CODESPARK ACADEMY

Una Introducción a Ciencias Computacionales
Preescolar y Primaria

Presentado por codeSpark

Edición HOUR OF CODE

codeSpark

codeSpark Guía del maestro

¡Gracias por su interés en enseñar ciencias computacionales a sus estudiantes! El conocimiento de ciencias computacionales y "pensamiento algorítmico" es cada vez más necesaria para tener éxito en nuestro mundo digital. Esta habilidad se está convirtiendo en un componente esencial de la alfabetización del siglo XXI. codeSpark creo *Los Foos* como introducción a "El ABC de Ciencias Computacionales."

Aunque es importante preparar a los niños para el ambiente de trabajo moderno, las ciencias computacionales son mucho más que conseguir un empleo en alta tecnología. Investigaciones muestran que las ciencias computacionales ayudan a los estudiantes a mejorar en áreas como matemáticas, lógica e incluso comprensión en la lectura. A menudo la gente piensa en programación o codificación como ciencias computacionales pero eso es sólo un elemento. Es esencia, las ciencias computacionales son el estudio de cómo utilizar el pensamiento lógico para identificar, simplificar, y resolver problemas complejos. No ceros y unos.

Estudios del MIT y la Universidad de Tufts demuestran que estudiantes de cinco años pueden aprender conceptos complejos de ciencias computacionales, especialmente cuando las barreras artificiales como la sintaxis de programación están fuera del camino.

codeSpark ha creado un enfoque único y poderoso para enseñar ciencias computacionales basado en investigación de vanguardia y cientos de horas de pruebas de prototipo. Los juegos para aprendizaje de codeSpark están diseñados sin palabras por lo que incluso los estudiantes no-lectores o aprendiendo el lenguaje Inglés pueden jugar y aprender de nuestro plan de estudios de gran alcance.

Jugando nuestros juegos, sus alumnos mejorarán sus habilidades de pensamiento crítico y mejorarán en otras disciplinas, ¡mientras se divierten mucho!

-- El equipo de codeSpark

¿Tiene preguntas o comentarios? Escríbanos a info@codespark.org

Consiga nuestra aplicación aquí: http://thefoos.com

Índice

Ciencias Computacionales para Prescolar y Primaria

codeSpark

4 Resumen

10 Glosario

11 Lecciones de la Estuche de Juego

16 Actividades Desconectadas

18 Rúbrica

Descarga el programa completo de 10 lecciones
en thefoos.com/hourofcode

Resumen

Ciencias Computacionales para Prescolar y Primaria

codeSpark

Resumen para el profesor:

Nuestros planes de lección proporcionan una iniciación divertida, flexible y atractiva en conceptos fundamentales de ciencias computacionales. Están enfocados en estudiantes de prescolar hasta 5° grado, pero hemos probado con éxito con estudiantes de hasta 8º grado.

Todos los planes de lección están diseñados para ser altamente adaptables. Usted será el mejor juez de a que sus estudiantes necesitan dedicar más tiempo y de que disfrutan más.

Además, todas las lecciones incluyen tanto una actividad con nuestro juego (*Academia codeSpark con los Foos*) como una actividad "desconectada", que no requiere una computadora u otro dispositivo conectado.

Materiales:

- Nuestro juego, *Academia codeSpark con los Foos*. Descárgalo en thefoos.com. Disponible gratuitamente en iPad, iPhone, dispositivos Android y la web (p. ej. en los principales navegadores Chrome, Safari, IE, etc.).
- Accesorios, según sea necesario: ver la lección para información específica.

No se necesita experiencia, pero se recomienda revisar la lección y jugar algunos niveles del juego antes de enseñarlos por primera vez.

Nota: Este curriculum de la Hora de Codigo te da aproximadamente **2 horas de clases**.

Si usted se registró para nuestro "Foosletter" en thefoos.com, le notificaremos cuando se amplíe el plan de estudios y haya nuevas versiones del juego.

También puede descargar nuestro temario completo de 10 lecciones en thefoos.com/hourofcode.

