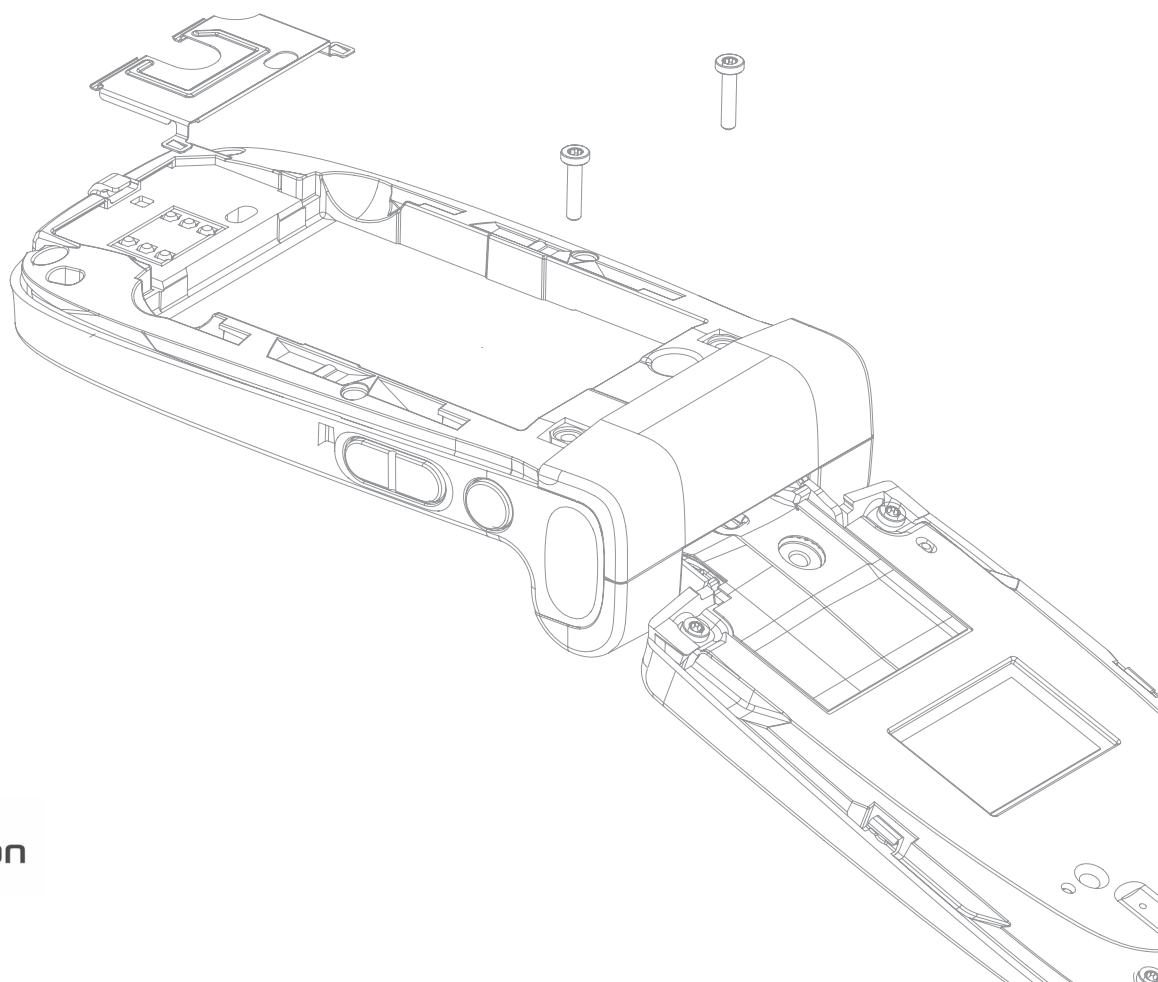


# 开发人员指南

2006 年 9 月

## 使用 Java™ Platform Micro Edition 的 3D 图形

适合索尼爱立信功能和入门级手机



Sony Ericsson

# 前言

## 本文档的用途

---

本文档介绍了在索尼爱立信 K700、S700、K500、Z500、V800、K300、J300、K750、W800、Z800、K600、V600、W600、W550、Z520、W900、W810、K610、K800、K790、Z530、W300、K510、K310、K320、W700、Z525、Z550、Z558、W850、Z710、W710、Z610 和 W830 系列手机中提供的对 Java™ ME 平台的 3D 支持。本文档还介绍了如何基于 Mascot Capsule® Micro3D Version 3 和适合 J2ME 的 Mobile 3D Graphics API (JSR 184) 来开发 Java 3D 应用程序。

本文档可使以下人员受益：

- 软件开发人员
- 企业买主
- IT 专业人员
- 技术支持工程师
- 业务决策人员

本文档假定读者熟悉 Java 语言。

这些开发人员指南的出版商为：

Sony Ericsson Mobile Communications AB,  
SE-221 88 Lund, Sweden

电话：+46 46 19 40 00  
传真：+46 46 19 41 00  
[www.sonyericsson.com/](http://www.sonyericsson.com/)

© Sony Ericsson Mobile Communications AB,  
2006。保留所有权利。在此授予您下载和 / 或打  
印本文档的许可。  
任何未在此明确授予的权利都予以保留。

第十四版（2006 年 9 月）  
出版号：EN/LZT 198 7387 R14A

本文档由 Sony Ericsson Mobile Communications AB  
（索尼爱立信）出版，不提供任何担保\*。Sony  
Ericsson Mobile Communications AB（索尼爱立  
信）随时可能对本文档中的印刷错误、当前的不准  
确信息进行必要的改进和更改，或对程序和 / 或设  
备进行改进，恕不另行通知。但是，这些修改将编  
入本文档的新版本中。印刷版本仅被视为临时的参  
考副本。

\* 所有暗示的担保，包括但不限于对适销性或针对  
特定用途的适用性的暗示担保，都不包含在内。对  
任何性质的意外或间接损坏，包括但不限于由于使  
用本文档中的信息而引起的利润或收入损失，索尼  
爱立信或其许可方概不负责。

# 索尼爱立信开发者世界

在 [www.sonyericsson.com/developer](http://www.sonyericsson.com/developer) 上，开发人员可以找到诸如手机白皮书、针对不同技术的开发人员指南、SDK（软件开发工具包）和相关 API（应用程序编程接口）等文档和工具。该网站还包含由索尼爱立信开发人员支持小组管理的论坛、内容广泛的知识库、提示与技巧、示例代码和新闻。

索尼爱立信还为专业开发人员提供技术支持服务。有关这些专业服务的更多信息，请访问索尼爱立信开发者世界网站。

## 文档约定

### 产品

本文档所涉及的索尼爱立信手机使用如下通用名称：

通用名称 系列	索尼爱立信手机
K700	K700i、K700c
S700	S700i、S700c、S710a
K500	K500i、K506c、K508i、K508c、F500i
Z500	Z500a
V800	V800、V802SE
K300	K300i、K300c、K300a
J300	J300i、J300c、J300a
K750	K750i、K750c、D750
W800	W800i、W800c
Z800	Z800i
K600	K600i、K608i
V600	V600i
W600	W600i
W550	W550i、W550c
Z520	Z520i、Z520c、Z520a

W900	W900i
W810	W810i、W810c、W810a
K610	K610i、K610c、K618i
K800	K800i、K800c、K800a
K790	K790i、K790c、K790a
Z530	Z530i、Z530c
W300	W300i、W300c
K510	K510i、K510c
K310	K310i、K310c、K310a
W700	W700i、W700c
Z525	Z525a
Z550	Z550i、Z550c、Z550a
W850	W850i、W850c
Z710	Z710i、Z710c
W710	W710i、W710c
Z610	Z610i
W830	W830i、W830c
Z558	Z558i、Z558c
K320	K320i、K320c

## 印刷体约定

在本文档中，代码以 Courier 字体编写，例如：

```
Vector3D position = new Vector3D( 0, 100, 256 );
```

# 商标及确认

Sun、Java 以及所有基于 Java 的商标和徽标是 Sun Microsystems, Inc. 在美国和其他国家 / 地区的商标或注册商标。

Mascot Capsule 是 HI Corporation 的商标。

Bluetooth™( 蓝牙 ) 是 Bluetooth SIG Inc. 的商标或注册商标。

此处提及的其他产品和公司名称可能是其各自所有者的商标。

# 文档历史记录

更改历史记录		
2004-07-04		第一版。
2004-08-23	版本 R2A	第二版。进行了较小编辑更改。增加了 K500 系列和 Z500 系列信息。
2004-09-20	版本 R2B	第二修订版。进行了较小编辑更改。
2004-10-19	版本 R3A	第三版。编辑更改。增加了 V800 系列。
2004-10-28	版本 R3B	进行了较小编辑更改。
2005-03-09	版本 R4A	增加了有关 K300、J300、K750、W800、Z800 和 K600 系列的信息。
2005-04-13	版本 R4B	进行了较小编辑更改。
2005-08-01	版本 R5A	增加了有关 V600、S600、W600 和 Z520 系列的信息。
2005-08-08	版本 R5B	S600 系列手机更改为 W550。
2005-09-30	版本 R6A	第六版。
2005-10-24	版本 R7A	第七版。增加了有关 W900 系列的信息。
2005-10-27	版本 R7B	进行了较小编辑更改。
2006-01-04	版本 R8A	第八版。增加了有关 W810 系列的信息。
2006-02-13	版本 R9A	第九版。增加了有关 K610 系列的信息。

