角色管理模块

# CharacterManager（角色管理，待定，目前用场景管理器DemoSceneManager代替功能）

## 变量

public GameObject **mainPlayer**：主角色GameObject

public GameObject **subPlayer**：副角色GameObject

public List<GameObject> **enemies**：场景内的所有敌人GameObject

public GameObject **boss**：BOSS的GameObject

public event EventHandler **gameOverEvent**：角色死亡后执行的所有操作（暂未使用）

## 方法

void **GameOver**()：角色死亡时执行gameOverEvent（暂未使用）

# PlayerControl（玩家角色控制器，执行各种行为）

## 变量

public float **playerMaxLife**：玩家残机最大值

public float **playerLife**：玩家残机

public float **playerMaxHP**：玩家最大HP

public float **playerHP**：玩家当前HP

public float **maxPPoint**：角色最大P点

public float **pPoint**：角色当前P点

public float **maxBluePoint**：角色最大蓝点

public float **bluePoint**：角色当前蓝点

public int **maxBeer**：回血道具最大携带量

public int **beer**：回血道具当前数量

public bool **isMainPlayer：**是否是主角色

public PlayerControl **changePlayer**：可更换的角色的PlayerControl组件

public float **playerChangeTime**：两次更换的间隔时间

public float **playerBaseAtk**：角色基础攻击力

public float **playerBaseDef**：角色基础防御力

public Animator **playerAnimator**：角色动画机

public List<string> **attackButtons**：角色攻击按键列表

public List<AAttackMode> **playerAttackModes**：角色攻击模块列表

public AMoveMode **playerMoveMode**：角色移动模块

public AHitMode **playerHitMode**：角色受伤模块

public float **playerSize**：角色判定大小（暂未使用，使用的胶囊形碰撞体）

[HideInInspector]

public float **playerAttackPoint**：计算后的角色攻击力

public float **playerDefensePoint**：计算后的角色防御力

public bool **isDead**：玩家角色是否死亡

## 方法

public void **SetAttackMode**(AAttackMode attackMode, int index)：设置角色攻击模块

public void **SetMoveMode**(AMoveMode moveMode)：设置角色移动模块

public void **SetHitMode**(AHitMode hitMode)：设置角色受伤模块

public int **GetAttackModeIndex**(int index) 获取当前第index号攻击模式编号

public string **GetAttackModeName**(int index) 获取当前第index号攻击模式名称

# EnemyControl（敌人角色控制器）

## 变量

public float **enemyMaxHP**：敌人最大HP

public float **enemyHP**：敌人当前HP

public Animator **enemyAnimator**：敌人角色动画机

public List<AAttackMode> **enemyAttackModes**：敌人攻击模块

public AMoveMode **enemyMoveMode**：敌人移动模块

public AHitMode **enemyHitMode**：敌人受伤模块

public float **enemySize**：敌人判定大小（暂未使用，使用的圆形碰撞体）

public float **enemyAttackPoint**：敌人攻击力

public float **enemyDefensePoint**：敌人防御力

public float **hatredRange**：仇恨距离

public List<GameObject> **itemDropped**：掉落物品

public List<float> **droppedRate**：掉落率

[HideInInspector]

public bool **isDead**：敌人角色是否死亡

## 方法

public void **SetAttackMode**(AAttackMode attackMode, int index)：设置角色攻击模块

public void **SetMoveMode**(AMoveMode moveMode)：设置角色移动模块

public void **SetHitMode**(AHitMode hitMode)：设置角色受伤模块

# BossControl（暂未使用）

# AAttackMode（攻击模块抽象类）

## 变量

public bool **isCannotAttack**：角色是否处于不能攻击状态

public int **attackModeIndex**：攻击模式的编号

public string **attackModeName**：攻击模式的名字

## 方法

public abstract void **AttackButtonDown**()：攻击键按下时的操作

public abstract void **AttackButtonUp**()：攻击键未按下时的操作

public abstract void **PowerUp**(int power)：增加攻击力

# AttackMode\_Sakuya\_01 : AattackMode（咲夜的第一种攻击模式，目前铃仙也使用此模式）

## 子弹参数相关变量

public Vector3 **relativeLaunchPosition**：发射点相对角色中心的位置

public float **relativeLaunchAngle**：发射方向相对角色朝向的角度（角度值，向前为0度，逆时针为正）

public int **bullentNumber**：一轮发射子弹数量

public float **bullentRange**：子弹覆盖范围，最两边子弹发射方向之间的夹角

public int **bullentWave**：一轮发射波数

public float **waveInterval**：每波间隔

public float **chargeFront**：攻击前摇，按下攻击键到发射出第一轮之间的时间

public float **chargeBack**：攻击后摇，最后一轮发出到下一次能产生前摇之间的时间（最好大于waveInterval）

public float **life**：子弹生存时间

public float **attackPointRatio**：子弹攻击力比角色攻击力的倍数

public bool **isCloseAttack**：是否近战

## 其它变量

public int **attackModeIndex**：攻击模式的编号

public string **attackModeName**：攻击模式的名字

public GameObject **bullentType**：发射的子弹预设体

public PlayerControl **playerControl**：角色控制器

public AudioSource **attackSEsource**：攻击音效（暂未使用）

[HideInInspector]

