**电子科技大学信息与软件工程学院**

**实 验 报 告**

**（实验）课程名称 软件工程基础**

**电子科技大学教务处制表**

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：张健顺 学 号：2015220103022 指导教师：郭建东**

**实验地点：信软楼西304/305 实验时间：2017.05.21**

**一、实验室名称：**基础实验室

**二、实验项目名称：**质量保证

**三、实验学时：4**

**四、实验原理：**

单元测试（unit testing），是指对软件中的最小可测试单元进行检查和验证。对于单元测试中单元的含义，一般来说，要根据实际情况去判定其具体含义，如C语言中单元指一个函数，Java里单元指一个类，图形化的软件中可以指一个窗口或一个菜单等。总的来说，单元就是人为规定的最小的被测功能模块。单元测试是在软件开发过程中要进行的最低级别的测试活动，软件的独立单元将在与程序的其他部分相隔离的情况下进行测试。

系统测试（System Testing），是将已经确认的软件、计算机硬件、外设、网络等其他元素结合在一起，进行信息系统的各种组装测试和确认测试，系统测试是针对整个产品系统进行的测试，目的是验证系统是否满足了需求规格的定义，找出与需求规格不符或与之矛盾的地方，从而提出更加完善的方案。系统测试发现问题之后要经过调试找出错误原因和位置，然后进行改正。是基于系统整体需求说明书的黑盒类测试，应覆盖系统所有联合的部件。对象不仅仅包括需测试的软件，还要包含软件所依赖的硬件、外设甚至包括某些数据、某些支持软件及其接口等。

1. **实验目的：**
2. 掌握软件编码规范
3. 掌握单元测试的方法，包括白盒测试和黑盒测试
4. 能对实际程序进行单元测试，能够撰写简单的测试报告
5. **实验内容：**
6. 编写三角形判断程序，并进行黑盒测试
7. 编写分数统计程序，并进行白盒测试
8. 选择系统中的2个关键模块进行编码实现
9. 使用白盒或黑盒方法测试实现的模块
10. **实验器材（设备、元器件）：**

一台PC机，win xp系统

1. **实验步骤：**

#### 步骤1：输入三个整数a、b和c分别作为三角形的3条边，判断由这3条边构成的三角形类型是：等边三角形、等腰三角形、一般三角形、或非三角形（即不能构成一个三角形）。

① 编写该程序，注意编程规范

② 按照等价类划分方法设计测试用例，并进行测试

③ 按照边界值方法设计测试用例，并进行测试

#### 步骤2：输入不超过50个数（以-1作为输入结束标志），计算其中有效的学生分数的个数，及有效分数的总分和平均值。

① 编写程序，注意编程规范

② 导出程序的控制流图，确定独立路径集合（基本路径集合）

③ 为每条路径设计测试用例，并进行测试

#### 步骤3：基于实验2的结果，选择2个关键模块进行编程实现。

#### 步骤4：使用白盒或黑盒方法测试实现的模块。

**九、实验数据及结果分析：**

（1）编写一个java程序，输入3个整数a，b和c，判断由这三条边构成的三角形是：等边三角形，等腰三角形，一般三角形或不能构成三角形。实现代码如下：

|  |
| --- |
| import java.util.Scanner;  public class Trigangle {  public static void main(String[] args){  int a,b,c;//三角形的三边  for(;;){  System.out.println("请输入三个正整数：");  Scanner in = new Scanner(System.in);    a = in.nextInt();  b = in.nextInt();  c = in.nextInt();  if(a<=0||b<=0||c<=0){  System.out.println("错误！输入了负数！");  };    if((a+b)>c&&(a+c)>b&&(b+c)>a){  if(a == b&&b==c&&a==c){  System.out.println("该三角形为等边三角形！");  }  else if(a==b||b==c||a==c){  System.out.println("该三角形为等腰三角形！");  }  else{  System.out.println("该三角形为一般三角形！");  }  }  else{  System.out.println("无法构成三角形！");  }  }    }  } |

