

HHardestUGame 设计文档

Team:

Larghetto

Member:

- Larghetto_ccEupho
- Larghetto_Boléro
- Larghetto_五亿个小铃铛
- Larghetto_没嘴见人朽木君
- Larghetto_王不会

功能模块

模块名（自顶向下层数）	模块描述
场景元素原型模块（2层）	提供地图中具有碰撞逻辑的元素的原始实现
控制模块（2层）	实现玩家的上、下、左、右移动逻辑
玩家模块（1层）	基于控制模块实现玩家的可视化
地图模块（0层）（依据地图数量设计）	根据设计出的游戏地图创建关卡，实现元素与玩家的交互逻辑
通信模块（1层）	实现游戏通关后玩家信息视图与后端数据库的交互
数据库模块（1层）	记录玩家通关信息
UI模块（0层）	使用按钮与文字等对游戏进行描述，与玩家交互
测试模块（0，1，2层）	基于常用测试方法对模块进行测试

具体代码实现

- 场景元素原型类Item，包含

```
constructor(root, info)// 根据传入的前端页面根标签与info字典中的信息，构造一个场景元素
```

- 控制类Controller，包含

```
constructor(root)// 根据传入的前端页面根标签，使用Set()实现用户按键监听
start()// 作为启动方法，启动一个按键监听实例
```

- 玩家类Player,包含

```

    constructor(root, info)// 根据传入的前端页面根标签与info字典中的信息, 构造一个玩家原型
    updateControl()// 根据键盘监听到的按键按下与释放调整玩家速度信息
    updateMove()// 实时根据玩家速度渲染玩家位置, 实现移动
    update()// 每一帧检测玩家操作, 实现逻辑
    hitBorder()// 判断玩家是否碰撞边界, 避免越界

```

- 地图模块Create (根据地图数量实现不同的类), 包含

```

    constructor()// 创建类并进行加载
    setup() // 构建游戏场景容器, 调整窗口大小, 启动帧渲染
    init()// 初始化场景内所有元素及玩家
    gameloop()// 每一帧执行的循环逻辑

    // 有限状态机
    play()// 游玩状态, 实现碰撞逻辑
    end()// 死亡状态, 提供再次进入本关卡入口
    succeed()// 通关状态, 提供进入下一关入口或胜利结算页面入口

    // 工具方法
    bindRestartAfterEnd()// 绑定状态机“游玩”、“死亡”相互转换时逻辑
    bindRightAnimation()// 绑定玩家右移时动画
    bindLeftAnimation()// 绑定玩家左移时动画
    unbindAnimation()// 在按键冲突时解绑玩家动画
    spriteAnimationPlay()// 根据按键监听器播放玩家动画

```

- 通信模块AddServlet, 包含

```

    public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException , IOException
    // 接收html中表单数据, 解析并发送至后端服务器
    // 后端服务器将数据存入数据库并将相应html发送至前端渲染

```

- 数据库模块druid.properties, 包含

```

// 部分数据库配置信息
#key=value
driverClassName=com.mysql.jdbc.Driver
url=jdbc:mysql://localhost:3306/test2?rewriteBatchedStatements=true
username=debian-sys-maint
password=7GVPWmzVBfp0imLY
#initial connection Size
initialSize=10
#min idle connecton size

```

```
minIdle=5
#max active connection size
maxActive=50
#max wait time (5000 mil seconds)
maxWait=5000
```

- UI模块, 包含

```
index.html // 显示游戏名称、“规则”按钮、“模式”按钮、“公告”按钮、“关于”按钮
rules.html // 描述游戏规则,
pattern.html // 提供模式选择按钮、返回按钮, 包括普通模式、大师模式
announce.html // 发布更新公告
ranking.html // 记录玩家通关信息
```

- 测试模块test, 包含 后端测试类testDao, 包含

```
@Test
public void testActorDao()// 测试写入数据库功能
public void testInsert()// 测试前后端通信插入功能
```

前端测试: 依据前端单元测试表, 在完成模块后依据由pattern.html演变而来的Createtest.html与由CreateCommom.js演变来的utilstest.js

使用行注释、控制台打印方法进行单元测试。详细信息见测试文档。