

Desarrollando Computational Thinking con Scratch.

MARÍA NIEVES CARRALERO COLMENAR. IES Pedro Mercedes.
Cuenca

Introducción

Herramientas nuevas y más sofisticadas están cambiando la forma en que la próxima generación aprenda a programar ordenadores. Los niños ahora pueden crear escenas complejas y juegos sin los comandos crípticos que alguna vez fueron la única forma de saber qué hacer computadoras. Los niños con más capacidad también puede utilizar algunas de las herramientas sofisticadas normalmente utilizados por los programadores profesionales, debido a que las herramientas de hoy son a menudo lo suficientemente fácil para alguien para recoger con sólo unos meses de estudio.

Mitchel Resnick, profesor de investigación de aprendizaje en el Media Lab del MIT, que ayuda a dirigir el proyecto de Scratch, dijo que Scratch es efectivo con los niños, ya que fomenta la colaboración.

La asignatura de Informática es una de las mejores asignaturas para ser impartida en un sistema bilingüe, debido a que puede verse como una asignatura que integra conocimientos de otras (matemáticas, lengua, física...) y que tienen una proyección más allá de ninguna frontera.

En este artículo se muestra un ejemplo de código usando Scratch.

Computational Thinking

El pensamiento computacional o “Computational Thinking” en inglés, ha sido definido por el ISTE (International Society for Technology in Education) y por la CSTA (Computer-supported telecommunications applications) como un proceso de solución de problemas que incluye, entre otros: analizar problemas, organizar y representar datos de manera lógica, automatizar soluciones mediante pensamiento algorítmico, usar abstracciones y modelos, comunicar procesos y resultados, reconocer patrones, y, generalizar y transferir.

Desarrollar el pensamiento computacional es la manera más adecuada de aprender una nueva forma de pensar y de resolver problemas; una manera de pensamiento que es fundamental y crítica en el siglo XXI, una etapa llena de diferentes tipos de tecnología y plagada de computadores en todos los espacios de nuestras vidas. A la importancia del desarrollo del pensamiento computacional, se han unido algunas voces que incluso reclaman que la computación sea una disciplina añadida en las escuelas como la lectura, la escritura o las matemáticas que todos los estudiantes deberían aprender.

Ejemplo

En el ejemplo mostrado en este artículo se trabajara con lenguajes de programación. En particular con Scratch. El grado en el que se puede aplicar es 4º ESO.



GALAXIAN

En el ejemplo mostrado se utilizan los elementos típicos del juego de los 80's Galaxian.

Rocket: El movimiento del rocket es muy sencillo. Se mueve con las teclas a derecha e izquierda.

Bullet: La bala se mueve desde el rocket hasta el bode superior. Si impacta con un sprite hace un sonido y desaparece. Solo se puede disparar una bala cada vez.

Enemies: Se mueven en horizontal bajando cada cierto tiempo. La velocidad es configurable usando variables.



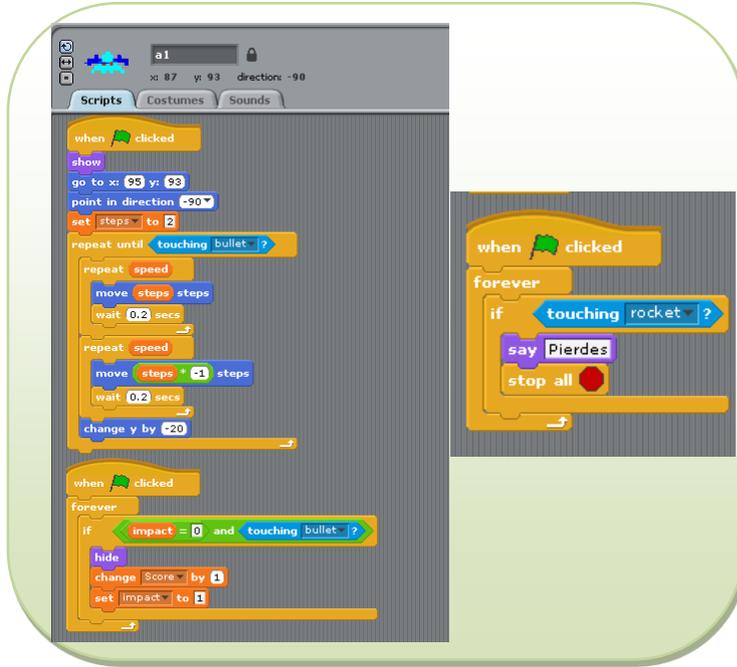
A screenshot of the Scratch script editor for the 'rocket' sprite. It shows three event-driven scripts: 1. 'when clicked' leading to 'go to x: 3 y: -126'. 2. 'when right arrow key pressed' leading to 'move 3 steps'. 3. 'when left arrow key pressed' leading to 'move 33 steps'.



A screenshot of the Scratch script editor for the 'bullet' sprite. It shows a single event-driven script: 'when space key pressed' leading to a sequence of actions: 'go to rocket', 'play sound Laser1', 'show', 'point in direction 90', 'set impact to 0', and a 'repeat until touching edge?' loop containing 'change y by 10'. After the loop, there is an 'if impact > 0' block containing 'play sound Zoop' and 'hide'.

A screenshot of the Scratch stage window titled 'mis-naves'. It shows a space-themed game with a rocket at the bottom and several rows of alien ships (a1-a5, b1-b7, c1-c5, d1-d3) at the top. The 'speed' is set to 5 and the 'Score' is 132. Below the stage is a 'New sprite' palette with various sprite options.

A screenshot of the Scratch 'Sounds' editor window. It shows two sound assets: 'Laser1' (1 KB) and 'Zoop' (5 KB). Each asset has a speaker icon, a play button, a stop button, and a delete button.



Conclusiones

En este artículo se ha mostrado un ejemplo de juego hecho con Scratch para desarrollar el Computational Thinking. Lo importante del código no es tanto el juego en sí, sino las estrategias utilizadas para dar solución al juego. Variables, condicionales, bucles y secuencias, son la base para la programación, y este ejemplo puede servir de modelo para hacer hincapié sobre ellos.



www.sociedadelainformacion.com

Edita:



Director: José Ángel Ruiz Felipe
Jefe de publicaciones: Antero Soria Luján
D.L.: AB 293-2001
ISSN: 1578-326x