e-Book

“THE ROLE OF ICT TOOLS IN MOTIVATING LEARNING AND INCREASING THE QUALITY OF EDUCATION”

Project Nr: 2018-1-RO01-KA229-049348_1

- **LICEUL TEHNOLOGIC DACIA PITEȘTI, ROMÂNIA**
- **LICEO SCIENTIFICO STATALE "MICHELE GUERRISI", CITTANOVA, ITALY**
- **PANEVĖŽIO 5-OJI GIMNAZIJA, PANEVĖŽYS, LITHUANIA**
- **BURDUR MESLEKI VE TEKNIK ANADOLU LISESI, BURDUR, TURKEY**

"Material produced with the financial support of the European Commission. The content of this material is the exclusive responsibility of the authors, and the National Agency and the European Commission are not responsible for how the content of the information will be used. "

Contents

PREFACE 4

I. About STEM - Science, Technology, Engineering and Mathematics 5

II. Programming using the STEM concept - Science, Technology, Engineering and Mathematics, Scratch programming language 7

What is Scratch? 7

Scratch Programming Environment 8

The Stage 9

Sprite List 9

Blocks Tab 11

Scripts Area 12

Costumes Tab 13

Sounds Tab 15

Backdrops Tab 15

Sprite Info 16

Toolbar 17

Paint editor 2

Setting the Center of an Image 2

Setting Transparent Color 3

III. Programming language Visual Basic Application (VBA) 10

IV. Robot Game: Build and program an autonomous LEGO MINDSTORMS robot 18

What is Lego? 18

Lego Mindstorms EV3 History 18

Fundamental principles for LEGO (theory) 20

Lego Mindstorms EV3 parts 20

EV3 Education Core builds 23

Examples of games that can be used during LEGO 25

Lego Mindstorms EV3 programming 25

V. e-learning platforms 27

About e-learning 27

About Moodle Platform 27

About Platform wand. education 38

Circuit elements – Tinkercad 43

Passive circuit elements 43

Active circuit elements 44

Platform Tinkercad 44

Assemblies using tinkercad And ardiuino platforms 45

THE SOURCES OF INFORMATIONS 48

PREFACE

This eBook is a final result of our project Erasmus Plus **”The role of ICT tools in motivating learning and increasing the quality of education” - Project Nr: 2018-1-RO01-KA229-049348.**

You can find some useful information about:

1. Programming using the STEM concept - Science, Technology, Engineering and Mathematics, Scratch programming language.
2. Programming language Visual Basic Application (VBA)
3. Robot Game: Build and program an autonomous LEGO MINDSTORMS robot
4. Applications of physics in mechanics and electronics presented using ICT tools

Teachers from the different countries dedicated themselves to this project.

Many thanks to your great effort.

We hope this eBook will be a guide in your teaching journey.

I. About STEM - Science, Technology, Engineering and Mathematics

STEM is an educational concept based on the idea of educating students in the following four areas: science, technology, engineering and mathematics, using a multi-disciplinary and applied approach.

STEM is a learning model that integrates the four disciplines, coherent and based on real-world applications.[1]

STEM education is a topical concept that helps students understand abstract concepts through experimentation and play. The development of technology makes it increasingly clear that the near future means artificial intelligence, the Internet of Things, Robotics. The STEM concept makes students' learning integrated into the creative process.

Technology includes topics such as computer programming, analysis and design, architecture and medicine, and engineering can include topics such as electronics, robots and machines of the future, so the key term when talking about STEM education is integration.

STEM education differs from traditional education in that learning is more coherent and closer to real life, making it easier for students to understand abstract concepts.

This is how students develop critical thinking, analytical and methods such as problem solving, through project, collaborative, learning through experience give much better results from an early age and becomes much more attractive.

An advantage of STEM education is that students can control part of the learning process, are more involved and learn much easier.

Education must constantly adapt to prepare students for the jobs of the future, which do not currently exist.

Through the US ICT strategic partnership project, we aimed to develop these competencies of STEM education through all activities, to approach different ways of learning, to learn from the experiences of partners in this field, to integrate and implement concepts from different educational systems. .

All these are materialized in this e -BOOK which is an important result of the project.

[1] https://www.crestem.org/index.php?id=scopul-si-obiectivele

II. Programming using the STEM concept - Science, Technology, Engineering and Mathematics, Scratch programming language

What is Scratch?

A computer program is just a set of instructions that tell a computer what to do. You write these instructions using a *programming language*, and that's where Scratch comes in. Most programming languages are *text based*, which means you have to give the computer commands in what looks like a cryptic form of English. For example, to display "Hello!" on the screen, you might write:

```
print('Hello!')                       (in the Python language)
std::cout << "Hello!" << std::endl;(in the C++ language)
System.out.print("Hello!");       (in the Java language)
```

Learning these languages and understanding their syntax rules can be challenging for beginners. Scratch, on the other hand, is a *visual* programming language. It was developed in the Massachusetts Institute of Technology (MIT) Media Lab to make programming easier and more fun to learn. In Scratch, you won't type any complicated commands. Instead, you'll connect graphical blocks together to create programs. Confused? Look at the simple program in Figure 1-1, and I'll explain.

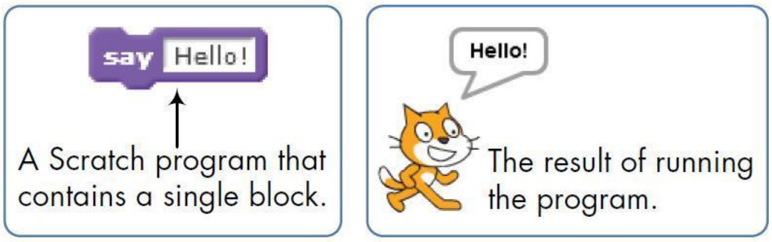
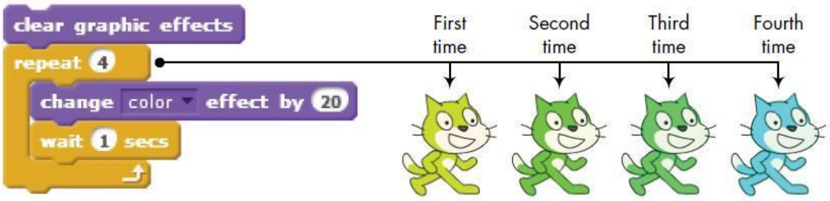

say Hello!
A Scratch program that contains a single block.
Hello!
The result of running the program.

Figure 1-1: When you run this Scratch block, the cat says "Hello!" in a speech bubble.

The cat that you see in Figure 1-1 is called a *sprite*. Sprites understand and obey sets of instructions that you give them. The purple block on the left tells the cat to display "Hello!" in a speech bubble. Many of the applications you'll create in this book will contain multiple sprites, and you'll use blocks to make sprites move, turn, say things, play music, do math, and so on. You can program in Scratch by snapping those color-coded blocks together as you would puzzle pieces or LEGO bricks. The stacks of blocks that you create are called *scripts*. For example, Figure 1-2 shows a script that changes a sprite's color four times.



```
clear graphic effects
repeat 4
    change color effect by 20
    wait 1 secs
```

Figure 1-2: Using a script to change the Cat *sprite's color*

This script waits for one second between color changes, and the four cats you see here show the sprite's new color after each change.

Scratch 2 version allows you to create projects directly in your web browser so you don't have to install any software on your computer, and we'll rely on Scratch's web interface for the material in this text. Now, that you know a little about this language, it's time to kick off our programming journey and learn how to use it.

Scratch Programming Environment

To start Scratch, go to the Scratch website (*http://scratch.mit.edu/*) and click the *TRY IT OUT* link. This should take you to Scratch's project editor interface, shown in Figure 1-3.

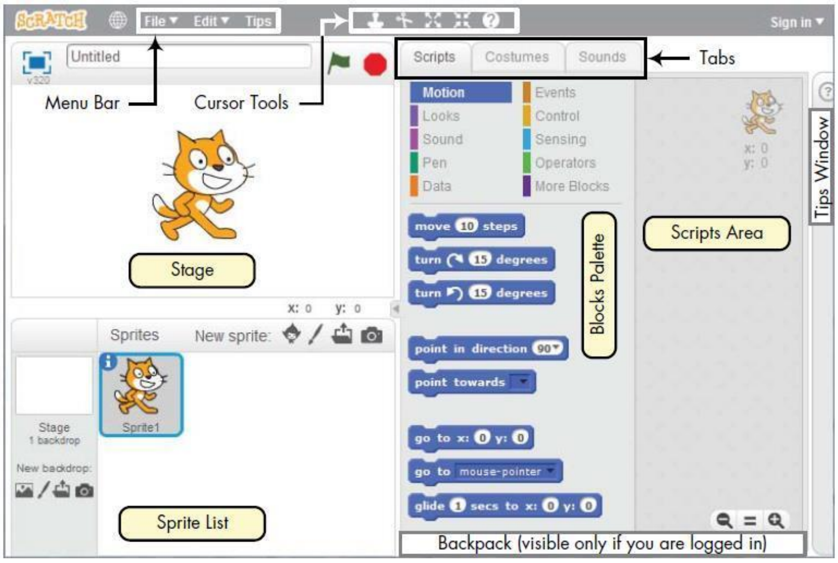


Figure 1-3: The Scratch user interface, where you'll build your programs

You should see a single window with at least the following three panes: the Stage (top left), the Sprite List (bottom left), and the Scripts tab (right), which contains the Blocks tab and the Scripts Area. The right pane also contains two additional tabs, Costumes and Sounds, which will be discussed later in this section. If you're logged into an account on the Scratch website, you should also see the Backpack (bottom right), which has buttons that let you share your project and use sprites and scripts from existing projects. Let's take a quick look at the three main panes.

The Stage

The *Stage* is where your sprites move, draw, and interact. The Stage is 480 steps wide and 360 steps tall, as illustrated in Figure 1-4. The center of the Stage has an *x*-coordinate of 0 and a *y*-coordinate of 0.

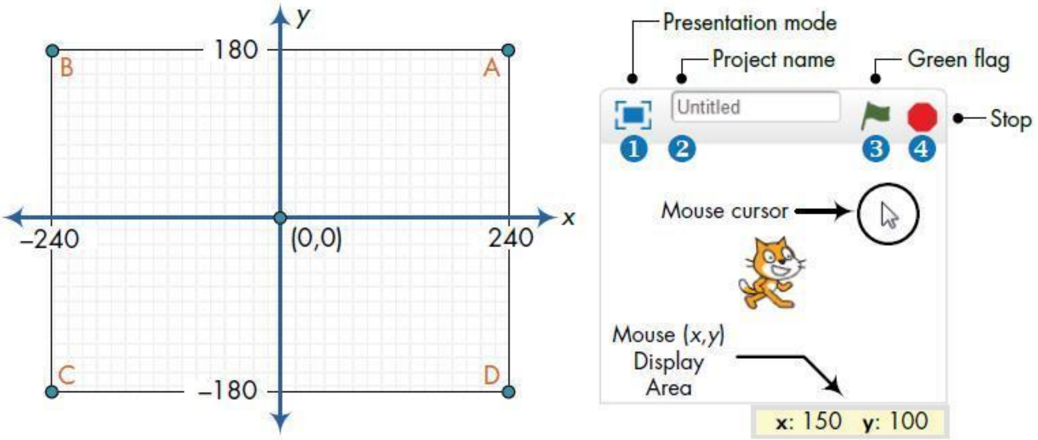


Figure 1-4: The Stage is like a coordinate plane with point (0,0) in the center.

You can find the (*x*,*y*) coordinates of any point on the Stage by moving the mouse cursor to that point and watching the numbers in the Mouse (*x*,*y*) Display Area, located directly below the Stage. The small bar located above the Stage has several controls. The Presentation mode icon (1) hides all scripts and programming tools and makes the Stage area take up almost your entire monitor. The edit box (2) shows the name of the current project. The green flag (3) and stop (4) icons let you start and end your program.

Sprite List

The Sprite List displays names and thumbnails for all the sprites in your project. New projects begin with a white Stage and a single cat-costumed sprite, as illustrated in Figure 1-5.

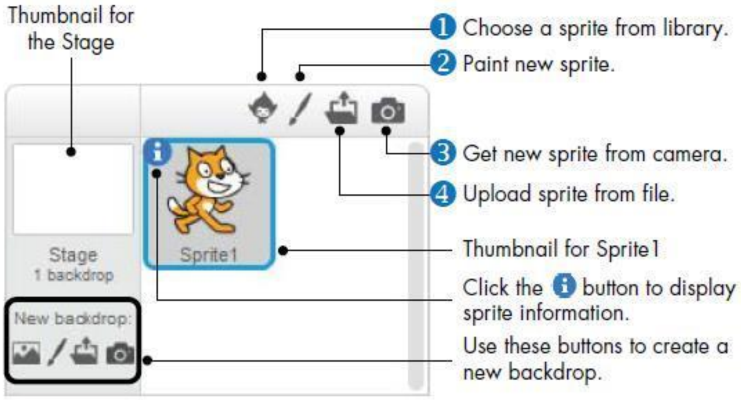

Thumbnail for the Stage
1 Choose a sprite from library.
2 Paint new sprite.
3 Get new sprite from camera.
4 Upload sprite from file.
Stage
1 backdrop
Sprite1
Thumbnail for Sprite1
Click the i button to display sprite information.
New backdrop:
Use these buttons to create a new backdrop.

Figure 1-5: The Sprite List for a new project

The buttons above the Sprite List let you add new sprites to your project from one of four places: Scratch's sprite library (1), the built-in Paint Editor (2) (where you can draw your own costume), a camera connected to your computer (3), or your computer (4).

Each sprite in your project has its own scripts, costumes, and sounds. You can select any sprite to see its belongings. Either (1) click the sprite's thumbnail in the Sprite List or (2) double-click the sprite itself on the Stage. The currently selected sprite thumbnail is always highlighted and outlined with a blue border. When you select a sprite, you can access its scripts, costumes, and sounds by clicking one of the three tabs located above the Scripts Area. We'll look at the contents of these tabs later. For now, right-click (or ctrl-click if you're using a Mac) the Cat sprite's thumbnail to see the popup menu shown in Figure 1-6.

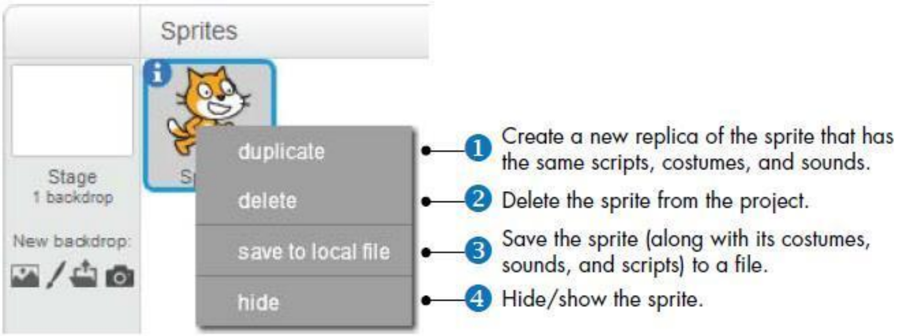

Sprites
Stage
1 backdrop
New backdrop:
duplicate
delete
save to local file
hide
1 Create a new replica of the sprite that has the same scripts, costumes, and sounds.
2 Delete the sprite from the project.
3 Save the sprite (along with its costumes, sounds, and scripts) to a file.
4 Hide/show the sprite.

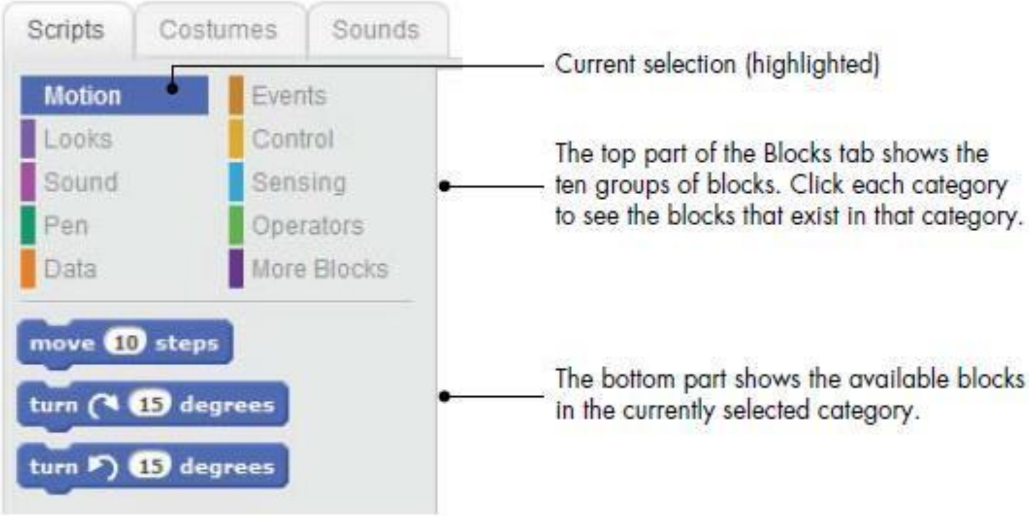
Figure 1-6: Right-clicking a sprite's thumbnail shows this handy menu.

The duplicate option (1) copies the sprite and gives the copy a different name. You can remove a sprite from your project with delete (2), and you can export a sprite to a *.sprite2* file on your computer using the save to local file option (3). (To import an exported sprite into another project, just click the Upload sprite from file button shown in Figure 1-5.) The hide/show option (4) allows you to change whether a sprite on the Stage is visible or not.

Along with thumbnails for your sprites, the Sprite List also shows a thumbnail of the Stage to the left (see Figure 1-6). The Stage has its own set of scripts, images, and sounds. The background image you see on the Stage is called a *backdrop*. When you start a new project, the Stage defaults to a plain, white backdrop, but you can add new backdrop images with any of the four buttons below the Stage's thumbnail. Click on the Stage icon in the Sprite List to view and edit its associated scripts, backdrops, and sounds.

Blocks Tab

Blocks in Scratch are divided into 10 categories (palettes): *Motion*, *Looks*, *Sound*, *Pen*, *Data*, *Events*, *Control*, *Sensing*, *Operators*, and *More Blocks*. Blocks are color coded to help you find related blocks easily. Scratch 2 has more than 100 blocks, though some blocks only appear under certain conditions. For example, blocks in the *Data* palette (discussed in Chapters 5 and 9) appear only after a variable or a list is created. Let's look at the various components of the Blocks tab in Figure 1-7.



Current selection (highlighted)

The top part of the Blocks tab shows the ten groups of blocks. Click each category to see the blocks that exist in that category.

The bottom part shows the available blocks in the currently selected category.

Figure 1-7: An enlarged view of the Blocks tab

Try clicking a block to see what it does. If you click **move 10 steps** on the *Motion* palette, for example, the sprite will move 10 steps on the Stage. Click it again, and the sprite moves another 10 steps. Click the **say Hello! for 2 secs** block (in the *Looks* palette) to make the sprite display "Hello!" in a speech bubble for two seconds. You can also access the help screen of a block by selecting *Block help* (the question mark icon) from the toolbar and clicking the block you're confused about. Some blocks require one or more inputs (also called *arguments*) that tell the block what to do. The number 10 in the **move 10 steps** block is an example of an argument. Look at Figure 1-8 to see the different ways blocks let you change their inputs.

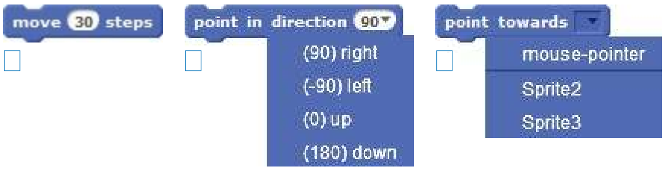


Figure 1-8: Changing the inputs of different types of blocks

You can change the number of steps in **move 10 steps** by clicking the white area where you see the 10 and entering a new number (1), perhaps 30 as you see in Figure 1-8. Some blocks, like **point in direction 90**, also have pull-down menus for their inputs (2). You can click the down arrow to see a list of available options and select one. This particular command has a white editable area, so you could also just type a value inside the white box. Other blocks, like **point towards** (3), will force you to choose a value from the drop-down menu.

Scripts Area

To make a sprite do interesting things, you need to program it by dragging blocks from the Blocks tab to the Scripts Area and snapping them together. When you drag a block around the Scripts Area, a white highlight indicates where you can drop that block to form a valid connection with another block (Figure 1-9). Scratch blocks only snap together in certain ways, eliminating the typing errors that tend to occur when people use text-based programming languages.

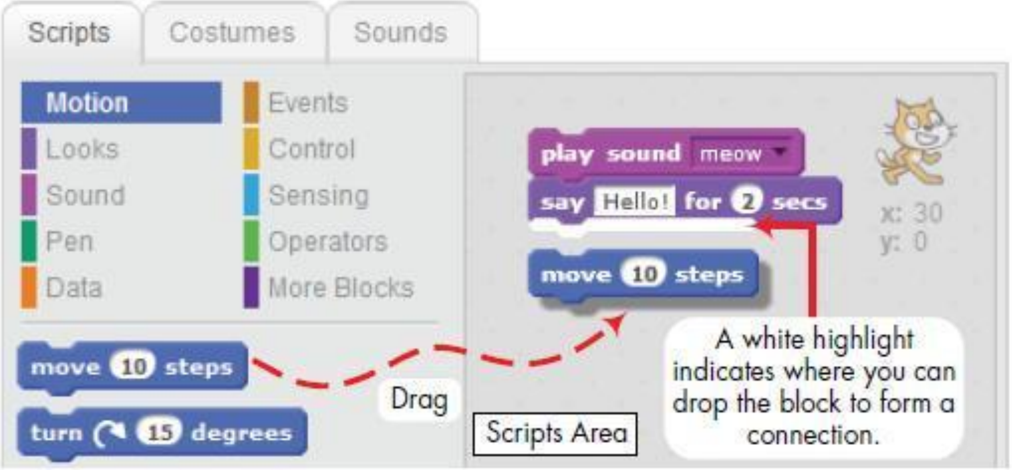


Figure 1-9: Drag blocks into the Scripts Area and snap them together to create scripts.

You don't need to complete scripts to run them, which means that you can test your script as you build it. Clicking anywhere on a script, complete or partial, runs the entire script, from top to bottom.

You can also easily disassemble a stack of blocks and test each individually. This will be an invaluable strategy when you're trying to understand long scripts. To move an entire stack of blocks, grab the top block of the stack. To detach a block in the middle of a stack and all the blocks below it, grab it and drag it. Go ahead and try this out. This feature also allows you to build your project one piece at a time. You can connect small chunks of blocks, test them to make sure they work as intended, and then combine them into larger scripts. You can even copy a stack of blocks from one sprite to another. Just drag the stack from the Scripts Area of the source sprite to the thumbnail of the destination sprite in the Sprite List.

