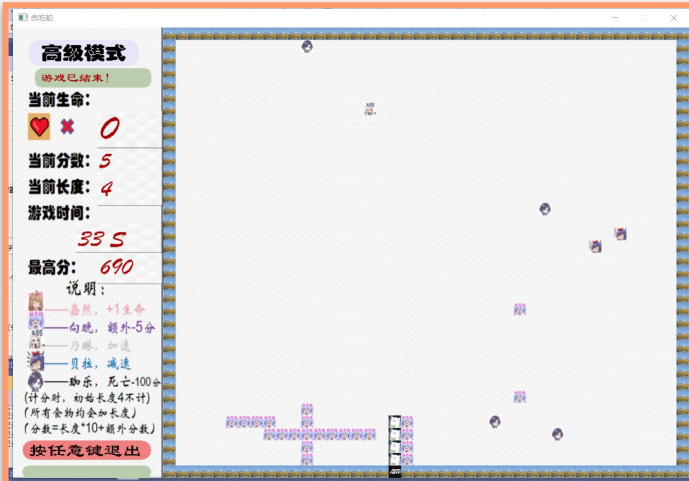
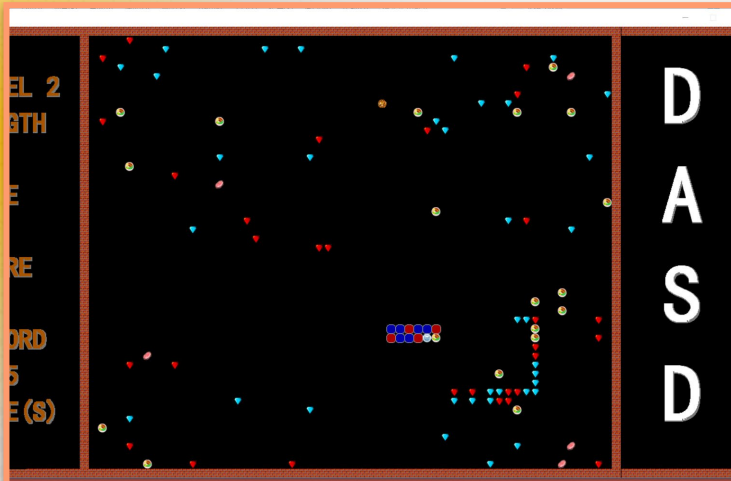
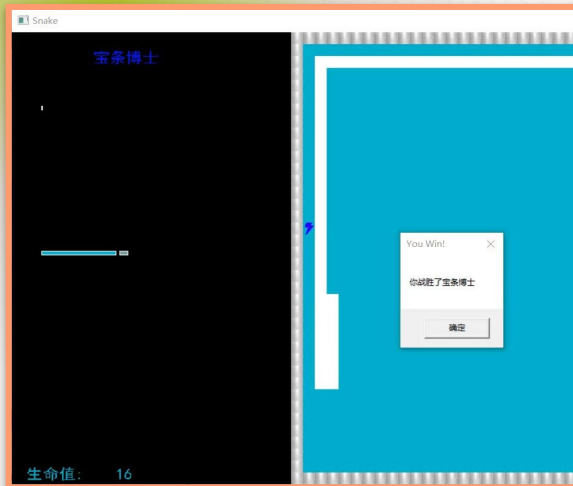
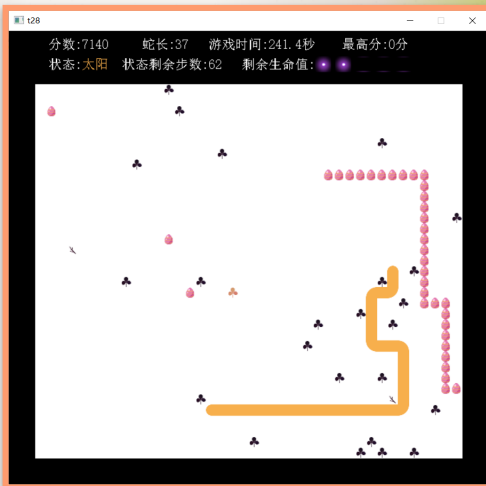
放飞想象与设计创意！

