

Code the Vaccination Game Layouts – Part I

Hello Everyone! Welcome back!

In our last video, we have not only successfully created the multiple screen layouts but also added sprites to them. In this video, we will learn more about the different behaviours, plugins and objects available in Construct for game development. Also, we will begin coding the screens and the sprites associated with them to make our game functional.

Before we proceed any further with coding sprites, let's try to understand in detail the 'Behaviour' feature provided by Construct. The first in this list is '8 Direction' behaviour. This behaviour allows an object to be moved up, down, left, right, diagonally, and is controlled by the arrow keys. It can also be set to 2 directions as in the Ping Pong game.

Next is the 'Car' behaviour. This behaviour allows an object to accelerate in both forward and backward direction and also gives a feeling of using a steering wheel just like you must have experienced in a bike or car racing game. It also has a simple "drift" feature which helps the object to skid around corners.

The 'Platform' behaviour facilitates a side-view 'Jump and Run' style movement. It supports slopes, moving platforms, "jump-thru" platforms and arbitrary angles of gravity just like you have seen in games like Mario.

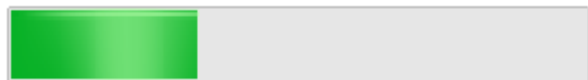
The 'Drag & Drop' behaviour enables objects to be dragged and dropped using either the mouse or touch. On multi-touch devices such as trackpads, touch screen smart phones, game consoles etc....., multiple objects can be dragged and dropped at once. An object can be dragged on click of the mouse or when the touch falls inside the object's collision polygon. It is dropped when the mouse button is released or the touch ends.

To explore all the behaviours of Construct 2, you can visit the link shown on the screen - <https://www.scirra.com/manual/86/behavior-reference>

After learning about the behaviour feature, we need to understand about the Audio plugin features which are available in construct. **Audio** object plays those audio files that have been imported onto the project. It must be inserted into each project that requires an audio playback. Audio files can be imported to a project by right-clicking the Sounds or Music folders in the Project Bar and selecting Import..., which brings up the Import Audio dialog.

One important object in the input section is the **Gamepad** object which allows you to detect input with the help of a connected console controller such as the PC gamepad or joystick. While a range of devices can work well with the Gamepad object, it works best with the XBox 360 controller, or similar other controllers.

In much the same way, the Progress bar object in the input section displays a bar which can be used to indicate the progress of a long-running operation or goal.



The **Local Storage** is another plugin that can **store data** locally on the user's device. For example, it can be used to store the personal best high-score.

To explore other Plugins in detail you can visit the link shown on the screen - <https://www.scirra.com/manual/106/plugin-for-reference>

Now let's begin coding the layouts...

Just like we had created layouts earlier, we will start by coding the home screen layout. We know that, we have one Play button and one Rule button sprite positioned on the Home screen. The Rule button will be used to go to the Rule screen and the Play button to go to the Game screen. To make the Play button functional, we need to add a touch function to it so that the required function gets performed once you touch or click on the button. To carry this out, go to the 'Home Screen' layout and double-click on the screen, scroll down to select 'Touch' option from the Input section and click on 'Insert'. The touch function automatically gets applied to all other screen layouts. So, in future you don't have to add this function individually to the other screen layouts.

Now go to the 'Home Screen Code' sheet. Left click on "Add event", select "Touch" and click on next. Select "On tap object" from Gesture Section and click on done. Then, click on 'click to choose' to select the object. Choose 'PlayButton' as an object and click on done. This will make your button clickable.

Now to make this click look more realistic we have to add an effect called as the "Button effect". We'll follow the same steps for creating Button Effect as done in our last game. We would recommend you to download the video description to follow the step by step process, if forgotten.

For adding Button Effect to Play Button:

Left-click on 'Add Action' > Select 'PlayButton' sprite – Next > Select "Set Scale" from size and position section – Next > Set scale as per your choice (We are setting it to 0.5)

Again left-click on 'Add Action' > Select 'System' option – Next > Select "Wait" from Time section - Next > Set time as 0.2 seconds

Again left-click on 'Add Action' > Select 'PlayButton' sprite – Next > Select "Set Scale" from size and position section – Next > Set scale as per your choice (We are setting it to 1)

Again left-click on 'Add Action' > Select 'System' option – Next > Select "Wait" from Time section - Next > Set time as 0.4 seconds

This action creates the "Button Effect".

Now let's look at the final part of programming the Play Button which will take you to the Game Screen layout and the Rule button to go to the Rule screen. Again follow the same process as followed in our last game. Feel free to follow the detailed steps from the video description.

To code the Play Button:

Left-click on 'Add Action' > Select 'System' option – Next > Select "Go to layout" from General section - Next > Select "Game Screen" layout from the drop down – Done

To code the Rule Button:

Left-click on 'Add event' > Select 'RuleButton' sprite – Next > Select 'On clicked' option from Button section - Done

Left-click on 'Add Action' > Select 'System' option – Next > Select "Go to layout" from General section - Next > Select "Rule Screen" layout from the drop down – Done

Let's do a quick check and see if all the events and actions added to the buttons are working correctly.

Great! We have successfully coded the Home screen. In our next video, we will begin coding the sprites present in the Game Screen layout of our game.

Bye for now. See you again in the next video!