****

**2017—2018学年第2学期**

**《程序设计实训》课程报告**

题目：太空战机2.0

机房名称 文理楼102

时 间 2018年5月1日至2018年6月24日

开课系室 计算机应用技术系

指导教师

小组成绩（百分制）

成员成绩（百分制）

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 成绩 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

2018年 6 月 24 日

# 说明

1. 《程序设计实训》课程作为《程序设计》课程的后续课程，旨在对学生进行程序设计能力的综合训练，以提高学生问题求解能力、创新能力和团队协作能力。所有参加本课程的学生都必须参与程序设计和实践报告撰写工作。
2. 《程序设计实训》结束后，需要提交**实践报告（电子及打印版）、程序源代码、答辩PPT、演示视频**。未提交相关资料者不得参加成绩评定。
3. 指导教师需对学生实践报告填写评语并在相应位置填写成绩并签字。
4. 实践报告要求严格按照本报告模板撰写，条理清晰、内容详尽、论述准确，字数不少于5000字。本表格可根据内容增加页面，但要保证格式一致。

|  |
| --- |
| 题目 太空战机2.0 |
| 一、问题提出 |
| 基于所学程序设计课的基础上，在程序设计实训期间，我们掌握了利用Funcode制作小游戏的技能。  我们组做的游戏是【太空战机2.0】，选择这款游戏的原因是它本身具有一定的提高和挑战性，另外，我们组认为这款游戏可改进的地方有很多，因此这款游戏的制作无疑是充满乐趣和探索的。  我们认为，【太空战机（模板）】是一款简单易懂的射击类游戏，那么，如何才能将它改造的更加高级，更加吸引玩家注意力呢？以这两个问题为基础，我们在游戏的各方面针对游戏外观和游戏性能两方面做了改进。 |
| 二、概要设计 |
| 2.1功能描述 **1、游戏规则介绍**  ①太空战机在太空中飞行时，太空飞机发出的子弹与敌机子弹碰撞后可相互抵消；  ②太空战机与敌机发生碰撞时，敌机被消灭，太空战机血量减少；  ③敌方BOSS的子弹无法与太空战机抵消，只能躲避；  ④未被消灭的敌机在与游戏边界碰撞后会重新弹回，直到被消灭为止  **2、游戏玩法简介**  ①空格键开始游戏，“空格键开始”消失，载入地图，将太空战机设置在地图左侧边界附近。  ②游戏时间不限，玩家血量为0时游戏结束。  ③地图下角设置两行数字，第一行代表玩家当前分数，第二行代表历史最高纪录，若一局结束玩家总分数高于历史最高纪录，则刷新第二行分数。  ④太空飞机击中敌方子弹分数加10，击中敌方战机分数加50，击毁敌方BOSS分数加500。  ⑤玩家用WSAD控制太空战机上下左右移动，空格键发射子弹。  ⑥敌方战机数量和产生时间随机。 2.2流程图 **1、游戏框架**  太空战机  游戏初始界面功能  游戏功能  游戏中界面功能  子弹特效与爆炸特效  显示当前分数和历史最高分  WASD控制上下左右，空格键发射子弹，消灭敌机  按空格键开始游戏，左下角第一行分数显示为零  显示初始界面  **2、游戏功能优化**  太空战机  性能  特效  利用边界反弹使未消灭敌机继续留在游戏界面内，增大游戏难度  游戏可玩性增强  太空战机子弹伤害值增大  采用全新背景，敌机类型增加  BOSS子弹无法消灭，冲击波特效更具挑战性！  子弹外观改变，玩家子弹更炫酷！ |
| 三、详细设计 |
| **1、主函数模块**  主函数模块控制游戏的主循环，此函数被不停的调用，引擎每刷新一次屏幕，此函数就被调用一次。主要用来处理游戏的开始、进行和结束等各种状态。  **2、初始化模块**  主要是实现对游戏的初始化。每局开始前进行初始化清空上一局数据，消除对新一局游戏数据的干扰。  **3、玩家战机功能模块**  通过dOnKeyDown、dOnKeyUp、LoopTick等函数控制玩家战机的移动与子弹发射。   1. 判断键盘输入 2. 设置精灵速度   **4、敌方战机功能模块**  1）实现敌方战机的生成、移动与发射子弹。且战机从视界边界右侧的竖直随机位置生成，生成时间在20秒到30秒之内，速度方向分为水平方向与竖直方向，速度大小随机。子弹从敌机诞生之后固定时间发射。  2）按照原版设计，若敌机在视界范围之内未被子弹消灭，则其将飞出视界边界。但是我们小组改良后的版本将其设定为不会飞出视界边界，存在至被玩家战机消灭或者游戏结束为止。  3）本小组的创新亮点在于为敌方战机设定了三种小型战机外观与一种大型“BOSS”战机外观，大大提升了游戏的趣味性与观赏性。