UNIVERSITÀ DEGLI STUDI DI ROMA TRE
Dipartimento di Discipline Scientifiche
Via della Vasca Navale, 84 – 00146 Roma, Italy.

Mijn meester Escher a nice appendix of a course in graphic

FLAVIO POLETTI, ANDREA VITALETTI

RT-INF-15-1996 **June 1996**

ABSTRACT

The Dutch title, in English "My teacher Escher", efficaciously summarizes the goal of this paper: to run through some typical subjects of a course in graphic again, by some works of the famous Dutch artist.

The purpose of our work is not to list the mathematical rules used in graphics but to provide some examples of how we can use them to make simple but interesting pictures.

For each subject, we first introduce Escher's picture chosen to illustrate it, then we briefly explain in natural language how we build the picture and then we show how to implement it in the powerful graphic language PLaSM.

Contents

1 Introduction

"... most of the people more easily understand a picture in an indirect way, through written words, rather than directly, through the picture itself".[5, pag. 6] With these words Escher not only wants to emphasize the importance of explaining the subject, but also the importance of understanding the graphic structure of his work; from this point of view it becomes interesting to discover the mathematical rules behind Escher's works.

Usually, in academic courses, the attention is focused on formal aspects. For students, as we are, it may be helpful and stimulating to see the problem from a more creative point of view, when possible.

We believe that this work could be a useful set of examples to improve one's understanding about the subjects of the Graphical Informatics course; it has been natural for us to choose Escher's works, for the manifest richness of topics contained therein that cover the principal subjects of the course.

Another purpose of this work is to show some of the potentialities of the functional language PLaSM, which is based on FL language, for example its conciseness: it is surprising how we can obtain great results (as in *Möebius Strip*) using a few code lines.

The first part of this work deals with plane transformations, that are rotations, translations and reflections. The goal is achieved through two examples: *Reptiles* and *Study for the regular division of the plane with human fugures.* Both figures represent a regular division of the plane; we have reached this effect using only linear planar transformations, which are modelled by their associated matrices in homogeneous coordinates. It is interesting that PLaSM gives us rotations around the origin, translations and scaling as primitives, so it is enough to give the typical parameters (as angle, length...) to obtain the desired transformation. Moreover, the STRUCT graphic primitive has been intensively used to organize the objects in logical hierarchical sets, to improve readability. Note that the use of the structure primitive is present in the whole code production, because it is powerful and rather concise.

The second part is about curves and surfaces. We use the iterated application of the 3D transformations to show how to build curves and surfaces in the space; in this way, we want to emphasize the importance of the logical process that is under the construction of such figures. The regularity of some of Escher's works has helped us to find its mathematical model with no loss of time. It is fascinating the idea of building a complex 3D object starting from a single point that at every step of the composition acquires a new degree of freedom.

The last part is perhaps the most fascinating, dealing with perspective and impossible worlds. We have chosen three examples, all representing absurd tridimensional objects. It is important to emphasise that these are only prospectic tricks, and not mathematical ones, so we had some difficulties because we had to represent tridimensional non-sense objects in a real tridimensional contest! For example, in Belvedere, the impossible figure has been represented viewing a composition of fully tridimensional parts from the unique point of view that makes the figure a non-sense. Moreover, in Ascending and descending, to fall in the perspective trick you must consider only two of the four castle walls: if we were architects looking at the plant of the castle, we would immediately realize the foolishness of the designer. The last figure, Other World I, is an example of perspective projection, but it is also a good example of using 3D transformations!

2 2D Transformations.

2.1 Introduction.

The part of this work which relates to affine transformations (in the plane, for our purposes), can be well-represented by a whole class of author's studies about cyclic regular divisions of the plane [4, pagg. 13-14].

A generic plane division is called *cyclic* if there exists a translation that makes the figure coincide with its non-translated version. In general, if you find a translation which has this property, you will also be able to find other translations like that. Take a point randomly : all its repetitions in the plane, that are the class of points in the plane that can be obtained by the application of one of permitted translations (how many times we like), will form a reticolous in the plane which is independent of the initial choosen point.

Regular divisions of the plane can also have other interesting properties, which are three in number and are called symmetries :

- Rotation : rotating the plane around the rotation points, the figure comes back to its initial shape before the rotation of the whole round angle; an example can be found in two of the generated images, respectively from *Reptiles* and *Metamorphosis I*;

- Reflection : there exist some lines in the plane which are symmetry axes for the figure, so that the figure can be folded on itself following those lines; an example of this type of symmetry can be the regular division *Three Elements*;

- Pushing reflection : there exist lines in the plane that become axes of symmetry if we push (that is, translate) one of the half planes adequately. The regular division which has only this symmetry is *Encounter*.

Initially, Escher tried by itself to solve the problem of the regular division of the plane, reaching scarce results; later, he knew that the problem had already been an object of study for mathematicians and christallographers, so the artist took their results to produce more than 150 sketches of regular plane divisions. Escher said that dividing the plane with figures which leaved no hole was ' a difficult job', but not impossible, if we consider his production in this area! Formally, in fact, there exist only 17 different divisions, but if we leave mathematics' rigidity and enter into an artist's mind we can find infinite different motifs.

To conclude, it must be told that the artist studied not only for regular plane divisions, but also for other types of divisions, like that of a sphere (see *Sphere with Fish*, *Sphere with Angels and Demons* and, last but not least, *Sphere with Human Figures*).

