

SilverCoders

DIGITAL LITERACY IMPROVEMENT THROUGH EFFECTIVE
LEARNING EXPERIENCES FOR ADULTS



CHALLENGE 17 COIN FETCHER

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

STRUCTURE OF THE CHALLENGE

DESCRIPTION

In this challenge we are going to create a very simple game: because our character Kenney is very greedy we will make him move so that he can fetch some coins.

GENERAL GOAL

This lesson intends to give the basic understanding of the Gdevelop environment and how it can be used to code. We will focus on the main steps and actions to start developing a game and how to use events and objects to do it.

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

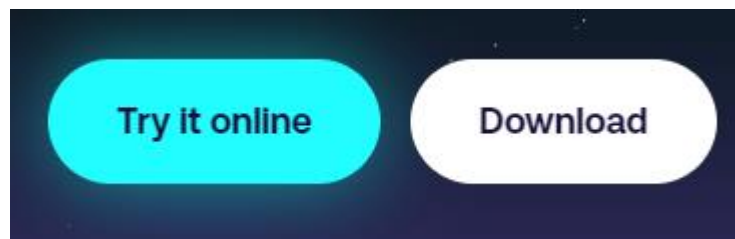
INSTRUCTIONS

WELCOME TO THE CODING AND PROGRAMMING CHALLENGES.

For this set of challenges we will be using GDevelop, a game editor, that is, an application that allows you to create games. What is good about GDevelop is that you can create games with just some basic programming skills. To start doing it, this specific challenge provides information about the major elements of the GDevelop editor: events and objects.

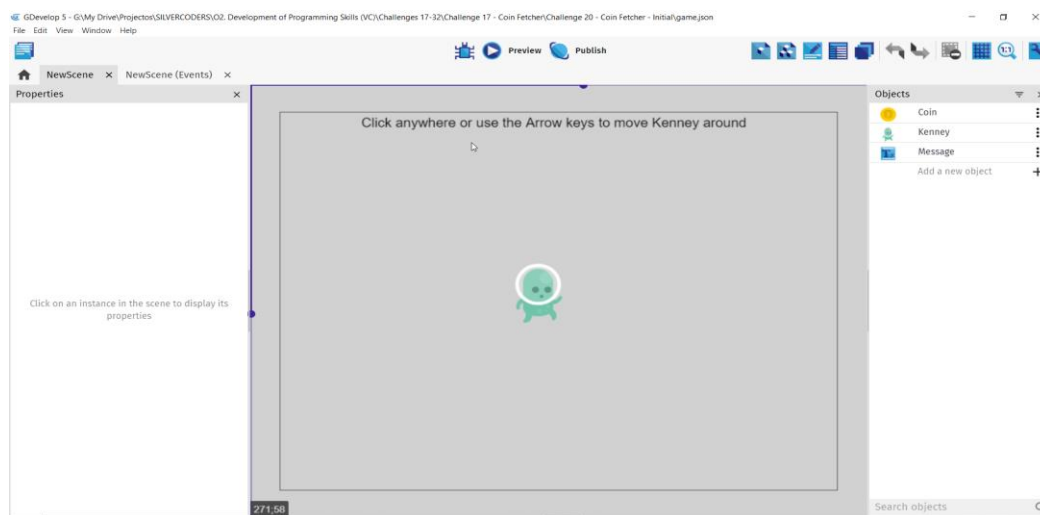
But let's start by opening the GDevelop environment. Your trainer should have explained you how to access, download and use the game editor and you also have two supporting videos on that. But here is a summarized list of instructions:

- Goto <https://gdevelop.io/>
- You can choose to use the game editor online or to download it.



- If you choose “Download” the application will be transferred to your computer and then you have to install it. Otherwise you can just use your browser to run the Gdevelop editor.
- When it is installed, open it.
- Then open the Challenge 17 - Coin Fetcher – Initial. This is your basic setup for the game – in each challenge we will provide such a basic setup and we explain how to reach the final version of the game

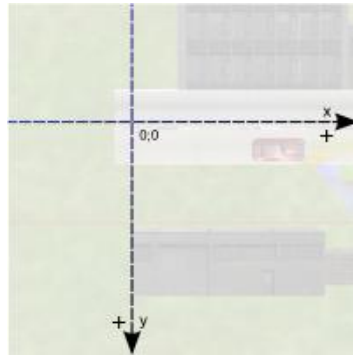
When you open the initial setup for Challenge 17, this is what you get.