2006-02-28	版本 R10A	第十版。增加了有关 K800、K790、Z530、W300、K510 和 K310 系列的信息。
2006-04-12	版本 R11A	第十一版。增加了有关 W700 和 Z525 系列的信息。
2006-05-19	版本 R12A	第十二版。增加了有关 Z550、W850、Z710 和 W710 系列的信息。
2006-08-22	版本 R13A	第十三版。增加了有关 K618i 和 Z610 系列的信息。
2006-09-25	版本 R14A	第十四版。增加了有关 K320、Z558 和 W830 系列以及 Z550a 手机型号的信息。

# 目录

<b>概述 .....</b>	<b>8</b>
索尼爱立信 Java 平台 .....	9
3D 引擎 .....	10
Java ME 集成 .....	10
Micro3D Ver.3 功能 .....	11
JSR 184 — 适合 Java ME 平台的 Mobile 3D Graphics API .....	11
适合 Java™ ME 平台的索尼爱立信 SDK .....	12
<b>Mascot Capsule Micro3D Version 3 .....</b>	<b>13</b>
图形命令 .....	14
对象导入 .....	14
Micro3D Version 3 类 .....	16
<b>适合 Java ME 平台的 Mobile 3D Graphics API (JSR 184) .....</b>	<b>17</b>
图形命令 .....	18
对象导入 .....	18
JSR 184 类 .....	19
<b>使用入门 .....</b>	<b>21</b>
Micro3D Version 3 模型转换 .....	22
Micro3D Version 3 的转换流 .....	22
针对 Micro3D Version 3 的 Java ME 平台应用程序设置 .....	23
Mobile 3D Graphics API (JSR 184) .....	26
<b>参考 .....</b>	<b>27</b>
有用链接 .....	28
索尼爱立信 3D 知识库 .....	28
常见问题 .....	28
Mascot Capsule Version 3 .....	29
Mascot Capsule Version 4 (JSR 184) .....	33

# 概述

人们对使用移动设备和显示 3D 内容的需要在不断增长。此类 3D 内容可能是游戏的一部分，例如游戏场景和角色。或者，也可能是以 3D 图形这种独特方式显示信息的某个程序。

新一代索尼爱立信手机 K700、S700、K500、Z500、V800、K300、J300、K750、W800、Z800、K600、V600、W600、W550、Z520、W900、W810、K610、K800、K790、Z530、W300、K510、K310、K320、W700、Z525、Z550、Z558、W850、Z710、W710、Z610 和 W830 系列，Java 平台 JP-3、JP-4、JP-5、JP-6 和 JP-7 提供了内置的 3D 图形支持，可以处理此类内容。本节给出了有关这些手机中 3D 技术的简要概述。这里假定读者具有关于 3D 图形和 MIDP 的一般知识。



# 索尼爱立信 Java 平台

为帮助内容开发人员改编适合多数手机的应用，索尼爱立信为其手机制定了 Java 平台策略。索尼爱立信手机有两种主要的 Java 平台方式，一种方式针对其基于 Symbian OS 的智能手机，另一种方式针对其功能手机和大众市场手机，即非基于 Symbian OS 的手机。通常，每个平台版本都在多个手机型号中使用。

下表列出了本文所涉及的索尼爱立信手机的 Java 平台版本。有些平台功能是可选的，也就是说是可以配置的。例如，只有当特定手机真正支持 Bluetooth(蓝牙) 无线技术时，它才支持 Bluetooth™ API (JSR 82)。

JP = Java 平台

	功能	手机
JP-3	CLDC 1.1、MIDP 2.0、JTWI (JSR 185)、JSR 120、JSR 135、Nokia UI API 1.1、JSR 184、Mascot Capsule Ver. 3	K700、K500、F500、Z500、S700、K300 和 J300 系列
JP-4	CLDC 1.1、MIDP 2.0、JTWI (JSR 185)、JSR 120、JSR 135、Nokia UI API 1.1、JSR 184、Mascot Capsule Ver. 3	V800 和 Z800 系列
	可选：VSCL 2.0	V800 系列
JP-5	CLDC 1.1、MIDP 2.0、JTWI (JSR 185)、JSR 120、JSR 135、Nokia UI API 1.1、JSR 184、Mascot Capsule Ver. 3、JSR 75	K750、W800、K600、V600、Z520、W700 和 Z525 系列
	可选：JSR 82	K750、W800、K600、V600、Z520、W700 和 Z525 系列
JP-6	CLDC 1.1、MIDP 2.0、JTWI (JSR 185)、JSR 120、JSR 135、Nokia UI API 1.1、JSR 184、Mascot Capsule Ver. 3、JSR 75、JSR 172、JSR 205	W600、W550、W900、W810、Z530、W300、K510、K310、K320、Z550 和 Z558 系列
	可选：JSR 82	W600、W550、W900、W810、Z530、W300、K510、K320、Z550 和 Z558 系列
JP-7	CLDC 1.1、MIDP 2.0、JTWI (JSR 185)、JSR 120、JSR 135、Nokia UI API 1.1、JSR 184、Mascot Capsule Ver. 3、JSR 75、JSR 172、JSR 205、JSR 234（仅限于相机功能）	K610、K800、K790、W850、Z710、W710、Z610 和 W830 系列
	可选：JSR 82	K610、K800、K790、W850、Z710、W710、Z610 和 W830 系列

## 3D 引擎

索尼爱立信手机具有 HI Corporation 的 3D 呈现引擎。该引擎的手机版本是 Mascot Capsule Engine Micro3D Edition (“Mascot Capsule”)。此 3D 呈现引擎软件支持在索尼爱立信手机上显示 3D 图形数据的应用程序的实时运行。

支持索尼爱立信 Java 3D 的手机中的 3D 引擎支持以下程序

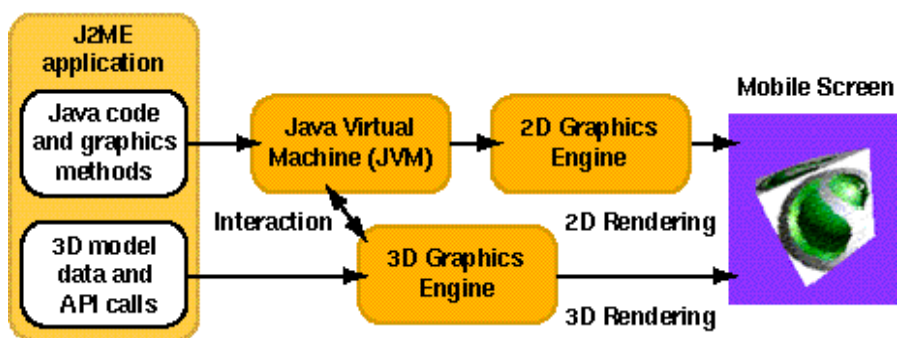
- Mascot Capsule Engine Version 3, 以下简称 “**Micro3D Ver.3**”
- Mascot Capsule Engine Version 4, 该程序支持最新版本的 Mobile 3D Graphics API(M3G) 这一针对 Java ME 平台的新型移动标准, 以下简称 “JSR 184”。

采用 JP-3、JP-4、JP-5、JP-6 和 JP-7 的手机都配备了 Micro3D Ver.3 和 JSR 184, 可实现 3D 图形。3D 引擎由一个压缩软件引擎组成, 该引擎可在手机平台严格的内存和处理能力约束下实时呈现 3D 图形。

## Java ME 集成

在下面的内容中, “3D 引擎” 一词指的是标准 JSR 184 或 Micro3D Ver.3 引擎。

Mascot Capsule 3D 图形引擎可与索尼爱立信手机中所提供的 Java ME 平台无缝集成。要编写 3D MIDlet, 您需要首先使用 Java 语言设计其控制逻辑和图形算法。通过 3D 引擎自带的 Micro3D Ver.3 API 直接访问它, 或者凭借 JSR 184 API 通过 KVM, 可以从 MIDlet 操纵 3D 引擎。3D 引擎可与手机的 JVM 协同工作以呈现 3D 图像, 如图 1 所示。