2.2 Reptiles (1943).

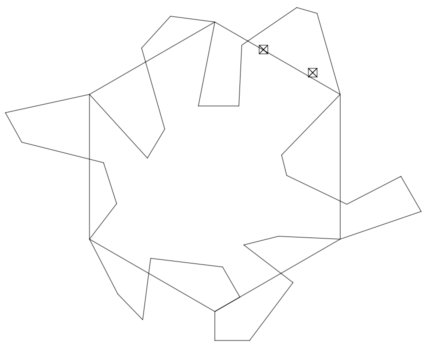


Figure 1: elmodule

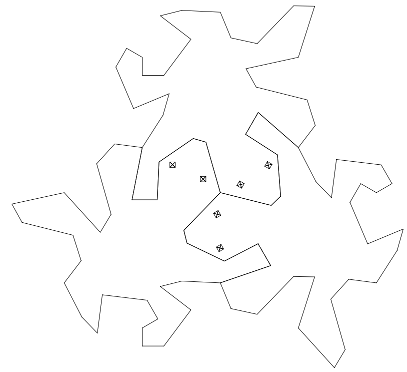


Figure 2: rotmodule

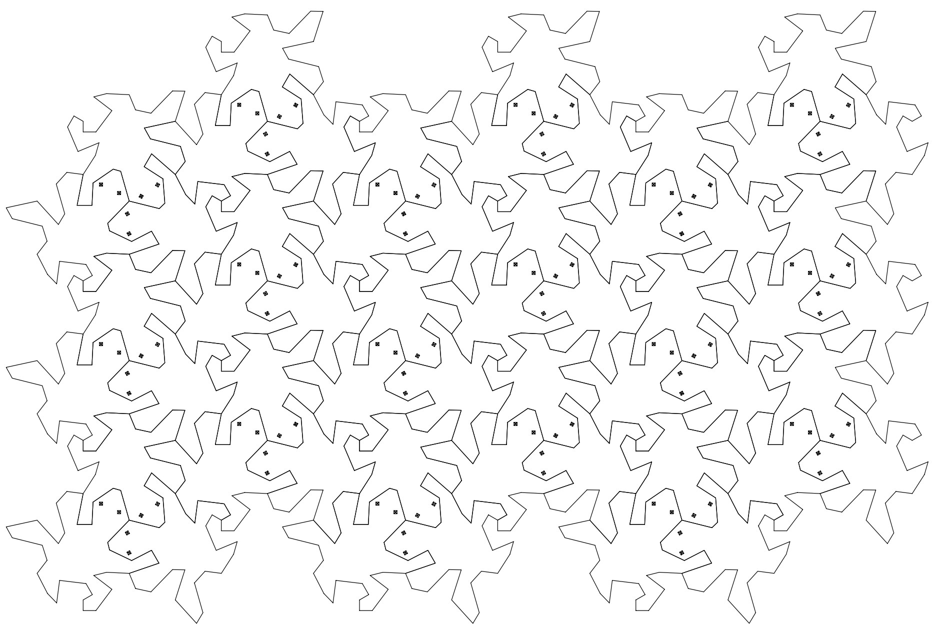


Figure 3: flooring

2.2.1 How to realize it.

Looking over Escher's picture, it is possible to draw some general rules to make similar pictures.

Look at a regular hexagon, and mark one of its' vertex as the "origin" of the picture.

It's obvious that the picture is build by a rotation of $\frac{2\pi}{3}$ and $-\frac{2\pi}{3}$ around the origin of the hexagon.(2)

Now, we must only understand how to build the picture inside the hexagon.

Let's look at the six vertexes of the hexagon and from the origin, number each vertex starting from number one.

The even points indicate rotation points; for every rotation point it's defined the polyline that has as its first point the rotation point itself, and as its last point the next odd vertex in the order.

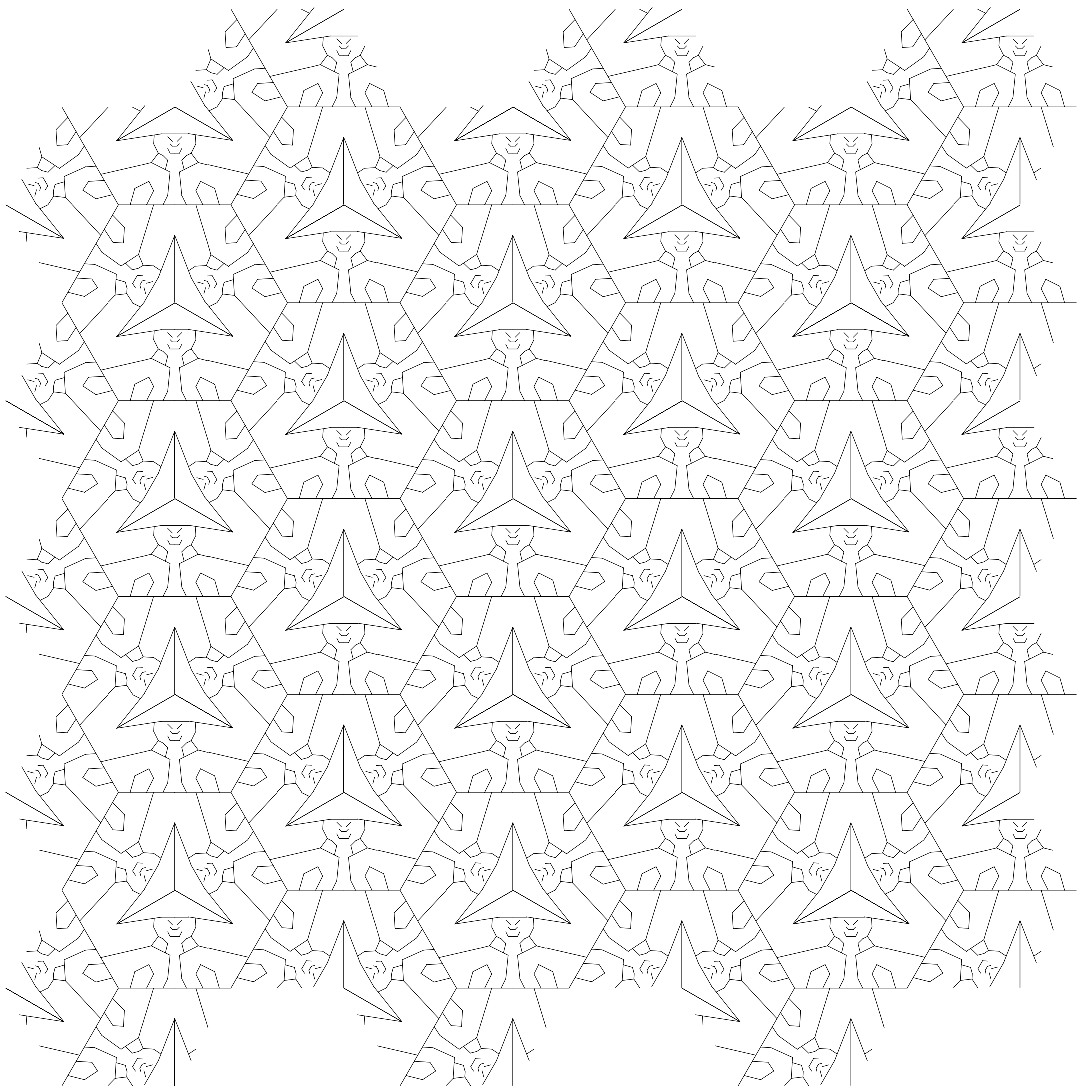
To make a consistent picture with Escher's one, it's enough to rotate every polyline of $\frac{2\pi}{3}$ around its own rotation point.(1)

It's an easy exercise to understand how to realize the flooring(3) by the traslation of rotmodule.

2.2.2 PLaSM implementation

```
DEF polyline = MKPOL~[ID,cells,pols]
           WHERE
            cells = TRANS~AA:FROMTO~[[k:1,len-k:1],[k:2,len]],
            pols = LIST~INTSTO~(len-k:1)
           END;

DEF extcoox = 50*sin:(PI/3);
DEF extcooy = 50*cos:(PI/3);
DEF l = sqrt:((extcoox*extcoox)+(extcooy*extcooy));

DEF firstpl = polyline:<<0,0>,<-:8,28>,<-:15,30>,<-:34,17>,<-:35,-:4>,<-:49,-:4>,
                        <-:extcoox,extcooy>>;
DEF secondpl = polyline:<<0,0>,<0,-:10>,<12,-:10>,<27,10>,<10,23>,<22,26>,
                         <extcoox,extcooy>>;
DEF thirdpl = polyline:<<0,0>,<20,-:22>,<26,-:12>,<18,16>,<28,27>,
                        <extcoox,extcooy>>;

DEF elemodule = STRUCT:<
        STRUCT:<T:<1,2>:<-:11,6>,cuboid:<3,3>>,
        STRUCT:<T:<1,2>:<-:28,14>,cuboid:<3,3>>,
        STRUCT:<firstpl,R:<1,2>:<2*PI/3>,firstpl>,
        STRUCT:<T:<1,2>:<-:extcoox,-:(50+extcooy)>,
                STRUCT:<secondpl,R:<1,2>:<2*PI/3>,secondpl> >,
        STRUCT:<T:1:(-:(2*extcoox)),STRUCT:<thirdpl,R:<1,2>:<-2*PI/3>,thirdpl>>
                       >;
DEF rotmodule = STRUCT:<elemodule,R:<1,2>:<2*PI/3>,
                        elemodule,R:<1,2>:<2*PI/3>,elemodule>;

DEF floor_height (n::IsIntPos) = (STRUCT~##:n):<rotmodule,T:2:(3*l)>;
DEF floor_module (n::IsIntPos) =
        STRUCT:< floor_height:n,T:<1,2>:<x,y>,floor_height:n >
          WHERE
           x=3*(l*cos:(PI/6)),
           y=l+(l*sin:(PI/6))
          END;
DEF flooring (n,m::IsIntPos) = (STRUCT~##:m):< floor_module:n,T:1:x >
          WHERE
           x=6*(l*cos:(PI/6))
          END;
```

