

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

# 3D游戏开发步步高系列课程（1）

## 3D视频游戏开发介绍

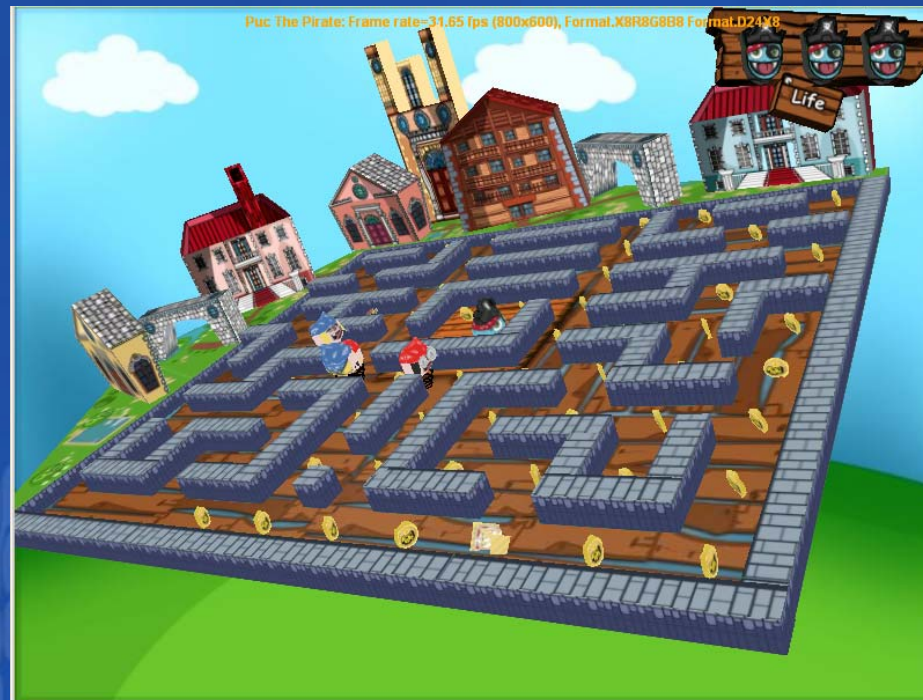
付仲恺  
微软特邀开发专家

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 主要目标

- 学习Direct3D开发知识
  - 创建3D游戏“Puc the Pirate”



您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 系列课程概要

- 3D游戏基本概念介绍
- 创建3D动态和静态物体
- 摄像机处理操作
- 游戏物体行为编程
- 背景处理和物体之间的碰撞

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 系列课程概要

- 人工智能（AI）的编程
- 显示文本和抬头显示（HUD）
- 菜单界面编程
- 添加灯光，音乐和音效



# 本次课程概要

- 介绍
- 探究必须的几种坐标系系统
- 对于渲染阶段的理解
  - 变换
  - 光栅化
- 为什么要使用Microsoft® Visual C#®

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 预习知识

- 基本的计算机知识
- 基本的编程开发知识
- C#开发知识

## Level 200

# 课程概要

- 介绍
- 探究几种常用的坐标系系统
- 对于渲染阶段的理解
  - 变换
  - 光栅化
- 为什么要使用Microsoft® Visual C#®

# 介绍

- 3D图像处理引擎完成3D图像处理过程
  - 程序控制
  - 几何变换
  - 特效
  - 等等
- 渲染是在3D场景中生成2D图像的处理过程





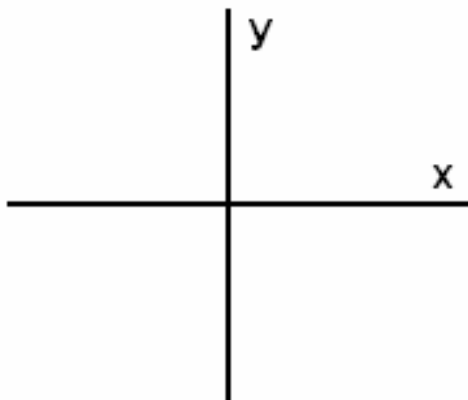
# 渲染处理

- 由两个主要部分组成
  - 几何变换 – 应用于 *顶点*
  - 三角形光栅化 – 应用于 *像素点*
- 固定功能管线
- 像素和顶点着色器
  - 替代特定特效的固定功能管线

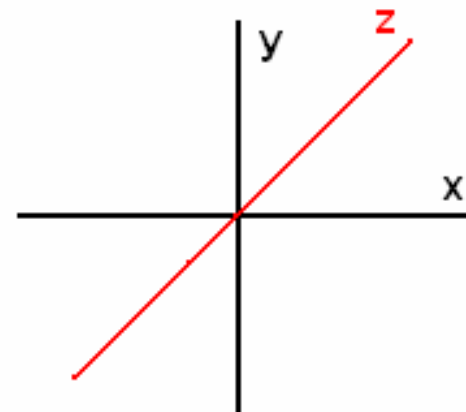
# 坐标系系统

- 用于描述物体位置和方向的空间
- 最简单的坐标系系统是笛卡尔平面

**2D Coodinate System**



**3D coordinate system**



# 重要的3D几何学知识

- 3D物体由多边形组成
  - 多边形由按照指定顺序描述的顶点集合组成
    - 三角形是最简单的多边形
    - 例如, 使用三角形来描述一个立方体, 那么每个面都需要使用2个三角形来描述, 并且由于立方体一共有6个面, 因此我们需要12个三角形来描述一个立方体
- 每个顶点包含:
  - x, y, z坐标值
  - 颜色
  - 用于计算灯光的法线
  - 纹理坐标, 通常是 (u, v)

