



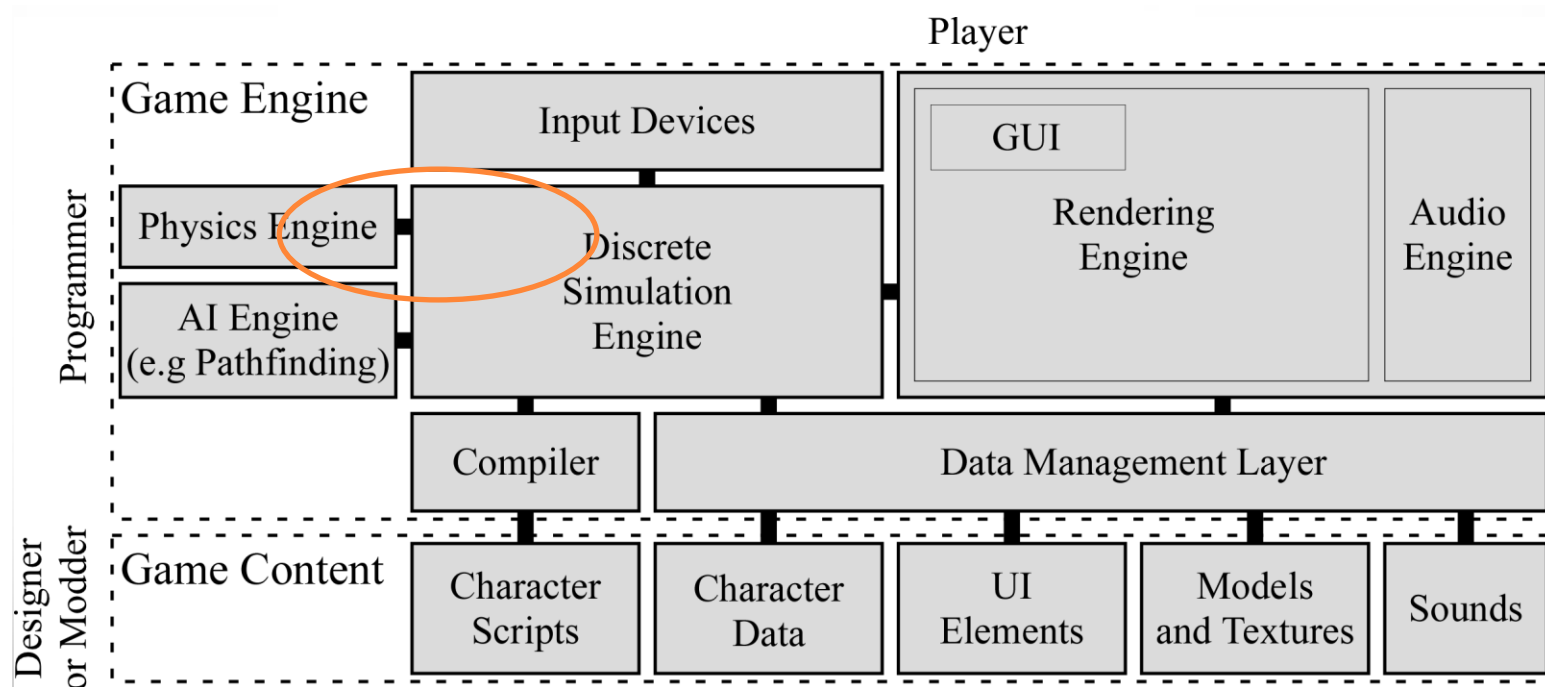
# INTRODUCTION TO COMPUTER 3D GAME DEVELOPMENT

Physics Engine and Collision Detect

潘茂林, [panml@mail.sysu.edu.cn](mailto:panml@mail.sysu.edu.cn)

中山大学·软件学院

# 游戏引擎架构



# 目录

- 物理引擎基础
  - 运动建模
  - 碰撞与几何
- Unity物理引擎实现
  - 刚体组件
  - 碰撞器组件
  - 课堂实验（一、二）
- 离散物理引擎
  - 常见问题
  - 物理引擎编程
- 力与场
- 面向对象设计思考
  - GoF设计模式：Adapter模式



# 物理引擎基础

## (1) 物体运动类型

### ○ 运动学 (Kinematics)

- 不考虑外部力作用的运动
- 仅考虑物体位置、速度、角度 ...
- 将一个物体作为几何部件
- 实现：使用变换

### ○ 动力学 (Dynamics)

- 考虑外部力对物体运动的影响
- 包括重力、阻力、摩擦力等，以及物体的重量和形状，甚至弹性物体
- 通常模拟物体在现实世界中的运动
- 通常将一个物体当作刚体



# 物理引擎基础

## (2) 物理引擎的职责与重要性

### ○ 物理引擎（Physics Engine）

- 物理引擎是一种软件组件，用于仿真物理世界运动。这种仿真包括刚体力学、流体力学以及碰撞检测。
- 物理引擎通过为刚性物体赋予真实的物理属性，在外部力的作用下，计算运动、旋转和碰撞。

### ○ 物理引擎与加速

- PhysX
  - NVIDIA GPU
- Havok
  - Intel CPU
  - AMD CPU+GPU



# 物理引擎基础

## (3) 物理计算的学习问题

- 物理仿真是复杂的
  - 原理与计算，这不是几节课能解决的问题
  - 您可能不解足够运动学知识，高深的数学
  - 计算复杂，CPU、GPU大厂出手收购说明一切
- 最简单的方式 – 使用物理引擎，你还需要：
  - 了解物理运动的基本知识
  - 物理引擎常用的 API
  - 物理引擎使用问题与场景
  - 有效使用物理引擎



