



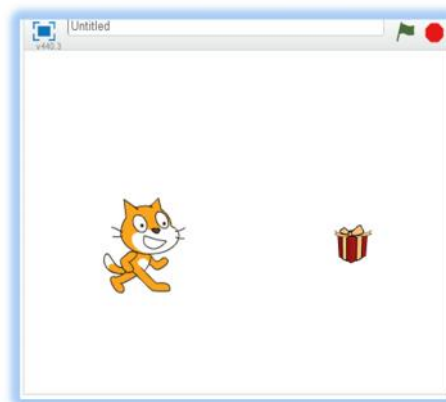
INTRODUCCIÓN A LA PROGRAMACIÓN CON SCRATCH 2.0 Parte IV

TEMA 2: Interacción I

Si, nuestros objetos pueden interactuar con su entorno, con el fondo del escenario y con el resto de objetos que se encuentren en el...

Vamos a darle un regalito al gato, colocando un nuevo objeto llamado GIFT desde la biblioteca general, le ajustamos el tamaño para que no sea tan grande.

Si movemos el gato sobre el regalo, pasará por detrás sin pestañear. ¿Por qué?, pues por que el regalito, ha sido el último objeto en ser creado y por tanto, ocupa una posición superior en el escenario, es decir, se pone por delante del resto de objetos que ya estuvieran en el.



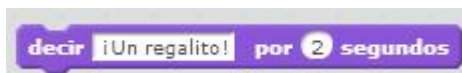
Intentaremos, que cuando el gato, toque el regalo, exclame un mensaje de voz (bocadillo de texto), diciendo alguna tontería, como ¡Un regalito!.

De forma similar a cuando controlamos si el gato estaba tocando un borde, podemos preguntar (durante el movimiento), si está tocando el regalo.



Pero en esta ocasión, si usamos un bloque de control SI básico.

Y para que el gato exclame un mensaje, usamos un bloque de la categoría apariencia



Todo junto...dentro del POR SIEMPRE junto al otro SI-SINO.

(Nota: Yo reduciría el tiempo para que el gato no se quede parado demasiado)





Como veis, en este caso, el que detecta que ha tocado el regalo es el gato. Podemos ver, que el propio regalo detecte que ha sido tocado por el gato...

Seleccionamos de la biblioteca el objeto regalito o GIFT, que es su nombre y colocamos en programas este bloque de control, similar al que usamos con el gato.



Obviamente, para que este bloque se ejecute, hemos de meterlo dentro de un EVENTO de al PRESIONAR Bandera Verde.

Además, si queremos que el bloque se esté ejecutando (analizando) constantemente durante el juego, deberemos meterlo todo dentro de un bloque de control POR SIEMPRE.

Quedaría así:



Ahora, solo nos queda establecer que queremos que ocurra cuando el gift detecte que ha sido tocado.

Por ejemplo, que se esconda y así, que parezca que ha desaparecido.

De la categoría “apariencia”, disponemos de un bloque que permite al objeto esconderse/mostrarse.





El problema, lo veréis cuando probéis el juego....si, ahora, cuando el gato toca el gift, el gift desaparece pero el gato ya no dice nada.

Esto, ocurre por lo anteriormente explicado...la ejecución es tan rápida, que prácticamente el gift ha desaparecido antes de que el gato diga nada.

Para evitar esto, hay que darle tiempo al gato haciendo que el “esconder” se produzca un poco más tarde, pero poco más ¿eh?



CLONES

¿Tengo que contaros lo que es un clon?, supongo que todos sabéis, que un clon es una copia exacta de un objeto o ser vivo, con sus mismas propiedades y características, sobre todo, físicas (lo de que además tengan el mismo comportamiento, está por ver).

En Scratch, podemos crear clones de un mismo objeto, sabiendo que en nuestro caso, no solo se parecerán físicamente, si no que tendrán el mismo comportamiento, pues junto a la imagen, se copian todos los sonidos y programas que tuviera el objeto original.

La gran diferencia entre un clon y su original, es que el original no se puede destruir, sólo esconder/mostrar, sin embargo, el clon puede ser destruido. Además, dispone de eventos propios para que sepa lo que tiene que hacer en cuanto es creado.

