Playful Turtle Geometry in the Paradise

Károly Farkas, *farkas.karoly@nik.uni-obuda.hu*
Dept of Software Technology, Óbuda University

László Csink *csink.laszlo@nik.uni-obuda.hu*
Dept of Software Technology, Óbuda University

Ildikó Tasnádi, *tasiildi@gmail.com*
Nemes Nagy Ágnes Secondary School of Humanities

Abstract

We present a turtle microworld that we have been developing and using for several years in a grammar school. We think that the example is important from a didactic viewpoint. The presentation will be done parallel on two different monitors in two Logo dialects; thereby comparisons are possible between the Imagine and MicroWorlds Ex dialects.

The innovative part of our paper is the generation of various polygons, spiral curves and rosettes with the same Logo procedure, just by changing the parameters.

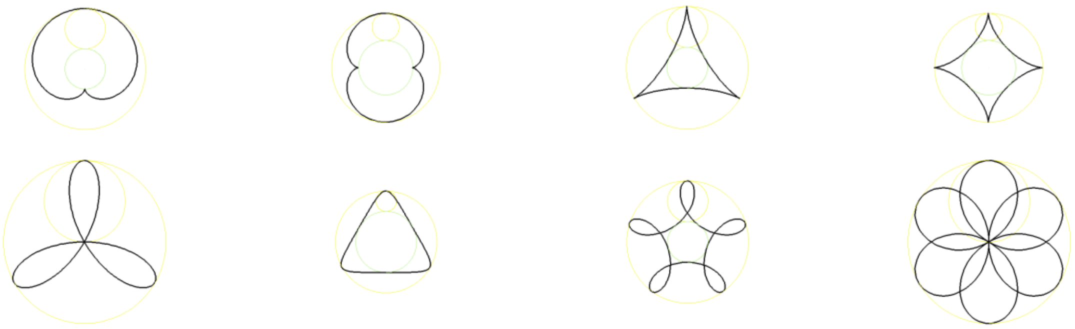


Figure 1. Cardioid, nephroid, deltoid, asteroid, rosettes, produced with the same procedure

The generation of these curves may invoke aesthetical pleasure and is an example how to develop mathematical and computational thinking. Understanding and remembering is more effective when it is linked to primeval stimuli. Turtle geometry is effective due to it's visualise and the experience of motion. Body syntonic significantly helps in understanding the algorithms which can be executed and lived through by the students as well. We think that some special curves are friendlier and easier to grasp if they are superposed by simple turtle motions showing curve ontology.

It is an important didactic advantage that the parameters of the curve generation function and the command parameters can be directly linked to various properties of the moving turtles; thereby the curve transformations can be easily realised and better understood using syntonicity[*]. Function analysis is thus more playful and enjoyable.

In our Paradise we move three people. Adam is performing some motion Eve is doing the same or some other motion independently from Adam, while Cain performs both motions. For superposition we use two ways, either methods attached to the turtles, or an infinite 'super' method.

Keywords

Superposition, playfulness, system, microworlds, genesis

ΠΑΙΖΟΝ ΜΗ ΒΙΑ ΔΙΔΑΣΚΕ ΤΟΥΣ ΠΑΙΔΑΣ! (Πλάτων)
Teach the children with play! (Plato)

'Take my hand. I'm a stranger in Paradise.' (Stranger in Paradise… Borodin - Tony Bennett)
http://www.tsrocks.com/t/tony_bennett_texts/stranger_in_paradise.html

1. Pedagogical Issues

The generation of curves may not only invoke aesthetical pleasure and but it is an example how to develop mathematical and computational thinking, how to use the computer effectively in teaching mathematics and physics, as well as computing. When studying mathematics, many students lose the thread when trigonometric functions and curves are discussed. Students are unable to link these curves to some reality. It helps if many curves are constructed at mathematics classes, but this is time consuming and not enjoyable for most students. Using a computer makes it easier to present many examples, also to keep the attention of students, and the visualisation is more dynamic as well.

When the various curves are produced by moving and animating turtles; the inherent knowledge can be discovered. Understanding and remembering is more effective when it is linked to primeval stimuli.

Turtle geometry is effective due to its visualisation and the experience of motion. Body syntonic – the possibility of linking the steps of drawing to the motion of our own body – significantly helps in understanding the algorithms.