2.3 Study for the regular division of the plane with human fugures (1936).



2.3.1 How to realize it

The regular division of the plane with the nice Chinese man who is the result in the woodcut Metamorphosis I (1937), had already been studied by Escher in 1936. If we observe the figure, it is clear that we can obtain the whole figure, starting from that of a single Chinese, by rototranslation of this 'base'. On the other side, we have preferred to start from another point of view, that is observing the structure in which the whole figure is imprisoned. This structure is a hexagonal one. Our method consists in proceeding top-down-like :

- at the first level, we observe that the figure can be obtained by translation of the base hexagon we mentioned earlier;
- at the second level, the problem is how to build the base hexagon; it can be formed by rotating around a fixed point its third part, that is a rhombus;
- at the third level it comes the rhombus' problem; as it is symmetrical, we can build it by reflection;
- at the fourth and last level we find the half-rhombus; as it has no particular property, we must build it directly.

2.3.2 PLaSM implementation

PLaSM code analysis may suggest a bottom-up approach to the problem, but it derives from the order we have given to the code itself. The construction of the half rhombus is obtained using only the primitive 'polyline' (which is defined apart using the implementation suggested during the course) : the real figure has been 'linearized' to let us need only the polyline function.

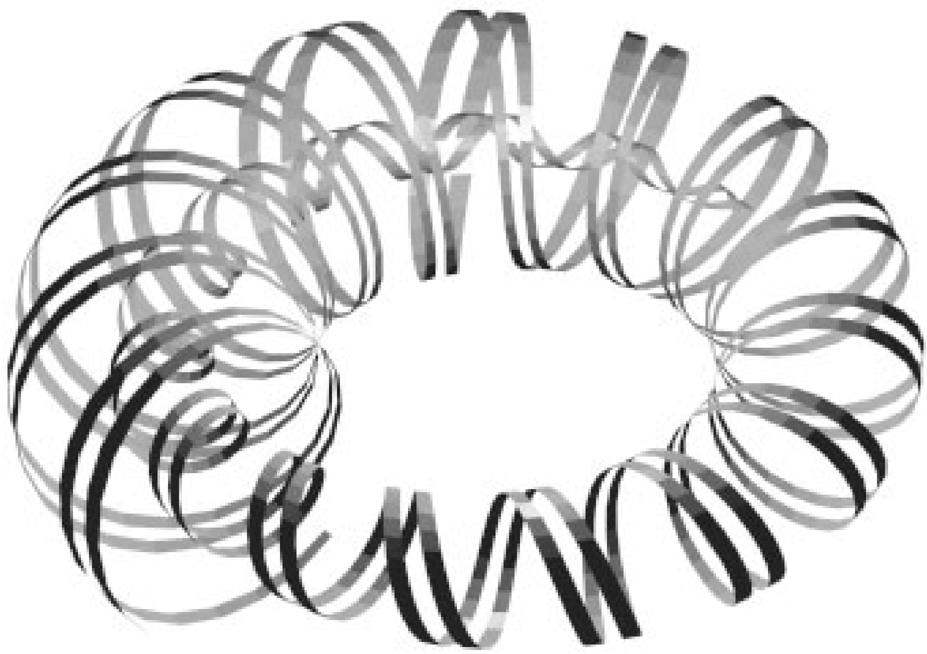
The generated figure represents only three hexagons, as they are sufficient to give an idea of Escher's way of proceeding.

```
DEF PolyLine = MKPOL~[ID,cells,pols]
        WHERE
         cells = TRANS~AA:FROMTO~[[k:1,len-k:1],[k:2,len]],
         pols  = LIST~INTSTO~(len-k:1)
        END;

DEF Trouser1 = PolyLine : <<9,43>,<14.2,26>>;
DEF Trouser2 = PolyLine : <<14,60>,<10,61>,<7,65>>;
DEF Trouser3 = PolyLine : <<9,17>,<6,15>>;
DEF Hat1  = PolyLine : <<0,73.2>,<6,73.2>>;
DEF Hat2  = PolyLine : <<0,1.5>,<0,30>,<7,12>>;
DEF Eye   = PolyLine : <<1,70>,<3,72>>;
DEF Smok  = PolyLine : <<10,61>,<2,57>,<3,47>,<5,43>>;
DEF Mouth = PolyLine : <<0,65>,<2,65>,<3,67>>;
DEF Head  = PolyLine : <<9,69>,<7,65>,<5,64>,<3,62>,<4,58>>;
DEF End   = PolyLine : <<0,43>,<24,43>>;
DEF Hand  = PolyLine : <<12,43>,<9,49>,<11,53>,<17,50>,<19,43>>;
DEF Nose  = PolyLine : <<0,68>,<1,68>,<2,69>>;

DEF HalfChinese = STRUCT : <
                     Trouser1, Trouser2, Trouser3,
                     Hat1, Hat2, Eye, Smok,
                     Mouth, Head, End, Hand, Nose
                           >;

DEF Chinese = STRUCT : < HalfCinese, S:1:(-1), HalfCinese >;
DEF Base = T:<1,2>:<24,-43>:Chinese;
```

```
DEF Base_l = R:<1,2>:(2*PI/3):Base;
DEF Base_r = R:<1,2>:(-2*PI/3):Base;
DEF Hexagon = STRUCT : < Base, Base_l, Base_r >;
DEF alias (axle,num::IsIntPos;distance::IsReal) =
         STRUCT~CONS:(AA:(T:asse:(AA:*:(DISTL:<distance,0..-:<num,1>>)));

DEF double_shift (d_x,d_y::IsReal) = STRUCT~[ID,T:<1,2>:<d_x,d_y>];
DEF my_matrix (row,column::IsIntPos) =
   (alias:<1,column,144>)~(double_shift:<72,41.7>)~(alias:<2,row,83.4>);

DEF fig = mymatrix:<5,3>:Hexagon;
```

3 Curves and surfaces.

3.1 Spirals (1953).



From 1953 to 1958, Escher worked on five pictures about spirals, the first of which is a two colour xylography, *Spirals*.

Spirals is the consequence of a challenge.

In the graphic collection of Amsterdam Rijksmuseum, Escher saw an ancient book about perspective: "La pratica della perspectiva" by Daniele Barbaro (Venice 1569). The incipit of a chapter was adorned with a torus whose surface was made of some twisted spirals. The print and the geometric shapes weren't very accurate; Escher remained irritated by this, and decided to remake not only the original picture, but also a torus, the smaller diameter of which grows from one extremity to the other, in such a way that we can imagine it to be like a snake that bites its own tail.

A spiral of the spirals, a kind of egoist as Escher used to call it later.

The problems about this project were quite difficult and they took some months of outlines. The result is a fascinating picture where the author communicates his astonishment about the pure laws of the shape. Those who would care to see all the preparatory studies for this publication, would be surely astonished by the accuracy the author put in it [1, pagg. 97-98].

