

**《软件工程》项目报告**

**题目： 贪吃蛇**

**课程名称： 软件工程**

**专业班级： CS1803**

**学 号： U201814802**

**姓 名： 江祉涵**

**同组成员： 江祉涵**

**指导教师：**

**报告日期：**

**计算机科学与技术学院**

**任 务 书**（黑体小2号加粗居中）

**一 总体要求**（黑体4号加粗,字母、阿拉伯数字为Time New Roman4号加粗）

1. 综合运用软件工程的思想，协同完成一个软件项目的开发，掌软件工程相关的技术和方法；

2. 组成小组进行选题，通过调研完成项目的需求分析，并详细说明小组成员的分工、项目的时间管理等方面。

3. 根据需求分析进行总体设计、详细设计、编码与测试等。

**二 基本内容**（黑体4号加粗,字母、阿拉伯数字为Time New Roman4号加粗）

根据给出的题目任选一题，自行组队，设计与开发中软件过程必须包括：

**1. 问题概述、需求分析：**正确使用相关工具和方法说明所开发软件的问题定义和需求分析，比如NABCD模型，Microsoft Visio，StarUML等工具 (20%)；

**2. 原型系统设计、概要设计、详细设计**：主要说明所开发软件的架构、数据结构及主要算法设计，比如墨刀等工具（35%）；

**3. 编码与测试**：编码规范，运用码云等平台进行版本管理，设计测试计划和测试用例（30%）；

**4．功能创新**：与众不同、特别吸引用户的创新（10%）；

**5. 用户反馈**：包括用户的使用记录，照片，视频等（5%）。

**目 录**（黑体小2号加粗居中）

**任务书** I

**1问题定义** 1

1.1项目背景及意义 1

1.2项目基本目标 3

1.3可行性分析 7

1.4人员管理和项目进度管理 9

**……**

2需求分析 20

2.1 E-R图、数据流图 20

2.2用例图等 23

2.3原型系统设计 25

2.3.1 ×××××× 30

**……**

3 概要设计和详细设计 40

3.1 系统结构 40

3.1.1功能说明 42

3.1.2接口设计 43

3.2类图等 43

3.3关键数据结构设计 45

3.4关键算法设计 47

3.5数据管理说明 47

3.5.1 ×××××× 45

**……**

4 实现与测试 50

4.1 实现环境和代码管理 50

4.2 关键函数说明 52

4.3 测试计划和测试用例 53

4.4结果分析 57

4.4.1 ×××××× 57

5总结 65

5.1 用户反馈 65

5.2 总结 68

6 体会 69

**附录 ××××××** 70

（章为宋体小4号加粗，其余宋体小4号，字母、阿拉伯数字为Time New Roman小4号）

**1 问题定义**

**1.1项目背景与意义**

1. N (Need 需求）：

随着时代的发展，人们的生活节奏不断地加快，可以利用的空闲时间不断减少，同时生活压力也不断的增大，人们更需要一个简单的方式来缓解压力，舒缓心情。游戏就成了最好的解决现代生活压力的手段。手游的繁荣导致端游的没落，但同样也有大量用户更加青睐于端游，这部分用户需要能进行快节奏的端游能够即玩即停，来利用空余时间愉悦身心，同时这类用户也更加希望进行更有挑战的游戏，所以也需要有足够的游戏性和难度来吸引住用户。随着IP时代的来临，一个好的IP更加能够吸引住游客的目光。而这些在快节奏的时代中任执着于端游的客户更有一种情怀的倾向。所以用贪吃蛇，推箱子，俄罗斯方块等知名经典小游戏更加容易吸引住他们的目光。用户也可以通过这些经典游戏IP来回忆往昔的自己和岁月。

1. A (Approach 方法)：

用贪吃蛇作为载体，串连起这些经典知名的IP，通过不断的边长的身体来触发不同的游戏，同时将这些游戏进行一些改动处理能够更加切合整个主题。为了满足即玩即停的需求，要求能过进行存档保存用户的游戏状态。同时为了适应快节奏的现代生活，在游戏初期要尽量降低游戏难度，使游戏简单上手，例如减少障碍物数量，较慢的移动速度，较弱的敌人等等。同时为了能够很好的留住用户，则需要中后期能够有较强的游戏性和更高的难度。例如更多的障碍物更，快的速度，更加厉害的敌人。利用ps绘画人物和地图等资源，利用c++作为脚本语言，通过UNITY来编写整个程序。

1. B (Benefit 好处)：

该游戏即能够满足快节奏生活的用户也能满足对游戏性有要求的用户，同时也包含了各种经典的老IP游戏，拥有强大的情怀加成。前期快速而又简单的游戏能够快速吸引用户，较低的上手难度不会使一些轻度玩家失去兴趣。同时后期较为更快移动速度更复杂的地形，和更强的敌人能够满足硬核玩家的需求。同时这种递进的难度也能使轻度玩家慢慢适应游戏，起到将轻度用户也能转化为忠实用户的作用。以此来增加用户黏度。用户也能通过这些游戏放松在这高压社会下的心里压力，缓解疲劳。老游戏玩家能感慨当年，新玩家也能找到游戏的乐趣所在。

1. C (Competitors 竞争)

同类的游戏一般较为古老，不再适应现在社会，用户的交互性差，UI界面不够符合现代用户。同时游戏的难度过度较差，游戏内容也十分单一，游戏性不够。而新现代的游戏只是模仿了经典游戏并没有做到创新，容易使用户提不起兴趣。而该款游戏拥有最新的unity编辑，拥有美观的UI界面，更加友善良好的人机交互，同时再游戏上也做到保留经典同时创新，不仅仅是将几款经典老游戏串联在一起，而是将其融合在一起，发挥出1+1大于2的效果，赋予了老游戏新的生命力。能够做到既易于上手，又有强大游戏性来防止用户流失，即能满足老玩家的情怀，又能给新玩家一个通道去接触游戏。

1. (Delivery 推广)：

通过UNITY的游戏云计划将游戏发布到将游戏发布在网上供人们下载，将作品参加MWU中国区比赛，能够使更多的任了解这款游戏。同时将项目发送再github上可供用户下载。

**1.2 项目基本目标**

通过Unity编辑一个可以运行的贪吃蛇游戏，完成蛇头的生成，蛇身的生成以及蛇身跟随蛇头移动，判断碰撞过程，如果碰撞到食物则销毁食物并增加蛇身。判断到墙壁则进行穿越操作，使蛇身蛇头以此从对面的墙壁出现。食物生成器，能够在规定的范围内生成食物以及限时的高分道具。墙壁生成器，使得在蛇行动范围内生成墙壁阻碍蛇的移动并根据分数调整生成墙壁的时间来提高难度。蛇的移动速度也会根据分数的变化而加速。在蛇头碰到蛇身或者墙壁时判断蛇死亡，结束游戏并记录下当前得分并通过与之前的最高分进行比较得到新的最高分。

