

## 积木建模 I

### Begin

本文没有那么严格，主要面向普通人群，只是粗略在分类和讨论和介绍。

### 积木与基元

以下[基元]一词和[积木]一词在许多意义上等价，但是[基元]的概念更宽，[基元]可以包含[积木]，[积木]一词比较受限于视觉可见的大尺寸实体，以及受限于现实世界中的积木概念。在偏理论一点的环境，[基元]是更专业一点描述，对应有 **primitive** 也会出现在许多模型语义分割的论文里，[基元]模型主要就是立方体、球形之类的三维软件中最常见的基本模型，这些模型也可以认为是一个[积木]，但是如果我们将一个点称为[基元]时，把他当成一个[积木]好像不太能接受。

[积木]这个词更直观，更偏向于现实世界，简单一点的就是幼儿园里有的，球，圆锥，方块等，复杂一点的有需要一点建模去生成的部分，比如一个楼梯等。上面也提到了，点、线、面本身在积木上是不太好表示的，因为实体模型总是有厚度的，积木不能进行任意放大、缩小等操作。因此，主要是二级基元，一根细直线，其实也可以看作是一个可视半径较小的圆柱。

复杂一点的就是乐高积木，高达拼接玩具等。

通常来讲，我们平常能看到最常见的模型，按表达方式有：多边形面片模型、四面体与六面体网格模型、体素模型、点云模型、参数曲面模型等，他们本身之间也存在直接和间接的转换。

普通大众比较熟悉的，游戏中常见的是多边形面片模型，体素模型[我的世界]，工业界中，考虑到要把模型制作成实体金属等，常见的是实体模型，也可以理解为体素，虽然四面体与六面体是个比体素概念更宽泛的体素，为了简化，我们这里体素专门指的就是立方体[可以三个坐标轴方向进行拉伸变成长方体]。

那么我这里介绍的是积木建模，一部分是角色[物件]建模，一部件是场景建模。角色建模主要聚焦于构成一个模型，但是这个模型有许多积木构成。场景建模聚焦于生成一个场景，会有大量的模型，每个模型可以是一个积木，也可以是各种积木组成的一个模型。

这里说的[角色]一词，包括常见的模型，车、房子，一个体积稍大点的房子，加上院子、装修、花草就是一个小场景，但是缩小到一个参考人物模型大小时，也可以称为一个物件。

### 基元分级

通常来讲，基元的复杂度不会太高，就是些基本图案，以下尝试先按复杂度进行简单划分，划分并不是绝对的，只是为了方便后续的粗颗粒度上的讨论。

#### 一级

单个点线面，由不同空间维度决定最低基元了，

## 二级基元

Cuboid, Sphere, BezierCurve, Cone, Cylinder, Line, Rectangle 等等这些都可以作为 2 维和三维的图形的基本单元。

通常来讲，因为三级基元也可以三角面组成或线段组成，也可以看起来比较复杂，因此这里的 Line 主要是指较低复杂度的直线段，或者由更少参数表达的直线段或其他类型的基元模型等。举例来说：一条 BezierCurve 本身是一个基元模型，但是如果把它变成了点阵逼近的曲线，并且由 Line[point1,point2,...]来表达时，它就可以是二级基元模型，也可以不是二级基元模型。具体还是要看这条曲线的复杂程度，比如是不是简单曲线，是不是封闭曲线，是否打了很多节。当我们重新去拟合这条 Line 时，可能需要多个简单一点的 BezierCurve。

同样，对于可三角化的模型来讲，不同的种类的基元，复杂程度和表达方式也是不一样的，其中，如果把所有的基元三角化，那么圆和曲面的三角形数量相对于平面风格模型来说是需要更多的。

## 三级基元

三级基元本身基本上就是一个独立的有一定复杂度的模型了，可以拿出来作为一个独立的模型来构建场景。比如复现一些现实世界中的物件、建筑、场景等。

高级的基元可以由低级的基元表达或生成或复现或逼近，也可以由最低级基元直接生成。

## 积木系统

虚拟世界中自由度当然很高，我这里分为面向玩家还是面向作者简单讨论。

网络游戏中的家园系统和积木系统[三级基元和二级基元]，我的世界就是典型的面向玩家的积木系统[一级、二级基元，可以理解为一个砖头或立体体素就是一个点]，其中家园系统更是更广义的积木系统，家园系统有可能附带一个积木系统。

