**T-Rex Runner Plus软件详细设计说明书**

**1引言**

**1.1编写目的**

T-Rex Runner Plus是以Google Chrome搭载的T-Rex Runner基础开发的跑酷冒险类单机游戏，在原有游戏的基础上增加了道具、血量、成就等新内容。该系统旨在满足各类用户的休闲娱乐需求。玩家通过主菜单进入游戏、查看成就、退出游戏。进入游戏后，玩家使用键盘实时操纵小恐龙运动，系统对用户操作进行可视化反馈。

本说明书给出了T-Rex Runner Plus系统的设计说明，包括最终实现的软件必须满足的功能、性能、接口和用户界面、设计约束等。

目的在于：

·为编码人员提供依据；

·为修改、维护提供条件；

·项目负责人将按计划书的布置和控制开发全过程；

·项目质量保证组将按此计划书做阶段性和总结性的质量验证和确认。

本说明书的预期读者包括：

·项目开发人员，特别是编码人员；

·软件维护人员；

·技术管理人员；

·项目负责人和全体参加人员。

**1.2背景**

待开发软件系统的名称：**T-Rex Runner Plus**

本项目的任务提出者：郭天傲，刘一鸣，马涵聪，冉瑾瑜

开发者：郭天傲，刘一鸣，马涵聪，冉瑾瑜

用户：需要使用本软件进行游戏娱乐的人

运行该程序系统的计算中心：安装有Visual Studio的计算机