Costumes Tab

You can change what a sprite looks like by changing its costume, which is just an image. The Costumes tab contains everything you need to organize your sprite's costumes; you could think of it like a clothes closet. The closet can have many costumes, but a sprite can wear only one at a time. Let's try changing the Cat sprite's costume now. Click the thumbnail of the Cat sprite and select the Costumes tab. As illustrated in Figure 1-10, the Cat has two costumes: costume1 and costume2. The highlighted costume (costume1 in this case) represents the sprite's current costume. Use these buttons to add a new costume to the sprite.

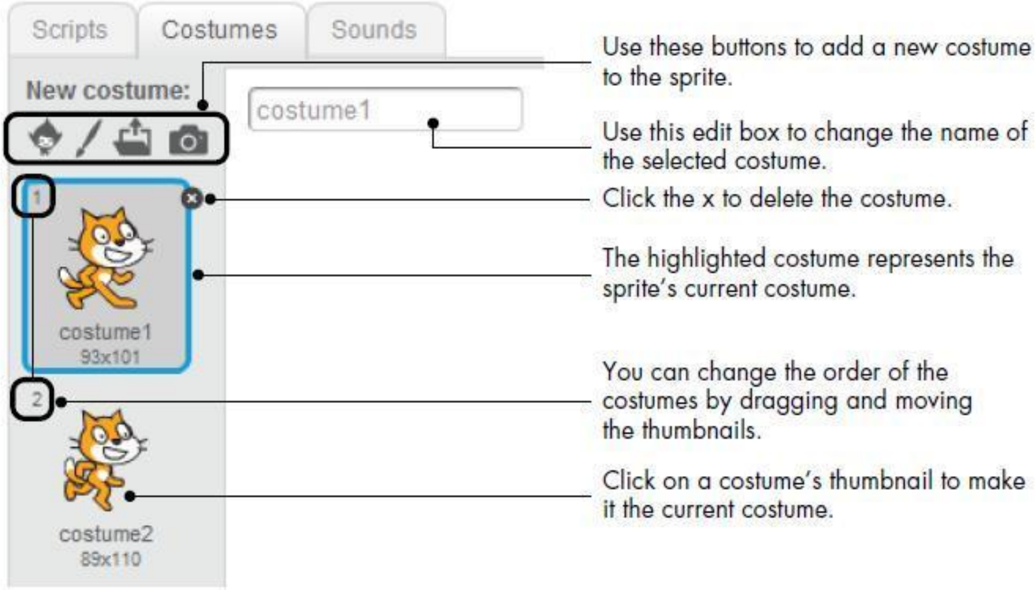

Scripts
Costumes
Sounds
New costume:
costume1
Use these buttons to add a new costume to the sprite.
Use this edit box to change the name of the selected costume.
Click the x to delete the costume.
The highlighted costume represents the sprite's current costume.
You can change the order of the costumes by dragging and moving the thumbnails.
Click on a costume's thumbnail to make it the current costume.
1
costume1
93x101
2
costume2
89x110

Figure 1-10: You can organize all the costumes for a sprite from the Costumes tab

.

If you right-click on a costume's thumbnail, you'll see a pop-up menu with three options: (1) duplicate, (2) delete, and (3) save to local file. The first option adds a new costume with an image identical to that of the costume you duplicated. The delete option deletes the selected costume. The last option allows you to save the costume to a file. You can import that costume and use it in a different project using the Upload costume from file button (the third button in Figure 1-10). Go ahead and try these options out.

Sounds Tab

Sprites can also play sounds, which liven up your programs. You can, for example, give a sprite different sounds to use when it's happy or sad. If your game contains a sprite that looks like a missile, you could make the missile generate different sounds when it hits or misses a target. The buttons in the Sounds tab will help you organize the different sounds your sprites can play. As shown in Figure 1-11, Scratch even provides a tool you can use to edit sound files. I won't discuss the details of this tool in this book, but I encourage you to experiment with it on your own.

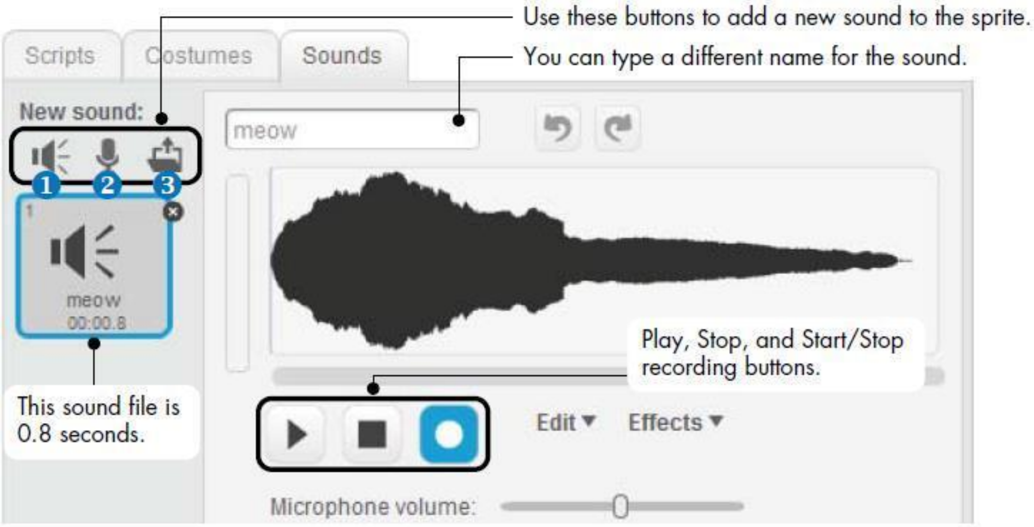

Use these buttons to add a new sound to the sprite.
You can type a different name for the sound.
Scripts
Costumes
Sounds
New sound:
1
2
3
meow
meow
00:00.8
Play, Stop, and Start/Stop recording buttons.
This sound file is 0.8 seconds.
Edit ▾
Effects ▾
Microphone volume:

Figure 1-11: The Sounds tab allows you to organize the sounds of a sprite .

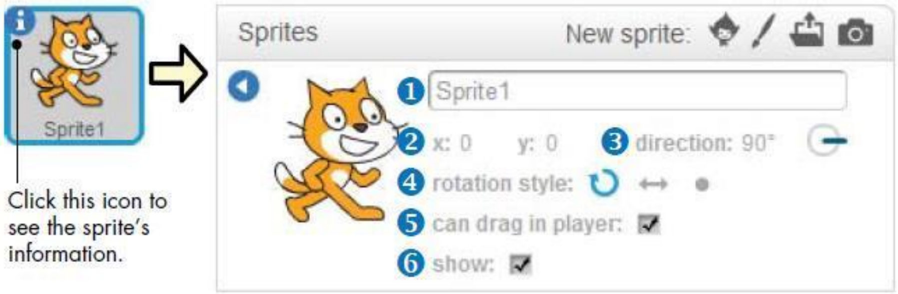
Most of the time, you'll need only the three buttons at the top of the Sounds tab. They allow you to choose a sound from Scratch's sound library (1), record a new sound (2) (if you have a microphone), or import an existing sound file from your computer (3). Scratch can read only MP3 and WAV sound files.

Backdrops Tab

When you select the thumbnail of the Stage in the Sprite List, the name of the middle tab changes from *Costumes* to *Backdrops*. Use this tab to organize the Stage's background images, which you can change with your scripts. For example, if you're creating a game, you might show one backdrop with instructions to begin and then switch to another when the user starts the game. The Backdrops tab is identical to the Costumes tab.

Sprite Info

You can view the sprite info area by clicking the small ℹ icon at the top-left edge of a sprite's thumbnail, as illustrated in Figure 1-12. This area shows the sprite's name, its current (*x*,*y*) position and direction, its rotation style and visibility state, and whether it can be dragged in Presentation mode. Let's briefly talk about each of these options.

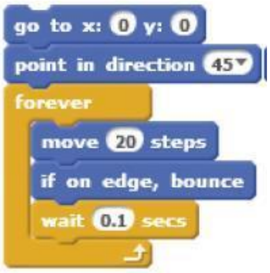


Click this icon to see the sprite's information.

1 Sprite1
2 x: 0 y: 0 3 direction: 90°
4 rotation style:
5 can drag in player: ☑
6 show: ☑

Figure 1-12: Sprite info area.

The edit box (1) at the top of this area allows you to change the sprite's name. You'll use this box many times in this book. The *x* and *y* values (2) show the sprite's current position on the Stage. Drag the sprite onto the Stage and watch what happens to these numbers. The sprite's direction (3) indicates which direction the sprite will move in response to a movement block. Drag the blue line emanating from the center of the circle icon to rotate the sprite. The three rotation-style buttons (4) (named Rotate, Left-right flip, and No rotate) control how the costume appears as the sprite changes its direction. To understand the effect of these buttons, create the script shown in Figure 1-13 and then click each of these buttons while the script is running. You can find the **wait** block in the *Control* palette. The Can drag in player checkbox (5) indicates whether or not the sprite can be dragged (using the mouse) in Presentation mode. Switch to Presentation mode with this box checked/ unchecked and try to drag the sprite across the Stage to understand the effect of this checkbox. The Show checkbox (6) allows you to show/hide the sprite at program design time. Try it out and see what happens.



```
go to x: 0 y: 0
point in direction 45
forever
    move 20 steps
    if on edge, bounce
    wait 0.1 secs
```

Figure 1-13: Script for demonstrating rotation styles

Toolbar

Let's take a quick look at Scratch's toolbar in Figure 1-14, starting with some of the buttons. Use the Duplicate and Delete buttons to copy and remove sprites, costumes, sounds, blocks, or scripts. The Grow button makes sprites bigger, while the Shrink button makes them smaller. Just click the button you want to use and then click on a sprite (or a script) to apply that action. To return to the arrow cursor, click on any blank area of the screen. You can use the Language menu to change the language of the user interface.

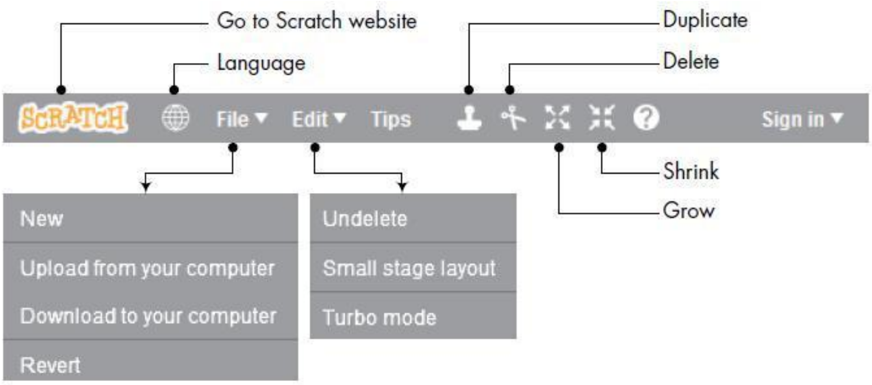


Figure 1-14: Scratch's toolbar

From the File menu, you can create new projects, upload (open) an existing project from your computer, download (save) the current project to your computer, or revert (undo) all your changes to the current project. Scratch 2 projects have an *.sb2* file extension to distinguish them from projects created in the previous version of Scratch (*.sb*). In the Edit menu, Undelete will bring back the last block, script, sprite, costume, or sound you deleted. The Small stage layout option shrinks the Stage and gives the Scripts Area more room. Selecting Turbo mode increases the speed of some blocks. For example, executing a **move** block 1,000 times may take about 70 seconds in normal mode and about 0.2 seconds in Turbo mode.

Now that you've seen the essentials of the Scratch toolbar, we'll talk briefly about Scratch's built-in Paint Editor.

Paint editor

You can use the Paint Editor (Figure 1-15) to create or edit costumes and backdrops. (Of course, you're free to use your favorite image-editing program, too.) If you want to learn more about Scratch's Paint Editor, check out *ScratchPaintEditor.pdf* (located in the online resources, which can be downloaded from *http://nostarch.com/learnscratch/*).

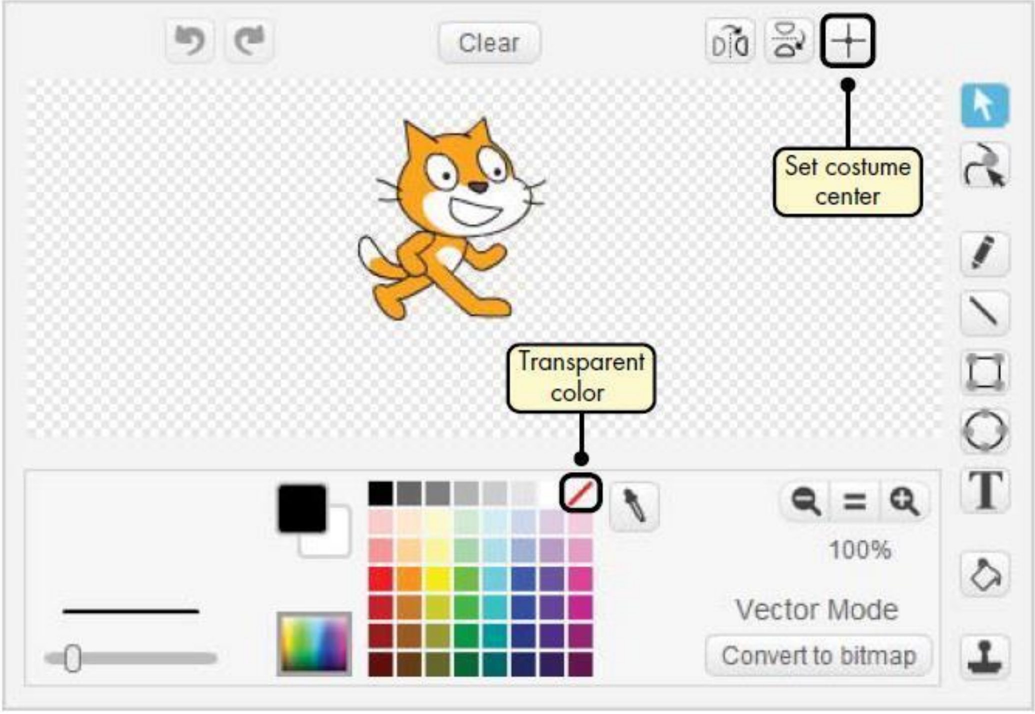


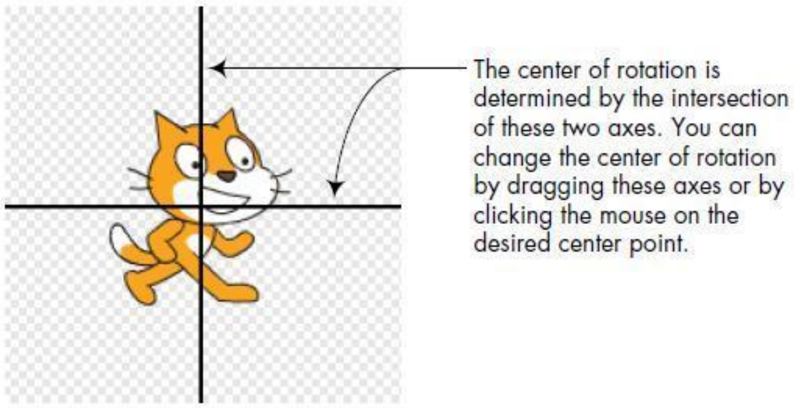
Figure 1-15: Scratch's Paint Editor

For now, there are two important features you'll need to know about: setting the center of an image and setting the transparent color. I'll explain these features in the following sections.

Setting the Center of an Image

When you command a sprite to turn (left or right), it will turn with respect to a reference point—the center of its costume. The Set costume center button (in the upper-right corner of the Paint Editor) allows you to choose that center. When you click this button, you'll see crosshairs on the drawing area, as shown in Figure 1-16.

The center point is determined by the intersection of these two axes, so to shift a costume's center, just drag them to a new position. To hide the axes, click the same button again.

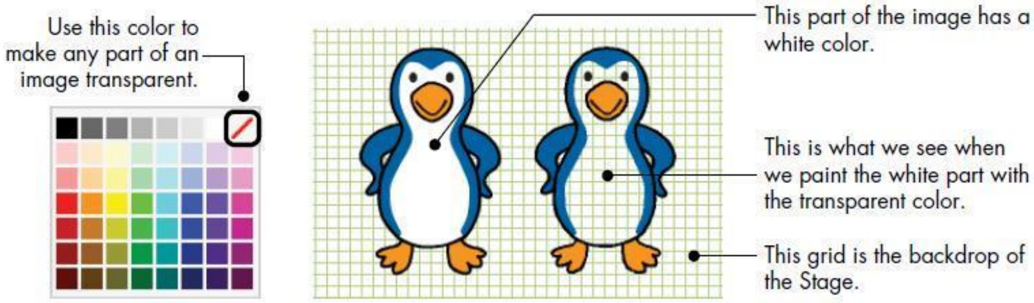


The center of rotation is determined by the intersection of these two axes. You can change the center of rotation by dragging these axes or by clicking the mouse on the desired center point.

Figure 1-16: Changing a costume's center after clicking the Set costume center button

Setting Transparent Color

When two images overlap, the top image will cover some part of the bottom image. Similarly, sprites cover parts of the Stage. If you want to see what the Stage looks like behind an image, you need to use the Paint Editor to make at least part of that image *transparent*, as the penguin on the right is in Figure 1-17. In the *Color* palette, just click the square with a diagonal red line and paint with that "transparent" color to make something invisible. You can think of this icon as a "No Color" sign, similar to a "No Smoking" sign with a red bar across a cigarette.



Use this color to make any part of an image transparent.

This part of the image has a white color.

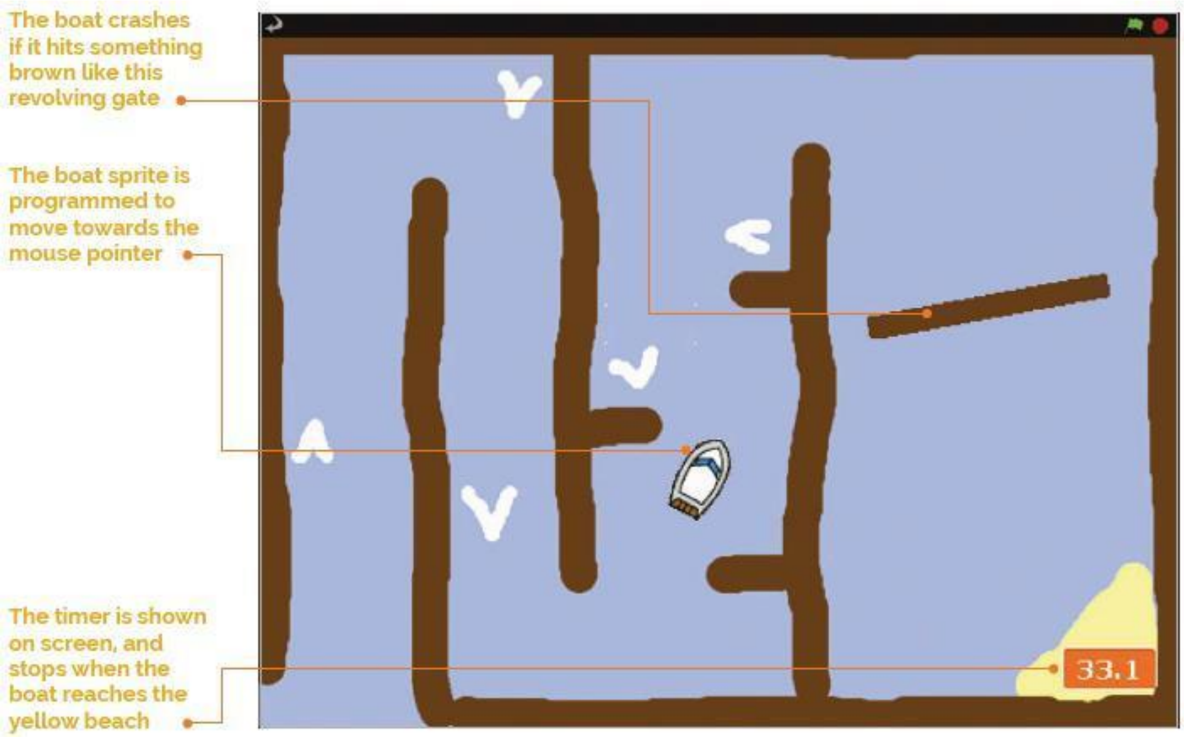
This is what we see when we paint the white part with the transparent color.

This grid is the backdrop of the Stage.

Figure 1-17: You can make any part of an image transparent by filling it with the "transparent" color .

Now that you know your way around the Scratch interface, we'll put that knowledge to good use and make something fun. Roll up your sleeves and get ready: We're making a game!

BOAT RACE



The boat crashes if it hits something brown like this revolving gate

The boat sprite is programmed to move towards the mouse pointer

The timer is shown on screen, and stops when the boat reaches the yellow beach

Now, you'll be making your own arcade game in which the player attempts to guide a boat safely around a maze-like course – including a revolving gate – to the finish in as fast a time as possible. You can even design your own custom course if you like. As well as moving a sprite towards the mouse pointer,this project involves collision detection, using the **touching color** Sensing block to determine whether the boat has hit something. Let's dive in and start coding…

STEP-01

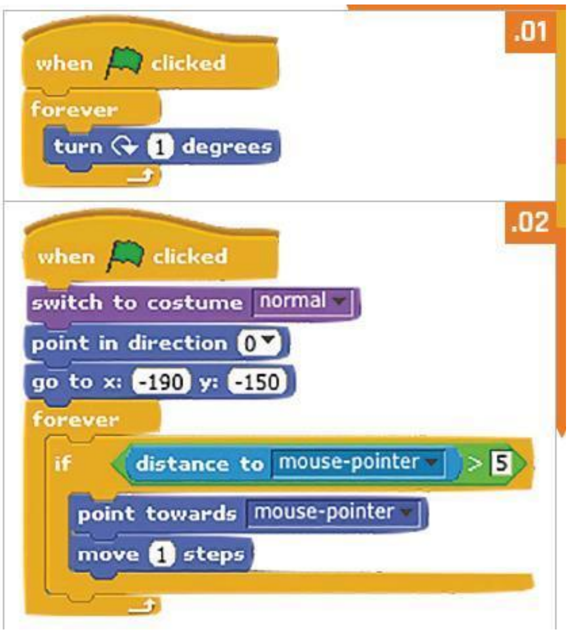
Prepare your artwork

First, delete the cat! You should then import the two sprites, for the boat and gate. Since they're not in the Scratch 1.4 library, you can download them (**magpi.cc/scratch_art**). Just click the star/folder icon above the Sprite List (bottom right), then navigate to the folder where you've stored the downloaded graphics for this project. Import the Boat and Gate sprites. If you aren't designing your own course, you can also download and import our Course backdrop: click Stage in the Sprite List, select the Backgrounds tab (top middle), then click Import and navigate to the folder.