**图 1。**使用 Mascot Capsule Micro3D Ver.3, 无需特殊操作就可以显示 3D 内容。只需在 Java 应用程序内添加对其 3D API 的调用。

开发人员不需要进行任何特殊操作就可以将 3D 功能添加到 Java ME 平台 MIDlet。您可以使用您喜欢的 Java 工具编写 MIDlet, 以创建 3D 应用程序。要访问 3D 引擎, 您只需添加对 3D API 的调用, 不需要任何特殊链接或其他后处理。

## Micro3D Ver.3 功能

---

全世界许多的手机都已经使用 Micro3D Ver.3 引擎。因而，3D 引擎基于功能强大、经过实地测试的代码。Micro3D Ver.3 引擎使用 32 位整数即可执行所有图形操作。简单地说就是，该引擎不需要浮点硬件或图形加速器就可以运行。

Micro3D Ver.3 引擎为索尼爱立信开发人员提供的几种功能包括：

- 访问、合成和显示在资源中存储的复杂 3D 对象的能力
- 定义 3D 对象的表面特征，例如明暗处理、透明度和纹理映射
- 操纵对象，例如旋转、缩放或移动
- 动画支持

这些功能将在本文档后面介绍。有关 Mascot Capsule 引擎的更多信息，请访问：

<http://www.mascotcapsule.com/>。

## JSR 184 — 适合 Java ME 平台的 Mobile 3D Graphics API

---

JSR 184 是适合手机中 3D 的标准 API。预计它将包括在支持 3D 的所有手机中。该引擎在计算过程中使用浮点算法以达到最高精度。

JSR 184 为索尼爱立信开发人员提供的几种功能包括：

- 访问、合成和显示在资源中存储的复杂 3D 对象的能力
- 定义 3D 对象的表面特征，例如明暗处理、透明度和纹理映射
- 操纵对象，例如旋转、缩放或移动
- 动画支持
- 雾化效果
- 变形效果
- 网格化，包括界面网格化
- 纹理平铺
- 多色光

## 适合 Java™ ME 平台的索尼爱立信 SDK

---

适合 Java™ ME 平台的索尼爱立信 SDK（以下简称“SDK”）包含

- Micro3D Ver.3 API 和 API 文档
- JSR 184 API。  
有关 JSR 184 API 文档，请访问 <http://www.jcp.org/en/jsr/detail?id=184>。
- Micro3D Ver.3 MIDlet 模拟
- JSR 184 MIDlet 模拟
- 设备工具 — 设备浏览器和设备连接代理工具
- 对适合 Java ME 平台应用程序的机上调试 (ODD) 的支持。开发人员将能够对其 MIDlet 执行实时源代码调试，以便进行标准 Java ME 平台、Micro3D Ver.3 和 JSR 184 开发。

索尼爱立信手机的开发人员指南为开发人员提供了设备专用的 Java 技术规范，包括手机内存、堆大小、画布大小、受支持的 MMAPI、平台和网络功能等等。

有关 SDK 和开发人员指南，请访问[索尼爱立信开发者世界](#)。

# Mascot Capsule Micro3D Version 3

本节将详细介绍 Mascot Capsule Micro3D Version 3（以下简称 Micro3D Ver.3）。这个常用的版本使用专利 3D Java API。

## 图形命令

---

Micro3D Ver.3 图形 API 以立即模式运行，在这种模式下，会将图形命令发出给图形管道，然后由呈现引擎执行该命令。这些命令可执行以下功能：

- **3D 对象描述。**这些命令可描述组成 3D 对象的三角形、多边形和四边形平面的坐标和颜色。这些命令还可制定应用于每个对象的纹理映射和颜色混合过程。还支持点精灵 (Point Sprites)。
- **3D 环境配置。**这些命令可调整场景灯光的位置和特征、其透视图（平行或投射）、要应用的明暗类型以及是否使用半透明混合。
- **管理呈现过程。**这些命令可描述在呈现操作中使用的颜色混合类型，并指定剪裁区域的坐标。这些命令还可引导引擎对对象执行变换（如旋转或移动）。刷新命令指示在手机屏幕上呈现和绘制场景的时间。
- **图形原型。**这些命令可绘制点、线、三角形和四边形。

可将命令存储在列表中。这样便可将 3D 对象存储为一个包含指定该对象的形状、颜色以及其明暗和混合特征的命令列表。要绘制对象（假定是一个火箭），应用程序会将存储了对火箭描述的命令列表发出给 Micro3D Version 3 图形管道。这样，3D 引擎便可绘制出该火箭。在 Micro3D Ver.3 中，只有当调用 `Flush()` 时才会在屏幕上绘制像素，而不是在发出命令后立即绘制。

## 对象导入

---

由于使用图形命令描述任何中等复杂度的 3D 对象都是很困难的，因此 Micro3D Ver.3 允许您导入由 3D 制作程序生成的 3D 对象数据。此数据作为资源存储在 MIDlet 的 JAR 文件中，可通过 Micro3D Ver.3 `Figure` 类进行访问。此类包含 3D 对象信息。可将各种不同的属性（如对象的明暗类型）与此类的一个实例相关联。其他关联可指定诸如旋转、缩放或移动等操作。也可将命令列表附加到 `Figure` 的各个实例。

除了简化 3D 模型的设计以外，对象导入还为开发人员提供了其他两个重要优势。第一，对象导入可降低更新 3D 场景的计算开销。不需要进行计算就可以生成 3D 对象本身。可将任何呈现或变换操作直接应用于由 `Figure` 实例维护的多边形信息。

第二，`Figure` 类有助于实现模型的动画效果。模型的当前 3D 对象布置代表一个空间方位信息。操作描述类 `ActionTable` 与 `Figure` 相关联，并存储可更改模型空间方位信息的动作。通过利用程序控制在一段时间内更改模型的空间方位信息，操作表可以使模型产生动画效果。也可从 JAR 资源中导入 `ActionTable`。

这样，您不仅可以使 3D 制作程序生成 3D 对象，而且还可以让程序使这些对象产生动画效果。HI Corporation 为多个常用的 3D 建模程序提供了插件，这些插件可将模型的几何和动画序列信息导出为 Micro3D Ver.3 引擎可以读取的文件格式。有关插件和工具的更多信息，请参见第 21 页的“使用入门”。

`Figure` 可以立即绘制或呈现，也就是说将处理所有变换和效果，但是像素数据的实际生成要推迟到执行刷新命令为止。这样，通过首先布置并更新场景的 `Figure` 对象，然后执行刷新图形命令，便可以模拟 3D 保留模式。“保留”呈现命令的主要原因是：所有呈现的图形都会自动按深度排序（Z 向排序）。

请注意，Micro3D Ver.3 不支持将 `Figure` 分组为将组成图形“世界”的页和节点，这是保留模式操作的一

项功能。在许多情况下，呈现方案将适合于大多数 3D 操作。

## Micro3D Version 3 类

Micro3D Ver.3 API 包含十个类，这些类用于管理对 3D 内容的显示和控制。这些类中的一部分用于存储对象信息（例如前面介绍的 Figure 类），而其他类将属性附加到这些对象的实例。其他类可描述在场景中使用的光和纹理。一个类就可处理所有呈现操作，而实用程序类提供可帮助设计图形算法的方法。下表给出了有关这些类的简短摘要。

类	说明
ActionTable	存储用于控制相关 3D 模型的移动的操作数据。可从 JAR 资源中读取操作表数据
AffineTrans	处理用于缩放或移动 3D 对象的矩阵数学
Effect3D	包含与特定 3D 对象相关联的呈现效果数据。此类效果包括透明度、明暗类型和灯光
Figure	存储 3D 对象的几何信息 以及空间方位信息信息。可从 JAR 资源中读取此数据
FigureLayout	3D 对象的所有呈现信息（例如其位置、大小和方向）的容器
Graphics3D	执行所有呈现功能。必须绑定到要在其中绘制像素的 LCDUI Graphics 对象
Light	包含灯光信息，例如其方向和强度
Texture	存储对象或环境（背景）的纹理数据。可从存储为 JAR 资源的 .bmp 文件中读取此数据
Util3D	包含在 3D 算法（特别是 <code>sin()</code> 、 <code>cos()</code> 和 <code>sqrt()</code> ）中使用的实用程序方法
Vector3D	包含构成 3D 矢量的方法，或者从这些矢量中提取信息（例如，矢量的 x、y 和 z 分量）。还可执行某些矢量数学例程

