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

Licence
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

Disclaimer
The sole responsibility for the content of this document lies with the authors. It does not necessarily reflect the opinion of the European Union.
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 ................................................................. 6
Part II: Learning Outcomes and Topics for F2F Workshops ........................................... 9

Workshop 1: Game Based Learning (GBL) and Unplugged Activities .............................. 10
  Workshop schedule ..................................................................................................... 11
  Presentation: Introduction to Workshop 1 ............................................................... 13
  Session 1: Game based learning ............................................................................. 15
  Presentation: Games in education ........................................................................... 17
  Presentation: Integration of games into the lecturing process ................................. 20
  Presentation: Serious games evaluation framework ............................................... 24
  Session 2: GBL with unplugged activities ............................................................... 26
  Presentation: GBL with unplugged activities ......................................................... 28
  Session 3: Using Web 2.0 tools for creating content for unplugged activities .......... 33
  Presentation: Using Web 2.0 tools for creating content for unplugged activities ...... 35
  Session 4: Designing learning scenarios ............................................................... 39
  Presentation: Designing learning scenarios ............................................................ 41
  Session 5: Designing learning scenarios using a graphical tool ............................... 44
  Presentation: Designing learning scenarios using a graphical tool ......................... 46
  Session 6: Designing learning scenarios for unplugged activities ............................ 49

Workshop 2: Problem Based Learning (PBL), Online Quizzes and Logical Tasks .............. 51
  Workshop schedule ..................................................................................................... 52
  Presentation: Introduction to Workshop 2 ............................................................... 54
  Session 1: Introduction to Problem Based Learning (PBL) ...................................... 56
  Presentation: Introduction to Problem Based Learning (PBL) .................................. 58
  Session 2: Problem-solving in logical games ......................................................... 61
  Presentation: Digital tools within the process of problem-solving ............................ 63
  Presentation: Role-playing games .......................................................................... 69
  Session 3: Online quizzes and logical tasks ............................................................ 75
  Presentation: Online quizzes and logical tasks ....................................................... 77
  Session 4: Using Web 2.0 tools for creating quizzes and logical tasks .................... 81
  Presentation: Using Web 2.0 tools for creating quizzes and logical tasks ............... 83
  Session 5: Designing learning scenarios for logical tasks ....................................... 88

Workshop 3: Games and Tools for Programming ............................................................ 90
  Workshop schedule ..................................................................................................... 91
Presentation: Introduction to Workshop 3 .................................................................93
Session 1: Inquiry Based Learning ........................................................................96
  Presentation: Inquiry Based Learning .................................................................98
Session 2: Basic programming concepts ...............................................................101
  Presentation: Basic programming concepts .......................................................103
Session 3: Learning programming with games and stories .....................................107
  Presentation: Learning programming with games and stories ............................109
Session 4: Introduction into visual programming with Scratch ..............................114
  Presentation: Introduction into visual programming with Scratch .......................116
Session 5: Implementing Computational Thinking and programming with GBL tools ....121
  Presentation: Implementing Computational Thinking and programming with GBL tools .....123
Session 6: micro:bit in classroom ........................................................................126
  Presentation: micro:bit in classroom ................................................................128
Session 7: Designing learning scenarios ...............................................................132
Part III: Annexes ..................................................................................................134
Annex 1: Learning scenario template .................................................................135
Annex 2: Game/story scenario template .............................................................137
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
Introduction to the Workshop 1

Agenda
- Main goals of the workshops
- Terms: Algorithmic Thinking and Computational Thinking
- Introduction round: The participants introduce themselves
- Introduction and enroling to the course “Games for Learning Algorithmic Thinking” in Moodle

Purpose of workshops
- Encouraging the integration of coding and algorithmic thinking into daily teaching through different subjects in students’ curricula in a fun and attractive way
- Special focus: using educational strategies of Game-Based Learning (GBL) and gamification in order to foster creativity, logical thinking, and problem-solving skills
- General goals: improving students’ attitudes towards coding and the development of algorithmic thinking of younger students, including the “fun” towards coding and increasing students’ interest in the option of future career in the ICT and STEM areas (for the long term)

Computational and algorithmic thinking
- Computational thinking: A fundamental approach for developing programming and problem-solving skills.
- Algorithmic thinking: one of the concepts of computational thinking important for formulating problems.
- A program is a step-by-step instruction for operations with a specific order of steps.
- The problem is previously known, we can try to write the program, decompose, and analyze it (algorithm).
- We can use and combine different computational thinking skills.

Algorithmic thinking
- Promotes precision and systematic thinking.
- Appropriate not only for problems that are solved by computers but also for application in other areas and everyday life.
- Developing algorithms and digital works.
- Model thinking: understanding the structure and function of the program.
- Students develop an understanding of concepts that can be incorporated into future professional life.
- Unplugged activities: simulate algorithmic thinking through paper and pencil without the use of computers.
- Digital games.

GBL and unplugged activities
- Learning outcomes:
  - describe principles of game-based learning
  - use Web 2.0 tools for creating content for unplugged activities
  - create learning scenarios and playtest with peers.
  - Workshops: practice different GBL and unplugged activities in different school subjects.
Introduction round

• Short introduction to others.
• What was your motivation for joining the GLAT education?
• What do you expect from the workshops?

Enrolling to the e-course

Individual activity

LMS Moodle e-course

https://real.tecnico.ulisboa.pt/TeLdeLis

Questions

Let’s start…
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


Duration: 3 hours (135 minutes)
<table>
<thead>
<tr>
<th>Topic/Sub-topics</th>
<th>Learning Objectives</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. GAMES AND LEARNING</strong></td>
<td>Participants will understand the psychological and cognitive aspects of game based learning.</td>
<td>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).</td>
</tr>
<tr>
<td>1.1. Introduction to games</td>
<td>Identify the concepts of games</td>
<td></td>
</tr>
<tr>
<td>1.2. Games in human development</td>
<td>Understand the role of games in cognitive development</td>
<td></td>
</tr>
<tr>
<td>1.3. Games and learning theories</td>
<td>Explore the features on serious games from learning theories point of view</td>
<td></td>
</tr>
<tr>
<td><strong>2. INTEGRATION OF GAMES INTO LEARNING PROCESS</strong></td>
<td>Participants will be able to find, evaluate and select suitable serious games and integrate them into learning process.</td>
<td>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).</td>
</tr>
<tr>
<td>2.1. Identification and evaluation of suitable serious games</td>
<td>Explore games available on recommended portals or on the web</td>
<td></td>
</tr>
<tr>
<td>2.2. Integration of games into the learning process</td>
<td>Create a game based „learning package“</td>
<td></td>
</tr>
</tbody>
</table>
Presentation: Games in education

Workshop 1: Game Based Learning (GBL) and unplugged activities
Session 1: Game Based Learning (GBL)
Games in education

Agenda
- Games in child development
  - Educational games
  - Entertainment 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
  - International
  - Supported by ICT

Influence of games on child development
- Importance of children'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:
  - Natural games to new environment
  - Learning through play
- During play, the child discovers basic concepts from the real world and the first fundamental relationships between them are created.

Games in human development

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 curiosity and behaviour. Repetition of learned concepts results in an established part of the mental processes."
"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 (p. 1544, 2006)
- Technology usage fluency
- Multitasking
- Individualization and personalization
- Increased connectedness
- Immediate
- Multiple media types
- Engagement and working attitude
- Sociality and team spirit

Additional characteristics of “gamers”
- Ability to follow instructions
- Proficiency of problem-solving strategies
- Quick thinking
- Random access to resources
- Increased hand-eye coordination and fine motor skills
- Spacial reasoning
- Stimulating learning experience
Games for Learning Algorithmic Thinking

Games and learning theories 1/2

- Behaviorism
  - direct, immediate response
  - positive (correction), negative (punishment), reinforcement
  - games that have a skill and practice concept (e.g., multiplication table)
  - Constructionism
  - learning is an active process of constructing knowledge, based on prior knowledge that the user already has

Games and learning theories 2/2

- Constructivism
  - learning process involves the learner in constructing meaning, new ideas, and in building new knowledge
  - learning takes place within a social context and interaction between learners and the game is a necessary part of the learning process
  - Teacher's task: guiding and providing feedback
  - Examples: role-playing, adventure, detective games, simulation...

Educational games

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

Educational game elements

- Educational games must include:
  - competing players
  - challenge (a problem to be solved)
  - rules of engagement
  - interaction with the environment and control
  - continuous feedback
  - particular game or tasks to solve
  - 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 motivation before the lecture
  -4. games as a part of the lecture
  -5. games as an activity in the classroom
  -6. games as an activity in 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
  - with the teacher in the classroom process - limited time for use of alternative learning resources in formal education

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

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

Exploring existing games

- Explore some of the following games and think whether they are suitable for inclusion in the learning process:
  - http://www.ordorner.com/
  - http://www.ordorner.com/games
  - http://libreoffice.com
  - http://freepdf ea.org/games
  - http://home.tilda.org.in/\~shashshash/alexander_eu.html
  - https://freemadewithflash.com/

Preparing “learning package”

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

- Light Bot (HR, SL, EN)
  - http://freepdf ea.org/games/lightbot.html
- http://freepdf ea.org/games/lightbot.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 lesson, repetition, homework, ...)?
- Are there any instructions? How will they be delivered to students?
- What will the teacher have to do?
- 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
**GLAT – Workshop Syllabus and Materials**

### Project:
2017-1-HR01-KA201-035362

### 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 “sleep learning”?
  - Do the students need to solve non-standard 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)
  - solution based
  - possible to repeat the actions and improve results
- Is Critical Thinking incorporated in the game design?
  - decision making
  - players need to solve some puzzles
  - need to understand different perspectives (e.g., other players goals and strategies)

### Games for Learning Algorithmic Thinking

### Analysing examples
Group activity

### Example 1.1. Science for kids
- Fun science games for kids: subtle learning more about science and technology
- Free online activities to try with something for everyone
  - chemistry
  - geography
  - physics
  -...