仅供参考，具体要求以OJ为准



02

主题设计



把作业编程变成快乐编程

视听效果

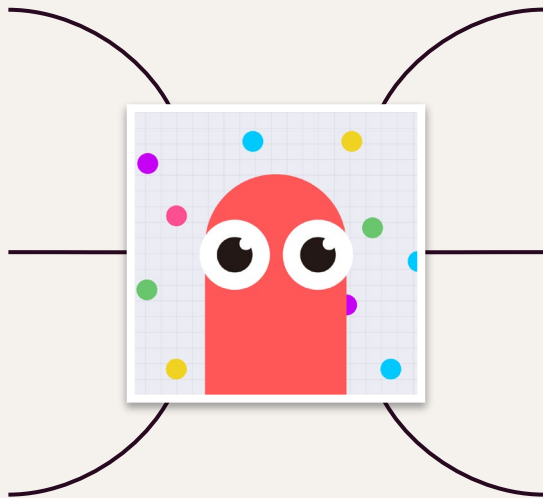
添加BGM与音效；元素贴图；
模拟动态效果等

主题设计

个人兴趣爱好；影视或音乐
作品；时事热梗等

丰富规则

食物效果；蛇存在多种状态；
限时奖励；障碍；传送门等



情节模式

逃离模式；与敌对者战斗；
拯救模式；夺宝模式等

惊喜彩蛋

隐藏成就；点击触发事件；
特殊语音等

Your Design

...

这么多复杂的设计，如何实现？

Game_Control 外层简单示例

```
int Game_Level = 0;
int Game_Life = 0;
int Game_Score = 0;
int Game_Time = 0;
int** Game_Map = NULL;
```

```
void Game_NewGame()
{
    Game_Initial(Game_Mode);
    Walls walls(Game_Map);
    Foods foods(Game_Map);
    Snake snake(Game_Map);

    while (Game_Life)
    {
        Single_Game(snake, walls, foods); //单局游戏
        Between_single(snake, walls, foods); //局间处理
    }
}
```

不同游戏模式只针对局间处理不同，单局游戏的实现是相同的

相同模式下，新游戏和读取存档只针对初始化不同，其余实现也是相同的

一种参考的实现方法

```
void Game_Top_Control()
{
    int mode = 0;
    while (1)
    {
        mode = choose_menu(); //选择操作类型

        if (mode == start_bottom)
        {
            Game_Level = choose_level(); //选择游戏难度
            Game_NewGame();
        }
        else if (mode == load_bottom)
        {
            Game_LoadGame();
        }
        else if (mode == rank_bottom)
        {
            draw_rank();
        }
    }
}
```

03

结构设计

结构设计没有标准答案，希望回来看PPT的你不要照搬这一节的内容，而且也没什么好搬的

这个程序要做什么？

01

单局游戏

初始化(食物, 蛇, 墙, UI等);

```
while (游戏结束判断) {
```

蛇的移动 (吃食物) ;

食物判断与更新;

UI界面与地图更新;

}



02

局间处理

- 食物与墙的更新（占蛇身）
- 状态（生命值）与UI更新
- 蛇和食物的重新生成

04

历史记录

- 制定编码解码规范
- 增删查改

03

游戏结束

- 更新历史记录与榜单
- 析构！

05

加载游戏

- 从文件进行对象初始化

关于划分的一些问题

程序分几个部分 (class) 比较合适?

控制

蛇

食物

墙壁

...

每部分之内存储什么数据 (private)?

位置

状态

数量

比率

...

每部分之间共享什么数据 (public)?

每部分执行哪些操作?

判定

修改

生成

绘制

...

我们希望各模块独立性强，模块间关联尽量小（高内聚，低耦合）

设计思路一

```
class Game
```

```
char map[height][width];
```

```
char headToward;
```

```
pair<int, int> tailPos;
```

```
pair<int, int> headPos;
```

```
void move(); // 键盘读方向
```

...



