

# SilverCoders

MEJORA DE LA ALFABETIZACIÓN DIGITAL MEDIANTE MÉTODOS EFECTIVOS  
DE APRENDIZAJE PARA ADULTOS



RETO #22

## CAZADOR DE GEOMETRÍA AVANZADA

PROGRAMA DE CODIFICACIÓN  
**PARA ADULTOS**  
**+55**



SILVER CODERS

ERASMUS+ No. 2020-1-SE01-KA227-ADU-092582



Co-funded by  
the European Union

*Este documento refleja únicamente la opinión del autor y la Agencia Nacional y la  
Comisión Europea no son responsables del uso que pueda hacerse de la información que  
contiene.*

# ESTRUCTURA DEL RETO

## DESCRIPCIÓN

Se le ha proporcionado una configuración para que recuerde los elementos más importantes del entorno GDevelop: la escena y la hoja de eventos. Los eventos disponibles permiten al jugador mover al monstruo y atrapar las piezas geométricas que, ahora sí, están cayendo. Se le pedirá que mejore el juego, haciéndolo más dinámico (también aparecen bombas que pueden matar a su monstruo).

## OBJETIVO GENERAL

En este reto vamos a mejorar el juego Geometry Hunter, haciéndolo más interactivo y frenético.

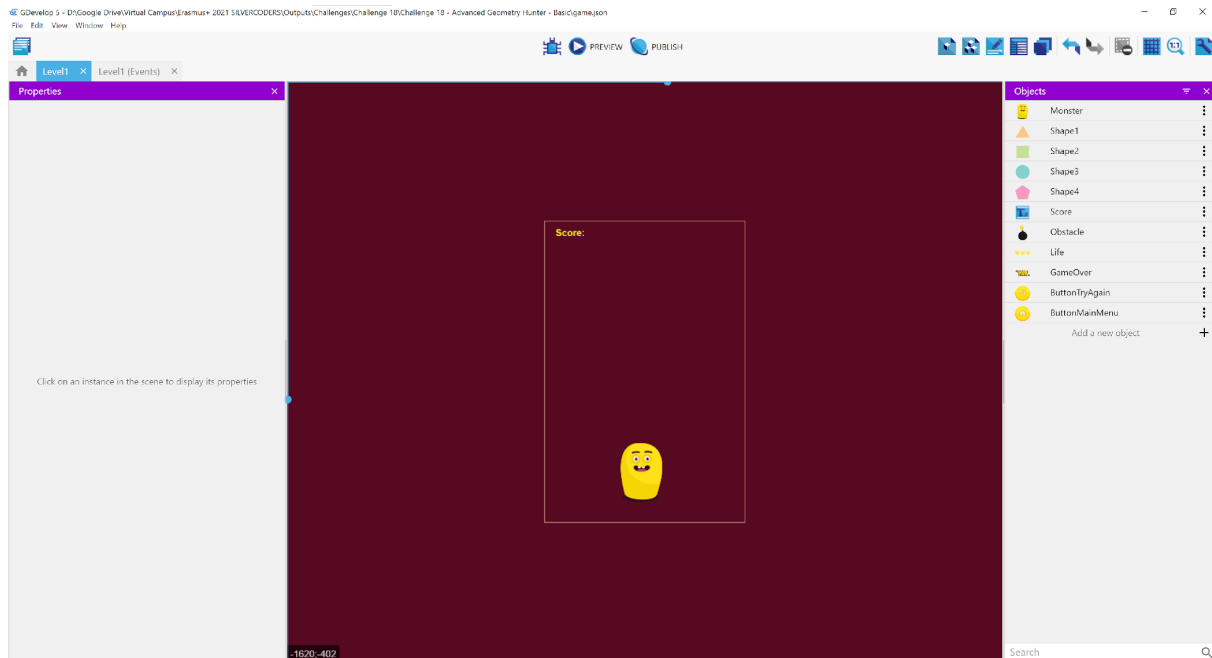
## OBJETIVOS DE APRENDIZAJE

Al final de este reto, serás capaz de...:

- Entender cómo trata el código un ordenador y cuál es el papel de un compilador.
- Estar familiarizado con el concepto de lenguajes de bajo y alto nivel y entender cuáles son sus diferencias y lo que se requiere para codificar en cualquiera de ellos.
- Tener experiencia con una suite de programación visual y ser capaz de codificar una pequeña pieza de software estándar con ella.
- Saber qué son las sentencias y las líneas de comandos y qué significan para un compilador.
- Ser capaz de escribir instrucciones utilizando una sintaxis correcta y con un mínimo de errores.
- Saber qué son los operadores, qué hacen y qué símbolos corresponden a cada operador.
- Comprender la asignación de valores a variables y cómo modificarlos.
- Conocer todas las operaciones aritméticas básicas y cómo utilizarlas.
- Reconocer y saber utilizar todas las estructuras de datos relacionadas con los números.
- Conocer las estructuras vinculadas al uso de texto, como cadenas y caracteres.
- Saber utilizar correctamente las sentencias If para ejecutar código según una determinada condición fija definida.
- Saber utilizar el editor de Gdevelop.
- Comprender los conceptos de escenas, eventos y objetos.

