

COMPUTATIONAL STATISTICS AT HIGH SCHOOL – CALCULATING THE ELLIPSE AREA BY THE MONTE CARLO METHOD AND ANALYZING THE DICE POKER GAME OF CHANCE

Jan Benacka
Faculty of Natural Sciences
Constantine the Philosopher University,
Nitra, Slovakia

Abstract

The paper reports on two experiments in which computerized statistical methods were used with high school students of age 18 – 19 to solve two real-life tasks. In the first experiment, the students formulated a hypothesis on the area of ellipse and verified it by the Monte Carlo method. The area formula was deduced from the analogy between the formulas for the area of circle, square and rectangle. The Monte Carlo experiment was carried out by a Visual C# program developed in the lesson. In the second experiment, the students calculated the probability of the dice poker hands by combinatorial formulas and verified it by a Visual C# program developed in the lesson that simulated the game and calculated the relative frequency of the hands. The students answered questionnaires to find out their opinion of the lessons.

Keywords: discrete modelling, programming, probability, simulation.

Introduction

It is generally known that there is a huge demand for STEM specialists in the world [1, 2], especially computer science experts. The reason is that high school students find STEM difficult and boring. Developing computer applications that model and solve real problems might be a way to increase the interest of high school students. Stephenson, Gal-Ezer, Haberman and Verno [3] claim that “Computer Science should be taught using real-world applications rather than specialized educational tools. Computer science instruction should

include integrative and interdisciplinary knowledge”.

Linn and Eylon [4] illustrate exciting opportunities that technology offers in knowledge integration. Keller, Scheuner, Serafini and Steffen [5] reported on an introductory unit in cryptology in which analytical skills and critical thinking were required. There were 53 students from which 80% found the unit interesting and 70% found it motivating to learn more about the topic. Taub, Armoni and Ben-Ari [6] reported on computer modelling in physics with Easy Java Simulations software package. Benacka and Reichel [7] reported on computer modelling in physics with Delphi. There were 43 students from which about 95% found the lesson interesting and more than 60% would be interested in simulating other physical phenomena. Hromkovič and Steffen [8] claim that “Informatics expands science also by giving new dimensions to the fundamental categories of science such as determinism, nondeterminism, randomness, proof, simulation, correctness, efficiency etc. Many of its discoveries are surprising and attractive to young people. For instance, exchanging a deterministic control by a randomized one, whose decisions are partially influenced by random bits, can essentially decrease the amount of work necessary to reach the intended goals”.

An everyday phenomenon that everybody faces, either consciously or unconsciously, is randomness and risk. Since the 1970s, studies of risk have grown into a major interdisciplinary field of research. Risk is regarded in the mathematics curriculum as application of

probability and as a key element of the curriculum in science areas such as nuclear physics, genetics, immunization, and so on. The Oxford English Dictionary [9] defines risk as “a situation involving exposure to danger” and puts its origin in the middle of the 17th century in French “risque” and Italian “risco”. The Stanford Encyclopedia of Philosophy [10] defines the knowledge about risk as follows: “When there is a risk, there must be something that is unknown or has an unknown outcome. Therefore, knowledge about risk is knowledge about lack of knowledge”.

The modern study of risk began in 1654 when Blaise Pascal and Pierre Fermat interchanged a series of letters on mathematical problems related to bets on dice throws, especially the problem of points: Two players who have equal chances of winning each round contribute equally to a prize pot, and agree that the first player to have won a certain number of rounds will take the pot. How to divide the pot fairly if the game is interrupted before either player has achieved victory? The problem, also called the problem of division of the stakes, is considered the one that motivated the beginnings of modern probability theory [11]. Rolling dice has been a rewarding activity to introduce pupils and students to randomness and probability theory [12, 13].

