Computer Graphics
&
Multimedia Techniques

Unit-6

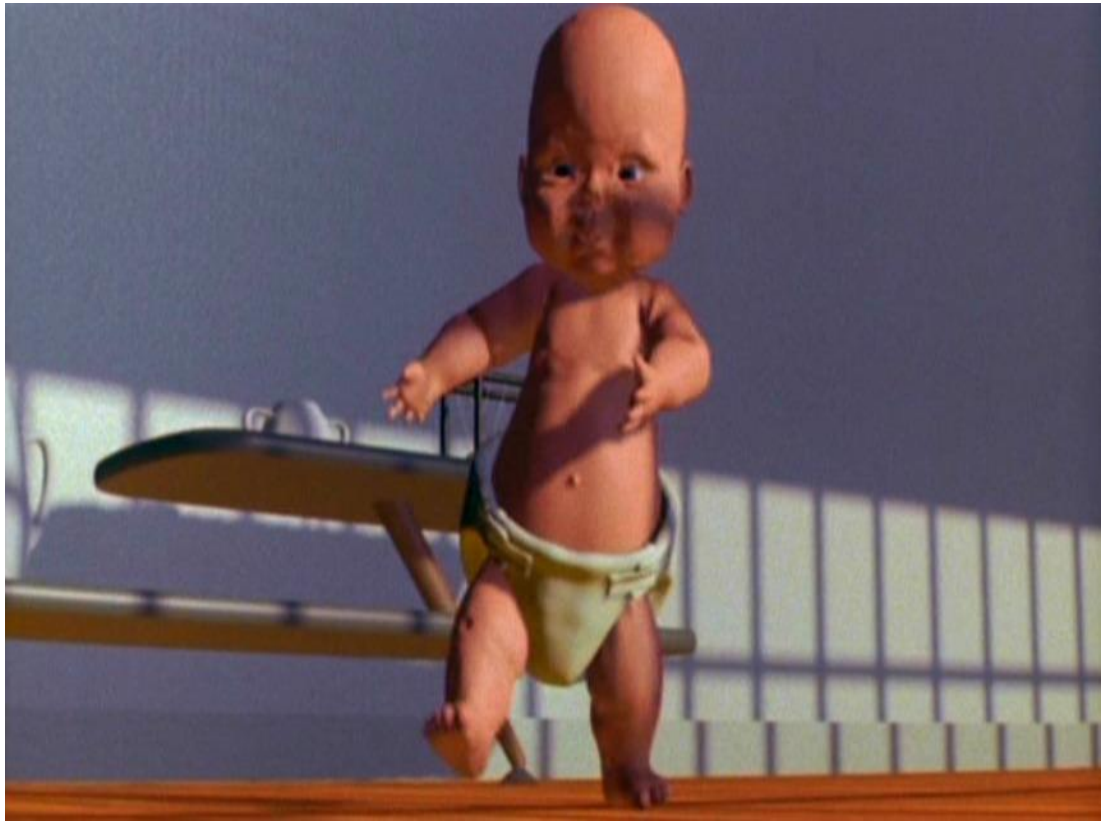
Animations & Realism

Design of Animation sequences:

In general, an animation sequence is designed with the following steps:

- Storyboard layout
- Object definitions
- Key-frame specifications
- Generation of in-between frames

This standard approach for animated cartoons is applied to other animation applications as well, although there *are* many special application that do not follow this sequence. Real-time computer animations produced by Bight simulators, for instance, display motion sequences in response to *settings* on the aircraft controls and visualization applications *are* generated by the solutions of the numerical models. For *frame-by-frame* animation, each frame of the scene is separately generated and stored. Later, the frames can be recoded on film or they can be consecutively displayed in "real-time playback" mode.



Animation sequences:

Storyboard Layout: The *storyboard* is an outline of the action. It defines the motion sequence as a set of basic events that *are* to take place. Depending on the type of animation to be produced, the storyboard could consist of a set of rough sketches or it could be a list of the basic ideas for the motion.

An Object Definition: An *object definition* is given for each participant in the action. Objects can be defined in terms of basic shapes, such as polygons or splines. In addition, the associated movements for each object are speeded along with the shape.

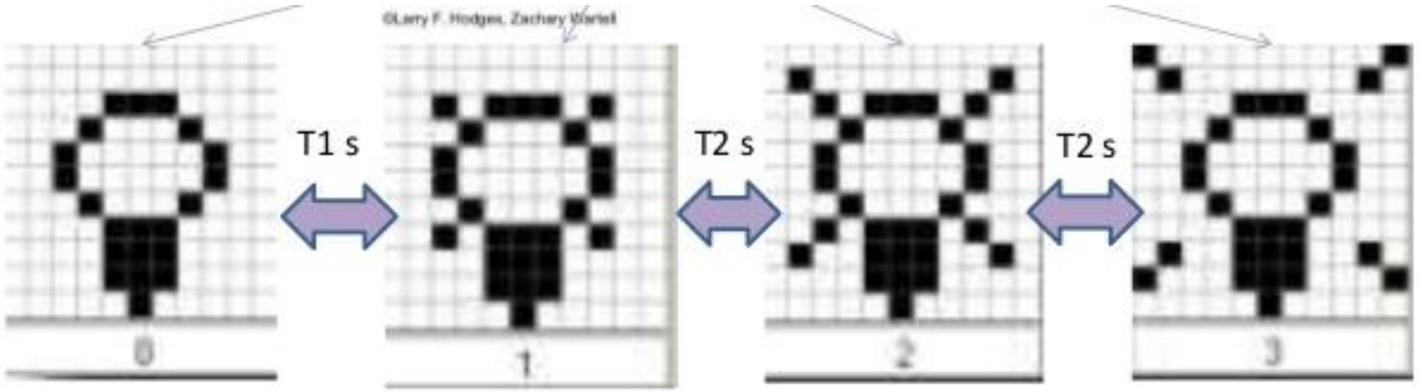
Key-Frame Specification: A key-frame is a detailed drawing of the scene at a certain time in the animation sequence. Within each key frame, each object is positioned according to the time for that frame. Some key frames are chosen at extreme positions in the action; others are spaced so that the time interval between key frames is not *too* great. More key frames are specified for intricate motions than for simple, slowly varying motions.

Animation sequences:

In-between frames (tweening): In-between are the intermediate frames between the key frames. The number of in-betweens needed is determined by the media to be used to display the animation. Film requires 24 frames per second, and graphics terminals are refreshed at the rate of 30 to 60 frames per second. Typically, time intervals for the motion are set up so that there are from three to five in-betweens for each pair of key frames. Depending on the speed specified for the motion, some key frames can be duplicated. For a 1 minute film sequence with no duplication, we would need 1440 frames. With five in-betweens for each pair of key frames, we would need 288 key frames. If the motion is not too complicated, we could space the key frames a little farther apart. eat. More key frames are specified for intricate motions than for simple, slowly varying motions.

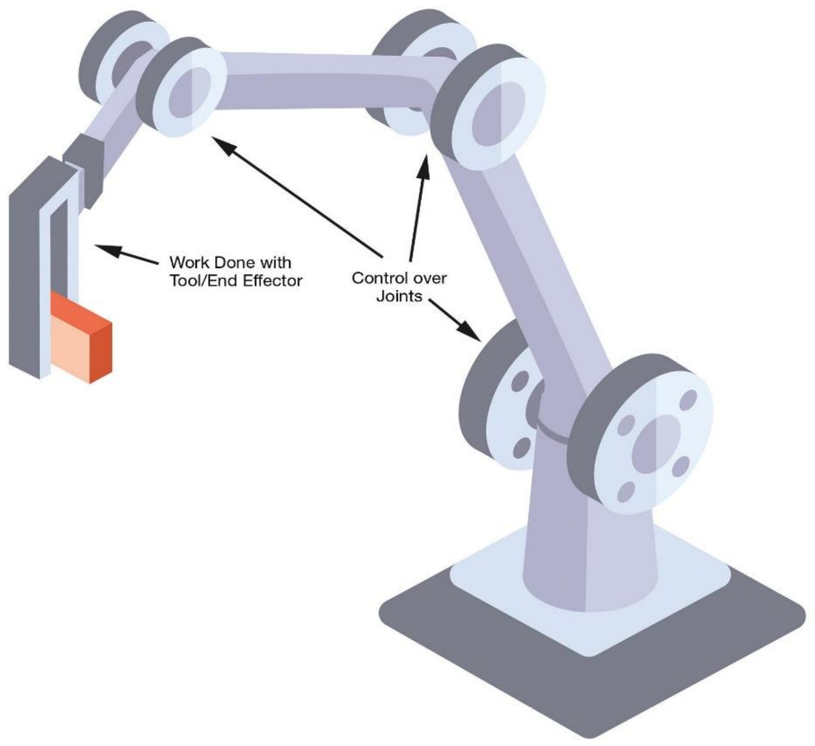
Raster Animation:

On raster systems, we can generate real-time animation in limited applications using *raster operations. A* simple method for translation in the **xy** plane is to transfer a rectangular block of pixel values from one location to another. Two dimensional rotations in multiples of *90"* are also simple to perform, although we can rotate rectangular blocks of pixels through arbitrary angles using antialiasing procedures. To rotate a block of pixels, we need to determine the percent of area coverage for those pixels that overlap the rotated block. Sequences of raster operations can be executed to produce real-time animation of either two-dimensional or three-dimensional objects, as long as we restrict the animation to motions in the projection plane. Then no viewing or visible-surface algorithms need be invoked.