（2）等价类划分，如表8-1。

表8-1 等价类划分表

|  |  |  |
| --- | --- | --- |
| 输入条件 | 有效等价 | 无效等价 |
| a,b,c是否为正数 | （1）a>0,b>0,c>0 | （2）a,b,c其中一个或多个为负数 |
| a,b,c是否为整数 | （3）a,b,c为整数 | （4）a,b,c其中一个或多个为分数 |
| a,b,c是否构成三角形三边 | （5）a+b>c,a+c>b,b+c>a | （6）其中一个或多个不满足 |
| 三角形是否特殊 | （7）a=b，a=c，b=c满足一项（8）a=b，a=c，b=c满足多项 | (9)a!=b,b!=c,a!=c |

（3）等价类测试用例，如表8-2所示，等价类测试结果如图8-1所示。

表8-2 测试用例表

|  |  |  |
| --- | --- | --- |
| 用例 | 输出 | 包括的等价类 |
| a=6,b=6,c=6 | 三角形为等边三角形！ | 1，3，5，8 |
| a=6,b=6,c=8 | 三角形为等腰三角形！ | 1，3，5，7 |
| a=6,b=7,c=8 | 三角形为一般三角形！ | 1，3，5，9 |
| a=1,b=2,c=4 | 无法构成三角形！ | 1，3，6，9 |
| a=-6,b=6,c=5 | 错误！输入了负数,无法构成三角形！ | 2，3，6，9 |
| a=6.6,b=7,c=8 | 错误报告 | 1，4，5，9 |



8-1（a） 等价类测试结果



8-2（b）等价类测试结果

（4）边界值测试用例，如表8-3所示，边界值测试结果如图8-2。

表8-3 边界值测试用例

|  |  |
| --- | --- |
| 边界测试用例选取 | 输出值 |
| a=0,b=5,c=6 | 错误！输入了负数！ 无法构成三角形！ |
| a=-1,b=5,c=6 | 错误！输入了负数！ 无法构成三角形！ |
| a=121,b=122,c=121 | 该三角形为等腰三角形！ |
| a=5,b=5.01,c=4.99 | 报错 |
| a=1,b=1,c=1 | 该三角形为等边三角形！ |
| a=9999,b=9999,c=9999 | 该三角形为等边三角形！ |



图8-2 边界值测试结果

**实验2：**

（1）编写一个C程序，输入不超过50个数（以-1作为输入结束标志），计算其中有效的学生分数的个数，及有效分数的总分和平均值。程序如下所示：

|  |
| --- |
| #include<stdio.h>  #include <math.h>  #define N 51  int main(void)  {  int i,num=0;  int a[N];  float sum = 0;  printf("请输入成绩（少于50份），并且以‘-1’结束您的输入。\n输入成绩：");  for (i = 0; i < N; i++) //录入成绩  {  scanf\_s("%d", &a[i]);  if (a[i] == -1)  {  break;  }  }    for (i = 0; i < N; i++) //计算成绩  {  if (a[i] != -1 && a[i]>-1 && a[i] < 101)  {  sum = sum + a[i];  num = num + 1;  }  else if (a[i] == -1)  {  break;  }  }  printf("\n");  printf("有效的学生人数为:%d。\n", num);  printf("他们的平均成绩为:%.2f。\n", sum / num);  return 0;  } |

