

计算机图形学

期末大作业



实验名称 游戏《纪念碑谷》场景的模拟实现

实验日期 2016/12/19

组 员 1452244 李松霖 1452238 杨力

1452246 杨斌诚

1452254 孔思尹 1452288 李文

指导老师 Junqiao Zhao 赵君峤

一、 Project Title

游戏《纪念碑谷》场景的模拟实现

二、 Motivation

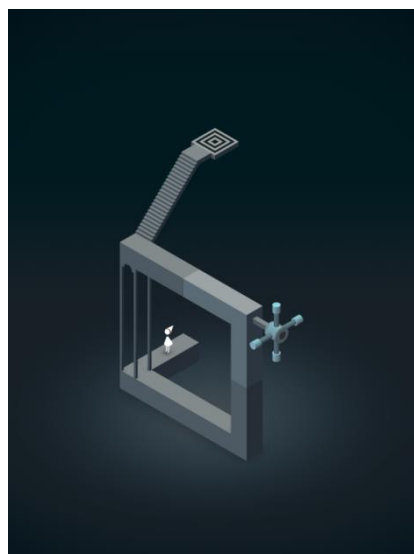
- a) 《纪念碑谷》是 USTWO 公司开发制作的解谜类手机游戏，于 2014 年正式发行。结合自顶向下的等距设计、鲜艳的色彩搭配和萦绕心头的音效，玩家需要通过全部关卡将一位叫做“Ida”的沉默公主引导至碑谷。



- b) 《纪念碑谷》大量的运用了矛盾空间的思想，比如说潘洛斯三角形（Penrose triangle）、莫比乌斯环等等。矛盾空间的思想来源于荷兰“图形艺术家”摩里茨·科奈里斯·埃舍尔(M.C.Escher)的作品，其展示了“一个不可能的世界”，在立体空间给人一种错觉，模糊 2D 平面与 3D 立体之间的界限，不遵守物理世界的规则，这种思想突破了我们传统空间几何的认知体验，我们想要同此次项目向大家表达这种思想

三、 Goal of the Project

选取《纪念碑谷》的某一关卡（如下图）来模拟其游戏场景和游戏操作：



纪念碑谷关卡截图

基本的目标主要包括：

- 整个游戏场景的构建
- 实现物体的旋转、拖拽等动态变化
- 主人公艾达的构建以及其移动路径的构建

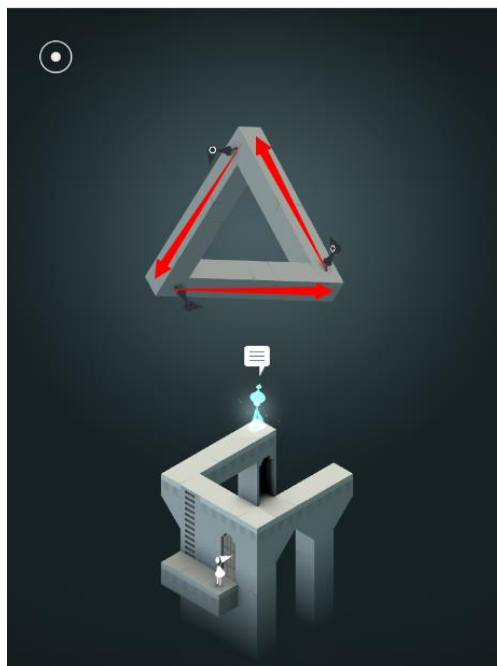
四、 Team Member and Distribution

Group 11	
1452238 杨力	主人公绘制
1452244 李松霖	主人公动画的实现
1452246 杨斌诚	寻路算法的设计实现
1452288 李文	场景构建
1452254 孔思尹	场景的变换（旋转） 阴影、光照的处理

