Integrating Programming into the Mathematics Curriculum: Combining Scratch and Geometry in Grades 6 and 7

Klaus-Tycho Förster

SIGITE 2016 Boston, MA, USA — September 30

Algorithms – in Class

- Algorithms - a fundamental competence for teaching mathematics and computer science
  (A. Engels, M. Klika / U.-P. Tietze / H. Wolpers, H.-G. Weigand, W. Herget, A. Schwill, S. Schubert, E. Modrow, … …)

- Algorithms – relevant everyday and across disciplines!
  (B. Schmidt-Thieme 2005)

Algorithms – and Programming

- Useful for checking the correctness of algorithms and their formalization

- Conducive for criticism of the (correct?) result

- Important for the review and evaluation of more complex modeling

Programming – in Mathematics

- Feurzeig et al. (1969):
  - Students often have difficulties to talk about mathematical problem solving (especially about the process afterwards) or to gather first experiences in it
  - *“Programs are more discussable than traditional mathematical activities: one can talk about their structure, one can talk about their development, their relation to one another, and to the original problem.”*

Programming – What is a Program?

- Transformation from Input to Output
  - IPO principle
  - Analogous: Function with domain and codomain

- What sort of task is programming?

- Math. tasks consist of 3 components (cf. Bruder 2008)
  - What is given
  - Transformation
  - Result

Programming - Characterization

Proof tasks →

Gegebenes	Transformationen	Gesuchtes	Bezeichnung des Aufgabentyps	Beispielaufgabe
×	×	×	gelöste Aufgabe, Musteraufgabe, Aufgabe zur Fehlersuche	– Stimmt das?… – Wo steckt der Fehler?
				Zahl soll bestimmt werden.
×	–	×	Beweisaufgabe, Spielstrategie finden	– Beim Nimm-Spiel gewinnt Frank immer. Wie macht er das? *Es liegen 20 Streichhölzer auf dem Tisch. Zwei Spieler spielen gegeneinander. Gewonnen hat derjenige, der das letzte Streichholz nehmen kann, wenn entweder ein, zwei oder drei Hölzer pro Zug genommen werden dürfen.* – Warum ist die *p*-*q*-Formel zur Lösung quadratischer Gleichungen immer richtig?
			erfinden	
–	–	–	Problemsituation mit offenem Ausgang (Trichtermodell)	Führe eine Befragung zu einem gegebenen Thema bei deinen Mitschülern durch und stelle die Ergebnisse vor.

From: Bruder, Leuders, Büchter 2008; Markup from Strecker 2009

- *Finding a transformation which maps input values to output values, or finding an algorithm, can be compared with a proof task in mathematics. (cf. Strecker 2009)*

Proofs – In Mathematics

N = 833	USA	Deutschland	Japan
Arithmetik			
verallgemeinerte Bestimmungsaufgaben	34 (42 %)	19 (66 %)	
verallgemeinerte Grundaufgaben	46 (57 %)	10 (34 %)	
Beweisaufgaben			
Reflexionsaufgaben	1 (1 %)		
gesamt (Arithmetik)	81 (100 %)	29 (100 %)	
Algebra			
verallgemeinerte Bestimmungsaufgaben	147 (80 %)	115 (93 %)	33 (69 %)
verallgemeinerte Grundaufgaben	36 (20 %)	6 (5 %)	13 (27 %)
Beweisaufgaben		1 (1 %)	
Reflexionsaufgaben		1 (1 %)	2 (4 %)
gesamt (Algebra)	183 (100 %)	123 (100 %)	48 (100 %)
Geometrie			
verallgemeinerte Bestimmungsaufgaben	146 (74 %)	80 (70 %)	25 (43 %)
verallgemeinerte Grundaufgaben	48 (24 %)	30 (26 %)	2 (3 %)
Beweisaufgaben		1 (1 %)	23 (40 %)
Reflexionsaufgaben	2 (1 %)	4 (4 %)	8 (14 %)
gesamt (Geometrie)	196 (100 %)	115 (100 %)	58 (100 %)
gesamt	460	267	106

Distribution of tasks in a TIMSS video study in grade 8.

From Neubrand 2002; Markup from Strecker 2009

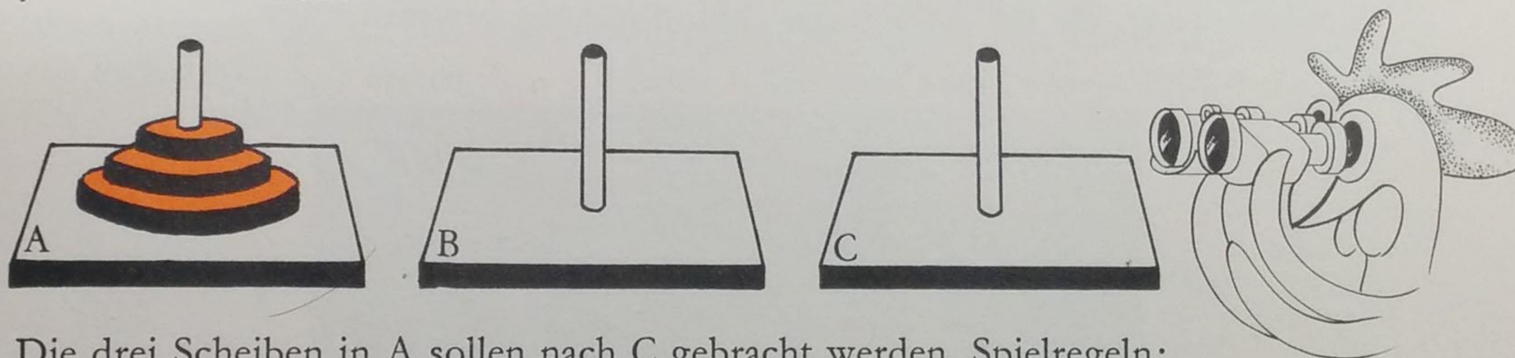
Programming – in Mathematics

- In the 1980s, German math books often had programs in Basic for, e.g., the approximation of square roots or for values of binomial coefficients (cf. Oldenburg 2011)

- Often w. other programming languages as well

Let’s go even further back..

