Movieclip

三、三角学

角：两条线四个角

角度弧度：degree 180=pi rotation是角度

坐标系：通常是颠倒的坐标系 左上00 向右向下为正 **此时角度也是反的**，向下为正，向上为负 相对差角

三角形的边 :相对于角的 对边斜边 邻边 通常是把角放在原点00 邻边斜边垂直于坐标轴 sin cos

Arcsin arccos arctan反函数是由正弦余弦求角度

**反正切比较常用，可以由坐标求得角度**

旋转 ：角度是从x轴的平行开始算的 用斜边的两个端点坐标算 arctan dlt y/dlt x

正弦波形，,x均匀变化，y=sinx 的值 画出所有（x,y） 使用弧度

平滑地上下移动 （周期运动） y=y0+sin(angle)\*10 angle+=1 每次循环执行一次

线性垂直运动 x匀速运动 ，y上下移动 sin angle

心跳运动（缩放） scale= sin angle \*0.5

双角波形 两个角度 两个sin angle 分别x ,y或者任意两个不同属性 实现不规律的运动

圆和椭圆

正弦余弦配合圆形运动 一个角度 x=x0+cos angle\*n y=y0+sinangle\*n angle每桢增加

N可以看成是半径radius

椭圆运动： cos\*rx sin\*ry

勾股定理 求第三边

两点间距离

四、渲染技术

RGB颜色 十六进制 透明度 32位色（多出8位就是透明度）

色彩合成 位移操作符 << >> 提取红绿蓝

绘图api 清除（每桢） 填充 缩放模式 端点拐点

矩阵 颜色转换 滤镜 投影模糊发光斜角渐变斜角 颜色矩阵 卷积 置换图

动态滤镜

位图

五、速度和加速度

速度向量 有方向的速度

每桢增加 距离 速度 加速度

X+=vx y+=vx 斜向速度 沿着某个角度运动，速度为speed vx=speed cos （angle）这里angle是固定值

向量加法

箭头跟随鼠标旋转 rotation= 反正切 angle= arctan dy/dx 算鼠标坐标 角度可以变化

让箭头沿着方向运动：vx= cos angle speed

旋转速度 rotation+=vr vr就是每桢角度 vScale vAlpha

Angle+=vr

加速度 vx+=ax x+=vx 匀速运动把ax设为0 静止把vx设为0 可以通过按键修改速度加速度，控制 比如方向

**每桢自动运动**

重力加速度 每桢都 固定g vy每桢改变 ，y+=vy

沿着边上的加速度 或者力（force） 固定角度angle ax=cos angle\*force

通过控制 修改角度angle 或者使用vr转速度 提高速度或加速度 来使 对象自动运动

六、边界和摩擦力 顶 边缘 墙 地面

惯性：匀速运动 只有施加外力才改变 摩擦力：空气

来个数组push

边界超出范围： 移除 或者移回,反弹 left buttom 判断if x >right <

返回对象池

屏幕环绕，让对象从另一边出来

反弹：修改速度方向

摩擦力 反向的固定力 ax vx x

简单的摩擦力 每桢把速度\*0.9 0.8

七、交互

对对象进行点击 按下 移动 弹上 监听移动时 默认是所有在对象上的移动吧，所以开启移动监听时机是按下时，在松开后立刻清除掉监听

对区域点击：瞬间移动

拖拽 用mousemove事件或者 startdrag

投掷：拖拽，松开时沿着方向运动 拖动角度 拖动速度 修改vx vy 这个vx,vy如何算出来 或者用a 如何模拟出来 拖动的起始坐标，拖动用的时间 瞬时速度还是平均速度？ 拖动中计算瞬时速度 出蹚速度

八、缓动和弹性运动 滑动到目标点停止 在目标点两端来回，最后停在目标

速度变化：离目标点越远 移动越快

缓动曲线

要显得自然 每桢都要求距离distance vx=distance\*0.1 x+=vx 距离渐渐缩小，速度也渐渐缩小 easing=0.1 缓动系数

到达目标时应该让缓动停止，不是把速度变0完事吧，除非以后还要运动

缓动其实永远无法到达目标点，但是像素大小是有限的 我们可以用一个最小单位值 distance<min 差距在一像素内

可以把目标点设为动态的 比如鼠标， 小球缓动跟随效果

缓动可用于透明度 目标透明度 每桢改变透明度 alpha+=vAlpha vAlpha=(target-alpha)\*easing

旋转 目标角度 rotation+=

颜色

高级缓动

弹性运动springing 有像皮绳模型和弹簧模型（像皮绳可以是从0拉长，弹簧有固定长度）

弹性值 0.1 spring

目标位置

Ax和距离相关 ax=distance\*spring vx+=ax x+=vx

运动状态 ： 来回运动 永远不会停止 起点速度是0，加速度最高 目标点加速度0，速度最高

过了目标点后，加速度反向增加，速度降低，到了另一边速度为0

然后速度反向增加

需要一个摩擦力(这里简化为系数) friction 0.95 每次把速度vx直接\*friction

速度小于某个值就让其停止（vx等于0或直接从循环移除）

动态目标点 动态起点 如果目标和起点之间有某种连接（两球连一弹簧）

画一根像皮绳 drawLine 画线 动态画线

在运动中 发现可以通过拉动绳子动态修改双方距离（distance 然后ax就会突然变大变小，从而实现拖动弹簧连接物体的效果 注意此处目标点和此时的x都是实时的，所以distance也是实时的）