① 非常**直观**，对于简单的贪吃蛇很好实现

② 很好实现图形化界面输出（仅针对网格）



① 可扩展性一般，几乎没有OOP思想

这个程序要做什么？

01

单局游戏

初始化(食物, 蛇, 墙, UI等):

while (游戏结束判断) {

蛇的移动(吃食物):

食物判断与更新：

UI界面与地图更新:

5000

02

局间处理

- 食物与墙的更新（占蛇身）

- 状态（生命值）与UI更新

- 蛇和食物的重新生成

22

04

历史记录

力大如牛

03

游戏结束

- 更新历史记录与榜单

- 析构:

加

120

从文件进行对象初始化

[illegible]

设计思路二

```
typedef pair<int,int> COORD;  
U P L R H / W / 1 2 3 4 5 / ...
```

class Snake

```
char headToward;  
COORD tailPos;  
COORD headPos;
```

```
-----  
void create(char** map);  
void move(char** map);
```

class Food

```
vector<COORD> foods;  
-----  
void renew(char** map);
```

初始: wall → snake → food

class Control

```
char map[height][width];  
Snake snake;  
Food food;  
-----  
void run();
```

可以是其他返回类型吗?
bool? enum? COORD?

① 保留了原本的直观与易输出优点

② OOP思想, 提升了程序的可扩展性

可以根据自己的设计想法, 完善或修改这个设计



一些思考

每次刷新时全部重画还是只画更新的区块？如何知道哪些块更新了？

- 一般情况下，只需要更新蛇头、蛇尾、刷新的食物；
- `vector<COORD>buffer;`

墙需不需要封装？食物和墙可不可以同属一个class？

对于多种不同种类与功效的食物，如何方便管理？

- 每种食物可以单独封装管理（刷新机制、属性等），Control传被吃的食物坐标给Food，Food再根据食物种类调用相应对象的方法

一些思考

如何设计人工蛇与AI蛇的成员与方法？

- 封装snake基类，派生AISnake和HumanSnake两类，改move方法

map[i][j]可以存多少信息？是不是一定要单独存一张MAP？

不想拘泥于网格状，想做连续状态，如何设计？

- 数据结构可基本保留，但原来的MAP也许不太适用（与区块结合？）
- 更复杂的状态判定与管理（比如物体长宽与判定边界的设计）
- 注意绘制时的绘制顺序



?

06-29 22:22:12



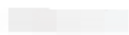
06-29 22:22:37

助教大人，开个价吧，好商量好商量



06-29 22:22:40

[动画表情]



06-29 22:23:14

你捞一捞我，我去了嘉定给你当🐮做🐻



陈宇飞老师

这种必须降档，没的商量



04

注意事项

可能还需要考虑的一些问题



实时

- 能否及时绘制？(sleep?)
- 能否及时读键盘改方向？
- 能否实时更新多方状态？



美观

- 贴图的图像处理
- UI界面与地图的设计
- 简洁（cmd何尝不简洁）



主次

- 优先考虑完成基础项
- 其次考虑完成加分项

警告

警告：对于上述加分项，你可能需要一个星期到三个星期去完成，平均下来你可能至少要在本次作业花费 20 小时的时间，故如果你希望做加分项并且有很高的完成度，请提前至少两周开始你的大作业（且不保证两周内人人皆可完成加分）。

不要卡DDL

感谢观看

CREDITS: This presentation template was created by Slidesgo, and includes icons by Flaticon and infographics & images by Freepik

一些好图，奖励你课下还来看这份PPT

能不能把游戏背景换成二刺螈美少女

当然可以

每吃一个美少女都会为我欢呼

就得是那种，一运行，电脑屏幕一黑

然后慢慢亮起，占全屏的3a感觉

然后不能有鼠标点击这种low的

要滚轮



LV34 群主 陈宇飞

是学一门高（搞）快乐的科（课）程



LV34 群主 陈宇飞

五一长假，大家搞搞压缩，后面的作业都是搞快乐的，所剩不多了

经由这次的编写，我的毅力以及对于类的了解都大幅上升了。我也逐渐了解了将需要改变的参数放在一个类之中所带来的便捷性。之后程序制作中我都会大量使用这样类似的编码结构的（吧大概）。

