角色管理模块

# CharacterManager（角色管理，待建，变量和方法名待定）

## 变量

public GameObject **player**：角色GameObject

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

public event EventHandler **gameOverEvent**：角色死亡后执行的所有操作

## 方法

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

# PlayerControl（玩家角色控制，执行各种行为）

## 变量

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

public List<AAttackMode> **playerAttackModes**：玩家攻击模块列表

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

public AHitMode **playerHitMode**：角色受伤模块（暂未使用）

public playerSize：玩家判定大小（暂未使用）

public playerHP：玩家HP（暂未使用）

public isDead：玩家角色是否死亡（暂未使用）

## 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

# 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

## 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（暂未使用）

# AttackMode\_Sakuya\_01 : AattackMode（咲夜的第一种攻击模式）

## 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 bool **isCloseAttack**：是否近战

## 其它变量

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）

Vector3 **launchPosition**：实际发射位置

float **launchAngle**：实际发射角度

float **directionAngle**：角色方向角（角色朝向与y轴的夹角（角度值，逆时针为正向））

float **chargeFrontTime**：前摇计时

float **nextAttackableTime**：下次可攻击的时间点

## 方法

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

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

void **Launch**()：发射一波子弹

# ABullent（子弹抽象类）

## 变量

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

# Bullent\_Sakuya\_01 : ABullent（咲夜的第一种子弹）

## 变量

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 int **attackPoint**：子弹攻击力（暂未使用）

public float **bullentSize**：子弹攻击判定半径（暂未使用）

public GameObject **hitEffect**：命中特效产生的GameObject

Vector3 **direction**：子弹实际速度方向，世界坐标

float **startTime**：子弹产生的时间点

public List<GameObject> **enemies**：场景中敌人列表（暂未使用）

public int **effect**：攻击效果的编号（暂未使用）

## 方法

void **CollisionDet**()：碰撞检测（可能被OnTriggerEnter2D代替）

void **BullentTranslate**()：子弹移动

void **OnTriggerEnter2D**(Collider2D collider)：碰撞检测

# Bullent\_Sakuya\_02 : ABullent（咲夜的第二种子弹，与01相同）

## 变量

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 int **attackPoint**：子弹攻击力（暂未使用）

public float **bullentSize**：子弹攻击判定半径（暂未使用）

public GameObject **hitEffect**：命中特效产生的GameObject

Vector3 **direction**：子弹实际速度方向，世界坐标

float **startTime**：子弹产生的时间点

public List<GameObject> **enemies**：场景中敌人列表（暂未使用）

public int **effect**：攻击效果的编号（暂未使用）

## 方法

void **CollisionDet**()：碰撞检测（可能被OnTriggerEnter2D代替）

void **BullentTranslate**()：子弹移动

void **OnTriggerEnter2D**(Collider2D collider)：碰撞检测

# AMoveMode（移动模块抽象类）

## 变量

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

public float **moveSpeed**：角色移动速度

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

## 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）

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

## 变量

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 **dashTime**：冲刺持续时间

public float **dashSpeed**：冲刺速度

public float **dashChargeTime**：冲刺后摇

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

bool **isMove**：角色是否移动中

float **inputX**：左右按键

float **inputY**：上下按键

bool **isDash**：角色是否冲刺中

float **dashStartTime**：角色冲刺开始时间

Vector2 **playerDirection**：角色朝向

## 方法

public override void **Move**()：角色移动

public override void **IsDelayed**()：硬直时的操作

# 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（通常状态的受伤模式）（暂未使用）

## 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