# 更多的3D几何信息

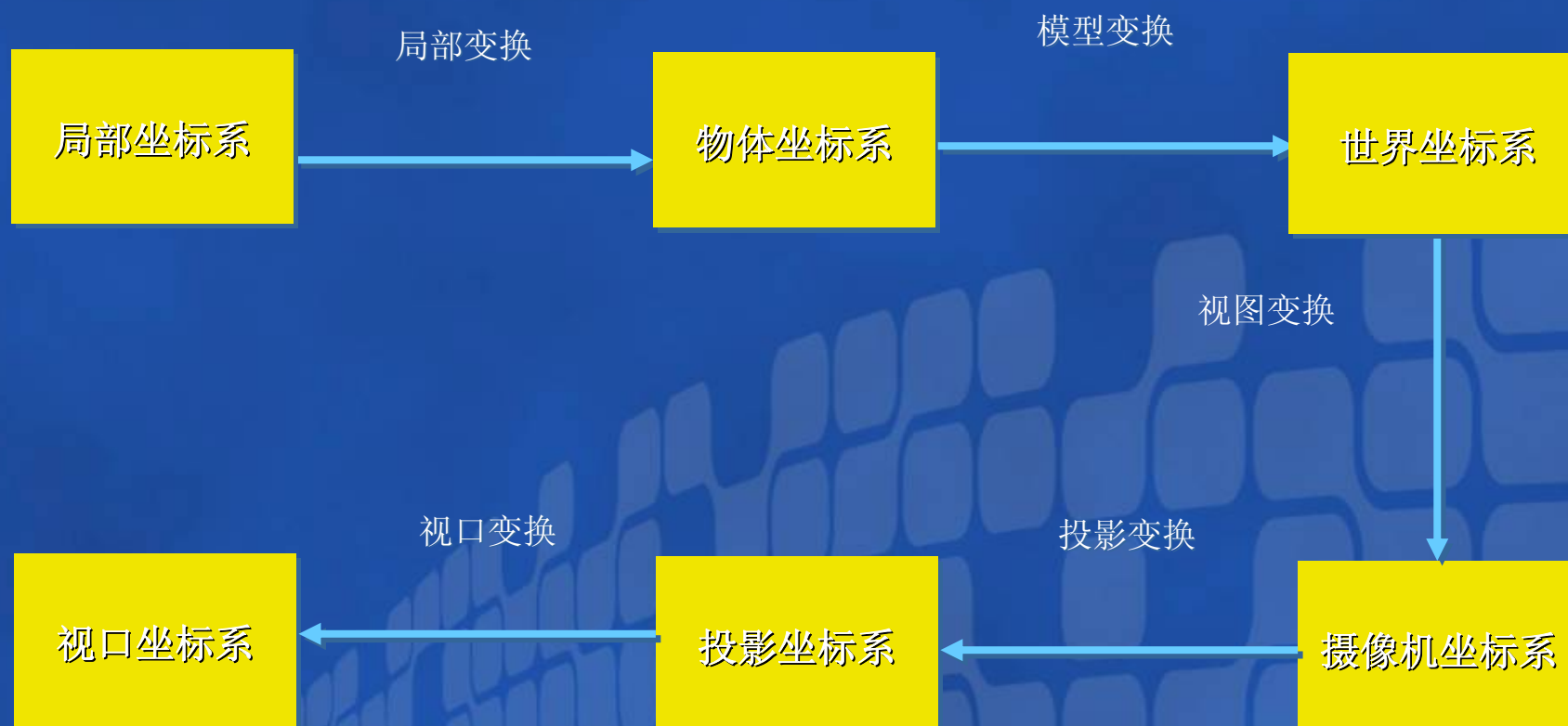
- 通过所有的变换，将顶点从物体局部坐标系变换到视口坐标系系统
- 平移，旋转和缩放等变换操作通常使用矩阵来执行
- 在投影之后，每个顶点都在投影平面上有一个新的 $x$ 和 $y$ 值用于描述它的位置，同时还有一个描述深度的 $z$ 值。
- 在管线处理的最后阶段，将纹理填入到各个三角形或者表面当中

