Games for Learning Algorithmic Thinking

Workshop Syllabus and Materials

Co-funded by the
Erasmus+ Programme
of the European Union

Group of Authors
For the *University of Rijeka, Department of Informatics*
Nataša Hoić-Božić, Martina Holenko Dlab, Marina Ivašić-Kos, Ivona Franković

For the *University of Rijeka, Faculty of Teacher Education*
Jasminka Mezak, Petra Pejić Papak

For the *Tallinn University, Centre for Educational Technology*
Mart Laanpere, James Sunney Quaicoe, Viktoria Humal

For the *Ss. Cyril and Methodius University in Skopje, Faculty of Computer Science and Engineering*
Ana Madevska Bogdanova, Katerina Zdravkova, Vladimir Trajkovik

For the *University of Ljubljana, Faculty of Education*
Jože Rugelj, Irena Nančovska Šerbec, Mateja Bevčič, Anja Luštek

For the *South-West University „Neofit Rilski", Faculty of Mathematics and Natural Sciences*
Daniela Tuparova, Kostadin Samardziev, Ivanichka Nestorova

Editors
Nataša Hoić-Božić (*University of Rijeka, Department of Informatics*)
Jasminka Mezak (*University of Rijeka, Faculty of Teacher Education*)
Martina Holenko Dlab (*University of Rijeka, Department of Informatics*)

English Language Proofreading
Ivanichka Nestorova (*South-West University „Neofit Rilski"*)

Graphic Design and Digital Processing
Martina Holenko Dlab (*University of Rijeka, Department of Informatics*)

Contact
glat@inf.uniri.hr

Preface

This publication is part of the educational material created in the context of the Erasmus+ project GLAT - "Games for Learning Algorithmic Thinking".

The general goal of the project is encouraging the integration of computational and algorithmic thinking, problem-solving skills, logic and creativity into the daily teaching through different subjects in students' younger ages in a fun and attractive way using Game Based Learning (GBL). One of the main activities of the project was the organization of education for primary junior grade teachers in the form of a blended learning e-course.

This publication is a syllabus of education designed during the project GLAT. The emphasis is placed on the f2f (classroom-based) workshops, which are combined with online learning during which the teachers are mentored by the experts who conduct the education.

The first part of the *Workshop Syllabus* provides general information on GLAT education, which includes the main goals of the education, expected learning outcomes, the target audience, and the required background knowledge. It is also stated that three two-day workshops with a total duration of 48 school hours (45 minutes each) and the use of the Moodle Learning Management System for the online part of the course are predicted during the training.

The second part of the publication lists the schedules for all three workshops: *Workshop 1: Game-Based Learning (GBL) and Unplugged Activities*, *Workshop 2: Problem Learning (PBL), Online Quizzes and Logic Tasks*, and *Workshop 3: Games and Tools for Programming*. The schedules provide for each of the sessions of the workshops: learning outcomes, topics (with handouts of presentations for lectures), evaluation methods, and tasks for the independent work of the learners after the workshops (during the online part of the education).

Finally, in the third part of the publication, templates created for the purposes of GLAT education were added as annexes.

The syllabus presents an introduction to other content created during the project GLAT: presentations for the sessions of the workshops, the GLAT Teacher's Guide, learning scenarios prepared by teachers that serve as examples of good practice, and the GLAT Moodle e-course available after logging into the MoD learning system. The course backup can also be restored to own empty Moodle course.

It should be emphasized that GLAT education is not intended for independent learning, but it requires mentoring for the participants. Therefore, these materials will be useful to educational institutions and individual educators who wish to launch their own courses or subjects based on the syllabus and learning materials produced within the project.

Considering that the results of the project GLAT are available not only in Croatian but also in English, and under a license that allows them to be freely shared, modified and transform, we believe that they will be a useful starting point that experienced educators will be able to use and appropriately adapt to future students in their own countries.

In addition to the entire project team actively involved in the preparation of GLAT publications and all other results, the completion of the syllabus was also contributed by primary junior grade teachers from Croatia – participants of GLAT education. We thank them for their irreplaceable help in the form of active participation, hard work and suggestions.

Editors

Contents:

Part I:
Information About GLAT E-Course

1. Overall goals for GLAT e-course

- Participants will learn about innovative teaching methodologies in the ICT area such as Game Based Learning (GBL), Problem Based Learning (PBL), Inquiry Based Learning (IBL), teamwork.
- Participants will learn how to use digital didactic games (serious games) in different school subjects for encouraging algorithmic thinking, problem-solving skills, logic and creativity with their students.
- Participants will design and implement a learning scenario, a document in which the teacher develops innovative ideas to carry out educational activities by means of modern teaching methods with the use of appropriate digital content and tools, in order to carry out educational activities for encouraging algorithmic thinking.

2. Target group of participants

- Focus group of about 15-20 primary grade school teachers

3. Required background knowledge

- Basics ICT skills
- No prior knowledge of programming is required

4. Duration of the course

- up to 8 months:

Workshop 1 - 16 hours and up to 2 months for preparing the 1st assignment.
Workshop 2 - 16 hours and up to 2 months for preparing the 2nd assignment.
Workshop 3 - 16 hours and up to 4 months for preparing the 3rd, final assignment.

5. Main learning outcomes

Participants will be able to:

- Describe the principles of Game Based Learning (GBL)
- Apply digital educational games into different school subjects
- Use Web 2.0 tools for creating content for unplugged activities, e.g. posters, leaflets…
- Create learning scenarios in order to develop innovative ideas for carrying out game based unplugged activities
- Describe principles of Problem Based Learning (PBL)
- Use digital tools within the process of problem solving
- Recognize the methodology of role-playing in educational games
- Choose and create logical tasks and quizzes suitable for algorithmic thinking development in different school subjects
- Use Web 2.0 tools for creating logical tasks and online quizzes
- Create learning scenarios in order to develop innovative ideas for carrying out logical tasks and online quizzes
- Describe principles of Inquiry Based Learning (IBL)
- Understand basic concepts of programming
- Use simple game based tools for learning programming
- Create learning scenarios in order to develop innovative ideas for applying programming concepts and developing algorithmic and computational thinking through game based tools

6. Learning strategy

- The blended model of e-learning that combines face-to-face (f2f) and online teaching methods (asynchronous content delivery methods, guided design, forums and discussion boards)
- All f2f teaching methods at the workshops encourage individual activities, group activities, and whole-group discussions (in addition to teacher presentations and demonstrations).

7. Learning environment

- For the purpose of the course, an e-course is established in LMS Moodle.
- All learning materials from the workshops' f2f parts will be available on the learning platform as well as other necessary information and materials needed for the realization of the course.

8. Evaluation

- Completed versions of all the learning scenarios will be reviewed and implemented in the classrooms by the participants.
- Participants' satisfaction with the education will be measured by the questionnaire or interviews.

Part II:
Learning Outcomes and Topics for F2F Workshops

Workshop 1:
Game Based Learning (GBL) and Unplugged Activities

Workshop 1 – Game Based Learning (GBL) and Unplugged Activities

Workshop schedule

Day 1

Introduction to Workshop 1

Duration: 1 hour (45 minutes)

Introductory presentation: Introducing and explaining the main goals of the workshop, defining algorithmic thinking.
Introduction round: The participants introduce themselves
Introduction and enrolling to the e-course in Moodle LMS

Session 1: Game Based Learning (GBL)

Duration: 3 hours (135 minutes)

Lecture: Games in education
Group work: Exploring educational games and preparing a „Learning package"

Lecture: Integration of games into the lecturing process
Demonstration: Examples of simple games in different school subjects
Group work: Exploring existing educational games

Session 2: GBL with unplugged activities

Duration: 1 hour (45 minutes)

Lecture: What are unplugged activities and how to use them in the classroom?
Demonstration: Examples of unplugged activities for different school subjects, providing propaedeutic for algorithms and programming (e.g Plant a seed, Find the hidden words, Guess the number, Walking in the maze, etc.)
Group work: Discussing new examples of unplugged activities

Session 3: Using Web 2.0 tools for creating content for unplugged activities

Duration: 3 hours (135 minutes)

Presentation: Advantages of using Web 2.0 tools for unplugged activities

Group work: Exploring examples and resources
Demonstration: Creating content for unplugged activities using Web 2.0 tools (Canva, Sketchpad)
Individual work: Creating content for unplugged activities using Web 2.0 tools

Group work: Creating examples of unplugged activities for different school subjects

Day 2

Session 4: Designing learning scenarios

Duration: 2 hours (90 minutes)

Lecture: Definition of learning scenarios, how to design learning scenarios
Demonstration: Examples of scenarios in written forms (with games and unplugged activities)
Group work: Preparing learning scenarios using prepared template

Session 5: Designing learning scenarios using a graphical tool

Duration: 2 hours (90 minutes)

Presentation: Visualising learning scenarios with LePlanner
Demonstration: Examples of GBL scenarios in graphical forms
Individual work: Exploring LePlanner tool
Group work: Designing an unplugged game based learning scenario

Session 6: Designing learning scenarios for unplugged activities

Duration: 3 hours (135 minutes)

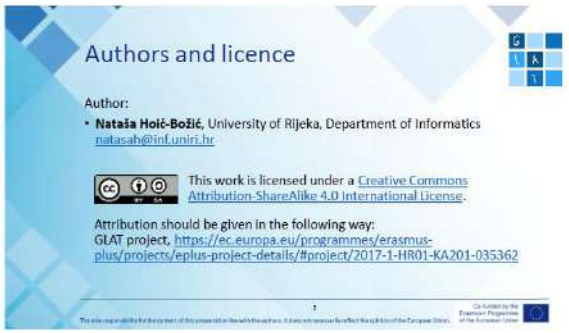
Individual work: Developing learning scenarios for carrying out an unplugged activity in written form and in graphical form using LePlanner (developing the first version of the 1st learning scenario)
Group work: Review and discussion about the developed scenarios

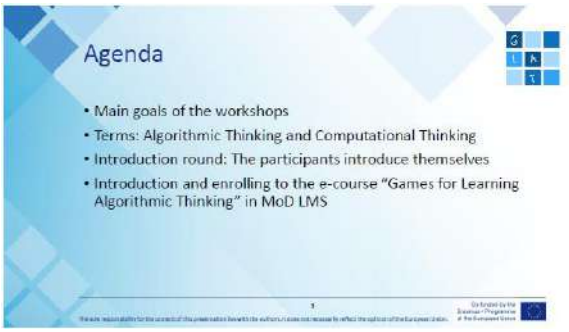
Conclusion of Workshop 1

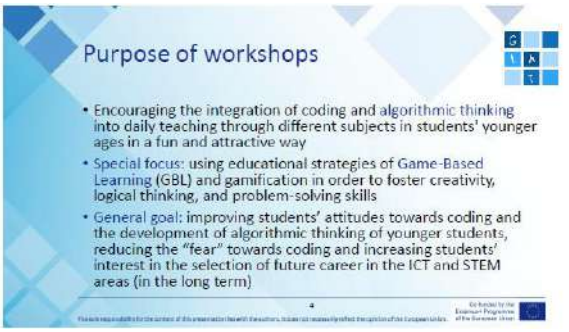
Duration: 1 hour (45 minutes)

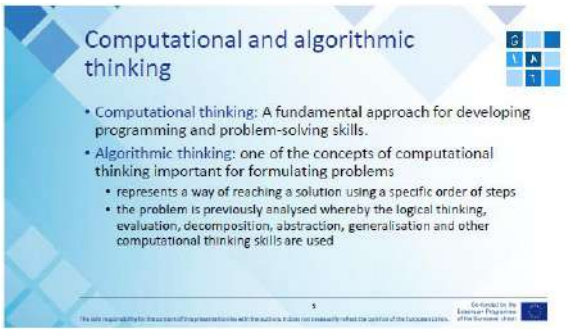
Whole-group activity: Debriefing
Closing talk: Introducing and explaining the follow-up activities (developing the 1st learning scenario for an unplugged activity)

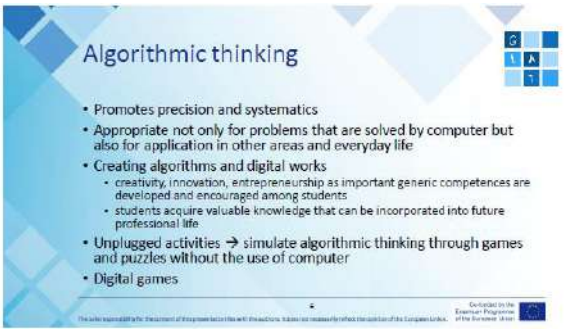
Presentation: Introduction to Workshop 1

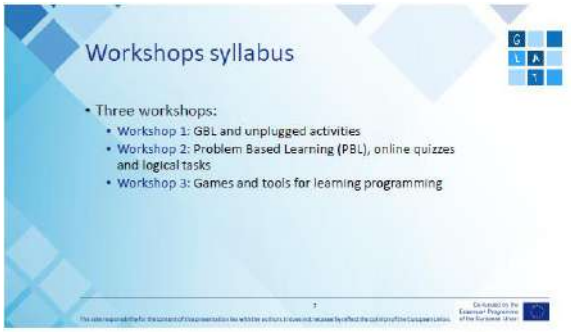


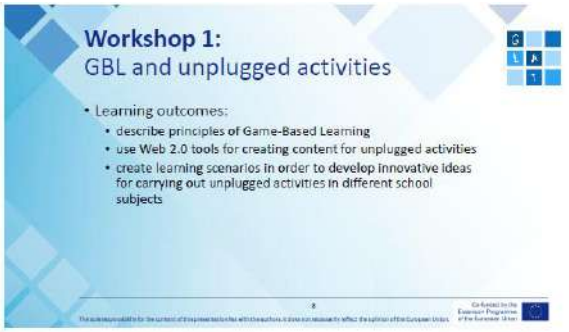


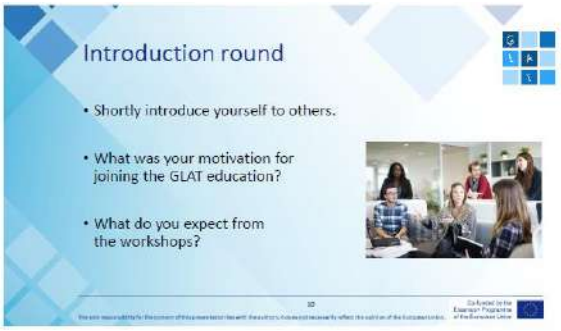


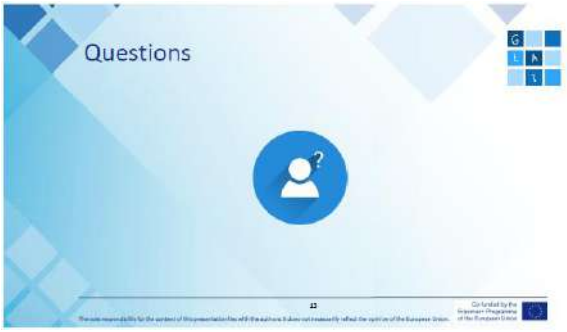












Workshop 1 – Game Based Learning (GBL) and Unplugged Activities

Session 1: Game based learning

Expected Learning Outcomes

- Recognize psychological and cognitive aspects of Game Based Learning
- Identify the importance of using educational computer games in courses
- Find, evaluate and select suitable serious games and integrate them into the learning process

Teaching Methods/Approaches

- Teacher presentation and demonstration
- Discussion
- Individual activity
- Group activity - collaboration
- Peer evaluation

Sources of Training Materials

- Portal izobraževalnih iger, http://hrast.pef.uni-lj.si/igre (Accessed: 14.6.2019.)
- SEGAN portal, http://seriousgamesnet.eu (Accessed: 14.6.2019.)

Duration: 3 hours (135 minutes)

Topic/Sub-topics	Learning Objectives	Evaluation
1. GAMES AND LEARNING	*Participants will understand the psychological and cognitive aspects of game based learning.*	Learners explore and analyse examples of games in order to point out typical characteristics of games (individual activity). Learners analyse examples of games considering corresponding learning theories in order to check the possibility of integration of the game into the learning process (group activity).
1.1. Introduction to games	Identify the concepts of games	
1.2. Games in human development	Understand the role of games in cognitive development	
1.3. Games and learning theories	Explore the features on serious games from learning theories point of view	
2. INTEGRATION OF GAMES INTO LEARNING PROCESS	*Participants will be able to find, evaluate and select suitable serious games and integrate them into learning process.*	Learners choose a didactic game, suitable for achieving predefined learning goals, and create a „learning package" which will be reviewed by the teacher and the colleagues (group activity).
2.1. Identification and evaluation of suitable serious games	Explore games available on recommended portals or on the web	
2.2. Integration of games into the learning process	Create a game based „learning package"	

Presentation: Games in education

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Workshop 1: Game Based Learning (GBL) and unplugged activities

Session 1: Game Based Learning (GBL)

Games in education

Authors and licence

Authors:

- **Jože Rugelj**, University of Ljubljana, Faculty of Education
  Joze.Rugelj@pef.uni-lj.si
- **Mateja Bevčič**, University of Ljubljana, Faculty of Education
  Mateja.Bevcic@pef.uni-lj.si
- **Anja Luštek**, University of Ljubljana, Faculty of Education
  Anja.Lustek@pef.uni-lj.si

Agenda

- Games in child development
- Educational games
- Educational games in the learning process

Introduction

- Characteristics of efficient approaches to learning:
  - student-centered
  - active
  - problem-based
  - directed to higher-order educational goals
  - motivational
  - supported by ICT

Educational games can integrate most of these characteristics!

Influence of games on child development

- Importance of child's play on the development of emotional, social, physical and cognitive growth
- Play is one of the most important activities for the development of important skills for life, regardless of age or level of development:
  - quick adoption to new circumstances
  - handling change with ease
- During play, the child discovers basic concepts from the real world and the first fundamental relationships between them are created

Games in human development

Vygotsky (1896 - 1934): "Play contains in a concentrated form all developmental tendencies - the most significant psychological achievements of the early childhood occur while children engage in play."

Jean Piaget (1896 – 1980):

- "Play is the incorporation of new intellectual material into the already existing cognitive structures, without a corresponding alteration of the structures themselves."
- "Play is the consolidation of newly learned behaviour. Repetition of learned concepts makes them an established part of the mental repertoire."

Jerome Bruner (1915 – 2016): "Play provides a **comfortable** and **relaxed atmosphere** in which children can learn to **solve** a variety of **problems**, making them able to efficiently **cope** with **complex problems** of the real world."

Specific characteristics of Digital Generation learners (Prensky, 2001)

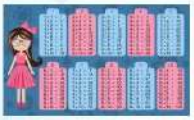
- Technology usage fluency
- Multi-tasking
- Individualization and personalization
- Increased connectedness
- Immediacy
- Multiple media types
- Engagement and working attitude
- Sociality and team spirit

Additional characteristics of "*gamers*"

- Ability to follow instructions
- Possession of problem-solving strategies
- Quick thinking
- Random access to resources
- Increased hand-eye coordination and fine motor skills
- Spatial reasoning
- Stimulating learning experience

Games and learning theories 1/2

- **Behaviourism**
  - one correct answer, immediate response
  - positive response (happy sound, positive character reaction that stimulate positive emotions)
  - games that have a drill and practice concept build in (e.g. multiplication table)
- **Constructivsm**
  - learning is an **active** process of constructing knowledge, built **recursively** on knowledge that the **user already has**
  - game **environment** where the students can construct their own knowledge



Games and learning theories 2/2

- **Constructivism**
  - own representation of knowledge
  - learning occurs when the learners' exploration uncovers an inconsistency between their current knowledge representation and their experience
  - learning takes place within a social context and interaction between learners and their peers is a necessary part of the learning process
- **Teacher's task:** guiding and providing feedback
- **Examples:** role-playing, adventures, detective games, strategies,...

Games for Learning Algorithmic Thinking

Educational games

Educational games

- What makes a computer game educational (serious)?
  - Well-defined learning goals (blurred in the game)
  - promotion of development of strategies and skills of learners
- Elements contributing to **educational values** of games:
  - sensual stimuli
  - fantasy
  - challenge
  - curiosity
- But even serious games **have to be funny!**

Eductional game elements

- Educational games must include:
  - compelling storyline
  - challenge (i.e., a problem to be solved)
  - rules of engagement
  - interaction within the environment and control
  - continuous feedback
  - particular goals or outcomes to achieve
- The task must not be perceived as **so difficult** that there is **no chance of success**

Educational games in the learning process

- Scaffolding needed to support students' perception that the challenge is achievable
  - different levels of mastery for students in the class (**individualization**)
- Student responses used to structure learning
- Timely corrective and progress-acknowledging feedback
  - correct mistakes
  - build understanding progressively
  - recognize incremental progress

Methodologies for using games for learning

- Suggested methodologies:
  - games as a **motivation before** the lecture
  - **teacher playing** a game **during** the lecture
  - game as a **group activity** in the classroom
  - game as a **home activity/independent** learning
- **Important:** Use educational games only when they increase the effectiveness of learning.

Implementation of games into the learning process

1. **phase - Identification and evaluation of suitable games**
2. **phase - Integration of games**
   - very often playing games is a time-consuming process - limited time for use of alternative learning resources in formal education

"Learning package"

- Consisting of briefing, post-game discussion, and reflection
- The teacher has to prepare the "learning package" taking into account:
  - students' background and previous knowledge
  - learning goals
  - curriculum
  - technical resources
  - her own competences
- Each implementation has to be evaluated by the teacher to determine to what extent learning goals have been achieved

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Exploring existing games

Individual activity

Exploring existing games

- Explore some of the following games and think whether they are suitable for inclusion in the learning process:
  - http://hrast.pef.uni-lj.si/games
  - https://www.tynker.com/
  - http://lightbot.com/flash.html
  - https://code.org/
  - https://codecombat.com/play
  - https://home.jeita.or.jp/is/highschool/algo/index_en.html
  - https://blockly-games.appspot.com/

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Preparing "learning package"

Group activity

Preparing „learning package"

Each group should choose one of the following games, prepare a "learning package" and present it to the other groups

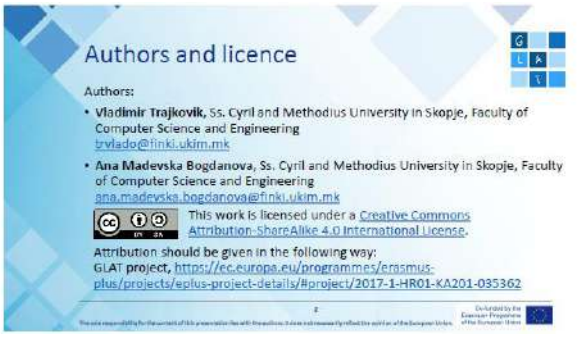
- Reality Show Restaurant (SL, EN)
  - http://hrast.pef.uni-lj.si/games/website/restavracija.html
- Luke saviour (SL, EN)
  - http://hrast.pef.uni-lj.si/games/website/LukecResiteli.html
- Light Bot (HR, SL, EN)
  - http://lightbot.com/flash.html

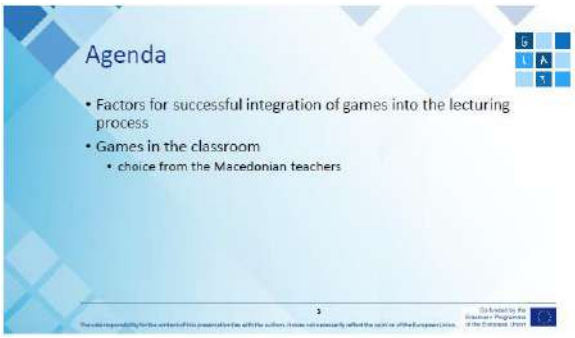
Questions to help prepare a "learning package"

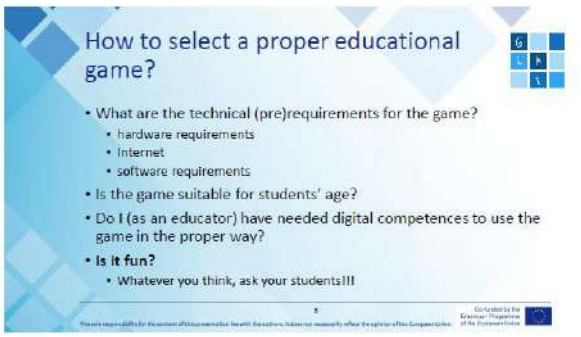
- To whom is the game intended?
- How will the game be included in the classroom (introduction to the new material, during the class, repetition, tasks, ...)?
- Are there any instructions? How will they be delivered to students?
- What role will the teacher have?
- Is additional material needed (worksheets, ...)?
- What activities will be conducted before and after playing the game?
- How will the game analysis (reflection) be performed?

Questions

Presentation: Integration of games into the lecturing process







How do you know whether the game is fun for the students?

- Is there any way to progress in the game (levels)?
- Is it competitive (can you compare results)?
- Is there any award?
- Is it easy to understand?
- Is it cooperative?
- Can you play it outside the classroom?
- ...

When does the game have educational value? (1/2)

- Can you relate game levels with educational goals?
- Are there any game elements that are opposite to educational goals?
  - e.g. violence
- Can the game be used for more than one educational goal in more than one subject (or other lecture in the same subject)?
- Does the game enable "deep learning"?
  - Do the students need to solve non-routine problems while playing the game?

When does the game have educational value? (2/2)

- Is Problem-Solving incorporated in the game design?
  - fact gathering, their evaluation, usage and creating some actions (or feedback)
  - reflex based
  - possible to repeat the actions and improve results
- Is Critical Thinking incorporated in the game design?
  - players need to solve some puzzles
  - decision making
  - need to understand different perspectives (e.g. other players goals and strategies)
- ...

Is the game useful?

The quality of learning depends on:

- Quality of experience while playing the game
  - Technology-related (objective, quantitative)
  - Fun-related (subjective, qualitative)
- Quality of achieved knowledge
  - short-term, fact-based (e.g. is it suitable for multiple-choice questions based assessment?)
  - long-term, expandable (e.g. can be used within the projects in other lectures and/or subjects?)

Games for Learning Algorithmic Thinking

Analysing examples

Group activity

Games for Learning Algorithmic Thinking

1. General

Example 1.1. Science for kids

- Fun science games for kids while learning more about science and technology
- Free online activities to try with something for everyone
  - chemistry
  - biology
  - physics
  - ...

http://www.sciencekids.co.nz/gamesactivities.html

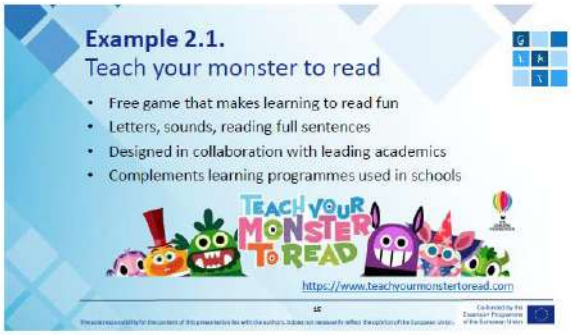
Science Kids
Fun science & technology for kids!

