

J. Ángel Velázquez Iturbide

Una Indagación del
Comportamiento del Lenguaje
ScratchJr

Número 2021-03

Serie de Informes Técnicos DLSI1-URJC

ISSN 1988-8074

**Departamento de Lenguajes y Sistemas Informáticos I
Universidad Rey Juan Carlos**

Índice

1	Introducción	1
2	Bloques del lenguaje ScratchJr.....	2
3	Metodología	6
3.1	Bloques de movimiento.....	7
3.2	Bloques de apariencia.....	7
3.3	Bloques de control.....	8
3.4	Bloques disparadores.....	8
3.5	Bloques de finalización	9
4	Resultados	9
4.1	Cuestiones generales	9
4.2	Bloques de movimiento.....	10
4.3	Bloques de apariencia.....	10
4.2	Bloques de control.....	11
4.2	Bloques disparadores.....	11
4.2	Bloques de finalización	12
5	Conclusiones	12
	Agradecimientos	12
	Referencias.....	13
	Apéndice: Programas de Prueba.....	14
	Bloque “comenzar al tocar” – Programa 1	14
	Bloque “fijar velocidad” – Programa 2	16
	Bloque “fijar velocidad” – Programa 3	18
	Bloque “parar” – Programa 4	19
	Bloques de mensajes – Programa 5	20
	Bloques de mensajes – Programa 6	22
	Personajes y guiones – Programa 7	25
	Páginas – Programa 8.....	26

Una Indagación sobre el Comportamiento del Lenguaje Basado en Bloques ScratchJr

J. Ángel Velázquez Iturbide

Departamento de Informática y Estadística, Universidad Rey Juan Carlos,
C/ Tulipán s/n, 28933, Móstoles, Madrid
angel.velazquez@urjc.es

Resumen. Hemos realizado un análisis del comportamiento de los programas desarrollados en el lenguaje de programación basado en bloques ScratchJr. Se presenta la metodología de indagación utilizada y se presentan las características encontradas del lenguaje. El informe incluye, como apéndice, los programas de prueba utilizados para la indagación.

Palabras clave: Lenguajes de programación, semántica dinámica, máquina nocional, ScratchJr.

1 Introducción

Los lenguajes de bloques gozan de gran difusión y popularidad para la enseñanza de la programación a niños y jóvenes, ya que son sencillos de aprender y permiten una alta expresividad. Esta situación contrasta con la del aprendizaje de los lenguajes textuales, que son el principal reto del aprendizaje de la informática en los grados universitarios de informática. No obstante, existen diferencias entre ambos tipos de lenguaje que explica esta situación aparentemente paradójica [1].

Los lenguajes basados en bloques, como ScratchJr o Scratch, no están exentos de dificultades, como ya comprobamos en la propia enseñanza universitaria [2]. Entre otros aspectos, es necesario proporcionar a los alumnos una definición precisa del comportamiento de los programas (lo que se llama conocer su máquina nocional).






En este informe presentamos una indagación planificada del lenguaje ScratchJr. Existen descripciones del lenguaje pero son informales e incompletas [3][4][5]. El objetivo del informe no es presentar una descripción completa del comportamiento del lenguaje ScratchJr sino indagar en los aspectos menos evidentes, para así poder realizar dicha descripción completa.




La estructura del informe es la siguiente. El apartado 2 presenta brevemente los bloques de lenguaje ScratchJr, agrupados en categorías. El apartado 3 contiene la metodología utilizada para la indagación, en particular la indagación planificada, y el apartado 4, las conclusiones obtenidas. Un apéndice contiene los programas e pruebas desarrollados.




2 Bloques del lenguaje ScratchJr




No realizamos una descripción del lenguaje ScratchJr aquí, ya que puede encontrarse en cualquiera de las referencias anteriormente citadas. Sin embargo, resumimos las categorías de bloques para contextualizar mejor la descripción de los casos de prueba.



