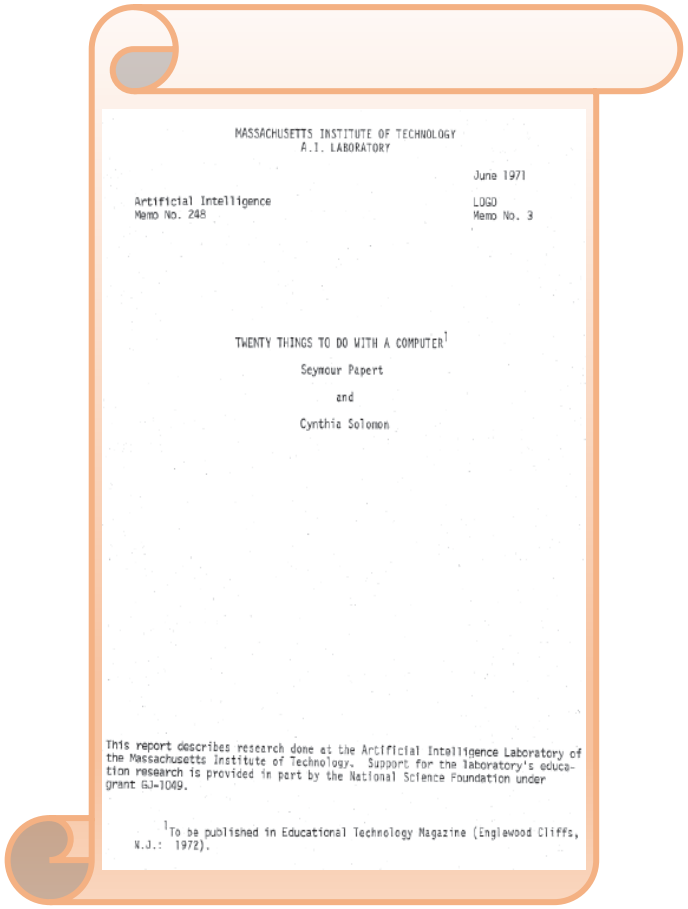
20 things to do with a computer – преди 50 години и днес



MASSACHUSETTS INSTITUTE OF TECHNOLOGY
A.I. LABORATORY

June 1971

Artificial Intelligence
Memo No. 248

LOGO
Memo No. 3

TWENTY THINGS TO DO WITH A COMPUTER[1]

Seymour Papert

and

Cynthia Solomon

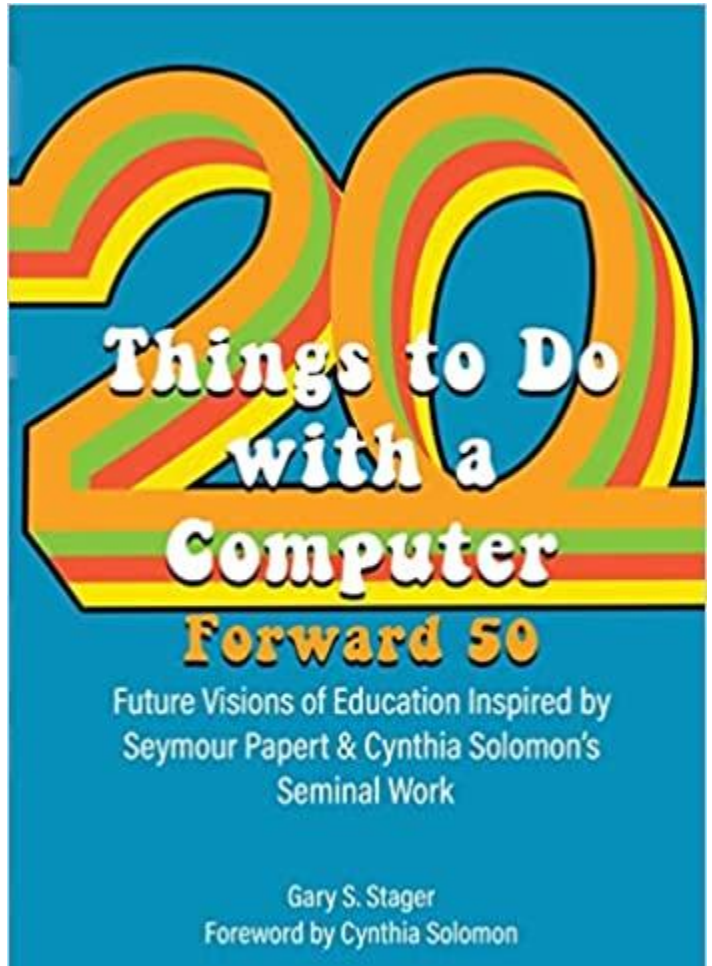
This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's education research is provided in part by the National Science Foundation under grant GJ-1049.

[1]To be published in Educational Technology Magazine (Englewood Cliffs, N.J.: 1972).

Евгения Сендова
ИМИ-БАН

Национален семинар по математическо образование

27 ноември 2021



Защо?

- **Gary Stager:** *My goal was to honor Seymour & Cynthia while inspiring readers to think about powerful ideas for the next 50 years. I cast a wide net and invited lots of authors, spanning three generations, whether I agree with them or joy. The book contains essays by pioneers, scholars, school administrators, and classroom teachers.*
- **Sylvia Martinez:** *For myself, it was great to revisit the original paper and think about how much it offered, and how much is still left to do. My hope is that this book is not read as a historical document, but as a reminder that children can do so much more than we currently ask of them in most classes, Even if just a few teachers read this and think, "I can do more…" that's a good outcome*.

Contents

Contents

Foreword i
Cynthia Solomon

Introduction 1
Gary Stager

Section 1 — Our Vision

Twenty (Poetic) Things to Do with a Computer 11
Angi Chau

Progress 13
Gary Stager

Spiraling 19
Artemis Papert & Cynthia Solomon

Adventures on the Road to Mathland 25
Dan Lynn Watt

Twenty Things is the Foundation for Constructionism 39
Fred Martin

Life in Logoland 43
Marian B. Rosen

The First Thing I Did with a Computer 49
Bryan P. Sanders

Twenty Things and Onwards 51
Stephen Heppell

Purple Constructionism 55
Nettrice Gaskins

My Rules 57
Jennifer Orr

Radical Ideas: Joy and Empowerment 61
Ken Kahn

An Eternal Source of Inspiration or What the Bulgarian Turtle Told Achilles This Time 63
Evgenia (Jenny) Sendova

I Learn How to Code and Lengthen My Reach #makingliberation 93
Susan Klimczak

The Best Day Ever 101
Cathy Hunt

Section 2 — Intellectual Timidity

The Computer Programs the Child 105
Audrey Watters

Where Have All the Metaphors Gone? 107
Gary S. Stager

Hitchhiking to Logo's Mathland 117
Molly Lynn Watt

When You Wake Seymour Papert in the Middle of the Night 127
Gary Stager

2020 Things to Do with a Computer 129
Dale Dougherty

Intellectual Timidity 135
Eleonora Badilla-Saxe

Seeking the Magic: One Teacher's Evolution Story 139
Carol Sperry

Bad Design is Violence: Powerful Ideas and Powerful Politics in Constructionist Education 145
Paulo Blikstein

Thou Shalt Not Write Curriculum 151
Peter Rawitsch

The Wider Walls 153
Bill Kerr

Let's Kill the Pencil 157
Gary Stager

Whatever Happened to the Revolution? 159
Geraldine Kozberg

Section 3 — Personal Computing

A Dreamer Given Sight 165
David Loader

Personal Computing 171
Gary Stager

Kid Power 181
Dennis O. Harper

Education Technology @ Fifty Something: My Personal Perspective 187
Karen J. Billings

To Be or Not to Be ... (a Programmer) 191
John Stetson

Change Is Certain, Progress Is Not 195
Ron Canuel

Twenty Things and Moving Forward: An Embodied-based Scenario for Kindergarten Children 199
José Armando Valente

Math Classes Are Failing the AI Age Workforce 207
Conrad Wolfram

The End of Knowing – and a New Way to Learn 209
Sugata Mitra

The Evolution of Logo Connections to the Physical World, From 1971 to Today 221
David D. Thornburg

Learning Together 227
David Cavallo

Section 4 — Recursion Line

Thoughts on XX 237
Tom Lough

Reflections on Papert and Solomon from the History and Civics Classroom 241
Heather Allen Pang

A Language for Making Physical Things 245
Leo McElroy

Things to Do with a Computer in an English/Language Arts Classroom (Besides Word Processing) 253
Kate Tabor

Twenty Things to Make with Biology 257
Yasmin B. Kafai & Justice T. Walker

For Real: Some Modern Things to Do with a Computer 267
Martin Levins

Twenty Ways to Facilitate Twenty Things to Do with a Computer 273
Carmelo Presicce, Giulio Bonanome, & Angela Sofia Lombardo

A Place for Experimentation: The Library 281
Carolyn Foote

Computer Modeling, Data Collection, and Programming in Middle School Science 283
Donna Collins

