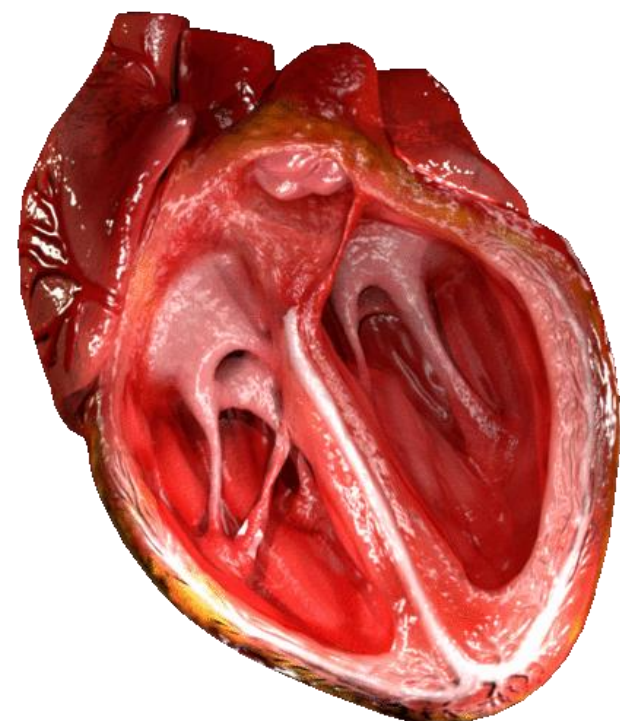




ATL Game Development Platform

Day 09

Creating a animation and Introduction to ATL Game Development Platform



**By Mr. Jitender Kumar &
Ms. Supriya Kadam from
Learning Links Foundation**

Agenda of the day

01

Reflections of Day - 8



02

Introduction to variable, Declaring Instant and Global Variable, Introduction to arrays, Adding an array in construct - 2,



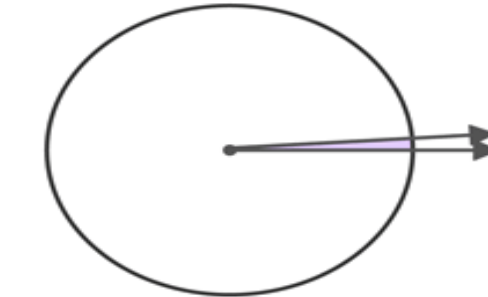
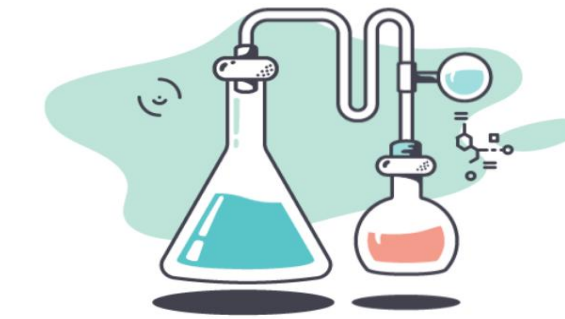
03

Creating an animation using Platform and scroll Behavior, Introduction to ATL Game Development Platform

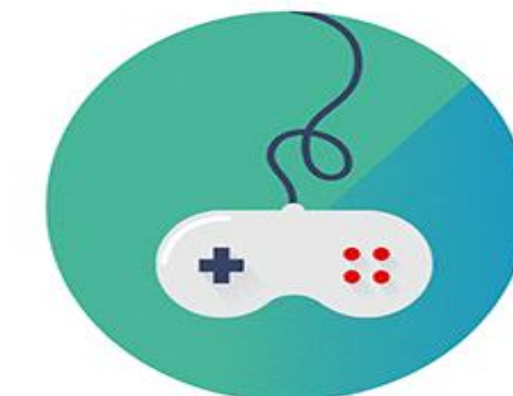


04

Home Assignment, Q&A



$\frac{1}{360}$
1°

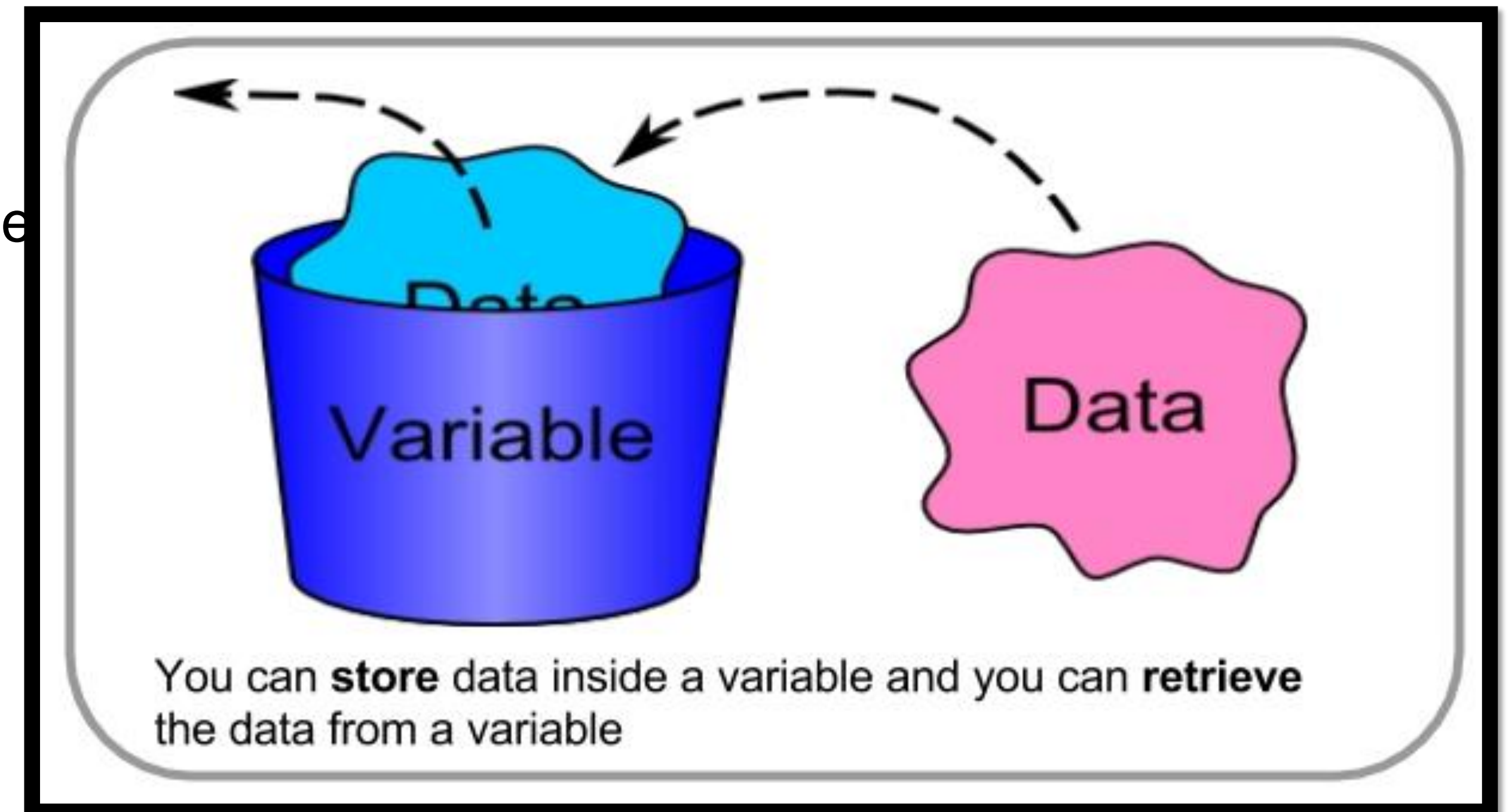


Reflections of Day 8

- Adding images, Editing Images
- Controlling the sprite with keyboard
- Organizing Event Sheet
- Adding Mouse Control
- Adding and destroying the enemy
- Adding spawned and collision

Introduction to variables

- Variables are the names you give to computer memory locations which are used to store values in a computer program.
- For example, assume you want to store two values 10 and 20 in your program and at a later stage, you want to use these two values. Let's see how you will do it. Here are the following three simple steps –
 - Create variables with appropriate names.
 - Store your values in those two variables.
 - Retrieve and use the stored values from the variable



Variables in construct 2

- Event Variables are number or text values which are either global to the project or local to a range of events.
- They are modified using the Event Variable dialog. To add an event variable, right-click on an event, another variable, or an empty space in the event sheet, and select Add global variable or Add local variable.
- Variables at the root level of the event sheet (not indented beneath anything else) become global variables, whereas variables in groups or sub-events become local variables.
- Event variables are modified with the system actions in the Global & local variables category. They can be retrieved by simply using their name in expressions.

- **Global variables :**

- Global variables show a globe icon. They are always at the top level of an event sheet - they are not sub-events or inside any groups.
- Global variables store their values between layouts. Events in any layout can access any global variable, even if it was created in a different event sheet that is not included.
- Global variables can be moved to another event sheet by cutting and pasting them.
- After being cut, references to the global variable will disappear because it has been removed; this is normal and nothing to worry about. When you paste the global variable, the references that disappeared will reappear again.



