

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

介绍游戏中的“精灵”与“动画”

吴淦

微软开发合作部

内容介绍

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

- 游戏开发系列课程
- 本次课程需具备的条件
- 课程介绍
- 游戏中元素介绍
- Demo
- 提供学习的相关信息
- Q & A

您的潜力. 我们的动力

Microsoft®
微软(中国)有限公司

游戏开发系列课程

- 向广大的游戏爱好者提供更多的游戏开发的基础知识。
- 介绍如何使用 Microsoft® Visual C#® 2005 Express Edition 进行程序设计，这是微软为初级开发者提供的一种轻量级的集成开发环境（IDE）。



本次课程需具备的条件

- 不需要任何编程经验
- 在网站上下载相应的演示程序
- 安装Visual C# 2005 Express
 - <https://www.microsoft.com/china/msdn/express/csharp.aspx>
 - [DirectX 9.0 SDK Update - \(April 2005\)](#)

内容介绍

- 介绍游戏中的“精灵”和“动画”
 - 背景
 - 视点
 - 动画
 - 帧
 - 精灵
 - 帧延迟
 - 透明
 - 游戏实现
 - 背景
 - 精灵

“世界”维度

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 在“世界”是什么样子的？
 - 整个游戏发生的环境
 - 是被定义了长度和宽度的矩形

视点

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

●什么是视点

- 在“世界”的坐标系中要被显示的窗体
- 窗体定义显示区域
- 视角定义什么区域被显示
- 视角是显示窗口中的一个矩形区域
- 默认情况下，它是整个应用程序窗口
- 视角以像素为单位变化

位图

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

●什么是位图

- 一张位图是由数千个像素组成
- 每个像素都有自己的位置和质量
- 像素质量包括颜色值、亮度等

●RGB 亮度

- 显示器显示的颜色是由红、绿、蓝三种颜色组成
- 每个像素有一个由红、绿、蓝组成的值
- 基础颜色的亮度可以创建一个颜色
- 换句话说，颜色其它是由红、绿、蓝三种颜色变化亮度组成的

背景

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 什么是背景

- 背景是一张图片或位图，它将在“世界”中代替一个固定位置
- 游戏对象可以显示在背景的上层

- 位置

- 背景会在“世界”中被指定显示位置

“动画”

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 什么是“动画”

- 一系列图片，按照顺序播放，并在播放每张图片间有特定的延迟
- 每张图片（帧）有细微的不通
- 当我们连续播出的时候，就给人以动画的效果

- 当“动画”播完我们将如何做？

- 再次从第一帧开始播放
- 设置从特定的帧开始播放
- 保持最后一帧

帧

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

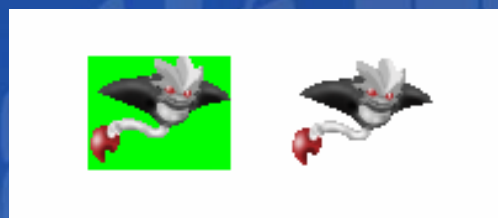
●什么是“帧”

- 帧就是一张简单的图片
- 图片是动画的一部分
- 动画是由每一帧组成的
- 当每一帧被顺序播放时，动画的感觉就产生了



帧和透明

- 一帧是由一张图片（位图）组成
- 图片是一个矩形或方形的点的矩阵,不是所有的图片都是矩形外形
- 在矩形中不被使用的区域，将被填充为透明
- 在播放中将忽略已经被设定为透明的像素
- 换句话说，透明的像素将不被重绘



精灵

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 精灵是显示在背景上层的画
- 也被称为游戏对象
- 通常，精灵表示游戏中所有运动的部分
- 精灵可以包括：
 - 敌人
 - 背景活动的部分
 - 子弹

精灵

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 精灵的状态是根据图片、位置、色值等变化的
- 因为如此，精灵需要存各种信息
 - 即时位置
 - 即时速度
 - 即时方向
 - 可见度
 - 帧
 - 动画