Theo's Rockets 295
Gary Stager

Ten Things to Do with the Internet 299
Tom Lauwers

Three European Robotics Projects Inspired by Twenty Things 303
Michele Moro & Dimitri Almisis

A Robot Petting Zoo 305
John Umekubo

Collaboration Rules! Twenty Insights About Online Collaborative Learning 311
Yvonne Marie Andres

Make a Turtle! 317
Miles Berry

#11: Make a Music Box and Program a Tune 323
Walter Bender & Devin Ulibarri

The Future is Computational 333
Gary Stager

Isn't it Time for Us to Grow Up? 345
Seymour Papert

Toolbox 346
Contributor Biographies 347
Twenty Things to Do With a Computer (1971) 359
Seymour Papert & Cynthia Solomon

Also from Constructing Modern Knowledge Press 401

An Eternal Source of Inspiration or What the Bulgarian Turtle Told Achilles This Time

Evgenia (Jenny) Sendova

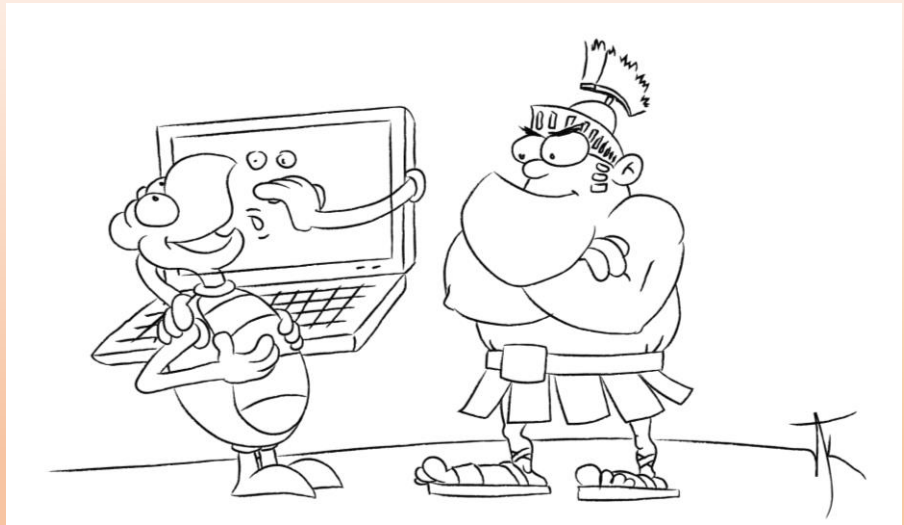
But there is a world of difference between what computers can do and what society will choose to do with them.—Seymour Papert

Prologue

Scene 1 – Enter the narrator

It was not long ago that I took part in a discussion on "innovations in education." The moderator was very surprised when I said that a good innovation in contemporary education would be for students to go to school with love—as was the case with the so called "cell schools" (the first and only schools in Bulgaria from the fifteenth to eighteenth century) and in more recent times in the schools that the Research Group on Education of the Bulgarian Academy of Sciences ran as an experiment from 1978 to 1999 in 2% of the Bulgarian schools (Sendov, 1987). These days not only in Bulgaria but also around the world, innovations are often associated with technology such as multimedia, interactive boards, smartphones, etc. and the innovativeness of a school is measured by the number of laptops and not by what the teachers or the students have *chosen to do with them.*

Scene 2 – Enter Achilles and the Turtle



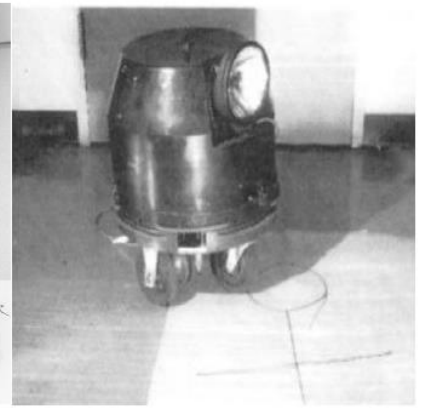
Защо именно тези герои?

- Lewis Carroll “What the Tortoise Said to Achilles” (1895)
- Douglas Hofstadter (1980) *Gödel, Escher, Bach*

Scene 2 – Enter Achilles and the Turtle

Achilles: By CS article, you mean an article about computer science?

Turtle: No, my dear miseducated friend! I mean Cynthia (Solomon) and Seymour (Papert), the authors of this fundamental article! They are illustrating MY geometry first with a physical robot-turtle (the floor turtle) and then with what they call a "display turtle"—a small triangle.

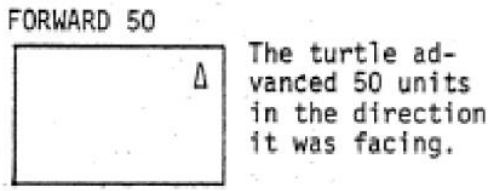


Achilles: A triangle? Why on earth, a triangle? Did Kandinski make your portrait?

Turtle: I wish he did but in fact, my original Logo image exposes and emphasizes my new role as a drawing instrument (which some more modern environments hide behind cats, bees, and other animals).

Achilles: That is true. I remember your dialog with primary teachers who told you once: *We don't work with turtles, we work with other animals!*

Turtle: But Logo people know that under any "mask" it's me, the turtle, which can draw figures and become different characters if needed.



Act 1

Scene 1 – Drawing a man (Thing #2)

Achilles: I have heard that in drawing, nothing is better than the first attempt.

Turtle: Maybe for Picasso this might be right, but in programming we say that if a program works when first executed, there is certainly a bug in it.

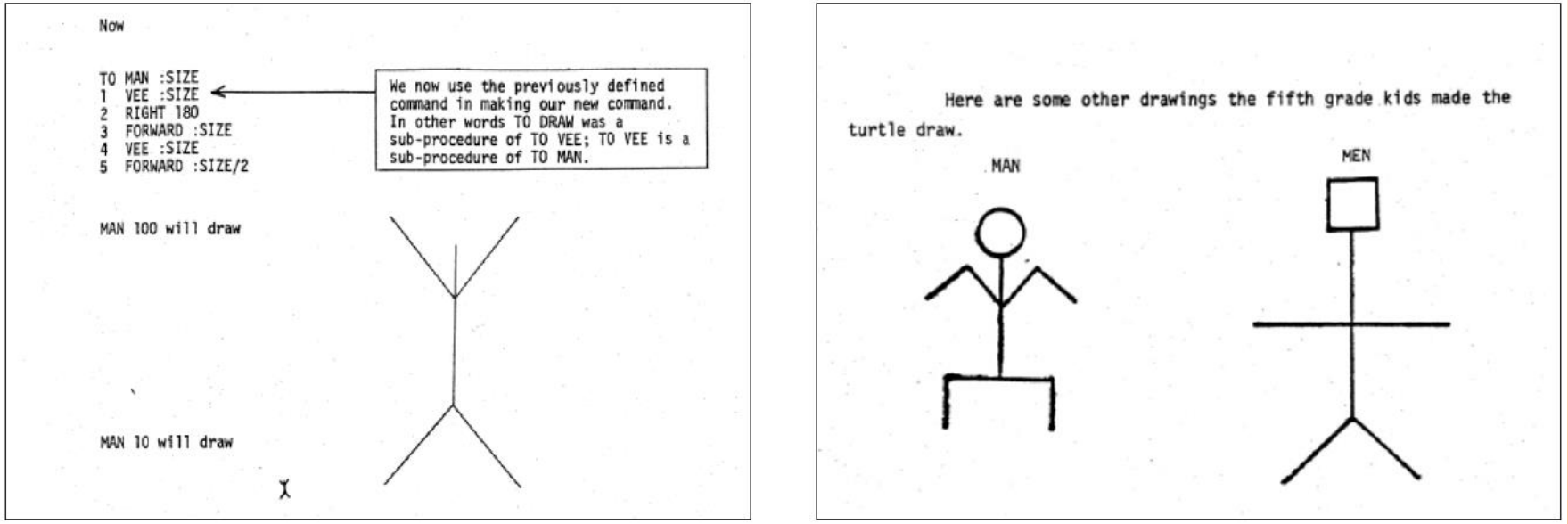


Fig. 1. *Thing #2 of the CS article as proposed by CS (left) and performed by children (right)*

