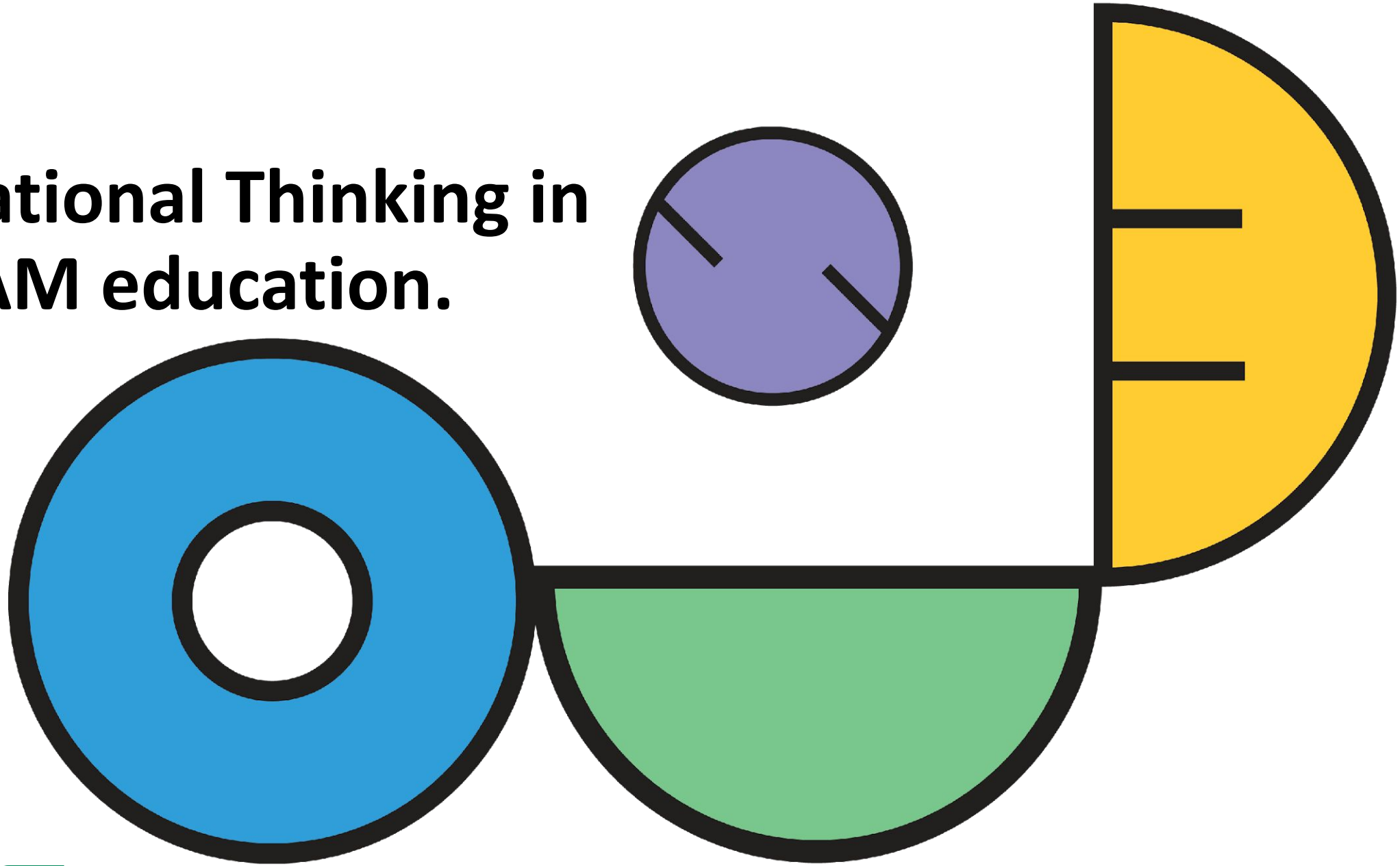
DOSE
DEVELOPMENT OF STEAM EDUCATION

Computational Thinking in STEAM education.



Funded by the Erasmus+ Programme of the European Union

NACIONALINĖ ŠVIETIMO AGENTŪRA

Context

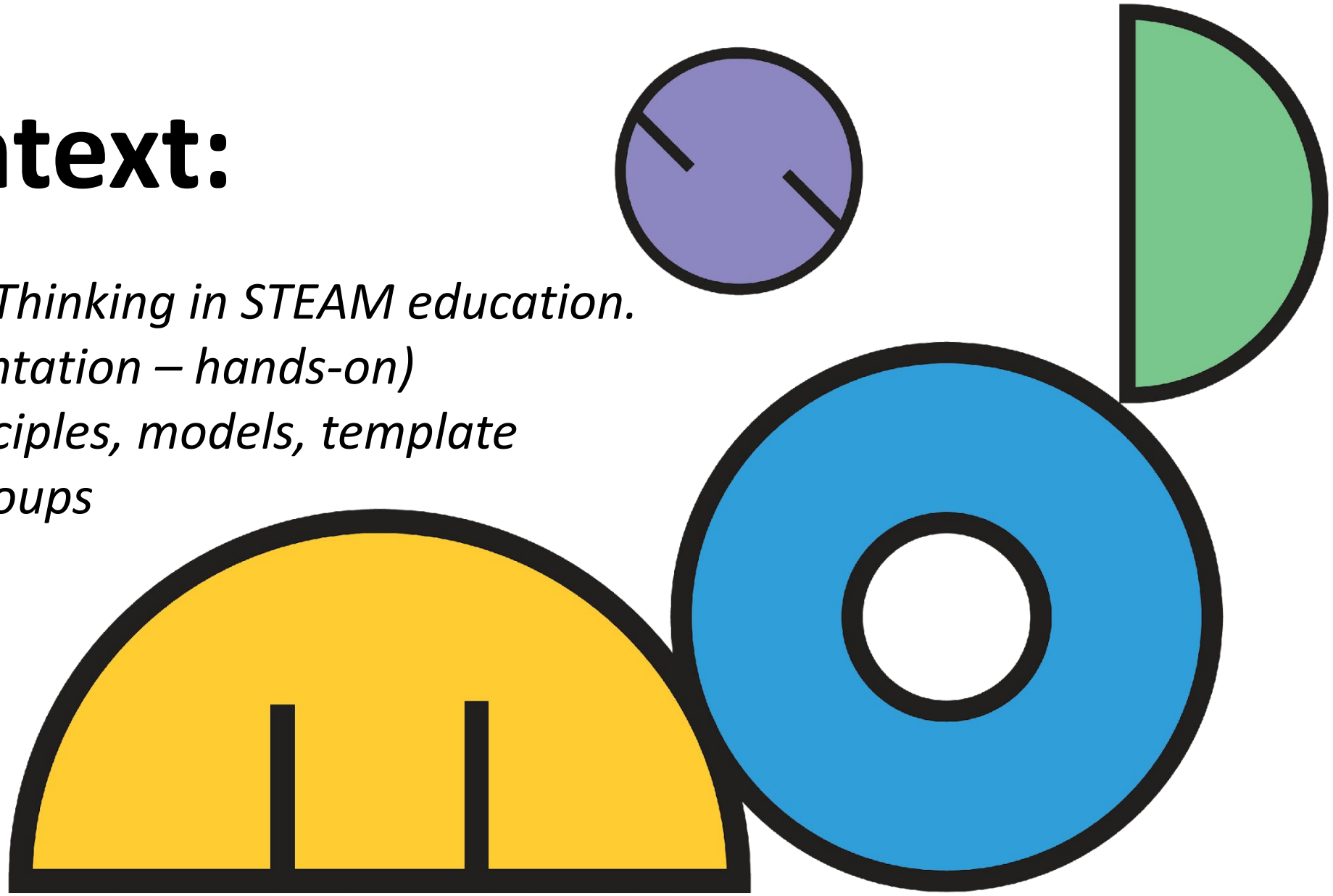
Description of the activity	Introduction to Computational Thinking and some ideas on how to connect it into STEAM (to develop further and expand). Computational Thinking in STEAM education refers to integrating the principles of problem-solving and logical reasoning from computer science into the fields of Science, Technology, Engineering, Arts, and Mathematics (STEAM).In this module, you will learn how to implement Computational Thinking through 3D printing and creative coding
Target group(s):	Teachers, School principals, Teacher trainers.
Keywords:	Computational Thinking, coding, problem-solving, logical reasoning, algorithmic thinking,programming, STEAM, creativity, innovation.
Duration of activity:	2 hours
Description of activity environment and materials	Brainstorming session, theoretical part, analysis of examples, sharing of experience. Reading through this presentation provides tools to carry out 3D printing, and coding. It also shows you the underlying principles and models of computational thinking in STEAM education in order to reflect upon the activity.

Funded by the Erasmus+ Programme of the European Union

The Development of STEAM education is co-funded by the Erasmus+ Programme of the European Union (Grant Agreement No. 2020-1-LT01-KA201-077935). The content of this document is the sole responsibility of the organizer and does not represent the opinion of the European Commission (EC), nor is the EC responsible for any use that might be made of information contained.

Context:

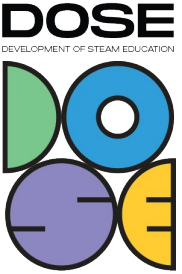
- *Computational Thinking in STEAM education.*
- *Example (presentation – hands-on)*
- *Underlying principles, models, template*
- *Discussion in groups*



Funded by the Erasmus+ Programme of the European Union

NACIONALINĖ ŠVIETIMO AGENTŪRA

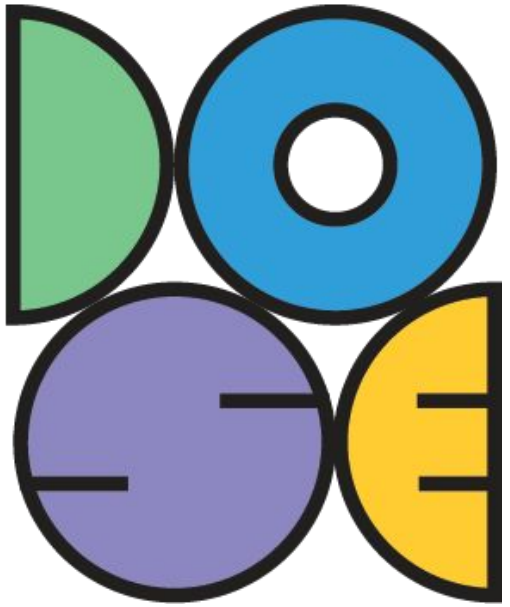
DOSE
DEVELOPMENT OF STEAM EDUCATION



COMPUTATIONAL THINKING IN STEAM

What is computational thinking and
how to take advantage of it in STEAM education?