# 物理引擎基础

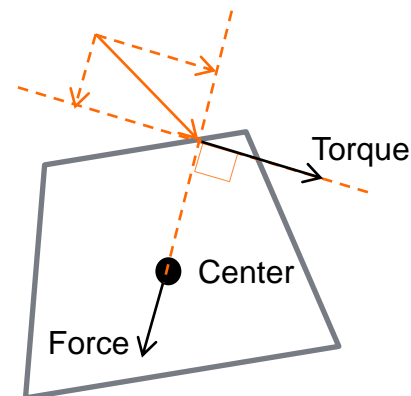
## (4) 运动与物体建模

### ○ 通常认为物体为刚体

- 刚体在运动中，物体内部任意两点几何关系保持不变
- 现实世界中不存在严格的刚体！
- 齿轮、绳索、滑轮不属于刚体，但属于该引擎
- 因此：物理引擎不只是刚体

### ○ 物体运动是点模型，仅考虑

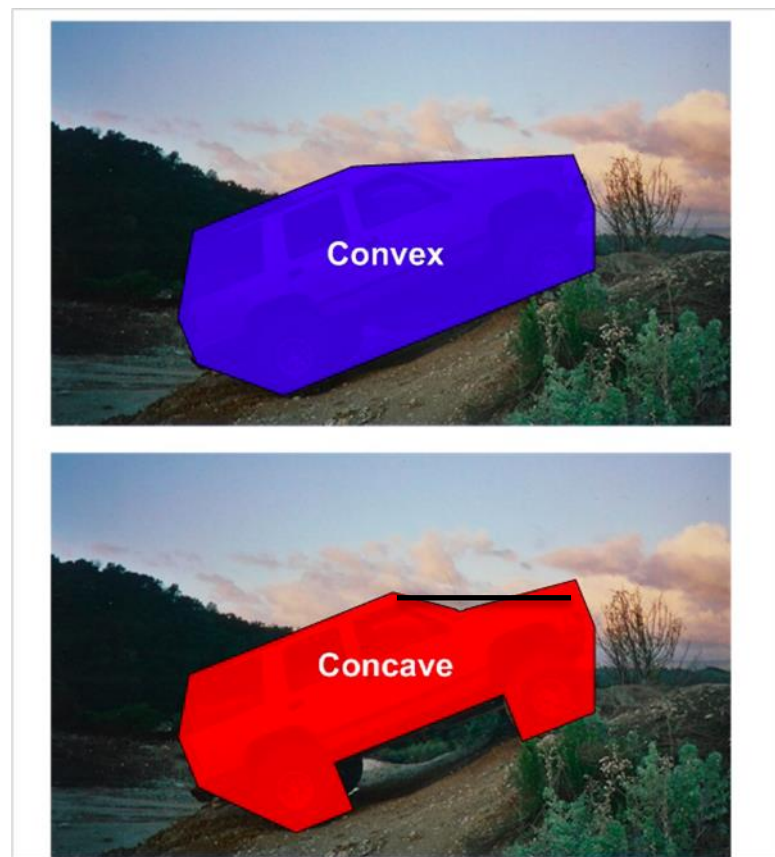
- 质量（重量）
- 中心
- 质心（不需要考虑形状）
  - 假设1：物体是均质的
  - 假设2：中心与质心重合
- 力（Force）与力矩（Torque）



# 物理引擎基础

## (5) 碰撞与几何

- 当物体发生碰撞
  - 质点模型不能用？
  - 必须解决在那点相撞
- 可计算的模型
  - 凸（**Convex**）物体
  - 如何判断凸多边形？
- 获取可计算的形状
  - 分解为多个凸物体组合
  - 2D：物体分解为若干三角形
  - 3D：凸物体表面用三角形，分解为四面体表示





# UNITY 物理引擎实现

## (1) 常用物理组件

- 内置支持引擎
  - physX 引擎
- Rigidbody 刚体组件
  - 物体运动控制属性
- Collider 碰撞器
  - 物体碰撞与碰撞检测属性
- Mesh 网格
  - 物体形状属性
- Joint 连接器（自学）
- 其他
  - 衣服、常力、用户定义的力场等等



# UNITY 物理引擎实现

## (1) 刚体(Rigidbody)组件

### ○ 为游戏对象添加刚体

- 菜单 component → physics → rigidbody

- 刚体属性

- 质量 mass

- 阻力 Drag

- 角阻力

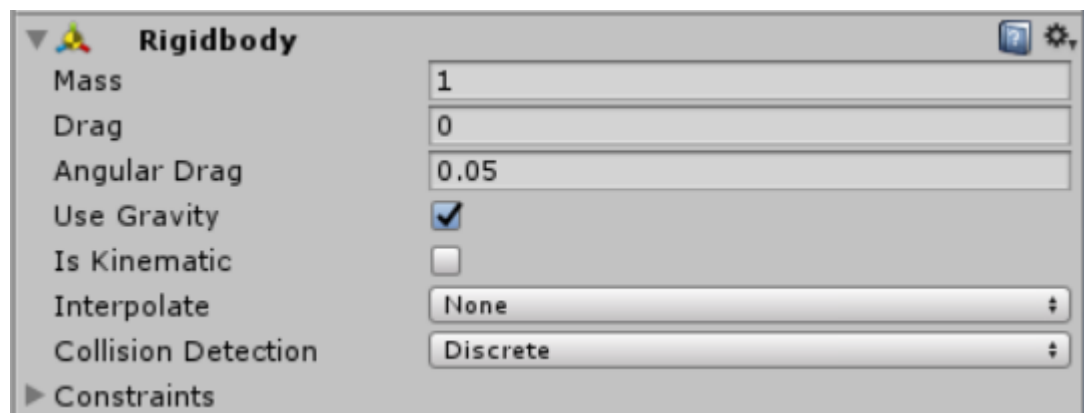
- 使用重力

- 是运动学控制。表示该物体是静态的，不受力影响

- 插值。官网说法当运动抖动时需要插值

- 碰撞检测。默认是离散检查。高速物体需要连续检测

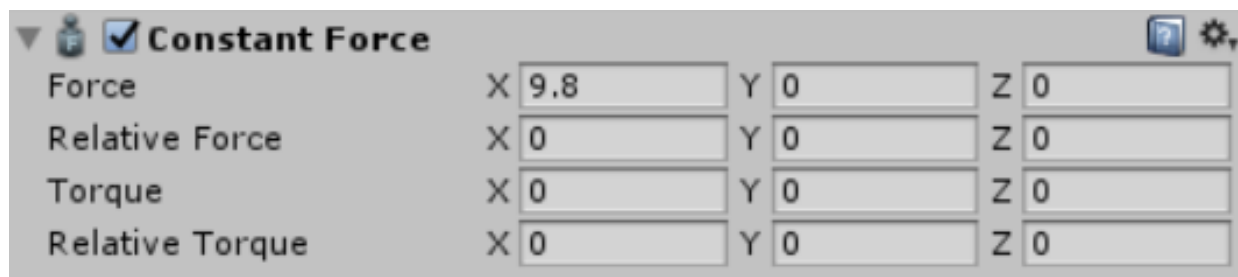
- 控制运动自由度。



# UNITY 物理引擎实现

## (1) 力与运动基础

- 系统默认的力作用在刚体重心
  - 力矩，产生旋转
- 力与运动
  - $\vec{f} = m * \vec{a}$
- 为物体添加力
  - 菜单 component → physics → constant force



- 力（Force）是一个作用在重心的向量
- 力矩（Torque）是产生角加速度的力



# 课堂实验（一）

## 验证力与刚体运动

### ○ 新建项目 physics

- 自由落体

1. 添加一个 cube
2. 为 cube 添加刚体，使用 重力
3. 运行（观察物体自由落体）

- 与静态物体碰撞

1. 添加 plane 使得  $y=0$ ；并使 cube 的  $y=3$
2. 运行（观察到 cube 落在 plane 上）

- 力的作用

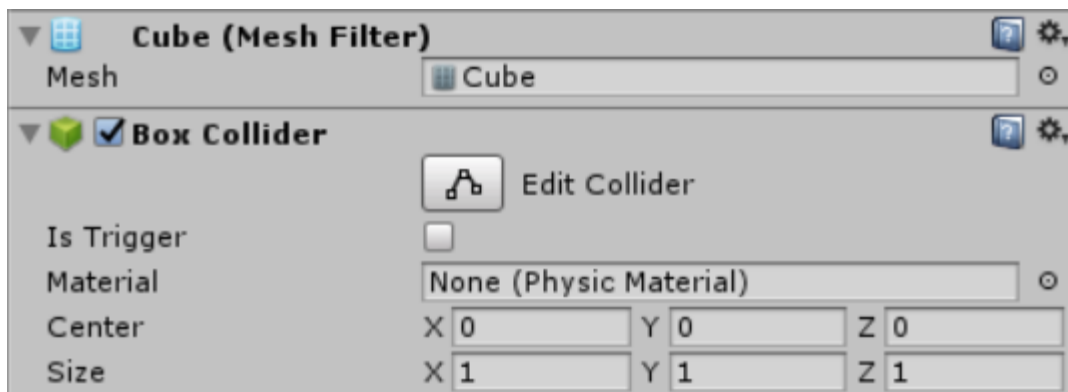
1. Cube 添加 Force 组件
2. 设置力  $y=9.8$  或  $y=10$
3. 运行（请问系统默认  $g=9.8$  或  $10$ ）
4. 设置力  $f=(10,0,0)$
5. 运行（观察由于碰撞产生的旋转）