有关这些类及其方法的详细信息，可以从适合 Java™ ME 平台的索尼爱立信 SDK 所附带的 JavaDoc 参考中获得。



# 适合 Java ME 平台的 Mobile 3D Graphics API (JSR 184)

本节将详细介绍适合 Java ME 平台的 Mobile 3D Graphics API（以下简称 JSR 184）。有关这个标准化 API 的更多信息，请访问

<http://www.jcp.org/en/jsr/detail?id=184>。

## 图形命令

---

JSR 184 支持保留和立即的场景呈现模式。JSR 184 保留模式的优势在于：它使不甚了解 3D 技术的开发人员也能快速地构建复杂的 3D 模型。保留模式隐藏了许多低级别、琐碎的细节，从而简化了 3D 世界的设计。

与 Micro3D Ver.3 不同，JSR 184 并不执行可存储到列表中并发出到图形管道中的图形命令。

由于保留模式功能是在立即模式操作的基础上建立的，因此保留模式还得益于硬件加速。这种设计方法还允许通过 JSR 184 同时使用保留和立即模式操作。通过同时使用这两种模式，程序员便可最大程度地发挥手机有限资源的作用。您可以针对不同的 3D 图形任务使用相应的模式，从而有效地平衡呈现速度与资源消耗。

请注意，JSR 184 的设计目的是可在 OpenGL ES 的基础上实现。而索尼爱立信 JSR 184 实现并不支持 OpenGL ES 命令。

## 对象导入

---

使用 JSR 184，您可以导入包含几何对象数据和动画信息的 3D 内容。对象数据还可以包含属性信息，例如材质、雾化、纹理、合成模式及其他信息。使用 3D 制作程序，并使用插件转换器将 3D 内容导出为 .m3g 文件，可以在 PC 和工作站上生成 3D 内容。JSR 184 规范包含有关这种文件格式的详细说明。该文件可存储为能在应用程序中被 `Loader` 类读取的 JAR 资源。`Loader` 在读取该资源时会生成 3D 对象实例。可以导入并显示具有动画效果、灯光和相机视图的完整的场景图。

由于 .m3g 文件格式是一种开放标准，因此它能够在高端平台上生成 3D 内容，然后很容易地导入该内容，并在执行 JSR 184 的多种手机平台间共享。

# JSR 184 类

与 Micro3D Ver.3 相比，JSR 184 包含许多类：总共有三十种，如下表所示。数量众多的类增加了执行固件占用的空间。API 的设计目的是降低 RAM 的使用率，具体方法是按引用存储大部分对象，而不是生成对象的实例。

JSR 184 支持**保留**和**立即**模式的操作。保留模式使用通过节点结构树链接 3D 世界中所有几何对象的场景图。图中的每个节点都代表一个几何对象，并包含以下信息：其外观、空间位置以及其相对于其他节点的行为方式。您可以使用 Node 的子类对象（例如光源、精灵和网格）组成 3D 世界。使用 Group 类，您可以将未排序的 node 对象收集在一起，而 World 类可定义一个特殊的 Group 节点，该节点可作为 3D 世界中所有节点的顶层容器。要使用保留模式显示 3D 世界视图，应对 World 节点执行 Graphics3D 呈现方法。

TriangleStripArray 和 VertexArray 类可用作构建块，您可以利用这些构建块组成更复杂的 3D 对象和模型。Node 的其他子类（如 Mesh）也可用于构建更复杂的 3D 几何对象。其他 Node 子类可控制场景图的灯光和视角（分别为光源和相机）。有一些类可定义 3D 对象的视觉属性（材料、雾化、合成模式及其他属性），这些类可在立即模式或保留模式下使用。

本文档不提供对这些类的详细介绍。有关更多信息，请参考位于 <http://www.jcp.org/en/jsr/detail?id=184> 上的 JSR 184 规范文档。下表给出了有关这些类的简短摘要

类	说明
AnimationController	管理用于构成动画序列的一组对象的位置和速度。
AnimationTrack	包含针对一个目标对象控制单个动画属性的信息。动画序列包含一组由 AnimationController 处理的 AnimationTrack。
Appearance	存储一组组件对象的呈现属性。这些属性可描述每个对象的材质特征、其多边形、对象在场景中的混合方式以及所有雾化效果。它还可指定纹理映射特征以及包含的图像。
Background	用于指定清除或填充给定 viewport（绘图区域）的颜色或图像。
Camera	场景图中用于确定场景视角的节点。用于从 3D 到 2D 呈现场景图。它还可确定场景图的哪些元素是可见的（剪裁）。
CompositingMode	一种 Appearance 组件，包含在像素合成操作中使用的属性。
Fog	一种 Appearance 组件，包含用于将雾化效果应用于像素的属性。
Graphics3D	在呈现图像时应用的图形上下文。此对象将绑定到呈现目标，该目标为 Canvas、易变的 Image 或 CustomItem。呈现目标将收到呈现操作输出。此类可处理此 API 的所有绘图操作。
Group	一种场景图节点，可将一组未排序的节点存储为其子节点。
Image2D	存储可用作纹理、背景或精灵图像的二维图像。
IndexBuffer	描述如何连接顶点，以使这些顶点组成一个几何对象。作为一个抽象类，您可使用 TriangleStripArray 构建对象。
KeyFrameSequence	存储动画数据。可将数据存储为带有时间标记和矢量值的关键帧的序列。每个关键帧都存储给定时间点的动画特征值。

Light	表示光源的场景图节点。灯光具有颜色、强度和类型（环绕、定向、全方向和聚光）。
Loader	下载在图形操作中使用的节点组件、场景图节点以及整个场景图。该数据可包括 Camera 和 Light、Appearance 和 Material 属性类，以及诸如 AnimationTrack 和 KeyFrameSequence 的动画类。用于导入现有的 3D 内容。
Material	一种 Appearance 组件，存储在灯光计算中使用的材质属性。
Mesh	表示 3D 对象的场景图节点。使用多边形平面描述对象。Mesh 包含 IndexBuffer 实例中定义的三角形带，以及其存储在 Appearance 实例中的其视觉属性。
MorphingMesh	表示顶点变形多边形网格的场景图节点。MorphingMesh 与 Mesh 相似，但是前者的顶点是在 VertexBuffer 中使用加权值计算得到的。这使得 MorphingMesh 对象能够更改其形状。
Node	所有场景图节点的抽象基类。Camera、Group、Light、Mesh 和 Sprite3D 都是其子类。
Object3D	可成为 3D 世界一部分的所有对象的抽象基类。
PolygonMode	包含多边形级属性的外观组件。这些属性包括背面 / 正面选择、灯光计算、透视纠正、明暗处理和缠绕。
RayIntersection	包含通过 Group.pick() 方法向其添加的光线的对象。RayIntersection 存储对与每条光线交叉的网格或精灵的引用，以及有关交叉点的信息。
SkinnedMesh	表示经过轮廓动画处理的多边形网格的场景图节点。允许顶点组彼此独立地更改，同时使多边形网格平滑变形。提供了一种动画特征的有效方法。
Sprite3D	使用 3D 位置表示 2D 图像的场景图节点。
Texture2D	一种 Appearance 组件，存储 2D 纹理图像以及管理如何将图像应用于子网格的属性。
Transform	一个 4x4 的矩阵，包含用于执行变换操作的值。
Transformable	Node 和 Texture2D 的抽象基类。它定义用于处理节点和纹理变换的常用方法。
TriangleStripArray	定义一个用于构成几何对象的三角形带数组。
VertexArray	表示顶点位置、法线、颜色或纹理坐标的整数矢量数组。
VertexBuffer	包含对定义一组顶点特征的 VertexArrays 的引用。
World	一种特殊 Group 节点，是场景图的顶层容器。

有关这些类及其方法的详细信息，可以从适合 Java™ ME 平台的索尼爱立信 SDK 所附带的 JavaDoc 参考中获得。

# 使用入门

Java ME 平台执行环境结合了 JSR 184 和 Micro3D V3。这表示可使用常见的 Java 开发工具来编写 3D 应用程序。但是，生成 Mascot Capsule 为组成并呈现 3D 模型而导入的文件是一个更复杂的过程，将在下面进行介绍。

## Micro3D Version 3 模型转换

在市场上有许多商用 3D 制作工具，可用于生成适用于 PC 应用程序和游戏机的 3D 模型和动画。通常，这些工具所生成的模型和动画数据不能与 JSR 184 或 Micro3D Ver.3 环境兼容。举例来说，3DS Max、LightWave 和 Maya 就是 3D 建模开发工具。

HI Corporation 提供了许多可在这些制作工具内使用的插件，以便将模型和动画数据导出到中间文件。随后，可将这些文件转换为 Micro3D 对象和操作格式文件。这样，对于为游戏控制台创建 3D 模型的应用程序开发人员来说，就可以利用同样的制作工具和专业知识为移动应用程序制作 3D 内容。