**1.3 可行性分析**

1.技术可行性分析：

通过unity构造蛇头作为物体，根据实际改变蛇头的位置完成舌头的移动，通过构造预制体蛇身作为蛇身，每次吃食物时创建蛇身，移动时蛇身的坐标替换成前一节蛇身的坐标。将食物作为预制体，在每次吃掉一个食物后利用random函数随机获得区域内的横竖坐标来完成食物的生成。

2.市场可行性分析：

通过市场上《贪吃蛇》的火爆以及之前《flappy bird》可知，消费者对于一个简单的操作的游戏是有一定兴趣的。同时大部分消费者对于简单操作又方便的游戏来打发时间是由一定的兴趣的。而且作为一款经典的游戏，也无需宣传，大多数人对此都有一定的了解，使受众范围巨大。

1. 社会可行性分析：

贪吃蛇作为一款经典的游戏，既无血腥暴力的画面，也无炫光之类的会引发癫痫的内容，是一款老少咸宜，大众都能接受的游戏，不存在社会层面上的问题。

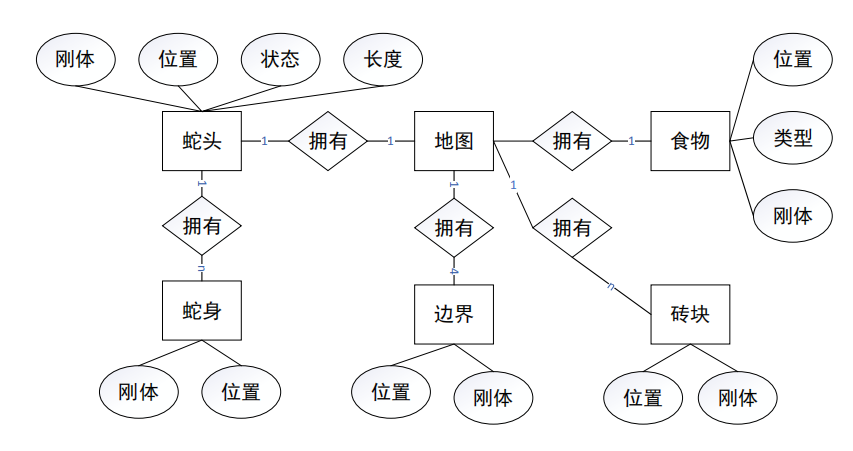
**1.4人员管理和项目进度管理**

只有本人一人完成所有内容。

**2 需求分析**

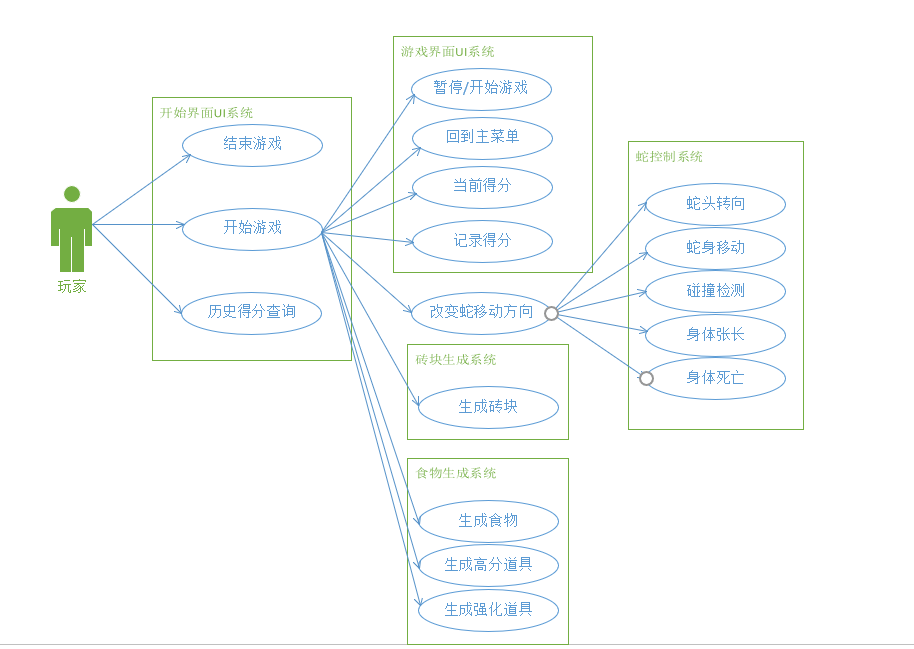
**2.1 E-R图、数据流图等**

贪吃蛇有一个地图，地图上有一个蛇头，上下左右四个边界来控制贪吃蛇碰到四周的情况，还有食物以及砖块。蛇头拥有刚体，坐标位置，状态，和长度四个基本属性，刚体决定蛇的碰撞体积，位置决定蛇头在屏幕中显示的位置，状态表面蛇是否会撞砖块而死，长度则表面蛇身的个数。同样的蛇身也有刚体来完成碰撞判定，位置来确定显示的位置。边界同样拥有刚体来判断碰撞，位置来确定游戏的边界。食物有位置来确定显示的位置，刚体来判断碰撞，以及不同种类的食物来带的特殊效果。砖块一样有位置确定显示的位置以及刚体来判断碰撞。整个贪吃蛇系统的E-R图如图2-1所示。



**2.2 用例图等**

玩家首先进入开始场景，在这个界面可以看到上次得分以及历史的得分，同样可以选择开始游戏来进入游戏场景，游戏场景分为UI界面和游戏界面，在UI界面玩家可以看到当前得分以及最高分，可以进行暂停和继续操作。在游戏界面玩家可以通过WASD来控制蛇上下左右移动，食物道具以及砖块的产生有各个系统自动产生。整个贪吃蛇的用例图如图2-2所示。



**2.3 原型系统设计**

通过unity编写原型系统，主要分为2个场景，开始场景以及游戏场景，在开始场景中玩家可以看到上次得分以及历史记录，以此来获得成就感，同时可以选择开始游戏，鼠标点击开始游戏的按键会切换到游戏场景，两个场景背景一样但UI界面不同，游戏场景增加了暂停/继续按钮以及回到开始场景按钮，同时显示当前得分以及历史记录方便玩家发现与记录的差距。玩家通过WASD键控制蛇上下左右移动，当碰撞到边界时蛇头会从另一边出现完成边界设定，游戏范围内会生成食物，同一时间只会有一个食物，当蛇头碰到食物，食物会消失并生成新的食物，蛇也会增长一节身体。当蛇头碰到蛇身或则砖块时会发出死亡音效以及动画，最后返回开始场景。玩家可以点击右上角x按钮来关闭游戏，游戏的得分会记录在data文件夹中。

