

Programación en Bachillerato. Crear video-juegos como recurso para comprender bucles y condicionales.

Nieves Carralero Colmenar.

IES Ramón y Cajal (Albacete). Junta de Comunidades de Castilla-La Mancha. España.

Resumen

La herramienta gráfica libre de programación Scratch ha sido desarrollada por el Lifelong Kindergarten Group de los laboratorios Media-Lab en MIT. Es un lenguaje gráfico de programación que hace fácil crear historias interactivas, animaciones, juegos, música y arte, para ponerlos en la web. Su filosofía es la de programar conectando bloques predefinidos. En este artículo se muestra cómo enseñar las estructuras condicionales y bucles en la creación de un juego con Scratch.

1. INTRODUCCIÓN

En Scratch [SCRATCH] se da la posibilidad de que, por medio de una mezcla de sonidos, imágenes y animaciones, se puedan lograr productos interactivos y creativos. Scratch abre un nuevo mundo de posibilidades para la programación de bachillerato, ya que ellos podrían subir su contenido creado a Internet, compartirlo, y así incentivar a otros a lograr algo similar o mejorar lo realizado.



Scratch funciona para Windows, Linux y Mac. En el sitio oficial de la aplicación (<http://scratch.mit.edu/>) se pueden subir trabajos y presentarlos en comunidad para ser compartidos por otros usuarios. Además, en los últimos dos años, desde el propio MIT y otras organizaciones se han convocado concurso para los mejores desarrollos usando Scratch.

La principal aportación de Scratch es que está destinado a usuarios de temprana edad, lo que lo hace directamente aplicable como herramienta didáctica destinada a enseñar programación en particular e informática en general. Cada vez más, los alumnos están interesados en la programación como medio creador de diferentes utilidades, aplicaciones y juegos, apareciendo este interés cada vez más en edades tempranas. Scratch ofrece una perspectiva avanzada en conocimiento pero sencilla en manejo.

Para aquellos que llevan muchos años en la docencia de la Informática en las aulas y que no han entendido hasta el momento lo que ofrece Scratch, se puede decir que Scratch es un sustituto avanzado de herramientas “ancestrales” como Logo. [CARRALERO, NIEVES, 2011]

2. ¿POR QUÉ CREAR UN VIDEO-JUEGO?

La creación de video-juegos como recurso para enseñar las estructuras básicas de un lenguaje de programación es una de las alternativas más atractivas que un docente puede ofrecer a su alumnado de Bachillerato en el área de Tecnologías de la Información y la Comunicación. Dos son los motivos principales que se destacan como ventaja:

- Con los video-juegos se acercan los contenidos del currículo a un mundo con el que el alumnado está muy familiarizado fuera del aula. Desde esta perspectiva, trabajar con video-juegos permite enlazar los que el alumnado aprende fuera del aula con lo que aprende dentro, favoreciendo así el trabajo por competencias.
- Se aleja al alumnado del tándem *informática-matemáticas*. La mayor parte de las primeras enseñanzas relacionadas con la programación están en la línea de utilizar un lenguaje de programación (C, Java, Basic, etc.) para resolver algoritmos matemáticos: sacar número primos, descomponer un número en factores, etc. Esta perspectiva, aunque tiene una indiscutible utilidad desde el punto de vista de la competencia matemática, suele crear un rechazo en el alumnado, al no ser capaz de asociar y visionar la programación como base de la tecnología del mundo real.

La manera más simple para definir a un video-juego es: una interacción entre algún tipo de máquina y una persona, la persona realiza alguna acción y la máquina, o los gráficos en una máquina, reaccionan a esa acción de alguna forma. En un vídeo-juego siempre hay tres elementos principales:

1. Entrada (controles, joysticks, pads, ratón etc.).
2. Procesamiento (Cálculos de posiciones y colisiones, generar sonidos, cambiar estados, etc..).
3. Salida (desplegar gráficos, tocar sonidos, mover dispositivos externos)

Por lo tanto, en este artículo se entiende video-juego como una interacción en la que un objeto reacciona ante una entrada devolviendo una salida.

3. ¿QUÉ VIDEO-JUEGO CREAR?