# UNITY 物理引擎实现

## (2) 碰撞(COLLIDER)组件

- 有形物体的属性
  - 一个 mesh 组件
  - 一个或多个 Collider 组件
- Collider 常见类型
  - 基本碰撞器
    - Box Collider
    - Sphere Collider
    - Capsule Collider
  - 复杂碰撞器
    - Mesh Collider
    - Wheel Collider
    - Terrain Collider



# UNITY 物理引擎实现

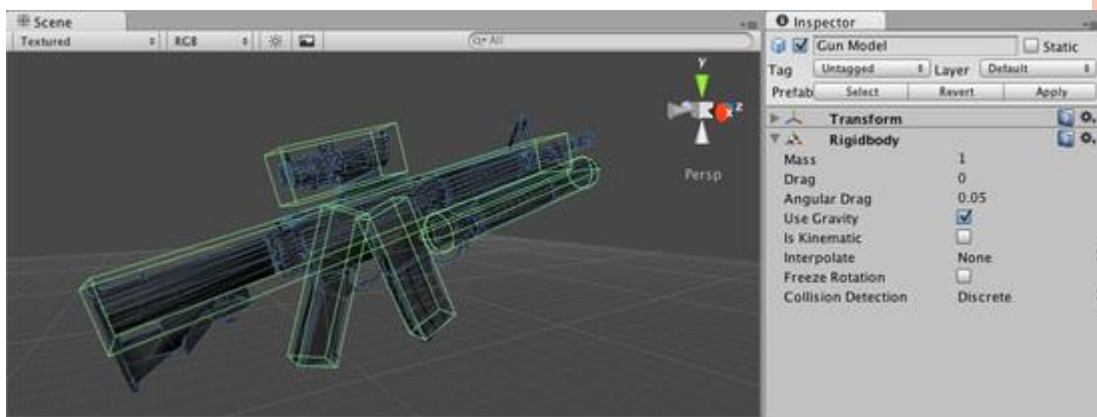
## (2) 复合碰撞器(COMPOUND COLLIDER)

### ○ Mesh 碰撞器是万能的？

- Too young! 能做碰撞计算的只能是凸的！！
- 处于性能的考虑，基本碰撞器是最有效的
- 但是：... ..

### ○ 复合（组合）物体

- 复杂形状分解
- 用变换组合
- 组合1：？
  - 仅 Root 对象拥有刚体，并拥有多个基本碰撞器
- 组合2：？
  - Root对象拥有刚体，每个对象拥有基本碰撞器



# UNITY 物理引擎实现

## (2) 碰撞交互(COLLIDER INTERACTIONS)

### ○ 一些概念

- 静态碰撞器：没有刚体属性物体的碰撞器
- 刚体碰撞器：有刚体属性物体的碰撞器，且是凸的
- 运动学刚体碰撞器

### ○ 发生碰撞的条件

|  | 静态碰撞器 | 刚体碰撞器 | Collider<br>运动学刚体碰撞器 |
|--|-------|-------|----------------------|
| Static Collider 静态碰撞器                    |       | Y     |                      |
| Rigidbody Collider 刚体碰撞器                 | Y     | Y     | Y                    |
| Kinematic Rigidbody Collider<br>运动学刚体碰撞器 |       | Y     |                      |



# UNITY 物理引擎实现

## (2) 碰撞事件(COLLISION EVENTS)

### ○ 碰撞消息与事件

- 消息

- OnCollisionEnter, OnCollisionExit, OnCollisionStay

- 事件与碰撞结构

- 处理消息的行为（脚本）接收该物体上刚体/碰撞器发出的消息
  - 例如：

```
5 public class CollisionRev : MonoBehaviour {
6
7     void OnCollisionEnter(Collision collision) {
8         Debug.Log ("Collision @ " + Time.frameCount.ToString());
9         Debug.Log ("GameObject is :" + collision.gameObject.name);
10        if (collision.collider) Debug.Log ("Collider belong to :" + collision.collider.gameObject.name);
11        if (collision.rigidbody) Debug.Log ("Rigidbody belong to :" + collision.rigidbody.gameObject.name);
12        foreach (ContactPoint contact in collision.contacts) {
13            Debug.DrawRay (contact.point, contact.normal, Color.white);
14        }
15    }
```





# 课堂实验（二）

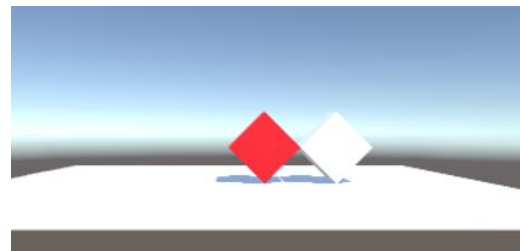
## 碰撞与碰撞事件接收

### ○ 碰撞与事件

- 实验对象：plane, cube（刚体），事件接收程序
- plane + 接收程序的输出？ cube + 接收程序的输出？

### ○ 组合碰撞器

- 实验对象：plane, cube1(0,0,0), cube2(1,1,0)(red), root(0,3,0) 是 cube 的容器，事件接收程序
- 组合1
  - root 是刚体，cube 有碰撞器
- 组合2
  - root 是刚体，且有两个碰撞器， cube 碰撞器
- 组合3
  - root 不是刚体，cube各自拥有刚体和碰撞器
- 事件接收器挂载不同物体，并观察现象



# 离散物理引擎

## (1) 离散计算与轨迹

### ○ 线性动态：以简单抛物线为例

- 力:  $F(p,t)$
- $p$ : 当前位置
- $t$ : 当时时间

### ○ Update 的公式

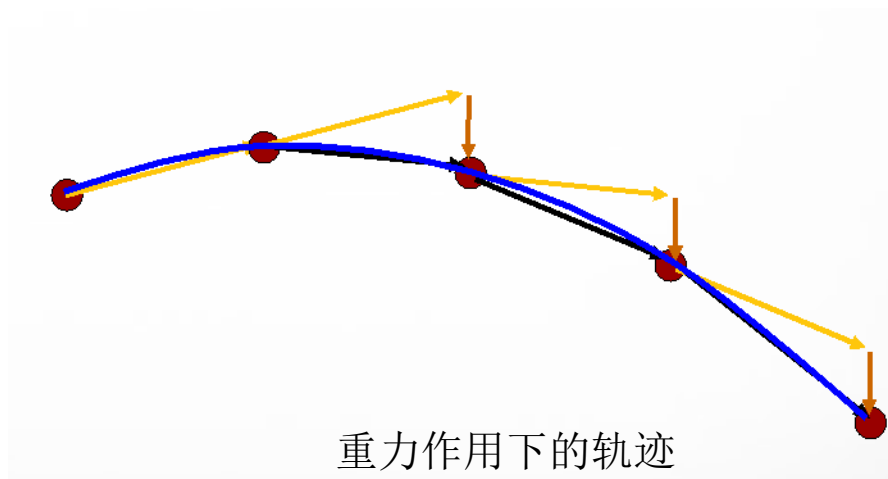
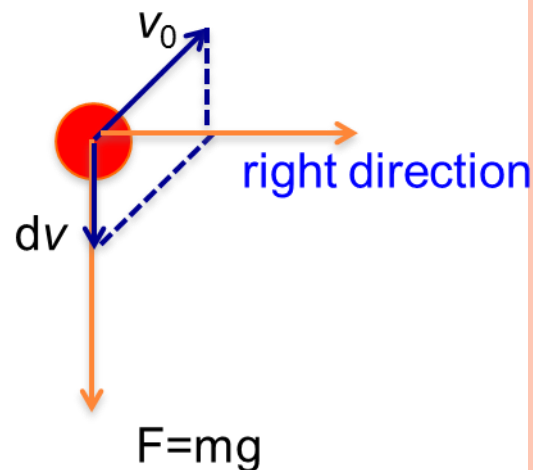
$$a_i = F(p_i, i\Delta t)/m$$