其中对“BOSS”战机的属性进行了调整，使其近似于玩家战机，需要多次被击中才能被消灭，稍微增加了游戏的难度。  **5、精灵碰撞模块**  本模块主要用来实现各模块之间的碰撞：战机与子弹之间的碰撞、子弹与子弹之间的碰撞以及战机与战机之间的碰撞。特别指出，我们对原版游戏中提供的子弹素材进行了修改，使用了多种子弹外观，使游戏画面与碰撞效果看上去更加的炫酷。  1）玩家战机的初始生命值可以通过修改代码来由玩家决定，每当玩家战机与敌机普通子弹发生一次碰撞，玩家战机生命值减少一百；当与“BOSS”战机发射的光波型子弹发生碰撞，玩家战机生命值减少二百。直至玩家战机生命值降低至零时游戏结束。  2）当玩家战机子弹与敌机普通子弹发生碰撞时，二者同时消失。但是玩家战机子弹无法与“BOSS”子弹相互抵消，玩家战机只能通过移动来躲避它。  3）当小型敌机与玩家战机子弹发生碰撞时，二者同时消失。当“BOSS”型敌机与玩家子弹发生碰撞次数少于10次时，敌机保留，子弹消失。当碰撞次数达到10次时两者同时消失。  4）当玩家战机与敌机碰撞时，二者相对于对方相当于子弹，遵循上面1）与2）的规则。  5）每局游戏开始时玩家得分为零。子弹相互抵消一次，玩家得分加10分；消灭一架小型敌机，玩家得分加50分；消灭一架“BOSS”型敌机，玩家得分加500分。当游戏结束时，得分清零，并将本次得分与最高得分记录（初始设为零值）进行比较，若是高于记录则用本局得分数值代替最高得分记录。  **6、链表操作模块**  敌机与子弹的生成都依靠此模块来实现。精灵生成时添加节点，游戏结束时删除节点。 |
| 存在问题及解决办法 |
| 1. 对于Funcode运行操作不熟悉，最初进度缓慢   解决办法：将十个Funcode模板按照步骤重新制作一遍，了解程序结构，掌握主要制作方法，提炼需要的内容。   1. 背景设置为滚动图，在运行的时候会出现部分图像拉伸的情况   解决办法：调试了许多次以后依旧没有找到解决方法……  3、在编写CreatBullet2函数（BOSS发射的特殊导弹）时，在设定发射时间间隔以及碰撞效果时遇到了无法生效的困难  解决方法：多次调试时候解决了这个难题。  4、空中战机在碰到BOSS无法消除的光波子弹后战机损坏特效持续时间太长  解决方法：多次调试，还是无法达到最优效果。  5、PlaySound函数无法使用  解决方法：直接在FunCode界面进入游戏时播放音乐。 |
| 五、运行效果 |
| 初始1.PNG  图1 游戏制作完成效果如图  运行界面.PNG  图2 游戏初始界面  历史记录为0，当前分数为0，按空格键开始游戏。  特效优化---子弹.PNG  图3 玩家战机与敌方战机，大BOSS同屏。  如图3所示，我们对子弹进行了造型优化，玩家战机用易于区分的黄色光波，BOSS子弹用覆盖范围较广的蓝色光波，敌方三种普通类型战机的子弹沿用初始素材，提高区分度。  BOSS的光波子弹覆盖范围广，杀伤力强且无法抵消，针对这种情况，玩家只能躲避，增大了游戏的可玩性。  玩家中弹展示.PNG  图4 玩家中弹展示  当玩家战机中弹时，在玩家战机附近产生爆炸特效，并且玩家战机有机身爆炸并掉渣的特效。  此时，战机被普通子弹击中，则血量减100，被敌方BOSS击中，血量减200，当玩家血量减至0或小于0时，游戏结束。  历史最高分.PNG  图5 历史最高记录  游戏刚开始时，当前分数记录为0，第二行显示历史记录最高分，当前分数超过历史记录而游戏没有结束时，第二行记录不变，在此轮游戏结束后再进行变更。  设置历史记录最高分是为了增加游戏的挑战性，吸引更多的玩家加入游戏。  破纪录.PNG  图6 记录的连续刷新  当历史记录被刷新后，玩家仍然可以继续游戏，制造更高的记录。  敌机反弹 BOSS子弹.PNG  图7 敌机在游戏边界处的反弹  如图7，敌方战机在未被消灭时，不会从边界处消失，而是利用了“BOUNCE”  设置使敌方战机弹回游戏界面，如果不及时消灭敌方战机，则越积越多，从而增加了游戏的挑战性。  暴击.PNG  图8 暴击  如图8，蓝色光圈为BOSS子弹，不可消除，在我方战机与BOSS子弹碰撞后，会出现如图大面积掉落战机零件的画面，此时玩家战机血量减2 0 0，因此，为避免不必要的牺牲，玩家会选择躲避BOSS子弹，我们将BOSS子弹尺寸设置为占游戏界面三分之二，减小玩家逃脱率，从而达到增大游戏挑战性的效果。 |