3.1.1 How to realize it

- Let us consider the point of coordinates $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$
- Let us apply it to the rotation matrix around y axis

$$\begin{bmatrix} \cos u & 0 & \sin u \\ 0 & 1 & 0 \\ -\sin u & 0 & \cos u \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \sin u \\ 0 \\ \cos u \end{bmatrix} \qquad u \in [0..2\pi]$$

- Let us translate the circumference obtained on the Y axis of an amount that is the radius of the spiral $\begin{bmatrix} radius + \sin u \\ 0 \\ \cos u \end{bmatrix}$
- Let us apply a rotatioin matrix around the Z axis

$$\begin{bmatrix} \cos u & -\sin u & 0 \\ \sin u & \cos u & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} radius + \sin u \\ 0 \\ \cos u \end{bmatrix} = \begin{bmatrix} \cos u \cdot (radius + \sin u) \\ \sin u \cdot (radius + \sin u) \\ \cos u \end{bmatrix}$$

- The spiral that we have obtained isn't satisfactory for two reasons: it makes only one circle around itself and it has always the same radius.To solve these problems we use a new constant *frequency* that specifies how many circles the spiral must make around itself, and a function of u that specifies the magnitude of the radius.

$$\begin{bmatrix} \cos u \cdot (radius + u \cdot \sin frequency \cdot u) \\ \sin u \cdot (radius + u \cdot \sin frequency \cdot u) \\ u \cdot \cos frequency \cdot u \end{bmatrix}$$

- Now we have obtained "our" spiral, to obtain the others it's enough to introduce a delay factor θ

$$\begin{bmatrix} \cos u \cdot (radius + u \cdot \sin frequency \cdot u + \theta) \\ \sin u \cdot (radius + u \cdot \sin frequency \cdot u + \theta) \\ u \cdot \cos frequency \cdot u + \theta \end{bmatrix}$$

- The spirals we have obtained are curves, while Escher's spirals are surfaces.

  Let us look at two spirals:

$$\alpha(u) = \begin{bmatrix} \cos u \cdot (radius + u \cdot \sin frequency \cdot u) \\ \sin u \cdot (radius + u \cdot \sin frequency \cdot u) \\ u \cdot \cos frequency \cdot u \end{bmatrix}$$

$$\beta(u) = \begin{bmatrix} \cos u \cdot (radius + u \cdot \sin frequency \cdot u + \frac{\pi}{2}) \\ \sin u \cdot (radius + u \cdot \sin frequency \cdot u + \frac{\pi}{2}) \\ u \cdot \cos frequency \cdot u + \frac{\pi}{2} \end{bmatrix}$$

The ruled surface $X(u,v) = \alpha(u) + v \cdot \beta(u)$ exactly defines the desired surface.

3.1.2 PLaSM implementation

```
DEF Intervals ( size::IsReal ; n::IsIntPos) = QUOTE:(#:n:(size/n));
DEF Udomain = Intervals:<(3*PI),500>;
DEF Vdomain = Intervals:<1,1>;
DEF domain = T:1:(PI/4):(Udomain*Vdomain);

DEF spiral_alfa( radius,freq,offset1::IsReal ) = [fx,fy,fz]
        WHERE
         fx = cos~u * ((k:radius) + (u * sin~((k:freq * u)+k:offset1))),
         fy = sin~u * ((k:radius) + (u * sin~((k:freq * u)+k:offset1))),
         fz = u * cos~((k:freq * u)+k:offset1),
         u = s1
        END ;

DEF spiral_beta( radius,freq,offset1,offset2::IsReal) = [fx,fy,fz]
        WHERE
         fx = cos~u * ((k:radius) + (u * sin~((k:freq * u)+k:offset2)),
         fy = sin~u * ((k:radius) + (u * sin~((k:freq * u)+k:offset2))),
         fz = u * cos~((k:freq * u)+k:offset2),
         u = s1-(k:((offset2 - offset1)/freq))
        END;

DEF ruled( radius,freq,offset1,offset2::IsReal) = MAP:[fx,fy,fz]:domain
        WHERE
         fx = s1~(spiral_alfa:<radius,freq,offset1>) +
              ((s1~(spiral_beta:<radius,freq,offset1,offset2>) -
              s1~(spiral_alfa:<radius,freq,offset1>))* v),
         fy = s2~(spiral_alfa:<radius,freq,offset1>) +
              ((s2~(spiral_beta:<radius,freq,offset1,offset2>) -
              s2~(spiral_alfa:<radius,freq,offset1>)) * v),
         fz = s3~(spiral_alfa:<radius,freq,offset1>) +
              ((s3~(spiral_beta:<radius,freq,offset1,offset2>) -
              s3~(spiral_alfa:<radius,freq,offset1>)) * v),
         v = s2
        END;

DEF Spirals = STRUCT:< ruled:<20,15,0,(PI/4)>,ruled:<20,15,(PI/2),(3*PI/4)>>;
```

3.2 Möebius Strip I (1961).

In 1960, an English mathematician, whose name I have forgotten, incited me to design a composition about Möebius strips. At that time, I did not quite know what they were [1, pagg. 99-101]. Although this affirmation, even in 1946, in his coloured lithography *Cavaliers*, and then in 1956, in *Cigni*'s xilography, Escher had represented figures which had a considerable topological meaning, and which had a noticeable point of contact with Möebius strip.

The mathematician had made him observe that Möebius strip has some strange properties: for example, it could be cut along its length without dividing it in two separated parts, as it has only one face with a unique margin. Escher underlines the first property in 1961 with *Möebius Strip I* , and the second one in 1963 with *Möebius Strip II.* The definition of those stripes is due to August Ferdinand Möebius (1790-1868), who used them to demonstrate some particular topological properties.

3.2.1 How it is realized.

We have generated a model for both strips, that is the cut one and the original Möebius strip.

To obtain a Möebius strip, it is sufficient to glue the short edges of a long paper strip after having given a half torsion to it. For the PLaSM model, we have followed the same approach:

- let us consider the segment $\begin{bmatrix} 0 \\ 0 \\ v \end{bmatrix}$, where v$\in [-1,1]$;

- let's give to it a half rotation on the y axis, having $\begin{bmatrix} v \sin \frac{u}{2} \\ 0 \\ v \cos \frac{u}{2} \end{bmatrix}$, where u $\in [0, 2\pi]$;

- let's translate on the x axis, having $\begin{bmatrix} r + v \sin \frac{u}{2} \\ 0 \\ v \cos \frac{u}{2} \end{bmatrix}$;

- at last, let's give a u parameter rotation on the z axis, to obtain $\begin{bmatrix} (r + v \sin \frac{u}{2}) * \cos u \\ (r + v \sin \frac{u}{2}) * \sin u \\ v \cos \frac{u}{2} \end{bmatrix}$.

If we want to obtain the cut Möebius strip, it is sufficient for us to change the v domain:

- $v \in [-1, 1] \rightarrow v \in [-1, -0.2] \cup [0.2, 1]$.