Los 25 tipos de bloques del lenguaje se agrupan en seis categorías [6]:


Bloques de movimiento	
	Mueve el personaje a la derecha cierto número de casillas
	Mueve el personaje hacia arriba cierto número de casillas
	Gira el personaje a la derecha cierta cantidad, produciendo 1 un giro de 30°
	Mueve el personaje a la izquierda cierto número de casillas
	Mueve el personaje hacia abajo cierto número de casillas





<p>Girar a la izquierda</p> 	<p>Gira el personaje a la izquierda cierta cantidad, produciendo 1 un giro de 30°</p>
<p>Saltar</p> 	<p>Hace saltar el personaje cierto número de casillas y luego lo baja a la situación original.</p>
<p>Ir al inicio</p> 	<p>Reinicia la posición del personaje a su posición inicial. (Para dar una nueva posición inicial, arrastra el personaje a dicha posición.)</p>



Bloques de apariencia	
<p>Decir</p> 	<p>Muestra un mensaje dado en un bocadillo (de comic) emitido por el personaje</p>
<p>Disminuir</p> 	<p>Reduce el tamaño del personaje</p>
<p>Ocultar</p> 	<p>Desvanece al personaje hasta hacerlo invisible</p>



<p>Crecer</p> 	<p>Aumenta el tamaño del personaje</p>
<p>Restablecer tamaño</p> 	<p>Devuelve el personaje a su tamaño original</p>
<p>Mostrar</p> 	<p>Hace visible gradualmente al personaje</p>

Bloques de sonido	
<p>Pop</p> 	<p>Reproduce el sonido “pop”</p>
<p>Reproducir sonido grabado</p> 	<p>Reproduce un sonido grabado por el usuario</p>

Bloques disparadores	
<p>Al presionar bandera verde</p> 	<p>Comienza el guión cuando se pulsa la bandera verde</p>

<p>Comenzar al tocar</p> 	<p>Comienza el gui3n cuando el personaje es tocado por otro personaje</p>
<p>Enviar mensaje</p> 	<p>Envía un mensaje del color seleccionado</p>
<p>Comenzar al pulsar</p> 	<p>Comienza el gui3n cuando se pulsa el personaje</p>
<p>Comenzar con mensaje</p> 	<p>Comienza el gui3n cuando se envíe un mensaje del color seleccionado</p>

<p>Bloques de control</p>	
<p>Esperar</p> 	<p>Para el gui3n durante cierto tiempo (en d3cimas de segundos)</p>
<p>Fijar velocidad</p> 	<p>Cambia la velocidad a la que ciertas instrucciones se ejecutan</p>

<p>Parar</p> 	<p>Para todos los guiones del personaje</p>
<p>Repetir</p> 	<p>Repita cierto número de veces las instrucciones que están dentro del bloque</p>

Bloques de finalización	
<p>Finalizar</p> 	<p>Indica el final del guión</p>
<p>Ir a la página</p> 	<p>Salta a la página seleccionada del proyecto</p>
<p>Repetir indefinidamente</p> 	<p>Ejecuta este guión continuamente</p>

3 Metodología

El proceso de indagación se realizó en dos fases, una de familiarización y otra de indagación propiamente dicha.

En la fase de familiarización se construyeron todos los programas mostrados en las tarjetas de ScratchJr desarrolladas por sus creadores [4] para conocer el lenguaje y su entorno de programación. Las tarjetas 3 y 9 se usaron también en la fase siguiente.

Asimismo, eventualmente se realizaron otros programas. Destacamos un programa (véase Programa 1, en Apéndice) donde, de forma no intencionada, se comprobó que el bloque “comenzar al tocar” puede activarse en cualquier momento en que dos personajes se toquen, incluso contra la voluntad del usuario.

En una segunda fase, se comenzó indagando sobre aspectos concretos del comportamiento de los bloques de manera informal. Sin embargo, enseguida se comprobó que un estudio riguroso exigía un diseño sistemático de casos de prueba que cubriera todos los aspectos de todas las instrucciones del lenguaje.

Veamos los casos de pruebas diseñados, desglosados por categorías de bloques. No se realizaron pruebas con los bloques de sonido.

3.1 Bloques de movimiento

Las pruebas para esta categoría fueron sencillas, limitándose en general a la ejecución de uno o varios bloques bajo ciertas condiciones. Para poder apreciar su efecto, se activó la cuadrícula.

Primero se realizaron pruebas sobre el sistema de referencia espacial:

- Tamaño en casillas del escenario.
- Posición inicial por defecto de cada personaje añadido.
- Si el sistema de referencia es circular (p.ej. un personaje que sale por la derecha aparece por la izquierda).