五、 General Thought

《纪念碑谷》是如何欺骗人们的视觉的呢？

- a) 在《纪念碑谷》中运用了大量的不可能图形（impossible figure），如下图的潘洛斯三角形



纪念碑谷关卡截图

不可能图形是在现实世界中不可能客观存在的事物，只会在二维世界存在的一种图形。它是由人类的视觉系统瞬间意识地对一个二维图形的三维投射而形成的光学错觉。

b) 不可能图形只能在某个角度进行观察，如下图



潘洛斯三角形

当以图一的角度观察时，我们看到的只是三条无规律拼接的边，可是当我们把视角转向图三的时候，奇迹就发生了，三条边拼接成了一个三角形。这就是游戏《纪念碑谷》的核心思想：**See things in different angles!**

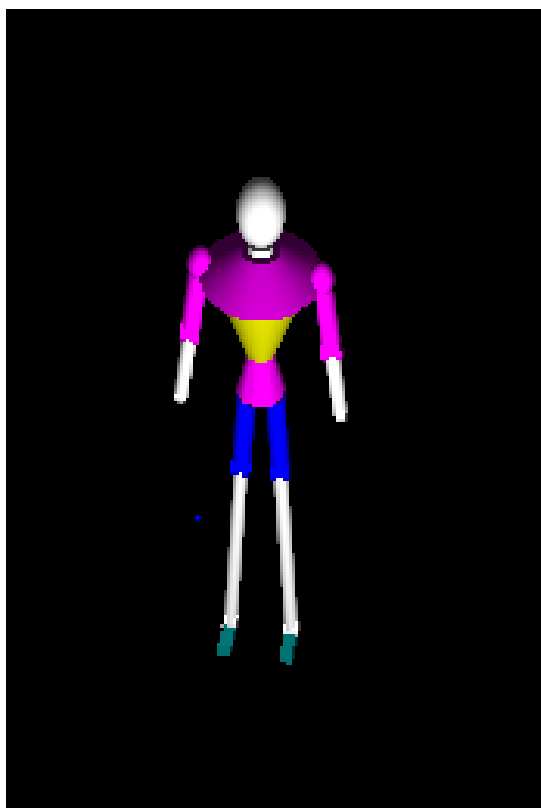
c) 因此，《纪念碑谷》其实就是通过将人们从 2D 平面看见的路径，在 3D 的游

戏场景中加以实现。人们看见的效果其实只是 3D 游戏场景的一个投影。而如果要实现游戏中这种“错视”的效果，我们必须采用平行投影，至于这样才能保证当我们旋转角度的时候，三条边能够顺利拼接起来，形成不可能图形。

六、 Implementation

➤ 人物的绘制

人物的绘制实际上就是画一个 robot：



robot

- I. robot 其实是用一系列的**基本几何体**来拼接起来的，基本几何体主要包括球体、圆柱、圆台、长方体等等，利用 opengl 的库函数 gluSphere、gluCylinder、gluDisk 等等可以实现。
- II. 在具体实现时，我们是把人物分成 head、shoulder、body、waist、arm、leg 几个部分来分别绘制的，然后定义一个整体绘制函数 drawrobot 来