# 课程概要

- 介绍
- *探究几种常用的坐标系系统*
- 对于渲染阶段的理解
  - 变换
  - 光栅化
- 为什么要使用Microsoft® Visual C#®

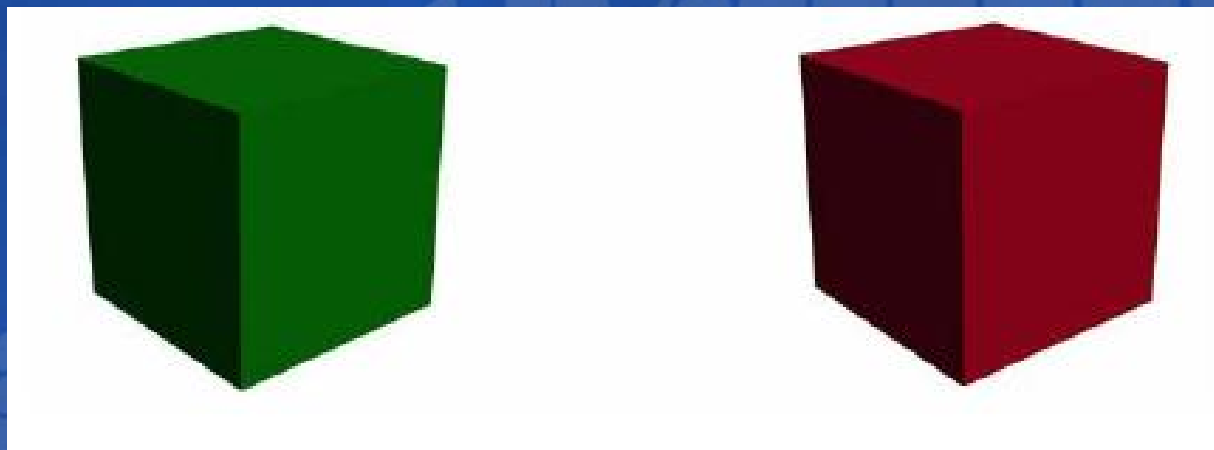


# 主要空间坐标系系统



# 物体坐标系系统

- 使用层次模型来表示由各个部分“装配”起来的物体
- 每个物体都有自己的坐标系
- 下面的图像显示了两个在它们自己的坐标系系统中所看到的正方体

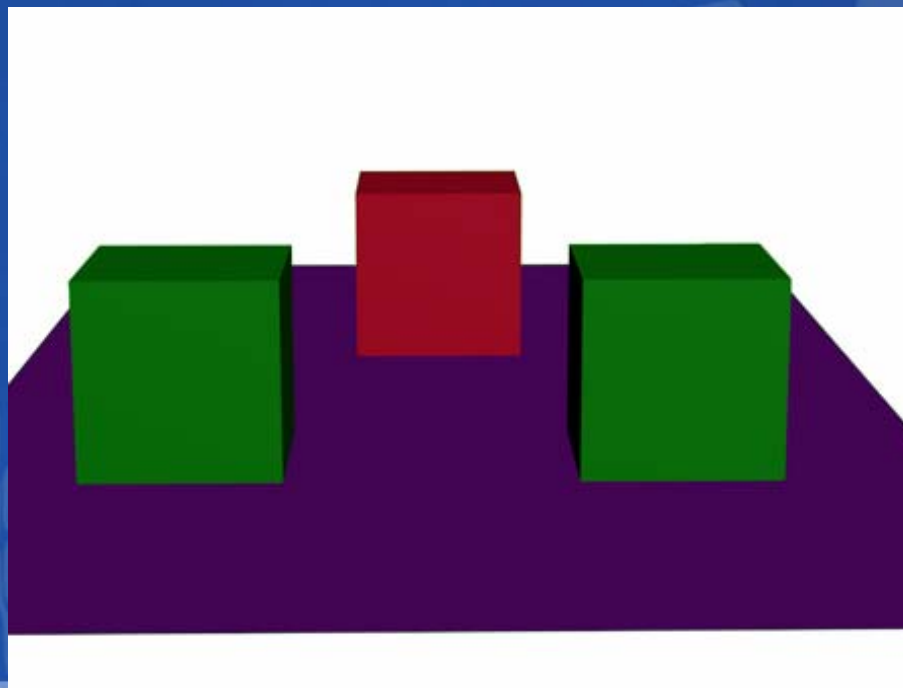


# 世界坐标系系统

- 也被称为“全局坐标系系统”
- 所有对象实例都能够缩放，平移和旋转的空间
- 所有的几何体都在同一坐标系中，使用同一坐标系原点

# 世界坐标系系统

- 下面的图像在世界坐标系系统中显示了两个绿色的立方体和一个红色的立方体的实例



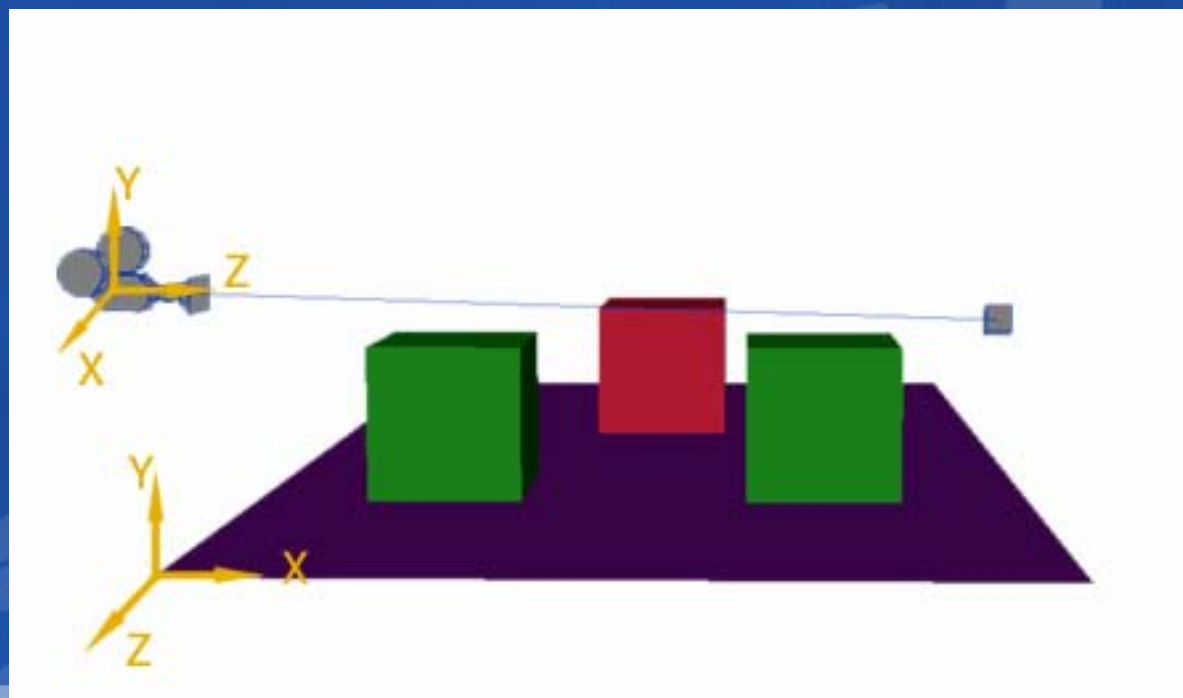