Employing risk as a research tool began in 1946 when Stanislaw Ulam invented the Monte Carlo method while working on nuclear weapons projects at Los Alamos [14]. The method allows solving deterministic problems by using a nondeterministic approach, which is by generating suitable random numbers and observing that fraction of the numbers obey some property. The method is useful for obtaining numerical solutions to problems which are too complicated to be solved analytically. Danesh [15] discussed ways to incorporate Monte-Carlo simulation techniques for introductory science and mathematics teaching and also for enriching computer courses, and described the Monte-Carlo method for modelling physical systems. Fox, Grim and

Hogan [16] proposed a block of exercises to emphasize particular aspects of Monte Carlo simulation, which the students would complete with a project to demonstrate their understanding of the fundamental concepts of the method.

This paper reports on an experiment in which computerized statistical methods were used with 45 high school students of age 18 – 19 to solve two real-life tasks. In the first lesson, the students formulated a hypothesis on the area of ellipse and verified it by the Monte Carlo method. The area formula was deduced from the analogy [17] between the formulas for the area of circle, square and rectangle. The Monte Carlo experiment was carried out by a Visual C# program developed in the lesson. In the second lesson, the students calculated the probability of the dice poker hands by combinatorial formulas and verified it by a Visual C# program developed in the lesson that simulated the game and calculated the relative frequency of hands. The students answered questionnaires to find out their opinion of the lessons.

Determining the Area of Ellipse by the Monte Carlo Method

Calculating area is a task to which mathematicians give attention since antiquity. Integral calculus gives the result if the antiderivative of the boundary curve is known. Otherwise, the task has to be solved in another way, for example by using the Monte Carlo method. Random number generator is essential when using the Monte Carlo method.

Theory

Let there be an ellipse of center $S(0,0)$, eccentricity e , and foci $E(-e,0)$ and $F(e,0)$. Let a and b be the length of the semi major and semi minor axis. Area S of the ellipse is given by the following formula, which is usually not taught at upper secondary level

$$S = \pi ab. \tag{1}$$

It holds for point $X(x, y)$ on the ellipse or inside the ellipse that

$$|EX| + |FX| \leq 2a, \quad (2)$$

where $|EX|$ is the distance between points E and X (Fig. 1, dark grey points). Applying the formula for the distance of two points yields

$$\sqrt{(x+e)^2 + y^2} + \sqrt{(x-e)^2 + y^2} \leq 2a. \quad (3)$$

Let the area of the ellipse be unknown U . Let us make a circumscribed rectangle to the ellipse so that the sides are parallel with axes x and y . It holds for point $X(x, y)$ of the rectangle that $x \in [-a, a]$ and $y \in [-b, b]$ (Fig. 1, light grey points). Area A of the rectangle is

$$A = 4ab. \quad (4)$$

Let us generate N random points $X(x, y)$ so that $x \in [-a, a]$ and $y \in [-b, b]$. Let M of them lie on or inside the ellipse. If $n \rightarrow \infty$, then [18]

$$\frac{U}{A} = \frac{M}{N}, \quad (5)$$

from which area U can be easily computed.

Visual C# Model

The model is shown in Figure 1. It is available at link https://www.dropbox.com/s/su14ufzjs2pvq42/ellipse_mc.zip?dl=0. The zip file contains a folder with Visual C# files (version 2010 Express), an exe file (runs on Windows), and a docx and pdf file with code explained.

Lesson

The experiment was carried out with twenty three students of age 18 – 19 in two groups. A 90 minute lesson in optional subject Visual C# programming was taught to each group by the author (“teacher” in the following text). The aim

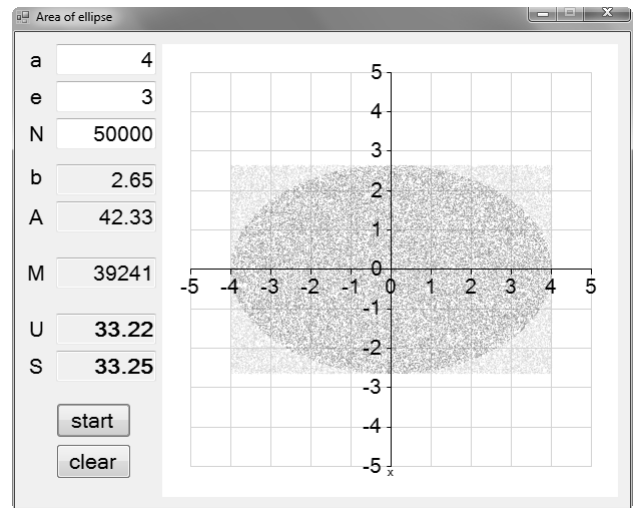
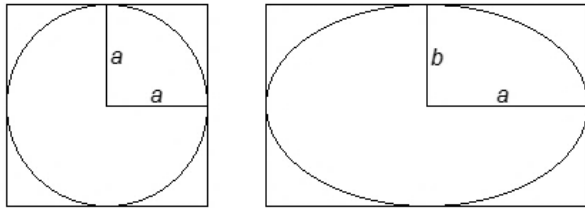


