



# GameManager

说明：目前游戏分为四个状态，分别是菜单状态、准备、进行游戏和游戏结束

## 重要属性：

```
public static GameManager Instance;

private int currentScore;           // 当前分数
private float gameTime;             // 游戏时间
private float shootTime = 0;        // 射击冷却时间

public Animation gunAnimation;       // 射击动画
public bool superMode = false;      // 超级模式开启标志
private GameState gameState;        // 当前游戏状态
```

## Awake

1. 设置单例Instance

## 游戏状态设置

1. 根据初始游戏状态类型执行不同逻辑
  - a. 初始是菜单状态
    - i. 使用DuckController<创建一个菜单鸭子>
    - ii. 调用UIManager<进入菜单状态>
  - b. 初始是准备状态
    - i. 调用UIManager<进入准备状态>
    - ii. 播放ready go音效

- iii. 有限播放动画，动画播放完毕<**开始游戏**>
- c. 初始是游戏状态
  - i. 重置当前分数和游戏时间
  - ii. 更新默认时间/分数/武器cd UI
  - iii. 调用UiManager和DuckManager的<**进入游戏**>方法进入游戏

## Update

1. 根据当前每一帧中游戏状态执行不同逻辑
  - a. 如果当前是菜单状态
    - i. <**检查玩家是否点击到了鸭子**>
    - ii. 如果当前菜单鸭还未死亡并且玩家点击到了鸭子
      1. 执行<**射击鸭子**>方法并设置<**鸭子死亡**>
      2. 切换到游戏状态
  - b. 如果当前是游戏状态
    - i. 检查当前还有无剩余时间，当没有剩余时间时游戏切换到结束状态
    - ii. 更新游戏倒计时时间并更新对应UI
    - iii. 更新射击CD并更新对应UI
    - iv. 检查当前是否进行了射击（射击cd  $\leq 0$  并且玩家点击了屏幕），如果玩家射击了则执行下列逻辑
      1. 重置武器cd
      2. <**检查玩家是否点击到了鸭子**>
        - a. 如果点击到了鸭子：
          - i. 根据鸭子配置更新游戏分数和对应UI
          - ii. 击到的鸭子执行死亡方法<**鸭子死亡**>
          - iii. 执行<**射击鸭子**>方法，播放动画和音效
        - b. 如果没有点击到：使用枪射击刚刚点击位置<**射击某点**>

- c. 如果当前是结束状态
  - i. 调用DuckManager<停止游戏>方法停止生成鸭子
  - ii. 调用UIManager<游戏结束>显示结束UI和当前分数

## Start

1. 播放进入菜单场景音效
2. 设置当前游戏状态为菜单状态

## 开始游戏

1. 将游戏状态设置为开始游戏

## 检查玩家是否点击到了鸭子

1. 如果 Input.touchCount > 0并且Input.GetTouch(0).phase == TouchPhase.Began则认为现在攻击到了鸭子
  - a. 使用相机发射射线扫屏幕上点击的点，并使用 `Physics2D.Raycast` 得到 `RaycastHit2D` 的信息
  - b. 如果当前hitInfo.collider不为空并且当前hitInfo有DuckController组件则认为当前点到了鸭子，此时返回duckController

## 射击某点

1. 根据传入的target position设置当前打击动画的position
2. 判断当前枪是否需要旋转，例如打击屏幕左侧时枪偏左打击屏幕右侧枪偏右
3. 播放射击动画
4. 播放射击音效

## 射击鸭子

1. 获得当前鸭子位置和配置信息中的枪射击偏移结果，累加后调用<射击某点>方法

## 2. <打击鸭子的音效>