Desde la línea argumental de la sección anterior, en este artículo se propone la creación de un video-juego para aplicar principalmente las estructuras condicionales (if-then-else) y los bucles (while, repeat)

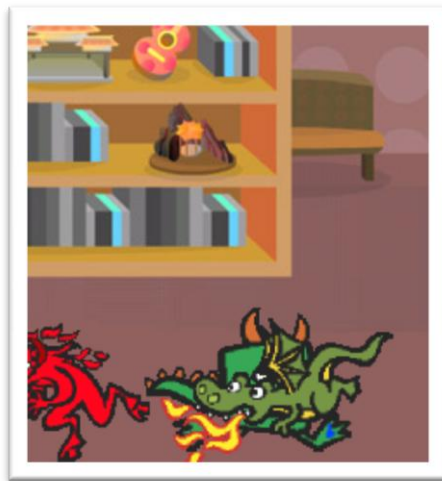
Para ello, se propone un juego tipo *arcade*, en el que la complejidad de los gráficos es muy baja, y los movimientos no se corresponden fielmente con las leyes físicas de la naturaleza (características típicas de este tipo de juegos).

Se propone un personaje que salta hacia arriba (con un ligero desplazamiento hacia adelante) cuando el usuario presiona la tecla "espacio". Además, el personaje puede moverse hacia adelante o atrás con las flechas del teclado. El personaje está puesto en un escenario inmóvil y queda a la espera de que aparezcan personajes-automático que debe saltar para no colisionar con ellos. Si el personaje colisiona con otros, éste cambia su aspecto y emite un sonido para dar la sensación de que lo han aplastado, y después de 1 segundo, continúa su camino. El ciclo de apariciones es infinito, por lo que cuando un personaje-automático llega al final de la pantalla, éste espera 3 segundos para volver a aparecer por el lado contrario.

La siguiente figura muestra una captura del juego en ejecución: El muñeco verde con cuernos es el personaje que en ese momento está saltando. Los personajes-automáticos que se mueven de izquierda a derecha son el demonio rojo, el dragón verde y el gato naranja.



Cuando el personaje colisiona con un personaje-automático entonces éste último cambia de disfraz y queda estático durante un segundo (además emite un sonido). Pasado el segundo continúa hasta el final de la pantalla. La siguiente captura muestra al dragón alcanzado por el personaje.



4. PASOS A SEGUIR

Antes de proponer el juego directamente al alumnado, el docente debe haber explicado las sentencias condicionales y los bucles y haber hecho ejemplos sencillos con números o letras para que el alumnado interiorice el proceso.

Posteriormente, el docente debe explicar cómo Scratch maneja las estructuras condicionales y los bucles. Además, también sería interesante explicar otros aspectos relacionados con este video-juego en particular como, por ejemplo: cómo hacer colisiones, cómo mover al personaje y hacer que salte.

El objetivo principal que se busca es que el alumnado aplique por sí mismo una solución para hacer que los personajes-automáticos se muevan sólo.

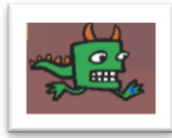
El resto de aspecto pueden ser facilitados y resueltos por el docente: el docente puede explicar cómo se hacen las colisiones y cómo hacer para que el personaje salte al presionar la barra de desplazamiento y se mueva adelante y atrás con las flechas de dirección.

5. SOLUCIÓN

La siguiente imagen muestra el juego en un determinado momento en el que el protagonista salta para no colisionar con los personajes-automáticos.



A continuación se muestra la solución del video-juego para cada personaje:



Protagonista:

El protagonista tiene asignado el siguiente código:



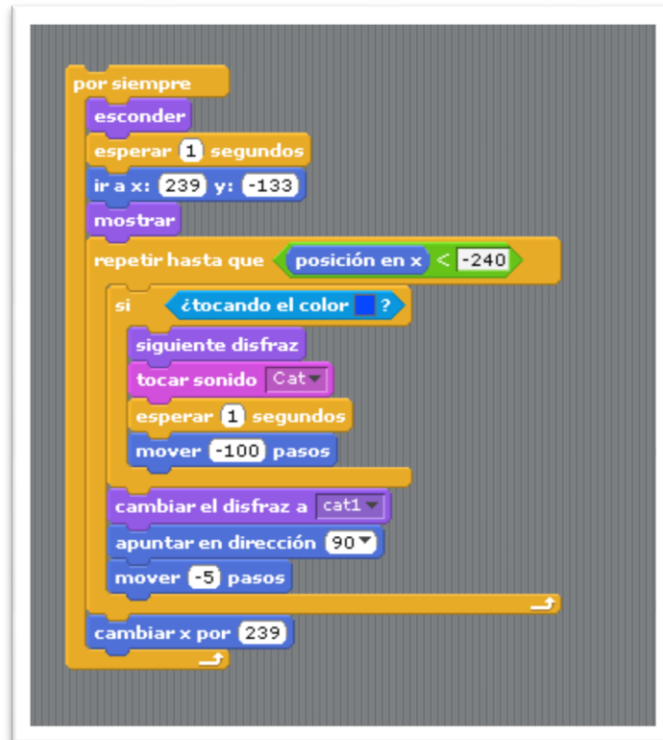
Se atienden tres eventos:

1. Al presionar la barra espaciadora el personaje salta. Se usa un bucle que se repite 11 veces para subir en cada vez 30 pasos en el eje de las y. luego espera 0.5 segundos antes de repetir otro bucle que hace lo mismo pero restando 30 pasos (-30) al eje de las y, para dejarlo en la misma posición inicial.
2. Al presionar la tecla "flecha derecha" se mueve el personaje 10 pasos y cambia de disfraz. El siguiente disfraz disponible es uno que cambia la posición de las patas del personaje para dar sensación de movimiento.
3. Al presionar la tecla "flecha izquierda" se mueve el personaje -10 pasos y cambia de disfraz. El siguiente disfraz disponible es uno que cambia la posición de las patas del personaje para dar sensación de movimiento.



Protagonista-automático

Este protagonista tiene asignado el siguiente código:



EL código tiene las siguientes partes:

1. Un bucle exterior que se hace “por siempre” para que nunca termine el movimiento.
2. De inicio se esconde, espera un segundo y luego se coloca en la posición más a la derecha de la pantalla.
3. Un bucle que se repite hasta que se llega a la final de la pantalla. (x=-240)
4. En ese bucle se detecta la colisión si este personaje-automático choca con un objeto con color **azul**. Para que esta colisión tenga efecto, al personaje se le ha puesto en un pie una mancha de ese **azul** que haga de “seña de colisión”.



5. Si colisiona, entonces cambia de disfraz para simular que lo aplasta, se emite un sonido y se adelanta al personaje-automático para que salga de la colisión. La siguiente imagen muestra al personaje-automático colisionado.