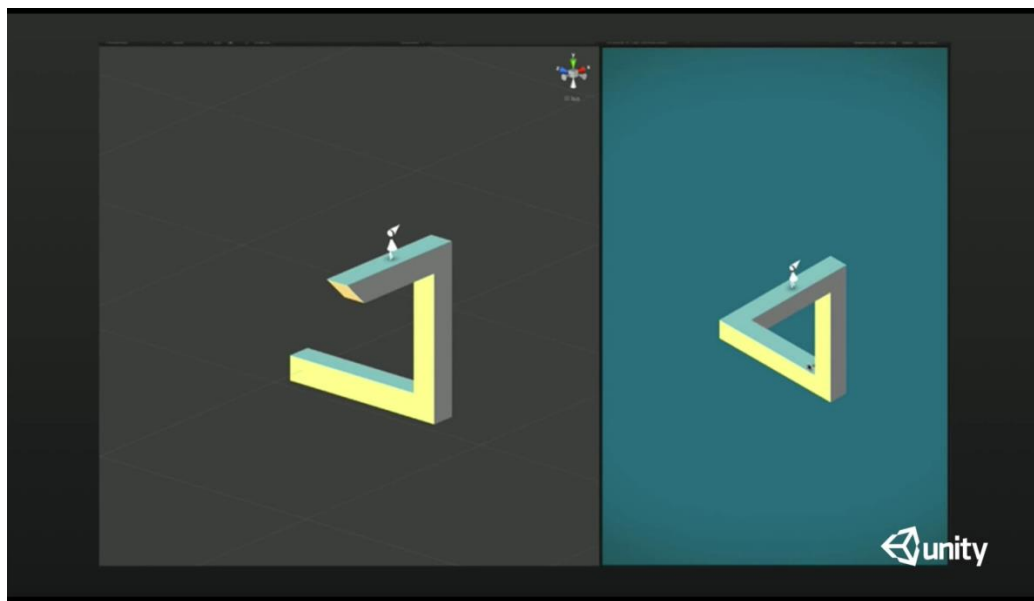
依次调用各部分，这样便实现了“局部绘制、整体运动”的目的

➤ 人物的动画的实现

通过帧的变化来实现，每一帧就是一个状态，动画实现实际上就是状态机的状态转换，我们在每个状态定义好手脚的摆动的幅度，这样人物就会经历迈步张臂——并腿并手——迈步张臂这样的状态转换。我们这里设定的人物的一步是 30 帧，而在最后一帧的时候，我们会把人物复位，变为直立站台，这样整个过程看起来就很流畅了

➤ 寻路算法的设计实现

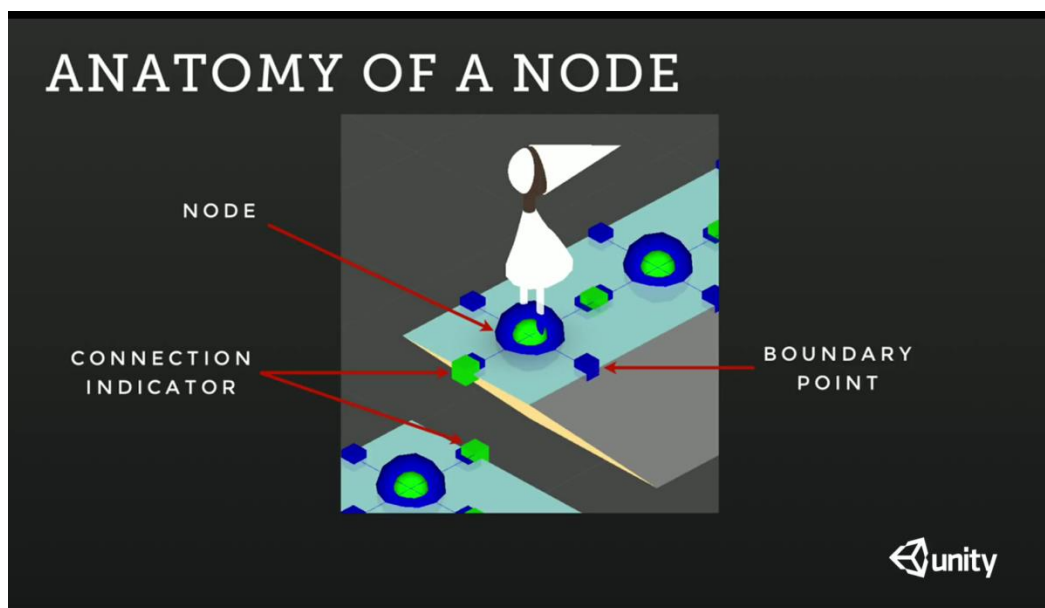
在场景中物体发生变化（选择、拖拽）之后，怎么判断主人公当前可移动的路径呢？如下图：