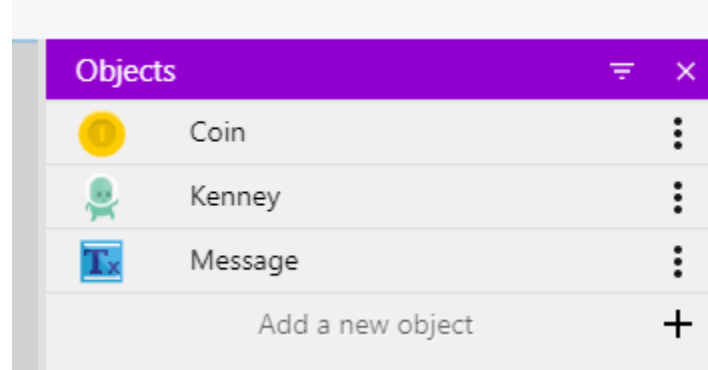
On the center, you have the visual layout – this is what your game will look like. On the right, you have the list of objects that you will use in the game. When you select an object (double click on it) its properties will show up on the left. Just play around to get familiar with the GDevelop environment. Click on the objects to see their properties. When you are done, reload the game again to prevent any changes you might have made. Now let's focus on the two most important aspects of creating a game with GDevelop: objects and events.

Objects

Everything that is displayed on the screen is called an “Object”. Different kind of objects can be used to display different game elements on the screen. For example, most game graphics are “Sprite” objects and texts can be displayed with “Text” object. Objects on the GDevelop screen/scene have X and Y coordinates. These coordinates correspond to the horizontal position (X-axis) and the vertical position (Y-axis) on the Cartesian plane. The X-coordinate decreases as you go to the left and increases as you go to the right. The Y-coordinate increases as you go down and decreases as you go up.



In Challenge 17, you start with 3 objects (actually 3 types of objects).



Kenney is our hero, our avatar. He will try to pick Coin(s). The Message object is a text box that allows us to write the instructions on the screen.

Events

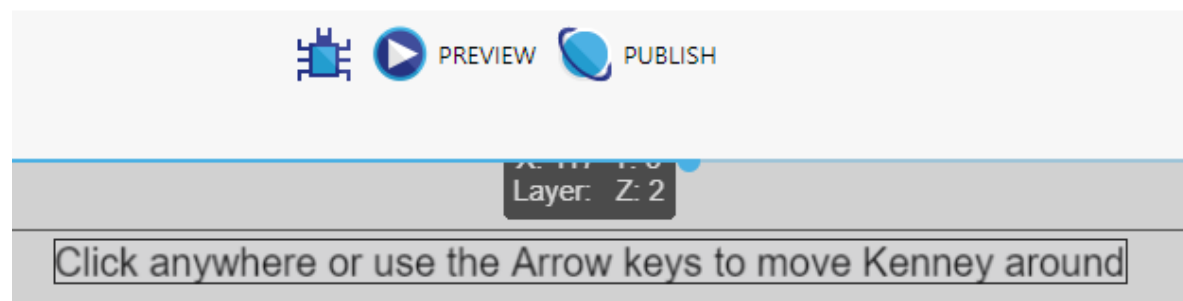
Events are used to create the rules of your game by coding and programming. They represent a sequence of instructions composed of conditions and actions. Conditions can be

thought of as “if” and actions can be thought of as “then”, for the purpose of making things happen in your game. “If” the conditions are true/met, “then” the actions will happen. Most conditions and actions refer to objects so Conditions run a test on the object and actions manipulate the objects. They can change the object position, appearance, etc... You can create events on the tab “New Scene (events)”