Asimismo, se realizaron pruebas con los bloques de movimiento:

- Comportamiento espacial de bloques de avance: si cada paso se corresponde con una casilla de la cuadrícula, si admite un número de pasos nulo o negativo. Solamente se realizó con uno de los dos bloques de avance.
- Comportamiento temporal de bloques de avance: si el tiempo de avance de cierto número de casillas depende del número de pasos a dar o del número de bloques implicadas en el avance. En este caso, se realizaron dos pruebas distintas, comparando en cada una el tiempo empleado por dos personajes que avanzaban de forma distinta:
 - Comprobar si un bloque de avance de 1 paso tarda lo mismo que uno bloque de 5 pasos.
 - Comprobar si un bloque de avance de 10 pasos tarda lo mismo que dos bloques de 5 pasos.
- Comportamiento espacial de bloques de salto: Igual.
- Comportamiento espacial de bloques de giro: cuántos grados corresponde cada giro, si admite un número de giros nulo o negativo. Solamente se realizó con uno de los dos bloques de giro.

3.2 Bloques de apariencia

Solamente se realizaron dos pruebas:

- Bloque “decir”: Se determinó la longitud máxima del texto de un bloque.

- Bloques “crecer” y “disminuir”. Se intentó determinar el incremento o decremento del tamaño de un personaje sobre la cuadrícula.

3.3 Bloques de control

Se realizaron pruebas sobre los cuatro bloques:

- Bloque “repetir”. Se indagó si estos bucles admiten:
 - Cero iteraciones.
 - Un número negativo de iteraciones.
 - El número máximo de iteraciones.
- Bloque “esperar”. Igual que el bloque anterior.
- Bloque “fijar velocidad”.
 - Dado que este bloque afecta a la velocidad futura de otros bloques, se desarrolló un programa (inspirado en la tarjeta 3 de [2], véase Programa 2 en Apéndice) en el que se comparaba la velocidad del comportamiento de tres personajes cuyos programas sólo diferían en su velocidad. A veces se probó el efecto con la ejecución de un bloque, a veces su repetición indefinida. Los bloques probados son:
 - Movimiento, avanzar.
 - Movimiento, girar.
 - Movimiento, saltar.
 - Apariencia, hablar.
 - Apariencia, aumentar y disminuir de tamaño.
 - Apariencia, ocultarse.
 - Control, repetir (cinco avances).
 - Control, esperar.
 - Se quería saber si la velocidad se mantiene a través de múltiples ejecuciones. Se usó un programa parecido al anterior, con varios personajes que avanzan continuamente.
 - Se quería saber si la velocidad se asocia al personaje o al guión actual (véase Programa 3 en Apéndice).
- Bloque “parar”. Se preparó un programa con dos personajes iguales pero con distintos guiones (véase Programa 4, Apéndice). El objetivo era comprobar si la ejecución del bloque por un personaje afecta a:
 - El guión donde se ejecuta.
 - Otros guiones del mismo personaje.
 - Otros personajes.

3.4 Bloques disparadores

Se realizaron pruebas sobre dos tipos de bloques:

- Bloque “comenzar al tocar”. Una vez detectado el comportamiento difícilmente controlable de este bloque (comentado al inicio de esta Sección 3,

véase Programa 1, en Apéndice), se realizaron diversas pruebas para intentar controlar que este bloque deje de activarse.

- Bloques de mensajes. Como es obvio, los bloques “comenzar con mensaje” y “enviar mensaje” se estudiaron conjuntamente. Se elaboraron dos programas distintos:
 - Un programa de elaboración propia (véase Programa 5, Apéndice) para comprobar que el comportamiento asociado a un envío y recepción de mensaje es como el de una llamada a subrutina, más que el de la comunicación síncrona entre procesos. Se realizaron cuatro variaciones para estudiar diversos detalles.
 - Un programa basado en la tarjeta 9 de [2] (véase Programa 6, Apéndice) para comprobar el comportamiento de estos bloques cuando intervienen varios receptores.

3.5 Bloques de finalización

Las pruebas se centraron en el bloque de salto de página. Al intervenir ámbitos, se hicieron varias pruebas:

- Personajes. Se comprobó si pueden hacerse copias del mismo personaje, cada uno con distintos guiones (véase Programa 7, Apéndice).
- Páginas. Se comprobó:
 - El número máximo de páginas.
 - El número máximo de personajes por página.
 - Si es posible tener el mismo personaje en distintas páginas, pero con guiones distintos en cada una, y si se ejecutan solamente los guiones de la nueva página (véase Programa 8, Apéndice).
 - El efecto de la bandera verde sobre los personajes, según la página activa.