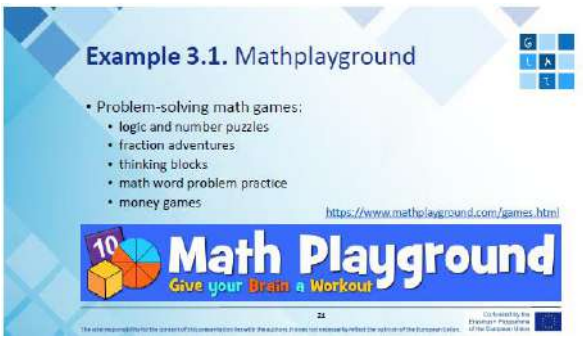
Example 1.2. IXL

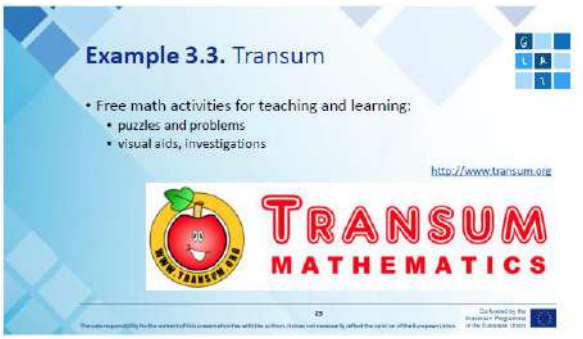
- Immersive, adaptive learning
- Math, language, arts, science
- K12 education
- 10 Free practice problems

IXL LEARNING

https://www.ixl.com





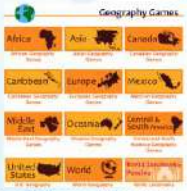


Games for Learning Algorithmic Thinking

4. Geography

Example 4. Travel the world

- Tutorials and games:
  - world's continents
  - countries
  - capitals
  - landscapes
- Games for beginners, intermediate and advanced learners



http://www.sheppardsoftware.com/Geography.htm

Games for Learning Algorithmic Thinking

5. Creative thinking

Example 5.1. Minecraft

- Building in a 3D procedurally-generated world with a variety of different cubes
- Requiring creativity from players
- Other activities:
  - exploration
  - resource gathering
  - crafting
  - combat
- Cost: 23$

https://minecraft.net

Example 5.2. Roblox

- Online gaming platform for kids and teens
- Students can create adventures, play games, role play, and learn with their friends in a family-friendly, immersive, 3D environment

https://www.roblox.com

Games for Learning Algorithmic Thinking

Exploring games

Group activity

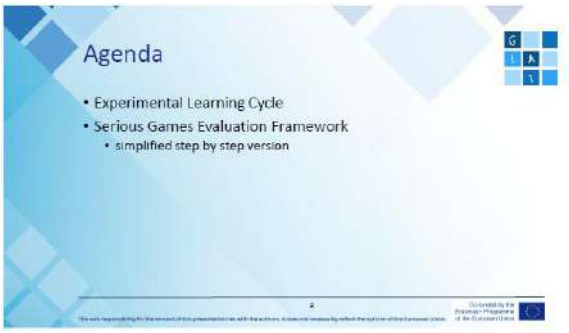
Exploring games

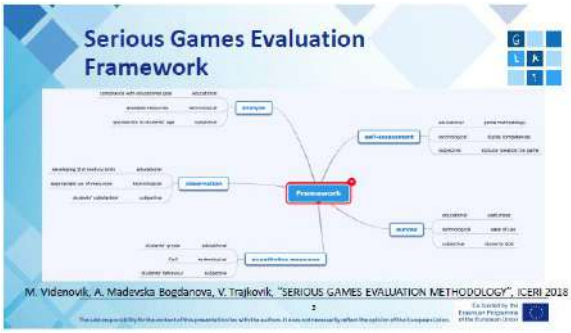
Each group should choose one of the presented websites or games, explore it in more detail, and discuss its possible integration in the classroom.

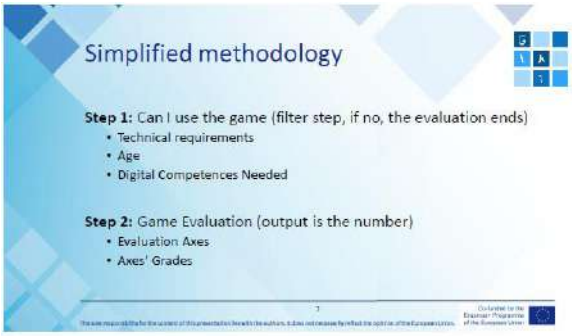
Share your ideas with the teacher and the other groups.

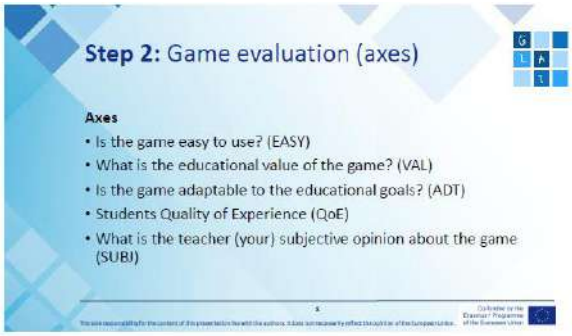
Questions

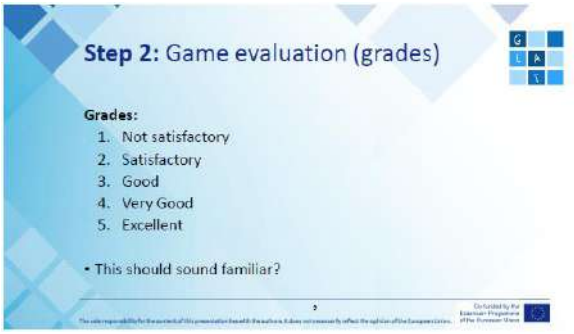
Presentation: Serious games evaluation framework

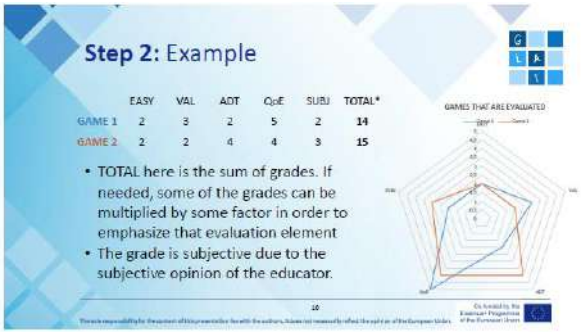


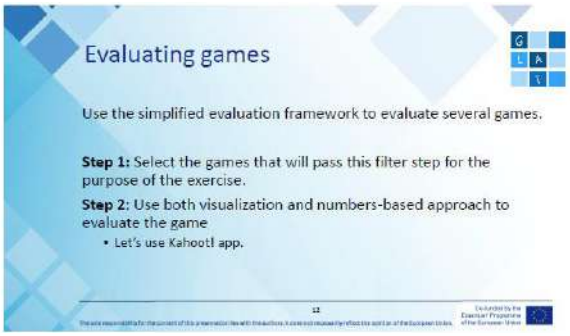


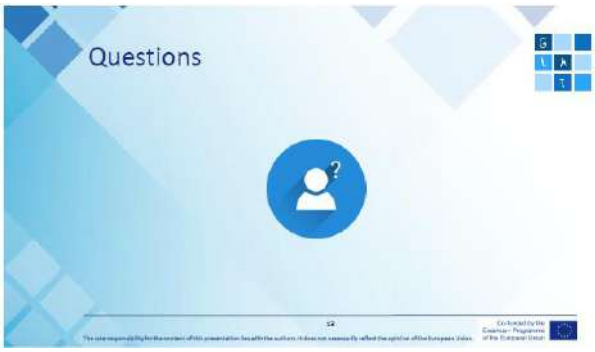












Workshop 1 – Game Based Learning (GBL) and Unplugged Activities

Session 2: GBL with unplugged activities

Expected Learning Outcomes

- Find examples of unplugged activities for the development of algorithmic thinking in different school subjects
- Analyze and compare existing examples
- Modify existing examples of unplugged activities for different school subjects

Teaching Methods/Approaches

- Teacher presentation and demonstration
- Discussion
- Group activity - collaboration

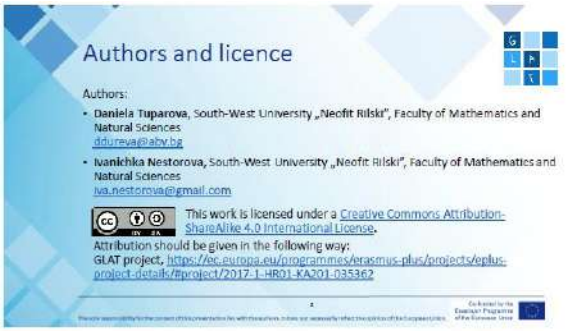
Sources of Training Materials

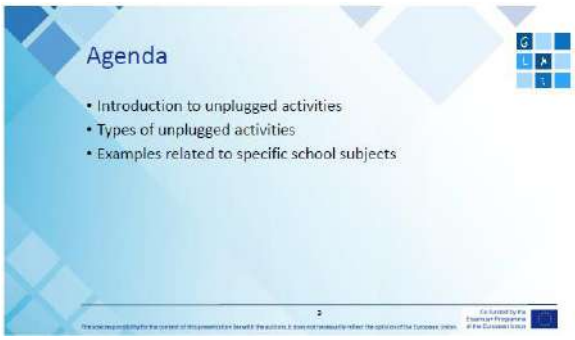
- CS Unplugged: http://csunplugged.org/ (14.12.2017.)
- Code Studio: https://studio.code.org/courses, https://code.org/curriculum/unplugged (14.12.2017.)

Duration: 1 hour (45 minutes)

Topic/Sub-topics	Learning Objectives	Evaluation
1. UNPLUGGED ACTIVITIES	*Participants will be able to describe and explain the characteristics of unplugged activities for the development of algorithmic thinking, analyze and classify existing examples of unplugged activities.*	Learners explore, analyze and classify existing examples of unplugged activities in order to transfer given examples to another school subject (group activity).
1.1. Introduction to unplugged activities for the development of algorithmic thinking	Describe and explain the characteristics of unplugged activities for algorithmic thinking development	
1.2. Examples of unplugged activities in different school subjects	Analyze and classify existing examples	
2. DESCRIPTION OF UNPLUGGED ACTIVITIES EXAMPLES	*Participants will be able to describe their own examples of unplugged activities appropriate for different school subjects.*	Learners describe new examples for unplugged activities (group activity - discussion).
2.1. New examples of unplugged activities	Propose examples of unplugged activities for algorithmic thinking from tales, everyday life, etc.	

Presentation: GBL with unplugged activities





Example 1: Hidden fruits (English, Math)

- Find the hidden fruits in the grid and put the picture of relevant fruit near the alphabet signs of it name.
- How many pictures of the fruits do you need for each word?

A	P	P	L	E	H	E	R	I	L
P	O	G	R	A	P	E	S	R	E
I	C	H	E	R	R	Y	L	O	M
S	T	R	A	W	B	E	R	R	Y
A	R	T	P	E	A	R	S	O	N
W	A	T	E	R	M	E	L	O	N
I	B	A	N	A	N	A	R	Y	N

Example 2: Numbers (Math, Computer science)

Task for students:

1. Find the two digit numbers with equal digit.
2. Order them in ascending order.
3. Replace the numbers with relevant alphabet sign.
4. Which word do you obtain?
5. Explain how to use it?

34-M	11-P	43-N	25-F	12-B	31-G
43-N	22-A	33-S	33-S	13-C	12-B
21-L	12-B	21-L	44-W	13-C	13-C
23-E	13-C	23-E	55-O	66-R	77-D
13-C	25-F	43-N	23-E	13-C	25-F

Example 3: Word search (English, Computer science)

1. Follow given instructions to find the hidden word. Start from upper left corner.
2. What is the meaning of the word?

P	A	A	O	P	Y
C	S	S	C	A	D
O	M	W	O	T	E
U	P	U	R	D	I
M	N	T	E	R	M

1. Go right
2. Go down
3. Go right
4. Go down
5. Go right
6. Go down
7. Go right

Idea for activity

- Divide the class in two groups and organize contest
  - first group has to hide words in the grids and describe with arrows how to find the words
  - second group has to follow the algorithm for moving and to find the words and explain it
- Time for word finding could be set
- Badges for founded word could be awarded

2. Real life algorithms

- Recognizing algorithms in our daily lives:
  - making sandwich
  - preparing tea
  - cleaning teeth
  - preparing school backpack
  - natural phenomenon

Games for Learning Algorithmic Thinking

Analysing examples

Group activity

Example 1: Code.org – Plant a seed

- Students create an algorithm to help each other plant a seed
- They cut out the steps for planting a seed from the provided worksheet and work together to choose the six correct steps from nine options

https://studio.code.org/s/course1/stage/6/puzzle/1

Example 2: Dance moves

- Students recognize dance structures
- They label the entire dance performance with the agreed marks
- They connect dance structures with the corresponding part of the music background

https://www.youtube.com/watch?v=PPNbH1shbII

Example 3: Code.org - Getting Loopy

- Students are introduced to the programming concept of loops (repeated instructions) through a dance activity (simple choreography).

https://studio.code.org/s/course1/stage/12/puzzle/1

Idea for activity

- Make photos with different statement of the hands or legs
- Introduce the concept *loop*
- Tasks for students:
  - to arrange dance or gymnastic exercise
  - to perform dance or gymnastic exercise according to given algorithm

3. Algorithms and analogies for concepts related to specific school subjects

- Ordering of rules
  - preforming mathematical operations
  - grammar
- Writing a song as algorithm (set od instructions)

Games for Learning Algorithmic Thinking

Analysing examples

Group activity

Example 1: Adding numbers (Math)

- Ordering of steps (rule) to add two three-digit numbers
- Steps are written on the paper and cut
  - students have to order steps
  - or the steps are given in wrong order and students have to find error in the algorithm

253
+471

Example 2: Present Simple vs. Present Continuous (English)

- Rule for determining the appropriate tense

General present	Permanent activity	Unlimited duration
Immediate present	Temporary activity	Limited duration

1. I like tea.
2. Keep quiet! The baby is sleeping.

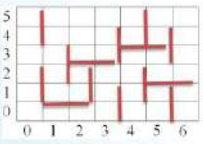
Example 3: Loops in the song (Music)

- Students should find part in song that have to be repeated

https://www.youtube.com/watch?v=9uqvGiZJQqg

4. Moving through a maze

- Moving characters through a maze using arrows
- Plan a route from the start location
- Use arrows to describe path



The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Union.

Co-funded by the Erasmus+ Programme of the European Union

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Analysing examples

Group activity

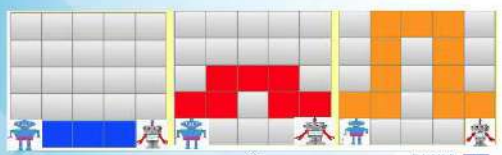
Example 1: Code.org - Happy maps

- Students create simple algorithms to move a character through a maze

https://studio.code.org/s/course1/stage/1/puzzle/1

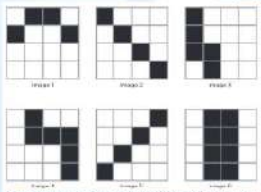
Example 2: Describing path

- Students use arrows to describe path between robots.



Example 3: Code.org - Graph Paper Programming

- Students write an algorithm using a set of predefined commands to direct their classmates to reproduce a drawing (to color squares in on graph paper)



https://studio.code.org/s/course2/stage/1/puzzle/1

Example 4: Code.org - Move it

- Students learn how to think ahead in multiple steps, as they plan a short route from the start location to the hidden smiley face, up to three steps away
- Starting point is the piece of paper imprinted with the compass rose

https://studio.code.org/s/course1/stage/2/puzzle/1

5. Tales and algorithms

- Recognizing algorithms in familiar stories and fairy tales:
  - The Wild Swans tale by Hans Christian Andersen → Elisa knits shirts for her eleven brothers
  - Hansel and Gretel → they execute an algorithm to get home from the forest
  - Cinderella → looking for a girl who will fit the shoe

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

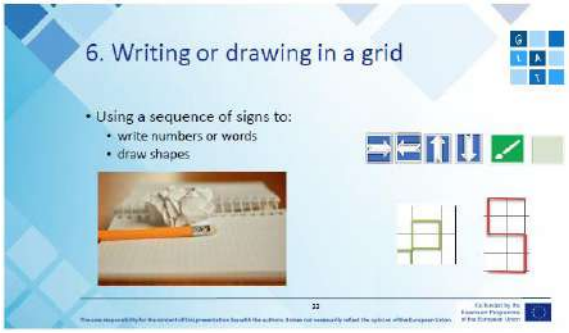
Analysing examples

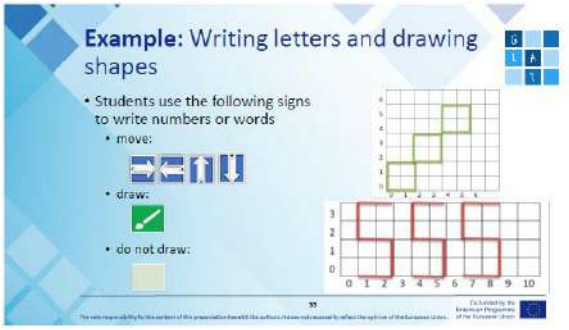
Group activity

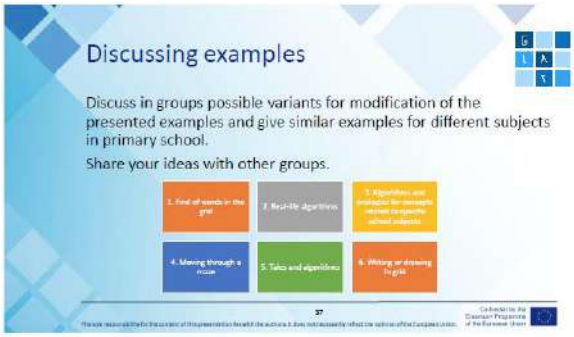
Example: Cinderella

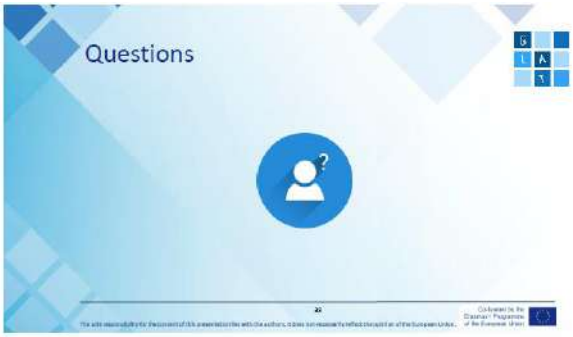
Find Cinderella

1. Find a girl
2. Try the shoe
3. If the shoe fits, then Cinderella is found, in other case go to step 1









Workshop 1 – Game Based Learning (GBL) and Unplugged Activities

Session 3: Using Web 2.0 tools for creating content for unplugged activities

Expected Learning Outcomes

- Identify the advantages of Web 2.0 tools for unplugged activities
- Create content for unplugged activity using Web 2.0 tools
- Create new examples for unplugged activities

Teaching Methods/Approaches

- Teacher presentation and demonstration
- Discussion
- Individual activity
- Group activity - collaboration

Sources of Training Materials

- Ljubić Klemše, N. "Web 2.0 alati i e-učenje u primarnom obrazovanju", Pogled kroz prozor, 2010.: https://pogledkrozprozor.wordpress.com/2010/11/27/web-2-0-alati-i-e-ucenje-u-primarnom-obrazovanju/ (2.12.2017.)
- Ljubić Klemše, N. "Web 2.0 alati i e-učenje u primarnom obrazovanju - II. dio", Pogled kroz prozor, 2010.: https://pogledkrozprozor.wordpress.com/2010/12/20/web-2-0-alati-i-e-ucenje-u-primarnom-obrazovanju-ii-dio/ (2.12.2017.)

Portals with tools and resources for teachers:

- Tools for Educators: http://www.toolsforeducators.com/ (11.12.2017.)
- The Teacher's Corner: https://worksheets.theteacherscorner.net/ (11.12.2017.)
- Education World: http://www.educationworld.com/tools_templates/index.shtml (11.12.2017.)

Web 2.0 tools:

- Canva: https://www.canva.com (1.12.2017.)
- Sketchpad: https://sketch.io/sketchpad/ (1.12.2017.)

Duration: 3 hours (135 minutes)

Topic/Sub-topics	Learning Objectives	Evaluation
1. WEB 2.0 TOOLS FOR CREATING CONTENT FOR UNPLUGGED ACTIVITIES	*Participants will be able to identify the advantages of using Web 2.0 tools for unplugged activities.*	Learners explore examples and resources in order to discuss the potentials of Web 2.0 tools for unplugged activities (group activity).
1.1. Introduction to the Web 2.0	Identify the advantages of using Web 2.0 tools	
1.2. Investigate examples of Web 2.0 tools	Use the preselected Web 2.0 tools to create drawings, posters, leaflets, etc.	
2. CREATING CONTENT FOR UNPLUGGED ACTIVITIES	*Participants will be able to create content for chosen unplugged activity.*	Learners create content for unplugged activities (individual activity) which will be evaluated by the teacher.
2.1. Presenting worksheet template	Create a worksheet (using the prepared template) and other content for the preselected task of unplugged activity	
2.2. Creating content		
3. DEVELOPMENT OF EXAMPLES OF UNPLUGGED ACTIVITIES	*Participants will be able to create examples of unplugged activities appropriate for different school subjects.*	Learners discuss potentials of Web 2.0 tools and other resources and create new examples for unplugged activities (group activity).
3.1. Modification and adaptation of examples for another school subject	Create new examples of unplugged activities based on given examples Give new examples of unplugged activities for algorithmic thinking from tales, everyday life, etc.	
3.2. Development of examples of unplugged activities		

Presentation: Using Web 2.0 tools for creating content for unplugged activities

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Workshop 1: Game Based Learning (GBL) and unplugged activities

Session 3: Using Web 2.0 tools for creating content for unplugged activities

Authors and licence

Authors:

- **Jasminka Mezak**, University of Rijeka, Faculty of Teacher Education jasminka.mezak@ufri.uniri.hr
- **Daniela Tuparova**, South-West University „Neofit Rilski", Faculty of Mathematics and Natural Sciences ddureva@abv.bg

Attribution should be given in the following way:
GLAT project, https://ec.europa.eu/programmes/erasmus-plus/projects/eplus-project-details/#project/2017-1-HR01-KA201-035362

Agenda

- Introduction - Web 2.0
- Web 2.0 tools for preparing *unplugged* activities
  - Canva
  - Sketchpad

Games for Learning Algorithmic Thinking

Web 2.0

Characteristics of Web 2.0

- Using Web as platform
- Accessing applications through Web browser
  - regardless of location (device)
  - reduced cost for software
- User participation in content creation
- Social networking
- „2.0" functionalities
  - commenting, sharing, tagging
- Rich user experience

Web 2.0

- **Fraze** →T. O'Reilly i D. Dougherty (2004)
- Describes *changes* in the way the Web is used as an Internet service
- It does not refer to technical improvements
- Impact on learning and teaching

E-learning 2.0

- Independence of time and place of learning (where and when we learn?)
- **Student in the centre** of learning process
  - constructivsm
- **Collaborative** learning
  - interaction
  - joining the community of practice → collective knowledge
- Using a variety of tools
  - Web 2.0 tools

Web 2.0 tools

- Web 2.0 tools → applications available on the Web that have the characteristics of Web 2.0
- Main types (according to the main purpose):
  - replacement for classic (desktop) applications
  - creation and exchange of multimedia content
  - exchange of ideas and creative learning
  - networking and communication
  - social bookmarking

Advantages and disadvantages of Web 2.0 tools

- **Advantages**
  - availability, price
  - diversity
  - stimulate activity and creativity
  - ...
- **Disadvantages**
  - questionable reliability
  - limited features of free versions
  - ads, advertisements
  - ...

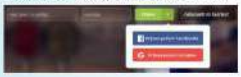
Web 2.0 tools for preparing *unplugged* activities

Canva

- Link: https://www.canva.com
- Login: Facebook, Google account
- Use: free (basic version)
- Purpose: making presentations, graphics, posters, posters
  - predefined templates
  - insert of ready-made graphical elements (images, icons, wallpapers, ...)
  - insert text
  - insert own files
  - save and export files

Login

- Login: user account, Facebook account, Google account



Canva – user interface

Sketchpad

- Link: https://sketch.io/sketchpad/
- Login: it is not necessary
- Use: free (basic version)
- Purpose: creation of vector graphics
  - draw and color
  - write text
  - insert ready-made graphical elements (images, icons,...)
  - insert own files
  - save (!) and export files

Sketchpad – User interface

Preparing materials for *unplugged* activities

- Creating drawings (Canva, Sketchpad)
- Creating worksheets for students (MS Word, Google Documents)

Worksheet template

Games for Learning Algorithmic Thinking

Creating worksheets

Individual activity

Task 1 – Walking through the maze

- **Task** - Show with the arrows the steps that will take little bear to his mother.

Task 1: Worksheet

Task 1: Expected solution

Task 1: Creating the necessary materials

- Draw a maze (Canva)
  - select a template to create a collage (6x6)
  - color fields and background, insert pictures of trees and flowers
  - insert the pictures of little bear and mother bear
  - save the drawing
- Use a template to create a worksheet (MS Word)
  - type task text
  - insert a picture of the maze
  - insert the arrows for cutting on the second page of the document

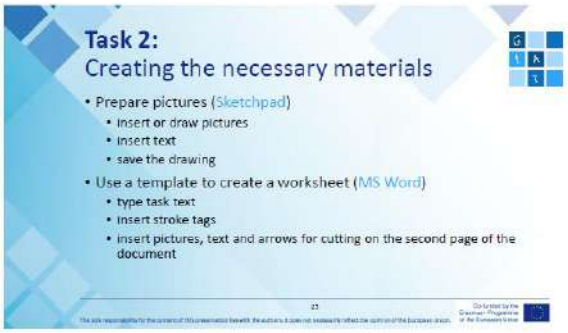
Task 2: Algorithm for the song

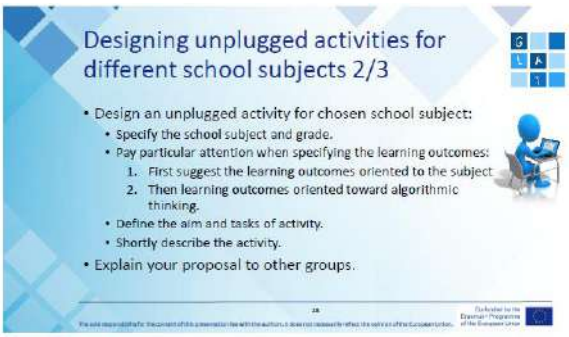
- **Task** – Show algorithm that includes repetitions of a song „Wash your hands" (https://www.youtube.com/watch?v=M4D3p6tg42I) with pictures.

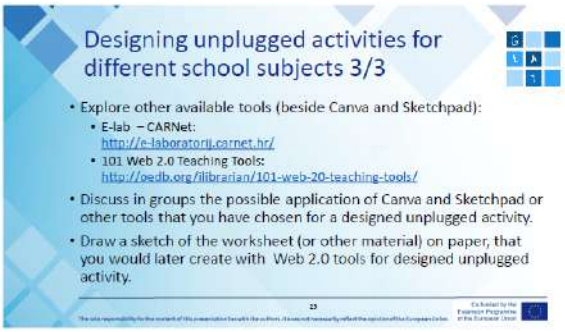
WASH, WASH, WASH YOUR HANDS
PLAY YOUR HANDY GAME
RUB AND SCRUB, SCRUB AND RUB
GERMS GO DOWN THE DRAIN.
WASH, WASH, WASH YOUR HANDS
PLAY YOUR HANDY GAME
RUB AND SCRUB, SCRUB AND RUB
DIRT GOES DOWN THE DRAIN.

