Lección 21

Movimiento complejo de Sprite

Lección en línea
Ver en Code Studio

Propósito

Los estudiantes aprenden a combinar las propiedades de velocidad de los sprites con el patrón de contador para crear movimientos de sprites más complejos. En particular, los estudiantes aprenderán cómo simular la gravedad, hacer un sprite y permitir que un sprite flote hacia la izquierda o hacia la derecha. En los niveles finales de Code Studio, los estudiantes combinan estos movimientos para animar y controlar un solo sprite y crear un juego simple en el que un personaje vuela y recoge una moneda. Se anima a los estudiantes a hacer sus propias adiciones al juego en el nivel final.

Esta lección muestra la combinación de herramientas, en particular las abstracciones aprendidas en lecciones anteriores, les permite construir nuevos comportamientos para sus sprites. Esto resalta el punto más amplio de que las abstracciones no sólo simplifican el código, sino que también pueden usarse como componentes básicos de un comportamiento aún más complejo.

La Lección presenta algunas de las programaciones más desafiantes. Los estudiantes combinarán múltiples construcciones de programación, incluyendo las propiedades de velocidad, el patrón del contador, la interacción del usuario y la detección de colisiones. Los patrones que los estudiantes usan en esta Lección generalmente son útiles para construir juegos y los estudiantes pueden reutilizarlos más adelante.

Secuencia para el aprendizaje

Conocimiento inicial (10 min)

Ampliación del conocimiento (60 min)

Transferencia del conocimiento (10 min)

Recursos

¡Atención!
Por favor, haga una copia de cada documento que planee compartir con los estudiantes.

Para los profesores:

- Contenido de la Lección- code.org

Objetivos

Los estudiantes serán capaces de:

- Usar velocidad de sprite con el patrón de contador para crear diferentes tipos de movimiento de sprite.
- Explicar cómo las construcciones de programación individual pueden combinarse para crear un comportamiento más complejo.

Estrategia de aprendizaje

Conocimiento inicial (10 min)

Comentario: Los cuatro conceptos principales que los estudiantes aprenderán en la Lección de hoy son isTouching(), keyDown(), sprite.velocityX/ sprite.velocityY. Pida a los estudiantes que hablen con un compañero y recuerden mutuamente para qué se usa cada una de estas cuatro construcciones y cómo funcionan. Como clase, defina cada uno según lo conversado.

Observaciones: En las últimas actividades aprendimos mucho de construcciones de programación poderosas que nos permitieron crear juegos mucho más interesantes. Hoy vamos a explorar cómo la combinación de estas nos dará aún más control sobre los tipos de juegos que podemos hacer.

Ampliación del conocimiento (60 min)

Transición: Dirija los estudiantes a Code Studio, Contenido de la Lección - code.org. Con la excepción de la discusión posterior al primer nivel, los estudiantes trabajarán en Game Lab hasta el final de la lección.

Resumen: La clase aprende a combinar las propiedades de velocidad de los sprites con el patrón de contador para crear un movimiento de sprite más complejo, como simular la gravedad, hacer que un sprite salte y permitir que un sprite flote hacia la izquierda o hacia la derecha. En los niveles finales, la clase combina estos movimientos para animar y controlar un solo sprite y construir un juego simple en el que un personaje vuela y recoge monedas.

Discute: Después de que los estudiantes hayan hecho predicciones sobre cómo se ejecutó rápidamente el código en el primer nivel, analicen por qué vieron que el automóvil comenzó a moverse más rápido. Ten en cuenta que mientras que antes usaban el patrón de contador para aumentar la posición de un automóvil, ahora se usa para aumentar la velocidad del automóvil.

Este nivel introduce el nuevo patrón de programación principal de esta Lección, combinando el patrón de contador con las propiedades de velocidad de los sprites. Anime a los estudiantes a tomar en serio sus predicciones antes de ejecutar realmente el código.

Niveles 2-4: Estos niveles les dan a los estudiantes la práctica de usar la velocidad dentro del patrón del contador.

