

Didactic Scenario

1. Title

Exploring the World of Code: Creating Digital Stories

2. Keywords

Code, Programming, Digital History, Creativity, Technology, Interactivity, Fiction, Screenwriting, Digital Tools, Exploration, Code Education

3. Basic information

STEAM Subject: TECHNOLOGY

Typical interaction time with the instructional scenario in teaching hours for in-school work: 3 hours

General description of the scenario:

<u>Phases</u>	<u>Stage</u>	<u>Time</u>
Introduction to Programming	Preparatory Stage	30 minutes
Creating Digital Stories	Implementation Stage	90 minutes
Presentation and Feedback	Evaluation Stage	60 minutes

Age group: 10-12 years old

Estimated difficulty level:

Very Easy	Easy	Moderate	Challenging	Very Challenging
		X		

Teaching resources

Materials:

- Computers or Tablets: To access online programming tools and create digital stories.
- Programming Software: Tools such as Scratch or other similar software suitable for children.
- Internet Access: To use online resources and tools.
- Headphones or Speakers: For students to hear sounds and music that they will use in their stories.
- Projector or Screen: For presenting digital stories in class.
- Paper and Pencils: For initial drawing and organizing their ideas.
- Pictures and Graphics: Materials or digital resources that students can use for their stories.
- Guides and Manuals: Material that will provide guidance for using the programming software.

School infrastructure:

- Computers or Tablets: Equipment with sufficient computers or tablets for all students, or at least in groups, to work on their digital projects.
- Good Internet Connection: Fast and reliable Internet connection to access online programming tools and other educational resources.
- Projector or Screen: For presenting student work and demonstrating programming tools in class.
- Adequate Workspace: Tables or desks that allow students to work in groups or individually comfortably.
- Headphones or Speakers: Space that allows the use of headphones or speakers to listen to the sounds and music that will be added to the digital stories.
- Online Learning Management Platform: If possible, using a platform like Google Classroom to submit projects and share information.

Additional material from external sources/online tools:

- Scratch (programming language) (<https://scratch.mit.edu/>)
Usage: Simple programming language that allows students to create programs for robots and other projects.
- Code.org (<https://code.org/>): A source of free programming lessons and interactive activities that encourage creativity and exploration.
- Storybird (<https://storybird.com/>): Online platform that allows students to create digital stories and integrate images with text.
- ED-ED "THINK LIKE A CODER"

This 10-episode series follows a girl, Ethic, and her robot boyfriend, Hedge, as they try to save the world. In their effort, the two friends will be asked to solve a series of programming problems. They embark on a quest to collect three objects and must work their way through a series of programming puzzles.

(<https://www.youtube.com/watch?v=KFVdHDMcepw&list=PLJicmE8fK0EiFngx7wBddZDzxogj-shyW>)

Differentiated instruction for students with different abilities and learning styles in the same class:

- Custom Activities. Create activities with different levels of difficulty. Students with more advanced programming knowledge can experiment with more complex stories, while beginners can focus on basic structures and characters.
- Tool Options. Offer a variety of tools and platforms for building the stories (eg Scratch, Storybird, Google Slides) so students can choose the tool that works best for them.
- Teamwork: Create groups of students with different abilities so that the most able support others, promoting collaboration and mutual learning.
- Individual Support: Provide individualized guidance to students who need more help by offering extra material or homework guidance.
- Self-assessment: Encourage students to assess their own work, enabling them to recognize their progress and set targets for improvement.

Developed by: Development Center of Thessaly

4. Educational Problem

This scenario solves the problem of students' difficulty understanding and applying mathematical and technological concepts, as well as their lack of interest in programming. Many students often perceive math and technology as abstract and difficult subjects, which prevents them from exploring or applying them creatively. Through creating digital stories, students have the opportunity to combine learning with creation, using programming to express their ideas. This not only enhances understanding of math and technology concepts but also promotes creativity, collaboration and problem-solving skills, making learning more enjoyable and experiential.

5. Learning Objective (-s)

1. Understanding Programming: Students will gain basic programming knowledge and understand how to use digital tools to create stories.
2. Creative Expression: They will develop their creative abilities, combining math and art in their digital storytelling.
3. Problem Solving Skills: Students will learn to identify and solve problems during the creation process, improving their critical thinking.
4. Collaboration and Communication: Through group work, students will enhance collaboration and communication skills, learning to work effectively with others.
5. Understanding Mathematical Concepts: They will recognize how mathematical concepts, such as numbers and patterns, can be applied to digital storytelling.
6. Self-Assessment and Feedback: They will be encouraged to assess their own work and provide feedback to their peers, enhancing their ability to learn from experience.