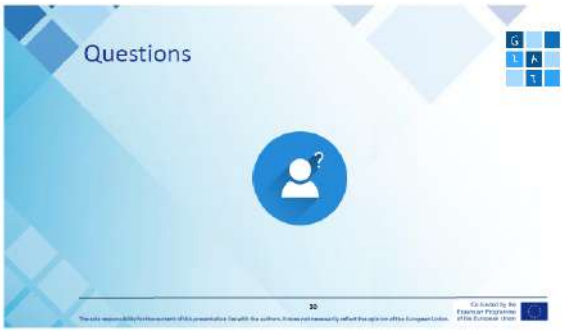
Task 2: Worksheet

Task 2: Expected solution









Workshop 1 – Game Based Learning (GBL) and Unplugged Activities

Session 4: Designing learning scenarios

Expected Learning Outcomes

- Identify the concepts of learning scenarios
- Analyze and compare existing examples of learning scenarios in written forms
- Using the learning scenario to create an unplugged activity

Teaching Methods/Approaches

- Teacher presentation and demonstration
- Discussion
- Group activity - Collaboration
- Peer evaluation

Sources of Training Materials

- E-škole scenariji poučavanja. CARNet: https://scenariji-poucavanja.e-skole.hr/ (5.12.2017.)
- Collaborative Education Lab - Learning scenarios: http://colab.eun.org/learning-scenarios/ (10.12.2017.)
- Code Studio – katalog: https://studio.code.org/courses (7.12.2017.)

Duration: 2 hours (90 minutes)

Topic/Sub-topics	Learning Objectives	Evaluation
1. LEARNING SCENARIOS	*Participants will be able to describe and explain the concepts of the learning scenario, analyze and compare existing examples of learning scenarios.*	Learners explore and analyze existing examples of learning scenarios in order to point out good and bad features (group activity).
1.1. Introduction to leaning scenarios	Identify the concepts of learning scenarios	
1.2. Investigate examples of existing learning scenarios	Analyze and compare existing examples	
2. DESIGNING LEARNING SCENARIOS FOR UNPLUGGED ACTIVITIES	*Participants will be able to create a learning scenario for chosen unplugged activity.*	Learners choose one unplugged activity among offered to create a learning scenario that will be evaluated by the teacher and the colleagues (group activity).
2.1. A learning scenario template	Create a learning scenario using a prepared template for preselected unplugged activity	
2.2. Filling out a learning scenario template		

Presentation: Designing learning scenarios

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Workshop 1: Game Based Learning (GBL) and unplugged activities

Session 4: Designing learning scenarios

Authors and licence

Author:

- **Jasminka Mezak**, University of Rijeka, Faculty of Teacher Education jasminka.mezak@ufri.uniri.hr

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Attribution should be given in the following way:
GLAT project,https://ec.europa.eu/programmes/erasmus-plus/projects/eplus-project-details/#project/2017-1-HR01-KA201-035362

Agenda

- Introduction
- Learning scenarios with examples
- Examples of learning scenarios for selected activities
- Practical work - Creating learning scenarios

Educational process

- Preparation
- Implementation
- Evaluation

- Contemporary approach, applying the appropriate strategies, methods and teaching techniques

Teacher competences

- Teaching skills for planning, preparation, performance and teaching lesson
- Include active learning:
  - understanding
  - expressing own attitude
  - critical thinking
  - creative problem solving
- Skills for monitoring and evaluating students

Planning and preparation

- Designing the learning environment:
  - one task
  - several tasks
  - learning unit
  - the entire teaching subject
- Defining:
  - resource and learning materials
  - equipment (technology)
  - activities

Learning scenarios

- Documents that contain innovative and creative ideas for teaching activities using contemporary teaching methods with the use of appropriate digital content and tools
- Textual or graphical form (LePlanner)
- Included in the teaching process as a whole teaching unit or as a part of the teaching unit

Elements of Learning scenario

The main elements that the scenario should contain:

- Description of activities
- The learning outcomes that will be realized with specified activities
- Methods and forms of teaching
- Tools for realizing the learning outcomes

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Analysing examples

Group activity

Example 1: CARNet project „e-škole"

https://scenarij-poucavanja.e-skole.hr/

Example 2: Code.org – Plant a seed

https://code.org/curriculum/course1/6/Teacher

Example 3: Colaborative Education Lab

http://colab.eun.org/learning-scenarios

Example 4: LePlanner

https://leplanner.net/#/

Games for Learning Algorithmic Thinking

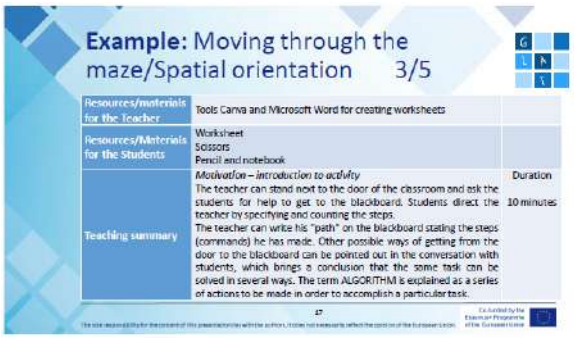
Example of a GLAT learning scenario

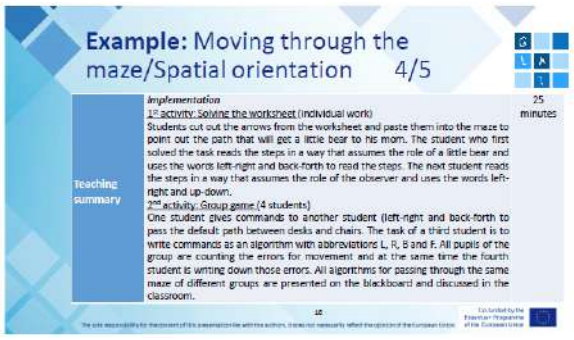
Example: Moving through the maze/Spatial orientation 1/5

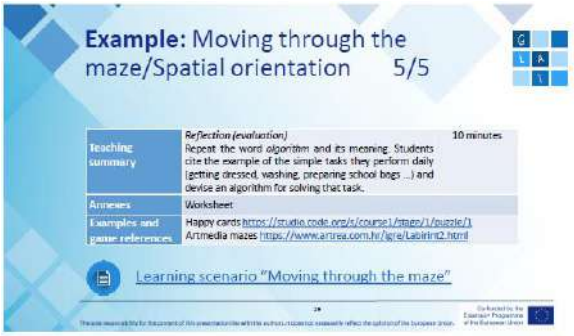
Learning Scenario Title	Moving through the maze/Spatial orientation
Course / Grade	Science, 1st grade
Learning Outcomes	*General learning outcomes* Determine the direction of movement: left-right, up-down and back and forth Moving in different directions in space (left-right, back and forth) Suggest the steps to navigate by the given path Suggest the steps to move from the starting to the ending position *Specific LO oriented on algorithmic thinking* Interpret the term algorithm Determine a set to reach a predetermined goal
Aim, Tasks and Short Description of Activities	As part of the Science lesson, for repetition and practicing the concepts left and right, up and down and back and forth, the students will solve the worksheet with maze by placing the arrows to determine the default path. Students will lead each other from the initial position to the final position in the classroom. Students will get familier with the term algorithm as a sequence of commands needed to accomplish the goal - arriving at the default location.

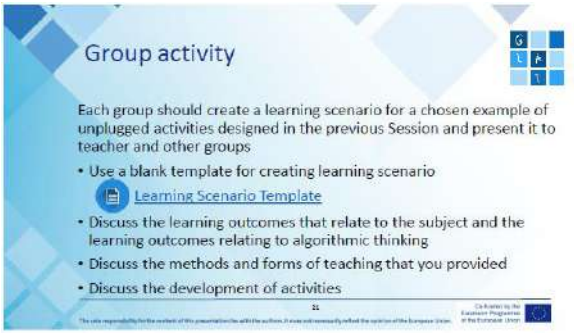
Example: Moving through the maze/Spatial orientation 2/5

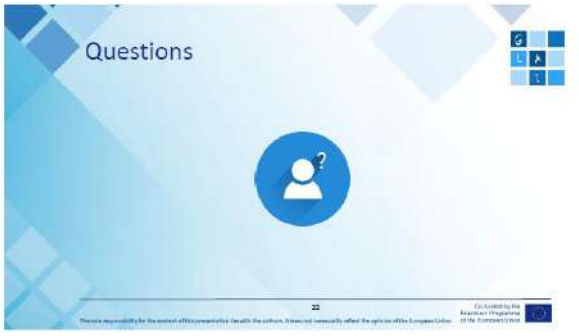
Keywords	Left-right, up-down, back-forth, algorithm, command
Correlation and Interdisciplinarity	Science, Physical Education and Informatics
Duration of Activities	45 minutes
Teaching Methods	Game-Based Learning Dialog method Problem-solving method
Teaching Forms	Frontal teaching Individual work Group work by four students
Tools	none











Workshop 1 – Game Based Learning (GBL) and Unplugged Activities

Session 5: Designing learning scenarios using a graphical tool

Expected Learning Outcomes

- Use LePlanner as a tool for designing learning scenarios
- Plan, create and instruct unplugged algorithmic thinking activities for students using LePlanner

Teaching Methods/Approaches

- Teacher presentation and demonstration
- Discussion
- Individual activity
- Group activity - collaboration
- Peer evaluation

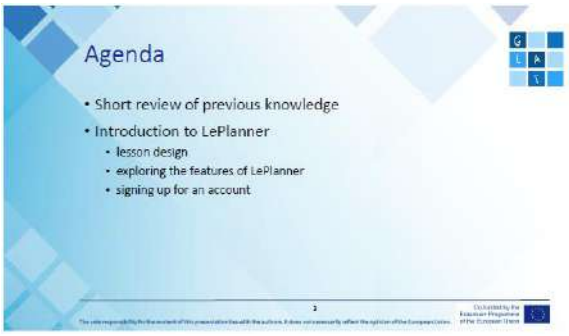
Sources of Training Materials

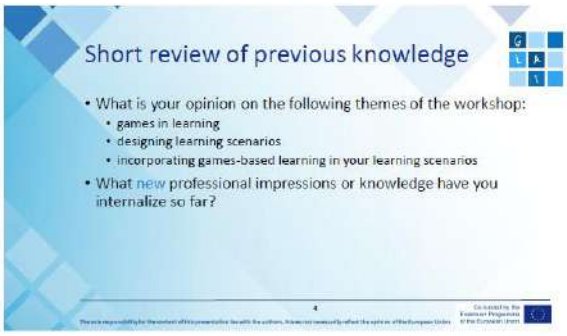
- LePlanner: https://leplanner.net/#/ (15.12.2017.)
- LePlanner – Creative Classroom Collection: https://beta.leplanner.net/#/tags/CreativeClassroomCollection (5.12.2017.)
- Codecombat – Dungeons of Kithgard: https://codecombat.com/play/level/dungeons-of-kithgard (15.12.2017.)

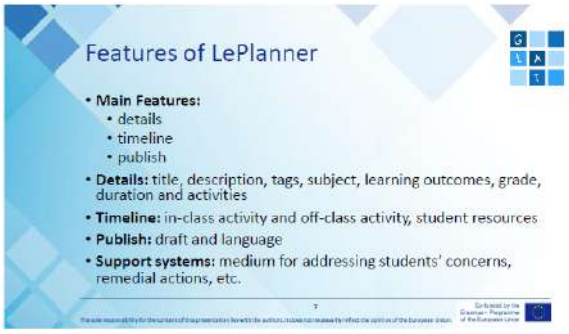
Duration: 2 hours (90 minutes)

Topic/Sub-topics	Learning Objectives	Evaluation
1. PLANNING AND CREATING LEARNING SCENARIOS	*Participants will be able to explore the features of the tool LePlanner for the creation of learning scenarios, create, evaluate, edit, and publish lesson scenarios.*	Learners create a dummy account for LePlanner log in accounts, create a dummy lesson, and prototyping (individual activity).
1.1. Introduction to LePlanner	Explore the key features of the LePlanner	
1.2. Creating a learning scenario	Create a sample leaning scenario(s)	Learners create a real log account in LePlanner, create a lesson, and publish it.
1.3. Reviewing created learning scenario (evaluating, editing and publishing scenarios)	Explore the features of the timeline for creating course contents	
2. DESIGNING LEARNING SCENARIOS USING LEPLANNER	*Participants will be able to create (unplugged, game based) learning scenarios using LePlanner and demonstrate the teaching of the planned lesson(s)*	Learners are engaged in micro-teaching using the designed lesson plan as a group activity.
2.1. Designing Game Based Learning Scenarios and participating in demonstration lessons	Explore online games, and create an unplugged game based learning scenario	

Presentation: Designing learning scenarios using a graphical tool







Lesson design - Student resources

- Expected learning outcomes/competence
- Student activities
- Support learning resources
- Technology
- Level of interaction/Collaborations medium
- Feedback channels

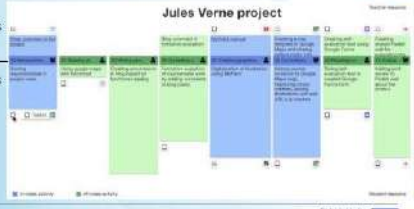
LePlanner: Defining activities

LePlanner: Defining resources

LePlanner: Timeline view (scenario)

resources and comments for teachers

resources and comments for students



LePlanner: Text view (scenario)

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Exploring LePlanner

Individual activity

Exploring LePlanner

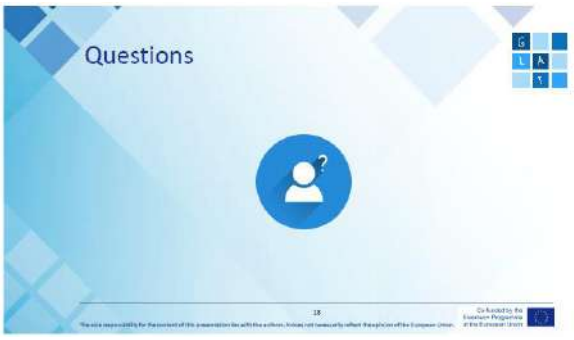
- Explore existing lesson scenarios developed using LePlanner.
- Create user account.
- Create a lesson scenario with several in-class and off-class activities.

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Designing and implementing a learning scenario

Group activity



Workshop 1 – Game Based Learning (GBL) and Unplugged Activities

Session 6: Designing learning scenarios for unplugged activities

Instructions for the participants

Expected Learning Outcomes

- Create learning scenarios in order to develop innovative ideas for carrying out game based unplugged activities
- Implement learning scenarios for different courses in the classroom with the students from 1st to 4th grades of primary school

Individual Assignment:

Your task is to prepare the learning scenario for carrying out an unplugged activity in written form and in graphical form using LePlanner. You could choose any school subject and any lesson within the subject for students from your class, considering that the activity should be completed in two months.

This is the **first version of the 1st learning scenario** which you will continue to design with the online help of your mentor. The completed version of the scenario will be **reviewed** by the mentor and the final refined versions you will **implement** in the classrooms with your students.

You are also supposed to write the **reflection** on conducted activities.

Duration: up to 2 months for the whole assignment

	ASSIGNMENT STEPS
1.	Choose a school subject – plan the activities that will be carried out in your class next month.
2.	Use the **Learning Scenario Template** form (Annex 1) for textual version and LePlanner for the graphical version of your scenario.
3.	Specify the **Learning outcomes**: • state ***general learning outcomes*** related to the course that will include game based unplugged activities • state the ***learning outcomes oriented on algorithmic thinking***
4.	Describe the **Aim and tasks** of teaching and give a **Short description of activities**. Plan the activities that will integrate games into the lecturing process, providing propaedeutic for algorithms and programming such as: • Finding words in the grid • Real-life algorithms • Algorithms and analogies for concepts related to specific school subjects • Moving through a maze • Tales and Algorithms • Writing or drawing in grid The activities should not include work on computer/tablet/smartphone, just unplugged activities.
5.	Specify the **Keywords**, **Correlation, and Interdisciplinarity** with other courses or topics, and **Duration of activities**.

6.	Point out **Learning and teaching strategies and methods**. Specify the **Teaching forms**: combine individual and group work.
7.	Choose Web 2.0 **Tools** that will be used for creating the content for unplugged activities. Point out all **Resources/materials** which will be required for the teacher as well as for students.
8.	Elaborate the **Teaching summary** as **Motivation (Introduction)**, **Implementation** and **Evaluation (Reflection)**. This part develops in detail the previously mentioned short description of activities.
9.	Create suitable content for unplugged activities, e.g. posters, worksheets, leaflets…. Pay attention to the copyright for images, videos, and other materials collected from the web. Photographing your students requires written parents' consent.
10.	In **Annexes** box provide examples and tasks you have created by yourself as well as a link to the graphical version of the learning scenario in LePlanner.
11.	**Examples and game references** box should contain sources you will use for the activities.
FOLLOW-UP ACTIVITIES	
1.	Upload your completed first version of the learning scenario to the Moodle e-course. Mentor will review and correct your scenario.
2.	Upload your final version of the learning scenario considering the mentor's suggestions and corrections.
3.	After mentor's approval, implement your learning scenario in the class with your students.
4.	Post a **reflection** on conducted activities in the forum: • Write a more extensive description of the implementation of the activity in your class. • Describe how your students have accepted learning activity. • Describe the achievement of all planned learning outcomes, both general and oriented on algorithmic thinking. • Define what you would like to change before the next implementation of the scenario.

Workshop 2: Problem Based Learning (PBL), Online Quizzes and Logical Tasks

Workshop 2 – Problem Based Learning (PBL), Online Quizzes and Logical Tasks

Workshop schedule

Day 1

Introduction to Workshop 2

Duration: 1 hour (45 minutes)

Introductory presentation: Workshop 1 - follow-up activities
Introduction to Workshop 2

Session 1: Introduction to Problem Based Learning (PBL)

Duration: 1 hour (45 minutes)

Lecture: Definition and key principles of Problem Based Learning
Demonstration: Learning scenarios illustrating PBL
Group work: Design a PBL lesson

Session 2: Problem-solving in logical games

Duration: 3 hours (135 minutes)

Lecture: Digital tools within the process of problem-solving
Demonstration: How to use problem-solving process in logical games
Group work: Exploring examples and resources

Presentation: Methodology – Role-playing games
Group work: Role-playing games

Session 3: Online quizzes and logical tasks

Duration: 3 hours (135 minutes)

Lecture: Logical tasks and quizzes in the classroom
Demonstration: Examples of logical tasks and quizzes for different school subjects, providing propaedeutic for algorithms and programming
Group work: Exploring examples and resources

Day 2

Session 4: Using Web 2.0 tools for creating quizzes and logical tasks

Duration: 4 hours (180 minutes)

Presentation: Advantages of using Web 2.0 tools for creating quizzes and logical tasks
Group work: Exploring examples and resources
Demonstration: Creating quizzes and logical tasks using Web 2.0 tools (Kahoot, Wizer, Match the memory)
Individual work: Creating quizzes and logical tasks using Web 2.0 tools

Group work: Creating examples of quizzes and logical tasks for different school subjects

Session 5: Designing learning scenarios for logical tasks

Duration: 3 hours (135 minutes)

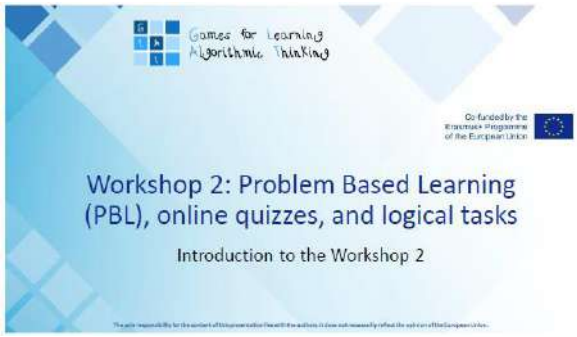
Individual work: Preparing learning scenarios based on PBL and logical tasks in written form and in graphical form using LePlanner (developing the first version of the 2nd learning scenario)
Group work: Review and discuss about the developed scenarios

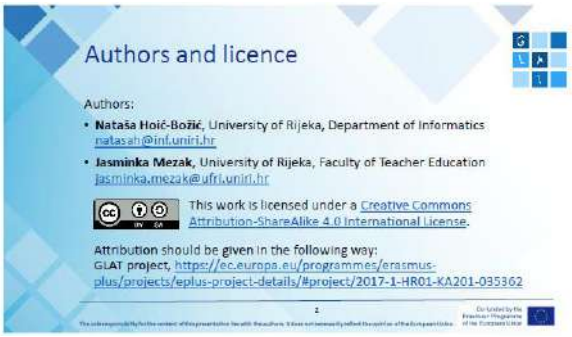
Conclusion of the Workshop 2

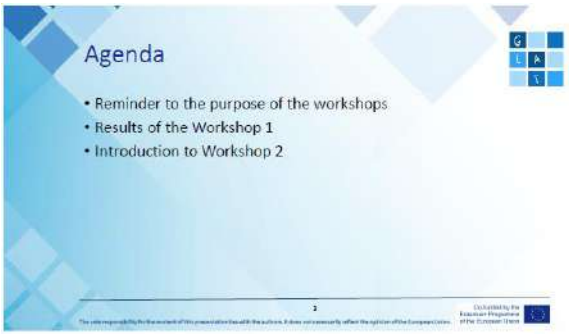
Duration: 1 hour (45 minutes)

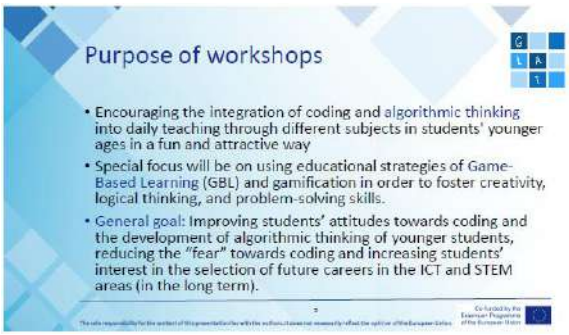
Whole-group activity: Debriefing
Closing talk: Introducing and explaining the follow-up activities (developing the 2nd learning scenario based on PBL and logical tasks)

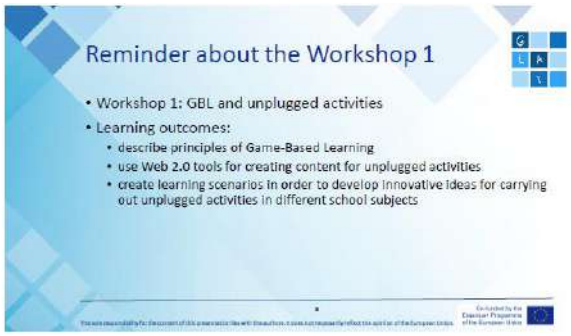
Presentation: Introduction to Workshop 2











Learning scenarios carried out in schools for different subjects

- The outstanding learning scenarios as examples of good practice

Rhythmic and dance structures	Playing and revising about the traffic	Addition and subtraction to 20

Available at: https://glat.uniri.hr/?page_id=2371

What is next?

Workshop 2:
Problem-Based Learning (PBL), online quizzes, and logical tasks

- Learning outcomes:
  - Describe the principles of Problem-Based Learning
  - Use Web 2.0 tools for creating logical tasks and online quizzes
  - Create learning scenarios in order to develop innovative ideas for carrying out logical tasks and online quizzes
  - Implement learning scenarios for different courses in the classroom with the students from 1st to 4th grades of primary school

In the following workshop pay attention to... 1/3

Computational and algorithmic thinking (CT & AT)

- Include elements in learning scenarios - along with the learning outcomes related to the courses, include those that relate to CT & AT

Learning scenarios

- After the Workshop 2, it is mandatory to implement at least one scenario
- Use games, quizzes or logical tasks prepared for computer/tablet/smartphone for at least one example (not only *unplugged* activities)
- Use PBL and group work

In the following workshop pay attention to... 2/3

About a review of the activity conducted in the classroom

- The teacher's review or reflection is very important.
- Write more extensive reflections, particularly on whether the learning outcomes related to algorithmic and computational thinking have been achieved.
- State the number of students who participated in the learning activity.
- Describe how your students have accepted learning activity.
- Describe the achievement of all learning outcomes in the scenario.
- Define what would you like to change before the next implementation of the scenario.

In the following workshop pay attention to... 3/3

Other

- Pay attention to the copyright for images, videos, and other materials collected from the web.
- Photographing and interviewing students requires written parents' consent

Questions

Let's start...

The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Union.

Co-funded by the Erasmus+ Programme of the European Union

Workshop 2 – Problem Based Learning (PBL), Online Quizzes and Logical Tasks

Session 1: Introduction to Problem Based Learning (PBL)

Expected Learning Outcomes

- Demonstrate competence in the principles, process and application of Problem Based Learning in learning situations/scenarios.
- Use the principles of PBL to design lessons for stimulating algorithmic thinking in problem-solving engagements.

Teaching Methods/Approaches

- Teacher presentation and demonstration
- Discussion
- Group activity - collaboration
- Peer-review

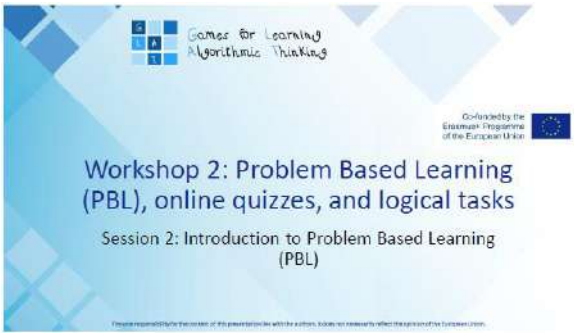
Sources of training materials

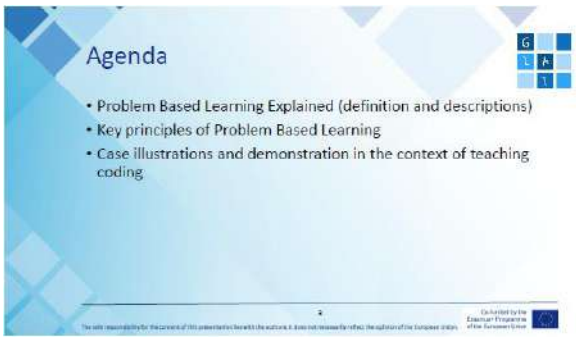
- Video "Learning scenario designing environment LePlanner.net": https://vimeo.com/168032150 (20.8.2018.)
- LePlanner: https://beta.leplanner.net/#/ (20.8.2018.)
- Rõbtšenkov, Romil: LePlanner a tool for creating learning scenarios: http://htk.tlu.ee/event/wp-content/uploads/2016/05/romil_robtsenkov.pdf (20.8.2018.)
- Clayton, Graham & Pierpoint, Peter: Problem Based Learning: A Would-be Practitioner's Guide, University of Plymouth Business School, 1996.: http://www2.uwe.ac.uk/faculties/BBS/BUS/Research/NTFS/problem.pdf (20.8.2018.)
- Christiansen, E. T., Kuure, L., Mørch, A., & Lindström, B. (Eds.): PROBLEM BASED LEARNING FOR THE 21st CENTURY: New Practices and Learning Environments (1 ed.), Aalborg Universitetsforlag, 2013.: http://vbn.aau.dk/files/187818413/PROBLEM_BASED_LEARNING_FOR_THE_21st_CENTURY_WEB.pdf (20.8.2018.)
- Jurković, Violeta. (Ed.): Guide to Problem Based Learning, Ljubljana: Slovene Association of LSP Teachers. 2005.: http://www.sdutsj.edus.si/SDUTSJ_Guide_%20to_%20PBL.pdf (20.8.2018.)