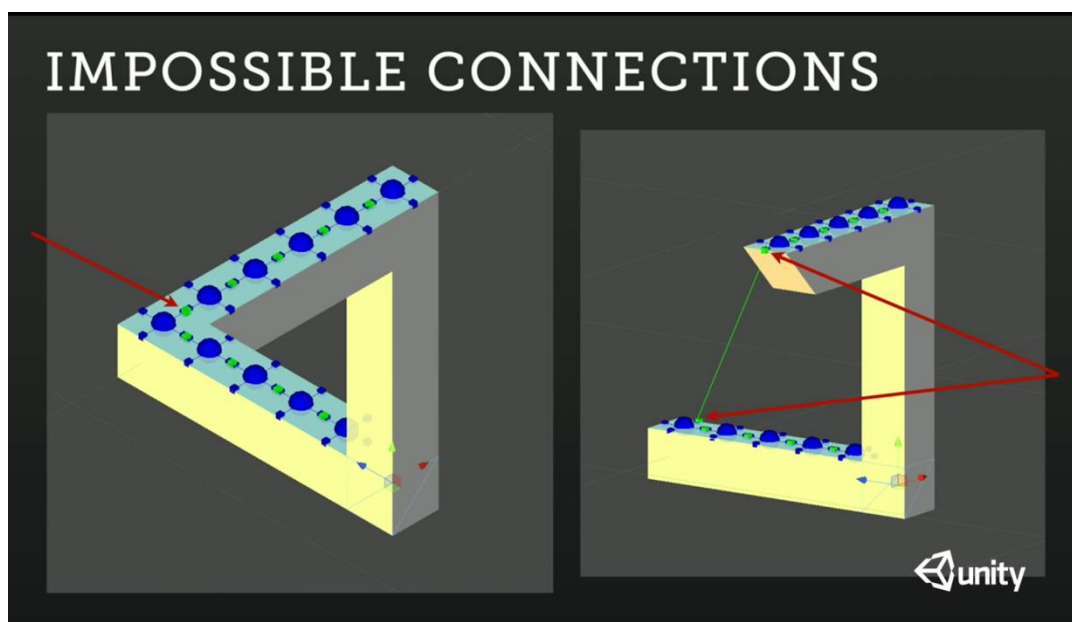
monument valley team's presentation 截图

我们在玩游戏的时候看到的效果感觉主人公是在一条路上走的（上右图中主人公可以从三角形最上面的角通过中间的角到达下方的角）。可是在 3D 的场景中两条路（上左图中根本没有中间的角）根本没有交汇，那主人公是怎么走过去的呢？

monument valley team 的技术主管 Peter 在应 unity 邀请给游戏开发者做的 presentation（文末有链接）中提到了解决办法：对路径进行标记：