Colleagues taking part at this conference well know the polygons and the Papert-type algorithms of drawing a circle. These algorithms can be easily executed and lived through by students too. It is generally agreed that the other mathematical curves can also be more easily understood and remembered using turtle geometry. We claim that more special curves that are not so easy to generate by traditional methods are also more understandable and friendly if the curve ontology is presented.

We generate the curves in an intrinsic way, from the point of view of the turtle in a relative coordinate system, thereby by attracting attention to the essential features of the curve.

It is an important didactic advantage that the parameters of the curve generation function and the command parameters can be directly linked to various properties of the moving turtles; thereby the curve transformations can be easily realised and better understood using syntonicity. Function analysis is thus more playful and enjoyable.

2. The Stage of Micro World

We move three main and later several minor actors in our Paradise. Let us create Adam, Eve and Cain:

```
newturtle "Adam setc 116 st pd          new "Turtle [name Adam pencolor 9]
newturtle "Eva setc 16 st pd            new "Turtle [name Eva pencolor 12]
newturtle "Cain st pd setpensize 3      new "Turtle [name Cain penwidth 3]
```

Adam performs some motion, Eve does the same independently from Adam, and Cain mimics both parents and performs both motions. Typically for turtle geometry, motions are put together from basic steps and turns, and Cain mixes these two kinds of elements. If these elements are infinitesimal, the resulting curve will be continuous. As we shall see, the sufficiently small element is often one turtle step or turn, or even a fraction of these.

We use two techniques for superposition.

- Superposition using the methods attached to the turtles

```
Set "Adam "onclick [forever [run :a]]        Adam'setEvent "onclick [run :a]
Set "Eva "onclick [forever [run :b]]         Eva'setEvent "onclick [run :b]
Set "Cain "onclick[forever[run :a run :b]]   Cain'setEvent "onclick [run :a run :b]
```

If, for example, the value of :a is fd 1 and :b is rt 1, then Cain steps ahead a little and turns, thereby drawing a circle.

```
make "a [fd 1] make "b [rt 1] everyone [clickon]
(Imagine version, see in the end of attachment.)
```

This is a novel realisation of the Papert-type circle drawing algorithm. The superposition has been successful.

In our next example we put Cain and Eve beside Adam, they all stand on the x-axis looking towards north. Then, if :a and :b are both equal to fd 1 , let us have a look at Cain's action:

```
make "a [fd 1] make "b [fd 1] everyone [clickon]
```

The result of two uniform rectilinear motions is also a uniform rectilinear motion whose speed is the vectorial sum of the speeds of the original motions. Cain, however**, does not do this in any Logo dialects**! He moves slower than expected. We think that the reason of this fault is that we use a Neumann-type computer having one processor only, so the three turtles do not work in parallel, and so Cain misses some steps. If we are not content with the type and genesis of the resulting curve but we are also interested in its relationship with its components, we need to find another way to represent superposition.

- Infinitesimal 'super' procedure

```
to sup :a :b
ask [Cain Adam][run :a]
ask [Cain Eva] [run :b]
sup :a :b
end
```

Running this procedure the turtles will execute our commands in a scheduled and exact way. For example,

```
sup [fd 1] [fd 1]
```

results in Cain's path being twice longer than that of his parents. When running **sup** we have to bear in mind that Cain acts twice in each sup loop, so some usual turtle methods have to be adapted. To draw the **inspi** curves, e.g., the following command can be used

```
make "fi 0 sup [fd 5] [rt :fi make "fi :fi + 1 / 2]
```

In the following we use both the above methods. To demonstrate the composite motions we normally attach methods to the turtles (animate Adam and Eve). When we wish to construct the exact resulting curve we use the **sup** procedure.

3. One of the Components is a Carrier Motion in a Straight Line

Adam's task is to increase the x coordinate in a uniform way: make "a [setx xcor + 1], while Eve moves in various ways.

3.1. Eve rotates

```
make "a [setx xcor + 1] make "b [rt 1] everyone [clickon]
```

or

```
sup [setx xcor + 1] [rt 1]
```

The result a sommersaulting turtle.