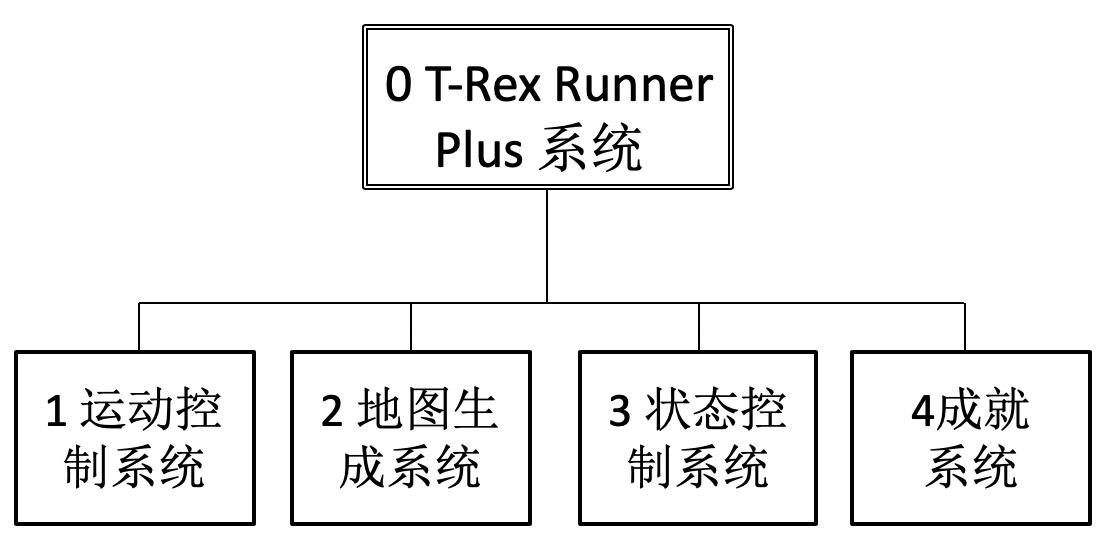
**1.3定义**

　　本系统：对T-Rex Runner Plus软件系统的简称。

**1.4参考资料**

《软件工程》课程课件、《软件工程方法与实践》

**2程序系统的结构**



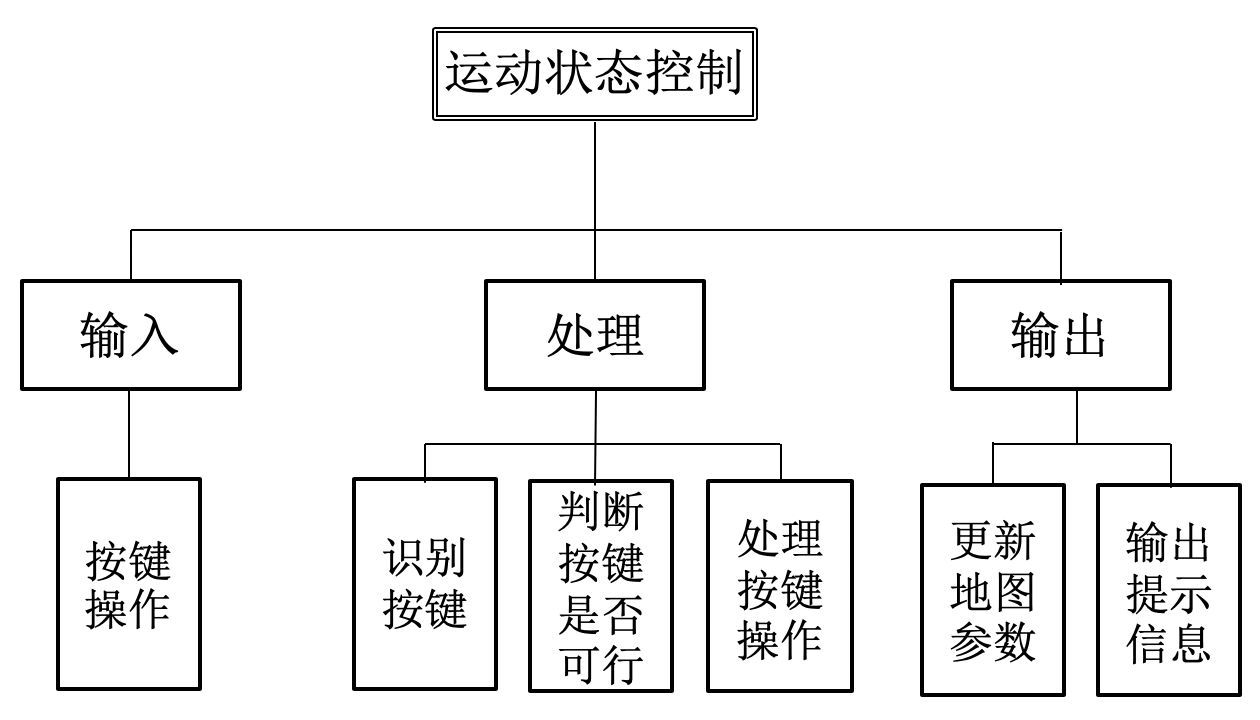
**3 程序1设计说明**

　　从本章开始，逐个地给出各个层次中的每个程序的设计考虑。以下给出的提纲是针对一般情况的。 对于一个具体的模块，尤其是层次比较低的模块或子程序，其很多条目的内容往往与它所隶属的上一层模块的对应条目的内容相同，在这种情况下，只要简单地说明这一点即可。

**3.1程序描述**

运动控制系统负责根据用户的输入按键信息调整控制小恐龙的运动和行为。若用户按下空格键，则小恐龙跳起；若用户按下‘R’键，则重新开始游戏；用户按下‘ ’键，则加速小恐龙的下落；若用户按下‘P’键，则暂停或继续游戏。该部分子程序有用于重置游戏程序resetGame，不可重人，无覆盖要求，采用顺序处理，使用事件驱动模型。

**3.2功能**



运动状态控制程序采用事件驱动模型，用户按键输入触发事件。程序先识别按键输入是否合法即是否为空格、‘R’键、‘ ’键、‘P’键之一，然后结合状态控制系统判断按键操作在当前游戏状态下是否可行，若可行则根据按键操作处理，更新地图要素相关参数。