**3 概要设计和详细设计**

**3.1 系统结构**

1. UI界面系统结构

UI界面能使玩家暂停或则开始游戏，暂停游戏需要设置暂停图标为按钮，并设置在鼠标抬起时触发该效果，通过冻结unity时间器来完成暂停功能，并同时将按钮图片替换成开始游戏的图标。点击开始游戏图标并在鼠标抬起时触发效果，回复unity的时间器来完成游戏的继续。同时UI界面要能显示出当前的得分以及最高分，这需要记录下最高分并且在每局游戏开始的时候读入记录的最高分。

1. 蛇的移动

蛇头控制需要完成在不接受键盘蛇头朝着一个方向移动，蛇头的移动采用物体的坐标的改变来完成，设置每次改变的步长，以及改变频率，步长固定，频率即为蛇的移动速度，可以根据得分情况改变速度。蛇身的移动是通过改变蛇身的坐标完成的，通过将前一节身体的坐标赋给这一节来完成蛇身的移动。

1. 碰撞处理

蛇头与其他物体的碰撞，通过unity自带的碰撞体积来完成碰撞检测，并通过给不同的物体赋予不同的标签来完成对碰撞种类的判定。对食物的碰撞判定，如果检测到碰撞的物体是食物，则创建一节蛇身并将其移动到队尾完成蛇的成长，并增加得分，并销毁食物。对高分物体的碰撞判定，如果检测到碰撞物体是高分物体则同样的使蛇身增强，但会给一个随机50到150分的得分并销毁高分物体。若碰撞检测到的物体为蛇身，则结束游戏，储存当前得分。如果检测到的碰撞物体为强化道具，则将其状态改变为无敌，并更换身体的颜色，持续10s并销毁强化道具。如果碰撞的物体是砖块，则判断是否处于无敌状态，如果处于无敌状态则销毁砖块，否则则调用死亡函数。如果碰撞的是墙壁则根据是墙壁的位置将蛇头的坐标改编为对面墙壁的起始位置，穿越墙壁。

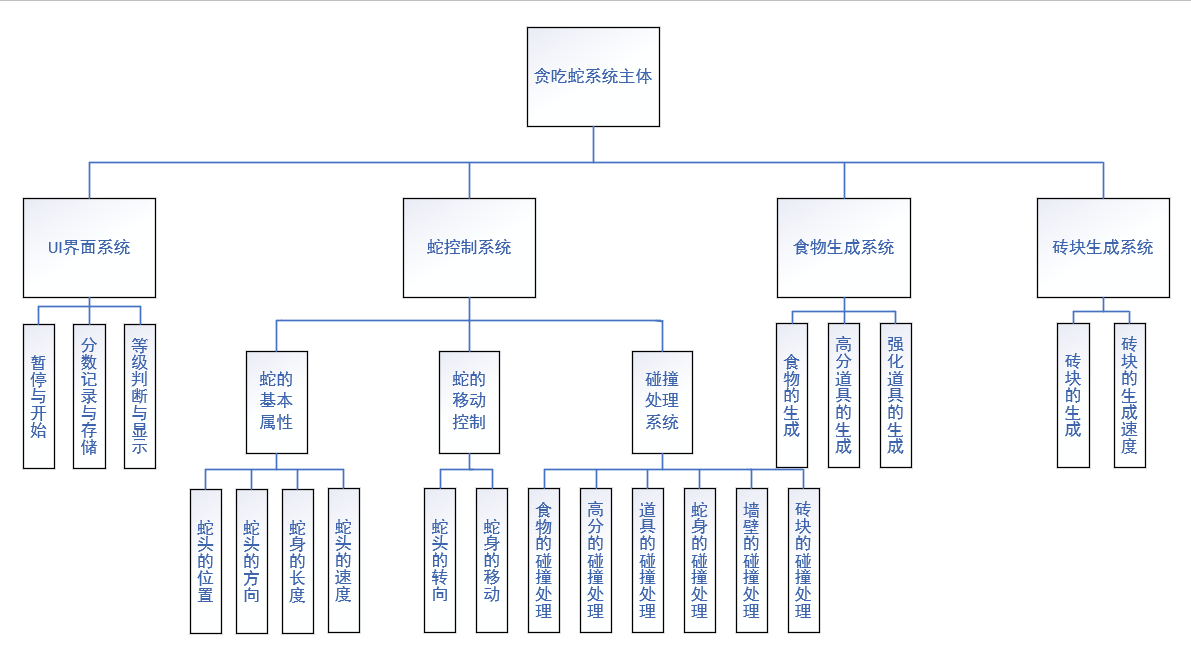
1. 食物及道具生成器。

在游戏界面内通过随机函数生成食物的x，y坐标生成食物，并判断是否与砖块或则蛇身发生碰撞，若发生碰撞则销毁该食物，重新生成食物，由于计算机反应速度快，人无法察觉这一过程，通过该方法来避免生成的食物与砖块或则蛇身重合。根据一定的概率生成高分道具以及强化道具，高分道具以及强化道具和食物生成一样，但有不同的预制体，以及在10s中之内会销毁自身。初始化时先生成一个食物，在每个食物被吃掉后调用食物生成函数来完成时间内只有一个食物。

1. 砖块的生成器

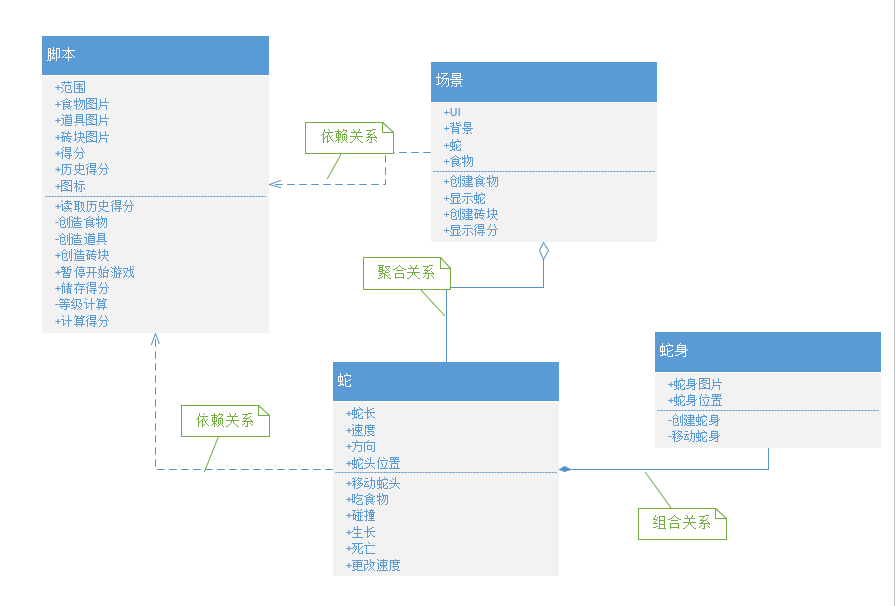
在游戏界面内通过随机函数生成砖块的x,y坐标，并判断是否有其他砖块或则食物或则蛇身发生碰撞，如果发生则销毁该砖块，并重新生成砖块来避免重合现象。砖块在生成间隔会随着分数的提高而不断的缩短以此来增加游戏难度。