5. ‚Turm von Hanoi‘.



Die drei Scheiben in A sollen nach C gebracht werden. Spielregeln:
a) Es darf jedesmal nur eine Scheibe bewegt werden.
b) Eine Scheibe darf nicht zweimal nacheinander bewegt werden.
c) Eine größere Scheibe darf nie auf einer kleineren liegen.
d) Eine Scheibe darf nur bei A, B oder C abgelegt werden.

Wie viele Züge braucht man mindestens? Und bei vier Scheiben?
(Als Scheiben kannst du Münzen nehmen, anstelle der Bretter A, B, C einen Bogen Papier, auf dem die drei Punkte A, B, C markiert sind.)

From: PLUS, mathematisches Unterrichtswerk, **1976** (grade **7**)
Editors: J. Schönbeck & H. Schupp

But Today?

- Programming in mathematics: Replaced by
  - CAS (computer algebra systems)
  - Interactive geometry software
  - Spreadsheet software

- Nearly no programming in mathematics education today (cf. Oldenburg 2011)

Algorithms and Programming!

- Should programming be part of mathematics education? Of course! (cf. Kortenkamp 2005)

- Concepts such as *loops*, *procedures*, and especially *variables*, are central in the learning of mathematics and offer many educational opportunities (cf. Kortenkamp 2005)

Tried and Trusted: Logo/Turtle-Graphics

- Logo: "Didactical" programming language (67)

- Extension: Turtle-Graphics (S. Papert)
  - Programmable turtle with a pen

- E.g., REPEAT 4 [FORWARD 100 RIGHT 90]
  - Results in a square

Tried and Trusted: Logo/Turtle-Graphics

http://projects.csail.mit.edu/films/aifilms/sources.php
http://www.youtube.com/watch?v=xMzojQFyMo0

Tried and Trusted – in Mathematics?

- Controversial in the 1980s (Weigand 1989)
  - P. Bender (1987): *Critique of Logo philosophy*.
    In: JMD 8 (1987), No. 1/2, pp. **3—103**
  - J. Ziegenbalg *Comments on the „Critique of Logo philosophy“*
    In: JMD 8 (1987), No. 8, pp. 305—313
- Fuchs 1996: Programming in mathematics is just for an elite circle with knowledge of the specific syntax

Tried and Trusted – in Geometry!

- However, the formation of the angle concept could be supported (Bender, *Contra Logo* 1986)
- Various empirical studies have shown the educational value of Logo (Clements et al. 2008)
- *„The good old idea of Turtle graphics still has an enormous potential."* (Oldenburg, Rabel und Schuster 2012)
- A high degree of interaction with geometry lessons can be easily reached (Hromkovič 2012)

Programming – How?

- More hours for programming?
  - Cuts for which school subjects?
  - Longer hours for children?

- Changing curriculums?
  - Again, what to remove?
  - Already too many things anyway?

Integrated into the Curriculum!

- Programming not for the sake of programming
- But for its multitude of possibilities, e.g.,
  - Problem solving, proofs, formalism, being able to criticize, modeling, modularization
- Where it is useful!
  - By using a programming language whenever it can be helpful for the students (cf. Kortenkamp 2005)

“Ancient” Demands…

- Propaedeutic of Algorithms
  - Algorithmic behaviors in everyday life and in the teaching of mathematics

- Possible topics include, e.g.,
  - Numerical methods
  - Simulations
  - Modeling
  - Geometry
  - Statistics

Programming – Difficult?

+ No (just have to know how…)
+ Just get used to a small “vocabulary”

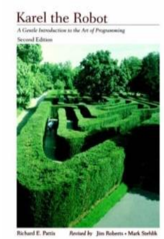
- Bracket problems…
- Syntax checking
- Cryptic error messages

- **New approaches**

Some examples

Etoys & Squeak 1996

Logo 1967

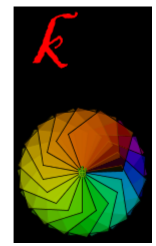


Karel the Robot 1981

Alice 1999

Kara 2000

Agentsheets 1991



Kojo 2010

BlueJ 1999

Turbo Pascal 1983

Delphi 1995

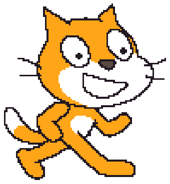
Lego Mindstorms 1998

Greenfoot 2006

Hypercard 1987

Scratch

- Visual programming language
- Released 2007
- Developed by the Lifelong Kindergarten research group at the MIT Media Lab
- Freely available for Windows, MacOS, Linux
- Source code available for non-commercial development



Scratch – Also established in Math

- Building upon and enriching grade four mathematics standards with programming curriculum
  (C. M. Lewis & N. Shah, SIGCSE 2012)
- Designing interactive activities within Scratch 2.0 for improving abilities to identify numerical sequences
  (L. Zavala et. Al., IDC 2013)
- A case study on mathematics learning during design and computing
  (F. Ke, Computers & Education 73, 2014)
- And many more..

COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

11/09 VOL.52 NO.11

Scratch
Programming for All

Communications Surveillance

An Interview with Ping Fu

Usable Security: How To Get It

E-Paper's Next Chapter

Turing Lecture
by Edmund M. Clarke, E. Allen Emerson, and Joseph Sifakis

Association for Computing Machinery

Scratch – Worldwide!

- Over 16 million programs at http://scratch.mit.edu/
  - Mostly children from 8 to 16
  - But also adults of all ages
- Available in roughly 50 languages
- Many extensions (e.g., Snap!)
- Also used at universities such as Harvard, Berkeley…
- As well as in some matriculation examinations

Visual Programming

- Instead of

```
for ( int i=0; i<10; i++ )
{
        ...
}
```

- Simply
  - Drag'n'Drop

Visual Programming – Hello World!