DOSE
DEVELOPMENT OF STEAM EDUCATION



Funded by the
Erasmus+ Programme
of the European Union

NACIONALINĖ
ŠVIETIMO
AGENTŪRA

Computational Thinking in STEAM education
What is it?

- Computational Thinking in STEAM education refers to integrating the principles of problem-solving and logical reasoning from computer science into the fields of Science, Technology, Engineering, Arts, and Mathematics (STEAM). It emphasizes skills like decomposition (breaking down complex problems), pattern recognition, abstraction, and algorithmic thinking, fostering creativity and innovation across diverse disciplines. This approach equips students with versatile problem-solving abilities and prepares them for the modern, technology-driven world.

- *Please share your thoughts and experiences (group work).*

Funded by the Erasmus+ Programme of the European Union
NACIONALINĖ ŠVIETIMO AGENTŪRA

WHAT IS COMPUTATIONAL THINKING?

- Computational Thinking (CT) is something programmers need to be able to program.
- CT involves programming elements, strategies and thinking processes.
- Brennan and Resnick (2012) divided CT into three areas: Concepts, Practices and Perspectives.
  - … with additions from Zhang and Nouri (2019)

K. Brennan ja M. Resnick. 2012. New Frameworks for Studying And Assessing the Development of Computational Thinking. http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf
L. Zhang ja J. Nouri. 2019. A Systematic Review of Learning Computational Thinking Through Scratch in K-9. https://doi.org/10.1016/j.compedu.2019.103607

Funded by the Erasmus+ Programme of the European Union
NACIONALINĖ ŠVIETIMO AGENTŪRA

DOSE
DEVELOPMENT OF STEAM EDUCATION

Computing Progression Pathways

Pupil Progression	Algorithms	Programming & Development	Data & Data Representation	Hardware & Processing	Communication & Networks	Information Technology
	• Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) • Understands that computers need precise instructions. (AL) • Demonstrates care and precision to avoid errors. (AL)	• Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text e.g. programmable robots etc. (AL) • Executes, checks and changes programs. (AL) • Understands that programs execute by following precise instructions. (AL)	• Recognises that digital content can be represented in many forms. (AB) (GE) • Distinguishes between some of these forms and can explain the different ways that they communicate information. (AB)	• Understands that computers have no intelligence and that computers can do nothing unless a program is executed. (AL) • Recognises that all software executed on digital devices is programmed. (AL) (AB) (GE)	• Obtains content from the world wide web using a web browser. (AL) • Understands the importance of communicating safely and respectfully online, and the need for keeping personal information private. (EV) • Knows what to do when concerned about content or being contacted. (AL)	• Uses software under the control of the teacher to create, store and edit digital content using appropriate file and folder names. (AB) (GE) (DE) • Understands that people interact with computers. • Shares their use of technology in school. • Knows common uses of information technology beyond the classroom. (GE) • Talks about their work and makes changes to improve it. (EV)
	• Understands that algorithms are implemented on digital devices as programs.(AL) • Designs simple algorithms using loops, and selection i.e. if statements. (AL) • Uses logical reasoning to predict outcomes. (AL) • Detects and corrects errors i.e. debugging, in algorithms. (AL)	• Uses arithmetic operators, if statements, and loops, within programs. (AL) • Uses logical reasoning to predict the behaviour of programs. (AL) • Detects and corrects simple semantic errors i.e. debugging, in programs. (AL)	• Recognises different types of data: text, number. (AB) (GE) • Appreciates that programs can work with different types of data. (GE) • Recognises that data can be structured in tables to make it useful. (AB) (DE)	• Recognises that a range of digital devices can be considered a computer. (AB) (GE) • Recognises and can use a range of input and output devices. • Understands how programs specify the function of a general purpose computer. (AB)	• Navigates the web and can carry out simple web searches to collect digital content. (AL) (EV) • Demonstrates use of computers safely and responsibly, knowing a range of ways to report unacceptable content and contact when online.	• Uses technology with increasing independence to purposefully organise digital content. (AB) • Shows an awareness for the quality of digital content collected. (EV) • Uses a variety of software to manipulate and present digital content: data and information. (AL) • Shares their experiences of technology in school and beyond the classroom. (GE) (EV) • Talks about their work and makes improvements to solutions based on feedback received.(EV)
	• Designs solutions (algorithms) that use repetition and two-way selection i.e. if, then and else. (AL) • Uses diagrams to express solutions. (AB) • Uses logical reasoning to predict outputs, showing an awareness of inputs. (AL)	• Creates programs that implement algorithms to achieve given goals. (AL) • Declares and assigns variables. (AB) • Uses post-tested loop e.g. 'until', and a sequence of selection statements in programs, including an if, then and else statement. (AL)	• Understands the difference between data and information. (AB) • Knows why sorting data in a flat file can improve searching for information. (EV) • Uses filters or can perform single criteria searches for information.(AL)	• Knows that computers collect data from various input devices, including sensors and application software. (AB) • Understands the difference between hardware and application software, and their roles within a computer system. (AB)	• Understands the difference between the internet and internet service e.g. world wide web. (AB) • Shows an awareness of, and can use a range of internet services e.g. VOIP. • Recognises what is acceptable and unacceptable behaviour when using technologies and online services.	• Collects, organises and presents data and information in digital content. (AB) • Creates digital content to achieve a given goal through combining software packages and internet services to communicate with a wider audience e.g. blogging. (AL) • Makes appropriate improvements to solutions based on feedback received, and can comment on the success of the solution. (EV)
	• Shows an awareness of tasks best completed by humans or computers. (EV) • Designs solutions by decomposing a problem and creates a sub-solution for each of these parts. (DE) (AL) (AB) • Recognises that different solutions exist for the same problem. (AL) (AB)	• Understands the difference between, and appropriately uses if and if, then and else statements. (AL) • Uses a variable and relational operators within a loop to govern termination. (AL) (GE) • Designs, writes and debugs modular programs using procedures. (AL) (DE) (AB) (GE) • Knows that a procedure can be used to hide the detail with sub-solution. (AL) (DE) (AB) (GE)	• Performs more complex searches for information e.g. using Boolean and relational operators. (AL) (GE) (EV) • Analyses and evaluates data and information, and recognises that poor quality data leads to unreliable results, and inaccurate conclusions. (AL) (EV)	• Understands why and when computers are used. (EV) • Understands the main functions of the operating system. (DE) (AB) • Knows the difference between physical, wireless and mobile networks. (AB)	• Understands how to effectively use search engines, and knows how search results are selected, including that search engines use 'web crawler programs'. (AB) (GE) (EV) • Selects, combines and uses internet services. (EV) • Demonstrates responsible use of technologies and online services, and knows a range of ways to report concerns.	• Makes judgements about digital content when evaluating and repurposing it for a given audience. (EV) (GE) • Recognises the audience when designing and creating digital content. (EV) • Understands the potential of information technology for collaboration when computers are networked. (GE) • Uses criteria to evaluate the quality of solutions, can identify improvements making some refinements to the solution, and future solutions. (EV)
	• Understands that iteration is the repetition of a process such as a loop. (AL) • Recognises that different algorithms exist for the same problem. (AL) (GE) • Represents solutions using a structured notation. (AL) (AB) • Can identify similarities and differences in situations and can use these to solve problems (pattern recognition). (GE)	• Understands that programming bridges the gap between algorithmic solutions and computers. (AB) • Has practical experience of a high-level textual language, including using standard libraries when programming. (AB) (AL) • Uses a range of operators and expressions e.g. Boolean, and applies them in the context of program control. (AL) • Selects the appropriate data types. (AL) (AB)	• Knows that digital computers use binary to represent all data. (AB) • Understands how bit patterns represent numbers and images. (AB) • Knows that computers transfer data in binary. (AB) • Understands the relationship between binary and file size (uncompressed). (AB) • Defines data types: real numbers and Boolean. (AB) • Queries data on one table using a typical query language. (AB)	• Recognises and understands the function of the main internal parts of basic computer architecture. (AB) • Understands the concepts behind the fetch-execute cycle. (AB) (AL) • Knows that there is a range of operating systems and application software for the same hardware. (AB)	• Understands how search engines rank search results. (AL) • Understands how to construct static web pages using HTML and CSS. (AL) (AB) • Understands data transmission between digital computers over networks, including the internet i.e. IP addresses and packet switching. (AL) (AB)	• Evaluates the appropriateness of digital devices, internet services and application software to achieve given goals. (EV) • Recognises ethical issues surrounding the application of information technology beyond school. • Designs criteria to critically evaluate the quality of solutions, uses the criteria to identify improvements and can make appropriate refinements to the solution. (EV)
	• Understands a recursive solution to a problem repeatedly applies the same solution to smaller instances of the problem. (AL) (GE) • Recognises that some problems share the same characteristics and use the same algorithm to solve both. (AL) (GE) • Understands the notion of performance for algorithms and appreciates that some algorithms have different performance characteristics for the same task. (AL) (EV)	• Uses nested selection statements. (AL) • Appreciates the need for, and writes, custom functions including use of parameters. (AL) (AB) • Knows the difference between, and uses appropriately, procedures and functions. (AL) (AB) • Understands and uses negation with operators. (AL) • Uses and manipulates one dimensional data structures. (AB) • Detects and corrects syntactical errors. (AL)	• Understands how numbers, images, sounds and character sets use the same bit patterns. (AB) (GE) • Performs simple operations using bit patterns e.g. binary addition. (AB) (AL) • Understands the relationship between resolution and colour depth, including the effect on file size. (AB) • Distinguishes between data used in a simple program (a variable) and the storage structure for that data. (AB)	• Understands the von Neumann architecture in relation to the fetch-execute cycle, including how data is stored in memory. (AB) (GE) • Understands the basic function and operation of location addressable memory.(AB)	• Knows the names of hardware e.g. hubs, routers, switches, and the names of protocols e.g. SMTP, IMAP, POP, FTP, TCP/IP, associated with networking computer systems. (AB) • Uses technologies and online services securely, and knows how to identify and report inappropriate conduct. (AL)	• Justifies the choice of and independently combines and uses multiple digital devices, internet services and application software to achieve given goals. (EV) • Evaluates the trustworthiness of digital content and considers the usability of visual design features when designing and creating digital artifacts for a known audience. (EV) • Identifies and explains how the use of technology can impact on society. • Designs criteria for users to evaluate the quality of solutions, uses the feedback from the users to identify improvements and can make appropriate refinements to the solution. (EV)
	• Recognises that the design of an algorithm is distinct from its expression in a programming language (which will depend on the programming constructs available). (AL) (AB) • Evaluates the effectiveness of algorithms and models for similar problems. (AL) (AB) (GE) • Recognises where information can be filtered out in generalizing problem solutions. (AL) (AB) (GE) • Uses logical reasoning to explain how an algorithm works. (AL) (AB) (DE) • Represents algorithms using structured language. (AL) (DE) (AB)	• Appreciates the effect of the scope of a variable e.g. a local variable can't be accessed from outside its function. (AB) (AL) • Understands and applies parameter passing. (AB) (GE) (DE) • Understands the difference between, and uses, both pre-tested e.g. 'while', and post-tested e.g. 'until' loops. (AL) • Applies a modular approach to error detection and correction. (AB) (DE) (GE)	• Knows the relationship between data representation and data quality. (AB) • Understands the relationship between binary and electrical circuits, including Boolean logic. (AB) • Understands how and why values are data typed in many different languages when manipulated within programs. (AB)	• Knows that processors have instruction sets and that these relate to low-level instructions carried out by a computer. (AB) (AL) (GE)	• Knows the purpose of the hardware and protocols associated with networking computer systems. (AB) (AL) • Understands the client-server model including how dynamic web pages use server-side scripting and that web servers process and store data entered by users. (AL) (AB) (DE) • Recognises that persistence of data on the internet requires careful protection of online identity and privacy.	• Undertakes creative projects that collect, analyse, and evaluate data to meet the needs of a known user group. (AL) (DE) (EV) • Effectively designs and creates digital artefacts for a wider or remote audience. (AL) (DE) • Considers the properties of media when importing them into digital artefacts. (AB) • Documents user feedback, the improvements identified and the refinements made to the solution. (AB) • Explains and justifies how the use of technology impacts on society, from the perspective of social, economical, political, legal, ethical and moral issues. (EV)
	• Designs a solution to a problem that depends on solutions to smaller instances of the same problem (recursion). (AL) (DE) (AB) (GE) • Understands that some problems cannot be solved computationally. (AB) (GE)	• Designs and writes nested modular programs that enforce reusability utilising sub-routines wherever possible. (AL) (AB) (GE) (DE) • Understands the difference between 'While' loop and 'For' loop, which uses a loop counter. (AL) (AB) • Understands and uses two dimensional data structures. (AB) (DE)	• Performs operations using bit patterns e.g. conversion between binary and hexadecimal, binary subtraction etc. (AB) (AL) (GE) • Understands and can explain the need for data compression, and performs simple compression methods. (AL) (AB) • Knows what a relational database is, and understands the benefits of storing data in multiple tables. (AB) (GE) (DE)	• Has practical experience of a small (hypothetical) low level programming language. (AB) (AL) (DE) (GE) • Understands and can explain Moore's Law. (GE) • Understands and can explain multitasking by computers. (AB) (AL) (DE)	• Understands the hardware associated with networking computer systems, including WANs and LANs, understands their purpose and how they work, including MAC addresses. (AB) (AL) (DE) (GE)	• Understands the ethical issues surrounding the application of information technology, and the existence of legal frameworks governing its use e.g. Data Protection Act, Computer Misuse Act, Copyright etc. (EV)