# 摄像机坐标系统

- 也被称为“参考坐标系系统”
- 指明在世界空间中的某一角度中任意放置的摄像机的位置，方向和方位的空间。
- 所有物体的位置都根据摄像机的中心和方向重新变换



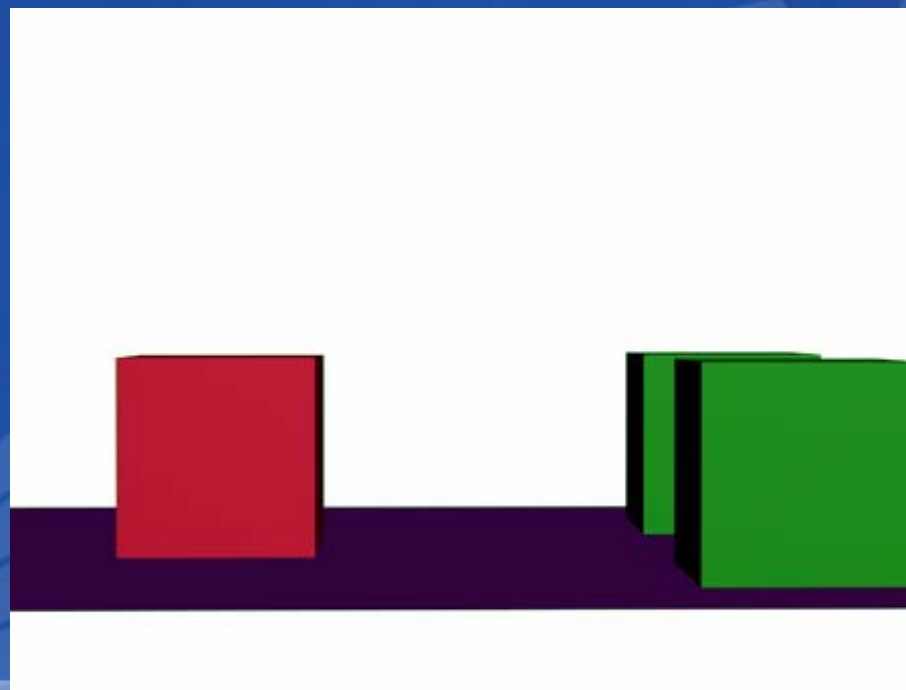
# 摄像机坐标系系统

- 下图显示了摄像机在世界坐标系系统中从左侧观看物体



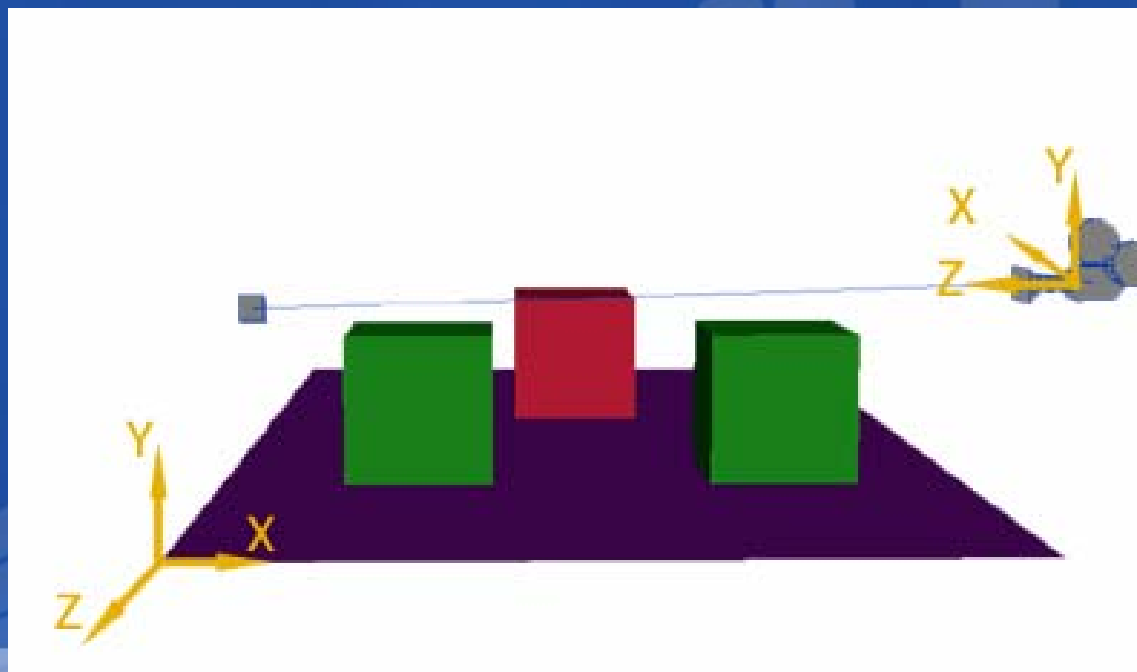
# 摄像机坐标系系统

- 下图显示了从左侧摄像机坐标系系统，或者摄像机点中所看到的相同的场景。



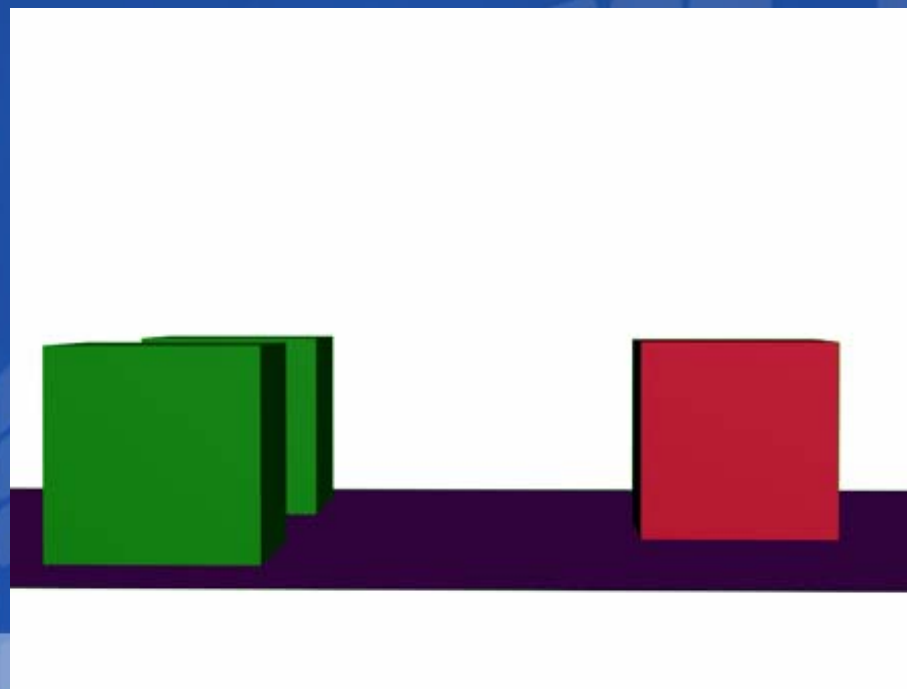
# 摄像机坐标系系统

- 下面的图像显示了在世界坐标系系统内摄像机从右侧观看世界的场景



# 摄像机坐标系系统

- 下面的图像显示了在右侧摄像机坐标系系统内或者摄像机点中所看到的相同的场景



