

# SilverCoders

DIGITAL LITERACY IMPROVEMENT THROUGH EFFECTIVE  
LEARNING EXPERIENCES FOR ADULTS



DESAFIO #22

## ADVANCED GEOMETRY HUNTER

**CODING TRAINING PROGRAMME  
FOR +55 ADULTS**



**SILVER CODERS**

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



**Co-funded by  
the European Union**

*This document reflects only the author's view and the National Agency and the European Commission are not responsible for any use that may be made of the information it contains*

# ESTRUTURA DO DESAFIO

## DESCRIÇÃO

Foi-lhe fornecida uma configuração que se destina a recordar os elementos mais importantes do ambiente Gdevelop: a cena e a folha de eventos. Os eventos disponíveis permitem ao jogador mover o monstro e apanhar as peças geométricas que agora estão a cair. Ser-lhe-á pedido que melhore o jogo, tornando-o mais dinâmico (as bombas também aparecem e podem matar o seu monstro).

## OBJETIVO GERAL

Neste desafio vamos melhorar o jogo do Geometry Hunter, tornando-o mais interativo e frenético.

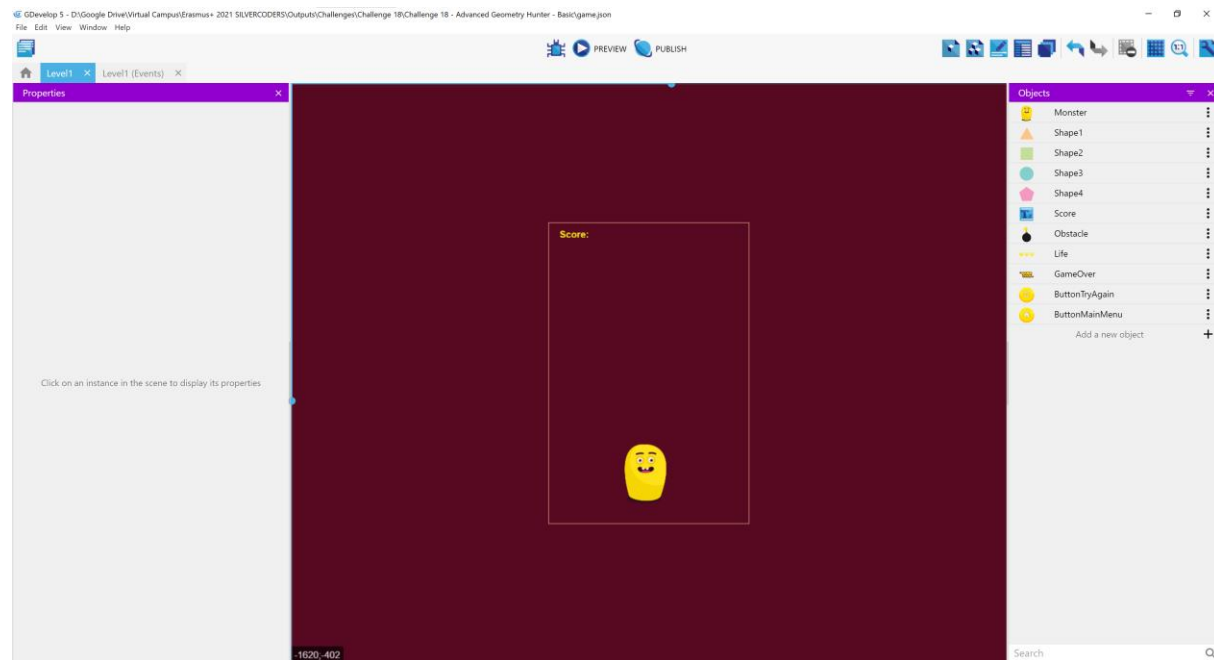
## OBJETIVOS DE APRENDIZAGEM

No final deste desafio, poderás...

1. Entender como o código é tratado por um computador e qual é o papel de um compilador.
2. Estar familiarizado com o conceito de línguas de baixo e alto nível e compreender quais são as suas diferenças e o que é necessário codificar em qualquer uma delas.
3. Ter experiência com uma suíte de programação visual e ser capaz de codificar uma pequena peça de software padrão com ele.
4. Saber o que são declarações e linhas de comando e o que significam para um compilador.
5. Escrever instruções utilizando a sintaxe correta e com erros mínimos.
6. Saber o que são os operadores, o que fazem e quais os símbolos que representam os operadores.
7. Ser capaz de entender a atribuição de valores a variáveis e como mudá-los.
8. Conhecer todas as operações aritméticas básicas e como usá-las.
9. Reconhecer e saber como utilizar todas as estruturas de dados relacionadas com os números.
10. Conhecer as estruturas ligadas ao uso do texto, como cordas e caracteres.
11. Utilizar Se as declarações forem corretamente executadas de acordo com uma determinada condição fixa definida.
12. Saber como usar o editor da Gdevelop
13. Entender os conceitos de cenas, eventos e objetos

# INSTRUÇÕES

1. Comece por abrir o editor da Gdevelop.
2. Utilize o menu **Ficheiro** para **abrir** o jogo Challenge 22 – Basic
3. Isto deve ser o que recibes.