Computational Thinking Concept: AB – Abstraction; DE – Decomposition; AL – Algorithmic Thinking; EV – Evaluation; GE – Generalisation

Note: Each of the Progression Pathway statements is underpinned by one-or-more learning outcomes (due for publication in 2014), providing greater detail of what should be taught to achieve each Progression Pathway statement and National Curriculum point of study. Reviewed by Simon Humphreys and Sue Sentance of Computing At School, CAS Master Teachers, and by teachers and academics from the wider CAS community. Computational thinking mapping undertaken by Mark Dorling, Cynthia Selby and John Woollard.

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

DOSE
DEVELOPMENT OF STEAM EDUCATION

CONCEPTS

Common programming concepts also translate outside programming – they are the “things”.

Sequences	“series of individual steps or instructions that can be executed by the computer”
Loops	“a mechanism for running the same sequence multiple times”
Events	“one thing causing another thing to happen”
Parallelism	“sequences of instructions happening at the same time”
Conditionals	“ability to make decisions based on certain conditions, which supports the expression of multiple outcomes”
Operators	“support for mathematical, logical, and string expressions, enabling the programmer to perform numeric and string manipulations”
Data	“storing, retrieving, and updating values”
Input and Output	specific inputs results consistently in specific outputs inside the programs (i.e. functions) and outside while using the program

Funded by the Erasmus+ Programme of the European Union

NACIONALINĖ ŠVIETIMO AGENTŪRA

PRACTICES

Practices refer to the processes of construction – i.e. the action.

Being incremental and iterative	“Designing a project is an adaptive process, one in which the plan might change in response to approaching a solution in small steps.”
Testing and debugging	“It is critical for designers to develop strategies for dealing with – and anticipating – problems.”
Reusing and remixing	“Building on other people’s work has been a longstanding practice in programming.” “Reusing and remixing support the development of critical code-reading capacities and provoke important questions about ownership and authorship.”
Abstracting and modularizing	“Building something large by putting together collections of smaller parts is an important practice for all design and problem solving.”
Predictive thinking	Outputs are predicted while programming and compared to the actual outputs of the program to see if the program works the way it should.
Reading, interpreting and communicating code	To be able to read und understand the code is necessary in programming (especially in debugging) and communicating it to others in computational terms is needed when working with others.
Multimodal design	Using different medias (such as sound and movies) in programs.

Funded by the Erasmus+ Programme of the European Union

NACIONALINĖ ŠVIETIMO AGENTŪRA

DOSE
DEVELOPMENT OF STEAM EDUCATION

PERSPECTIVES

Perspectives describe shifts in perspectives while programming.

Expressing	“Computational thinker sees computation as something they can use for design and self-expression.”
Connecting	“Creativity and learning are deeply social practices: there is value of creating with others and value of creating for others in designing computational media.”
Questioning	“Computational thinker feels empowered to ask questions about and with technology: they don’t feel disconnected from the complex technologies of everyday life.”
User interaction	When the interaction between the user and the computer or program is taken into account while designing and programming, the programs will be more intuitive, user friendly and accessible.

Funded by the Erasmus+ Programme of the European Union

NACIONALINĖ ŠVIETIMO AGENTŪRA

DOSE
DEVELOPMENT OF STEAM EDUCATION

CONNECTING CT TO STEAM

CT is tightly related to programming via the elements of programs and the act of programming, and programming can be used in relation to any subject!

- programming your own calculator with complicated functions (Mathematics)
- programming drawings and animations (Art)
- and so on!

CT is also related to thinking about, using and developing technology which, in todays society, we can find every where.

- thinking about how devices and apps are made
- using different devices and other pieces of technology (for example as part of science project)
- developing new devices and apps – and new ways to use them!

Funded by the Erasmus+ Programme of the European Union

NACIONALINĖ ŠVIETIMO AGENTŪRA

Other ideas on how CT connects to STEAM?

Discuss!

Funded by the Erasmus+ Programme of the European Union

CT IN STEAM IN PRACTICE – EXAMPLES

Math & CT Learning Paths in ViLLE

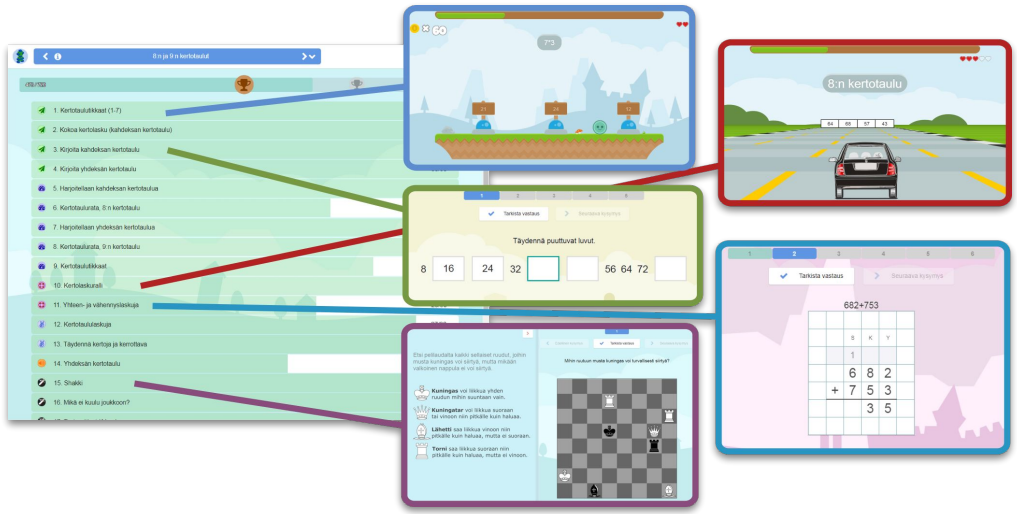
ScratchMaths project

Funded by the Erasmus+ Programme of the European Union

NACIONALINĖ ŠVIETIMO AGENTŪRA

WHAT IS VILLE?