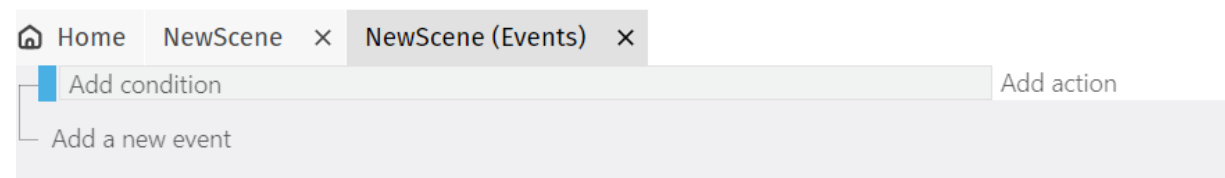
Now that you have understood this, let's create the Coin Fetcher game. In the game, Kenney (an object) is a character with the job of fetching coins. We drop a coin by clicking on the mouse (the coin will drop on the screen position of the mouse cursor) and Kenney will move to catch that coin. If we click on another position of the screen the coin will be displaced and Kenney will move to the new location. If Kenney picks a coin he will stop and wait for us to drop another coin.

On top of the screen you have the »Preview« button that allows you to see how your game runs. You can try it now, a new window will appear but nothing will happen as we just have the visuals of the game, we are missing the code (that is your task).

ge 17 - Coin Fetcher\Challenge 17 - Coin Fetcher - Initial\game.json *



To create the code for the game click on the tab “NewScene (Events)” – this is where we will insert our code (right now it is empty). Click on the button “Add an Event”. It will create an empty event (you can see it on the top, with the empty condition on the left and the empty action on the right).



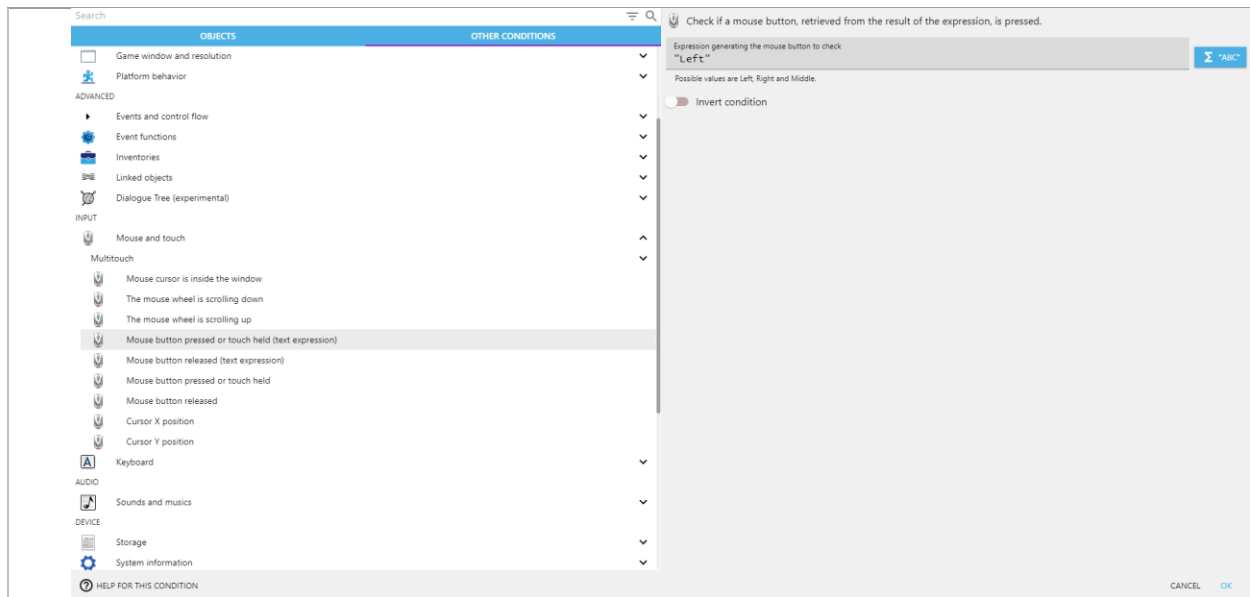
To start, we will make a Coin appear every time we click on the left mouse button.

