

基本功能：

- 接收玩家的键盘输入
- 对怪物、场景等进行更新
- 保存玩家的数据

## 设定

无尽关卡

## 流程

人 --> 场景 --> 打怪 --> 切换场景 --> 打怪 --> 切换场景 --> ……

## 模块

```
|——初始化 startup()函数
|      |——初始化画布
|      |——调用实例对象的initialize()函数 --> 读取图片资源
|      |——设置随机数种子 --> 开始时游戏的随机设置不同
|
|——绘制菜单 menu()函数
|      |——画菜单界面
|      |——接收操作 --> 开始/退出游戏
|      |——从文件中读取最高分、关卡记录并展示
|
|——画游戏界面 draw()函数
|      |——调用实例的show()函数 --> 按一定层次显示出人、景
|      |——游戏结束时跳转到结束界面
|
|——数据更新 updateWithoutInput()函数
|      |——调用实例的movement()等函数 --> 更新物体的坐标
|      |——调用change()函数 --> 更新主角坐标、切换场景
|
|——接收键盘输入 updateWithInput()函数
|      |—— --> action()函数 --> GetAsyncKeyState()函数 --> 接收WASD和方向键的输入 =>实现人物
```

斜向移动和射击

| |----->pause()函数 --> 检测暂停  
|  
|-----数据重置 dateReset()函数  
| |-----游戏结束时重置数据 --> 游戏结束后能再次进行游戏  
|  
|-----结束界面 deadUpdate()函数  
| |----->展示游戏结束界面及分数数据  
| |----->最高分记录存入保存文件

## 其它函数模块和难点

putMonster() --> 检测切换场景 --> 怪物随机布点

impact() --> 避免位置放置重叠

碰撞检测();

=>伤害检测, 碰撞反馈, 怪物布置

图片绘制可能出错

数据残余

## 类

类的创建=>实现物体的创建

不同的类具有各自的函数来完成职能

怪物类

1. initialize()函数, 完成素材的读取。
2. show()函数, 实现素材的绘制。
3. movement()函数, 实现坐标的变化。
4. damageJudgment()函数, 实现伤害判断的检测。
5. monsterClear()函数, 在游戏结束和实例死亡后初始化数据。