整体的系统结构图如图3-1。



**3.2 类图等**

贪吃蛇主要分为四大类，场景类包含了游戏场景的背景，音效，按键图标以及蛇位置的定位点等等。脚本类是一个挂无主体的脚本的一个类，包含UI控制脚本，食物生成脚本，砖块生成脚本等，控制除蛇以外的所有活动逻辑，例如产生食物，计算得分，产生砖块等等。蛇类为玩家可以操控的主体，存放蛇的基本信息以及控制蛇移动以及移动碰撞判断等逻辑实现，蛇身是组成蛇的一部分，内含已经制作好的蛇身的预制体，以及创造蛇身并更改蛇身位置的逻辑操作。类图如图3-2所示。

****

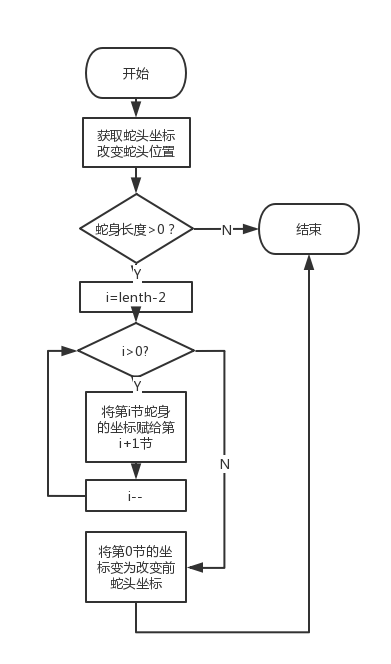
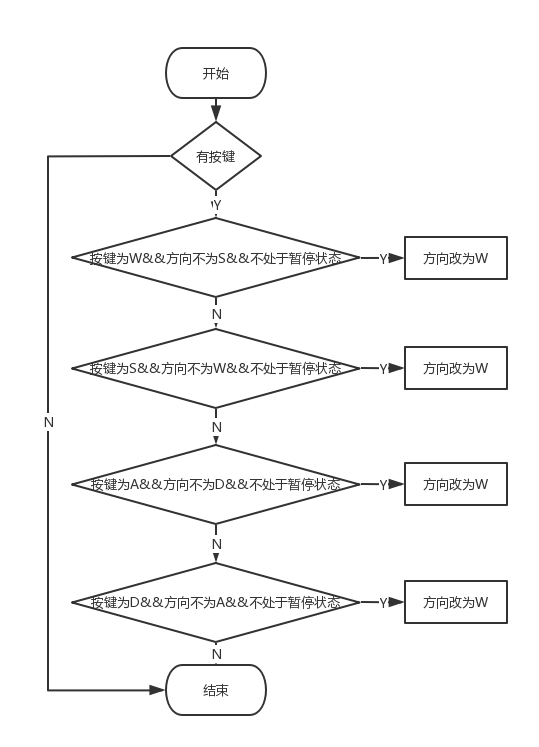
**3.3关键数据结构定义**

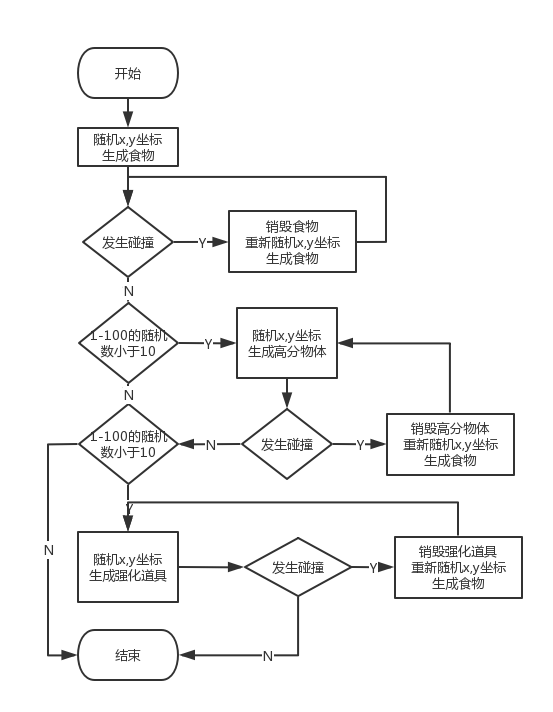
系统中最关键的数据便是坐标，在贪吃蛇中使用整数坐标方便使用蛇头通过加减一个整数step来改变坐标以完成蛇的移动，食物的显示，砖块的显示都依靠坐标来完成位置的确定。除开作品另一个重要的数据便是预制体，这是unity自带的一种数据，通过事先创造好的物体的各类属性如图片，碰撞体积，大小等等以便后续使用这些物体，在蛇身增长的时候便是通过创造已制作好的蛇身的预制体来创造蛇身再确定其位置信息，同理食物和砖块也是先创建预制体再确定坐标来完成最终的显示。