精灵活动性

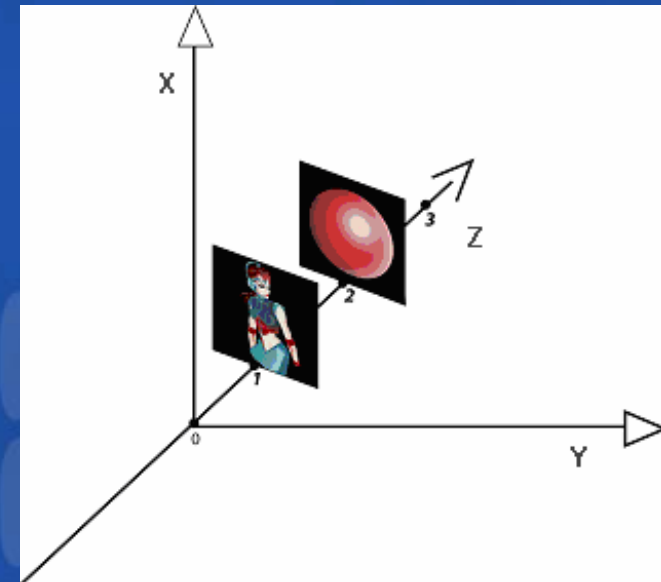
- 精灵可以设定为活动或不活动
- 当精灵不活动时，所有的行为是失效和不可见的
- 不活动的精灵在游戏循环中不能执行操作
- 不活动的精灵在需要精灵原型时候经常被使用
- 精灵原型在运行时用于创造同样的精灵

精灵可视性

- 在每个游戏循环中，代码将处理所有的游戏对象
- 处理游戏对象包括运用和绘制
- 当精灵可见时，它将被运用和绘制
- 当精灵不可见时，它也可以被运用，但不被绘制
- 伴随着精灵创建过程，有一帧或多帧的动画将被显示出来
- 当精灵需要可见时，游戏对象需要一个视觉上真实的动画存在

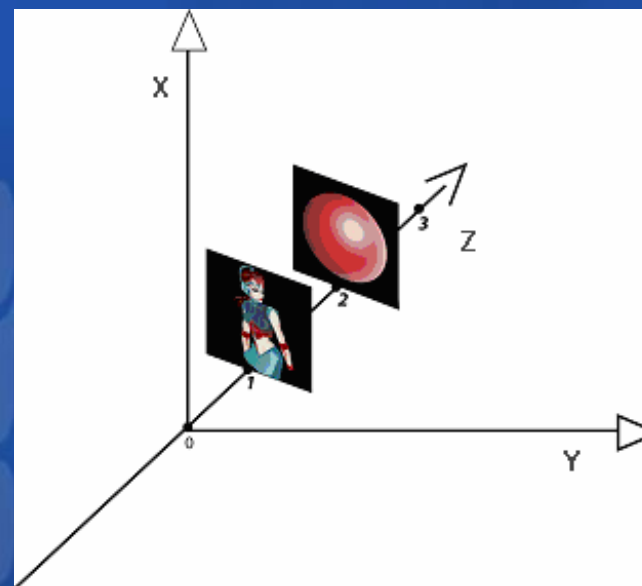
ZOrder

- 为了让大家理解ZOrder, 我们必须先要模拟一个3D坐标系
- Z轴表示深度
- ZOrder值越大说明屏幕上的对象越深
- 图片1的ZOrder 值比图片2的ZOrder值低, 所以球形图片应显示在人物图片下方



ZOrder

- 因为是 2 D 的应用程序，所以ZOrder只能影响图片显示的顺序不能影响图片的尺寸
- 所有的游戏对象都会受到ZOrder轴的影响



您的潜力，我们的动力

Microsoft
微软(中国)有限公司

DEMO

项目运行

添加背景
添加精灵
添加动画



MSDN Webcasts
MSDN Webcasts

背景精灵

- 背景是指定的图片，并且需要两个相同的背景图片（精灵）
- 这是为了容易滚动
- 因为背景是一张图片，所以必须被作为精灵添加到游戏中



添加背景

- 在游戏中直观的表现精灵就是动画
- 一个动画由一帧或多帧组成
- 每一帧是由一张图片和一个延迟组成
- 一张图片是一个位图文件
- 所有的创建精灵的代码都将在
`StarTrooperGame.cs`文件中的`StarTrooper`类中
执行