每桢动态

弹簧链 多个小球链起来 一个球的坐标是另一个球的目标点

先以缓动为例 小球跟随鼠标缓动 一个小球跟随另一个小球

受到多个弹簧影响的时候 多目标点 可以拉起任何一个目标点

添加多个力 只要多加几次ax就行了 vy+=ay vy+=g 有几个力在一个循环加几次

拖动的小球不同应该用不同的事件处理程序，使被拖动的球变为其他球的目标点

关于多目标点会有 多个力 多个不同的距离 速度是力的和 一个球，三个控制点

目标偏移 不用弹性线 用弹簧 长度不可能压缩为0 弹簧有正常长度 此时目标点应该相对于弹簧改变的位移

弹簧连接的两个目标 拉力会把两个球往中间拉 ，目标点在弹簧的中央？ 两边拉力一样

九、碰撞检测

像素检测 （检查非透明部分）

包围盒检测

点和对象的检测

对象和对象的碰撞检测

弹性碰撞，撞上了反弹

距离碰撞检测 圆形之间

多物体碰撞检测 循环遍历所有对象，需要两层遍历 n\*n次

格子检测

十、坐标旋转 角度反弹 如何在斜面反弹

指物体绕着某点旋转移动坐标， 不是自身旋转，自身旋转用rotation

Angle+=vr

Centerx x=ceterx+cos angle \*raduis 绕着中心 半径radius 旋转angle个角度

高级坐标旋转 x1=cosangle \*x-sin angle \*y y1=cox angle \*y+sin angle \*x

多物体旋转 对象遍历设计

角度反弹 反射角等于入射角 撞墙很好算，只用反一个方向

如果撞上的是斜面呢？

先旋转坐标系把斜面改平面，旋转速度 。算出反弹角度，速度，位置后 再以横轴为跟 旋转回原来的

碰撞检测getBounds x y w h left right buttom 碰撞检测 边界

多角度反弹 多个斜面

十一、撞球物理

质量 动量 p=mv 有方向 碰撞之后 动量守恒

动能守恒 动能损失 mv^2

单轴 vx 一条线上的碰撞

双轴 碰撞 碰撞角度 vx vy 和角度相关

旋转坐标系让碰撞的两球连线和x轴平行 旋转v角度

然后只用求vx 的动量和能量得到新vx1 vx2 此时y方向是无撞击的 两球vy不变

旋转回去

多物体碰撞：两层循环遍历

十二、粒子引力和万有引力

相同的对象 大量 具有碰撞等 拥有同样的行为 用对象池等结构保存

重力

万有引力 万有引力常数 force=G\*m1m2/distance^2 这个G在游戏中没必要

A1=Force/m1 a2=force/m2 注意这个力有方向 可以分解成x y 方向的

用 dx/两点距离 代替求角度

碰撞检测

粒子之间会互相吸引

行星系统：轨道运动

节点之间用弹力代替引力

画节点连线

十三、正向运动学：行走

互相连接的关节 链条 手臂

固定端 自由端

正向运动 走路 反向运动学 拖动 伸臂

关节 关节位置 关节旋转 关节是长方形 关节末端

画个圆角矩形 两个圆（枢轴）

可以用滑块控制旋转值

单关节 绕着枢轴旋转 .rotation 调整锚点

双关节 第一关节旋转 连接的第二关节rotation相同角度的同时 还要进行坐标轴旋转 改 x,y坐标

自动控制 平稳地前后摇摆 循环摆动 sin 正弦 rotation=angle x= angle要有周期性

手脚摆 动 两条 大小腿摆动角度不同

走动 重力 碰撞（对地面） 反作用力

屏幕环绕：从另一边穿出来

十四、反向运动学 ：拖拽（关节被外力拖动，可以移动） 和伸展（固定一端，关节不能移动，可以认为是固定在墙上的拖拽）

目标点

单关节伸展 ：向着目标点旋转 自由的端点跟随着要抓的目标点 算x,y坐标差，算出rotation =arctandy/dx 跟随鼠标

单关节拖拽： 旋转的同时 第二个枢轴跟随着鼠标 x,y鼠标 被鼠标拖着

拖两个关节 第二关节 的目标点 是第一关节的枢轴，跟随旋转和位移

多个关节拖拽：

多关节伸展 两关节向一个方向旋转 第二关节跟随枢轴 位移（坐标轴旋转）

固定一端 抓住鼠标 伸展，判断是否抓住

抓住物体 伸展加碰撞检测

反向运动学用三角定理 关节形成的三角，不是直角 余弦定理（） 由两边和角 求对边 两个关节是定长 对边求角度

十五、3D基础

Z 轴 右手左手坐标系

透视法：z深度增加时 将缩放变小 缩放公式

反弹

运动

十六、3D线条和填充

点

点线组成3d图形 旋转

填充

十七、背面剔除、3D灯光

十八、矩阵数学 旋转、缩放、平移 X/Y/Z三个方向

加法

乘法

缩放

旋转

十九、实用技巧

布朗运动 vx+=random（ 正或负之间） 加入摩擦力 每次\*0.8

随意乱飞

可以画线 画出布朗轨迹

随机分布（随机放置） 多个点随机放

方形分布中心点附近 x= random+centerx y 如果点很多类似于方形

圆形分布 最大半径 随机半径 随机的角度 x=cosangle \*r

偏向分布 越中心越多 x=random\*distance 把动态距离作为