Global number Variable1 = 0

Local variables

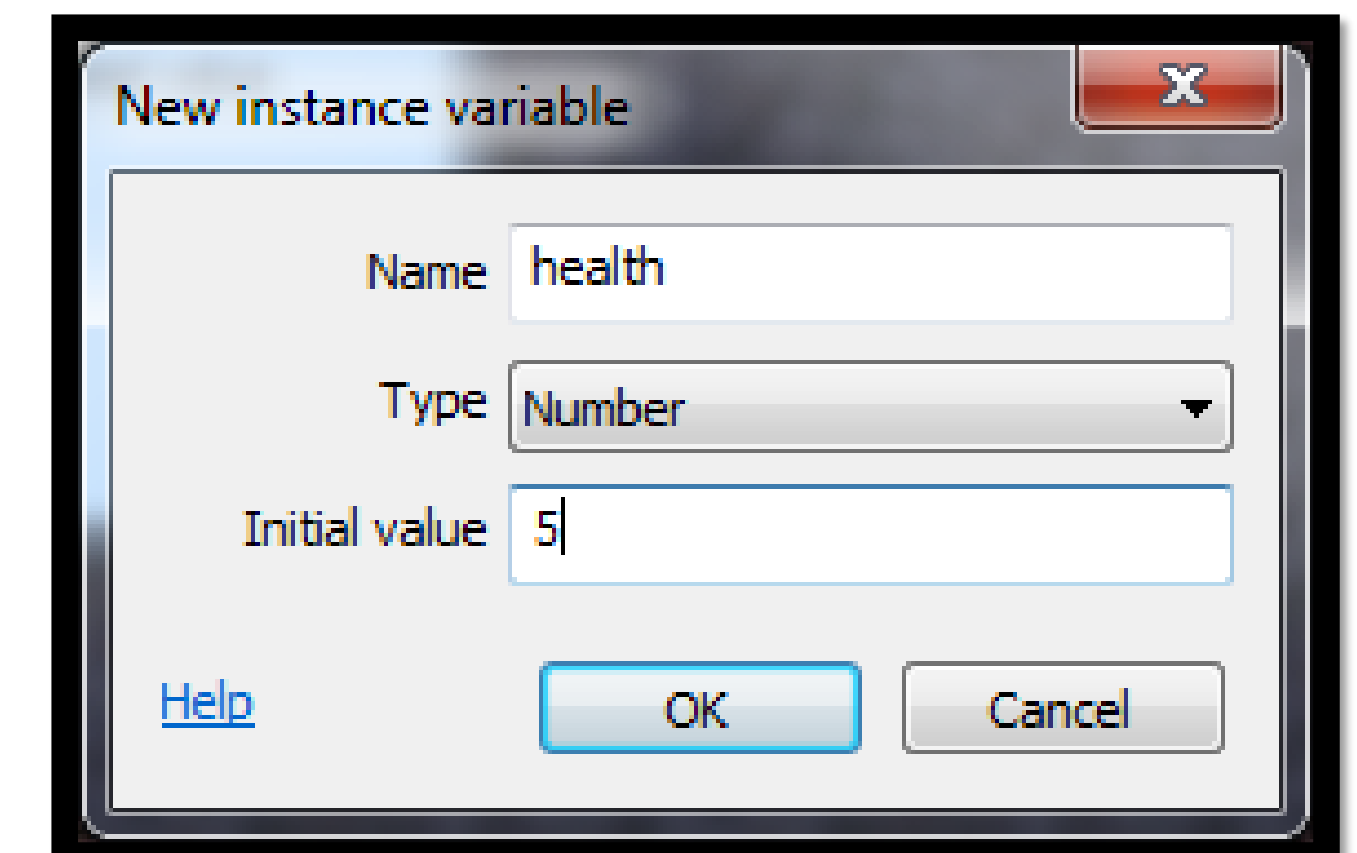
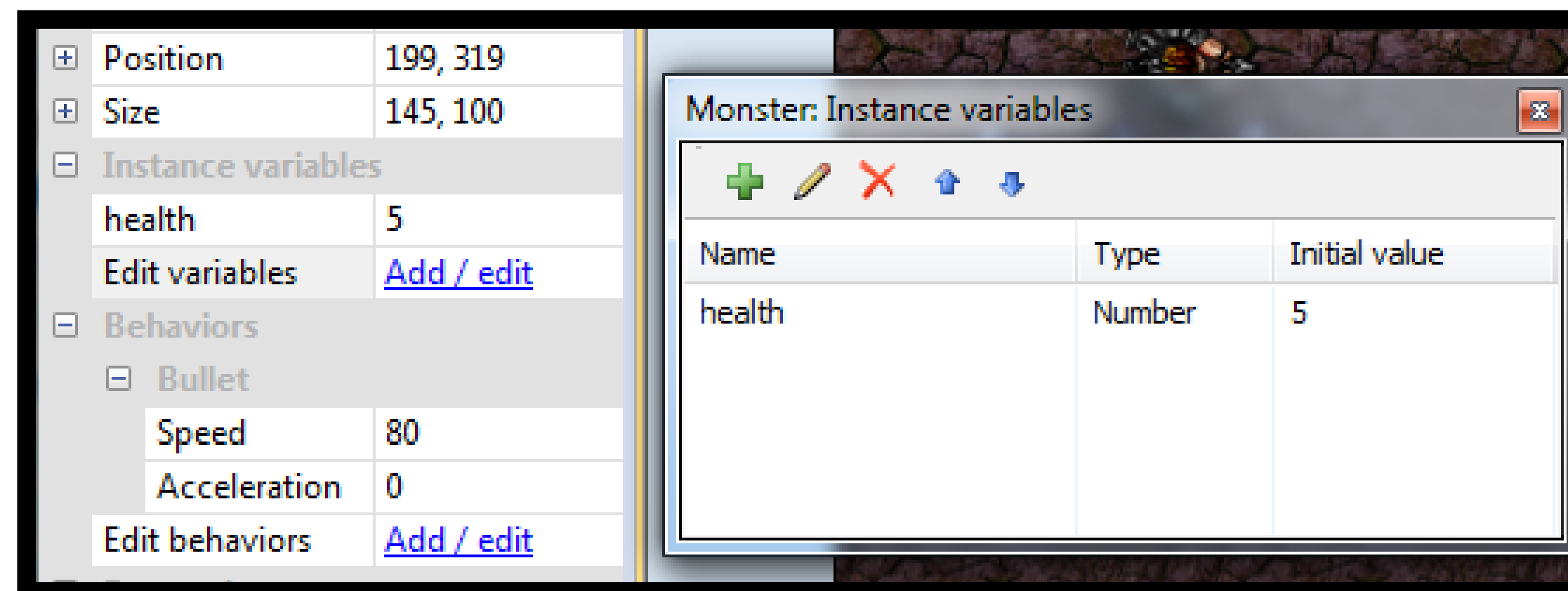
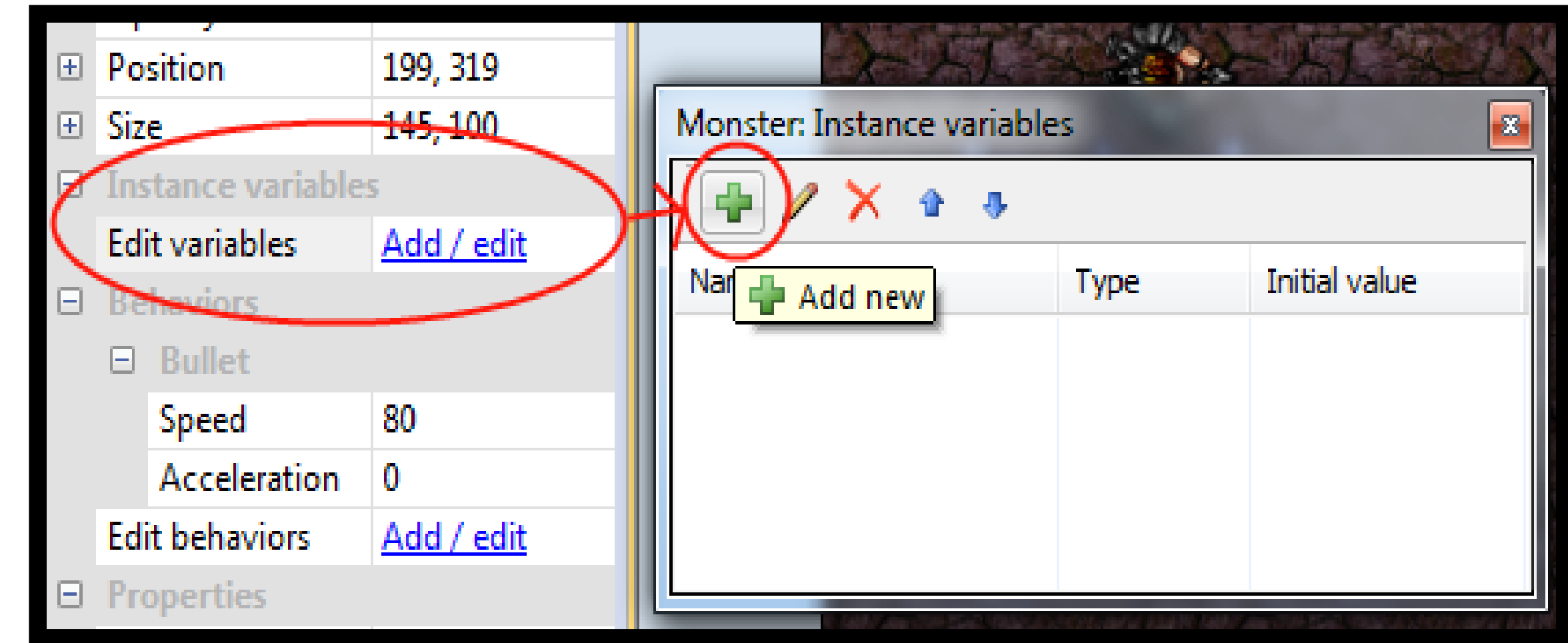
- Local variables are variables placed nested under other events, or inside a group. They also show with a different icon to global variables.
- The main difference between global and local variables is local variables can only be accessed in their scope. A local variable's scope is its level of sub-events.
- Local variables convenient for temporarily holding variables over a short range of events, such as to calculate an average value (where a temporary 'sum' variable may be necessary).
- It also helps keep the project simple, since it prevents the need to create more global variables, which appear everywhere in the project even if they are not needed everywhere.
- The scope of local variables is designed to mimic how the scope of variables works in real programming.



