

SilverCoders

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



RETO #28

TIC TAC TOE

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

Vamos a crear un juego del tipo tres en raya. Está pensado para ser jugado por dos personas.

OBJETIVO GENERAL

Vamos a crear un juego del tipo Tres en raya, pensado para ser jugado por dos personas. También aprenderemos sobre Arrays, una forma de almacenar datos.

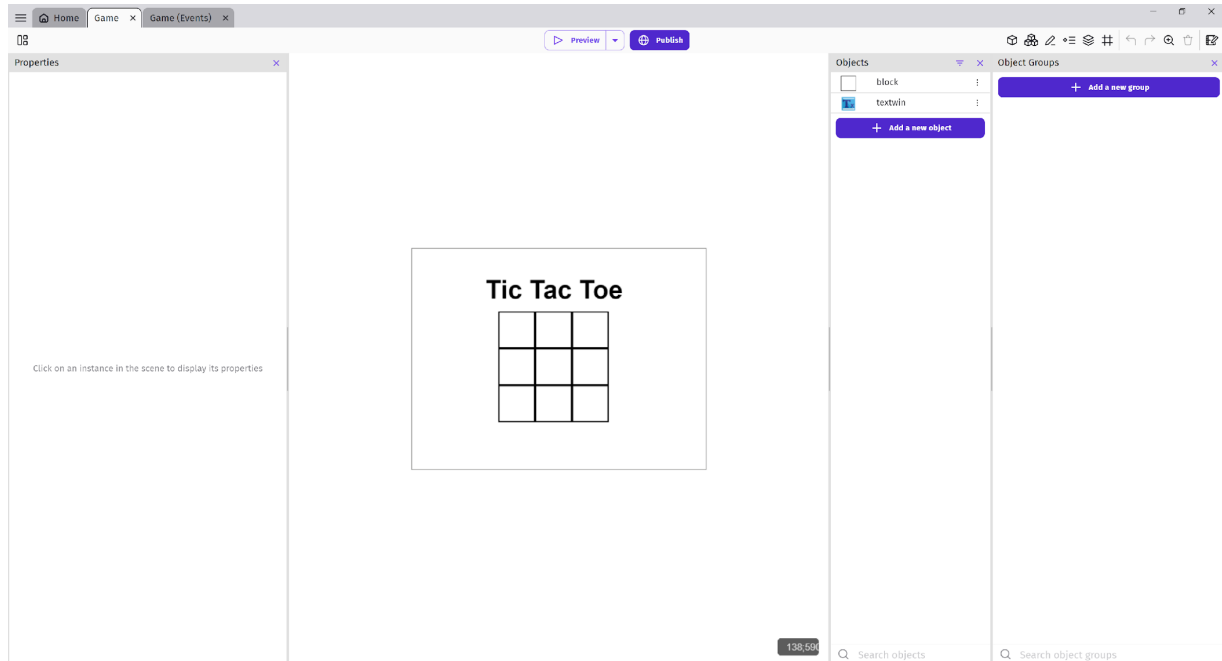
OBJETIVOS DE APRENDIZAJE

Al final de este desafío, usted será capaz de ...:

- 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 comando 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.
- Ser capaz de utilizar correctamente las sentencias If para ejecutar código según una determinada condición fija definida.
- Ser capaz de utilizar Arrays.