Lo más normal, es que el objeto original del que vamos a crear clones, no sea usado realmente, por lo que debería ser ocultado/escondido nada más empezar el juego.

Ahora bien, ¿Quién debe dar la orden de crear los clones, cuantos y cuando?

En un principio, haremos que sea el propio escenario nada más empezar, cree tantos clones como nos interese. Y que el propio objeto original se esconda también nada mas empezar.

Si lo veis bien, os daréis cuenta que simplemente es separar el código del objeto tratado (en este caso el Gift), haciendo que al Presionar Bandera Verde, se esconda y el resto de programa que sería el cómo debe actuar el clon a partir de ese momento, hay que ponerlo en el evento que controla el momento en que se crea un clon.

Añadiremos dentro, en vez de ESCONDER, un Borrar este clon cuando el propio clon detecte que ha sido tocado por el gato.

Quedaría así:



Ahora solo resta crear los clones, como dijimos, desde el programa o código propio del escenario.

Pero al ejecutar, vemos que el regalo no aparece.



Si, resulta que al haber escondido el original, los clones por defecto aparecen escondidos, vamos, que estar, están, pero no se ven.



Le añadimos al EVENTO “Al comenzar como clon” del regalo, el bloque MOSTRAR al principio, antes del por siempre claro, si no estaría mostrándose constantemente.

Y ahora sí, aparece en la misma posición donde estaba el original (que ahora está escondido)

¿Y si queremos crear 5 clones en vez de uno solo?, pues ponemos el bloque Crear clon de Gift dentro de un simple repetir 5 veces dentro de programas del ESCENARIO. .



Pero ahora, al ejecutar, parece que sigue apareciendo un solo clon...¿Por qué?

Pues claro, como sabemos un clon tiene inicialmente las mismas propiedades que su original, por lo que aparecen, uno encima de otro en las mismas coordenadas donde se dejó dicho original.

Y claro, al “coger” el regalo, sólo aparece un único mensaje porque todos, los 5 regalos desaparecen a la vez.

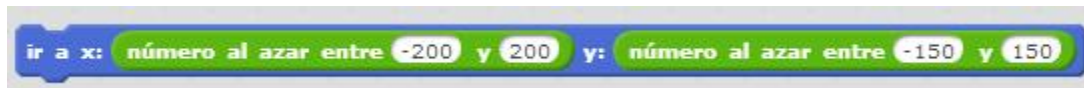
Deberíamos de conseguir que cada regalo clon, aparezca en unas coordenadas del escenario distintas, de forma aleatoria como en cualquier juego.

En el programa de inicio como clon, antes de ser mostrado, le diremos dónde debe ir con un bloque de movimiento.

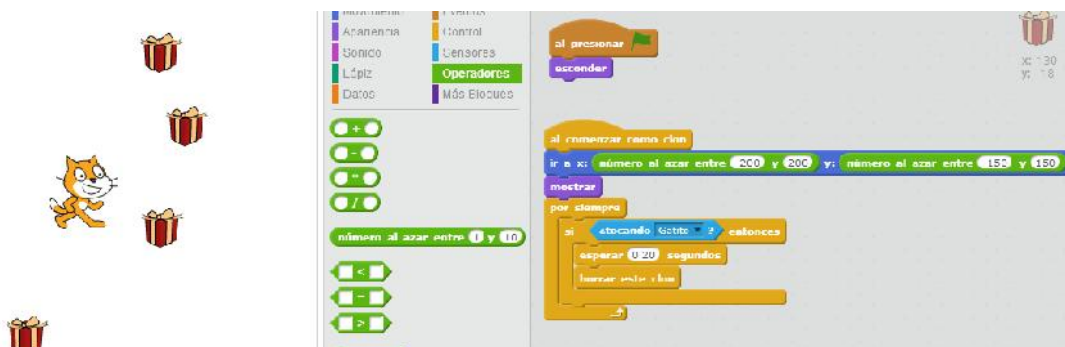


Pero si queremos que cada clon, vaya a unas coordenadas aleatorias, debemos cambiar los números X e Y por dos número calculados de forma aleatoria (sabiendo que la X va desde -240 hasta 240 y la Y va desde -180 hasta 180).