### Example 1.2. IXL
- Immersive, adaptive learning
- Maths, language, arts, science
- KS2 education
- 30 free practice problems

---

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

Example 2.1. Teach your monster to read
- Free game that makes learning to read fun
- Letters, sounds, reading full sentences
- Designed in collaboration with reading academics
- Complements learning programmes used in schools

Example 2.2. Learn English, kids
- Fun games in English:
  - listen and write
  - read and write
  - speak and spell
  - fun and games

3. Mathematics

Example 3.1. Mathplayground
- Problem-solving math games:
  - logic and number puzzles
  - logical adventures
  - thinking blocks
  - math word problem practice
  - money games

Example 3.2. Maths4fun
- Maths explained in easy language:
  - algebra
  - data
  - geometry
  - measurement
  - numbers
  - dictionary
  - games
  - puzzles
  - work sheets

Example 3.3. Transum
- Free math activities for teaching and learning:
  - puzzles and problems
  - visual aids, investigations

Example 3.4. Fractions
- Fractions games:
  - equivalent fraction game
  - add fractions
  - converting fractions into decimals
  - fraction word problem games
  - subtracting mixed fraction
4. Geography

Example 4. Travel the world
- Tutorials and games:
  - World’s continents
  - Countries
  - Landmarks
  - Landscapes
- Games for beginners, intermediate and advanced learners

Example 5.1. Minecraft
- Building in a 3D procedurally-generated world with a variety of different cultures
- Requiring creativity from players
- Other activities:
  - Exploration
  - Resource gathering
  - Building
  - Combat
- Cost: 235

Example 5.2. Roblox
- Online gaming platform for kids and teens
- Students can create adventures, role-play, and learn with their friends in a highly-detailed, immersive, 3D environment

Exploring games
- Group activity

Questions

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.
Presentation: Serious games evaluation framework

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

Session 1: Game Based Learning (GBL)

Serious games evaluation framework

Agenda

- Experimental Learning Cycle
- Serious games Evaluation Framework
  - Simplified step by step version

Theory behind: Experimental Learning Cycle

- Concrete Experience
  - Active Experimentation (doing/trying out what you have learned)
  - Reflective Observation (reflecting on the experience)
- Abstract Conceptualization
  - Abstracting/relating new concepts with old ones

Serious Games Evaluation Framework

Simplified step by step version

Simplified methodology

Step 1: Can I use the game (filter step, if no, the evaluation ends)
- Technical requirements
- Age
- Special Considerations Needed

Step 2: Game evaluation (axes)

- Axess
  - Is the game easy to use? (USW)
  - What is the educational value of the game? (Vol)
  - Is the game adaptable to the educational goal? (AUG)
  - Student Quality of Experience (CoQ)
  - What is the teacher (author) subjective opinion about the game (USM)
Step 2: Game evaluation (grades)

Grades:
1. Not satisfactory
2. Satisfactory
3. Good
4. Very Good
5. Excellent

* This should sound familiar.

----

Step 2: Example

<table>
<thead>
<tr>
<th>GAME 1</th>
<th>GAME 2</th>
<th>GAME 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>EASY</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>TKG</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>ACT</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>CHG</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>TOT</td>
<td>11</td>
<td>16</td>
</tr>
</tbody>
</table>

* TOTAL here is the sum of grades. It is multiplied by some factor to emphasize that evaluation element.
* The grade is subjective due to the subjective opinion of the educator.

----

Evaluating games

Group activity

Use the simplified evaluation framework to evaluate several games.

Step 1: Select the games that will pass this filter step for the purpose of the exercise.
Step 2: Use both visualisation and numbers-based approach to evaluate the game.

Let's use known app.

Questions
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


Duration: 1 hour (45 minutes)
<table>
<thead>
<tr>
<th>Topic/Sub-topics</th>
<th>Learning Objectives</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. UNPLUGGED ACTIVITIES</strong></td>
<td>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.</td>
<td>Learners explore, analyze and classify existing examples of unplugged activities in order to transfer given examples to another school subject (group activity).</td>
</tr>
<tr>
<td>1.1. Introduction to unplugged activities for the development of algorithmic thinking</td>
<td>Describe and explain the characteristics of unplugged activities for algorithmic thinking development</td>
<td></td>
</tr>
<tr>
<td>1.2. Examples of unplugged activities in different school subjects</td>
<td>Analyze and classify existing examples</td>
<td></td>
</tr>
<tr>
<td><strong>2. DESCRIPTION OF UNPLUGGED ACTIVITIES EXAMPLES</strong></td>
<td>Participants will be able to describe their own examples of unplugged activities appropriate for different school subjects.</td>
<td>Learners describe new examples for unplugged activities (group activity - discussion).</td>
</tr>
<tr>
<td>2.1. New examples of unplugged activities</td>
<td>Propose examples of unplugged activities for algorithmic thinking from tales, everyday life, etc.</td>
<td></td>
</tr>
</tbody>
</table>
Presentation: GBL with unplugged activities

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

Session 2: GBL with unplugged activities

Agenda
- Introduction to unplugged activities
- Types of unplugged activities
- Examples related to specific school subjects

Unplugged activities
- Activities that enable teaching and learning without using computers
- Teaching through engaging games and puzzles that use simple cards, strips, patterns...
- Facilitates development of computational and algorithmic thinking
- Can be used in different parts of the lesson and for different curricula
- Can be for collaborative and individual use

Types of unplugged activities
1. Finding words in the grid
2. Real-life algorithms
3. Algorithms and analogies for number-related to specific school subjects
4. Moving through a maze
5. Tales and Algorithms
6. Writing or drawing in grid

Analysing examples
Group activity
Example 1: Hidden fruits
(English, Math)
- Find the hidden fruits in the grid and put the pictures of relevant fruit near the alphabet signs of its name.
- How many pictures of the fruits do you need for each word?

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?

Example 3: Word search
(English, Computer science)
1. Follow given instructions to find the hidden words. Start from upper left corner.
2. What is the meaning of the words?

Idea for activity
- Divide the class into two groups and organize content:
  - First group has to write words in the grid and explain 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.
- Benefits for focused word could be rewarded.

2. Real life algorithms
- Recurring algorithms in our daily lives:
  - Making sandwiches
  - Preserving food
  - Cleaning teeth
  - Preparing school back-to-school
  - Natural phenomena

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.

Example 2: Dance moves
- Students recognize dance attributes
- They label the entire dance performance with the agreed marks
- They connect dance structures with the corresponding part of the music background
Example 3: Code.org - Getting Loopy

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

Idea for activity

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

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

- Ordering of rules
  - Arranging mathematical operations
  - Grammar
- Writing a song in algorithm
  - Set of instructions

Analysing examples

Group activity

Example 1: Adding numbers (Math)

- Ordering of steps (rules) to add two three-digit numbers
- Steps are written on the paper and cut
  - Students have to order steps
- A set of steps is given to write order and students have to find error in the algorithm

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

- Rule for determining the appropriate tense
  - General present
  - Permanent activity
  - Temporary activity
  - Unlimited duration
  - Limited duration

Example 3: Loops in the song (Music)

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

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.
Example 1: Code.org - Happy maps
- Students create simple algorithms to move a character through a maze.

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 preformatted commands to direct their classroom to reproduce a drawing (e.g., color squares in on graph paper).

Example 4: Code.org - Move it
- Students learn how to think ahead in multiple steps, e.g., they plan a short route from the start location to the hidden candy, factory, up to three steps away.
- Starting point is the piece of paper imprinted with the compass rose.

Example 5: Tales and algorithms
- Recognising algorithms in familiar stories and fairy tales.
  - The Wild Swans tale by Hans Christian Andersen (7) algorithms steps for how seven brothers.
  - Hamlet and Othello: they execute an algorithm to get home from the forest.
  - Cinderella: Making a girl go to the ball.

Example: Cinderella
1. Find Cinderella
2. Find a glass slipper
3. If the shoe fits, then Cinderella is found. In other cases go to step 1.
6. Writing or drawing in a grid

- Using a sequence of signs to:
  - write numbers or words
  - draw shapes

Example: Writing letters and drawing shapes

- Students use the following signs to write numbers or words
- place:
- draw:
- do not draw:

Discussing examples

Discuss in groups possible variants for modifications of the presented example and plan similar examples for different subjects in primary school. Share your ideas with other groups.

Questions
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


Portals with tools and resources for teachers:

- The Teacher’s Corner: https://worksheets.theteacherscorner.net/ (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)

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

Agenda
- Introduction - Web 2.0
- Web 2.0 tools for preparing unplugged activities
  - Canvas
  - GooglePoint

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
- 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 (author(s) and when we learn)
- Student in the center of learning process
- Personalization
- Collaborative learning
  - interaction
  - forming the community of practice
  - collecting knowledge
- Using a variety of tools
  - Web 2.0 tools
  - applications available on the Web that have the characteristics of Web 2.0
  - Multi-purpose (according to the main purpose)
    - replacement of face-to-face applications
    - education and exchange of multimedia content
    - exploration of ideas and creative learning
    - extending and exchanging
    - social bookmarking
Advantages and disadvantages of Web 2.0 tools

**Advantages**
- Accessibility, price
- Diversity
- Stimulate activity and creativity
-...

**Disadvantages**
- Limited access to reliability
- Limited features of free versions
- Ads/Advertisements
-...

---

**Canva**
- Link: [https://www.canva.com](https://www.canva.com)
- Login: Facebook, Google account
- User: Free (Basic version)
- Purpose: making presentations, graphics, posters, posters
- Predefined templates
- Insert 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/](https://sketch.io/sketchpad/)
- Login: It is not necessary
- User: Free (basic version)
- Purpose: creation of vector graphics
- Draw and color
- Write text
- Insert ready-made graphical elements (images, icons, ...)
- Insert own files
- Save (Q) 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
**Task 1 - Walking through the maze**

- **Task**: Show with the arrows the steps that will take Little Bear to his mother.

**Task 1: Worksheet**

- **Draw a maze**
  - Select a template to create a maze
  - Add fields and backgrounds, insert pictures of trees and flowers
  - Insert pictures of little bear and mother bear
  - Give the worksheet a title

- **Use a template to create a worksheet (MS Word)**
  - Type task text
  - Insert a picture of the maze
  - Insert the arrows for routing on the second page of the document

**Task 1: Expected solution**

**Task 2 - Algorithm for the song**

- **Task**: Show algorithm that includes repetitions of a song “Wash your hands” [https://www.youtube.com/watch?v=M4s0Jm0kL8](https://www.youtube.com/watch?v=M4s0Jm0kL8) with pictures:
  - **WASH, WASH, WASH YOUR HANDS**
  - **RUB AND SCRUB, SCRUB AND RUB**
  - **GOING DOWN THE DRAIN**
  - **WASH, WASH, WASH YOUR HANDS**
  - **PLAY YOUR HAND GAME**
  - **RUB AND SCRUB, SCRUB AND RUB**
  - **DIRT GOES DOWN THE DRAIN**

**Task 2: Worksheet**

- **Draw a maze**
  - Select a template to create a maze

- **Draw a placemat**
  - Select a template to create a placemat

**Task 2: Expected solution**

The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Union.
Task 2: Creating the necessary materials
- Prepare pictures (Storyboard)
- Create or draw pictures
- Insert text
- Insert text
- Use a template to create a worksheet (Excel, Word)
- Insert images
- Insert text
- Insert teacher's notes
- Insert student's notes
- Insert text

Designing unplugged activities for different school subjects 1/3
- Explore available resources on the following portals:
  - Tools for educators:
    - http://www.opensource.education.com/
    - https://teacher.creativecommons.net/
  - Education website:
    - http://www.education.com/teacher/lesson/activity/...
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/](https://scenariji-poucavanja.e-skole.hr/) (5.12.2017.)
- Code Studio – katalog: [https://studio.code.org/courses](https://studio.code.org/courses) (7.12.2017.)