# INSTRUCCIONES

- Comience abriendo el editor **Gdevelop**.
- Utilice el menú Archivo para abrir el juego Desafío 22 - Básico
- Esto es lo que debería obtener



- Pulsa el botón **Vista Previa** para jugar. Puedes mover el monstruo hacia la izquierda y hacia la derecha con las flechas de tu teclado y debes atrapar las formas geométricas que ahora caen. Por cada pieza que atrapes, se sumará un punto a tu puntuación.
- Repite el juego tantas veces como quieras. Para repetir tienes que cerrar la ventana del juego y pulsar el botón **Vista Previa** en el editor.
- Ahora que ya sabes cuál es la mecánica del juego (lo que puedes hacer en el juego) vamos a ver cómo se hace. Nos centraremos en las diferencias con el último desafío.

GAME START	
<p>At the beginning of the scene</p> <p>Add condition</p>	<p>Start (or reset) the timer "ShapeCreation"</p> <p>Add action</p>

- Cuando el juego comienza creamos un Timer llamado ShapeCreation que es un objeto que siempre está contando el tiempo en segundos.

## SHAPES

⌚ The timer "ShapeCreation" is greater than 2 seconds  
Add condition

🎨 Among objects **Shapes**, create object named "Shape" : ToString(RandomInRange(1,4)) at position RandomInRange(80, 640-80); 100 (layer: )  
🔧 Change the angle of **Shapes**: set to RandomInRange(0,360)  
🔧 Change the scale of **Shapes**: set to RandomFloatInRange(0.8, 1.6)  
⌚ Start (or reset) the timer "ShapeCreation"  
Add action

- Cuando el temporizador ShapeCreation alcanza los 2 segundos se crea una nueva forma que puede ser aleatoriamente una de las cuatro formas diferentes. Para hacer el juego más divertido estas formas son escaladas y rotadas. Después reiniciamos el temporizador a 0 para empezar a contar de nuevo.

## Move shape according to the game speed

Add condition

➡ Add to **Shapes** an instant force, angle: 90 degrees and length: Variable(GameSpeed) pixels  
🔄 Rotate **Shapes** at speed 90 deg/second  
Add action

- La forma "cae" añadiéndole una fuerza vertical.
- Hagamos el juego un poco más difícil. Vamos a lanzar también algunas bombas que pueden quitar vidas al monstruo si es alcanzado. Para ello vamos a utilizar el objeto **Obstáculo**

## OBSTACLE

⌚ The timer "ObstacleCreation" is greater than 5 seconds  
Add condition

🎨 Create object **Obstacle** at position RandomInRange(80, 640-80);-100 (layer: )  
⌚ Start (or reset) the timer "ObstacleCreation"  
Add action

## Move obstacle according to the game speed

Add condition

➡ Add to **Obstacle** an instant force, angle: 90 degrees and length: 1.5\*Variable(GameSpeed) pixels  
🔄 Change the z-order of **Obstacle**: set to 4  
Add action

- El código para el objeto **Obstáculo** es similar al código para las formas. Tenemos un temporizador que lo controla y lo hacemos caer.

Repeat for each instance of Obstacle:

🔪 **Obstacle** is in collision with **Monster**  
Add condition

✖ Delete **Obstacle**  
💥 Damage **Monster**, removing 1 from its health  
🔊 Play the sound killed.wav, vol.: , loop: no  
Add action

- Pero si hay una Colisión con el Monstruo, se daña y obtiene 1 punto menos de salud

PROPERTIES
BEHAVIORS
VARIABLES
EFFECTS

Flash

Half period (time during which object is invisible), in seconds

0,1

Health

Damage cooldown (in seconds) (0 for no cooldown)

0,8

Health

3

Maximum health (0 for no maximum)

3

StayOnScreen

Bottom margin, in pixels

0

Left margin, in pixels

0

Right margin, in pixels

133

Top margin, in pixels

0

?

