



Games for Learning Algorithmic Thinking

Workshop Syllabus

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

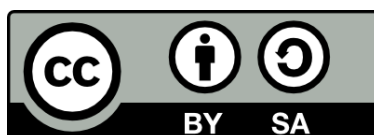
Martina Holenko Dlab (*University of Rijeka, Department of Informatics*)

Contact

glat@inf.uniri.hr

Licence

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).



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, 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	5
Part II: Learning Outcomes and Topics for F2F Workshops	8
Workshop 1: Game Based Learning (GBL) and Unplugged Activities	9
Workshop schedule.....	10
Session 1: Game based learning	12
Session 2: GBL with unplugged activities	14
Session 3: Using Web 2.0 tools for creating content for unplugged activities.....	16
Session 4: Designing learning scenarios	18
Session 5: Designing learning scenarios using a graphical tool	20
Session 6: Designing learning scenarios for unplugged activities.....	22
Workshop 2: Problem Based Learning (PBL), Online Quizzes and Logical Tasks	24
Workshop schedule.....	25
Session 1: Introduction to Problem Based Learning (PBL)	27
Session 2: Problem-solving in logical games.....	29
Session 3: Online quizzes and logical tasks.....	31
Session 4: Using Web 2.0 tools for creating quizzes and logical tasks.....	33
Session 5: Designing learning scenarios for logical tasks	35
Workshop 3: Games and Tools for Programming.....	37
Workshop schedule.....	38
Session 1: Inquiry Based Learning.....	40
Session 2: Basic programming concepts	42
Session 3: Learning programming with games and stories	44
Session 4: Introduction into visual programming with Scratch	46
Session 5: Implementing Computational Thinking and programming with GBL tools	48
Session 6: micro:bit in classroom.....	50
Session 7: Designing learning scenarios.....	52
Part III: Annexes	54
Annex 1: Learning scenario template.....	55
Annex 2: Game/story scenario template	57





Games for Learning
Algorithmic Thinking



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





Games for Learning
Algorithmic Thinking



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)





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

Session 1: Game based learning

Expected Learning Outcomes

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

Teaching Methods/Approaches

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

Sources of Training Materials

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

Duration: 3 hours (135 minutes)





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





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

Session 2: GBL with unplugged activities

Expected Learning Outcomes

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

Teaching Methods/Approaches

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

Sources of Training Materials

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

Duration: 1 hour (45 minutes)





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





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

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

Expected Learning Outcomes

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

Teaching Methods/Approaches

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

Sources of Training Materials

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

Portals with tools and resources for teachers:

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

Web 2.0 tools:

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





Duration: 3 hours (135 minutes)

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





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

Session 4: Designing learning scenarios

Expected Learning Outcomes

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

Teaching Methods/Approaches

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

Sources of Training Materials

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

Duration: 2 hours (90 minutes)





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





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

Session 5: Designing learning scenarios using a graphical tool

Expected Learning Outcomes

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

Teaching Methods/Approaches

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

Sources of Training Materials

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

Duration: 2 hours (90 minutes)





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





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

Session 6: Designing learning scenarios for unplugged activities

Instructions for the participants

Expected Learning Outcomes

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

Individual Assignment:

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

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

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

Duration: up to 2 months for the whole assignment

ASSIGNMENT STEPS	
1.	Choose a school subject – plan the activities that will be carried out in your class next month.
2.	Use the Learning Scenario Template form (Annex 1) for textual version and LePlanner for the graphical version of your scenario.
3.	Specify the Learning outcomes : <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 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)





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

Session 1: Introduction to Problem Based Learning (PBL)

Expected Learning Outcomes

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

Teaching Methods/Approaches

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

Sources of training materials

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





Duration: 1 hour (45 minutes)

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





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

Session 2: Problem-solving in logical games

Expected Learning Outcomes

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

Teaching Methods/Approaches

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

Sources of training materials

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

Duration: 3 hours (135 minutes)





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



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

Session 3: Online quizzes and logical tasks

Expected Learning Outcomes

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

Teaching Methods/Approaches

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

Sources of training materials

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

Web 2.0 tools:

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

Duration: 3 hours (135 minutes)