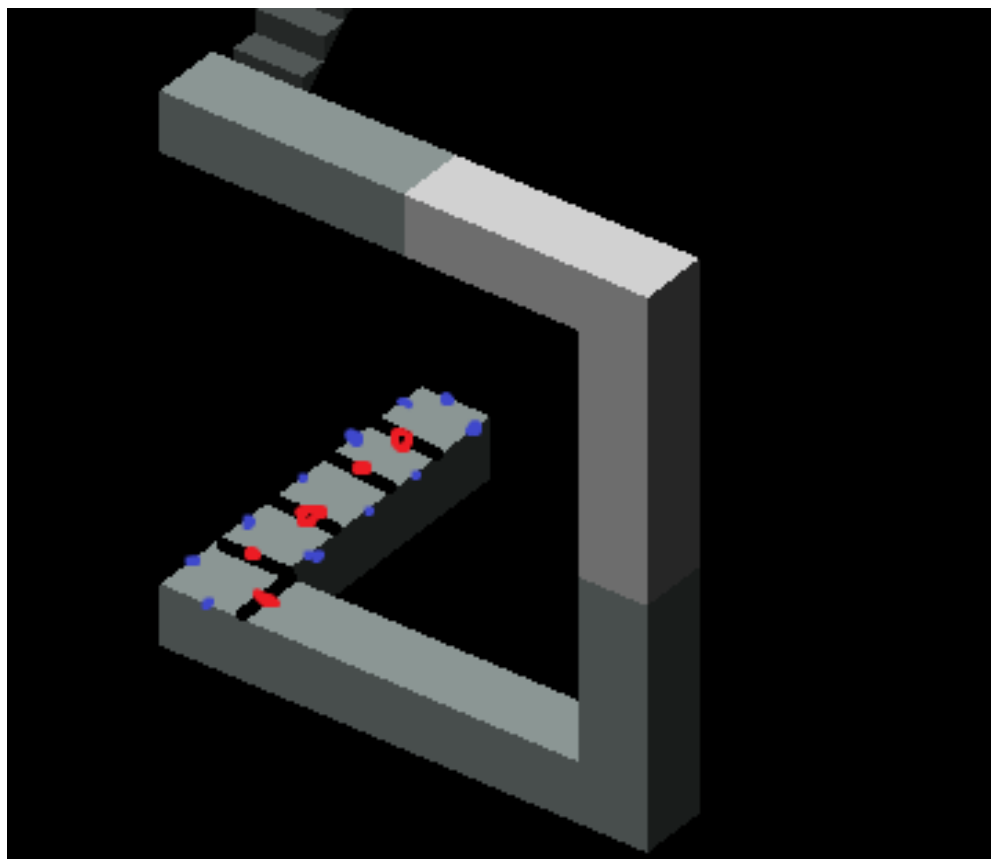


由图中我们可以看到，可供主人行走的矩形路面都用 Node 进行了标记，Node 由蓝色的 Boundary Point 和绿色的 Connection Indicator。游戏的寻路算法是基于 Node 组成的路径网络的，而通过对 Connection Indicator 进行处理，可以决定不同的 Node 之间是否相连。通过这种方法，技术人员可以轻松地根据视觉效果在 3D 场景中标出路径，下面是效果展示：



基于这个思想，我们对路径进行分割，将之分成一个个 node，每个 node 记录它当前的位置以及四个边界与其他 node 相连的情况，如下图（相连则用红

色点表示，无相连用蓝色点表示)：



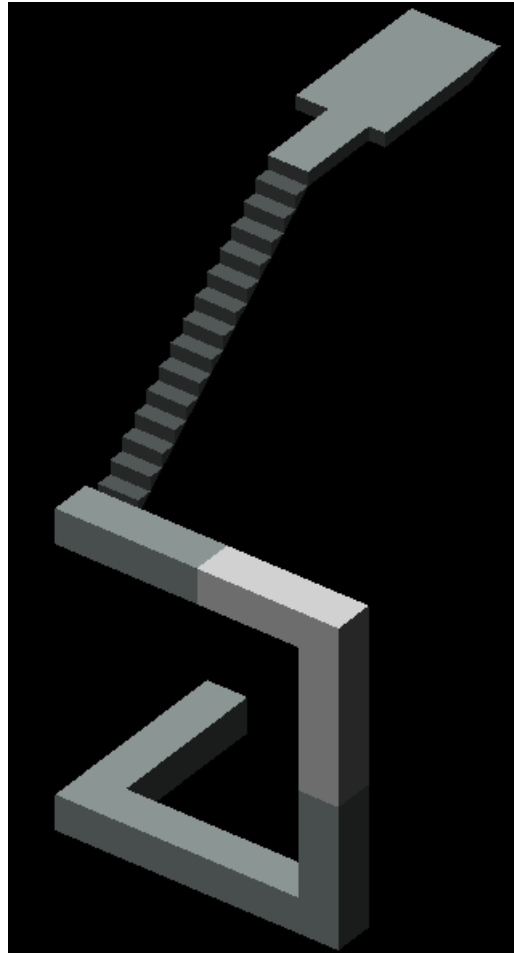
在程序中是这样定义每个 node 的：

```
struct Node{
    struct Node *left;
    struct Node *right;
    struct Node *front;
    struct Node *back;
    float pos[3];
}node[MAX_NODENUM];
```