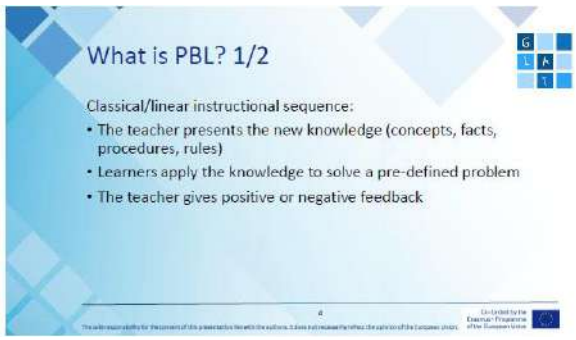
Duration: 1 hour (45 minutes)

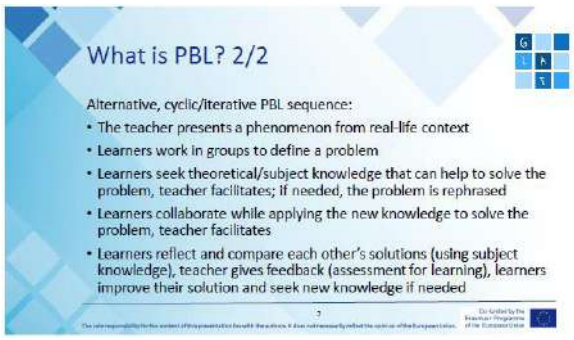
Topic/Sub-topics	Learning Objectives	Evaluation
1. INTRODUCTION TO PROBLEM BASED LEARNING	*Participants will able to demonstrate competence in using the principles and process of the PBL approach to solving problems.*	Learners explore and distinguish PBL from other learning approaches and argue for its relevance or otherwise – relative to the various learning scenarios.
1.1. Problem Based Learning Explained (definition and descriptions)	Explain and describe PBL – with the emphasis of relevant descriptors	
1.2. Key principles of Problem Based Learning	Identify and describe the key principles and characteristics underpinning PBL	
2. PRACTICAL – HANDS-ON PBL ACTIVITIES	*Participants will be engaged in the identification of PBL related cases, demonstrate the ability to address them and review proposed solutions.*	Learners identify real-life or learning cases where PBL is applicable, define the problem and use PBL principles to propose solutions to the problems as a group activity. Learners are engaged in peer-reviewing discussion of proposed or administered solutions.
2.1. Modelling the PBL approach in conceptual problem cases/learning scenarios	Identify real-life and learning cases and where PBL could be applied Propose process and procedures in the PBL approach	

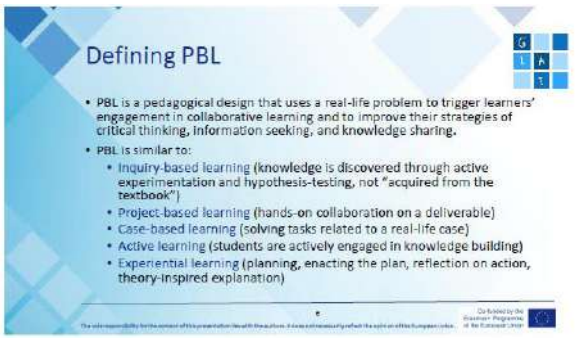
Presentation: Introduction to Problem Based Learning (PBL)

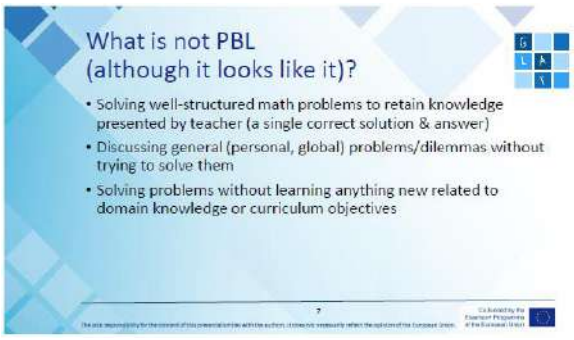


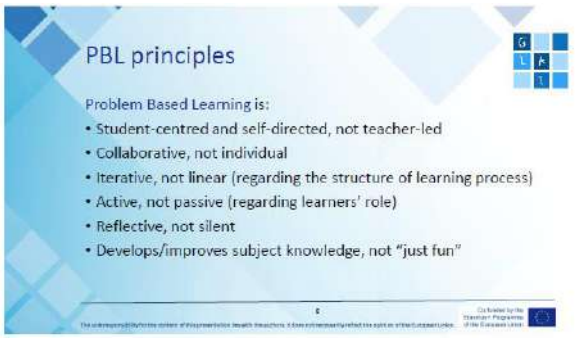












Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Discussing about PBL

Group activity

Well, this is embarrassing...

- The next steps would have been lecturing about problem types and quality, design process, scaffolding and assessment in the context of PBL, but ...
- Was the beginning of this workshop PBL (did I teach what I preach?)
- How should I have designed it, if I wanted this session to become truly a PBL experience for you as learners?

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Exploring case illustrations

Group activity

Example 1: 4th grade science lesson, Moldova

- Topic: elastic force and friction
- Task: in groups of 4, compile and program a robot that would illustrate your story that explains the phenomena

https://youtu.be/0qsqokc_WSY

Example 2: PBL teacher training in Estonia

- Trigger: the most well-known children song (Duck the Engineer)
- Task: define and solve a problem, using coding and/or robotics

https://youtu.be/0qsqokc_WSY

Example 3: 5th grade science lesson in Estonia - www.progetiiger.ee

- Topic: renewable energy
- Task: build a model of a wind generator using Strawbees & Quirkbot

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

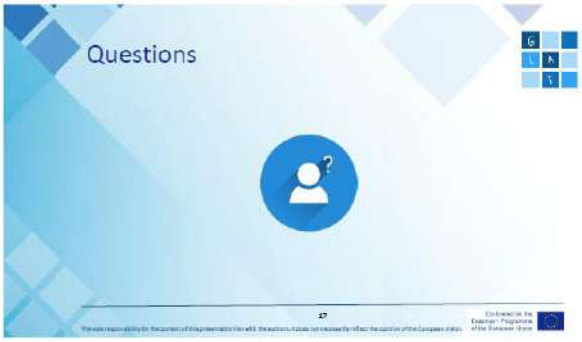
Identifing cases where PBL could be applied

Group activity

Identify real life and learning cases where PBL could be applied

Peer-to-peer activity:

- Select a well-known Croatian children song or cartoon
- Define a problem
- Propose initial solution
- Assessment criteria: explain task types, design process, scaffolding and assessment in PBL based on this examples
- Discuss the proposed PBL cases



Workshop 2 – Problem Based Learning (PBL), Online Quizzes and Logical Tasks

Session 2: Problem-solving in logical games

Expected Learning Outcomes

- Understand the process of problem-solving
- Being able to develop the methodology for using problem-solving in role-playing by mutual collaboration

Teaching Methods/Approaches

- Teacher presentation and demonstration
- Discussion
- Individual activity
- Group activity - collaboration

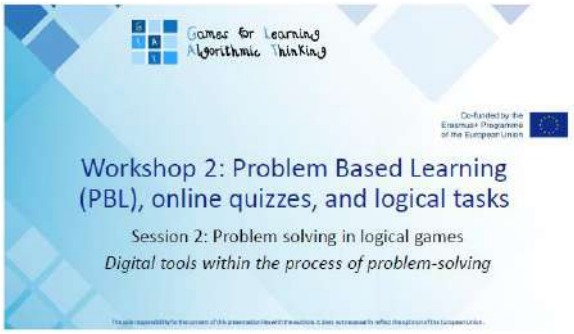
Sources of training materials

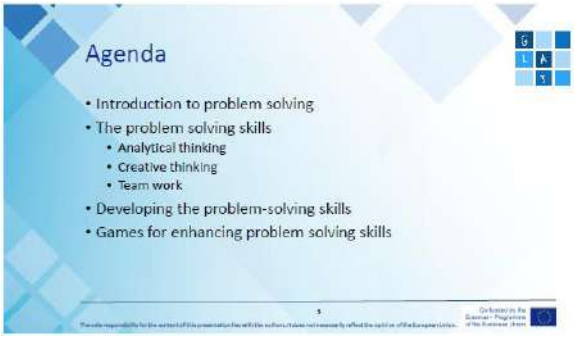
- Digital competence, Europass: https://europass.cedefop.europa.eu/resources/digital-competences (14.6.2019.)
- Production of Creative Game Based Learning Scenarios – A handbook for teachers, ProActiveEU Life-Long Learning project: http://www.ub.edu/euelearning/proactive/documents/handbook_creative_gbl.pdf (21.8.2018.)

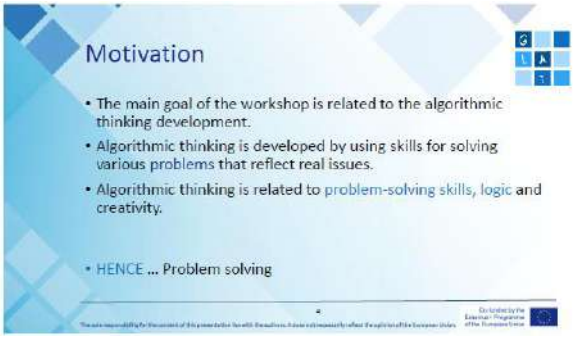
Duration: 3 hours (135 minutes)

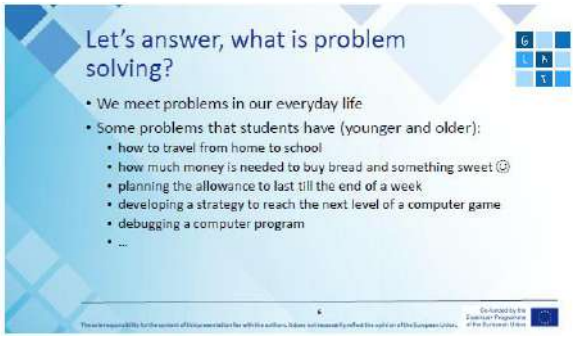
Topic/Sub-topics	Learning Objectives	Evaluation
1. DIGITAL TOOLS WITHIN THE PROCESS OF PROBLEM-SOLVING	*Participants will recognise the process of problem-solving.*	Learners explore and analyse examples of problem-solving techniques in order to point out typical characteristics of logical reasoning.
1.1 Introduction to problem-solving	Identify the concepts of: Analytical ability, Creative Thinking, Initiative, Logical Reasoning	
1.2 Developing problem-solving skills	Understand the role of analytical and creative skills in the process of problem-solving	
1.3 Problem-solving within games and puzzles	Explore the logical features in serious games	
3. ROLE-PLAYING METHODOLOGY	*Participants will recognise the methodology of role-playing in serious games.*	Learners explore and analyse examples of role-playing and knowledge gathering to understand the practice of solving tasks by the active participation of the students in the class and online (work in groups).
2.1 Developing the skills for mutual collaboration accepting different responsibilities (roles) participating in games that support algorithmic thinking	Introducing the power of simulation of playing various complementary roles focused on problem-solving and their implementation Implementing various in-class role-playing scenarios	

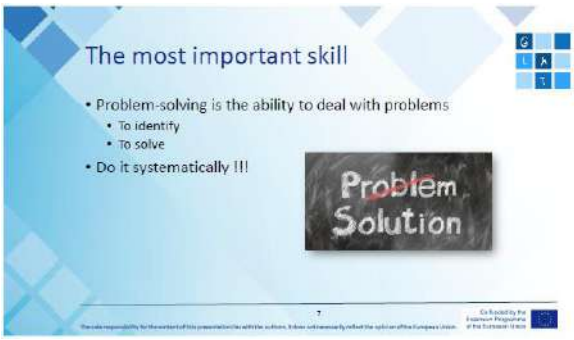
Presentation: Digital tools within the process of problem-solving

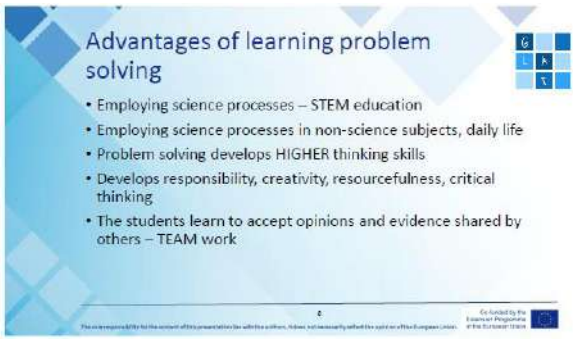












Problem solving is a process...

- **Problem-solving** is a process—an ongoing activity in which we take what we know to discover what we don't know.
- Problem-solving involves three basic functions:
  - seeking information
  - generating new knowledge
  - making decisions

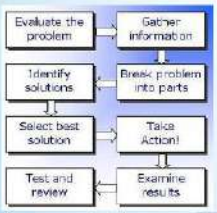
What does it take to be able to solve a problem?

- Problem solving involves both analytical and creative skills.
- The following skills are key to problem-solving:
  - analytical ability
  - creative thinking
  - initiative
  - persistence
- Analytical and critical thinking skills helps to evaluate the problem and to make decisions.

The five steps model

- A logical and methodical approach in finding solution using 5 five-steps model
- Has direct applications to many areas of the curriculum and everyday life
- Steps:
  1. Understand the problem.
  2. Analyze the problem.
  3. Identify various solutions.
  4. Try out a solution.
  5. Evaluate the results.

In other words.. stages to solving a problem



Games for Learning Algorithmic Thinking

Analysing examples

Group activity

Let's learn through example

- The 5 steps through an example
- The following examples are from:
  - Dabar, Croatia, http://ucitelji.hr/vjezbaliste/ or
  - Bebras Computing Challenge, http://www.bebraschallenge.org/

Emily has broken her favourite bracelet. The broken bracelet now looks like this:

Question:

Which of the following four bracelets shows what the bracelet looked like when it was whole?

1. Understand the problem

- It's important that students understand the nature of a problem and its related goals.
- Encourage students to frame a problem in their own words.
- Spread-sheets helps.

- What do you know

- What do you need to find

1. Understand the problem – the example

Let's do it!

- How would you describe the problem?

...

- We should find the bracelet that connects the ends of the string and fits the other parts

2. Analyzing the problem

- Describe any barriers (obstacles)
- In short, what is creating the problem?
  - Encourage students to verbalize these obstacles is always an important step.
- Techniques to understand the nature of a problem and its conditions:
  - List all related relevant facts.
  - Make a list of all the given information.
  - Restate the problem in your own words.
  - List the conditions that surround a problem.
  - Describe related known problems.

2. Analyze the problem – the example

- The data:
  - There are 6 different shapes.
- The conditions:
  - The ends of the string must be next to each other in the bracelet.
  - The shapes needs to be in the same order in the bracelet.
- Obstacles:
  - The order is not symmetrical.

3. Identify various solutions

- There are MANY strategies and no single strategy will work for all problems.
- Some problem-solving possibilities:
  - Create visual images.
  - Create a table.
  - Use physical objects.
    - By moving objects around on a table or desk, students can develop patterns and organize elements.
  - Work backward.
  - Look for a pattern.
  - Create a systematic list (table).

3. Identify various solutions – the example

- The possible solutions are offered
  - See if the solution is meeting the conditions and avoids the obstacles
- The student can draw their idea of a solution and then compare to the offered ones
- The student can move the objects to organize elements

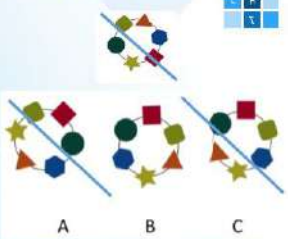
4. Try out a solution

Important clues:

- Keep accurate and up-to-date records of student thoughts, proceedings, and procedures.
- Try to work through a selected strategy or combination of strategies until it becomes evident that it's not working.
  - It needs to be modified.
  - It is yielding inappropriate dana.
- Monitor with great care the steps undertaken as part of a solution.

4. Try out a solution – the example

- Eliminate the ones that don't meet the first condition:
  - the ends of the string must be next to each other: D
- Eliminate the ones that don't meet the second condition:
  - The shapes needs to be in the same order in the bracelet as in the string: A, C

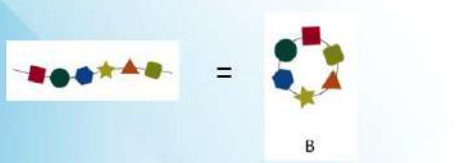


5. Evaluate

- Students should SELF - measure their problem solving skills
- Independence!
- Maybe ask the students questions such as
  - "How do you feel about your progress so far?"
  - "Are you satisfied with the results you obtained?"
  - "Why do you believe this is an appropriate response to the problem?"

5. Evaluate – the Example

- The solution is B



Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Analysing examples

Group activity

Example

- Solving mathematical problems by:
  - guessing the answer and then checking that the guess fits the conditions of the problem.
- "Guess and Check" - a problem-solving strategy

- The problem:

Ben knows 100 football players by name. Ten are from Real Madrid. The rest are Juventus and Barsa. He knows the names of twice as many Barsa as Juventus. How many Juventus players does he know by name?

The steps for the example

1. Understand the Problem
- This involves finding the key pieces of information needed to find the answer.
- May require reading the problem several times, and/or students putting the problem into their own words.

We need to find out how many players does Ben knows from Juventus, by following the given facts and conditions

The steps for the example

2. Analyze
- Gather data, conditions, obstacles
- Data: 10 Real Madrid's
- Conditions:
  - the number of Juventuses and Barsas should equal 90
  - there are twice as many Barsas as Juventuses

The steps for the example

3. Identify various solutions
- Choose a Strategy
  - Use the "Guess and Check" strategy.
  - Guess and check is often one of the first strategies that students learn when solving problems.
  - This is a flexible strategy that is often used as a starting point when solving a problem, and can be used as a safety net, when no other strategy is immediately obvious !!!

The steps for the example

4. Try a solution
- Now, solve the problem. You may want to set up a table to record the guesses.

Guess Number	Juventus	Barsa	Real Madrid	Total
First Guess	10	20	10	40

- Guess a greater number of Juventuses.

Guess Number	Juventus	Barsa	Real Madrid	Total
First Guess	10	20	10	40
Second Guess	20	40	10	70

- Now guess a greater number of Juventuses.

Guess Number	Juventus	Barsa	Real Madrid	Total
First Guess	10	20	10	40
Second Guess	20	40	10	70
Third Guess	40	80	10	130

Finally, the answer!

Guess Number	Juventus	Barsa	Real Madrid	Total
First Guess	10	20	10	40
Second Guess	20	40	10	70
Third Guess	40	80	10	130
Fourth Guess	30	60	10	100

The steps for the example

5. Evaluate the solution
- Read the problem again to be sure the question was answered.
  - Yes, I found the number of Juventuses.
- Check the math to be sure it is correct.

30 doubled is 60.

30 + 60 + 10 = 100

- SOLUTION!

Games for Learning Algorithmic Thinking

Analysing examples

Group activity

Let's try together

- Working in groups solve the problem by following and explaining the steps:

1. Understand the problem
2. Analyze
3. Identify various solutions
4. Try out a solution
5. Evaluate the results

Games for Learning Algorithmic Thinking

Creative thinking and team work

Creative thinking

- Sometimes, creative thinking will be necessary
  - ideas for resolving the problem and find fresh approaches

Don't forget... TEAM work

- Team working is often a key component in problem-solving
  - Not everyone has analytical and critical thinking potential
- The students learn to accept opinions and evidence shared by others
  - Latter about Role playing in team working

Why groups are more effective decision makers

- Only one might not have all the knowledge or resources to find the solution
- Groups "see" from different angles
- Group easier test different ideas before one is selected and implemented

WS2 Rijeka 28-29 August 2018

Games for Learning Algorithmic Thinking

Developing the problem-solving skills with games

Developing the problem-solving skills

Most problem-solving skills are developed through everyday life and experience by utilizing:

- 'Mind games' such as cryptic crosswords, Sudoku, chess, bridge,...
- Computer games – the chosen ones can include:
  - strategic planning
  - critical and statistical analysis
  - assessing the pros and cons of different courses of action

Enhancing Problem solving skills with games

- Utilizing digital tools
- We need digital competence to choose the right games – edu-websites
- www.funbrain.com
  - an award-winning interactive learning and "edutainment Web site" links K-8 children, parents, and teachers. It gives a choice of math problems and the level you want to practice.
- www.educationworld.com
  - High energy, engaging games in math, Language Arts, Science, Social Studies, The Arts, and thinking games all to be found here either by subject or grade level.

Some games for elementary school

- https://www.funbrain.com/pre-k-and-k-playground
  - Fun games for warming up
- https://www.funbrain.com/games/pig-pile
  - Fun game for logical que of steps
- https://www.funbrain.com/games/word-derby
  - English language
- https://www.funbrain.com/games/inkster
  - Math (grade 3-4)

Some fun

- Games that develop problem solving skills
  - https://www.funbrain.com/games/pig-pile
  - https://www.funbrain.com/games

Games for Learning Algorithmic Thinking

Discussing examples

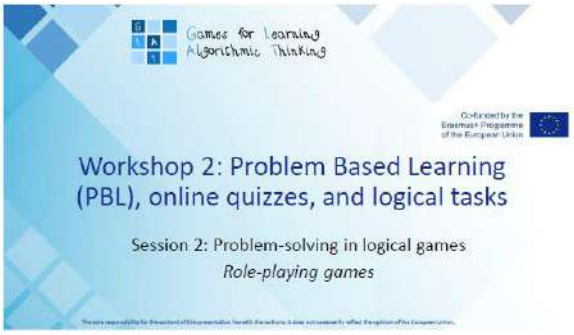
Group activity

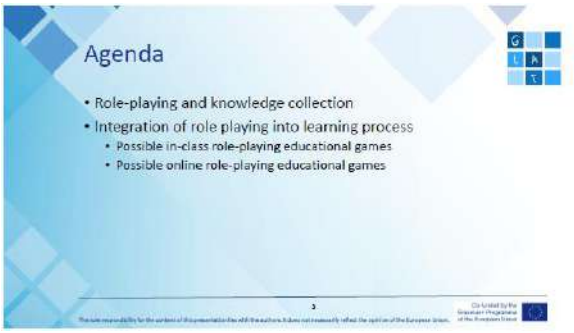
Discussing examples

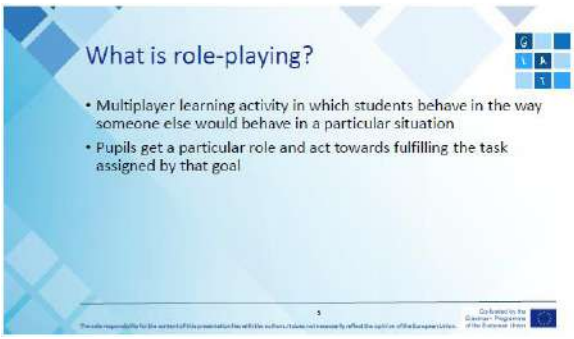
- Explore and discuss in groups possible usage of games that develop problem solving skills as well as other games and math tasks at www.funbrain.com in different subjects in primary school.
- Share your ideas with other groups.

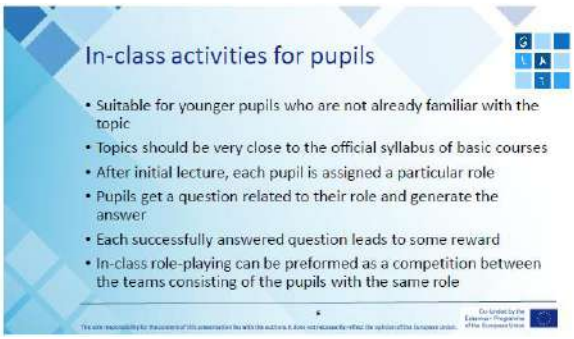
Questions

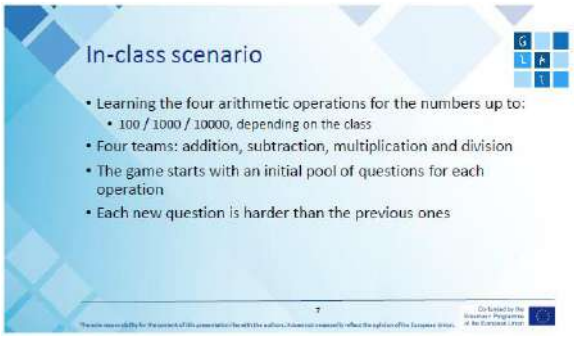
Presentation: Role-playing games

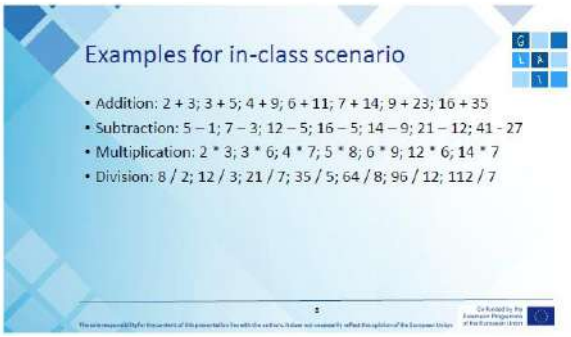












In-class role-playing 1/2

- Teacher assigns the roles to all the pupils
- The board is divided into four sectors, one for each team
- Pupils with the same role make one team
- Teams are named (for example: flowers, butterflies, birds, fishes)

https://easydrawingguides.com/how-to-draw-cartoon-flowers/

In-class role-playing 2/2

- All team members make a queue.
- Teacher poses the question.
- The pupil answers the question, other team members can help
- If the answer is correct, teacher draws a part of the corresponding team symbol.
- After one tour of the game, the roles are exchanged.

All the drawings are copied from https://easydrawingguides.com/how-to-draw-cartoon-flowers/

More advanced level

- Prerequisite: successfully finished basic level by all the four teams
- Expressions with several operations
- Teams announce that they know the right answer
- The fastest team answers the question
- If the answer is wrong, the second team can try as well
- If no one gives the right answer, teacher should explain why
- The expressions should be gradually introduced:
  - 2+5−6 8−4+9 6+8−3−5 9−4+5−2+3
  - 2*5*6 8/4*9 6*8/3*5 9*4*5/2/3
  - 2+5*6 8/4+9 6*8+3*5 9*4+5+2*3…

Role-playing online scenario – an example

- The goal is to paint an image, for example, one or several cartoon heroes, which are invisible at the beginning of the game
- The computer game teacher is a painter, who reacts to pupil's responses
- Again, pupils are divided into teams
- Each team has its own color: yellow, blue, green, and pink
- Questions are randomly generated
- Correct answer leads to one small part of the image of the heroes painted in the color of the team
- When the game ends, the whole image is completed

Games for Learning Algorithmic Thinking

Computer role-playing games

Computer role-playing games

- One of the most important genres on PC
- Include complex, controllable relationships with companions or non-playable characters
- Forums and guides exist in support of this play style
- Boys usually play massively multiplayer online role-playing games (MMORPG)
  - A virtual SF or fantasy world
  - Players interact with one another to conquer the virtual world
- Girls prefer the canonical role-playing games (RPG)
  - Emphasize story
  - Let you inhabit a customizable character through skill points, inventory, and dialogue decisions

Famous RPGs

- Pokemon
- Final Fantasy
- Dragon Quest
- World of Warcraft
- Mass Effect 2
- Ever Quest
- Guild Wars
- Massively played
- Generate enormous income

Edutainment software and role-playing

- Edutainment software supports role-playing
- Games are adjusted to pupil's age (complete beginners, elementary school, intermediate, exam levels)
- Typical examples:
  - Mathematics: Math Quest, Bunny Math Race for Kids, Monster Squeeze
  - English: Catch the Burglar, Free letters and sounds game, Emergency Services Role Play Cards, Drama Word Talent
  - Science: Jobi's Animal Barn, Classcraft, Cytosis (board game)
  - History: The Oregon Trail, Medieval Merchants (board game), Civilization Revolution, The Settler

The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Union.