4. Prima o botão **pré-visualização** para reproduzir o jogo. Pode mover o monstro para a esquerda e para a direita com as setas no teclado e deve pegar as formas geométricas que agora estão a cair. Por cada peça que apanhares, um ponto é adicionado à tua pontuação.
5. Repita o jogo quantas vezes quiser. Para repetir, tem de fechar a janela do jogo e premir o botão **Pré-visualização** no editor.
6. Agora que sabe o que são as mecânicas do jogo (o que pode fazer no jogo) vamos ver como se faz. Vamos focar-nos nas diferenças para o último desafio.

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

8. Quando o jogo começa, criamos um **Timer** chamado **ShapeCreation** que é um objeto que está sempre a contar o tempo em segundos.

SHAPES	
⌚ The timer "ShapeCreation" is greater than 2 seconds Add condition	🎨 Among objects <b>Shapes</b> , create object named "Shape" + ToString(RandomInRange(1,4)) at position RandomInRange(80, 640-80);-100 (layer: ) 🔄 Change the angle of <b>Shapes</b> : set to RandomInRange(0,360) 📏 Change the scale of <b>Shapes</b> : set to RandomFloatInRange(0.8, 1.6) ⌚ Start (or reset) the timer "ShapeCreation" Add action

1. Quando o **Timer ShapeCreation** atinge 2 segundos, uma nova forma é criada e pode ser aleatoriamente uma das quatro formas diferentes. Para tornar o jogo mais divertido, estas formas são dimensionadas e giradas. O reposição do Temporizador para 0 para começar a contar novamente.

Move shape according to the game speed	
Add condition	➡ Add to <b>Shapes</b> an instant force, angle: 90 degrees and length: Variable(GameSpeed) pixels 🔄 Rotate <b>Shapes</b> at speed 90 deg/second Add action

2. A forma «cai» adicionando-lhe uma força vertical.
4. Vamos dificultar um pouco o jogo. Vamos lançar também algumas bombas que podem remover vidas do monstro se for atingido. Para isso vamos usar o objeto **obstáculo**

OBSTACLE	
⌚ The timer "ObstacleCreation" is greater than 5 seconds Add condition	🎨 Create object <b>Obstacle</b> 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 <b>Obstacle</b> an instant force, angle: 90 degrees and length: 1.5*Variable(GameSpeed) pixels 🔄 Change the z-order of <b>Obstacle</b> : set to 4 Add action

5. O código para o objeto **obstáculo** é semelhante ao código para as formas. Temos um temporizador que o controla e o fazemos cair.

Repeat for each instance of Obstacle:	
🚗 <b>Obstacle</b> is in collision with <b>Monster</b> Add condition	✖ Delete <b>Obstacle</b> 💥 Damage <b>Monster</b> , removing 1 from its health 🔊 Play the sound killed.wav, vol.: , loop: no Add action

