

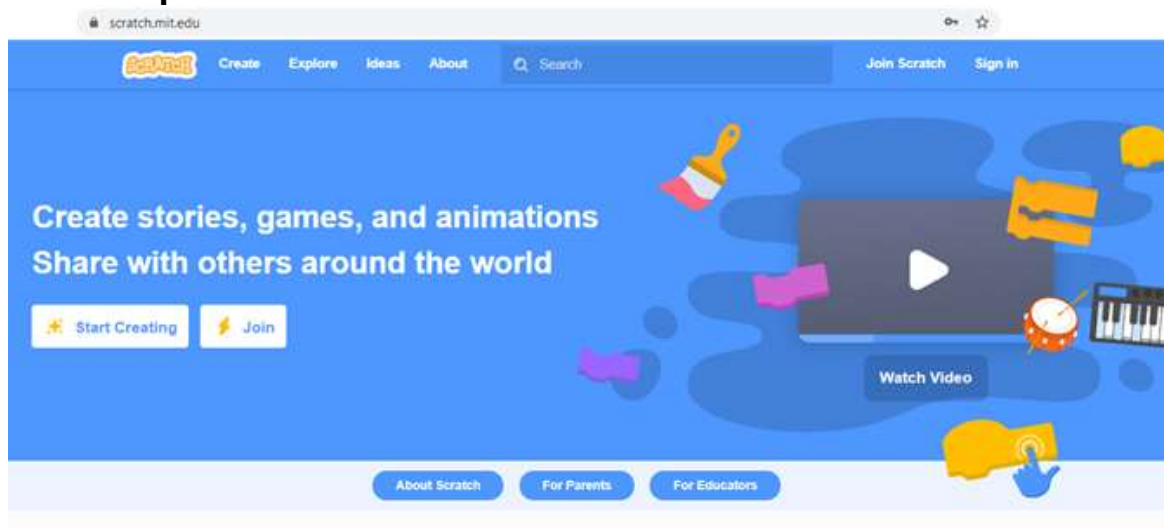


Programming SCRATCH

Step-by-step

Material needed:

- One computer or tablet with Internet connection












Step-1 – creating your account

- Access at the website <http://scratch.mit.edu> and click at “Join Scratch”
- Then choose for yourself a username that you think suitable for you, it will be your programmer ‘*nickname*’. ☺
- **(Very important!! Don't use your real name)**
- Choose for your self a complex password (with letters, numbers and Special Characters) with at least 8 characters long
- After that, reply your birthday, and your country.
- - And for last, a valid email address that you access. ... register done!



- Step-2 – Learn blocks and their behaviour

Types of Blocks

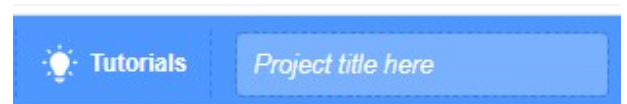
 Motion	Motion - deal with the movement of sprites.
 Looks	Looks – control how a sprite looks
 Sound	Sound - are the blocks that control sound
 Events	Events - control events and the triggering of scripts
 Control	Control - blocks that control scripts
 Sensing	Sensing - associate with sprites and the stage detecting conditions.
 Operators	Operators - blocks that perform math functions and string handling.
 Variables	Variables - blocks that hold variables and lists.
 My Blocks	My Blocks - are user-made custom blocks

- Step-3 – Creating the game project “MAZE”

- After login, press the Create hyperlink to start your own project.