Co-funded by the Erasmus+ Programme of the European Union

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Analysing examples

Group activity

Example 1: Bunny Math Race

- Available from http://beiz.com/bunnymathrace.html
- FREE download: https://itunes.apple.com/app/id661137493
- Race between up to 6 bunnies
- Intended for children from 3 to 8 years
- Reward: more carrots ☺

Bunny Math Race - Video

https://www.youtube.com/watch?v=fYH0iKsWzRc

Example 2: Jobi's Animal Barn

- Available from: https://www.yellephant.net/
- Free download: https://appsonwindows.com/apk/2002373/
- Features:
  - How do farmers grow and harvest vegetables?
  - How to clean the barn: remove spider webs, collect chicken eggs, help small ducklings find their mom, wash the piggies, feed the cows?
  - How to take care of sheep: feeding, shearing and guarding?
  - How do farm animals look like?
  - How to look after them?
  - How do root vegetables look like?
- Intended for children from 3 to 8 years

Jobi's Animal Barn - Video

https://www.youtube.com/watch?v=8xjjQSlwTE0

Advantages of role-playing

- Stimulates active learning - children learn by doing
- Amusing
- Very popular and easy to use
- Important part of child development
- Builds confidence, creativity communication and problem solving
- Fosters the development of motor skills
- Role-playing can be very competitive
- Stimulates interaction, mutual communication and collaboration
- Leads towards achieving common goals

Games for Learning Algorithmic Thinking

Role-playing scenarios with problem solving

Reminder of the in-class scenario

- How does a scenario fulfill the five criteria:
  1. Understand the problem
  2. Analyze the problem
  3. Identify various solutions
  4. Try out a solution
  5. Evaluate the results

In-class scenario

- Learning the four arithmetic operations for the numbers up to:
  - 100 / 1000 / 10000, depending on the class
- Four teams: addition, subtraction, multiplication and division
- The game starts with an initial pool of questions for each operation
- Each new question is harder than the previous ones

1. Understand the problem

- Motivation:
  - Each pupil should demonstrate the capability to perform the four basic arithmetic operations.
- Advantage of role-playing for problem perception
  - Four eyes watch better than two.
  - Any misunderstanding can be avoided by interactive communication between team members.
  - Team leader can negotiate with the teacher to resolve prospective dilemmas.

Explanation of the problem

- Remember what is your role (+ - * /).
- See the question carefully (2 + 3; 5 – 1; 2 * 3; 8 / 2 ...).
- Check whether it represents your role.
- Guess the result of this operation (5; 4; 6; 4 ...).
- Tell it to all the others.
- If you don't know the result, pass the task to next member of your team (or ask your team members for help).

2. Analyze the problem

- Gather data:
  - arithmetic expressions with two numbers and one arithmetic operator
- Obstacles:
  - Expressions are sorted from easiest to most complex.
  - Expressions with all the arithmetic operators of same complexity are randomly ordered.
- Conditions:
  - Find the correct answer.

3. Identify various solutions

- The solution part is either a problem solving task (like the example on the previous slides), or a multiple choice question task (like in the Bunny Math Race).
- For both approaches, the most appropriate strategy is "Guess and check."
- Pupils guess the prospective correct answer and check it with the exact result.

Role-playing and various solutions

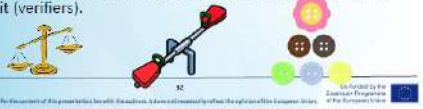
- This strategy is the most natural for the pupils, who will definitely first use their own fingers to check the correctness.
- Several pupils will sometimes generate more than one solution.
- Their mutual communication increases the willingness to propose the correct answer.
- Teacher's help should be encouraged in favor of delegating the role to other pupils, or to the whole team.

4. How to check the answer correctness?

- Scale
- Seesaw
- Buttons

Try the solution

- Seesaw is useful for the easiest exercises.
- Scales are applicable for more complex tasks including multiplication.
- Colorful buttons are the universal solution verifier.
- By using them, twice as much roles can be assigned, four for those who guess the answer (guessers), and four for those who verify it (verifiers).



Creative thinking and active learning

- The verifiers should arrange the buttons in a way that proves the correct solution.
- The verifiers should arrange the buttons in a way that disproves the wrong solution.
- In both cases, the verifiers learn by doing, or even better, learn by becoming.
- Roles are exchanged after one tour, each pupil is a prospective verifier.

5. Evaluation

- If the answer is correct, one part of the corresponding team symbol is drawn on the board.
- The direct correct answer means a full part of the drawing.
- The correct answer obtained after several attempts will be drawn fully after all the attempts.
- With such approach the winners will be visible soon.

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Analysing role-playing games examples

Group activity

The problem- Beavers in the elevator

- 3 beavers enter in an elevator on 9th floor. The first beaver goes to the 1st floor, the second to 16th, and the third to the 20th floor.
- In what order should the elevator go to make the shortest route possible?

A. 9 →16→20→1
B. 9 →20→16→1
C. 9 →1→16→20
D. 9 →16→1→20

Available from: http://ucitelji.hr/vjezbaliste/

Your assignment

- How does the problem „Beavers in the elevator" fulfil the five learning criteria?
- Which are the five criteria:
  1. Understand the problem
  2. Analyze the problem
  3. Identify various solutions
  4. Try out a solution
  5. Evaluate the results

Understand the problem

- Three pupils enter the elevators on the 9th floor
- Role 1 should arrive to 1st floor
- Role 2 should arrive to 16th floor
- Role 3 should arrive to 20th floor
- The elevator should travel as short as possible

Analyze the problem

- There are many solutions
- Start from 9th floor and go down
- Start from 9th floor and go up
- If the elevator goes up, it can stop on the:
  - 16th floor and carry on up or down
  - 20th floor and carry on up or down
  - ...
- If the elevator goes down, it can stop on the:
  - 1st floor and then go up
  - ...

Identify various solutions

- A. 9 -> 16 -> 20 -> 1
- B. 9 -> 20 -> 16 -> 1
- C. 9 -> 1 -> 16 -> 20
- D. 9 -> 16 -> 1 -> 20
- E. 9 -> 20 -> 1 -> 16
- F. 9 -> 1 -> 20 -> 16

The game - prerequisites

- Plates with the numbers 1., 16. and 20. (the role)
- Three sets of tiles, each consisting of the numbers from 1 to 20
- Paper ladder with numerate fields
- Paper ladder is put on the floor

1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	2

- The class is divided into k triples
- If the number of pupils is 3*k – 1, the teacher becomes a member of one team
- If the number of pupils is 3*k + 1, one pupil becomes a judge

41

Playing the game

1. All the pupils from the team step on the field **9**
2. Randomly selected role (let's call it **A**) takes all pupils towards its floor
3. Pupils go field by field, counting the number of steps they made
4. Pupil **A:**
   1. Arrives at its destination
   2. Finishes the journey
   3. Takes all the three tiles with the corresponding number of steps
   4. Shares them to other pupils
5. The game is repeated with the remaining two pupils

42

Try out the solutions

Potential solutions:	Path length:
• A. 9 -> 16 -> 20 -> 1	• (16 - 9) + (20 - 16) + (20 - 1) = 30
• B. 9 -> 20 -> 16 -> 1	• (20 - 9) + (20 - 16) + (16 - 1) = 30
• C. 9 -> 1 -> 16 -> 20	• (9 - 1) + (16 - 1) + (20 - 16) = 27
• D. 9 -> 16 -> 1 -> 20	• (16 - 9) + (16 - 1) + (20 - 1) = 41
• E. 9 -> 20 -> 1 -> 16	• (20 - 9) + (20 - 1) + (16 - 1) = 45
• F. 9 -> 1 -> 20 -> 16	• (9 - 1) + (20 - 1) + (20 - 16) = 33

43

Evaluate the results

- All the teams play the game
- They remember the order of finishing the game
- The pupil of each team with three tiles sums their amounts
  - For solution A. 7 + 4 + 19 = 30
  - For solution B. 11 + 4 + 15 = 30
  - For solution C. 8 + 15 + 4 = 27
  - For solution D. 7 + 15 + 19 = 41
  - For solution E. 11 + 19 + 15 = 45
  - For solution F. 8 + 19 + 4 = 33
- The shortest path is C. 9 -> 1 -> 16 -> 20

44

Questions

45

The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Union.

Co-funded by the Erasmus+ Programme of the European Union

Workshop 2 – Problem Based Learning (PBL), Online Quizzes and Logical Tasks

Session 3: Online quizzes and logical tasks

Expected Learning Outcomes

- Choose logical tasks suitable for different school subjects and providing propaedeutic for algorithms and programming
- Create new examples of logical tasks suitable for different school subjects and providing propaedeutic for algorithms and programming

Teaching Methods/Approaches

- Teacher presentation and demonstration
- Individual activity
- Group activity - collaboration

Sources of training materials

- Bebras, International Challenge on Informatics and Computational Thinking: https://www.bebras.org/ (30.6.2018.)
- LearningApps: https://learningapps.org/ (4.7.2018.)
- Teaching London Computing: https://teachinglondoncomputing.org/ (4.7.2018.)
- e-laboratorij CARNet, ankete/kvizovi: http://e-laboratorij.carnet.hr/category/interaktivni-sadrzaji/ (4.7.2018.)

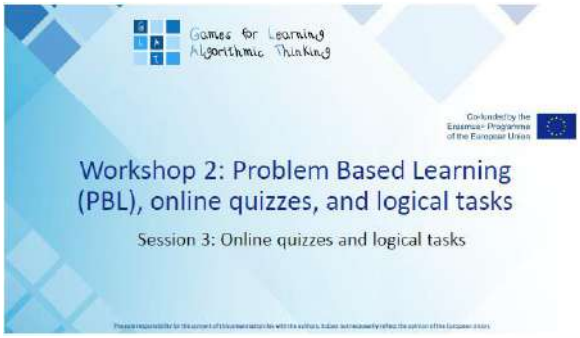
Web 2.0 tools:

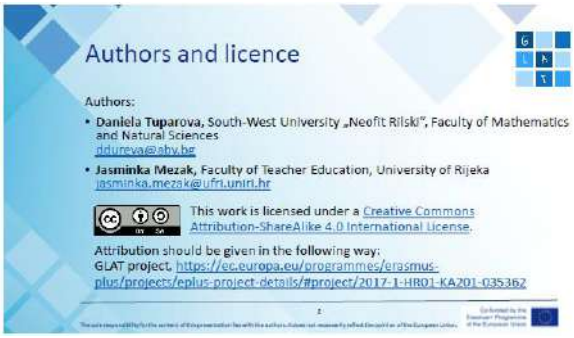
- Learningapps: https://learningapps.org/ (4.7.2018.)
- Kahoot: https://kahoot.com/ (30.6.2018.)
- Wizer: https://app.wizer.me/ (30.6.2018.)
- Match the memory: https://matchthememory.com/ (4.7.2018.)

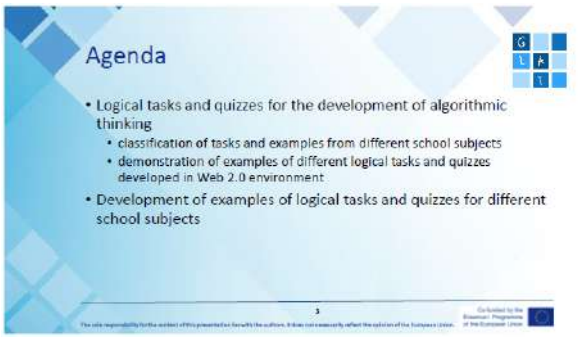
Duration: 3 hours (135 minutes)

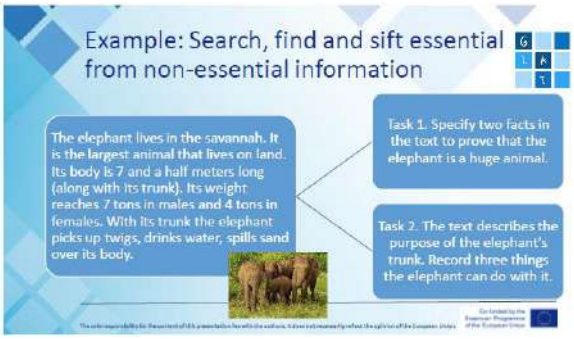
Topic/Sub-topics	Learning Objectives	Evaluation
1. LOGICAL TASKS AND QUIZZES FOR DEVELOPMENT OF ALGORITHMIC SKILLS AND THINKING	*Participants will be able to classify logical tasks for propaedeutic of algorithm and programming and construct quiz appropriate for implementation.*	Learners explore examples and resources in order to discuss different types of logical tasks for the development of algorithmic skills and their application in school.
1.1. Classification of tasks for the development of algorithmic skills and thinking; examples from different school subjects	Classify logical tasks providing propaedeutic for algorithms and programming	
1.2. Main requirements for online quizzes development	Construct quizzes appropriate for online implementation	
1.3. Demonstration of examples of different logical tasks and quizzes developed in Web 2.0 environment (Learningapps.org, Kahoot, etc.) and applicable in school subjects	Experiment with existing examples of logical tasks and quizzes in the form of games	
2. DEVELOPMENT OF EXAMPLES OF LOGICAL TASKS AND QUIZZES	*Participants will be able to create examples of logical tasks, appropriate for different school subjects.*	Learners modify some of the examples and discuss the possibilities of implementation of the tasks in school subjects and lessons (group activity).
2.1. Modification and adaptation of examples of logical tasks for different school subjects	Create new examples of logical tasks based on given examples Give new examples of logical tasks for algorithmic thinking	
2.2. Development of examples of logical tasks		

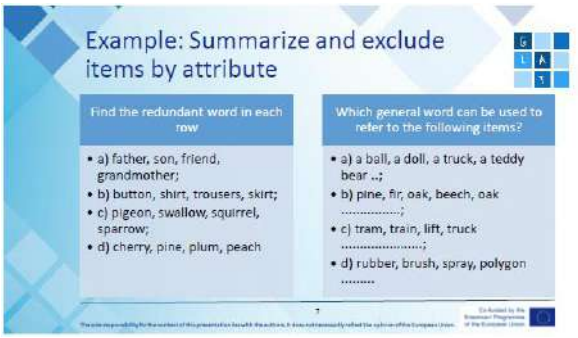
Presentation: Online quizzes and logical tasks

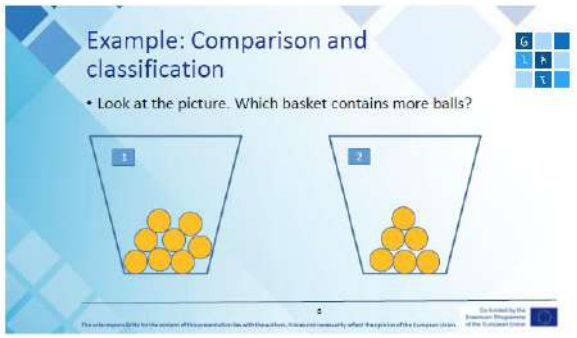






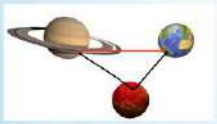






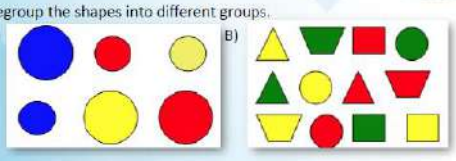
Examples: Comparison and classification

- Earth, Jupiter and Mars are planets from the Solar System. Jupiter is larger than Earth, and Mars is smaller than Earth. Arrange the planets by size, starting with the smallest.



Example: Comparison and classification

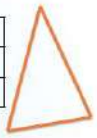
- Regroup the shapes into different groups.
- A) B)



Example: Define and follow the sequence of actions

- Which is the correct sequence for finding the circumference (perimeter) of a triangle? Use numbers 1, 2 and 3 to mark the sequence in the table.

	Sum up the lengths of the sides
	Write and record the answer
	Measure the length of the sides



Examples: Description of a sequence of actions

- Write the sequence of actions to find the side length of a square in a given perimeter.
- $p = 4. a \;\; a = p:4$

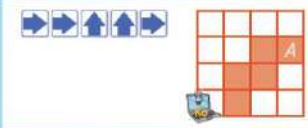
Example: Detection of errors in an algorithm

- Fix any errors:

To clean my teeth:
1. I get a brush.
2. Brush my teeth.
3. Place some paste on the brush.
4. Rinse with water.

Example: Detection of errors in an algorithm

- Ann used arrows to write an algorithm in Algo with commands for moving in 4 directions. The aim of the algorithm is Ko to reach point A on the way colored in brown. Find the errors in the code.



Example: Detection of errors in an algorithm

- Bobby is in front of the hot drinks machine. He tried to buy hot chocolate in the following way:
  1. Take the cup of hot chocolate.
  2. Put a coin.
  3. Press the hot chocolate button.
- Help Bobby to perform the actions in the right order so that he could get hot chocolate from the vending machine.

Examples: Detection of errors in an algorithm

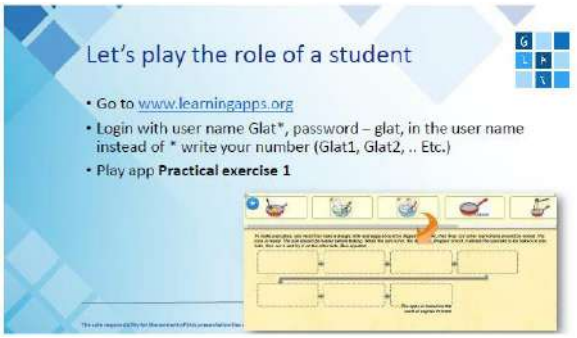
An element of a user's profile is hidden in the grid.
A) What kind of element is it?
B) Which is the right code for Tabby to find this element?

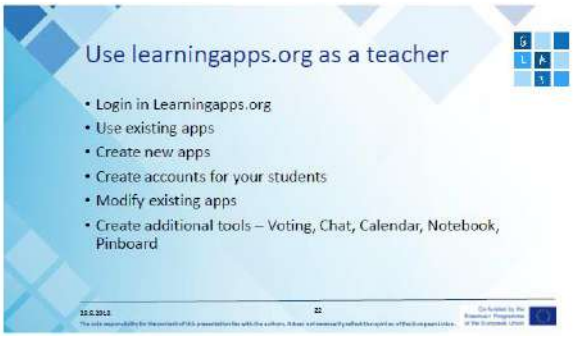
	a	w	o	r
p	i	a	t	d
a	r	o	a	l
s	a	l	r	e

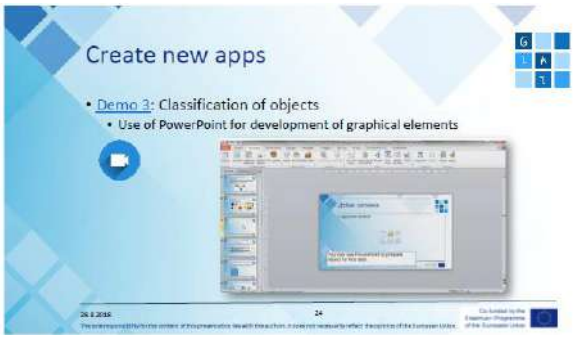
A)
B)
C)

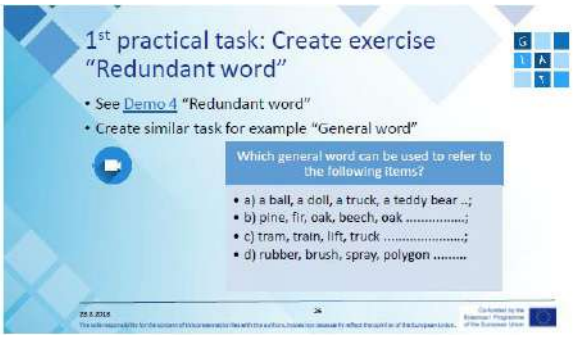
The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Union.

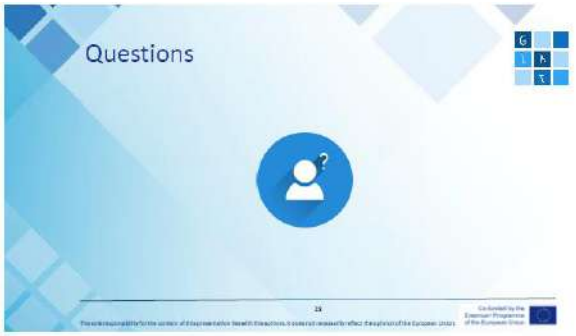
Co-funded by the Erasmus+ Programme of the European Union











Workshop 2 – Problem Based Learning (PBL), Online Quizzes and Logical Tasks

Session 4: Using Web 2.0 tools for creating quizzes and logical tasks

Expected Learning Outcomes

- Identify the advantages of Web 2.0 tools for quizzes and logical tasks.
- Create quizzes, logical tasks, and interactive worksheets using Web 2.0 tools.
- Create new examples for quizzes, logical tasks, and interactive worksheets.

Teaching Methods/Approaches

- Teacher presentation and demonstration
- Discussion
- Individual activity
- Group activity - collaboration

Sources of training materials

- Dabar, međunarodno natjecanje iz informatike I računalnog razmišljanja: http://ucitelji.hr/dabar/ (30.6.2018.)
- Bebras, International Challenge on Informatics and Computational Thinking: https://www.bebras.org/ (30.6.2018.)
- LearningApps: https://learningapps.org/ (4.7.2018.)
- E-laboratorij CARNet, ankete/kvizovi: http://e-laboratorij.carnet.hr/category/ankete-kvizovi/ (4.7.2018.)
- E-laboratorij CARNet, ankete/kvizovi: http://e-laboratorij.carnet.hr/category/interaktivni-sadrzaji/ (4.7.2018.)

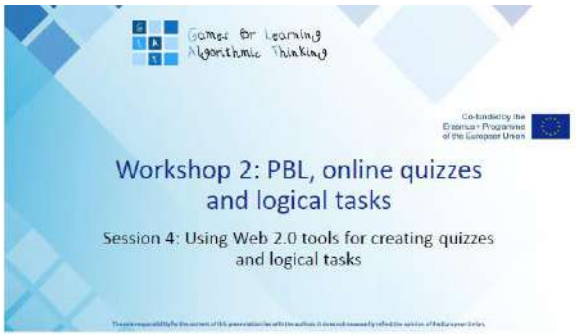
Web 2.0 tools:

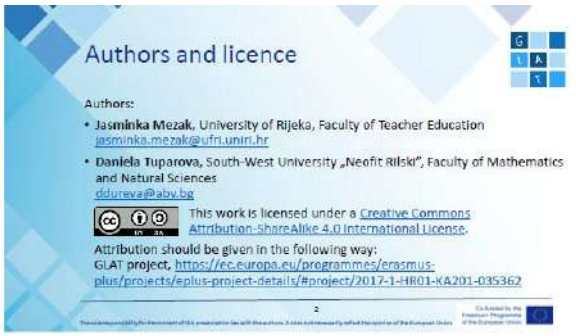
- Kahoot: https://kahoot.com/ (30.6.2018.)
- Wizer: https://app.wizer.me/ (30.6.2018.)
- Match the memory: https://matchthememory.com/ (4.7.2018.)

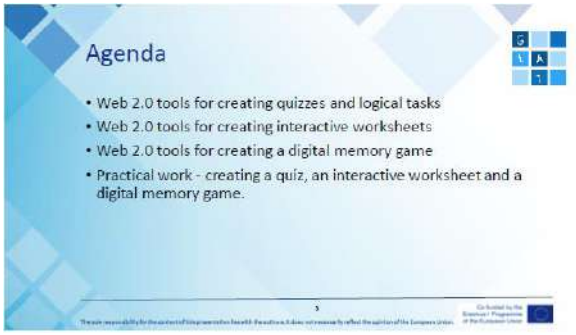
Duration: 3 hours (135 minutes)

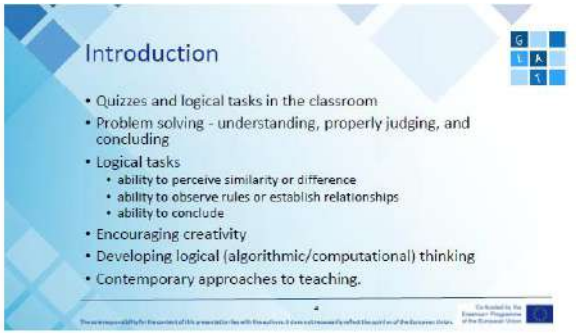
Topic/Sub-topics	Learning Objectives	Evaluation
1. WEB 2.0 TOOLS FOR CREATING QUIZZES AND LOGICAL TASKS	*Participants will be able to identify the advantages of using Web 2.0 tools for creating quizzes and logical tasks.*	Learners explore examples and resources in order to discuss the potentials of Web 2.0 tools for creating quizzes and logical tasks (group activity).
1.1. Investigate examples of Web 2.0 tools for creating quizzes and logical tasks.	Use the preselected Web 2.0 tools (Kahoot, Wizer, Match the memory) to create quizzes, interactive worksheets, memory cards, etc.	
2. CREATING QUIZZES AND INTERACTIVE WORKSHEETS	*Participants will be able to create an online quiz and interactive worksheet.*	Learners solve online quizzes and interactive worksheets (group activity) made by teacher. Learners create an online quiz and an interactive worksheet (individual activity) which will be evaluated by the teacher.
2.1. Creating an online quiz	Create an online quiz and interactive worksheet for the preselected task	
2.2. Creating interactive worksheet		
3. DEVELOPMENT OF EXAMPLES OF LOGICAL TASKS	*Participants will be able to create examples of logical tasks, appropriate for different school subjects.*	Learners discuss the potentials of digital tools and create new examples for logical tasks that encourage algorithmic/computational thinking (group activity).
3.1. Modification and adaptation of examples of logical tasks for additional school subjects	Create new examples of logical tasks based on given examples Prepare new examples of logical tasks for algorithmic thinking	
3.2. Development of examples of logical tasks		

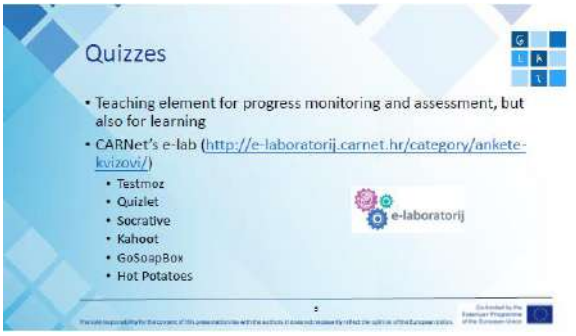
Presentation: Using Web 2.0 tools for creating quizzes and logical tasks

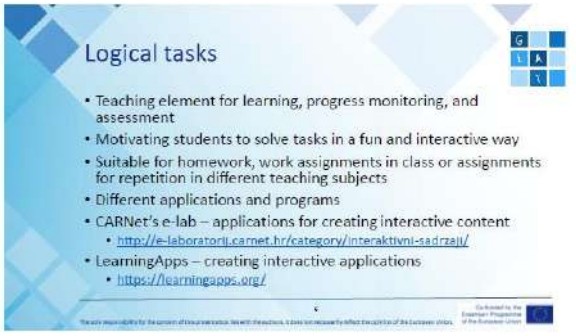


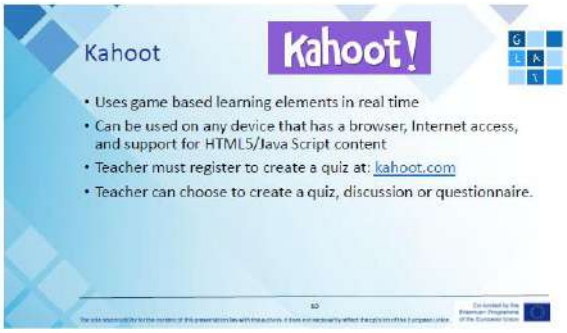


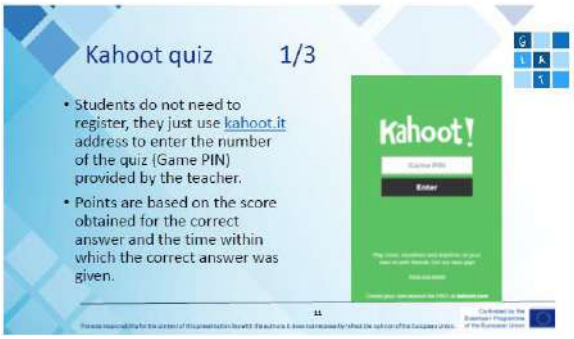


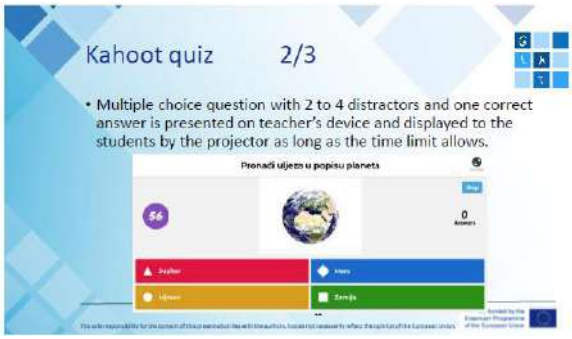


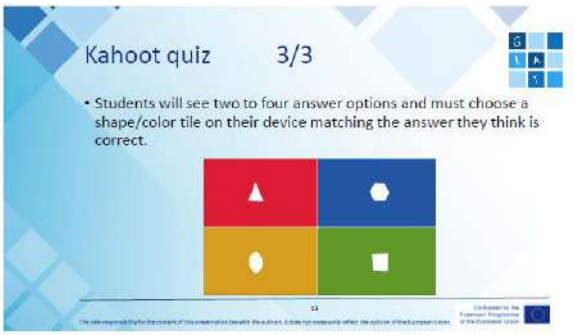












Use Kahoot! as a teacher

- Login at kahoot.com and create an account

Task:

- Create a quiz with at least 3 multiple-choice questions
- Include cover image
- The 1st question is text-based, e.g. mathematical task
- The 2nd question and the 3rd question contain graphics (distractors are text-based), e.g. about Science course.