Duration: 2 hours (90 minutes)
<table>
<thead>
<tr>
<th>Topic/Sub-topics</th>
<th>Learning Objectives</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. LEARNING SCENARIOS</strong></td>
<td>Participants will be able to describe and explain the concepts of the learning scenario, analyze and compare existing examples of learning scenarios.</td>
<td>Learners explore and analyze existing examples of learning scenarios in order to point out good and bad features (group activity).</td>
</tr>
<tr>
<td>1.1. Introduction to leaning scenarios</td>
<td>Identify the concepts of learning scenarios</td>
<td></td>
</tr>
<tr>
<td>1.2. Investigate examples of existing learning scenarios</td>
<td>Analyze and compare existing examples</td>
<td></td>
</tr>
<tr>
<td><strong>2. DESIGNING LEARNING SCENARIOS FOR UNPLUGGED ACTIVITIES</strong></td>
<td>Participants will be able to create a learning scenario for chosen unplugged activity.</td>
<td>Learners choose one unplugged activity among offered to create a learning scenario that will be evaluated by the teacher and the colleagues (group activity).</td>
</tr>
<tr>
<td>2.1. A learning scenario template</td>
<td>Create a learning scenario using a prepared template for preselected unplugged activity</td>
<td></td>
</tr>
<tr>
<td>2.2. Filling out a learning scenario template</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Presentation: Designing learning scenarios

Workshop 1: Game Based Learning (GBL) and unplugged activities
Session 4: Designing learning scenarios

Agenda
- Introduction
- Designing 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 lessons
- Include active learning:
  - understanding
  - gaining own stimulus
  - critical thinking
  - creative problem solving
- Skills for monitoring and evaluating students

Planning and preparation
- Designing the learning environment:
  - one task
  - learners
  - learning unit
  - the entire teaching subject
- Defining:
  - resources and learning materials
  - equipment and technology
- Substitutes

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 format (Difference)
- 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
  - Tests for realizing the learning outcomes

Authors and licence
Author:
- Jasmina Miletic, University of Bijeljina, Faculty of Teacher Education:
  jasmina.miletic@uni-bijeljina.hr

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

Attribution should begin in the following way:
GLAT project: https://ec.europa.eu/programmes/erasmus-plus/ Call: 2017-1-HR01-KA201-035362

The sole responsibility for the content of this publication lies with the authors. It does not necessarily reflect the opinion of the European Union.
**Example: Moving through the maze/Spatial orientation 3/5**

**Teaching scenario**

- Title: Moving through the maze
- Description: Students are divided into groups and are given a map of the maze. The goal is to navigate through the maze as quickly as possible while avoiding obstacles.

**Questions**

- What strategies did you use to navigate the maze?
- How did you overcome challenges?

---

**Example: Moving through the maze/Spatial orientation 4/5**

**Teaching scenario**

- Title: Moving through the maze
- Description: Students are given a set of instructions to follow, which include identifying landmarks and making decisions based on their position in the maze.

**Questions**

- What were the key landmarks you used to navigate the maze?
- How did you make decisions along the way?

---

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

**Teaching scenario**

- Title: Moving through the maze
- Description: Students are given a more complex maze with multiple paths and dead-ends. They must use critical thinking and problem-solving skills to find the correct route.

**Questions**

- What were the biggest challenges you faced in navigating this maze?
- How did you adapt your strategy as you progressed through the maze?
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.)

Duration: 2 hours (90 minutes)
<table>
<thead>
<tr>
<th>Topic/Sub-topics</th>
<th>Learning Objectives</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. PLANNING AND CREATING LEARNING SCENARIOS</strong></td>
<td>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.</td>
<td>Learners create a dummy account for LePlanner log in accounts, create a dummy lesson, and prototyping (individual activity).</td>
</tr>
<tr>
<td>1.1. Introduction to LePlanner</td>
<td>Explore the key features of the LePlanner</td>
<td></td>
</tr>
<tr>
<td>1.2. Creating a learning scenario</td>
<td>Create a sample leaning scenario(s)</td>
<td>Learners create a real log account in LePlanner, create a lesson, and publish it.</td>
</tr>
<tr>
<td>1.3. Reviewing created learning scenario (evaluating, editing and publishing scenarios)</td>
<td>Explore the features of the timeline for creating course contents</td>
<td></td>
</tr>
<tr>
<td><strong>2. DESIGNING LEARNING SCENARIOS USING LEPLANNER</strong></td>
<td>Participants will be able to create (unplugged, game based) learning scenarios using LePlanner and demonstrate the teaching of the planned lesson(s)</td>
<td>Learners are engaged in micro-teaching using the designed lesson plan as a group activity.</td>
</tr>
<tr>
<td>2.1. Designing Game Based Learning Scenarios and participating in demonstration lessons</td>
<td>Explore online games, and create an unplugged game based learning scenario</td>
<td></td>
</tr>
</tbody>
</table>
**Presentation:** Designing learning scenarios using a graphical tool

**Agenda**
- Short review of previous knowledge
- Introduction to LePlanner
  -Scenario design
  - Exploring the features of LePlanner
  - Signing up for an account

**Short review of previous knowledge**
- What is your opinion on the following themes of the workshop:
  - games in learning
  - designing learning scenarios
  - incorporating game-based learning in your learning scenarios
- What new professional impressions or knowledge have you internalized so far?

**LePlanner**
- Tool for designing, visualization and sharing learning scenarios
- Developed at School of Digital Technologies, Tallinn University
- Available online: [https://beta.leplanner.net](https://beta.leplanner.net)
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)

LePlanner: Text view (scenario)

Exploring LePlanner
- Individual activity

Designing and implementing a learning scenario
- Group activity
Designing an unplugged game-based learning scenario

- Select group name and subject of interest
- Design unplugged game-based learning scenario to last for about 10-15 minutes on the LePlanner
- Present created learning scenario to other groups
- Review learning scenarios created by other groups

LePlanner Instructional Sheet
## 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

<table>
<thead>
<tr>
<th>ASSIGNMENT STEPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Choose a school subject – plan the activities that will be carried out in your class next month.</td>
</tr>
<tr>
<td>2. Use the <strong>Learning Scenario Template</strong> form (Annex 1) for textual version and LePlanner for the graphical version of your scenario.</td>
</tr>
</tbody>
</table>
| 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

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

Introduction to the Workshop 2

Agenda
- Reminder to the purpose of the workshops
- Results of the Workshop 1
- Introduction to Workshop 2

Main goals of the project
- Encouraging the integration of algorithmic thinking into daily teaching through different subjects in the first to fourth grade of primary school
- Training of teachers including the acquisition of contemporary knowledge and skills connected to different ICT-related innovative teaching methodologies such as Problem-Based Learning (PBL), Inquiry-Based Learning (IBL), Game-Based Learning (GBL)
- Creating blended learning e-course in LMS (platforms, materials in English and partially in Croatian) for further use in the partner countries and beyond

Purpose of workshops
- Encouraging the integration of coding and algorithmic thinking into daily teaching through different subjects in students’ younger ages in a fun and attractive way
- Special focus will be on using educational strategies of Game-Based Learning (GBL) and gamification in order to foster creativity, logical thinking, and problem-solving skills
- General goal improving students’ attitudes towards coding and the development of algorithmic thinking of younger students, reducing the “fear” towards coding and increasing students’ interest in the selection of future careers in the ICT and STEM areas in the long term

Workshops
- Three workshops:
  - Workshop 1: Game-based learning (GBL) and unplugged activities
  - Workshop 2: Problem-based learning (PBL) - online quizzes and logical tasks
  - Workshop 3: Games and tools for learning programming

Reminder about the Workshop 1
- Workshop 1: GBL and unplugged activities
- Learning outcomes:
  - basic principles of Game-Based Learning
  - VanLehn’s 5 phases for creating content for unplugged activities
- Create learning scenarios in order to develop innovative ideas for carrying out unplugged activities in different school subjects.
Learning scenarios carried out in schools for different subjects
- The outstanding learning scenarios as examples of good practice

Workshop 2: Problem-Based Learning (PBL), online quizzes, and logical tasks
- Learning outcomes:
  - Describe the principles of Problem-Based Learning
  - Use with students for creating logical tasks and online quizzes
  - Create learning contexts in order to develop innovative ideas for solving logical tasks and online quizzes
- Relevant learning scenarios for different courses in the classroom with students from 3rd to 9th grades of primary school.

In the following workshop pay attention to...
1/3
Combinatorial and algorithmic thinking (CT & AI)
- Include elements of learning scenarios along with the learning activities related to the course, include those that are close to CT & AI
- Learning scenarios:
  - After the workshop 2, it is mandatory to implement at least one scenario
  - Use games, quizzes, or logical tasks prepared for computer/teachers/teachers; not just one example (not only unopenedicking)
  - 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.
- Skills from various reflections, particularly those in the learning scenarios related to algorithms and computational thinking have been reviewed.
- State the number of students who participated in the learning activity.
- Describe how your students have engaged in the learning activity.
- Comment on the achievement of all learning outcomes in the scenarios.
- Define what would you like to change before the next implementation of the scenarios.