**3.3性能**

　　该程序采用事件驱动模型，性能、精度主要取决于Visual Studio响应按键事件的性能和精度。

**3.4输入项**

　　用户通过键盘按键输入操作指令，程序采用事件驱动模型。

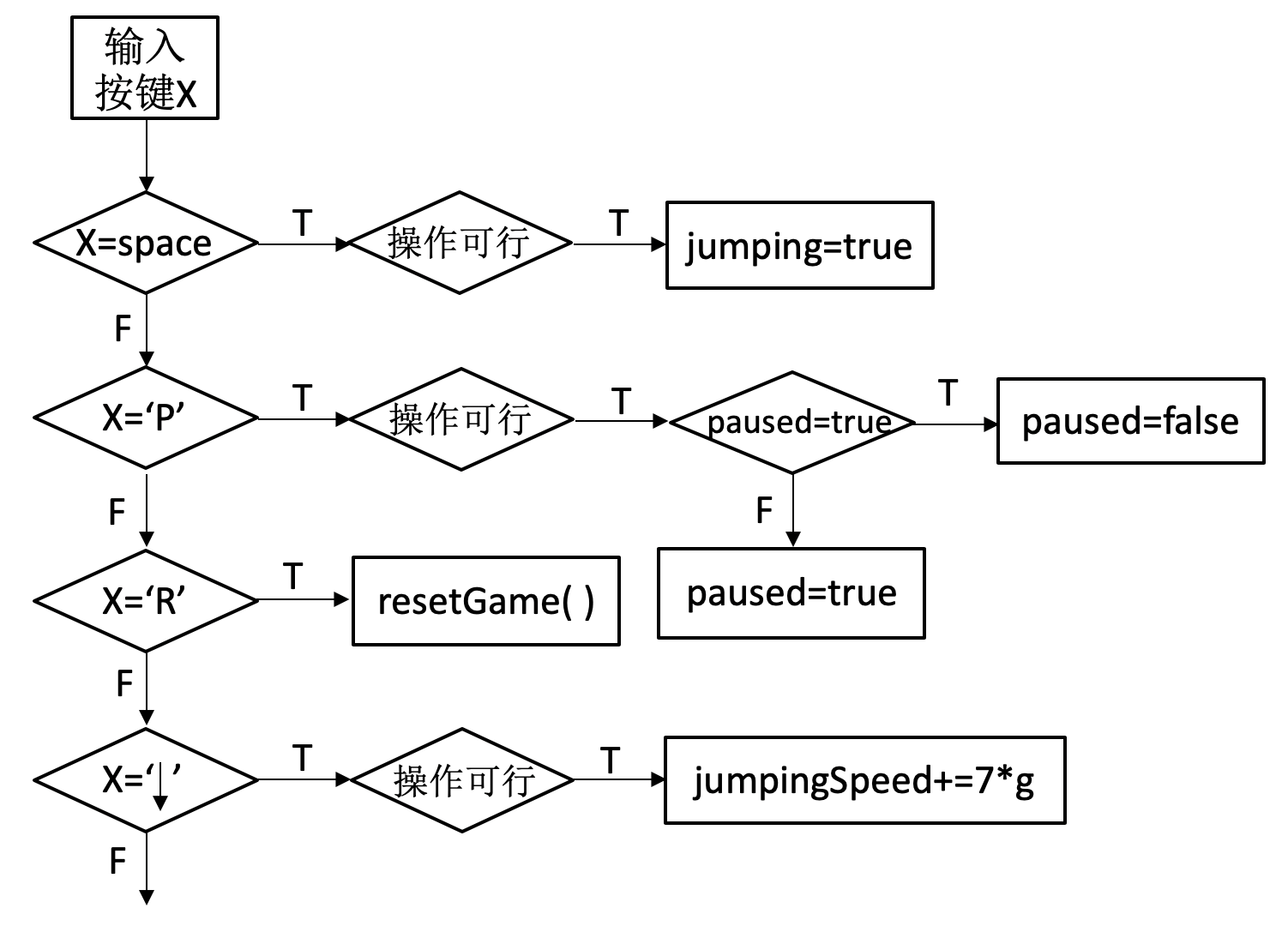
**3.5输出项**

　　布尔值jumping记录小恐龙是否起跳，若用户按下空格键且当前状态下小恐龙操作可行，则将jumping赋值为true输出到地图生成系统和状态检测系统。布尔值paused记录当前游戏是否为暂停状态，若用户按下‘P’键且当前状态下游戏处于进行状态（paused值为false），则将paused赋值为true输出到地图生成系统和状态检测系统并输出提示信息到屏幕；若用户按下‘P’键且当前状态下游戏处于暂停状态（paused值为true），则将paused赋值为false输出到地图生成系统和状态检测系统。若用户按下‘R’键，则调用重置游戏函数resetGame。jumpSpeed为整型数据，记录小恐龙下落速度，若用户按下‘ ’键，且操作可行，则将jumpSpeed加上7g输出到地图生成系统和状态检测系统。