- #1 Digital Learning Environment in Finland
- 20+ Years of Finnish educational excellence with gamification and AI, various subjects & topics
- Up to 3 hours more active learning per week
- SaaS: Software as a Service (web-platform)



Turku Research Institute for Learning Analytics trila.fi

Laakso, MJ., Kaila, E. & Rajala, T. **ViLLE – collaborative education tool: Designing and utilizing an exercise-based learning environment.** Educ Inf Technol 23, 1655–1676 (2018). https://doi.org/10.1007/s10639-017-9659-1

Funded by the Erasmus+ Programme of the European Union

NACIONALINĖ ŠVIETIMO AGENTŪRA

DIGITAL AND GAMIFIED LEARNING PATH IN MATHEMATICS & COMPUTATIONAL THINKING

Grades: 1 – 12, **aligned with any curricula**

Weekly lessons: **457** (40+ per grade level)

Exercises: **17 973** (25 - 35 per lesson)

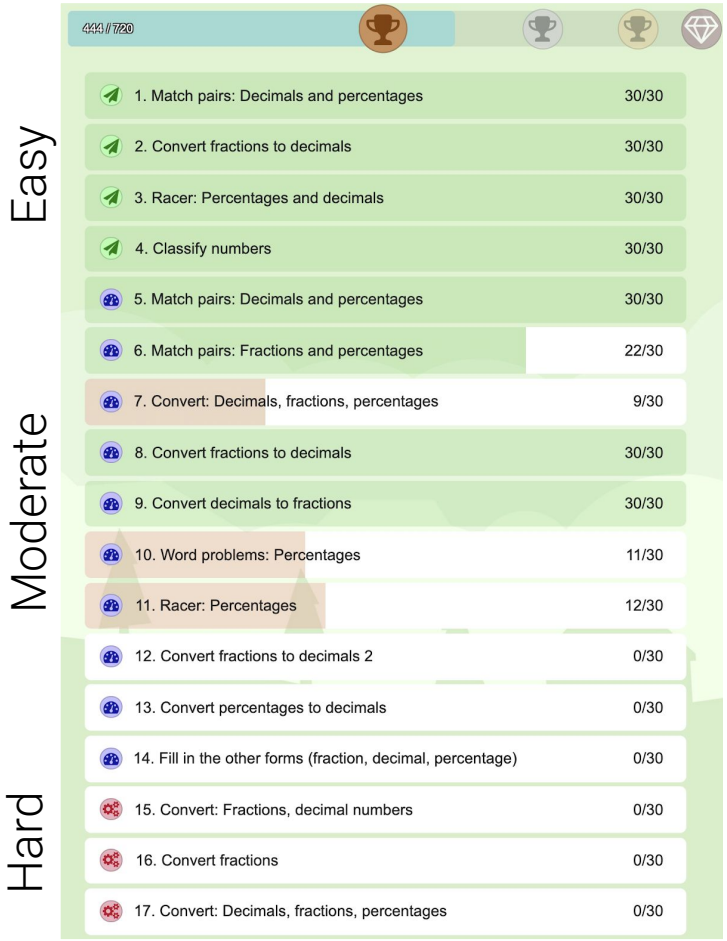
Co-designed and co-created with the teachers:
- Weekly ready-made-lessons for teachers
- **Computational thinking tasks included!**
- Personalization and differentiation made easy
- **Integrates to teachers existing workflows!**

Funded by the Erasmus+ Programme of the European Union

NACIONALINĖ ŠVIETIMO AGENTŪRA

BASIC CONCEPT

- 1 lesson per week + homework
  - 25-35 exercises, 350-500 tasks
  - 45-90 minutes to complete a lesson
  - Computational thinking & logical exercises
- Active learning & gamification & continuous assessment
- Easy personalization & differentiation for learners
- Integrates into teacher existing work!
- Also used in Bebras Challenge that promotes Informatics and Computational Thinking!

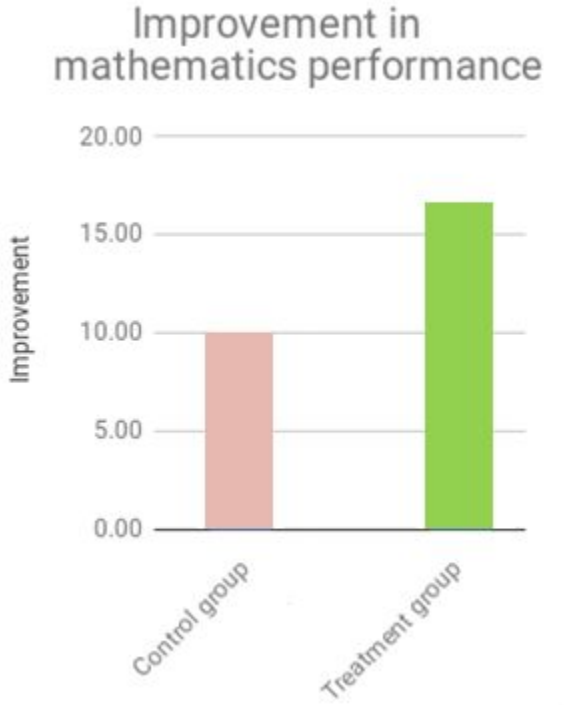


Pluhár, Z. et al. (2022). **Bebras Challenge in a Learning Analytics Enriched Environment: Hungarian and Indian Cases.** In: Bollin, A., Futschek, G. (eds) Informatics in Schools. A Step Beyond Digital Education. ISSEP 2022. Lecture Notes in Computer Science, vol 13488. Springer, Cham. https://doi.org/10.1007/978-3-031-15851-3_4

Funded by the Erasmus+ Programme of the European Union
NACIONALINĖ ŠVIETIMO AGENTŪRA

RESEARCH BASED: IMPACT ON STUDENTS RESULTS

- In 15 weeks study (3rd grade):
  - student performance improved 12% (39% more than in control group) and
  - arithmetic fluency improved 11 % (45% more than in control group).
- Children chose to work more than 50% extra at home and during weekends (3rd grade). 71% less mistakes overall (2nd grade).
- Marks increased by one whole mark on average. The improvement was permanent as observed over 2 years (grades 5-6, 15-20%).

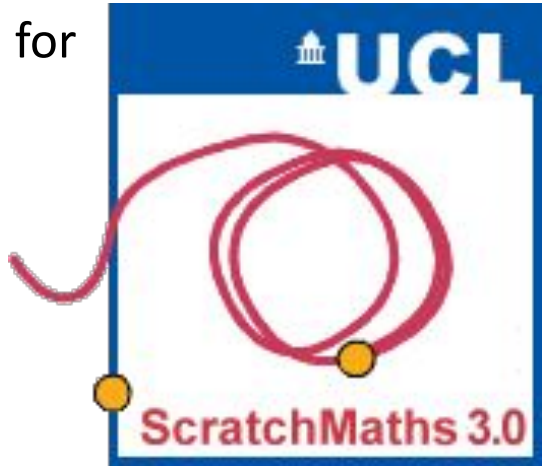


15 week study, 3rd gr, Mar 2018

Funded by the Erasmus+ Programme of the European Union
NACIONALINĖ ŠVIETIMO AGENTŪRA

SCRATCHMATHS

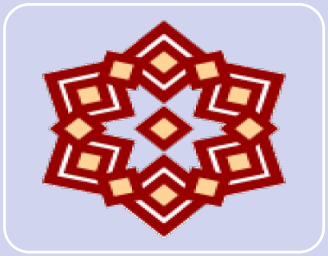
- ScratchMaths is a two-year computing and mathematics-based curriculum for grades 5 and 6.
- Its aim is to enable pupils to engage with and explore important mathematical ideas through learning to program.
- It uses Scratch – a free online programming environment.



https://www.ucl.ac.uk/ioe/research/projects/ucl-scratchmaths/ucl-scratchmaths-curriculum

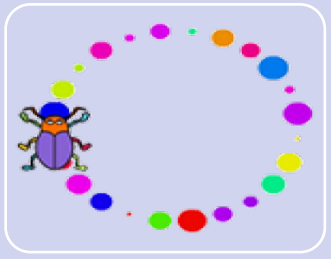
Funded by the Erasmus+ Programme of the European Union
NACIONALINĖ ŠVIETIMO AGENTŪRA

SCRATCHMATHS MODULES



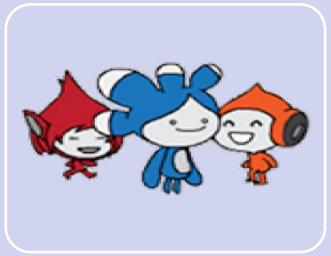
1: Tiling patterns

- Theme: repeating patterns
- CT: sequence, loops, algorithms, debugging
- Math: symmetry, angles, negative numbers



2: Beetle geometry

- Theme: creating drawings
- CT: sequence, loops, randomness, expressions, initialisation
- Math: geometry

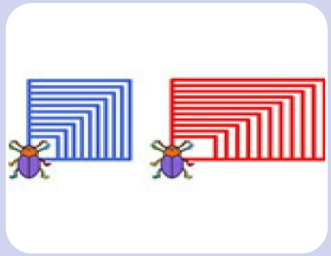


3: Interactive sprites

- Theme: interactive behaviours between multiple sprites
- CT: parallelism
- Math: coordinates, multiplication, factors

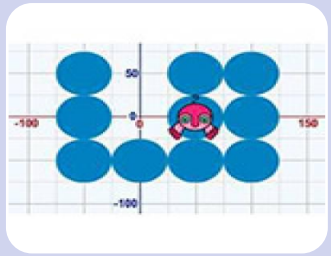
4: Building with numbers

- Theme: exploring place value
- CT: broadcasting
- Math: place values



5: Exploring mathematical relationships

- Theme: exploring poportionality and ratio
- CT: variable
- Math: poportionality, ratio



6: Coordinates and geometry

- Theme: exploring coordinates
- CT: variable
- Math: coordinates, scale

Funded by the Erasmus+ Programme of the European Union

NACIONALINĖ ŠVIETIMO AGENTŪRA

GET FAMILIAR WITH SCRATCHMATHS

- Choose a module and get to know it!
  - If programming or Scratch is not familiar, choose one from the beginning. Otherwise choose something that interest you.
- Would you use the module? Would you include everything in the module? Would your students like the module?
- Discuss with partner!
- https://www.ucl.ac.uk/ioe/research/projects/ucl-scratchmaths/ucl-scratchmaths-curriculum