public bool **isCannotAttack**：角色是否处于不能攻击状态

## 方法

public override void **AttackButtonDown**()：攻击键按下时的操作(由PlayerControl每帧调用)

public override void **AttackButtonUp**()：攻击键未按下时的操作(由PlayerControl每帧调用)

public override void **PowerUp**(int power)：增加攻击力（暂未使用，目前直接修改PlayerControl.pPoint）

# AttackMode\_Enemy\_01 : AattackMode（敌人第一种攻击模式）

## 变量

变量与AttackMode\_Sakuya\_01相同，仅在给敌人添加此脚本组件时，填写参数不同。

## 方法

方法与AttackMode\_Sakuya\_01相同，由于敌人为自动AI，仅由EnemyControl每帧调用AttackButtonDown方法。

# ABullent（子弹抽象类）

## 变量

public float **life**：子弹生存时间（AttackMode会对其赋值）

public float **attackPoint**：弹幕攻击力（AttackMode会对其赋值）

# Bullent\_Sakuya\_01 : ABullent（咲夜的第一种子弹，前方扇形直线发射子弹）

## 变量

public float **life**：子弹生存时间（AttackMode会对其赋值）

public float **attackPoint**：弹幕攻击力（AttackMode会对其赋值）

public float **velocity**：子弹速度

public float **acceleration**：子弹加速度

public float **angularVelocity**：子弹角速度

public Vector3 **rotationCenter**：旋转中心，相对子弹自身的局部坐标

public float **bullentSize**：子弹圆形碰撞体半径

public GameObject **hitEffect**：命中特效产生的GameObject（暂未使用）

public List<GameObject> **enemies**：场景中敌人列表（暂未使用，目前使用Layer对碰撞体进行分组来识别是否需要进行命中判定）

public int **effect**：攻击效果的编号（暂未使用）

# Bullent\_Sakuya\_02 : ABullent（咲夜的第二种子弹，前方小刀划过）

## 变量

public float **life**：子弹生存时间（AttackMode会对其赋值）

public float **attackPoint**：弹幕攻击力（AttackMode会对其赋值）

public float **velocity**：子弹速度(近战子弹为零)

public float **acceleration**：子弹加速度(近战子弹为零)

public float **angularVelocity**：子弹角速度（近战武器砍人效果，子弹回旋效果）

public Vector3 **rotationCenter**：旋转中心，相对子弹自身的局部坐标

public float **bullentSize**：子弹圆形碰撞体半径

public GameObject **hitEffect**：命中特效产生的GameObject（暂未使用）

public List<GameObject> **enemies**：场景中敌人列表（暂未使用，目前使用Layer对碰撞体进行分组来识别是否需要进行命中判定）

public int **effect**：攻击效果的编号（暂未使用）

# Bullent\_Sakuya\_03 : ABullent（咲夜的第三种子弹，前方投掷抛物线子弹，落地后爆炸产生攻击判定）

## 变量

public float **life**：子弹生存时间（AttackMode会对其赋值，由于此子弹落地爆炸，生存时间由其它参数决定，因此life值只要足够大就行）

public float **attackPoint**：弹幕攻击力（AttackMode会对其赋值）

public float **distance**：子弹落地距离

public float **hight**：子弹抛物线高度

public float **velocity**：弹幕速度

public float **acceleration**：弹幕加速度

public float **angularVelocity**：弹幕角速度

public Vector3 **rotationCenter**：旋转中心

public float **bullentSize**：子弹圆形碰撞体半径

public GameObject **hitEffect**：命中特效产生的GameObject（暂未使用）

public List<GameObject> **enemies**：场景中敌人列表（暂未使用，目前使用Layer对碰撞体进行分组来识别是否需要进行命中判定）

public int **effect**：攻击效果的编号（暂未使用）

public Animator **bullentAnimator**：爆炸动画机

# Bullent\_Bat\_01 : ABullent（敌人的第一种子弹，前方扇形直线发射子弹）

## 变量

与第7节Bullent\_Sakuya\_01 : ABullent相同

# AMoveMode（移动模块抽象类）

## 变量

public float **moveSpeed**：角色移动速度

[HideInInspector]