**3.6算法**

运动状态控制程序采用事件驱动模型，用户按键输入按键为空格、‘R’键、‘ ’键、‘P’键之一才会触发事件。若用户按下空格键并且当前小恐龙非跳起状态，则将jumping赋值为true输出到地图生成系统和状态检测系统。若用户按下‘R’键，则直接调用resetGame函数，将游戏进程相关参数重置即可。若用户按下‘P’键当前状态下游戏处于进行状态（paused值为false），则将paused赋值为true输出到地图生成系统和状态检测系统并输出提示信息到屏幕；若用户按下‘P’键且当前状态下游戏处于暂停状态（paused值为true），则将paused赋值为false输出到地图生成系统和状态检测系统。若用户按下‘ ’键且当前小恐龙正处于跳起后下落阶段，则将记录下落速度的值jumpSpeed加上7g并输出到地图生成系统和状态检测系统。

**3.7流程逻辑**



**3.8注释设计**

　　在判断条件分支安排注释，说明按键操作可行状态条件及其后续相关操作。

**3.9尚未解决的问题**

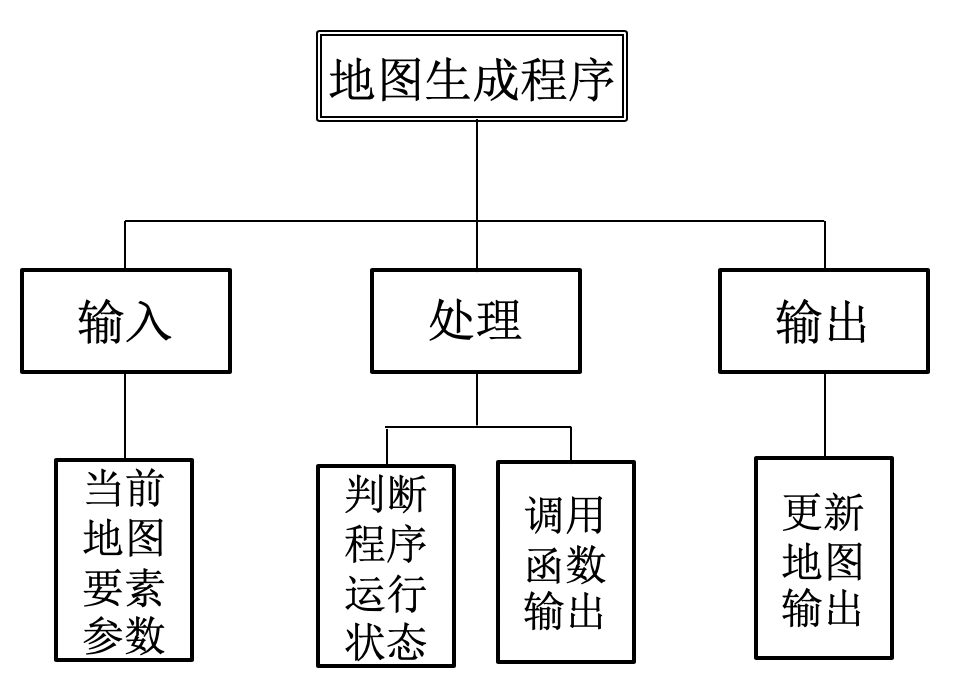
　　游戏改进为双人游戏的相关操作处理。

**4 程序2设计说明**

**4.1程序描述**

地图生成系统根据相关的地图要素参数，如小恐龙位置、障碍物、道具、血条、结束界面等，生成图形用户界面，用于可视化。使用常驻内存，不含子程序，不可重人，无覆盖要求，按时间顺序和触发事件调用不断刷新界面。