3.2. Eva performs an alternating motion

Let us create the alternating method:

```
to alter
fd 1 if (ask "Cain [abs ycor]) > 90 [rt 180]
end
sup [setx xcor + 1] [alter]
```

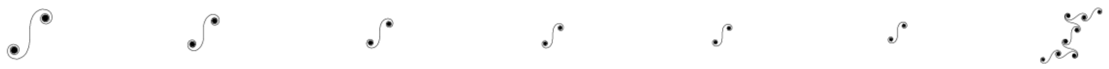
The result is the sawtooth wave.

3.3. Eve rotates with increasing speed

```
make "alfa 0 sup [fd 8] [rt :alfa make "alfa :alfa + d / 2]
```

We got an inward spiral (curve of increasing curvature) where the angle is incremented each time. Changing the starting angle :alfa and the **d** difference value we get the well-known series of curves.

alfa = 0 , **d** is 1, 2, 3, … 7



alfa=1 , **d** is 1, 2, 3, … 7

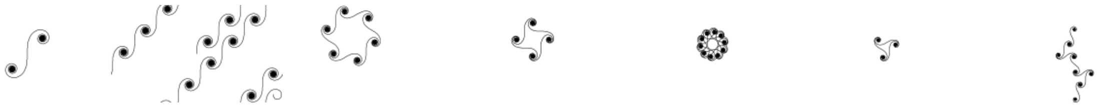
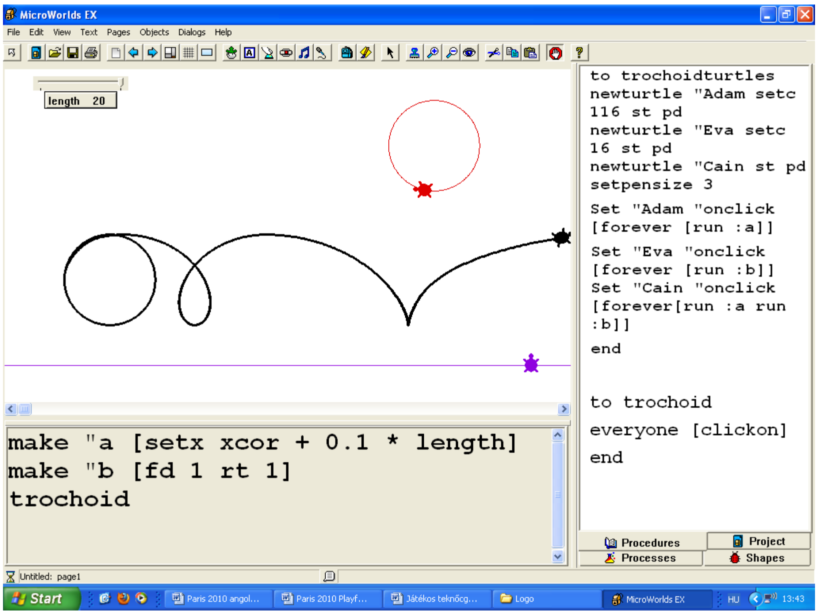


*Figure 2. The elements of the first and second row of the **inspi** curve matrix*

3.4. Eve circulates

```
sup [setx xcor + 1] [fd 1 rt 1]
```

The result is a cycloid. Changing Adam's (or Eve's) step length we get various cycloids.



```
make "a [setx xcor + 0.1 * length]
make "b [fd 1 rt 1]
trochoid
```

```
to trochoidturtles
newturtle "Adam setc
116 st pd
newturtle "Eva setc
16 st pd
newturtle "Cain st pd
setpensize 3
Set "Adam "onclick
[forever [run :a]]
Set "Eva "onclick
[forever [run :b]]
Set "Cain "onclick
[forever[run :a run
:b]]
end

to trochoid
everyone [clickon]
end
```

Figure 3. The cycloid is being transformed by extending Adam's step length

3.5. Eve performs harmonic oscillation

The projection of circular motion to any line parallel with a diameter yields harmonic oscillation. This is the syntonic interpretation as we teach it to our students.