# 投影坐标系系统

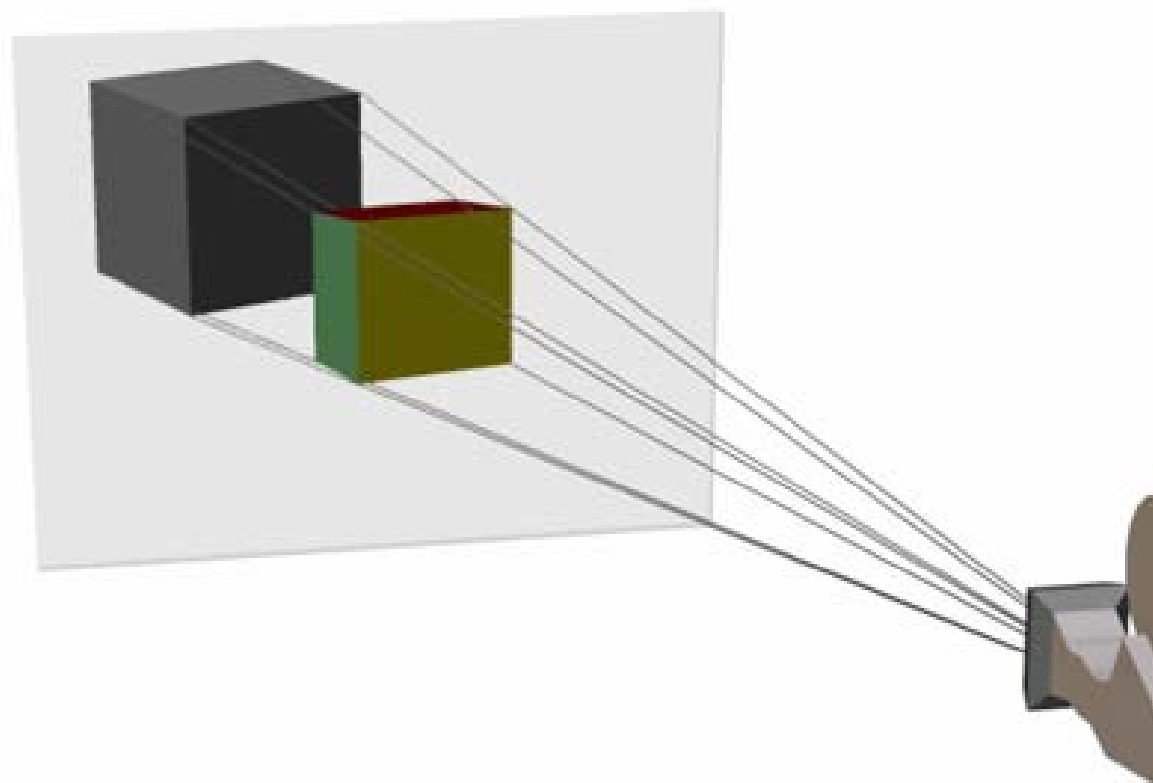
- 也被称为“裁剪坐标系系统”
- 由视图截锥和投影方法定义的空间
- 裁剪并且投影3D物体到2D视图平面



您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 投影坐标系系统



# 视口坐标系系统

- 在显示窗口中图像被实际绘制的矩形区域
- 由原点和窗口的延伸方向定义
- Z值通常被保留
- 有时候也被称作“2.5D”

# 课程概要

- 介绍
- 探究几种常用的坐标系系统
- *对于渲染阶段的理解*
  - 变换
  - 光栅化
- 为什么要使用Microsoft® Visual C#®

# 3D处理管线

- 为了实现 *时间平行性*
- 类似于装配线
- 通过将任务分割为一系列子任务来完成
- 子任务由特定的硬件来执行
- 各个处理阶段 *并发操作*
- 连续的任务在子任务层面上重叠执行

