Universidad de Guantánamo

Ciencia y Progreso
Publicada en línea: abril, 2021, RNPS: 2477, ISSN: 2707-7098
http://cienciayprogreso.cug.co.cu

Conocer la interfaz gráfica de Scratch, una necesidad para su uso

Knowing the Scratch graphical interface, a necessity for its use

Esteban Fernández-Sánchez

Universidad de Guantánamo
Correo(s) electrónico(s)
estebanfs@cug.co.cu
ORCID: https: //orcid.org/: 0000-0001-9493-2527

Recibido: 20 de septiembre de 2020
Aceptado: 26 de noviembre de 2020

Resumen

El artículo tiene como finalidad valorar la importancia y potencialidades de Scratch como lenguaje de programación visual en el contexto del Tercer Perfeccionamiento del Sistema Nacional de Educación en Cuba; ofrece un procedimiento didáctico sustentado en el método deductivo y la analogía, permite transitar por el contenido objeto de estudio desde los elementos más generales que caracterizan este lenguaje de programación hasta llegar a la identificación de los aspectos más comunes presentes en la interfaz gráfica del editor a utilizar, facilitando la adquisición del sistema de conceptos y procedimientos que contribuirán a la resolución de problemas desde las entidades educativas.

Palabras clave: Scratch; Lenguaje de programación visual; Interfaz de trabajo; bloques de programación

Abstract

The purpose of the article is to assess the importance and potential of Scratch as a visual programming language in the context of the Third Improvement of the National Education System in Cuba; it offers a didactic procedure based on the deductive method and analogy, allowing to go through the content under study from the most general elements that characterize this programming language to the identification of the most common aspects present in the graphic interface of the editor to be used, facilitating the acquisition of the system of concepts and procedures that will contribute to the resolution of problems from the educational entities.

Keywords: Scratch; Visual programming language; Working interface; Programming blocks;

Introduction

In Cuba, as of 2017, the Third Improvement of the National Education System is implemented, which arises from the need to update textbooks, curricula, subject programs, methodological orientations, based on the development that science, technique and technology are reaching and the demands that humanity is posing in order to put the Cuban school at the height of these times. (Jiménez and Verdecia, 2019).

To achieve the aforementioned purpose, the Ministry of Education (MINED) and the Ministry of Higher Education (MES) have established working agreements aimed at the development of training and initial training actions respectively, both for teachers of the computer science branch who are in practice and those who are in training so that they can face with quality the teaching-learning process of programming in Scratch.

Such training and initial training actions have been developed since the school year 2017 - 2018; in the case of teachers in training who belong to the Bachelor's degree in Computer Education, it is introduced within the Study Plan E in this same school year as a subject of its own.

However, from the exchange of ideas with methodologists, teachers and students of the computer science branch, it has been possible to know certain limitations to introduce Scratch as a visual programming language in the different educational levels, for example: there is insufficient bibliography that from the didactic point of view, constitute bases of orientation for the study of introductory contents derived from the graphic interface of Scratch in its different versions, and thus be able to identify for its efficient use the existing relations between its fundamental elements and make them available for the resolution of diverse problems.

For this reason, the purpose of this article is, firstly, to assess the importance and potential of this new programming language and, secondly, to offer a didactic procedure that allows, as an introductory aspect, to characterize the Scratch graphical interface on the basis of the establishment of the existing relationships between its fundamental elements.

Development

In the study plan of the Bachelor's Degree in Computer Science Education, it is stated that, although it is not an objective of the Programming Languages and Techniques discipline to train programmers, it is

necessary that Computer Science teachers acquire a culture of programming and prepare themselves to teach the fundamentals of programming in the future from the last two grades of primary education and in basic secondary education. (Curriculum E, 2016, p. 29)

In order to comply with the objectives of the Third Improvement of the National Education System in Cuba and as part of the updating of subject programs, in the case of computer science, the study of the Scratch programming language is carried out in primary and basic secondary education, being necessary to introduce it also from the initial training of the graduate in computer education so that he/she can lead the teaching - learning process of this new programming language once graduated from the different educational levels, however:

What is the Scratch programming language?

How did it come about?

What is Scratch's potential as a programming language?

How can we access the Scratch editors?

How to install them?

How to identify the elements shown in the Scratch editors' work interface?

In order to answer these questions and the objective of this article, it is proposed to use a didactic procedure that consists in a first moment, to initially characterize Scratch as a programming language and in a second moment, to identify the elements of the graphical interface of one of its editors, all this from the study of the fundamental concepts and those procedures that are necessary to elaborate.