Let us introduce a new turtle, Lucifer:

```
newturtle "Lucifer setc 56 st pd
sety ask "Lucifer [ycor]
```

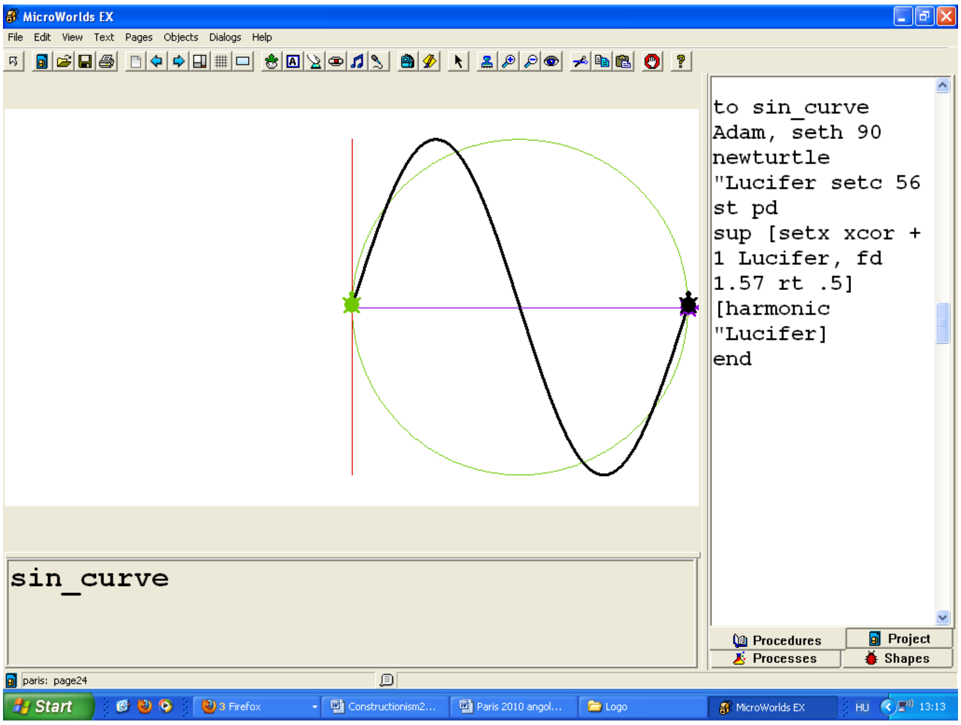
Our procedure is the following:

```
Sup [setx xcor + 1 Lucifer, fd 1.57 rt .5] [harmonic "Lucifer]

to harmonic :Mrx
;Mrx circles
sety ask :Mrx [ycor]
end
```

(Lucifer’s step size is π/2 as he moves twice in each sup loop.)

The resulting sine curve is



```
to sin_curve
Adam, seth 90
newturtle
"Lucifer setc 56
st pd
sup [setx xcor +
1 Lucifer, fd
1.57 rt .5]
[harmonic
"Lucifer]
end
```

```
sin_curve
```

*Figure 4. Generation of sine using **sup***

The pair Eve-Cain was necessary only to explain the generation of harmonic motion. The model can be simplified if only Eve is rotating and Cain is taking the harmonic projection of this circular movement.

```
to sine
sup [setx xcor + 1 Eva, fd 1.57 rt .5] [harmonic "Eva]
end
```

By changing a parameter (e.g. with a slide) we can extend or shrink the sine curve.

3.6. The cosine curve

One way to generate the cosine curve is to use the procedure harmonicyx whose values display the projection of the circular movement to the x-axis, as the cosine function is the projection of the moving line to the x-axis in the unit circle. If we wish to draw the main cycle, let us move Eve ahead with a quarter-circle from the starting point.

```
to cos
Eva, repeat 90 [fd 3.14 rt 1]
sup [setx xcor + 2 Eva, fd 3.14 rt 1] [harmonicyx "Eva]
end
to harmonicyx :Mrx
sety ask :Mrx [xcor]
end
```

3.7. The tangent

It was hard to find a realistic pattern in nature for the demonstration of the tangent. However, we found an example in stage lighting. Let us make the light of a profile spot pass up repeatedly to a canvas and drag the canvas across. The highlight will show pieces of the tangent curve.

To the rectilinear carrier motion we have to superpose a vertical motion created by a light scanning up on the screen. The light source will be played by Lucifer (his name means literally ’light-bearer’), the moving point on the canvas will be the Snake. Lucifer keeps performing flips. The Snake is running back and forth on the canvas, linked to Lucifer (Lucifer is rotating him). The Snake faces Lucifer while Lucifer’s light is rising; while they look to the same direction when Lucifer’s light moves downwards.