Resumen

Ciencias Computacionales para Prescolar y Primaria

codeSpark

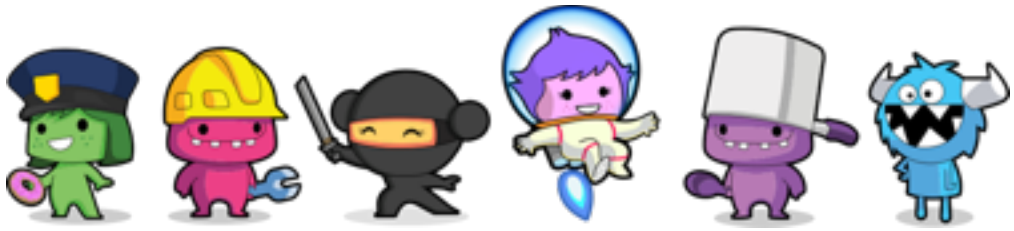
¿Qué son las ciencias computacionales?

Las ciencias computacionales o "CS" se enfocan en resolver problemas con conjuntos muy específicos de instrucciones porque los ordenadores sólo hacen exactamente lo que se les dice que hagan. Pensamos en los ordenadores como inteligentes pero en realidad ¡tenemos que decirles qué hacer! Pues no pueden anticipar lo que queremos de ellos; solo los científicos de la computación pueden dar las instrucciones precisas que necesitan los ordenadores para actuar. Aprender a pensar como científico computacional o un programador ayuda a los niños a analizar problemas, pensar en secuencias lógicas y utilizar lenguaje preciso para dar instrucciones.

La primera lección se enfoca en la identificación de objetos comunes que sólo funcionan cuando les da las instrucciones adecuadas. Entonces vamos a poner esta idea a trabajar programando el Foo policía – el primer personaje que conocemos en nuestro juego.

¿Que son los Foos?

Los Foos son personajes adorables y lindos, recientemente descubiertos por científicos. Son muy pequeños y viven dentro de cada computadora, ¡incluyendo teléfonos inteligentes, tabletas y ordenadores de clase!



Cada Foo puede caminar, saltar, lanzar, comer y navegar en su mundo, llamado "Fooville". Algunos Foos tienen habilidades especiales que los hacen únicos, por ejemplo:

- **Foo Policía** - puede perseguir y capturar el error (Glitch).
- **Foo Cocinero** - puede hacer muchas clases de alimentos.
- **Foo Ninja** – se puede encoger o crecer más grande.

Resumen

Ciencias Computacionales para Prescolar y Primaria

codeSpark

- **Foo Astronauta -** puede viajar en cuatro direcciones diferentes.
- **Foo Constructor** - puede hacer cajas y también explotarlas.

Pero, al igual que los ordenadores, los Foos solo hacen lo que les dicen. Los estudiantes deben aprender a darles comandos específicos, o programar los Foos, en un orden específico.

¡**Cuidado!** El personaje azul con los cuernos blancos es el error (**Glitch**). Es una fuerza del caos en Fooville. A veces hace un lío, el aveces lanza cosas alrededor y a veces aparece inesperadamente.