Click on »Add condition«

Select »Other conditions«

Select »Mouse and touch« and then »Mouse button pressed or touch held«

On the top-right, choose »Left«

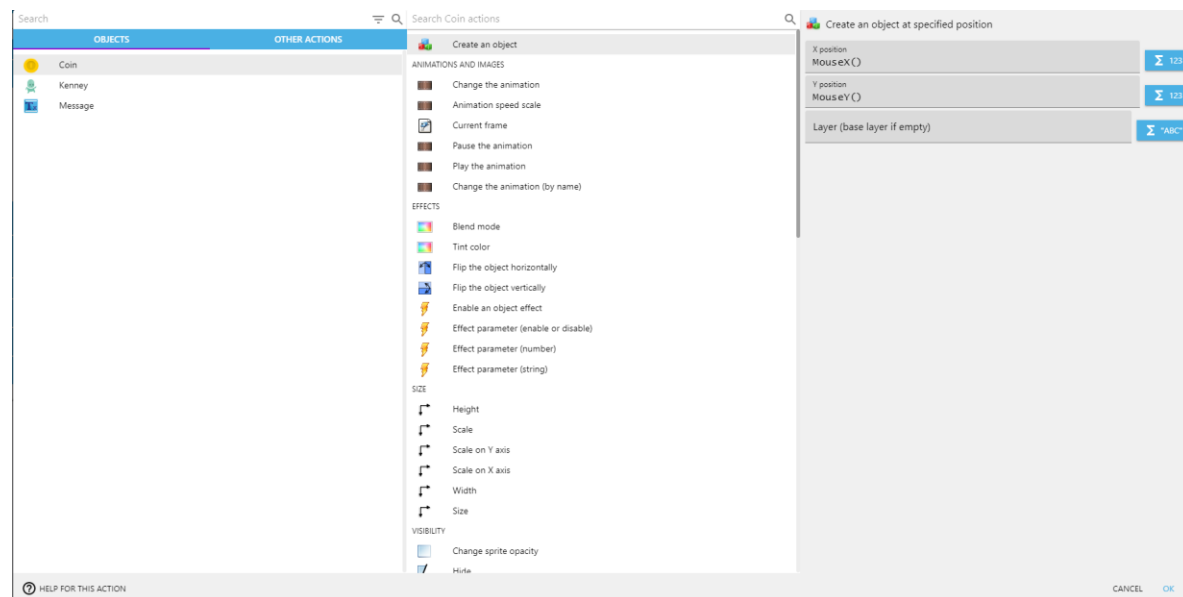


We have just created a condition that will be true when the user clicks the left button. Now, when that happens we must make a Coin appear.

Click on »Add action« and then on the Coin

Select »Create an Object«

Now, on the right we have to input the position where the Coin will appear. You can put numeric values but we want the Coin to appear on the position of the mouse cursor. So we will write `MouseX()` and `MouseY()`. These are two functions that tell us the position of the cursor and we will use that to position our Coin.



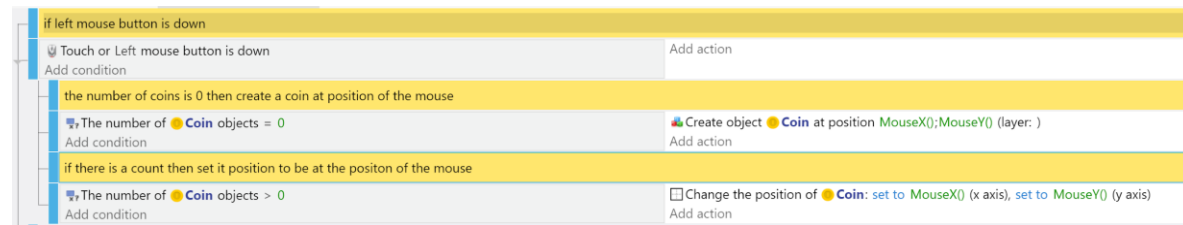
Do »Preview«

A Coin should appear everytime you click. And you have created your first event that operates an Object.

However we should improve this as we just want one Coin in the scene at any moment (now a Coin appears every time we click the mouse button). So:

- If there are no Coins in the visual layout, we create a Coin that will appear on the position of the cursor.
- If there is already a Coin, we will just move it to the position of the cursor.

This is the corresponding code – try to add it yourself.



You notice that now we have three events. But two of them are indented to the right, it is like they belong to the first event. And that means that they will only be called if the first event is true.

In yellow, we have "Comments". Comments have no influence on the code, but they help us understand what the code will do (they describe the logic of the code). You can add comments by clicking the right mouse button over the »Add event« or on the top-right menu.

Now, we want to make Kenney move to catch a Coin.

Moving objects can be achieved by applying forces (pushing) to the objects.