Scene 2 – Geometric human figures

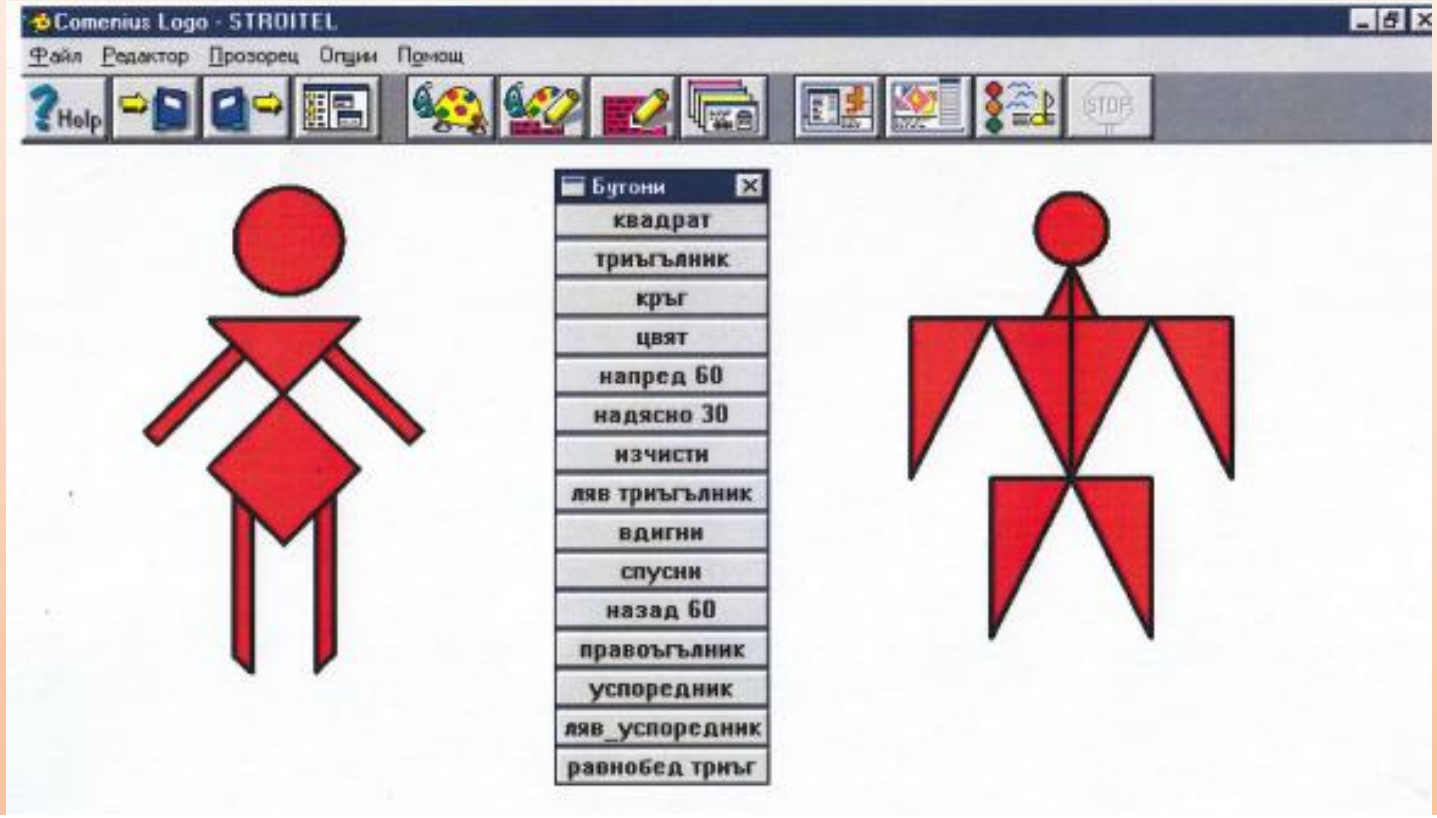


Fig. 2. *Work of 11-year old students in Comenius Logo (the buttons' names are in Bulgarian)*

Scene 3 – Visual modeling à la Pencho Balkanski and Sonia Delaunay

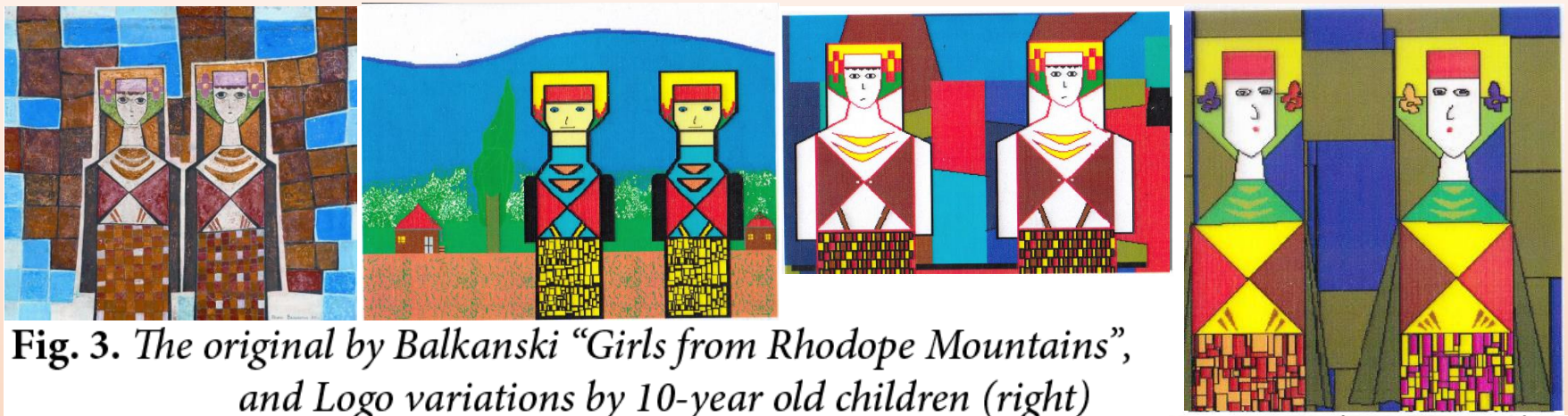


Fig. 3. *The original by Balkanski "Girls from Rhodope Mountains", and Logo variations by 10-year old children (right)*

Fig. 4. *Students' computer variations of Sonia Delaunay's models*

Scene 4 – From one step to 3D animation

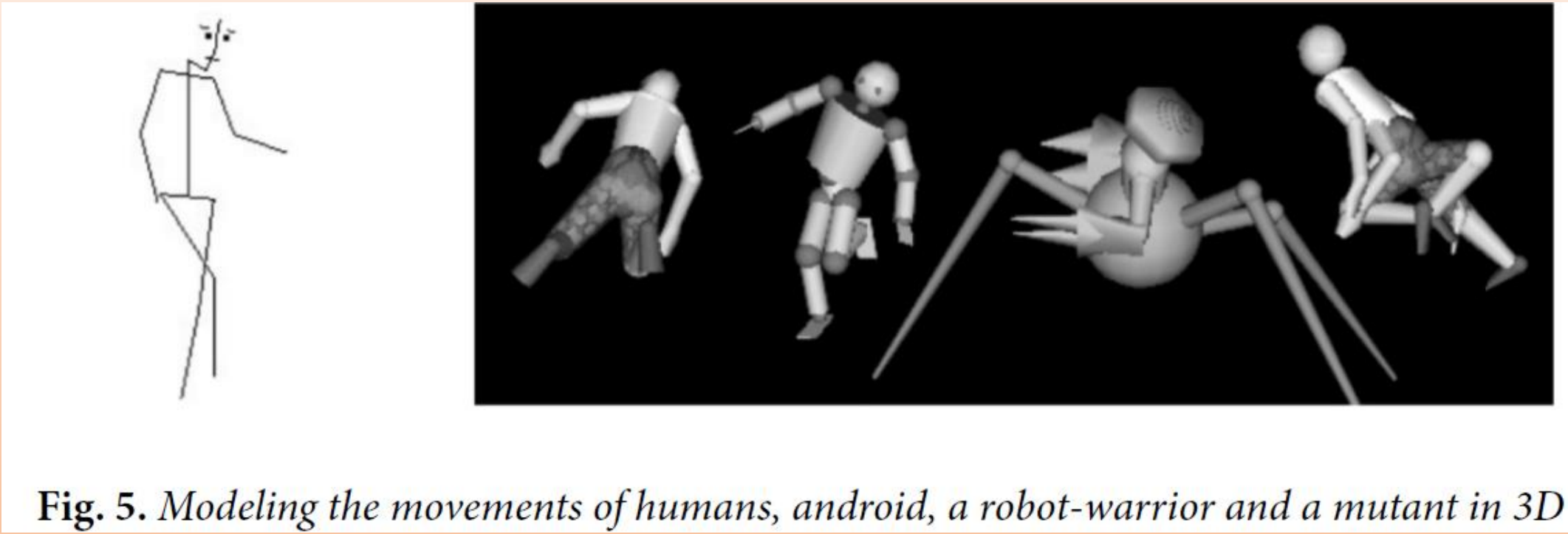


Fig. 5. *Modeling the movements of humans, android, a robot-warrior and a mutant in 3D*

Pavel Boytchev, “Turtle Metamorphoses (from FD 1 to 3D animation)”