（2）绘制程序流程图，如图8-3所示，由流程图得基本路径图如8-4所示

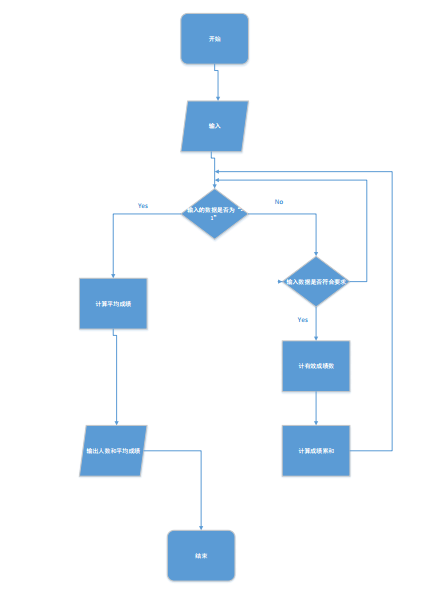


图8-3 程序流程图

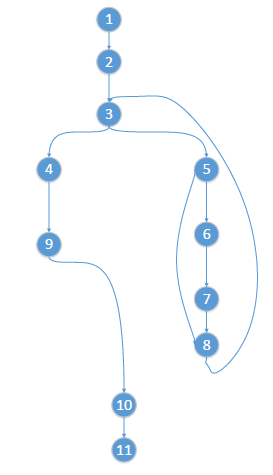


图8-4 基本路径图

（3）为每条路径设计测试用例，并进行测试得到结果，如下表：

Path1：1-2-3-4-9-10-11

|  |  |
| --- | --- |
| 测试用例序号 | 1 |
| 测试目的及说明 | 测试独立路径Path1 |
| 测试输入 | -1 |
| 期望测试结果 | 有效的学生人数为:0。  他们的平均成绩为:0.00。 |
| 判断准则 | 与期望结果是否相同 |
| 实际测试结果 | 有效的学生人数为:0。  他们的平均成绩为:0.00。 |
| 测试日期 | 2017/6/4 |
| 测试评价 | 通过测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |

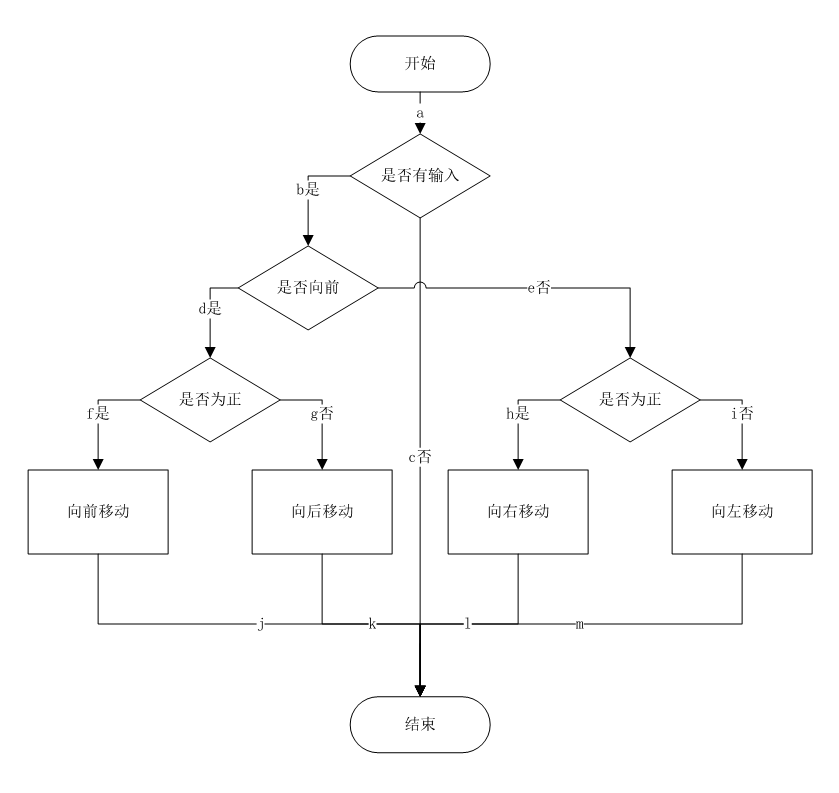
Path2：1-2-3-5-8-3-4-9-10-11

|  |  |
| --- | --- |
| 测试用例序号 | 2 |
| 测试目的及说明 | 测试独立路径 |
| 测试输入 | -100 150 -1 |
| 期望测试结果 | 有效的学生人数为:0。  他们的平均成绩为:0.00。 |
| 判断准则 | 与期望结果是否相同 |
| 实际测试结果 | 有效的学生人数为:0。  他们的平均成绩为:0.00。 |
| 测试日期 | 2017/6/4 |
| 测试评价 | 通过测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |

Path3：1-2-3-5-6-7-8-3-4-9-10-11

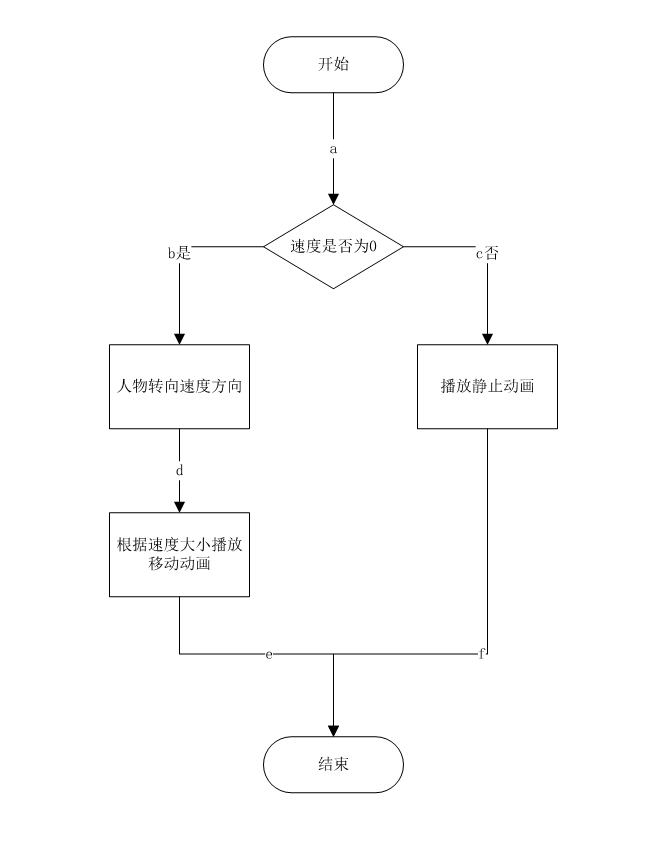
|  |  |
| --- | --- |
| 测试用例序号 | 3 |
| 测试目的及说明 | 测试独立路径 |
| 测试输入 | 100 90 50 60 50 60 70 -1 |
| 期望测试结果 | 有效的学生人数为:7。  他们的平均成绩为:68.57。 |
| 判断准则 | 与期望结果是否相同 |
| 实际测试结果 |  |
| 测试日期 | 2017/6/4 |
| 测试评价 | 通过测试 |
| 问题标识 | 无 |
| 问题处理 | 无 |