The methods used can be found in the attachment.

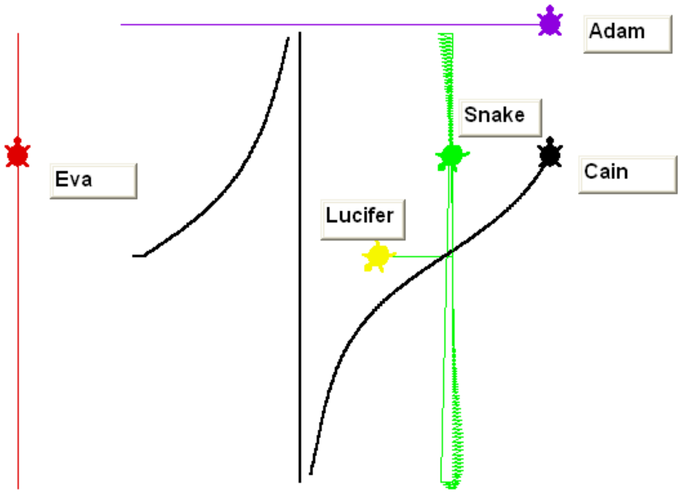


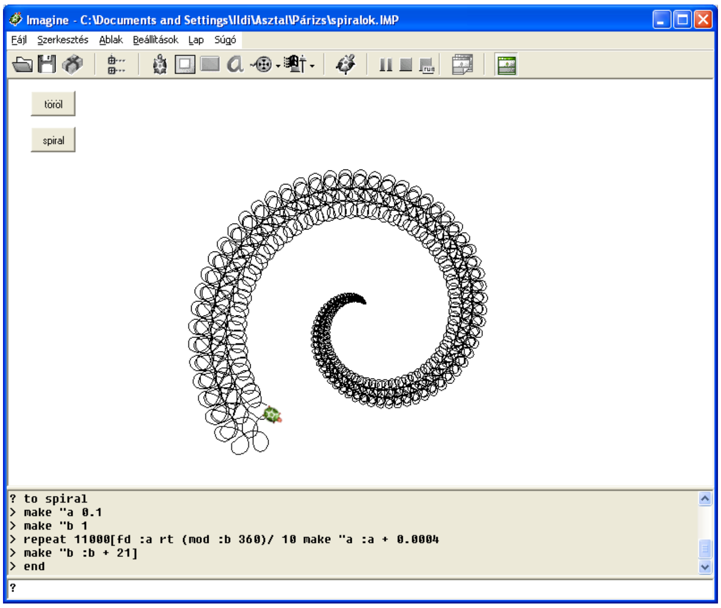
Figure .5 Generation of tangent in the Paradise

4. Adam Performs Rectilinear Accelerated Motion

```
make "a initialA make "b initialB sup [fd :a make "a :a + deltaA] [rt :b make
"b :b + deltaB]
```

When Eve rotates at constant speed (constant4 is zero), we get the well known **polispi** curves.

When Eve gains speed while rotating, we get various spectacular spirals, such as


Imagine - C:\Documents and Settings\Ildi\Asztal\Párizs\spiralok.IMP
Fájl Szerkesztés Ablak Beállítások Lap Súgó
töröl
spiral

```
? to spiral
> make "a 0.1
> make "b 1
> repeat 11000[fd :a rt (mod :b 360)/ 10 make "a :a + 0.0004
> make "b :b + 21]
> end
?
```

Figure 6. A 'real' spiral

5. Adam is Rotating

If Eve is also rotating, Cain is rotating at the speed of Eve and Adam added together. If Eve performs rectilinear motion, we get the curves as before, with Eve and Adam changing their roles.

6. Adam Rotates Eve, Eve Rotates Cain

This way Cain draws various curves resembling planet motions. When Adam's and Eve's direction of rotation is the same, the curves are epicycloids (see the first two curves in Figure 7). When Adam and Eve rotate in the contrary direction, we get hypocycloids. Using various parameters of the same procedure we generate the cardioids, the nephroids, the deltoids, the astroids and the rosettes, as well as their looped and extended versions. Adam, Eve and Cain may represent the Sun, the Earth and the Moon, respectively. The curves are the Moon's orbits in absolute coordinates.