Figure 1: Visual C# application.

of the lesson was to present a method of formulating a hypothesis by using analogy, learn about ellipse, learn the Monte Carlo method, and practice Visual C#. In programming, the students were familiar with loop “for” and with generating integer random numbers. They were not familiar with generating non-integer random numbers from a given interval and with working with a component “chart”.

The reason for calculating the ellipse area was that (1) not all students had learned about ellipse before the lesson, (2) there is a simple formula for the area, which is not taught at high school as it is typically derived by using calculus, and (3) the definition of ellipse is quite clear from the “gardener’s ellipse” [19].

The teacher started the lesson with a discussion about planar figures and the area formulas. Some students knew that ellipse is the result of an oblique cut on a cylinder, but nobody knew the area formula. The teacher carried out the following experiment: he inserted tightly a long cylinder in a square prism shaped box and cut the prism in two positions, first perpendicularly to the longitudinal axis and then askew to it. He projected Figure 2 as the result, but without the formulas.



Square: $A = (2a)^2 = 4 \cdot a \cdot a$ Rectangle: $A = (2a) \cdot (2b) = 4 \cdot a \cdot b$
 Circle: $A = \pi a^2 = \pi \cdot a \cdot a$ Ellipse: $A = \pi \cdot a \cdot b$

Figure 2: Hypothesis on ellipse area.

The teacher invited the students to draw hypotheses on the area of the depicted ellipse by using analogy between the areas of the circle, square and rectangle. After applying the method of questioning [20], the students formulated the hypothesis that the area is πab . The teacher informed the students that the hypothesis would be verified in the rest of the lesson by using a computerized method called the Monte Carlo method.

The teacher explained the principles of the Monte Carlo method by using Figure 3. He invited the students to imagine that 1 million sand grains were sprinkled evenly over the figure, and asked the students how many grains was in the grey rectangle and why? The students answered that about 250 thousand as the area is a quarter of the area of the big rectangle. From that, the students derived the notion that the ratio of the areas and the ratio of the number of grains is the same, which should also hold for the ellipse and the circumscribed rectangle. After some questioning, the students derived equation (5). The teacher said that sprinkling sand would be simulated by generating random points, that is, point with random coordinates. He retold briefly the story of the Monte Carlo method.

The teacher pointed out that a condition has to be derived to find out whether a random point is inside the ellipse or not. He demonstrated the “gardener’s ellipse” construction on the board with the help of two student: “Drive sticks E and F in the ground, take a string longer than distance $|EF|$, fix the ends to the sticks, stretch

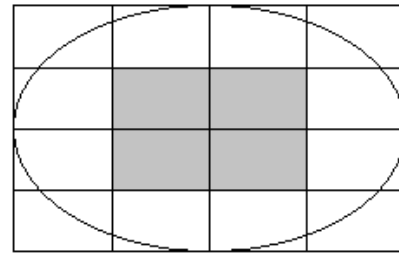


Figure 3: Showing the principle of the Monte Carlo method.

it with third stick X , and draw a curve, an ellipse, around points E and F . The intersections of the ellipse with line EF are the main vertices, typically called A and B . Then, segment AB is the major axis. Center S of segment AB is the center of the ellipse. The intersections of the ellipse with the perpendicular to EF at S are the co-vertices, typically called C and D . Then, segment CD is the minor axis. Distance $e = |ES| = |FS|$ is the eccentricity, distance $a = |AS| = |BS|$ is the length of the major half-axis, and $b = |CS| = |DS|$ is the length of the minor half-axis. It follows from the construction that $|EX| + |FX|$ is constant, actually $|EX| + |FX| = 2a$, which is clear if point X equals point A or B . If point X equals point C or D , then it can be seen that $a^2 = e^2 + b^2$.” The students derived inequality (3) as the condition to find out if point X is inside the ellipse or not.