- *public class Hello*

  {

  public static void main (String argv[])

  {

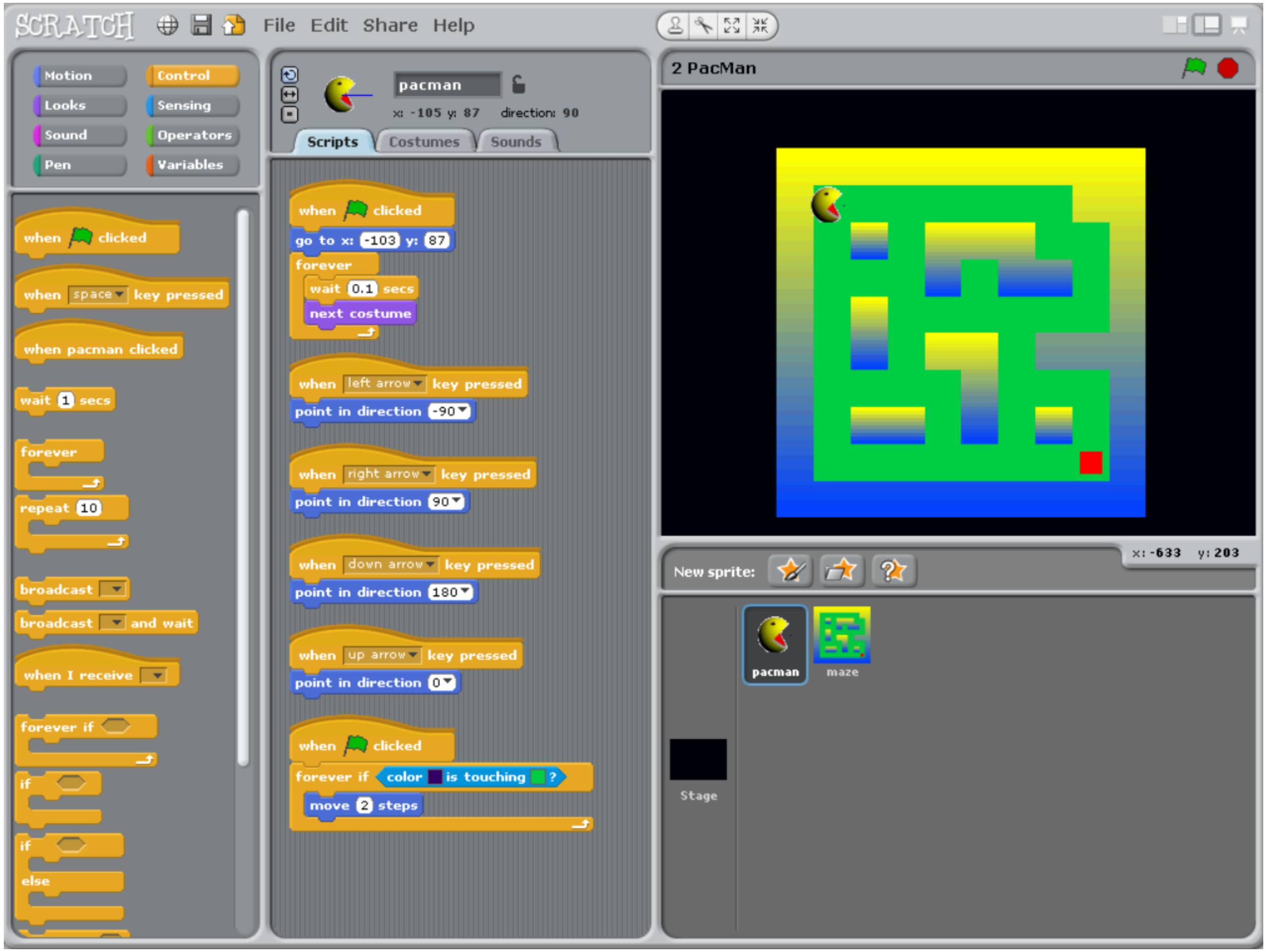
  System.out.println(„Hello World");

  }

  }

- 

Interface of Scratch



Advantages of Visual Programming

- Commands are self-explanatory
- No syntax errors
- Mistakes are often “visible”
- Easy to get first programs working

Integrated Approach - Geometry

- Describing a geometrical construction is nothing less than giving an appropriate algorithm for it (Holland, 1974)
- Only describing the construction algo-rithmically enforces checking the correctness of every single step of the construction (Holland, 1973)

Describing Geometric Constructions

According to Riemer 2011:

- Akin to chalk and cheese between what
    - (i) students write when describing a construction, and
    - (ii) mathematically correct solutions

- Early digital thinking in this area leads to fascinating new possibilities

Describing Geometric Constructions

- Strong improvements in the students' descriptive skills when the construction is performed in the computer first (Weigand and Weth 2002)
- The final form of the description of a geometric construction is the algorithm, which then can be translated to a computer language (Schmidt-Thieme 2009)
- Allows joint usage of geometry and variables

Integrated Approach – Variables?

- Concept of variables is one the more difficult aspects of didactics in computer science
- However: The understanding of variables in both the worlds of math and computer science could be beneficial (cf. Serafini 2011)
- In Germany: Variables formally introduced mostly not before grade 7 in mathematics

Overview of the Study

- Grade 6: regular polygons and tessellations
- Grade 7: (describing) triangle constructions
- Grade 9: long-term evaluation

Overview of Grade 6 Lessons

- Short introduction to Scratch
  - Especially simple Turtle-graphics & loops
- Construction of regular triangles
- Construction of regular polygons
  - No variables
- Tessellations
    - Regular tilings: Triangles, squares, hexagons