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





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

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

Expected Learning Outcomes

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

Teaching Methods/Approaches

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

Sources of training materials

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

Web 2.0 tools:

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

Duration: 3 hours (135 minutes)





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





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

Session 5: Designing learning scenarios for logical tasks

Instructions for the participants

Expected Learning Outcomes

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

Individual Assignment:

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

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

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

Duration: up to 2 months for the whole assignment

ASSIGNMENT STEPS	
12.	Choose a school subject – plan the activities that will be carried out in your class next month.
13.	Use the Learning Scenario Template form (Annex 1) for textual version and LePlanner for the graphical version of your scenario.
14.	Specify the Learning outcomes : <ul style="list-style-type: none"> - state general learning outcomes related to the course that will include problem teaching and logical tasks - state learning outcomes oriented on algorithmic thinking
15.	Describe the Aim and tasks of teaching and give a Short description of the activities . Plan the activities that will encourage your students for seeking the information, critical and logical thinking as well as collaborating while solving the problem. The activities should include work on computer/tablet/smartphone (not only unplugged activities).
16.	Specify the Keywords, Correlation, and Interdisciplinarity with other courses or topics, and the Duration of activities .
17.	Point out the Learning and teaching strategies and methods . Specify the Teaching forms : combine individual and group work; since this is a problem teaching, collaborative learning should be included.
18.	Choose Tools or games that will be used for quizzes or logical tasks on computer/tablet/smartphone for at least one example.





	Point out all Resources/materials which will be required for the teacher as well as for students.
19.	Elaborate the Teaching summary as Motivation (Introduction), Implementation and Evaluation (Reflection) . This part develops in details previously mentioned short description of activities. Logical tasks or quizzes can be used in each part of teaching summary (you will add links to the developed online tasks later).
20.	Create suitable quizzes and/or logical tasks with chosen tools. Pay attention to the copyright for images, videos, and other materials collected from the web. Photographing your students requires written parents' consent.
21.	In Annexes box provide examples and tasks you have created by yourself as well as link to the graphical version of the Learning scenario in LePlanner. Links should be direct to the created tasks prepared for solving by students (not to the tasks open in editor).
22.	Examples and game references box should contain sources you will use for the activities.
FOLLOW-UP ACTIVITIES	
5.	Upload your completed first version of learning scenario to the Moodle e-course. Mentor will review and correct your scenario.
6.	Upload your final version of learning scenario considering mentor's suggestions and corrections.
7.	After mentor's approval, implement your learning scenario in the class with your students.
8.	Post a reflection on conducted activities in the forum: <ul style="list-style-type: none"> • 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 planned learning outcomes, both general and oriented on algorithmic thinking. • Define what you would like to change before the next implementation of the scenario.





Games for Learning
Algorithmic Thinking



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)





Workshop 3 – Games and Tools for Programming

Session 1: Inquiry Based Learning

Expected Learning Outcomes

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

Teaching Methods/Approaches

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

Sources of Training Materials

- 4 Phases of Inquiry Based Learning - A Guide For Teachers: <https://www.teachthought.com/pedagogy/4-phases-inquiry-based-learning-guide-teachers/> (5.1.2019.)
- Inquiry Based Learning in the Science Classroom: <https://www.edutopia.org/practice/inquiry-based-learning-science-classroom> (5.1.2019.)
- What is Enquiry-Based Learning (EBL)?: <http://www.ceebl.manchester.ac.uk/eb/> (5.1.2019.)

Duration: 1 hour (45 minutes)





Topic/Sub-topics	Learning Objectives	Evaluation
1. INQUIRY BASED LEARNING	<i>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.</i>	Learners discuss concepts of Inquiry Based Learning and describe an example of lesson (group activity).
1.1. Introduction to Inquiry Based Learning	Describe principles of Inquiry Based Learning Apply concepts of Inquiry Based Learning	
1.2. Project Based Learning	Compare Inquiry Based Learning with Project Based Learning. Analyse and compare existing examples of using Inquiry Based Learning in different school subjects	





Workshop 3 – Games and Tools for Programming

Session 2: Basic programming concepts

Expected Learning Outcomes

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

Teaching Methods/Approaches

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

Sources of Training Materials

Games:

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

Duration: 1 hour (45 minutes)