## 面向玩家

### 游戏家园系统

家园系统中主要是三级基元，游戏公司制作一些完整的基础模型，比如房子，门窗等家具，玩家只能限制性地修改，因为涉及到性能优化和赚钱等各种问题，玩家一般是不能任意 DIY 的，家具是要花钱或花游戏时间来获取的。家园系统如果附带了积木系统，也可以支持更低级的基元。这个不需要我多说了，B 站搜索一下家园就有许多作品。

### 积木系统

## 面向作者

### 面向开发者

面向开发者，主要就是各种行业的专业人士，建模工作者等。一般使用的是通常使用的所有技术，只要有某个特性，有时间有资源一般不会限制自己不去用它。

一级、二级、三级基元都会由不同的人员使用。但是这里提到的基元，通常是不受限制的二级基元，就是同样一个曲面，我可以自定义曲面。基元的限制性问题的会在后文谈到。

### 面向玩家作者

开发商提供一些基础的平台或工作，玩家可以创造一些 UGC 内容来分享。因为受限于工具，

和其他各种原因，有可能表达能力受限，因此不是那么自由。

此时，二级基元有可能是限制性二级基元，比如我想要一个曲面模型，不好意思，没有导入模型的工具，或自定义曲面的工具，或者待开发商开放一些工具，比如我的世界要转换成指定格式的体素化模型，也有一些工具软件的支持。

### 玩家 UGC 创作需求与开发者提供工具之间的矛盾

首先这个矛盾的来源并不只是双方直接的，可能是第三方或整个世界。

双方之间直接的，比如开发者就是要进行风格限制，玩家想瞎搞，这是直接的。

因为玩家的功能受限，表达能力受限，所以有限制性二级基元==>一级基元==>二级基元==>三级基元这么一条迂回曲折的道路。

当然整个过程中，也会因此发现一些不同的风格，是直接创作时无法想到和简单复现的。

### 直接导入模型

让普通玩家直接导入模型这件事情，远没有普通人认识的这么简单。只讨论一小部分问题，举例来说，他可以分为导入原创模型和导入网上直接下载模型。

导入原创模型理论上来说是受支持的，但是如何跟非原创的或非原创可任意商用模型等等区分开来是个难点。这方面涉及到一个版权问题，如何来控制，是否有授权、是否是自用、是否可商、是否可改、是否可二传、是否可仿照等等细节问题很多。理论上讲，最简单的是人工审核，但是人也不知道你这个模型是哪来的呀，作者自己录制建模过程证明视频？

还有一个问题是性能，你要知道 3D66 网上随便下一个模型，导出 obj 文件后可能是 1 个 G 大小，一百万个面。。。开发者有一套成熟的方案、系统和流程以及专业人士来解决减面、适配、导入使用问题。特别地：如果落地系统中，底层模型普通是几万面的，这里就有一个性能矛盾，即使是专业一点的玩家，愿意自己做这个事情，还是需要开发商提供工具支持，其他问题我就不多说了。

所以这件事情，一些简单的解决方案是，支持一个受限制的模型导入，比如支持一个 10 万个面的模型的导入，或对接常见的模型文件格式，或支持自定义积木，或支持曲面工具，或支持基元模型。

### 自定义积木[基元]

如上面提到，在这样一个系统中，我们想要自定义积木功能，提出了一些需求背景，然后回到积木建模这件事情上来，这才是本文的主题。

### 积木建模

积木建模的本质就是用更低级的积木组装成更高级的积木的过程，其中不限制高级积木向低级积木的投影[球缩小表示一个点，砖头压扁变成一个平面面片，一个模型某一个轴向的缩放分量变为 0 等操作]