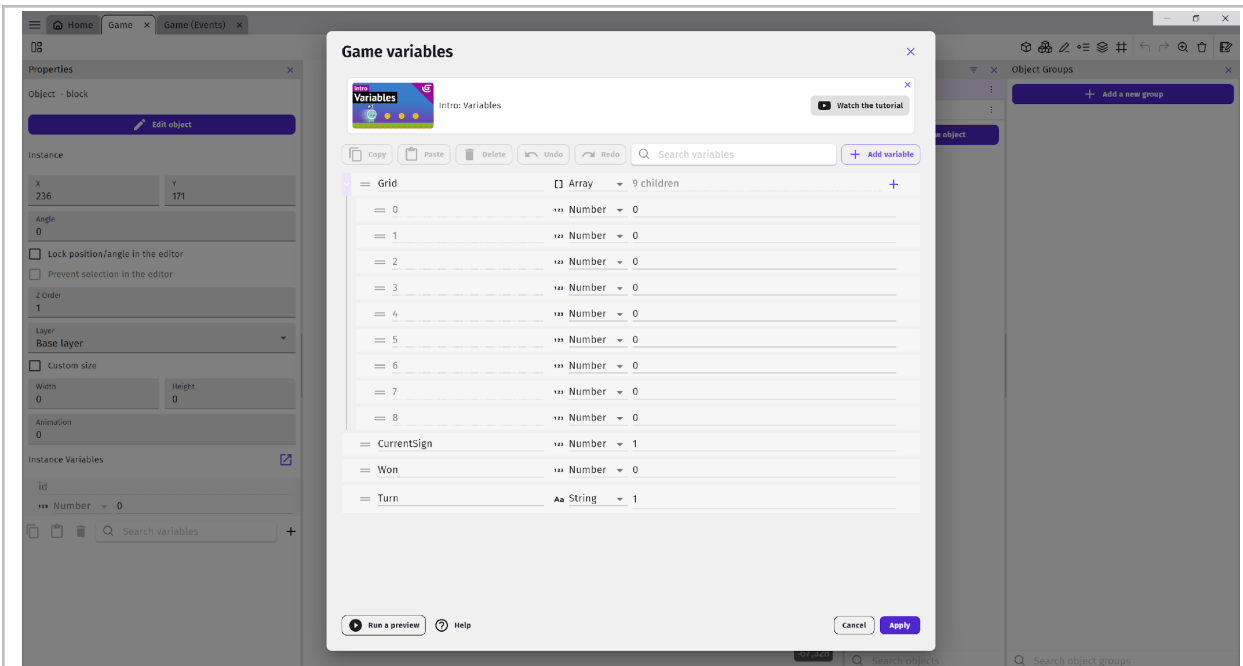
INSTRUCCIONES

Esta es tu configuración inicial. En este caso hemos proporcionado los objetos básicos que necesitarás para el juego. Como de costumbre empieza a revisarlos cuidadosamente.



Hay varios aspectos importantes en esta configuración:

- Cada ficha del tablero es un sprite de bloque. Cada instancia o copia de bloque tiene una variable llamada id que lo identifica. Así, las baldosas de arriba son 0, 1 y 2. Las baldosas de la fila media son 3, 4 y 5 y las de abajo son 6, 7 y 8.
- El sprite del bloque tiene 3 frames: uno para el espacio vacío (frame 0), otro para la X (frame 1) y otro para la O (frame 2).
- La escena tiene creadas varias variables:
- CurrentSign indica qué frame (o signo debe ser representado cuando elegimos una ficha).
- Won nos dice si alguien ya ha ganado.
- Turn nos dice si es el jugador 1 o 2 el que juega
- La variable más importante es Grid, un array con 9 posiciones que nos indica qué símbolo está en una determinada posición. Cuando empezamos todas las posiciones son 0 (vacías).



También tenemos el código que inicia el juego y tenemos la estructura para el resto del código.

When starting the scene, assign the first Turn to a random player 0 or 1, where 0 is X, and 1 is O	
At the beginning of the scene	Change the scene variable Turn : set to Random(1)
Add condition	Add action
Check whose Turn it is and set the appropriate animation to the block	
Left mouse button was released	Add action
Add condition	
Each box (block) has different ids representing there distance from the first box (Box at top left is 0 and the increments by 1 from there to the right and to the bottom row)	
The Grid index that matches the id of the block is set to 1 if it X and 2 if it is Y	
The scene variable Turn = 0 The cursor/touch is on block The number of the animation of block = 0 Add condition	Change the number of the animation of block : set to 1 Change the scene variable Turn : set to 1 Change the scene variable Grid[block.Variable(id)] : set to 1 Add action
The scene variable Turn = 1 The cursor/touch is on block The number of the animation of block = 0 Add condition	Change the number of the animation of block : set to 2 Change the scene variable Turn : set to 0 Change the scene variable Grid[block.Variable(id)] : set to 2 Add action

Este código establece aleatoriamente el jugador inicial. Luego comprueba si hemos pulsado una ficha vacía, pone allí el símbolo del jugador y rellena la posición de la rejilla correspondiente con el valor correcto.

Lo que queda es comprobar si un jugador uno. Eso significa comprobar si ha conseguido poner 3 símbolos iguales en una línea horizontal, vertical o diagonal. Lo haremos comprobando la matriz Grid. Empecemos con las líneas horizontales:

Winning system

The value of each box is stored in an array (from 0 to 8) and their value can be 0 (means empty) 1 (means X) 2 (means O)

```
[0 | 1 | 2]
[3 | 4 | 5]
[6 | 7 | 8]
```

The scene variable ☒ Won = 0

Add condition

CurrentRow represents the current row we are checking (if there is a match of 3). CurrentColumn represents the current Column we are checking (if there is a match of 3). CurrentSign represents the current sign (1= X or 2= O) we are checking for a match

Checking for horizontal matches,
CurrentRow is added to the the current Grid index (box) value so that we are iterating through all the columns with 3 values neighboring them instead of having seperate events for each row that is,
CurrentRow increments by 3
When CurrentRow is 0, (0+0 = 0 | 1+0 = 1 | 2+0 = 2)

```
[0 | 1 | 2]
[ | | ]
[ | | ]
```