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 parental consent

Questions

Let's start...
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

### Duration
1 hour (45 minutes)

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

Workshop 2: Problem Based Learning (PBL), online quizzes, and logical tasks
Session 2: Introduction to Problem Based Learning (PBL)

Agenda
- Problem Based Learning Explained (definition and descriptions)
- Key principles of Problem Based Learning
- Case illustrations and demonstration in the context of teaching coding

What is PBL? 1/2
Classical/linear instructional sequence:
- The teacher presents the new knowledge (concepts, facts, procedures, rules)
- Learners apply the knowledge to solve a pre-defined problem
- The teacher gives positive or negative feedback

What is PBL? 2/2
Alternative, cyclical/iterative PBL sequence:
- The teacher presents a phenomenon from real life context
- Learners work in groups to define a problem
- Learners seek theoretical/practical knowledge that can help to solve the problem, teacher facilitates if needed, the problem is explored
- Learners collaborate while applying the new knowledge to solve the problem, teacher facilitates
- Learners reflect and compare each other’s solutions (using subject knowledge), teacher gives feedback (assessment of learning), learners improve their solution and seek new knowledge if needed

Defining PBL
- PBL is a pedagogical design that uses real-life problems to trigger learners’ engagement in collaborative learning and to promote their development of critical thinking, information seeking, and knowledge sharing.
- PBL is similar to:
  - Inquiry-based learning (knowledge is discovered through active experimentation and hypothesis testing, and it is applied to the real world)
  - Project-based learning (learners learn while engaged in a real-life project)
  - Case-based learning (learners learn through case studies)
  - Active learning (students are actively engaged in knowledge building)
  - Problem-based learning (students are taught through reflection on actual, theory-oriented exploration)

What is not PBL (although it looks like it)?
- Solving well-structured mathematical problems to retain knowledge presented by teacher (a single correct solution & answer)
- Discussing general (personal, global) problems/dilemmas without trying to solve them
- Solving problems without learning anything new related to domain knowledge or curriculum objectives

PBL principles
- Problems Based Learning is:
  - Student- and self-directed, not teacher-led
  - Collaborative, not individual
  - Interactive, not linear (regarding the structure of learning process)
  - Reflective, not silent
  - Develops/improves subject knowledge, not just fun

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

Exploring case illustrations
Group activity

Example 1: 4th grade science lesson, Moldova
- Topic: elastic force and friction
- Task: in groups of 4, compose and program a robot that would illustrate your theory that explains the phenomena

Example 2: PBL teacher training in Estonia
- Trigger: the most well-known children song (The Unicorn)
- Task: define and solve a problem, using coding and/or robotics

Example 3: 5th grade science lesson in Estonia - www.progettiger.ee
- Topic: renewable energy
- Task: build a model of a wind generator using strawberries & OurRibo

Identifying cases where PBL could be applied
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? What should this session become truly a PBL experience for you as learners?
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


Duration: 3 hours (135 minutes)
<table>
<thead>
<tr>
<th>Topic/Sub-topics</th>
<th>Learning Objectives</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. DIGITAL TOOLS WITHIN THE PROCESS OF PROBLEM-SOLVING</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1 Introduction to problem-solving</td>
<td>Participants will recognise the process of problem-solving.</td>
<td>Learners explore and analyse examples of problem-solving techniques in order to point out typical characteristics of logical reasoning.</td>
</tr>
<tr>
<td>1.2 Developing problem-solving skills</td>
<td>Identify the concepts of: Analytical ability, Creative Thinking, Initiative, Logical Reasoning</td>
<td></td>
</tr>
<tr>
<td>1.3 Problem-solving within games and puzzles</td>
<td>Understand the role of analytical and creative skills in the process of problem-solving</td>
<td></td>
</tr>
<tr>
<td><strong>3. ROLE-PLAYING METHODOLOGY</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1 Developing the skills for mutual collaboration accepting different responsibilities (roles) participating in games that support algorithmic thinking</td>
<td>Participants will recognise the methodology of role-playing in serious games.</td>
<td>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).</td>
</tr>
</tbody>
</table>
Presentation: Digital tools within the process of problem-solving

Workshop 2: Problem Based Learning (PBL), online quizzes, and logical tasks
Session 2: Problem solving in logical games
Digital tools within the process of problem-solving

Agenda
• Introduction to problem solving
• The problem solving skills
• Logical thinking
• Creative thinking
• Reasoning
• Developing the problem-solving skills
• Games for enhancing problem solving skills

Motivation
• The main goal of the workshop is related to the algorithmic thinking development.
• Algorithmic thinking is developed by using skills for solving various problems that reflect real issues.
• Algorithmic thinking is related to problem-solving skills, logic and reasoning.
• MINDOR - Problem solving

Why we talk about problem solving?
• Problem solving skills lead towards developing algorithmic thinking
• Problem solving is part of everyday life...
• They continually make and execute algorithms
• Are design series of activities

This world is fully comprehensible only for those who are familiar with the basics of these activities.

Let’s answer, what is problem solving?
• We meet problems in our everyday life:
• Some problems that students have (younger and older):
• Have to travel from home to school
• How much money is needed to buy bread and something sweet?
• Planning the allowance to last till the end of the week
• Developing a strategy to reach the next level in a computer game
• …

The most important skill
• Problem solving is the ability to deal with problems
• To identify
• To solve
• Do it systematically!!!

Advantages of learning problem solving
• Employing science processes – STEM education
• Employing science processes in non-science subjects, daily life
• Problem solving develops HIGHER thinking skills
• Develops responsibility, creativity, resourcefulness, critical thinking
• The students learn to accept opinions and evidence shared by others – TEAM work
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
  - perseverance.
- 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 five steps model.
- Has direct applications to many areas of the curriculum and everyday life.
- Steps:
  1. Understand the problem.
  2. Analyse the problem.
  3. Identify various options.
  4. Try out the solution.
  5. Evaluate the results.

In other words... stages to solving a problem

- Evaluate the problem.
- Infer the information.
- Identify the problem.
- Brainstorm possible solutions.
- Select a solution.
- Test and review.
- Iterate.

Let's learn through example

- The 5 steps through an example
- The following examples are from:
  - Dobbs Edibles, http://dobb.ediblesltd.co.uk/

Analysing examples

Group activity

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

Question:
What 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 relevant 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 see all of these obstacles as always an important first step.
- Techniques to understand the nature of a problem and its conditions:
  - Gather relevant facts.
  - Make a list of all the given information.
  - Solve the problem in your own words.
  - Indicate the conditions that remain a problem.
  - Describe related known problems.

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 making students work in a team or daily, students can develop strategies and combine elements.
  - Remind them to:
    - Seek a problem.
    - Create a systematic list (table).

4. Try out a solution

Important clues:
- Keep accurate and up-to-date records of student thoughts, proceedings, and procedures.
- Try to mark 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 data.
- Monitor with care the steps undertaken as part of a solution.

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 have reached?”
  - “Why do you believe this can improve your performance to the problem?”

---

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

- The solution is B

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 180 football players by name. Ten are from Real Madrid. The rest are Juventus and Barca. He knows the names of twice as many Barca 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

2. Analyze
   - Gather data, conditions, obstacles.
   - Data: 10 Real Madrids
   - Conditions:
     - the number of Juvenats and Barca should equal 90
     - there are twice as many Barca as Juventus

3. Try a solution
   - New, solve the problem. You may want to set up a table to record the guesses:
     - Options: Barca, Juventus, Real Madrid
     - First guess: 30 Barca, 60 Juventus, 10 Real Madrid
     - Total: 100

   - New guess: 40 Barca, 50 Juventus, 10 Real Madrid
     - Total: 100

Finally, the answer!

<table>
<thead>
<tr>
<th>Game</th>
<th>Barca</th>
<th>Juventus</th>
<th>Real Madrid</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Guess</td>
<td>20</td>
<td>80</td>
<td>10</td>
<td>120</td>
</tr>
<tr>
<td>Second Guess</td>
<td>40</td>
<td>50</td>
<td>10</td>
<td>150</td>
</tr>
<tr>
<td>Third Guess</td>
<td>20</td>
<td>60</td>
<td>10</td>
<td>150</td>
</tr>
<tr>
<td>Fourth Guess</td>
<td>30</td>
<td>60</td>
<td>10</td>
<td>150</td>
</tr>
</tbody>
</table>

Co-funded by the Erasmus+ Programme of the European Union
The steps for the example

5. Evaluate the solution
   - Read the problem again to be sure the question was answered.
   - Check the number of times.
   - Check the math to be sure it is correct.

30 divided is 60.
30 + 60 + 10 = 100
- SOLUTION:

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

Creative thinking

- Sometimes, creative thinking will be necessary
- Ideas for solving 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
- Letter about their playing in team work

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

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:

- Mental games such as crosswords, Sudoku, puzzles, bridge,
- Computer games such as puzzles, card games, brain teasers,
- Decision-making games.

The chosen areas can include:

- Strategic planning
- Logical and statistical analysis
- Assessing the pros and cons of different courses of action

Enhancing Problem solving skills with games

- Utilizing digital tools
  - www.oreo.com
  - www.fjunbrain.com
  - www.educationworld.com

  - High energy engaging games in math, science, social studies, the arts, and thinking games. All can be found free online by subject or grade level.

Some games for elementary school

- http://www.funbrain.com/pre-k-and-k-playground
- http://www.fjunbrain.com/games/pre-k
- http://www.fjunbrain.com/games/second-grade
- http://www.fjunbrain.com/games/word-emoji
- http://www.fjunbrain.com/games/word-emoji
- http://www.fjunbrain.com/games/word-emoji
- http://www.fjunbrain.com/games/word-emoji
- http://www.fjunbrain.com/games/word-emoji
- http://www.fjunbrain.com/games/word-emoji

Some fun

- Games that develop problem solving skills
  - http://www.fjunbrain.com/games/word-emoji
  - http://www.fjunbrain.com/games/word-emoji
  - http://www.fjunbrain.com/games/word-emoji

Discussing examples

- Group activity

Questions
Presentation: Role-playing games

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

Session 2: Problem-solving in logical games
Role-playing games

Agenda
• Role-playing and knowledge collection
• Integration of role playing into learning process
• Provides context for role-playing of educational games
• Possible online role-playing educational games

What is role-playing?
• Multiplayer learning activity in which students behave in the way someone else would behave in a particular situation
• Pupils get a particular role and act towards fulfilling the task assigned by that role

In-class activities for pupils
• Suitable for younger pupils who are not already familiar with the topic
• Topics should be very close to the official syllabus of basic courses
• After initial lectures, each pupil is assigned a particular role
• Pupils get a question related to their role and generate the answer
• Each successfully answered question leads to extra reward
• In-class role playing can be performed as a competition between the teams consisting of the pupils with the same role

Examples for in-class scenario
• Addition: 2 + 8; 8 + 5; 4 + 9; 6 + 11; 7 + 14; 9 + 26; 10 + 35
• Subtraction: 5 – 1; 7 – 3; 12 – 5; 15 – 5; 14 – 9; 21 – 12; 41 – 27
• Multiplication: 2 * 3; 3 * 6; 4 * 7; 7 * 8; 6 * 9; 12 * 6; 10 * 7
• Division: 6 / 2; 12 / 3; 21 / 7; 35 / 7; 64 / 8; 96 / 12; 112 / 7

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 one
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).