Games for Learning Algorithmic Thinking

Web 2.0 tool Wizer

Wizer

wizer.me

- Creating interactive worksheets
- The Wizer worksheet allow quick creation of a different question types: open questions, multiple choice, matching pairs, fill in the blank, fill on an image, tables, etc.
- Worksheets may contain text, images, audio, video, links
- The number of points can be assigned for each question.

Wizer sign up

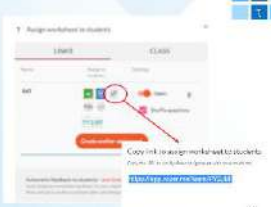
- Teacher must register to create a worksheet at https://app.wizer.me
- Students do not need to register, they just use the address with PIN provided by the teacher.
- Thy can solve a worksheet in the classroom or at home.

Wizer question types

- Wizer question types: Open answer, multiple choice, matching, fill in the blank, fill on an image, table activity, sorting into groups and drawing.

Solving a Wizer Worksheet

- Option: Assign to learners
- After solving the tasks on the worksheet, students choose „Hand in work" to submit the results.



Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Solving an interactive worksheet

Group activity

Solving Wizer worksheet

- Let's play the role of a student and solve the worksheet at https://app.wizer.me/learn/PY2JBF
- Sign in with nick name and use password PY2JBF

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Creating Wizer worksheet

Individual activity

Use Wizer as a teacher

- Login at https://app.wizer.me and create free teacher account

Task:

- Create a worksheet with at least 3 different types questions
- Examples of types: Open answer, multiple choice, matching, fill in the blank, fill on an image.

Games for Learning Algorithmic Thinking

Match the Memory

Match the Memory

Match The Memory

- Creating digital memory games with customized combination of images and text
- Teacher can register to create a game at https://matchthememory.com/
- Students do not need to register, they just use the address of the memory game.

Creating a memory game 1/2

- Teacher needs to enter the address and the title of the game, and to define the number of cards.
- The theme is selected for the card background and the type of card is defined: text-text, image-image, text-image, text A-text B, image A-image B.

Creating a memory game 2/2

- Pairs are edited under the category Cards by defining corresponding type, images and text.

Memory games list

- By selecting My account, there is a list of created games that can be re-edit, delete or play.

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Solving a memory game

Group activity

Solving Match the Memory game

- Let's play the role of a student and solve the game at https://matchthememory.com/math10

Creating a memory game

Individual activity

Use Match the memory as a teacher

- Login at https://matchthememory.com/
- Task:
  - Create a memory game with 4 cards
  - Type of the game should be Picture – Only
  - Upload six pictures
  - Play the game
  - Change type of game into Text-Picture
  - Change type of game into Picture A – Picture B.

Creating quizzes and logical tasks

Group activity

Creating quizzes and logical tasks 1/2

- Each group should choose a different school subject.
- Set up the topic and assign learning outcomes and evaluation elements.
- Discuss the learning outcomes that you want to accomplish using the quiz, worksheet or memory games that relate to the subject and those relating to algorithmic thinking.

Creating quizzes and logical tasks 2/2

- Place the evaluation elements into task with:
  - Kahoot! quiz
  - Match the memory game
  - Wizer.me
- Present your quiz, memory game and worksheet to other groups.

Questions

Workshop 2 – Problem Based Learning (PBL), Online Quizzes and Logical Tasks

Session 5: Designing learning scenarios for logical tasks

Instructions for the participants

Expected Learning Outcomes

- Create learning scenarios in order to develop innovative ideas for carrying out logical tasks and online quizzes
- Implement learning scenarios for different courses in the classroom with the students from 1st to 4th grades of primary school

Individual Assignment:

Your task is to prepare the learning scenario based on PBL and logical tasks in written form and in graphical form using LePlanner. You could choose any school subject and any lesson within the subject for students from your class, considering that the activity should be completed in two months.

This is the **first version of the 2nd learning scenario** which you will continue to design with the online help of your mentor. The completed version of the scenario will be **reviewed** by the mentor and the final refined versions you will **implement** in the classrooms with your students.

You are also supposed to write the **reflection** on conducted activities.

Duration: up to 2 months for the whole assignment

ASSIGNMENT STEPS	
12.	Choose a school subject – plan the activities that will be carried out in your class next month.
13.	Use the **Learning Scenario Template** form (Annex 1) for textual version and LePlanner for the graphical version of your scenario.
14.	Specify the **Learning outcomes**: - state ***general learning outcomes*** related to the course that will include problem teaching and logical tasks - state ***learning outcomes oriented on algorithmic thinking***
15.	Describe the **Aim and tasks** of teaching and give a **Short description of the activities**. Plan the activities that will encourage your students for seeking the information, critical and logical thinking as well as collaborating while solving the problem. The activities should include work on computer/tablet/smartphone (not only unplugged activities).
16.	Specify the **Keywords**, **Correlation, and Interdisciplinarity** with other courses or topics, and the **Duration of activities**.
17.	Point out the **Learning and teaching strategies and methods**. Specify the **Teaching forms**: combine individual and group work; since this is a problem teaching, collaborative learning should be included.
18.	Choose **Tools** or games that will be used for quizzes or logical tasks on computer/tablet/smartphone for at least one example.

	Point out all **Resources/materials** which will be required for the teacher as well as for students.
19.	Elaborate the **Teaching summary** as **Motivation (Introduction)**, **Implementation** and **Evaluation (Reflection)**. This part develops in details previously mentioned short description of activities. Logical tasks or quizzes can be used in each part of teaching summary (you will add links to the developed online tasks later).
20.	Create suitable quizzes and/or logical tasks with chosen tools. Pay attention to the copyright for images, videos, and other materials collected from the web. Photographing your students requires written parents' consent.
21.	In **Annexes** box provide examples and tasks you have created by yourself as well as link to the graphical version of the Learning scenario in LePlanner. Links should be direct to the created tasks prepared for solving by students (not to the tasks open in editor).
22.	**Examples and game references** box should contain sources you will use for the activities.
FOLLOW-UP ACTIVITIES	
5.	Upload your completed first version of learning scenario to the Moodle e-course. Mentor will review and correct your scenario.
6.	Upload your final version of learning scenario considering mentor's suggestions and corrections.
7.	After mentor's approval, implement your learning scenario in the class with your students.
8.	Post a **reflection** on conducted activities in the forum: • Write more extensive description on implementation of the activity in your class. • Describe how your students have accepted learning activity. • Describe the achievement of all planed learning outcomes, both general and oriented on algorithmic thinking. • Define what you would like to change before the next implementation of the scenario.

Workshop 3: Games and Tools for Programming

Workshop 3 – Games and Tools for Programming

Workshop schedule

Day 1

Introduction to Workshop 3

Duration: 1 hour (45 minutes)

Introductory presentation: Workshop 2 - follow-up activities
Introduction to the Workshop 3

Session 1: Introduction to Inquiry Based Learning

Duration: 1 hour (45 minutes)

Lecture: Definition of Inquiry Based Learning (IBL). Comparing IBL with Project-Based Learning
Demonstration: Examples of IBL implementation and Project-Based Learning in primary education
Group work: discussing concepts of Inquiry Based Learning and describing an example of lesson

Session 2: Basic programming concepts

Duration: 1 hour (45 minutes)

Lecture: Basic programming concepts: sequence, branching, loop, variables
Demonstration: Games for learning programming: Games Run Marco, Blockly-games, Code.org
Group work: analysing the existing didactical games and discussing the advantages and disadvantages of the games and possibilities for didactical implementation

Session 3: Learning programming with games and stories

Duration: 2 hours (90 minutes)

Lecture: Development of Computational Thinking (CT) with games and stories
Demonstration: Didactic computer stories and games (in ScratchEd community)

Session 4: Introduction into visual programming with Scratch

Duration: 3 hours (135 minutes)

Lecture: Introduction into visual programming with Scratch
Group work: Creating stories and games with Scratch

Individual work/Group work: Storytelling with Scratch

Day 2

Session 5: Implementing Computational Thinking and programming with GBL tools

Duration: 1 hour (45 minutes)

Lecture: Introducing programming in the classroom from teacher's perspective
Demonstration: Video presentation of Scottie Go! usage as a way to learn programming
Group work: Comparing Scottie Go! with Scratch

Session 6: micro:bit in classroom

Duration: 3 hours (135 minutes)
Lecture: Presenting micro:bit programming and how it differs from Scratch; How to apply critical thinking using micro:bit in different school subjects
Demonstration: Creating simple examples for different school subjects with micro:bit
Group work: Creating basic micro:bit applications for different school subjects

Session 7: Designing learning scenarios

Duration: 3 hours (135 minutes)

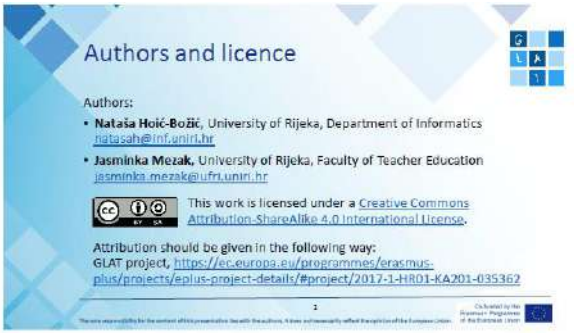
Individual work: Preparing learning scenarios using written form (developing the first version of the 3rd learning scenario based on IBL and Scratch/micro:bit educational game)
Group work: Review and discussion about the developed scenarios

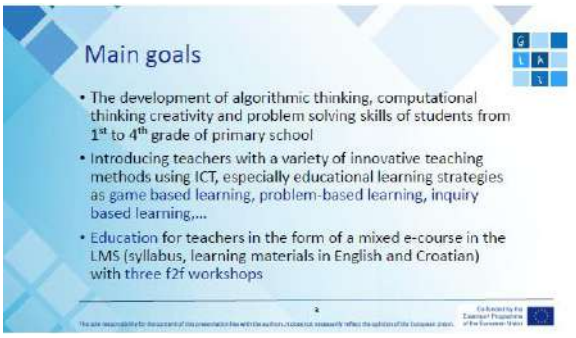
Conclusion of the Workshop 3

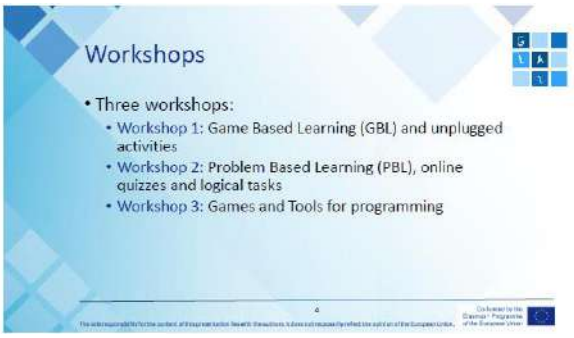
Duration: 1 hour (45 minutes)

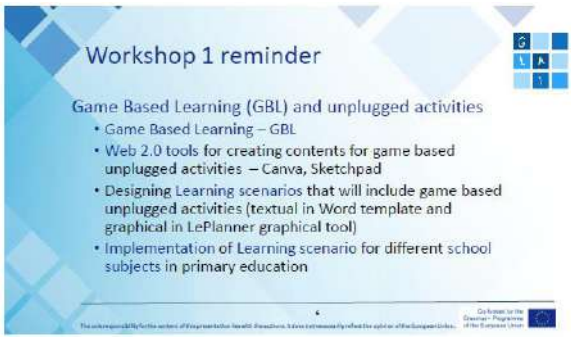
Whole-group activity: Debriefing
Closing talk: Introducing and explaining the follow-up activities (developing the 3rd learning scenario based on IBL and Scratch/micro:bit educational game)

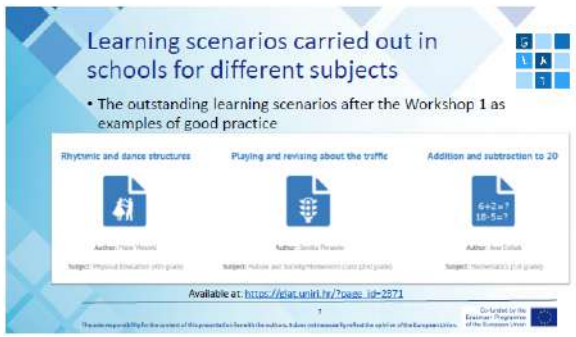
Presentation: Introduction to Workshop 3

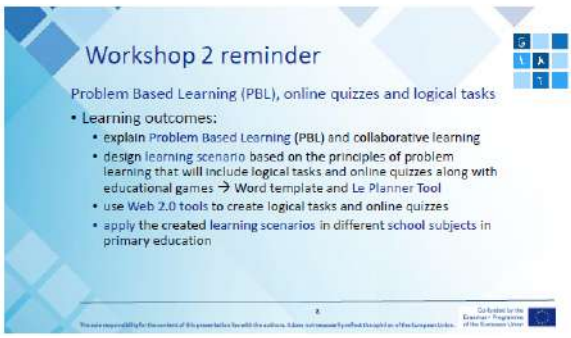












Learning scenarios carried out in schools for different subjects

- The outstanding learning scenarios after the Workshop 2 as examples of good practice

Addition to 5	Months of the year	Spatial orientation
1+3=? 2+1=?		

Available at: https://glat.uniri.hr/?page_id=2571

Games for Learning Algorithmic Thinking

What is next?

Workshop 3

Games and tools for programming

- Learning outcomes:
  - describe the principles of IBL
  - explain the basic programming concepts
  - use digital tools to create interactive educational games
  - develop learning scenarios that will include, along with educational games, concepts of programming and computational thinking for different school subjects in primary education
  - apply the created learning scenarios in different school subjects in primary education.

Workshop 3 – 1st day

- Introduction to Inquiry Based Learning
- Basic programming concepts
- Learning programming with games and stories
- Examples:
  - Run Marco (https://www.brainpop.com/games/runmarco/)
  - Blocky-games (https://blockly-games.appspot.com/)
  - Code.org (https://studio.code.org)
  - Scratch (https://scratch.mit.edu/)
- Creating stories and games with Scratch

Workshop 3 – 2nd day

- Introducing programming in primary education
- Examples:
  - ScottieGo (https://scottiego.com/en/)
  - micro:bit (https://microbit.org/hr/)
- Creating basic micro:bit applications for different school subjects
- Designing learning scenarios for IBL with programming tools

In the following workshop, pay attention to ... 1/2

- Computational and algorithmic thinking
  - incorporate them in learning scenarios, i.e. with the learning outcomes related to the subjects, include those that relate to these elements
- Learning scenarios
  - after the Workshop 3 the use of a single game example is required, but you are also encouraged to design and implement more examples
  - use Scratch or micro:bit for at least one example
  - use the principles of IBL and team work of students
  - problem solving elements (logic games, quizzes, ...) can also be included

In the following workshop, pay attention to ... 2/2

- Reflection on conducted activity
  - Teacher's review or reflection is very important.
  - Write more extensive reflections, especially about achieving the learning outcomes related to algorithmic and computational thinking
  - Indicate the number of students with whom the activity was performed.
  - Describe how your students have adopted these activities.
  - Describe the achievement of all learning outcomes in the scenario.
  - Describe the possible changes to the scenario before the next implementation.

Questions

The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Union.

Co-funded by the Erasmus+ Programme of the European Union

Workshop 3 – Games and Tools for Programming

Session 1: Inquiry Based Learning

Expected Learning Outcomes

- Describe principles of Inquiry Based Learning
- Explain steps in designing IBL activity (research question, exploring, presenting).
- Analyse and compare existing examples of using Inquiry Based Learning in different school subjects

Teaching Methods/Approaches

- Teacher presentation and demonstration
- Discussion
- Group activity - collaboration

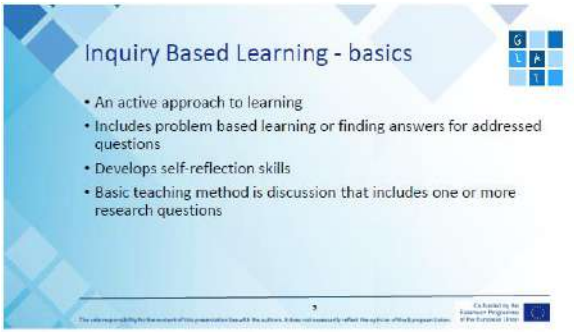
Sources of Training Materials

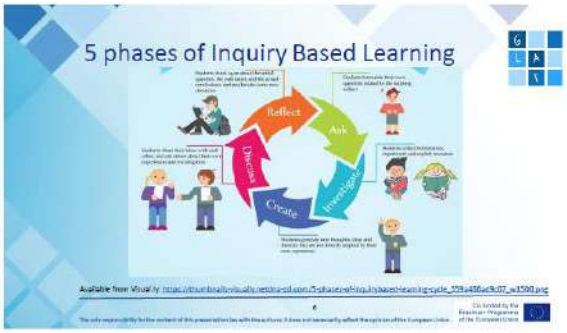
- 4 Phases of Inquiry Based Learning - A Guide For Teachers: https://www.teachthought.com/pedagogy/4-phases-inquiry-based-learning-guide-teachers/ (5.01.2019.)
- Inquiry Based Learning in the Science Classroom: https://www.edutopia.org/practice/inquiry-based-learning-science-classroom (5.01.2019.)
- What is Enquiry-Based Learning (EBL)?: http://www.ceebl.manchester.ac.uk/ebl/ (5.01.2019.)

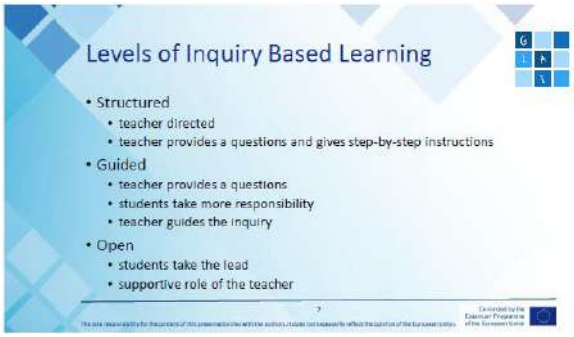
Duration: 1 hour (45 minutes)

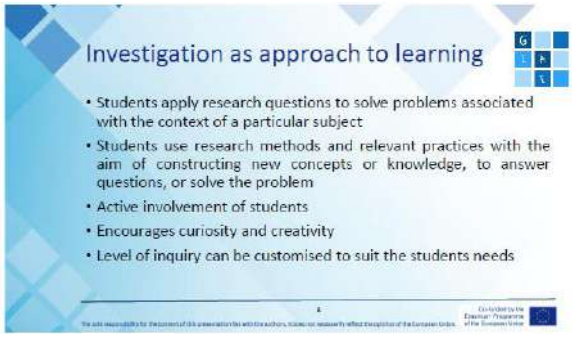
Topic/Sub-topics	Learning Objectives	Evaluation
1. INQUIRY BASED LEARNING	*Participants will be able to describe and explain the principles of Inquiry Based Learning, analyse and apply concepts of Inquiry Based Learning in different school subjects.*	Learners discuss concepts of Inquiry Based Learning and describe an example of lesson (group activity).
1.1. Introduction to Inquiry Based Learning	Describe principles of Inquiry Based Learning Apply concepts of Inquiry Based Learning	
1.2. Project Based Learning	Compare Inquiry Based Learning with Project Based Learning. Analyse and compare existing examples of using Inquiry Based Learning in different school subjects	

Presentation: Inquiry Based Learning









Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Project Based Learning

Project Based Learning

- A form of Inquiry Based Learning that involves independent student research
- Students are assigned a task based on facts, problems that need to be solved, or the goal to be achieved
- Based on the concept of collaborative learning
- Students develop problem-solving skills, critical thinking, communication and interpersonal skills, and take the responsibility for themselves and others

K-W-L strategy chart

KWL		
What I Know	What I Want to Know	What I Learned

Inquiry Based Teaching strategy

Dutt-Doner, K. and Grande, M. (2011): http://www3.canisius.edu/~grandem/catholicconference/imageinquiregif.gif

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Discussing example

Group activity

Example 1/2

Research questions:

- By collecting old paper, we preserve forests? Why?
- Can you calculate how much paper we need to recycle for saving one tree?

Example 2/2

Students will:

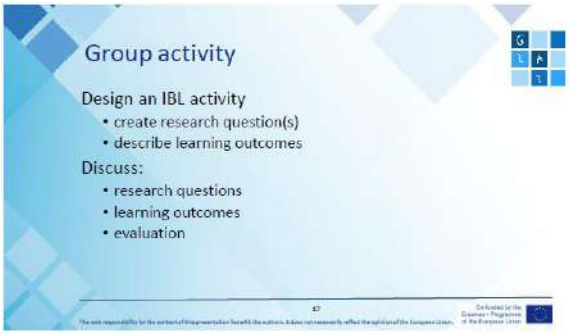
- Investigate and interpret the process of producing paper from wood (search and extract relevant information).
- Investigate and interpret the paper recycling process (search and extract essential information).
- Create a poster or graphically present both procedures (define algorithm or sequence of commands).
- Establish and implement a calculation (write down the steps in the calculation process).

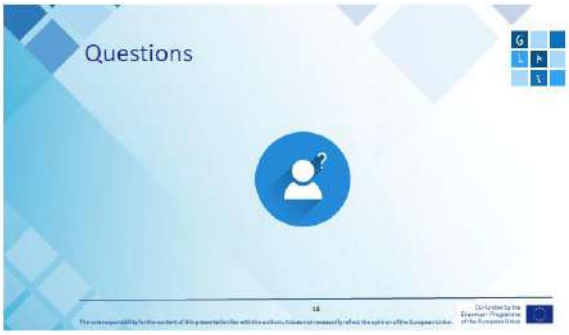
Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Designing IBL activity

Group activity





Workshop 3 – Games and Tools for Programming

Session 2: Basic programming concepts

Expected Learning Outcomes

- Identify the basic programming concepts
- Recognise the basic programming concepts in examples of different educational games
- Analyse and compare existing examples of computer games for learning programming

Teaching Methods/Approaches

- Teacher presentation and demonstration
- Discussion
- Group activity - collaboration

Sources of Training Materials

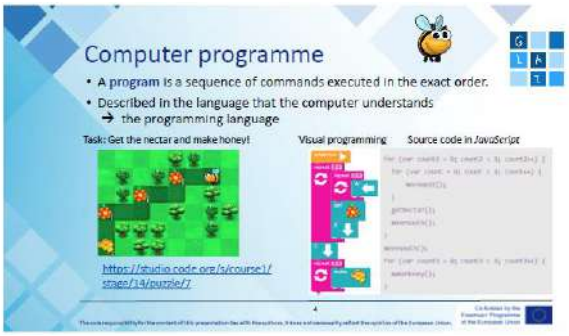
Games:

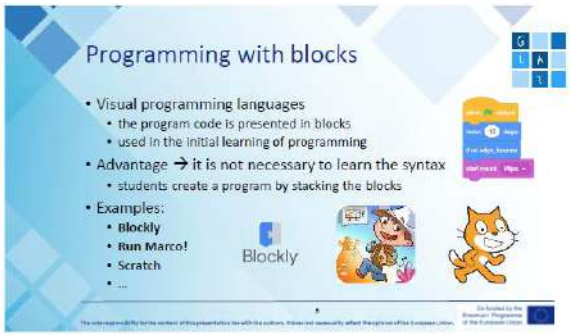
- Run Marco: https://runmarco.allcancode.com/ (5.1.2019.)
- Blockly-games: https://blockly-games.appspot.com/?lang=en (5.1.2019.)
- Code.org: https://studio.code.org/ (5.1.2019.)

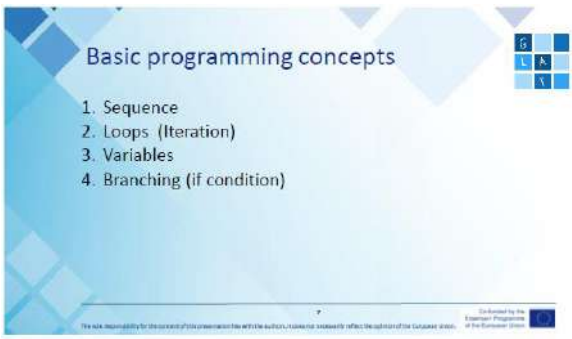
Duration: 1 hour (45 minutes)

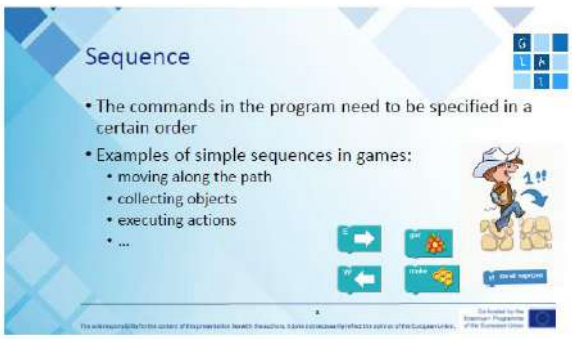
Topic/Sub-topics	Learning Objectives	Evaluation
1. BASIC PROGRAMMING CONCEPTS	*Participants will be able to identify the basic programming concepts.*	Learners explore existing educational games for learning programming, point out and discuss usage of programming concepts (group activity).
1.1. Introduction to basic programming concepts	Describe the basic programming concepts (sequence, branching, loop, variables)	
1.2. Educational computer games for learning basic programming concepts	Recognise the basic programming concepts in examples of different educational games Analyse and compare existing examples	

Presentation: Basic programming concepts









Games for Learning Algorithmic Thinking

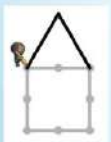
Co-funded by the Erasmus+ Programme of the European Union

Analysing examples

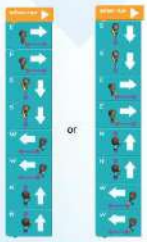
Group activity

Sequence – Example 1

Code.org: Artist



or



https://studio.code.org/s/course1/stage/10/puzzle/1

Sequence – Example 2

Code.org: Bee

https://studio.code.org/s/course1/stage/7/puzzle/3

Loop

- The loop is a construct that causes a group of one or more commands to be invoked repeatedly until some end condition is met:
  1. number of repetitions is known in advance
  2. number of repetitions is not known
- Examples of using loops in games:
  - moving characters
  - performing the action
  - ...

