

| Tema | COMPLEJIDAD Y AUTOMATIZACIÓN DEL JUEGO | |
|---------------------------------|---|--|
| Descripción de la clase | El alumno aprenderá a introducir diferentes tipos de complejidad en el juego a medida que aumenta la puntuación. El alumno también aprenderá cómo automatizar el juego usando IA (Inteligencia Artificial). | |
| Clase | PRO-C7 | |
| Duración de la clase | 50 minutos | |
| Objetivo | <ul style="list-style-type: none"> • Obtener más información sobre los diferentes operadores lógicos en JavaScript. • Aumentar la velocidad del sprite de pelota a medida que aumenta la puntuación. • Hacer que los ladrillos se muevan hacia abajo, hacia la paleta. • Asignar IA a la paleta para que pueda jugar el juego por su cuenta. | |
| Recursos requeridos | <ul style="list-style-type: none"> • Recursos para maestras: <ul style="list-style-type: none"> ○ Iniciar sesión en Code.org. ○ Laptop con conectividad a internet. ○ Auriculares con micrófono. ○ Cuaderno y bolígrafo. • Recursos para alumnos: <ul style="list-style-type: none"> ○ Iniciar sesión en Code.org ○ Laptop con conectividad a internet ○ Auriculares con micrófono ○ Cuaderno y bolígrafo | |
| Estructura de la clase | Diapositivas de Rompiendo el Hielo Actividad dirigida por la maestra Actividad dirigida por el alumno Conclusión | 10 minutos 15 minutos 15 minutos 10 minutos |
| ROMPIENDO EL HIELO - 10 minutos | | |



La maestra inicia la presentación de diapositivas desde la diapositivas 1 a la 11

Consulta las notas en tu pantalla y sigue las instrucciones de cada diapositiva.

Detalles de la actividad

¿Cómo has estado? ¿Estás emocionado de aprender algo nuevo?

Ejecuta la presentación desde la diapositiva 1 a la diapositiva 3.

Sesión de PyR

Pregunta

Respuesta

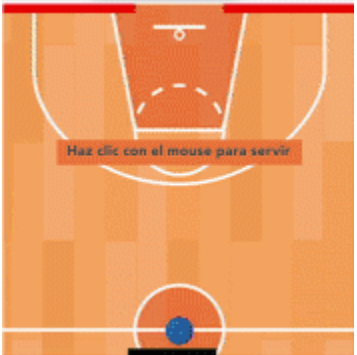
Selecciona el bloque de código que movería el balón y establecería gameState - *estado del juego*, en PLAY cuando el jugador haga clic en el mouse.




```
function mousePressed() {
  ball.velocityX = 4;
  ball.velocityY = -3;
  gameState = "play";
}
```

A.

A

| | |
|---|---|
| <p>A. <pre>function mousePressed() { ball.velocityX = 0; ball.velocityY = 0; gameState = "play"; }</pre></p> <p>B. <pre>function mousePressed() { ball.velocityX = 4; ball.velocityY = -3; }</pre></p> <p>C. <pre>function pressed() { ball.velocityX = 4; ball.velocityY = -3; gameState = "play"; }</pre></p> | |
| <p>¿Cuál de las siguientes instrucciones se utiliza para mover la paleta del jugador de manera horizontal con el mouse?</p>  <p>A. <code>player_paddle = mouseX;</code> B. <code>player_paddle.move = World.mouseX;</code> C. <code>player_paddle.y = World.mouseY;</code> D. <code>player_paddle.x = World.mouseX;</code></p> | <p>D</p> |
| <p>Continúa con la sesión de Rompiendo el hielo</p> | |
| <p>Detalles de la actividad</p> <p>Ejecuta la presentación desde la diapositiva 4 a la</p> | <p>Solución/Lineamientos</p> <p>Narra la historia usando</p> |