- Then you should title the project, with a name you think suitable





Step-4 – Creating Stage Backdrop

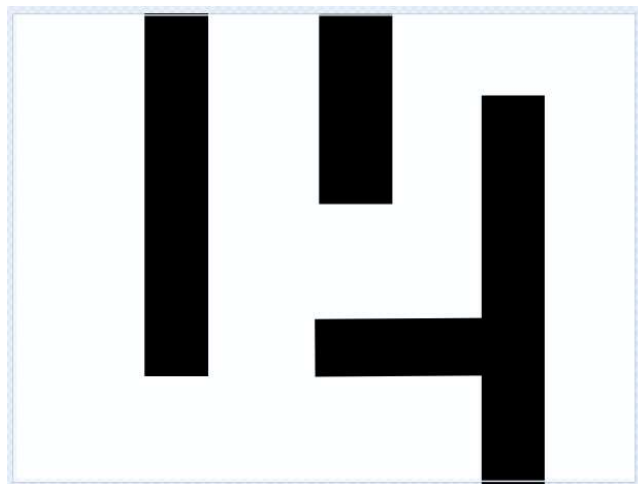
2nd - click here

3rd - click here to start building your maze using drag and drop with the mouse

1st click here

Attention - The walls of the maze must be larger than the objects that you are going to use.

You may and should create at beginning simple maze as show above



Once the backdrop created we continue forward, it is always possible to make changes at the backdrop.



Step-5 – Creating the Sprite

For this simple exercise game, we going to create 2 simple sprites.

- A ball
- a small rectangle that will be our goal.

3rd draw here the simple forms to the project for the other form repeat the procedure

If you wish you may delete this sprite, it will be of no use for this maze

2nd click here → Paint

1st click here → Choose a Sprite

After doing it you should have two sprites as shown above

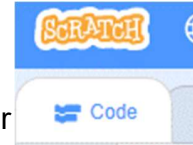


Don't forget to name the sprite





Step-6 – Coding the game objective



- We going to start the coding by assuring we select the separator

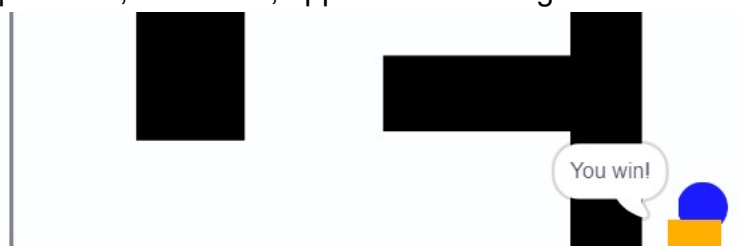
Now we are going to code, each of the sprites, once this is a guided DiY we share the code, it works with drag and drop to place it

Code for the Finish

- Make sure you have the sprite finish selected

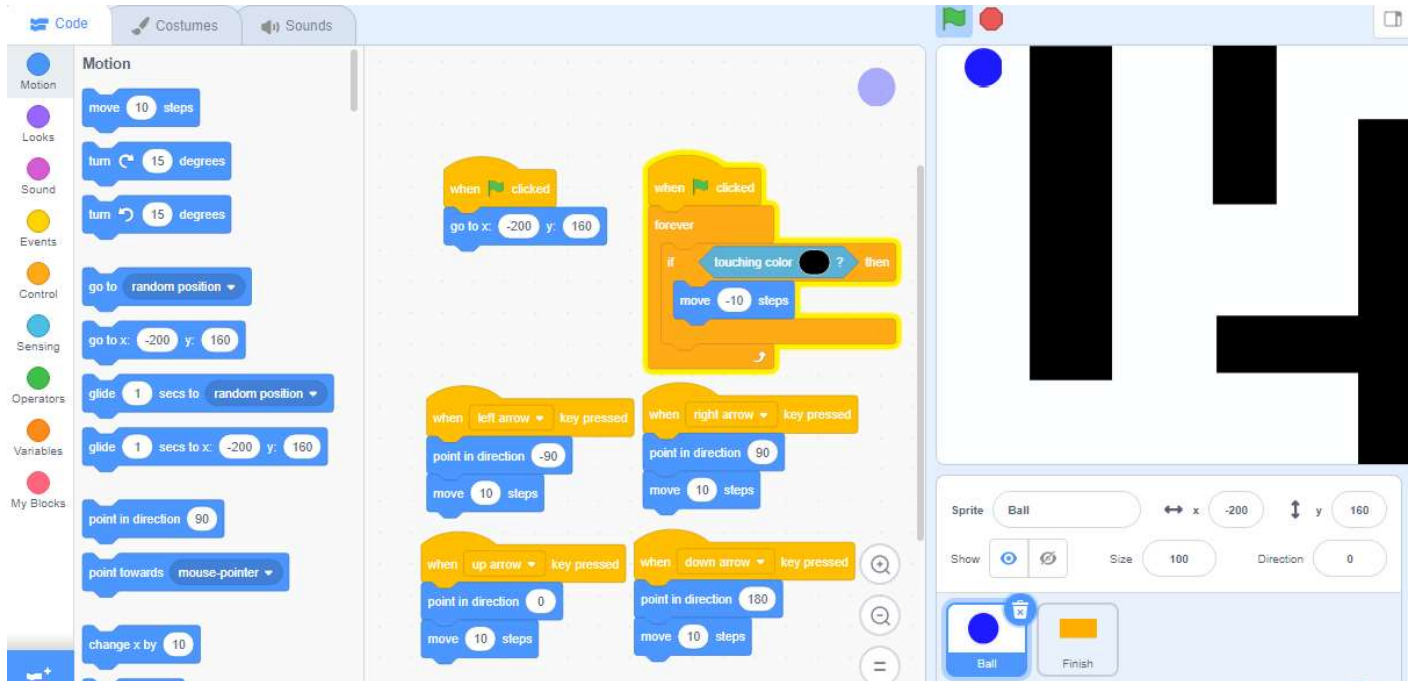
- Confirm that the sprite Finish is placed in a part of the maze that allows the Ball to arrive through the maze