# 3D图像处理管线

- 应用程序处理阶段通过软件在CPU中实现
- 几何变换和光栅化在GPU（图像处理单元）中实现



3D图像处理管线概念图



# 应用程序处理阶段

- 软件实现
- 游戏引擎：碰撞检测和响应，动画，AI，用户输入的读取和解析
- 准备GP所使用的图元，属性和相关函数
- 加速和优化非常重要

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 应用程序处理阶段组件

- 游戏逻辑
- 人工智能
- 动画物体
- 摄像机控制
- 剔除算法
- 碰撞检测和响应
- 游戏物理和动力学特性
- 几何运算库

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 游戏逻辑

- 控制游戏流和层级流
- 用户界面控制
- 输入/输出 (I/O) 处理
- 物体的导入和加载

您的潜力, 我们的动力

**Microsoft**  
微软(中国)有限公司

# 人工智能

- 不可玩角色反应
- 不可玩角色行为
- 不可玩角色路径查找
- 脚本事件

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 动画物体

- 刚性物体
- 可变形物体
- 关节物体



# 摄像机控制

- 第一人称视角
- 第三人称视角
- 跟踪正在移动物体的路径
- 在不同的摄像机角度和方向之间内插数值（Quaternions）以提供合理的路径
- 重播

您的潜力. 我们的动力

**Microsoft**  
微软(中国)有限公司

# 几何运算库

- 向量和矩阵操作
- 距离和角度测量
- 交集和包含运算
- 搜索和排序算法

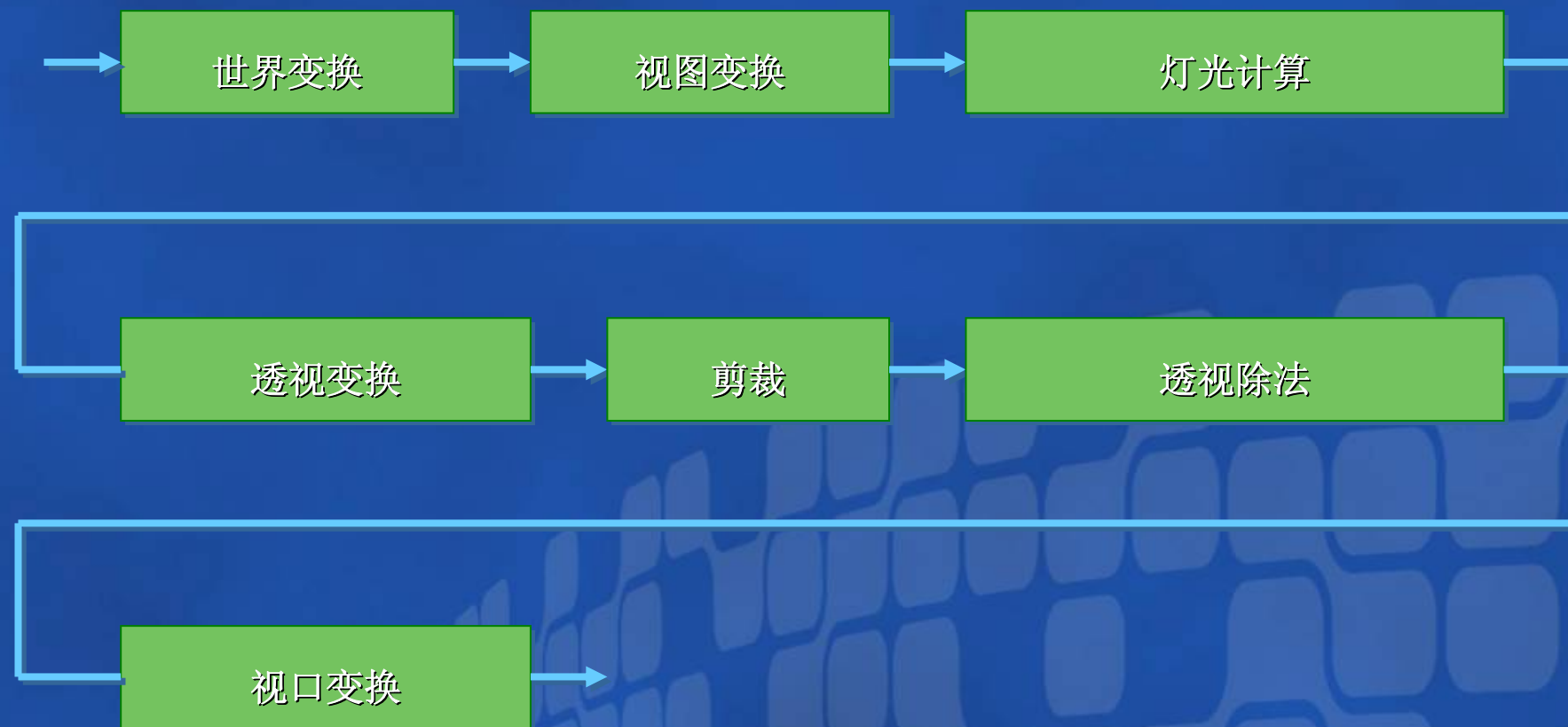
# 剔除算法

- 背面剔除
  - 如果一个绘制元素的法线方向背离视点观察方向，则该绘制元素必不可见
- 摄像机平截体内部的对象之间的遮挡测试
  - 遮挡剔除：如果一个绘制元素被其他不透明绘制元素（组）所遮挡，则该绘制元素必不可见
- 空间分割
  - 八叉树（Octree）
  - 二元空间分割树（BSP）
- 潜在可见集合（PVS）
- 层次细节（LOD）

# 游戏的物理和动力学特性

- 正向和反向运动
- 动力学逆过程（已知力求运动）
- 刚体和柔体物体对象
- 碰撞检测和响应

# 几何处理阶段



变换处理阶段管线



# 几何变换

- 将3D的几何体变换到2D的屏幕上
- 在这个阶段不会产生任何绘制操作 – 完全的数学计算
- 由许多的坐标系系统和在这些系统之间相互转换的操作方法组成
  - 矩阵运算
- 如果程序员对这些变换不熟悉, 极易混淆

# 世界变换

- 将所有的物体转换为统一的全局坐标
- 由不同的模型组成并且通过 *世界变换* 创建
  - 缩放
  - 旋转
  - 平移

# 视图变换

- 转为从摄像机的角度看到的世界坐标系
- 设置摄像机位置和方向
  - 设置注视点和参考点
- 设置灯光位置, 方向和属性
- 以摄影机为原点, 从原点到注视点为Z轴, 再加上参考点, 确定Y-Z平面, 构成视图坐标系系统 (VCS)
- 创建无限的棱锥面
- 将场景从WCS变换为VCS

# 灯光和阴影

- 对每个顶点计算RGBA颜色数值
- 灯光模型方程应用于一组输入参数
- 使用顶点和像素着色器可以实现灵活的灯光处理

# 灯光

- 输入参数
  - 顶点位置坐标
  - 顶点法线坐标系
  - 灯光源位置坐标
  - 灯光源属性（点光，聚光灯，方向光）
  - 表面材质属性（漫反射和环境光反射，散射，镜面反射）
- 操作
  - 物体颜色由光源和表面属性决定
  - 为每个顶点计算法线向量
  - 在每个顶点上应用照明模型
  - 每个顶点都保存漫反射和环境光反射，散射和镜面反射颜色



# 透视转换

- 介绍前后裁剪面
- 将无限的视图棱锥体截为有限的视图截锥
- 将VCS场景转换为齐次剪裁坐标系系统（CCS, 视口坐标系系统的另一个名字 – 2.5D）
- 将有限的视窗体扭曲为长方体（实现近大远小的效果）
- 在CCS中裁减算法非常简单