实验3



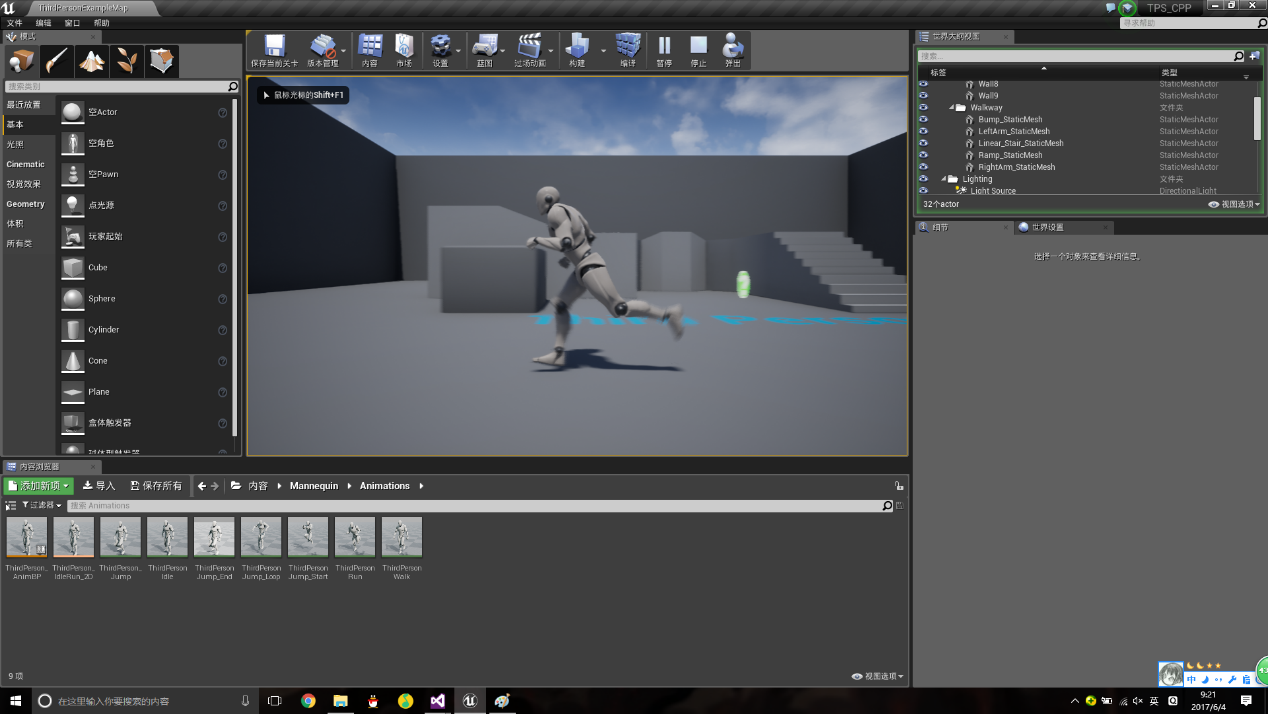
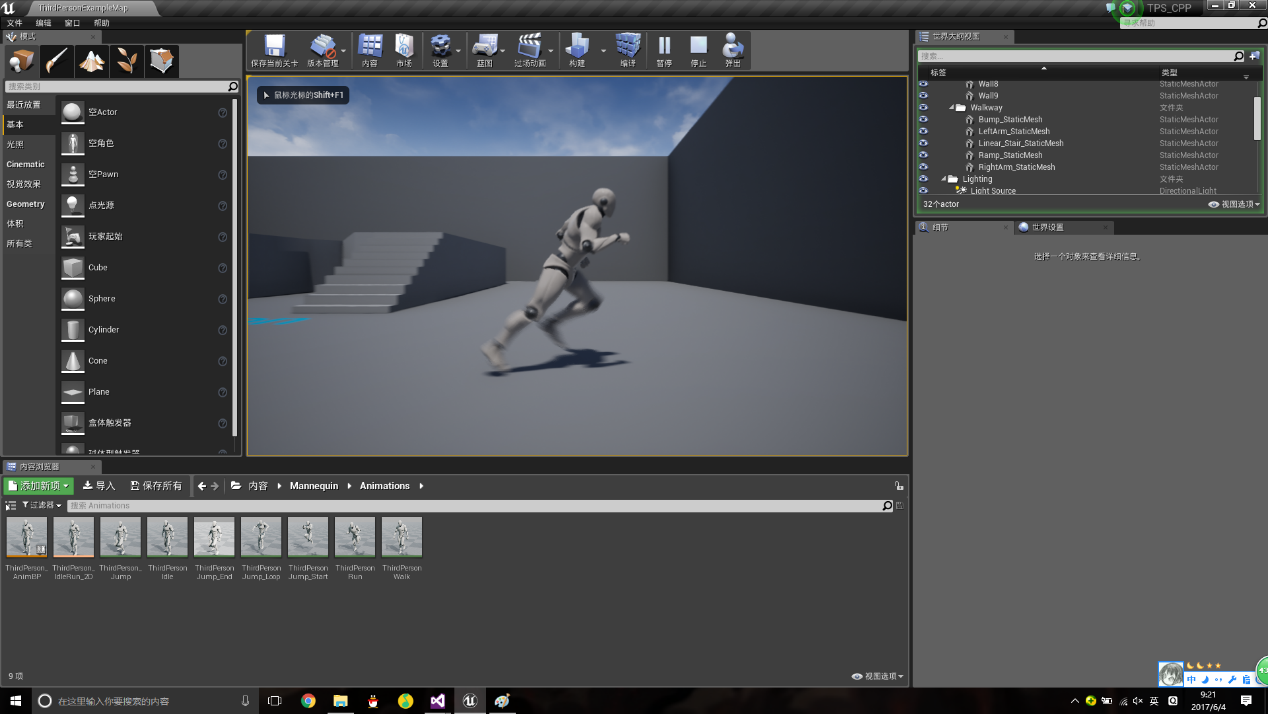