### 动机和必要性

从我个人角度，我首先是作为玩家来做这件事情，是因为上述提到的创造力与落地积木系统的创造力支持程度不匹配这个矛盾。

从家园游戏开发者角度，必要性是给玩家提供积木系统这件事情。

从行业开发者角度，大家已经在普通使用积木建模了[想想基元，假如一个样条曲面也作为基元看待时]，但是最终使用的时候，有时还是会三角化后再输出使用。而体素建模渲染本身在图形行业这几年也是一种新的方式，也有一些特别的优势。但是模型是否有必要三角化是待具体考虑的。以此样条曲面的例子为，即使建模系统和落地积木系统都支持样条曲面，但是如何对接是一个问题，有可能不如三角化直接。

从研究者角度，首先基元分割本身也是语义分析中的一个点，比如点云模型的语义分割，简化看待此问题时，一个点就是一个立方体积木。  
其次，是否存在新的基元表达方式是可以探索的。

从艺术创作角度，这是一种独特的风格，或者过程中可以生成一些独特风格的作品。

### 积木建模本身的必要性

如果一个模型，我可以直接用参数或系统支持的更直接的方式来生成，如果不是为了追求特殊风格，我可能没有必要使用积木来搭起来逼近。

所以有时是为了适配终端落地的系统，从头部就开始进行积木建模，具体我们可以在[3D 软件与积木建模]章节里聊聊。

### 积木系统

这里指狭义的积木系统，每个积木有一些基本可变参数，每个积木作为一个整体，比如可以进行按三个轴缩放、位移、旋转、镜像等操作。

积木是否可以相交、重叠、堆叠

在游戏系统中，两个积木不一定可以相交，或者可能可以受限制相交，比如定义横梁和门窗是可以相交的，比如考虑到有物理碰撞的积木是不能相交的等等。

在一个面向玩家作者的编辑器中，积木可能可以相交，积木相交表达能力更强。在本文的轮胎案例中，就是积木相交的一个情况。

### 极小化积木建模

极小化积木建模的意思是这样的，一个圆环，最小可以由多少个低级基元表示？

简单举例来说，一个圆环，可以由 20 个积木生成，也可以由 2000 个三角形积木？那么 20 就是我要的极小值。

极小化积木建模有时是可以节省资源的，比如所有 2 级基元模型，不管三角化表示时顶点数或面数是多少，性能都是一样的时候。

### 正向积木建模

积木建模与阵列

10 年的时候我就对剑法、剑阵的物资非常感兴趣，因此开始设计与制作一些特别喜欢的，并最后输出成视频作品。期间虽然对数字图像处理一直有玩，但直到 20 年的时候，因为对线条风格窗花产生了兴趣，于是把这两者结合了一下，产生了更多 3D 模型化的作品。

## 案例演示@线条风格建模@门窗

这一块算法背景涉及到模式图案的生成，比如 `HatchFilling` 和 `PatternFilling` 等填充算法，是我比较喜欢的功能，以后再扩展介绍一下。这种算法属于把一个基本图案当成一个基元，然后使用参数控制，不断在一个形状区域内填充。

## 案例演示@城市

## 案例演示@轮胎

正向的难点

懒

要人工参与，要设计，因此比较适用于简单的模型

## 逆向积木建模

逆向积木建模的意思是，我有一个初始模型，点云也好，三角面模型也好，我把他转化成最终的积木系统适配的模型

## 积木建模之三角网格建模

案例演示@阴阳兔子@LowPoly 风格

法拉利 2020 款，历史有许多作品是这种分组三角化积木模型

## 积木建模之空间矩形建模

主要指的是建模的目标是大部分是矩形为主的比较适配。矩形建模的一个好处是适配于房子等自带矩形属性的模型，不容易出现局部问题特别大的拼接缝隙。

## 积木建模之立方体素建模

此类建模方式，一般在我的世界和体素风小游戏中使用得较多，是强风格化的。但是也不一定“好看”，许多人并不喜欢。

而我做得较多的一种是拿来生成，底层是体素化的，但是看起来就是普通的房子这种，体素可视风格没有那么强。我这么做的动机的一部分原因是有一定的性能上的优化需求，同一个房子，用三角形拼接和用体素拼接后者经过优化后需要的积木数量更少。风格上，我一直在探索强风格化与弱风格化之间的平衡问题。

