**《畏光》游戏系统简要设计文档及代码框架**

### 一、​****​核心系统​****​

1. ​**​计分系统** (GameManager)

全局分数存储变量：score

1. ​**​移动系统​**​ (PlayerMovement)

水平移动：左右输入控制速度

跳跃：支持二段跳机制

地面检测：通过Physics2D.Overlap碰撞检测

动画状态机：idle/running/jumping/falling/dead五种动画

1. ​**​死亡系统** (PlayerLife)

死亡处理：播放音效/动画dead，更改死亡次数

重生机制：返回重生点

死亡计数器：显示死亡次数

1. ​**​物品与收集​**​

​**樱桃收集​**​ (item\_collector)

碰撞检测：接触樱桃时销毁物体

分数更新：实时同步到GameManager

UI显示：动态更新得分文本

​**​存档点​**​ (SavePoint)

触碰设置：保存位置并更新颜色标识

首次触发：奖励分数

### 二、​**​**环境交互​****​

1. ​**​机关系统​**​

​**​平台移动​**​ (WaypointFollower)

路径点循环移动

移动双模式：单向瞬移回起点/摇摆移动

​**​粘性平台​**​ (StickyPlatform)

玩家/箱子吸附：设置物体父子关系

脱离恢复：重置父物体

1. ​**​陷阱系统​**​

​**​死亡区域​**​ (DeadZone)

玩家触碰：立即死亡

应用：地刺，锯片，悬崖下方

​**​聚光灯​**​ (LightMove)

移动/旋转：双模式（摇摆/单向）

射线检测：识别玩家触发角色死亡

1. ​**​按钮机关​**​ (Button)

触发模式：按下/松开双事件

事件绑定：可配置Event事件数组

有效触发体：玩家或箱子灯可以在检查器中选择所需标签检测

### 三、​**​**关卡系统​****​

1. ​**​终点机制​**​ (Finish)

碰撞检测：玩家触碰通关

流程：播放音效 →完毕后加载下一关

1. ​**​关卡管理​**​

开始菜单 (StartMenu)：启动第一关

结束菜单 (EndMenu)：重置分数并重载场景

场景加载：使用SceneManager索引控制

### 四、​**​**事件系统​****​

1. ​**​基础架构​**​ (Event)

抽象类：定义Trigger()接口

1. ​**​具体实现​**​

​**​物体显隐​**​ (ObjectEnableEvent)：控制目标物体激活状态

### 五、​**​**辅助功能​****​

1. ​**​摄像机跟踪​**​ (Cemara)

固定Y轴偏移跟随玩家

1. ​**​旋转物体​**​ (Rotate)

持续Z轴旋转

1. ​**​UI系统​**​

文本：分数/死亡次数显示

音效反馈：收集/死亡/跳跃音效

对话框：玩梗文案/解谜提示

### 六、代码框架

#### 1.游戏管理系统

class GameManager {

<<Singleton>>

- static \_instance GameManager实例

+ static Instance 访问器

+ int score 全局分数

}

class EndMenu {

+ void ReloadGame() 重载游戏：重置分数并加载指定场景

}

class item\_collector {

- int cherries 当前分数

- Text cherriesText 分数显示UI

- AudioSource collectSoundEffect 收集音效

+ void Score() 主动加分方法

+ void OnTriggerEnter2D(Collider2D) 樱桃收集碰撞检测

}

#### 2.玩家控制系统

class PlayerMovement {

- Rigidbody2D rb 刚体组件

- Animator anim 动画控制器

+ float speed 移动速度

- bool isGround 是否在地面

- bool isJump 是否跳跃中

- int jumpCount 剩余跳跃次数

+ float jumpForce 跳跃力度

- void IsGrounded() 地面检测

- void Jump() 跳跃逻辑处理

- void GroundMovement() 地面移动控制

- void UpdateAnimationState() 更新动画状态

}

class PlayerLife {

- Rigidbody2D rb 刚体组件

- Animator anim 动画控制器

- Transform rebirthPoint 重生点位置

- int dieNumber 死亡次数

+ void Die() 死亡处理：播放动画/音效

+ void Rebirth() 重生：回到重生点

}

#### 3.环境交互系统

class Button {

<<enum>> TriggerMode 触发模式枚举

- TriggerMode mode 当前触发模式（松开取消/重复点击取消）

- HashSet<Collider2D> pressingObjects 按压物体集合

- bool isPressed 当前按压状态

- void UpdateButtonState() 更新按钮状态

- Event[] onPressEvents 按下事件数组

- Event[] onReleaseEvents 松开事件数组

- bool IsValidPresser() 验证有效触发体

- void HandlePress() 处理按下事件

- void HandleRelease() 处理松开事件

+ void ResetButton() 重置按钮状态

}

class DeadZone {

+ bool effectOnBox 是否影响箱子

+ void OnTriggerEnter2D(Collider2D) 死亡区域碰撞检测

}

class Finish {

- bool levelCompleted 关卡完成标志

+ void OnTriggerEnter2D(Collider2D) 终点碰撞检测

- void CompleteLevel() 完成关卡加载下一关

}

class SavePoint {

+ static SpriteRenderer lastSave 最后激活的存档点

+ Transform SavePointPosition 存档位置

- bool firstGet 首次获取标志

}

class StickyPlatform {

- Transform boxParent 箱子的原始父物体

+ void OnTriggerEnter2D(Collider2D) 进入平台时吸附物体

+ void OnTriggerExit2D(Collider2D) 离开平台时释放物体

}

#### 4.机关系统

class LightMove {

+ bool enableMovement 启用移动

+ bool enableRotation 启用旋转

- Light2D spotlight 聚光灯组件

+ int rayCount 射线数量

+ float rayLength 射线长度

- Vector2[] CalculateRayDirections() 计算射线角度

+ LayerMask detectionMask 检测层级

- void HandleRayHit() 射线命中处理

+ void HandleMovement() 处理移动逻辑

+ void HandleRotation() 处理旋转逻辑

+ void CastSpotlightRays() 发射检测射线

}

class WaypointFollower {

- GameObject[] waypoints 路径点数组

- int currentWaypointIndex 当前路径点索引

+ float speed 移动速度

+ bool teleport 是否传送模式

+ void Update() 移动逻辑更新

}

class Rotate {

+ float speed 旋转速度

+ void Update() 每帧旋转更新

}

#### 5.事件系统

class Event {

<<abstract>>

+ abstract void Trigger() 事件触发抽象方法

}

class ObjectEnableEvent {

- GameObject targetObject 目标物体

- bool desiredState 期望状态

+ override void Trigger() 设置物体激活状态

}

#### 6.辅助系统

class Cemara {

- Transform player 玩家位置

+ void Update() 摄像机跟随玩家

}

class StartMenu {

+ void StartGame() 开始游戏加载第一关

}

#### 7.主要类关系

PlayerMovement -- PlayerLife : 依赖

Button ..> Event : 依赖

ObjectEnableEvent --|> Event : 继承

LightMove ..> PlayerLife : 灯光检测到玩家时调用Die()

item\_collector --> GameManager : 读写分数

EndMenu --> GameManager : 重置分数

DeadZone --> PlayerLife : 触发死亡

Finish --> SceneManager : 场景加载

SavePoint --> item\_collector : 到达存档点触发加分