# 剪裁

- 删除长方体外部的场景
- 实现Sutherland-Hodgeman多边形裁剪算法
  - 每次用裁剪窗口的一条边裁剪多边形。
  - 裁剪窗口的一条边以及延长线构成的裁剪线，把平面分成两个部分：
    - 一部分包含窗口，称为可见一侧
    - 另一部分称为不可见一侧

# 透视除法

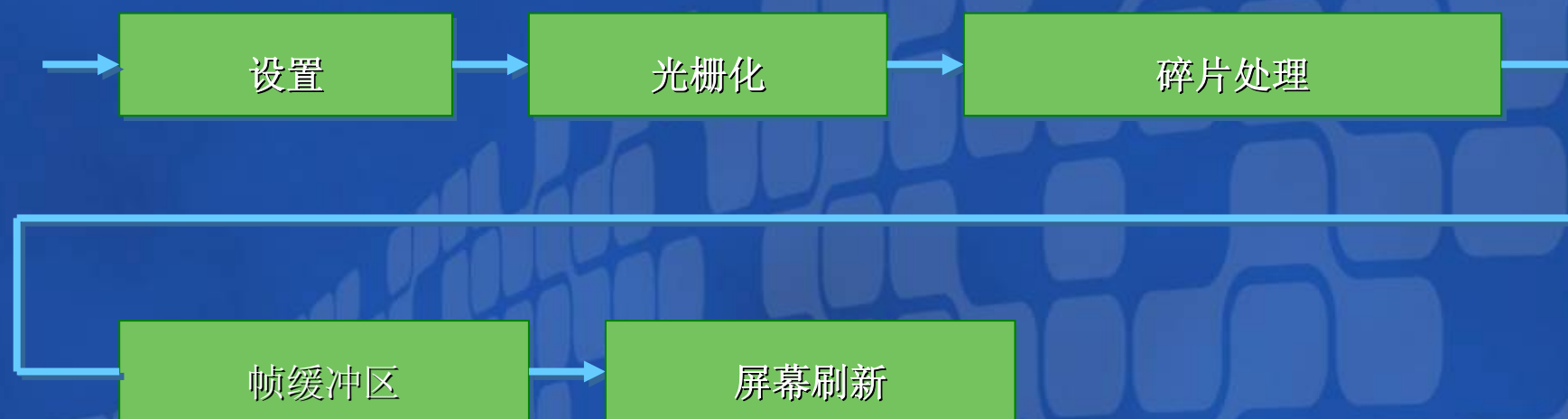
- 投影3D物体到2D视图平面
- 将可视长方体压扁到屏幕坐标系系统上
- 通过齐次坐标分割齐次CCS坐标
  - 光栅希望以 $x/w$ 、 $y/w$ 和 $z/w$ 坐标的形式得到顶点，同时还要  
求RHW，即齐次 $w$ 的倒数
- SCS是设备无关的
- SCS中不包含2D视图之外的像素

# 屏幕映射变换

- 视口是包含SCS映射内容的显示设备区域。它定义了被显示的2D图像的位置
- 视口变换重新解释SCS坐标系为相应的硬件像素坐标

# 光栅化处理阶段

- 将几何图元转换为二维图像的过程
  - 图象的每个点都包含诸如颜色、深度、纹理数据等信息
  - 像素点和它的相关信息合起来称作碎片



光栅化处理阶段管线



# 设置

- 如果背面（**back-face**）剔除被打开，只光栅化前面（**front-facing**）的三角形
  - 从三角形的前面看三角形，顶点是按照顺时针方向排列，背面剔除可以剔除那些背离屏幕（顶点逆时针方向排列）的三角形
- 为三角形光栅化计算参数
  - 边斜度
  - 颜色渐变层
  - 贴图坐标梯度

# 光栅化

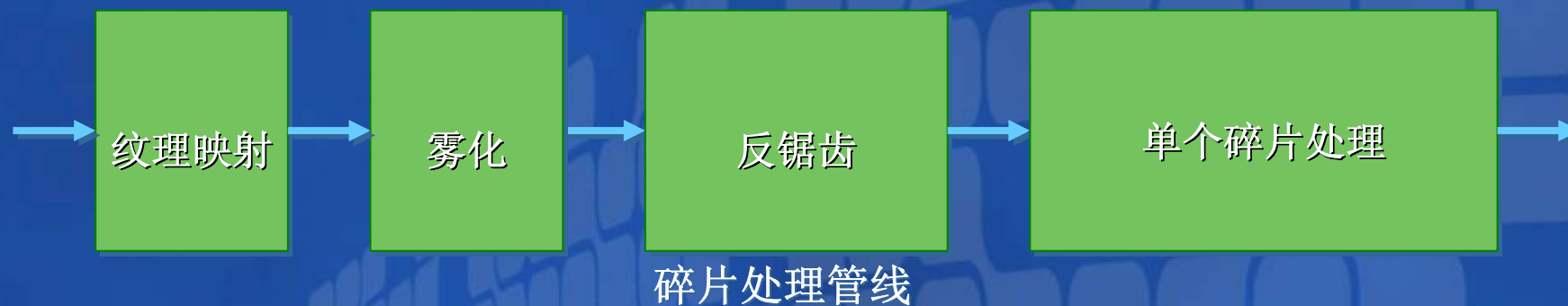
- 计算三角形所覆盖的像素的坐标
- 计算每个内部像素的数据
  - 颜色
  - 深度
  - 纹理坐标
  - RGBA颜色
- 处理凸凹形多边形
- 转换多边形为线段和像素点

# 碎片处理

- 处理内像素的颜色和深度值
  - 材质查找
  - 雾化
  - 反锯齿
  - 深度测试
  - 混合
- 保存内部像素的颜色和深度值
  - 决定像素是否可见
  - Z值表示了像素点到视口的距离
  - 将最后计算出来的颜色写入到帧缓冲区
  - 将深度值写入深度缓冲区