添加背景

每张图片都是图片类对象用于存储位图

```
Picture background = new Picture("Background.bmp", Color.FromArgb(0, 255, 0));
```

第一个参数是位图名字，第二个参数是透明度

图片对象被添加到游戏中是为了作为帧使用的

```
Game.Add(background);
```

添加背景

帧是使用图片的帧类的对象

```
Frame backGroundFrame = new Frame(background, 0);
```

第一个参数是图片对象，第二个参数是图片延迟

动画对象是由若干帧组成

```
Animation backGroundAnimation = new Animation();
```


添加背景

每一帧添加到动画对象中

```
backGroundAnimation.Add(backGroundFrame);
```

创建一个BackGround对象bg,Background类是由Spirte类派生的

```
Background bg = new Background();
```

把已经创建的动画对象添加到bg对象中

```
bg.Add(backGroundAnimation);
```

添加背景

设置精灵位置

```
bg.Position = new Point(320, 240);
```

涉及bg对象的宽和高

```
bg.ScaleX = 640.0f / background.Width;  
bg.ScaleY = 480.0f / background.Height;
```

把bg对象的ZOrder属性设置为10

```
bg.ZOrder = 10;
```

把精灵添加到游戏中

```
Game.Add(bg);
```

添加背景

重复添加其它的精灵对象

```
Background bg2 = (Background)bg.Clone();  
bg2.Position = new Point(320, -240);  
bg2.ScaleX = 640.0f / background.Width;  
bg2.ScaleY = 480.0f / background.Height;  
bg2.ZOrder = 10;  
Game.Add(bg2);
```

这时我们已经创建我们自己的精灵对象，并添加到了游戏中

添加背景

全部代码如下

```
Picture background = new Picture("Background.bmp", Color.FromArgb(0, 255, 0));
Game.Add(background);

Frame backGroundFrame = new Frame(background, 0);
Animation backGroundAnimation = new Animation();
backGroundAnimation.Add(backGroundFrame);
Background bg = new Background();
bg.Add(backGroundAnimation);
bg.Position = new Point(320, 240);
bg.ScaleX = 640.0f / background.Width;
bg.ScaleY = 480.0f / background.Height;
bg.ZOrder = 10;
Game.Add(bg);

Background bg2 = (Background)bg.Clone();
bg2.Position = new Point(320, -240);
bg2.ScaleX = 640.0f / background.Width;
bg2.ScaleY = 480.0f / background.Height;
bg2.ZOrder = 10;
Game.Add(bg2);
```

添加骑兵

骑兵是动画，所以需要几张图片

每张图片被添加到游戏中

每张图片都将被作为一帧被创建

```
Picture trooper01 = new Picture("trooper01.bmp", Color.FromArgb(0, 255, 0));  
Game.Add(trooper01);  
Picture trooper02 = new Picture("trooper02.bmp", Color.FromArgb(0, 255, 0));  
Game.Add(trooper02);  
Picture trooper03 = new Picture("trooper03.bmp", Color.FromArgb(0, 255, 0));  
Game.Add(trooper03);  
Picture trooper04 = new Picture("trooper04.bmp", Color.FromArgb(0, 255, 0));  
Game.Add(trooper04);  
Picture trooper05 = new Picture("trooper05.bmp", Color.FromArgb(0, 255, 0));  
Game.Add(trooper05);  
Picture trooper06 = new Picture("trooper06.bmp", Color.FromArgb(0, 255, 0));  
Game.Add(trooper06);
```


添加骑兵

每张图片被用于创造一帧

```
Frame afTrooper01 = new Frame(trooper01, 5);  
Frame afTrooper02 = new Frame(trooper02, 5);  
Frame afTrooper03 = new Frame(trooper03, 5);  
Frame afTrooper04 = new Frame(trooper04, 5);  
Frame afTrooper05 = new Frame(trooper05, 5);  
Frame afTrooper06 = new Frame(trooper06, 5);
```

每帧都被添加到游戏中

```
Animation trooperAnimation = new Animation();  
trooperAnimation.Add(afTrooper01);  
trooperAnimation.Add(afTrooper02);  
trooperAnimation.Add(afTrooper03);  
trooperAnimation.Add(afTrooper04);  
trooperAnimation.Add(afTrooper05);  
trooperAnimation.Add(afTrooper06);
```