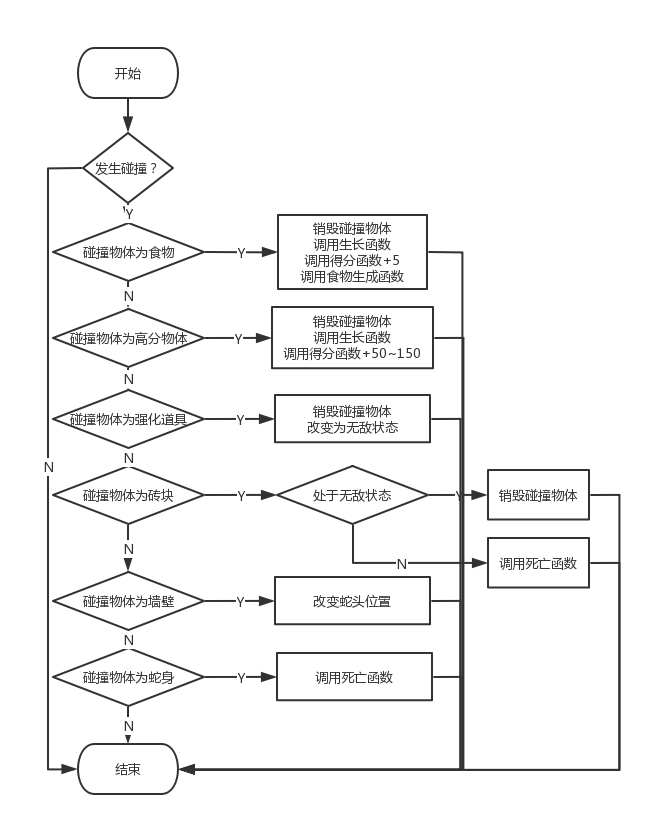
**3.4 关键算法设计**

1. 蛇头移动算法，蛇头移动算法通过按键改变蛇头的方向来起到控制蛇上下左右移动的操作。同时要判断不能180度转向，例如向右移动时向左转不生效同时要判断是否处于暂停状态如果处于暂停状态则不生效，流程图如图3-3所示。

1. 蛇身移动算法，蛇身移动算法是通过将蛇身看作一个数组，每个蛇身获得前一节蛇身的坐标来完成蛇身的移动，最前面的一节获得蛇头的位置。流程图如图3-4所示。
2. 碰撞检测算法，通过unity的刚体判断碰撞的发生，并通过对应物体的不同标签来判断碰撞的对象，并进行响应的操作，碰撞物体标签为食物，则增长蛇身，销毁碰撞物体，重新调用食物生成函数，调用得分函数。碰撞物体为高分道具，则增长蛇身，销毁碰撞物体，调用得分函数并取分数为随机的50-150分。碰撞物体为强化道具则更改蛇身状态，销毁碰撞物体。碰撞物体为砖块，判断是否处于无敌状态，处于则摧毁碰撞物体，否则调用死亡函数。碰撞物体为墙壁，则根据上下左右墙壁将蛇头的坐标变为对面墙壁的起始位置。碰撞物体为蛇身则调用死亡函数。流程图如图3-5所示。
3. 食物生成算法，随机一个食物的x,y坐标，创建食物，如果和蛇身，砖块碰撞，则销毁食物并重新随机坐标再次创建食物。zhuang再通过随机1-100小于10创建高分道具，其中通过随机函数随机其x,y坐标，并将其再十秒钟之内销毁。再通过1-100随机小于10创建强化道具同高分道具一样。流程图如图3-6所示。

****

****

****

**3.5 数据管理说明**

整个贪吃蛇只需储存分数即可，在每次死亡函数被调用时，判断得分是否大于最高分，若大于做高分则利用unity的数据存储函数，将最高分的值存入带有标签的bestscore的存储文件中，在每次UI界面利用unity自带的函数在开始时读取bestscore中的数值来显示历史记录，其他的图片，图标音效等再导入素材时导入asset文件夹中，程序运行时会根据设定好的逻辑从对应的路径中提取素材显示。

**4 实现与测试**

**4.1实现环境与代码管理**

游戏引擎采用unity2020.1.16f1c1，游戏平台采用unity hub 2.4.4。脚本语言采用c#，脚本编辑器采用Visual Studio2019。开发系统软件基于64位Windows 10家庭版，硬件CPU基于Intel core i7-8750H。使用github进行代码管理，迁入记录如图4-1所示。



图4-1 码云平台代码签入示例

**4.2 关键函数说明**

Update()函数完成蛇头的转向功能。

Move()函数完成蛇头以及蛇身的移动。

Grow()函数完成蛇身的增加啊。

Die()函数完成蛇死亡音效播放，特效显示，并存入分数判断是否产生新的历史记录。

Pause()函数完成游戏的暂停与继续。

UpdateUI(int s = 5, int l = 1)函数完成分数记录与显示。

MakeFood(int type)函数根据不同的输入值生成食物或则高分道具或则强化道具。

OnTriggerEnter2D(Collider2D collision)函数完成碰撞检测。当与食物碰撞时，调用Grow()函数增长蛇身，调用UpdateUI(int s = 5, int l = 1)函数跟新分数。并调用MakeFood(int type)函数根据概率生成食物或则高分道具或则强化道具。当碰撞到砖块或则蛇身时调用Die()函数完成蛇死亡。碰撞为墙壁时改变蛇头位置。

**4.3 测试计划和测试用例**

由于贪吃蛇代码逻辑较为简单，采用黑盒测试来检测每个功能是否正常工作即可。黑盒测试顾名思义是将程序看作一个不透明的黑盒子，在不清楚程序内部结构以及特性的情况下在程序接口进行测试，只负责检测功能是否正确只考虑程序是否能适当的接收输入数据并产生正确的输出数据。

黑盒测试最常见的测试有：功能性测试、容量测试、安全性测试、负载测试、恢复性测试、标杆测试、稳定性测试、可靠性测试等。

1. 主场景UI测试，主场景只对鼠标点击开始按钮产生反应进入游戏场景，对于其他操作不进行回应。测试时在开始按钮旁进行点击，随机按下键盘上的按键并无反应，再按下开始按钮时进入游戏场景。主场景界面如图4-2所示。
2. 游戏场景UI测试，游戏场景UI有暂停/继续按钮以及返回按钮。游戏场景如图4-3所示。在按下暂停按钮后按钮图标改变，蛇停止不同且不能操控。按下WASD键以及其他按键，发现蛇并不会有任何变化，图标也发生改变。结果如图4-4所示。按下继续键图标又变回原图标蛇也会进行移动，可以通过WASD控制蛇移动。结果如图4-5所示。
3. 蛇头转向测试，在游戏开始后分别按下WASD改变蛇头的方向，在蛇头朝向W时按下S键，蛇头朝向S时按下W键，蛇头朝向A时按下D键,蛇头朝向D时按下A键发现并不会改变方向。在此只展示一组实验结果如图4-6，4-7所示。
4. 吃食物测试，当蛇头碰撞到食物时，食物消失蛇身增长。结果如图4-8，4-9所示。
5. 边界测试，当蛇头碰撞到左边界便从右边界出，碰撞到右边界便从左边界出，碰撞到上边界便从下边界出，在此只展示一组实验结果如图4-10，4-11所示。
6. 死亡测试，当蛇头碰撞到蛇身时发生死亡结果如图4-12所示，当蛇头不出无敌状态下碰撞到砖块如图4-13所示。
7. 无敌状态测试，在吃下强化道具后碰撞砖块，使砖块被摧毁。如图4-14，4-15所示。
8. 高分道具以及强化道具计时消失测试，当强化道具和高分道具在10s中后会自动消失，在此展示一组实验结果如图4-16，4-17所示。