3.2.2 PLaSM implementation

Code's understanding is straightforward:

```
DEF Intervals (size::IsReal;n::IsIntPos) = QUOTE:(#:n:(size/n));
DEF Udomain = Intervals:<(2*PI),30>;
DEF Vdomain = QUOTE : <0.8,-0.4,0.8>;
DEF domain = T:2:(-:1):(Udomain*Vdomain);

DEF Moebius (radius :: IsReal) = MAP : [fx,fy,fz] : domain
        WHERE
         fx = cos~u * ((k:radius)+(v*sin~(u/k:2))),
         fy = sin~u * ((k:radius)+(v*sin~(u/k:2))),
         fz = v * cos~(u/k:2),
         u = s1, v = s2
        END;
```

4 Projections.

4.1 Introduction.

Projections have been diffusely used by Escher to obtain various optical illusions, and allowed him to obtain various effects, as in the case of figures which are impossible from a tridimensional point of view: *M.C.Escher's well-known prints provide examples of (graphical) representation of nonsense 3-D objects. Conditions under which collections of 2-D lines correspond to nonsense polyhedra have been studied in the literature on scene analysis* [2, footnote on page 442]; because of this, sometimes it has been difficult to obtain the same Escher's effect using a mathematical model.

We have chosen three Escher's works, that are *Ascending and Descending*, *Belvedere* and *Other World I*. The first two works can be considered as 'impossible worlds', while the third is an example of 'compared worlds'.

4.2 Ascending and Descending (1960).

I recently read a book "Godel Escher Bach: an Eternal Golden Braid" by D.R. Hofstadter [3, pag. 11], in which I found an interesting interpretation of Escher's print "Klimmen en Dalen". Hofstadter said that some of Escher's prints are the best visual representation of the mathematical concept of "strange ring". A "strange ring" consists, going up or down a hierarchical system, in going back at the starting point.

Indeed, the illusion of "Up and Down"(6) hasn't been invented by Escher, but by Penrose, an English mathematician, in 1958.

In the lithography "Up and Down", we see a stairway in which we could go up or down without arriving higher or lower. We can understand the illusion if we cut the building: doors, windows, columns, all that should be on an horizontal plane is on a spiral that grows upward; we would expect the stairway to be on an inclined plane, on the contrary it is on a horizontal plane.

To show how it's possible to build a stairway on a horizontal plane, let's try to build it.

Let us look at the perspective plane ABCD and divide the segments AB, BC, CD, DA into equal parts, two for example(5). Now draw a vertical segment in each point that divides the boundary segments of the plane, and join the extremity of the vertical segments as shown in figure. The result is evident.

We have understood the trick: the stairway is on a horizontal plane, while the other elements of the castle are on a spiral that grows upward. So the anterior face of the building seems to be plausible, but if Escher should have drawn the posterior face on another sheet, we should have observed that the whole building would have collapsed(4).

4.2.1 How to realize it

The mathematical model of the castle is obtained assembling some polyhedrons to build the "flat stairway". The peculiar view obtained by the perspective, shows the desired effect.

4.2.2 PLaSM implementation

```
DEF Wall (width,height,offset::IsReal) = MKPOL:<
            <<0,0,0>,<width,0,0>,<width,width,0>,<0,width,0>,<0,0,height>,
             <width,0,height>,<width,width,height+offset>,<0,width,height+offset>>,
            <<1,2,3,4,5,6,7,8>>,
            <<1>>>;
DEF Corner (width,height,offset::IsReal) = MKPOL:<
            <<0,0,0>,<width,0,0>,<width,width,0>,<0,width,0>,<0,0,height+(offset/2)>,
             <width,0,height+(offset/2)>,<width,width,height+(offset/2)>,
                                         <0,width,height+(offset/2)>>,
            <<1,2,3,4,5,6,7,8>>,
            <<1>>>;
```