**4.2功能**



该程序主要功能是根据游戏状态和地图要素信息输出地图。

**4.3性能**

　　按一定时间间隔以及用户的按键操作不断刷新地图，要求地图的连贯性较好，图形显示尽可能平稳不要有闪烁现象。

**4.4输入项**

　　记录游戏是否结束healthflag、记录血量整型health、记录分数双精度浮点型score、记录障碍物速度整型obstacleSpeed、小恐龙位置信息。

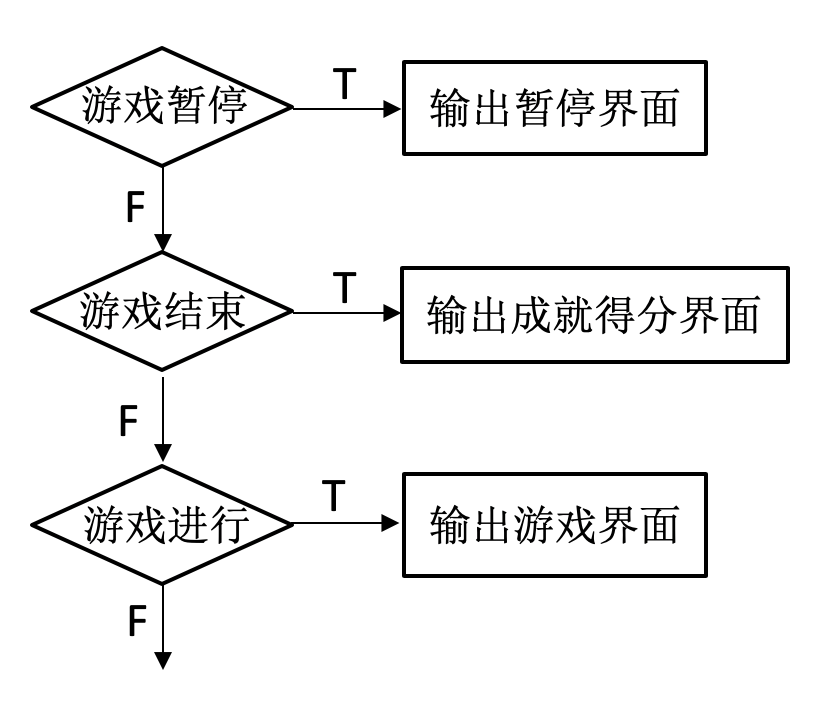
**4.5输出项**

　　根据地图要素参数信息输出图形到屏幕。

**4.6算法**

首先判断游戏是处于暂停状态、进行状态还是结束状态，再根据各自状态相关地图要素参数，生成图形用户界面。

**4.7流程逻辑**

****

**4.8注释设计**

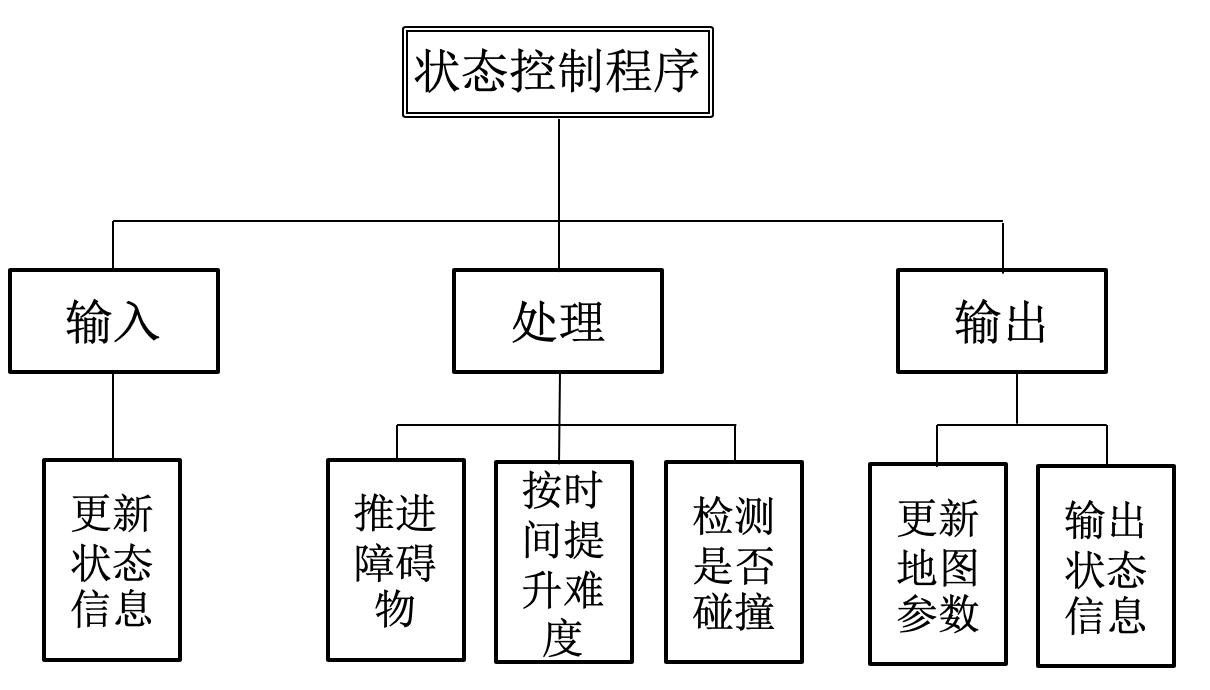
　　注明各地图要素含义。

**5 程序3设计说明**

**5.1程序描述**

状态控制程序负责按照时间序列不断推进障碍物、提高游戏难度，控制小恐龙跑动、跳起、下落状态，游戏暂停、继续状态的切换，配合碰撞检测系统和地图生成系统输出。使用常驻内存，包含子程序用于检测小恐龙与障碍物是否相碰撞，不可重人，无覆盖要求，按时间顺序和触发事件调用不断刷新。

**5.2功能**

****

根据按时间序列推进的状态信息，如游戏难度、障碍物、跳起下落等，和按键操作触发的状态信息，如跳起、加快下落、暂停等，检测小恐龙与障碍物是否碰撞，推进场景变换，以及暂停和游戏界面、游戏界面与成就界面的切换。

**5.3性能**

　　与运动控制系统有关，受时间间隔取值影响，应当尽量保证小恐龙运动、障碍物推进运动连续。

**5.4输入项**

　　记录游戏是否结束healthflag、记录血量整型health、记录分数双精度浮点型score、记录障碍物速度整型obstacleSpeed、小恐龙位置信息、游戏时间、记录游戏是否处于暂停状态布尔值paused、记录小恐龙下落速度整型jumpingSpeed。