在生成食物，高分道具，强化道具时只能在规定的范围内产生，此时采用白盒测试。白盒测试：是一种测试用例设计方法，在这里盒子指的是被测试的软件，白盒，顾名思义即盒子是可视的，你可以清楚盒子内部的东西以及里面是如何运作的，因此白盒测试需要你对系统内部的结构和工作原理有一个清楚的了解，并且基于这个知识来设计你的用例。将坐标地址设置为随机值的极限值，即最大值和最小值生成食物，最后结果生成在范围内在此展示一组实验结果如图4-18所示。

****

图4-2 主场景

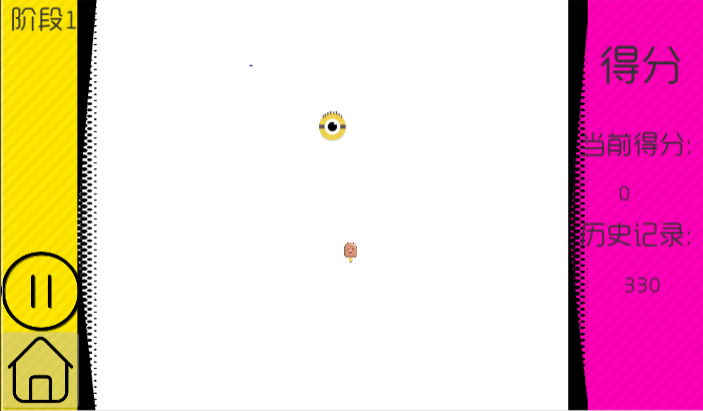
****

图4-3 游戏场景

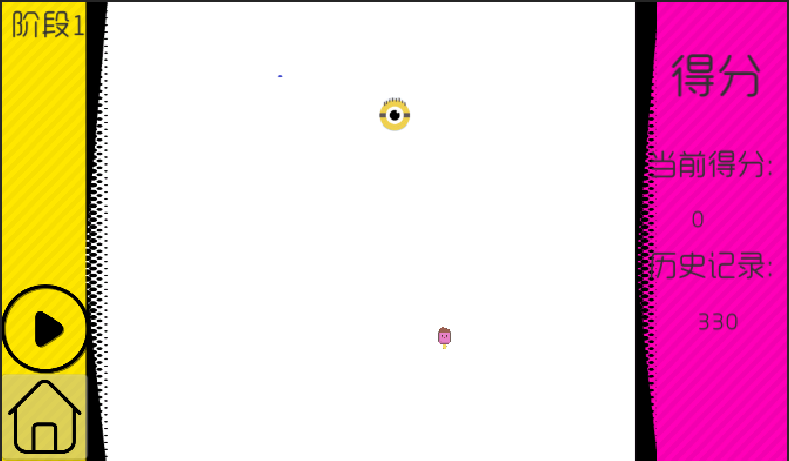
****

图4-4 按下暂停按钮图标改变无法操控

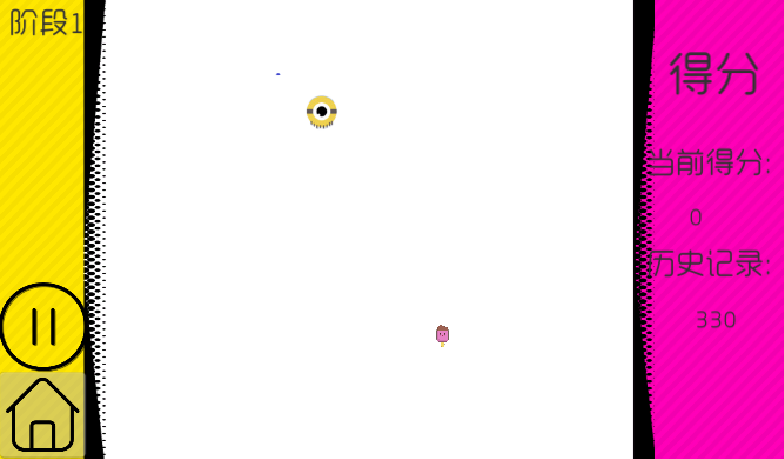
****

图4-5 按下继续按钮图标改变恢复操控

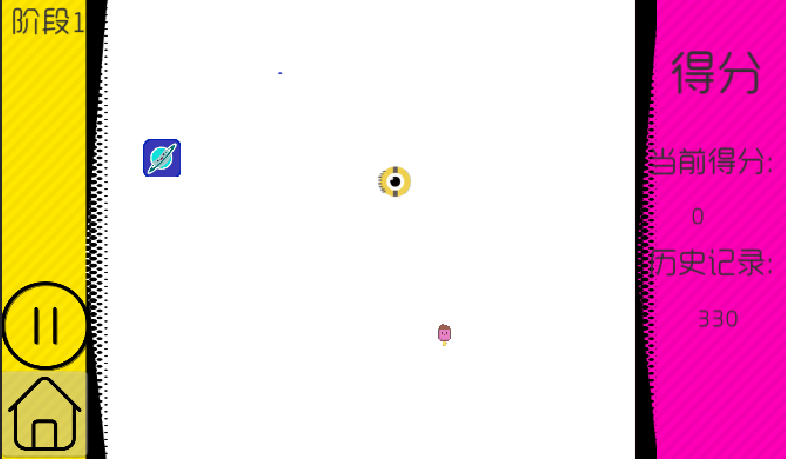
****

图4-6 按下A向左转

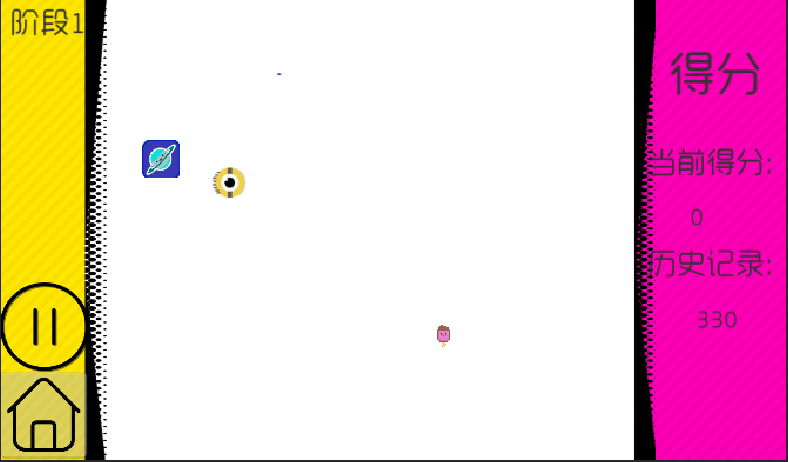
****

图4-7 按下D无法向右转

****

图4-8 蛇头碰撞到食物前

****

图4-9 蛇头碰撞到食物后重新生成食物蛇长长

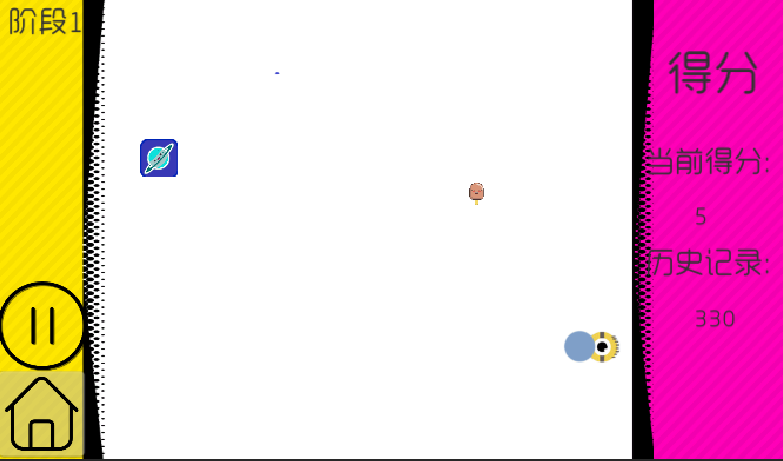
****

图4-10 蛇头碰撞右墙壁之前

****

图4-11 蛇头碰撞右墙壁之后从左墙壁出现

****

图4-12 蛇头碰撞身体后死亡

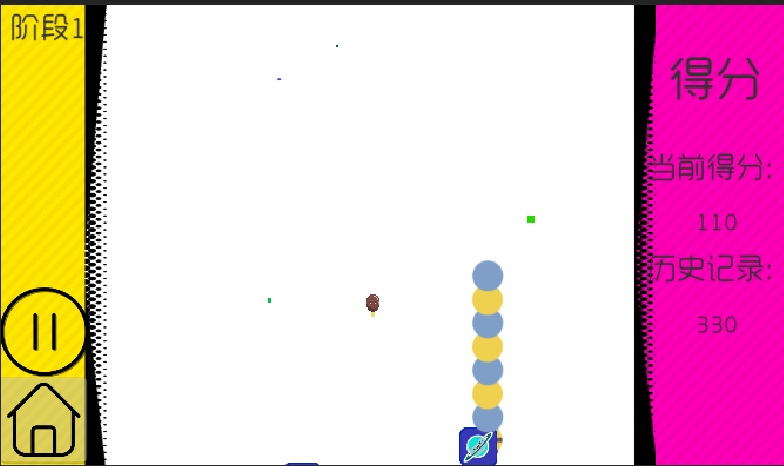
****

图4-13 蛇头碰撞砖块后死亡

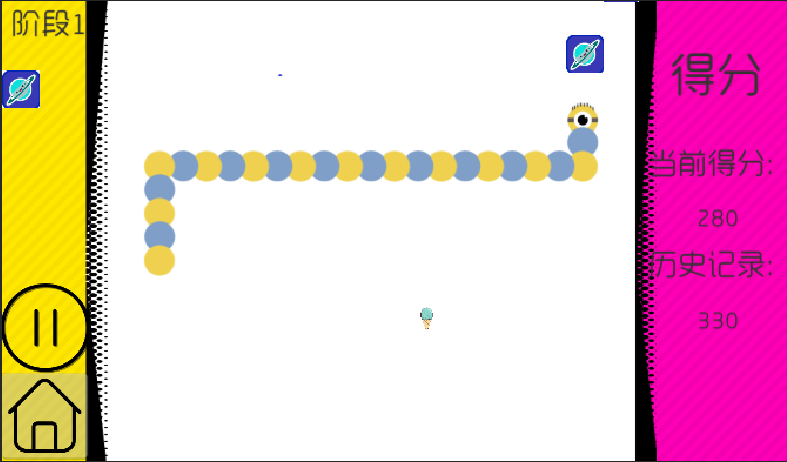