在整个编程的过程中，我发现了，我在编程的过程中依旧存在许多的低级错误，例如循环参数没有置零等。这些错误导致我的debug过程出现了许多意想不到的麻烦，拖累整体进度，因而我之后会设置更多的参数，并且进行详细记录，避免参数错误的问题发生。

（在最后，小小的说一些无关紧要的无聊事。编写贪吃蛇的过程中，我经历了一个档的滑，原来想来软院的结果没了。所以在某段时间内我陷入了摆烂，一点都不想干活因为感觉没有意义了。然后后来就，继续写了。没错我只是在为我的贪吃蛇没有那么高的完成度找借口而已。而且，我还凑到了字数，哈哈quq）

第 2 页

头在地证明了找编程能力的坚持提高。

但还是开心的，因为打开游戏就能看到自己喜欢的歌手——房东的猫。设计游戏 UI 及背景美化的时候是最开心的，因为从网上浏览寻找房东猫图片时总是看不够:~)。

这条小贪吃蛇，现在可以叫它“贪吃的猫”，因为程序中的蛇头是小黑猫！在它的设计过程中，我遇到了许许多多的问题，也曾苦恼好久，尤其 UI 设计对于我是一个以前未曾实践过的领域。但现在，不管怎样都走过来了，我相信，这些问题的遭遇和解决都会成为我能力提升的必要经历。看到跑得欢快的贪吃猫，是快乐的编程成果！

几个小小的思考点：（参考easyx文档和其他资料解决）

1. 游戏封面：用卡米拉的绝美照片作为封面背景，如何使照片与窗口大小相适应（调loadimage函数参数）
2. 使用图片作为食物，如何使图片大小适应小方块（loadimage和putimage函数组合，并调节参数），将图片都放入pic文件夹，相对路径怎么写（如“pic\\c.jpg”）

这个游戏主要是在吃家没反应过来时的时候蛇吃掉，优先吃正前方食物可以利用蛇运动的时间间隔想好下一步打算。

这次作业，又一次让我体会到画出流程图的重要性。想清楚再动手，减少山一样的代码。刚开始一头雾水，一厢情愿地想着，痴痴地望着屏幕，却一种无力感涌上心头。后来呵，一步一步画出流程图，朦胧的影儿逐渐清晰..... 此时，天朗气清！突破！！

1.6.1. 食物

蓝宝石：分值为5，刷新率为35%，吃到后增加一节蓝色身体。

红宝石：分值为5，刷新率为35%，吃到后增加一节红色身体。

马蹄酥：分值为10，刷新率为5%，吃到后增加一节蓝色身体。

猪肉：分值为-10，刷新率为5%，吃到后增加一节红色身体。

加分盒饭：分值为50，刷新率为15%，吃到后增加一节红色身体，并好吃到想跳舞，触发跳舞特效，伴随着《本草纲目》按照右侧状态栏的提示顺序按下“W”“S”“A”“D”，若跳错动作则会导致受伤，死亡。

减分盒饭：分值-50，刷新率5%，吃到后增加一节蓝色身体，因为难吃所以不想跳舞。

1.6.2. 状态栏

moveStep() 函数并不实际绘制蛇体，只是根据当前方向进行蛇头节点的数据计算。

游戏所有的绘制绝大部分为独立函数，以 draw 开头，不涉及具体数据计算。绘制基于 easyx 库，并利用外载图片进行美化。游戏主题为房东的猫，新生代民谣组合，代表作《秋酿》，《下一站茶山刘》，《云烟成雨》等。

```
1 class Food {
```

祝你写出令自己满意的贪吃蛇！
快乐编程！期末愉快！