

Lección 19

Velocidad

Propósito

Después de una breve revisión de cómo utilizaron el patrón de contador para mover los sprites en las lecciones anteriores, los estudiantes reciben las propiedades que establecen la velocidad y rotación. A medida que utilizan estas nuevas propiedades de diferentes maneras, desarrollan las habilidades necesarias para crear un juego básico de desplazamiento lateral.

En esta Lección, se les enseña a usar los bloques de velocidad para simplificar el código para mover un sprite a través de la pantalla. Esto marca un cambio en cómo se introducen los nuevos bloques. Mientras que los bloques anteriores se presentaron como habilitantes de comportamientos completamente nuevos, ahora se presentan como un código simplificador que los estudiantes podrían haber escrito con los bloques disponibles anteriormente. En las siguientes lecciones, los estudiantes verán cómo este método les permite producir comportamientos de sprites más interesantes.

Secuencia para el aprendizaje

Conocimiento inicial (15 min)

Ampliación del conocimiento (75 min)

Transferencia del conocimiento (5 min)

Objetivos

Los estudiantes serán capaces de:

- Usar los bloques velocidad y rotación para crear y cambiar los movimientos de los sprites.
- Describir las ventajas de simplificar el código mediante el uso de bloques de mayor nivel.

Lección en línea
[Ver en Code Studio](#)

Recursos

¡Atención!

Por favor, haga una copia de cada documento que planea compartir con los estudiantes.

Para los profesores:

- [Contenido de la Lección- Code.org](#)

Código

- `sprite.rotationSpeed`
- `sprite.velocityX`
- `sprite.velocityY`

Estrategia de aprendizaje

Conocimiento inicial (15 min)

Demuestre: Pida que un voluntario pase al frente de la clase y actúe como su sprite. Digamos que le dará instrucciones al sprite como si fuera un programa de Game Lab.

Cuando su estudiante esté listo, organice el lugar para que tengan algo de espacio frente a ellos y pida que “avancen por 1”. Deben dar un paso adelante. Luego repita el comando varias veces, cada vez que esperas que el estudiante avance un paso. Debería apuntar a que la repetitividad de estas instrucciones sea clara. Después de que tu estudiante haya completado esta Lección, pide que vuelvan a donde comenzaron. Esta vez repite la demostración, pero solicita al alumno que “Avance por 2” y haga que el alumno tome 2 pasos cada vez. Una vez que el alumno haya hecho esto varias veces, pida a la clase que dé una ronda de aplausos e invita a los estudiantes volver a su asiento.

Preguntar: Sólo estaba dando instrucciones a mi Sprite, pero parecían ser bastante repetitivas. ¿Cómo podría haber simplificado o simplificado mis instrucciones?

Discute: Dé a los estudiantes un minuto para escribir sus pensamientos antes de invitarlos a compartir con un compañero. Luego haga que la clase comparta sus pensamientos. Puede escribir sus ideas en el pizarrón.

Observaciones: Una forma de simplificar estas instrucciones es decirle a nuestro sprite que siga moviéndose por 1 o 2, o por todos los pasos que queramos. Esto haría las instrucciones más fáciles de entender para los humanos, y como estamos a punto de ver, también hay una manera similar de simplificar nuestro código.

Muestra: Video de Sprite Velocity.

Pantalla: En el pizarrón, escriba las siguientes piezas de código.

- `sprite.velocityX = 4;`
- `sprite.velocityY = -1;`
- `sprite.rotationSpeed = 2;`

Indicación: Acabamos de ver cómo estos nuevos bloques nos

Meta: la demostración anterior debería haber reforzado el hecho de que repetir las mismas instrucciones es algo que nunca harías en la vida real. En su lugar, se le ocurriría una forma de capturar que las instrucciones deberían repetirse, como “seguir avanzando por 1”

Comprobar la comprensión: esto es solo una comprobación rápida para entender después del video. Se les pide rápidamente a los estudiantes que vean si entendieron el punto principal del video y, si no, tienen la oportunidad de reforzarlo antes de pasar a los niveles de Code Studio. Deje la traducción de los bloques de velocidad a si patrón de contador asociado en el tablero para hacer referencia a lo largo de la Lección

ayudan a simplificar el patrón de contador que usamos para mover los sprites. En una hoja de papel, anote el patrón de contador que está reemplazando cada una de estas líneas de código.

Discute: Haga que los estudiantes escriban sus ideas en una hoja de papel antes de hablar sobre sus respuestas como clase. Finalmente, escriba las respuestas correctas en el tablero al lado de cada una, como se muestra a continuación.