During the process of formation of the concepts derived from the study of Scratch as a programming language, a description of the fundamental aspects that identify it will be made using for it in certain moments as logical ways of obtaining knowledge, the deductive and the analogy, therefore, these concepts will be made known to then identify their essential characteristics, this same procedure will be used in the formation of some procedures.

In the same vein, Scratch is a free visual programming language, oriented to teaching and learning, which allows the creation of interactive stories, animations, games, music, art, didactic materials, etc., as well as sharing the creation of different projects on the web and with other developers.

This programming language arises from the development of a project that was born between the company MIT Laboratories and the University of California, Los Angeles in 2003; the purpose was to teach programming to children and adolescents between the ages of 8 and 16 years.

> The term Scratch comes from the name given to a hip hop technique called scratchering by which disc jockeys experiment with music by playing vinyl records back and forth with their hands, mixing music clips in a creative way. Scratch programming attempts to be somewhat similar, mixing graphics, animations, photos, music, and sound in a highly interactive way. (Cynthia, 2015, p. 20)

There are certain potentialities that identify Scratch as a programming language, being highly demanded by children, adolescents and young people who dabble in this activity, because:

- It is a programming language formed by a set of objects that from their shapes and meanings define the structure of the programs to be developed, generally integrating multimedia elements.
- Its editors show a simple and attractive graphical interface in more than 50 languages, where through it you can program and perform various tasks without having to write a single letter of code, simply by performing operations with the mouse on different objects organized by blocks and colors which are also called Sprites.

Sprites is an English word that translated into Spanish means mascot, which are used in the creation of cartoons in order to guide the end user in the interaction with each of the elements present in the work interface of the program or project developed.

- It favors group work, due to the possibility of sharing objects, programs and projects (scratching technique).
- It facilitates the use of multimedia elements by easily incorporating audio, text and images.
- Scratch is multiplatform (Windows, Mac, Linux, etc.).

From the aforementioned assumptions, it is necessary to know how to access Scratch. From this point of view, it is possible to access Scratch precisely through its editors and free of charge in several Internet sites, however, reference is made in this article to the main Scratch site with the following Web address: https://scratch.mit.edu/download.

From this site, it is possible to download the latest version of Scratch, both portable and desktop, in addition to facilitating the work directly through an online editor called Scratch Offline Editor.

This material will work with Scratch version 3.12, which will serve as a basis for identifying the elements of its graphical interface, which is very important to be able to use it to develop the projects or programs desired, hence the need to know in depth its fundamental characteristics.

The value of the previous idea, makes some suggestions for the study of the graphical interface of Scratch editors, for this it is necessary to identify and relate its components with its functionalities, which will contribute in the first place to the description of its fundamental characteristics and in second place to establish the links or relations that exist between each one of them.

It is also possible to make comparisons with other graphical interface of other visual programming language editors such as Visual Basic or Live Code and thus determine differences and similarities, which will contribute to the analysis of the content under study.

Under these circumstances, authors such as Albornoz, Vialart, Berón and Montejano, (2017), state that the Graphical User Interface (GUI) is the fundamental part of any application, where the process of human-computer interaction takes place, whether to interact with an operating system, software or website; these authors also consider that the design and study of a graphical interface should not be considered a secondary and unimportant task, since it is responsible for providing a smooth and pleasant interaction.

Next, from the image shown, the most important elements of the Scratch 3.12 graphical interface, the latest improved version to be published at the beginning of 2020, are identified, for which it is also suggested:

- Observe and identify each of its elements, and investigate its meaning.
- Make a summary table or list of these elements, adding also their functionalities and relationships in case they exist.

Imagen 1 Interfaz gráfica Scratch 3.12: Fuente: Elaboración propia

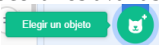
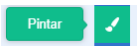
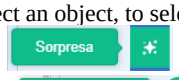
ELEMENTS	DESCRIPTION
TITLE BAR	Displays the name of the Scratch editor you are working with, as well as the minimize, maximize and close buttons.
MENU BAR	It contains the following menus: Language , to facilitate the selection of a language among 50 available; the File menu, allows you to create a new project, upload or save a project to the computer; Edit menu, facilitates testaurar the default values of an object, as well as accelerate the execution of projects through the Turbo Mode option; Tutorials menu, through which you can access video tutorials that facilitate learning and finally the Scratch project option, which allows you to name the project or program.
SCENARIO	Space where the Scratch mascot is located, it shows the objects in their interaction and results of the current project, it also has a toolbar.  which makes it easy to increase or decrease the size and position of the stage, and on the right-hand side there are two buttons  the green flag to run and the pink exagon to stop the program.
OBJECT POSITIONING AREA	This area shows the position of the object that has been selected, as well as its size and direction. It also presents two buttons that allow you to show or hide the object that has been selected.
LIST OF OBJECTS AND SCENARIOS INCLUDED IN A PROGRAM	Displays thumbnail images of all Objects and Scenarios available in the project. Here you can also find the Choose an object button.  through which you can access other buttons such as  Select an object, to select a new object,  Paint, to paint a new object at will,  Surprise, from which an object is added at the publisher's will and  Upload an object, option to upload an object saved on the computer.
PROGRAM EDITING AREA, DISGUISES BACKGROUNDS AND SOUNDS OF THE ACTIVE OBJECT	Area where the programming blocks are dragged to form stacks to create the program code..
PROGRAMMING BLOCK CATEGORIES PROGRAMMING	Allow access to the family of a given programming block group..
PROGRAMMING BLOCK AREA PROGRAMMING	They are sets of objects characterized as instructions, which are part of a given category and are used to form the code of a program joined together in the form of a stack.
ADD EXTENSIONS	Area through which extensions can be added, what are the extensions, other categories to be inserted by the user at will, which show a family of specific programming blocks.