Local number Variable1 = 0

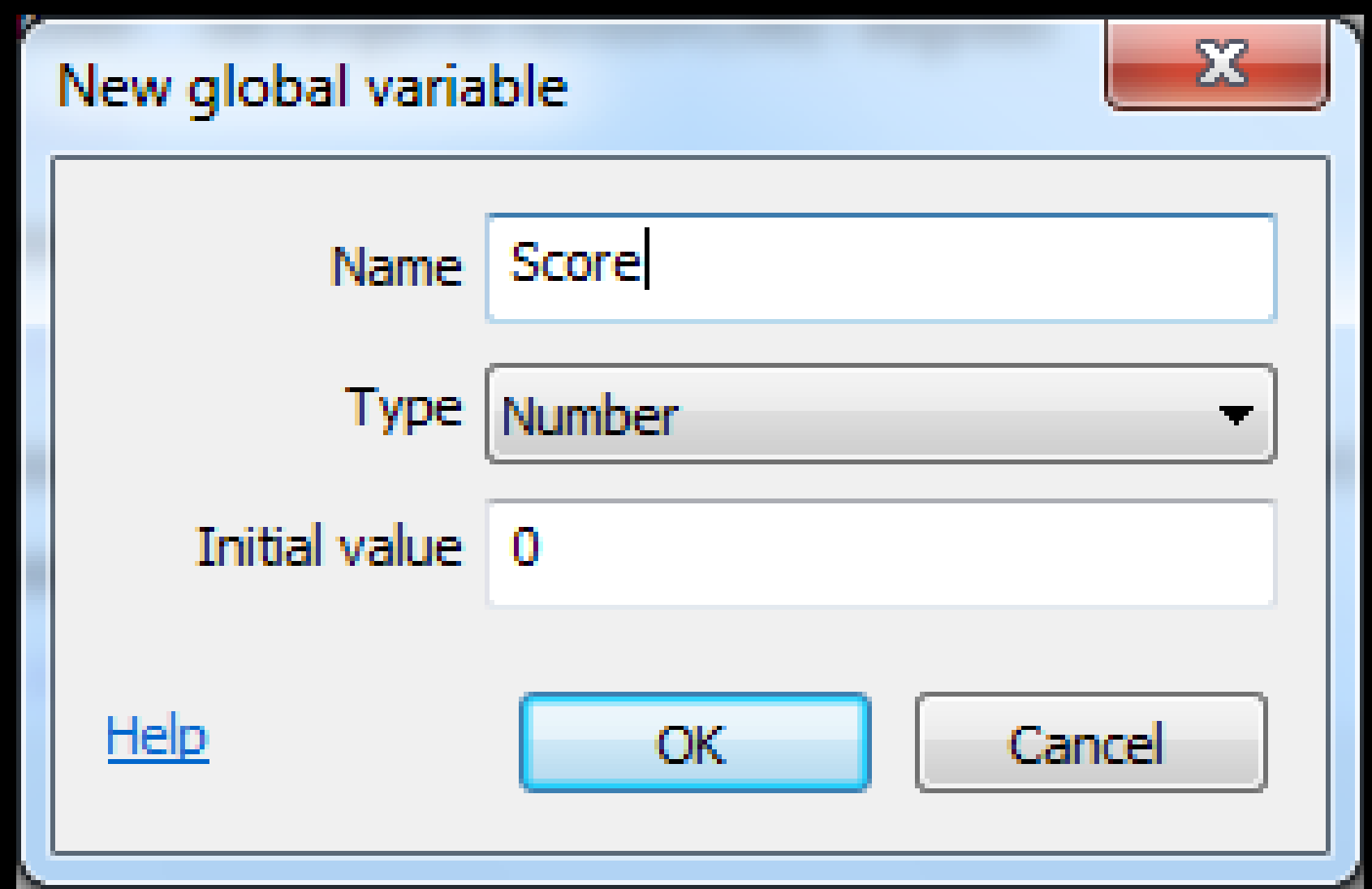
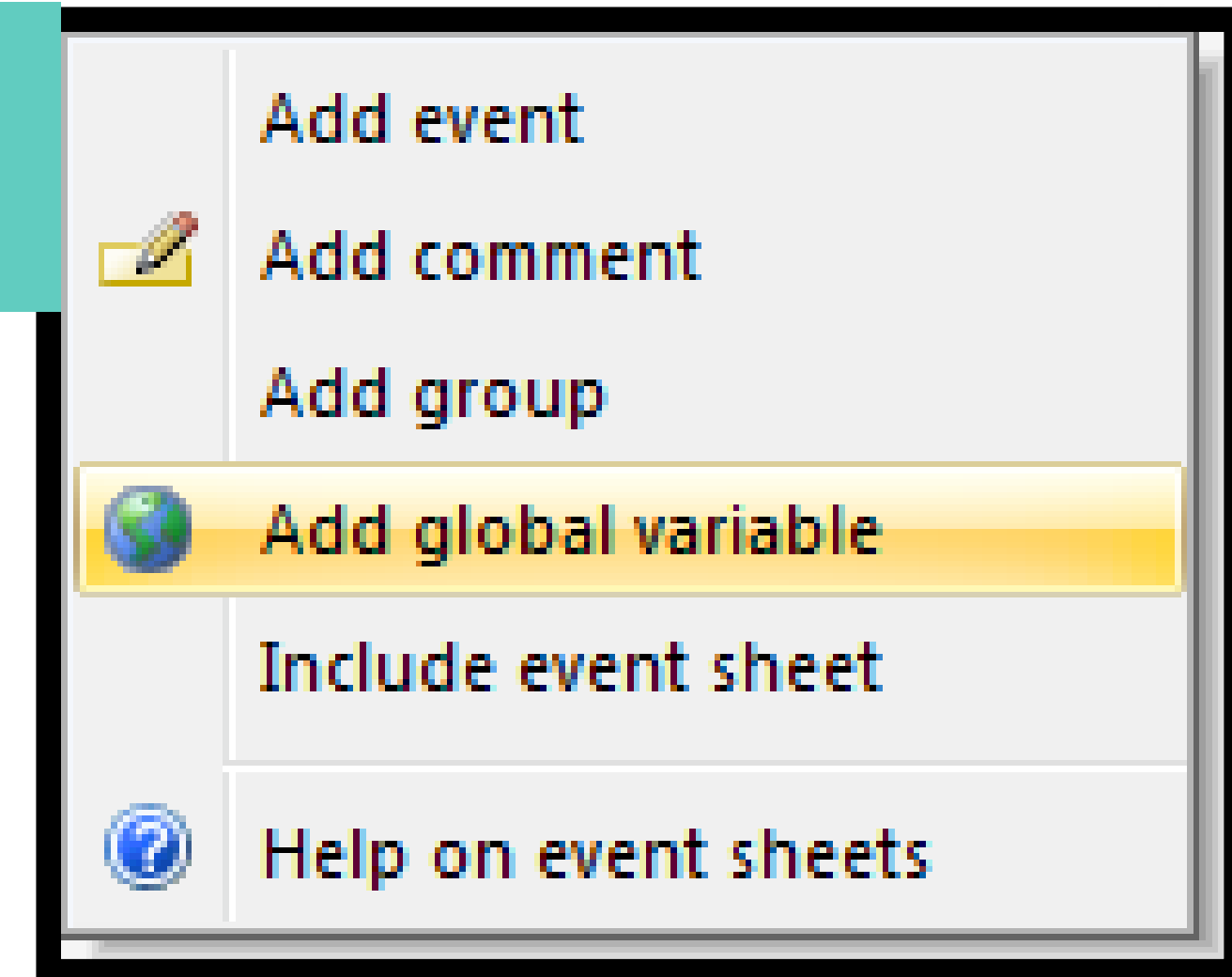
Declaring Instance variables

- Switch to the layout and select a object. This will show the object's properties in the properties bar. Click Add/edit by Edit variables.
- The Instance Variables dialog appears. It looks similar to the Behaviors dialog but instead allows you to add and change instance variables for the object.
- Click the green Add button to add a new one.
- In the dialog that pops up, type "variable name", type, and for *Initial value*.
- Once done click OK.



Declaring Global variables

- Right-click the space at the bottom of the event sheet, and select *Add global variable*.
- Enter "variable name", type & initial value. Click OK
- Now the global variable appears as a line in the event sheet. It's in this event sheet, but it can be accessed from any event sheet in any layout.