More advanced level

- Eight teams, each successfully answer a basic level by all four teams.
- Expressions with several operations.
- Teams answer as quickly as they can.
- If the answer is wrong, the second team can try as well.
- If no one answers the right answer, the teacher should explain why.
- The expressions should be gradually increased.

Role-playing online scenario – an example

- The goal is to paint an image, for example, one or several cartoon heroes, which is visible at the beginning of the game.
- The computer game teacher is a painter, who reacts to pupils’ responses.
- Pupils are divided into teams.
- Each team has two colors: yellow, blue, green, and pink.
- Questions are randomly generated.
- Correct answers lead to one small part of the image of heroes painted in color of the team.
- When the game ends, the whole image is completed.

Computer role-playing games

- One of the most important games on PC.
- Include complex, controllable relationships with companions or non-playable characters.
- Features and games exist in support of this play style.
- Very similar to massively multiplayer online role-playing games (MMORPG).
- A digital 3D or 2D world.
- Users interact with one another to uncover the virtual world.
- Games prefer the canonical role-playing games (RPG).
- Enriched story.
- Story that evolves through the actions of the character through skill points, inventory, and dialogue decisions.

Famous RPGs

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

Edutainment software and role-playing

- Edutainment software supports role-playing.
- Games are adjusted to pupils’ age (complete beginners, elementary school, intermediate, advanced levels).
- Typical examples:
  - Mathematics, Mind Quest, Barrio Multi-Race for kids, Monster Math.
  - English: Dino the Dinosaur, (free version and school game), (learning process); Role Play Game, Drama Word Tramp.
  - Science: Jack’s Animal Barn, Classic K. (Olympus Board game).
  - History: The Dragon Tail, Medieval Merchants (board game), Civilization revolution, etc.
Example 1: Bunny Math Race

- Available from: http://www.youtube.com/watch?v=FRHgM5yZdyc
- FREE download: https://itunes.apple.com/app/id503137823
- Race between 12 & 18 hamsters
- Intended for children from 3 to 8 years
- Reward: more carrots 😊

Example 2: Jobi’s Animal Barn

- Available from: https://www.youtube.com/watch?v=FRHgM5yZdyc
- Free download: https://itunes.apple.com/app/id503137823
- Features:
  - How do learners grow and harvested vegetables?
  - How to clean the barn, remove spider webs, collect chicken eggs, help small ducklings find their moms, wash the piggies, feed the cows?
  - How to take care of messy tactics, mining and growing?
  - How do farm animals look like?
  - How to look after them?
  - How the next vegetables look like?
- Intended for children from 3 to 8 years

Advantages of role-playing

- Stimulates active learning - children learn by doing
- Amazing
- 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

Reminder of the in-class scenario

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

Role-playing scenarios with problem solving
In-class scenario

- Learning the four arithmetic operations for the numbers up to:
  - 100 / 1000 / 10000, depending on the class
- Four basic arithmetic operations: 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 ability to perform four basic arithmetic operations.
  - Advantage of role-playing for problem perception
    - Four roles rather than two.
    - Any misunderstanding can be avoided by intensive 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: 3 = 1; 2 * 3: 3 / 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 operation
- Obstacles:
  - Expressions are sorted from easiest to most complex.
  - Expressions with the 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 slide), or a multiple choice question task (like in the Brilli Math Pack).
- 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
- See saw
- Buttons

Try the solution

- Scales is useful for the easiest exercise.
  - Scales are applicable for more complex tasks including multiplications.
- Colorful buttons are the universal solution verifier.
  - By using them, twice as much votes can be assigned, first for those who guess the answer (green), and then for those who verify it (yellow).
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.

The problem - Beavers in the elevator

- 3 beavers enter in an elevator on 5th floor. The first beaver goes to the 5th floor, the second to 9th and the third to the 8th.
- Which order should the elevators go to make the shortest route possible?

A. 5-9-2
B. 8-9-2
C. 5-1-9
D. 5-9-1
E. 9-2-8-5

Analysis

Your assignment

- How does the problem „Beavers in the elevator” fulfill 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 elevator on the 6th floor.
- Rule 1 should arrive to 1st floor.
- Rule 2 should arrive to 8th floor.
- Rule 3 should arrive to 20th floor.
- The elevator should travel as short as possible.

Identify various solutions

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

Analyze the problem

- There are many solutions.
- Start from 5th floor and go down.
- Start from 6th floor and go up.
- If the elevator goes up, it can stop on:
  • 10th floor and carry on up or down.
  • 20th floor and carry on up or down.
- If the elevator goes down, it can stop on:
  • 1st floor and then go up.
-
The game - prerequisites

- Plates with the numbers 1, 16, and 20 (the rule)
- Three sets of tiles, each consisting of the numbers from 1 to 20
- Paper ladder with numerical order
- Paper ladder is put on the floor
- The class is divided into 5 teams
- If the number of pupils is 3n - 1, the teacher becomes a member of one team
- If the number of pupils is 3n + 1, one pupil becomes a Jester

Playing the game

1. All the pupils from the team step on the field
2. Randomly selected role (let's call it A) takes all pupils towards
   the floor
3. Pupils go field by field, counting the number of steps they make
4. Pupil A:
   1. Arrives at its destination
   2. Finishes the journey
   3. Tosses of 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

Try out the solutions

Potential solutions:
- A: 9 - 16 - 20 - 2
- B: 9 - 20 - 16 - 1
- C: 9 - 1 - 16 - 1
- D: 9 - 16 - 1 - 20
- E: 9 - 20 - 1 - 15
- F: 9 - 1 - 20 - 16

Path lengths:
- (20 - 9) + (20 - 12) + (20 - 1) = 20
- (20 - 9) + (20 - 16) + (16 - 1) = 30
- (20 - 9) + (20 - 15) + (15 - 1) = 30
- (20 - 9) + (20 - 12) + (20 - 1) = 31
- (20 - 9) + (20 - 15) + (15 - 1) = 45
- (20 - 9) + (20 - 12) + (20 - 1) = 46

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: 3 + 4 + 15 = 22
- For solution B: 3 + 4 + 15 = 22
- For solution C: 4 + 5 + 4 = 13
- For solution D: 7 + 15 + 1 = 23
- For solution E: 1 + 3 + 12 = 16
- For solution F: 9 + 19 + 4 = 34
- The shortest path is C: 9 - 1 - 10 - 20

Questions
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

- LearningApps: https://learningapps.org/ (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)
<table>
<thead>
<tr>
<th>Topic/Sub-topics</th>
<th>Learning Objectives</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. LOGICAL TASKS AND QUIZZES FOR DEVELOPMENT OF ALGORITHMIC SKILLS AND THINKING</strong></td>
<td>Participants will be able to classify logical tasks for propaedeutic of algorithm and programming and construct quiz appropriate for implementation.</td>
<td>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.</td>
</tr>
<tr>
<td>1.1. Classification of tasks for the development of algorithmic skills and thinking; examples from different school subjects</td>
<td>Classify logical tasks providing propaedeutic for algorithms and programming</td>
<td></td>
</tr>
<tr>
<td>1.2. Main requirements for online quizzes development</td>
<td>Construct quizzes appropriate for online implementation</td>
<td></td>
</tr>
<tr>
<td>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</td>
<td>Experiment with existing examples of logical tasks and quizzes in the form of games</td>
<td></td>
</tr>
<tr>
<td><strong>2. DEVELOPMENT OF EXAMPLES OF LOGICAL TASKS AND QUIZZES</strong></td>
<td>Participants will be able to create examples of logical tasks, appropriate for different school subjects.</td>
<td>Learners modify some of the examples and discuss the possibilities of implementation of the tasks in school subjects and lessons (group activity).</td>
</tr>
<tr>
<td>2.1. Modification and adaptation of examples of logical tasks for different school subjects</td>
<td>Create new examples of logical tasks based on given examples</td>
<td></td>
</tr>
<tr>
<td>2.2. Development of examples of logical tasks</td>
<td>Give new examples of logical tasks for algorithmic thinking</td>
<td></td>
</tr>
</tbody>
</table>
Presentation: Online quizzes and logical tasks

Agenda

- Logical tasks and quizzes for the development of algorithmic thinking
- Classification of tasks and examples from different school subjects
- Demonstration of examples of different logical tasks and quizzes developed in Web 2.0 environment
- Development of examples of logical tasks and quizzes for different school subjects

Classification of tasks for the development of algorithmic skills and thinking

Example: Search, find and sift essential from non-essential information

Example: Summarize and exclude items by attribute

Example: Comparison and classification
GAMES FOR LEARNING ALGORITHMIC THINKING

Analysing examples - learningapps.org
Individual activity

Let's play the role of a student

- Go to www.learningapps.org
- Login with user name: GLAT, password: glat, in the user name instead of * write your number (GLAT1, GLAT2, etc.)
- Play any Practical exercise 1

About Web 2.0 tool - learningapps.org

- Use the link Show Tutorial

Use learningapps.org as a teacher

- Create an account

Use existing apps

- Demo 1: Use of existing apps
- Demo 2: How to create apps from existing in learningapps.org?

Create new apps

- Demo 3: Classification of objects
- Use of Prune-Plot for development of graphical elements
1st practical task: Create exercise “Redundant word”

- See Example 4 “Redundant word”
- Create similar task for example “General word”

Which general word can be used to order the following listed?

- a) ball, cat, truck, taxi, bear......
- b) phone, clock, brush, ear... ...
- c) paper, men, lift, skirt ...
- d) bread, brush, spray,ologna......

2nd practical task: Creating an app

- Prepare one app to learningapp.org
- You can use resources materials, given in this workshop, find suitable images, audio or video on internet or use examples from the presentation
- Some additional useful apps:
  - Algorithms and Design
  - Software Engineer
  - deCode

Questions
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/](http://ucitelji.hr/dabar/) (30.6.2018.)
- LearningApps: [https://learningapps.org/](https://learningapps.org/) (4.7.2018.)

Web 2.0 tools:

- Kahoot: [https://kahoot.com/](https://kahoot.com/) (30.6.2018.)
- Wizer: [https://app.wizer.me/](https://app.wizer.me/) (30.6.2018.)
- Match the memory: [https://matchthememory.com/](https://matchthememory.com/) (4.7.2018.)
**Duration:** 3 hours (135 minutes)

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

Workshop 2: PBL, online quizzes and logical tasks
Session 4: Using Web 2.0 tools for creating quizzes and logical tasks

Agenda
- Web 2.0 tools for creating quizzes and logical tasks
- Web 2.0 tools for creating interactive worksheets
- Practical work: creating a quiz, an interactive worksheet and a digital memory game.