Table 1 Elements of the Scratch 3.12 graphical interface: Source: Own elaboration

All the elements shown are important for the development of the programs or projects, however, it is necessary to study some of them in depth, such as the programming blocks.

> The blocks represent the programming structures, and the actions that can be performed within the program (moving an object, playing a sound, etc.). Each block has a different shape, and there are certain pieces that can be joined together, and others that cannot. Fitting the pieces together where joining is possible builds syntactically correct programming structures. (Alonso, 2017, p. 52)

But what are the elements that distinguish them from other objects in the Scratch work interface?

1. They are objects also called instructions.

2. They are organized within the code of a linked program in the form of a stack.

3. They are derived from a category according to their color and functionality.

4.They make the other objects that are inserted in the scenario work.

From the above, it is derived that the programming blocks are accessed from their Categories, these being classification criteria based on color codes: Motion, Appearance, Sound, Events, Control, Sensors, Operators, Variables, and My Blocks (See Table 2). To access the programming blocks corresponding to a category, click on the desired category and its corresponding blocks will be automatically displayed.

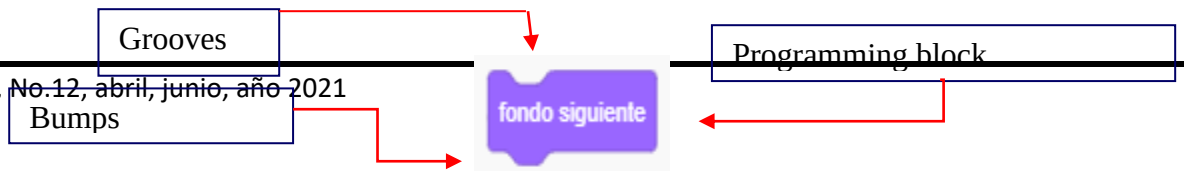
CATEGORY	Allow:	CATEGORY	Allow:
Movement	Displacement, orientation and position of objects on the stage.	Sensors	Objects can interact with the environment created by the user.
Appearance	Modify the appearance of objects.	Operators	Insert different types of operators.
Sound	Insert, change, stop different types of sounds.	Variables	Carrying out work with variables and data.
Events	Establish certain actions to be carried out at the beginning of the projects.	My Blocks	Display blocks created by user and selected through the Add Extension option.
Control	Insert conditional statements "if-else", "forever", "repeat" y "stop".	Add Extension	

Table 2 List of programming blocks: Source: Own elaboration

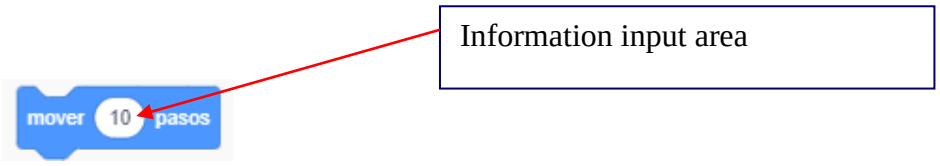
Note that in Scratch there are three types of programming blocks that are classified according to their form and functionality, namely:

1. **Stacking Blocks**

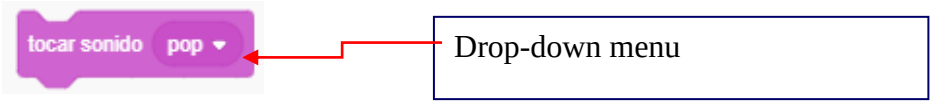
   The stacking blocks have protruding protrusions and/or notches at the top allowing them to fit together to form stacks, for example:



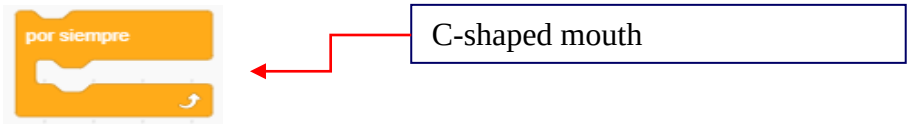
Some of these blocks have an information entry area inside, where you can type in a number:



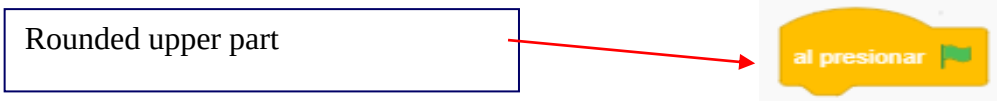
Others allow vou to select an item from a drop-down menu, such as meow in the block:



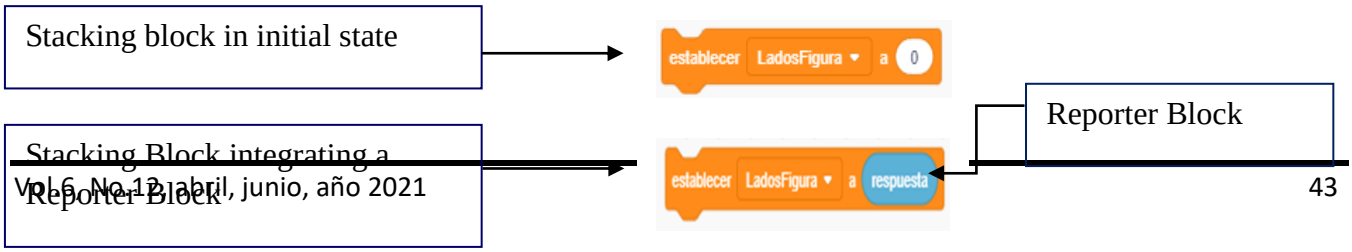
There are also blocks with a C-shaped mouth, into which you can insert other stacking blocks:



2. Hats blocks: they have a rounded top, the first to be placed in the stacks. They wait for an event to happen, for example, a key is pressed, and then they execute the blocks below them:



3 Reporter Blocks: they are designed to fit into the information entry area of other blocks based on their shape.



Once the elements mentioned above have been analyzed, it is necessary to show a small example of how to select and insert programming blocks into the code or programming area, since this is the basis for the creation of programs or projects using the Scratch graphic interface:

First, select with a click the category that contains the programming blocks to be used.

2. Second, select and drag the desired programming block to the code area.

These steps must be executed for each of the programming blocks to be used during the development of a program or project, and in this way a stack of programming blocks will be formed, which will constitute the main code as shown below where it is shown as an example to determine the number of patients who are tested to see if they are carriers of the Covid-19 virus and if they are positive to report whether they were asymptomatic or not.

Image 2 Stack of programming blocks: Source: Own elaboration

There are several types of programming blocks in this stack, which are linked according to their shapes, color and functionality; knowing where they are located and what relationships are evident in each of them from the Scratch graphical interface is of vital importance, as it contributes to achieving higher levels of efficiency and effectiveness during the teaching-learning process of this new programming language, thus facilitating the development of problem-solving skills, even in children from an early age.

Conclusions

The present didactic procedure was elaborated taking into account the need to motivate children, adolescents, young people and society in general in the study of programming taking as a premise the

updating of the subject programs in the computer science branch according to the requirements of the Third Improvement of the National System of Education in Cuba.

Likewise, it contributes to the understanding and identification of the fundamental elements that characterize the Scratch programming language and the graphic interface of its editors, facilitating the resolution of problems and the search for new knowledge from different educational contexts.

Bibliographic references

Jiménez, R. y Verdecia, E. (2019). La educación cubana desde un prisma renovador. Recuperado de http://scielo.sld.cu/pdf/reds/v8n1/2308-0132-reds-8-01-7.pdf.

MES. (2016). Plan de Estudio E. Carrera Licenciatura en Educación Informática. La Habana, (Material digital).

Cynthia, E. (2015). Diseño de actividades educativas en Scratch para la dinamización del Museo de Informática. Trabajo Fin de Grado en Ingeniería Informática. Universidad Politécnica de Valencia.

Albornoz, M.; Berón, M. y Montejano, G. A. (2017). Interfaz gráfica de usuario: el usuario como protagonista del diseño. Recuperado de http://sedici.unlp.edu.ar/handle/10915/62078.

Alonso Urbano, D. (2017). *Scratch como herramienta para la enseñanza de la programación en la Educación Primaria*. Tesis de Doctorado. Universidad "Camilo José Cela", Madrid.