Funded by the Erasmus+ Programme of the European Union

NACIONALINĖ ŠVIETIMO AGENTŪRA

3D printing and creative coding workshop concept

Tools and Materials:

- PCs or laptops for the students
- BlocksCAD: Free, ''Programmable'' CAD tool with the use of blocks, similar to the environment of Scratch (https://www.blockscad3d.com/editor/)
- Cura: Open-source slicing application for 3D printers (https://ultimaker.com/software/ultimaker-cura)
- 3D printer: Any hobbyist/DIY 3D printer is sufficient
- Filament: The suggested filament for most projects is PLA
- Writing utensils (pencils, paper sheets, sticky notes, etc.) for ideation/drafts
- CAD and Coding commands cheat sheet

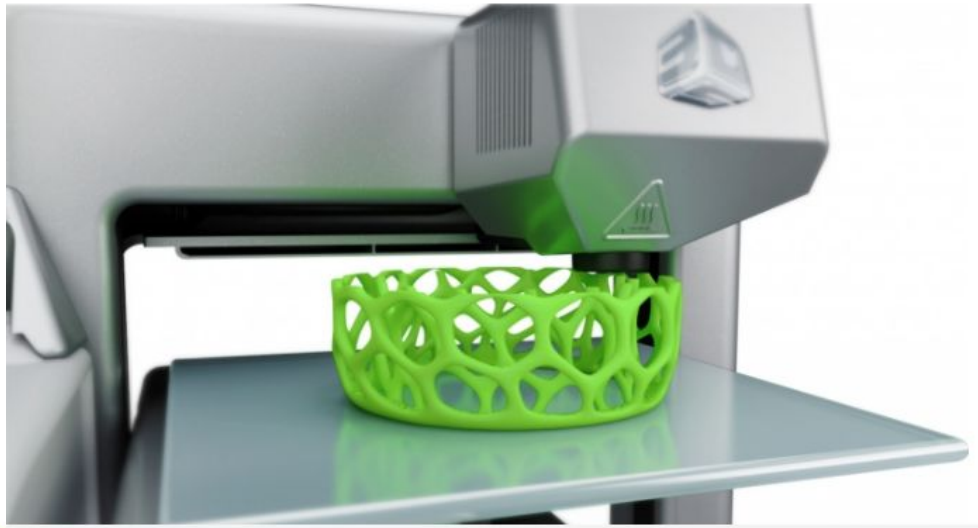
Teacher:

- Acts as facilitator, not an instructor

Aim

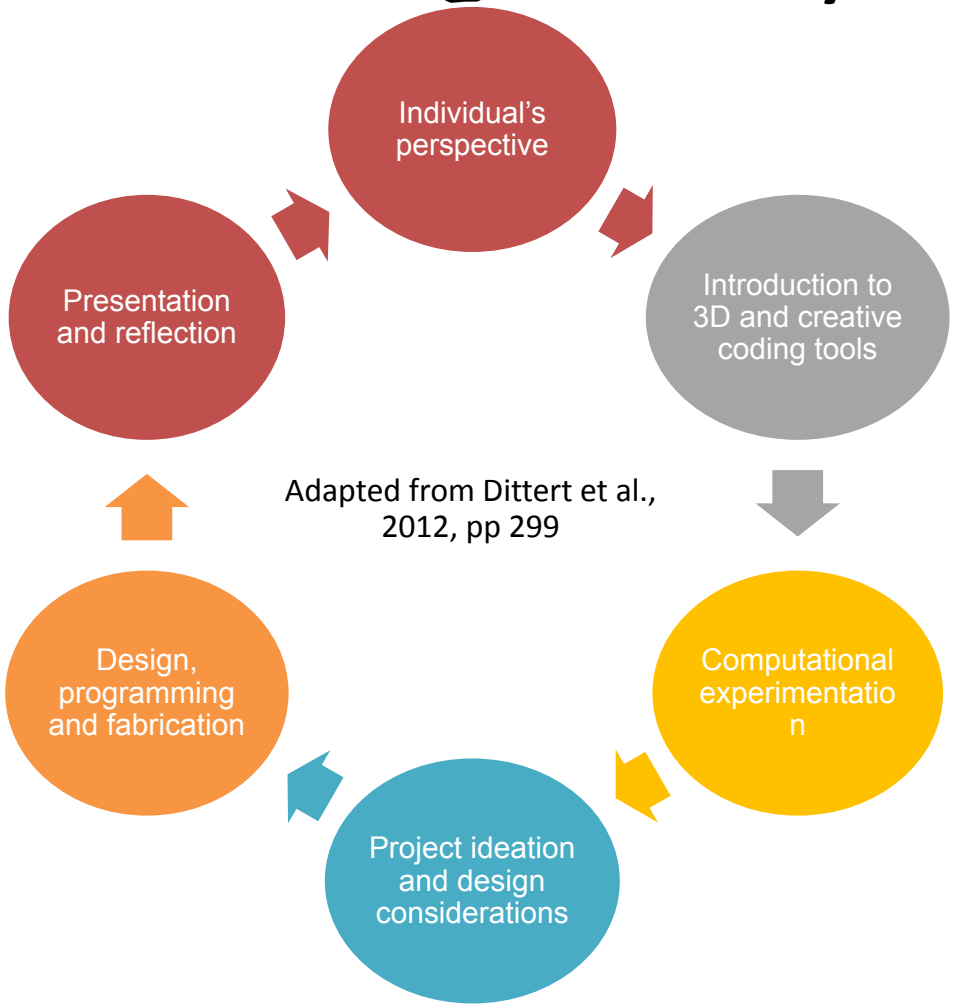
Students become designers and learn how to overcome challenges in STEAM to create personally meaningful artifacts

They are encouraged to use mathematics concepts, coding and computational thinking aspects through visualization to design 3D models that can be fabricated



Funded by the Erasmus+ Programme of the European Union

Concept of the learning activity



Tools, previous projects, materials
Computational experimentation

Experimentation with CAD and creative coding commands through short tutorials and examples

Brainstorming, drafts, design considerations based on their experience with computational experimentation

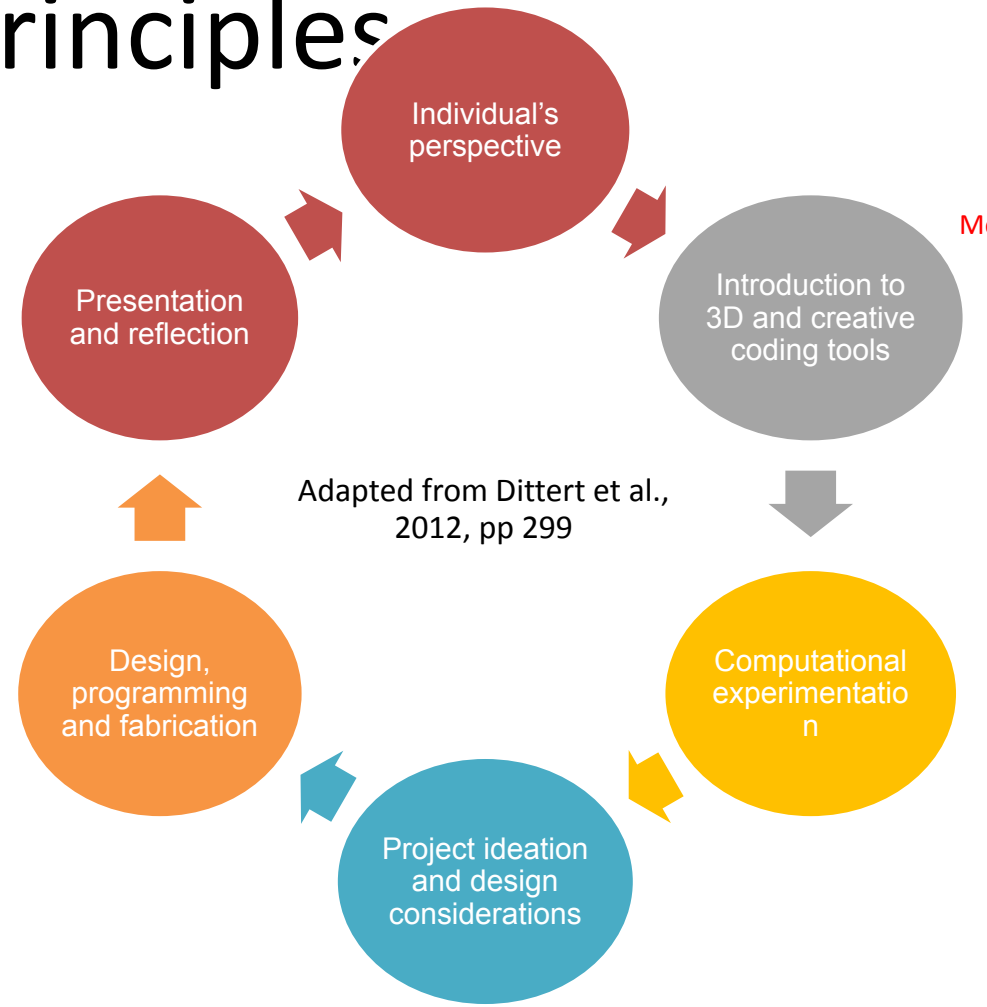
Making phase, tackling technical issues to design and fabricate the intended model

Reflection of the learning activity, exchange ideas and feedback

Adapted from Dittert et al., 2012, pp 299

Funded by the Erasmus+ Programme of the European Union

Underlying principles



Adapted from Dittert et al., 2012, pp 299

Tools, previous projects, materials

Computational experimentation

Motivate and engage individuals in project-based learning activities

Experimentation with CAD and creative coding commands through short tutorials and examples

Learning through trial and error, learning-by-doing

Brainstorming, drafts, design considerations based on their experience with computational experimentation

Collaboration, peer-to-peer learning critical thinking

Making phase, tackling technical issues to design and fabricate the intended model