STEP-02

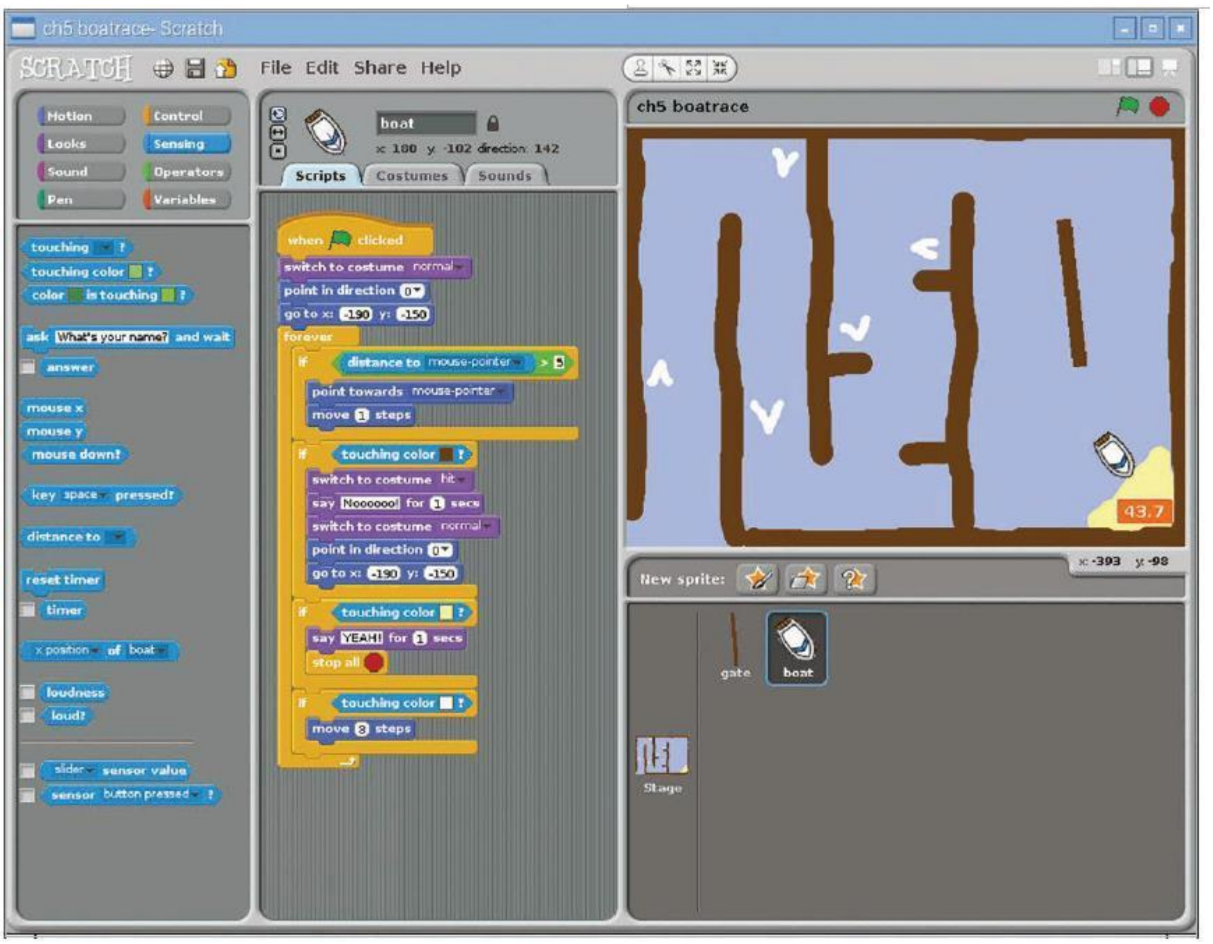
Design a course

You could just edit our course. Alternatively, to create a brand new one, click on the Stage in the Sprite List, then the Backgrounds tab, and Paint. Use the paint bucket tool to fill the canvas with a blue colour for the water. Then use a brown colour – which should be the same as in the Gate sprite – to draw the walls of the course. Use a yellow colour to draw some sand for the finish. Finally, add some white arrows which will act as speed boosters. Once this is done, let's make our Gate sprite rotate by adding the simple code in **Listing 1** to its Scripts area.

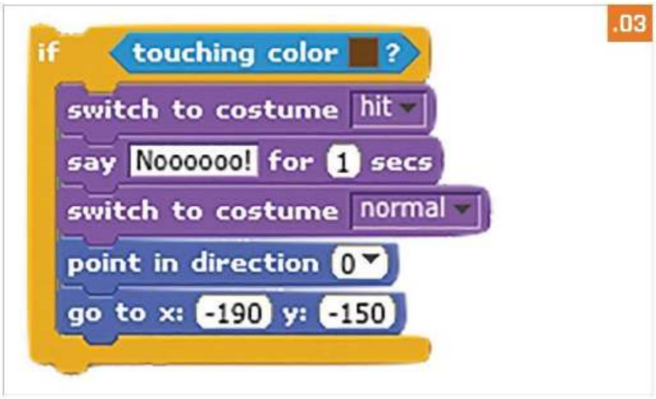


```
when [flag] clicked
forever
    turn ↻ 1 degrees
```

```
when [flag] clicked
switch to costume normal
point in direction 0
go to x: -190 y: -150
forever
    if distance to mouse-pointer > 5
        point towards mouse-pointer
        move 1 steps
```



```
when [flag] clicked
switch to costume normal
point in direction 0
go to x: -190 y: -150
forever
    if distance to mouse-pointer > 5
        point towards mouse-pointer
        move 1 steps
    if touching color ?
        switch to costume hit
        say Noooooo! for 1 secs
        switch to costume normal
        point in direction 0
        go to x: -190 y: -150
    if touching color ?
        say YEAH! for 1 secs
        stop all
    if touching color ?
        move 3 steps
```


.03

```
if touching color ?
    switch to costume hit
    say Noooooo! for 1 secs
    switch to costume normal
    point in direction 0
    go to x: -190 y: -150
```

STEP-03

Controlling the boat

In this game we'll be controlling the boat with a mouse – using the code in **Listing 2** in the Scripts tab of the Boat sprite. To do this, we simply point it towards 'mouse pointer' and move it 1 step at a time, within a **forever** loop. To stop it from moving when near the pointer, we put the control code in an if block that only tells it to move if the distance to the pointer is greater than 5. Try out the code and guide the boat: at the moment, it sails straight through barriers.

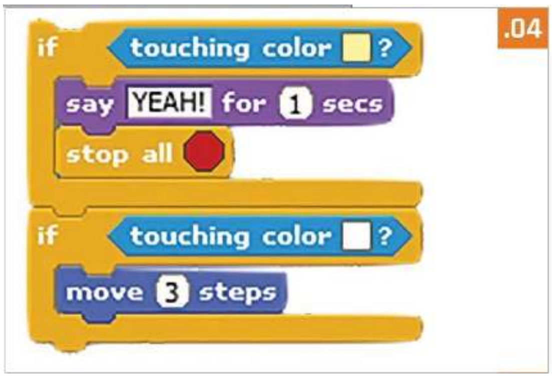
STEP-04

Make it crash!

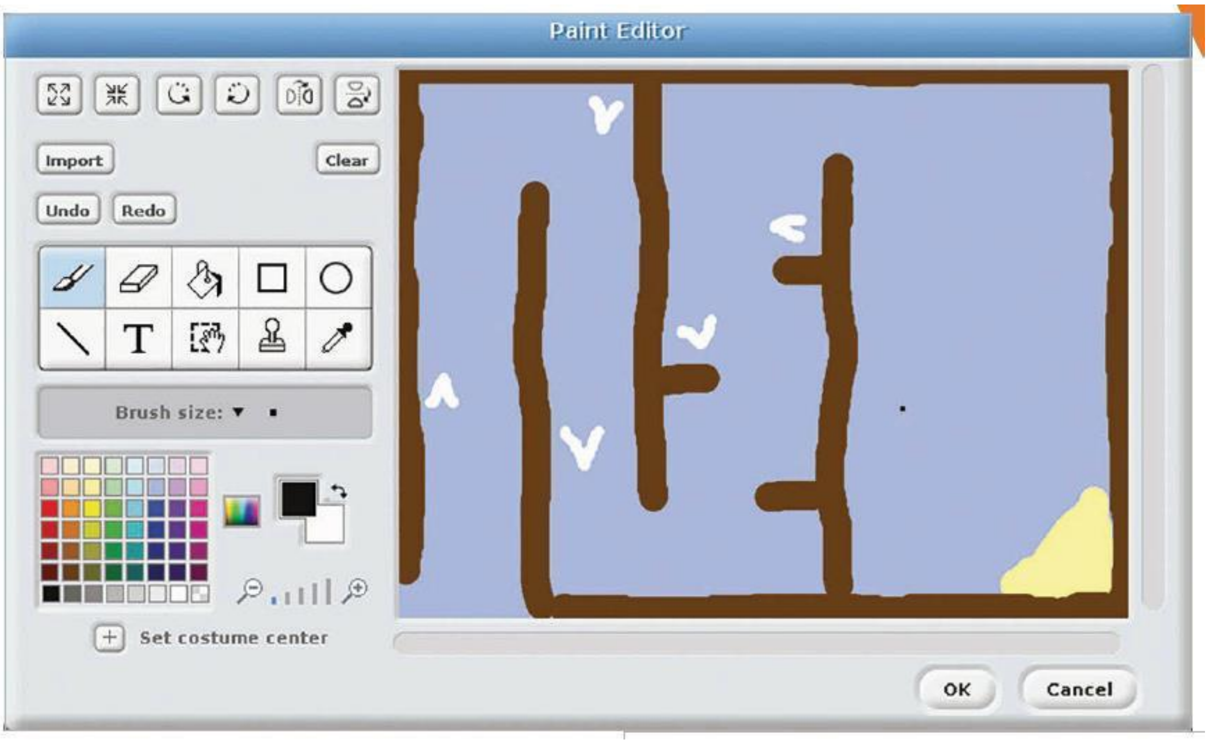
What we need is some collision detection to check whether the boat has hit a hazard. Within your **forever** block, add the code from **Listing 3** under your boat control code. Here, we use the **touching colour** Sensing block to see if the boat has hit anything brown: click the colour

square to get a dropper tool, then click on a brown part of the course. When it crashes, we switch the boat's costume, say 'Noooooo!', then place it back at the start point (in its normal costume). Let's add two more **if touching color** blocks, shown in **Listing 4**, to our forever loop. The first checks whether the boat has reached the yellow beach, which acts as the finish line, and stops the program. The second detects the white of

our booster arrows and moves the boat three steps.



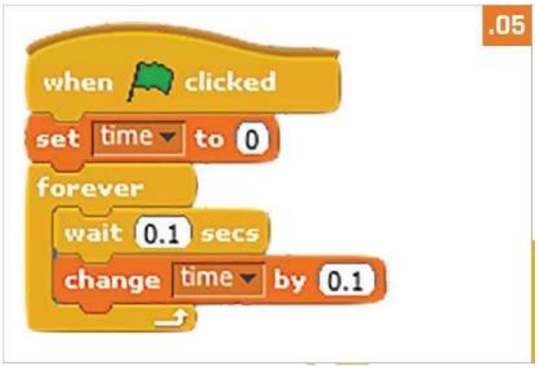
```
if touching color 🟨 ?
  say YEAH! for 1 secs
  stop all
if touching color ⬜ ?
  move 3 steps
```



STEP-05

Boosters and time

To make our game a bit more exciting, we need a timer. Click the Stage and add the **Listing 5** code to its Scripts area. This sets the time to zero at the start of the game, then gradually increases the **time** variable in line with real time – you'll need to create the latter in Variables and make sure it's ticked so that it's shown on the stage.



```
when ⚑ clicked
set time ▾ to 0
forever
    wait 0.1 secs
    change time ▾ by 0.1
```

STEP-06

Taking it further

You could easily add a sound effect for when the boat crashes, using a Sound block. You could even add background music, composing it using Sound blocks with various drums, instruments, and notes. The best time(s) could also be stored in a variable or list.

III. Programming language Visual Basic Application (VBA)

Visual Basic is a third-generation event-driven programming language from Microsoft known for its Component Object Model (COM) programming model first released in 1991 and declared legacy during 2008.

Microsoft intended Visual Basic to be relatively easy to learn and use.

But is it really «Basic»? Not really It's called «Basic» because it is the descendent of BASIC, a programming language used in the 90s.

Despite not having any visual element, the «Visual» part was added because of Microsoft's naming convention, and because it enables the use of a graphical user interface (GUI).

Why should you use Visual Basic?

It allows you to create databases and front-end applications;

ActiveX technology lets you use functionalities of other applications, like Microsoft Word or Excel

Internet capabilities make it easy to access documents and applications across the web

The programmed applications are .exe files you can freely distribute

Derivative languages

VBScript is the default language for Active Server Pages.

Visual Basic .NET is Microsoft designated successor to Visual Basic 6.0. It compiles and runs using the .NET Framework.

OpenOffice Basic is a Visual Basic compatible interpreter that originated in Star Office suite.

Gambas is a Visual Basic inspired free software programming language for the Linux operating system

WinWrap Basic is a third-party VBA variant used with various software

LotusScript is a VBA variant available in Lotus SmartSuite and Lotus Notes.

Later versions of Corel WordPerfect Office implement access to VBA as one of the macro/scripting languages

Earlier versions of Microsoft Word use a variant of Visual Basic called WordBasic

Logo of Microsoft Visual Studio, application that uses Visual Basic .NET,

Visual Basic’s modern substitute.

Visual Basic for Applications (VBA) is included in many Microsoft applications (Microsoft Office), and also in many third-party products.

Visual Basic development ended with 6.0, but in 2010 Microsoft introduced VBA 7 to provide extended features and add 64-bit support.

It is largely the same language as Visual Basic 6.0 and uses the same runtime library.

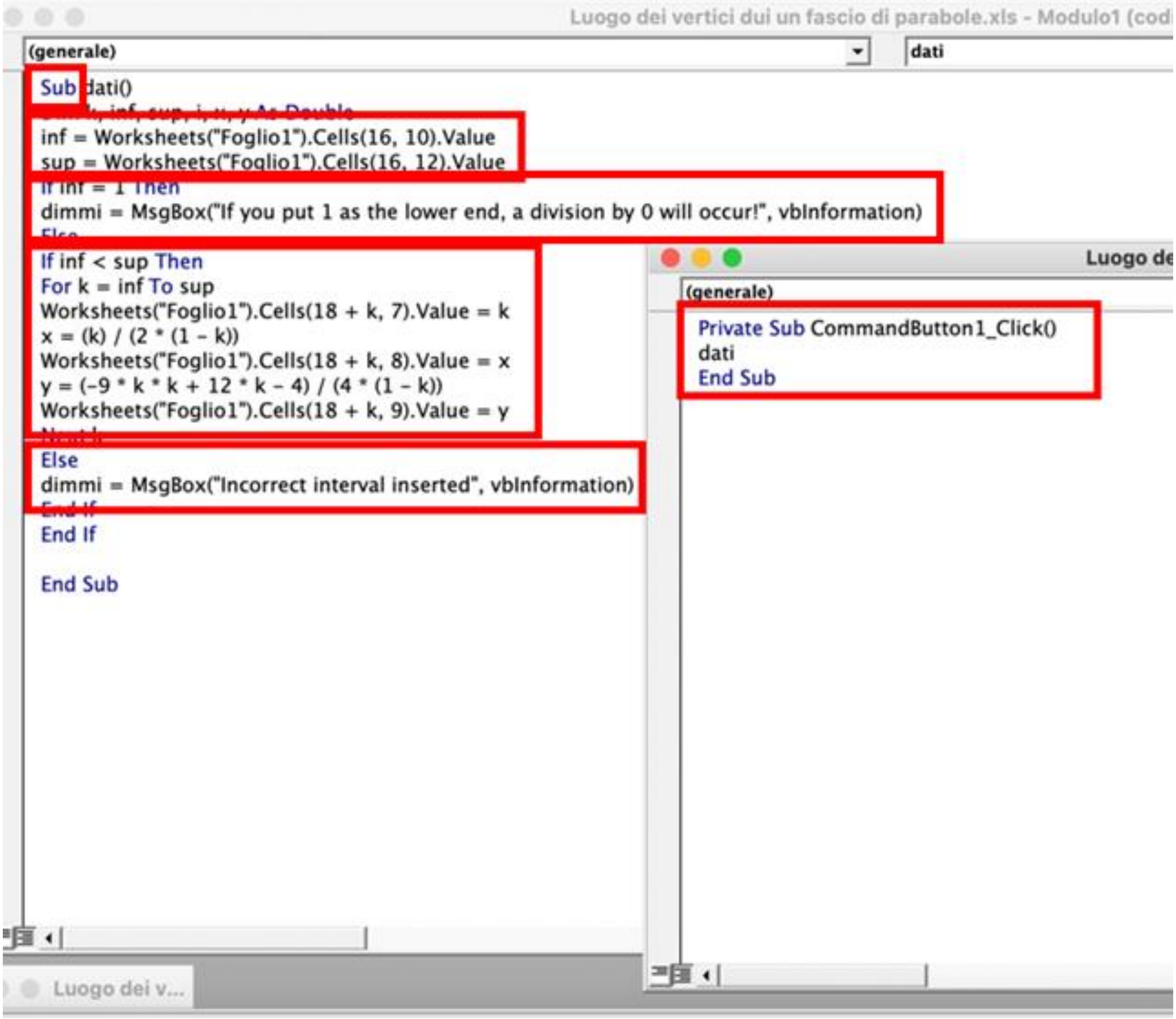
It can be used in various methods in Microsoft office applications, especially to enhance the experience of Microsoft Excel.

Visual Basic Application (VBA) is a programming language for digital innovation. The use of this programming language has the aim to strengthen the knowledge and skills of students and teachers. It may be applied to the study of Maths, Physics, ICT in an innovative way. VBA is mainly used in the Office Excel package and allows the implementation of worksheets with mathematical formulas that the program does not have. The positive repercussions can be appreciated in all areas of Science because VBA allows the creation of worksheets for the application and adaptation of formulas of this. It allows the management of results obtained from experiences in physics, chemistry and mathematics laboratories. It is possible to create programmes to solve problems arising from different discussions (Problem solving).The teacher poses the problem to the students who, through the application of the formulas, will give various solutions. This will create the conditions for a dynamic creativity, typical of young people and will favour their intellectual curiosity. So the students will be able to use VBA and apply it to Science, Maths and Physics.). VBA will make the teaching of Sciences attractive because it will provide practical and tangible results.

As an **example**, we used VBA tools to create a problem solver for Maths and Physics.

Maths example

Here is a code example in building a calculator of the locus of vertexes of parabolas family:

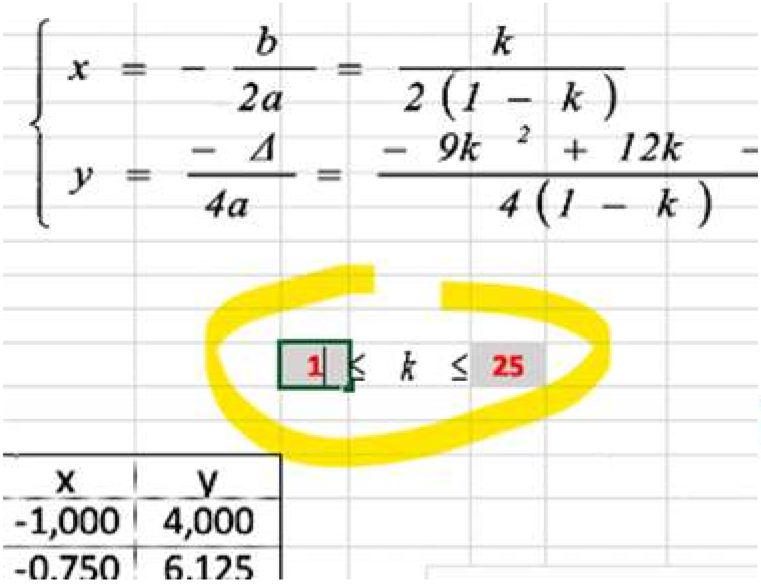

Luogo dei vertici dui un fascio di parabole.xls - Modulo1 (cod
(generale)
dati
Luogo de
(generale)
Luogo dei v...

```
Sub dati()
Dim k, inf, sup, i, x, y As Double
inf = Worksheets("Foglio1").Cells(16, 10).Value
sup = Worksheets("Foglio1").Cells(16, 12).Value
If inf = 1 Then
dimmi = MsgBox("If you put 1 as the lower end, a division by 0 will occur!", vbInformation)
Else
If inf < sup Then
For k = inf To sup
Worksheets("Foglio1").Cells(18 + k, 7).Value = k
x = (k) / (2 * (1 - k))
Worksheets("Foglio1").Cells(18 + k, 8).Value = x
y = (-9 * k * k + 12 * k - 4) / (4 * (1 - k))
Worksheets("Foglio1").Cells(18 + k, 9).Value = y
Next k
Else
dimmi = MsgBox("Incorrect interval inserted", vbInformation)
End If
End If

End Sub
```

```
Private Sub CommandButton1_Click()
dati
End Sub
```

Let’s break down the code.

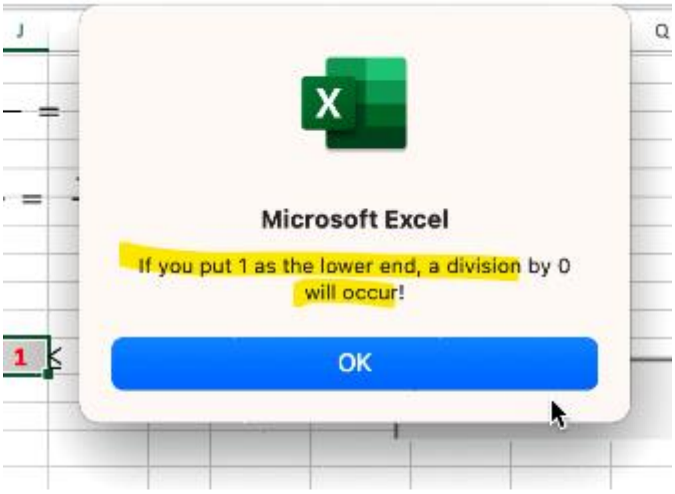
- Sub is short for Subroutine; Subroutine means putting one or more input parameters to get multiple results.
- Inf indicates the lower end of the equation;
- Sup indicates the upper end of the equation;
- The value of x and y depends on the value of the parameter k
- If the lower end is the number 1, an error message appears:



$$\begin{cases} x = -\dfrac{b}{2a} = \dfrac{k}{2(1-k)} \\ y = \dfrac{-\Delta}{4a} = \dfrac{-9k^{2}+12k\ -}{4(1-k)} \end{cases}$$

$1 \leq k \leq 25$

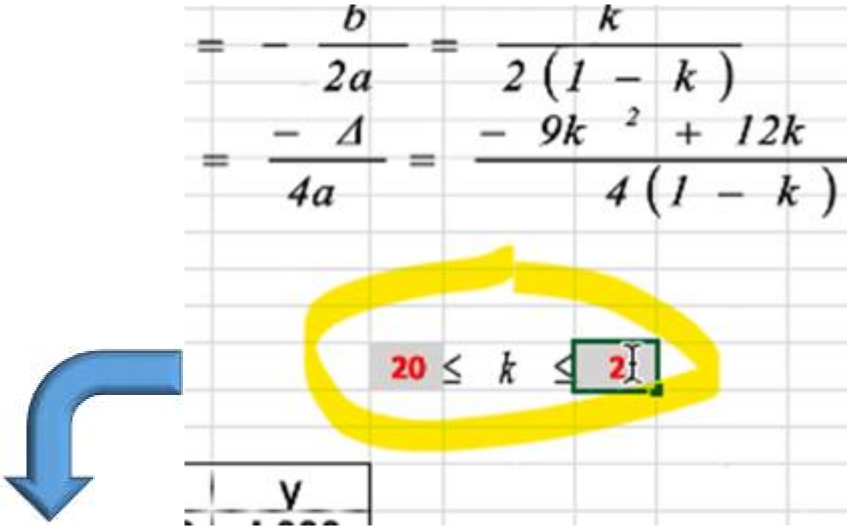
x	y
-1,000	4,000
-0,750	6,125



Microsoft Excel

If you put 1 as the lower end, a division by 0 will occur!