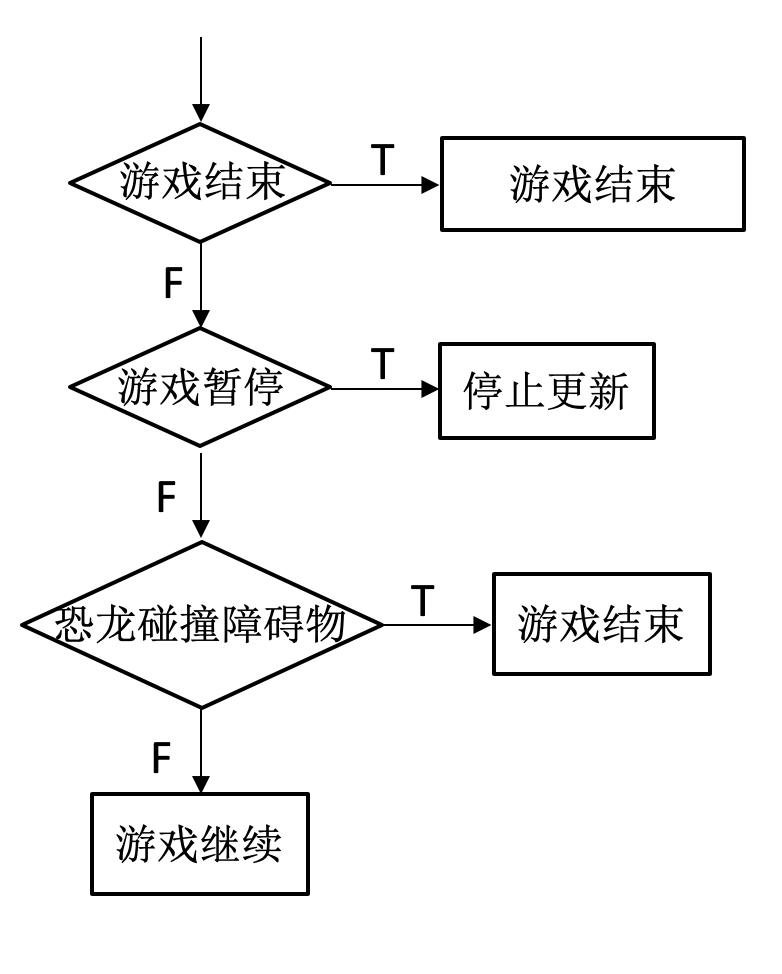
**5.5输出项**

　　输出更新的地图各项参数，用于地图生成板块生成图形用户界面。

**5.6算法**

　　根据小恐龙坐标与障碍物坐标判断二者是否相撞判断游戏是否结束，再按照时间序列推进障碍物并增加难度，判断是否暂停游戏。

**5.7流程逻辑**



**5.10注释设计**

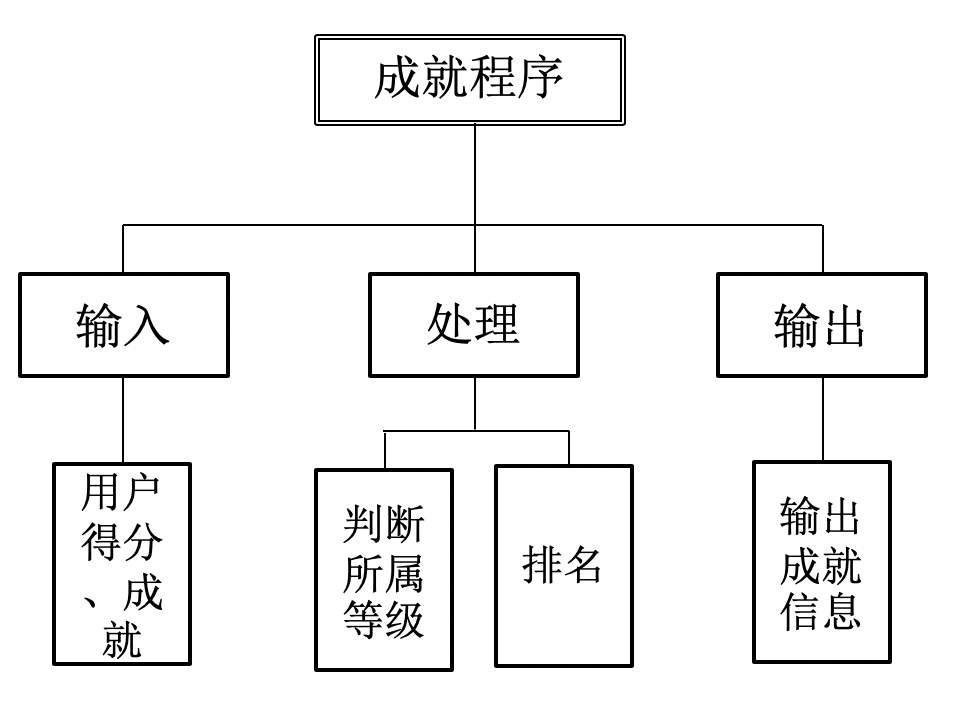
　　判断小恐龙与障碍物相撞时给出说明注释；游戏难度随时间增加给出注释。

**6 程序4设计说明**

**6.1程序描述**

　　成就系统负责游戏结束后，根据用户得分给出用户成就，按最高血量达到20、50、100分为续命达人、续命大师、续命之神三个成就，按得分达到100、500、1000分为得分能手、得分狂人、得分王者。使用常驻内存，不含子程序，不可重人，无覆盖要求。