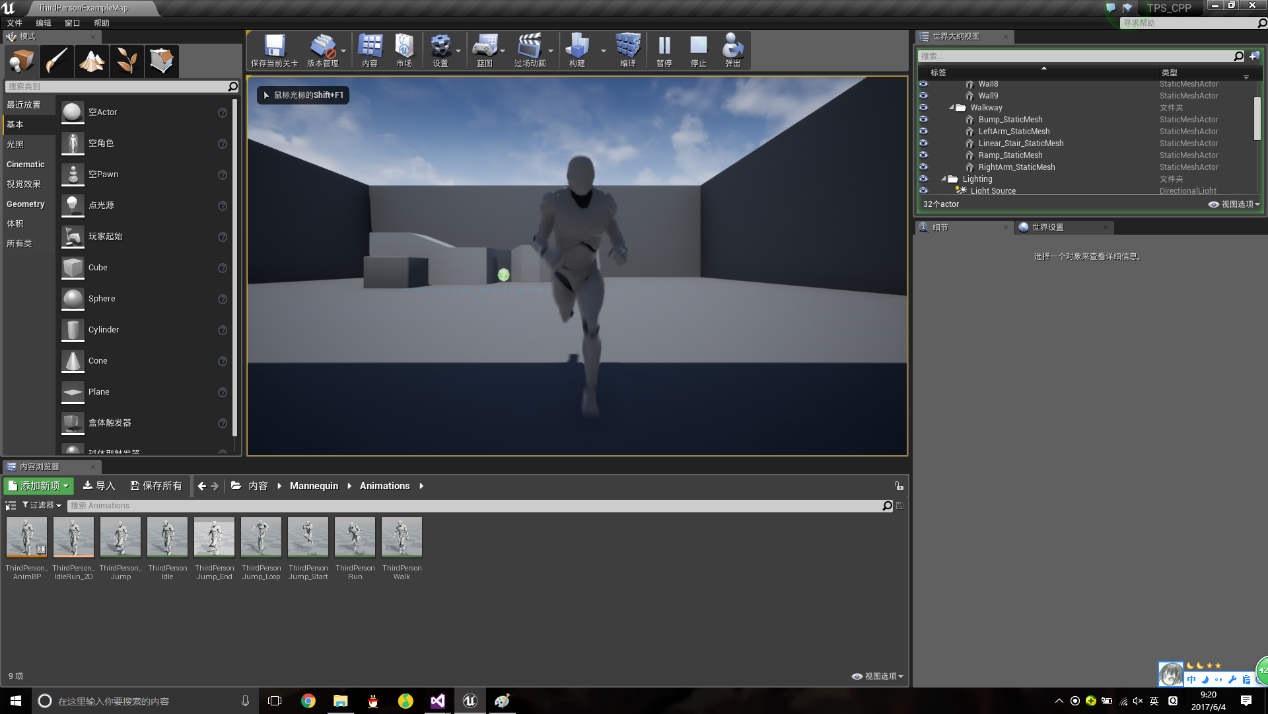
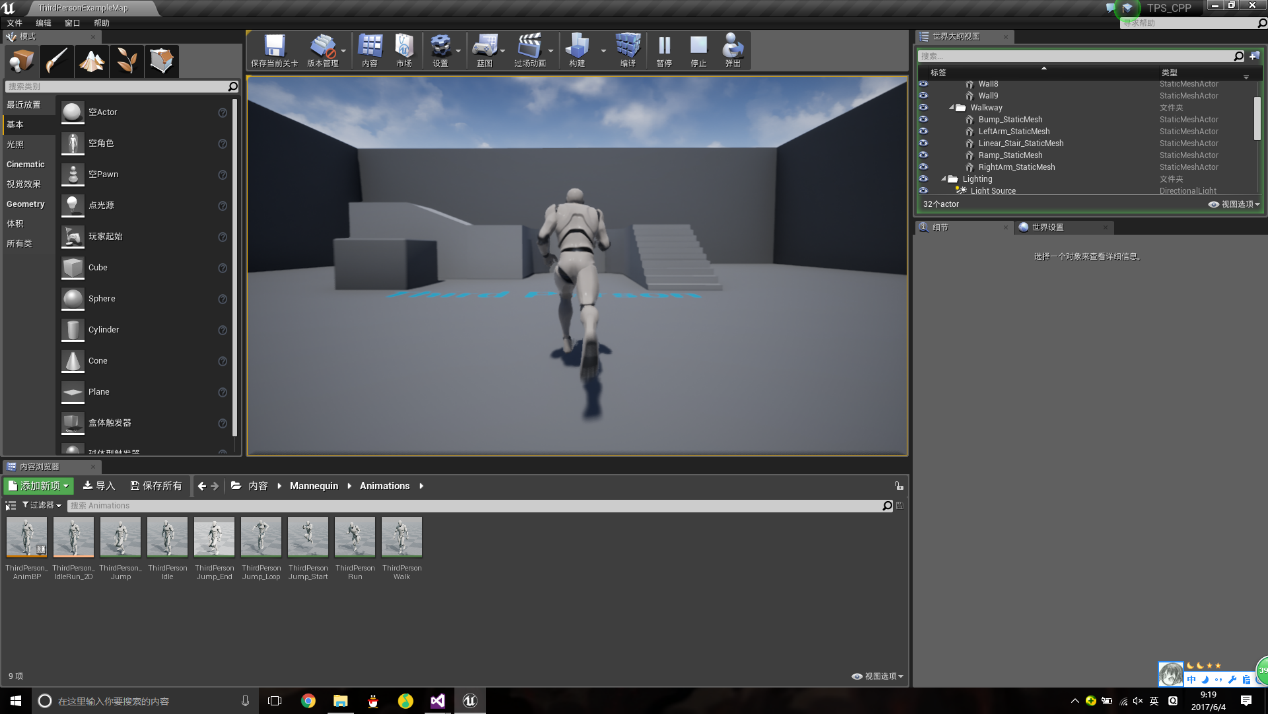
角色移动流程图