****

图4-14 蛇头无敌状态下碰撞砖块前

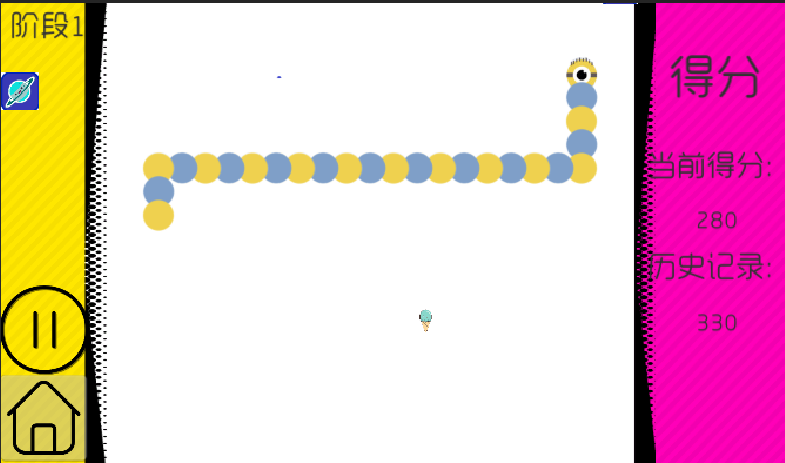
****

图4-15 蛇头无敌状态下碰撞砖块后摧毁砖块

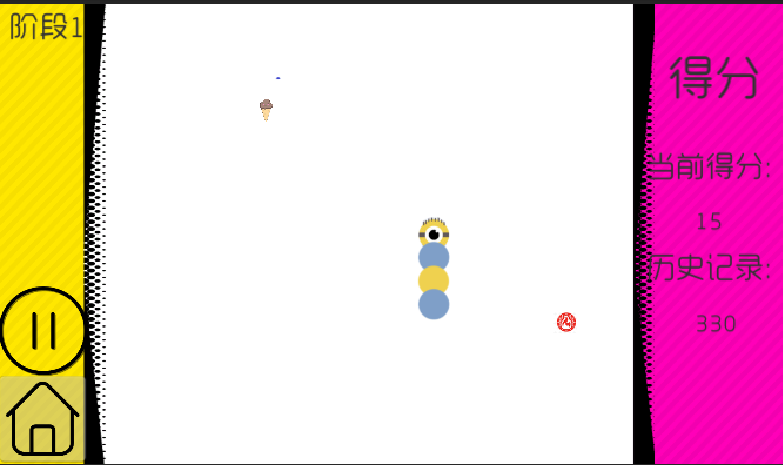
****

图4-16 强化道具出现

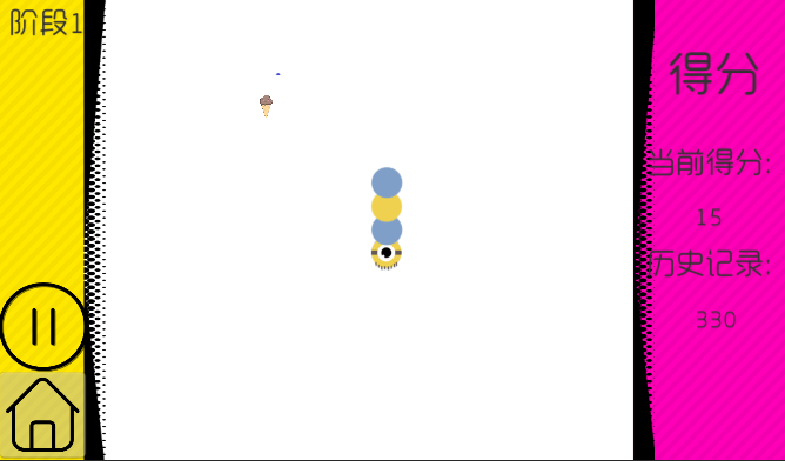
****

图4-17 强化道具出现10s后消失

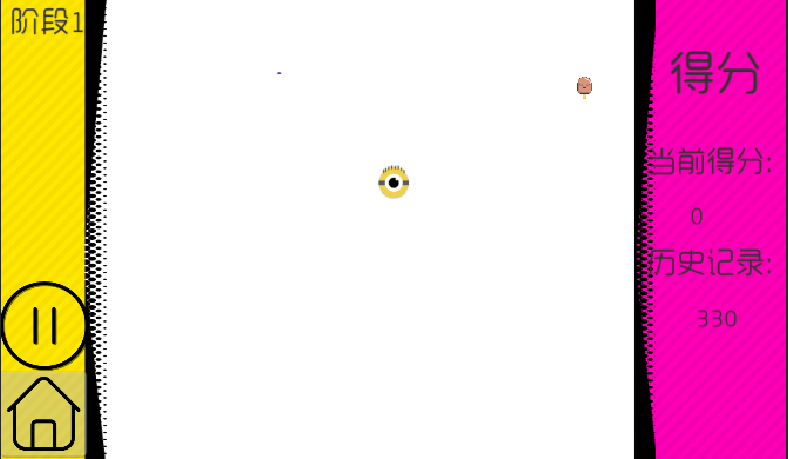
****

图4-18 食物的极限位置

**4.4 结果分析**

玩家能通过按钮切换场景，并能进行暂停，也可以看到展示的实时的得分以及历史记录表面UI系统达到目标。同时玩家可以通过WASD控制蛇的移动，可以进行吃食物增长身体，碰到墙壁时从另一边钻出，在碰撞到蛇身或则在非无敌状态下碰到砖块死亡，以及无敌状态下碰撞到砖块摧毁砖块。食物也会生成在指定范围内，高分道具以及强化道具会限时消失。最高分会被记录。测试结果达到预期目标。

**5 总结**

**5.1 用户反馈**

包括用户的使用记录，照片，视频等。

**5.1 全文总结**

（1）完成对整个贪吃蛇的构想。

（2）完成对贪吃蛇基础功能的实现，例如上下左右移动，蛇身能跟随蛇头移动，食物的随机生成，吃食物会长长身体，在碰撞到身体时会死亡等等基础功能。

（3）对其他有趣功能的实现，例如限时的高分道具以及强化道具，随时间推移不断变快的蛇的移动速度，四周墙壁的穿越设定，随时间变化不断变密集的砖块设定以及在摧毁砖块的强化道具的设定。

（4）UI界面的设计

（5）UI界面的实现以及图片及其它资源的实现例如给游戏加上背景图片，给蛇头蛇身食物等添加合适图片，给按键添加合适的图标，采用更加美观的字体以及其他美术方面的改进。

