SilverCoders

DIGITAL LITERACY IMPROVEMENT THROUGH EFFECTIVE LEARNING EXPERCIENCES FOR ADULTS

CHALLENGE #19

SUPER ADVANCED COIN FETCHER

CODING TRAINING PROGRAMME

FOR +55 ADULTS

SILVER CODERS

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

Co-funded by the European Union

STRUCTURE OF THE CHALLENGE

DESCRIPTION

This lesson takes the game developed in the previous challenge and develops it further, making it more complex and attractive while exploring other aspects of GDevelop.

GENERAL GOAL

This lesson continues to promote the understanding of the Gdevelop environment and how it can be used to code. It focuses on additional relevant GDevelop concepts like Scenes, Timers and Behaviours.

LEARNING OBJECTIVES

In the end of this challenge, the learner will be able ...:

- To have experience with a visual programming suite and to code a small piece of software with it.
- To know what statements and command lines are.
- To write instructions using correct syntax.
- To be able to use If statements correctly to execute code according to a certain defined fixed condition.
- To use the Gdevelop editor
- To understand the concepts of scenes, events and objects
- To understand the concept of variables.
- To understand the concept of Scenes, Timers and Behaviours.

INSTRUCTIONS

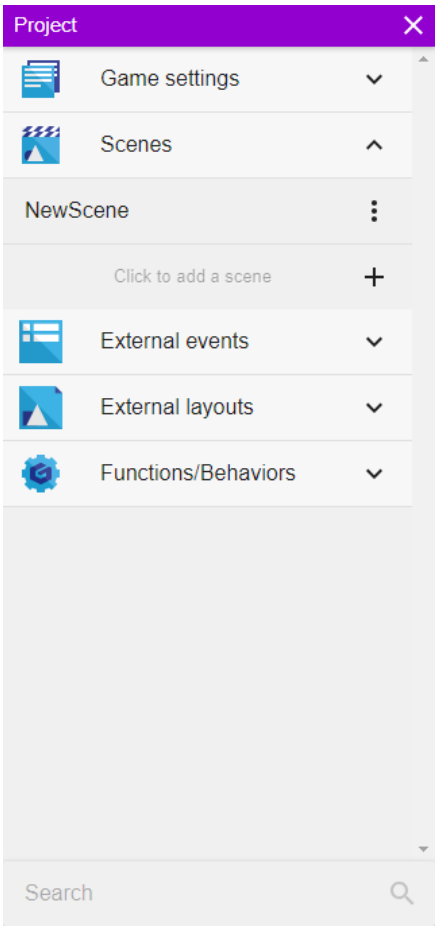
Welcome back to the coding and programming challenges.

In this challenge we will develop further the game with Kenney while we learn about Scenes, Behaviours and other aspects of Gdevelop.

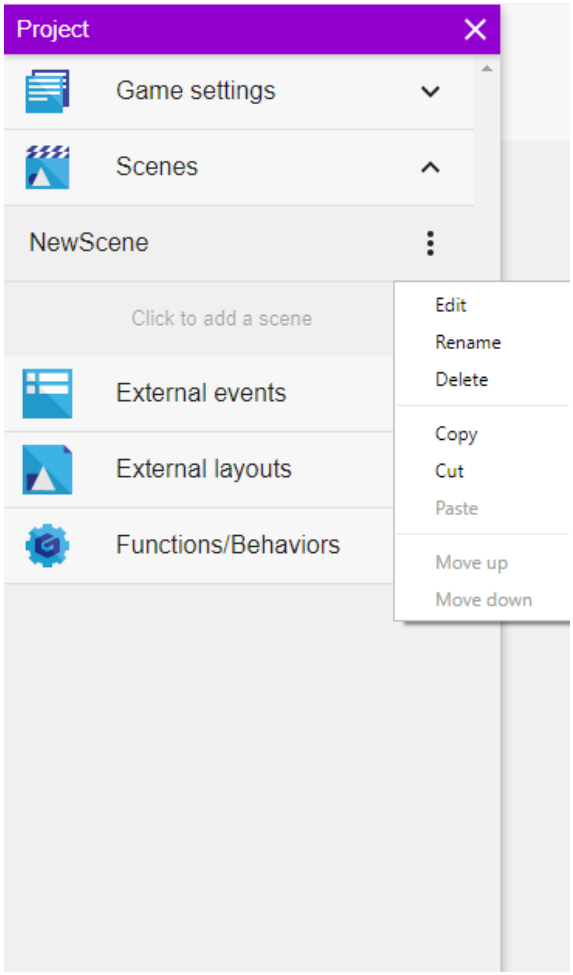
- Open the Challenge 19 – Super Advanced Coin Fetcher – Initial. This is the same game that you created in the last challenge.