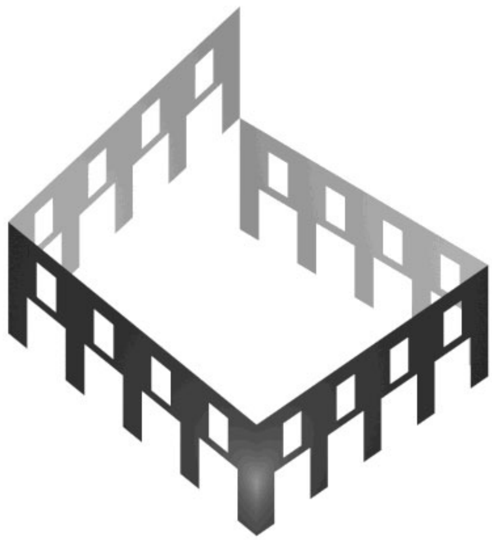


Figure 4: strange building

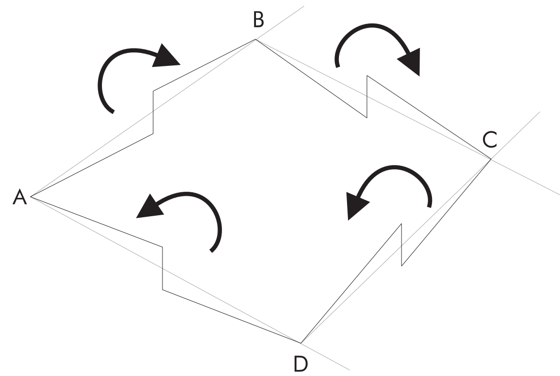


Figure 5: stairwy on a plane

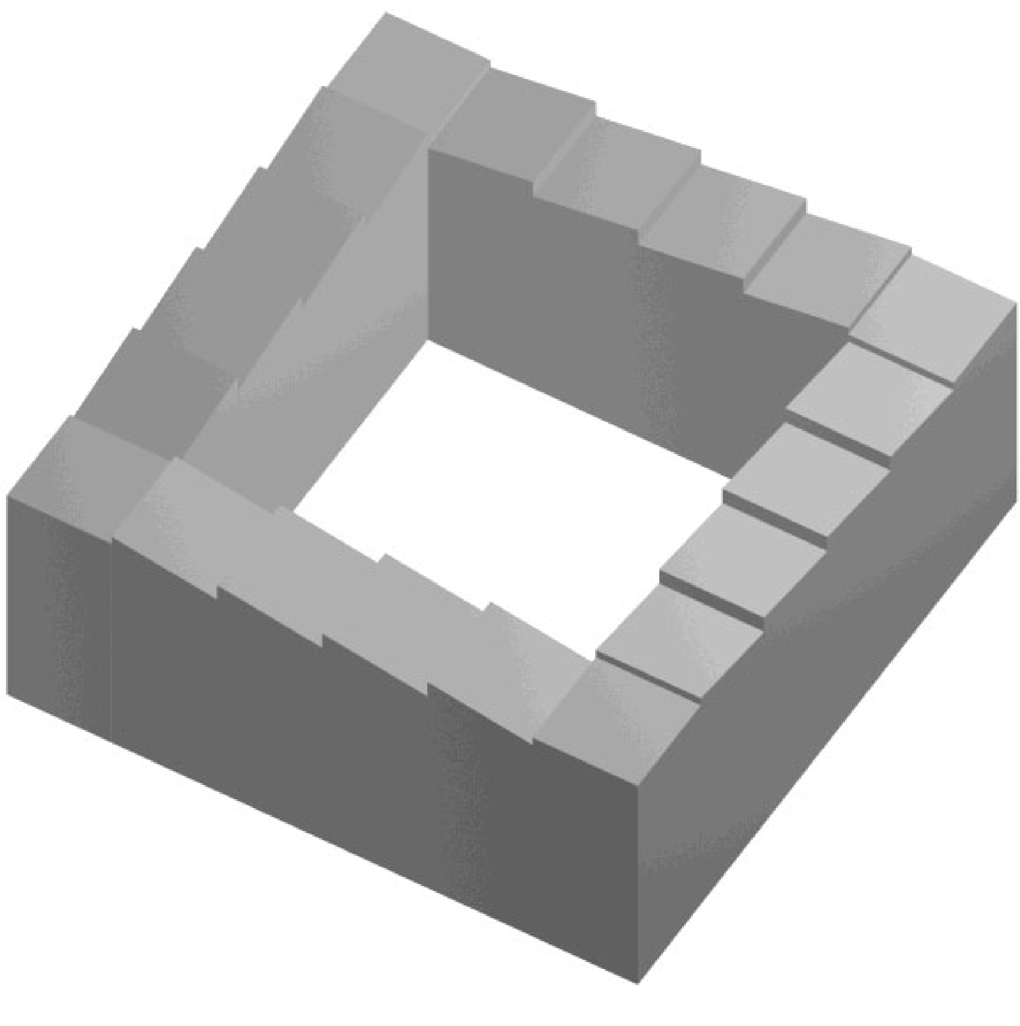


Figure 6: Up and Down

```
DEF Module (width,height,offset::IsReal;numrip::IsIntPos) =
        STRUCT:<Corner:<width,height,offset>,
                T:2:width,(STRUCT~##:numrip):<Wall:<width,height,offset>,T:2:width>>;
DEF SemiCastle (width,height,offset::IsReal;numrip::IsIntPos) =
        STRUCT:<Module:<width,height,offset,numrip>,
                T:1:((numrip+2)*width),R:<1,2>:(PI/2),
                Module:<width,height,offset,numrip>>;
DEF Castle (width,height,offset::IsReal;numrip::IsIntPos) =
        STRUCT:<SemiCastle:<width,height,offset,numrip>,
                T:<1,2>:<((numrip+2)*width),((numrip+2)*width)>,
                R:<1,2>:PI,SemiCastle:<width,height,offset,numrip>>;

DEF Module = mkpol:<<<0,6>,<0,13>,<8,13>,<8,6>,<3,7>,<3,11>,<5,11>,<5,7>,<0,0>,
                    <0,6>,<2,6>,<2,0>,<6,0>,<6,6>,<0,0>,<8,6>,<8,0>>,
                   <<1,5,8,4>,<1,5,6,2>,<2,6,7,3>,<3,7,8,4>,<9,10,11,12>,
                    <13,14,15,16>>,
                   <<1,2,3,4,5,6>>>;
DEF UnitModules(numrip::IsIntPos) =
     S:<1,2>:<1/lung,1/13>:((STRUCT~##:numrip):<Module,T:1:6>)
        WHERE
         lung = (numrip * 6) + 2
        END;
DEF Sh2d(a::IsReal) = MAT:<<1,0,0>,
                           <0,1,0>
                           <0,a,1>>;
DEF strip(x,y,sh::IsReal) =
        R:<2,3>:(PI/2):(EMBED:1:(Sh2d:sh:(S:<1,2>:<x,y>:(UnitModules:4))));
DEF SemiSpir(x,y,sh::IsReal) =
        STRUCT:<strip:<x,y,sh>,
                T:<1,3>:<x,-:(sh*x)>,R:<1,2>:(PI/2),strip:<x,y,sh>>;
DEF Spir(x,y,sh::IsReal) =
        STRUCT:<SemiSpir:<x,y,sh>,
                T:<1,2,3>:<x,x,-:(2*sh*x)>,R:<1,2>:PI,SemiSpir:<x,y,sh>>;
```

4.3 Belvedere (1958).

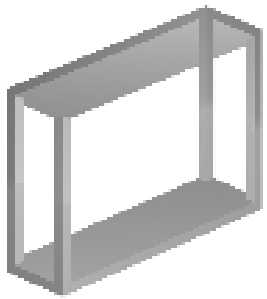


Figure 7: point of view

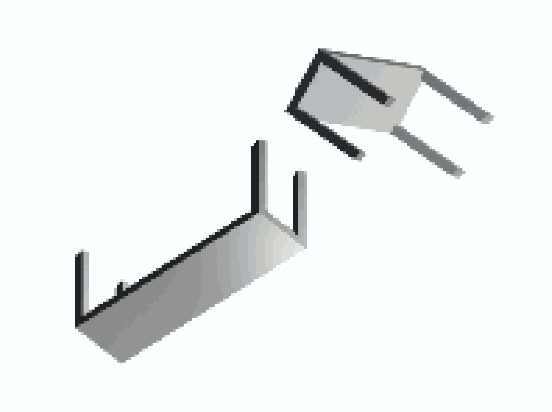


Figure 8: two parts

The original name for this lithography was 'The house of ghosts', which was later converted in the definitive one (*Belvedere*) for this work is not so ghostly, even if it gives some anxious impression when we look at it. It is a good example of impossible architecture, and for this reason it can be associated to the lithography *Waterfall.*

The principal trick in *Belvedere* consists in its intrinsic bidimensionality. In fact, looking at every figure that could have some tridimensional interpretation in some part, we are accustomed to force this impression to the whole figure, even to parts that are not tridimensional.

With this, we dare not to say that it is impossible to build tridimensional objects whose projection leads to the effect we are speaking about, but simply that the meaning we are lead to give to the final picture is not its real one, but something impossible.

Our way of representing *Belvedere* is an example: it consists of an object, formed by two parts(8), which has nothing interesting for itself, but that assumes a very particular meaning if it is seen from a certain point of view(7). Actually, our way of building a model for *Belvedere* is not the same as that followed by Escher : in fact, he draw this work observing that the global impossible effect could be reached by projecting the upper part using some view parameters, and the lower using other dual parameters. Following this way, it would be impossible for us to represent it by the PLaSM language, for the program treats with tridimensional objects, and Minerva viewer does not allow us to project two objects in the same scene with different view parameters.

The real trick used by Escher is suggested by the author himself in two ways:

- it is revealed by the young man sitting close to the prison, for the object he has in his hands is a simplified model of the whole picture;
- it is revealed (maybe to the young man himself!) by the paper lying on the floor, at young man's feet.

It is not strange for us the perplexity on his face: living in our tridimensional world, it would be rather difficult for him even to hold that curious object in his hands!

Our version of *Belvedere* is a simplified one, and does not represent objects that, in the real picture, are intended to underline the figure's ambiguity. A clear example is the ladder from the lower floor to the upper one: it starts remaining in the inside, and arrives to the upper floor standing in the outside! The effect is also put in evidence by the two persons standing on the right looking at the mountains: they are looking through the same couple of columns, but they are actually orthogonal!

4.3.1 How it is realized.

Escher's real effect is clear if we cut the figure in two parts with a horizontal line. The two halves are tridimensionally coherent, but it is also clear that they have been obtained by being projected in two different view systems that are impossible for us TO COMPREHEND.

Our approach to obtain *Belvedere*'s stylizazion has been different from Escher's one, even if the result is similar; we simply asked : " Does it exists a tridimensional object such that it leads to the desired effect if observed from the right point of view? The answer had to be affirmative:

- Dr. Cochran had succeeded in taking a photo of such an object;
- joking apart, it must be observed that any projection is only a sequence of transformations; so, all we need to do is to apply to one of the two halves a transformation, so that compounding it with the prospectic transformation of the other half, the final result was the right prospectic transformation. In fact, if we say
  - RHT the right half transformation
  - OHO the other half transformation
  - M the transformation we are looking for

  it holds that

  $$RHT = OHO \circ M$$

  and so, as we consider projections without eliminating the third dimension

  $$M = OHO^{-1} \circ RHT$$

  and the whole property is nothing but the associative property of transformations.