Desde la categoría OPERADORES, arrastramos dos bloques de **Número al azar**. Y los modificamos para ponerlos dentro del bloque de movimiento.



Fijaos, que he reducido las coordenadas para que los clones no se salgan del escenario.



Ahora sí, todo junto nos permite coger clones de la misma caja.

Vamos a mejorar un poco la gracia del juego. Por ejemplo, que cada vez que se coja una caja, suene un Popii.

Accedemos a la pestaña de sonidos del Gift, y de la biblioteca de sonidos, categoría efectos, le agregamos el sonido “pop”.

Ahora, añadimos un bloque de programa de categoría “Sonido”, que haga el play de este sonido en el mismo momento en que desaparece al ser el regalo tocado por el gato.



Hagamos un poco más interesante el juego, es muy fácil recoger los cinco regalos. ¿Y si estos clones desaparecerían al cabo de un tiempo?

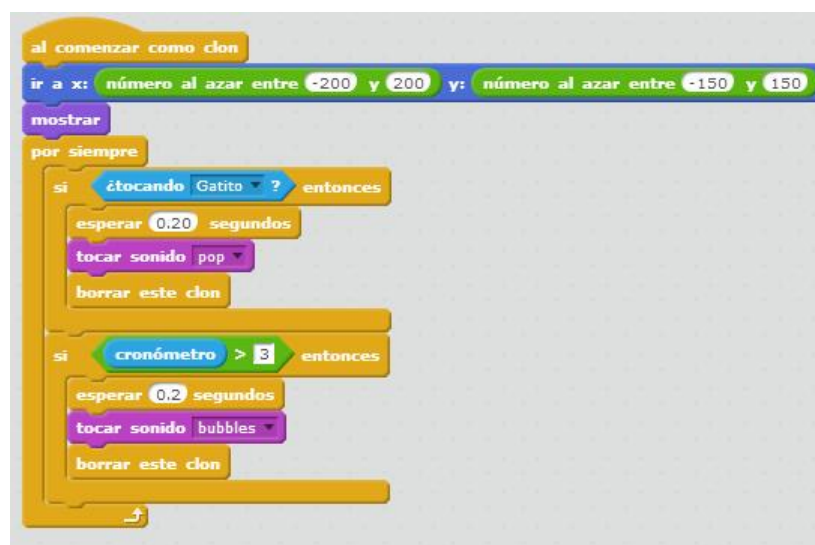
Podemos usar el reloj interno del ordenador, poniéndolo a cero para que cada clon pueda decidir cuando borrarse.

Haremos que sea el ESCENARIO el que reinicie el cronómetro a cero con un bloque de sensor del tipo “Reiniciar cronómetro”.

Y después, añadimos al programa del clon, que otra condición para ser borrado, es que el valor actual del cronómetro sea por ejemplo mayor a 5 segundos.



Podemos duplicar el bloque SI que ya tenemos y modificarlo, añadiendo a nuestra biblioteca de sonidos previamente otro efecto distinto, como el de “bubbles”.



Pero claro, todos los regalos desaparecen a la vez. ¿Cómo lo podemos evitar?

Efectivamente, con el socorrido “Número al azar”, podemos hacer que cada clon desaparezca en un tiempo distinto, sustituyendo el número por el bloque correspondiente...



¡¡Pues no!!, parece que prácticamente hace lo mismo....analicemos lo que tenemos...

Resulta, que el cálculo al azar del tiempo, lo realizamos CADA VEZ que el bloque por siempre se repite, cuando ese valor, debería ser calculado de antemano antes incluso que el clon aparezca en pantalla, para que nunca cambie.

Por eso, haremos aquí una introducción a las **VARIABLES**.

Si tengo que obtener un número al azar antes del bloque POR SIEMPRE que después podamos usar dentro, deberé guardarlo en una variable (ya explicada).