6. Mas se houver uma colisão com o Monstro, está danificada e ele recebe menos 1 ponto de saúde.

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

▶

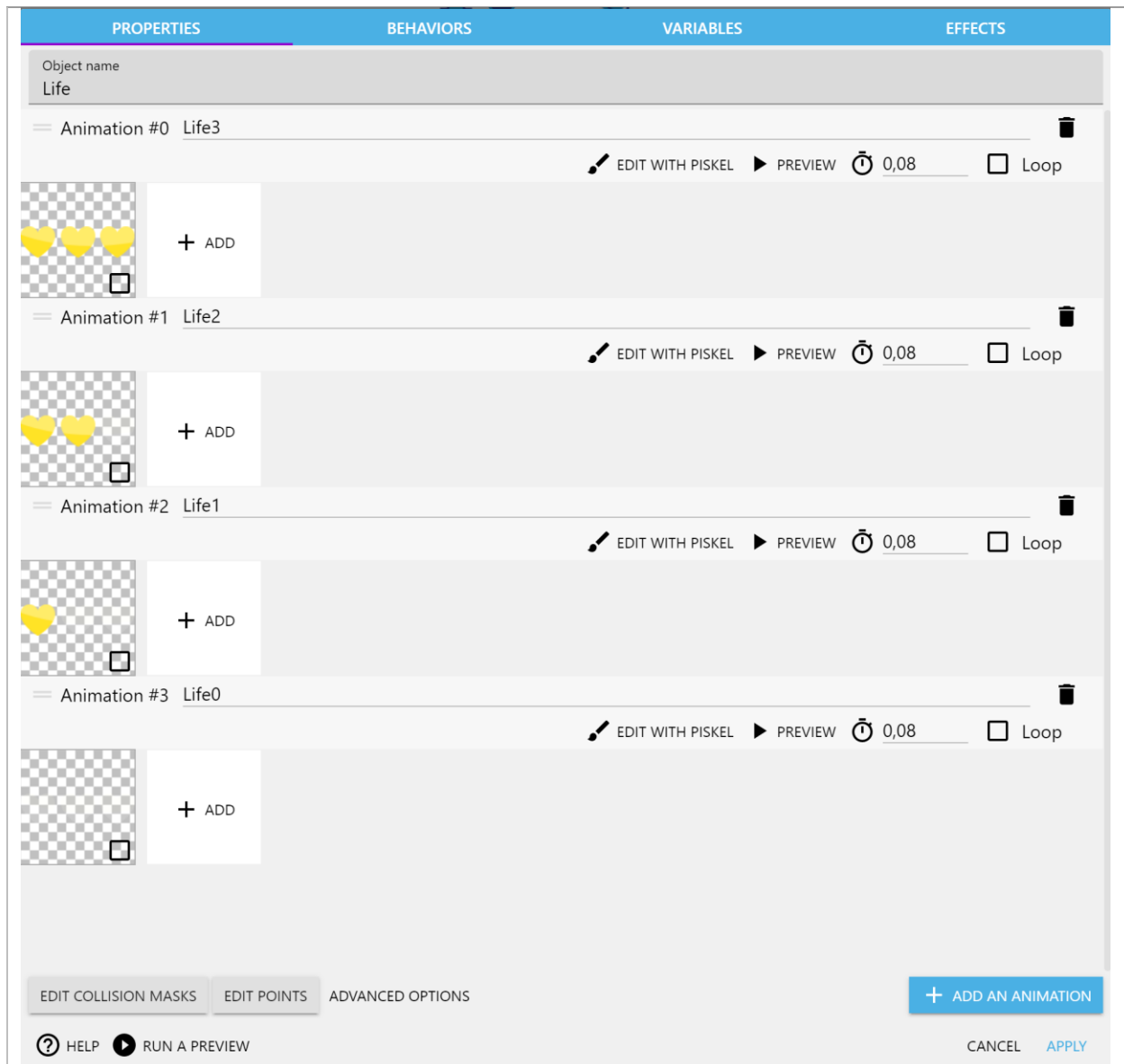
RUN A PREVIEW

+ ADD A BEHAVIOR

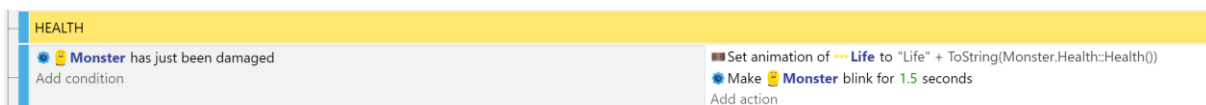
CANCEL

APPLY

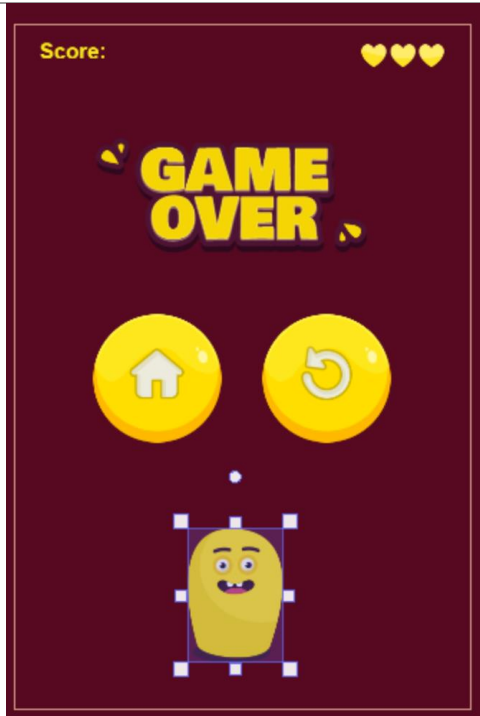
1. A saúde é um **Comportamento**, uma propriedade padrão que podemos associar aos objetos. A saúde máxima é 3.



2. Para representar visualmente a saúde do monstro, usaremos o objeto **Life** . Este Sprite tem 4 quadros, cada um representando um estado de saúde.



3. Quando o monstro estiver danificado, o Sprite move-se para o quadro seguinte.
4. Quando o monstro está sem vida, o monstro está morto. Temos de mostrar o botão GameOver e vamos criar dois botões, um para reiniciar o jogo e outro para sair do jogo. Isto é o que a cena inicial parece.



1. Não queremos ver estes objetos no início, por isso escondemo-los.

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

2. E mostramos-lhes quando o monstro está morto.

<ul style="list-style-type: none"> <li> Monster is dead</li> <li>Add condition</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> <li>Add action</li> </ul>
---	---

3. Agora verificamos em que botão o jogador clicou

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

### Challenge 22 (Basic)