4 Resultados

En este apartado realizamos un resumen de los hallazgos determinados experimentalmente, sin que sea una descripción estructurada del comportamiento de los programas ScratchJr.

4.1 Cuestiones generales

Se han determinado algunas cuestiones relacionadas con el escenario y las páginas de los programas ScratchJr:

- Escenario:
 - Cuadrícula de 20 casillas de ancho por 15 de alto.
 - Cada nuevo personaje añadido al escenario se sitúa en el centro, casilla horizontal 11 y vertical 8.

- El escenario es circular, es decir, un personaje que sale por la derecha aparece por la izquierda y si sale por arriba aparece por abajo, y viceversa.
- Personajes:
 - El estado de cada personaje viene dado por: posición en los ejes X e Y, su orientación, su tamaño, si es visible, si está diciendo algo y velocidad de ejecución de sus bloques.
 - El estado de cada personaje se restaura cada vez que se pulsa la bandera verde, salvo la velocidad de ejecución, que se mantiene. Por tanto, puede distinguirse entre estado inicial del personaje y estado en ejecución.
 - El estado inicial de cada personaje puede cambiarse mediante su manipulación con el ratón o la ejecución de bloques de movimiento o apariencia.
- Páginas:
 - Número máximo de páginas: 4.
 - Número máximo de personajes por página. Indeterminado: hemos añadido 24 personajes, momento en el que hemos abandonado la indagación porque es un número de personajes que raramente va a alcanzarse.

4.2 Bloques de movimiento

Se dedujeron los siguientes comportamientos:

- Comportamiento espacial de bloques de avance:
 - Cada paso se corresponde con una casilla de la cuadrícula.
 - Se admite un número de pasos nulo o negativo, avanzando en este último caso en dirección contraria a la orientación del personaje.
- Comportamiento temporal de bloques de avance:
 - El tiempo de avance es proporcional al número de pasos a dar, independientemente del número de bloques ejecutados.
- Comportamiento espacial de bloques de salto:
 - Cada salto se corresponde con una casilla de la cuadrícula.
 - Se admite un número de saltos nulo o negativo, análogo al avance.
- Comportamiento espacial de bloques de giro:
 - Cada giro corresponde a 30°.
 - Se admite un número de giros nulo o negativo, análogo al avance.

4.3 Bloques de apariencia

Se dedujeron los siguientes comportamientos:

- Bloque “decir”. Pudo escribirse el texto “La longitud máxima de un texto coincide con la lon”. Por tanto, la longitud máxima del texto es de 50 caracteres.

- Bloques “crecer” y “disminuir”:
 - Hemos comprobado que el gato puede aumentar de tamaño hasta 34 veces y puede disminuir 7. Sin embargo, el pollo, que es un personaje más pequeño que el gato, pudo aumentarse en 41 y disminuir en 5. Por tanto, el número de incrementos y decrementos posibles dependen del tamaño del personaje.
 - No hemos encontrado ninguna referencia clara para medir el incremento del tamaño en función de la cuadrícula. De forma visual, con la cuadrícula activada, parece que la altura del gato y del pollo se doblan tras 10 incrementos.

4.2 Bloques de control

Se dedujeron los siguientes comportamientos:

- Bloque “repetir”:
 - Admite cero iteraciones, en cuyo caso realiza una.
 - No admite un número negativo de iteraciones.
 - El número máximo de iteraciones es 99 (espacio de 2 dígitos).
- Bloque “esperar”. Igual que el bloque anterior.
- Bloque “fijar velocidad”:
 - Resulta afectada la velocidad de ejecución de todos los bloques de movimiento y todos los de apariencia, salvo el bloque “decir”.
 - No resultan afectados el resto de bloques, que son de control.
 - La velocidad queda fijada de una ejecución para otra.
 - La velocidad se asocia cada personaje, no a cada guión.
- Bloque “parar”:
 - No para el guión donde se ejecuta, pero sí paran otros guiones del mismo personaje.
 - No afecta a los guiones de otros personajes.

4.2 Bloques disparadores

Se dedujeron los siguientes comportamientos:

- Bloque “comenzar al tocar”. Se ratificó la dificultad de evitar la activación de este bloque.
- Bloques de mensajes. El comportamiento básico de este bloque es igual que una llamada a subrutina. Si hay una cadena de envío de mensajes entre guiones, se quedan pendientes de reactivar en orden inverso, como en la pila de las llamadas a métodos. Podemos añadir los siguientes matices:
 - Si el guión receptor acaba en un bucle infinito, el guión parado del emisor no retoma el control.
 - Si el guión receptor no tiene bloque de finalización, el emisor recibe la notificación de fin del guión receptor.