| | |
|---|---|
| <p>diapositiva 11 para establecer el planteamiento del problema.</p> <p>Los siguientes son los entregables de la sesión de Rompiendo el Hielo:</p> <ul style="list-style-type: none"> • Acerca de los operadores lógicos. • Asignar inteligencia artificial a la paleta. | <p>gestos con las manos y métodos de modulación de voz, para atraer más interés de los alumnos.</p> |
| <p style="text-align: center;">La maestra finaliza la presentación de diapositivas</p>  | |
| <p style="text-align: center;">ACTIVIDAD DIRIGIDA POR LA MAESTRA 1 (10 - 15 minutos)</p> | |
| <p style="text-align: center;">La maestra comienza a compartir su pantalla</p> | |
| <p style="text-align: center;"><u>DESAFÍO</u></p> <ul style="list-style-type: none"> • Implementar operadores lógicos para diferentes casos de uso. • Incrementar la velocidad del sprite de la pelota en cada golpe con el ladrillo. | |
| <p style="text-align: center;">Actividad dirigida por la maestra (20 minutos)</p> | |
| <p style="text-align: center;">Acción de la maestra</p> | <p style="text-align: center;">Acción del alumno</p> |
| <p>Hola, <nombre_del_alumno>, ¿Cómo estás?</p> <p>En la última clase completamos el Juego de Escape. ¿Lo jugaste después de la clase?</p> <p>¡Estupendo! Hoy agregaremos algunas complejidades al juego y también asignaremos inteligencia artificial a la paleta para que pueda jugar el juego por sí sola. ¿Estás emocionado?</p> | <p>REA: Variada.</p> <p>REA: Sí / No.</p> <p>REA: sí</p> |
| <p>Pero antes de presentar la IA, aprendamos sobre los operadores lógicos.</p> <p>¿Recuerdas que en la última clase verificamos si el grupo de sprites 'bricks' (<i>ladrillos</i>) estaba vacío o no?</p> <p>¿Cómo lo hicimos?</p> | <p>REA: Sí.</p> <p>REA: Verificamos si el</p> |

| <p>¡Correcto! Habíamos usado un operador '!' (Not) para el mismo:</p> <pre><< if (! Bricks [0]) >></pre> <p>! es un operador not lógico. Devuelve true (<i>verdadero</i>) si el operando es false (<i>falso</i>) y viceversa. Por ejemplo, !false = true &! true = false. <i>!falso = verdadero &! verdadero = falso</i></p> <p>Por eso, bricks[0] será verdadero cuando bricks[0] sea falso o esté vacío.</p> | <p>grupo bricks no estaba vacío.</p> | | | | | | |
|---|--------------------------------------|----------|--------|------------------|---------------|---------------|------------------|
| <p style="text-align: center;">OPERADOR LÓGICO NOT</p> <table border="1" style="margin: auto;"> <thead> <tr> <th>Operando</th><th>NOT(!)</th></tr> </thead> <tbody> <tr> <td>TRUE (verdadero)</td><td>FALSE (falso)</td></tr> <tr> <td>FALSE (falso)</td><td>TRUE (verdadero)</td></tr> </tbody> </table> | | Operando | NOT(!) | TRUE (verdadero) | FALSE (falso) | FALSE (falso) | TRUE (verdadero) |
| Operando | NOT(!) | | | | | | |
| TRUE (verdadero) | FALSE (falso) | | | | | | |
| FALSE (falso) | TRUE (verdadero) | | | | | | |
| <p>Hoy aprenderemos algunos operadores lógicos más, como:</p> <p>1) Operador &&, que devuelve verdadero solo si ambos operandos son verdaderos; de lo contrario, devuelve falso. Como sugiere el nombre 'and' (y) eso significa que ambos operandos deben ser verdaderos.</p> <p>Por ejemplo, necesito un pastel y un juguete, lo que significa que necesito ambos.</p> <p>2) Operador , que devuelve falso sólo si ambos operandos son falsos; de lo contrario, devuelve verdadero si cualquiera de los operandos es verdadero.</p> <p>Por ejemplo, necesito un pastel o un auto. Estaré feliz si consigo alguno de ellos o ambos. Estaré triste, solo si no consigo ninguno de ellos.</p> | | | | | | | |
| <p style="text-align: center;">LÓGICO Y</p> | | | | | | | |

FAL SE (fal so) **FAL SE (fal so)** **FAL SE (fal so)**

¿Dónde podemos utilizar estos operadores?