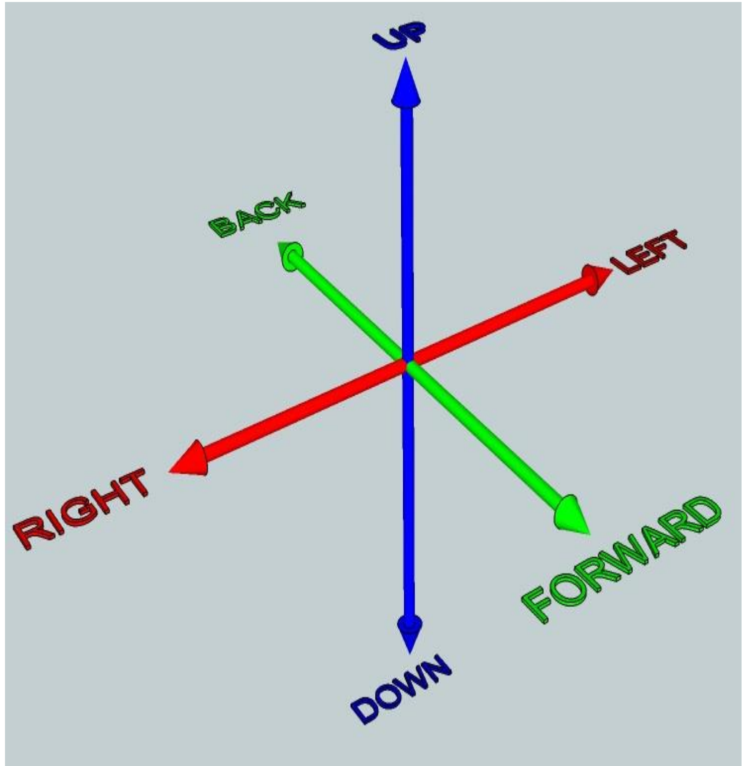
Key Frame Systems:

Key-frame systems are specialized animation languages designed simply to generate the in-betweens from the user-specified key frames. Usually, each object in the scene is defined as a set of rigid bodies connected at the joints and with a limited number of degrees of freedom. As an example, the single-arm robot in Figure has six degrees of freedom, which are called arm sweep, shoulder swivel, elbow extension, pitch, yaw, and roll. We can extend the number of degrees of freedom for this robot arm to nine by allowing three-dimensional translations for the base. If we also allow base rotations, the robot arm can have a total of 12 degrees of freedom. The human body, in comparison, has over 200 degrees of freedom.



Key Frame Systems:

We generate each set of in-betweens from the specification of two (or more) key frames. Motion paths can be given with a kinematic description as a set of spline curves, or the motions can be physically based by specifying the force acting on the objects to be animated. For complex scenes, we can separate the frames into individual components or objects called **cels** (celluloid transparencies), an acronym from cartoon animation. Given the animation paths, we can interpolate the positions of individual objects between any two times. With complex object transformations, the shapes of objects may change over time. Examples are clothes, facial features, magnified detail, evolving shapes, exploding or disintegrating objects, and transforming one object into another object. If all surfaces are described with polygon meshes, then the number of edges per polygon can change from one frame to the next. Thus, the total number of line segments can be different in different frames.



Motion Specification:

There are several ways in which the motions of objects can be specified in an animation system. We can define motions in very explicit terms, or we can use more abstract or more general approaches.

Direct Motion Specification:

The most straightforward method for defining a motion sequence is ***direct*** specification of the motion parameters. Here, we explicitly give the rotation angles and translation vectors. Then the geometric transformation matrices are applied to transform coordinate positions. Alternatively, we could use an approximating equation to specify certain kinds of motions.

Goal-Directed Systems:

At the opposite extreme, we can specify the motions that are to take place in general terms that abstractly describe the actions. These systems are referred to as goal directed because they determine specific motion parameters given the goals of the animation. For example, we could specify that we want an object to "walk“ or to "run" to a particular destination.

Kinematics and Dynamics:

We can also construct animation sequences using kinematic or dynamic descriptions. With a kinematic description, we specify the animation by giving motion parameters (position, velocity, and acceleration) without reference to the forces that cause the motion. For constant velocity (zero acceleration), we designate the motions of rigid bodies in a scene by giving an initial position and velocity vector for each object.

Morphing:

Transformation of object shapes from one form to another is called morphing, which is a shortened form of metamorphosis. Morphing methods can he applied to any motion or transition involving a change in shape. Given two key frames for an object transformation, we first adjust the object specification in one of the frames so that the number of polygon edges (or the number of vertices) is the same for the two frames.

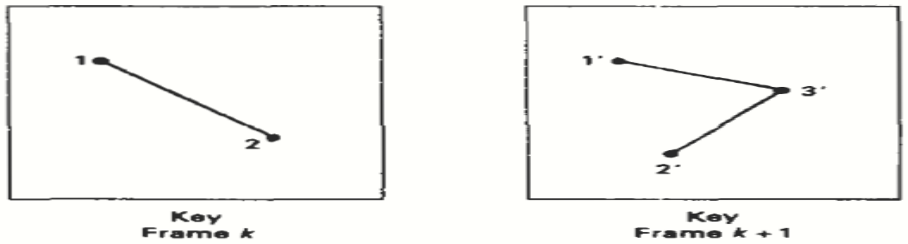


Figure 16-6
An edge with vertex positions 1 and 2 in key frame k evolves into two connected edges in key frame $k + 1$.

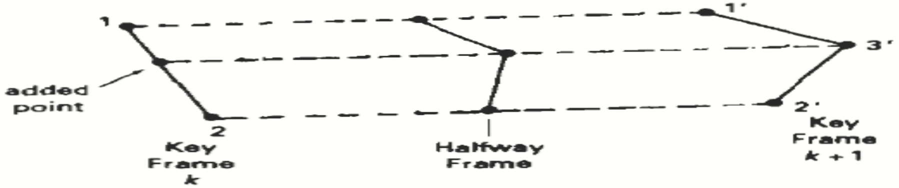


Figure 16-7
Linear interpolation for transforming a line segment in key frame k into two connected line segments in key frame $k + 1$.

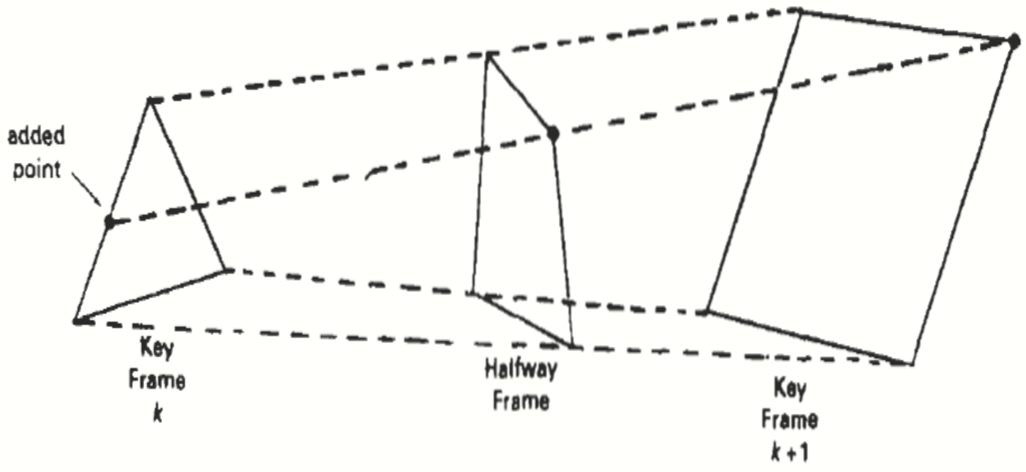
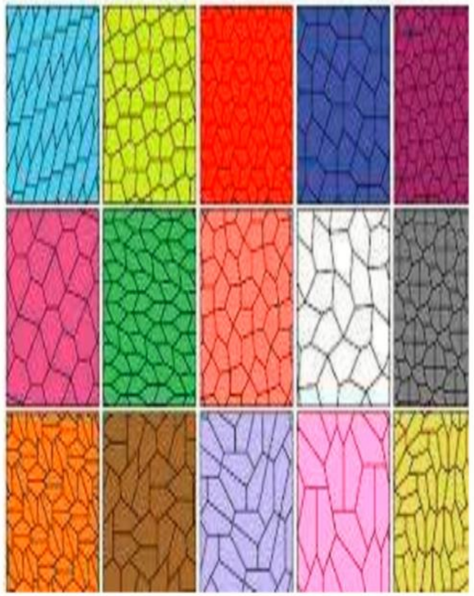


Figure 16-8
Linear interpolation for transforming a triangle into a quadrilateral.