- Si no existe guión receptor, el emisor continúa inmediatamente con el siguiente bloque del envío de mensaje.
- Si varios guiones reciben un mensaje, lo hacen simultáneamente. El guión emisor no sigue hasta que todos los receptores han terminado.
- De la misma forma que en ciertos casos la recursividad es equivalente a la iteración (como la recursividad final), a veces puede conseguirse el mismo efecto de varias formas en paso de mensajes. En concreto, sea un emisor y varios receptores en cadena. Puede conseguirse el mismo efecto si el emisor realiza una operación tras el retorno del envío de mensaje que si el último personaje en la cadena de mensajes, le envía otro mensaje, que el emisor recibe.

4.2 Bloques de finalización

Se dedujeron los siguientes comportamientos:

- Personajes:
 - Pueden hacerse copias del mismo personaje, cada uno con sus distintos guiones.
 - Los guiones están asociados a un personaje, que define un ámbito.
- Páginas:
 - Los personajes están asociados a una página, que define otro ámbito.
 - Una consecuencia de los dos ámbitos identificados es que un evento sólo afecta a los guiones de los personajes de la página activa. Esto es importante para el comportamiento de los bloques disparadores, incluyendo la bandera verde.

5 Conclusiones

Hemos presentado una indagación sobre el lenguaje de bloques ScratchJr. Una parte se ha realizado de forma no planificada pero otra ha sido planificada. Como resultado, conocemos mejor el comportamiento, no documentado, de los distintos elementos y bloques del lenguaje.

Los resultados presentados no constituyen una descripción estructurada del lenguaje ni una máquina nocional. Sin embargo, dado que ScratchJr es un lenguaje pequeño, no parece una tarea compleja ni laboriosa. Esta tarea queda pendiente como trabajo futuro.

Agradecimientos. Este trabajo se ha financiado con el proyecto de investigación S2018/TCS-4307 de la Comunidad Autónoma de Madrid. Este proyecto también está financiado con los fondos estructurales FSE y FEDER.

Referencias

1. Velázquez-Iturbide, J. Á., coord.: Informe del grupo de trabajo SCIE/CODDII sobre la enseñanza preuniversitaria de la informática. Sociedad Científica Informática de España (SCIE), 2018. <https://www.scie.es/wp-content/uploads/2021/02/Informe-SCIE-CODDII-2018-06.pdf>
2. Martínez Valdés, J.A., Velázquez Iturbide, J.Á., Hijón Neira, R.: A (relatively) unsatisfactory experience of use of Scratch in CS1. En: Proceedings of 5th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'07). ACM DL, 2017, 6 págs., DOI [10.1145/3144826.3145356](https://doi.org/10.1145/3144826.3145356)
3. Bers, M.U., Resnick, M.: The Official ScratchJr Book: Help Your Kids Learn to Code. No Start Press (2016)
4. Bers, M.U., Sullivan, A.: ScratchJr Coding Cards: Creative Coding Activities. No Start Press (2018)
5. Flannery, L.P., Kazakoff, E.R., Bontá, P., Silverman, B., Bers, M.U., Sullivan, A.: Designing ScratchJr: Support for early childhood learning through computer programming. En: Proceedings of the 12th International Conference on Interaction Design and Children (IDC'13). ACM DL, 2013, págs. 1-10, DOI [10.1145/2485760.2485785](https://doi.org/10.1145/2485760.2485785)
6. ScratchJr – Aprende – Guía de bloques. <https://www.scratchjr.org/pdfs/block-descriptions.pdf>

Apéndice: Programas de Prueba

Incluimos a continuación los programas de prueba desarrollados y citados en el Apartado 3, organizados por el bloque que pretenden indagar.

Bloque “comenzar al tocar” – Programa 1

Se presenta una situación con cuatro personajes en un desierto. El sol está situado tras el cacto, asciende en el cielo y desaparece. El cacto varía de tamaño al ser tocado por cualquier otro personaje. La serpiente avanza continuamente hacia su derecha (en la figura se ha colocado “en el cielo” para que no toque al cacto). Finalmente, el camello tiene dos guiones: uno para avanzar a su derecha indefinidamente y otro que, al tocar a otro personaje, retroceda rápidamente y pare su movimiento.