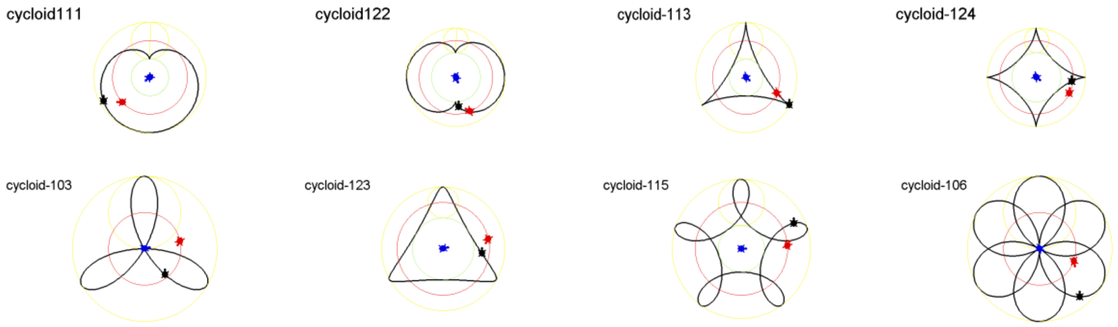


Figure 7. Some basic 'motives' of the listed curves

```
to cycloid :e :i :z
everyone  [pu  seth  0  setx  0]  Eva,
sety 80
ifelse :e = 1 [Cain, pu sety 80 * :i
/ (1 + :i) pd Eva, seth 180] [Cain,
pu sety 80 + 80 / (:i + 1) pd]
make "d2 80 / (:i + 1)
roll
end

to roll
Adam, rt :e
make "alfa1 heading
Eva, setpos list 80 * sin :alfa1 80
* cos :alfa1 seth heading + :e
rt :z
make  "alfa2  heading  make  "x2  xcor
make "y2 ycor
Cain,  setpos  list  :x2  +  :d2  *  sin
:alfa2 :y2 + :d2 * cos :alfa2
roll
end
```

```
to cycloid :e :i :z
 ask all [pu seth 0 setxcor 0]
 ask "Eva [setycor 80]
 ifelse  :e  =  1  [ask  "Cain  [pu
   setycor 80 * :i / (1 + :i) pd]
   Eva'seth 180]
   [ask "Cain [pu setycor 80 + 80 /
   (1 + :i) pd]]
   make "d2 80 / (:i + 1)
   roll
  end

to roll
  Adam'rt :e
  make "alfa1 Adam'heading
   ask  "Eva  [setpos  list  80  *  sin
   :alfa1    80    *    cos    :alfa1
   seth heading + :e rt :z]
  make "alfa2 Eva'heading
  make "x2 Eva'xcor
  make "y2 Eva'ycor
  ask "Cain [setpos list :x2 + :d2 *
 sin :alfa2 :y2 + :d2 * cos :alfa2]
  roll
end
```

7. Both Components are Harmonic Oscillations

sup [harmonic "Lylith][harmonicxx "Lucifer]

The result is the Lissajous curve. To generate the various Lissajous curves in a spectacular way we use our method technique. To draw the resulting curve we construct motion elements as composites.

```
to Lissajous
Lucifer, fd 1 rt 1
Lylith, fd 1.5 * parameter / 10  rt 1 * parameter / 10
Adam, harmonic "Lylith
Eva, harmonicxx "Lucifer
Cain, setpos list ask "Lylith [xcor] ask "Lucifer [ycor]
Lissajous
end
```

Lucifer and Lylith keep rotating. The parameter changes the ratio of the speeds. harmonicxx is the projection of circular motion on the x-axis.

Figure 8. Some Lissajous curves

8. Summary of Curves

:a	:b	Resulting curve	Note
fd i	fd i	line	Vectorial sum of paths
fd i	rt i	circle	
increase xcor	rt i	somersault	
increase xcor	fd i rt i	cycloid	
increase xcor	alternating	sawtooth	
increase xcor	harmonic	sine	
Increase xcor	tangent value	tg	Tg is the ycor value of highlight
fd k	rt k	polygon	
fd k	rt inc k	inspi	
fd increase k	rt k	polispi	
fd increase k	rt inc k	spiral	
rt 1	rt 1	rotating	
rotate ”Eva	rotate ”Cain	rosetta	Adam rt 1 : epicykloid Adam lt 1 : hipocykloid
harmonic	harmonic	Lissajous	Lucifer and Lylith circle

i = motion element of infinitesimal size
k = constant

Summary

To generate curves with superposition, we need only relatively simple mathematical operations. Thus we can teach new chapters of geometry using turtle geometry in earlier age and with more effectiveness. An increasing set of curves can be systematised in a playful microworld, thus aiding the development of system approach.