这样，当我们场景发生变化的时候，相应修改对应 node 的指向，如此便更新了路径。主人公行走时，主需要判断当前主人公所在的 node 和下一步要走的 node 是否相连，有则允许行动，否则主人不动。

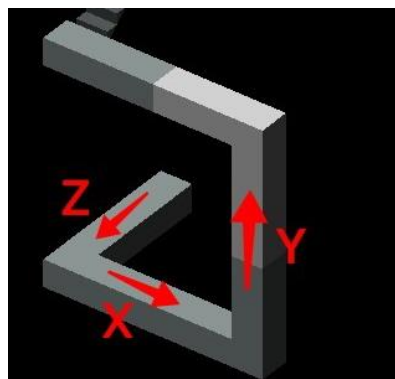
➤ 场景的绘制

场景的绘制其实也不是很难，最后的实现效果是这样的：



场景

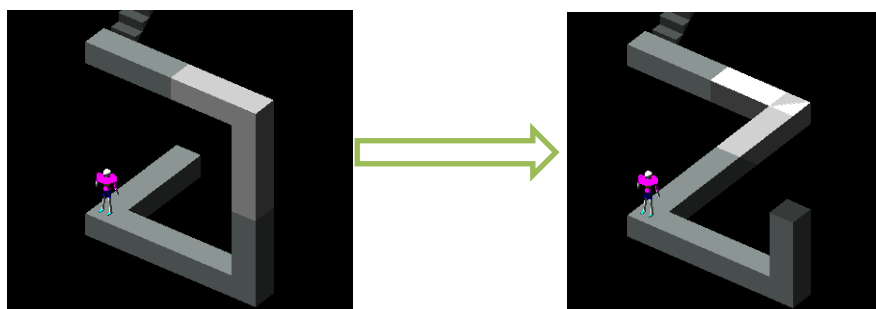
通过观察可知，我们的场景都是由一些大小不一的长方体所构成的，由于场景底部的潘洛斯三角形的三条边在空间中是相互垂直的，因此为了方便起见，我们把他们都建在与坐标轴平行的边上，如下图：



在场景的具体绘制时，比如说底部的潘洛斯三角形，我们并不是一条一条边这样去绘制长方体，而是先在 Z 走绘制一个基准长方体，而后通过 transformation（先平移后缩放）的方法绘制出其他的长方体。而在绘制楼梯的时候，我们也是从第一阶开始递归下去绘制。