When CurrentRow is 3, (0+3 = 3 | 1+3 = 4 | 2+3 = 5)

```
[ | | ]
[3 | 4 | 5]
[ | | ]
```

When CurrentRow is 6, (0+6 = 6 | 1+6 = 7 | 2+6 = 8)

```
[ | | ]
[ | | ]
[6 | 7 | 8]
```

The scene variable ☒ Grid[0+Variable(CurrentRow)] = Variable(CurrentSign)
The scene variable ☒ Grid[1+Variable(CurrentRow)] = Variable(CurrentSign)
The scene variable ☒ Grid[2+Variable(CurrentRow)] = Variable(CurrentSign)

Add condition

Change the scene variable ☒ Won: set to Variable(CurrentSign)

Add action

En los comentarios del código se da una explicación completa.

Ahora, para las líneas verticales.

Checking for vertical matches
CurrentColumn increments by 1
When CurrentColumn is 0, (0+0 = 0 | 3+0 = 1 | 6+0 = 1)

```
[0 | | ]
[3 | | ]
[6 | | ]
```

When CurrentColumn is 1, (0+1 = 0 | 3+1 = 1 | 6+1 = 1)

```
[ | 1 | ]
[ | 4 | ]
[ | 7 | ]
```

When CurrentColumn is 2, (0+2 = 0 | 3+2 = 1 | 6+2 = 1)

```
[ | | 2]
[ | | 5]
[ | | 8]
```

The scene variable ☒ Grid[0+Variable(CurrentColumn)] = Variable(CurrentSign)
The scene variable ☒ Grid[3+Variable(CurrentColumn)] = Variable(CurrentSign)
The scene variable ☒ Grid[6+Variable(CurrentColumn)] = Variable(CurrentSign)

Add condition

Change the scene variable ☒ Won: set to Variable(CurrentSign)

Add action

Y finalmente para las diagonales.

Checking for diagonal matches,
[| | 2] [0 | |]
[| 4 |] OR [| 4 |]
[6 | |] [| | 8]

The scene variable ☒ Grid[2] = Variable(CurrentSign)
The scene variable ☒ Grid[3] = Variable(CurrentSign)
The scene variable ☒ Grid[6] = Variable(CurrentSign)

Add condition

Change the scene variable ☒ Won: set to Variable(CurrentSign)

Add action

The scene variable ☒ Grid[0] = Variable(CurrentSign)
The scene variable ☒ Grid[4] = Variable(CurrentSign)
The scene variable ☒ Grid[8] = Variable(CurrentSign)

Add condition

Change the scene variable ☒ Won: set to Variable(CurrentSign)

Add action

Change the scene variable ☒ CurrentRow: add 3
Change the scene variable ☒ CurrentColumn: add 1
Change the scene variable ☒ CurrentSign: add 1

Add action

Ahora tenemos que ocuparnos del cambio de turno al siguiente jugador.

Add action	
When CurrentSign is greater than 3, it is reset to 1 (X)	
<input checked="" type="checkbox"/> The scene variable <input checked="" type="checkbox"/> CurrentSign > 2 Add condition	<input checked="" type="checkbox"/> Change the scene variable <input checked="" type="checkbox"/> CurrentSign: set to 1 Add action
When CurrentRow is greater than 6, that is at row 3 (last row)	
<input checked="" type="checkbox"/> The scene variable <input checked="" type="checkbox"/> CurrentRow > 6 Add condition	<input checked="" type="checkbox"/> Change the scene variable <input checked="" type="checkbox"/> CurrentRow: set to 0 Add action
When CurrentRow is greater/equal than 3, that is at column 3 (last column)	
<input checked="" type="checkbox"/> The scene variable <input checked="" type="checkbox"/> CurrentColumn ≥ 3 Add condition	<input checked="" type="checkbox"/> Change the scene variable <input checked="" type="checkbox"/> CurrentColumn: set to 0 Add action

Y si alguien lo hace, felicitémosle.

<input checked="" type="checkbox"/> The scene variable <input checked="" type="checkbox"/> Won ≠ 0 <input checked="" type="checkbox"/> Trigger once Add condition	Add action
The title is changed according to which player won	
<input checked="" type="checkbox"/> The scene variable <input checked="" type="checkbox"/> Won = 1 Add condition	<input checked="" type="checkbox"/> Change the text of <input checked="" type="checkbox"/> textwin: set to "X Won" Add action
<input checked="" type="checkbox"/> The scene variable <input checked="" type="checkbox"/> Won = 2 Add condition	<input checked="" type="checkbox"/> Change the text of <input checked="" type="checkbox"/> textwin: set to "Y Won" Add action

RECURSOS

Reto 28 (Básico)