Overview of Grade 7 Lessons

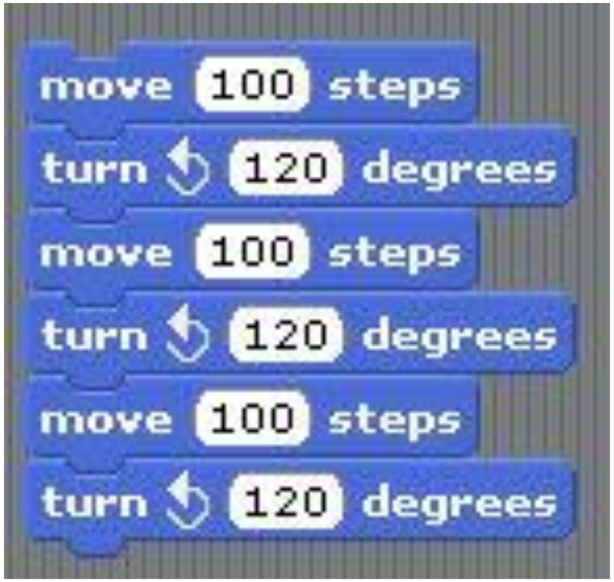
- Congruent triangle constructions w. Scratch
  - Regular triangle (introduction)
  - SAS: side-angle-side
  - ASA: angle-side-angle (or SAA or AAS)
  - Parametrized triangle construction

- Remaining types SSS & SSA:
  - With pen and paper and dynamic geometry softw.
  - (both require intersection of cycles)

Classroom Study

- High school in northern Germany
- One class over multiple years
  - Grade 6: 15 girls, 13 boys
  - Grade 7: 12 girls, 12 boys
  - Grade 9: 12 girls, 12 boys
- Three further classes for comparison

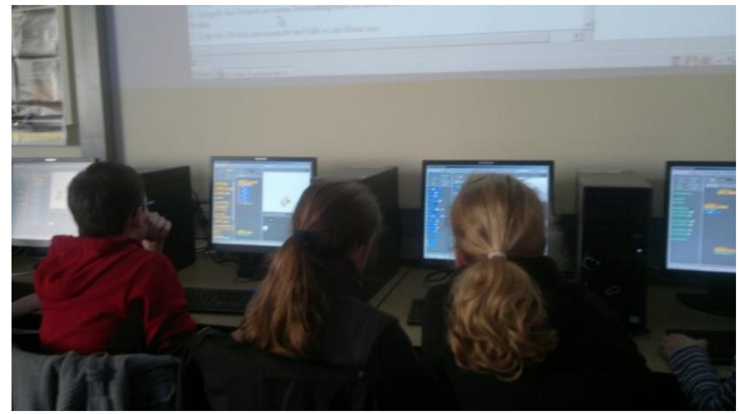
Regular Triangles



Blocks to create a regular triangle, step by step.

The same construction, but now with a loop.

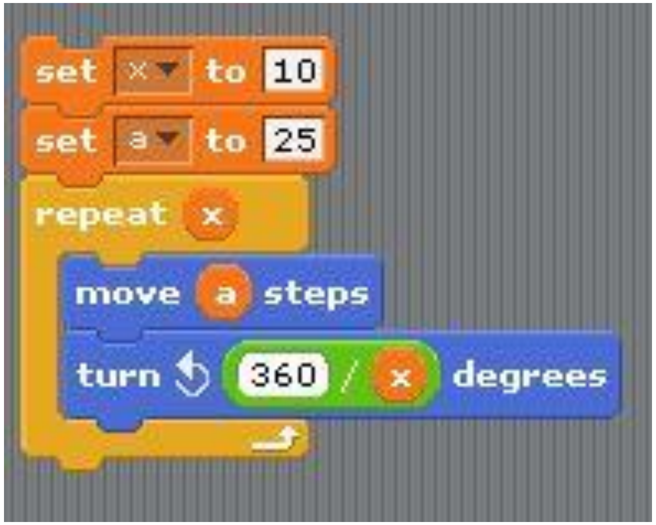
Regular Polygons



To construct regular polygons with a differing number of corners, one would replace the 10s with the desired number.

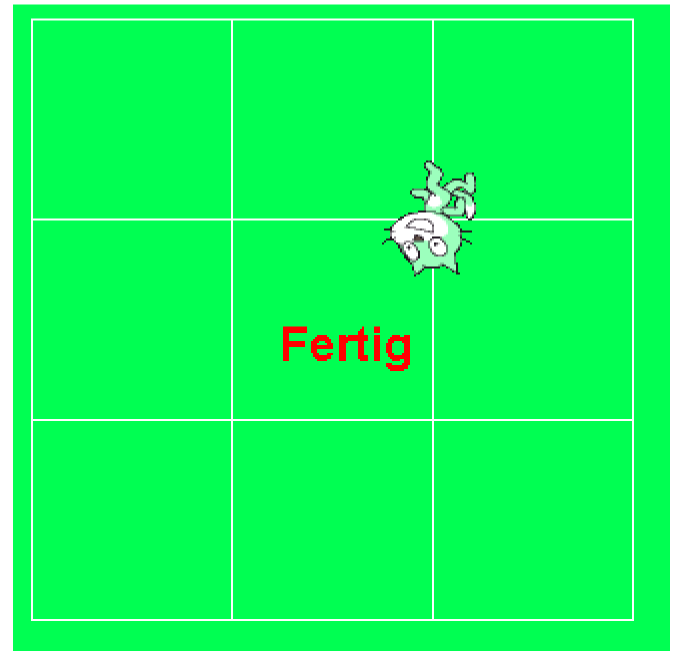
Regular Polygons

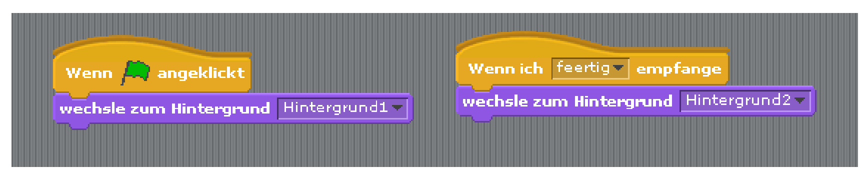
To construct regular polygons with a differing number of corners, one would replace the 10s with the desired number.