Tiling (tessellation) the plane:

A tessellation or tiling of the plane is a collection of plane figures that fills the plane with no overlaps and no gaps. One may also speak of tessellations of parts of the plane or of other surfaces. Generalizations to higher dimensions are also possible.

- Use one or more geometric shapes
- Tessellation(without gaps) of flat surface
- Shape repeated
- Moving infinity
- Covering entire plane
- Used arts,mosaics,wall papers,tiled floor



Types of tiling

- Monohedral tiling
- Dihedral tiling
- Drawing tiling
- Reptiles

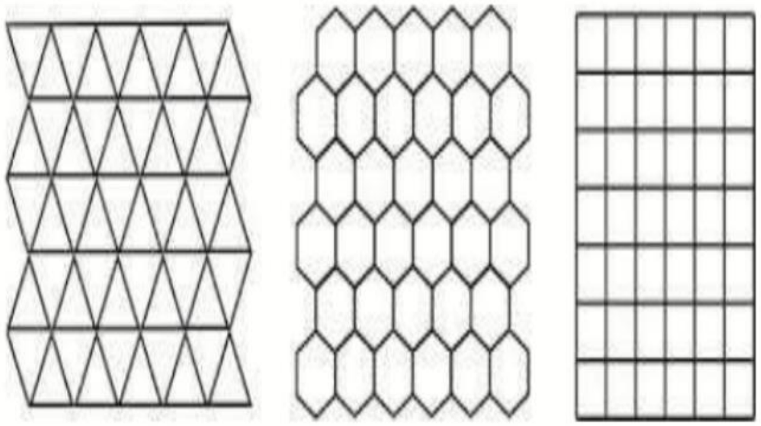
Monohedral tiling

- Based on single polygon
- **Types**

1. Regular tiling
2. Patterns
3. Cario tiling
4. Polymino
5. Polyiamond

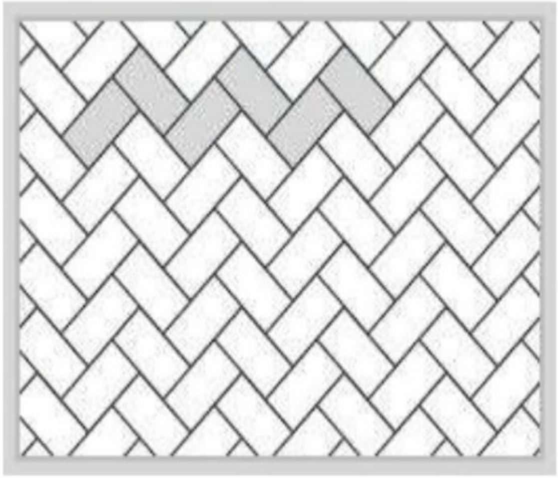
Monohedral Tiling:

Regular Tiling



Patterns

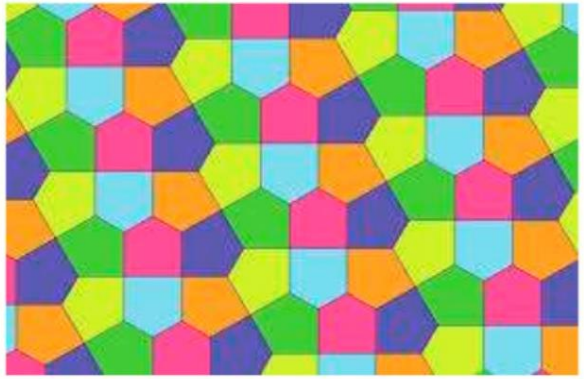
- Shifting the tessellation in particular direction



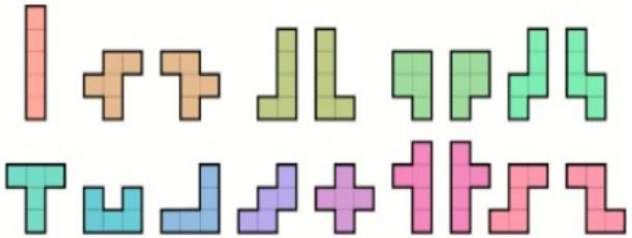
Monohedral Tiling:

Cairo tiling

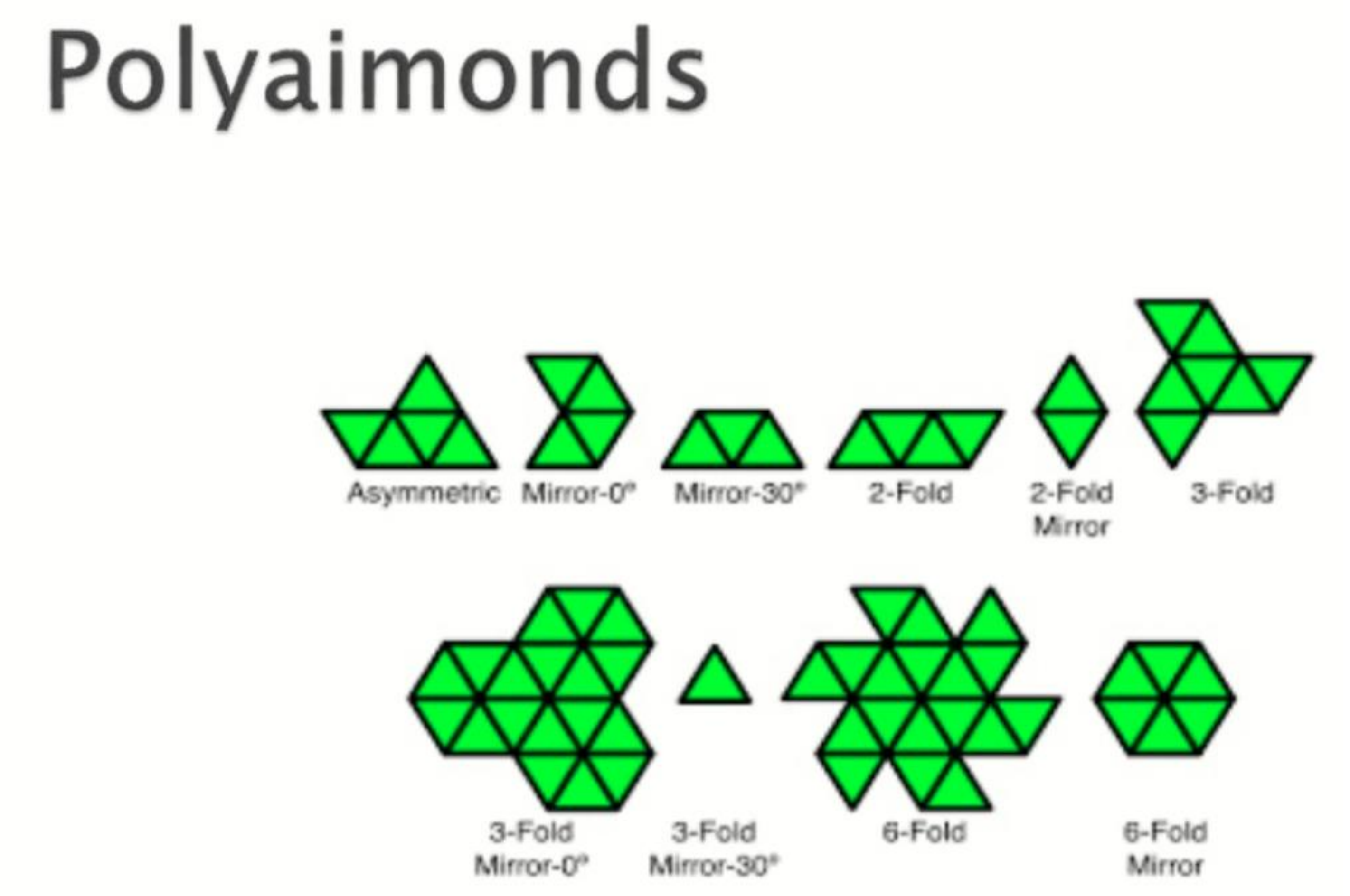
- Four pentagon fit together to form hexagon
- Used to tile the plane
- Many street in cairo,Egypt in this pattern