Quizzes
- Teaching element for progress monitoring and assessment, but also for learning
- CBMNet's e-learning [http://e-laboratoriya.com/category/elearning]
  - Teachers
  - Students
  - Assessments
  - Resources
  - Discuss
  - Hot Potatoes

Logical tasks
- Teaching element for learning, progress monitoring, and assessment
- Motivating students to solve tasks in a fun and interactive way
- Suitable for homework, work assignments in class or assignments for competition in different teaching subjects
- Different applications and programs
  - CBMNet's e-learning applications for creating interactive content
    - [http://e-laboratoriya.com/category/elearning/]
  - LearningApps: creating interactive applications
    - [https://learningapps.org/]

Examples
- Interactive logical games and tasks (in Croatian):
  - Dooler [http://dooler.hr/]
  - Logic [http://logic.hr/]
  - Attessa - online games
    - [https://www.attessa.com/hr/softver/]
  - Talak's Portal - online games
    - [http://www.gismanica.com/hr/online_games.htm]
Web 2.0 tool Kahoot!

Kahoot quiz 1/3
- Students do not need to register; they just use Kahoot.it address to enter the number of the quiz (Game PIN) provided by the teacher.
- Points are based on the score obtained for the correct answer and the time within which the correct answer was given.

Kahoot quiz 2/3
- Multiple choice question with 2 to 4 distractors and one correct answer is presented on teacher’s device and displayed to the students by the projector as long as the time limit allows.

Kahoot quiz 3/3
- Students will see two to four answer options and must choose a shape/color tile on their device matching the answer they think is correct.

Solving online quizzes
- Group activity

Solving Kahoot quiz
- Let’s play the role of a student and solve the quiz at Kahoot.it

Creating Kahoot quiz
- Individual activity
Use Kahoot! as a teacher

- Login at Kahoot.com and create an account.
- Create a quiz with at least 3 multiple-choice questions.
- Include cover design.
- The 1st question is text-based, e.g. mathematical task.
- The 2nd question and the 3rd question contain graphics (instructions are text-based), e.g. about science course.

Wizer

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

Wizer question types

- Wizer question types: open answer, multiple choice, matching, fill in the blank, fill in 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.

Solving Wizer worksheet

- Let’s play the role of a student and solve the worksheet at https://www.wizer.me/signup/.
- Sign in with nick name and use password P72@8F.
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 of questions:
  - Examples of types: Open answer, multiple choice, matching, fill in the blank, fill in an image.

Match the Memory

- Creating digital memory games with customizable 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

- Select 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, text, image, image, text-image, text-text, text-A, text-B, image A, image B.

Creating a memory game 2/2

- Rules are edited under the category Cards by defining corresponding types images and text.

Memory games list

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

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/match2](https://matchthememory.com/match2)

Use Match the memory as a teacher

- Login at [https://matchthememory.com/](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 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 game 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:
  - Multiple choice
  - Find the memory game
  - Worksheet
- 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

<table>
<thead>
<tr>
<th>ASSIGNMENT STEPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>12. Choose a school subject – plan the activities that will be carried out in your class next month.</td>
</tr>
<tr>
<td>13. Use the Learning Scenario Template form (Annex 1) for textual version and LePlanner for the graphical version of your scenario.</td>
</tr>
</tbody>
</table>
| 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 |
<p>| 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. |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>19.</td>
<td>Elaborate the <strong>Teaching summary</strong> as <strong>Motivation (Introduction)</strong>, <strong>Implementation</strong> and <strong>Evaluation (Reflection)</strong>. 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).</td>
</tr>
<tr>
<td>20.</td>
<td>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.</td>
</tr>
<tr>
<td>21.</td>
<td>In <strong>Annexes</strong> 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).</td>
</tr>
<tr>
<td>22.</td>
<td><strong>Examples and game references</strong> box should contain sources you will use for the activities.</td>
</tr>
</tbody>
</table>

### 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.

---

Point out all **Resources/materials** which will be required for the teacher as well as for students.
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

Main goals
- The development of algorithmic thinking, computational thinking, creativity, and problem-solving skills of students from 3rd to 4th grade of primary school.
- Introducing teachers to a variety of innovative teaching methods using ICT, especially educational learning strategies as game-based learning, problem-based learning, inquiry-based learning.
- Education for teachers in the form of a mixed e-course in the LMS (Syllabus, learning materials in English and Croatian) with three DIT workshops.

Workshops
- Three workshops:
  - Workshop 1: Game Based Learning (GBL) and un-plugged activities
  - Workshop 2: Problem Based Learning (PBL), online quizzes and logical tasks
  - Workshop 3: Games and Tools for programming

Workshop 1 reminder
Game Based Learning (GBL) and unplugged activities
- Game Based Learning – GBL
- Web 2.0 tools for creating contexts for game-based unplugged activities – Scratch, Sketchpad
- Designing learning scenarios that will include game-based unplugged activities (template in Scratch template and graphical in Lifesaver graphical tool)
- Implementation of learning scenario for different school subjects in primary education

Workshop 2 reminder
Problem Based Learning (PBL), online quizzes and logical tasks
- Learning activities:
  - Exploring problem-based learning (PBL) and collaborative learning
  - Design learning scenario based on the principles of problem learning that will include logical tasks and online quizzes along with educational games
  - Use Web 2.0 tools to create logical tasks and online quizzes
- Finally, the created learning scenarios in different school subjects in primary education
Learning scenarios carried out in schools for different subjects
• This outstanding learning scenarios after the Workshop 2 as examples of good practice.

Workshop 3
Games and tools for programming
• Learning outcomes:
  • describe the principles of IFL
  • explain the basic programming concepts
  • use digital tools to create interactive educational games
  • develop learning scenarios that will include, among other things, digital games, attempts of programming and computational thinking for different school subjects in primary education.

• Apply the created learning scenarios in different school subjects and primary education.

Workshop 3 – 1st day
• Introduction to Inquiry Based Learning
• Basic programming concepts
• Programming with games and stories
• Examples:
  • Run Make (https://www.runmake.com/games/university)
  • Weeky games (https://workshoprunmake.com/)
  • Code.org (https://www.code.org)
  • Scratch (https://scratch.mit.edu)

• Learning stories and games with Scratch

Workshop 3 – 2nd day
• Introducing programming in primary education
• Examples:
  • Scratch (https://www.scratch.org)
  • Creating basic micro:bit applications for different school subjects
  • Designing learning scenarios for IFL with programming tools

In the following workshop, pay attention to...
• Computational and algorithmic thinking
  • Incorporate them in learning scenarios, i.e. the learning outcomes related to the subjects, include them in these exercises.
  • Learning scenario:
  • If the learning activity is used as a single general example is required, but you can also encourage to design and implement these examples.
  • Use Scratch or micro:bit for at least one example.
  • Use the principles of IFL and learn tools of students.
  • Problem-solving elements (e.g., games, quizzes) are also be included.

In the following workshop, pay attention to...
• Reflection on conducted activity
  • Teachers review reflection is very important
  • Share more detailed reflections, regularly about achieving the learning outcomes related to algorithmic and computational thinking
  • Identify the number of students with whom the activity was performed.
  • Describe how your students have conducted these activities.
  • Describe the achievements of all learning outcomes in the scenario.
  • Describe the overall changes to the scenario before the next intervention.

Questions
•
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

- 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)
<table>
<thead>
<tr>
<th>Topic/Sub-topics</th>
<th>Learning Objectives</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. INQUIRY BASED LEARNING</strong></td>
<td><em>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.</em></td>
<td>Learners discuss concepts of Inquiry Based Learning and describe an example of lesson (group activity).</td>
</tr>
</tbody>
</table>
| 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

Workshop 3: Games and tools for programming
Session 1: Inquiry Based Learning

Agenda
- Introduction to Inquiry Based Learning
- Project-based learning as part of ILT
- Inquiry and learning with games

Inquiry Based Learning - basics
- An active approach to learning
- Includes problem-based learning or finding answers for addressed questions
- Develops self-reflection skills
- Basic teaching method is discussion that includes one or more research questions

Levels of Inquiry Based Learning
- Structured
  + Teacher directed
  + Teacher provides questions and guidance step-by-step instructions
- Guided
  + Teacher provides questions
  + Students take more responsibility
  + Teacher guides the inquiry
- Open
  + Students take the lead
  + Supportive role of the teacher

Investigation as approach to learning
- Students apply research questions to solve problems associated with the content of a particular subject
- Students use research methods and relevant practices with the aim of constructing new concepts or knowledge, to answer questions, or solve the problem
- Active involvement of students
- Encourages curiosity and creativity
- Level of inquiry can be customised to suit the students needs
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.

Example 1/2

Research questions:
- By collecting old paper, could 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 (research and extract relevant information).
- Investigate and interpret the paper recycling processes (research and extract essential information).
- Create a poster or graphically present both processes (define algorithms or sequence of commands).
- Establish and implement a calculation (write down the steps in the calculation process).
Group activity

Design an IBL activity
- create research questions
- describe learning outcomes

Discuss:
- research questions
- learning outcomes
- evaluation
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.)
- Code.org: https://studio.code.org/ (5.1.2019.)