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



Workshop 3 – Games and Tools for Programming

Session 3: Learning programming with games and stories

Expected Learning Outcomes

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

Teaching Methods/Approaches

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

Sources of Training Materials

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





Duration: 2 hours (90 minutes)

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



Workshop 3 – Games and Tools for Programming

Session 4: Introduction into visual programming with Scratch

Expected Learning Outcomes

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

Teaching Methods/Approaches

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

Sources of Training Materials

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

Duration: 3 hours (135 minutes)





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



Workshop 3 – Games and Tools for Programming

Session 5: Implementing Computational Thinking and programming with GBL tools

Expected Learning Outcomes

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

Teaching Methods/Approaches

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

Sources of Training Materials

- Computational Thinking: <https://code.org/curriculum/course3/1/Teacher> (4.1.2019.)
- Scottie Go! for Computational Thinking <https://www.youtube.com/watch?v=hXZOGFal6vc&t=16s> (4.1.2019.)

Duration: 1 hour (45 minutes)





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



Workshop 3 – Games and Tools for Programming

Session 6: micro:bit in classroom

Expected Learning Outcomes

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

Teaching Methods/Approaches

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

Sources of Training Materials

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

Duration: 3 hours (135 minutes)





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



Workshop 3 – Games and Tools for Programming

Session 7: Designing learning scenarios

Instructions for the participants

Expected Learning Outcomes

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

Individual Assignment:

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

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

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

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

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

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

ASSIGNMENT STEPS	
23.	Choose a school subject – plan the activities that will be carried out in your class next month.
24.	Use the Learning Scenario Template form (Annex 1) for textual version and LePlanner for graphical version of your scenario.
25.	Specify the Learning outcomes : <ul style="list-style-type: none"> - state general learning outcomes related to the course that will include problem teaching and logical tasks - state learning outcomes oriented on algorithmic thinking
26.	Describe Aim and tasks of teaching and give a Short description of activities . Plan the activities that will encourage your students for seeking the information, critical and logical thinking as well as collaborating while solving the problem according to the principles of inquiry based learning (IBL). The activities should include a game on computer/tablet/smartphone (not only unplugged activities).





27.	Specify the Keywords, Correlation and Interdisciplinarity with other courses or topics, and Duration of activities .
28.	Point out Learning and teaching strategies and methods . Specify the Teaching forms : use the principles of IBL and team work of students. Problem solving elements (logic games, quizzes, ...) can also be included.
29.	Choose Tools and games that will be used on computer/tablet/smartphone for at least one example. Mandatory is to use of at least one story or game example developed in Scratch or micro:bit. Point out all Resources/materials which will be required for the teacher as well as for students.
30.	Use Game/story template (Annex 2) to prepare the description of the story or game. For now, prepare just a draft (fill in elements: Title of the game, Type (Scratch or micro:bit), Course/Grade, Learning outcomes, Goal of the game). Pay attention to the copyright for images, videos, and other materials collected from the web. Photographing your students requires written parents' consent.
31.	Elaborate the Teaching summary as Motivation (Introduction), Implementation and Evaluation (Reflection) . This part describes in detail previously mentioned short description of activities. It should be based on IBL and activities with students for designing a story/game.
32.	In Annexes box provide a link to the graphical version of the learning scenario in LePlanner. You will add link to the developed online story later.
33.	Examples and game references box should contain a link to the Scratch story and to the other sources you will use for the activities.
FOLLOW-UP ACTIVITIES	
9.	Upload your completed first versions of learning scenario and draft of the game/story description to the Moodle e-course. Mentor will review and correct your scenario and story description.
10.	Upload your final version of learning scenario with story description considering mentor's suggestions and corrections.
11.	After mentor's approval, implement the part of learning scenario about the story development in the class and design game with your students. Complete the game/story description and upload it in the Moodle e-course. Mentor will provide you with the finished story/game for your learning scenario.
12.	Implement the last part (Reflection and evaluation) of your learning scenario in the class with your students and play the final version of the game together with them.
13.	Post a reflection on conducted activities in the forum: <ul style="list-style-type: none"> • 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

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





Annexes	
Examples and game references	





Annex 2: Game/story scenario template

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

Appendix - Instructions for Storytelling in Scratch

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

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

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





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

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

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