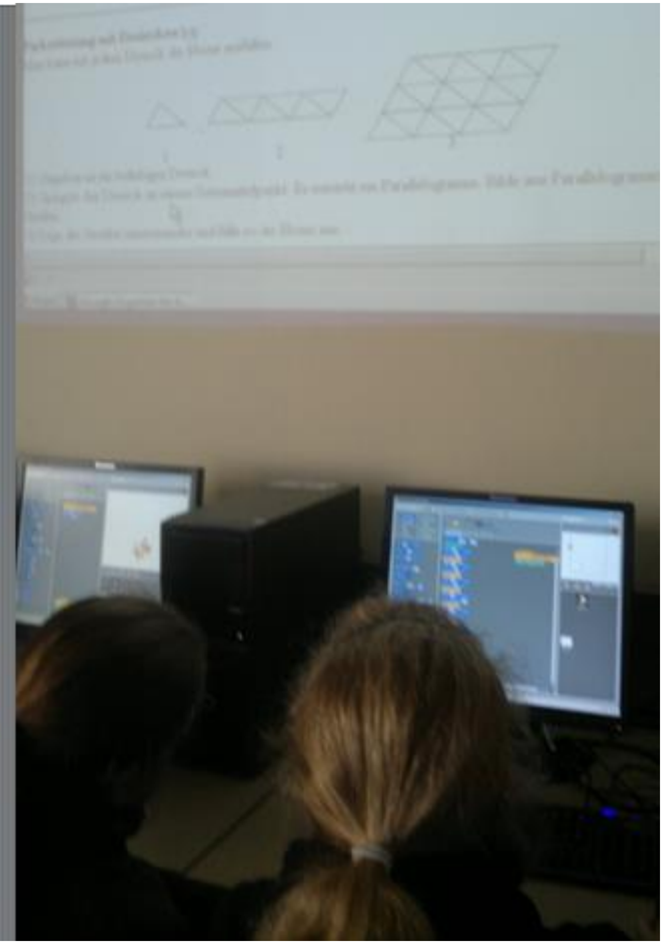
A generalized form with added variables.

Tessellations

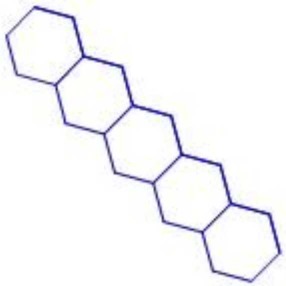


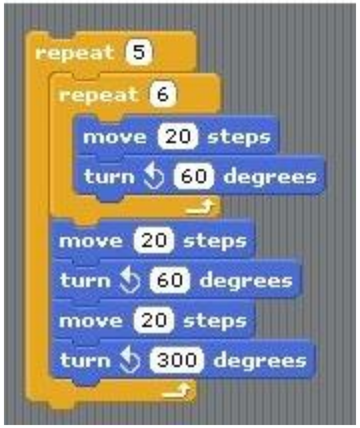


Tessellations

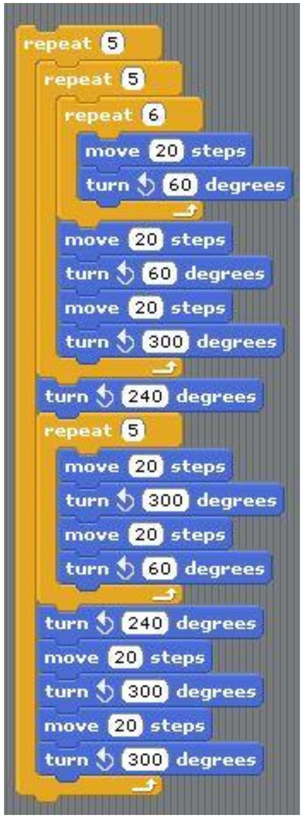


Tessellations



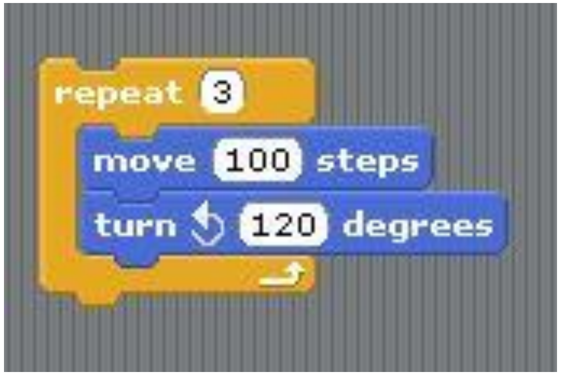


By adding five additional blocks of code, multiple hexagons can be drawn in a row.



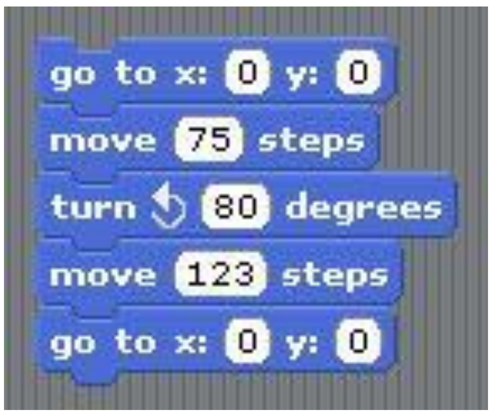
However creating a 5×5 tessellation requires many more additional blocks of code.