Generalmente se utilizan en las declaraciones if (si) para escribir condiciones complejas. Por ejemplo, para verificar el año bisiesto, necesitamos ver si el mes es 'Feb' y el número de días es '29', entonces es un año bisiesto.

Hoy escribiremos algunos programas para practicar estos operadores lógicos, pero antes quiero presentarte uno de los operadores matemáticos más importantes utilizados en programación, es decir, el operador **módulo**, representado por un símbolo de porcentaje (%).

El operador mod '%' devuelve el residuo cuando el número **1** se divide por el número **2**. Por ejemplo, cuando **5** se divide entre **2** el residuo será **1**.

¿Puedes decirme dónde lo encontraré en la Caja de herramientas?

¿Puedes encontrar el símbolo % en tu teclado?

.

Digamos que, si tenemos que escribir un programa para comprobar si el número es divisible por 2 y además el

REA: Variada.

REA: La pestaña Math.

REA: En el número 5 (Mayús + 5)

número debe ser mayor que 60... ¿Cuáles serán las condiciones y el operador lógico para este planteamiento del problema?

REA: Condición para ser divisible por 2: $x \% 2 == 0$
Condición para números mayores de 60: $x > 60$
Operador: AND

La maestra hace clic en el enlace de la [Actividad de la maestra 1](#) y luego en remix. La maestra escribe el código para definir una nueva función llamada 'check()' – verificar, para el planteamiento del problema mencionado anteriormente.

Mostrar el cuadro de herramientas
Espacio de trabajo:

```

1
2 function check(num1) {
3
4     if(num1%2==0 && num1>60)
5     |     console.log("verdadero");
6     else
7     |     console.log("falso");
8
9 }
10
11 check(32);
12 check(62);
13 check(65);
14

```

Comandos para depurar

Interrumpir

Step over

Consola

"false"
"verdadero"
"false"

¡Estupendo! Nuestro programa está funcionando bien. Solo la entrada '62' satisface ambas condiciones. '32' falla en la segunda condición y es menor que 60, mientras que '65' falla en la primera condición y no es divisible por 2.


¿Probemos uno más? ¿Me ayudarías?

REA: sí

| | |
|--|--|
| <p>Escribe un programa para verificar si el número es divisible por 2 o si es menor que 30.</p> <p>¿Cuáles serán las dos condiciones y el operador lógico en este caso?</p> <p><i>La maestra edita la función 'check()' para este planteamiento del problema, y cambia la entrada a 15, 32 y 65. La maestra ejecuta el código.</i></p> | <p>REA: Condición1: $x \% 2 == 0$; Condición 2: $x < 30$; Operador: OR</p> |
|  | |
| <p>Genial, ahora que conoces los operadores lógicos, volvamos al juego.</p> <p>Hoy intentaremos aumentar la complejidad del juego.</p> <p>¿Qué podemos hacer para aumentar la complejidad del juego?</p> | <p>REA: Variada.</p> |

| | |
|---|---|
| <p>¿Podemos aumentar la velocidad de la pelota a medida que se destruyen los ladrillos?</p> <p>¿Cómo podemos aumentar la velocidad de la pelota?</p> <p><i>La maestra hace clic en el enlace de la Actividad de la maestra 2 y hace clic en remix. La maestra escribe el código para aumentar la velocidad de la pelota en cada golpe de ladrillo.</i></p> | <p>REA: Sí.</p> <p>REA: Aumentando la velocidad de la pelota.</p> |
| <pre>function brickHit(ball, brick) { playSound("sound://category_hits/puzzle_game_button_04.mp3") brick.remove(); score = score+5; ball.velocityX *= 1.05; ball.velocityY *= 1.05; }</pre> | |
| <p>¿Notaste que, si la velocidad del sprite de pelota se vuelve demasiado rápida, el juego no funciona correctamente, especialmente la función bounceOff() (rebotar en)?</p> <p>Entonces, necesitamos decidir la velocidad máxima de la pelota en ambas direcciones.</p> <p>Según mi experiencia, una velocidad de 12 sería lo suficientemente buena para la complejidad.</p> <p><i>Nota: un valor de velocidad más alto (mayor que 12) puede crear algunos problemas en el juego. Si es necesario, deja que el alumno intente ejecutar el juego a una velocidad de 12, para asegurarse de que no haya ningún problema.</i></p> <p>¿Puedes decirme la condición para verificar si la velocityY es menor que 12? Entonces solo debemos aumentar la velocidad de la pelota, de lo contrario no aumentaremos</p> | <p>REA: sí</p> <p>REA: (Ball.velocityY <12)</p> |