The teacher projected the application (Figure 1). The students were familiar with the fact that random generators produce real numbers from 0 including to 1 excluding. After some questioning, they derived the equations for generating real numbers from a given interval (coded in program lines 39 and 40).

The students downloaded the template of the application that the teacher had prepared to save time (Figure 1 without the chart). The students put the chart on the form, and the teacher showed them how to set the chart parameters. Then, the students started coding. The teacher was working along with them, discussing the

steps of creation, and projecting his code to give feedback but with a delay so that the students could work independently. No obstacles occurred except for adding and clearing points into and from chart series (code lines 34, 35, 46 and 49), which was a new skill for the students. Obtaining the result, which proved the correctness of the hypothesis on the area formula, was clearly a hit. At the end of the lesson, the students answered a questionnaire to find out their opinion of the lesson.

Survey

The questionnaire comprised the following questions:

- A) *The lesson was (1 = very; 2 = quite; 3 = little; 4 = not) interesting.*
- B) *I understood (1 = all; 2 = most; 3 = little; 4 = nothing) of the algorithm.*
- C) *I learnt (1 = a lot; 2 = quite a lot; 3 = little; 4 = nothing) new programming.*
- D) *I learnt about ellipse before the lesson (1 = yes; 2 = no).*
- E) *I knew about the Monte Carlo method before the lesson (1 = yes; 2 = no).*
- F) *I would like to model and analyze the dice poker game (1 = yes; 2 = no).*
- G) *I am a boy (1 = yes; 2 = no).*

The absolute and relative number of answers is in Table 1. We take answers 1 and 2 in questions A – C and answer 1 in questions D – G for positive ones.

Table 1: Number of answers.

	Number				All %				Boys %				Girls %			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
A	19	4	0	0	83	17	0	0	83	17	0	0	80	20	0	0
B	14	9	0	0	61	39	0	0	61	39	0	0	60	40	0	0
C	8	12	3	0	35	52	13	0	33	61	6	0	40	20	40	0
D	6	17			26	74			33	67			0	100		
E	10	13			43	57			56	44			0	100		
F	16	7			70	30			67	33			80	20		
G	18	5			78	22										

The number of the positive answers is (all, boys, girls):

- A) *The lesson was interesting: 100%, 100%, 100%*
- B) *I understood the algorithm: 100%, 100%, 100%*
- C) *I learnt new in programming: 87%, 94%, 60%*
- D) *I learnt about ellipse before the lesson: 26%, 33%, 0%*
- E) *I knew about the Monte Carlo method before the lesson: 43%, 56%, 0%*
- F) *I would like to model the dice poker: 70%, 67%, 80%*
- G) *I am a boy: 78%*

Determining the Probability of the Dice Poker Hands by Simulation

Dice poker is known from the latter half of the 19th century [21]. When calculating the probability of hands, students are typically not much certain of the result. Verifying by writing down all alternatives, $6^5 = 7776$, is impossible. Computers allow using another method – to simulate the game and calculate the relative frequency of hands at a high number of throws.

Theory

The hands, number of positive results and the probability of hands are listed in Table 2.

Visual C# Models

Three models were developed. The first one just simulated the game (Figure 4). The second one found out the hand (Figure 5). The third one calculated the relative frequency of hands (Figure 6). The models are available at the link <https://www.dropbox.com/s/65ycn2uh08oxemk/dicepoker.zip?dl=0>. The zip file contains folders with Visual C# files (version 2010 Express), exe files (run on Windows), docx and pdf files with the code explained, and bmp figures (note that they must be in the same folder as the exe files).