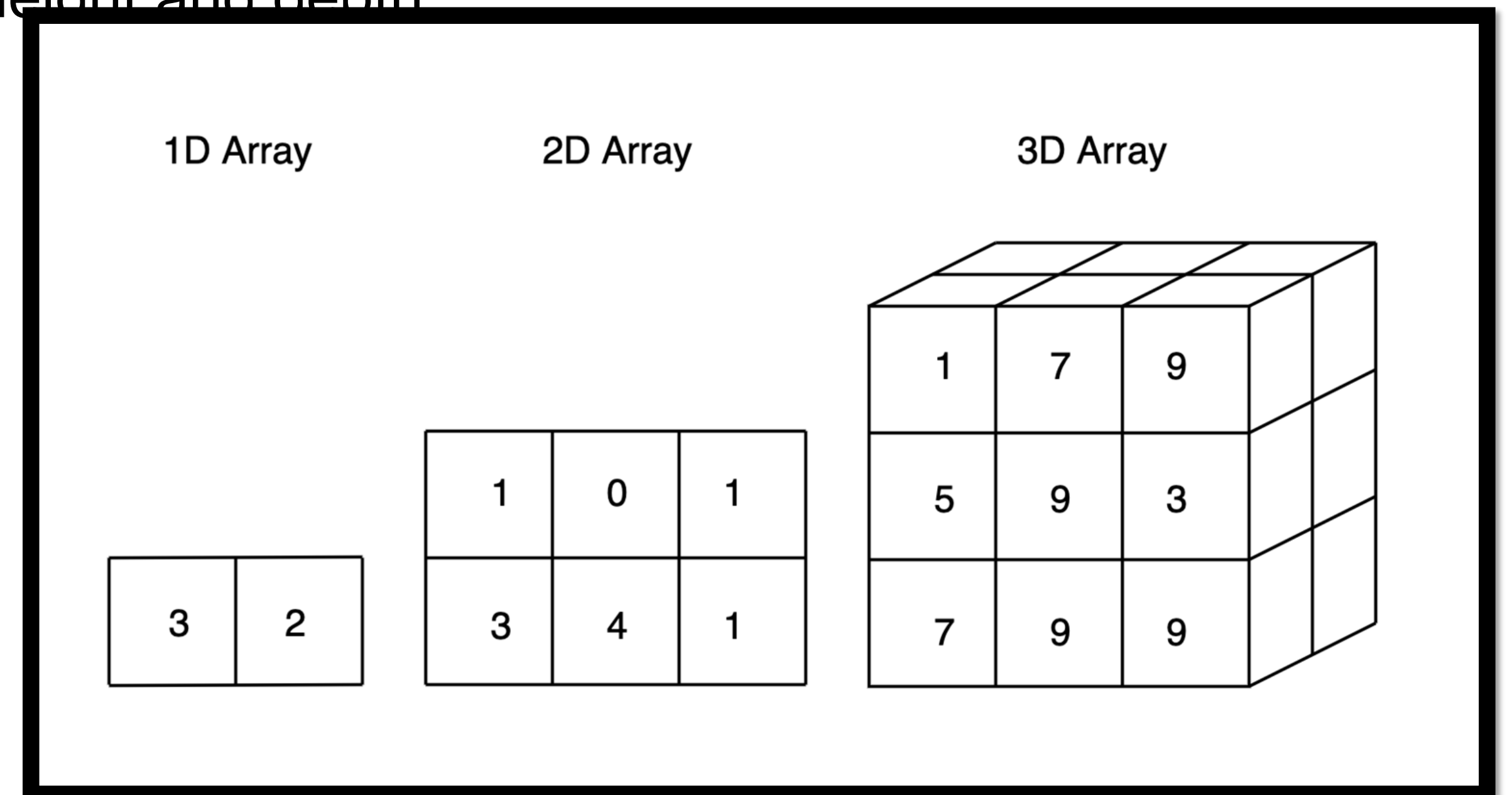


A screenshot of the 'New global variable' dialog box. It has a title bar with a close button (X). The dialog contains three input fields: 'Name' with the text 'Score', 'Type' with a dropdown menu set to 'Number', and 'Initial value' with the text '0'. At the bottom, there are three buttons: 'Help', 'OK', and 'Cancel'.

 *Global number* **Score = 0**

Array

- Array is an ordered list of cells; each cell can contain a value (a number or some text).
- An array can have 1, 2 or 3 dimensions, known as the x, y and z dimensions, also known as the width, height and depth.



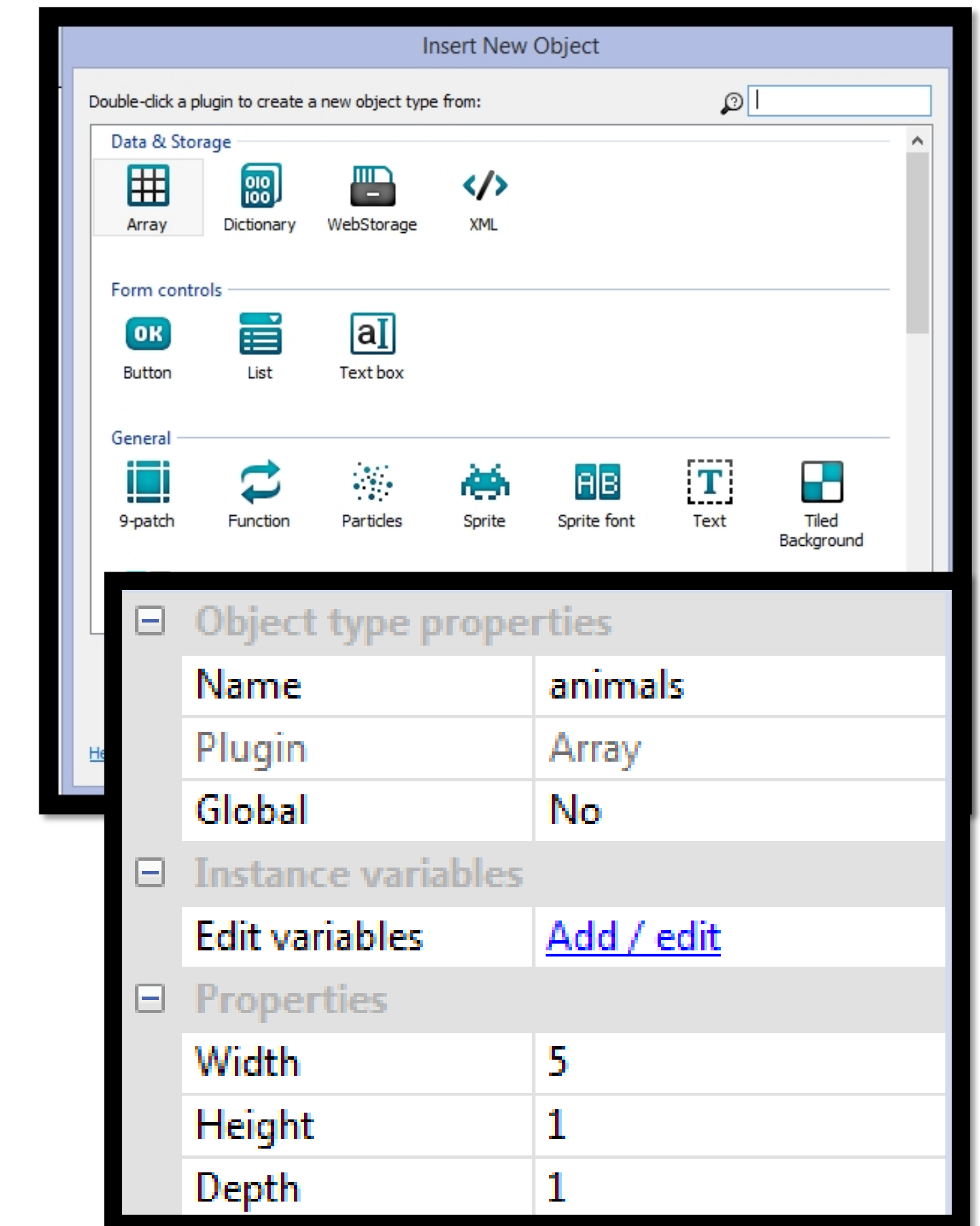
Adding an Array in Construct

- Let's create a list of five animals. Right-click or double-click on the layout and insert a new Array object. Call it "animals".
- Select the array in the objects window and let's look at its properties. The default array size is Width: 10, Height: 1, Depth: 1. We only want to store 5 animals in the list so go ahead and change the Width to 5.
- By default all elements are set to zero (0); so we can imagine the array looks like this:

- The position of a cell within the array is called its index. The indices are $x = 0$ to $x = \text{width} - 1$, where $x = 4$ is the last cell.

animals =

0	0	0	0	0
---	---	---	---	---

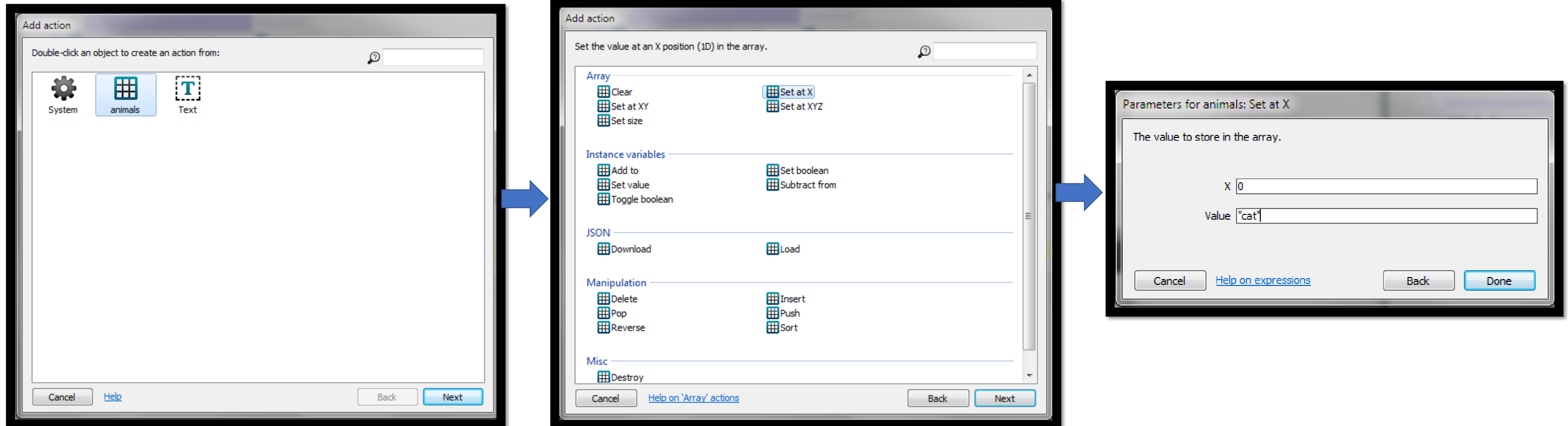


The screenshot shows the 'Insert New Object' dialog box in Construct 3. The 'Data & Storage' category is selected, and the 'Array' plugin is chosen. Below the dialog, the 'Object type properties' panel is visible, showing the following configuration for the 'animals' array object:

Object type properties	
Name	animals
Plugin	Array
Global	No
Instance variables	
Edit variables	Add / edit
Properties	
Width	5
Height	1
Depth	1


- **Setting values:**

- Let's set some values in the array. Add a "System: On start of layout" condition and the following action to set the value of the first cell to "cat".