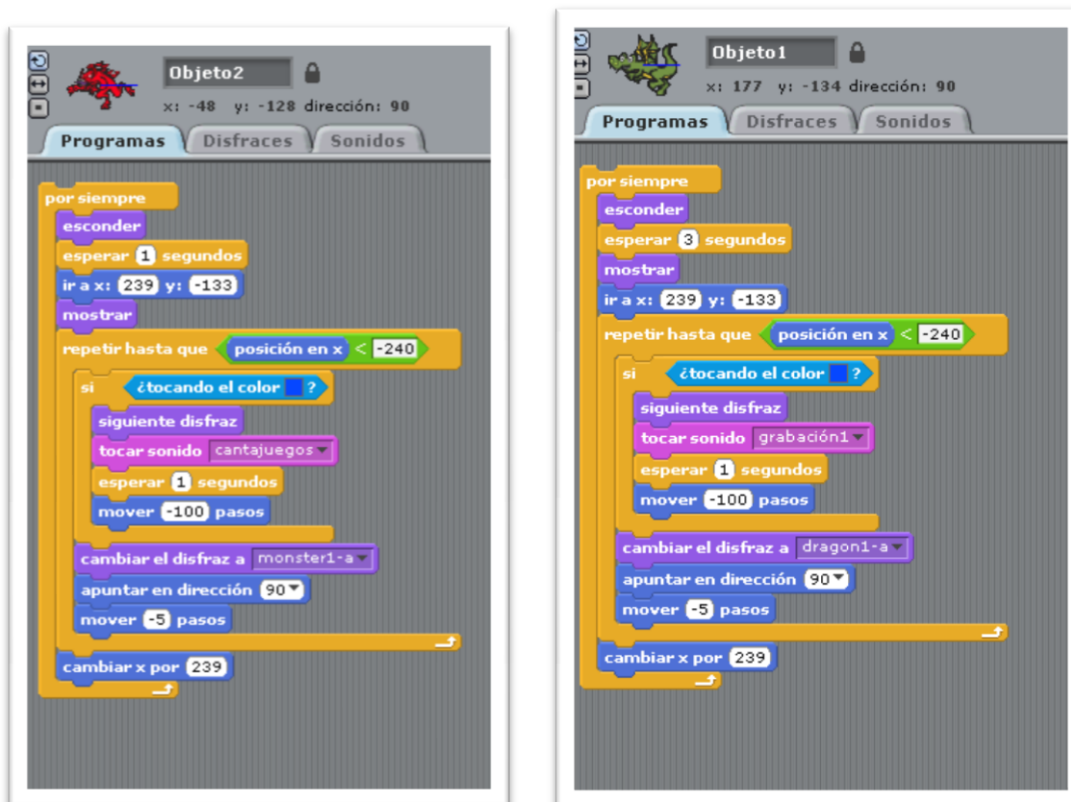


6. Si no colisiona, entonces avanza -5 pasos y continúa el bucle.

El mismo bucle es para el resto de personajes-automáticos. El hecho de que todos no se inicien al mismo tiempo, que si hay o no colisión condicione cuando llegan al final y que al principio espere 1 segundo antes de aparecer afecta a la frecuencia de aparición de todos ellos, por lo que a veces saldrán más espaciados y otras más amontonados, dificultando el juego.



A continuación se muestra el código de los otros dos personajes-automáticos:



4. CONCLUSIONES

Scratch es una aplicación sencilla para enseñar a programar. Su interfaz gráfica, alejada de los “poco amigables” lenguajes de programación textuales, tiene todo lo necesario para hacer Programación Orientada a Objetos.

Desde el punto de vista de las TIC, que el alumnado aprenda a programar sus aplicaciones es una necesidad, ya que la sociedad de la información lo requiere. Sin embargo, no es aconsejable que aprendan a programar como objetivo (con tareas como hacer algoritmos para sacar si un número es primo o no) sino que deberían aprender como medio para conseguir un fin.

El fin con Scratch es hacer proyectos multimedia atractivos que, al mismo tiempo que enseñan a los alumnos a programar, desarrollan contenidos, presentaciones, juegos, etc., para sus aulas y sus materias.

En este artículo se muestra el código de un video-juego sencillo que permite desarrollar los contenidos respecto a la programación al mismo que los motiva al hacer aplicaciones cercanas a sus inquietudes, que fácilmente pueden posteriormente ser reconvertidas en software aplicado.

5. BIBLIOGRAFÍA

[SCRATCH] <http://scratch.mit.edu/>

[CARRALERO, NIEVES, 2011] ENTORNOS PARA ENSEÑAR PROGRAMACIÓN EN SECUNDARIA. NUEVOS ENFOQUES. Revista digital “Quaderns Digitals / Quaderns” número 70. Diciembre de 2011. http://www.quadernsdigitals.net/index.php?accionMenu=hemeroteca.VisualizaArticuloIU.visualiza&articulo_id=11125

[CARRALERO, NIEVES, 2011b] SCRATCH. PROGRAMACIÓN FÁCIL PARA EDUCACIÓN PRIMARIA Y SECUNDARIA. Revista Digital Sociedad de la Información. Nº 29. Junio 2011. <http://www.sociedadelainformacion.com>

SOCIEDAD DE LA INFORMACION

www.sociedadelainformacion.com

Edita:



Director: José Ángel Ruiz Felipe

Jefe de publicaciones: Antero Soria Luján

D.L.: AB 293-2001

ISSN: 1578-326x