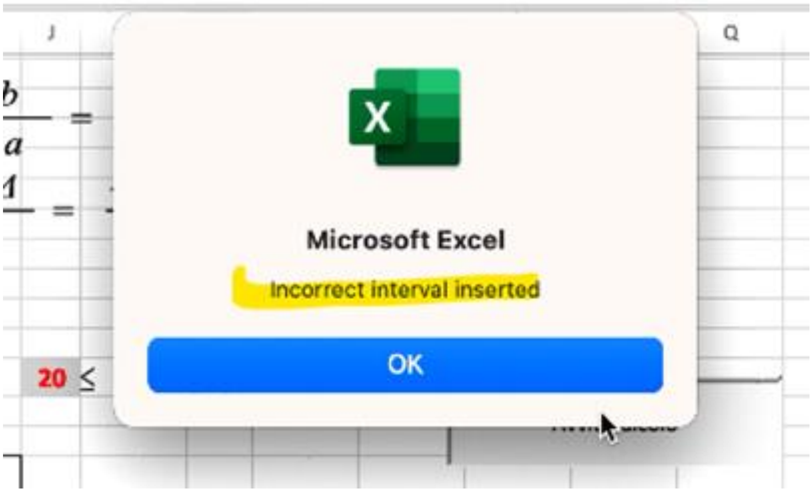
- If inf < sup is not true, another error message appears



$$= -\frac{b}{2a} = \frac{k}{2(1-k)}$$

$$= \frac{-\Delta}{4a} = \frac{-9k^2 + 12k}{4(1-k)}$$

$20 \leq k \leq 2$



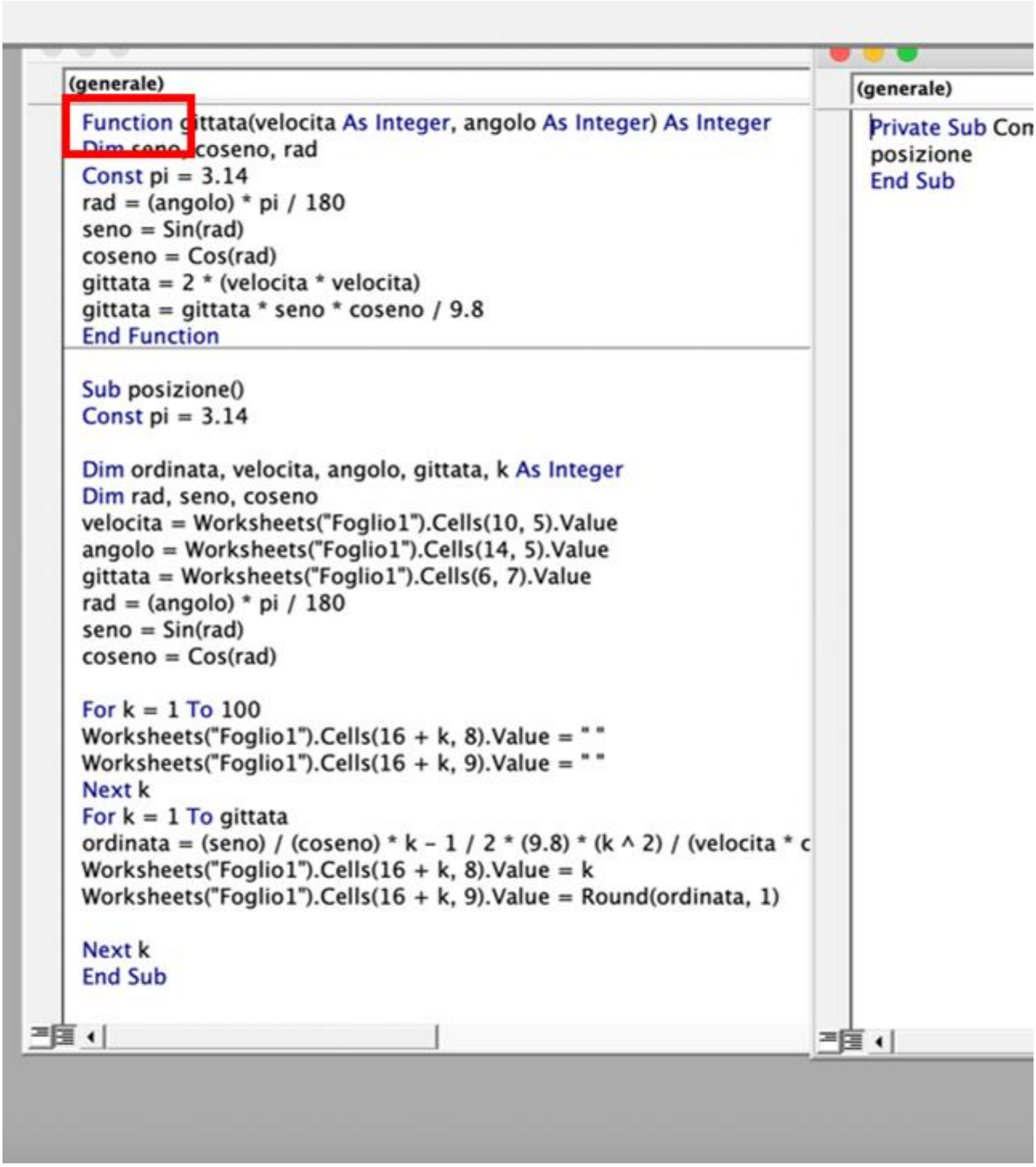
Microsoft Excel

Incorrect interval inserted

- The button can only work if associated with the corresponding macro, a series of inputs that give the corresponding results.

Physics example

Another **example**, using a different code, to calculate the range of a projectile:



```
Function gittata(velocita As Integer, angolo As Integer) As Integer
Dim seno, coseno, rad
Const pi = 3.14
rad = (angolo) * pi / 180
seno = Sin(rad)
coseno = Cos(rad)
gittata = 2 * (velocita * velocita)
gittata = gittata * seno * coseno / 9.8
End Function

Sub posizione()
Const pi = 3.14

Dim ordinata, velocita, angolo, gittata, k As Integer
Dim rad, seno, coseno
velocita = Worksheets("Foglio1").Cells(10, 5).Value
angolo = Worksheets("Foglio1").Cells(14, 5).Value
gittata = Worksheets("Foglio1").Cells(6, 7).Value
rad = (angolo) * pi / 180
seno = Sin(rad)
coseno = Cos(rad)

For k = 1 To 100
Worksheets("Foglio1").Cells(16 + k, 8).Value = " "
Worksheets("Foglio1").Cells(16 + k, 9).Value = " "
Next k
For k = 1 To gittata
ordinata = (seno) / (coseno) * k – 1 / 2 * (9.8) * (k ^ 2) / (velocita * c
Worksheets("Foglio1").Cells(16 + k, 8).Value = k
Worksheets("Foglio1").Cells(16 + k, 9).Value = Round(ordinata, 1)

Next k
End Sub
```

```
Private Sub Com
posizione
End Sub
```

Function is a different command: it means that one or more input parameters only give one solution.

IV. Robot Game: Build and program an autonomous LEGO MINDSTORMS robot

What is Lego?

Lego Mindstorms EV3 History

LEGO is a line of plastic construction toys that are manufactured by the Lego Group, a company based in Billund, Denmark. The Lego Group began manufacturing the interlocking toy bricks in 1949. As of July 2015, 600 billion Lego parts had been produced.

The first ever LEGO set made in 1949.

These sets were using older designs of the "modern" plastic LEGO brick (without the hollow underside tube). It wasn't until 1958 that the current design of LEGO bricks started being used instead.

Lego Mindstorms is a hardware/software platform which is produced by Lego for the development of programmable robots based on Lego building blocks.

There have been four generations of the Mindstorms platform: the original Robotics Invention System, NXT, NXT 2.0, and EV3. With each platform release, the motor and sensor capabilities expanded.

A programmable brick from the Robotics Invention System set

A brick of Lego Mindstorms EV3

Lego Mindstorms EV3 is the successor to the second generation Lego Mindstorms NXT 2.0 kit. It was released in stores on September 1, 2013. There are many competitions using this set, among them are the First Lego League and the World Robot Olympiad. The biggest change in the Lego Mindstorms EV3 is the technological advances in the programmable brick.

Lego Mindstorms EV3 parts

The computer/“Brick”

- Programs are executed from the Brick.

- Programs are downloaded to the Brick.

- Sends programmed information to motors.

- Receives information from sensors.

- It’s interface can be used for basic programming.

Large motor

Large motor is for the mobility requirements of the robot.

Medium motor

Medium motor is used primarily for moving parts, for a quick response of the robot, not for mobility.

Color sensor

- Detects color differences and sends that data to the Brick.

- Emits a set of color wavelengths. The color wavelength that is reflected back, and not absorbed, determines the color of the surface.

Ultrasonic sensor

- Detects distance to an object and sends that data to the Brick.

- Uses echolocation by emitting an ultrasonic wave that is received back after bouncing off an object.

- The time it takes to be received back determines the distance to that object.

Touch sensor

- Sends the state of either being pressed, not pressed, or bumped (pressed, then not pressed) to the Brick.

- Often used to determine if the robot has physically bumped into something.

Gyro sensor

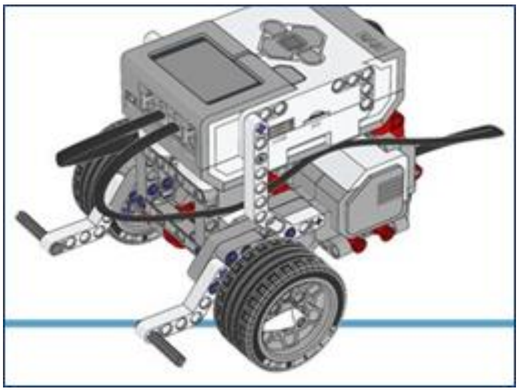
Detects the robots orientation and rotational motion and sends that data to the Brick for possible action according to programmed parameters.

EV3 Education Core builds

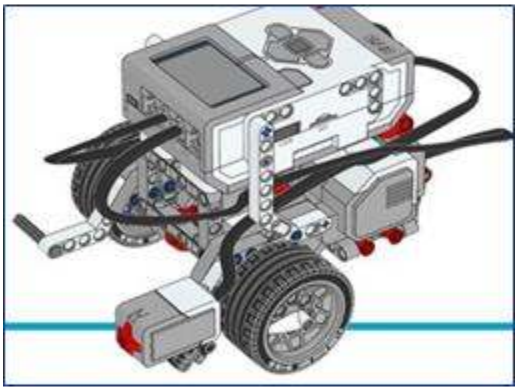
Driving base

It‘s one of the most universal builds for the Lego Mindstorms EV3 kit. It can be used to perform a multitude of different tasks.

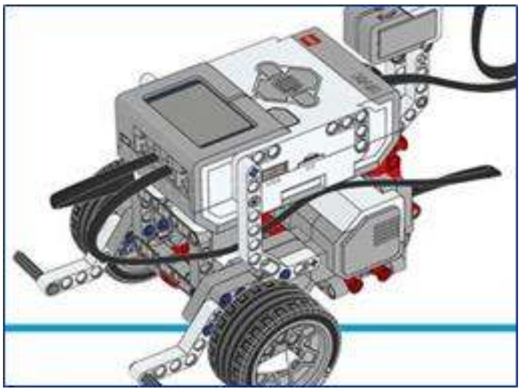
All of the other builds mentioned in the slide before are made from this build.



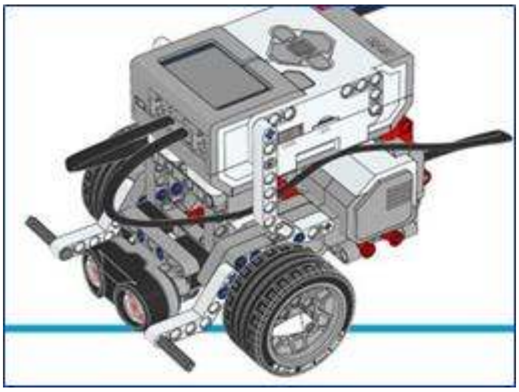
Touch sensor



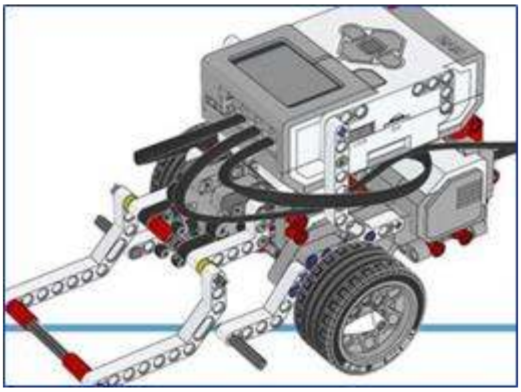
Gyro sensor



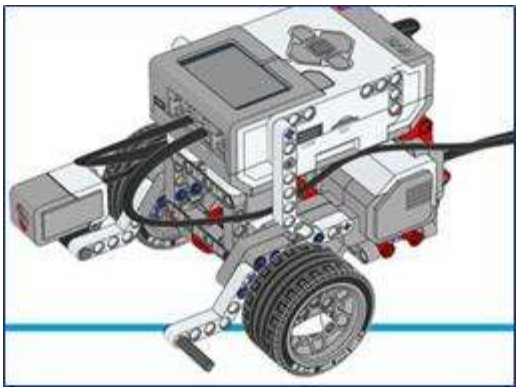
Ultrasonic sensor



Medium motor

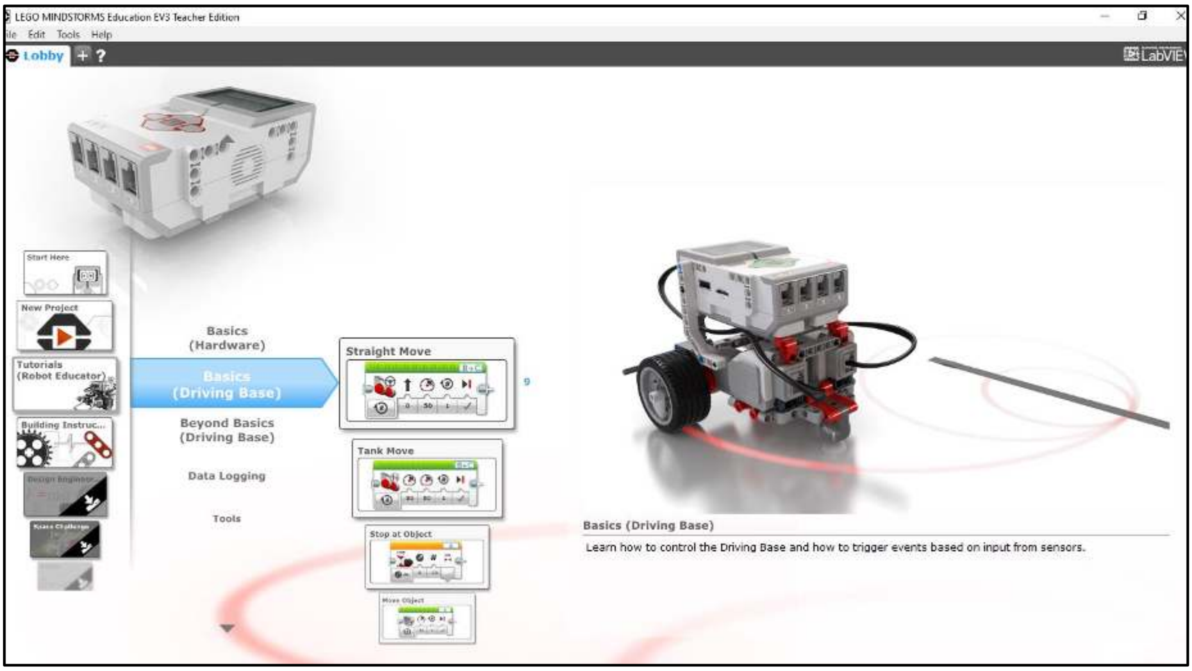


Color sensor



Examples of games that can be used during LEGO Lego Mindstorms EV3 programming

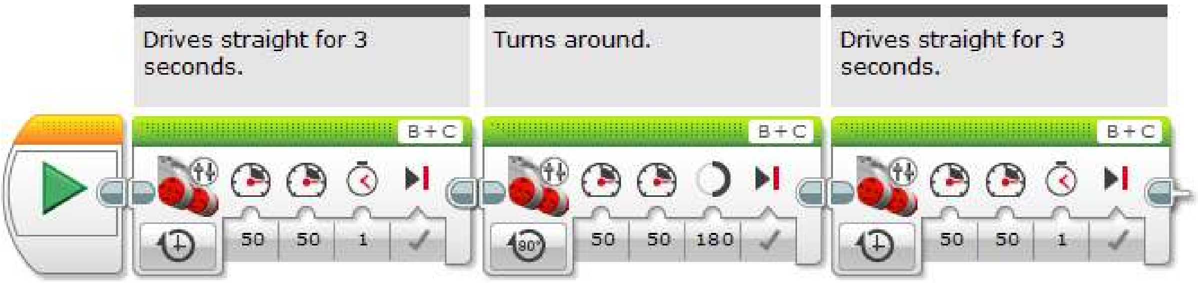
First of all, you need to have the required software.



Basics (Driving Base)

Learn how to control the Driving Base and how to trigger events based on input from sensors.

Lego Mindstorms EV3 programming interface is block-based, so it‘s very intuitive and easy to learn.



Drives straight for 3 seconds.

Turns around.

Drives straight for 3 seconds.

Example of a basic program

Tutorials on how to program your Lego Mindstorms EV3 using the software - EV3lessons.com / education.lego.com.

V. e-learning platforms

About e-learning

The use of modern technologies in the realization and presentation courses, the transition from traditional variants to training tools in the virtual space, to the approach called e-learning, to the use of virtual training environments, is a topical desideratum.

Courses in the computer network using current technologies are more attractive for students than using chalk on traditional board.

An **e-learning platform** is a teaching content management system and its distribution to students (teaching activities, assessment tests, homework, references to external pedagogical resources), as well as a virtual learning environment with socializing elements that can be customized.

Below we will briefly exemplify how to use two e-learning platforms:

1. Moodle Platform
2. https://app.wand.education/
3. https://www.tinkercad.com/

About Moodle Platform

Moodle (Modular Object-Oriented Dynamic Learning Environment) is one of the most used open-source platforms of Learning Management Systems (LMS).

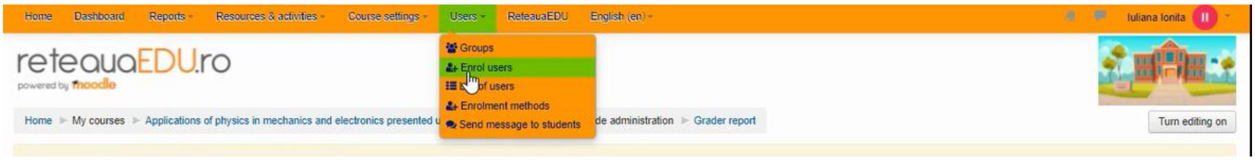
The most important functionalities of the platform are the following:

- Moodle provides a number of tools for assessing learners as well as managing training outcomes, which allows for a detailed analysis of the impact of the learning process.
- Provides tools that allow teachers to move from classic course materials and Power Point presentations to interactive and multimedia training materials.
- Moodle allows fast and efficient communication between platform users

One of the themes of the virtual mobility within the project was the use of the moodle platform by the participating students.

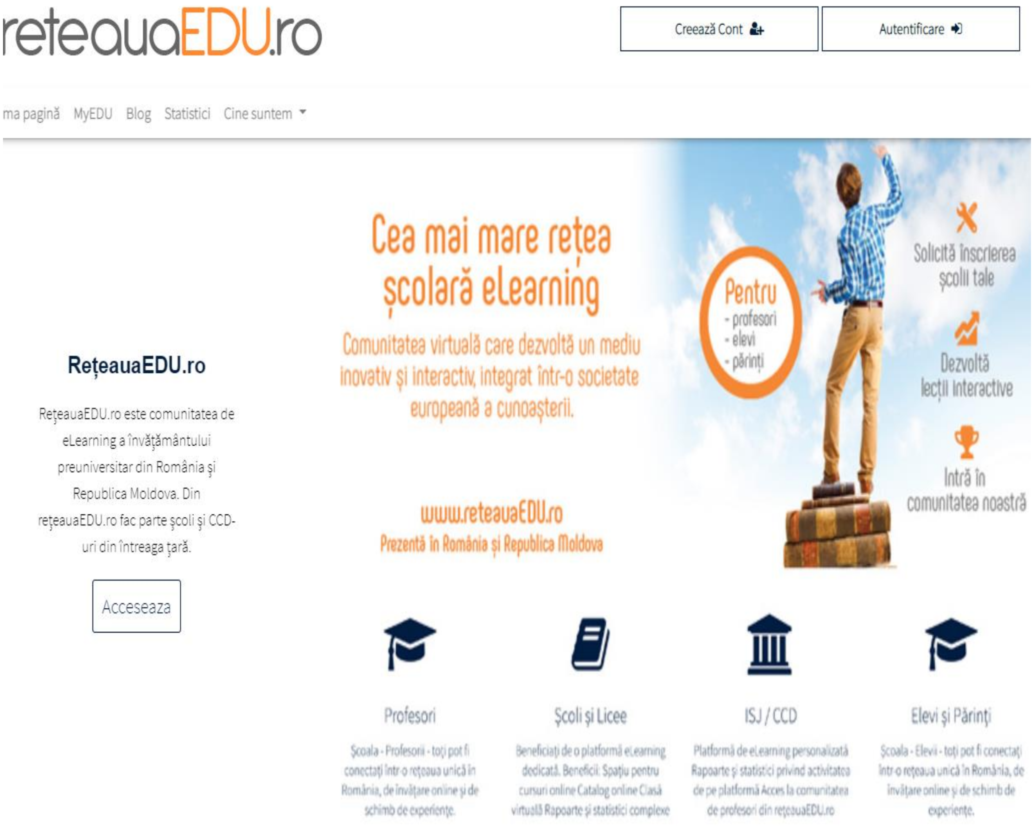
Thus, a course was created on the moodle platform of the high school, in which all the participating students were enrolled, as well as the accompanying teachers.