Table 2: Dice poker hands.

Hand	Number of positive results	%	Example
Five of a kind	$\binom{5}{5} \cdot 6 = 6$	0.08	44444
Four of a kind	$\binom{5}{4} \cdot 6 \cdot 5 = 150$	1.93	66663
Full house	$\binom{5}{3} \cdot 6 \cdot 5 = 300$	3.86	22211
Straight	$\binom{5}{1} \cdot 2 \cdot 4! = 240$	3.09	12345
Three of a kind	$\binom{5}{3} \cdot 6 \cdot 5 \cdot 4 = 1200$	15.43	55516
Two pair	$\binom{5}{2} \cdot 6 \cdot \binom{3}{2} \cdot 5 \cdot 4 \cdot \frac{1}{2} = 1800$	23.15	66224
One pair	$\binom{5}{2} \cdot 6 \cdot 5 \cdot 4 \cdot 3 = 3600$	46.30	33126
High card	$\binom{5}{2} \cdot 2 \cdot 4! = 480$	6.17	12346

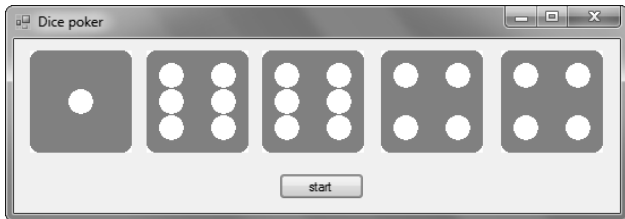


Figure 4: Simple dice poker model.

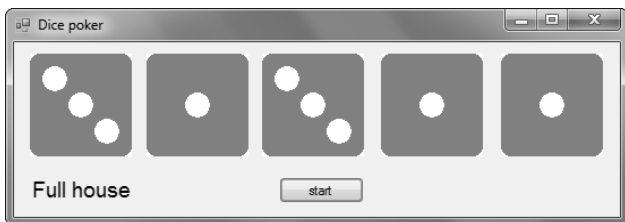


Figure 5: Dice poker model with hand name.

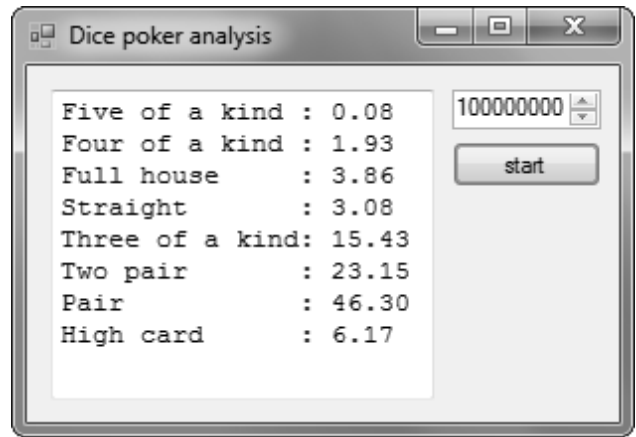


Figure 6: Relative frequency of hands.

Lessons

The experiment was carried out with twenty two students of age 18 – 19 in two groups. Two 90 minute lessons in optional subject Visual C# programming were taught to each group by the author (“teacher” in the following text). The reason for choosing the dice poker was that it can be easily simulated as the numbers in the throw may repeat. In programming, the students were familiar with loop “for”, with generating integer and real random numbers, and with arrays of numbers. Component picture box was new to them.

The lesson started with a demonstration of the game by using real dice. The teacher explained the rules. The students started developing the first model (Figure 4). They drew the dice faces in the MS Paint program and named them 1.bmp, 2.bmp, etc. The teacher was developing the application along with the students and projecting his work to give feedback but with a delay to enable the clever students work individually. He largely used the method of questioning [20] to help the students proceed. He explained how to work with picture boxes and how to make an array of them (code lines 21, 28). The students clearly enjoyed the result.