## 积木建模之有向平行六面体建模

我的世界游戏中或体素化模型场景中你不会看到斜面，也不会看的斜着的柱子。这里的平行六面体的一种特殊情况，就是把立方体在空间中旋转一个角度的效果。

特别地，如果计算一个空间点集的有向长方体包围盒时，求出来可能是由平行四边形或平行六面体表达。如果 `Parallelogram` 或 `Parallelepiped` 在积木系统是不受支持的，其实可以用一个立方体+旋转参数表达。

## 综合积木建模

考虑到必要性，如果积木系统中已经提供了一个标准球形基元，你再用三角形去拼一个标准



球形，可能是没有必要的，但是如何把这个球形识别出来，然后再用系统有的球形积木去融合或替换掉这个三角形拼接的环形模型就是一个难点啦。

这里就涉及到逆向建模中的模型分割问题，最理想的是有一个算法，比如语义分割模型能自动按模型部件的语义[这是一个球形，这是一个手指，这是一个头等语义]进行分割，否则我们就要在 3D 软件里进行人工分割。这一块功能也可以称为模型分组，这个分组可以分为积木级别的分组和基元级别的分组和贴图级别的分组等等，是个多层次的分组。

综合积木建模的意思就是不同的分组使用不同的积木[基元]来替换掉，会涉及到各种技术，上面提到的是一个核心的点，来源于整个模型不同的部位使用不同的技术。举例来说，想象一个人物模型，毛发、布料、模型等均是不同的，所用到的建模技术也是综合的。

逆向的难点

比如减面困难，效果未必达到预期。局部与整体在的冲突性问题。

对于生成某种风格的 3D 模型来说，语义分割+BoundingBox+BoundingConvex 是一个比较好的减面方案，但是语义分割受限于分割算法，它并不一定是用来解决你这个特定需求的。。。非标准曲面在指定面数限制下保持光滑是相对困难的，标准曲面比如一个圆环这种是相对简单的。

### 积木建模与限制性基元建模

在普通的建模中，我们对模型的三角化是任意的，也可以去控制某些区域的三角面的大小、密度、形状等等。

对于三角面模型，我任意一个三角形都可以由三个顶点的坐标表达，但是如果系统中只存在一个直角三角形呢，不能生成一个任意的三角形呢？此时一种解决方案是把这个任意的三角形，分解成两个直角三角形。

### 限制性基元

以上这两个直角三角形称为限制性基元。限制性基元有时是因为系统受限被动选择，有时是为了风格或某种别的目的主动选择。

### 3D 建模软件、Blender、Mathematica 与积木建模

以 Blender 为例，如果我们要实现一个积木建模，那么在 Blender 中可以按积木建模的方式进行，一是可以把 Blender 本身当成一个积木系统，二是与后处理系统对接，比如分组、切割模型的目标要依赖后处理系统的需求。

0@Blender=积木系统

1@Blender==>积木模型

2@Blender==>后处理系统==>积木系统

Mathematica 与积木建模[其中 Mathematica 可以替换成 Python\C++\Lua 或其他 3D 软件]作为 3D 软件，Mathematica 可以作为一个弱 3 维建模软件，强参数 3D 建模软件存在。作为脚本语言，Mathematica 可以作为一个脚本编辑语言存在，并且也可以融合其他语言的程序。

Mathematica==>积木模型

Blender==>Mathematica==>积木模型

Mathematica==>Blender==>Mathematica==>积木模型

---

End

纯手写还是很累人的，懒得写了。

有空把部分独立出来补充更多细节，以及分享更多好玩的案例吧。感兴趣的，有啥想法或哪些局部看不太清楚的可以评论里或其他各种方式交流沟通。

欢迎感兴趣、有时间、和有能力做低面建模的可以一起玩，也可以学着玩，可以一起来生成一些艺术作品，比如我经常需要 2 万面以下-10 万面以下的模型，但是没有时间仔细精修模型、配色和贴图等。