**6.2功能**

****

根据用户续命血量和得分情况判断用户所属成就登记，然后输出成就信息。

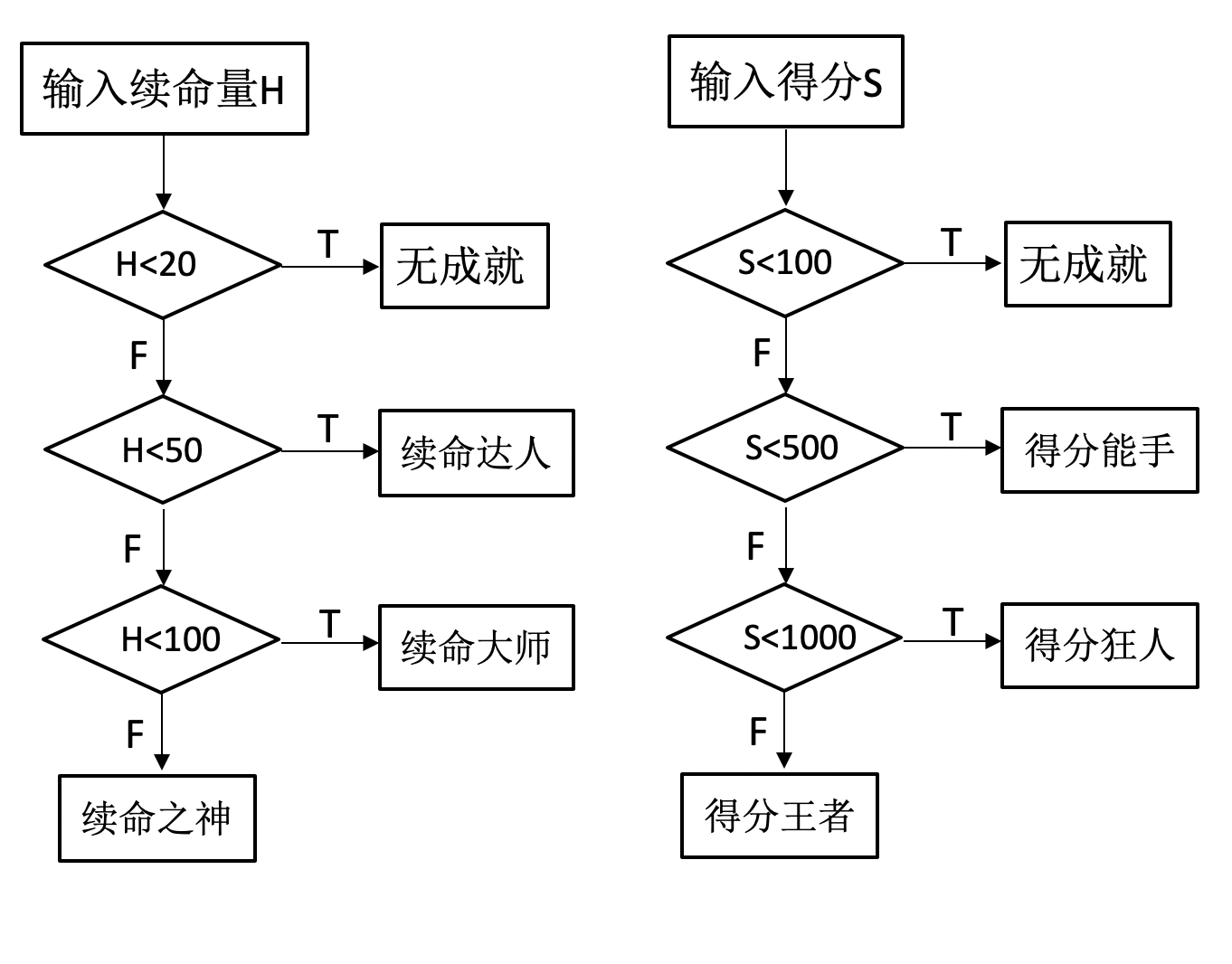
**6.3输入项**

　　记录续命血量整型health和记录得分的整型score。

**6.5输出项**

　　记录续命血量整型health、记录得分的整型score、续命成就的字符串、得分成就的字符串。

**6.7流程逻辑**



**6.10注释设计**

　　注释说明各得分段对应成就。