$$v_{i+1} = v_i + a_i \Delta t$$

$$p_{i+1} = p_i + v_i \Delta t$$

### ○ 结果

- 一组不太平滑的点
- 近似算法， $\Delta t$ 越大则偏差越大（求解器内部做插值）
- $\Delta t$ 要尽可能相等，否则就会感到跳跃

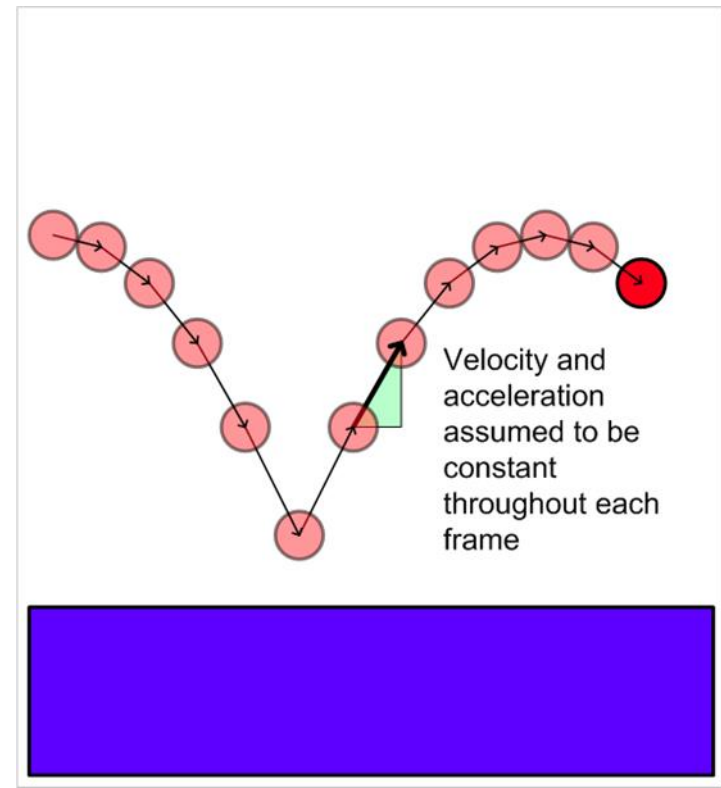


# 离散物理引擎

## (2) 线性动态计算问题

### 动画综合症 – Flipbook Syndrome

- 当你用慢动作播放运动...
- ?

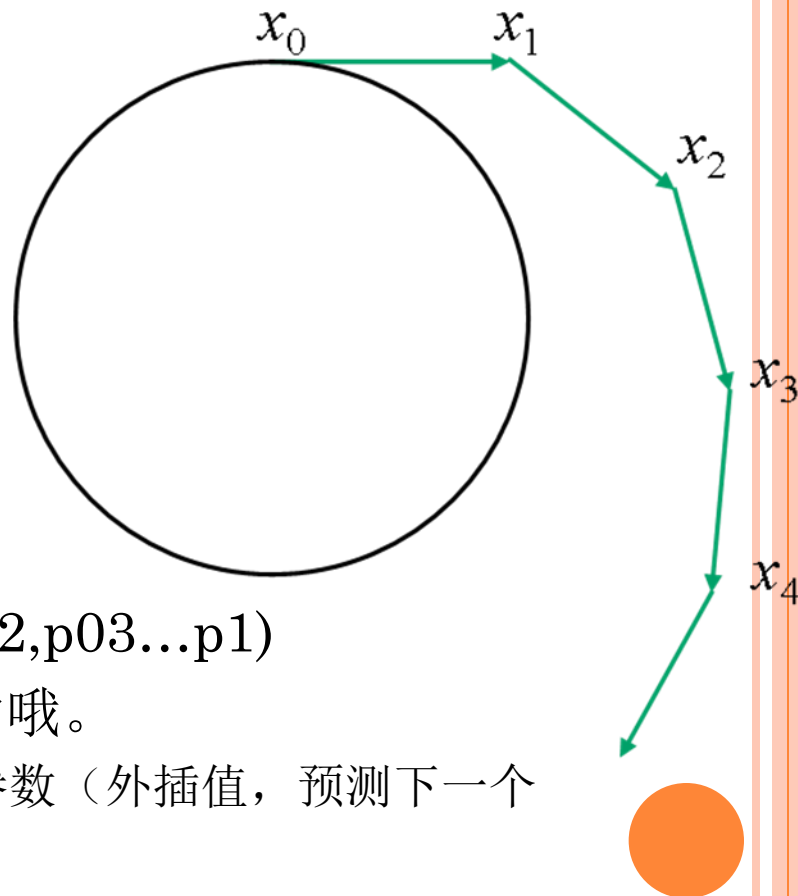


# 离散物理引擎

## (2) 线性动态计算问题

### 动画综合症 – Flipbook Syndrome

- 当你用慢动作播放运动...
- 当旋转运动比较快...
- 都是**积累误差**惹的祸!!!



### 常用解决方案

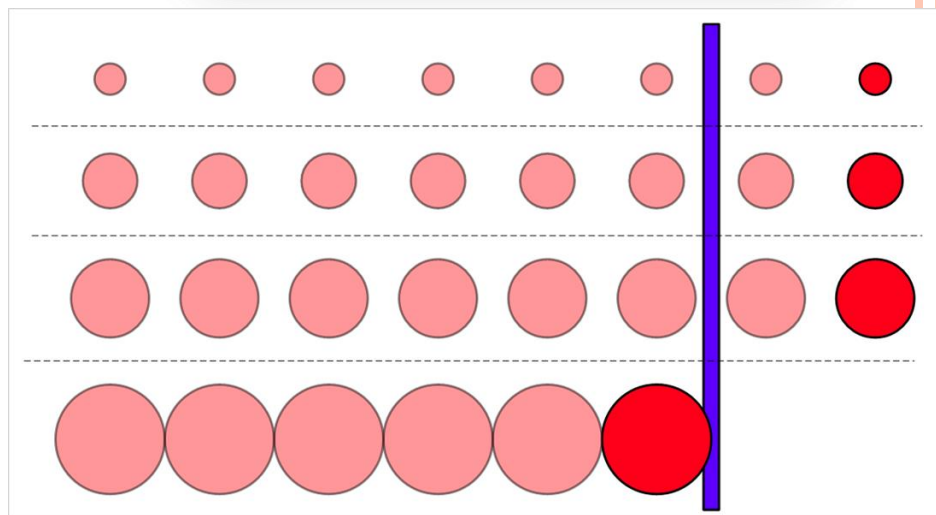
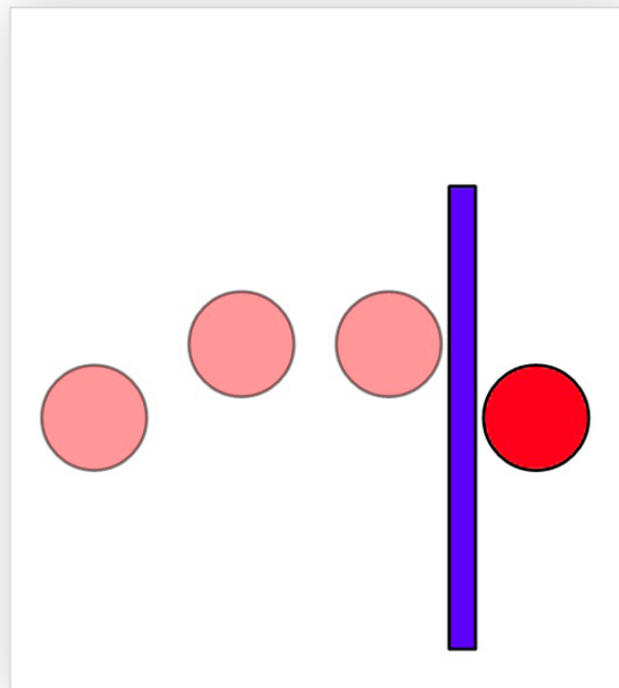
- 在每个计算帧做内插值( $p_0, p_{01}, p_{02}, p_{03} \dots p_1$ )
- 做旋转误差补偿, 但会使物体抖动哦。
  - `Rigidbody.Interpolate` 刚体的插值参数 (外插值, 预测下一个位置消除抖动)