Aunque la separación de tareas entre guiones parece sencilla, en la práctica la serpiente y el camello tocarán al cacto y entre sí y se hará casi imposible llevar al programa a un estado en el que no se estén activamente continuamente los guiones del cacto y del camello de “comenzar al tocar”.



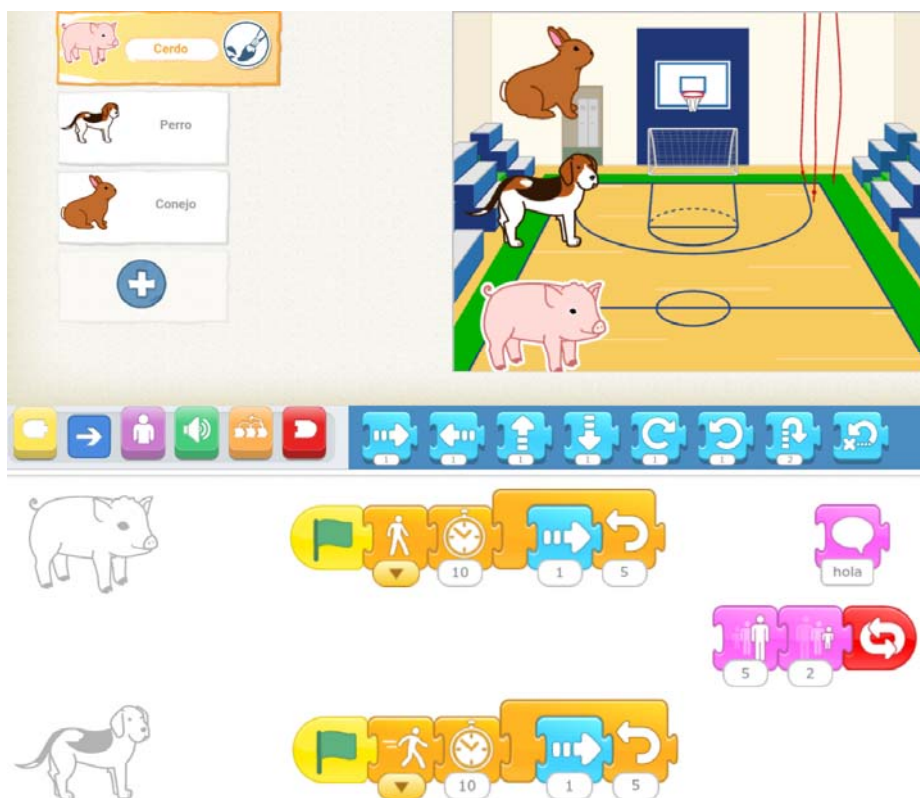


Bloque “fijar velocidad” – Programa 2

Sean tres personajes situados en la misma situación relativa y con mismo programa simple salvo la velocidad. El cerdo es el más lento y el conejo, el más rápido. Los tres personajes pueden estar en la misma posición horizontal o vertical, según resulte más adecuado para el bloque a probar (p.ej., para avanzar conviene que estén situados en vertical, mientras que para saltar conviene que estén en horizontal).

La figura muestra un ejemplo para el bloque de avanzar. Puede observarse que el programa de los tres personajes es el mismo, salvo en las opciones seleccionadas en su bloque de velocidad. Asimismo, a la derecha del guión del cerdo se encuentran algunos de los bloques que se han utilizado para probar el efecto de la velocidad (en estos casos, bloques de apariencia). A veces se terminó los guiones con el bloque de “repetir indefinidamente”.

Para probar el bloque “esperar”, se usó un programa parecido al de la figura con el bloque “decir”, aprovechando que ya se sabía que su velocidad no variaba.