At a theorical level, the problem was solved, but it remained the practical problem of calculating the M matrix. Does it exist a simple way to reach our goal without the explicit calculus of the M matrix? If we observe the simplified model, it is clear that the two halves are the same if we turn one of them by an angle of 180°.That is, if we cut the paper, hold one half on the table and turn the other one, the two drawings we see are quite similar. The important thing to notice is that such a rotation is done taking the direction of projection as rotation axis, so we can skip M's matrix calculus to obtain the final effect by a simple rotation.

Another problem was that the cut we talked about on a bidimensionl paper would result in a cut on a plane that is not parallel to the floor's plane; the result was that it was impossible for us to use a simple table to simulate one of the halves, and we designed the table's legs ad hoc. The base of every leg had to be done so that its image points were all at the same y coordinate.

At last, we had to choose the adequate point of view. Actually, this problem has been resolved before the last one, for it is necessary for the construction of the leg to know the exact view parameters. At the point where the two images have to be fused, Escher had more freedom than us, as he treated with circular columns where we use square-section elements; so, we had to choose the x coordinate in DOP equal to the y one. The height has been arbitrarily set to $\frac{2}{3}$.

To conclude, we have also chosen to use a parallel type projection, that is, an oblique axonometry with the view plane intersecating the three principal axes at the coordinate 1. On the viewer, we have to set the following parameters:

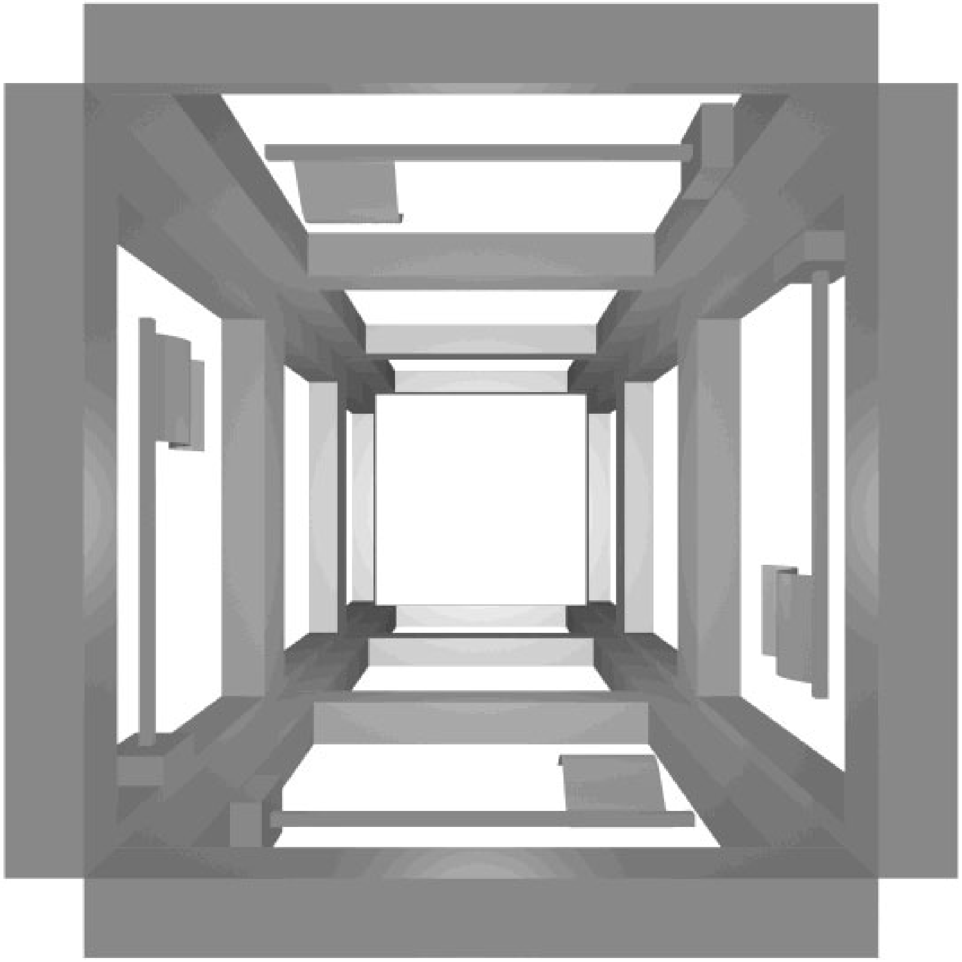
- Projection type : parallel
- $VRP = [0, 0, 0]$
- $VUP = [0, 0, 1]$
- $VPN = [1, 1, 1]$
- $PRP = DOP = [1, 1, 2/3]$.

4.3.2 PLaSM implementation

```
% BELVEDERE %

DEF IsVect = IsSeqOf:IsReal;
DEF IsMat = AND~[IsSeqOf:IsVect, EQ~AA:LEN];
DEF ScalarVectorProd (a::IsReal;v::IsVect) = (AA:*~DISTL):<a,v>;
DEF VectorModule = SQRT~+~AA:sqr;
DEF sqr = ID*ID;
DEF VectorDiff = AA:-~TRANS;
DEF VectorSum = AA:+~TRANS;
DEF UnitVector = ScalarVectorProd~[k:1/VectorModule,ID];
```

```
DEF VectorProd (u,v::IsVect)=<w1,w2,w3>
          WHERE
           u1=s1:u,u2=s2:u,u3=s3:u,
           v1=s1:v,v2=s2:v,v3=s3:v,
           w1=(u2*v3)-(u3*v2),
           w2=(u3*v1)-(u1*v3),
           w3=(u1*v2)-(u2*v1)
          END;
DEF ScalarProd = +~AA:*~TRANS;
DEF MatHomogenize (m::IsMat) = AL:<prima,(AA:AL~DISTL):<0,m>>
          WHERE
           prima = AL:<1,#:(LEN:m):0>
          END;
DEF Rotation (axis :: IsVect; angle :: IsReal) =
        ((MAT~TRANS):M)~(R:<1,2>:angle)~(MAT:M)
          WHERE
           M = MatHomogenize:<mx,my,mz>,
           mz = UnitVector:axis,
           my = mz VectorProd mx,
           mx = UnitVector:(<0,0,1> VectorProd axis)
          END;

DEF Plane = CUBOID:<25,7,1>;
DEF cut_leg (h::IsReal) = MKPOL:<Vertex, Cells, Pols >
          WHERE
           Vertex = <<0,0,0>,<1,0,0>,<1,1,0>,<0,1,0>,
                      <0,0,h>,<1,0,h+(1/3)>,<1,1,h+(2/3)>,<0,1,h+(1/3)>>,
           Cels = <<1,2,3,4,5,6,7,8>>,
           Pols = <<1>>
          END;
DEF leg (x,y,h::IsReal) = (T:<1,2,3>:<x,y,1>):(cut_leg:h);


DEF Belvedere_down (h::IsReal) = STRUCT:<
                                          Plane,
                                          leg:<0,0,h>,
                                          leg:<24,0,h>,
                                          leg:<0,6,h>,
                                          leg:<24,6,h>>;
DEF Belvedere_up (h::IsReal) =
        Rotation:<<1,1,2/3>,PI>:(Belvedere_down:h);

DEF Belvedere = STRUCT : < Belvedere_down:8,
                           T:<1,2,3>:<23.5,5.5,17.242424>,
                           Belvedere_up:8 >;
```

4.4 Other World (1946).



Escher drew two versions of *Other World* , one of which in 1946, and the other one year later, in 1947. Although he drew the second version for the reason that the first had not satisfied him at all (Other World I, in fact, remained a Mezzotint), we have decided to represent the former version, for it is more convenient for our goals. Therefore, 1947 version is not very coherent, as in two parts the arch is in front of the Simurgh (that strange anthropocephalus bird which smiles in both versions was a gift of the author's father-in-law brought from Baku, in Russia), while in the third it is on the other side of the bird. The principal defect of the first version is the necessity of using four visions to represent three different landscapes, and that the infinite tunnel fades in obscurity with the vanishing point.