# 离散物理引擎

## (3) 碰撞计算问题

### ○ Tunneling(穿越效应)

- 当物体运动足够快
- 或物体又小又快
  - 射击的子弹
  - 快速转动臂膀
- 不了解这些问题
  - 后果很严重



# 离散物理引擎

## (3) 碰撞计算问题

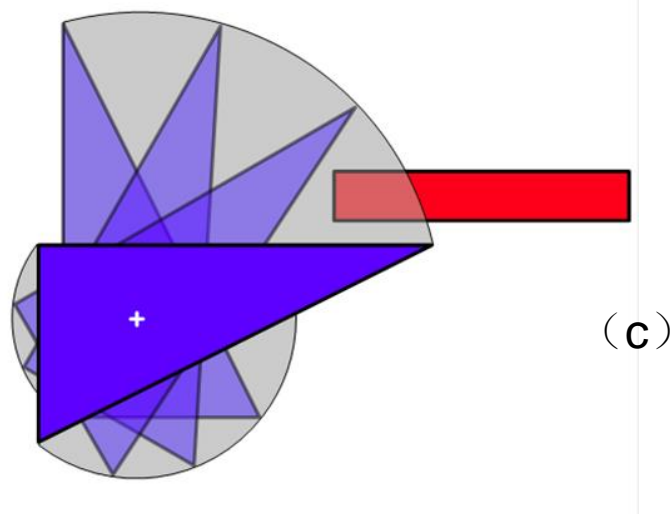
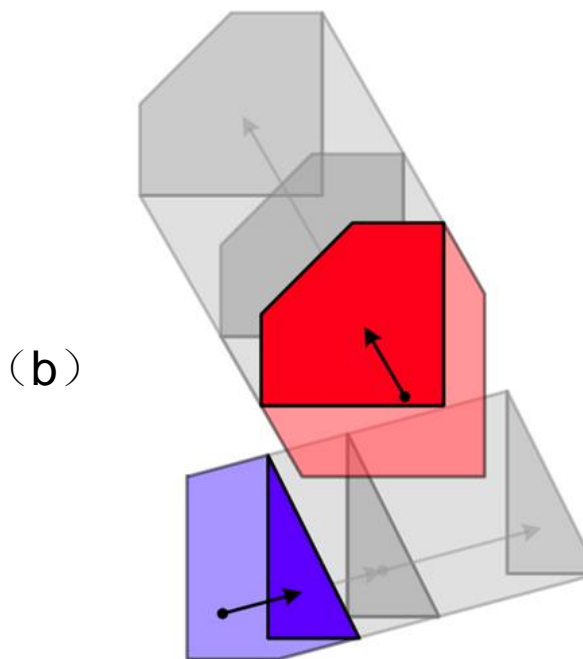
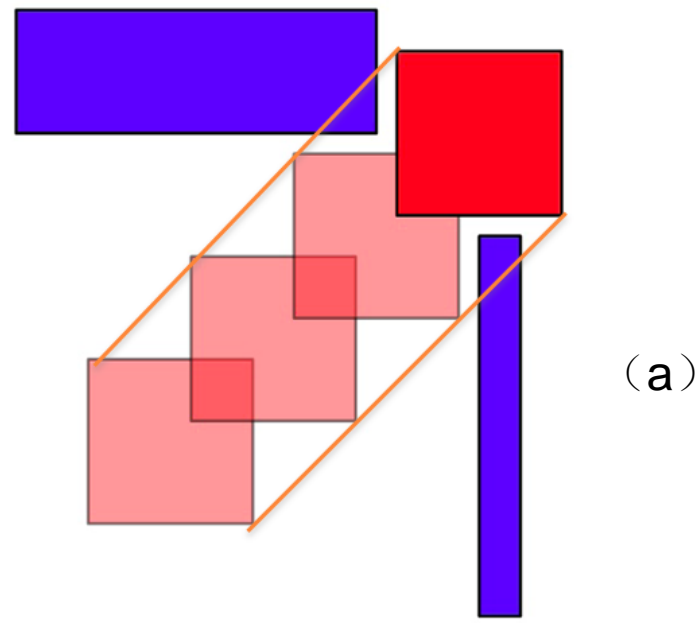
### ○ 解决方案

- 刚体碰撞检测属性

1. 连续检测 (a)

2. 连续动态检测 (b)

3. Wheel动态碰撞 (c)



# UNTIY物理引擎使用

## (1) 基本注意事项

### ○ Fixed-update

- 修改物理引擎相关组件属性，请使用Fixed-Update
- 确保物理引擎计算

### ○ 根据场景需要设置刚体属性

- 内插值、外插值的使用
- 连续检测
- 特殊情况，用射线碰撞+定时销毁子弹替代碰撞

### ○ 复合物体

- 父子对象不能同时拥有刚体

### ○ 其他

- 不要逐帧移动一个静态碰撞器
- 不要同时使用物理运动与运动学移动物体



# UNTIY物理引擎使用

## (2) 刚体组件属性与方法

### ○ 刚体运动常用属性

- 速度 (velocity) 角速度 (angular velocity)
- 质心 (center of mass)
  - 物体中心 (坐标原点) 与质心一致是特例
  - 特别是组合物体, 设置质心会使运动效果更真实
- 自定义内插值步数 (solver iteration count)
  - 提高/降低计算轨迹精度
- 刚体休眠 (sleep)
  - 设置最低速度, 减少引擎计算量
- 限制运动自由度





# UNTIY物理引擎使用

## (2) 刚体组件属性与方法

### ○ 对刚体施力或力矩

1. `Rigidbody.AddForce(Vector3 force, ForceMode mode);`
2. `Rigidbody.AddForceAtPosition(Vector3 force, Vector3 position, ForceMode mode);`
3. ... ..