Velocidad y el patrón contador

Usar una sprite.velocityX propiedad con el patrón de contador cambiará la velocidad de un sprite durante el programa. Esto hace

que el sprite se acelere. Practica un poco usando este patrón tú mismo.

Niveles 5 - 6: Estos niveles introducen el uso de la velocidad y el patrón del contador para ralentizar un sprite, eventualmente moviéndolo en la dirección opuesta. Esto lleva a la introducción de cómo este patrón podría usarse para simular la gravedad.

Niveles 7 a 9: Los estudiantes comienzan a trabajar en un juego de volante. En estos niveles usan las construcciones de programación que han aprendido para hacer que su personaje principal se mueva. El personaje responde a gravedad simulada, saltos y flotantes a izquierda y derecha.

Saltando

Aumentar la velocidad de un sprite dentro del patrón de contador puede simular la gravedad. Al agregar interacciones del usuario, puedes hacer que tu sprite parezca que también salte.

Flotante a la derecha

Ahora están usando el patrón de contador con la velocidad Y del sprite para simular la gravedad y saltar. Si usan la velocidad X del sprite en el patrón contrario, pueden hacer que tu sprite flote de lado a lado también.

Niveles 10 a 12: Los estudiantes agregan una moneda al juego para que su personaje la coleccione. En el último nivel, se les anima a actualizar el juego ellos mismos. Aprovecha esta oportunidad en particular para alentar a los estudiantes a usar otros patrones de programación que hayan aprendido, por ejemplo, crear un marcador.

Transferencia del conocimiento (10 min)

Compartir: Pida a los estudiantes que compartan con sus compañeros de clase qué adiciones hicieron a su último juego. Haga que los estudiantes se centren no sólo en cómo funciona el juego, sino también en cómo se ve el código para crear ese tipo de funcionalidad.

Preguntar: En su papel, hagan dos listas. Primero una lista de cosas nuevas que puedan programar después de la Lección de hoy. En la segunda lista anoten todos los nuevos bloques que aprendieron hoy.

Discute: Haga que los estudiantes compartan sus listas con sus compañeros de clase. Después comparte listas como una clase. Deberían haber enumerado muchos nuevos movimientos de

Objetivo: Esta conversación debe resaltar que los estudiantes no aprendieron ningún bloque nuevo en la Lección de hoy, sino que aprendieron nuevas formas de combinar bloques y patrones que habían aprendido previamente. El punto más amplio aquí es que la programación no siempre se trata de aprender nuevos bloques, sino de ser creativos al combinar las herramientas que ya sabes usar en el lenguaje.

sprites pero los estudiantes no han aprendido ningún bloque nuevo en esta Lección.

Después de avisar que todos los movimientos nuevos que crearon hoy se hicieron combinando bloques y patrones que ya aprendieron, pregunte a los estudiantes qué piensan, qué les puede decir esto sobre cómo los programadores desarrollan el código.

Indicación: Hoy hemos construido muchos movimientos de sprites nuevos, como la gravedad y el salto, pero nada de esto nos obligó a aprender nuevos bloques. ¿Crees que aprender a programar siempre significa aprender nuevos comandos?

Discute: Lidere un seguimiento rápido de su discusión inicial sobre este punto.

Observaciones: Vamos a seguir aprendiendo algunas herramientas más en Game Lab. Para crear nuevos tipos de programas, no siempre es necesario aprender nuevos bloques. La mayoría de las veces, la creatividad de la programación proviene de aprender a combinar cosas que ya conoces de maneras nuevas y creativas.

Meta: Reforzar el hecho de que aprender a programar no es solo memorizar bloques. Ser creativo con la programación a menudo significa idear maneras inteligentes de combinar los comandos y patrones que ya sabe usar.

Sugerencias para evaluación

Se sugieren los siguientes indicadores para evaluar formativamente los aprendizajes:

- Crean nuevos tipos de programas, combinando elementos que ya conocen de manera nueva y creativa.
- Perfeccionan el trabajo involucrando modificaciones y testeo.