Regular Triangles (Grade 7)

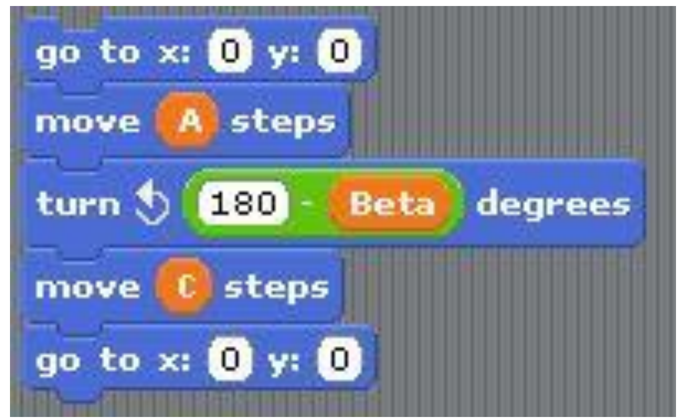


The regular triangle program from grade 6, with the length of each side replaced by the variable A

Congruent Triangle Constructions

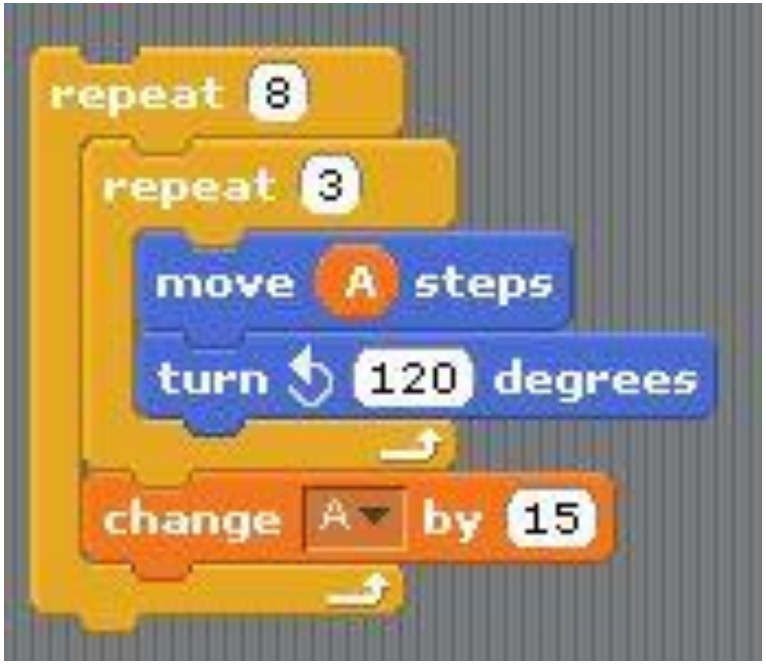


Construction of a triangle with SAS given by 75 – 100° – 123.



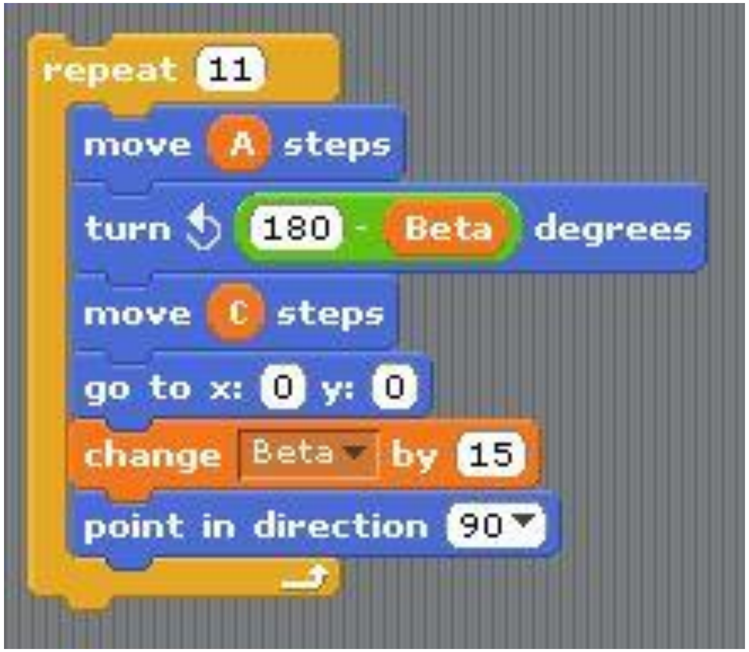
Generalized construction of all triangles given by SAS.

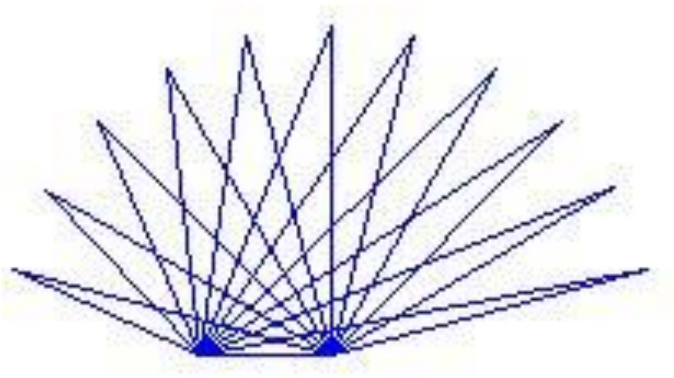
Parametrized Triangle Construction



The program from for regular triangles, with the variable A changing during the execution of the program