Scene 5 – Enter the Cat

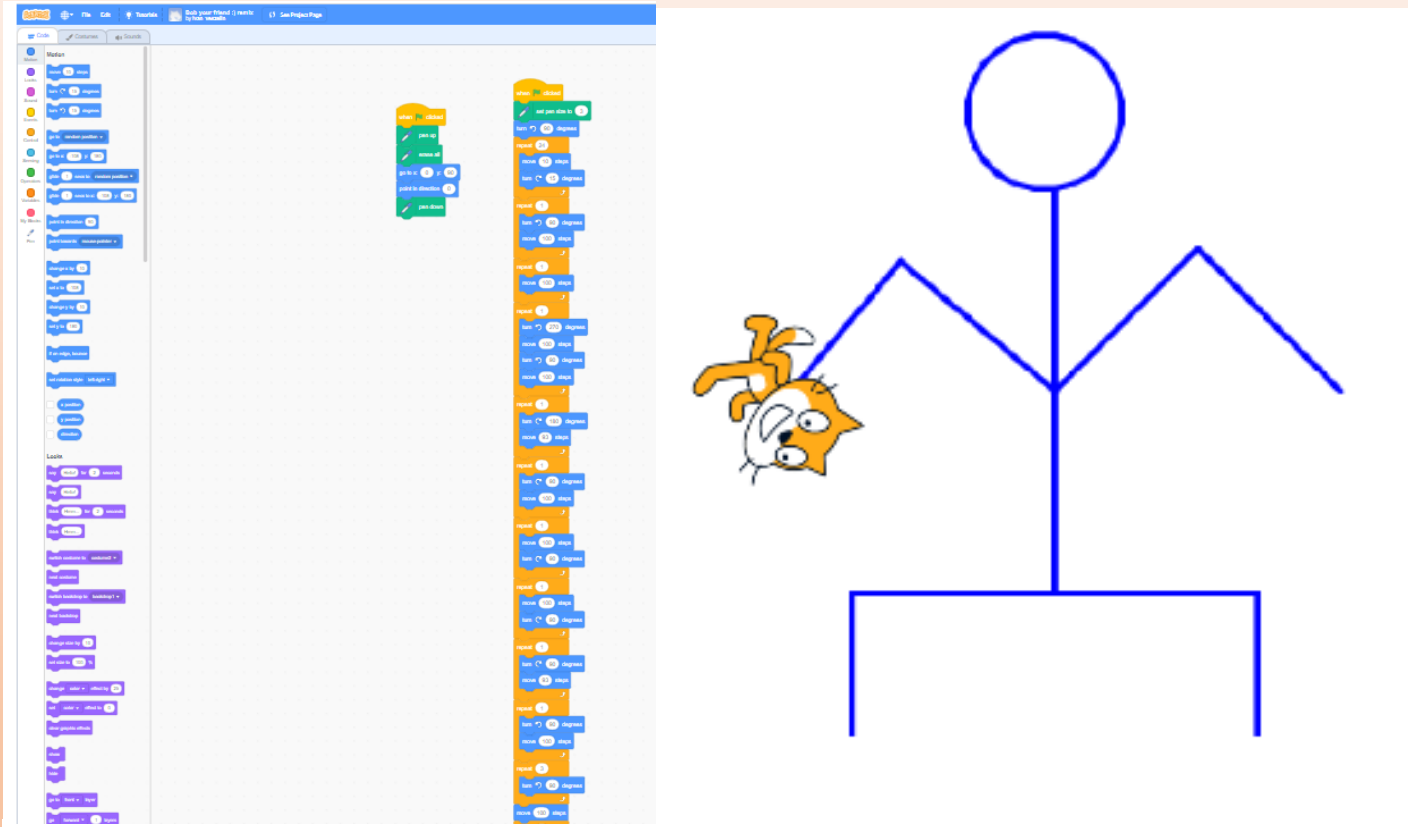


Fig. 6. *The first version of the man's figure using a chain of commands*

Scene 5 – Enter the Cat

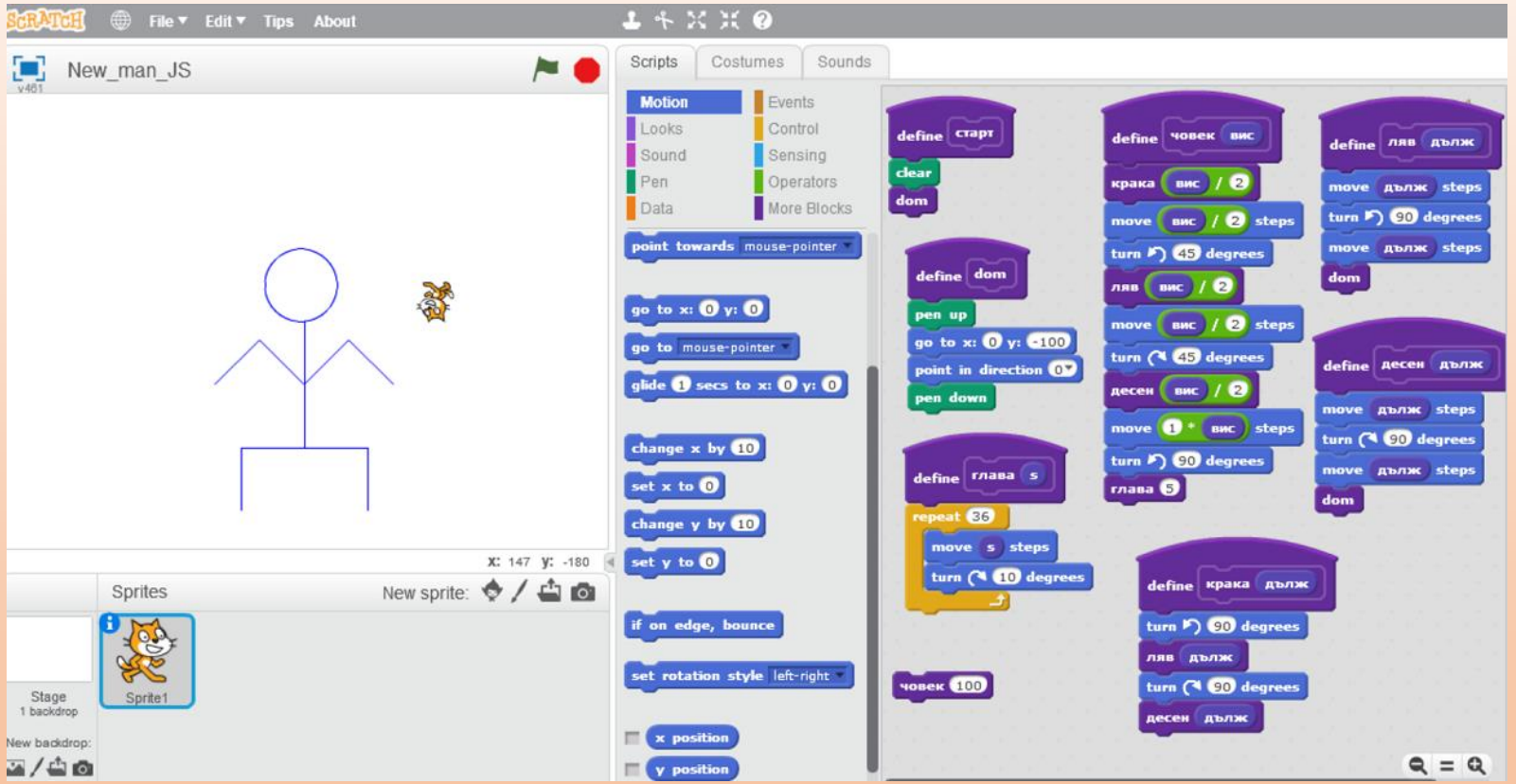


Fig. 7. *A second version of the man's figure above using subprocedures and an input for the size*