Duration: 1 hour (45 minutes)
<table>
<thead>
<tr>
<th>Topic/Sub-topics</th>
<th>Learning Objectives</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. BASIC PROGRAMMING CONCEPTS</strong></td>
<td><em>Participants will be able to identify the basic programming concepts.</em></td>
<td>Learners explore existing educational games for learning programming, point out and discuss usage of programming concepts (group activity).</td>
</tr>
<tr>
<td>1.1. Introduction to basic programming concepts</td>
<td><em>Describe the basic programming concepts (sequence, branching, loop, variables)</em></td>
<td></td>
</tr>
<tr>
<td>1.2. Educational computer games for learning</td>
<td><em>Recognise the basic programming concepts in examples of different educational games</em></td>
<td></td>
</tr>
<tr>
<td>basic programming concepts</td>
<td><em>Analyse and compare existing examples</em></td>
<td></td>
</tr>
</tbody>
</table>
Presentation: Basic programming concepts

**Agenda**
- Introduction - Programming languages
- Basic programming concepts
- Programming concepts in different education games

**Programming languages**
- Languages used to write computer programs
- Each programming language has:
  - Alphabet - characters set used
  - Syntax - rules of sentence formation
  - Semantics - describes the purpose, meaning, action performed by the commands
- Each programming language uses a set of words of special significance called the *keywords*

**Basic programming concepts**
1. Sequence
2. Loops (iteration)
3. Variables
4. Branching (if condition)

**Sequence**
- The commands in the program need to be specified in a certain order
- Examples of simple sequences in games:
  - Moving along the path
  - Collecting objects
  - Executing actions
  - ...
**Analysing examples**

Group activity

**Sequence – Example 1**

*Code.org: Artist*

https://code.org/inspectors/
hhtp://code.org/html/

**Sequence – Example 2**

*Code.org: Bee*

http://code.org/inspectors/
https://code.org/html/

**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 an action

**Loop – Example 1**

*Run Marco! Level 11*

**Variables**

- Variables are used to keep values (text, numbers, …) and reuse these values.
- Variables have a 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 useful for moving characters (number of steps, time, …)
  - number of lives, collected items, points, …
  - time left to finish the game
  - …
Analyzing examples

Group activity

Variables - Example 1

Code club: Scratch - Ghostbusters

Branching

- The program can take certain routes depending on the fulfilled conditions.
- Examples of using branching in games:
  - avoiding enemies
  - changing values of variables (end life, getting points)
  - showing and hiding characters/items
  - ending the game

Branching - Conditions and operators

- A path that has a fulfilled condition is selected

Branching - Example 1

Blockly Games: Bird

Branching - Example 2

Run Marco! Level 31
Exploring games

Choose one of the following games and explore it:

- Code org: https://studio.code.org/a/course
- Blocky games: https://workshop.gamanet.com/thang/en
- Run Marco: https://www.runmarco.com/tafvea/poonaar

Discuss chosen game and its possible application for learning the basic programming concepts.

Additional resources

- Scratch, https://www.getscratch.org/ (15.4.2015)
- Hopscotch: https://www.hopscotchapp.com (13.3.2015)
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.)
### Duration: 2 hours (90 minutes)

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

Agenda
- Wider meaning of computational thinking (CT)
- Development of CT through Scratch games and stories: Concepts, practice, perspective
- The role of the Scratch community
- Examples of digital stories and games within the Scratch community

What is Computational Thinking?
- There are many definitions of computational thinking, one of them: CT combines critical thinking skills with the power of computing to make decisions or find solutions.
- Skills needed to solve an equation, plan a project, develop an outline for a writing assignment, etc. include important problem-solving competencies that students need throughout their lifetimes
- CT can amplify problem-solving skills needed to address authentic, real-world issues.

Computational Thinking

CT Operational Definition
Developed by the CT4E and C4ET, which evolved with inputs from higher education, industry, and K-12 educators.
CT is a problem-solving process that includes (but is not limited to) the following characteristics:
- Formulating problems in a way that enables us to use a computer and other tools to help solve them.
- Structuring problems by organizing and analyzing data.
- Representing problems through abstractions such as models and simulations.
- Automating solutions through algorithms (a series of ordered steps), and identifying, analyzing, and implementing effective solutions with the goal of achieving the most efficient and effective combination of steps and resources.
- Overarching and transforming this problem-solving process to a wide variety of problems.

CT in the Classroom

Elementary school
- Data collection
- Algorithms and procedures
- CT in context
- Introducing algorithms
- Logical reasoning

Middle school
- Problem decomposition
- CT in context
- Algorithms
- Logical reasoning

High school
- Animation
Computational thinking

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 scenarios and approaches of CT are you generating?
  - Linking concepts and approaches of CT with the learning objectives of the selected subject.
- Personal 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 programming language – educational programming environment
- MIT Media Lab, 2003 – first prototype
- 2007 public launch
- Translated in many languages
- Scratch 1 (January 2019)

Scratch - teachers point 1/2

Report: ‘If ‘90s argued that programming languages should have:
- a “low floor” (easy target stories)
- a “high ceiling” (opportunities to create increasingly complex projects)

“Social skills” (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 result sources
- “Children knows more than teachers”
- What else?

How to start with Scratch?

Guided start:
- https://code.org/act/activity-cards
- Scratch Wiki
- Tutorials at Code Club and CS First
- ScratchEd: lots of support, resources
- http://scratcheds.uc.howard.edu/resources/all
- Creative computing curriculum:
  http://scratchedu.co.howard.edu/resides/
Let's Scratch: Individual activity

Let's Scratch (searching, remixing): Pair activity

Search for good project (games and stories) in Scratch:
- Search with your own keyword
  - Enter your keyword
  - Find the project
  - Click on the project
- Look for images
- Find the project
- Click on the project
- Download the project
- Open the project
- Customize the project

Remix examples:
- Choose an example:
- Play the example
- Modify the example
- Add a new character
- Change the background
- Add a new sound
- Add a new animation
- Add a new script

Scratch "Unstuck" strategies:
1. Read through your code
2. Experiment with your code
3. Search for examples
4. Work with others
5. Be persistent

Lesson from Scratch workshops:
- Our assumptions about children's skills in technology are rather wrong.
- Children don't need to learn everything we (teachers) think they need to know.
- We are not supposed to know everything about programming.
- A logistic strategy: Guide on the side
- Power of vulnerability:
  - Children learn from each other; we learn from them.
Explore Escape Rooms

- Search for Escape Room projects.
- Select 1-2 “good” Escape Rooms.
- Examine the code.
- Why is this project good?
- Add to your toolkit (Kamol).

Connecting Scratch with natural sciences

- TEALEF project site: https://sites.google.com/a/tealef-project.eu/tealef-project/
- Topics: biodiversity, ecosystems, invasive species
- Brian’s ecology game: https://越南.hko.hk/d/23b0.png

Scratch workshop ideas

Group activity

Workshop - My Scratch day

Think and discuss about what is of interest in your group:
- What is your idea for Workshop in Scratch?
- Explain to other groups:
- Which topics do you select for 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 messages through is difficult
- Need better teachers
- Programming games in Scratch is fun!

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.
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


Duration: 3 hours (135 minutes)
<table>
<thead>
<tr>
<th>Topic/Sub-topics</th>
<th>Learning Objectives</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION INTO VISUAL PROGRAMMING WITH SCRATCH</td>
<td>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.</td>
<td>Learners (in pairs) inspect prepared stories/games in Scratch and comment them.</td>
</tr>
<tr>
<td>1.1. Testing Scratch examples</td>
<td>Test already prepared examples</td>
<td>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.</td>
</tr>
<tr>
<td>1.2. Short introduction to Scratch elements</td>
<td>Explore how Scratch works - how to start, where and what are blocks, how to move blocks</td>
<td></td>
</tr>
<tr>
<td>2. CREATING PROJECTS IN SCRATCH</td>
<td>Participants will be able to create some projects with the help of lecturers’ instructions and recognize important programming concepts through different activities.</td>
<td>Learners with the help of teachers create few examples in Scratch.</td>
</tr>
<tr>
<td>2.1. Creating a presentation-story</td>
<td>Create a presentation-story in Scratch</td>
<td>Learners create their own simple project – story in Scratch.</td>
</tr>
</tbody>
</table>
Presentation: Introduction into visual programming with Scratch

Agenda
- Short introduction – how to work in Scratch (how to start, where and what are blocks, how to move them)
- Creating new projects together (participants follow instructions)
- Participants create their own projects - stories.

Creative computing is about:
- CREATIVITY
- ENVIRONMENT
- COMPUTING

Scratch is a free programming language where you can create your own interactive stories, games, and animations.

Join Scratch
Interactive activity

Creating Scratch accounts
- https://scratch.mit.edu/

Creating in Scratch environment
- Click on the "Create" tab located at the top left of the browser to start a new project.
- "Start creating" button.
Introduction in Scratch environment

1. Blocks palette:
   - Actions, Deluxe, Score, Sound, Pen, Blocks, Score, Pen, Deluxe, Score
   - Turning Operators, Move Blocks
2. Coding area
3. Stage area
4. Different sprites
5. Different backgrounds

Creating together in Scratch (Step-By-Step):
Getting started with Scratch 1/10

FIRST STEP-BY-STEP ACTIVITY

- CONSTRUCT BLOCKS
- MAKE BLOCKS
- INTRODUCTION

Description:
- Select one block from the palette
- Click on the block to make the character move

MOTION blocks

Creating together in Scratch (Step-By-Step):
Getting started with Scratch 2/10

Description:
- Drag the corner block into the stage area
- Click on the block to make the character move

MOTION blocks

Creating together in Scratch (Step-By-Step):
Getting started with Scratch 3/10

Description:
- Click the block and drag it to another position
- Type in the code to change the number
- Then, place the block as before

MOTION blocks

Creating together in Scratch (Step-By-Step):
Getting started with Scratch 4/10

Description:
- Drag the corner block and drag it to another position
- Type in the code to change the number
- Then, place the block as before

MOTION blocks

Creating together in Scratch (Step-By-Step):
Getting started with Scratch 5/10

Description:
- Drag a block named "Music Block" and place it in the stage area
- Click on the block to make the character move

MOTION blocks

Creating together in Scratch (Step-By-Step):
Getting started with Scratch 6/10

Description:
- Drag a block named "Green Flag" and place it in the stage area
- Click on the block to start whenever you want to start again

EVENT blocks
Creating together in scratch (Step-By-Step): Answer – Question 1

Creating together in scratch (Step-By-Step): Answer – Question 2

FOURTH
STEP-BY-STEP ACTIVITY

FIFTH
STEP-BY-STEP ACTIVITY

SIXTH
STEP-BY-STEP ACTIVITY

Creating together in scratch (Step-By-Step): Create a Pong Game 1/6

Creating together in scratch (Step-By-Step): Create a Pong Game 2/6

Creating together in scratch (Step-By-Step): Create a Pong Game 3/6

Background and Sprites

INFINITY LOOP

MOTION, EVENT, CONTROL
blocks
TASK – Create a game

1. Create a background
2. Use a sprite (ball) from the library
3. Add two paddles
4. Make a game

TASK – Create your own story

1. Choose a background
2. Choose sprites from the library that will represent you
3. Add a picture of your hometown
4. Make a story

Questions

Create Stories

Individual activity

Make your own game

Pair activity
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=hXZOGFal6vc&t=16s (4.1.2019.)