6. Phases of the Scenario

Phase 1

Title: Introduction to Programming

Indoor	Outdoor	Mixed
X		

Phase duration in minutes: 30 minutes

Detailed description of the scenario phase: In the 1st phase of the scenario, students are introduced to the basic concepts of programming in an interactive and participatory way. The instructor begins with a presentation that breaks down fundamental programming terms, such as statements, variables, and loops, using simple examples and visuals. Students participate in activities that include programming games, where they are asked to recognize and execute basic commands through games or puzzles. This process helps them understand the logic behind programming and prepare them to create their own digital stories afterwards.

Activity Sheets:

Activity Sheet - Phase 1: Introduction to Programming

Purpose: To understand the basic concepts of programming.

Activity 1: Identifying Commands

Directions: Read the following programming commands and match them with their description.

Principle
Repetition
If...then
Termination

Descriptions:

- A) It is used to start a program.
- B) Allows a command to be executed multiple times.
- C) It is used to test whether a condition is true.
- D) It signals the completion of the program.

Activity 2: Create a Small Algorithm

Instructions: Think of a simple activity (eg how to make a coffee) and write the algorithm step by step:

Activity 3: Class Discussion

Directions: Answer the following questions in class:

What are the most interesting commands you learned today?
How do you think programming can help us in our daily lives?

Phase 2

Title: Creating Digital Stories

Indoor	Outdoor	Mixed
X		

Phase duration in minutes: 90 minutes

Detailed description of the scenario phase: In Phase 2 of the scenario, "Creating Digital Stories," students apply their programming knowledge to design and program their own digital stories. First, students work in groups to develop their ideas, defining the characters, plot, and setting of their story. Then, using programming tools like Scratch, students create their story, adding code to combine graphics, sounds, and dialogue. As they work, they are encouraged to experiment with different functions and commands, discovering how mathematical concepts can be incorporated into the creation process. This phase enhances students' creativity,

teamwork and problem-solving skills, while also preparing them for the presentation of their projects.

Activity Sheets: N/A

Phase 3

Title: Presentation and Feedback

Indoor	Outdoor	Mixed
X		

Phase duration in minutes: 60 minutes

Detailed description of the scenario phase: In the 3rd phase of the scenario, the students have the opportunity to present the digital stories they created to the rest of the groups and to the class. Each group explains the process they followed, the programming elements they used, and the importance of the mathematical concepts they incorporated into their story. The presentations are accompanied by an open discussion, where classmates provide feedback, recognizing the strengths of each story and suggesting improvements. The teacher guides the discussion, focusing on learning experiences and the connection between programming and creativity. This process builds students' self-confidence, promotes critical thinking, and encourages collaboration and mutual respect.

Activity Sheets: N/A

7. Evaluation Methodology

Participation Observation:

During the input and creation phases, the teacher can monitor the active participation of the students. Use an observation board to note participation, interaction, and support among team members.

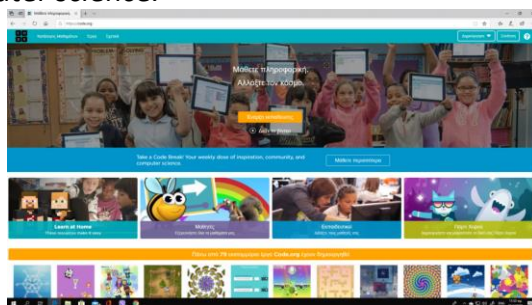
Evaluation of Creative Presentations:

In phase 3, evaluate digital story presentations based on criteria such as clarity, creativity, use of programming, and connection to mathematical concepts. Create a simple score column for the assessment.

8. Additional Resources for the teacher

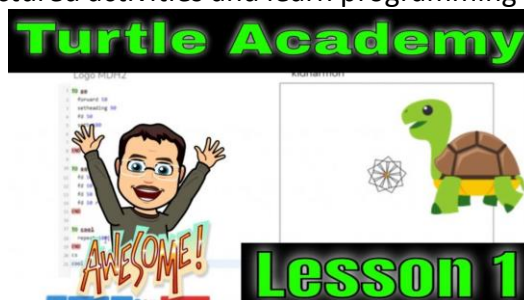
CODE.ORG (<https://code.org/>)

A non-profit initiative that aims to enable every student and every school to have the opportunity to learn computer science.



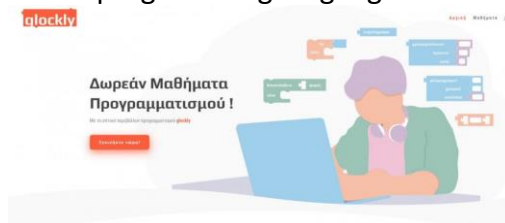
TURTLE ACADEMY (https://turtleacademy.com/?lang=el_EL)

Enter the amazing world of programming with the excellent Turtle Academy, now also available in Greek. Follow many structured activities and learn programming in an exciting way.