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Analysing examples

Group activity

Loop - Example 1

Run Marco! Level 11

https://www.brainpop.com/games/runmarco/

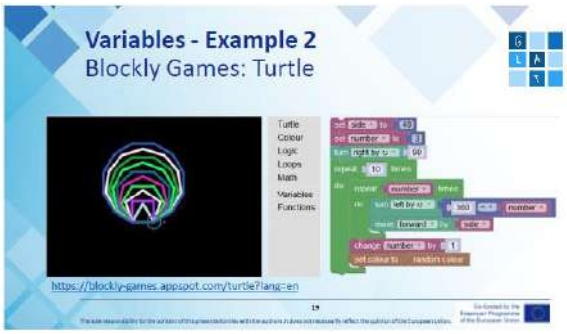
Loop - Example 2

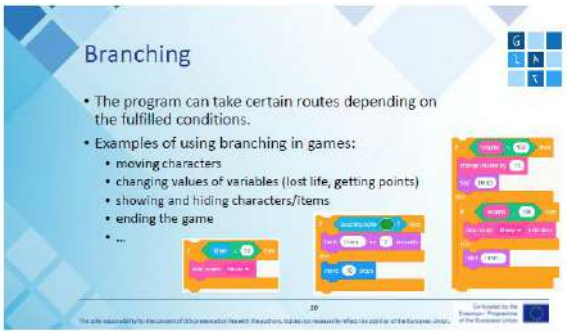
Code.org: Code with Anna and Elsa

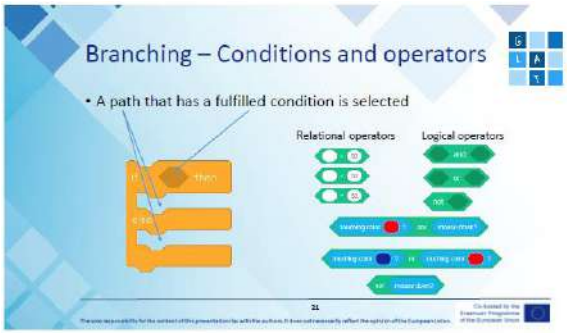
https://studio.code.org/s/frozen/stage/1/puzzle/5

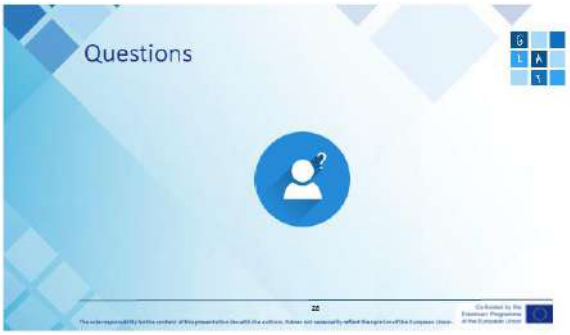
Variables

- Variables are used to keep values (text, numbers,...) and reuse these values.
- Variables have **name** and **value** (for most programming languages also the **type**)
- Examples of using variables in games:
  - text or sound - what will a character say
  - values used for moving characters (number of steps, turns,...)
  - number of lives, collected items, points,...
  - time left to finish the game
  - ...









Workshop 3 – Games and Tools for Programming

Session 3: Learning programming with games and stories

Expected Learning Outcomes

- Recognise the meaning of Computational Thinking (concepts, practices, perspective) development
- Understand the role of Scratch community and the process of creation in the Scratch community
- Find, analyse and compare different examples of games and digital stories in Scratch
- Change and remix a story/game

Teaching Methods/Approaches

- Teacher presentation and demonstration
- Discussion
- Individual activity
- Group activity - collaboration

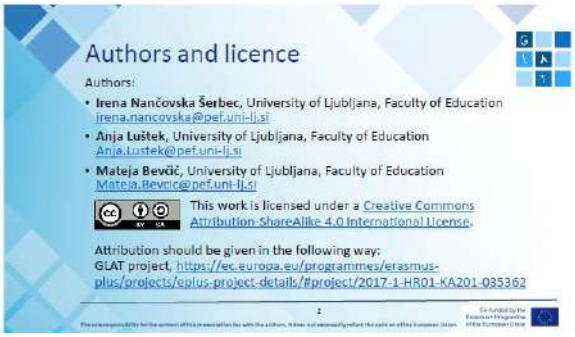
Sources of Training Materials

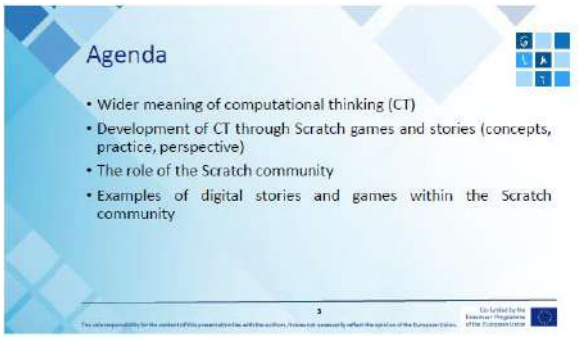
- ScratchEd teaching resources: http://scratched.gse.harvard.edu/resources/all (4.1.2019.)
- Computational Thinking with Scratch-developing fluency with computational concepts, practices and perspectives: http://scratched.gse.harvard.edu/ct/defining.html (4.1.2019.)
- Brennan, K. A. (2013). Best of both worlds: Issues of structure and agency in computational creation, in and out of school (Doctoral dissertation, Massachusetts Institute of Technology), http://hdl.handle.net/1721.1/79157 (4.1.2019.)
- Brennan, K., Balch, C., Chung, M. (2014). *Creative Computing*. Harvard Graduate School of Education. Retrieved from http://scratched.gse.harvard.edu/guide/files/CreativeComputing20140806.pdf (4.1.2019.)
- Brennan, K. (2015). Beyond right or wrong: Challenges of including creative design activities in the classroom. *Journal of Technology and Teacher Education*, 23(3), 279-299. Waynesville, NC USA: Society for Information Technology & Teacher Education, https://www.learntechlib.org/primary/p/151249/ (4.1.2019.)
- CS First, https://csfirst.withgoogle.com/en/home (4.1.2019.)

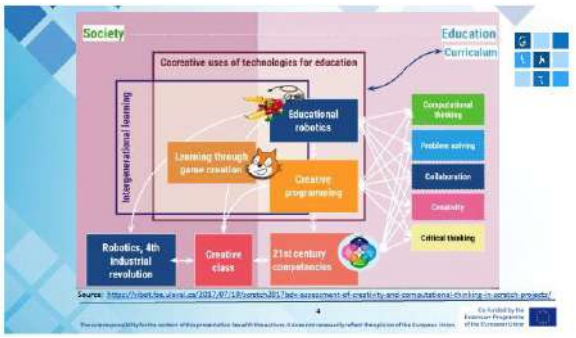
Duration: 2 hours (90 minutes)

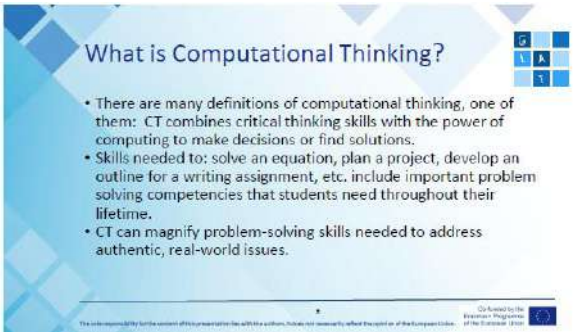
Topic/Sub-topics	Learning Objectives	Evaluation
1. COMPUTATIONAL THINKING (CT)	*Participants will be able to recognise the meaning of CT and to identify its concepts, practices, and perspectives.*	Learners give examples of computational thinking development from their practice and describe the computational thinking dimensions: concepts, practices, and perspective.
1.1. Introduction to Computational Thinking concepts	Describe the meaning of CT	
1.2. Practices and perspectives of CT	Identify the concepts, practices, and perspectives of CT development	
2. COMPUTATIONAL THINKING DEVELOPMENT WITH SCRATCH	*Participants will be able to understand the role of Scratch community and identify existing digital stories and games for the development of CT.*	Learners will search Scratch projects (games and stories) with own keywords, "run" the game and explain some functionalities, remix games and stories. Learners will explore and analyse others' projects of stories/games in Scratch. Learners will create a studio, add a project and think how to "unstuck" while developing Scratch projects with the support of community.
2.1. Scratch community	Explore the Scratch community and the process of creation in the Scratch community	
2.2. Scratch for creating games and stories	Change and remix existing digital stories and games in Scratch for different didactic purposes	
2.3. Workshops for developing games and stories	Analyse the presence of computational thinking concepts in the Scratch projects (stories and games) and workshops.	

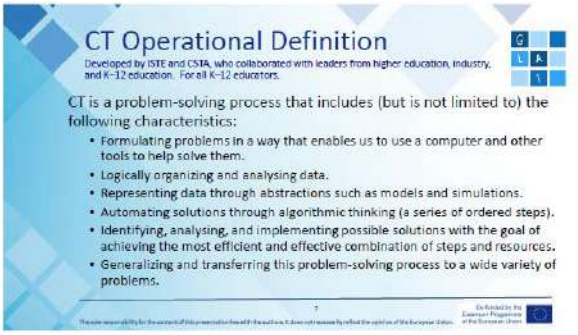
Presentation: Learning programming with games and stories

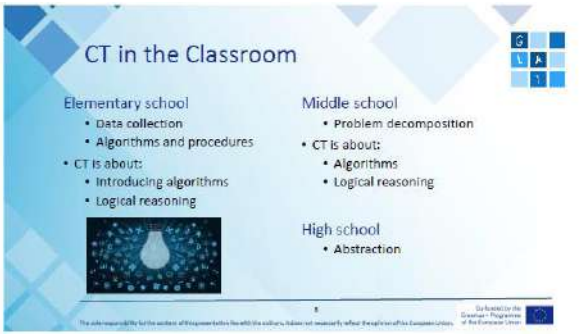












Computational thinking

Games for Learning Algorithmic Thinking

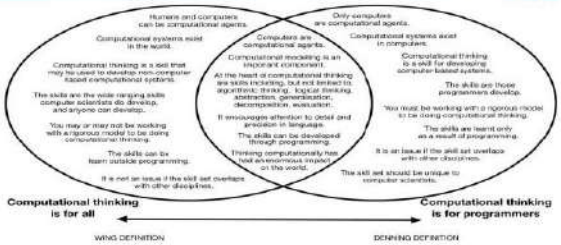
Explaining CT

Pair activity

Examples of CT

- Do you have idea what is CT? – explain to your pair
- Design activity (individually)
  - Context of activity (age group of students, subject, activity)
  - What concepts and approaches of CT is developing the activities?
  - Linking concepts and approaches of CT with the learning objectives of the selected subject.
- Present activity to your pair

Agreement and disagreement around what CT should be



Development of CT – Scratch

- Students develop CT through interactive stories, games and animations in Scratch.
- Scratch - visual programing language - educational programing environment
  - MIT Media Lab, 2002 – first prototype
  - 2007 public launch
  - translated in 70+ languages
  - Scratch 3 (January 2019).

Scratch - teachers point 1/2

Papert (in '80) argued that programming languages should have:

- a "low floor" (easy to get started)
- a "high ceiling"(opportunities to create increasingly complex projects over time).
- "wide walls" (supporting many different types of projects so people with many different interests and learning styles can all become engaged).

Scratch - teachers point 2/2

What is easy/possible in Scratch?

- start
- involve students with different knowledge
- a lot of books, support of community
- What else?

What is not easy?

- assessment
- to much sources
- "Children know more than teachers"
- What else?

How to start with Scratch?

Guided start:

- https://resources.scratch.mit.edu/www/guides/en/Getting-Started-Guide-Scratch2.pdf
- https://scratch.mit.edu/tips + Activity cards
- Scratch Wiki
- tutorials at Code Club and CS First

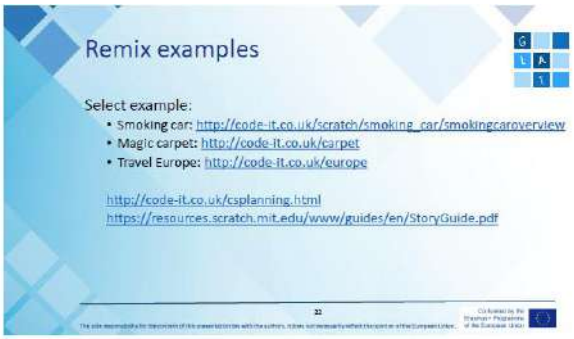
ScratchEd - lot of support, resources:

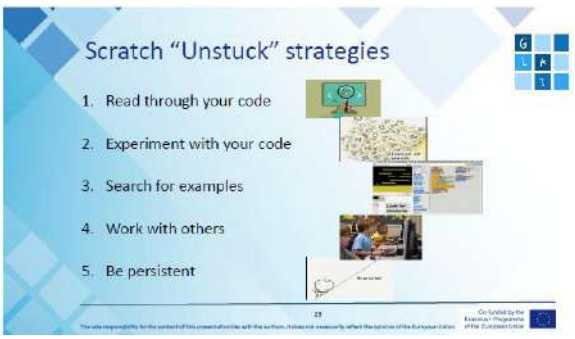
- http://scratched.gse.harvard.edu/resources/all

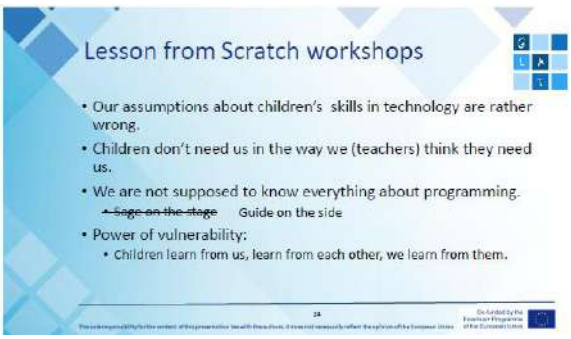
Creative computing curriculum: http://scratched.gse.harvard.edu/guide/

The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Union.

Co-funded by the Erasmus+ Programme of the European Union







CT, Brennan and Resnick (2012) 1/2

- CT through concepts, practices and perspectives

Concepts:

- **sequence:** identifying a series of steps for a task
- **loops:** running the same sequence multiple times
- **parallelism:** making things happen at the same time
- **events:** one thing causing another thing to happen
- **conditionals:** making decisions based on conditions (branching)
- **operators:** support for mathematical and logical expressions
- **data:** storing, retrieving, and updating values.

CT, Brennan and Resnick (2012) 2/2

Practices - elements of designers' learning and participation

- **experimenting and iterating:** developing little bit, step by step
- **testing and debugging:** making sure things work
- **reusing and remixing:** building on existing projects or ideas
- **abstracting and modularizing:** exploring connections between the whole and the parts.

Perspectives – metacognition, understandings of themselves

- **expressing:** computation is a medium of creation, "I can create."
- **connecting:** recognizing the power of creating with and for others, "I can do different things when I have access to others."
- **questioning:** feeling empowered to ask questions about the world, "I can (use computation to) ask and make sense of computational world."

Workshops

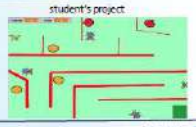
- Workshops for younger children – between 6-8 years old
- ScratchJr:
  - Take me for a walk
  - Space exploration
  - Magic in Scratch
- Based on:
  - Moving object on the graph paper
  - Games
  - Tablets
  - Tangible computing (bot Sphero)
  - Physical computing (Arduino)

Workshop - Maze game

- Workshop and academic project work description
- Summer School 17 students (age 8-10), 25 hours = 5 day x 5 hours https://scratch.mit.edu/studios/3491552/
- Students of pedagogic CS: academic project work 7 students (age >= 19), students' studio: https://scratch.mit.edu/studios/677448/

9 years olds' child project

student's project



Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Search Mazes

Individual activity

Explore Mazes

- Login to Scratch
- Search for Maze projects
- Select 2-3 "good" mazes
- Look inside the code
- Try to remix, create studio and add a project to your studio

Workshop - Escape Room game

- Workshops were organized for 2 different groups:
- Pupils: Summer School 8 student (age 9-11 children) 25 hours (5 days x 5 hours)
  - Studio: https://scratch.mit.edu/studios/4136527/
  - Activities: https://youtu.be/UBEojBWJa4
- Students of pedagogic CS (Faculty of Education, University of Ljubljana): academic project work 11 students (age >= 19)

9-10 years olds' child project

student's project

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Search Escape Room

Individual activity

Explore Escape Rooms

- Search for Escape Room projects.
- Select 1-2 "good" Escape Rooms.
- Look inside the code.
- Why is this project good?
- Add to your studio (remix).

Connecting Scratch with natural sciences

- TEALEAF project site: https://sites.google.com/a/tealeaf-project.eu/tealeaf-project/
- Topics: biodiversity, ecosystems, invasive species
  - Bart beetle games:
  - https://scratch.mit.edu/studios/3748271/

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Scratch workshop ideas

Group activity

Workshop - My Scratch day

Think and discuss about about topic of interest in your group.

- What is **your idea for workshop** in Scratch?

Explain to other groups:

- Why do you select this topic?
- How old children would you like to include?
- What do you expect?
- Exchange and discuss ideas with other groups.

Conclusions

- Computer science = problem solving
- Teaching coding is not related to ICT use
- Getting these message through is difficult
- We need educated teachers
- Programming games in Scratch is fun!

Questions

Workshop 3 – Games and Tools for Programming

Session 4: Introduction into visual programming with Scratch

Expected Learning Outcomes

- Understand the concept of computational creation in the context of Scratch
- Find and analyse different possibilities for own Scratch-based computational creation
- Become familiar with resources that support computational creation
- Establish Scratch accounts and create Scratch projects (stories)

Teaching Methods/Approaches

- Teacher presentation and demonstration
- Individual activity
- Group activity - collaboration
- Peer evaluation

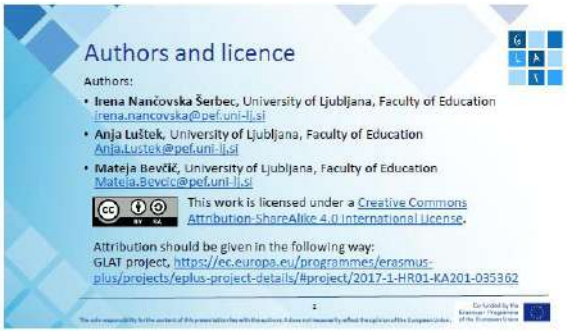
Sources of Training Materials

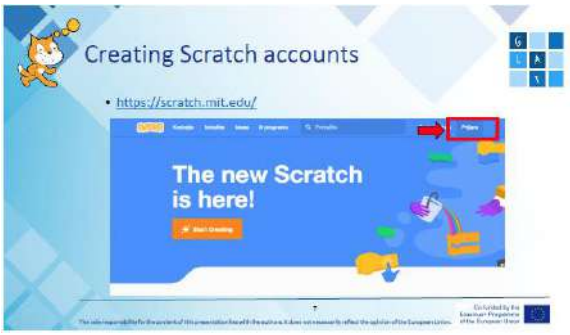
- Brennan, K., Balch, C., Chung, M. (2014). *Creative Computing*. Harvard Graduate School of Education. Retrieved from http://scratched.gse.harvard.edu/guide/files/CreativeComputing20140806.pdf (5.1.2019.)

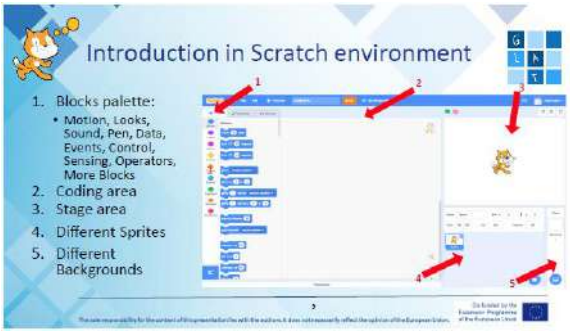
Duration: 3 hours (135 minutes)

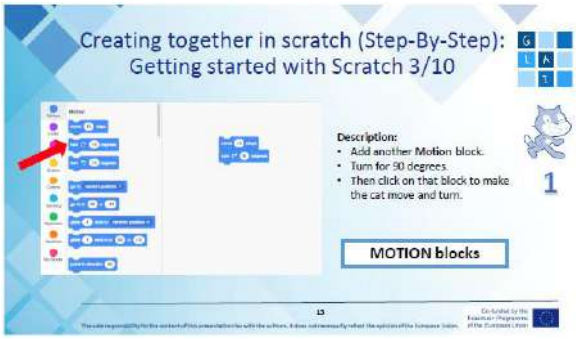
Topic/Sub-topics	Learning Objectives	Evaluation
1. INTRODUCTION INTO VISUAL PROGRAMMING WITH SCRATCH	*Participants will be able to understand the concept of computational creation in the context of Scratch and to imagine possibilities for their own Scratch-based computational creation.*	Learners (in pairs) inspect prepared stories/games in Scratch and comment them. Learners explore different parts of the Scratch interface (drag and drop blocks), experiment by clicking on each block to see what happens, snapping blocks together etc.
1.1. Testing Scratch examples	Test already prepared examples	
1.2. Short introduction to Scratch elements	Explore how Scratch works - how to start, where and what are blocks, how to move blocks	
2. CREATING PROJECTS IN SCRATCH	*Participants will be able to create some projects with the help of lecturers' instructions and recognize important programming concepts through different activities.*	Learners with the help of teachers create few examples in Scratch. Learners create their own simple project – story in Scratch.
2.1. Creating a presentation-story	Create a presentation-story in Scratch	

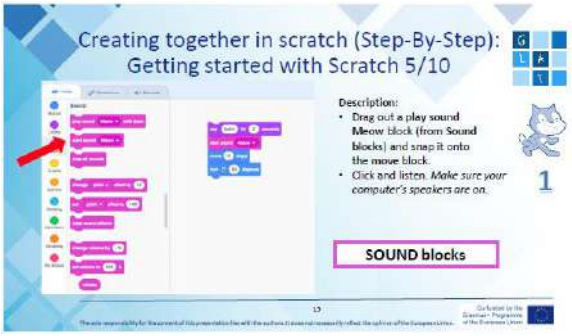
Presentation: Introduction into visual programming with Scratch