添加骑兵

使动画可以循环播放

创建一个骑兵

添加这个动画

设置骑兵的位置

把骑兵添加到游戏，并且设置它的原型

```
trooperAnimation.Play();  
trooperAnimation.Loop = true;  
Trooper trooper = new Trooper();  
trooper.Add(trooperAnimation);  
trooper.Position = new Point(320, 450);  
Game.Add(trooper);  
Trooper = trooper;
```

添加秃鹰

秃鹰也是动画，它也是多图片的
象创建骑士一样，我们把每张图片添加到游戏中，并且为
动画创建一帧

```
Picture condor01 = new Picture("condor01.bmp", Color.FromArgb(0, 255, 0));  
Game.Add(condor01);  
Picture condor02 = new Picture("condor02.bmp", Color.FromArgb(0, 255, 0));  
Game.Add(condor02);  
Picture condor03 = new Picture("condor03.bmp", Color.FromArgb(0, 255, 0));  
Game.Add(condor03);  
Picture condor04 = new Picture("condor04.bmp", Color.FromArgb(0, 255, 0));  
Game.Add(condor04);
```

添加秃鹰

使用每张图片创建一帧

```
Frame afcondor01 = new Frame(condor01, 5);  
Frame afcondor02 = new Frame(condor02, 5);  
Frame afcondor03 = new Frame(condor03, 5);  
Frame afcondor04 = new Frame(condor04, 5);
```

把每一帧添加到动画中

```
Animation condorAnimation = new Animation();  
condorAnimation.Add(afcondor01);  
condorAnimation.Add(afcondor02);  
condorAnimation.Add(afcondor03);  
condorAnimation.Add(afcondor04);
```

添加秃鹰

当游戏显示时，秃鹰是第二个动画，第二个动画与第一十分相似

```
Picture condorExplosion01 = new Picture("condorExplosion01.bmp",  
                                         Color.FromArgb(0, 255,  
0));  
Game.Add(condorExplosion01);  
  
Picture condorExplosion02 = new Picture("condorExplosion02.bmp",  
                                         Color.FromArgb(0, 255,  
0));  
Game.Add(condorExplosion02);  
  
Picture condorExplosion03 = new Picture("condorExplosion03.bmp",  
                                         Color.FromArgb(0, 255,  
0));  
Game.Add(condorExplosion03);
```


添加秃鹰

使用每张图片创建一帧

```
Frame afcondorExplosion01 = new Frame(condorExplosion01, 4);  
Frame afcondorExplosion02 = new Frame(condorExplosion02, 3);  
Frame afcondorExplosion03 = new Frame(condorExplosion03, 4);
```

把每一帧添加到动画中

```
Animation condorExplosion = new Animation();  
condorExplosion.Add(afcondorExplosion01);  
condorExplosion.Add(afcondorExplosion02);  
condorExplosion.Add(afcondorExplosion03);  
condorExplosion.Play();
```

添加秃鹰

- 使动画不能循环播放
- 创建一个秃鹰
- 把两个动画添加到秃鹰对象
- 为秃鹰设定原型

```
condorExplosion.Play();  
Condor condor = new Condor();  
condor.Add(condorAnimation);  
condor.Add(condorExplosion);  
Condor = condor;
```

创建其它精灵或者新游戏

备注：在游戏中，默认的精灵类是为了从中创建“骑兵”、“秃鹰”、“火”这些对象而创建的。然而，为了添加一个新精灵类，例如：`car` 类，我们可以添加一个文件叫 `Car.cs`，并且添加一个新类叫 `Car`，`Car` 类可以从精灵类派生

备注：一旦明白了游戏的流程，我们可以自己创建一个游戏。使用游戏引擎，我们可以创造一个游戏情节完全不同的游戏。例如我们写一个 `MyGame.cs` 的文件，创建一个 `MyGame` 的游戏。

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.msp>

下节课简介

- 下节课简介:


讲述让你的精灵动起来所相关的知识。运动所涉及的基本概念, 如位置、速度、方向、路程等。以及运动时产生的碰撞在不同坐标下的判定及其公式。

- 时间


11月22日

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