There are several ways in which the teacher of the course enrolls the students:



One of the methods is the one through which the administrator adds the cohort of students, and the students will receive the user and the login password for the course.

How to access?


reteauaEDU.ro
Creează Cont
Autentificare
ma pagină
MyEDU
Blog
Statistici
Cine suntem
Acceseaza
Cea mai mare rețea școlară eLearning
Comunitatea virtuală care dezvoltă un mediu inovativ și interactiv, integrat într-o societate europeană a cunoașterii.
www.reteauaEDU.ro
Prezentă în România și Republica Moldova
Pentru
- profesori
- elevi
- părinți
Solicită înscrierea școlii tale
Dezvoltă lecții interactive
Intră în comunitatea noastră

ReţeauaEDU.ro

ReţeauaEDU.ro este comunitatea de eLearning a învăţământului preuniversitar din România şi Republica Moldova. Din reţeauaEDU.ro fac parte şcoli şi CCD-uri din întreaga ţară.

Profesori

Şcoala - Profesorii - toţi pot fi conectaţi într-o reţeaua unică în România, de învăţare online şi de schimb de experienţe.

Şcoli şi Licee

Beneficiaţi de o platformă eLearning dedicată. Beneficii: Spaţiu pentru cursuri online Catalog online Clasă virtuală Rapoarte şi statistici complexe

ISJ / CCD

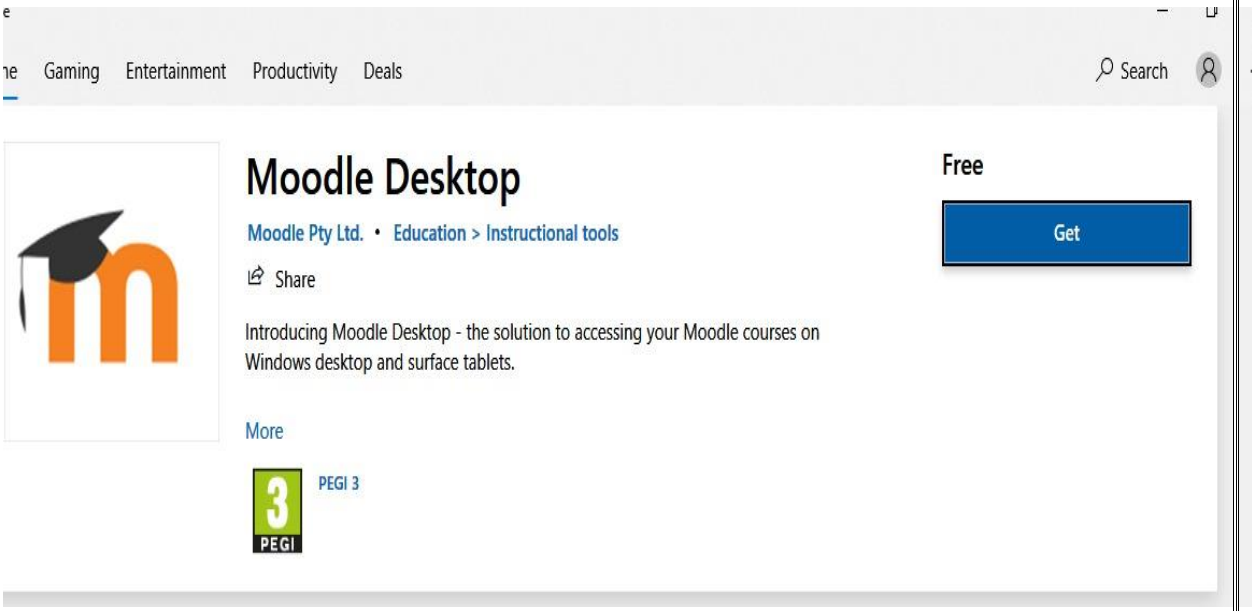
Platformă de eLearning personalizată Rapoarte şi statistici privind activitatea de pe platformă Acces la comunitatea de profesori din reţeauaEDU.ro

Elevi şi Părinţi

Şcoala - Elevii - toţi pot fi conectaţi într-o reţeaua unică în România, de învăţare online şi de schimb de experienţe.

Steps for accessing platforms:

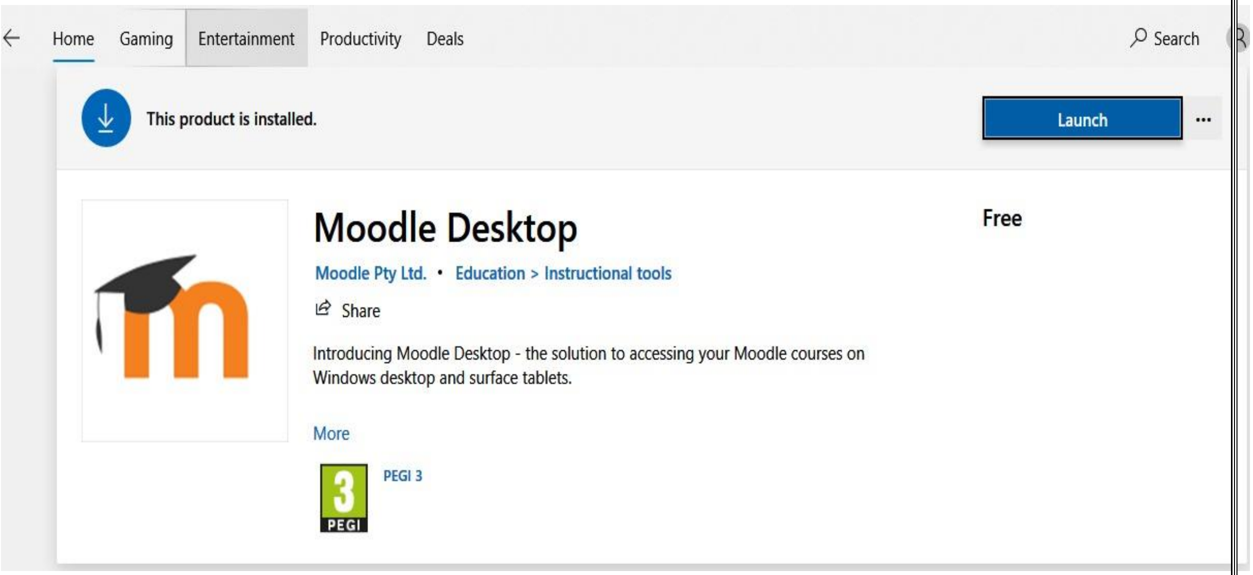
1. the application **Moodle Desktop** is downloaded from the Microsoft Store;



Moodle Desktop

Introducing Moodle Desktop - the solution to accessing your Moodle courses on Windows desktop and surface tablets.

2. Launch the application:

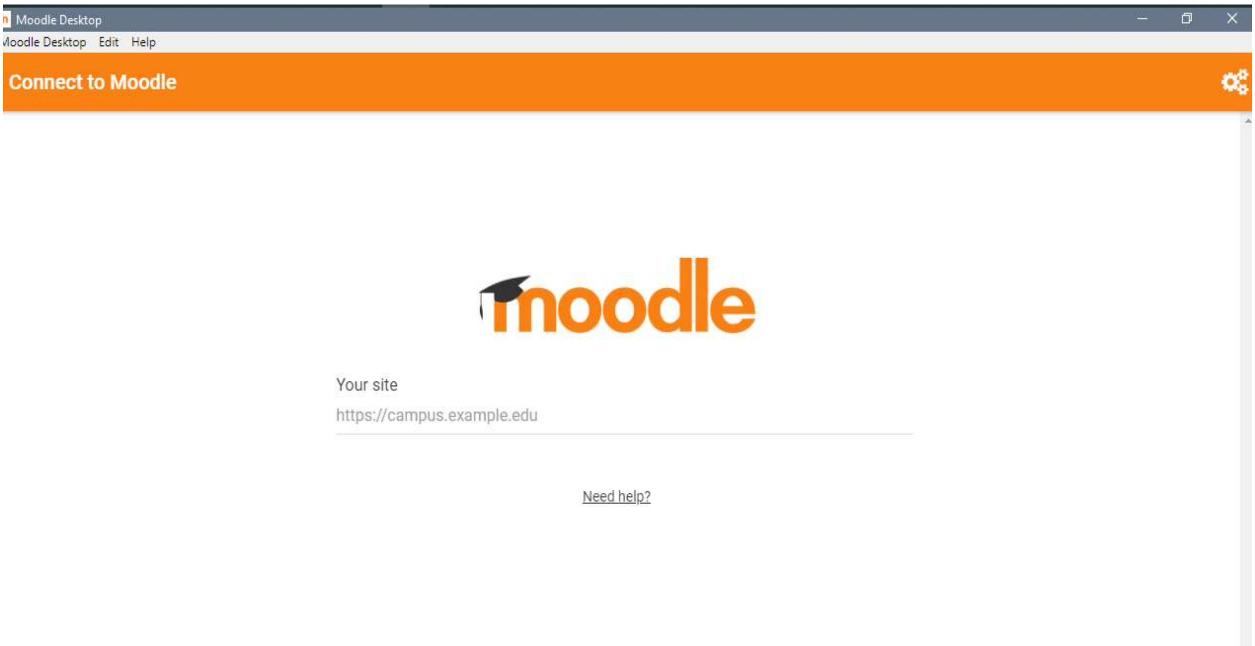


This product is installed.

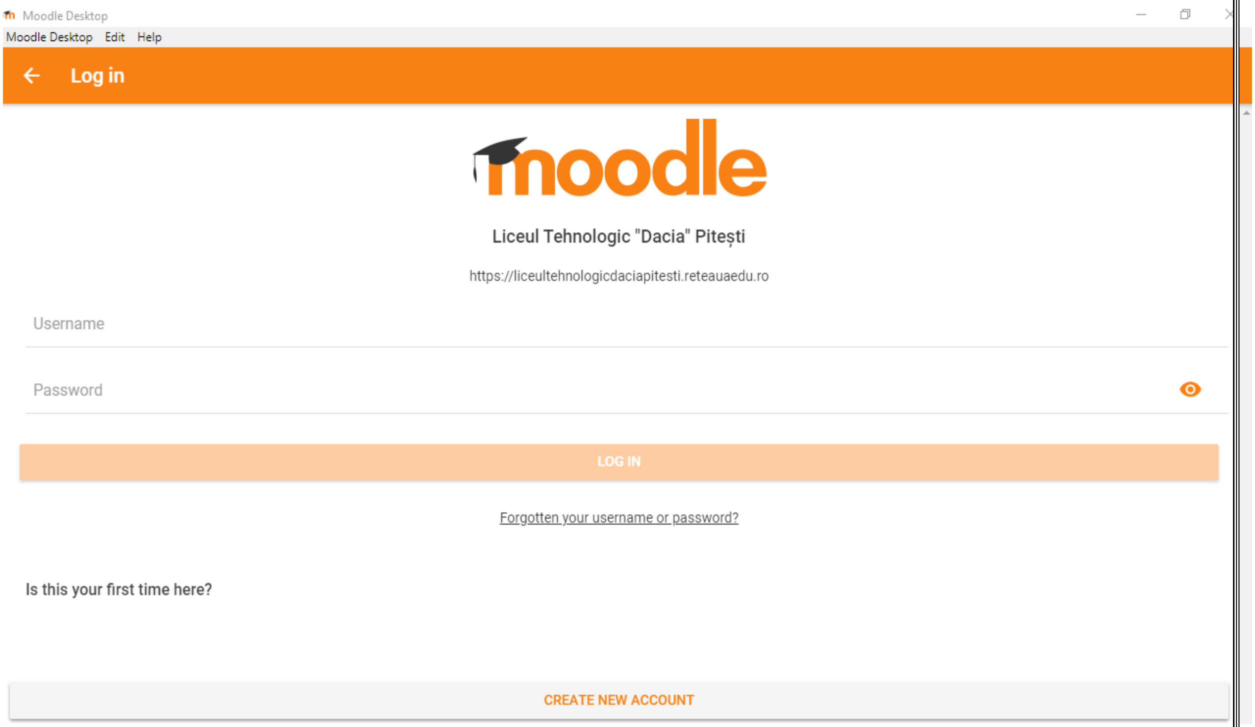
Moodle Desktop

Introducing Moodle Desktop - the solution to accessing your Moodle courses on Windows desktop and surface tablets.

3. Enter the site address



4. The username and password are entered from the teacher who enrolls in the course:

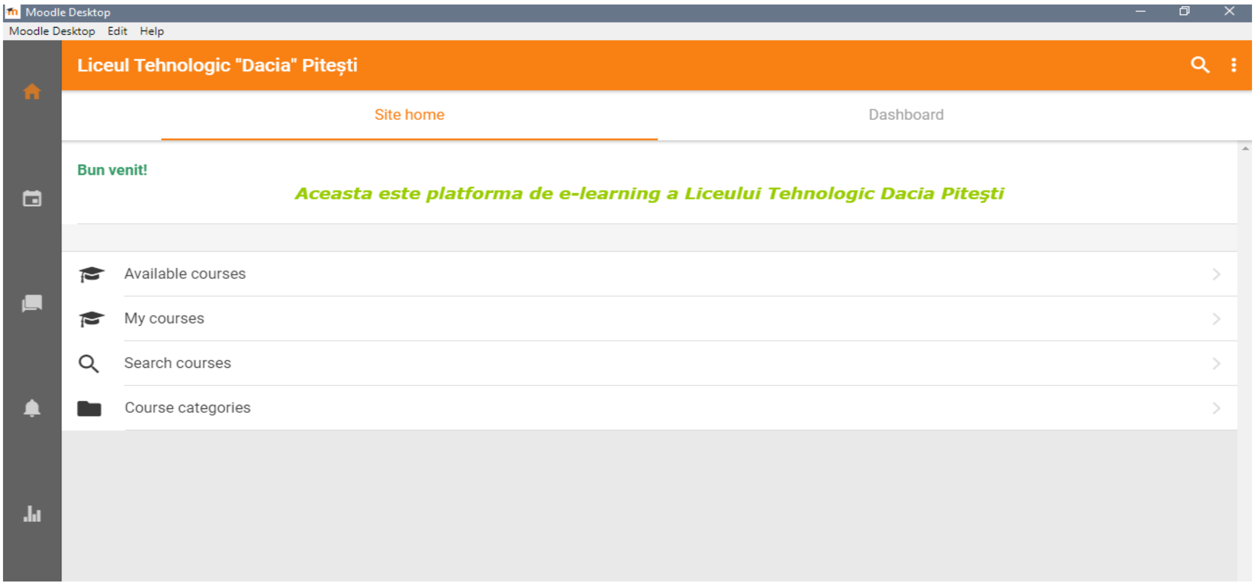


Liceul Tehnologic "Dacia" Pitești

https://liceultehnologicdaciapitesti.reteauaedu.ro

Is this your first time here?

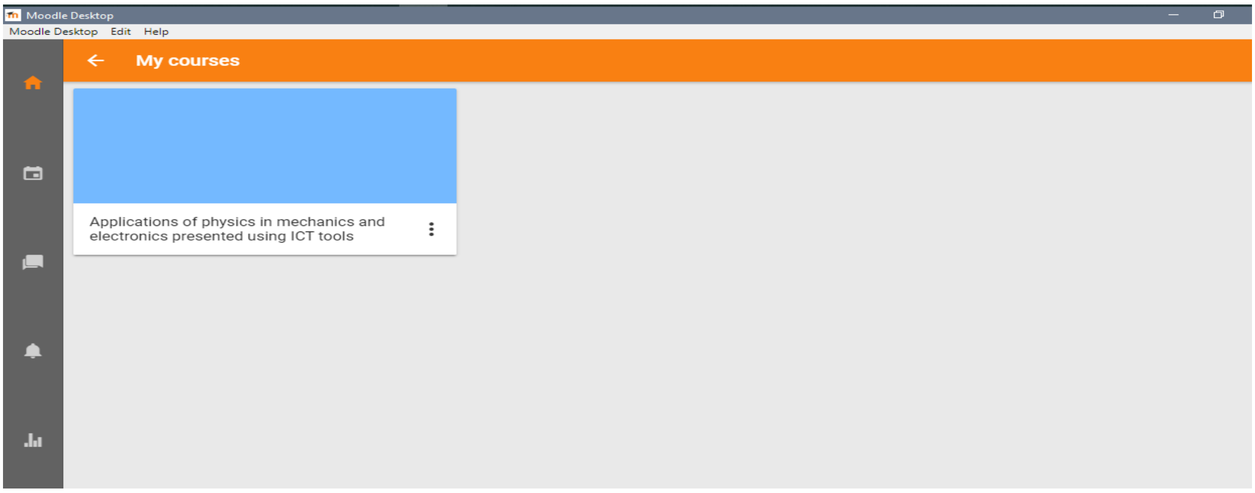
5. After logging in, the following window will appear:



Bun venit!

Aceasta este platforma de e-learning a Liceului Tehnologic Dacia Piteşti

6. The created course will be accessed:



Applications of physics in mechanics and electronics presented using ICT tools

A course created on the Moodle platform includes the following sections:

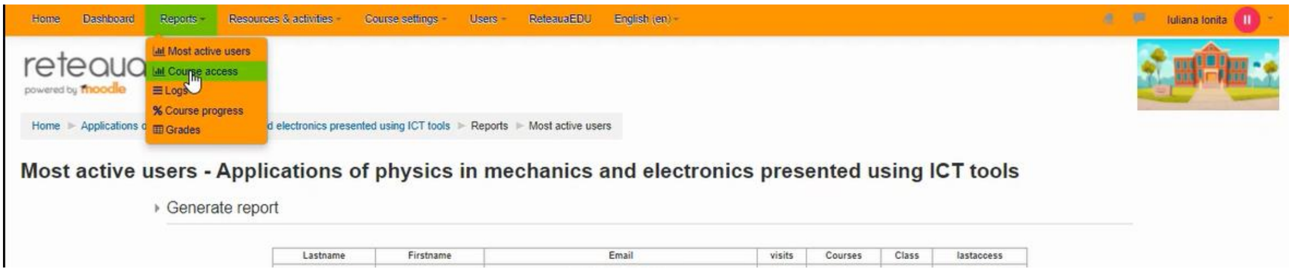
- Overview;
- Welcome message;
- Course description;
- What is the way to complete the course;
- Necessary tools;
- An initial questionnaire;
- A number of lessons allocated to the course;

- Each lesson will contain:
  - Study materials in the form of a presentation, PDF or video;
  - Practical activities;
  - Overview;
  - Welcome message;
  - Course description;
  - What is the way to complete the course;
  - An initial questionnaire;
  - Themes that will be uploaded on the platform in a certain format and with a certain deadline;
  - Worksheets;
  - Final questionnaire.

The moodle platform is easy to use and attractive for students.

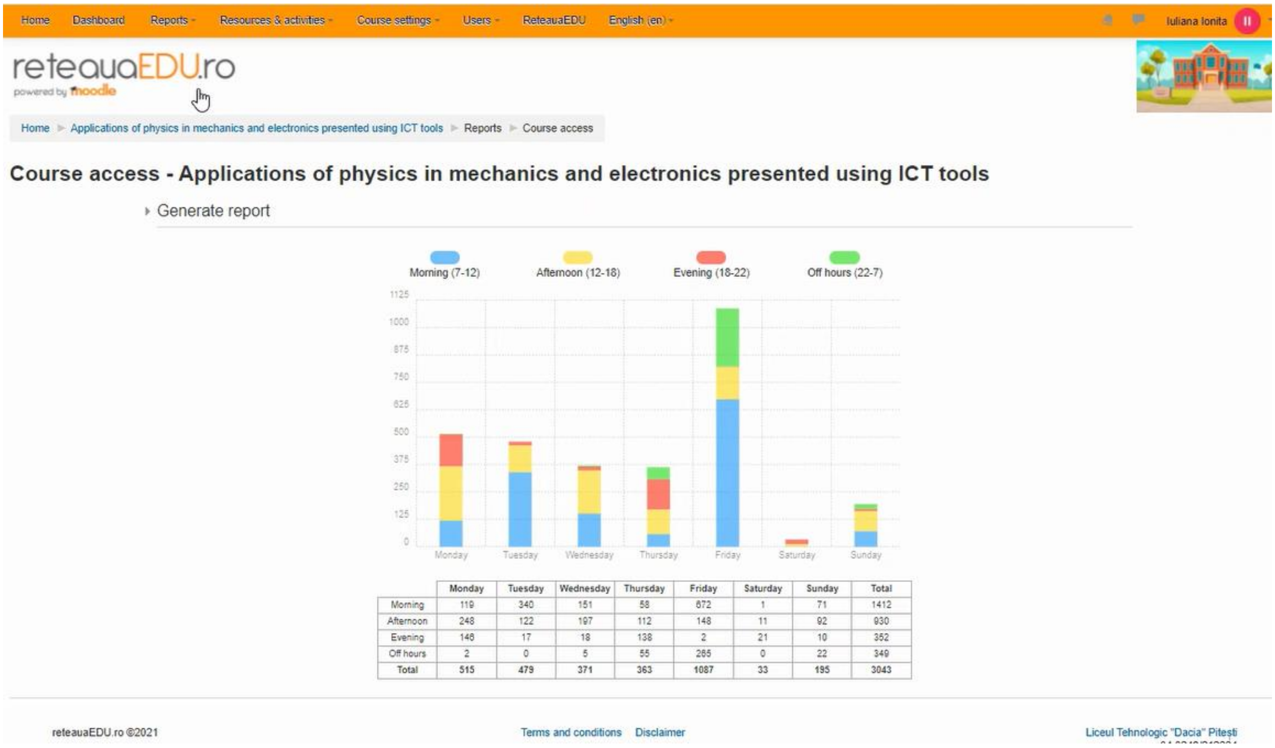
Teachers have the opportunity to generate reports such as:

1. the most active users,
2. reports on the access of the course,
3. the number of logins for certain periods of time,
4. course progress and the catalog of the course;
5. grader report.