Parametrized Triangle Construction



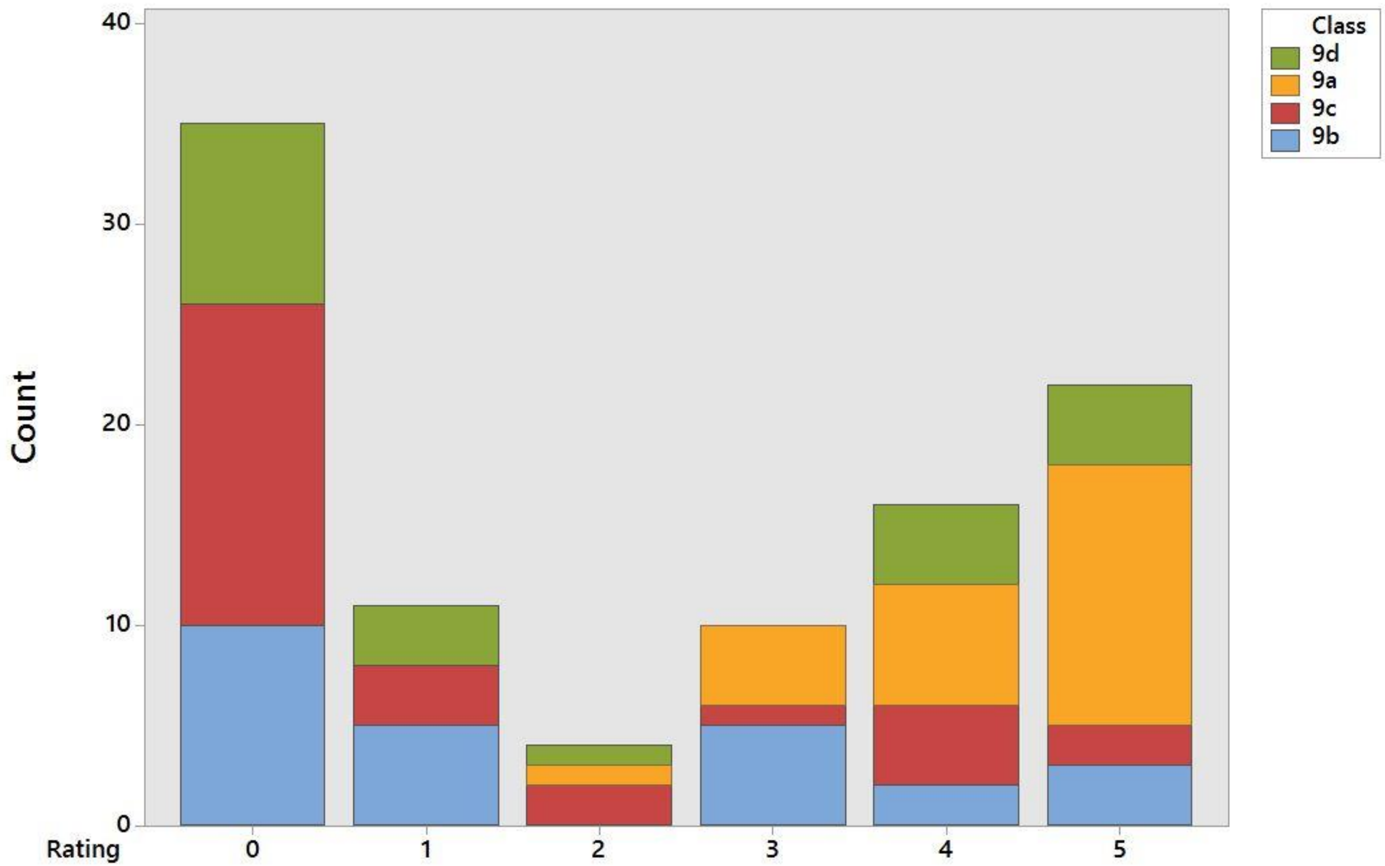


Program for a parametrized family of constructions of the type SAS, with its output next to it