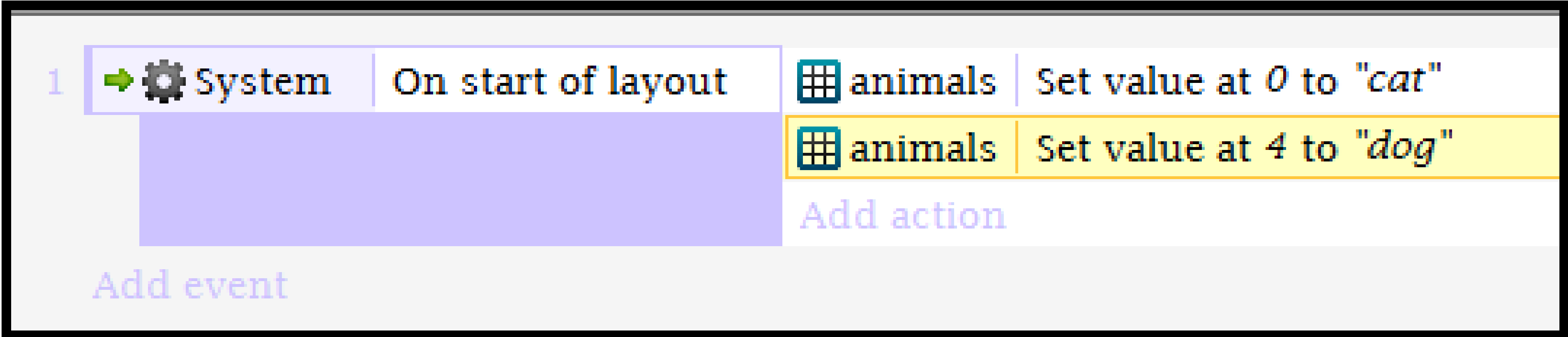
The first screenshot shows the 'Add action' dialog with 'animals' selected. The second screenshot shows the 'Set at X' action selected under the 'Array' category. The third screenshot shows the 'Parameters for animals: Set at X' dialog with 'X' set to 0 and 'Value' set to 'cat'.

- **Result:**

 animals

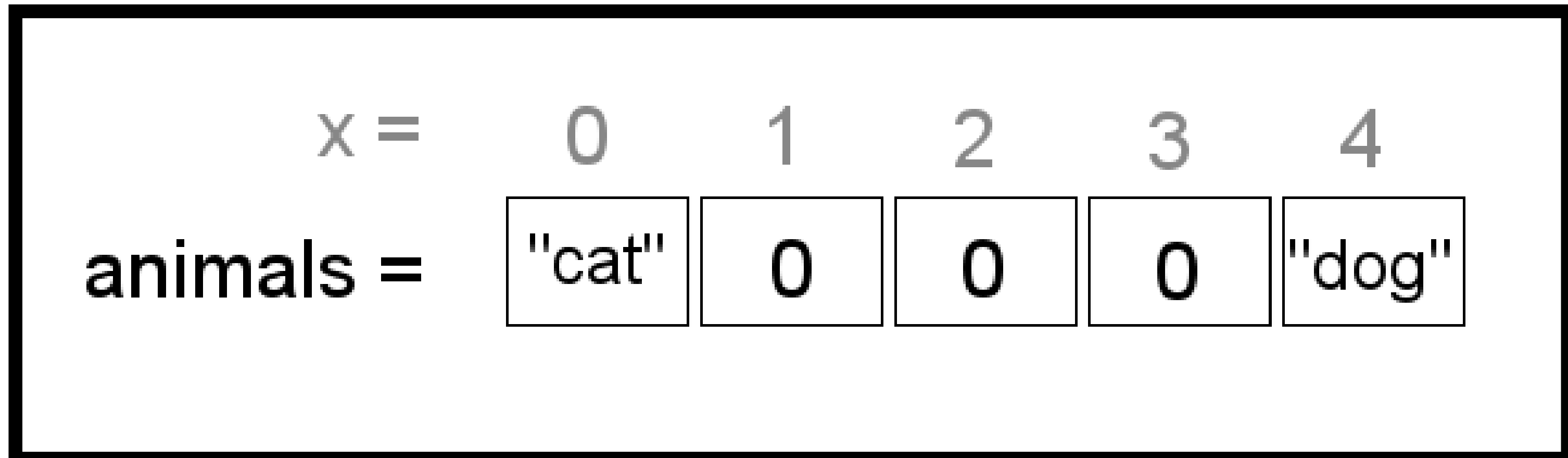
| Set value at 0 to "cat"

➤ Similarly do to set fifth cell's value to "dog":



	Event	Event Name	Target	Action
1	System	On start of layout	animals	Set value at 0 to "cat"
			animals	Set value at 4 to "dog"
				Add action
				Add event





➤ Now the array looks like this:














x =	0	1	2	3	4
animals =	"cat"	0	0	0	"dog"

Checking Array properties

- The debugger is very handy for viewing the data in your array. If we run "Debug Layout" and select the array in the debugger, it will show of all the array's data:

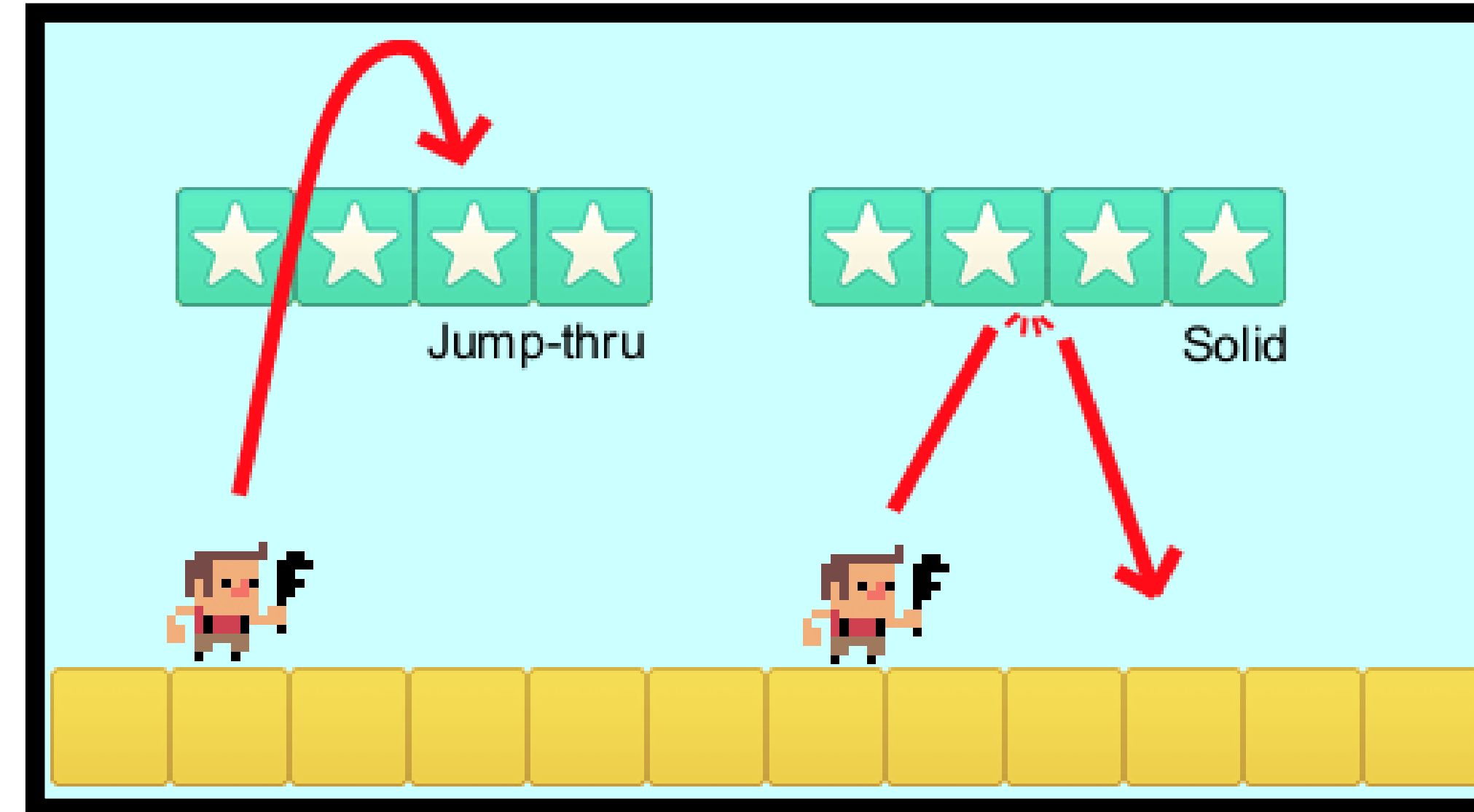
Common 	
Name	animals 
UID	0 
IID	0 

Array 	
Width	5 
Height	1 
Depth	1 
Total elements	5 

Array data 	
0	cat 
1	0 
2	0 
3	0 
4	dog 

Platform Behavior

- The Platform behavior implements a side-view "jump and run" style movement. It supports slopes, moving platforms, "jump-thru" platforms, and arbitrary angles of gravity.
- The Platform behavior will land on any objects with the Solid or Jump-thru behaviors.