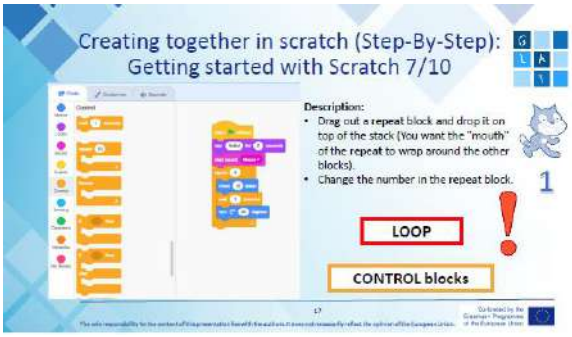


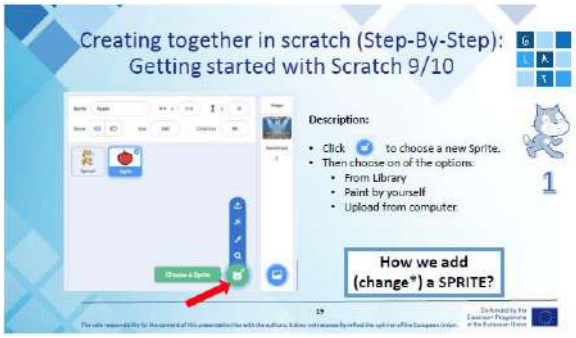


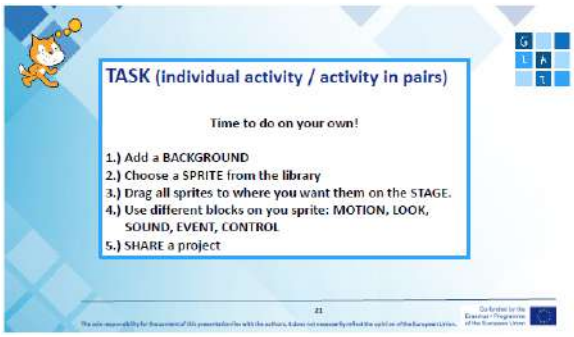


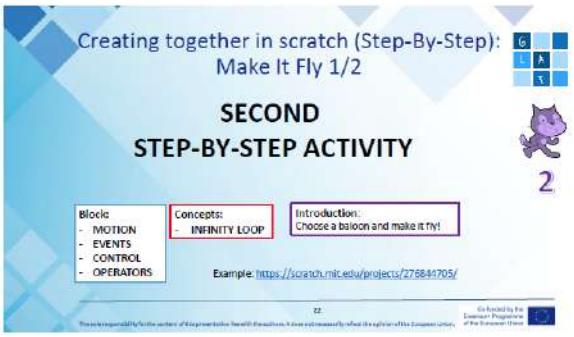


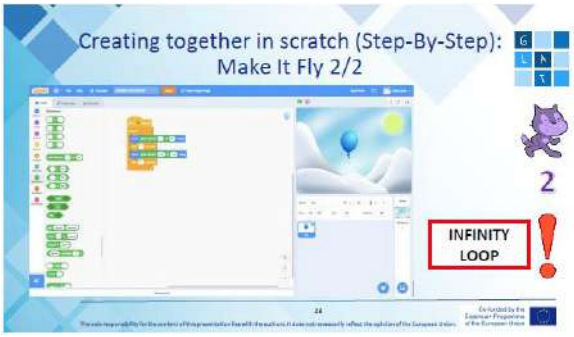


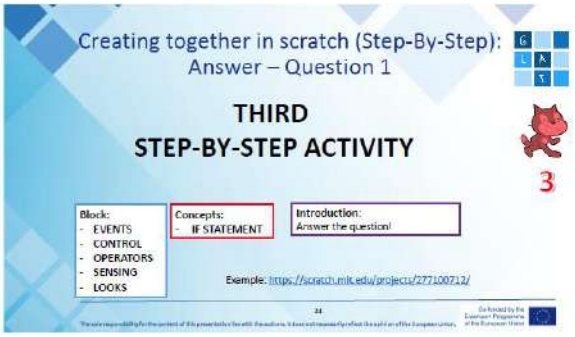


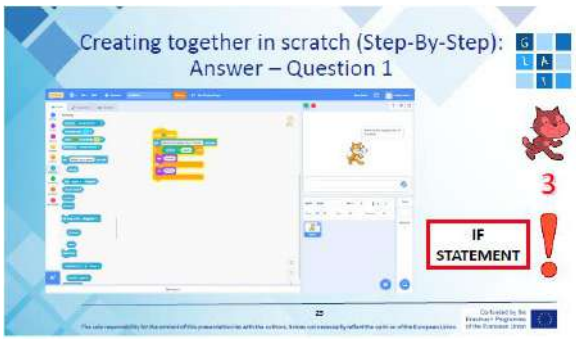


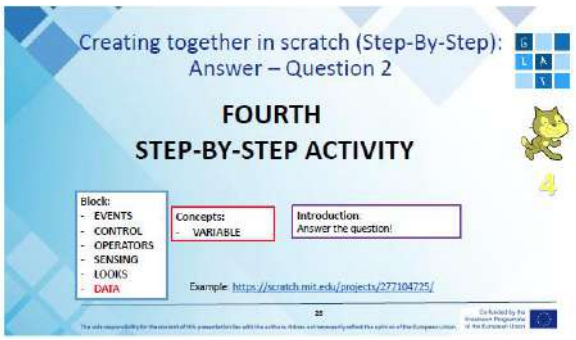


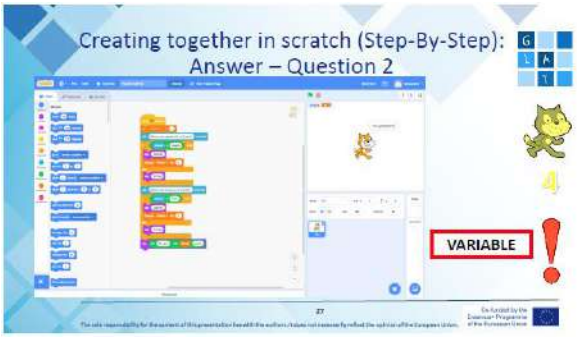


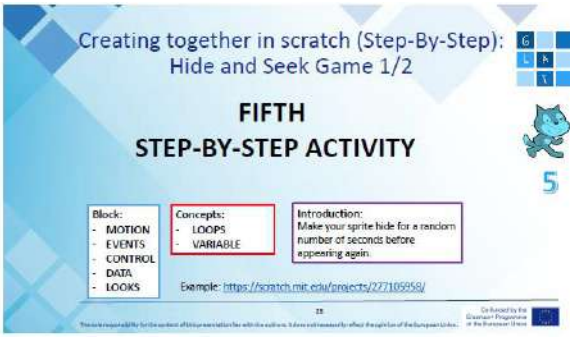


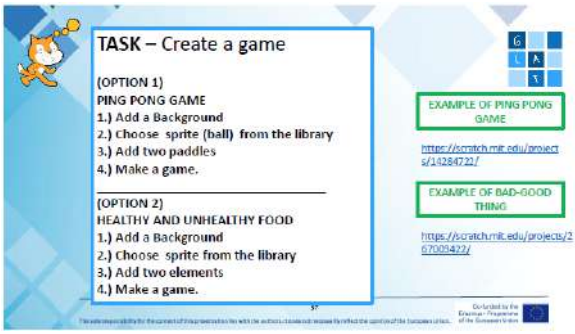


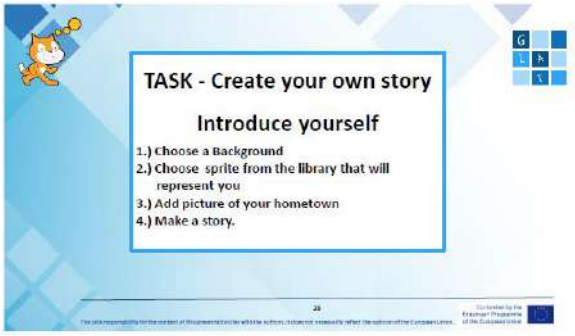


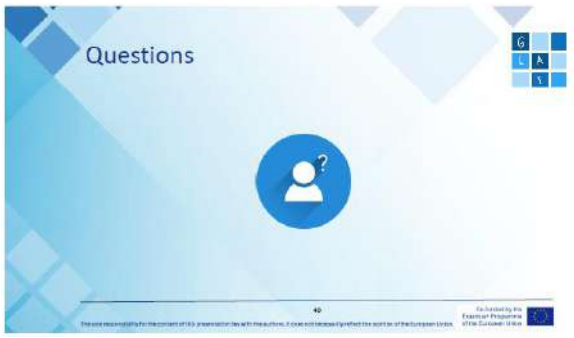












Workshop 3 – Games and Tools for Programming

Session 5: Implementing Computational Thinking and programming with GBL tools

Expected Learning Outcomes

- Understand the elements and process of computational thinking from teacher perspective
- Compare computational thinking with programming
- Being able to introduce game based learning tools with elements of coding in the classroom

Teaching Methods/Approaches

- Teacher presentation and demonstration
- Discussion
- Group activity - collaboration

Sources of Training Materials

- Computational Thinking: https://code.org/curriculum/course3/1/Teacher (4.1.2019.)
- Scottie Go! for Computational Thinking https://www.youtube.com/watch?v=hXZOGFaI6vc&t=16s (4.1.2019.)

Duration: 1 hour (45 minutes)

Topic/Sub-topics	Learning Objectives	Evaluation
1. HOW TEACHERS CAN INTRODUCE PROGRAMMING IN THE CLASSROOM FROM THEIR PERSPECTIVE	*Participants will be able to recognize the elements and the process of computational thinking and programming.*	Learners explore and analyse applications of GBL tools within the class in order to point out benefits of introduction of computational thinking and programming.
1.1. Cycles for learning about how to code with focus on computational thinking	Explore the concepts of learning programming from teacher's perspective	
2. USING GAME BASED LEARNING TOOLS WITH ELEMENTS OF CODING IN THE CLASSROOM	*Participants will be able to recognize the importance and the concept of collaborative games with coding.*	
2.1. Video presentation and discussion of game based learning tool Scottie Go!	Understand the role of GBL tools for coding and in development of computational thinking Introduce a way to incorporate technology and digital tools in engaged way	

Presentation: Implementing Computational Thinking and programming with GBL tools

Games for Learning Algorithmic Thinking

Co-funded by the Erasmus+ Programme of the European Union

Workshop 3: Games and tools for programming

Session 5: Implementing computational thinking and programming with GBL tools

Authors and licence

Authors:

- **Vladimir Trajkovik**, Ss. Cyril and Methodius University in Skopje, Faculty of Computer Science and Engineering
trvlado@finki.ukim.mk

Agenda

- How to introduce programming in your classroom from teachers perspective
- Introducing GBL tools with elements of coding in the classroom
  - Scottie Go!

„Whether you want to uncover the secrets of the universe, or you just want to pursue a career in the 21st century, basic computer programming is an essential skill to learn."

Stephen Hawking

Learning how to code

Encountered problems

- Teachers neglected
- Focus on results, not knowledge
- Trivial pedagogical approach

Disadvantages

- No real teacher training
- No digital competences guidelines for coding
- Risk to create opposite effect on pupils

Learning progamming in primary school

Approaches:

- Teacher lead and information oriented

OR

- Students engaged and skills development oriented

Teacher role:

- Educator that leads the class from the classroom front

OR

- Facilitator who encourages the class to think and question the world around students

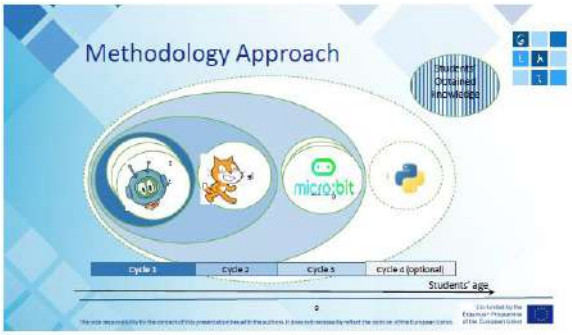
Possible solutions

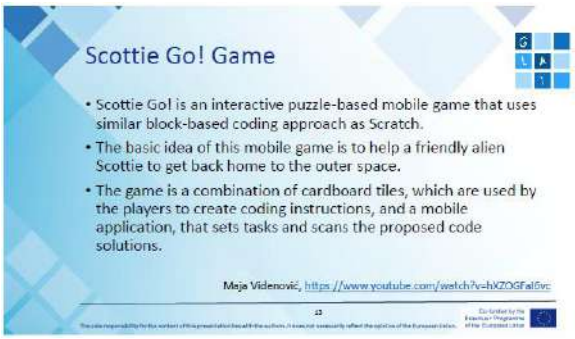
- Game Based Learning
- Flipped Classroom
- Cooperative and individual Cycles

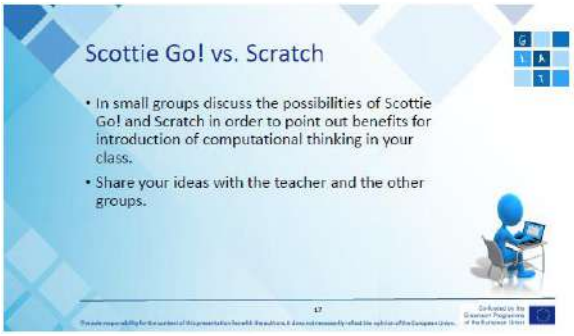
How to make a change?

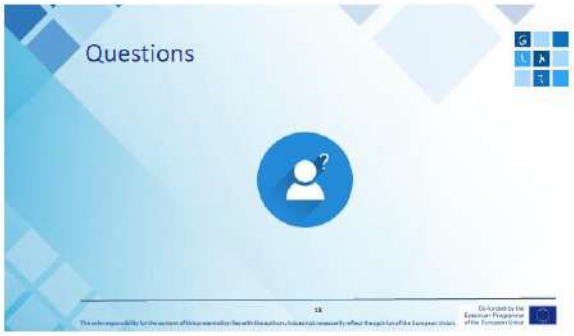
- Cooperative games, no technology
  - Focus on team work and cooperation to achieve educational goal
- Individual time to master at kid's own pace...
  - Focus on technology
- Team work, create something new, gamification...
  - Gamification is a process that introduces elements of games (e.g. competition among teams)

Remember: You have „superheroes" in your class...









Workshop 3 – Games and Tools for Programming

Session 6: micro:bit in classroom

Expected Learning Outcomes

- Recognize the elements and process of using micro:bit
- Compare micro:bit with Scratch
- Be able to develop activities using micro:bit

Teaching Methods/Approaches

- Teacher presentation and demonstration
- Individual activity
- Group activity - collaboration

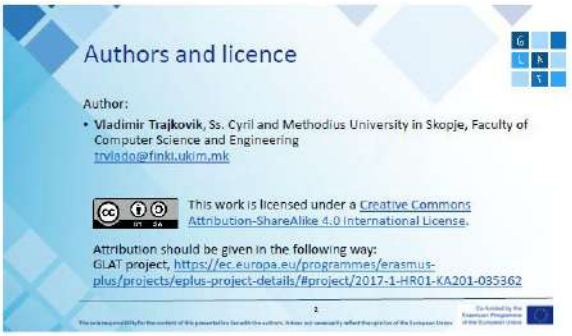
Sources of Training Materials

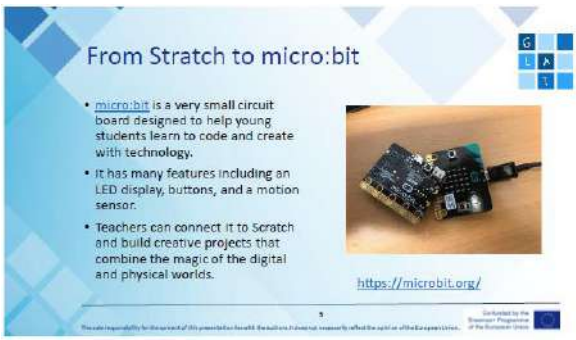
- Computational Thinking: https://code.org/curriculum/course3/1/Teacher (4.1.2019.)
- micro:bit: https://microbit.org/hr/ideas/ (4.1.2019.)
- BBC micro:bit edukacijski materijali: http://izradi.croatianmakers.hr/bbc-microbit-uvodna-stranica/ (4.1.2019.)

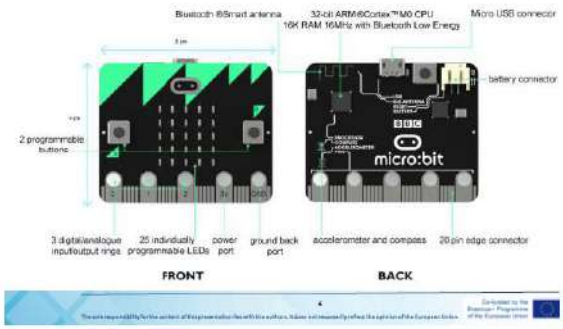
Duration: 3 hours (135 minutes)

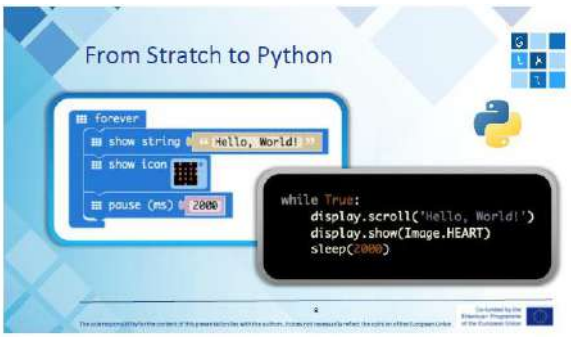
Topic/Sub-topics	Learning Objectives	Evaluation
1. INTRODUCTION OF MICRO:BIT	*Participants will be able to describe the functionalities and features of micro:bit and recognize and compare basic micro:bit applications with Scratch.*	Learners explore and analyse simple micro:bit applications and compare it with Scratch projects.
1.1. Introduction of micro:bit as tool for programming (basic concepts, how it differs from Scratch)	Explore the functionalities and features of micro:bit, micro:bit development environment, and basic event driven programming	
2. HOW TO APPLY MICRO:BIT IN DIFFERENT SCHOOL SUBJECTS	*Participants will be able to introduce basic micro:bit applications in their classroom.*	Learners explore and analyse micro:bit projects and explore possibility to apply them in their classes for active participation of their students (group activity).
2.1. Demonstration of using simple micro:bit application for different school subjects	Analyse existing applications suitable for different subjects Analyse examples of project-based learning using micro:bit	
2.2. Developing and adopting micro:bit application for different school subjects	Be able to alter micro:bit code in order to better match learning outcome	

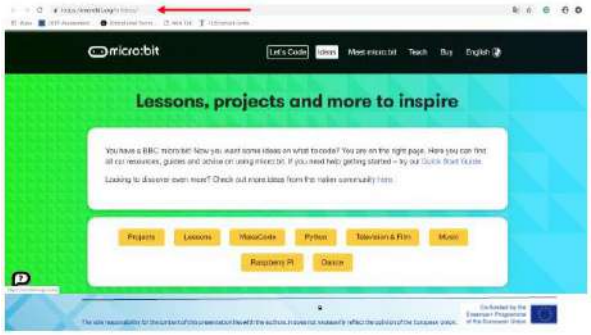
Presentation: micro:bit in classroom

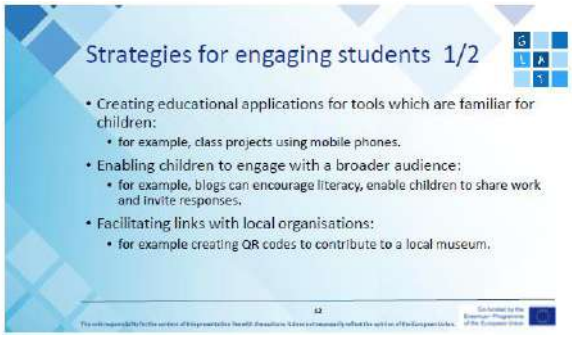


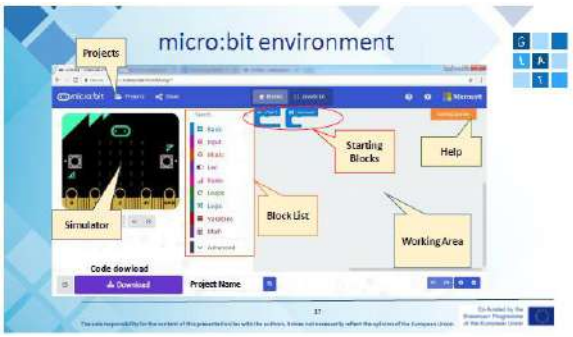


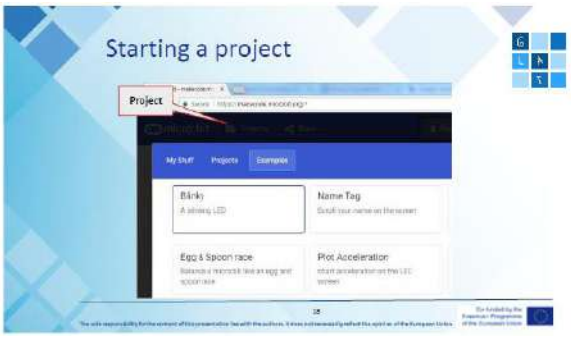


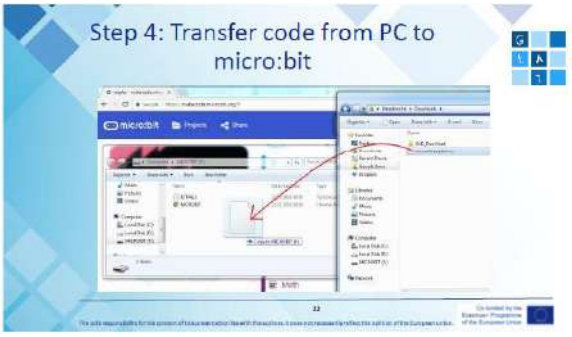


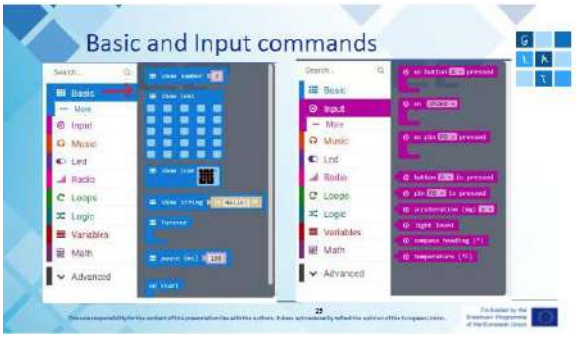


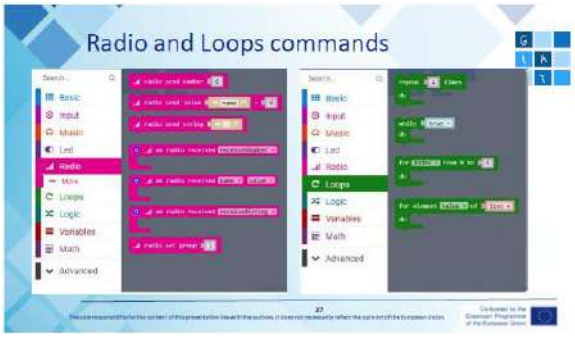


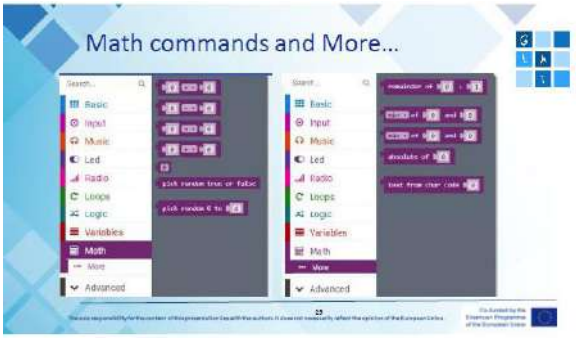


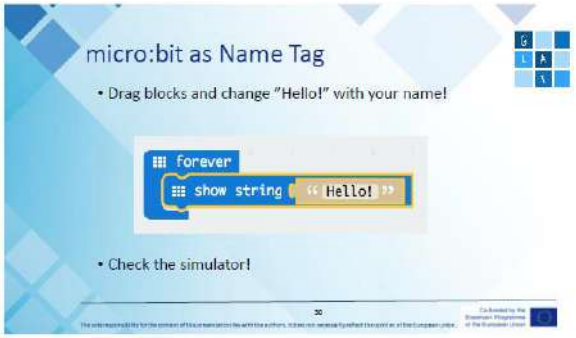


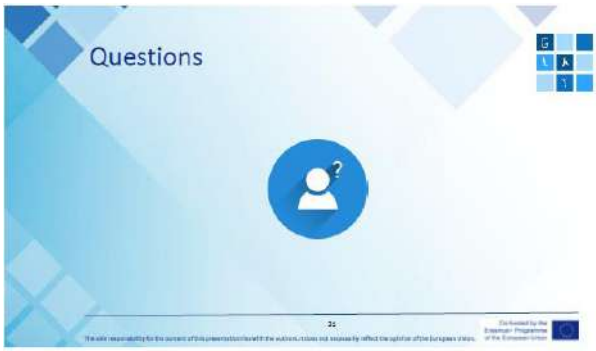












Workshop 3 – Games and Tools for Programming

Session 7: Designing learning scenarios

Instructions for the participants

Expected Learning Outcomes

- Create learning scenarios that will include, along with educational games, concepts of programming and computational thinking for different school subjects in primary education
- Apply the created learning scenarios in different school subjects in primary education with the students from 1st to 4th grades

Individual Assignment:

Your task is to prepare the learning scenario based on IBL and Scratch/micro:bit educational game in written form and in graphical form using LePlanner. You could choose any school subject and any lesson within the subject for students from your class, considering that the activity should be completed in three months.

This is the **first version of the 3rd learning scenario** which you will continue to design with the online help of your mentor.

In this scenario the use of at least one story or game example developed in Scratch or micro:bit is required. It is not necessary to develop the game by yourself. Your task is to fill in detailed **Game/story template** with the help of your students as a follow-up activity.

Completed version of the game description and learning scenario will be **reviewed** by the mentor. According to the descriptions in the Game/story template, mentor will organize the development of the game (e.g. help will be provided by the teachers or students of Informatics). The final refined versions you will **implement** in the classrooms with your students. Last step is to play the final version of the game together with your students.

You are also supposed to write the **reflection** on conducted activities.

Duration: up to 3 months for the whole assignment (including the development of story/game in Scratch)

	ASSIGNMENT STEPS
23.	Choose a school subject – plan the activities that will be carried out in your class next month.
24.	Use the **Learning Scenario Template** form (Annex 1) for textual version and LePlanner for graphical version of your scenario.
25.	Specify the **Learning outcomes**: - state ***general learning outcomes*** related to the course that will include problem teaching and logical tasks - state ***learning outcomes oriented on algorithmic thinking***
26.	Describe **Aim and tasks** of teaching and give a **Short description of activities**. Plan the activities that will encourage your students for seeking the information, critical and logical thinking as well as collaborating while solving the problem according to the principles of inquiry based learning (IBL). The activities should include a game on computer/tablet/smartphone (not only unplugged activities).

The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Union.

Co-funded by the Erasmus+ Programme of the European Union

27.	Specify the **Keywords**, **Correlation and Interdisciplinarity** with other courses or topics, and **Duration of activities**.
28.	Point out **Learning and teaching strategies and methods**. Specify the **Teaching forms**: use the principles of IBL and team work of students. Problem solving elements (logic games, quizzes, ...) can also be included.
29.	Choose **Tools** and games that will be used on computer/tablet/smartphone for at least one example. Mandatory is to use of at least one story or game example developed in Scratch or micro:bit. Point out all **Resources/materials** which will be required for the teacher as well as for students.
30.	Use **Game/story template** (Annex 2) to prepare the description of the story or game. For now, prepare just a draft (fill in elements: Title of the game, Type (Scratch or micro:bit), Course/ Grade, Learning outcomes, Goal of the game). Pay attention to the copyright for images, videos, and other materials collected from the web. Photographing your students requires written parents' consent.
31.	Elaborate the **Teaching summary** as **Motivation (Introduction)**, **Implementation** and **Evaluation (Reflection)**. This part describes in detail previously mentioned short description of activities. It should be based on IBL and activities with students for designing a story/game.
32.	In **Annexes** box provide a link to the graphical version of the learning scenario in LePlanner. You will add link to the developed online story later.
33.	**Examples and game references** box should contain a link to the Scratch story and to the other sources you will use for the activities.
	FOLLOW-UP ACTIVITIES
9.	Upload your completed first versions of learning scenario and draft of the game/story description to the Moodle e-course. Mentor will review and correct your scenario and story description.
10.	Upload your final version of learning scenario with story description considering mentor's suggestions and corrections.
11.	After mentor's approval, implement the part of learning scenario about the story development in the class and design game with your students. Complete the game/story description and upload it in the Moodle e-course. Mentor will provide you with the finished story/game for your learning scenario.
12.	Implement the last part (**Reflection and evaluation**) of your learning scenario in the class with your students and play the final version of the game together with them.
13.	Post a **reflection** on conducted activities in the forum: • Write a more extensive description on the implementation of the activity in your class. • Describe how your students have accepted learning activities, point out the parts about designing the game and playing the game. • Describe the achievement of all planed learning outcomes, both general and oriented on algorithmic thinking. • Define what you would like to change before the next implementation of the scenario.

The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Union.

Co-funded by the Erasmus+ Programme of the European Union

Part III:
Annexes

Annex 1: Learning scenario template

Learning Scenario Title		
Course/Grade		
Learning Outcomes	***General learning outcomes*** ***Specific LO oriented on algorithmic thinking***	
Aim, Tasks and Short Description of Activities		
Keywords		
Correlation and Interdisciplinarity		
Duration of Activities		
Learning and Teaching Strategy and Methods		
Teaching Forms		
Tools		
Resources/Materials for the Teacher		
Resources/Materials for the Students		
Teaching summary	Motivation-Introduction	Duration
	Implementation	
	Reflection and evaluation	

Annexes	
Examples and game references	

Annex 2: Game/story scenario template

Title of the game	
Type *(Scratch or micro:bit)*	
Course/ Grade	
Learning outcomes	
Goal of the game	
Characters and their roles	
Description of the game flow	
List of scenes / backgrounds	
Logical tasks within the story *(Note: select tasks that are aligned with your learning outcomes)*	
End of the game	

Appendix - Instructions for Storytelling in Scratch

The basic idea is to encourage students' algorithmic and computational thinking by including them as much as possible in designing the game/story, rather than just playing/reading it when finished.

Using Scratch, the whole story, which should have at least one logic game, will be designed. This game is used to direct the flow of the story according to the "if ... then ... else" principle as one of the algorithmic thinking concepts we would like to encourage in students.

The story will be designed together with the students, the amount of their participation will depend on their age. You should estimate how much help has to be provided to your students in this process.

For example, with the students you can design: **characters** (who will be the main character, who will be supporting characters, what they will look like, what they will do in the game,…), **scenes** (how many, what will they represent, what objects will be placed on them,…), the **goal and flow of the game** (what we want the main character in the game to do and achieve during the game), the **text** (written on the scene, or in the "bubbles", ...), **logical tasks** ("obstacles" for the main character which must be resolved or "skipped" because the continuation of the game depends on it (according to "if ... then ... else" model).

For logical tasks, students can also be asked, for example, what items are collected, what they look like, what elements will be "wrong", how points are gained or subtracted in the game, how to move a character towards a given object, etc.

The recommendation is that the story has no more than 3-4 scenes that are connected with 2-3 logic games to achieve the "if ... then ... else" flow of the game. The end of the story should depend on the results of the logic games played, so the endings of successfully solved games should differ from the endings of unsuccessfully solved games.