



Girls Who Code en casa

Juego del Meteor Catcher: Parte 1

Guía de referencia

Juego del Meteor Catcher: Parte 1 - Guía de referencia



En este documento encontrarás todas las respuestas a algunas de las preguntas de la actividad. Sigue la actividad y cuando veas este ícono, detente y revisa tus conocimientos aquí.

Paso 1: Identificar las partes de un juego

Las partes de un juego de Meteor Catcher

- Describir el **objetivo** del Meteor Catcher, el juego que acabas de jugar en el último paso. ¿Qué tiene que hacer un jugador o equipo para ganar el juego?

El objetivo del Meteor Catcher es atrapar tantos meteoritos como sea posible. En este momento, este juego no tiene un objetivo muy definido. Lo dejamos abierto para que puedas personalizar el tuyo después de construir el juego base.

- Los **componentes** del Meteor Catcher incluyen el meteorito, el capturador, las paredes y el jugador. Cada componente tiene propiedades únicas (p. ej., tamaño, color, forma, etc.) y acciones (es decir, las cosas que hace, los verbos que asocias con ese componente) que contribuyen al sistema del juego. Por ejemplo, una propiedad de meteorito sería redonda y una acción de meteorito incluiría caer desde la parte superior de la pantalla a la parte inferior.

Dedica **2-3 minutos** a pensar en las propiedades y acciones de cada componente de la tabla a continuación.

COMPONENTE	PROPIEDADES	ACCIONES
<i>¿Cuáles son las piezas esenciales para jugar?</i>	<i>¿Cuáles son los atributos o las características del componente?</i>	<i>¿Qué hace? ¿Qué verbos asocias con él?</i>
meteorito	<ul style="list-style-type: none">→ Redondo→ Verde azulado→ Diámetro aleatorio entre 10 y 40 píxeles	<ul style="list-style-type: none">→ Cae desde la parte superior de la pantalla hasta la parte inferior→ Se mueve a una velocidad aleatoria→ Comienza en diferentes ubicaciones en la parte superior de la pantalla→ Puede intersectarse con el capturador→ Puede intersectarse con el suelo (es decir, en la parte inferior de la pantalla)
capturador	<ul style="list-style-type: none">→ Redondo→ Blanco transparente→ 40 píxeles de diámetro	<ul style="list-style-type: none">→ Sigue al ratón.→ Puede “capturar” o recolectar meteoritos intersectándolos.
paredes	<ul style="list-style-type: none">→ 400 píxeles de ancho→ 400 píxeles de altura	<ul style="list-style-type: none">→ Se intersecta con meteoritos.→ La pared inferior hace que aparezca un nuevo meteorito si un meteorito la toca.
jugador	<ul style="list-style-type: none">→ ¡Le gusta el espacio!→ ¡Y las lluvias de meteoritos!	<ul style="list-style-type: none">→ Mueve el ratón para atrapar los meteoritos que caen.

Paso 3: Identificar las partes de un juego (cont.)

- Describir el **espacio** del juego. ¿En dónde ocurre? (Ten en cuenta que a veces el espacio puede ser más de una cosa. Por ejemplo, el ajedrez tiene lugar en el tablero de ajedrez, pero también tiene lugar en una sala de estar, parque o cafetería).

El juego se realiza en una ventana de 400 por 400 cuadrados en una página web. Desde la perspectiva del cuento, se lleva a cabo en el espacio exterior real.

- Definir el **desafío**. ¿Qué obstáculos se encuentran en el camino del jugador para alcanzar el objetivo?

El desafío es capturar los meteoritos que caen. Cada uno comienza en una ubicación diferente, cae a una velocidad diferente y se renderiza a un tamaño diferente cada vez que aparece un nuevo meteorito en la pantalla.

- Describir la mecánica **central del juego**. ¿Qué acciones o movimientos principales necesita hacer el jugador para jugar el juego? ¿Qué acciones o movimientos principales hace el jugador para impulsar el juego?

Los jugadores capturan meteoritos. Mueven el ratón por la pantalla para recogerlos.

- Escribir una lista de las **reglas del juego**. Las reglas determinan lo que podemos y no podemos hacer en nuestro juego. Pueden aplicarse a jugadores, a componentes, al espacio, etc.
 - ◆ Los meteoritos caen desde la parte superior de la pantalla hasta la parte inferior.
 - ◆ Solo cae un meteorito a la vez.
 - ◆ El capturador sigue al ratón.
 - ◆ El capturador del jugador debe intersectarse con el meteorito para capturarlo.
 - ◆ Si se captura un meteorito, desaparece y se dibuja un nuevo meteorito en la parte superior de la pantalla.
 - ◆ Si un meteorito toca la parte inferior de la pantalla, desaparece y se dibuja un nuevo meteorito en la parte superior de la pantalla.