| 六、小组分工 |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | 学号 | 姓名 | 报告分工 | 程序分工 | 贡献排序 | |  |  | 版面设计 | 主要代码编写 | 1 | |  |  | 概要、详细设计 | 素材选取、创新 | 2 | |  |  | 流程图、图片处理 | 创新讨论 | 2 |   表1 分工表 |
| 七、实践感想、经验及教训 |
| 1.实践感想  通过程序设计实训课程，我们小组充分积累了Funcode软件的使用经验，加深了一些基本函数的掌握，对于一般函数，我们做到了熟记，会用两点，对于其他FunCode函数，我们组做到了会充分应用的基本要求。  对于FunCode游戏设计，我们组做了充分的准备，并且从创新的角度发现问题，以原版游戏作为模板，添加新的特效，精灵，制作创新版的游戏，最终将【太空战机】游戏按照我们最初的创新思路一步一步完成了，对于我们来说是一个非常有意义的历练。  通过这次合作，我们组的三个成员都意识到了团队合作的重要性，对于一个好的制作团队来说，最大的问题在于内部分工和讨论。讨论可以促进团队的合作团结精神，优化团队思想，并且制作出更加优秀的作品，然而精确有效的工作分配是讨论结果的具现化，如果工作量分配不均匀或分配方式有问题，都有可能给制作团队带来巨大的损失，因此，在进行团队内部分工时，我们组根据个人的特长进行分配，不分工作的重要性，不计个人荣誉，最终将我们的作品完美的具现化出来，这是令人非常欣慰的事情。我们认为，每个人做的贡献都是一样多的，正是因为我们都有这样的想法，所以才能将我们的作品完善的更加全面。  起初我们对FunCode软件的认识是非常浅薄的，但是为了更好的完成小组作品，我们组一起阅读了老师提供的参考资料，一点一点系统地建立了FunCode与我们所学的C语言的联系。因此我们认为自学能力也是我们作品成功的一个主要原因。  2.实践经验  通过程序设计实训课程，我们收获了非常多的经验，尤其是在问题处理方面。  当游戏运行出问题时，要从头开始。首先要想程序出了什么问题，对程序进行不断的调试以后，再次运行，观察问题是否能消除，如果不能，还要从其他方面进行测试，然后再进行反思和总结。这是一个漫长而枯燥的过程，需要非常大的耐心，许多无法检查出来的错误要反复调试才能得出结果。  想问题的时候要从多个角度考虑。我们从固定的套路里获得的知识，不一定适用于所有的对象，因此这需要我们用辩证的眼光去看待资源的用处，理性处理问题，真理永远是实践的产物，资料是不可能原样照搬的，必须要加入自己的思路，才能形成新的作品。如果思考的方式和角度过于单一，就无法制作出优秀完美的作品，有时甚至连简单的问题也难以解决。  要在实践中不断地突破自我。刚开始的时候我们连海底世界的完成都非常困难，但是通过不断的练习积累了许多经验，最后制作出了具有提高能力的【太空战机】游戏，并通过不断的讨论，创新，设计，使游戏的性能和外观得到了巨大的改善。  要对每次试验都报以严谨的态度。每一次不起眼的试验都会成为一次实践数据，是得出经验的基础，因此，如果不对每次有益的改动进行保存，很可能会失去优化作品的机会。  3.实践教训  在我们的制作过程中出现了许多技术性错误和常见性错误，我们通过这些错误总结教训，减少了再次犯错的可能性。  每次做完程序要记得保存，不用的素材和附件不要过早删除，等作品完成后再统一处理。当我们的游戏完成了百分之八十左右时，由于过早删除了部分副本，导致游戏部分处理过的细节全部被撤销，为此，我们重新制作了游戏。这个非技术性错误给我们的教训是：不可操之过急。许多需要多次使用的素材和附件，以及游戏主干部分的东西一定要进行妥善保存和备份，这样才能不影响后续工作的正常进行。  不要对无法涉及的内容过分追求，本末倒置。在我们游戏制作的过程中，对于一些背景滚动图和特效的不完美，我们进行了数十次的修正和改善，但是效果并不明显，并且浪费了许多时间，有一次直接导致了游戏恢复初始状态。因此我们认为，对于一些不影响游戏正常运行的细小瑕疵，如果最初的几次修改无法完善，那么应该暂时忽略，不要过多的逗留。  制作游戏时，应该有一套完整的思路，细节的处理要留在最后。不能因为某些细节的处理影响了整个游戏的制作进度。对于游戏程序的编写，应该先编好主要内容，测试无误以后再添加创新内容，否则可能导致因代码过乱而使游戏制作的失败。  在一个团队制作游戏的过程中，要有选择性的吸纳资源，要理性的进行讨论和设计，必要时应该有所舍弃。因为我们能力和人力的有限，我们的游戏作品不会和正规游戏一样精致，这个时候不能急于求成，过于追求所有细节的完美刻画，应该一起讨论方案的可行性，选择必要的，舍弃不必要的，减轻压力，这样才能增强团队协作能力，制作出让所有队员满意的作品。  学生签字： |
| 指导教师评语  指导教师签字： |