在将模型和操作数据导出为其中间格式之后，可以使用多种实用工具进行后处理和最终转换。使用 **PAC.exe** 实用程序，您可以在中间模型数据文件中查看和调整特定类型的多边形数据的属性。转换器实用程序 **Micro3D\_Converter.exe** 可将中间 3D 数据和操作文件转换为 Micro3D 可以使用的文件格式。使用 **PVMicro.exe** 应用程序，您可以用 3D 模型自带的 Micro3D Ver.3 3D 格式来查看这些 3D 模型，以预览最终效果。

下表概述了在导出 / 转换过程中涉及的文件类型，以及在对 Micro3D Ver.3 消耗的 3D 模型和动画数据进行导出和转换的过程中涉及的文件类型。

文件扩展名	说明
.bac	包含从 3D 制作工具中导出的 3D 模型数据的中间文件
.tra	包含从 3D 制作工具中导出的动画（操作表）数据的中间文件
.mbac	包含已格式化的、以在 Micro3D 中使用的 3D 模型数据的最终文件
.mtra	包含已格式化的、以在 Micro3D 中使用的 3D 操作表数据的最终文件
.bmp	存储 8 位纹理图像数据

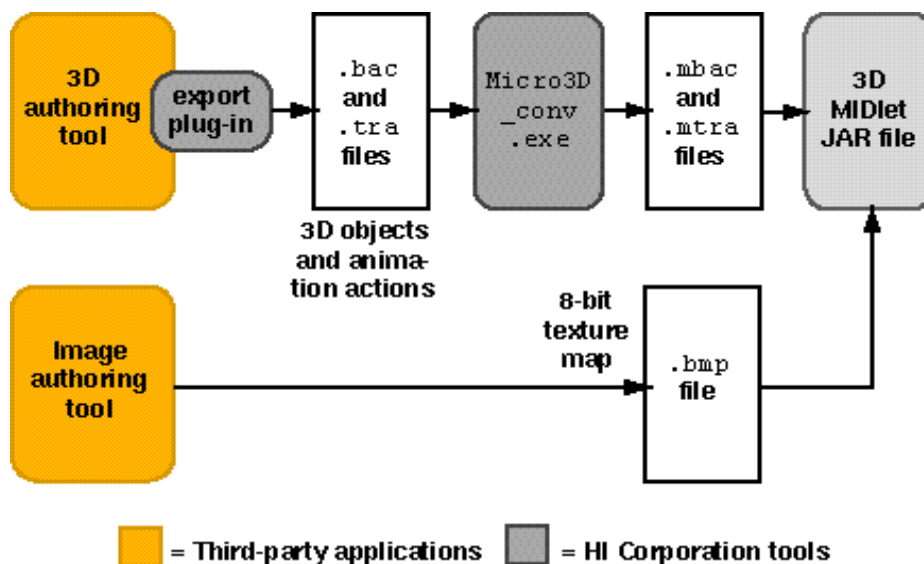
管理导出 / 转换过程所需要的所有插件、工具和文档都可以从 Mascot Capsule 网站 [http://www.mascotcapsule.com/toolkit/sony\\_ericsson/](http://www.mascotcapsule.com/toolkit/sony_ericsson/) 中获得。这些工具是由 HI Corporation 提供的，所有反馈和故障报告都应提交给此网站。

**注意：**所使用的 3D 建模程序的版本号一定要与导出插件的版本号相同。

## Micro3D Version 3 的转换流

- 使用市场上的各种商用 3D 制作工具创建 3D 模型。
- 使用可用插件从 3D 制作应用程序中导出数据。插件将输出中间 3D 格式文件（.bac 和 .tra）。
- 可选：使用 **PAC.exe** 工具在 .bac 文件中更改并编辑多边形属性。
- 使用 **Micro3D\_Converter.exe** 工具将中间格式文件（.bac 和 .tra）转换为 Micro3D Ver.3 格式文件（.mbac 和 .mtra）。

- 可选：使用 **PVMicro.exe** 工具在最终文件（.mbac 和 .mtra）中预览 3D 模型。
- 将最终文件（.mbac 和 .mtra）作为资源添加到 3D MIDlet 的 JAR 文件中。
- 将纹理映射（.bmp 文件）添加到 3D MIDlet 的 JAR 文件中。图像必须使用 8 位像素（256 色）。



## 针对 Micro3D Version 3 的 Java ME 平台应用程序设置

本节介绍了如何在移动应用程序内正确初始化 Micro3D 的图形环境并使用它显示 3D 图形。简单说，配置和使用 Micro3D Ver.3 图形引擎的正确顺序如下所示：

1. 加载 3D 模型数据。
2. 初始化模型的 3D 对象的所有属性。
3. 设置相机。
4. 设置灯光和任何效果。
5. 初始化图形环境，呈现模型并显示它。

## 加载 3D 模型数据

正如上面所述，通过导入 3D 对象数据来组成 3D 模型要容易得多。更确切地说，在生成 Figure 实例时，其 3D 几何信息和空间方位信息可从 .mbac 文件中导入。所有需要的动画数据都必须从 .mtra 文件中导入并存储在 ActionTable 实例中。最后，如果模型使用纹理映射，则要从 .bmp 文件中导入纹理映射。所有这些文件都作为 JAR 资源存储在 MIDlet 中。

对于每个 3D 对象，以下代码都适用：

```
// Declare the 3D model's elements

Texture texture;
Figure figure;
ActionTable action;

// Load (import) the 3D object data from JAR resources

try {
    texture = new Texture( "/texture.bmp", true);
    figure = new Figure ( "/figure.mbac" );
    action = new ActionTable( "/action.mtra" );
} catch (Exception e) {}
```

正如代码所示，导入数据使得构建 3D 模型的整个过程变得简单、容易。

## 初始化 3D 对象属性

加载图形几何、它的一些操作信息以及纹理。将操作信息和纹理映射与图形相关联：

```
//Use this texture with the 3D object
figure.setTexture( texture );

// Associate the action table with this object
figure.setPosture ( action, 0, 1 );
```

setPosture() 方法的参数可将 action 表与 figure 相关联。动画以模型的第一帧以及表中的第一个操作命令开始。

## 设置相机

通过定位观察相机，场景的视角已被设置。设置相机的观察角度矢量：

```
Vector3D position = new Vector3D( 0, 100, 256 );
Vector3D look = new Vector3D( 0, -2048, -4096 );
Vector3D up = new Vector3D( 0, 4096, 0 );
AffineTransform trans = new AffineTransform();
trans.lookAt( position, look, up );
```

这些操作可创建矢量，然后将矢量输送给特殊的 AffineTransform 方法 lookat()。此方法将所提供的矢量应用于矩阵变换，然后从指定的视角生成场景几何。



除了指定观察角度外，还需要将其他属性与 3D 对象相关联。FigureLayout 类可作为某些图形属性的容器，例如 Figure 视角 (POV) 变换、放大、位置数据和透视类型。FigureLayout 实例应按如下方法进行初始化：

```
FigureLayout layout; // Make an instance

// Set the distance between the 3D model and the camera.
layout.setPerspective( 1, 4096, 512 );

// Set where the camera's center is positioned in the display.
layout.setCenter( 128, 128 );

// Set the camera angle using the supplied parameters.
layout.setAffineTrans( trans );
```

布局设置好以后，就可以调整场景灯光。

## 设置灯光和任何效果

Micro3D Ver.3 场景有两种灯光：聚光和环绕光。这两种光都是白色的，因此没有色彩描述。以下代码可设置聚光的方向矢量并配置两种灯光的强度：

```
Vector3D lightDirVec = new Vector3D( -146, 293 439 );
int lightDir = 3730;
int lightAmbient = 1626;

// Make instance of Light from the prepared parameters
Light light = new Light( lightDirVec, lightDir, lightAmbient );
```

这些灯光特征将被添加到 Effect3D 实例中：

```
Effect3D effect;
effect.setLight( light );
```

请注意，也可将其他属性添加到效果实例中，例如明暗类型以及是否启用颜色透明。

## 初始化图形环境，呈现模型并显示它

准备好 Figure 及其多个属性后，就应该呈现并显示该图形。Graphics3D 类用于实现此目的，因为它包含所有呈现方法。但是，必须先将此类的输出“绑定”到 LCDUI Graphics 对象，这样最终结果才会出现在屏幕上。然后绘制场景，最后发布 Graphics3D 实例。以下代码显示了实现此目的方法：

```
Graphics3D g3d = new Graphics3D();
Graphics g;
// Bind the 3D graphics context to the MIDP Graphics object.
g3d.bind(g);

try {
    g3d.renderFigure( figure, 80, 100, layout, effect );
    g3d.flush(); // Draw the rendered figure on-screen
} catch( Exception e ) {}

// Release the MIDP Graphics object.
g3d.release( g );
```

请注意 renderFigure() 方法是如何将模型 (figure) 与其布局信息 (layout) 以及所有特殊效果 (effect) 一起提供的。在呈现过程中将这些属性应用于 Figure。flush() 方法将生成的像素展现在屏幕上。

