

LEARNING TO CODE

THEME	Programming
FORMAT	Collective (all students participate simultaneously)
PREPARATION TIME	1 hour
ACTIVITY LENGHT	30-45 minutes
DIFFICULTY LEVEL	Low

PEDAGOGICAL GOAL

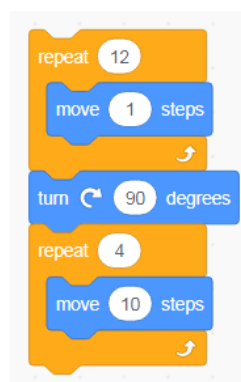
This activity seeks to introduce the idea of programming, simulating the operation of a computer program with blocks of paper or EVA. Following an ordered sequence of steps created by the class (code), a student must leave one point in the room to reach another predetermined point.

The activity has as specific objectives:

- Understand how a computer program works
- Relate programming to an ordered sequence of steps
- Understand the importance of creating precise instructions so that what we imagine is executed by our program
- Familiarize yourself with the way the blocks fit in Scratch

NECESSARY MATERIALS

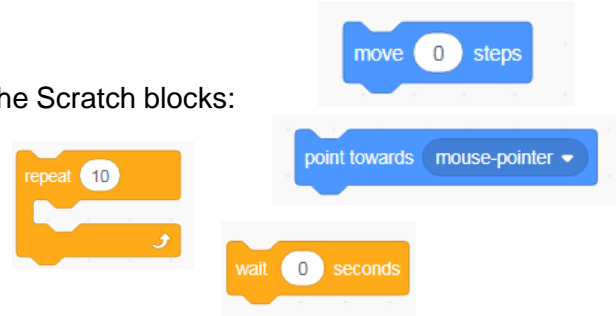
- EVA or paperboard (blue,yellow or orange, white)
- Scissors
- Paper
- Double-sided tape



LEARNING TO CODE

Preparation:

- Cut the sheets of EVA or cardboard into the shapes of the Scratch blocks:
 - 3 blocks: Move _____ steps (blue)
 - 2 blocks: Point towards _____ (blue)
 - 3 blocks: Repeat _____ (orange)
 - 3 blocks: Wait _____ seconds (orange)
 - 1 blocks: When thouch board (orange)
 - 1 blocks: Forever <____> (orange)
 - 10 white rectangles to be fitted within the previous blocks (____): 8 blank, 1 "left" e 1 "right"
 - 1 blank diamond to be inserted into the block "If" (< ____ >)



Leading the activity:

Start by gluing all the blocks to a corner of the board with double-sided tape. Divide the class into groups of 4 students. Then ask a student to apply to be the actor on the program. This student will start in one corner of the room and must reach another corner (the door, for example) following the instructions that the class creates. Suggest a route that involves turning 90 ° at least once.

Each group has a turn to suggest a new block to the program or to modify a block that is already part of the program. At each turn, the group must discuss which block to use or which modification to make to the program, go to the blackboard and add a new block to the program or modify it. Then, the student-actor executes the program on the blackboard and, at the end of the shift, returns to the starting point.

The activity is repeated with each group contributing to the final program until the objective is reached After completing the task, new ways of doing it can be suggested, and new concepts included, such as conditions (example: if someone claps, something happens.)

It is important to allow students to make mistakes during the process so that they realize the importance of providing accurate instructions for the program to work as we expect.

When an instruction provided does not make sense, make sure that the student-actor will perform that action as it is on the board, to show what will happen.

Discussion and reflection:

After completing the activity, discuss with your students the concepts covered in this class. See examples of some questions that can be used to start the discussion.

- What do you think is programming? When do we use that word?
 - After listening to students, relate the idea of programming to student comments. It is common to hear relationships with programming a clock, programming yourself to do something, or programming on television. Connect the examples presented to the idea of providing instructions to perform an action and to orderly sequences of events..
- Where is there programming in our day to day?
- What does a programmer do?

Credits:

Cassia Fernandez (LSITec/USP)