This complete set of coding, allows the sprite **Finish**, to know through a Sensing command that when the sprite **Ball**, touches it, appear the message “You Win!





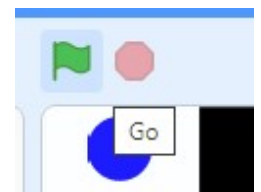
Code for the Ball



The code for the sprite ball is more complete, once we need it to have more 'actions', so it is defined the start, placing the sprite in the same coordinates every time we execute the code, also with the start it's implied that when the "Ball" touch a black color it will go back 10 steps.

You will need also to define the keys to control the sprite ball and the way it moves, in the example we used the cursor keys, but you may define other keys, or even put one more sprite Ball and that allows two players at the same time.

Once it is all settled, arrives the time to test it, so you go at the green Flag And enjoy.



If you confirm that work, it is possible to improve the game, for instance with:

- increase one other player at the same time
- Input sound when the ball touch the color
- Set a timer
- Create a new backdrop, and setting it as level 2

Now you are able to... try, fail, retry, improve, build, create and Above all...
To have the fun in learning!



3 common mistakes and how to avoid them:

Mistake 1: Complexity of the project

At start you should create easy projects, so that with the developing of projects your knowledges may enhance, while creating and evolving your know-how with the programming and its usages.

Mistake 2: Stop coding in the first fail

Coding not always go as we wish, sometimes we fail (it happens to everyone, sometimes a lot 😊), then we should stop to think, looking to our project (what i want to happen?, how it is supposed to happen?, how can I make it happen with the available tools?)

And sometimes we can achieve the knowledge that it is not yet possible to make it happen, but for sure something nearby is possible to do. 😊 above all it is important to be resilient.

In the coding it's common to place the code in the work sprite, and always take quite a time until we find the mistake 😊

Mistake 3: The place of the blocks...

It happens a lot some mistakes with the coding blocks, despite the fact that they don't fit where they can't be placed, but somehow, we try it... and amazing it doesn't work. 😊 also, is common use the wrong (lookalike) block.

It is not strange when we put the blocks in the wrong order... and then they don't work, amazed by that we stay all the times we do it.

And one of the most common at the place of blocks... put the wrong block in the bracket, it makes a huge difference whether a block is inside or outside a repeat block bracket, it is important check the blocks near the end to make sure that the right blocks are inside of it.