| | |
|--|---------------------------------------|
| más la velocidad | |
| <pre>function brickHit(ball, brick) { playSound("sound://category_hits/puzzle_game_button_04.mp3") brick.remove(); score = score+5; if(ball.velocityY<12) { ball.velocityX *= 1.05; ball.velocityY *= 1.05; } }</pre> | |
| <p>¿Notaste que la velocidad no aumenta en caso de bajar, pero la velocidad sigue aumentando en la dirección ascendente?</p> <p>Porque no limitamos los valores negativos de velocityY.</p> <p>Necesitamos escribir una condición más para comprobar que velocityY no debe bajar también de -12.</p> <p>¿Puedes decirme qué operador lógico debería usarse entre dos condiciones?</p> <p><i>La maestra ayuda al niño a escribir la condición, para verificar si la velocidad es > -12 y la velocidad < 12, entonces solo así aumenta la velocidad.</i></p> | <p>REA: sí</p> <p>REA: &&</p> |
| <pre>function brickHit(ball, brick) { playSound("sound://category_hits/puzzle_game_button_04.mp3") brick.remove(); score = score+5; if(ball.velocityY > -12 && ball.velocityY < 12) { ball.velocityX *= 1.05; ball.velocityY *= 1.05; } }</pre> | |
| ¡Estupendo! Hemos aumentado la complejidad del juego a | |

| | |
|--|--|
| <p>medida que aumenta la puntuación.</p> <p>Ahora es tu turno. Comparte tu pantalla conmigo.</p> | |
| <p>La maestra deja de compartir su pantalla</p> | |
| <p>ACTIVIDAD DIRIGIDA POR EL ALUMNO 10 - 20 minutos</p> | |
| <ul style="list-style-type: none"> • Pídele al alumno que presione la tecla ESC para volver al panel. • Guía al alumno para que comience a compartir la pantalla. • La maestra cambia a pantalla completa. | |
| <p>DESAFÍO</p> <ul style="list-style-type: none"> • El alumno logra que los ladrillos se muevan hacia abajo. • El alumno agrega IA a la paleta para moverse automáticamente. | |
| <div>  <p>La maestra inicia la presentación de diapositivas desde la diapositiva 12 a la 15.</p> <p>Consulta las notas en tu pantalla y sigue las instrucciones de cada diapositiva.</p> </div> | |
| <p>Guía al alumno para que abra su propio código actualizado del juego o haga clic en el enlace de la Actividad del alumno 1 y haga clic en remix.</p> <p>¿Puedes decirme más formas de aumentar la complejidad de la pelota?</p> <p>¿Quieres intentar que los ladrillos se muevan hacia abajo?</p> <p>Vamos a intentarlo. Es tu turno ahora.</p> <p>Dado que los ladrillos son un objeto de sprites. ¿Puedes mirar en la lista de métodos de la pestaña Group (<i>grupo</i>) y buscar algo para dar velocityY al grupo?</p> <p><i>La maestra ayuda al niño a escribir un código para asignar velocidad al grupo bricks. También explica que los ladrillos deben comenzar a moverse solo después de que el juego esté en estado play (jugar). Es decir, una vez que se hace clic con el mouse para iniciar la pelota y el juego.</i></p> | <p>REA: Variado.</p> <p>REA: Si.</p> <p>REA: setVelocityYEach()</p> |

```
function mousePressed()
{
  if(gamestate == "start")
  {
    gamestate = "play";
    ball.velocityY = -7;
    ball.velocityX = 7;
    bricks.setVelocityYEach(0.2);
  }
}
```

¡Eso fue genial!