Our work can be regarded as a continuation of research of, among others, Uzi Armon, Sergei Supronov and Izabella Foltynowicz. Our goal is to draw various curves playfully, to keep up motivation in informatics classes, and to demonstrate research to our students.

References

Abelson, H. and diSessa, A. (1986) *Turtle Geometry. The Computer as a Medium for Exploring Mathematics.* The MIT Press.

Alexandrov, K. and Soprunov, S. (1997) *One Approach to Mechanic Simulations in Logo*. In: M. Turcsányi-Szabó (ed.) Proceedings Sixth European Logo Conference, Budapest 114-121

Armon, U. (1997) *An Algorithm that Translates Intrinsic Equations of Curves into Intrinsic Procedures of These Curves*. In: M. Turcsányi-Szabó (ed.) Proceedings Sixth European Logo Conference, Budapest, August. pp. 378-389

Csink, L. and Farkas, K. (2008) *Turtle's Curves*. In: R. T. Mittermeyer M. M. Sysło (ed): Proceedings Informatics Education Contributing Across the Curriculum, Torun, Poland, 76-86

Farkas, K. (2003) Logo and native language. *Intrinsic procedures of some curves*. In: Secundino Correia (ed.): Proceedings of the 9th European Logo Conference, Porto pp. 69-79

Foltynowicz, I. (2007) *Cicloids and limacons in the turtle graphics*. In: Kalaš, I. (ed.) Proceddings of the 11th European Logo Conference, EuroLogo, Bratislava, Slovakia, August. p. 50.

Pólya, Gy. (1954) How to Solve It; Induction and Analogy in Mathematics. Princeton University Press, Princeton; and (1969) Patterns of Plausible Inference Princeton, N. J. Princeton.

http://www.freeweb.hu/mwlogo/Paris2010.html

Attachment

```
to tangent
;for Microworld Ex
Lucifer, st pd setx 0 seth 90 set "Lucifer "onclick [forever [lt 0,025]]
Snake, st setpos [0 0]
Adam, pu setpos [-200 180]
Eva, pu setx -280 seth 0
Cain, st pu setx -300
make "a [setx xcor + 0,05]make "b [sety ask "Snake [ycor]]
ask [Adam Eva Cain Lucifer][clickon]
end
to tg
make "k ask "Cain [ycor]
make "l1 (ask "Lucifer [heading]) < 165 make "l2 (ask "Lucifer [heading]) > 15
make "ji and :l1 :l2
make "l3 (ask "Lucifer [heading]) < 345 make "l4 (ask "Lucifer [heading]) >
195
make "bi and :l3 :l4
if :ji [ifelse :k < -170 [Cain, pu][Cain, pd] Snake, stick_to "Lucifer setx 57
]
ifelse not :k < 169  [Cain, pu make "a [setx xcor + 0,01]] [make "a [setx xcor
+ 0,05]]
if :bi [ifelse :k < -169 [Cain, pu][Cain, pd] Snake, stick_to2 "Lucifer setx
57]
tg
end

to stick_to :a
  towards :a  make "d distance :a
  make "alfa ask :a [heading] make "x ask :a [xcor]  make "y ask :a [ycor]
  setpos list :x + :d * sin :alfa :y + :d * cos :alfa
end

to stick_to2 :a
  towards :a make "d distance :a
  make "alfa ask :a [heading] make "x ask :a [xcor]  make "y ask :a [ycor]
  setpos list :x - :d * sin :alfa :y - :d * cos :alfa
end
```

The Microworlds ’ Everyone [clickon] ’ equal in Imagine:

```
Adam'forever [runEvent "onClick]
Eva'forever [runEvent "onClick]
Cain'forever [runEvent "onClick]
```

[*] We use ’syntonicity’ expression after Papert. Mindstorms, page 63.