You can specify:

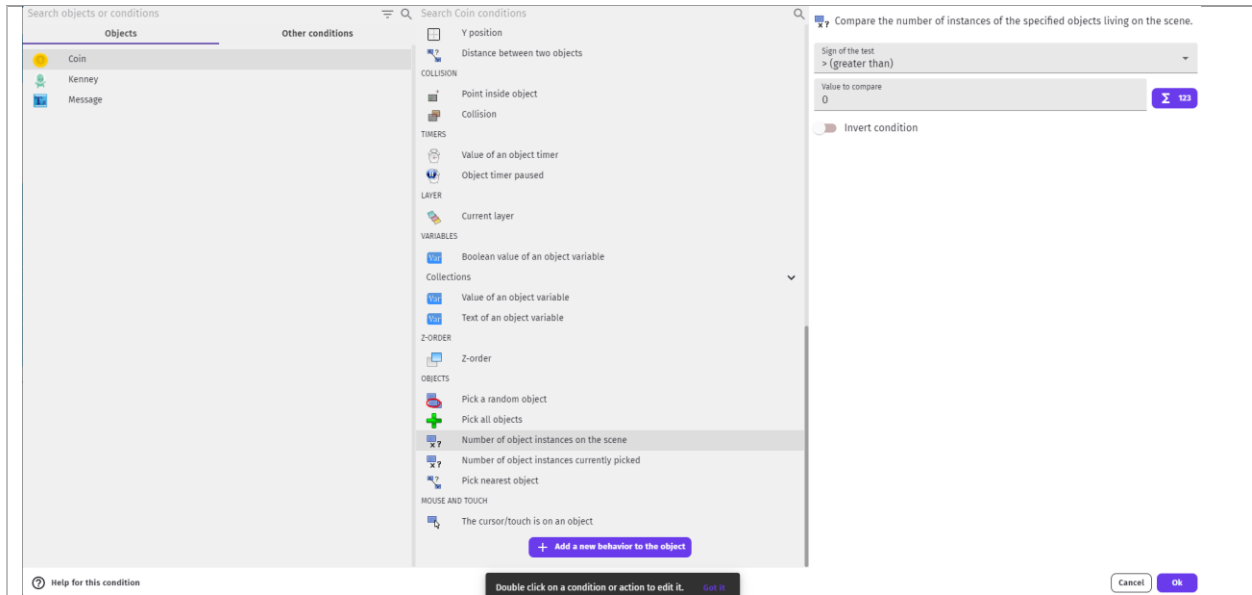
- the coordinates of a force on the X and Y axis, in pixels,
- or its polar coordinates (the force angle, in degrees, and length, in pixels),
- if the force is instant (will only push "a bit" the object) or continuous (will keep pushing the object until it's removed).

Select »Add new event«

Select »Add condition«

Select »Coin« and »Number of object instances on the scene«

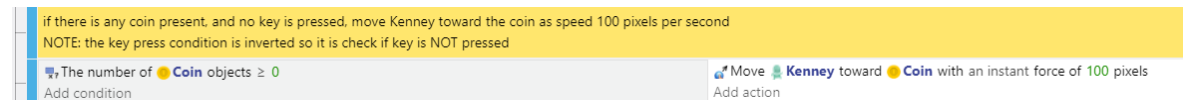
On the right choose »>(greater than)« and 0



On the action part, do

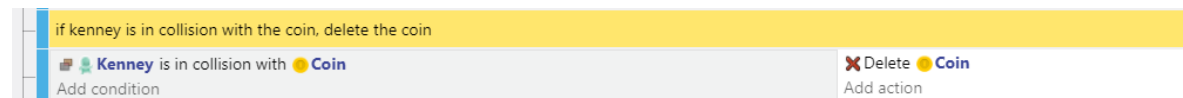
Select Kenney and »Add a force to move towards an object«

Choose Coin and add 100 pixels speed.



Now Kenney is moving to the Coin but nothing happens when he gets there. He can't collect the Coin!

Let's take care of that – when Kenney gets to the Coin (he collides with it) he will pocket it so it will disappear (be deleted). Here is the code – try inserting it.



HOPE YOU ENJOYED CREATING THE KENNEY GAME!

Note: it is possible to create events without conditions. In that case, the action will always be executed (as if the condition is always true).

RESOURCES

Challenge 17 (Initial)