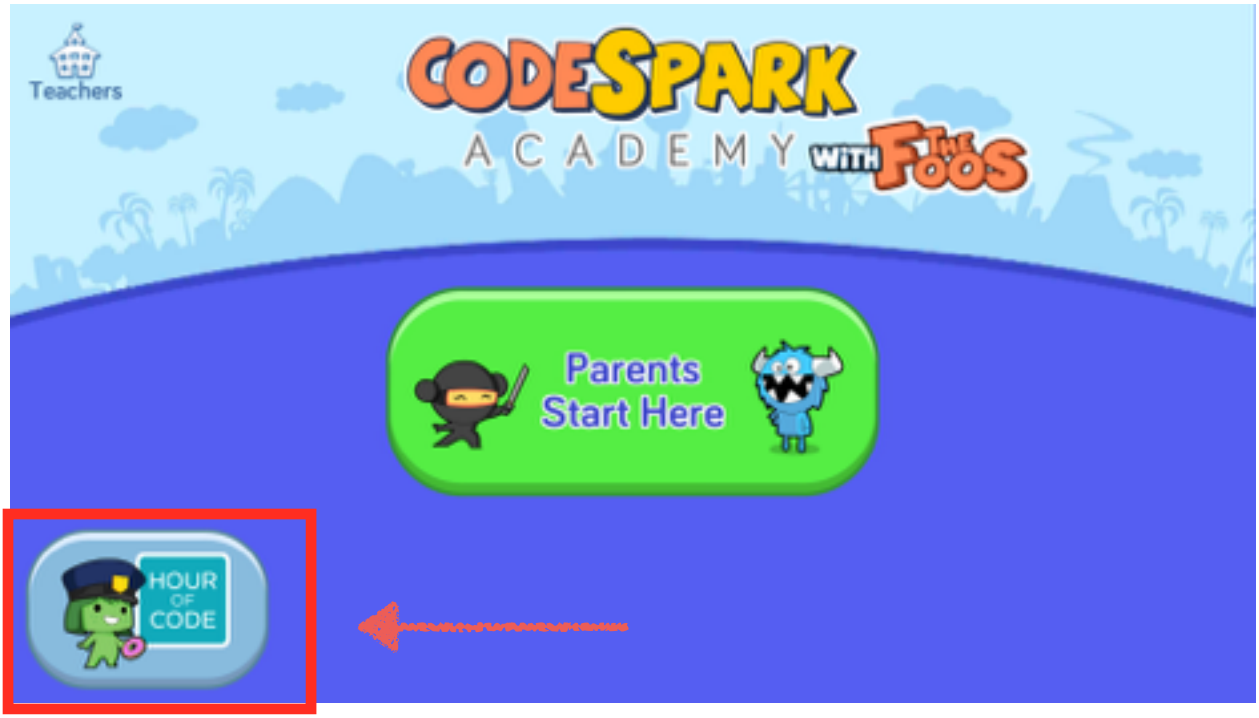
Resumen

Ciencias Computacionales para Prescolar y Primaria

codeSpark

Sugerencias y trucos

Para iniciar la hora del código, presione el botón "Hora del código" en la parte inferior izquierda de la pantalla de inicio.



En la siguiente pantalla puede seleccionar la experiencia que le gustaría para sus estudiantes: rompecabezas o el creador del juego.

