Code the Screen Layouts – Part IV

Hello everyone! Welcome back to the Intermediate course on game development!

In our last video, we continued to code the objects present in our Main Screen layout and were nearing completion. In this video, we will finish coding for the remaining objects present in the main screen. So, let's continue to do so....

If you recall, every sprite in the layout was carefully coded so that the game progresses flawlessly and holds the player's interest. However, the medicine box showed no special effects when it crossed the path of pathogens or came in contact with them. Further, we also need to set the timer so that the game gets over in 120 seconds.

Let us begin by setting the timer. For this, click on "Add event", select "System" option from the object window and click next. A new window will pop-up. Select 'Compare variable' option from global & local variables section and click next. You will now find a new window opening up to set the parameter for timer. From the variable section's drop down menu, select 'Timer' option and set the comparison to < Equal to > and Value 0. Then click on done. We will now add action to it. Click on "Add Action" and select "System" object and then click on next. Select "Go to layout" option from the general section and click on next. A new pop-up window will appear. From the drop down menu, select 'Game Over Screen' option.

Next, we would want to create an event in which the "Health" of the medicine box diminishes every time a pathogen comes in contact with it. This increases the difficulty in the game and also safeguards the medicine box while the game is being played.

To accomplish this, click on “Add event”, select “MedicineBox” option from the object window and click next. From the new window that pops-up, scroll down to select ‘On collision with another object’ option from collisions section and click next. Another new window will appear. Select the ‘click on choose’ option and then ‘Bacteria’, and click on done. Use the same logic for protozoa, virus & worm. So, let’s use of the Make Or block before we define the logic. In order to do that, right click on the ‘green arrow’ and select ‘Make Or Block’ from the sub-menu. You’ll see the whole event highlighted in yellow. Next, right-click on the MedicineBox image and select ‘Add another condition’ from the sub-menu to open the object window. Select “MedicineBox” option from the object window and click next. A new window pops-up again, scroll down to select ‘On collision with another object’ option from collisions section and click next. Another new window appears. Click on ‘click on choose’ option and select ‘Protozoa’, then click on done. Now, you can see that we have been able to add MedicineBox-Protozoa collision event using the same logic.

Repeat the same process for collision of MedicineBox with virus and worm.

For collision with Virus, right-click on the MedicineBox image and select ‘Add another condition’ from the sub-menu to open the object window. Select “MedicineBox” option from the object window and click next. A new window will pop-up, scroll down to select ‘On collision with another object’ option from collisions section and click next. A new window will appear again. Click on ‘click on choose’ option and select ‘Virus’, then click on done. Lastly, for collision with Worm, right-click on the MedicineBox image and select ‘Add another condition’ from the sub-menu to open the object window. Select “MedicineBox” option from the object window and click next. Another new window will pop-up, scroll down to select ‘On collision with another object’ option from collisions section and click next. You will see a new window appearing again. Click on ‘click on choose’ option and select ‘Worm’, then click on done.

Next, click on “Add Action”, select “System” object and click on next. Select “Subtract from” option from the Global and local variables section and click on next. A new window will pop-up. From the drop down, select the variable Health, change the value to 1, then click on done. Again, click on “Add Action”, select ‘Health’ object and click on next. A new window will appear, scroll down and select “Set text” option from the Text section and click on next. You will see a new pop-up window, type “Box Health - “&Health in the box. Use the global variable health and not the text one. This code is like an equation which will show the health of the medicine box on the screen. Again, click on “Add Action” and select “MedicineBox” object and click on next. Select “Set visible” option from the Appearance section and click on next for a new window to appear. Change the visibility from Visible to Invisible and click on done. Then click on “Add Action” again, select “System” object and click on next. Select “Wait” option from the Time section and click on next. A new window will pop-up again. Change the value from 1 to 0.1 second and click on done. Again, click on “Add Action” and select “MedicineBox” object and click on next. Select “Set visible” option from the Appearance section and click on next. A new window will appear and keep the visibility to Visible only and click on done. Lastly, click on “Add Action” and select “MedicineBox” object and click on next. Select “Set position” option from the Size & Position section and click on next. Set the value of X-position to 399 and Y-position to 383 in the new window that appears. However, you can decide the X & Y position for the medicine box as per your requirement. Click on done.

From the above action, every time the Medicine Box collides with any pathogen, the health of the medicine box gets reduced by 1 and the updated health gets displayed on the screen as an after effect. The medicine box will once again be set to the base position of (399, 383).

To further increase the complexity of the game, let’s consider having only 6 lifelines for the medicine box. As soon as all of them get exhausted, the game will get over. To do that, click on “Add event”, select “System” option from the object window and click next. A new pop-up window appears. Select ‘Compare variable’ option from global & local variables section and click next. A new window will pop-up again to set the parameter for Health of Medicine Box. From the variable section’s drop down menu, select ‘Health’ option and set the comparison to ’ = ‘ and Value 0. Click on done. We will now add action to it. Click on “Add Action” and select “System” object and click on next. Select “Go to layout” option from the general section and click on next. A new pop-up window will appear. From the drop down in layout, select ‘Game Over Screen’ option and click on done.

Now that we are done with this, let’s run the layout of our game to see if everything is working as we wanted. Great! I can see that the game is almost ready. In our next video, we will continue coding the next game screens of our game.

So, see you next time! Till then, Bye - Bye!