Scenes

A Scene is a part of a game which includes the visual layout and the events associated with that layout. When you start a new Scene if will be assigned a default name and properties. To change that you have to open the “Project Manager” on the “View” Menu.



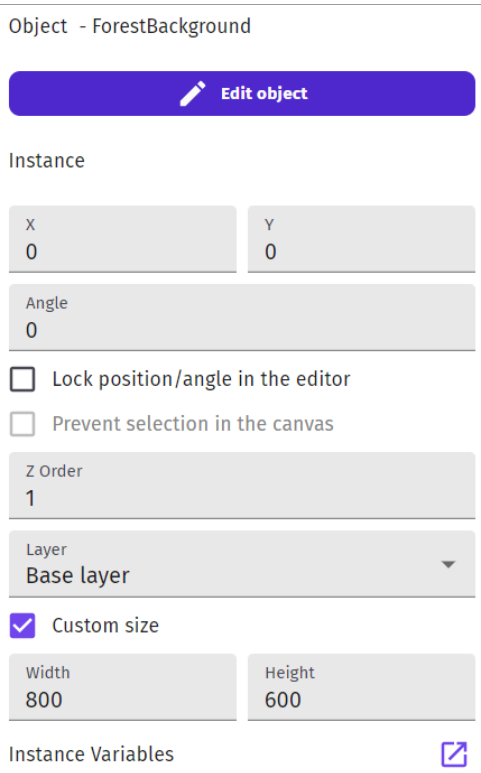
Let's change the name of the scene that we have. Go to “View”->”Show Project Manager” and click in the dots next to “NewScene”. Choose “Rename” and write “KenneyScene”.



Now, in the same menu “View->Show Project Manager”, create a new scene, name it “EntryScene” and on the dot menu choose “Set as entry scene”. When we start the game this will be the first scene.

This scene is empty so let’s create a Text Box and fill it with the title of the game “Kenney’s Game”. Then format it as you like and add it to the middle of the layout.

We can also add a nice background to it. Do “Add a new object”, “Asset Store” and choose one of the existing backgrounds (Note: that this is only possible if you are connected to the network). In the end do “Add to the scene”. You may have to configure the properties of the image to show it correctly.

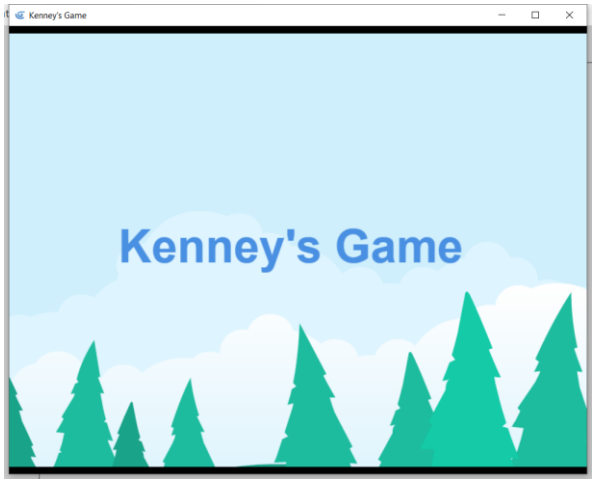


In our case, the object is called “ForestBackground”. We set the position to X=0, Y=0 to place it on the upper left corner and we defined the width=800 and height=600 to make it occupy the entire scene. We also define the “Z order” as 1.