Interdisciplinary, design-based learning, Be-greifbarkeit

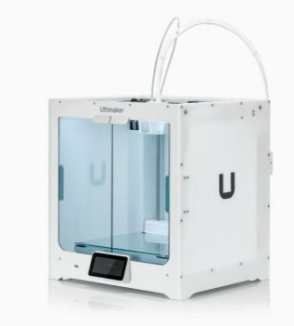
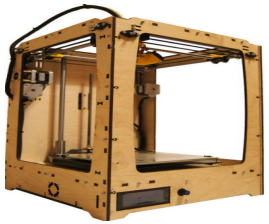
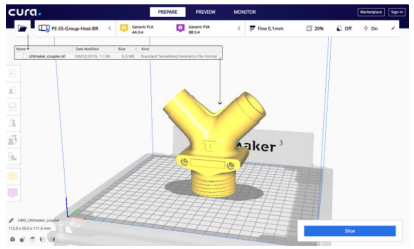
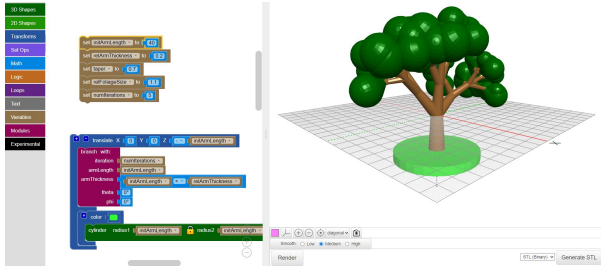
Reflection of the learning activity, exchange ideas and feedback

Experiential learning, critical pedagogy

Funded by the Erasmus+ Programme of the European Union

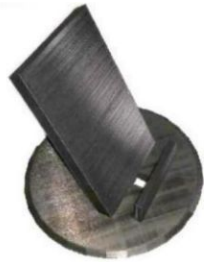
Introduction to 3D printing and creative coding tools

- **Tools**

- **Previous projects**

Including projects with computational concepts (e.g., iterations for the creation of repetitive patterns)

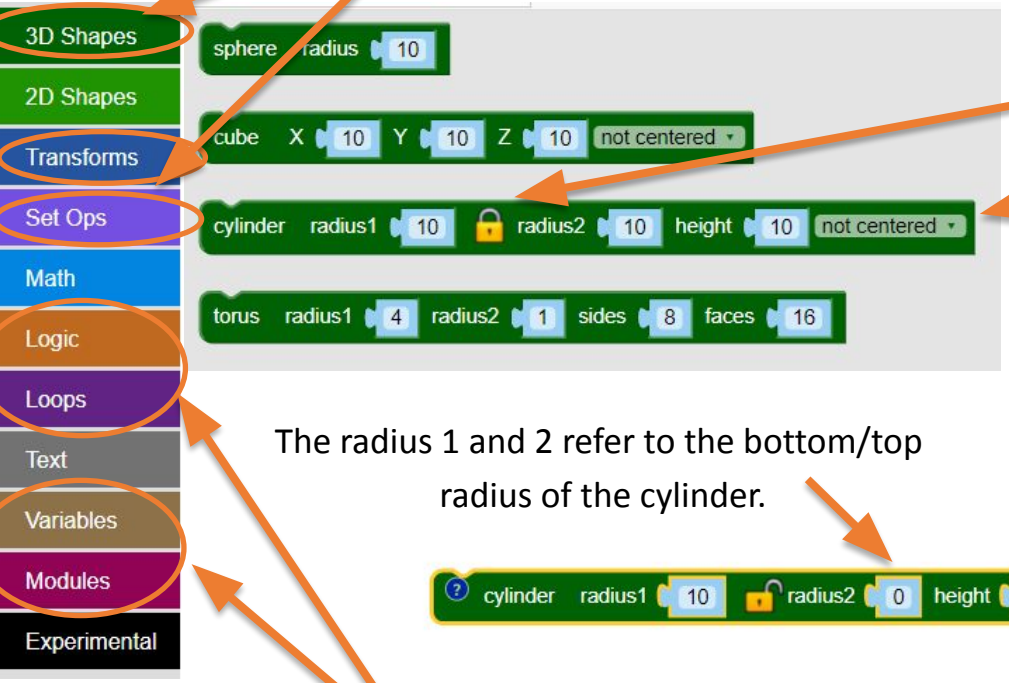
Including simple projects with the traditional CAD commands (shapes, transformations and CSG operations)

- **Materials**


Funded by the Erasmus+ Programme of the European Union

Getting started: Computational experimentation with simple CAD commands

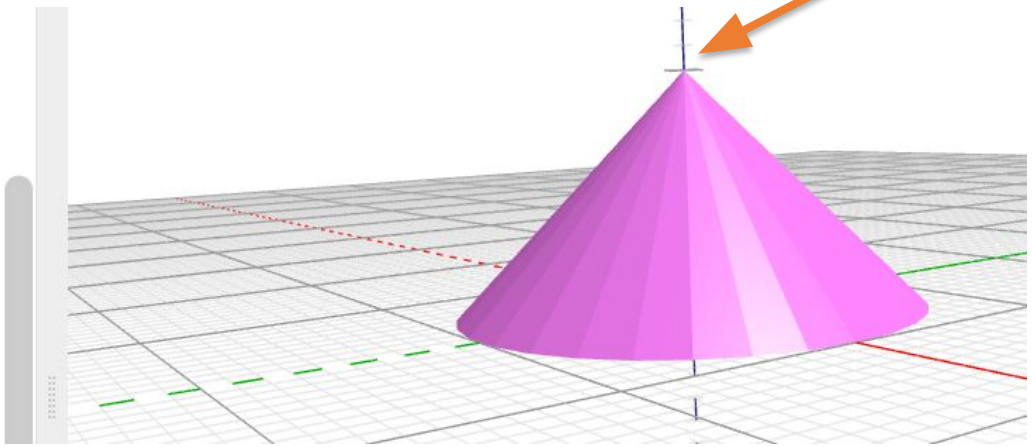
CAD commands



The locker icon allows the proportionally (or non) change of a radius

Center is O (0,0,0)

radius 2 is 0



The radius 1 and 2 refer to the bottom/top radius of the cylinder.

Computational concepts

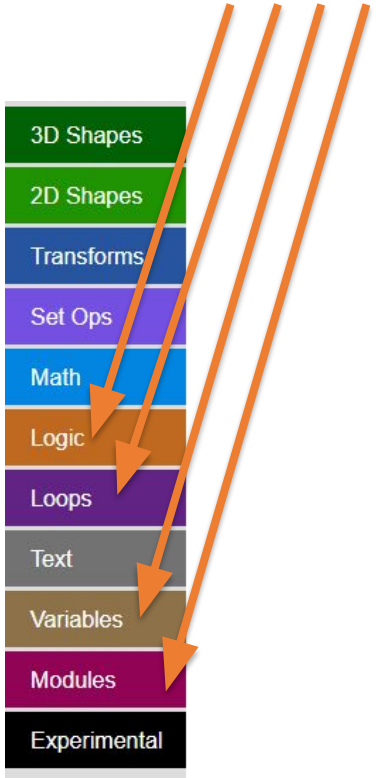
Funded by the Erasmus+ Programme of the European Union

NACIONALINĖ ŠVIETIMO AGENTŪRA

Radboud Universiteit

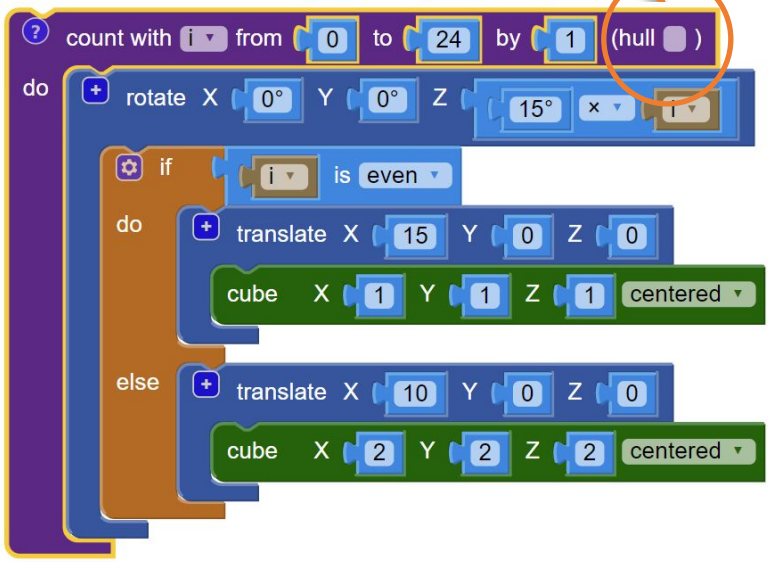
Computational experimentation with computational concepts (loops, conditional statements, modules/functions)

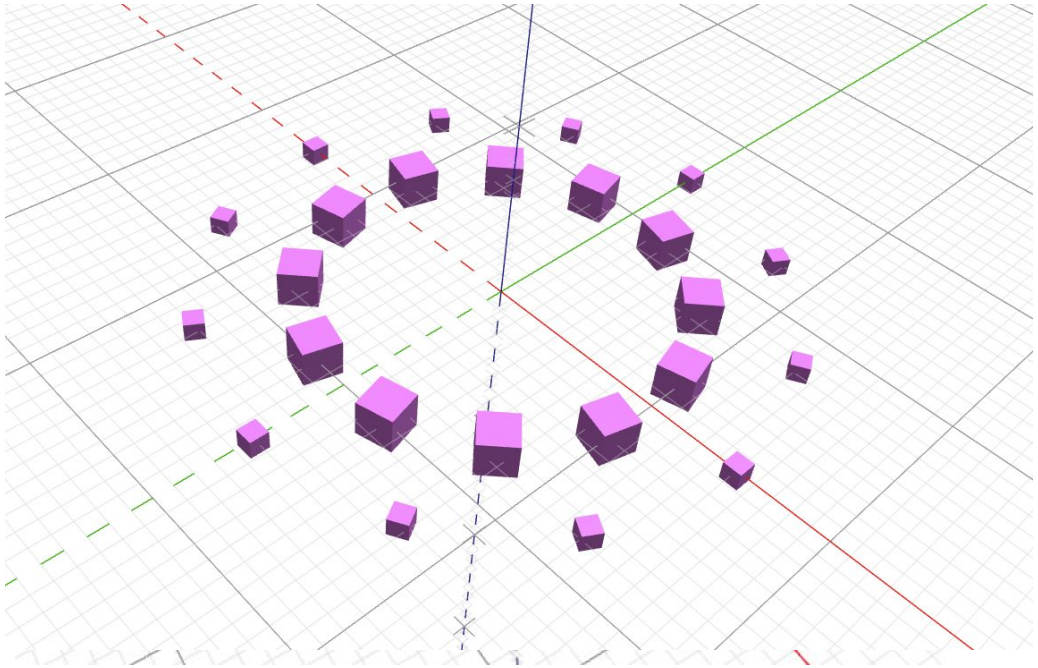
Navigation menu for including core CS concepts



Checking the "hull" option in the loop does pairwise hulls between iterations n and n+1. It can handle 2D or 3D input.