Bloque “fijar velocidad” – Programa 3

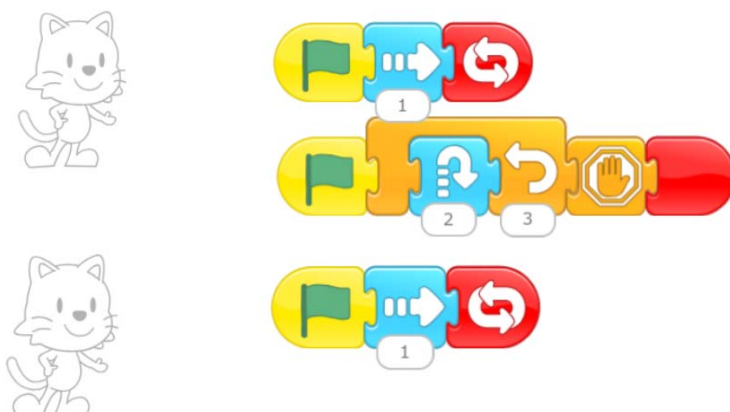
Sean dos personajes situados en la misma situación relativa y con mismo programa simple de avanzar repetidamente. Usando un bloque suelto de la paleta, hacemos que el primer personaje avance lo más rápido posible y el otro, lo más lento.

El escenario es igual que en el Programa 4. Para comprobar si la velocidad es una propiedad del personaje y no del guión, incluimos un bloque disparador “comenzar al tocar” que dé velocidad lenta al personaje rápido. Si se trata de una propiedad global del personaje, debe afectar a la velocidad de avance de su otro guión.



Bloque “parar” – Programa 4

Sean dos personajes iguales tales que cada uno tiene un guión que simplemente avanza indefinidamente. Además, el primer personaje tiene un guión adicional que contiene un bloque “parar”. Podrá comprobarse si la ejecución de dicho bloque paraliza el otro guión del mismo personaje así como el único guión del otro personaje. Podría haberse incluido un bloque “decir” tras el bloque “parar”, para anunciar que ha parado todo, pero es innecesario.



Bloques de mensajes – Programa 5

Se desarrolló un diálogo entre dos personajes (Gato y Pollo) para comprobar que el emisor siempre esperaba hasta que el guión activado terminaba. Se utilizaron los bloques de “decir” para poder rastrear la ejecución del programa.

Gato dice a Pollo sucesivamente: “Hola, Pollo”, “Si te tocan, avísame”. Y envía un mensaje a Pollo para que se entere. Pollo dice “De acuerdo” al recibir el mensaje. Tras acabar el envío, Gato contesta “Gracias, Pollo”.

Cuando se toca al Pollo, éste da un salto sobresaltado, dice “Ya, Gato” y le envía un mensaje. Gato da las gracias y desaparece.



Para comprobar aspectos adicionales, se introdujeron algunas variaciones para comprobar si Gato retoma la ejecución de su guión:

- El gui3n de Pollo que recibe el primer mensaje se hace acabar en un bucle indefinido.



- Se quita el bloque terminador del gui3n de Pollo.



- Se suprime el gui3n receptor de Pollo.



Bloques de mensajes – Programa 6

Se modificó la tarjeta 9, en la que un profesor y tres alumnos se envían mensajes de forma secuencial. En la modificación, no sólo dicen algo al recibir el mensaje, sino tras terminar su envío de mensaje. El diálogo es:

Profesor: “¡Bienvenidos!” “Mi nombre es sr. Martínez”

Niña negra: “Hola. Soy la primera en hablar”

Niño: “Hola. Soy el segundo en hablar”

Niña blanca: “Hola. Soy la tercera en hablar”

Profesor: “¡Qué bien que volvemos a encontrarnos!”

Niña blanca: “Ahora soy la primera en retomar la palabra”

Niño: “Y yo el segundo en retomar la palabra”

Niña negra: “Y yo la tercera”

Profesor: “Yo termino con el turno de palabra”





También se modificó el programa para comprobar el comportamiento de los personajes cuando hay recepción del mismo mensaje por parte de varios personajes. Se han probado varias variantes sobre el mismo programa del aula:

- El profesor envía un único mensaje a los tres alumnos y tiene un bloque “decir” tras el envío del mensaje.
- El niño y la niña negra reciben el mensaje y saludan al mismo tiempo.
- La niña blanca tarda un poco más en terminar (con un bloque “esperar”).
- La niña blanca no termina.



Personajes y guiones – Programa 7

Se incluyeron dos gatos que siempre avanzaban a la derecha, aunque uno lentamente y otro rápidamente.



Páginas – Programa 8

Se desarrollaron dos programas sencillos formados por dos páginas, cada una con un gato:

- El gato de la primera página corre indefinidamente y el de la segunda, salta indefinidamente.
- El gato de la primera página corre 5 pasos y se va a la siguiente página y el de la segunda, salta indefinidamente.