Nota para la maestra: ya que hemos agregado demasiada velocidad a la pelota, y ahora que los ladrillos también han comenzado a moverse, la velocidad de la pelota aumentará significativamente al rebotar en los ladrillos. Si aumenta aún más la velocidad de los ladrillos a 0.3 o más, la pelota podría pegarse a la paleta, por lo que restringe la velocidad de los ladrillos a 0.2 para un juego más suave.

¡Ahora es el momento de asignar IA a nuestra paleta!

¡Hagámoslo!

¿Puedes decirme, si la computadora tiene que mover la paleta, a qué debería referirse la computadora?

¡Correcto! La paleta debe seguir la posición x de la pelota, para vencer al jugador y golpear la pelota.

La maestra ayuda al niño a programar para agregar automatización al juego.

REA: a la posición 'x' de la pelota.

```
function gameplay()
{
  //paddle.x = World.mouseX;
  paddle.x = ball.x; //automatizado
  if(paddle.x < 60)
  {
    paddle.x = 60;
  }

  if(paddle.x > 340)
  {
    paddle.x = 340;
  }
  drawSprites();
}
```

¡Magnífico! ¿Observaste que ahora la computadora puede jugar el juego por sí sola?

Aunque las computadoras tienen un coeficiente intelectual cero, con algo de inteligencia proporcionada por los programadores/codificadores, pueden hacer maravillas.

¿Cómo te sientes?

REA: Variado.

Guía al alumno para que deje de compartir su pantalla

• CONCLUSIÓN - 5 - 10 minutos

RETROALIMENTACIÓN

- Aprecia y felicita al alumno por intentar aprender un concepto difícil.
- Pregúntale cómo se siente después de la sesión.
- Revisa y comprueba su comprensión.



La maestra inicia la presentación de diapositivas desde la diapositiva 16 a la 24

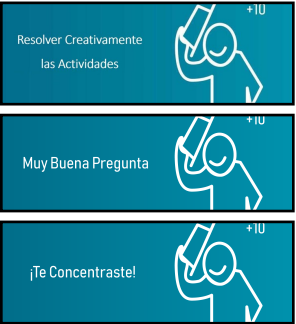
Consulta las notas en tu pantalla y sigue las instrucciones de cada diapositiva.

Detalles de la actividad

Solución/Lineamientos

Ejecuta la presentación desde la diapositiva 16 a la

| <p>diapositiva 24.</p> <p>Los siguientes son los entregables de la sesión de Rompiendo el hielo:</p> <ul style="list-style-type: none"> • Explica los hechos y las trivias. • El desafío de la próxima clase. • El proyecto del día. • La actividad adicional. | <p><i>Guía al alumno a desarrollar el proyecto y compártelo con nosotros.</i></p> |
|---|---|
| <p>Sesión de PyR</p> | |
| Pregunta | Respuesta |
| <p>¿Qué operador devuelve verdadero si el operando es falso y viceversa?</p> <p>A. OR - O B. AND - Y C. NOT - NO D. NO - NO</p> | <p>A</p> |
| <p>¿Qué hace el código resaltado en el siguiente fragmento del código?</p> <pre> function brickHit(ball, brick) { playSound("sound://category_hits/puzzle_game_button_04.mp3") brick.remove(); score = score+5; ball.velocityX *= 1.05; ball.velocityY *= 1.05; } </pre> <p>A. Aumenta la velocidad de los ladrillos B. Destruye el ladrillo C. Reproduce sonido D. Aumenta la velocidad de la pelota</p> | <p>D</p> |
| <p>¿Cuál de los siguientes comandos nos ayuda a automatizar la paleta?</p> | <p>A</p> |