Scene 6 – The dance of the triangular village people

Scene 6 – The dance of the triangular village people

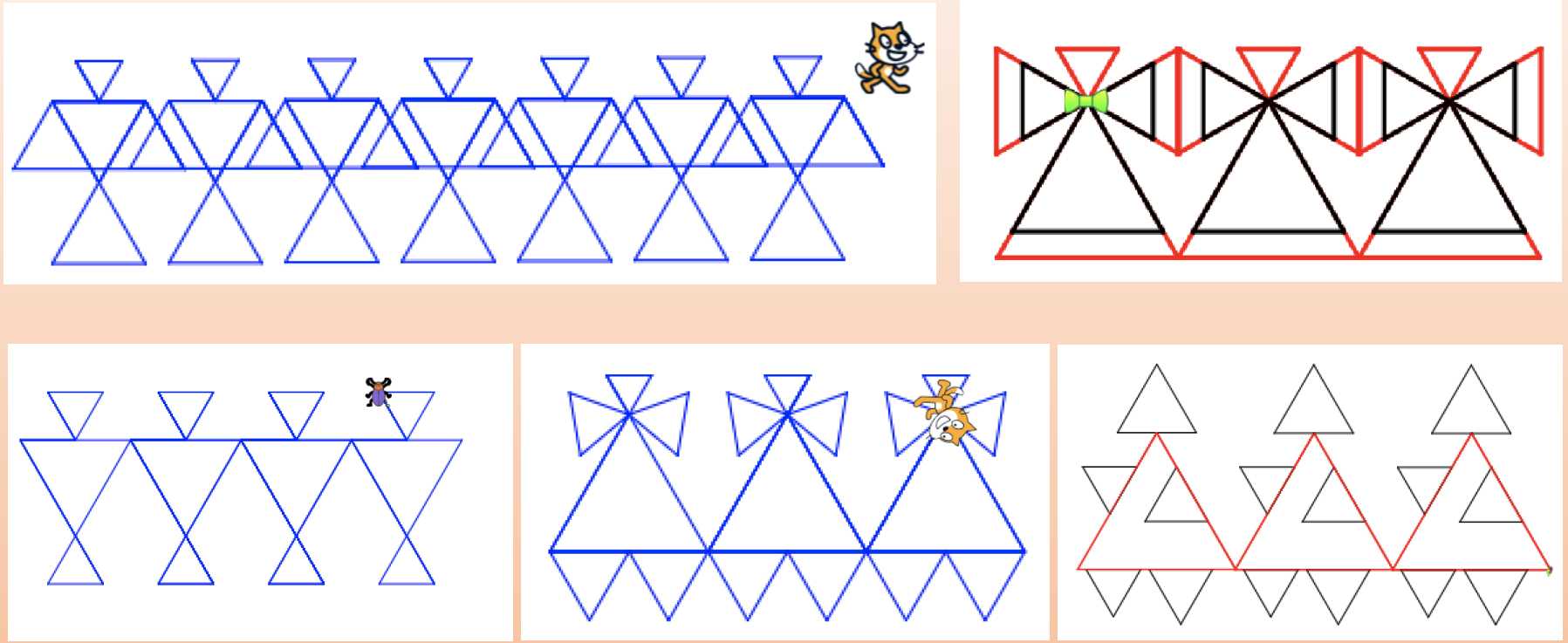
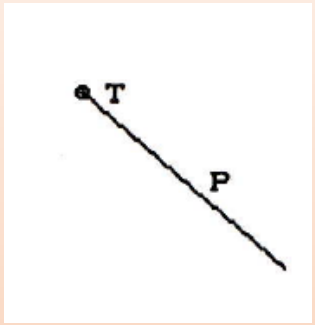


Fig. 8. *Triangular variations of the traditional Bulgarian "horo" dance, by fifth graders from Bulgarsko Shkolo, Sofia*

Intermission – time for a coffee break

But it is not only in the constructing of something that learning becomes truly meaningful for the learner. That creation process and the end product must be shared with others in order for the full effects of Constructionist learning to take root.

Act 2 Scene 1 – Modeling the pendulum in the spirit of Logo (Thing #17)



```
OBJECT "T POINT 0 84
OBJECT "ang 60
MAKE "length 100
OBJECT "V VECTOR 270+:ang :length
OBJECT "P SEGMENT :T :V
```

```
REPEAT 10 [OBJECT "ang :ang-12 WAIT 5]
REPEAT 10 [OBJECT "ang :ang+12 WAIT 5]
```

```
MAKE "angincmax 8
MAKE "angincmin 1
MAKE "maxang normalized RANDOM 360
OBJECT "ang :maxang
OBJECT "V VECTOR 270+:ang 100
OBJECT "P SEGMENT :T :V
MAKE "k (:angincmax – :angincmin)/:maxang
OBJECT "anginc "angincmax - :k* (ABS :ang)
MAKE "d (NEG SIGN :ang)
WHILE NOT ReadKey? [(IF (ABS :ang)>=:maxang[MAKE "d (NEG SIGN :ang])
   OBJECT "ang :ang+:d*:anginc WAIT 3]
```

Fig. 9. *Modeling the pendulum as illustrated in Geomland*

Scene 4 – Pendulum harmonics in 3D

Fig. 10. *Pendulum Harmonics (youtu.be/PyvWjtdL3I4)*

Act 3 Scene 1 – Can a computer play and compose music? (Things #11 and #12)

One (very bad) way to make the computer play Frere Jacques would be to write the following LOGO procedure:

```
TO FJ
1  PRINT "AAA!CCC!EEE!AAA!AAA!CCC!EEE!AAA!EEE!FFF!HHHHHHHH!EEE!FFF!HHHHHHH!..."
END
```

```
TO PENTATONIC
   OUTPUT [277 311 470 415 466]
END

TO DURATIONS
   OUTPUT [100 200 300 400]
END

TO ORIENTAL :N
   IF :N=0 [STOP]
   SOUND LIST PICK PENTATONIC PICK DURATIONS
   ORIENTAL :N-1
END
```

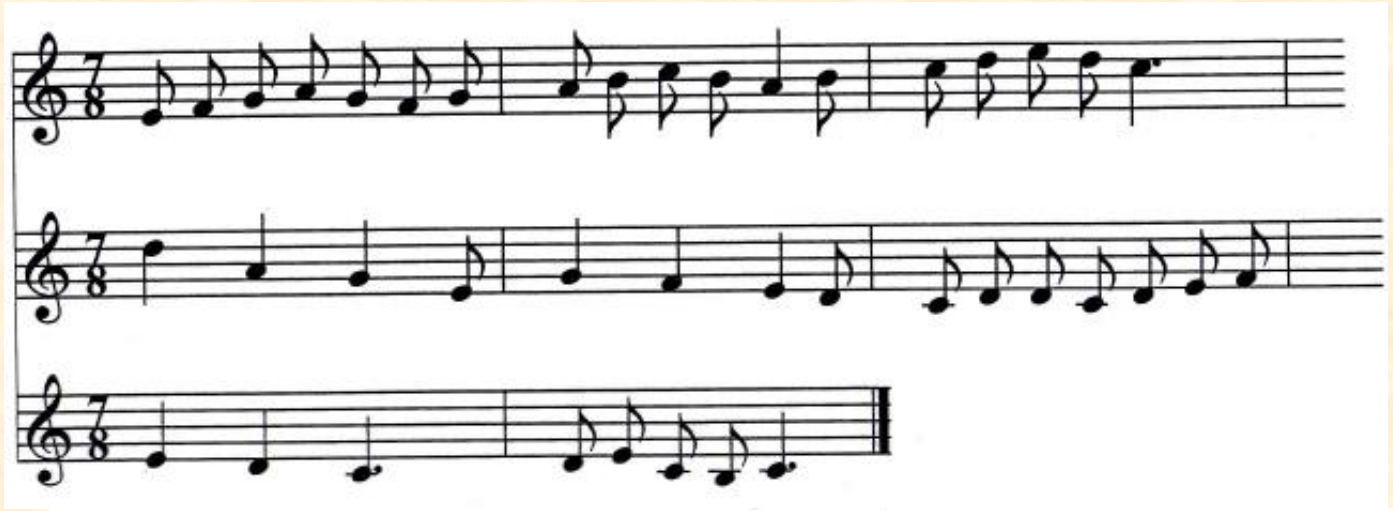


Fig. 11. *Ruchenitsa (Bulgarian folk dance in 7/8) generated by a Logo Writer program*

Scene 2 – Enter a musical structure

```
TO FRERE1
1  SING MUSIC OF "1! 3! 5! 1!"  "2 2 2 2"
END


TO FRERE2
1  SING MUSIC "5! 6! 8!"  "2 2 4"
END


TO FRERE3
1  SING MUSIC '8! 10! 8! 6! 5! 1!"  "1 1 1 1 2 2"
END


TO FRERE4
1  SING MUSIC "1! -8! 1!"  AND  "2 2 4"
END


TO FREREJACQUES
1  FRERE1
2  FRERE1
3  FRERE2
4  FRERE2
5  FRERE3
6  FRERE3
7  FRERE4
8  FRERE4
9  FREREJACQUES
END
```

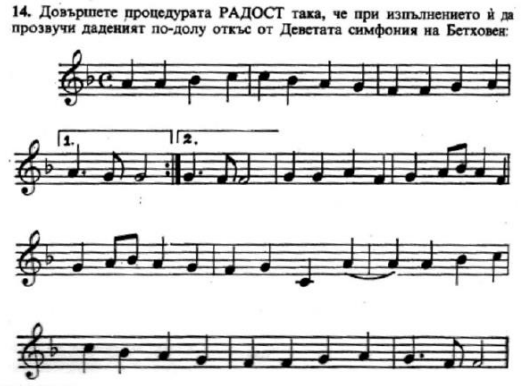


Fig. 13. *The main theme of the Ode to Joy by Beethoven as part of a Logo problem for fifth graders*

```
TO JOY
   THEME1 THEME1.1
   BAR1 BAR1.1 BAR1.2 BAR2
   THEME1.2
END

TO THEME1
   PLAY [A1 A1 B1 C2 C2 B1 A1 G1 F1 F1 G1 A1 A1 G1 G1]~
    [4  4  4  4  4  4  4  4  4  4  4  4  4. 8  2]
END
```

Scene 5 – What does Scratch offer as a marriage partner

Cat: Come on! Why do children need to complicate their lives with structures! See how joyful these programs are that children have done to perform *Ode to Joy* by Beethoven. They can google the notes, they can build scripts with sound blocks that play in sequence (Fig. 12 – left), what more do they need! They can even create multiple scripts which when played together in order, create chords (Fig. 12 – right).