HELP

▶

RUN A PREVIEW

+

ADD A BEHAVIOR

CANCEL
APPLY

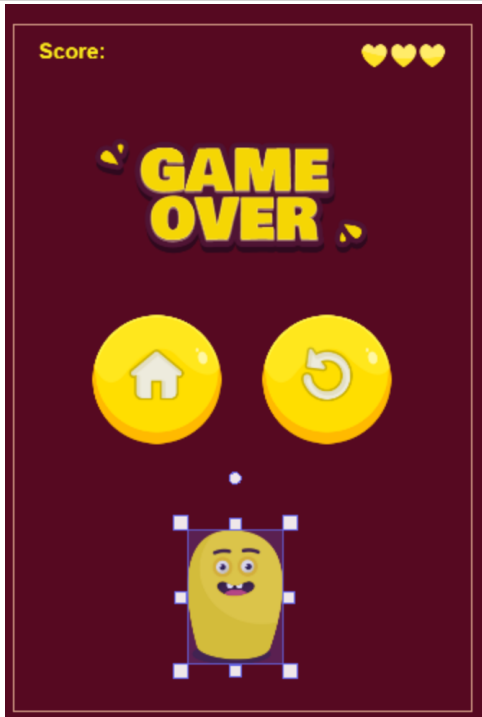
- La salud es un **Behaviour**, una propiedad estándar que podemos asociar a los objetos. La salud máxima es 3.

The screenshot shows the Piskel animation editor interface. At the top, there are four tabs: PROPERTIES, BEHAVIORS, VARIABLES, and EFFECTS. The PROPERTIES tab is active, showing the object name 'Life'. Below this, there are four animation slots, each with a title bar (e.g., 'Animation #0 Life3'), a preview icon, a duration of 0,08, and a 'Loop' checkbox. Each slot contains a 4-frame animation sequence of yellow hearts on a transparent background, with a '+ ADD' button and a small square icon. At the bottom, there are buttons for 'EDIT COLLISION MASKS', 'EDIT POINTS', 'ADVANCED OPTIONS', and a blue '+ ADD AN ANIMATION' button. A footer bar contains a help icon, 'HELP', 'RUN A PREVIEW', 'CANCEL', and 'APPLY'.

- Para representar visualmente la salud del monstruo utilizaremos el objeto Vida. Este Sprite tiene 4 cuadros, cada uno representando un estado de salud.

The screenshot shows the game engine's event editor. A yellow bar at the top is labeled 'HEALTH'. Below it, there are two sections: 'Add condition' and 'Add action'. The 'Add condition' section shows a condition: 'Monster has just been damaged'. The 'Add action' section shows two actions: 'Set animation of Life to "Life" + ToString(Monster.Health::Health())' and 'Make Monster blink for 1.5 seconds'.

- Cuando el monstruo es dañado el Sprite se mueve al siguiente frame.
- Ahora, cuando el monstruo está sin vida, el monstruo está muerto. Debemos mostrar el botón GameOver y crearemos dos botones, uno para reiniciar el juego y otro para salir del juego. Este es el aspecto de la escena inicial.



- No queremos ver estos objetos al principio así que los ocultamos.

GAME OVER	
<ul style="list-style-type: none"> <li>At the beginning of the scene</li> </ul>	<ul style="list-style-type: none"> <li>Hide  GameOver</li> <li>Hide  ButtonTryAgain</li> <li>Hide  ButtonMainMenu</li> </ul>

- Y les mostramos cuando el monstruo está muerto.

<ul style="list-style-type: none"> <li> Monster is dead</li> </ul>	<ul style="list-style-type: none"> <li>Set animation of  Life to "Life0"</li> <li>Set animation of  Monster to "MonsterDead"</li> <li>Delete  Shapes</li> <li>Delete  Obstacle</li> <li>Show  GameOver</li> <li>Show  ButtonTryAgain</li> <li>Show  ButtonMainMenu</li> </ul>
--	---

- Ahora comprobamos qué botón ha pulsado el jugador

The cursor/touch is on <b>ButtonTryAgain</b> Add condition	Set animation of <b>ButtonTryAgain</b> to "TryAgainNormal" Add action
The cursor/touch is on <b>ButtonTryAgain</b> Add condition	Set animation of <b>ButtonTryAgain</b> to "TryAgainHover" Add action
Touch or Left mouse button is down Add condition	Set animation of <b>ButtonTryAgain</b> to "TryAgainPressed" Add action
Left mouse button was released Add condition	Change to scene "Level1" Add action
The cursor/touch is on <b>ButtonMainMenu</b> Add condition	Set animation of <b>ButtonMainMenu</b> to "MainMenuNormal" Add action
The cursor/touch is on <b>ButtonMainMenu</b> Add condition	Set animation of <b>ButtonMainMenu</b> to "MainMenuHover" Add action
Touch or Left mouse button is down Add condition	Set animation of <b>ButtonMainMenu</b> to "MainMenuPressed" Add action
Left mouse button was released Add condition	Quit the game Add action

## RECURSOS

Reto 22 (Básico)