Implementing Platform Behavior for animation

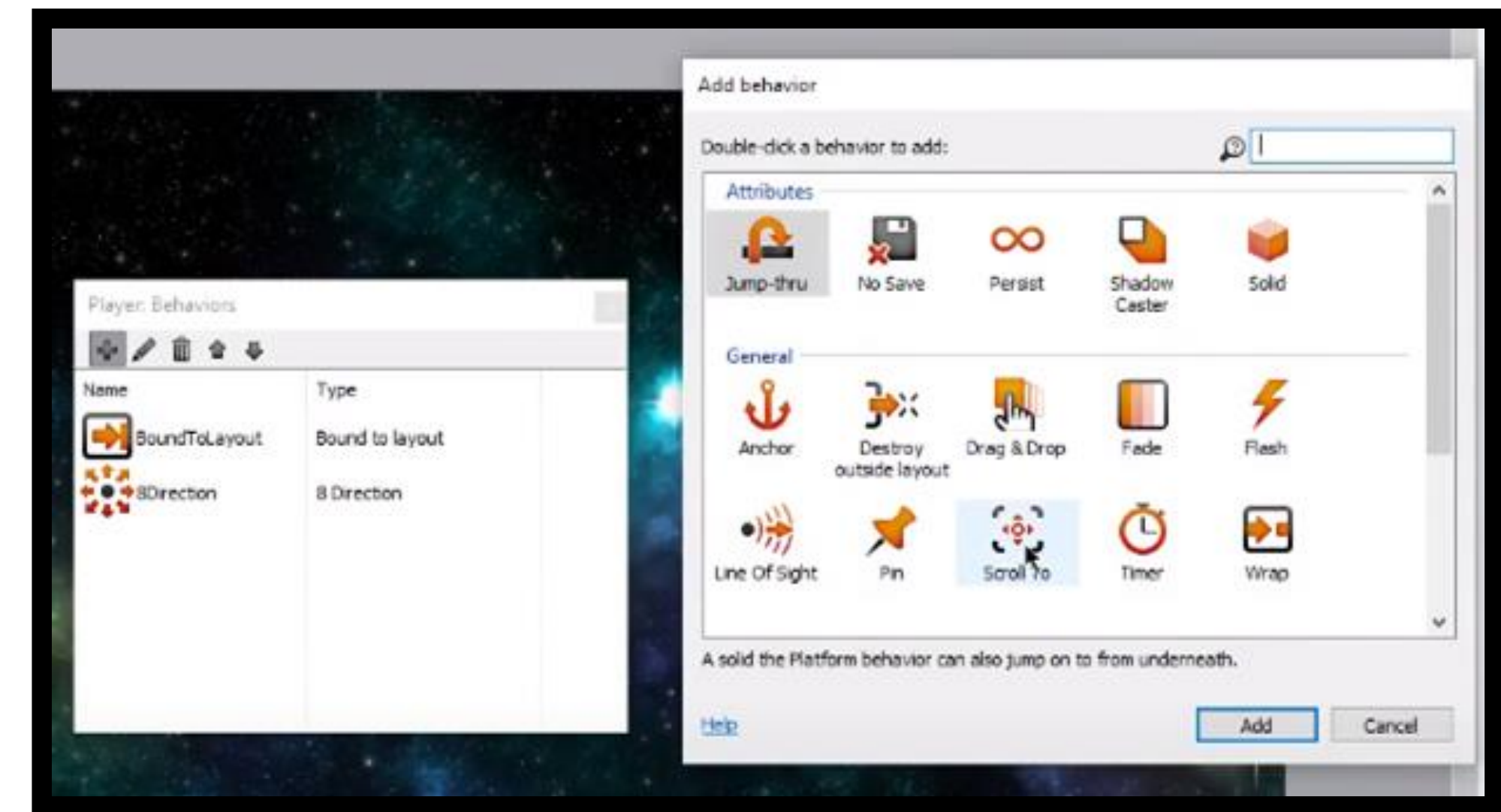
- It is recommended to use a invisible rectangle sprite with no animations with the Platform movement.
- Then, the animated player object can be positioned on top of that. Otherwise, the changing size and collision polygon of the object as its animation plays can interfere with the Platform movement's floor, wall and slope detection, causing a shaking or glitchy movement.
- **Mirroring the player:**
 - It is not necessary to duplicate all your artwork for the player for facing both left and right. Instead, simply draw all the player's artwork for facing to the right, and use the Sprite object's Set mirrored action to mirror the player's image.
 - Set the player mirrored when pressing the movement controls. An example is shown.
 - A common question is how to make the player shoot left and right, since even when mirrored the player will still shoot objects to the right. To solve this, use the Is mirrored condition, and if it is true, shoot to the left instead.

Scroll To Behavior

- The Scroll To behavior simply centers the view on the object with the behavior.
- It is a shortcut for the *Scroll to object* system action.
- However, it also provides a *Shake* action to shake the screen, and if multiple objects have the Scroll To behavior, it will center the view in between all of them.
- If we need more advanced scrolling, e.g. limited to certain regions or following the player after a delay, scroll to an invisible object which we control through events.
- To scroll, **the size of the layout must be bigger than the size of the window**, or the layout's *Unbounded scrolling* property must be set to *Yes*. Otherwise there is nowhere to scroll to and scrolling will have no effect.
- Scroll To has no properties, conditions or expressions.

Implementing Platform Behavior for animation

- Set the size of the layout bigger than the size of the window.
- Set player's behavior to Scroll To.
- Keep Initial State as Enabled from Properties bar
- Now, if we preview the layout we can see that the view follows the player staying centered on him as he explores the layout.



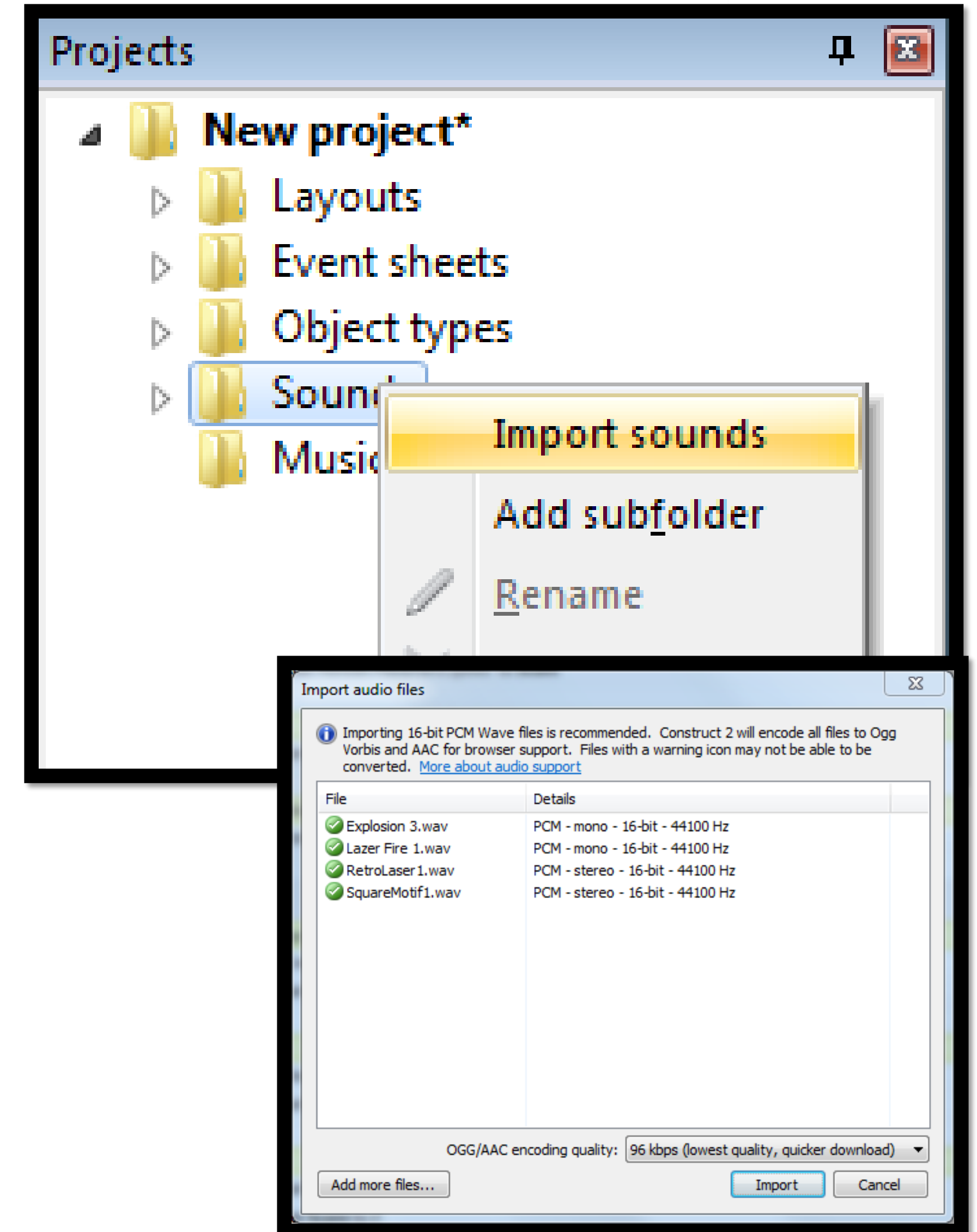
Adding Sound & music

- It takes three simple steps. :
 - Import sound files into your project.
 - Insert an Audio object to your project.
 - Add sound events in the event sheet.