### ○ 力的模式(ForceMode)

#### Values 值

##### Force

Add a continuous force to the rigidbody, using its mass.  
添加一个可持续力到刚体，使用它的质量。

##### Acceleration

Add a continuous acceleration to the rigidbody, ignoring its mass.  
添加一个可持续加速度到刚体，忽略它的质量。

##### Impulse

Add an instant force impulse to the rigidbody, using its mass.  
添加一个瞬间冲击力到刚体，使用它的质量。

##### VelocityChange

Add an instant velocity change to the rigidbody, ignoring its mass.  
添加一个瞬间速度变化给刚体，忽略它的质量。



# UNITY物理引擎使用

## (2) 刚体运动属性与方法

### ○ 代码替代重力

```
5 public class Gravity : MonoBehaviour {  
6  
7     void FixedUpdate () {  
8         Rigidbody rigid = this.gameObject.GetComponent<Rigidbody> ();  
9         if (rigid) {  
10             rigid.AddForce (Vector3.down * 9.8f);  
11         }  
12     }  
13 }  
14 |
```

### ○ 移动刚体

- 尽管不能使用transform移动刚体，刚体提供
  1. MovePosition (position : Vector3)
  2. MoveRotation (rot : Quaternion)
- 它们会在本帧物理引擎计算结束后执行



# UNITY物理引擎使用

## (3) 碰撞盒—作为触发器

### ○ 触发器 (Trigger)

- 一种碰撞盒，产生与运动无关的碰撞，并触发事件
- 用 Is Trigger 属性表示

### ○ 刚体碰撞器的代价

- 刚体的碰撞盒是近似形状（运动计算性能与效果折中）
  - 普通游戏角色的碰撞器仅是胶囊模型
  - 除了部分物理仿真应用，并不需要精确模型
- 不同刚体的碰撞盒不能相交（不能穿透）

### ○ 触发器的优势

- 碰撞计算是高效的
- 很多应用的碰撞不需要物理影响运动
  - 子弹“爆头”的实现
- 触发器可以放在任意物体，任意位置



# UNTIY物理引擎使用

## (3) 碰撞盒－作为触发器

### ○ 触发器事件（Trigger Event）触发条件

| Trigger messages are sent upon collision<br>碰撞后有触发信息 |                          |                             |  |                                    |                                       |  |
|--|--------------------------|-----------------------------|--|------------------------------------|---------------------------------------|--|
|  | Static Collider<br>静态碰撞器 | Rigidbody Collider<br>刚体碰撞器 | Kinematic Rigidbody Collider<br>运动学刚体碰撞器 | Static Trigger Collider<br>静态触发碰撞器 | Rigidbody Trigger Collider<br>刚体触发碰撞器 | Kinematic Rigidbody Trigger Collider<br>运动学刚体触发碰撞器 |
| Static Collider 静态碰撞器                                |                          |                             |  |                                    | Y                                     | Y  |
| Rigidbody Collider 刚体碰撞器                             |                          |                             |  | Y                                  | Y                                     | Y  |
| Kinematic Rigidbody Collider<br>运动学刚体碰撞器             |                          |                             |  | Y                                  | Y                                     | Y  |
| Static Trigger Collider<br>静态触发碰撞器                   |                          | Y                           | Y  |                                    | Y                                     | Y  |
| Rigidbody Trigger Collider<br>刚体触发碰撞器                | Y                        | Y                           | Y  | Y                                  | Y                                     | Y  |
| Kinematic Rigidbody Trigger Collider<br>运动学刚体触发碰撞器   | Y                        | Y                           | Y  | Y                                  | Y                                     | Y  |



# UNITY物理引擎使用

## (3) 触发器应用案例

### ○ 触发器与碰撞器的选择

- 飞碟游戏中飞碟
  - 触发器，胶囊体，如果碰撞后飞碟立即回收
  - 碰撞器，胶囊体，期望物体落地后弹跳一会
- 《反恐精英CS》玩家角色
  - 碰撞器，胶囊
  - 触发器，头部球/或长方体
- 天空飞行的鸟
  - 碰撞器，矩形体（可变，展翅与死后不展翅不一样）
- 炮塔
  - 碰撞器，胶囊体用于物体主体子对象
  - 触发器，胶囊体用于攻击范围子对象
- 子弹
  - 触发器，方体。条件，遇到任何物体立即回收



# UNITY物理引擎使用

## (3) 应用案例分析

### ○ 打靶游戏

- 靶标对象
  - Root
  - Target（显示网格）
  - Objective1..5（一圈一个，计分用）
- 箭对象
  - Root（射中后，变为运动学刚体）
  - 箭头（射中后，箭头会inactive）
  - 箭身（射中后，箭身插在靶标上）