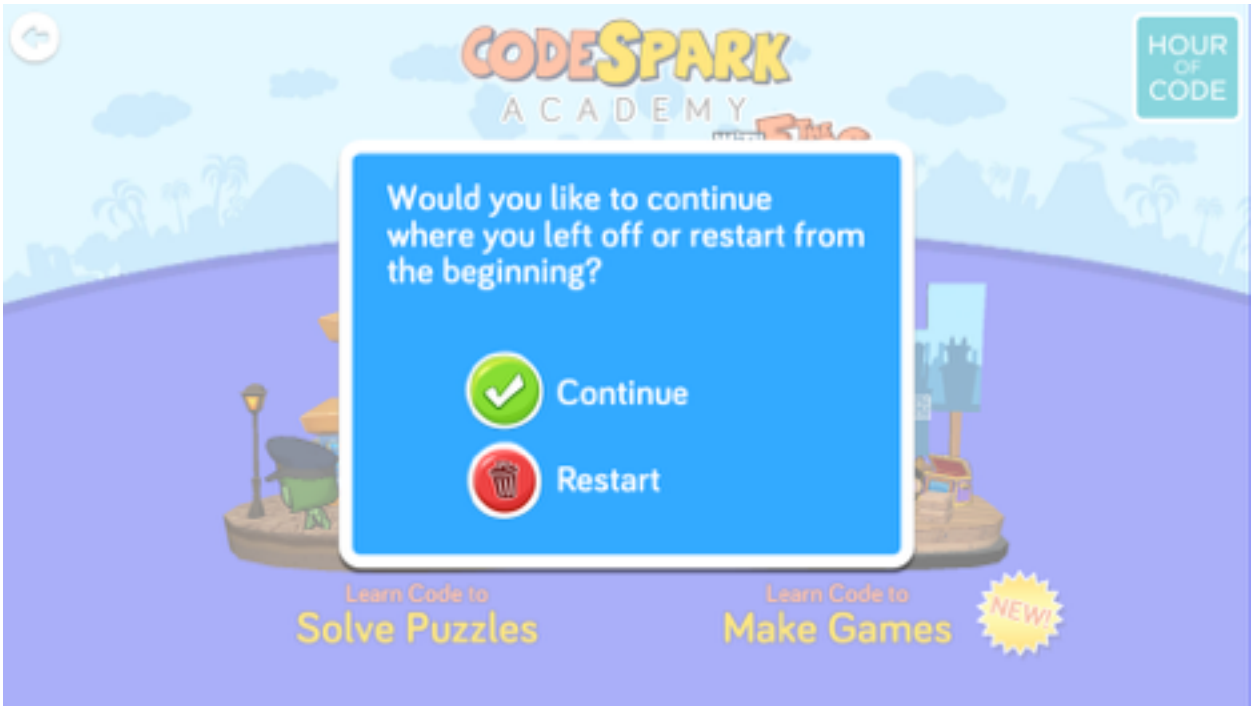
Resumen

Ciencias Computacionales para Prescolar y Primaria

codeSpark

Restablecer su progreso:

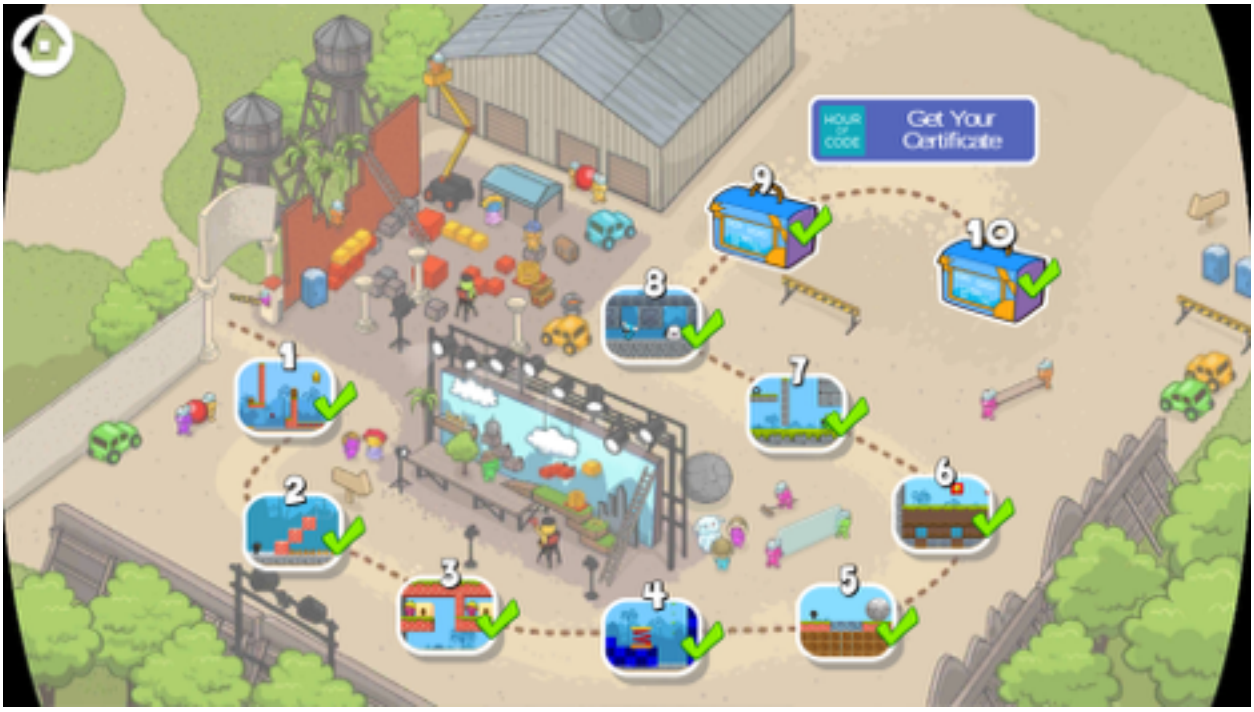
Para restablecer el progreso de un jugador, salga y vuelva a ingresar la hora del código. Debería obtener una ventana emergente que le pregunte si desea borrar su progreso.