- `sprite.x = sprite.x + 4;`
- `sprite.y = sprite.y - 1;`
- `sprite.rotation = sprite.rotation + 2;`

Observaciones: Estos nuevos bloques de “nivel superior” nos ayudan a escribir código que ya sabíamos cómo escribir. Están simplificando nuestro código para nosotros al ocultar algunos de los detalles innecesarios. Sin embargo, como estamos a punto de ver, estos nuevos bloques cambiarán el tipo de programas y juegos que podemos escribir.

Ampliación del conocimiento (75 min)

Aprendiendo a usar los bloques de velocidad

Transición: Los estudiantes se dirigen a Code Studio, [Contenido de la Lección-Code.org](#).

Recorra la sala: Estos niveles introducen las propiedades de velocidad X, velocidad Y velocidad de rotación que acaba de analizar con los estudiantes. Verifique con los estudiantes para ver cómo están y mantenga un registro de cuándo todos han llegado al final del nivel 10.

VelocidadX

Una forma de mover los sprites en Game Lab es con el patrón de contador. Por ejemplo, `sprite1.x = sprite1.x + 1` mueve un sprite por 1 píxel en cada fotograma del ciclo de dibujo. Este patrón es tan común que los sprites tienen una `velocityX`, propiedad que hace esto por ti.

Velocidad de rotación

Ya ha aprendido cómo hacer que su sprite gire usando el `rotation` bloque. Por ejemplo, cuando quería que su sprite girara dos grados cada vez que se dibujaba, colocabas `sprite.rotation = sprite.rotation + 2` dentro del círculo de extracción.

¿Dentro o fuera del lazo de dibujo?: Para los primeros rompecabezas, asegúrese de que los estudiantes estén configurando las velocidades y velocidades de rotación fuera del círculo de dibujo, inmediatamente después de que creen sus sprites. El código funcionará para los primeros rompecabezas, incluso si establecen las velocidades dentro del bucle de extracción, pero causará problemas más adelante.

Ahora puede usar los `rotationSpeed` para hacer que sus sprites giren una cierta cantidad cada vez que se dibujan. Si desea que su sol gire dos grados cada vez que se dibuja, puede usar `sun.rotationSpeed = 2` antes del ciclo de extracción, después de crear su sprite.

Control de velocidad

Usó `rotationSpeed` fuera del lazo de extracción para hacer que tu sprite girara cuando tu programa comienza. También puedes usar `rotationSpeed` dentro del ciclo de dibujo para cambiar la velocidad del sprite durante el juego. Por ejemplo, un Sprite puede comenzar a girar cuando el usuario presiona la barra espaciadora, y seguirá girando hasta que se le indique que se detenga.

Cambiando la velocidad con la posición

Una ventaja del uso de bloques de velocidad dentro de condicionales (if bloques) es que tu sprite se mantiene en movimiento, incluso después de que la condición deje de ser cierta.

Si los estudiantes terminan temprano, anímelos a experimentar con diferentes aspectos del juego de desplazamiento que crearon. Pueden cambiar las animaciones, crear un fondo, etc.

Transferencia del conocimiento (5 min)

Indicación: Hoy aprendieron algunos bloques nuevos. A primera vista, estos bloques hicieron el mismo tipo de cosas que ya habíamos hecho con el patrón de contador, pero nos simplificaron hacerlas. Sin embargo, al pasar por los rompecabezas, comenzaron a hacer algunos movimientos interesantes que no habíamos podido hacer antes.

- Describe uno de esos movimientos y cómo lo hiciste.
- Describe otro movimiento que te gustaría hacer, pero aún no sabes cómo.
- Describe otro bloque que te gustaría tener.
- ¿Tomaría algún argumento?
- ¿Qué haría?
- ¿Qué código escondería dentro?

Observaciones: Todos los movimientos que hicimos hoy son posibles sin los nuevos bloques, pero sería muy complicado codificarlos. Uno de los beneficios de bloques como la velocidad es que cuando no tenemos que preocuparnos por los detalles de los movimientos y acciones simples, podemos usar esa potencia cerebral adicional para resolver problemas más complicados. A medida que desarrolle su juego de desplazamiento lateral, seguiremos buscando nuevos bloques que simplifiquen las cosas, para que podamos construir juegos cada vez más complicados.

Sugerencias para evaluación

Se sugieren los siguientes indicadores para evaluar formativamente los aprendizajes:

- Utilizan bloques de velocidad y rotación, cambiando el movimiento de los sprites.
- Incorporan el código, los medios y las bibliotecas existentes en los programas originales, y le dan atribución.
- Prueban y perfeccionan sistemáticamente los programas.
- Documentan aquellos programas para que sean más fáciles de seguir, probar y depurar.