public bool **isCannontMove**：角色是否处于不能移动状态

public float **directionAngle**：角色朝向与y轴的夹角（弧度，逆时针为正向）

public Vector2 **characterDirection**：角色朝向

## 方法

public abstract void **Move**()：角色移动，每帧调用

public abstract void **IsDelayed**()：角色硬直时的操作（暂未使用）

public abstract void **SetDirection**(Vector2 direction)：设置角色朝向

# MoveMode\_Player\_Dash : AmoveMode（特殊移动为冲刺的移动模式）

## 变量

public float **moveSpeed**：角色移动速度

public PlayerControl **playerControl**：角色控制器

public Animator **playerAnimator**：角色动画机（后续可能被龙骨取代）

public Rigidbody2D **rb**：角色的刚体

public float **dashSpeed**：冲刺速度

public float **dashTime**：冲刺持续时间

public float **dashChargeTime**：冲刺后摇

[HideInInspector]

public bool **isCannontMove**：角色是否处于不能移动状态

public float **directionAngle**：角色朝向与y轴的夹角（弧度，逆时针为正向）

public Vector2 **characterDirection**：角色朝向

## 方法

public override void **Move**()：角色移动，每帧调用

public override void **IsDelayed**()：角色硬直时的操作（暂未使用）

public override void **SetDirection**(Vector2 direction)：设置角色朝向

# MoveMode\_Player\_Run : AmoveMode（特殊移动为加速跑的移动模式）

## 变量

public float **moveSpeed**：角色移动速度

public PlayerControl **playerControl**：角色控制器

public Animator **playerAnimator**：角色动画机（后续可能被龙骨取代）

public Rigidbody2D **rb**：角色的刚体

public float **dashSpeed**：冲刺速度

public float **dashTime**：冲刺持续时间

public float **dashChargeTime**：冲刺耗尽后的补充时间

[HideInInspector]

public bool **isCannontMove**：角色是否处于不能移动状态

public float **directionAngle**：角色朝向与y轴的夹角（弧度，逆时针为正向）

public Vector2 **characterDirection**：角色朝向

## 方法

public override void **Move**()：角色移动，每帧调用

public override void **IsDelayed**()：角色硬直时的操作（暂未使用）

public override void **SetDirection**(Vector2 direction)：设置角色朝向

# MoveMode\_Enemy\_01 : AmoveMode（敌方移动模式，仇恨距离外随机移动）

## 变量

public float **moveSpeed**：角色移动速度

public EnemyControl **enemyControl**：角色控制器

public Animator **enemyAnimator**：角色动画机

public Rigidbody2D **rb**：角色刚体

public float **moveTime**：移动时间

public float **moveInterval**：移动间隔时间

## 方法

public override void **Move**()：角色移动，每帧调用

public override void **IsDelayed**()：角色硬直时的操作（暂未使用）

public override void **SetDirection**(Vector2 direction)：设置角色朝向

# AhitMode（受伤模块抽象类）

## 方法

public abstract void **Hit**()：受伤后做的操作，每帧调用

public abstract void **BeHit**(int atkPoint, int effect)：触发角色受伤，atkPoint为攻击力，effect为攻击效果编号

# HitMode\_Player\_Normal : AHitMode（玩家通常状态的受伤模式）

## 变量

public PlayerControl **playerControl**：角色控制器

## 方法

public override void **Hit**()：受伤后做的操作，每帧调用（暂未使用）

public override void **BeHit**(int atkPoint, int effect)：触发角色受伤，atkPoint为攻击力，effect为攻击效果编号，攻击效果暂未使用

# HitMode\_Enemy\_Normal : AHitMode（敌人通常状态的受伤模式）

## 变量

public EnemyControl **enemyControl**：角色控制器

## 方法

public override void **Hit**()：受伤后做的操作，每帧调用（暂未使用）

public override void **BeHit**(int atkPoint, int effect)：触发角色受伤，atkPoint为攻击力，effect为攻击效果编号，攻击效果暂未使用

# AItem（物品抽象类）

## 变量

public int **effectPoint**：物品产生的效果量

public AudioClip **itemSE**：物品拾取时的音效

## 方法

public abstract void **ItemEffect**(GameObject player)：物品拾取到时产生的效果

# ItemPowUp : Aitem（P点）

## 变量

public int **effectPoint**：物品产生的效果量

public AudioClip **itemSE**：物品拾取时的音效

## 方法

public override void **ItemEffect**(GameObject player)：物品拾取到时产生的效果

# ItemDrop（物品掉落四散掉落效果）

## 变量

public AItem **item**：物品

public float **distance**：物品落地距离

public float **hight**：物品抛物线高度

public float **velocity**：物品掉落速度