Resumen

Ciencias Computacionales para Prescolar y Primaria

codeSpark



¿Qué hay en el Creador del Juego?

El Creador del Juego es que los jugadores pueden crear sus propios niveles de juego de vídeo utilizando dos de nuestros “estuches de juego", que son la pintura-por-números planes para la creación de su propio juego. Los jugadores pueden aplicar los principios de codificación que aprendieron en los niveles de rompecabezas para reprogramar cualquier objeto en el Creador de Juego.

Los primeros 8 niveles del Creador de Juego son tutoriales para que los estudiantes estén acostumbrados a todas las Herramientas del Creador de Juegos. Los niveles 9 y 10 son estuches de juego.

Glosario

Bucle: Un conjunto de instrucciones que es repetido una y otra vez.

Bucle infinito: Un conjunto de instrucciones que es repetido una y otra vez sin parar.

Remezcla: Aprovechar y adaptar la programación existente para crear una nueva versión.

Lección de la Estuche de Juego Haga su Propio Juego

codeSpark Academy with The Foos

codeSpark

Time:

45-60 Min

Materials:

Tabletas o computadores con *codeSpark Academy*
Comando y Parámetros Tarjetas de Baile

Objetivos de Aprendizaje:

Los estudiantes aprenderán que…

- Algunas secuencias son más eficientes, por lo tanto más deseables, que otras.
- Los bucles hacen las secuencias más eficientes.
- La eficiencia es muy importante porque las computadoras no tienen poder de procesamiento ilimitado.
- La diferencia entre un bucle y un bucle sin fin o infinito.

Vocabulary:

Bucle: Un conjunto de instrucciones que es repetido una y otra vez.

Bucle infinito: Un conjunto de instrucciones que es repetido una y otra vez sin parar.

Remezcla: Aprovechar y adaptar la programación existente para crear una nueva versión.

Lección de la Estuche de Juego
Haga su Propio Juego

codeSpark Academy with The Foos

codeSpark

Introduction:

Si tiene un proyector o Smartboard, abra el codeSpark Academy y el nivel de **Foo Constructor numero** 10 para toda la clase. Aquí es donde los bucles se introducen por primera vez. Señale el símbolo del bucle en la esquina inferior derecha y juegue a través del nivel para mostrar a los estudiantes cómo funcionan los bucles. Pida a los estudiantes que le ayuden a determinar cuántas veces desea que Foo Constructor repita una acción. Continúe jugando a través de los niveles de Foo Constructor y muestre a los estudiantes lo que sucede si el comando de bucle está configurado a muy pocas o demasiadas repeticiones.

Pregunte a los estudiantes, "¿qué pasaría si quisieran que una acción continuara sin cesar?"

Introduzca la idea de bucles sin fin y proporcione algunos ejemplos, como la tierra girando alrededor del sol, el tiempo y la electricidad. Dibuje el símbolo de infinito ∞ en el tablero, que se usa en codeSpark Academy para representar bucles sin fin.

Enseñar a los estudiantes "Esta es la Canción que Nunca Termina" como un ejemplo de un bucle sin fin.