➤ 场景的变化

场景的变化主要是指将场景中可旋转部分实现旋转：

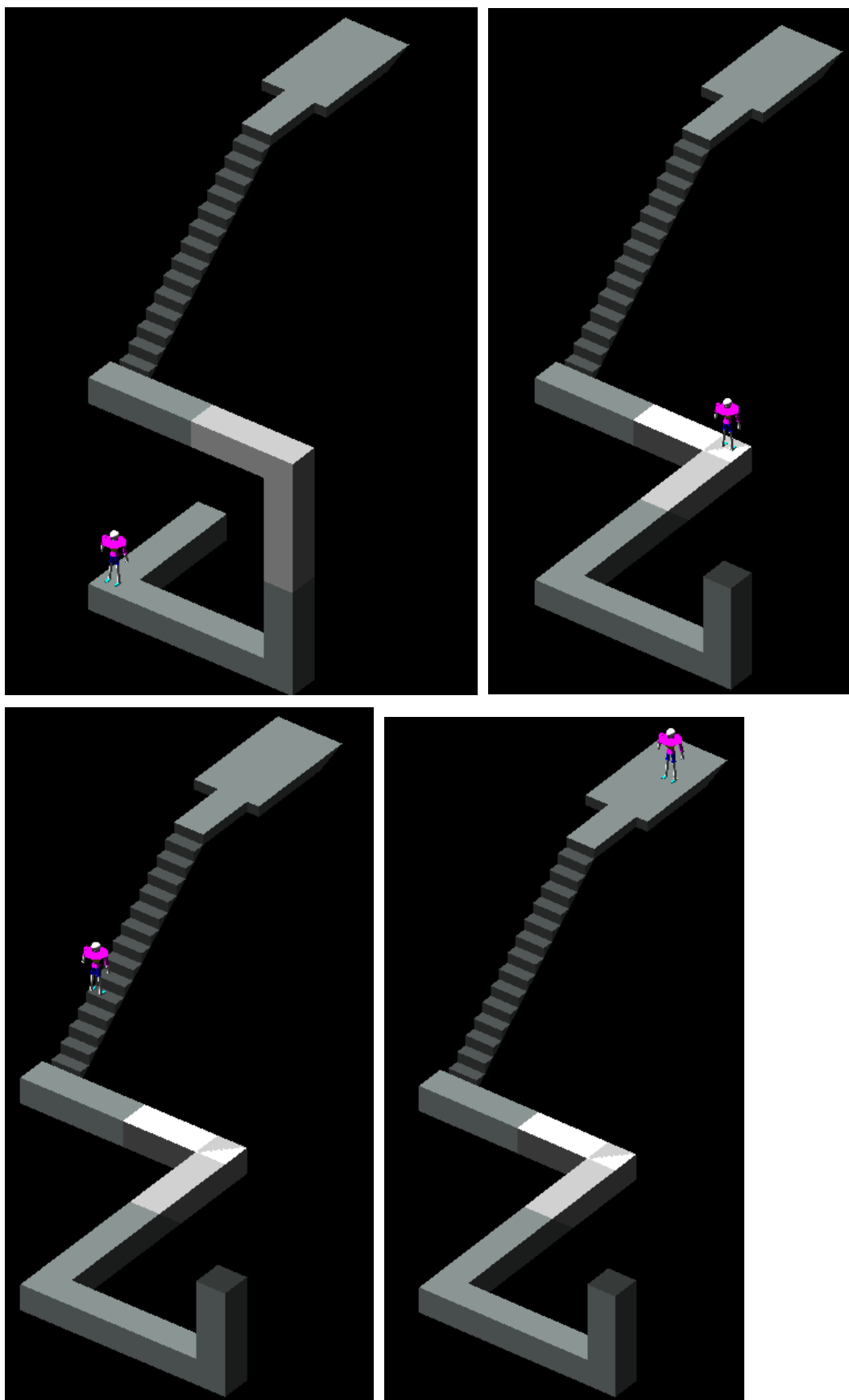


为此，我们需要：

- I. **定义投影方式和光照。**投影方式我们必须选用**平行投影**，这样保证旋转之后，两条边能够顺利地拼接起来。同时我们还需要定义光照，这样物体在旋转的过程中立体感更强，我们选用的是**点光源**，从侧边上方照向我们的场景。需要注意的是，我们在绘制长方体的各个面的时候，需要定义它们的法向量（从里向外），否则物体不会反射光。
- II. **可旋转部分着色。**可旋转部分着色应该比其他的浅一点，这样对比性更强
- III. **拼接部分微调。**在拼接部分为了看起来完全重合，需要进行一些微调（切割等）操作。

七、 Results

最后实现的效果如下：



八、References

- <http://www.zhihu.com/question/24656102> (纪念碑谷是如何欺骗人们的视觉的?)
- <http://www.zhihu.com/question/23312258> (《纪念碑谷》(Monument Valley) 游戏的空间结构是如何设计的?)
- <https://vimeo.com/89525141> (视频Monument Valley - Behind the Scenes on Vimeo)
- <http://www.tudou.com/programs/view/wmktBQ6Zeu8/> (视频ustwo at nordic game 2014: Making of monument valley in unity)
- <http://www.gameres.com/482496.html> (游戏场景设计干货:《纪念碑谷》潘洛斯视觉效果分析)