# Mobile 3D Graphics API (JSR 184)

---

用于 Micro3D Ver.3 的 3D 建模工具同样可用于创建在 Mobile 3D Graphics Version 4 中使用的数据。(m3g)

HI Corporation 提供了许多可在这些制作工具内使用的插件，以便将模型和动画数据导出到中间文件。

下表概述了在导出 / 转换过程中涉及的文件类型，以及在对 Micro3D Engine Version 4 (JSR 184) 消耗的 3D 模型和动画数据进行导出和转换的过程中涉及的文件类型。

文件格式	说明
H3T	从 3D 制作工具中导出的中间 3D 数据
M3G	在 JSR 184 中使用的 3D 数据格式
PNG	纹理的文件格式

在将模型和操作数据导出到其中间格式之后，可以使用多种实用工具进行后处理和最终转换。使用 **V4Converter.exe** 实用工具，您可以将 H3T 数据转换为 M3G 数据。使用 **V4Examiner.exe** 工具，您可以查看并检验 M3G/H3T 文件

管理导出 / 转换过程所需要的所有插件、工具和文档都可以从 Mascot Capsule 网站 [http://www.mascotcapsule.com/M3G/download/e\\_index.html](http://www.mascotcapsule.com/M3G/download/e_index.html) 中获得。这些工具是由 HI Corporation 提供的，所有反馈和故障报告都应提交给此网站。

## 参考

本章包含一些链接和一个常见问题解答部分。

## 有用链接

---

- 索尼爱立信开发者世界 ([http://developer.sonyericsson.com/site/znch/home/p\\_home.jsp](http://developer.sonyericsson.com/site/znch/home/p_home.jsp))
- 适合索尼爱立信手机的 Mascot Capsule 工具包 ([http://www.mascotcapsule.com/toolkit/sony\\_ericsson](http://www.mascotcapsule.com/toolkit/sony_ericsson))
- Mascot Capsule Micro3D V4 - m3g 工具和插件 ([http://www.mascotcapsule.com/M3G/download/e\\_index.html](http://www.mascotcapsule.com/M3G/download/e_index.html))
- 适合 Java ME 平台的 Mobile 3D Graphics API (JSR 184) (<http://www.jcp.org/en/jsr/detail?id=184>)
- Discreet 的 3DS Max (<http://www4.discreet.com/3dsmax/>)
- Newtek 的 Lightwave 7.5 (<http://www.newtek.com/products/lightwave/product/7.5/>)

## 索尼爱立信 3D 知识库

---

索尼爱立信手机利用 Mascot Capsule 技术提供了基于硬件的 3D 支持。Mascot Capsule 提供了两种形式的 3D 支持（Version 4 和 Version 3），分别提供了 JSR 184 和高性能专利 API。

对于在开发使用 Mascot Capsule 技术的 3D 内容时遇到的常见问题，本知识库尝试提供广泛的答案。

### 常见问题

---

**问题：**索尼爱立信手机使用 Mascot Capsule 作为 3D 引擎，这在 SDK 中反映出的准确度如何？

**答案：**模拟器只使用为在桌面环境中使用而设计的 Mascot Capsule 的略作修改版本。事实上，准确度是非常高的。

**问题：**是否可在模拟器中实现对 3D 应用程序的正确计时？

**答案：**为了模拟真实手机的速度，可以在“性能”标签下的索尼爱立信 WTK2 首选项中调整“图形延迟”首选项。但是一定要注意，输入也将延后。

**问题：**我导出的 3D Max 7 模型非常大，如何减小其大小？

**答案：**纹理通常会消耗大部分的空间，最好在一开始尝试减少使用的纹理数量。另一个注意事项是，HiCorp 为 Maya、3DStudio 和 LightWave 提供的 H3T 导出器总是导出 TrueColour PNG。请确保在导出 H3T 文件后，对 PNG 文件进行优化。

**问题：**在尝试运行一个使用 3D 功能的 MIDlet 时，我收到一个异常，根本原因是“无法加载 zayitlib.dll”？

**答案：**这表示在 SDK 中使用 Mascot Capsule 3D 引擎可准确反映手机实现过程。此错误表示在安装过程中无法正确注册库。关闭所有应用程序，然后尝试重新安装 SDK 以纠正问题。

**问题：**如果我按顺序加载多个 M3G 文件，即使所有引用都是空的且调用了 System.gc，但经过一段时间后，我还是收到一条“引发虚拟 IOException”消息？

**答案：**这是在较早的 Z500 固件中的已知问题，在以后的版本中已得到解决。

**问题：**哪些索尼爱立信手机支持 Mascot Capsule Micro3D 和 JSR 184？

**答案：**K700、S700、K500、Z500、V800、K300、J300、K750、W800、Z800、K600、V600、W600、W550、Z520、W900、W810、K610、K800、K790、Z530、W300、K510、K310、K320、W700、Z525、Z550、Z558、W850、Z710、W710、Z610 和 W830 系列

**问题：**从哪里可以获得 Mascot Capsule Version 3 Javadoc？

**答案：**API Javadoc 包含在适合 Java™ ME 平台的索尼爱立信 SDK 中。

**问题：**从哪里可以获得 JSR 184 Javadoc？

**答案：**JSR 184 Javadoc 包含在适合 Java™ ME 平台的索尼爱立信 SDK 中，也可以在 <http://jcp.org/aboutJava/communityprocess/review/jsr184/index.html> 中找到

**问题：**适合 Java® ME 平台的索尼爱立信 SDK 为 Java 3D 开发提供了哪些支持？

**答案：**该 SDK 包括：

- Mascot Capsule Version 3 API 和 Javadoc
- Mascot Capsule Version 4 API（适合 Java ME 的 Mobile 3D Graphics API，即 JSR 184）和 Javadoc
- 适合 Mascot Capsule Version 3 及 Mascot Capsule Version 4 (JSR 184) 的机上调试

## Mascot Capsule Version 3

---

**问题：**如何在 SDK 中构建项目？

**答案：**在 Ktoolbar 中创建一个新项目。选择 项目 -> 设置 菜单，在 API 选择 标签上选择“自定义”平台。选中相应的复选框“Mascot Capsule Ver.3 API”。现在，您将能够通过 Mascot Capsule V3 API 的支持构建您的项目。

**问题：**可使用哪些工具来创建模型？

**答案：**所有 Micro3D 转换工具都位于以下 Mascot Capsule 网站 [http://www.mascotcapsule.com/toolkit/sony\\_ericsson](http://www.mascotcapsule.com/toolkit/sony_ericsson)。

**问题：**可以使用哪些纹理大小？

**答案：**与 Mascot Capsule Version 4 不同，可以使用任意大小，例如 64 x 192。纹理图像文件必须小于 256x256 像素并使用 8 位（256 色）。

**问题：**我无法使用 Z 轴在手机上定位模型？

**答案：**请确保定义了 Z 轴位置。请参见 `FigureLayout` 类中的 `setPerspective()` 方法。或许是设置了平行投影，而不是透视投影？

**问题：**好像不支持在屏幕上定义并呈现您自己的几何模型。

**答案：**请参考 `Graphics3D::renderPrimitives`。

**问题：**能否将场景呈现到屏幕外的对象？

**答案：**可以。使用 `Graphics3D()` 中的 `bind()` 方法可将 3D 绘图定向到图形对象 `javax.microedition.lcdui.Image::getGraphics()`。

**问题：**Mascot Capsule v3 支持哪种图形格式？

**答案：**Mascot Capsule v3 仅支持未压缩的编制索引的 256 色 BMP。不支持在创建纹理时使用 BMP RLE 压缩。

**问题：**使用 `AffineTransform` `setRotation` 方法时，整数参数表示什么含义？

**答案：**将旋转角度指定为一个整数，假定  $360^\circ = 4096$ 。旋转是指在正轴方向看是顺时针旋转。

**问题：**什么是 .bac 和 .mbac 文件，它们有什么关系？

**答案：**.bac 文件是由 3D 建模工具使用 Mascot Capsule Version 3 插件生成的。此文件以适当的 Mascot Capsule 格式表示您的模型。.bac 文件将会与 Mascot Capsule 提供的 Micro3D 转换器工具结合使用，这样可生成 .mbac 文件，可对该文件进行优化以在手机上占用较小的空间和使用率。

**注意：**不能使用二进制 (bac) 文件格式来获取三角形的访问权限（以进行高级碰撞检测和直接的三角形变换）。为此，请使用 ASCII 格式。一定要了解哪些模型可以由 ASCII 文件表示，哪些不能。

**问题：**如何执行 ToonShading

**答案：**ToonShading 或 CelShading 无法由硬件执行，因为所有支持 3D 的手机上的当前硬件都不支持对图形管道的直接操纵（明暗处理）。但是，可以通过算术方法在软件中创建 CelShaded 模型。具体做法是在呈现所有模型之后，手动将像素绘制到图形缓冲区。如果操作不正确，这个方法的速度会非常慢。关键是不（或者几乎不）使用纹理来处理未经明暗处理的模型，而后通过像素绘制在图形缓冲区中直接对其进行明暗处理。