En la categoría DATOS, creamos una nueva variable, que deberá ser local a este objeto (por que al resto de objetos, este dato ni les va ni les viene para nada) y la llamamos por ejemplo “Tiempo”, para poder referirnos a ella posteriormente.



- Vemos que la pantalla en la categoría DATOS, cambia ahora y me ofrece nuevos bloques para esta variable.
- Podemos inicializar la variable a un valor por defecto (de momento es cero).
- Podemos cambiar el valor incrementando lo que ya tuviera mas el que le pongamos ahora, por defecto +1.
- Las variables, por defecto se muestran como si fueran un simple contador en el ESCENARIO, lo podemos ocultar si no queremos verlos.

Ahora, colocamos el bloque “Fijar Tiempo a (número aleatorio)” antes que el clon aparezca.





Y cuando preguntamos por el cronómetro dentro del POR SIEMPRE, usamos dicha variable.



Arrastrándola desde la categoría DATOS.

Bueno, me he pasado con lo de 40 segundos, ponedle menos.

Ahora, desaparecen de forma más gradual.

Fijaos que el contador de tiempo se muestra en la esquina superior izquierda del ESCENARIO, para no verlo podemos simplemente quitarle el Tick a la variable creada dentro de la categoría de programas de DATOS.



Tarea para vosotros, modificar el programa para que podamos llevar y visualizar un contador de regalos que SI hemos podido recoger a tiempo.



Ahora, esta variable, ha de poder ser modificada no por un solo clon como su contador de tiempo, sino por cualquier clon, para incrementarlo si ha sido recogido por el gato.

los objetos)

Por eso la definimos como Global (Para todos

En programas del ESCENARIO



En programas del REGALO/GIFT





Mejorar el juego:

Podemos añadir un peligro para el gatito, que una pelota se mueva de forma aleatoria por todo el escenario, y que si toca al gato, le reste uno en su puntuación.

Como la de la pelota no haremos copias, no hace falta convertirlo en clon.



Si queremos evitar, que la pelota al atravesar al gato, le reste constantemente, deberemos por ejemplo, hacer que la pelota rebote en otra dirección cuando se dé con el gato.

En la categoría de movimiento, el bloque dirección me devuelve en todo momento la dirección en grados que el objeto lleva en ese momento.

Para que la dirección sea la contraria, lo multiplicamos por -1. Y hacemos que coja esa nueva dirección.



Podríamos además, hacer que la dirección variara unos grados de inclinación, como si rebotara contra la pared, en cuyo caso, este efecto va incluido.



Es decir, le sumamos a la dirección un número aleatorio pequeño y después le cambiamos la dirección que lleva al cambiarle el signo.

No parece gran cosa, pero a veces el rebote queda en horizontal y de ahí no sale, con esto evitamos ese efecto tan pernicioso.

Terminamos de mejorar el juego, cambiando el fondo del escenario, por ejemplo.



Todo juego o programa, ha de tener un INICIO una consecución y un FINAL.

Esto lo lograremos usando varios fondos y haciendo que podamos pasar de uno otro según sea pulsando un “Botón” que diseñemos al efecto (por ejemplo, de INICIO comenzamos si se pulsa el botón INICIO del juego) o por tiempo o por otro motivo (por ejemplo, saltar al fondo de FIN si ya no quedan regalos en el escenario).

En nuestro juego, como ejemplo básico, realizaremos lo siguiente:



FONDO INICIO

Modificamos el color del fondo por defecto de nuestro escenario a otro:

Seleccionamos el objeto ESCENARIO y vamos a su pestaña FONDO para entrar en modo dibujo.

En el selector de color de abajo, escogemos por ejemplo este...



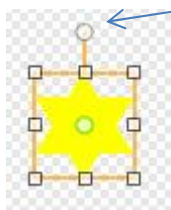
Nuestro objetos ya diseñados, pelota, gato y regalo, no deben aparecer en este fondo, así que entrando en la hoja de propiedades de cada uno de ellos en la biblioteca local, desmarcamos la propiedad,  **mostrar:**  más adelante haremos que estos objetos se muestren.