Paso 2: Escribir el pseudocódigo para tu juego

Declarar cualquier variable

HAZLO UNA VEZ

Establecer el tamaño del lienzo en 400 píxeles por 400 píxeles

HAZLO EN CADA LOOP

Establecer el color de fondo

Dibujar el meteorito

Hacer que el meteorito caiga

Dibujar el capturador para seguir al ratón

Encontrar/calcular la distancia entre el meteorito y el capturador

Probar para ver si el meteorito y el capturador se han intersectado. Si se intersectan,

Volver a dibujar el meteorito en la parte superior de la pantalla en una ubicación aleatoria,

Darle una nueva velocidad,

Establecer un diámetro nuevo.

Probar para ver si el meteorito y la pared inferior se han intersectado. Si se intersectan,

Volver a dibujar el meteorito en la parte superior de la pantalla en una ubicación aleatoria,

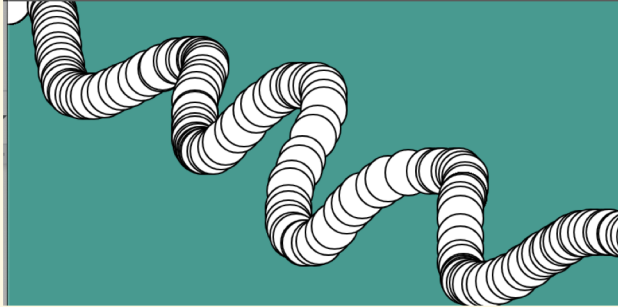
Recuerda: Hay más de una forma de escribir un programa, por lo que también hay más de una forma de escribir tu pseudocódigo.

Darle una velocidad,

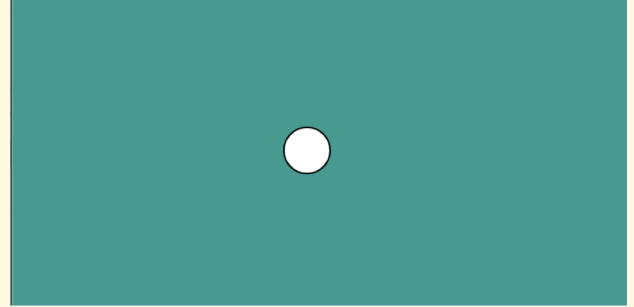
Establecer un diámetro nuevo.

Paso 5: Aprender sobre el flujo del programa

BORDADO 1



BORDADO 2



El bordado 1 tiene la función `background()` en `setup()`. Esto significa que el fondo solo se dibuja una vez. Dado que la función `ellipse()` está en `draw()`, p5 dibuja un nuevo círculo en la posición del ratón en la pantalla cada vez que el programa se ejecuta a través de un loop. Podrías describirlo de la siguiente manera: Llenar el fondo, dibujar un círculo, dibujar un círculo, dibujar un círculo, dibujar un círculo, dibujar un círculo, etc.

```
// Bordado 1
function setup() {
  createCanvas(400, 400);
  background(220);
}

function draw() {
  ellipse(mouseX, mouseY, 50, 50);
}
```

El bordado 2 tiene la función `background()` en `draw()`. Esto significa que el programa dibuja el fondo y el círculo cada vez que el programa pasa por el loop. Esto da la apariencia de que el círculo se mueve suavemente a través del espacio a medida que movemos el cursor, a pesar de que el programa está dibujando nuevos círculos como en el primer bordado. Podrías describirlo de la siguiente manera: Rellenar el fondo, dibujar un círculo, rellenar el fondo, dibujar un círculo, rellenar el fondo, dibujar un círculo, rellenar el fondo, etc.

```
// Bordado 2
function setup() {
  createCanvas(400, 400);
}

function draw() {
  background(220);
  ellipse(mouseX, mouseY, 50, 50);
}
```

Paso 6: Verificar la comprensión

Las acciones en `setup()` solo deben realizarse una vez, ya que solo estás realizando un lote. Las acciones en `draw()` deben ocurrir para cada dumpling que hagas. Ya que estás haciendo varios dumplings, colocamos estas acciones en `draw()`.

```
setup() {  
  Medir los ingredientes de relleno  
  Mezclar ingredientes de relleno  
  Recolectar los envoltorios de  
  dumplings  
}  
  
draw() {  
  Poner relleno de la cuchara en el  
  envoltorio  
  Cerrar envoltorio  
  Colocar el dumpling en la bandeja  
  Cocinar el dumpling  
  Sacar el dumpling de la bandeja  
  Comer el dumpling  
}
```