> *"Esta es la canción que no terminamos.*
> *Sigue y sigue adelante mis amigos.*
> *Alguien comenzó a cantarlo sin saber qué es,*
> *Y seguirán cantándolo para siempre sólo porque."*
>
> *(Repita una y otra vez)*

Lección de la Estuche de Juego
Haga su Propio Juego

codeSpark Academy with The Foos

codeSpark

Actividades de Juego:

Haga que los estudiantes jueguen en los primeros nueve niveles de la sección "Hacer Juegos". Estos niveles tutoriales ayudarán a los estudiantes a familiarizarse con los diferentes componentes del diseño del juego y cómo usarlos y adaptarlos para crear sus propios juegos (descripción proporcionada a continuación).

Descripción General de los Niveles de Tutorial 1-8:

Nivel 1: Aprende a jugar a través de un juego
Nivel 2: Aprenda cómo agregar paisajes (por ejemplo, ladrillos)
Nivel 3: Aprende a hacer que el personaje "camine hacia adelante"
Nivel 4: Aprende a hacer saltar a los personajes
Nivel 5: Aprenda a borrar los componentes del juego
Nivel 6: Aprende a explotar objetos y paisajes
Nivel 7: Aprende a hacer crecer personajes y objetos
Nivel 8: Aprenda a usar bucles para repetir las acciones

Después de que los estudiantes jueguen a través de los nueve niveles tutoriales, dos cofres del tesoro (Niveles 9 y 10) se desbloquearán. Ambos niveles comienzan con un breve video que muestra una visión general de cómo será el juego una vez que se haga, seguido por instrucciones interactivas sobre cómo programar un mecánico de juegos en particular utilizando diferentes conceptos de informática.

Pida a los estudiantes que trabajen individualmente o en pareja para trabajar en ambos niveles. Ambos niveles proporcionan "estuches de juego" que describen los diferentes componentes del juego que necesitan agregar para hacer el juego. Si los estudiantes trabajan en parejas, asegúrese de que se turnen para crear los componentes del juego y para jugar el juego. Esto podría incluir a los estudiantes que trabajan juntos en ambos componentes o un estudiante asume el papel de "programador" para crear el juego y el otro lo juega y luego cambian de oficios.

Lección de la Estuche de Juego
Haga su Propio Juego

codeSpark Academy with The Foos

codeSpark

Después de que los estudiantes terminen de hacer sus juegos de acuerdo con los planos, desafíelos para ajustar los diferentes componentes del juego y hacerlos suyos. A continuación se presentan algunos ejemplos de desafíos.

Ejemplos Desafíos

- ¿Qué sucede si cambia el símbolo de infinito a un número cuando usa bucles?
- ¿Cómo puede hacer que su juego sea más desafiante? ¡Intente agregar la mecánica diferente del juego para ver qué sucede!
- Si trabaja en pares programador/jugador, aliente a los estudiantes a remexclar el juego de sus parejas cambiando ciertos elementos (por ejemplo, donde se colocan los personajes enemigos, la dirección a la que saltan los personajes, añadiendo elementos adicionales que el Foo debe evitar).

Actividades Desconectadas:

Bulces de Baile

1. Divida a los estudiantes en parejas y proporcione a cada pareja las Tarjetas de Baile de Comando y Parámetros.

2. En parejas, haga que los estudiantes usen las tarjetas para crear sus propias bailes. Hay dos reglas para la baile:
   - La baile debe estar contenida dentro de un bulce sin fin.
   - Debe incluir al menos un comando y un parámetro.

3. Haga que los parejas escriban la secuencia de acciones y los parámetros que componen su baile.

4. Pida a cada pareja que presente su baile a toda la clase (¡actúela!) y pida a los otros estudiantes que identifiquen cuáles fueron los comandos y parámetros del baile.

Comando y Parámetros Tarjetas de Baile disponibles en la parte de atrás del libro.

