

跑酷游戏设计文档

2015013249 孙子平

1. 游戏介绍

《Escape the Deadline》是一款采用物理模拟的跑酷游戏。所有的游戏场景均从文件读取。

需要指出的是游戏中的物理模型。游戏中，若玩家与地面碰撞，将失去所有垂直于地面的动量分量。玩家的跳跃以垂直于地面方向的冲量实现。玩家不能在空中控制。

目前实现了三个地图。第一个地图为无限随机的跑酷地图，强制向右卷屏。第二个地图的重力加速度较小（这使得玩家可沿着墙壁向上爬行），地图有限，当玩家高度低于-5000 时会死亡，此地图考验技术。第三个地图为空白地图，玩家可进入命令模式后手动绘制。

注意：进入命令模式务必须确保输入法为英文模式。按住 shift 再按单引号。

2. 游戏流程

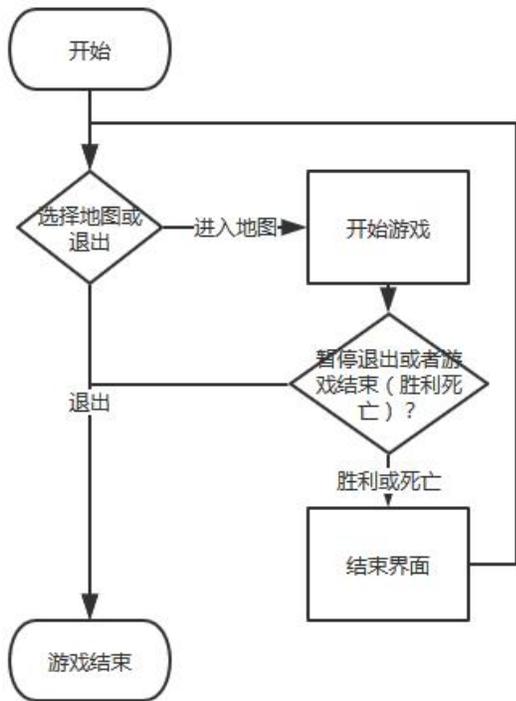


图 1：游戏流程图

3. 界面设计



图 2：界面设计

右上角为得分、跑动距离。玩家上方为血条。背景有坐标。左下角可显示命令行及错误信息。

4. 概要设计

4.1 系统架构

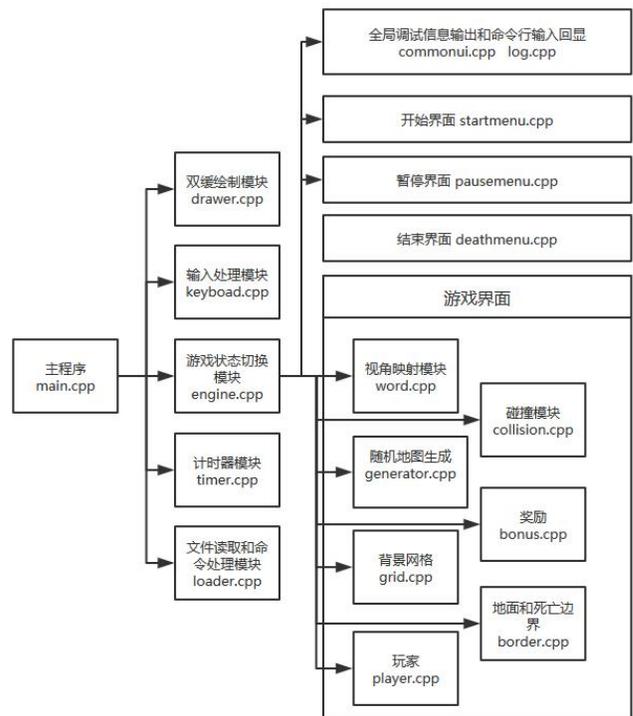


图 3：系统架构

4.2 核心代码流程

为了避免各底层模块如（双缓、输入、计时器、命令）以及负责游戏进行的（视角映射、碰撞）都采用了回调模式。Engine 模块负责初始化各个具体界面，并负责界面切换，统领所有的其他模块。而其他的各个模块则在其被需要的时候向底层模块注册他们的回调函数，并以此工作。

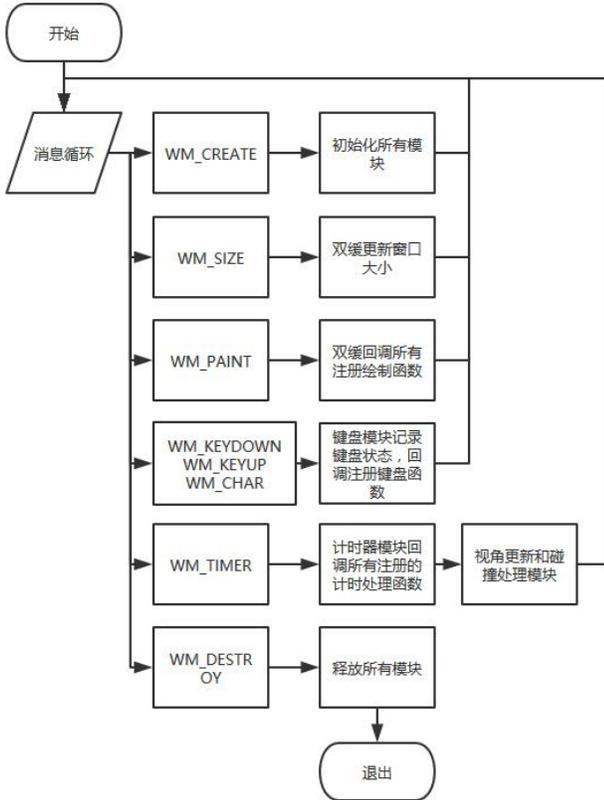


图 4：核心代码流程

5. 总结与感悟

学会了模块化设计。采用回调机制解耦。学会了 git 版本控制。顺带着也开始练 vim。颇有成就的地方有：游戏动画丰富；碰撞处理细致，采用了半平面算法，并根据速度方向进行碰撞预判；计时器模块采用了变频的设计，减少系统资源消耗；整体架构设计合理可扩展性强；地图为脚本文件，极为自由。也有些遗憾：碰撞处理，然而还是有一些 bug，这是初期接口设计的问题；Engine 的设计在项目初期忘记考虑了暂停界面下重新载入时的游戏状态切换，这导致了后期代码的累赘和混乱；缺乏一个动画管理的模块；缺乏一个没有来得及优化性能。