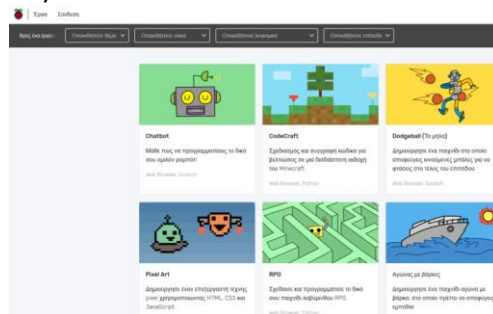
GLOCKLY (<https://glockly.com>)

glockly is based on Blockly, a visual programming language from Google with lessons in Greek.



RASPBERRY PI (<https://projects.raspberrypi.org/el-GR/projects>)

From the Raspberry Pi foundation, courses in scratch, python, html/css, etc. translated into Greek (a constantly growing list).



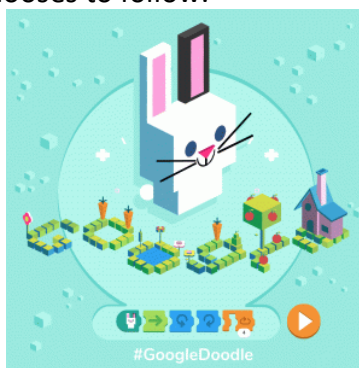
Blockly (<https://developers.google.com/blockly/>)

Google's Blockly is similar to the aforementioned Scratch but differs in that it makes it easier to code since kids can code in JavaScript, Python, PHP, Lua, Dart and other programming languages. In other words, by composing the blocks, this yields syntactically correct code in your programming language of choice.



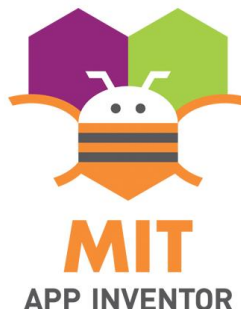
Celebrating 50 Years of Programming by Kids
(<https://www.google.com/logos/2017/logo17/logo17.html?hl=en>)

A game of simple programming and collecting carrots from rabbits was chosen by Google today to celebrate 50 years of programming languages for children, with today's google doodle. It is an interactive game, where young and old can have a good time, learning and having fun. A child who learns programming equips himself with skills that will be useful in the future, regardless of the profession he chooses to follow.



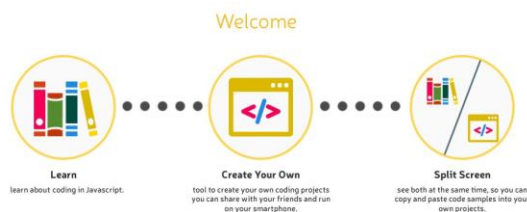
WITH App Inventor (<http://appinventor.mit.edu/>)

MIT App Inventor is a visual programming environment that allows anyone to create fully functional apps for smartphones and tablets. In short, with App Inventor they can have a simple app and run it on a mobile or tablet.



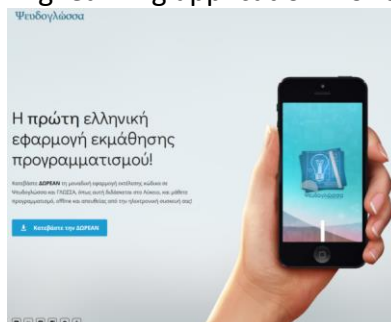
S2JS (<https://s2js.com/>)

S2JS (Scratch to JavaScript) teaches kids how to transfer the programming methods they learned in Scratch to the JavaScript programming language. It is mainly aimed at older children between the ages of 12 and 17 and assumes that they are already advanced in Scratch. S2JS shows kids how to achieve similar things in Javascript using Scratch examples and then walks them step-by-step through how to achieve the same result in Javascript. The result is that their applications can be easily used on computers as well as portable devices such as mobile phones.



PSEUDOGLOSSA (<https://pseudoglossa.com/#Home>)

The first Greek mobile programming learning application. For computers HERE.



EduBlocks (<https://edublocks.org/>)

For a smooth transition from block-based programming to Python, look no further! In this talk, 14-year-old Josh will introduce you to his project called EduBlocks, which is a drag-drop version of Python 3 that he created to help educators introduce programming languages like Python to younger kids. The goal of the project is to make the transition from block-based programs like Scratch to Python easier for students and teachers, as there is currently no solution that bridges this gap. Josh will share his journey so far with you, from how he came up with the idea when he was just 11 years old, the developments along the way, the exciting plans for the future and how schools today in over 72 different countries around the world they use EduBlocks on Raspberry Pi and micro:bit.