（6）对整个程序的测试与改进。

（7）将程序发布给亲朋好友进行用户测试。

（8）撰写实验报告。

**6 体会**

在开始编写贪吃蛇游戏时我才用了纯c++语言的实现方式，通过graphics图形库来实现贪吃蛇，编写难度极大并且并不美观，只能用方块的颜色来区分蛇头蛇身食物，并且难以调整整体游戏布局只能通过一点一点的尝试来不断完善布局对碰撞的处理十分困难只能在固定的点上判断碰撞，并且检测复杂。

后来采用了专业的游戏制作引擎unity来完成贪吃蛇的设计，unity很直观的就能完成整体界面的设计。Unity通过给物体上脚本来控制物体执行逻辑。可以将物体的属性和其执行逻辑完全分开来处理，可以用类似画图的方法改变物体的形状大小图片等，脚本代码只需处理其行动逻辑即可。同时unity带有刚体属性，在给物体附上刚体后就能自动进行碰撞检测，代码只需完成对不同碰撞发生后的处理逻辑即可，极大减少了实现难度，也给我实现类似高分道具能其他游戏内容创造了可能性。同时unity自带有数据存储和读取函数，只用调用对应的函数就能存入或则读取历史记录十分便利。

因此我深刻的意识到虽然用c++语言可以实现近乎所有程序，但对应特殊的程序如游戏来说使用游戏引擎会极大的减少工作难度和工作量，也能做的更加的美观。将物体的属性和行动逻辑分开会极大的降低代码的复杂度，也能提升物体的精美程度。

对于整个软件工程项目来说，我明白了不光只需要写好代码，同样的对目标人群的需求的调查一样重要，一项工程不只有代码部分连带从立项到开发到后期检查都同样的重要。同时通过系统结构的拆分能够使程序员对整个项目的构架更为清晰，更容易进行错误排除和改进。

对于我个人来说，这是第一次制作一款能过可以玩的游戏，真正意义上有游戏乐趣的游戏，极大的增加了我个人的成就感，感觉自己学习的内容有一定的作用，能完成一个小游戏。提高了我的自信心，同时也学会了使用unity这一十分遍历的游戏制作引擎。