**问题：**是否有对透视纠正的支持？

**答案：**`FigureLayout` 类提供了一个简便的 `setPerspective` 方法，可使用该方法进行透视纠正。

**问题：**可以将一个情景中的多个 3D 对象导出到一个 bac 文件中吗？

**答案：**可以，但是建议您将它们导出到单独的 bac 文件中。这样就可以在您的应用程序中单独加载这些对象。还有，这样可将手机上加载文件时出现的错误降到最低。有些手机可能会在加载过于复杂的文件时引发异常。

**问题：**常量 `PATTR_BLEND_NORMAL`、`PATTR_BLEND_ADD`、`PATTR_BLEND_HALF` 和 `PATTR_BLEND_SUB` 表示什么含义？

**答案：**这些命令用于混合多边形。下表显示了如何将多边形背景与纹理混合。

将背景中的 RGB 颜色分别与纹理中的 RGB 颜色混合。对于同一多边形，不能同时使用 Half、Addition 和 Subtract。

属性	背景	纹理
Normal	0%	+ 100%
Add	100%	+ 100%
Half	50%	+ 50%
Sub	100%	- 100%

问题：是否有方法可使用 Mascot Capsule Version 3 来访问 MIDlet 中 3D 对象的顶点？如果没有，我该如何检查模型是否触地？

**答案：**检查模型是否触地实际就是多边形碰撞问题。通常，开发人员都会创建某些类型的关卡编辑器，使用该编辑器为所有可移动的对象制作清晰的边界（通常是大的边界框）。这样，不必具有顶点访问权限，就可以通过算术方法确定一个对象是否接触到另一个对象。只需一个包含所有地面边界框和一个适合您模型的边界框或边界球的文件。

要为一个模型创建边界球，只需知道两个角度变量：模型的中心，以及球的半径。在转换模型时，大多数开发人员都会通过读取 ASCII 格式导出边界框。这样就不需要进行直接的顶点操纵。

问题：当远剪贴板的最大值为 32767 时，如何使用 Mascot Capsule Version 3 制作一个赛车游戏 MIDlet？要获得准确的正弦和余弦以及矢量分析，我通常需要对场景进行缩放，这样 4096 就表示 1.0。如果我这样做，我的视图圆锥只有 7 个单位，很难制作赛车游戏。

**答案：**虽然您可以使用许多技巧在固定点范围内制作赛车游戏，但是第一步就会开始降低准确性，且在光栅化步骤中会很大程度地降低准确性（因为视角范围少于 200 像素）。

问题：能否模拟雾化效果？

**答案：**可以，但是雾化的每个像素都涉及额外的颜色混合操作，这使得软件的执行速度非常慢。有许多方法可以模拟雾化效果，使用半透明带纹理的三角形就是其中一种。

问题：哪些不透明性（透明性）设置可用于 3D 对象中的纹理？

**答案：**可以使用 3 种不透明性设置：0%、50% 和 100%。如果想要通过只更改模型的材质颜色来改变纹理颜色，可以选择 50% 设置。

问题：是否有可用于 3D 碰撞检测的方法？

**答案：**与具有光线交叉执行方法的 Mascot Capsule Version 4 不同，Version 3 不支持这个功能。

问题：为了执行正确的碰撞检测框架，我们需要检索 Mascot Capsule Version 3 中 Figure 的所有顶点的功能。如何获得该功能？

**答案：**如果您选择使用 `renderPrimitives()`，您要几何数据。当使用从资源中加载的 Figure 时，无法检索几何数据。但实际上，您会拥有对象的一些更适合的表示来进行碰撞检测，通常为球体层次、OBB，也可能是（如果确实需要）已呈现几何的简化形式。您也可以跟踪游戏中的顶点，并使用 3D 图形来绘制场景。

**注意：**请不要使用二进制格式 (bac)，而使用 ASCII 格式来存储顶点数据和模型数据。

**问题：**能否对 MIDlet 中的 3D 模型使用平铺（循环）纹理？

**答案：**在您熟悉的 3D 工具中平铺纹理并使用 HiCorp 工具将纹理导出，就会获得想要的平铺效果。但是，这并不总是有效的，使用时应该注意。要生成地板和地面，请对带纹理的三角形或四边形使用 `renderPrimitives` 方法。

**问题：**在场景内使用的多边形数量是否有限制？

**答案：**不受限制。但是，性能将成为一个限制因素。

**问题：**在 Micro 3D 中似乎没有选择功能？能否指定背面选择？

**答案：**可以执行背面选择，以及使用 PAC 工具将其直接设置为图形数据。

可从 [http://www.mascotcapsule.com/toolkit/sony\\_ericsson](http://www.mascotcapsule.com/toolkit/sony_ericsson) 中下载 PAC 工具。

**问题：**为什么模型不能显示在查看器中？

**答案：**bac 数据的 UV 值都是整数。如果在 3D 工具的 UV 坐标值中显示小数，则在导出到 bac 文件时将出现偏差。错误范围可能只是多了或少了一个像素，但是这足以使纹理变形。目前，没有对此问题的解决方法。但是，作为改进您作品外观的一个方法，我们建议如眼睛和字母数字字符等模型应使用分辨率更高的纹理。此外，如果尺寸过于小，则在实际手机中显示时，细小的纹理可能会聚在一起。这是由于手机的屏幕分辨率较低造成的。

另一个注意事项是，导出没有任何相机或灯光的模型可能会导致查看器错误显示这些模型。



## Mascot Capsule Version 4 (JSR 184)

---

**问题：** 哪些工具和文档可用于创建 Mascot Capsule Version 4 (JSR 184) 模型？

**答案：** 所有官方的 Mascot Capsule 创建工具和插件都位于 [http://www.mascotcapsule.com/M3G/download/e\\_index.html](http://www.mascotcapsule.com/M3G/download/e_index.html)。

许多项目正在多方参与下进行，一些常用的链接包括：

- <http://sourceforge.net/projects/juinness/>  
芬兰的项目，致力于创建能够读取几乎所有常见的 3D 格式的 M3G 转换器。
- <http://fivedots.coe.psu.ac.th/~ad/jg/objm3g/>  
介绍如何将 OBJ 文件转换为 M3G 的教程。
- <http://www.renderware.com/>  
Renderware，一种在整个行业中都非常有名的程序。
- <http://www.jcodeworks.com/demomidp.html>  
M3G 导出器，可制作 OBJ 文件并变为可由提供的加载类读取的格式。

**问题：** 加载 M3G 时引发“嵌套的 URI”IO 异常？

**答案：** 当使用多个映射以及使用多于 256 色的纹理时，会出现这种情况。确保您的纹理映射使用的调色板不超过 256 色。在 3D Max 下，在导出器对话框中有一个“纹理工具”，其中有一个选项“优化的调色板”可达到此目的。

**问题：** 在使用自带的 3D Max M3G 插件进行导出时，我注意到输出要大于使用 Mascot Capsule 导出器插件导出的内容？

**答案：** 与 Mascot Capsule 导出器插件不同，3D Max M3G 插件不执行任何与计算三角形带相关的优化，因此可能生成更大的文件尺寸。我们建议使用 HiCorp 导出器插件。

**问题：** 是否有可应用于 3D 碰撞检测的方法？

**答案：** 没有，没有为碰撞检测定义任何 API。但是，`RayIntersection` 类可计算 3D 世界中的光线交叉。

**问题：** 可以使用哪些纹理大小？

**答案：** JSR-184 中的所有纹理的大小必须为  $2^n \times 2^n$ （例如 32x32、128x64）。建议的最大尺寸为 256x256（就堆内存而论，256 x 256 纹理通常上限）。

**问题：** 使用缩放工具在 3D Max 中对我的模型进行缩放，然后将其导出以进行预览时，模型不能按预期呈现。

**答案：** 3D Max 缩放工具对对象顶点不起作用，但可对变换矩阵应用缩放。您应该直接缩放顶点，以便确保在 3D Max 中的预览可以最接近地反映 Mascot Capsule 呈现。也可以通过 Mascot Capsule v4 API 随时缩放对象。

**问题：** 从 3D Max 中导出我的模型，但该模型没有像对 Mascot Capsule 预期那样呈现？

**答案：** 一定要使修改器堆栈很短，并在导出前折叠堆栈。某些修改器可能在内部包含这样的材质：在 3D Max 中不显示，但在由 Mascot Capsule 呈现时显示。在导出过程中，M3G 查看器也可能是很重要的。先在查看器中测试所有文件，然后再将其转换并插入游戏中。