Hulled loop

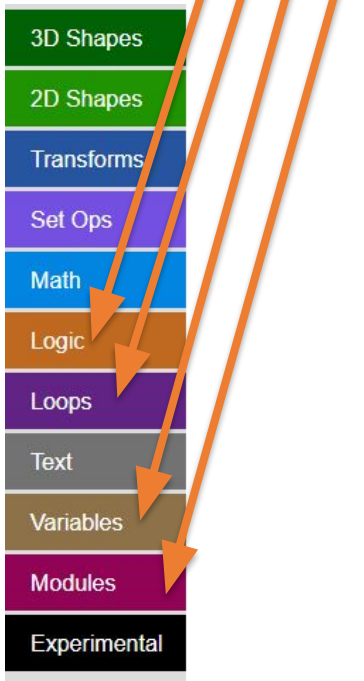




Funded by the Erasmus+ Programme of the European Union
NACIONALINĖ ŠVIETIMO AGENTŪRA
Radboud Universiteit

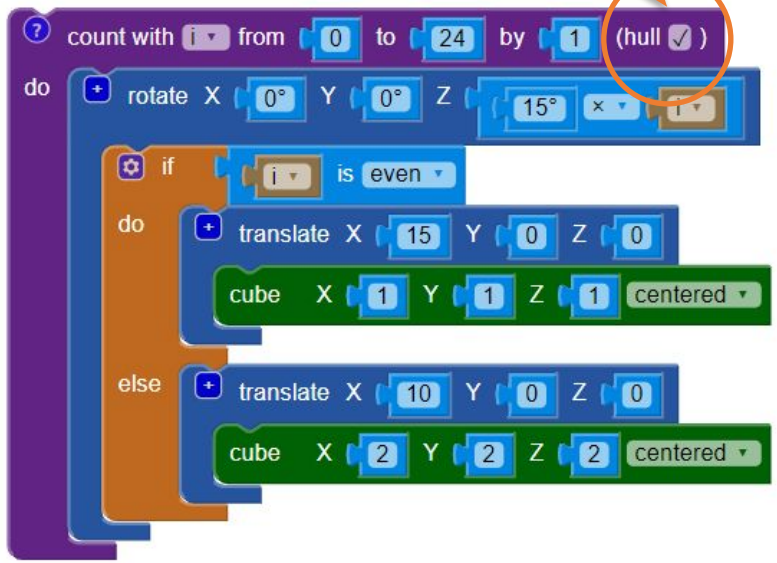
Computational experimentation with computational concepts (loops, conditional statements, modules/functions)

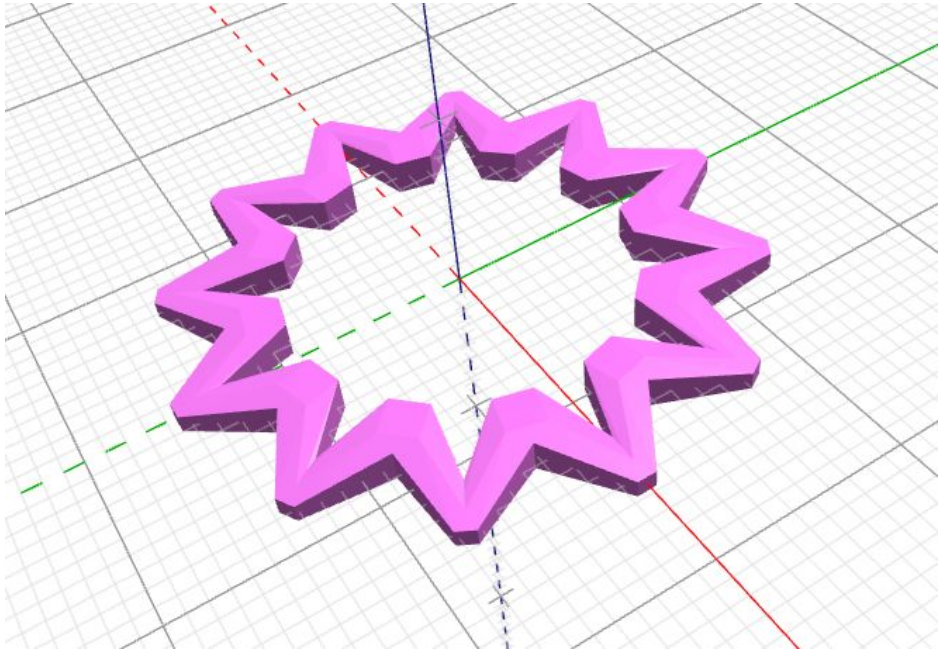
Navigation menu for including core CS concepts



Checking the "hull" option in the loop does pairwise hulls between iterations n and n+1. It can handle 2D or 3D input.

Hulled loop

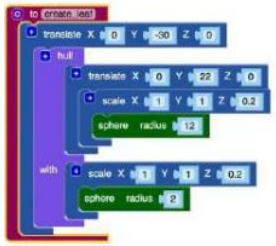


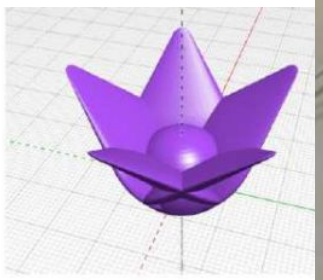


Funded by the Erasmus+ Programme of the European Union
NACIONALINĖ ŠVIETIMO AGENTŪRA
Radboud Universiteit

Project ideation and design considerations

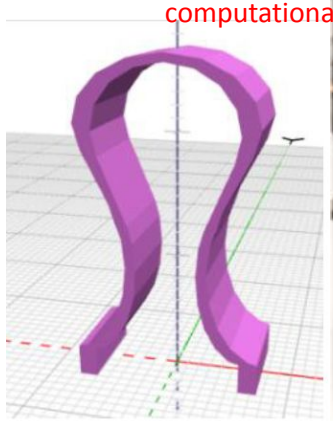
- Type of project/Purpose of project (e.g., daily use, decoration)
- Resources (materials, information requirements)
- Challenges (know-how-to, functionality)
- Limitations (materials, shape)
- Aesthetical choices (e.g., color of filament and choosing decorative patterns)

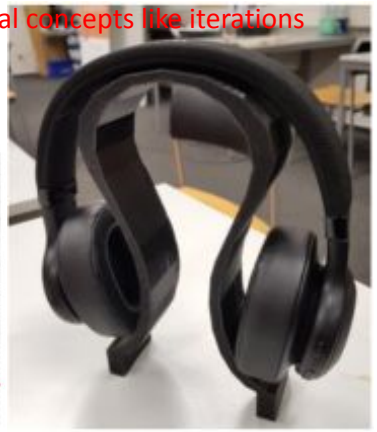




- Intended for decoration
- Transparent material
- Includes computational concepts

- Intended for daily use
- Measurement and consideration of the headphones dimensions
- Does not require the use of computational concepts like iterations