Import sound files into your project:

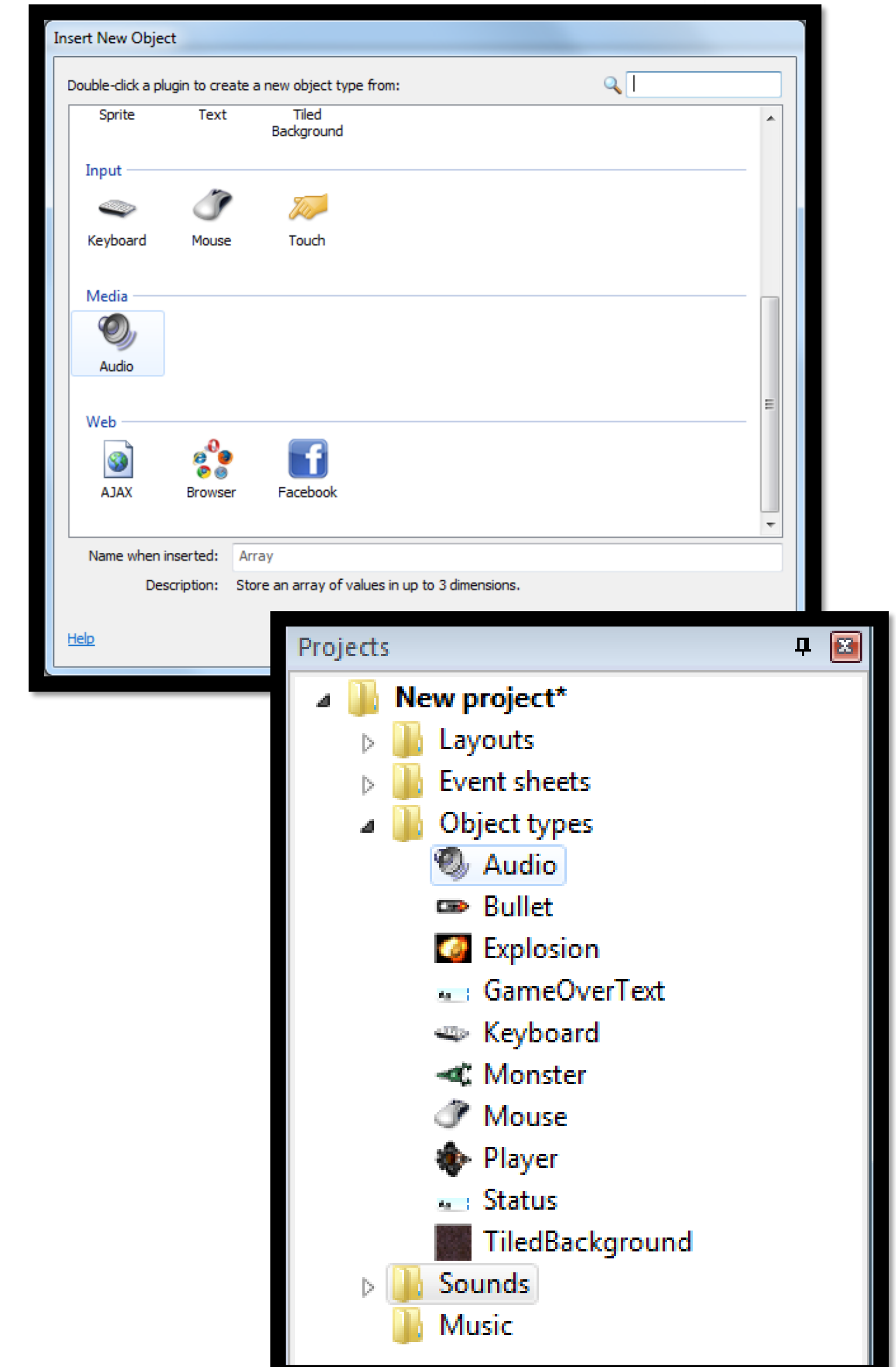
- In the Project Bar (on the right of the workspace - if it's not visible, click the Project tab at bottom right), right-click the Sounds or Music folders and select 'Import sounds' or 'Import music'.
- Select the sound files you need

Notes: Audio files in the Sounds folder are downloaded completely before they're played, while files in the Music folder are streamed (played as they're downloading).



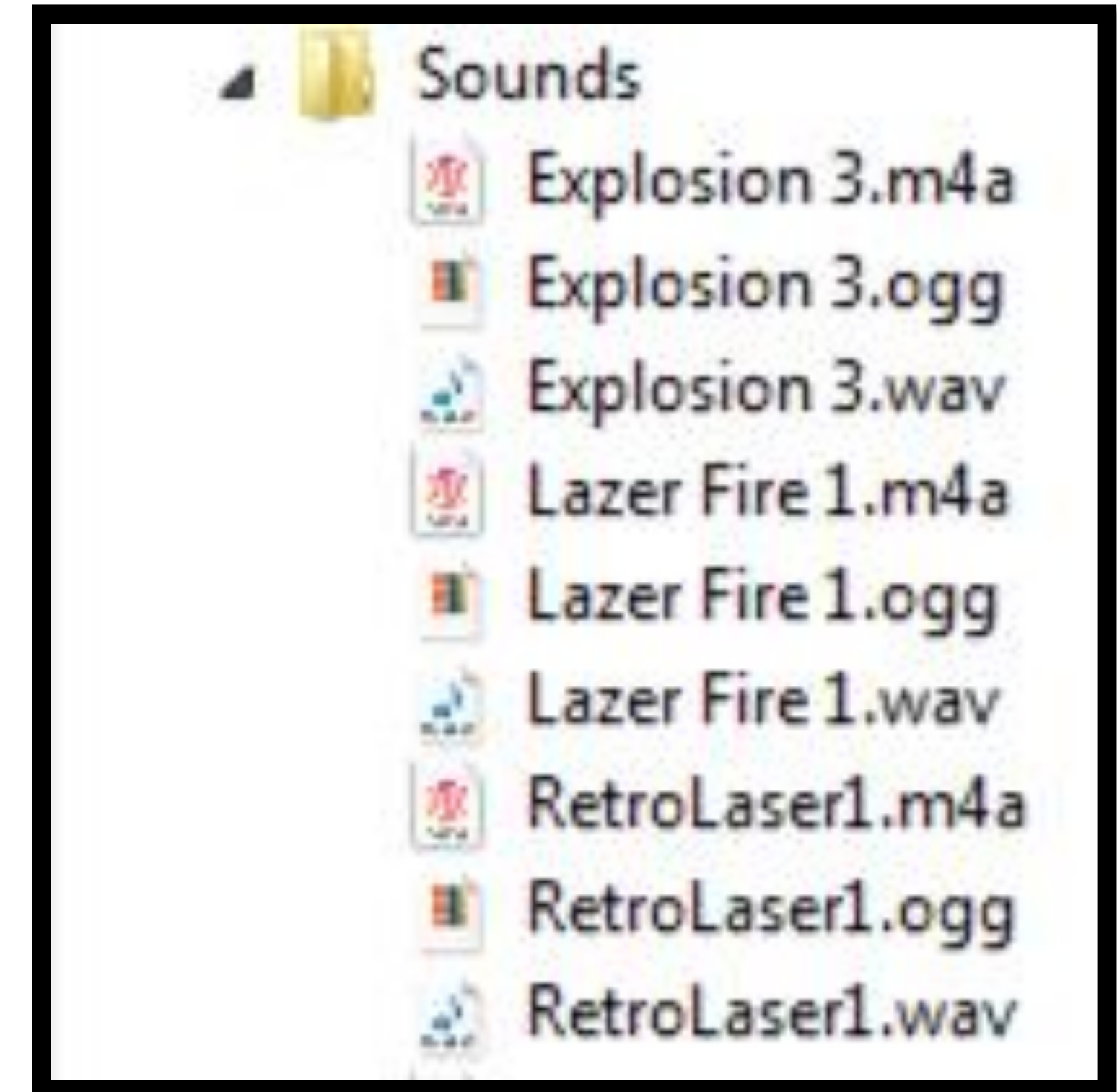
Insert an Audio object to your project

- Right-click in the layout sheet. In the dialog that opens, select 'Insert new object'.
- Under 'Media', select 'Audio' and click 'Insert'.
- In the Project Bar, an Audio object will appear under 'Object types', and its properties will appear in the Properties Bar.



Add sound events in the event sheet

- **Example:** we import the sound files we want –
 - bullet fire, an explosion for when a bullet monster is destroyed,
 - a different explosion for when your player is destroyed,
- We're taking our sound files from the free bundle download of music, sounds and sprites - for the bullets firing we'll use Lazer Fire 1.wav, for the monster destruction we'll import Explosion 3.wav, for the player destruction RetroLaser1.wav (it's got a suitably dismal quality).
- Import these and you'll see their .ogg and .m4a versions showing up in the Sounds folder in the Project Bar.