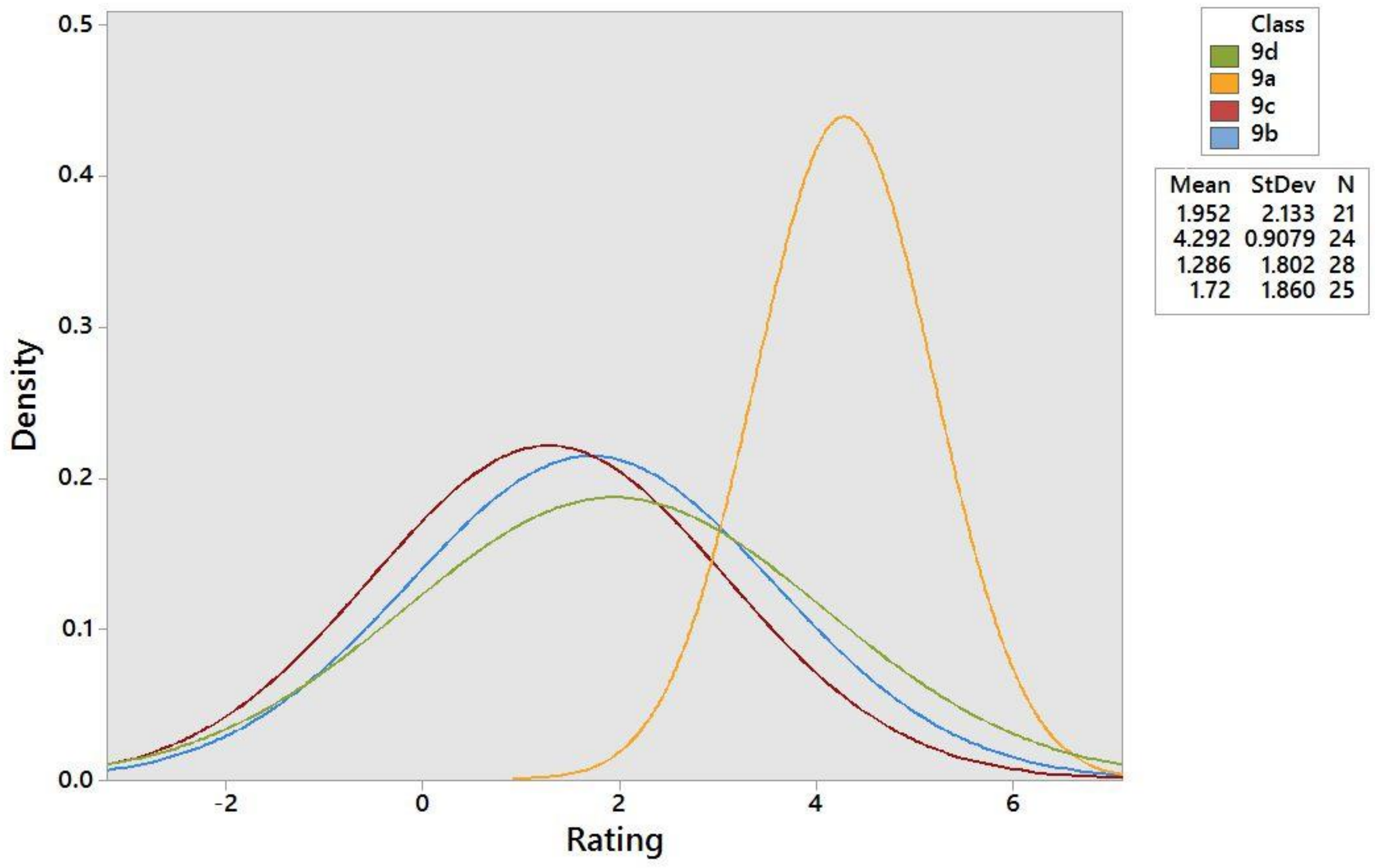
Long-Term Evaluation

- 4 classes (n=98), using the same curriculum
  - 1 class with our Scratch approach in grade 6/7
- Comparison via a classical task:
  - Using pen and paper, describe a congruent triangle construction (SAS: 3cm, 50°, 4cm)
  - Rated from 0 (barely any knowledge) to 5 (correct)

Stacked Column Chart Comparison



Fitted Probability Density Functions



Many more Integrated Possibilities

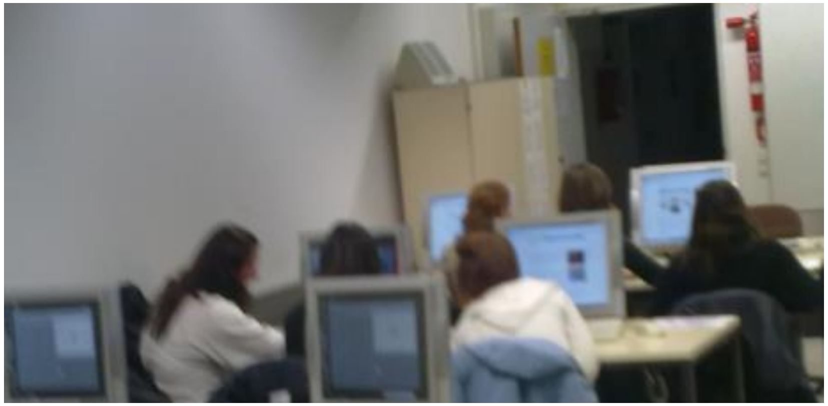
- Bisection method
- Random experiments
- Babylonian method
- Sieve of Eratosthenes
- Euclidean algorithm
- Negative numbers
- Coordinate systems

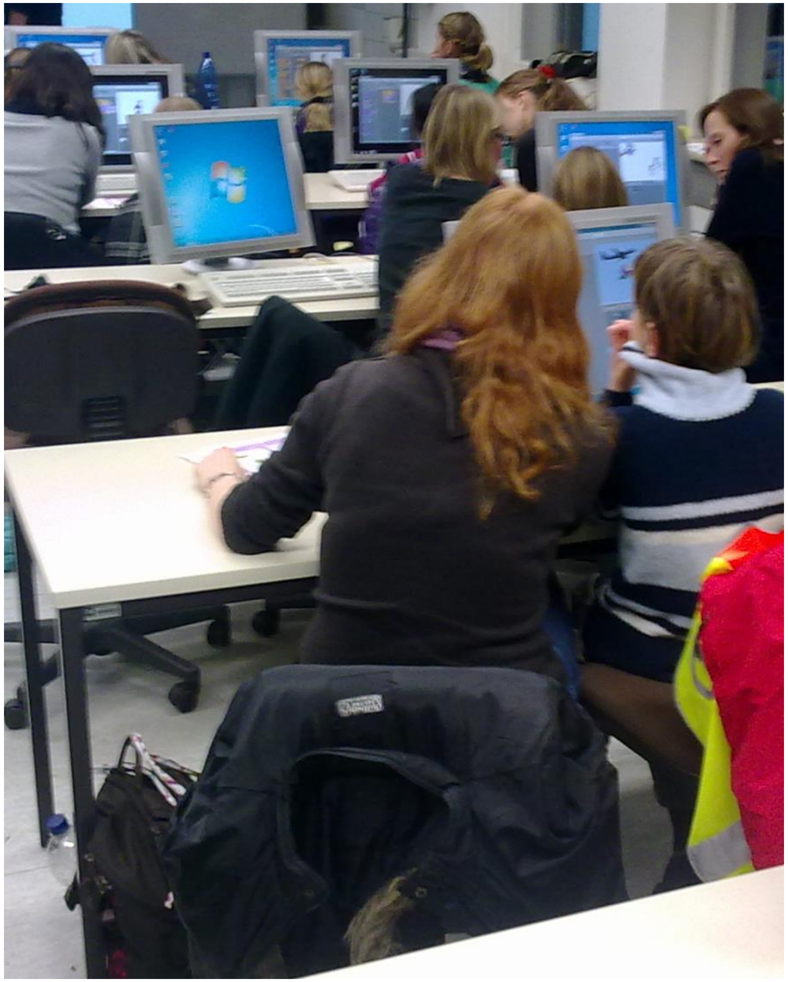
Many more Integrated Possibilities

- Further aspects of mathematics
  - Discrete math, e.g., graph theory
  - Space filling curves
  - Sierpinski triangle

- Also in, e.g.,
  - Physics: Simulation of a spring pendulum (Modrow 2013)
  - Art: Analyzing & simulating concrete art (Wörler 2013)

Also for Teaching Future Teachers





We envision that programming should be a standard tool in mathematics in schools, just as a calculator, compass, or ruler is; a cultural technique that is available to and useable by everyone

Integrating Programming into the Mathematics Curriculum: Combining Scratch and Geometry in Grades 6 and 7

<u>Klaus-Tycho Förster</u>

SIGITE 2016 Boston, MA, USA — September 30