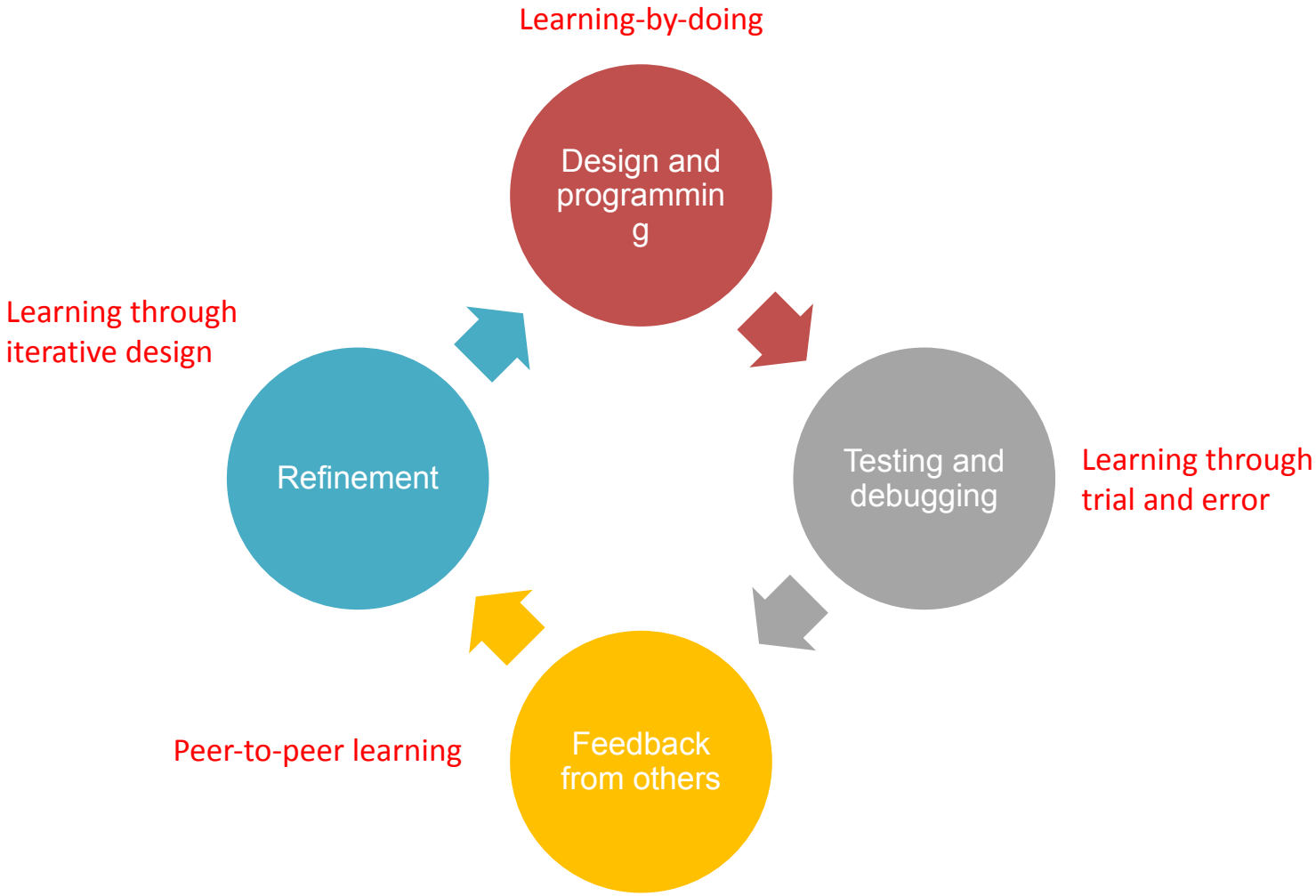


Funded by the Erasmus+ Programme of the European Union

NACIONALINĖ ŠVIETIMO AGENTŪRA

Radboud Universiteit

Design, programming and fabrication



Funded by the Erasmus+ Programme of the European Union
NACIONALINĖ ŠVIETIMO AGENTŪRA
Radboud Universiteit

Presentation and reflection

The final artifact of the learning activity:

- A product that can be shared and discussed with others
- Has a quality that it is comparable to that of industrial products
  - empowers individuals in making things
- A potentially personally meaningful product
  - used for personal and creative expression

Funded by the Erasmus+ Programme of the European Union

Connections with STEAM content

- **Science**: working in the 3D space, considering attributes of the 3D model (material, weight, structure, shape)
- **Technology**: (programmable and non-programmable) CAD design and 3D printing
- **Engineering**: CAD/computational design, prototyping, programming, functionality of final product, design scaling, design thinking
- **Art**: Creative expression, decorations, artistic artifacts, design aspects
- **Mathematics**: use of mathematical operations for the creation of simplex and more complex patterns and geometry in the 3D space.

Funded by the Erasmus+ Programme of the European Union

A Step-by-step tutorial (that still provides some space for creativity and personalization)

I. Forms

1. Use the sphere block and create a sphere model with a radius of 28 units of length.

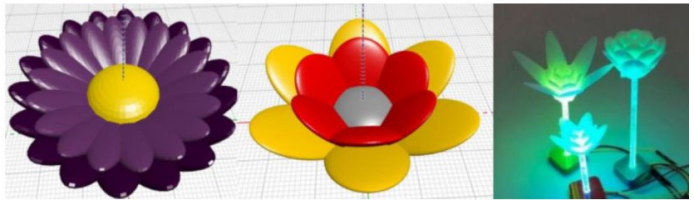
2. Can you model a box that has a height of 40, a length of 20, and a width of 10 units? Which side belongs to which axis in the coordinate system?

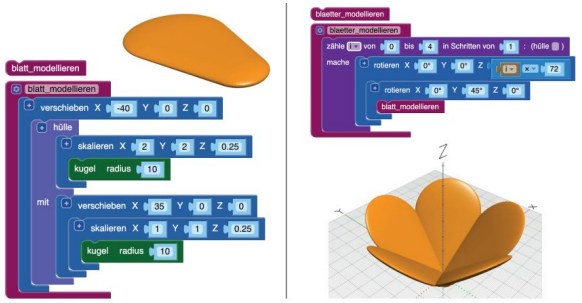
3. Examines the cylinder block (under menu item "3D shapes").

    a. What happens if you set the "Radius 2" to a value that is different from the "Radius 1" is different?

    b. What influence does the lock have on the cylinder block?

    c. What happens if you change the value for "not centered" to "centered"?





Funded by the Erasmus+ Programme of the European Union

NACIONALINĖ ŠVIETIMO AGENTŪRA

Radboud Universiteit

A Step-by-step tutorial (that still provides some space for creativity and personalization)

II. Transformations

4. Use a sphere block into the work surface.

a. Now use the scale command (under the menu item "Transformations") to stretch it so that it looks like an Easter egg. To do this, set the z-value of the scale command to a value that is is greater than 1 (e.g. 2).

b. What happens if you use a number for the z-value that is smaller than 1? (e.g. 0.2)? Make sure that you use a dot ‘’.’’ instead of commas.

c. Now adjust the shape of the sphere on the y-axis, so that it is looks like a flat leaf of a flower.

5. Use the sheet you designed in 4c. Which block do you need to move it 10 units of length along the y-axis?

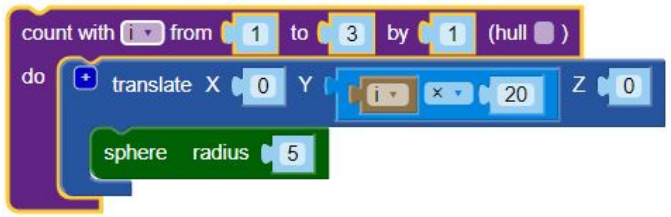
Hint: Look at the section "Transformations"

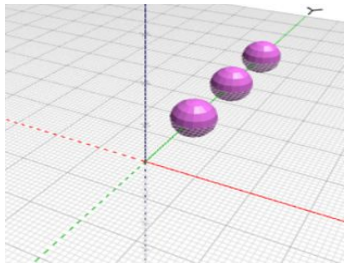
Funded by the Erasmus+ Programme of the European Union
NACIONALINĖ ŠVIETIMO AGENTŪRA
Radboud Universiteit

Step-by-step tutorial that still provides space for creativity and personalization

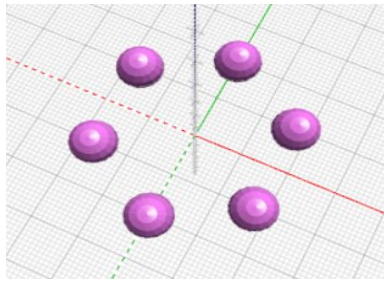
III. Loops and variables

6. For now, deactivate your sheet from task 5. Click with the right mouse button on the respective blocks and select the item "Deactivate block". Now you will be shown how to use the so called **loops** and **variables**, to create multiple objects of the same type (as shown in the image). The three spheres from task 3 with a radius of 5 units of length, all of which are 20 units of length apart (see picture), can be calculated using the following command:





7. How can you use this function to create a circular arrangement (as in the following figure)?



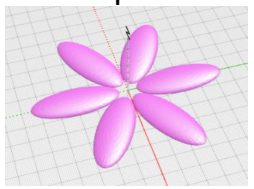
Hint: You need to create a "rotate" - block and insert the instruction with the "i" into it.

Funded by the Erasmus+ Programme of the European Union
NACIONALINĖ ŠVIETIMO AGENTŪRA
Radboud Universiteit

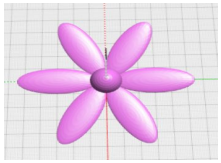
A Step-by-step tutorial (that still provides some space for creativity and personalization)

8. Now activate your flower leaf again, which you deactivated in task 6, (Right-click on the block and click "Activate block"). Now swap the green ball block in the loop against the blocks that create your leaf. What's happening?

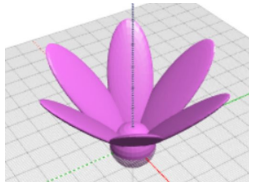
9. Change the value of the parameters in the loop and scale commands so that generated result is that of flower. A form could be similar to the one below, but the shape and number of the individual petals are up to you!



10. Flowers usually have a flower bud in the middle. How can you find this addition?



11. At which point do you have to add a rotate block so that the leaves of the flower are pointing up?



In order for the 3D printer to recognize the several parts that consist the flower as one 3D model, and print it altogether, you have to include all the blocks inside the Union block, under the Set Ops block.

Funded by the Erasmus+ Programme of the European Union
NACIONALINĖ ŠVIETIMO AGENTŪRA
Radboud Universiteit

Reflection

- What did go well and what did not in the activity?
- What did you like the most and what did you like the least?
- What did you learn during the workshop?
- Any suggestions for improving the learning activity?
- Did you manage to design what you intended? If not what were the reasons?
- What aesthetic/design elements would you like to improve or add to your model?

Funded by the Erasmus+ Programme of the European Union
NACIONALINĖ ŠVIETIMO AGENTŪRA
Radboud Universiteit

Further Discussion (Optional)

- Which aspects of this activity could address aspects of your respective STEAM subject?
- What is the role of the teacher in a ‘’computational making’’ learning activity? How can the teacher support the students when inexperienced with 3D printing and creative coding?
- What do you think about predefined and free-choice projects?

Predefined topics

Pros	Cons
Content specific/ easier aligned with existing curricula	Predefined topic might not be aligned with learner’s interests
Computationally rich projects	Learning activity might feel like step-by-step instruction
More efficient preparation for teachers/tutors	Less opportunities for personal and creative expression
There is still some space for personalization of projects (by adjusting the parameters and the relationships between them)	
Structuring the duration of the learning	

(To be completed/discussed by the activity participants)

Free-choice topics

Pros	Cons
Learning by creating personally meaningful projects – More aligned with students’ interests	Time management in different levels of the learning activity (e.g., structuring the learning activity, preparation)
More motivation for tackling the problems that inevitably occur	Inexperienced tutors/facilitators might have difficulties to support learners
Real-world/authentic problem solving	Some projects might provide less opportunities for STEAM learning
More opportunities for creativity, innovation, critical thinking, personal growth	

(To be completed/discussed by the activity participants)

Funded by the Erasmus+ Programme of the European Union
NACIONALINĖ ŠVIETIMO AGENTŪRA
Radboud Universiteit

DOSE

DEVELOPMENT OF STEAM EDUCATION

Funded by the
Erasmus+ Programme
of the European Union

NACIONALINĖ
ŠVIETIMO
AGENTŪRA