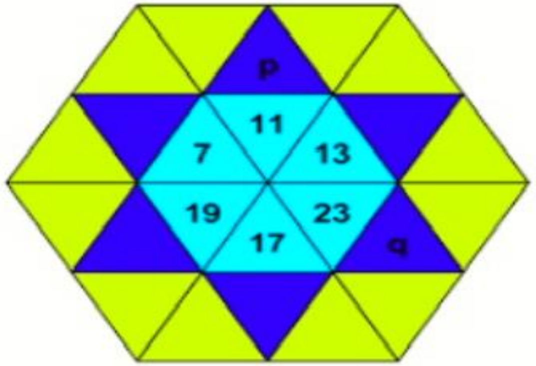
Polymino

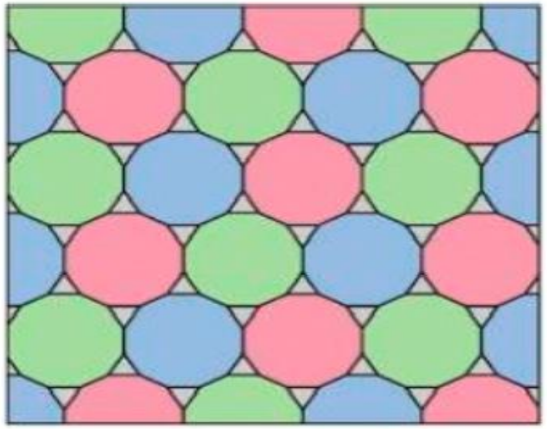


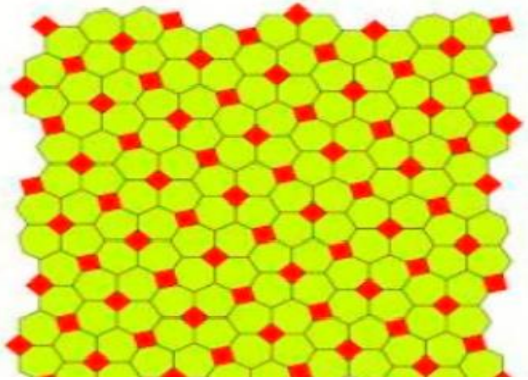
Monohedral Tiling:



Dihedral Tiling:

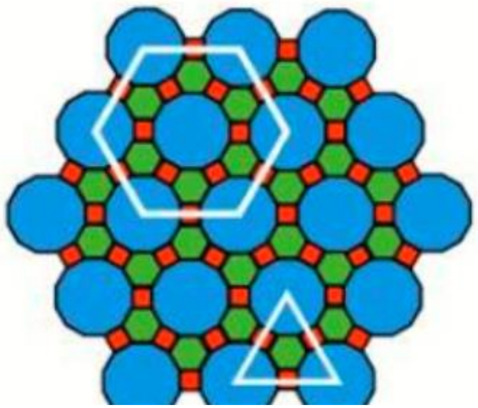






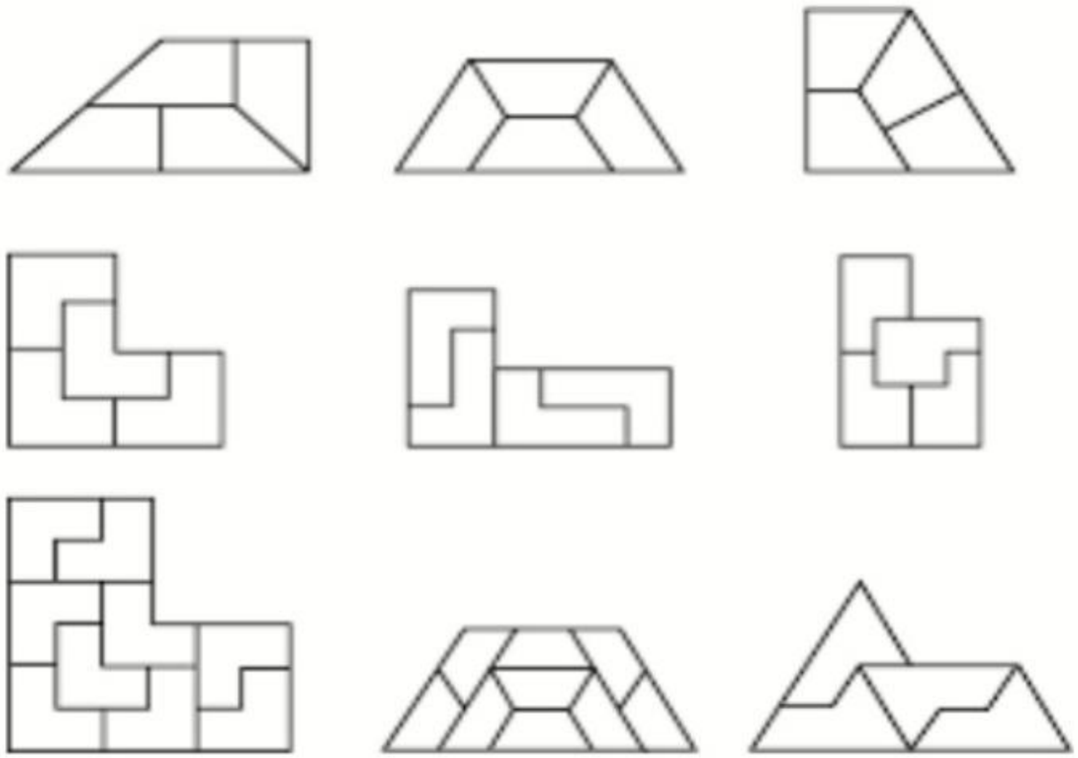
Drawing Tiling:

- Large window setup
- Tiles grouped together into single figure
- Single figure drawn again and again
- Non periodic figure include
- Small to large and large to small



Reptiles:

- Non periodic tiling
- Based on square, equilateral triangle



Fractals:

- A fractal is a never-ending pattern.
- Fractals are infinitely complex patterns that are self-similar across different scales.
- They are created by repeating a simple process over and over in an on-going feedback loop.

Types of Fractals

- Self Similar fractals
- Self Affine fractals
- Invariant fractals

Self Similar Fractals:

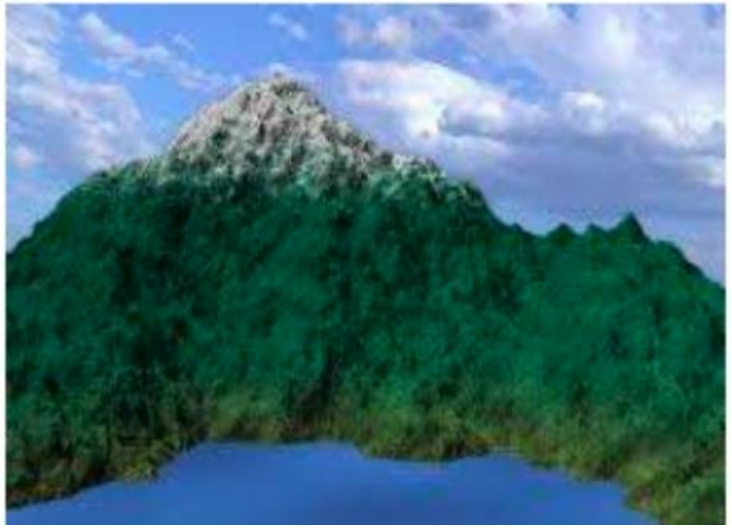
- Geometric figure is self similar
- Fractals appear identical at different scales

Iteration 5

Self Affine Fractals:

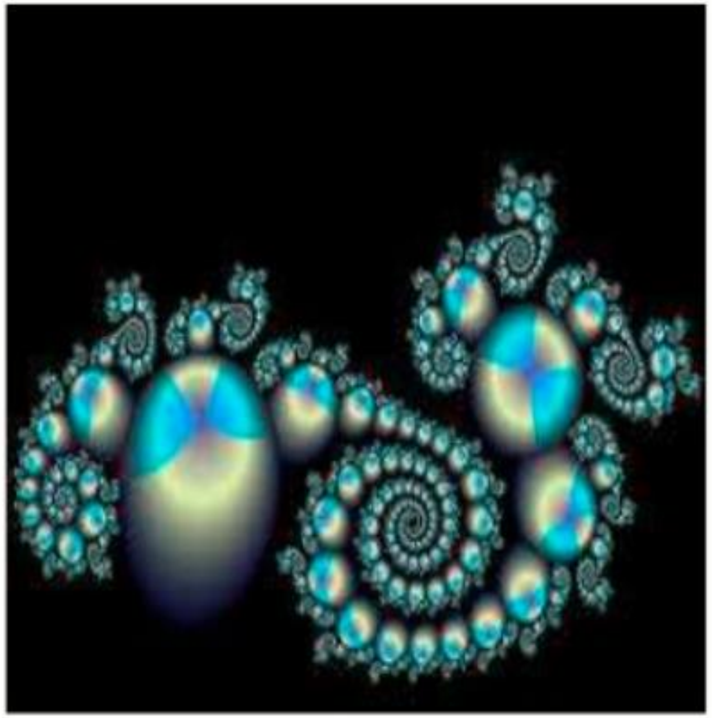
- Fractal appear approximately identical at different scales
- Model water,clouds,terrain

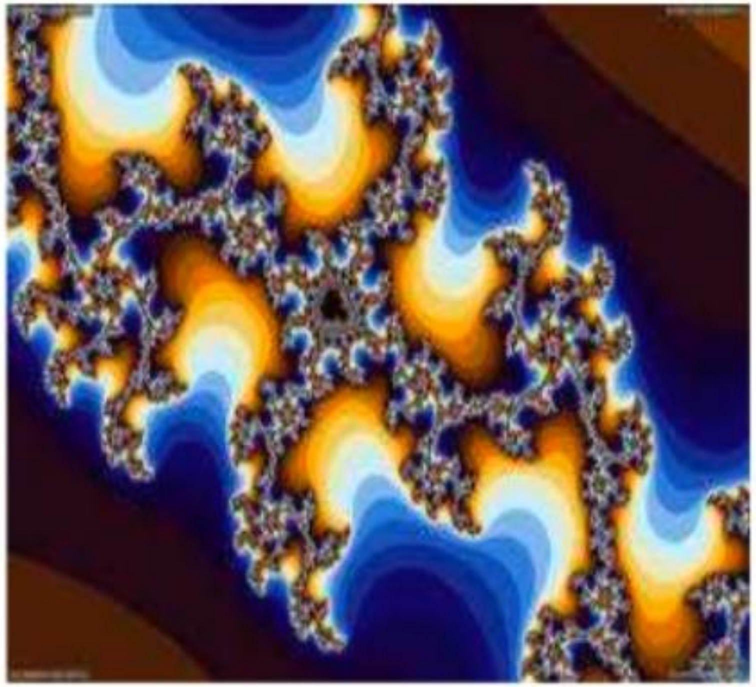




Invariant Fractals:

- Non linear transformation





Recursively Defined Curves:

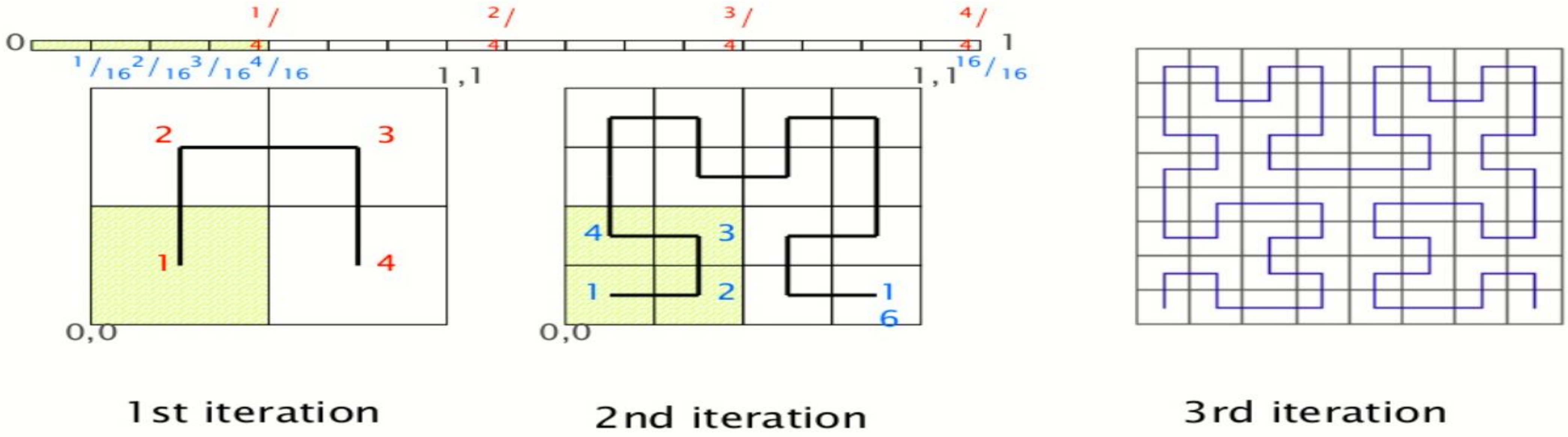
- Curves created by iterations
- Formulas repeated with slightly different values over and over again

Types

- Hilberts Curve
- Koch Curve
- Dragon Curve
- Space filling Curve/Piano Curve
- C Curve

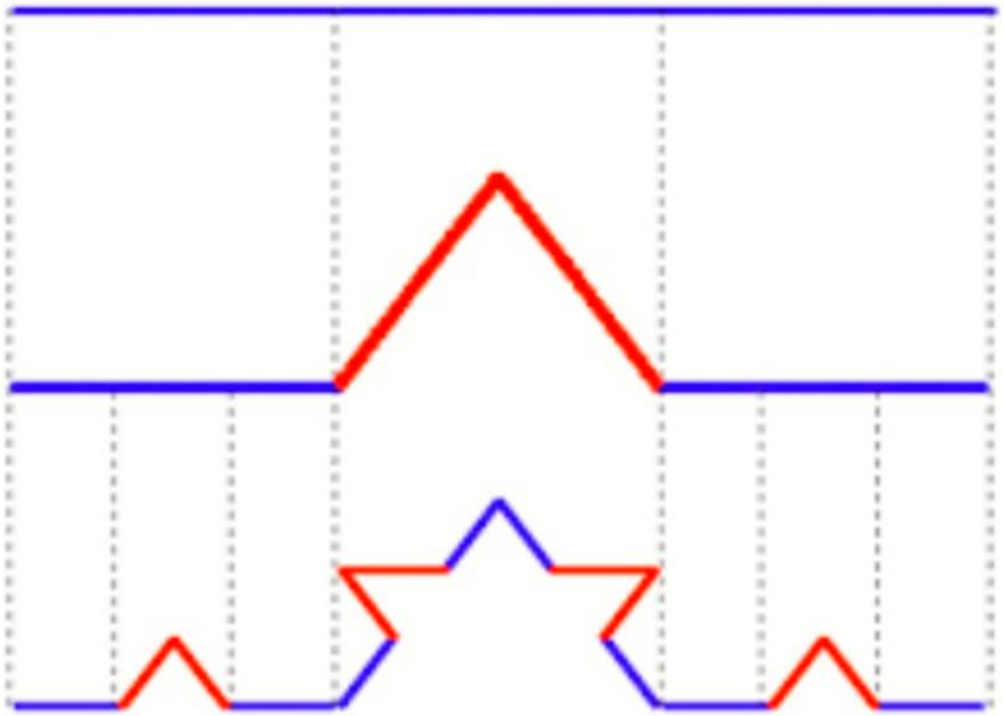
Hilberts Curves:

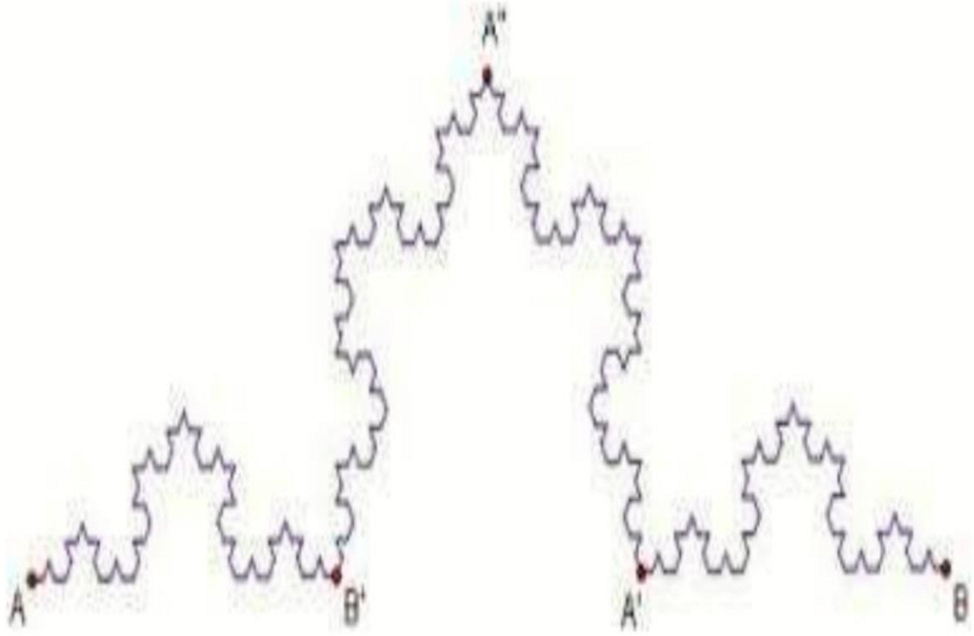
- It was described by the German mathematician David Hilbert in 1891.
- The Hilbert curve is a space filling curve.
- It visits every point in a square grid with a size of 2×2, 4×4, 8×8, 16×16, or any other power of 2.



Koch Curves:

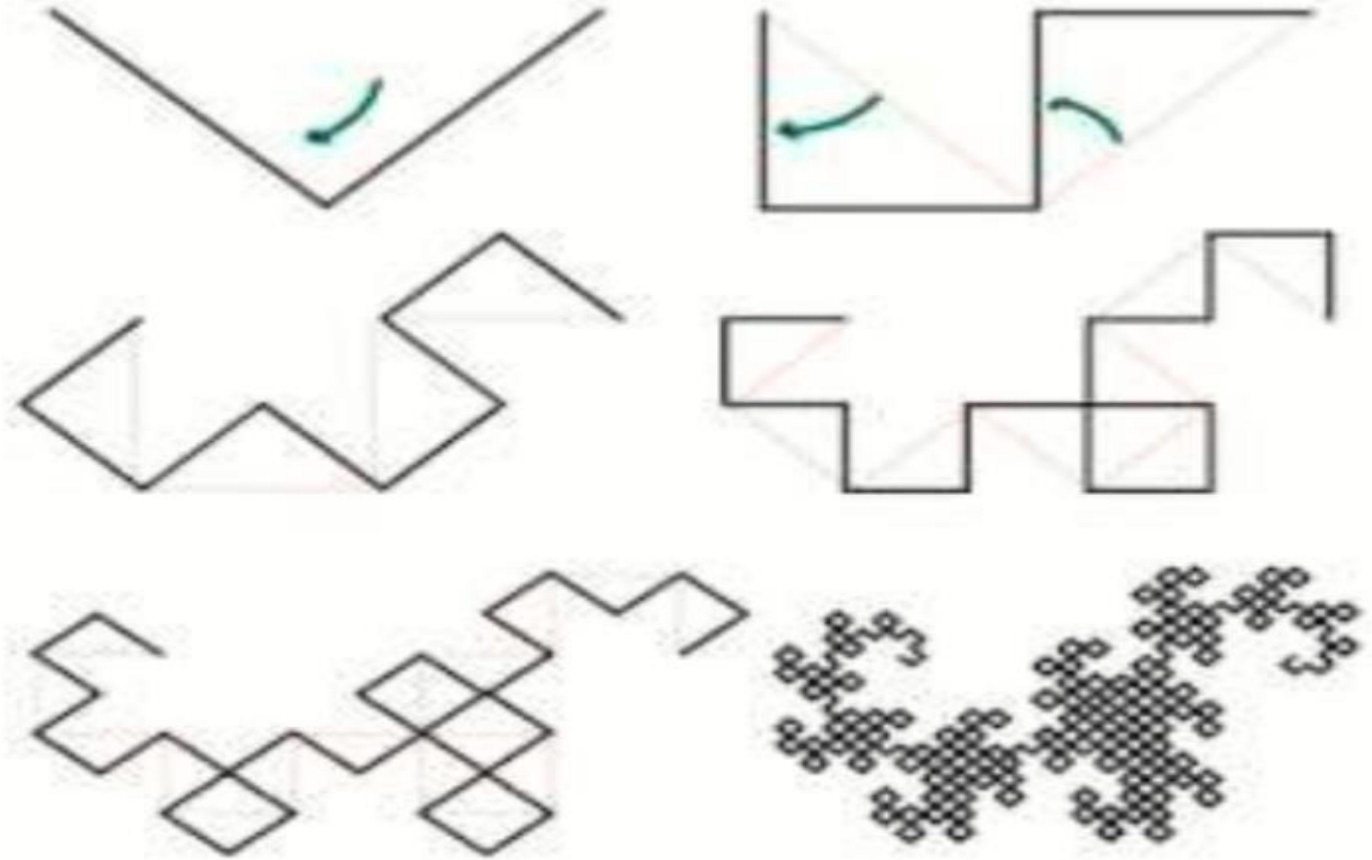
- Developed by Helga von Koch in 1904





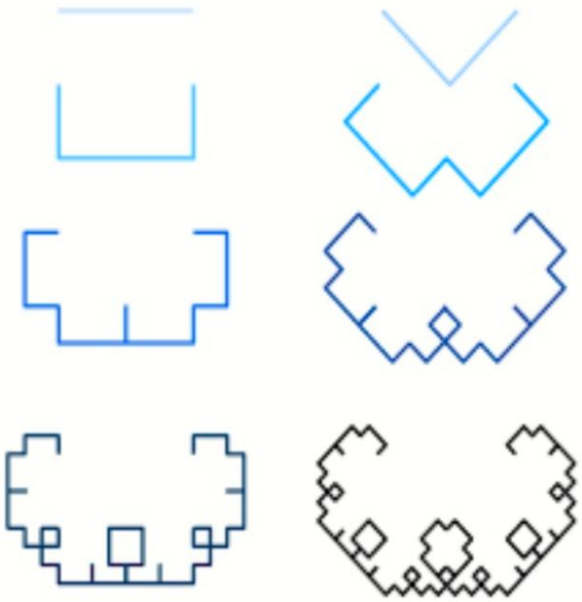
Dragon Curves:

- Self similar fractal curves



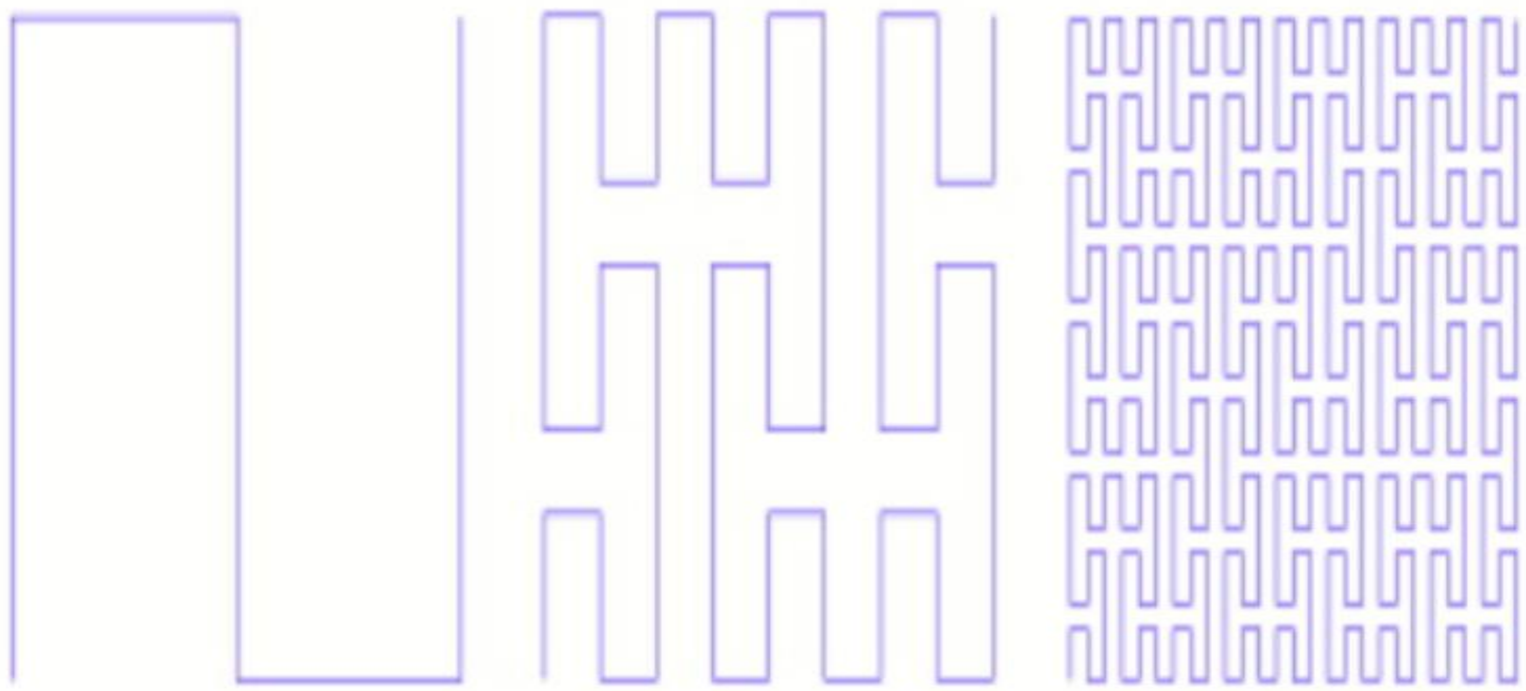
C Curves:

- Self similar fractals
- Described by Ernesto cesaro and Georg Faber in the year 1910



Space Filling Curve/Piano Curves:

- Developed by Italian mathematician Guiseppe peano in 1890
- Space filling curve

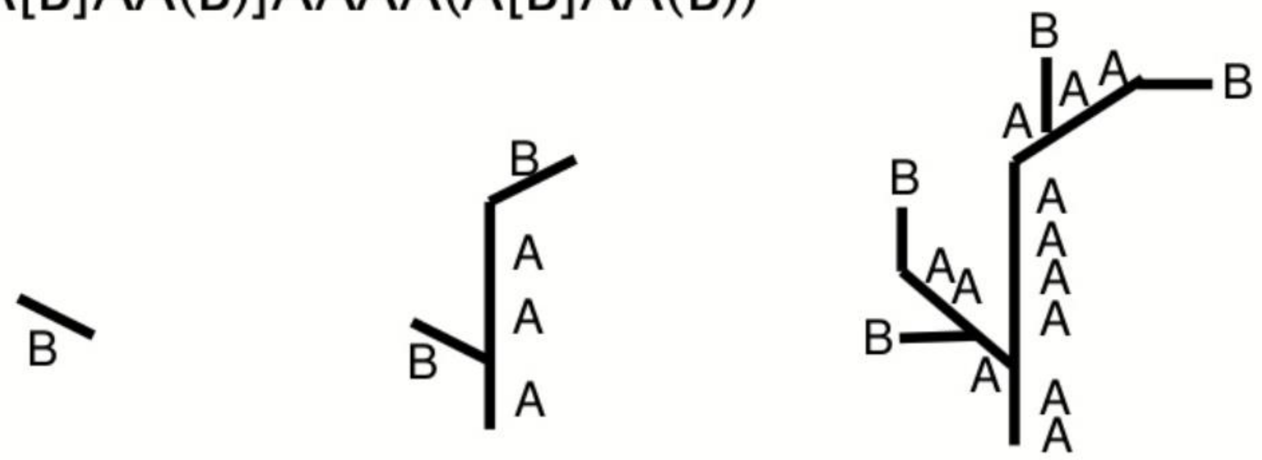


Grammar based models:

- Structure defined by language
- Languages described by a collection of productions
- example, A–>AA creates results of A, AA, AAAA,
- B–>A[B] creates results of B, A[B], AA[B], etc.

Graftals:

- Grammar based models...
  - [ ] for left branches
  - ( ) for right branches
  - A -> AA and B -> A[B]AA(B)
  - create a 2nd generation of:
    AA[A[B]AA(B)]AAAA(A[B]AA(B))



First Generation Second Generation

Graftals:

- Grammar based models...
  - ...use biological productions to simulate plants in development
  - ...describe the topology of plants
  - ...also describe the shape including the directions of branches and the arrangement of leaves

Graftals:

- To simulate the growth of plants using languages include information on...
  - ...the current age
  - ...the growth rate of each segment
  - ...the probabilities of death, dormancy, growth
  - ...the shape (depending on type and age)
  - ...the branch angles (depending on type and age)
  - ...the color and texture of each segment

Graftals:

- Pseudo code simulates the growth of plants using graftals:
  - For (each moment in time)
    - For (each bud that is still alive)
    - Determine whether the bud dies, is dormant, or grow
    - If (the bud does not die)
    - If (the bud is not dormant)
    - Create a portion of a stem, determining its direction, position, color, texture;

Create a new bud;

Graftals:

- Particle systems...
  - ...can be used to simulate fire, clouds, water, fog, smoke, fireworks, trees, and grass
  - ...are particularly useful for animating objects instead of just simulating static objects

Ray Tracing:

The basic ray-tracing algorithm provides for visible-surface detection, shadow effects, transparency, and multiple light-source illumination Many extensions to the basic algorithm have been developed to produce photorealistic displays. Ray-traced displays can be highly realistic, particularly for shiny objects, but they require considerable computation time to generate.

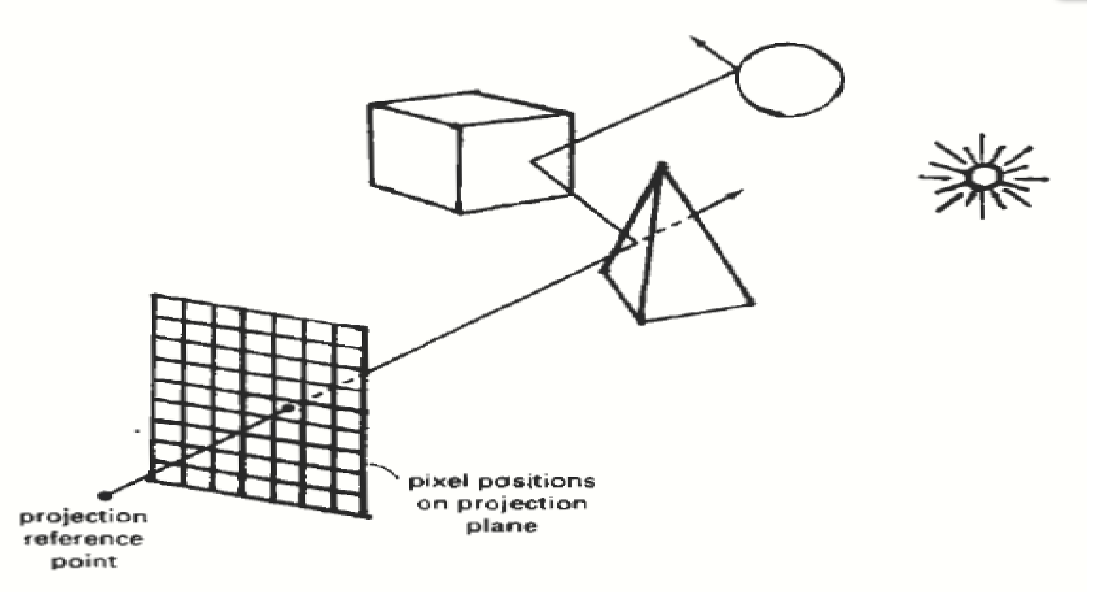


Figure 14-49
Tracing a ray from the projection reference point through a pixel position with multiple reflections and transmissions.

Ray Tracing:

We first set up a coordinate system with the pixel positions designated in the xy plane. The scene description is given in this reference frame. From the center of projection, we then determine a ray path that passes through the center of each screen-pixel position. Illumination effects accumulated along this ray path are then assigned to the pixel. This rendering approach is based on the principles of geometric optics. Light rays from the surfaces in a scene emanate in all directions, and some will pass through the pixel positions in the projection plane.