### ○ 课堂问题与课后作业

- 请回答每个对象子物体的刚体、碰撞器的设定？
- 课后请完成该第一人称游戏（使用物理）！
- 添加一个风向和强度标志，提高游戏难度



# UNTIY物理引擎使用

## (4) 碰撞计算原理与过程

### ○ Axis-Aligned Bound Box(AABB)

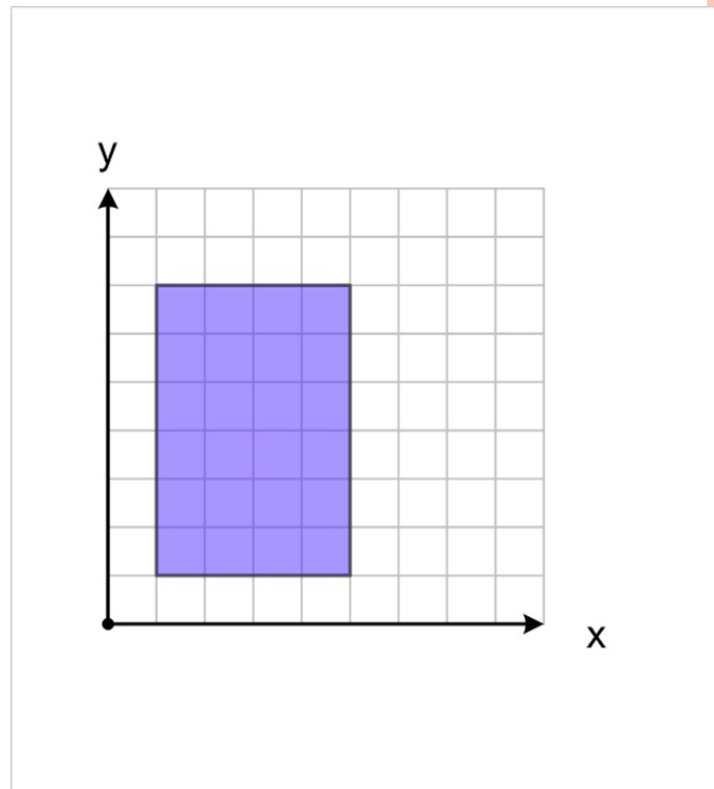
- 计算简单，高效
- 与事实差异大

### ○ 物理碰撞计算过程

1. AABB碰撞
2. 物体碰撞

### ○ Collider的重要属性与方法

- bound
- ClosestPointOnBounds 用来计算受到爆炸伤害点数
- Raycast 用来加速计算碰撞

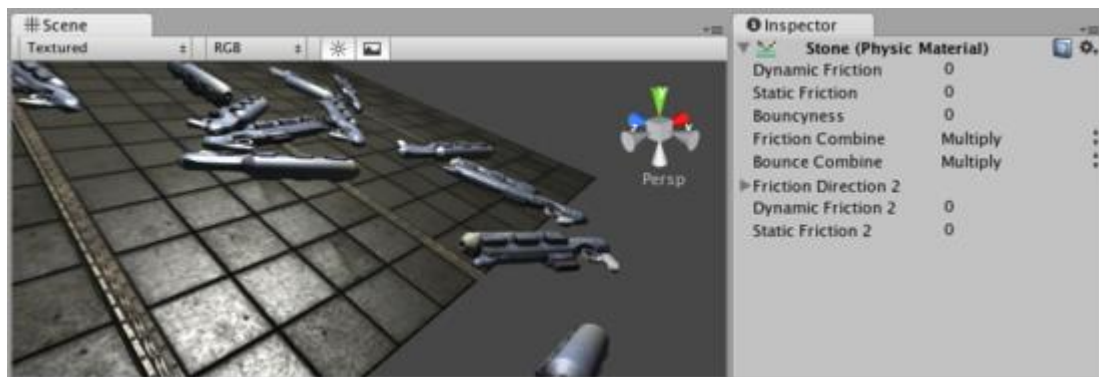


# UNTIY物理引擎使用

## (5) 物理材料与碰撞效果

### ○ 作用

- 物理材质用来调节碰撞物体的摩擦力和弹力效果。
- 创建物理材质：菜单 Assets → Create → Physic Material。然后从项目视图拖拽物理材质到场景的一个碰撞器上

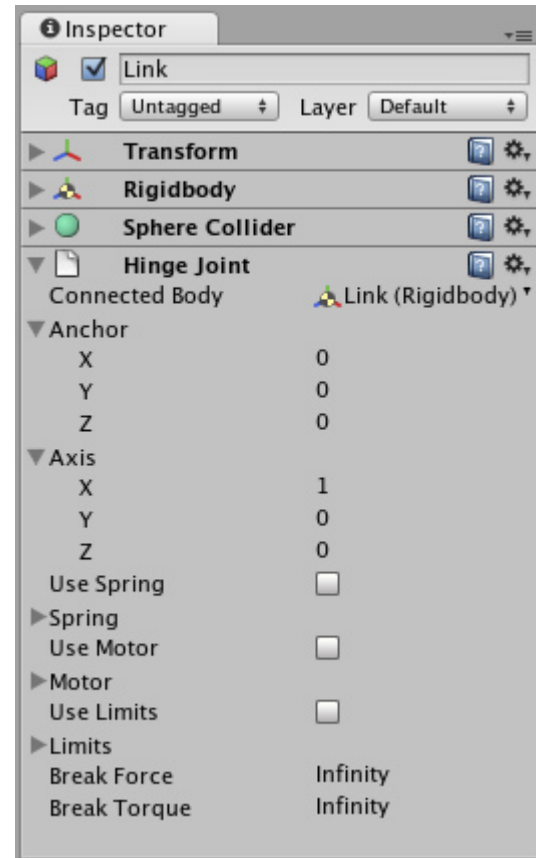
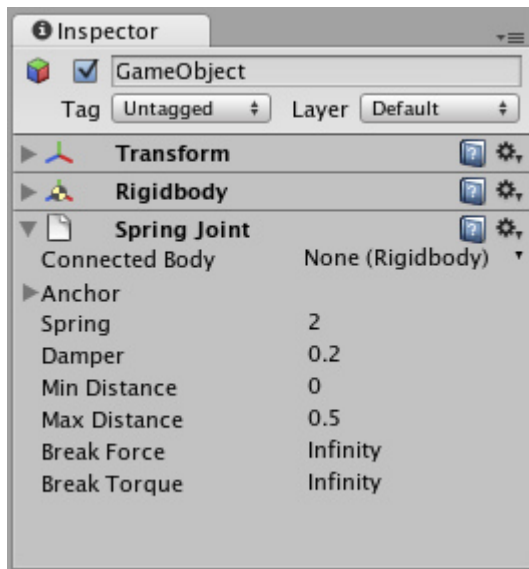




# UNTIY物理引擎使用

## (6) 关节与弹簧

- Hinge Joint 铰链关节
  - 用于连接两个刚体
- Spring Joint 弹簧关节

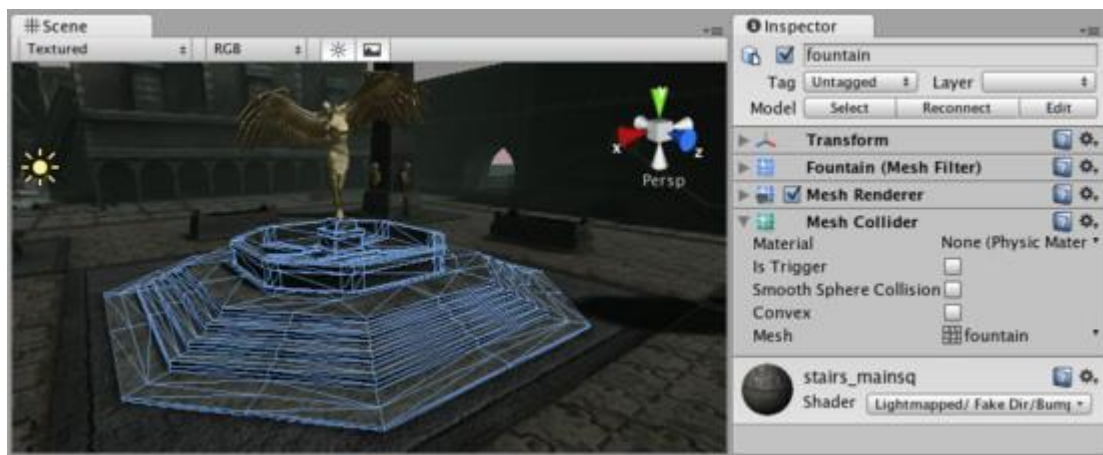


# UNITY物理引擎使用

## (7) 其他注意事项

### ○ Mesh Collider 网格碰撞器

- 能不用就不用（性能考虑）
- 只有凸的网格能与网格碰撞
- 任意网格可以与基本碰撞盒碰撞



### ○ 注意物体设计大小

- Unity 空间单位 1 大约 1米，所以人高大约2
- 物体缩小、放大会影响性能



# 物理引擎中级技术

## (1) 势能与场 (POTENTIAL FIELDS)

### ○ 现实世界的场

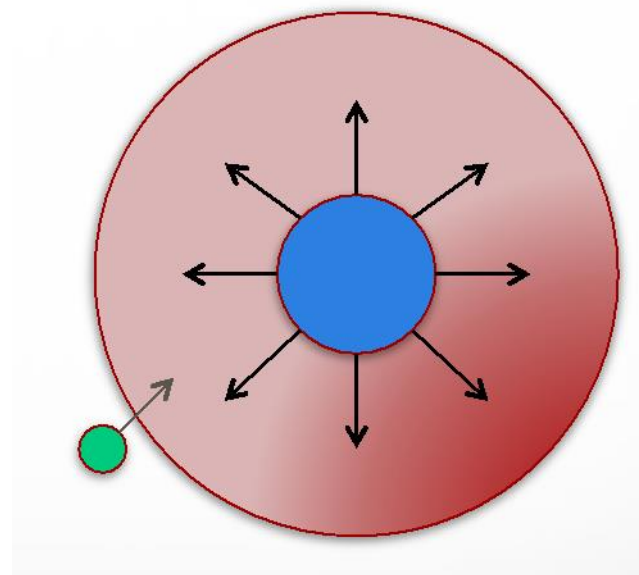
- 引力
- 电磁力场
- ... ..