# 碎片处理

- 内部像素传递路径:



# 纹理映射

- 图像数据从系统内存传递到在线纹理存储器中
- 使用内插值纹理坐标来查找纹理内存并且获取颜色数据
- 将纹理颜色数据输入到颜色融合函数中
  - Replace
  - Modulate
  - Add
  - Blend



# 纹理映射

- 多纹理在管线纹理处理阶段处理
- 每个独立单元被称为纹理环境发生器 (TEV)
- 第一个TEV单元结合纹理颜色, 顶点颜色 (或者第二个纹理颜色) 以及约束条件
- 处理结果被传递到下一个TEV单元
- 第二个和后继的TEV单元将另一个纹理或者内插值与前一个结果相结合

# 雾化

- 景深效果处理: 当效果被打开时, 物体随着距离的延长而淡出
- $Color_{final} = (f)Color_{fragment} + (1-f)Color_{fog}$ 
  - 雾化混合因子被定义为  $f \in [0, 1]$
- 雾化效果随着观察者距离的增加而增强
- 不同的雾化混合因子: 线形, 指数, ...
- 对每个顶点以及内插点进行计算, 或者对每个碎片进行计算

# 图像反锯齿

- 当高频信息通过低频像素采样时，会发生锯齿
  - 连续的数学表达式与离散的像素之间的误差
  - 低分辨率显示设备
- 反锯齿技术尝试减轻类似锯齿
- 根据碎片所覆盖的像素的总量调整像素颜色
- 在MIP映射（多纹理映像纹理）中，原始纹理被过滤为许多子纹理，形成纹理序列
  - 每一级纹理都比上一级纹理的高和宽图像小二次幂

# 单个碎片操作

- 光栅化产生的碎片只有通过一系列的测试后，才允许它修改相应的帧缓冲像素。
  - 通过测试，碎片数据可以直接替代帧缓冲中现有的数据
  - 否则，根据某些模式的状态，碎片数据可能与帧缓冲已有数据结合



# 单个碎片操作

- 剪切测试 (Scissor Test)
  - 判断碎片是否在视口中
  - 如果测试失败, 碎片将被抛弃
- Alpha Test
  - 将碎片的alpha值与固定参考值进行比较
  - 如果测试失败, 碎片将被抛弃
- 蒙板测试 (Stencil Test)
  - 蒙板缓冲是一组不可显示的位平面 (1-8)
  - 视口中的每个象素都有对应的蒙板值
  - 将碎片的蒙板值与蒙板参考值进行比较



# 单个碎片操作

- 深度测试
  - 判断新进入的碎片是否比以前已经渲染过的碎片更近
  - 将新进入的碎片深度值与深度缓冲器中的当前值进行比较
  - 如果测试失败，碎片将被抛弃
  - 否则，如果DB需要更新，则更新为碎片的深度值

# 单个碎片操作

- 混合
  - 透明, 合成, 绘画, ...
  - 将新进入的碎片的颜色与帧缓冲区中碎片所在位置的颜色相结合
  - 将计算出来的颜色部分截取在1.0以内
  - 将颜色写回到帧缓冲区
  - $ColorFinal = \alpha ColorFragment + (1 - \alpha) ColorFB$
- 逻辑运算
  - 逻辑运算应用于新进入的碎片的颜色和帧缓冲区中碎片所在位置的颜色

# 课程概要

- 介绍
- 探究几种常用的坐标系系统
- 对于渲染阶段的理解
  - 变换
  - 光栅化
- 为什么要使用**Microsoft<sup>®</sup> Visual C#<sup>®</sup>**

# 为什么使用C#

- C#充分利用托管资源
  - 自动垃圾回收
  - 没有内存泄露 – 保证
- Managed DirectX提供了简单的框架
  - 减少了繁重的初始化操作
- 代码运行于Microsoft .NET 虚拟机之上
  - 降低了代码复杂度
  - 便于Web接口的开发



# 什么是Puc the Pirate?

## Controls

- Right-click** to change the camera's position
- Mouse wheel** to zoom in and out
- Arrow keys** to move Puc
- Space** to jump
- M** to toggle music on/off
- P** to pause the game
- Esc** to exit the game

6

7

All Content ©Copyright 1996 - 2005 - DigPen Institute of Technology. ALL RIGHTS RESERVED.

## Game Play

Puc can jump over the would-be pirates to avoid colliding with them and losing a life. He starts with two lives and can gain another by catching a drumstick of chicken along the course. Puc must avoid bombs that the would-be pirates plant as they navigate around the maze, otherwise he might lose a life and end up in prison. There is no way to escape bombs except by jumping over them.

After Puc has collected all the coins, a trap door opens somewhere in the maze for Puc to jump through to complete the current level. If Puc loses all his lives before collecting all the coins, he will get trapped in prison and the game will end.

12

13

All Content ©Copyright 1996 - 2005 - DigPen Institute of Technology. ALL RIGHTS RESERVED.



# 课程总结


- 介绍
- 探究必须的几种坐标系系统
- 对于渲染阶段的理解
  - 变换
  - 光栅化
- 为什么要使用Microsoft® Visual C#®

# 获取更多MSDN资源


- **MSDN中文网站**  
<http://www.microsoft.com/china/msdn>
- **MSDN中文网络广播**  
<http://www.msdnwebcast.com.cn>
- **MSDN Flash**  
<http://www.microsoft.com/china/newsletter/case/msdn.aspx>
- **MSDN开发中心**  
<http://www.microsoft.com/china/msdn/DeveloperCenter/default.mspx>

# Question & Answer

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。

 **问题和解答 (无问题)** ▲ ×

在此会议中尚未解答任何问题。

要向演示者提问，请在此处键入问 

提问(A)

删除(D)

问题管理器(Q)

您的潜力，我们的动力

**Microsoft®**  
微软(中国)有限公司

**Microsoft®**

msdn  


**MSDN Webcasts**