问题：能够对 3D 模型使用平铺（循环）纹理？

**答案：**可以，支持平铺纹理。

问题：是否有方法直接更改对象的顶点，而不必更改为位置数组并将其设置为 VertexArray。直接引用位置将会更好。我认为设置方法是复制原型数组。

**答案：**很遗憾，只有一种引用对象顶点的方法，即复制在对象的 VertexArray 中存在的顶点，创建一个新数组，然后删除旧数组并执行垃圾收集。这与要制作每个帧的方法不同。但是，有时必须这样做，因为在对模型应用了许多转换之后，模型容易损坏。这时，必须将模型重新加载到内存中，以保留其原始顶点坐标。

问题：当我将第二个纹理应用于图形时，图形消失了？如何使用多个纹理？

**答案：**您必须为要在对象中使用的每个纹理指定纹理坐标：

```
vertexBuffer.setTexCoords(0, TEXTURE_ARRAY, (1.0f/255.0f), null);
vertexBuffer.setTexCoords(1, TEXTURE_ARRAY, (1.0f/255.0f), null);

appearance.setTexture(0, texture); // add the texture to the appearance.
appearance.setTexture(1, texture2); // add the second texture to the appearance
```

问题：我为一个三角形指定了三种不同的颜色（每个顶点一个颜色），但却只显示了一种颜色。这是为什么？

**答案：**要显示所有三种颜色并获得平滑的明暗效果，您必须为所使用的 PolygonMode 指定 SHADE\_SMOOTH 参数。

例如：

```
public Mesh createTriangle() {
    short []POINTS = new short[] {-1,-1, 0,
                                   1,-1, 0,
                                   0, 1, 0};

    int []INDICES = new int[] {0, 1, 2};
    byte []COLORS = new byte[] {127, 0, 0,
                                 0, 127, 0,
                                 0, 0, 127};

    int []LENGTH = new int[] {3};

    VertexArray POSITION_ARRAY, COLOR_ARRAY;
    IndexBuffer INDEX_BUFFER;

    POSITION_ARRAY = new VertexArray(POINTS.length / 3, 3, 2);
    POSITION_ARRAY.set(0, POINTS.length / 3, POINTS);
    COLOR_ARRAY = new VertexArray(COLORS.length / 3, 3, 1);
    COLOR_ARRAY.set(0, COLORS.length / 3, COLORS);
    INDEX_BUFFER = new TriangleStripArray(INDICES, LENGTH);

    VertexBuffer vertexBuffer = new VertexBuffer();
    vertexBuffer.setPositions(POSITION_ARRAY, 1.0f, null);
    vertexBuffer.setColors(COLOR_ARRAY);
    Mesh mesh = new Mesh(vertexBuffer, INDEX_BUFFER, null);
}
```

```

Appearance appearance = new Appearance();
PolygonMode polygonMode = new PolygonMode();
polygonMode.setPerspectiveCorrectionEnable(true);
polygonMode.setCulling(PolygonMode.CULL_NONE);
polygonMode.setShading(PolygonMode.SHADE_SMOOTH);
appearance.setPolygonMode(polygonMode);
mesh.setAppearance(0, appearance);
return mesh;
}

```

**问题：**我已创建了一个三角形，但是当我将该平面旋转 90 度后它消失了，直到旋转到 270 度后才再次显示。三角形的背面是黑色的？

**答案：**这是一种称为背面选择的方法，使用它可通过只显示每个多边形的前面而获得性能。您可以在 `PolygonMode` 类中指定该选择：

```

PolygonMode polygonMode = new PolygonMode();
polygonMode.setCulling(PolygonMode.CULL_BACK);

appearance.setPolygonMode(polygonMode);

```

**问题：**当 `VertexArray` 不允许使用浮动数字且纹理坐标必须是 0 到 1 之间的值时，如何在具有多个顶点的图形中设置纹理坐标？

**答案：**使用 0 到 255 之间的值输入您的纹理坐标。当您将纹理坐标应用于 `VertexBuffer` 时，按照  $1.0f/255.0f$  的比例缩小坐标：

```
vertexBuffer.setTexCoords(0, TEXTURE_ARRAY, (1.0f/255.0f), null);
```

不要忘记，纹理映射的宽度和高度必须是 2 的非负次幂（2、4、8、16、32、64、128、256）。

**问题：**.m3g 文件中的对象应该是已压缩的还是未压缩的？

**答案：**将 .m3g 文件在应用程序 .jar 文件中传送时，最好将对象数据保留为未压缩状态。这是由于 .jar 文件总是压缩的，因此压缩对象数据只会增加手机在加载 m3g 文件时的负荷，而在文件尺寸方面没有任何好处。

**问题：**能否为一个立方体只指定八个顶点并指定正确的纹理和法线坐标？

**答案：**可以绘制三个面共享一个顶点的立方体，并为其设置颜色。但是，纹理和法线坐标需要每个顶点都具有唯一的值，这意味着您必须为立方体的每个面指定四个顶点。

指定立方体的八个点，然后使用 `IndexBuffer` 指定绘制三角形的方式，请记住，立方体的每个面都由两个三角形构成。

例如：

```
short POINTS[] = new short[] {1, 1, 1, // 0
                               1,-1, 1, // 1
                               -1, 1, 1, // 2
                               -1,-1, 1, // 3
                               -1, 1,-1, // 4
                               -1,-1,-1, // 5
                               1, 1,-1, // 6
                               1,-1,-1}; // 7

int INDICES[] = {2, 3, 0, 1, // Front Face
                 4, 5, 2, 3, // Left Face
                 6, 7, 4, 5, // Back Face
                 0, 1, 6, 7, // Right Face
                 4, 2, 6, 0, // Top Face
                 3, 5, 1, 7}; // Bottom Face

int[] LENGTHS = new int[] {4, 4, 4, 4, 4, 4};

POSITIONS_ARRAY = new VertexArray(8, 3, 2);
POSITIONS_ARRAY.set(0, 8, POINTS);
INDEX_BUFFER = new TriangleStripArray(INDICES, LENGTHS);
```

要指定纹理和法线坐标，我们必须对立方体的每个面使用四个点：

```
short []POINTS = new short[] {-1, -1, 1,      1, -1, 1,      1, 1, 1,      -1,
1, 1, // Front
                               1, -1,-1,     -1, -1,-1,     -1, 1,-1,      1,
1,-1, // Back
                               -1, -1,-1,     -1, -1, 1,      -1, 1, 1,      -1,
1,-1, // Left
                               1, -1, 1,      1, -1,-1,     1, 1,-1,      1,
1, 1, // Right
                               -1, 1, 1,      1, 1, 1,      1, 1,-1,     -1,
1,-1, // Top
                               -1, -1,-1,     1, -1,-1,     1,-1, 1,      -1,-
1, 1}; // Bottom

short []TEXTCOORDINATES = new short[] {0, 1,      1, 1,      1, 0,      0, 0,
0, 1,      1, 1,      1, 0,      0, 0,
0, 1,      1, 1,      1, 0,      0, 0,
0, 1,      1, 1,      1, 0,      0, 0,
0, 1,      1, 1,      1, 0,      0, 0,
0, 1,      1, 1,      1, 0,      0, 0};
```

问题：相机调准没有对给定节点更新。

**答案：**必须呈现节点才能进行相机调准。

问题：为什么我的场景中的某些多边形发生闪烁？

**答案：**Micro3D 对每个单独的多边形都使用 Z 向排序方法。因此，重叠的多边形在呈现时可能发生闪烁。我们建议尽可能使用避免发生重叠的模型。

问题：如果要在移动终端上呈现多边形，则必须确保正确操作 PVMicro.exe，这种理解是否正确？

**答案：**任何不能在 PVMicro.exe 查看器中显示的 3D 内容也不能在手机上显示。

问题：什么是 UV 设置？

**答案：**在 Mascot Capsule 引擎上运行的内容必须具有所有多边形顶点的经过纹理映射的 UV 坐标。

问题：如何解决想要映射其中一些多边形，而不想映射其他多边形的问题？

**答案：**这种布置是不允许的。要求（并假定）所有多边形都具有纹理映射。这是一个限制要求。

问题：Micro 转换器没有指出任何问题，但是查看器没有显示任何图像，且操作帧计数增加了。如何在 PVMicro 中显示经转换的文件？

**答案：**因为会播放操作帧，所以问题可能与缩放有关。

问题：为我的模型的身体和手指指定的白色纹理未能在 PVMicro.exe 屏幕显示中成功显示。这些部分变成黑色。这是什么问题？

**答案：**Micro 3D 规范的调色板索引 0（零）被不加区分地转换为黑色。无论透明还是不透明，都必须将调色板索引 0 指定为纯黑 (0, 0, 0)。