It is the vanishing point itself which is the real star of the whole scene in the 1946 version, much more than 1947 one. In fact, the scene can be divided into four parts, each edged by the two diagonals of the drawing, representing the same scene taken by different points of view, that is different vanishing points. Those four vanishing points are then obliged to coincide putting the four figures together.

Let us imagine the space being divided in two parts: in one half, completely empty, we stand as observers; on the other half there is the whole universe, and, in particular, the moon promenade that Escher has chosen to represent. To separate the two worlds, a regular grid of arches. Under one of those arches there is the Simurgh, with its ironic smile. Now, if we observe it from four different points of view, holding view up vector parallel to Simurgh's height, the final effect is that of *Other World I*, when we oblige the four vanishing points to coincide (Nadir for the upper part, Zenith for the lower one, and those on the axis which is orthogonal to the separation plane). The final composition is astonishing, as it seems that every Simurgh is looking at what is happening in the... other world.

4.4.1 How it is realized

In our representation, we have substituted arches with square cells (modules), Simurgh with a flag and eliminated the oil lamp on the top of the arch. Constructive proceeding follows the line of reasonment we have just talked about: for every view we have isolated that portion of the world which would be visible, and the global structure has been obtained through rototranslations.

View parameters setting needed a prospectic type projection, as it is evident looking at Escher's work; our unique vanishing point will be that of the x axis. The view parameters are then:

- Projection type : perspective
- VRP = [20, 0, 0]
- VUP = [0, 0, 1]
- VPN = [18, 0, 0]
- PRP = COP = [18, 0, 0] (automatic in Minerva)
- Angle : 180°.

4.4.2 PLaSM implementation

```
% OTHER WORLD %

DEF partition (i::IsReal;n::IsIntPos) = QUOTE:(#:n:(i/n));
DEF Udomain = partition:<1,20>;
DEF Vdomain = partition:<1.8,1>;
DEF domain = Udomain*Vdomain;
DEF VectorSum = AA:+~TRANS;
DEF ScalarVectorProd (a::IsFun;v::IsVect) = (AA:*~DISTL):<a,v>;
DEF IsVect = IsSeqOf:IsFun;
```

```
DEF Bezier3D (q1,q2,q3,q4::IsSeqOf:IsReal) =
      <(x:q1*b1)+(x:q2*b2)+(x:q3*b3)+(x:q4*b4),
       (y:q1*b1)+(y:q2*b2)+(y:q3*b3)+(y:q4*b4),k:0 >
           WHERE
            x = k~s1,
            y = k~s2,
            b1 = (k:1-u)*(k:1-u)*(k:1-u),
            b2 = (k:3)*u*(k:1-u)*(k:1-u),
            b3 = (k:3)*u*u*(k:1-u),
            b4 = u*u*u,
            u = s1
           END;
DEF alpha = Bezier3D:<<1,1,0>,<-1,1,0>,<1,-1,0>,<-1,-1,0>>;
DEF beta =  <k:0,k:0,s2>;

DEF Flag_base = CUBOID:<1.4,1,0.5>;
DEF Stick = CUBOID : <0.2,0.2,7.5>;
DEF Bezier = (T:<1,2,3>:<-1.4,0.1,5.5>~R:<1,2>:(3*PI/4)):
                     (MAP:(CONS:(alpha VectorSum beta)) : domain);
DEF Flag = STRUCT : <Flag_base,T:<1,2,3>:<0.6,0.4,0.5>,Stick,Bezier >;
DEF Upper = CUBOID:<10,1,1>;
DEF Lateral = CUBOID:<1,1,8.7>;
DEF Base_Module = CUBOID:<10,1,0.3>;
DEF Module = STRUCT:< Base_Module, T:3:0.3 ,
                      T:1:9:Lateral,Lateral,T:3:8.7:Upper >;
DEF Module_and_Flag = STRUCT:<Module, T:<1,3>:<6.6,0.3>,Flag >;

DEF Horizontal_right =STRUCT:<T:2:11,Module,T:1:10,Module,
                              T:<1,2>:<20,1>~R:<1,2>:PI,Module_and_Flag>;
DEF Horizontal_left =STRUCT:<Module, T:1:10, Module, T:1:10, Module_and_Flag>;
DEF Vertical_superior =STRUCT:<T:<2,3>:<11,11>~R:<1,2>:(-:PI/2)~R:<2,3>:(-:PI/2),
                               Module, T:3:10, Module, T:3:10, Module_and_Flag>;
DEF Vertical_inferior =
       STRUCT:<T:<1,2,3>:<30,11,-1>~R:<1,2>:(-:PI/2)~R:<2,3>:(PI/2),
               Module_and_Flag, T:3:10, Module, T:3:10, Module>;
DEF OtherWorld = STRUCT:< T:<2,3>:<-6,-5>,
                          Horizontal_left,
                          Horizontal_right,
                          Vertical_superior,
                          Vertical_inferior >;
```

5 Conclusions

It is a peaty that people like Escher didn't have the opportunity to use powerful instruments like modern computers and programming languages, because with these instruments there are no limits on the objects that fantasy can create.

We believe that we have shown how simple and pleasant is playing with mathematics especially having a valid language like PLaSM to implement our experiments; we encourage the students to improve their own knowledge by programming because it is sure that the direct experience is the most valid support to the understanding of the course topics.

References

[1] Bruno Ernst, *Lo specchio magico di M.C.Escher*, TASCHEN 1992

[2] Aristides A.G. Requicha, *Representation for rigid solids: theory, methods and systems*, Computing Surveys, Vol. 12 N. 4 Dec. 90

[3] Douglas R. Hofstadter, *Goedel Escher Bach: un'eterna ghirlanda brillante*, Adelphi 1990

[4] D.Schattschneider e W.Walker, *M.C.Escher caleidocicli*, TASCHEN 1990

[5] M.C.Escher, *Grafica e disegni*, TASCHEN 1992

[6] Paoluzzi A., Pascucci V., Vicentino M., *Geometric Programming: A Programming Approach to Geometric Design*, acmTransactions on Graphics,Volume 14, Number 3, July 1995, 266-306

[7] Paoluzzi A.,Sansoni C. 1992 *Programming language for solid variational geometry* Comput. Aided Des. 24, 7, 349-366

[8] Backus J., Williams J. H., Wimmers E. L. 1990 *An introduction to the programming language FL*, Research Topics in Functional Programming, D.A. Turner, Ed, Addison Wesley, Reading, Mass.

[9] Foley J.D., Van Dam A., Feiner S. K., Hughes J.F. 1990 *Computer Graphics Principles and Practice*, Second Ed. Addison-Wesley, Reading, Mass.

Some books where you can find a selection of Escher's pictures :
Bruno Ernst, *Lo specchio magico di M.C.Escher*, TASCHEN 1992
Vari, *The world of M.C.Escher*, Abradale press 1988
M.C.Escher, *Grafica e disegni*, TASCHEN 1992