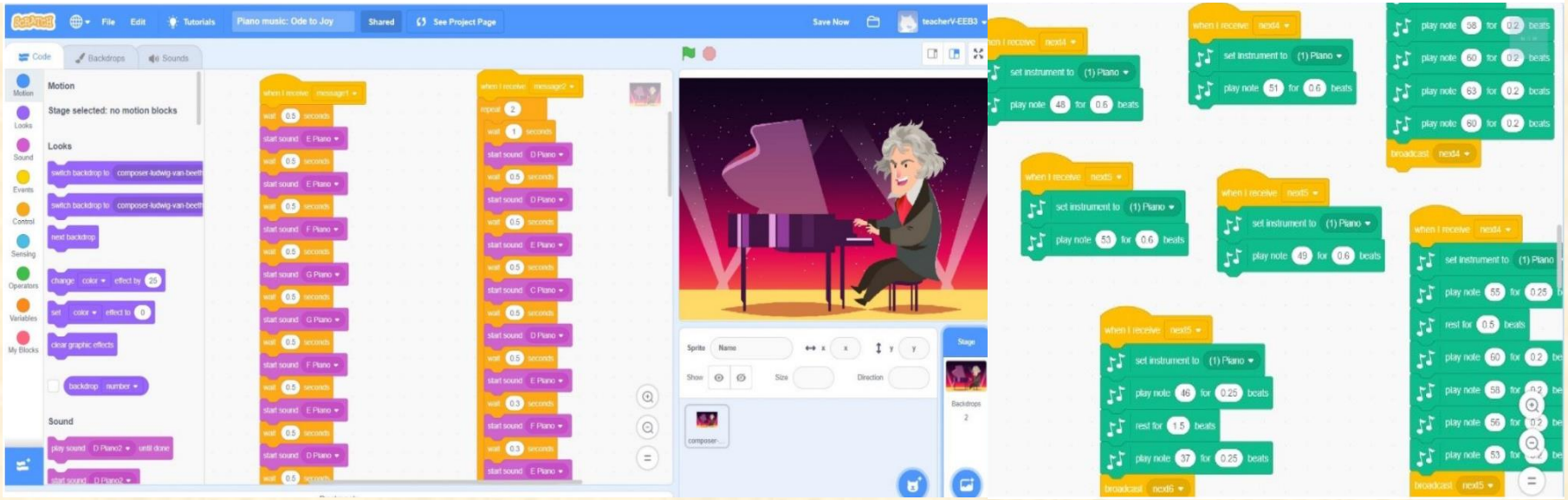
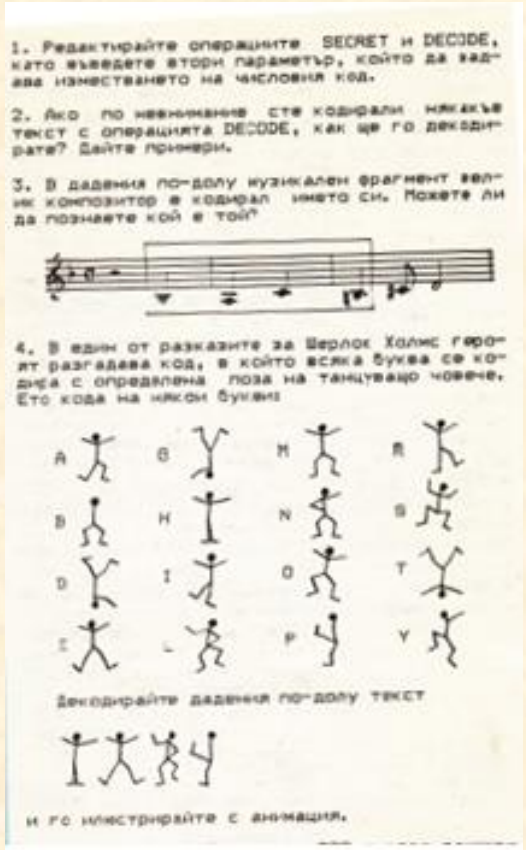


Fig. 12. *Examples of fragments of Ode to Joy in Scratch (Veselinova, 2020)*

Scene 6 – Adding to the intrigue



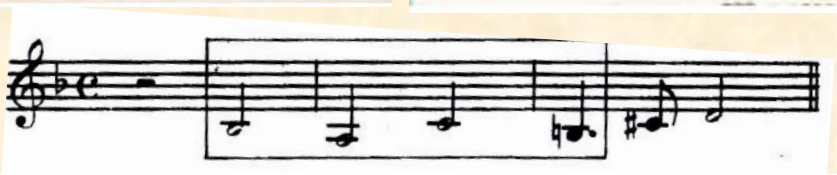


Fig. 14. *Language and mathematics (Logo) for 6th graders [(Nikolov & Sendova, 1984)*

Act 4 Scene 1 – Can the computer learn grammar? (Thing #15)

```
TO PLURAL :NOUN
   OUTPUT WORD :NOUN "S
END
```

```
TO PLURAL :NOUN
   If (LAST :NOUN)="Y [OUTPUT PLURAL_ENDING_IN_Y :NOUN]
   OUTPUT WORD :NOUN "S
END

TO PLURAL_ENDING_IN_Y :NOUN
   IF VOWEL? LAST BUTLAST :NOUN [OUTPUT WORD :NOUN "S]
    [OUTPUT WORD BUTLAST :NOUN "IES]
END

TO VOWEL? :L
   OUTPUT MEMBER? :L [A E I O U]
END
```

Act 4

Scene 3 – Modeling aphorisms

Let me start with Logo model of aphorisms in the style of Georg Lichtenberg:

The horse started resembling a donkey – like a translation from German to Dutch.

Achilles: I get the first one as conveying the idea of how much can be lost in translation—the author compares *pairs* of objects, the first object being similar in a certain sense to the second, but going beyond it.

Turtle: Precisely! We encouraged the students to use an associative list containing the *pair of similar words* from the original aphorism but enriching it with their own *pairs of similar words*, e.g.

[[horse donkey][masterpiece kitsch][wine vinegar][watermelon pumpkin]].

Then they create a procedure generating variations of the original aphorism:

```
TO APHORYSM :LIST1 :LIST2
   MAKE :PAIR1 PICK :LIST1
   MAKE :PAIR2 PICK :LIST2
   OUTPUT (SENTENCE "The FIRST :PAIR1~
    [started resembling]~
    LAST :PAIR1 [- like a translation from]~
    FIRST :PAIR2 "to LAST :PAIR2)
END
```

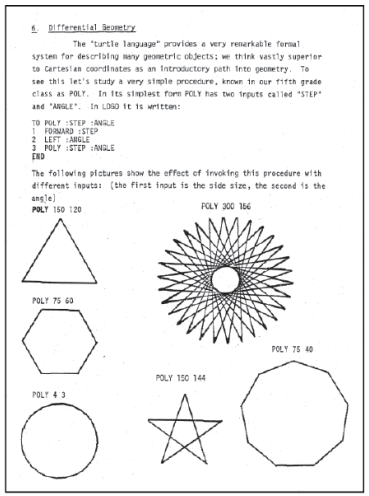
Achilles: Let me try it with appropriate inputs:

```
PRINT APHORYSM [[horse donkey][masterpiece kitsch][wine vinegar]]~
   [German Dutch] [English Greek][Russian Bulgarian]
```

I'll run it twice:

The wine started resembling vinegar – like a translation from Russian to Bulgarian.
The masterpiece started resembling kitsch – like a translation from English to Greek.

Act 5 Scene 1 – Enter recursion: polygons and spirals (Things #6 and #7)



6. Differential Geometry

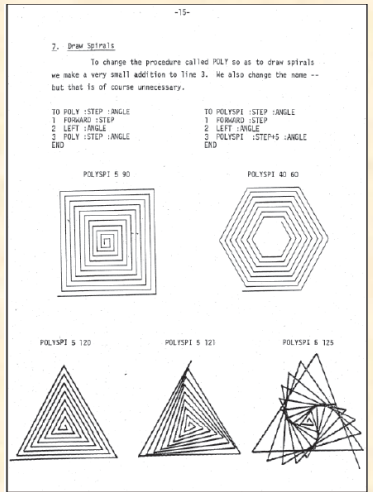
The "turtle language" provides a very remarkable formal system for describing many geometric objects; we think vastly superior to Cartesian coordinates as an introductory path into geometry. To see this let's study a very simple procedure, known in our fifth grade class as POLY. In its simplest form POLY has two inputs called "STEP" and "ANGLE". In LOGO it is written:

```
TO POLY :STEP :ANGLE
1  FORWARD :STEP
2  LEFT :ANGLE
3  POLY :STEP :ANGLE
END
```

The following pictures show the effect of invoking this procedure with different inputs: (the first input is the side size, the second is the angle)

POLY 150 120 · POLY 300 166 · POLY 75 60 · POLY 75 40 · POLY 150 144 · POLY 4 3

```
TO POLY :STEP :ANGLE
1  FORWARD :STEP
2  LEFT :ANGLE
3  POLY :STEP :ANGLE
END
```



-15-

7. Draw Spirals

To change the procedure called POLY so as to draw spirals we make a very small addition to line 3. We also change the name -- but that is of course unnecessary.