|  |  |  |
| --- | --- | --- |
| 测试用输入 | 输出结果 | 覆盖路径 |
| W=1,A=0,S=0,D=0 | 向前移动 | abdfj |
| W=0,A=1,S=0,D=0 | 向左移动 | abeim |
| W=0,A=0,S=1,D=0 | 向后移动 | abdgk |
| W=0,A=0,S=0,D=1 | 向右移动 | abehl |
| W=1,A=1,S=0,D=0 | 向左前方移动 | Abdfj,abeim |
| W=1,A=0,S=1,D=0 | 静止 | ac |
| W=1,A=0,S=0,D=1 | 向右前方移动 | Abdfj,abehl |
| W=0,A=1,S=1,D=0 | 向左后方移动 | Abeim,abdgk |
| W=0,A=1,S=0,D=1 | 静止 | ac |
| W=0,A=0,S=1,D=1 | 向右后方移动 | Abdgk,abehl |
| W=1,A=1,S=1,D=0 | 向左移动 | abeim |
| W=1,A=1,S=0,D=1 | 向前移动 | abdfj |
| W=1,A=0,S=1,D=1 | 向右移动 | abdgk |
| W=0,A=1,S=1,D=1 | 向后移动 | abehl |
| W=1,A=1,S=1,D=1 | 静止 | ac |



移动动画播放流程图

|  |  |  |
| --- | --- | --- |
| 测试用输入 | 输出结果 | 覆盖路径 |
| 速度为0 | 播放静止动画 |  |
| 速度不为0 | 旋转到当前速度方向并根据速度大小播放放移动动画 |  |



移动截图

**十、总结及心得体会：**

通过两个简单的程序设计，我们进行了白盒与黑盒方式的测试用例设计，并成功的将设计的测试用例在程序上实施了测试，得到了满意的结果。在熟悉测试软件的方法后，我们细化了学生事务管理系统的模块。在这个过程中，我们遇到了很多困难，通过询问老师，上网查询等方法最终解决了问题，成功地完成了实验。

**十一、对本实验过程及方法、手段的改进建议：**

希望能够通过一些具体的例子了解到白盒测试和黑盒测试在实际的开发中具体的应用场景，以及一些相应的要求和标准，以便更好地将测试应用到实际开发中

**报告评分：**

**指导教师签字：**