### ○ 社会中的场

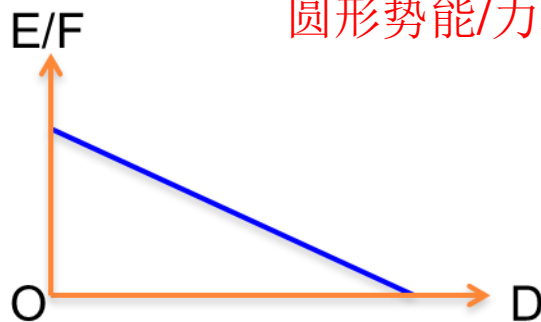
- 吸引力
- (食物) 诱惑力

### ○ 场与运动

- 场使运动更真实与智能
- 常见力场



圆形势能/力场



力与距离的关系



# 物理引擎中级技术

## (2) 势能与场应用

### ○ 爆炸效果

```
void OnCollisionEnter(Collision col)
{
    //如果碰撞物是上方落下的小球，进行爆炸处理
    if (col.transform.name == "Sphere")
    {
        //定义爆炸半径
        float radius = 3.0f;
        //定义爆炸位置为炸弹位置
        Vector3 explosionPos = transform.position;
        //这个方法用来返回球型半径之内（包括半径）的所有碰撞体collider[]
        Collider[] colliders = Physics.OverlapSphere(explosionPos, radius);

        //遍历返回的碰撞体，如果是刚体，则给刚体添加力
        foreach (Collider hit in colliders)
        {
            if (hit.rigidbody)
            {
                hit.rigidbody.AddExplosionForce(600, explosionPos, radius);
            }
            //销毁炸弹和小球
            Destroy(col.gameObject);
            Destroy(gameObject);
        }
    }
}
```

爆炸效果：<http://blog.csdn.net/jukaiblog/article/details/10343211>

# 物理引擎中级技术

## (2) 势能与场应用



# 课堂实验（三）

## 力场与爆炸效果

### ○ 完成爆炸效果

- 设计四个物体在 plane 上
- 从高空掉落一物体，在地面发生爆炸
- 产生炸飞的效果

### ○ 巡航与感应攻击

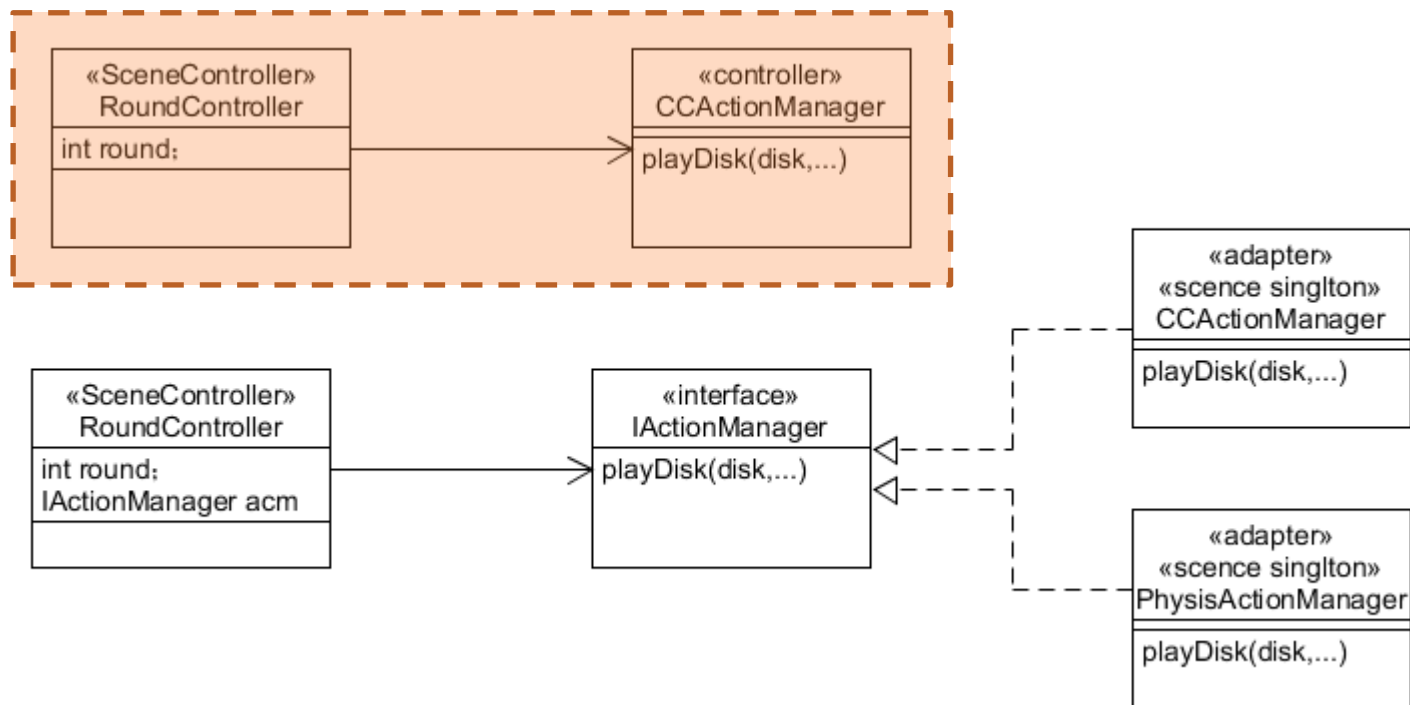
- A程序员开发一个巡航带自动攻击机器人
- 巡航设计：
  - 机器人使用巡航行为（FixedUpdate）
  - 巡航行为包含一组目标点
  - 当前目标点产生指向目标的恒速力。到达则使用下一个目标
- 攻击设计：
  - 机器人附加带触发器的攻击子对象
  - 当触发对象名为 boss\* 时发起攻击行动



# 面向对象设计思考

## (1) 业务需求

- 我们学完物理运动，现在的需求是：
  - 不想放弃 CCActionManager
  - 新建 PhysisActionManager
  - 新的设计如下图：



# 面向对象设计思考

## (2) ADAPTER 模式的价值

- 什么是 Adapter 模式?
- Adapter 模式使用场景
  - 你设计了电商网站，它有一个实用简单同一的支付接口，现在这个接口要对接工行、农行、...、微信、支付宝、财付通、...的接口。你明白设计如何应对业务变化的吗?
  - 在游戏中有哪些场景会用到 Adapter 呢?
  - ... ?



# 课程小结

## ○ 物理引擎基础

- 力与运动、物体碰撞的抽象
- Unity 物理系统的刚体、碰撞组件使用
- 离散系统动画综合症、穿越效应
  - 产生条件、原理、解决方案

## ○ 物理引擎编程

- 重心、力的使用
- 触发器的作用
- 碰撞器、触发器的综合应用
- 力场与智能

## ○ 面向对象设计技巧

- Adapter 模式



# 作业：

## ○ 修改飞碟游戏：

- 按设计图修改飞碟游戏
- 使它同时支持物理运动与运动学（变换）运动

## ○ 打靶游戏：

- 靶对象为 5 环，按环计分；
- 箭对象，射中后要插在靶上；
- 游戏仅一轮，无限 trials；
- 增强要求：
  - 添加一个风向和强度标志，提高难度