Developing the second model (Figure 5) started with discussing the algorithm. The students found out that it was necessary to remember how many times each number fell. They introduced array Count (code lines 23 and

31). To find out the hand, the teacher suggested starting with “Five of a kind”, the simplest case as there is five equal numbers. Hand “Four of a kind” was easy to find out alike (code lines 42, 43). Hands “Full house”, “Three of a kind” and “Two pair” appeared to be difficult to find out. The teacher suggested introducing boolean variables Pair and Three (lines 37, 38), which much helped the students resolve the problem (lines 46-60). The last obstacle was to find out if hand “High card” or “Straight” fell. It was clear to the students that “High card” must have contained both 1 and 6, while “Straight” must have contained either 1 or 6. After a discussion, the teacher pointed out that all other hands had already been resolved. The students realized that it was enough to put the two options before the others (lines 34, 35). The complete model gave the students clearly the feeling of satisfaction.

Before developing the third version (Figure 6), the teacher invited the students to calculate the probability of hands by combinatorial formulas. Typically, the students were not much sure about the result. The teacher projected the application in Figure 6, which proved Table 2 to be correct, and invited the students to adjust the second model. The students introduced variable NumberOfThrows and array Frequency (lines 21, 23), changed the name of the hands to numbers (lines 39, 40, 47, 48, 54, 62, 63, 65) and computed the frequency of hands in a “for” loop that repeated as many times as was the number of throws (line 27, 28, 68). Finally, they added the command for calculating and writing out the relative frequency of hands in the multiline textbox. The result obviously gave the students the feeling of a great achievement.

Survey

The students answered the following questionnaire at the end of the lesson:

- A) *The lesson was (1 = very; 2 = quite; 3 = little; 4 = not) interesting.*
- B) *I understood (1 = all; 2 = most; 3 = little; 4 = nothing) of the algorithm.*

- C) *I learnt (1 = a lot; 2 = quite a lot; 3 = little; 4 = nothing) new in C# programming.*
- D) *I would like to model and analyze the card poker game (1 = yes; 2 = no).*
- E) *I am a boy (1 = yes; 2 = no).*

The absolute and relative number of answers is in Table 3. We take answers 1 and 2 in questions A – C and answer 1 in questions D – E for positive ones.

Table3: Number of answers.

	Number				All %				Boys %				Girls %			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
A	10	12	0	0	45	55	0	0	44	56	0	0	50	50	0	0
B	8	14	0	0	36	64	0	0	33	67	0	0	50	50	0	0
C	13	9	0	0	59	41	0	0	50	50	0	0	100	0	0	0
D	20	2			91	9			89	11			100	0		
E	18	4			82	18										

The number of positive answers is (all, boys, girls):

- A) *The lesson was interesting: 100%, 100%, 100%*
- B) *I understood the algorithm: 100%, 100%, 100%*
- C) *I learnt new in C#: 100%, 100%, 100%*
- D) *I would like to model the card poker: 91%, 89%, 100%*
- E) *I am a boy: 82%*

Summary and Conclusions

In both experiments, the result of 100% in questions (A) and (B) suggests that the lessons were successful.

In the Monte Carlo experiment, 87% learned new skills in programming, 74% learned about ellipse, 57% learned about the Monte Carlo method, and 70% found the topic motivating to continue with other random processes. The boys answered much more positively than the girls except for question (F). However, sample sizes were too small to make strong conclusions, especially about gender differences.

In the dice poker experiment, the result of 100% in questions (C) is interesting as just working with picture boxes was new to the students; however, accomplishing the program clearly gave the students the feeling of having done a considerable step in their programming skills. The overall outcome of 91% in question (D) suggests that the topic was motivating to continue with other random processes.

The overall result implies that introducing random processes and computerized statistical methods has the potential to promote programming to high school students.