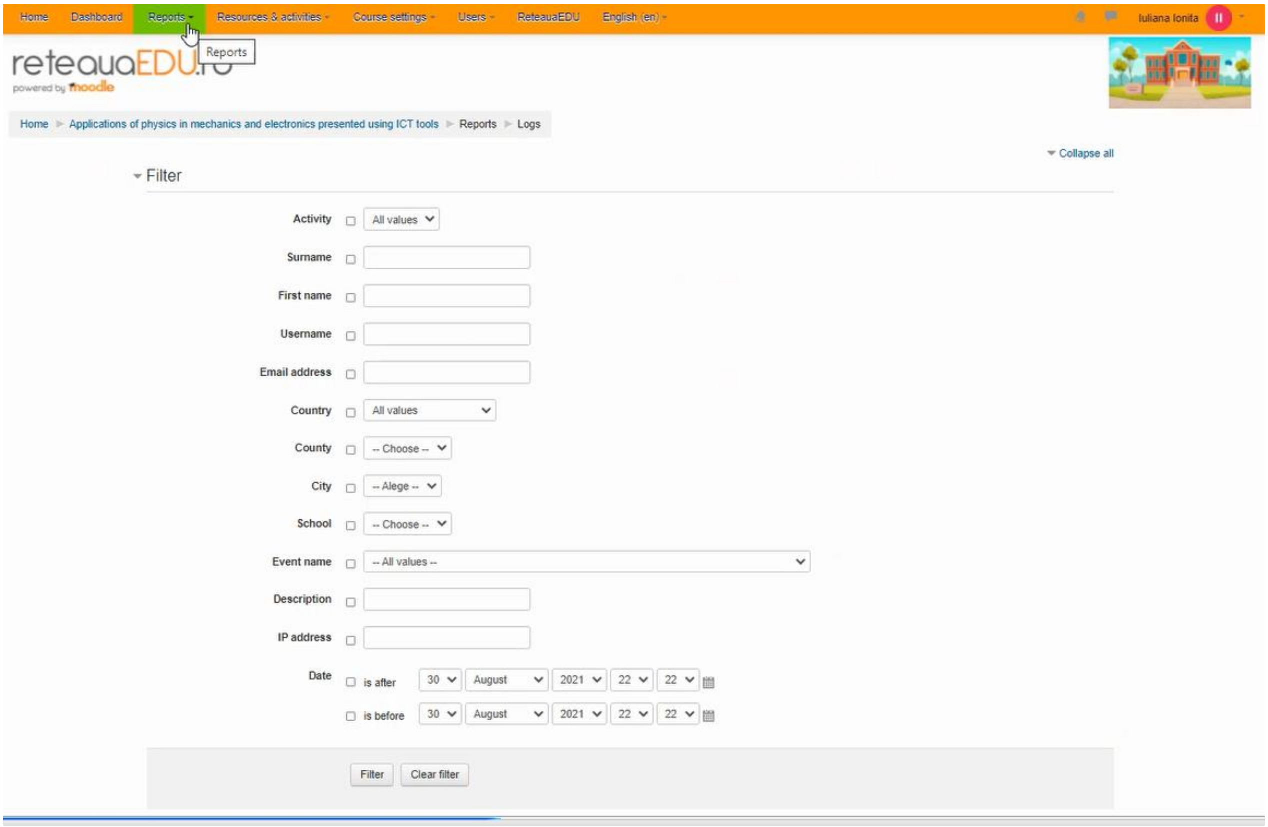
Most active users - Applications of physics in mechanics and electronics presented using ICT tools

Lastname	Firstname	Email	visits	Courses	Class	lastaccess

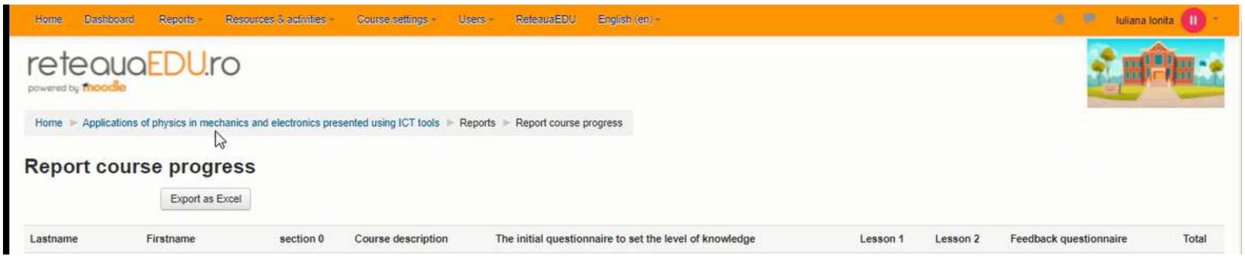


Course access - Applications of physics in mechanics and electronics presented using ICT tools

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Total
Morning	119	340	151	58	672	1	71	1412
Afternoon	248	122	197	112	148	11	92	930
Evening	146	17	18	138	2	21	10	352
Off hours	2	0	5	55	265	0	22	349
Total	**515**	**479**	**371**	**363**	**1087**	**33**	**195**	**3043**



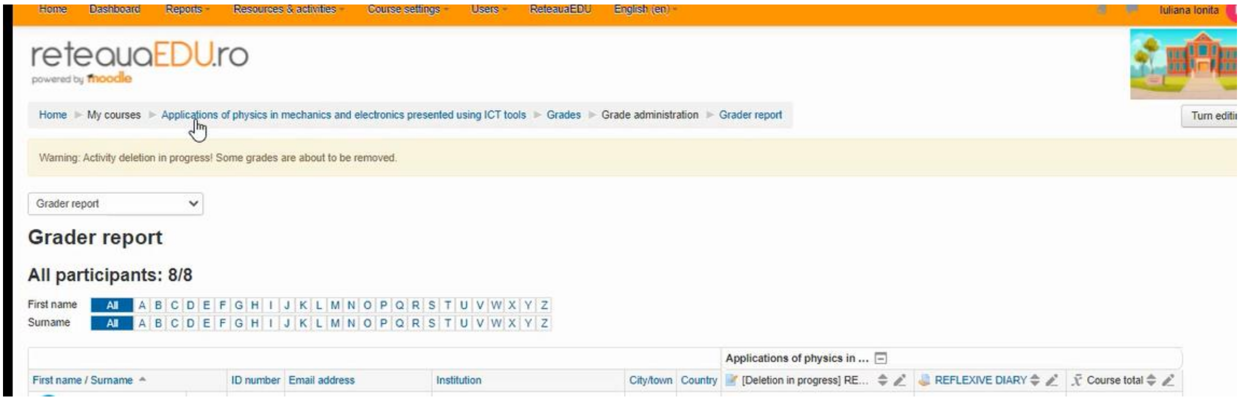
Activity	□	All values
Surname	□	
First name	□	
Username	□	
Email address	□	
Country	□	All values
County	□	-- Choose --
City	□	-- Alege --
School	□	-- Choose --
Event name	□	-- All values --
Description	□	
IP address	□	
Date	□ is after	30 August 2021 22 22
	□ is before	30 August 2021 22 22



Report course progress

Export as Excel

Lastname	Firstname	section 0	Course description	The initial questionnaire to set the level of knowledge	Lesson 1	Lesson 2	Feedback questionnaire	Total



Warning: Activity deletion in progress! Some grades are about to be removed.

Grader report

All participants: 8/8

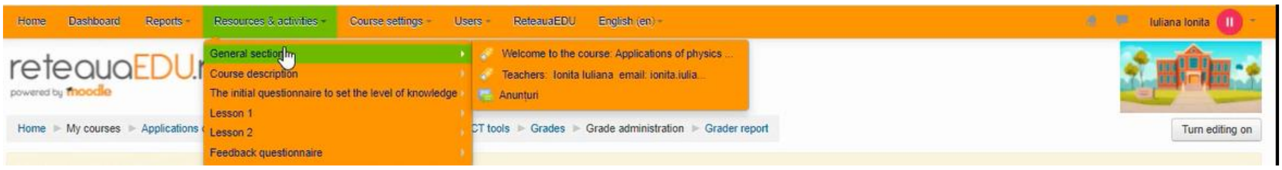
First name All A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

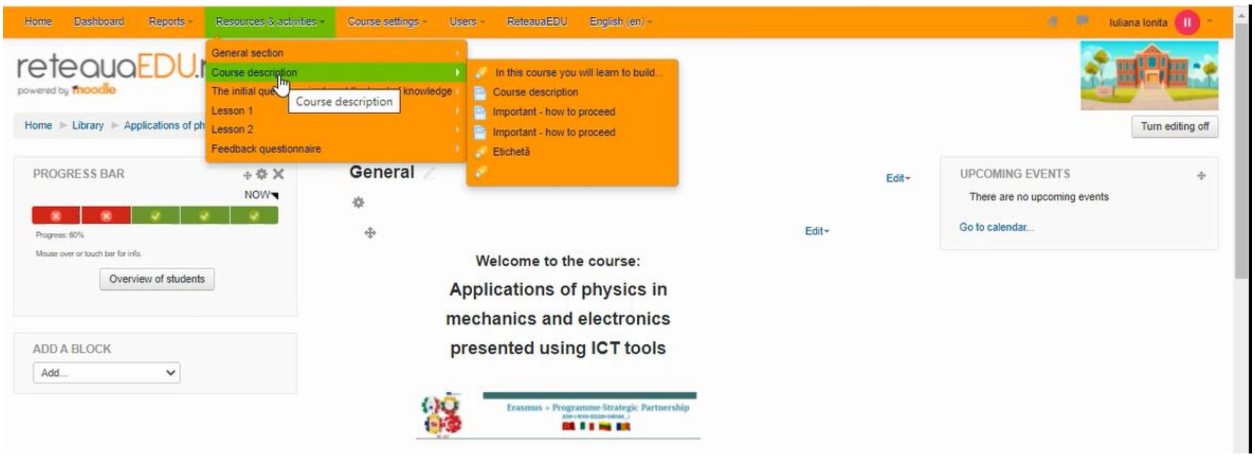
Surname All A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

						Applications of physics in ...		
First name / Surname	ID number	Email address	Institution	City/town	Country	[Deletion in progress] RE...	REFLEXIVE DIARY	Course total

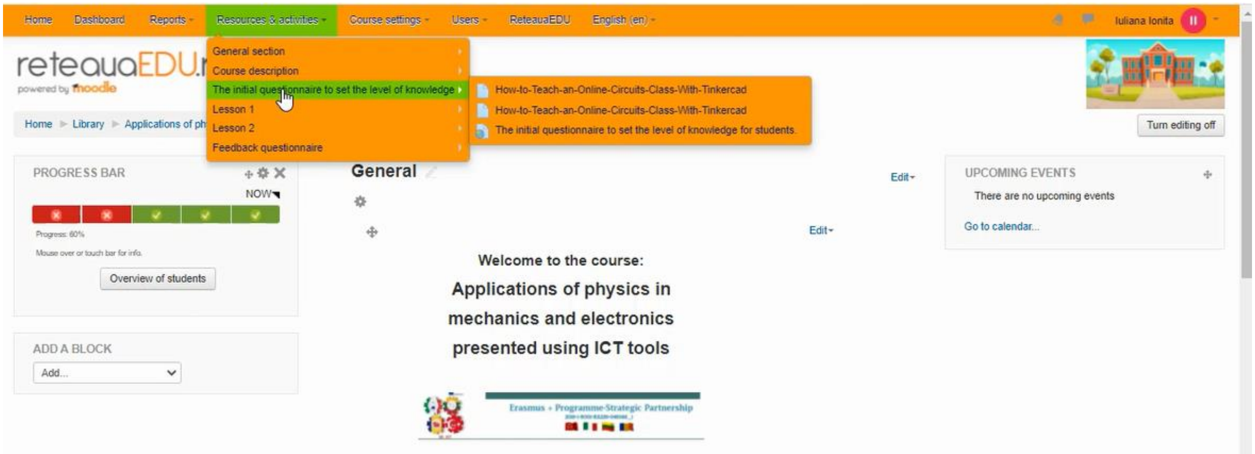
In the menu bar there is the option: ***Resources and activities*** of the course.

In the following :


Home
Dashboard
Reports
Resources & activities
Course settings
Users
ReteauaEDU
English (en)
Iuliana Ionita
reteauaEDU.ro
powered by moodle
General section
Course description
The initial questionnaire to set the level of knowledge
Lesson 1
Lesson 2
Feedback questionnaire
Welcome to the course: Applications of physics ...
Teachers: Ionita Iuliana email: ionita.iulia...
Anunturi
Home ▸ My courses ▸ Applications ... CT tools ▸ Grades ▸ Grade administration ▸ Grader report
Turn editing on



Welcome to the course:
Applications of physics in mechanics and electronics presented using ICT tools


Home Dashboard Reports Resources & activities Course settings Users ReteauaEDU English (en) Iuliana Ionita

reteauaEDU.r
powered by moodle

Home ▸ Library ▸ Applications of ph

- General section
- Course description
- The initial questionnaire to set the level of knowledge
  - How-to-Teach-an-Online-Circuits-Class-With-Tinkercad
  - How-to-Teach-an-Online-Circuits-Class-With-Tinkercad
  - The initial questionnaire to set the level of knowledge for students.
- Lesson 1
- Lesson 2
- Feedback questionnaire

Turn editing off

PROGRESS BAR
NOW
Progress: 60%
Mouse over or touch bar for info.
Overview of students

ADD A BLOCK
Add...

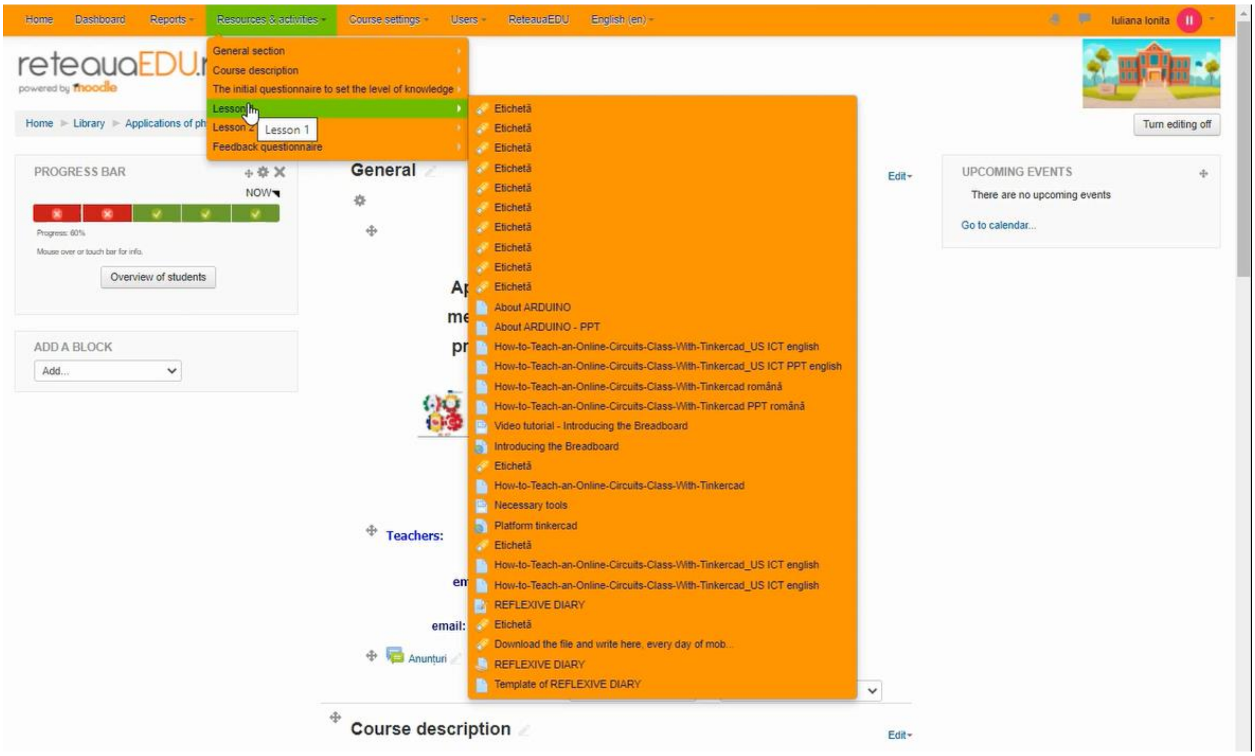
General

Edit
Edit

Welcome to the course:
Applications of physics in mechanics and electronics presented using ICT tools

Erasmus + Programme-Strategic Partnership

UPCOMING EVENTS
There are no upcoming events
Go to calendar...


Home Dashboard Reports Resources & activities Course settings Users ReteauaEDU English (en) Iuliana Ionita

reteauaEDU.r
powered by moodle

Home ▸ Library ▸ Applications of ph

- General section
- Course description
- The initial questionnaire to set the level of knowledge
- Lesson 1
  - Etichetă
  - Etichetă
  - Etichetă
  - Etichetă
  - Etichetă
  - Etichetă
  - Etichetă
  - Etichetă
  - Etichetă
  - Etichetă
  - About ARDUINO
  - About ARDUINO - PPT
  - How-to-Teach-an-Online-Circuits-Class-With-Tinkercad_US ICT english
  - How-to-Teach-an-Online-Circuits-Class-With-Tinkercad_US ICT PPT english
  - How-to-Teach-an-Online-Circuits-Class-With-Tinkercad română
  - How-to-Teach-an-Online-Circuits-Class-With-Tinkercad PPT română
  - Video tutorial - Introducing the Breadboard
  - Introducing the Breadboard
  - Etichetă
  - How-to-Teach-an-Online-Circuits-Class-With-Tinkercad
  - Necessary tools
  - Platform tinkercad
  - Etichetă
  - How-to-Teach-an-Online-Circuits-Class-With-Tinkercad_US ICT english
  - How-to-Teach-an-Online-Circuits-Class-With-Tinkercad_US ICT english
  - REFLEXIVE DIARY
  - Etichetă
  - Download the file and write here, every day of mob...
  - REFLEXIVE DIARY
  - Template of REFLEXIVE DIARY
- Lesson 2
- Feedback questionnaire

Lesson 1

Turn editing off

PROGRESS BAR
NOW
Progress: 60%
Mouse over or touch bar for info.
Overview of students

ADD A BLOCK
Add...

General

Edit

Teachers:

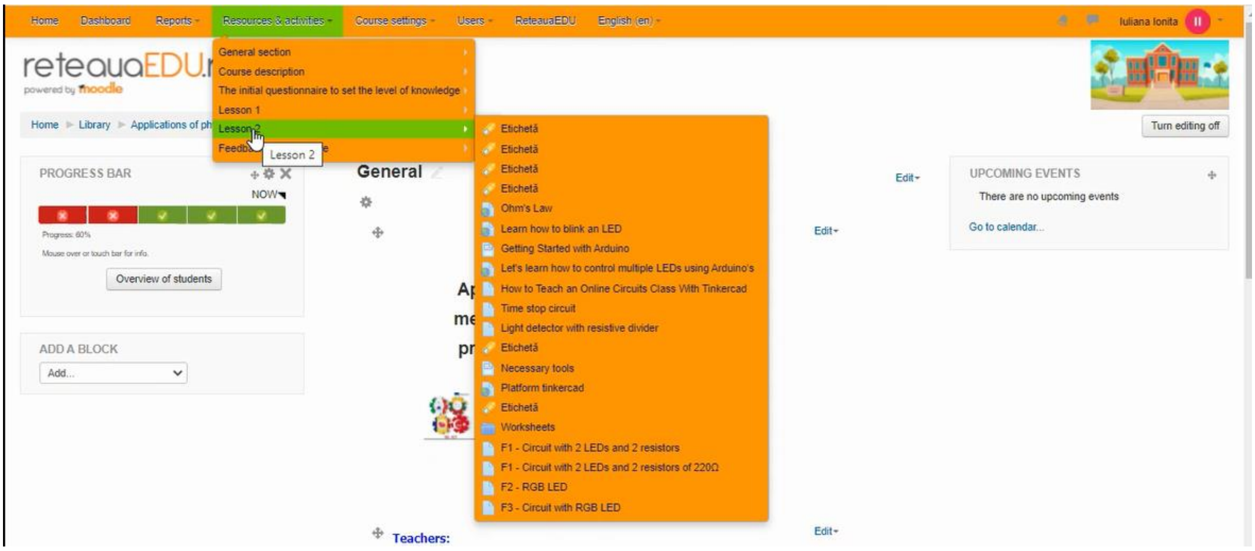
email:

Anunțuri

Course description

Edit

UPCOMING EVENTS
There are no upcoming events
Go to calendar...


Home Dashboard Reports Resources & activities Course settings Users ReteauaEDU English (en) Iuliana Ionita

reteauaEDU.r
powered by moodle

Home ▸ Library ▸ Applications of ph

- General section
- Course description
- The initial questionnaire to set the level of knowledge
- Lesson 1
- Lesson 2
  - Etichetă
  - Etichetă
  - Etichetă
  - Etichetă
  - Ohm's Law
  - Learn how to blink an LED
  - Getting Started with Arduino
  - Let's learn how to control multiple LEDs using Arduino's
  - How to Teach an Online Circuits Class With Tinkercad
  - Time stop circuit
  - Light detector with resistive divider
  - Etichetă
  - Necessary tools
  - Platform tinkercad
  - Etichetă
  - Worksheets
  - F1 - Circuit with 2 LEDs and 2 resistors
  - F1 - Circuit with 2 LEDs and 2 resistors of 220Ω
  - F2 - RGB LED
  - F3 - Circuit with RGB LED
- Feedb

Lesson 2

Turn editing off

PROGRESS BAR
NOW
Progress: 60%
Mouse over or touch bar for info.
Overview of students

ADD A BLOCK
Add...