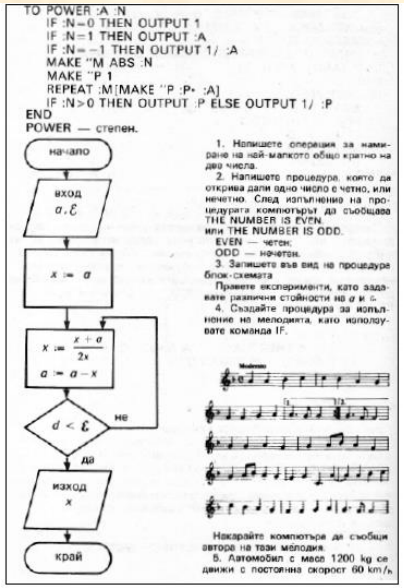
```
TO POLY :STEP :ANGLE
1  FORWARD :STEP
2  LEFT :ANGLE
3  POLY :STEP :ANGLE
END
```

```
TO POLYSPI :STEP :ANGLE
1  FORWARD :STEP
2  LEFT :ANGLE
3  POLYSPI :STEP+5 :ANGLE
END
```

POLYSPI 5 90 · POLYSPI 40 60 · POLYSPI 5 120 · POLYSPI 5 121 · POLYSPI 6 125

```
TO POLY :STEP :ANGLE
1  FORWARD :STEP
2  LEFT :ANGLE
3  POLY :STEP :ANGLE
END
```

```
TO POLY :STEP :ANGLE
1  FORWARD :STEP
2  LEFT :ANGLE
3  POLY :STEP+5 :ANGLE
END
```



```
TO POWER :A :N
  IF :N=0 THEN OUTPUT 1
  IF :N=1 THEN OUTPUT :A
  IF :N=-1 THEN OUTPUT 1/ :A
  MAKE "M ABS :N
  MAKE "P 1
  REPEAT :M[MAKE "P :P* :A]
  IF :N>0 THEN OUTPUT :P ELSE OUTPUT 1/ :P
END
```

POWER — степен.

начало · вход a, ε · $x := a$ · $x := \frac{x+a}{2x}$ · $a := a - x$ · $d < \varepsilon$ · не · да · изход x · край

1. Напишете операция за намиране на най-малкото общо кратно на две числа.
2. Напишете процедура, която да открива дали едно число е четно, или нечетно. След изпълнение на процедурата компютърът да съобщава THE NUMBER IS EVEN, или THE NUMBER IS ODD.
EVEN — четен.
ODD — нечетен.
3. Запишете във вид на процедура блок-схемата.
Правете експерименти, като задавате различни стойности на a и ε.
4. Създайте процедура за изпълнение на мелодията, като използувате командата IF.

Накарайте компютъра да съобщи автора на тази мелодия.
5. Автомобил с маса 1200 kg се движи с постоянна скорост 60 km/h

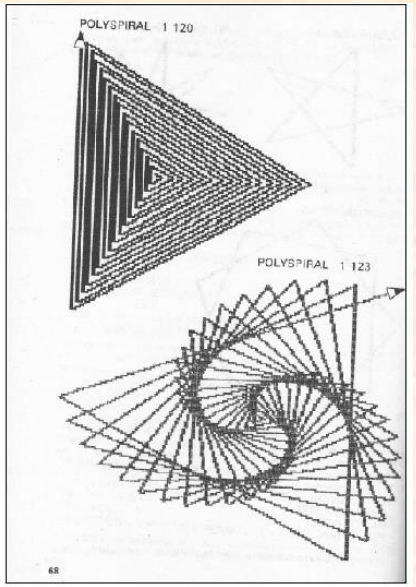
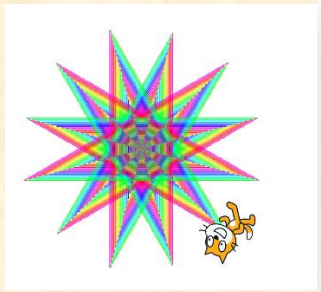
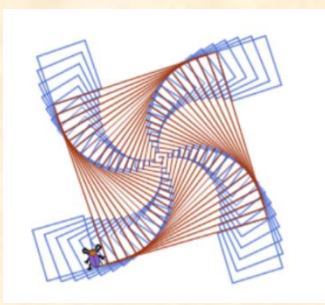
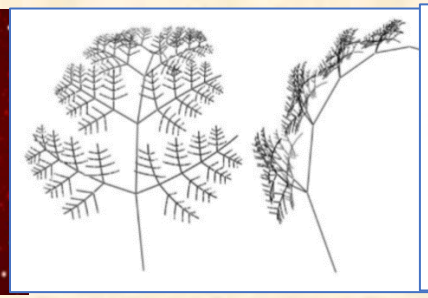


Fig. 15. *The first Logo book in Bulgarian (Nikolov, 1983)*

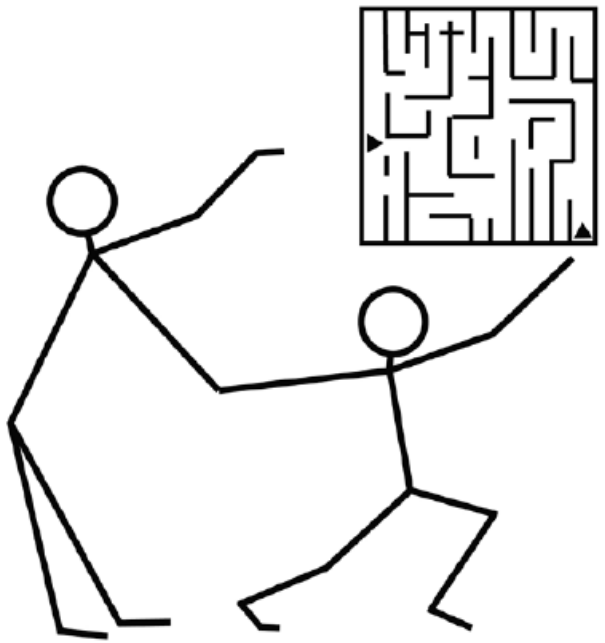






Scene 2 – Ten rules for creative teaching

1



You should not lead children to predetermined solutions. (In informatics there are often no right answers but rather right paths.)

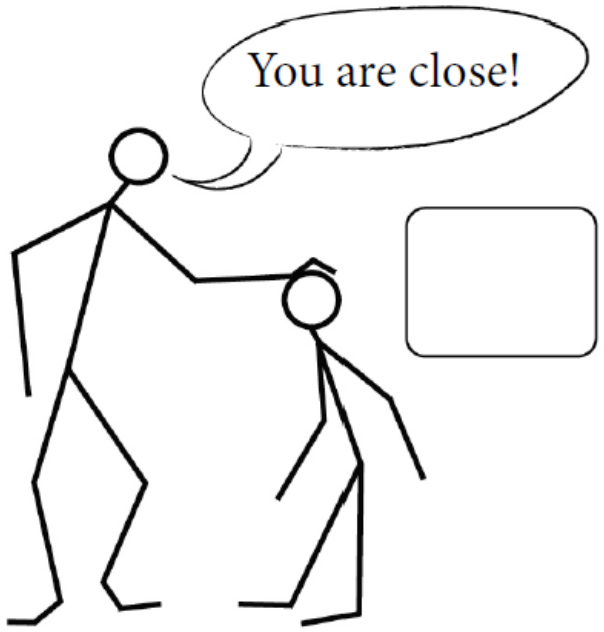
2

You should not deprive children of the pleasure of solving problems by themselves.

The cartoons are drawn originally by Todor Kolarov (1987) and remade by Yovko Kolarov.

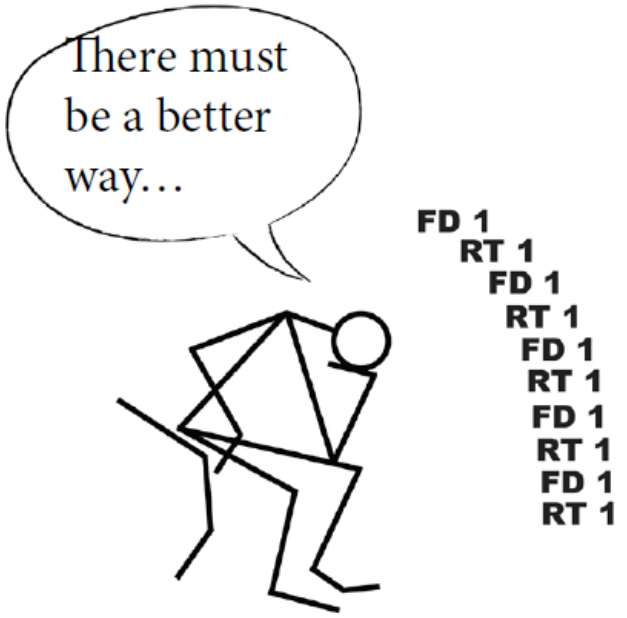
Scene 2 – Ten rules for creative teaching

3



You should not let children become discouraged.