Duration: 1 hour (45 minutes)
<table>
<thead>
<tr>
<th>Topic/Sub-topics</th>
<th>Learning Objectives</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. HOW TEACHERS CAN INTRODUCE PROGRAMMING IN THE CLASSROOM</td>
<td>Participants will be able to recognize the elements and the process of computational thinking and programming.</td>
<td>Learners explore and analyse applications of GBL tools within the class in order to point out benefits of introduction of computational thinking and programming.</td>
</tr>
<tr>
<td>FROM THEIR PERSPECTIVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1. Cycles for learning about how to code with focus on computational thinking</td>
<td>Explore the concepts of learning programming from teacher’s perspective</td>
<td></td>
</tr>
<tr>
<td>2. USING GAME BASED LEARNING TOOLS WITH ELEMENTS OF CODING IN THE CLASSROOM</td>
<td>Participants will be able to recognize the importance and the concept of collaborative games with coding.</td>
<td></td>
</tr>
<tr>
<td>2.1. Video presentation and discussion of game based learning tool Scottie Go!</td>
<td>Understand the role of GBL tools for coding and in development of computational thinking</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Introduce a way to incorporate technology and digital tools in engaged way</td>
<td></td>
</tr>
</tbody>
</table>
Presentation: Implementing Computational Thinking and programming with GBL tools

Workshop 3: Games and tools for programming
Session 5: Implementing computational thinking and programming with GBL tools

Agenda
- How to introduce programming in your classroom from teachers perspective
- Introducing GBL tools with elements of coding in the classroom
- “How to code

“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 competence guidelines for coding
- Risk to create opposite effect on pupils

Possible solutions
- Game Based Learning
- Flipped Classroom
- Co-operative and Individual Cycles

Learning programming in primary school
Approaches:
- Teacher lead and information oriented
- Student lead and skill development oriented
Teacher role:
- Educator that leads the class from the classroom front
- Facilitator who encourages the class to think and question the world around students

How to make a change?
- Cooperation games, no technology
- Focus on team work and cooperation, to achieve educational goal
- Individual time to master all kids can pair...
- Focus on technology
- Team work, create something new, gamification...
- Simplification, removing the parts of the classroom (e.g. competition writing lesson)

Remember: You have “superheroes” in your class...
Methodology Approach

Possible tools
- Scottie Go
- Code Combat
- Puzzles
- Scratch
- Code Monkey
- Minecraft
- Micro:bit
- Lego
- Microduino
- Python
- Java
- Ruby

Possible paths
- Scottie Go
- Scratch
- Micro:bit
- Python

Scottie Go! Game
- Scottie Go! is an interactive puzzle-based mobile game that uses a similar block-based coding approach as Scratch.
- The basic idea of this mobile game is to help a friendly alien Scottie to get back home to the outer space.
- The game is a combination of cardboard tiles, which are used by the players to create coding instructions, and a mobile application, that sets tasks and scans the proposed code solutions.

Scottie Go! and CT development

Video Presentation
Scottie Go for Computational Thinking

Discussing Scottie Go! vs. Scratch
Group activity
Scottie Gol vs. Scratch

- In small groups discuss the possibilities of Scottie Gol and Scratch in order to point out benefits for introduction of computational thinking in your class.
- Share your ideas with the teacher and the other groups.

Questions
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.)

Duration: 3 hours (135 minutes)
<table>
<thead>
<tr>
<th>Topic/Sub-topics</th>
<th>Learning Objectives</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. INTRODUCTION OF MICRO:BIT</strong></td>
<td>Participants will be able to describe the functionalities and features of micro:bit and recognize and compare basic micro:bit applications with Scratch.</td>
<td>Learners explore and analyse simple micro:bit applications and compare it with Scratch projects.</td>
</tr>
<tr>
<td>1.1. Introduction of micro:bit as tool for programming (basic concepts, how it differs from Scratch)</td>
<td>Explore the functionalities and features of micro:bit, micro:bit development environment, and basic event driven programming</td>
<td></td>
</tr>
<tr>
<td><strong>2. HOW TO APPLY MICRO:BIT IN DIFFERENT SCHOOL SUBJECTS</strong></td>
<td>Participants will be able to introduce basic micro:bit applications in their classroom.</td>
<td>Learners explore and analyse micro:bit projects and explore possibility to apply them in their classes for active participation of their students (group activity).</td>
</tr>
<tr>
<td>2.1. Demonstration of using simple micro:bit application for different school subjects</td>
<td>Analyse existing applications suitable for different subjects Analyse examples of project-based learning using micro:bit</td>
<td></td>
</tr>
<tr>
<td>2.2. Developing and adopting micro:bit application for different school subjects</td>
<td>Be able to alter micro:bit code in order to better match learning outcome</td>
<td></td>
</tr>
</tbody>
</table>
Presentation: micro:bit in classroom

Workshop 3: Games and tools for programming
Session 6: micro:bit in classroom

Agenda
- Introduction of micro:bit as tool for programming
- Creating basic micro:bit applications for different school subjects

Introduction to micro:bit programming
- how it differs from Scratch

From Scratch to micro:bit
- micro:bit is a very small, cheap, open-source board designed for engaging students with coding and computing education.
- It has many features, including an LED display, buttons, and a motion sensor.
- Teachers can connect it to Scratch and build creative projects that combine the map of the digital and physical worlds.

https://microbit.org/

From Scratch to Python
Strategies for engaging students 1/2

- Creating educational applications for tools which are familiar for children.
  - for example, class projects using mobile phones.
  - Enabling children to engage with a broader audience.
  - For example, blogs can encourage sharing, invite children to share work and invite reflection.

- Facilitating links with local organisations:
  - For example creating MIT badges to contribute to a local museum.

Strategies for engaging students 2/2

- Encouraging children to create digital artefacts:
  - for example, allowing children to make their own films using simple hardware and software.
  - Integrating digital literacy into children’s research skills:
  - Using social bookmarking sites for children to form groups and add their own bookmarks and evaluate those of others on a particular emergent topic.

Student involvement

- Student involvement may vary from engagement as contributors, through raising awareness of digital literacy, to helping digital literacy skills to others.
- Activities might include:
  - peer to peer learning, with more adept students supporting the digital literacy development of others.
  - Student leadership and learning about digital literacy as well as teachers, and in some cases parents and wider community.
  - Students mentoring other teachers - recognizing that there are aspects of digital literacy and ICT use of which students are more familiar.

Getting started with micro:bit

Individual activity
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)

<table>
<thead>
<tr>
<th>ASSIGNMENT STEPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>23. Choose a school subject – plan the activities that will be carried out in your class next month.</td>
</tr>
<tr>
<td>24. Use the Learning Scenario Template form (Annex 1) for textual version and LePlanner for graphical version of your scenario.</td>
</tr>
<tr>
<td>25. Specify the Learning outcomes:</td>
</tr>
<tr>
<td>- state general learning outcomes related to the course that will include problem teaching and logical tasks</td>
</tr>
<tr>
<td>- state learning outcomes oriented on algorithmic thinking</td>
</tr>
<tr>
<td>26. Describe Aim and tasks of teaching and give a Short description of activities.</td>
</tr>
<tr>
<td>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).</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td><strong>27.</strong> Specify the <strong>Keywords, Correlation and Interdisciplinarity</strong> with other courses or topics, and <strong>Duration of activities.</strong></td>
</tr>
<tr>
<td><strong>28.</strong> Point out <strong>Learning and teaching strategies and methods.</strong> Specify the <strong>Teaching forms:</strong> use the principles of IBL and team work of students. Problem solving elements (logic games, quizzes, ...) can also be included.</td>
</tr>
<tr>
<td><strong>29.</strong> Choose <strong>Tools</strong> 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 <strong>Resources/materials</strong> which will be required for the teacher as well as for students.</td>
</tr>
<tr>
<td><strong>30.</strong> Use <strong>Game/story template</strong> (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.</td>
</tr>
<tr>
<td><strong>31.</strong> Elaborate the <strong>Teaching summary</strong> as <strong>Motivation (Introduction), Implementation and Evaluation (Reflection).</strong> 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.</td>
</tr>
<tr>
<td><strong>32.</strong> In <strong>Annexes</strong> box provide a link to the graphical version of the learning scenario in LePlanner. You will add link to the developed online story later.</td>
</tr>
<tr>
<td><strong>33.</strong> <strong>Examples and game references</strong> box should contain a link to the Scratch story and to the other sources you will use for the activities.</td>
</tr>
</tbody>
</table>

**FOLLOW-UP ACTIVITIES**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>9.</strong> 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.</td>
<td></td>
</tr>
<tr>
<td><strong>10.</strong> Upload your final version of learning scenario with story description considering mentor’s suggestions and corrections.</td>
<td></td>
</tr>
<tr>
<td><strong>11.</strong> 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.</td>
<td></td>
</tr>
<tr>
<td><strong>12.</strong> Implement the last part (<strong>Reflection and evaluation</strong>) of your learning scenario in the class with your students and play the final version of the game together with them.</td>
<td></td>
</tr>
</tbody>
</table>
| **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 planned learning outcomes, both general and oriented on algorithmic thinking.
  - Define what you would like to change before the next implementation of the scenario. |
Part III: Annexes
## Annex 1: Learning scenario template

<table>
<thead>
<tr>
<th>Learning Scenario Title</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Course/Grade</td>
<td></td>
</tr>
<tr>
<td>Learning Outcomes</td>
<td>General learning outcomes</td>
</tr>
<tr>
<td></td>
<td>Specific LO oriented on algorithmic thinking</td>
</tr>
<tr>
<td>Aim, Tasks and Short Description of Activities</td>
<td></td>
</tr>
<tr>
<td>Keywords</td>
<td></td>
</tr>
<tr>
<td>Correlation and Interdisciplinarity</td>
<td></td>
</tr>
<tr>
<td>Duration of Activities</td>
<td></td>
</tr>
<tr>
<td>Learning and Teaching Strategy and Methods</td>
<td></td>
</tr>
<tr>
<td>Teaching Forms</td>
<td></td>
</tr>
<tr>
<td>Tools</td>
<td></td>
</tr>
<tr>
<td>Resources/Materials for the Teacher</td>
<td></td>
</tr>
<tr>
<td>Resources/Materials for the Students</td>
<td></td>
</tr>
<tr>
<td>Teaching summary</td>
<td>Motivation-Introduction</td>
</tr>
<tr>
<td></td>
<td>Implementation</td>
</tr>
<tr>
<td></td>
<td>Reflection and evaluation</td>
</tr>
<tr>
<td>Annexes</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Examples and game references</td>
<td></td>
</tr>
</tbody>
</table>
Annex 2: Game/story scenario template

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

**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.