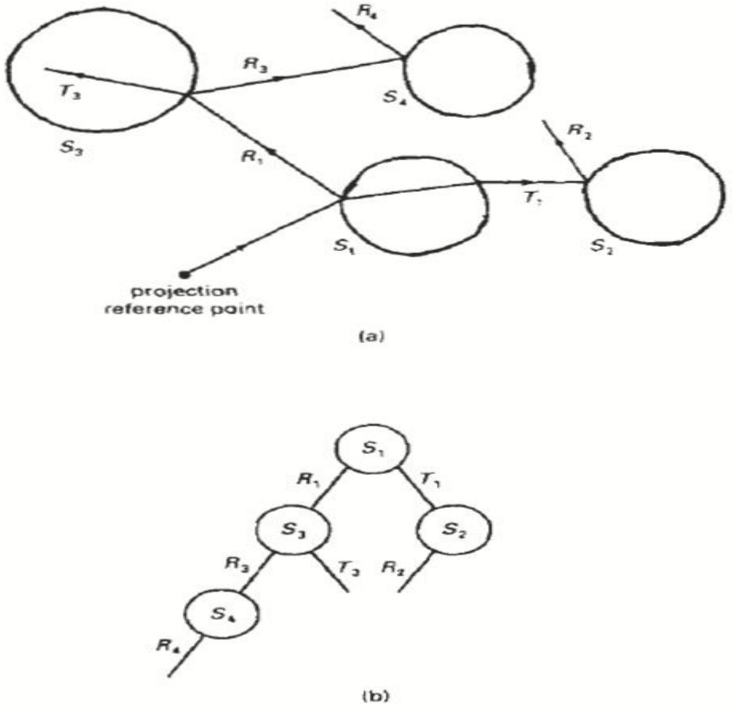
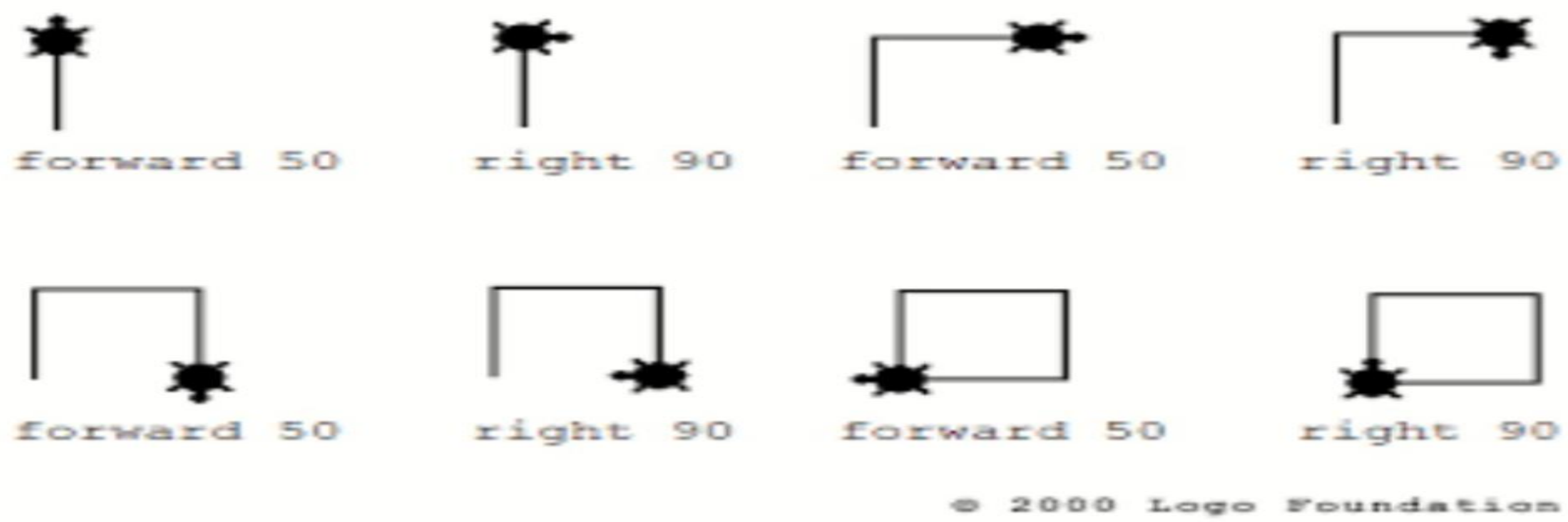


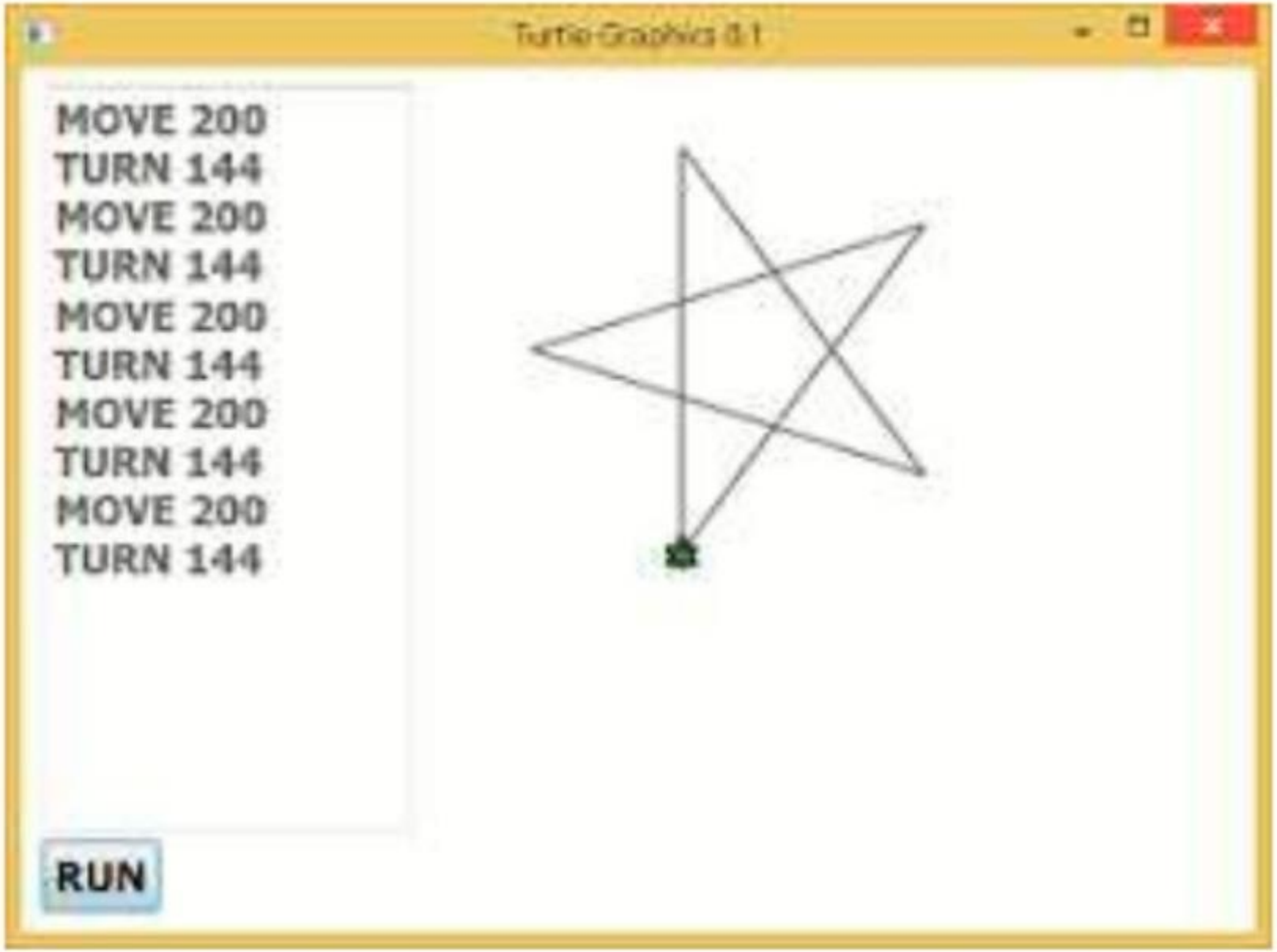
Figure 14-52
(a) Reflection and refraction ray paths through a scene for a screen pixel. (b) Binary ray-tracing tree for the paths shown in (a).

Turtle Graphics:

- Logo programming language
- Developed by feurzig & seymour papert in 1966
- Popular graphics language for kids



Turtle Graphics:




Thanks!