4



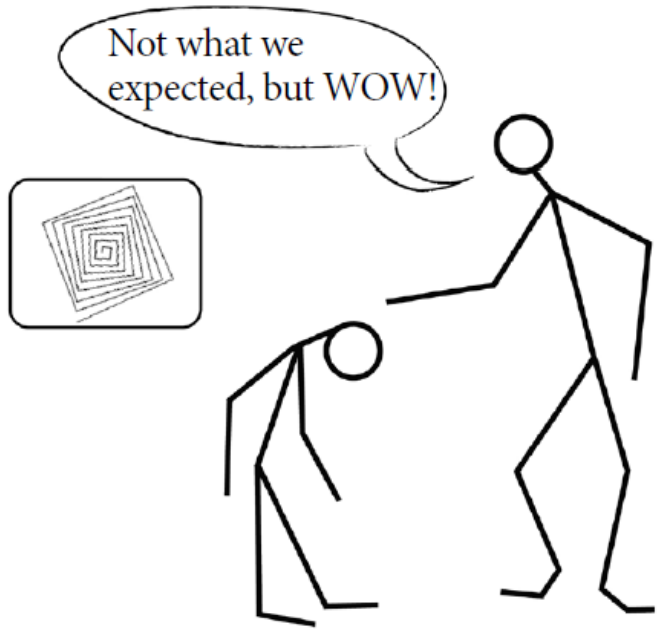
You should not show a Logo facility that is new to chidlren unless they see a need for it. (Or, as others put it, programming should not be a goal, but a tool.)

Scene 2 – Ten rules for creative teaching

5

You should not be afraid of cooperation with children as equal partners and certainly not afraid of learning from them.

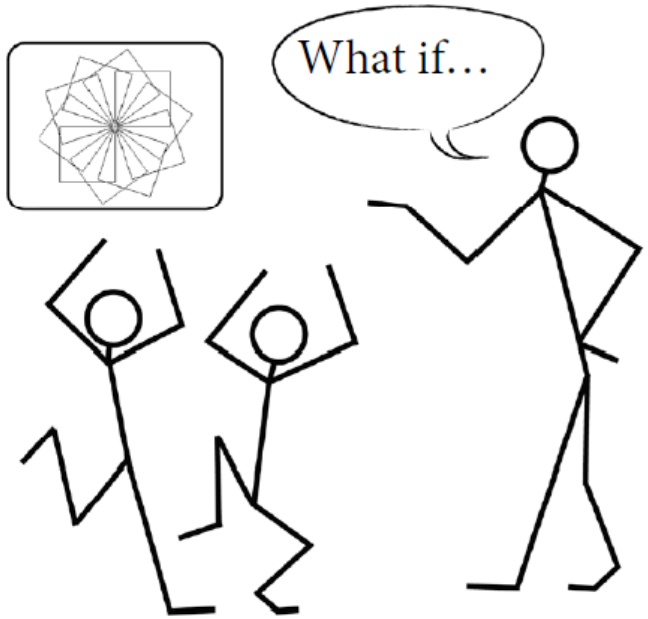
6



You should not automatically reject an unexpected result obtained in the process of programming but rather analyze and explore it.

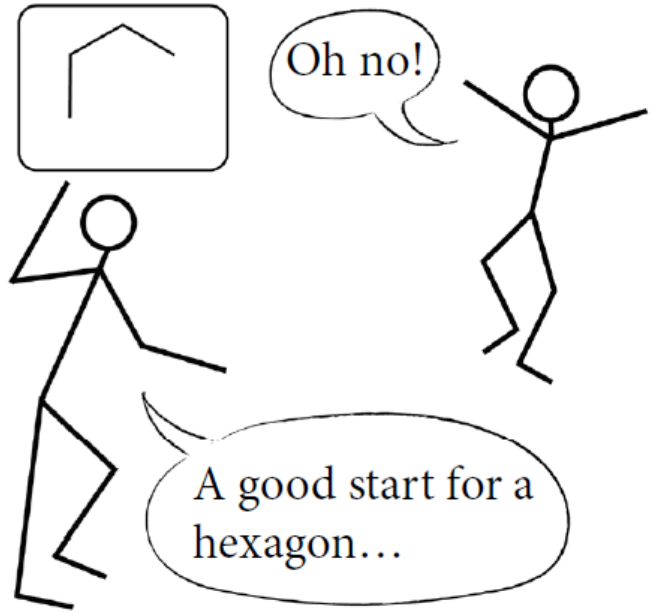
Scene 2 – Ten rules for creative teaching

7



You should not think of programs as being right or wrong but rather as artifacts that could be developed and improved.

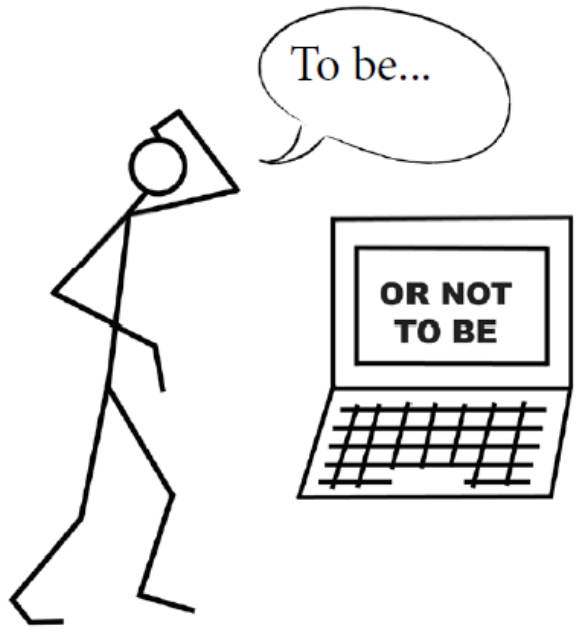
8



You should not let students be afraid of making mistakes. Moreover, it may well be that the approach of debugging is one of the most profound educational ideas of twentieth century.

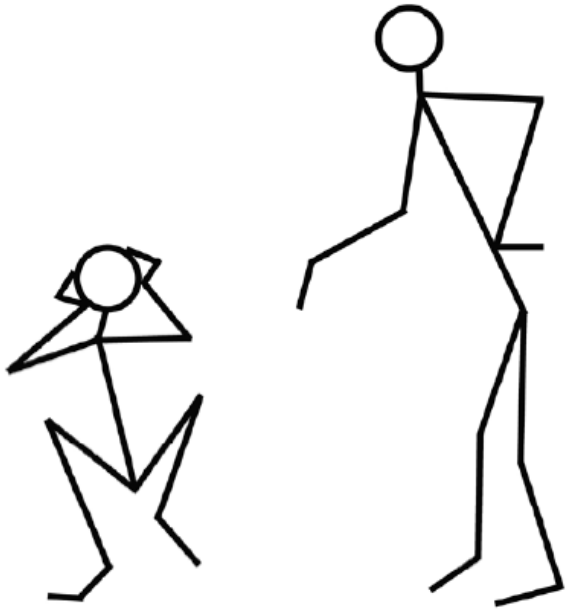
Scene 2 – Ten rules for creative teaching

9



As a teacher you should not be afraid of making mistakes yourself. (Sometimes it pays to allow mistakes to slip in, so as to show how to cope with such situations.)

10



Honor the natural wish of children to learn rather than to be taught.

Words of thanks

Acknowledgments to Gary Stager for the invitation and the great initiative, to Cynthia Solomon for being such an inspiration during the years with all her activities, and to Artemis Papert for sharing pictures and her love for turtle art. Special thanks go to Sylvia Martinez for her editorial help in making this essay sound more like a “horse” than a “donkey.”

Words of thanks

My deep appreciation goes to the international Logo community for keeping the Logo spirit alive through all these years with the Eurologo and Constructionism conferences and related books and journals.

My deep appreciation goes to the international Logo community for keeping the Logo spirit alive through all these years with the Eurologo and Constructionism conferences and related books and journals.

EURO LOGO 99

Sofia
Bulgaria
22 – 25 August 1999

Words of thanks

My special gratitude to the pillars of the Bulgarian Logo culture—Roumen Nikolov, fo creating the first Logo text book and the first Logo version in Bulgarian; Bojidar Sendov fo designing and leading the development of Geomland as a mathematical laboratory for explo ration in Euclidian geometry in a Logo spirit; Darina Dicheva for co-authoring a number o informatics textbooks in Logo style; Iliana Nikolova for leading a Logo center at the Facult of Mathematics and Informatics at Sofia University and a project for adapting Comenius Log for the Bulgarian Schools; Ivaylo Ivanov and Vesela Ilieva for developing educational resource for the primary school in Bulgaria; and finally to Pavel Boytchev who created Elica (a 3D Log environment), the Logo tree project, and a whole dynamic ecosystem in which versions o Logo emerge and communicate with a variety of programming languages, virtual educationa environments, educational video-clips, games, films, and exhibitions with artistic value. It wa his idea to present this dialog like a play with acts and scenes.

Благодаря!

The biggest THANK YOU goes to the numerous teachers and students who proved that th ***can*** do twenty things (and many more) with a computer thanks to a teacher and friend such Seymour Papert—a genius and dreamer with the curiosity of a child.