References

1. European Commission, “*Europe 2020*”, http://ec.europa.eu/europe2020/index_en.htm (accessed May 2016).
2. The White House, “*Report to the President: prepare and inspire: K-12 education in science, technology, engineering, and math (STEM) for America’s future*”, <https://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast-stem-ed-final.pdf> (accessed May 2016).
3. C. Stephenson, J. Gal-Ezer, B. Haberman, and A. Verno, *The New Education Imperative: Improving High School Computer Science Education, Final Report of the CSTA Curriculum Improvement Task Force*, CSTA, ACM (February 2005), http://csta.acm.org/Communications/sub/DocsPresentationFiles/White_Paper07_06.pdf (accessed May 2016).
4. M. C. Linn, and B. S. Eylon, *Science learning and instruction: Taking advantage of technology to promote knowledge integration*, Routledge, New York, 2011.
5. L. Keller, B. Scheuner, G. Serafini, and B. Steffen, “A short introduction to classical cryptology as a way to motivate high school students for informatics”, in *Proceedings of ISSEP 2011, LNCS 7013*, Springer, Berlin, 2011.
6. R. Taub, M. Armoni, and M. Ben-Ari, “The contribution of computer science to learning computational physics”, in *Proceedings of ISSEP 2013, LNCS 7780*, Springer, Berlin, 2013.
7. J. Benacka, and J. Reichel, “Computer modeling with Delphi – constructionism and IBL in practice and motivation for studying STEM”, in *Proceedings of ISSEP 2013, LNCS 7780*, Springer, Berlin, 2013.
8. J. Hromkovič, and B. Steffen, “Why teaching informatics in schools is as important as teaching mathematics and natural sciences”, in *Proceedings of ISSEP 2011, LNCS 7013*, Springer, Berlin, Germany, 2011.
9. Oxford Dictionaries, “Risk”, <http://www.oxforddictionaries.com/definition/english/risk?searchDictCode=all> (accessed May 2016).
10. S. O. Hansson, “Risk”, in *The Stanford Encyclopedia of Philosophy (Spring 2014 Edition)*. <http://plato.stanford.edu/archive/spr2014/entries/risk/> (accessed May 2016).
11. A. Hald, *A History of Probability and Statistics and Their Applications before 1750*. Wiley, Hoboken, NJ, 2003.
12. T. Lough, and S. Tipps, “Learning with Logo. Spinning, rolling, and flipping: more on probability”, *Journal of Computers in Mathematics and Science Teaching*, vol. 7, no. 3, 1988.
13. K. B. Smith, “Programming and probability: a practical application of principles and standards for school

mathematics”, *Journal of Computers in Mathematics and Science Teaching*, vol. 22, no. 1, 2003.

14. N. Metropolis, “The beginning of the Monte Carlo method”, *Los Alamos Science, special issue dedicated to Stanislaw Ulam*, 1987, <http://library.lanl.gov/cgi-bin/getfile?00326866.pdf> (accessed May 2016).
15. I. Danesh, “Incorporation of Monte-Carlo computer techniques into science and mathematics education”, *Journal of Computers in Mathematics and Science Teaching*, vol. 6, no. 4, 1987.
16. W. P. Fox, P. E. Grim, and D. L. Hogan, “Teaching reinforcement of stochastic behavior using Monte Carlo simulation”, *Journal of Computers in Mathematics and Science Teaching*, vol. 15, no. 4, 1996.
17. M. Hesse, *Models and Analogies in Science*, Sheen and Ward, London, 1966.
18. G. S. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications*, Springer, New York, 1995.
19. F. J. Dijksterhuis, “Moving around the Ellipse. Conic sections in Leiden, 1620-1660”, in *Silent Messengers: The Circulation of Material Objects of Knowledge in the Early Modern Low Countries*, LIT, Münster, 2011.
20. G. Petty, *Teaching Today: A Practical Guide*, Nelson Thornes, Cheltenham, 2004.
21. Encyclopaedia Britannica, “Poker dice”, <http://www.britannica.com/EBchecked/topic/466642/poker-dice> (accessed May 2016).

Biographical Information

Jan Benacka, Ph.D. is an assistant professor. He graduated in Mathematics, Physics and Informatics. He has 22 years experience in teaching the subjects at secondary school and 13 at university. His present research is focused on promoting STEM to high school students through computer modelling with Excel, Delphi/Lazarus and Visual C#. <http://www.ki.fpv.ukf.sk/~jbenacka/>