Ahora, colocaremos copias de objetos del gato, regalo y algunas estrellas para formar una especie de imagen de presentación. Claro que estos objetos, no serán los que se mueva, desaparecerán en cuanto cambiemos de fondo.

Lo centramos en el escenario y modificamos el giro de las estrellitas y tamaños según sea necesario.



Para cambiar el giro de cada estrellita, seleccionamos una por una y vamos a la pestaña Disfraces, con el ratón hacemos una selección rectangular alrededor de la estrella y se verá así...



Usamos este círculo para hacerlo girar al gusto

De momento, no realizamos mas cambios.

Para distinguir que estos objetos son solo para la presentación, podemos cambiarles el nombre.



Ahora nos creamos un “BOTON” para iniciar el juego, haciendo que al ser pulsado, se cambie de fondo, desaparezcan estos objetos y todo comience.

Insertamos un objeto Button3, que lleva dos disfraces, button3-a y button3-b (para cuando es pulsado).



En Disfraz, seleccionamos la herramienta de TEXTO

Con un color azul alegre, escribimos dentro INICIO, y lo centramos en el botón.

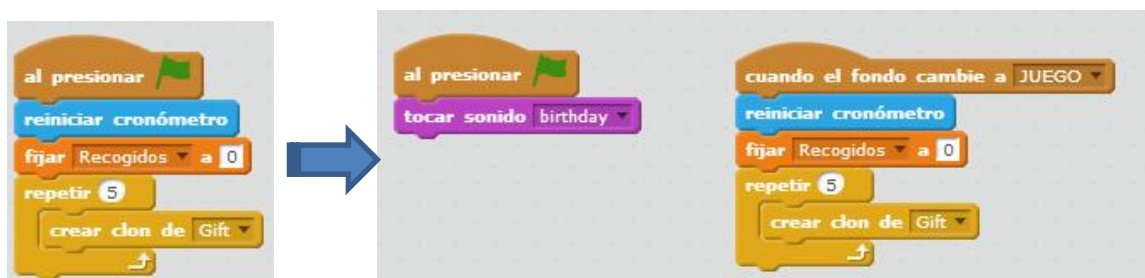


También podemos añadir al escenario un bucle musical de cumpleaños de la biblioteca.

Acciones a realizar

1. Hay que hacer, que al comenzar el juego, en el ESCENARIO se oculten los objetos del juego, los que vamos a manejar, y que los clones se creen sólo cuando cambiemos de fondo al JUEGO.
2. Hay que hacer que al comenzar el juego, se ponga el fondo musical en play y que al darle al botón INICIO, este sonido se pare antes de saltar al siguiente fondo.

ESCENARIO



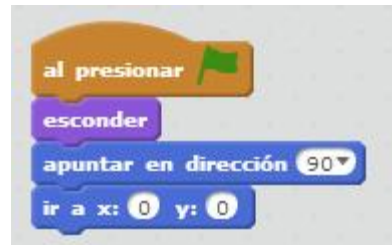


Al iniciar el juego ponemos la música de fondo y cuando el fondo cambie a JUEGO, creamos los clones e inicializamos el contador, además de reiniciar el cronómetro a cero.

Habría que añadir al iniciar bandera verde, que se pusiera el fondo al del INICIO.