Discusión Después:

- ¿Cuáles son los beneficios de usar bucles? (Pista: son más eficientes)
- ¿Cuáles son algunas situaciones de la vida real en las que los bucles sin fin serían más beneficiosos que los bucles regulares? ¿Cuáles son las situaciones en las que los bucles regulares pueden ser mejores?
- Ejemplos: cintas de correr, escaleras mecánicas, señales de giro, ciclo del agua

Aviso Profesional: Depuración - A menudo en informática, encontramos errores que hacen que nuestros programas hagan las cosas incorrectamente. Al crear su baile, los estudiantes pueden haber cometido errores en su código. Recuerde a los estudiantes que cometer errores es parte del proceso, y podemos aprender de cada error.

Actividades Desconectadas

Bucle Sin Fin Actividad de Baile

Tarjetas de Comandos y Parámetros

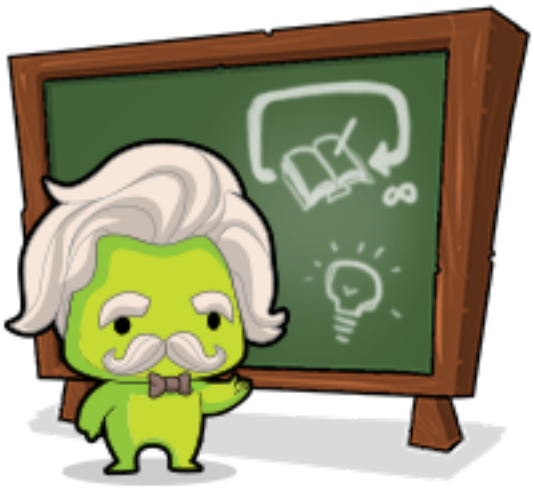
Las tarjetas grises son Comandos mientras que las tarjetas blancas son Parámetros.

DAR PALMADAS	DAR UN VUELTA	UNA VEZ	DOS VECES
CHASEQUEE LOS DEDOS	AGITAR LA MANO	TRES VECES	DERECHO
BRINCAR	SACUDIR LAS CADERAS	IZQUIERDA	ARRIBA
SALTAR CON UN PIE	MENEAR	ABAJO	RÁPIDO
ANADEAR	GOLPEAR EL PIE	LENTO	QUATRO VECES

Escala para la Evaluación de los Estudiantes

	Insatisfactorio	Competente	Dominio	Excelente
Conceptos	Niveles de rompecabezas no se completan.	Niveles de rompecabezas se completan con 1 estrella.	Niveles de rompecabezas se completan con 2 estrellas.	Niveles de rompecabezas se completan con 3 estrellas.
Ejecución	Código no funciona o tiene fallas importantes que impiden que funcione correctamente.	El código casi funciona, o tiene defectos menores.	El código funciona de la manera que el estudiante la intencióno pero no es el más eficiente.	El programa es funcional y refinado y se ejecuta de la manera más eficiente posible.
Comprensión de materiales	El estudiante no puede describir cómo debería funcionar su código y no es consciente de su proceso.	El estudiante puede describir cómo debería funcionar su código y comprende algo de su contenido.	El estudiante puede describir cómo debe funcionar su código y soluciona problemas, logrando sus resultados deseados.	El estudiante puede describir cómo funciona su código, cómo ellos lo escribieron y ayudar a otros a solucionar problemas de su código.
Esfuerzo	El estudiante muestra un mínimo esfuerzo, no utiliza la hora de clase efectivamente, y el trabajo es incompleto. El estudiante se niega a explorar más de una idea.	El estudiante hace lo suficiente para cumplir los requisitos mínimos. El estudiante tiene más de una idea pero no la persigue.	Completó el trabajo de una manera promedio, aunque podrían haber hecho más. El estudiante explora múltiples soluciones.	Completó el trabajo y superó las expectativas del profesor. El estudiante muestra disposición para explorar múltiples ideas y soluciones, y formula preguntas

Escala adaptada de: http://www.edutopia.org/pdfs/blogs/edutopia-yokana-maker-rubric.pdf



DESCARGA EL PROGRAMA COMPLETO DE 10 LECCIONES EN THEFOOS.COM/HOUROFCODE

Traducción por Sandra Cantu