Z Order

The “Z order” indicates which objects will be shown in front and the ones will be shown in the back. The lower the number the more backwards will be the object.

Select the TextBox you created and assign a “Z Order” of 2 to be in front of the background.



If you run the game, you will see something similar to this (depends on the background you

chose). But it doesn't do anything, right?

We will use a Timer so that after a certain time, the game will move to the scene with Kenney.

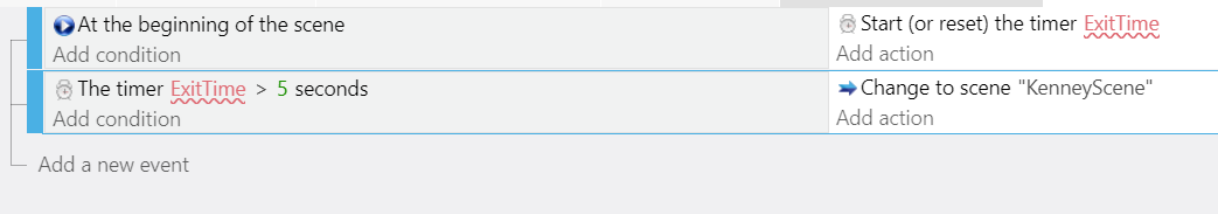
Timers and time

Timers can be run using actions inside events. You can then use conditions to check if enough time elapsed before running other actions or reseting the timer. Timers can be attached to scenes or to objects.

Special actions like "Wait X seconds" can be used to wait before launching the next actions in an event. This is useful for creating cut-scenes, timelines or just effects happening not immediately.

Finally, there are expressions to get the current time, day and time since the start of the game.

Create the following events:



What these events do is the following:

- When the scene starts we create a Timer called "ExitTIme" that starts counting the time
- When the Timer reaches 5 seconds, we move to the scene with Kenney

To make this a little bit more fun, we will animate the text before leaving to the next scene. We will use a behaviour associated to the Text Box for this.

Behaviors: pre-defined rules and logic for objects

Behaviors enhance an object with some pre-defined programming logic. They can be reasonably straightforward, automating simple tasks, or much more advanced tasks. For example:

- A behavior can be used to automatically remove an object from the game when it goes out of the screen.
- Another behavior can be used to move objects on the screen with the keyboard arrows.
- Yet another behavior can be used to allow the object to be dragged on screen with the mouse or by touching the object.
- The Physics behavior is an example of an advanced behavior which make your objects move in a realistic way, following the laws of physics.

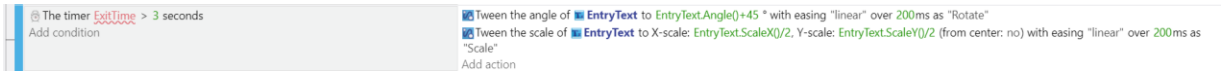
Behaviors will often come with their own variables that can be changed to customize the task it performs, but they can also be manipulated using events that are specific to that behavior.

On the Object menu (to the right), in the dotted menu option for the TextBox select “Edit behaviors”. Then do “Add Behavior” and select “Tween”. Then “Apply”.

A Tween is a behaviour that allows to change a property of an object from an initial state to a final state. It can be the scale (size) of the object, its position, the angle, etc. The change takes some time to unfold and we can program that time. What we did before was associating a behaviour with the TextBox object.

Next we need to indicate which type of Tween we want.

On the code, add the following actions:



So, after 3 seconds what will happen is that the TextBox starts rotating (first Tween, changes angle) and shrinking (second Tween, changes scale)

HOPE YOU ENJOYED! SEE THE FINAL VERSION OF THE CHALLENGE AS IT HAS SOME ADDITIONAL FEATURES.

RESOURCES

Challenge 19 (Initial)