GATITO



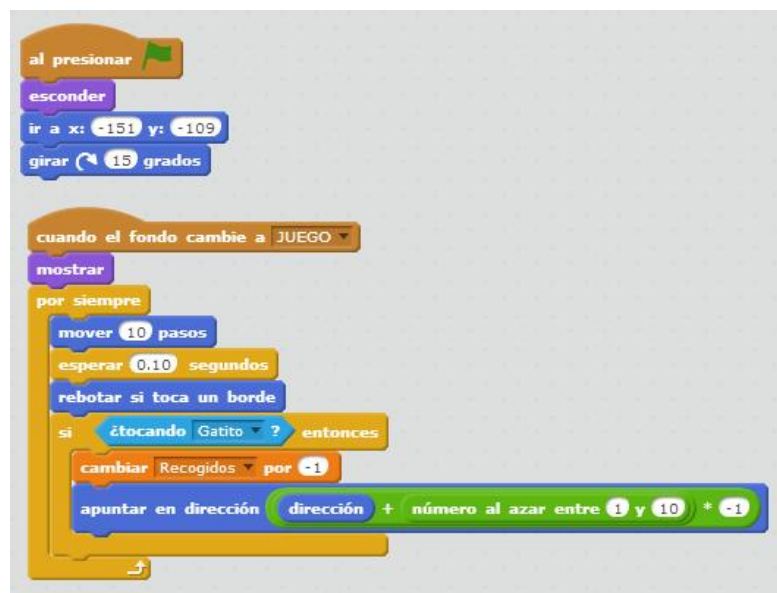
Ya añadimos



PELOTITA

Separamos código y añadimos el evento "Cuando cambie el fondo a JUEGO"

Escondemos la pelota al principio y la mostramos cuando el fondo sea JUEGO.



3. Finalmente, hay que hacer que al saltar al fondo JUEGO, los objetos de presentación se escondan.

En INICIO, INICIO2, INICIO3, INICIO4, INICIO5 y BOTÓN INICIO, en cada uno...



Y claro, parar la música de fondo cuando el escenario cree los clones, antes...



FONDO FIN

Lo más importante, es controlar cuando hemos terminado el juego, y esto puede ser por varias causas, en nuestro caso, normalmente es porque ya no quedan cajas de regalo que coger.

En este momento, habrá que hacer algo parecido, ocultar los objetos de juego y mostrar un fondo con nuevas imágenes presentando el fin del juego.

Una de las formas de controlar que no queden ya cajas, es mantener un contador como el de recogidos pero que indique que la caja se ha recogido o eliminado por tiempo, cuando este contador llegue a la cantidad inicial de clones que creamos (5), haremos que se salte de fondo.

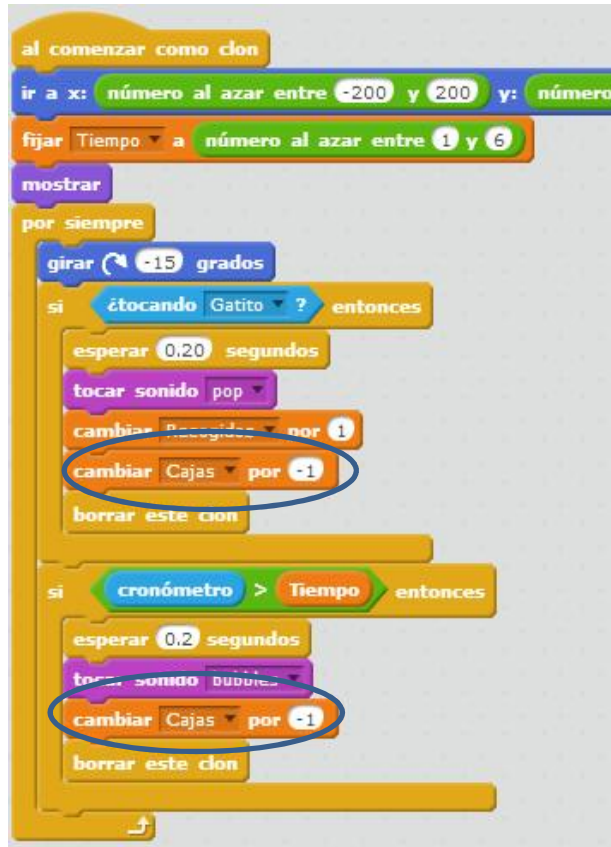


Esta variable, será modificada por cada clon cuando sea cogido o se elimine así misma por tiempo, incrementándose en una unidad.

El ESCENARIO, deberá de comprobar esta variable constantemente para saber en qué momento debe saltar se fondo.

Por tanto:

EN REGALO/GIFT



EN ESCENARIO



Ahora, los objetos del juego, cuando se cambie de fondo al de FIN, tienen que esconderse.

EN GATITO y PELOTA



Hay que diseñar el fondo FIN, podríamos, sabiendo los contadores como están, presentar un texto diciendo ¡Felicidades! O ¡Muy mal!.