| | |
|---|--|
| <p>A. paddle.x= ball.x B. ball.x = paddle.x C. paddle.y = ball.y D. paddle.velocityX = 5</p> | |
| <p>Felicidades por tu excelente trabajo.</p> <p>¡Impresionante!</p> <p>La próxima clase es una clase Capstone. Has estado aprendiendo nuevos conceptos en cada clase. Ahora es el momento de reducir un poco la velocidad y revisar todos los conceptos que hemos cubierto en las últimas 7 clases. Estarás creando un juego completamente nuevo por ti mismo en la clase Capstone.</p> <p>¿Estás emocionado?</p> | <p><i>Asegúrate de dar al menos 2 felicitaciones durante la clase por:</i></p> <div data-bbox="1015 554 1308 875">  </div> <p>REA: Sí</p> |
| <p>Descripción del proyecto</p> <p>PING PONG INTELIGENTE</p> <p>Objetivo del proyecto: En la clase 7 has aprendido cómo aumentar la complejidad del juego y aplicar inteligencia en el juego, para jugarlo automáticamente.</p> <p>En este proyecto tendrás que practicar y aplicar lo que has</p> | <p><i>El alumno interactúa con la maestra durante el proyecto.</i></p> |

| | |
|--|--|
| <p>aprendido hasta ahora y habilitar a un jugador de computadora inteligente en el juego como oponente.</p> <p>Historia: A Dodo le encanta jugar Ping Pong y ha dominado el juego. Le gusta que lo desafíen en el juego y, por lo tanto, decide cambiar el juego y hacerlo un poco más desafiante. También Dodo quiere habilitar la IA en el movimiento de la paleta de la computadora para que siempre tenga un fuerte oponente.</p> <p>¿Puedes ayudar a Dodo a habilitar la IA y subir de nivel las complejidades del juego?</p> <p>Estoy muy emocionada de ver la solución de tu proyecto y se que lo harás realmente bien.</p> <p>¡Adiós!</p> | |
| <p>La maestra finaliza la presentación de diapositivas</p> | |
| <p>La maestra hace clic en</p> | |



✕ Finalizar Clase

| Actividad | Nombre de la actividad | Enlaces |
|---------------------------|------------------------|---|
| Actividad de la maestra 1 | Actividad en blanco | https://studio.code.org/projects/gamelab/NUENU Nje68AbpKJcYNQ9YBN8WQnJUDzDOjBnGdTu hHM |
| Actividad de la | Operador AND | https://studio.code.org/projects/gamelab/iVh73BJ |

| | | |
|---|----------------------------------|---|
| maestra 1.1 Código de referencia | | bKLGHxo9ZcPh8htkC0ekbwMs6kU7tCZXYEWA |
| Actividad de la maestra 1.2 Código de referencia | Operador OR | https://studio.code.org/projects/gamelab/iVh73BJbKLGHxo9ZcPh8hsJ9tSPgdFT_1Kd9ENgJN54 |
| Actividad de la maestra 2 | Juego de Escape 1.6 | https://studio.code.org/projects/gamelab/iVh73BJbKLGHxo9ZcPh8hkBc6wUbtOeGovOT-f4iz9Q |
| Código de referencia de la actividad 2 de la maestra | Incremento de complejidad | https://studio.code.org/projects/gamelab/iVh73BJbKLGHxo9ZcPh8hvffEg8U23QiY3Pu_6bKmcA |
| Actividad del alumno 1 | IA en el Juego de Escape | https://studio.code.org/projects/gamelab/iVh73BJbKLGHxo9ZcPh8hvmkj65rm_S0Gkw1u4BgXnk |
| Referencia del código de la maestra (Juego de Escape 1.7) | Juego de Escape 1.7 | https://studio.code.org/projects/gamelab/ZCNJxY1WTGmnott5eEkVOPpp9IYR4tndIGOB4VKoA18 |
| Solución del proyecto | PING PONG INTELIGENTE | https://studio.code.org/projects/gamelab/wbXZ8KdA3cvMC_Q44c-moqOQUC78N50O1pnY1NuGkno |
| Enlace de apoyo visual de referencia de la maestra | Enlace de ayuda visual con notas | https://s3-whjr-curriculum-uploads.whjr.online/45efed18-f7fd-4ba3-9067-35a59ff888d4.html |
| Enlace de apoyo visual de referencia de la maestra | Enlace de ayuda visual sin notas | https://s3-whjr-curriculum-uploads.whjr.online/866d31da-412a-491e-922f-3e44bfcc77c9.html |
| Referencia del quiz en clase de la maestra | Quiz en clase | https://s3-whjr-curriculum-uploads.whjr.online/64c9a922-69e8-470b-a202-5da6510e2aca.pdf |