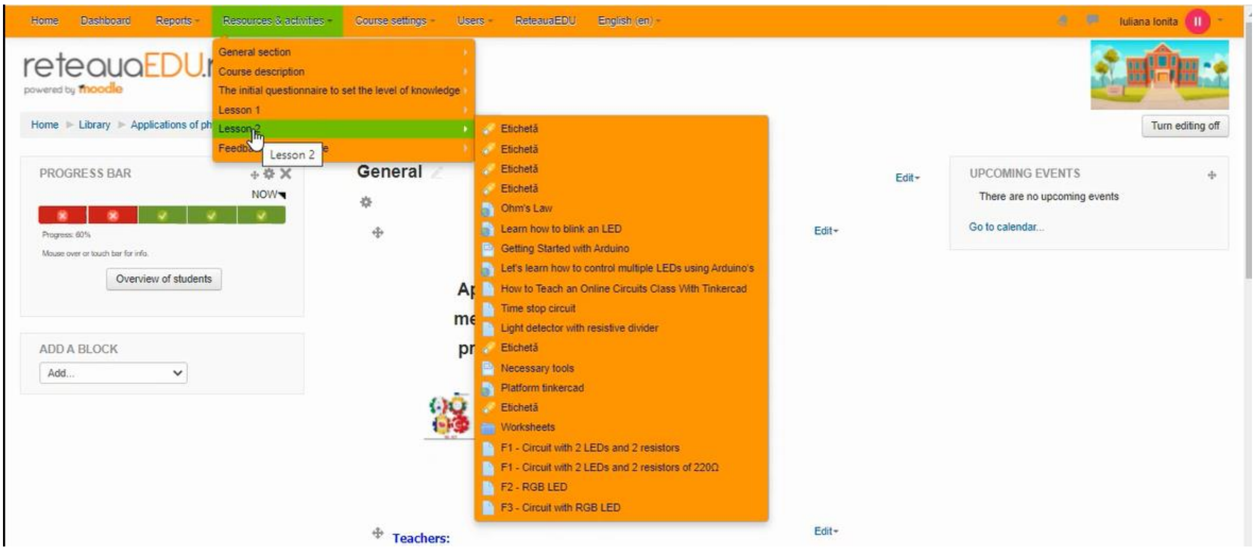
General

Edit
Edit

Teachers:

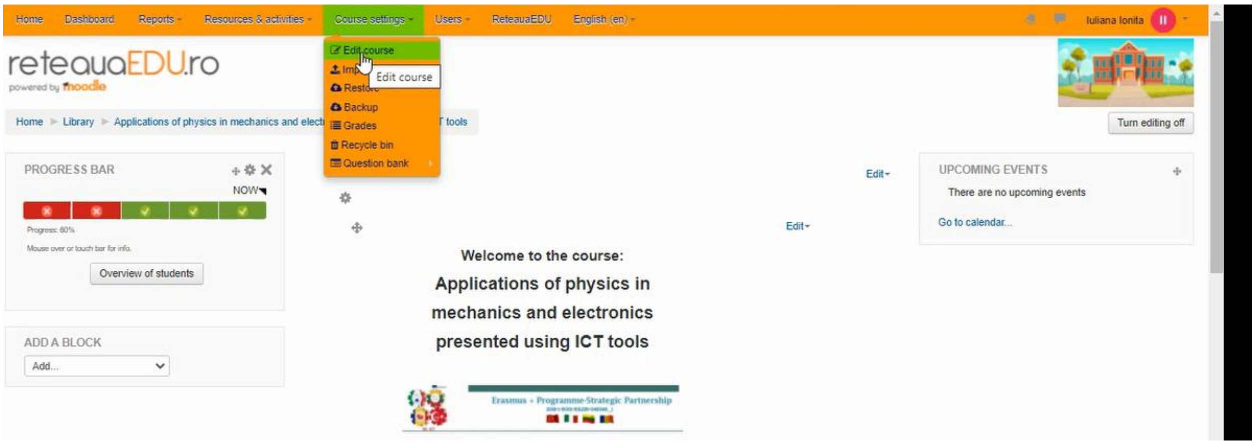
Edit

UPCOMING EVENTS
There are no upcoming events
Go to calendar...

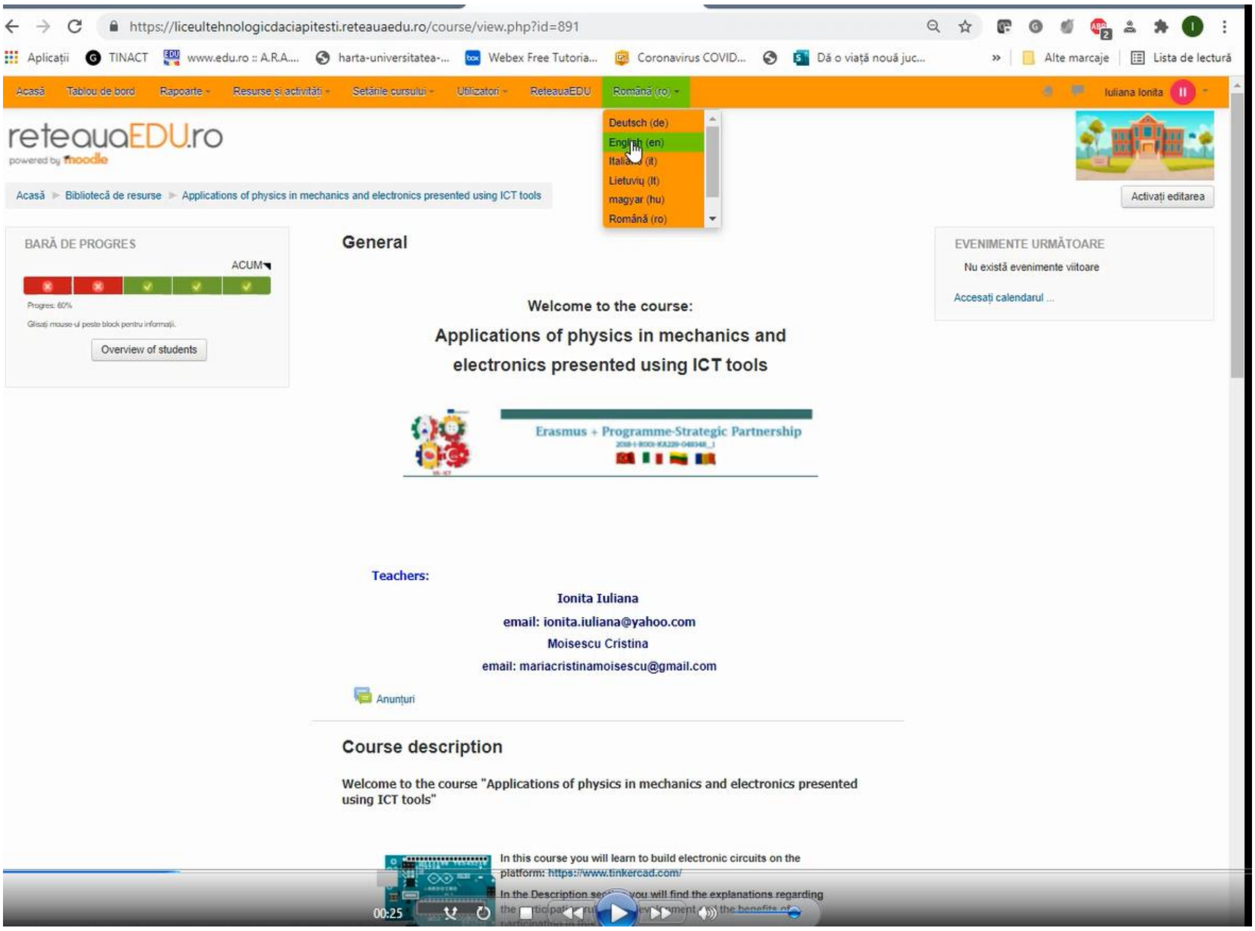

Home
Dashboard
Reports
Resources & activities
Course settings
Users
ReteauaEDU
English (en)
Iuliana Ionita
reteauaEDU.ro
powered by moodle
Home ▸ Library ▸ Applications of ph
Turn editing off
General section
Course description
The initial questionnaire to set the level of knowledge
Lesson 1
Lesson 2
Feedb
Lesson 2
Etichetă
Etichetă
Etichetă
Etichetă
Ohm's Law
Learn how to blink an LED
Getting Started with Arduino
Let's learn how to control multiple LEDs using Arduino's
How to Teach an Online Circuits Class With Tinkercad
Time stop circuit
Light detector with resistive divider
Etichetă
Necessary tools
Platform tinkercad
Etichetă
Worksheets
F1 - Circuit with 2 LEDs and 2 resistors
F1 - Circuit with 2 LEDs and 2 resistors of 220Ω
F2 - RGB LED
F3 - Circuit with RGB LED
PROGRESS BAR
NOW
Progress: 60%
Mouse over or touch bar for info.
Overview of students
ADD A BLOCK
Add...
General
Edit
UPCOMING EVENTS
There are no upcoming events
Go to calendar...
Teachers:

Another very useful menu of the platform, from the edit mode, is **Course Settings**, which contains the following submenus:

1. Edit course;
2. Import course;
3. Restore;
4. Backup;
5. Grades;
6. Recycle Bin;
7. Question Bank.



The language used on the platform can be set:



General

Welcome to the course:

Applications of physics in mechanics and electronics presented using ICT tools

Teachers:

Ionita Iuliana

email: ionita.iuliana@yahoo.com

Moisescu Cristina

email: mariacristinamoisescu@gmail.com

Course description

Welcome to the course "Applications of physics in mechanics and electronics presented using ICT tools"

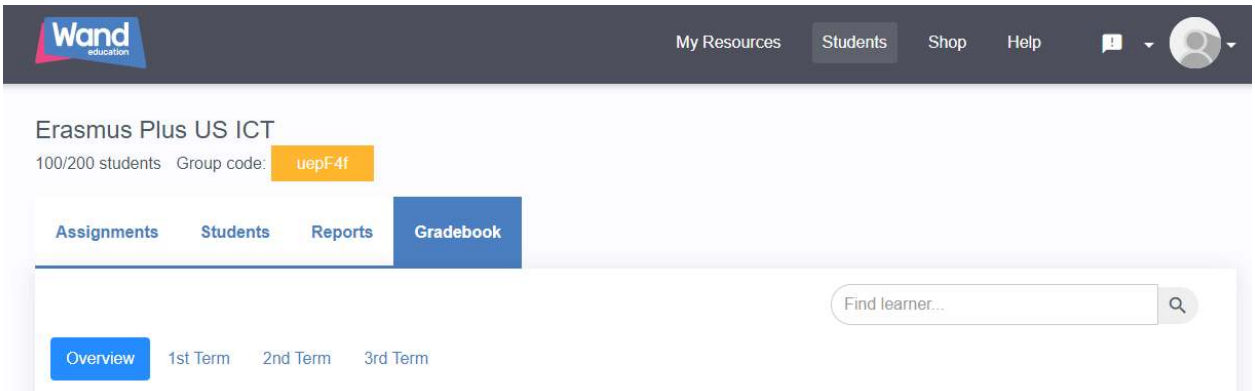
In this course you will learn to build electronic circuits on the platform: https://www.tinkercad.com/

In the Description sec[illegible] you will find the explanations regarding the [illegible]ticipati[illegible] [illegible]ment [illegible] the benefits of participating in this

About Platform wand. education

The platform *wand. education* is an e-learning platform that reduces the effort of preparing, teaching and evaluating lessons

It provides access to a library of pedagogical resources in continuous development. Lessons can be adapted or created. Worksheets, questionnaires can be made, the evaluation being done automatically. Automated reports are received to follow the progress of each student.[2]



Erasmus Plus US ICT

100/200 students Group code: uepF4f

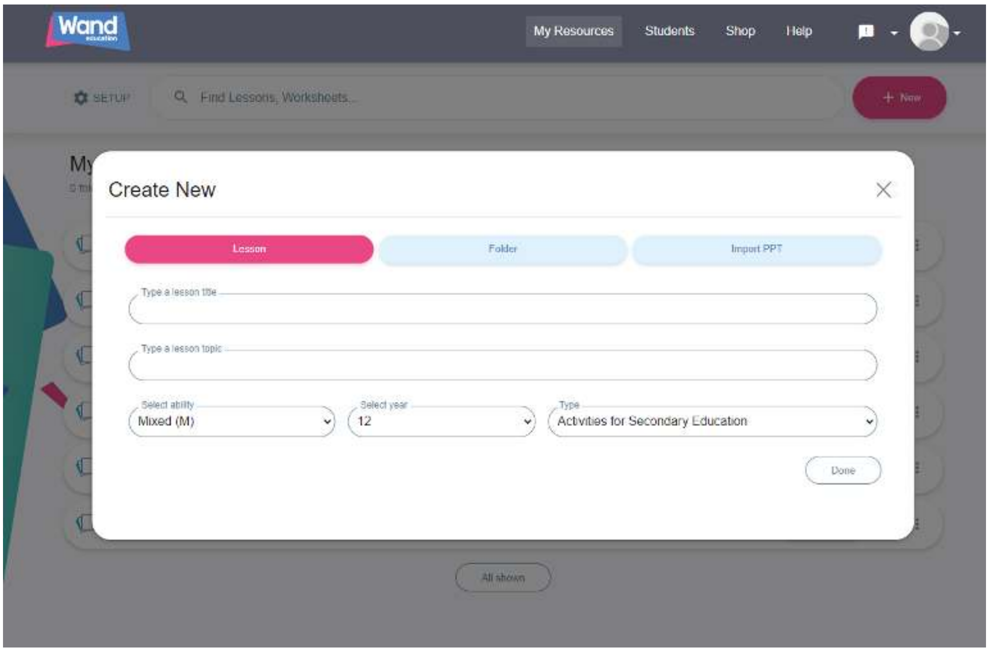
The interface is intuitive and easy to use by teachers of all specialties.

Classes can be created, students accessing the virtual class after creating a student account and receiving the class access code from the teacher.

Creating a lesson

Pressing the **New** button will start creating a new lesson. Pressing the New button will start creating a new lesson for a certain level or import a Power Point presentation.

[2] https://wand.education/?lang=ro



Create New

Type a lesson title

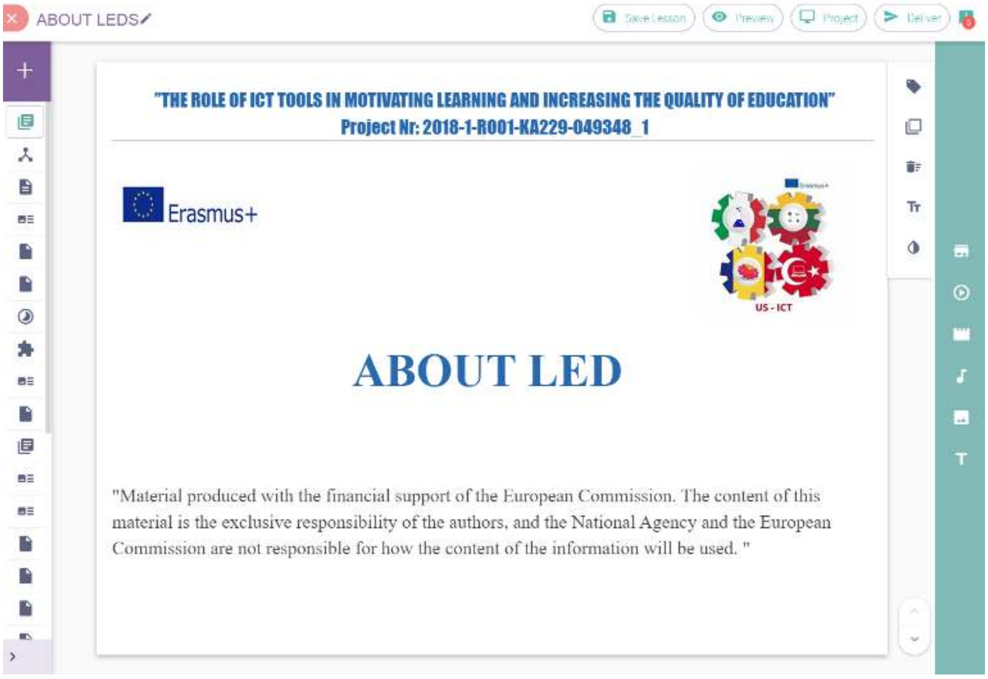
Type a lesson topic

Select ability: Mixed (M)

Select year: 12

Type: Activities for Secondary Education

Within the Erasmus Plus project, a presentation lesson was created on the wand.education platform:



"THE ROLE OF ICT TOOLS IN MOTIVATING LEARNING AND INCREASING THE QUALITY OF EDUCATION"
Project Nr: 2018-1-RO01-KA229-049348_1

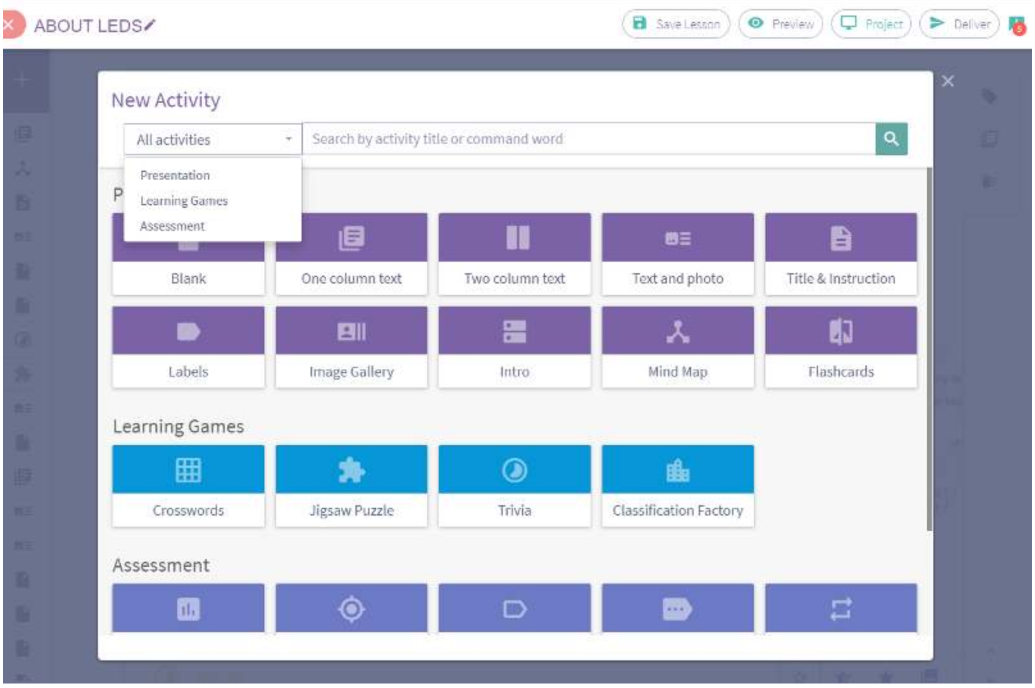
ABOUT LED

"Material produced with the financial support of the European Commission. The content of this material is the exclusive responsibility of the authors, and the National Agency and the European Commission are not responsible for how the content of the information will be used. "

The activities can be introduced in the lesson:

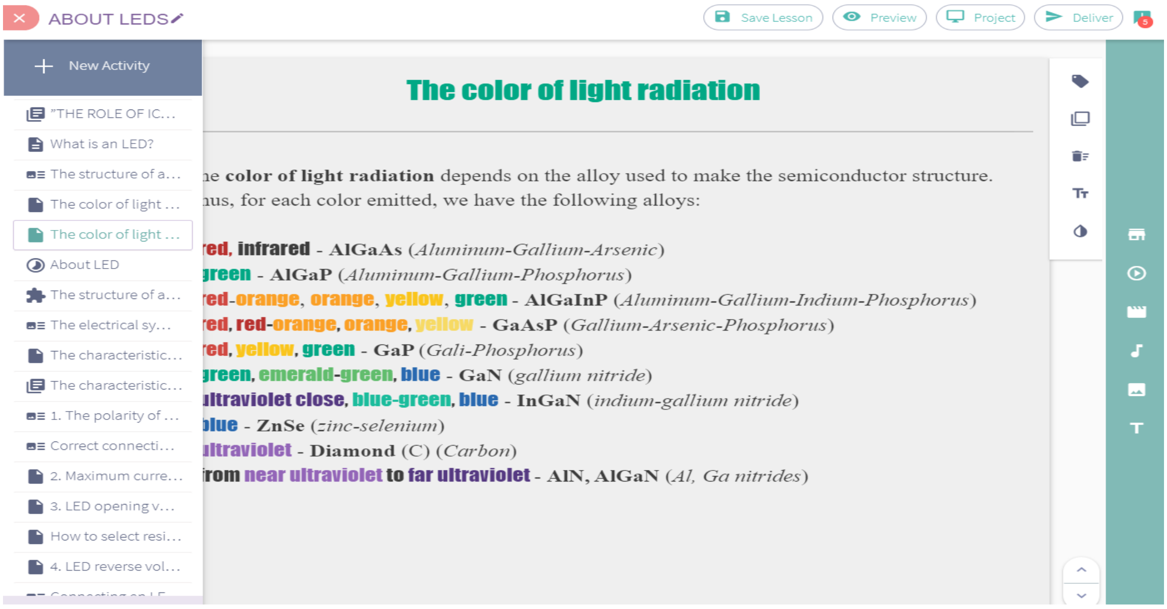
- ✔ Presentation;
- ✔ Learnig games;

- ✓ Assement.



The presentation of the lesson can have different shapes as seen in the image above.

Objects such as: images, movies, text can be inserted in each slide.



The color of light radiation

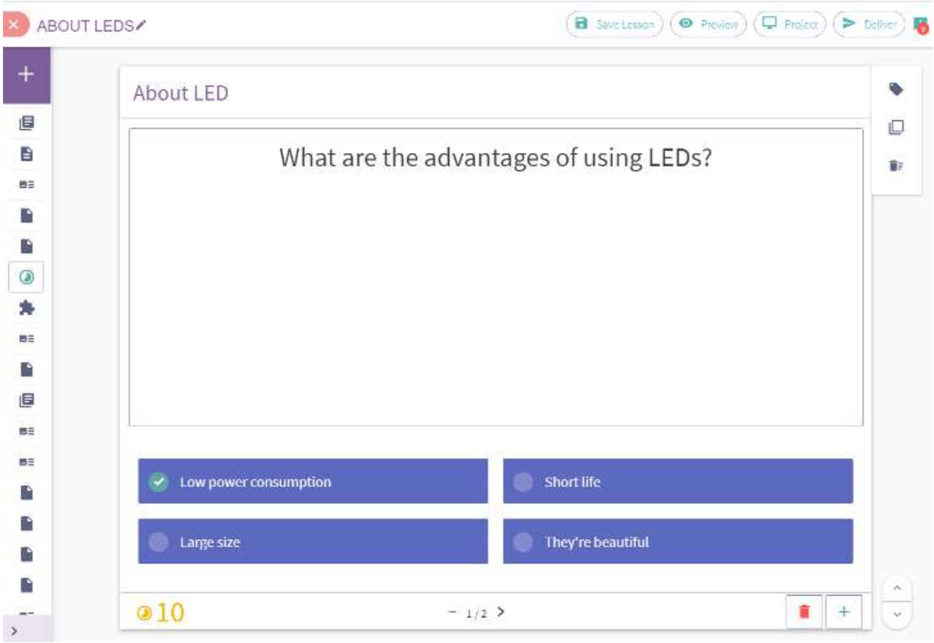
ne **color of light radiation** depends on the alloy used to make the semiconductor structure. nus, for each color emitted, we have the following alloys:

red, infrared - **AlGaAs** (*Aluminum-Gallium-Arsenic*)
green - **AlGaP** (*Aluminum-Gallium-Phosphorus*)
red-orange, orange, yellow, green - **AlGaInP** (*Aluminum-Gallium-Indium-Phosphorus*)
red, red-orange, orange, yellow - **GaAsP** (*Gallium-Arsenic-Phosphorus*)
red, yellow, green - **GaP** (*Gali-Phosphorus*)
green, emerald-green, blue - **GaN** (*gallium nitride*)
ultraviolet close, blue-green, blue - **InGaN** (*indium-gallium nitride*)
blue - **ZnSe** (*zinc-selenium*)
ultraviolet - **Diamond** (C) (*Carbon*)
from near ultraviolet to far ultraviolet - **AlN, AlGaN** (*Al, Ga nitrides*)

The types of learning games that can be inserted are:

- ✓ Crowswords;
- ✓ Jigsow Puzzle
- ✓ Trivia
- ✓ Classification Fasctory;

An example of Trivia learning game:

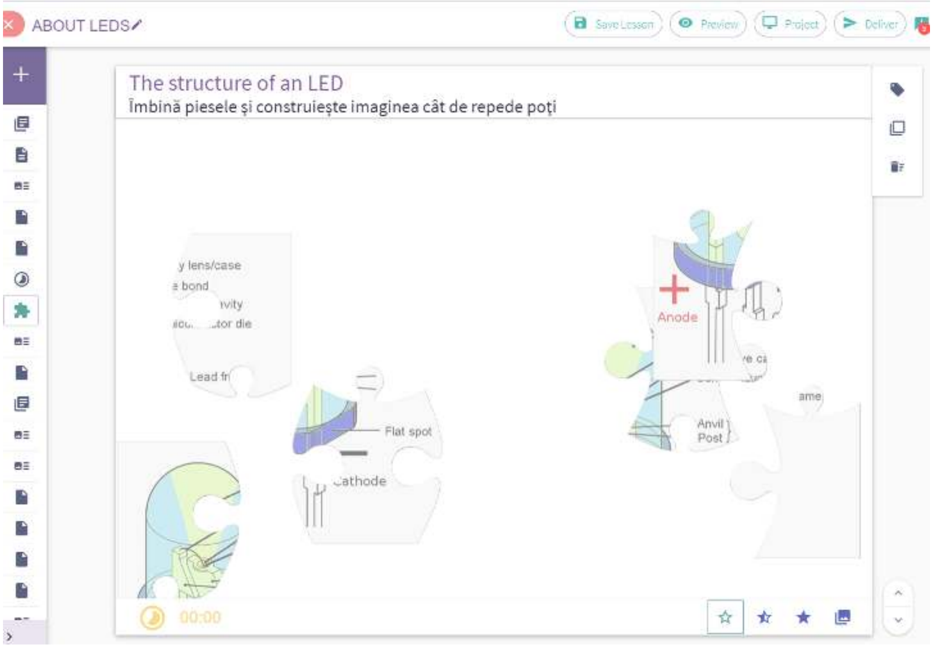


About LED

What are the advantages of using LEDs?