- Now, we insert an Audio object.
- Add the sound events

Shoot a bullet on left click.

3	→ Mouse	On Left button Clicked	Player	Spawn Bullet on layer 1 (image point 1)
			Audio	Play Lazer Fire 1 (tag "")
Add action				

If a bullet hits a monster, create an explosion and damage the monster. Also increase the speed monsters are spawning at.

4	→ Bullet	On collision with Monster	Bullet	Spawn Explosion on layer "Main" (image point 0)
			Explosion	Set angle to <i>random(360)</i> degrees
			Monster	Subtract 1 from health
			Bullet	Destroy
			System	Add 1 to MonsterSpeed
			Audio	Play Explosion 3 (tag "")
Add action				

Kill the player if a monster comes too close!

9	→ Monster	On collision with Player	Player	Destroy
			GameOv...	Set Visible
			Audio	Play RetroLaser1 (tag "")
Add action				

ATL GAMING SOLUTION

The ATL Gaming solution, in other terms a module, is a learning opportunity that introduces students, across the country, to an interactive and immersive platform for game development!



Aligned to UN
Sustainable Development
Goals

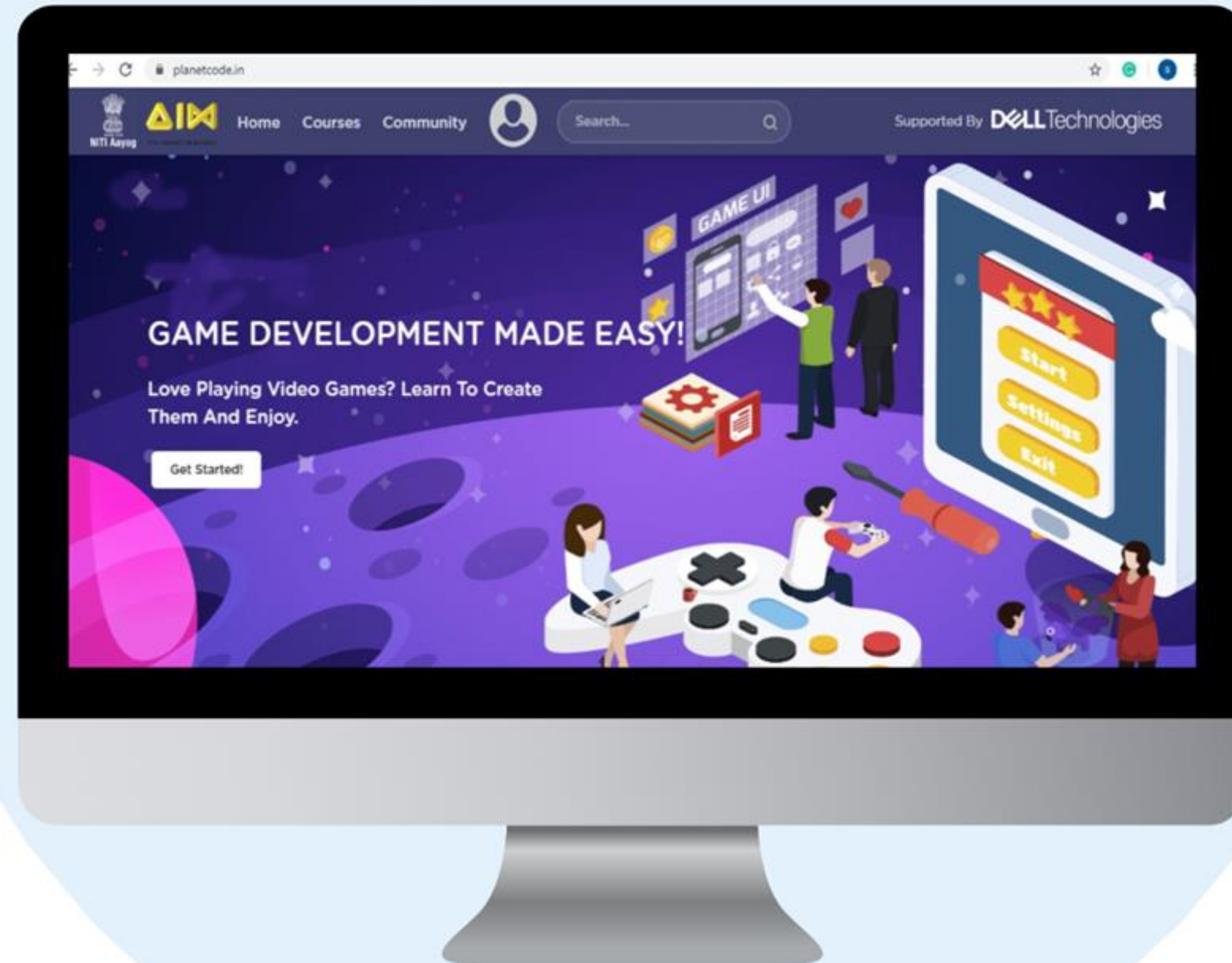
Interactive way to
Learn Programming

Online & No Age
limit

Completely FREE &
Open Source

Only requires a Tech
device & Internet

Key features of our platform





Beginner Level

*UNSDG 13
Climate Change*



Intermediate Level

*UNSDG 3
Health and Wellbeing*



Advanced Level

*UNSDG 15
Life on Land*

What courses are available for students?



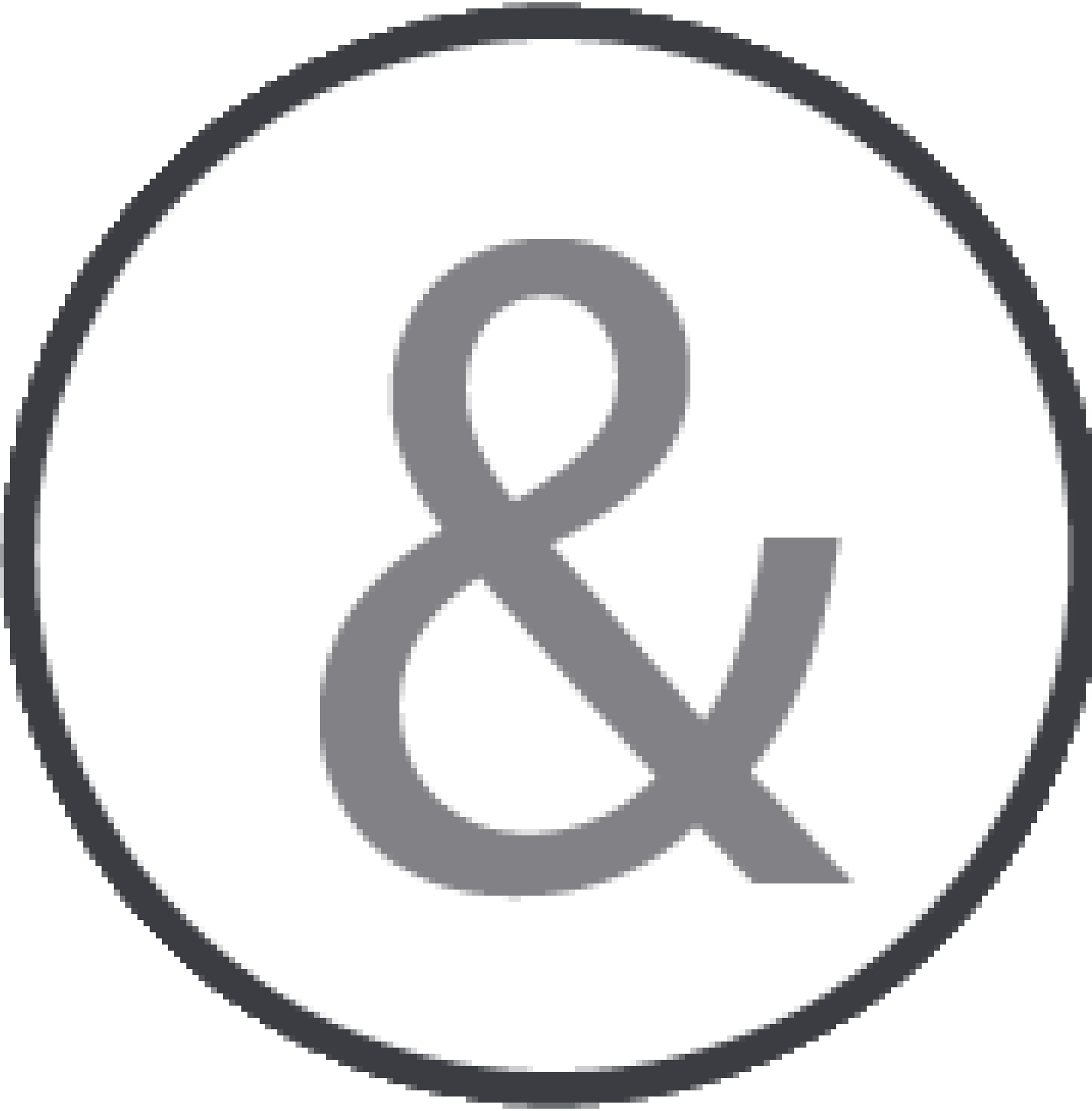
Accessing the Gaming Portal



Let's Practice - Homework of the day

Add controlling elements to the games

(Half baked game can be downloaded from <https://rb.gy/b7mkf6>)



Thank You!
For more info, please write to:

tech@learninglinksindia.org

To learn more about visit www.planetcode.in

[For Feedback visit https://rb.gy/mi3xw9](https://rb.gy/mi3xw9)