- ✓ Low power consumption
- ○ Short life
- ○ Large size
- ○ They're beautiful

An example of Jigsow Puzzle:

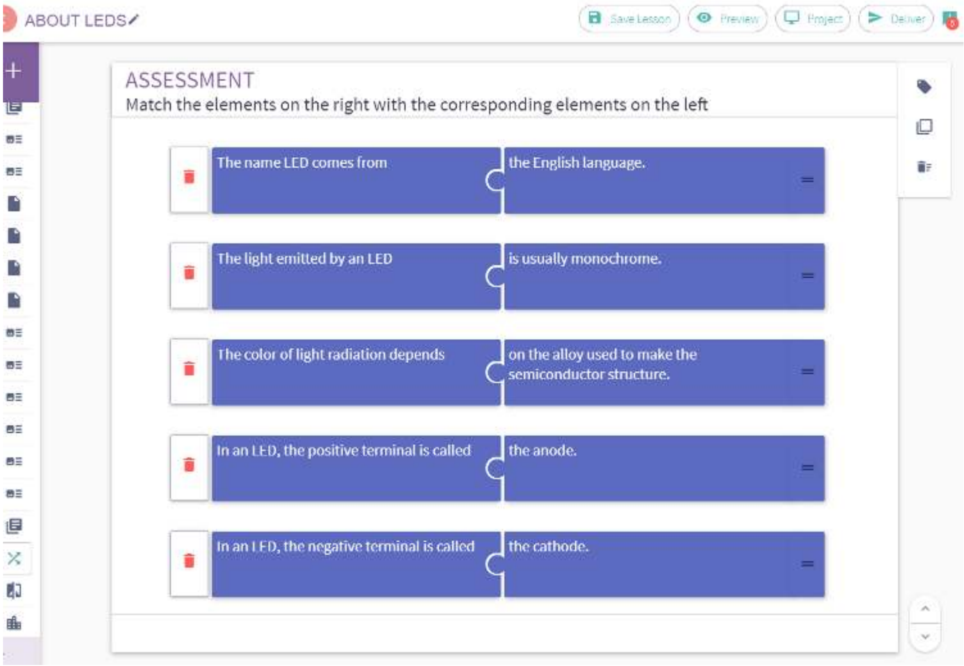


The structure of an LED

Îmbină piesele şi construieşte imaginea cât de repede poţi

The assessments can be in several ways:

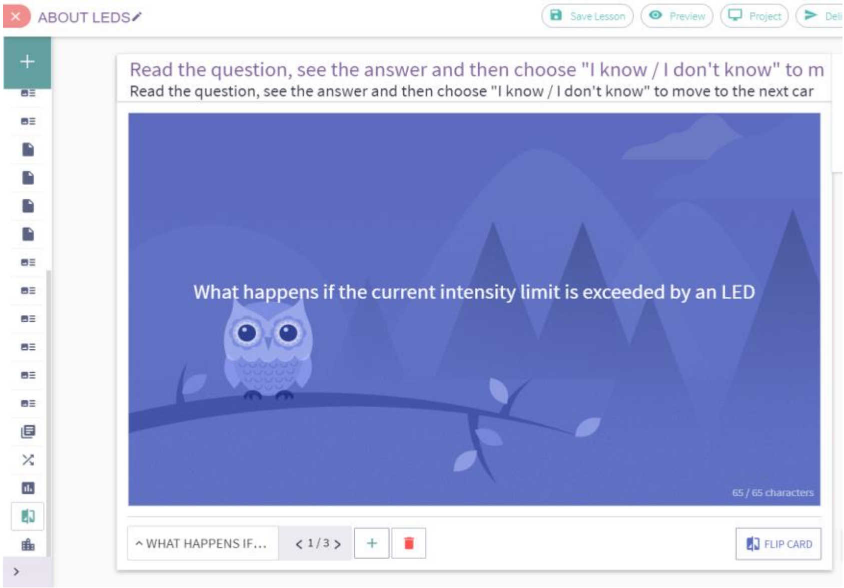
- ✔ summative assessment;
- ✔ hotspot;
- ✔ image matching;
- ✔ image fill in;
- ✔ ranking.



ASSESSMENT

Match the elements on the right with the corresponding elements on the left

The name LED comes from	the English language.
The light emitted by an LED	is usually monochrome.
The color of light radiation depends	on the alloy used to make the semiconductor structure.
In an LED, the positive terminal is called	the anode.
In an LED, the negative terminal is called	the cathode.



Read the question, see the answer and then choose "I know / I don't know" to m

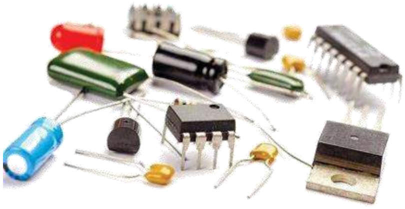
Read the question, see the answer and then choose "I know / I don't know" to move to the next car

What happens if the current intensity limit is exceeded by an LED

The platform is friendly, attractive and offers many educational resources made by teachers from different countries.

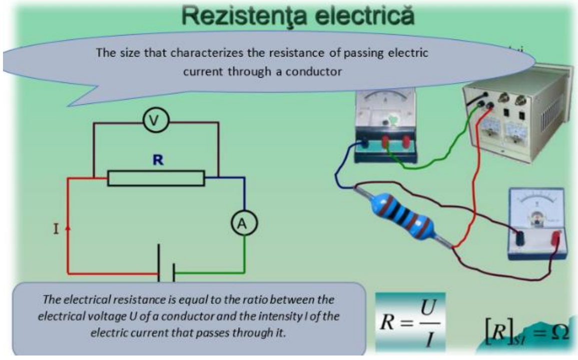
Circuit elements – Tinkercad

Circuit elements are devices used to make electrical circuits. They are of two types: passive (resistors, capacitors, coils) and active (diodes, transistors, integrated circuits, voltage sources). Active, as well as passive, elements can control / influence the flow of electricity in a circuit.



Passive circuit elements

1. Resistors (resistors) are components that limit the intensity of power when it passes through that resistor, according to Ohm's law (R = U / I). The unit of measurement used for electrical resistance (the physical quantity that characterizes it is the symbol ohm (Ω), with its multiples and submultiples). The value of the electrical resistance of a resistor is written on it in clear numbers and letters or using the color code.

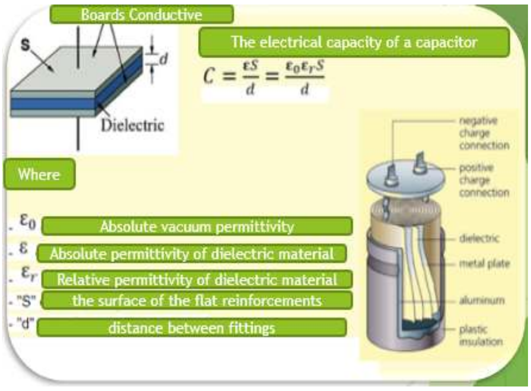


Rezistenţa electrică

The size that characterizes the resistance of passing electric current through a conductor

The electrical resistance is equal to the ratio between the electrical voltage U of a conductor and the intensity I of the electric current that passes through it.

$$R = \frac{U}{I} \qquad [R]_{SI} = \Omega$$



The electrical capacity of a capacitor

$$C = \frac{\varepsilon S}{d} = \frac{\varepsilon_0 \varepsilon_r S}{d}$$

Where

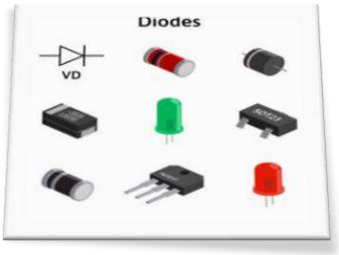
- ε_0 Absolute vacuum permittivity
- ε Absolute permittivity of dielectric material
- ε_r Relative permittivity of dielectric material
- "S" the surface of the flat reinforcements
- "d" distance between fittings

The capacitor is a passive component, the capacity of the plane capacitor is given by the dielectric medium that separates the two armatures and has the permittivity, the distance between the armatures - d, and the plane armature surface -S

When passing through an electrical circuit, the initial electric current creates a magnetic field around the conductors through which it passes. This field creates in the electrical conductor some electric power induced by the opposite direction, which tends to slow the initial one. This slowing phenomenon occurs only when the initial power changes its value (increases or decreases) and is called self-induction. If an (initial) electric power flows through a part, the inductance represents the capacity of that circuit element to oppose the change of the initial power intensity, by creating (inducing) a electric power of opposite direction

Active circuit elements

An active circuit element is any type of component that can control the movement of electrons (current) electrically. For a circuit to be called an electronic circuit, it must contain at least one such active element.



Active elements include, but are not limited to, vacuum tubes, transistors, semiconductor rectifiers, and triacs.

All active devices control the current through them. Some active devices do this through a voltage and others through a current. Those that use a static voltage as a control signal are called voltage controlled devices. Those that use another current to control the current in question are known as current controlled devices.

Platform Tinkercad

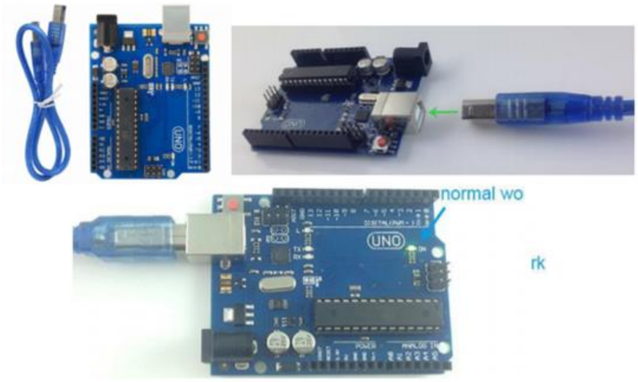
Tinkercad is a free online collection of software tools that help people all over the world think, create and make. We’re the ideal introduction to Autodesk, the leader in 3D design, engineering and entertainment software.

Assemblies using tinkercad And ardiuino platforms

The assemblies are intended to be physically checked using a prototype and various components, so that only the program can be simulated, but it is advisable to create the electronic assembly.

An Arduino board consists of an 8-bit, 16-bit or 32-bit Atmel AVR microcontroller with complementary components that facilitate programming and incorporation into other circuits.

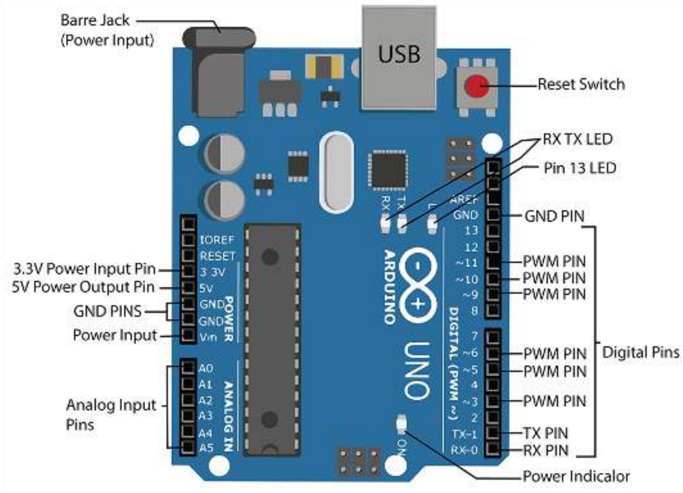
An important aspect of Arduino is that it has standard connectors, which allow the user to connect the board with the processor to various interchangeable modules called shields. Some shields communicate with the Arduino directly via digital or analog pins, but others are individually addressable via the I²C serial main line allowing the use of several modules in parallel.



The 14 digital input / output pins are located at the top, and the 6 analog inputs are on the right side, below the microcontroller.

Unlike most previous programmable circuit boards, the Arduino does not need separate pieces of hardware (programmer) to load new code onto the board using a simple USB cable. In addition, the Arduino IDE uses a simplified version of C ++, making it easier to learn programming. Arduino offers a standard form factor that eliminates microcontroller functions in a more affordable package. UNO is one of the most popular boards in the Arduino family.

It only takes 10 lines of code in the Arduino IDE to be able to make a screen come to life.

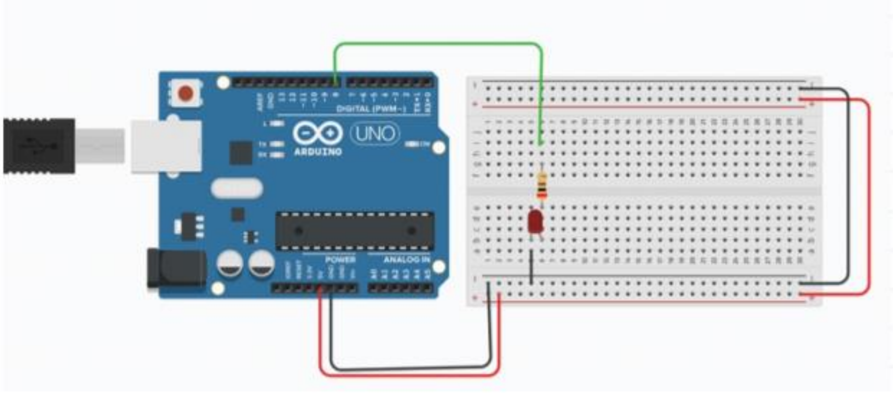


Examples of applicability

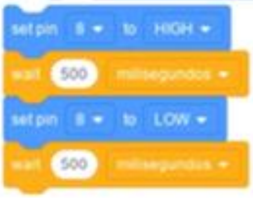
The flashing LED. Using the Tinkercad circuits, we can create a circuit with Arduino Uno, on a connection board using breadboard and a LED connected to the Arduino 8 digital PIN.

For programming we can use blocks (pseudocodes) or codes we can write a program that makes the LED flash once every second. After the virtual realization of the circuit we simulate the circuit from Tinkercad.

There is an Arduino output when the voltage is higher than 5V. The LED, when it illuminates, has a power of almost 2V, so Vr = 5V-Vled = 3V. For the LED to shine correctly we will use an amount of electricity of 15mA, so R = 3V / 150mA = 200 ohm The code is the following: Pin 8 is used and we will design in codes and blocks.



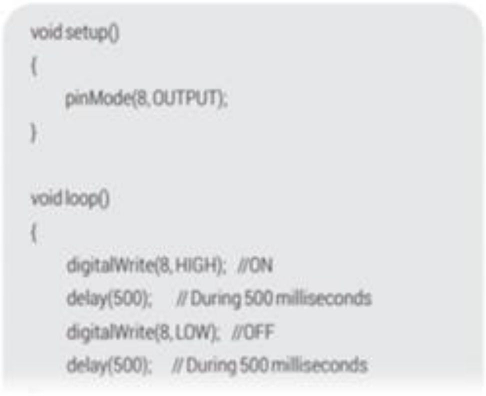
Blocuri



```
set pin 8 to HIGH
wait 500 milisegundos
set pin 8 to LOW
wait 500 milisegundos
```

It runs in an infinite loop

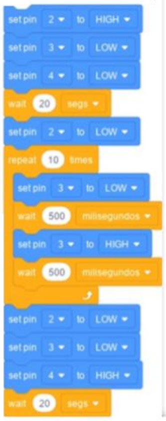
Cod



```
void setup()
{
    pinMode(8, OUTPUT);
}

void loop()
{
     digitalWrite(8, HIGH);  //ON
     delay(500);      // During 500 milliseconds
     digitalWrite(8, LOW);  //OFF
     delay(500);     // During 500 milliseconds
```

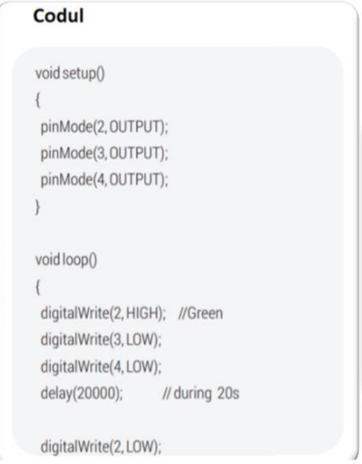
Using the Tinkercad circuits, a breadboard connection board and three LEDs (red, yellow, green), connected to pins 2, 3, 4, we can make a circuit with Arduino Uno. Using the blocks, we will write a program that creates a traffic light: the green light will be on for 20 seconds, then the yellow light will flash 10 times and then the red light will be on for 20 seconds

Blocuri

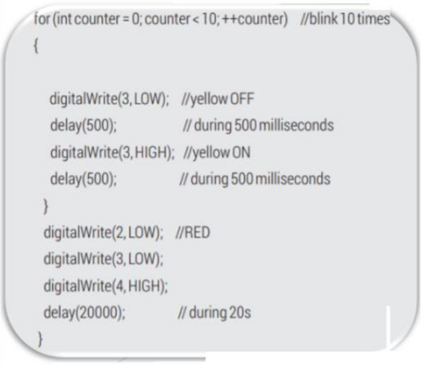


```
set pin 2 to HIGH
set pin 3 to LOW
set pin 4 to LOW
wait 20 segs
set pin 2 to LOW
repeat 10 times
  set pin 3 to LOW
  wait 500 milisegundos
  set pin 3 to HIGH
  wait 500 milisegundos
set pin 2 to LOW
set pin 3 to LOW
set pin 4 to HIGH
wait 20 segs
```

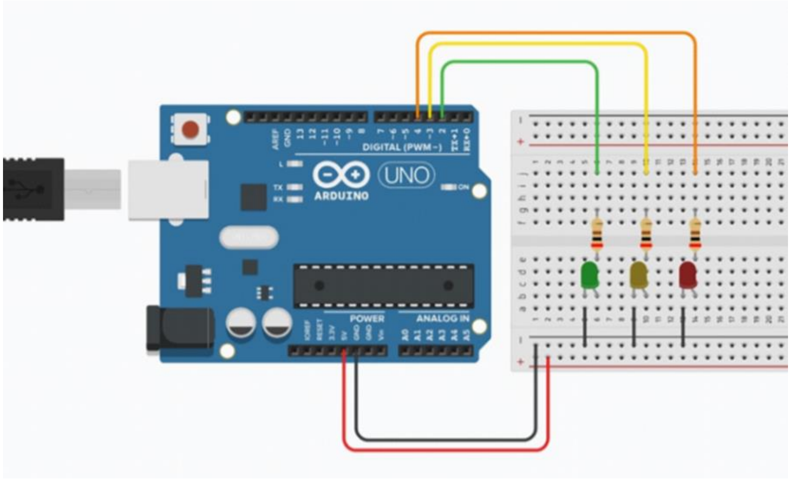
Codul



```
void setup()
{
 pinMode(2, OUTPUT);
 pinMode(3, OUTPUT);
 pinMode(4, OUTPUT);
}

void loop()
{
 digitalWrite(2, HIGH);   //Green
 digitalWrite(3, LOW);
 digitalWrite(4, LOW);
 delay(20000);        // during 20s

 digitalWrite(2, LOW);
```



```
for (int counter = 0; counter < 10; ++counter)   //blink 10 times
{

  digitalWrite(3, LOW);   //yellow OFF
  delay(500);            // during 500 milliseconds
  digitalWrite(3, HIGH);  //yellow ON
  delay(500);            // during 500 milliseconds
 }
 digitalWrite(2, LOW);   //RED
 digitalWrite(3, LOW);
 digitalWrite(4, HIGH);
 delay(20000);           // during 20s
}
```



THE SOURCES OF INFORMATIONS

https://www.crestem.org/index.php?id=scopul-si-obiectivele

https://blog.robofun.ro/2019/09/11/educatia-stem-ce-este-si-de-ce-este-importanta-pentru-copii/

https://www.britannica.com/technology/Visual-Basic

http://msdn.microsoft.com/

https://www.html.it/guide/guida-visual-basic/

https://wand.education/?lang=ro

Learn to program with SCRATCH - Majed MARJI

Learn to code with SCRATCH - The MapPi Team

https://en.wikipedia.org/wiki/Lego

https://en.wikipedia.org/wiki/Lego_Mindstorms_EV3

https://education.lego.com/en-us/product-resources/mindstorms-ev3/downloads/building-instructions

https://education.lego.com/en-us/lessons

https://www.tinkercad.com/things/2n2GLkEXBh4-turn-on-1-led-using-arduino-in-limba-romana-romanian-language-

Microsoft MVP Windows Development Passionate

NET Developer focused on Windows, Xamarin and Azure

Co-Founder of DotNetToscana User Group

Authors word processing:

1. ***Arianna Sturniolo***
2. ***Hüseyin Yaşar Konak***
3. ***Ionita Iuliana***
4. ***Jolita Stankevičienė***
5. ***Moisescu Cristina***
6. ***Măcaneata Anca***
7. ***Soare Angelica***

”THE ROLE OF ICT TOOLS IN MOTIVATING LEARNING AND INCREASING THE QUALITY OF EDUCATION”

Project Nr: 2018-1-RO01-KA229-049348_1

"Material produced with the financial support of the European Commission. The content of this material is the exclusive responsibility of the authors, and the National Agency and the European Commission are not responsible for how the content of the information will be used. "