

9. User Interface (UI)

Series

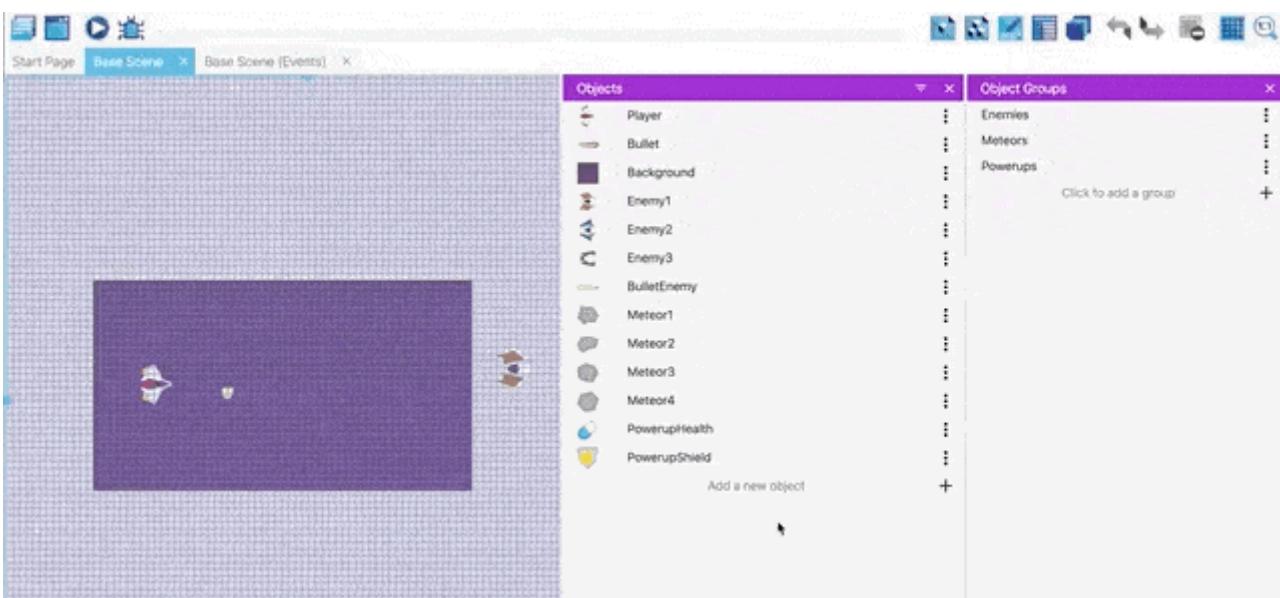
You are reading **Part 9** of the [Space Shooter Tutorial](#).

1. [Space Shooter, Part 1](#)
2. [Space Shooter, Part 2](#)
3. [Space Shooter, Part 3](#)
4. [Space Shooter, Part 4](#)
5. [Space Shooter, Part 5](#)
6. [Space Shooter, Part 6](#)
7. [Space Shooter, Part 7](#)
8. [Space Shooter, Part 8](#)
9. [Space Shooter, Part 9](#)
10. [Space Shooter, Part 10](#)
11. [Space Shooter, Part 11](#)
12. [Space Shooter, Part 12](#)
13. [Space Shooter, Part 13](#)

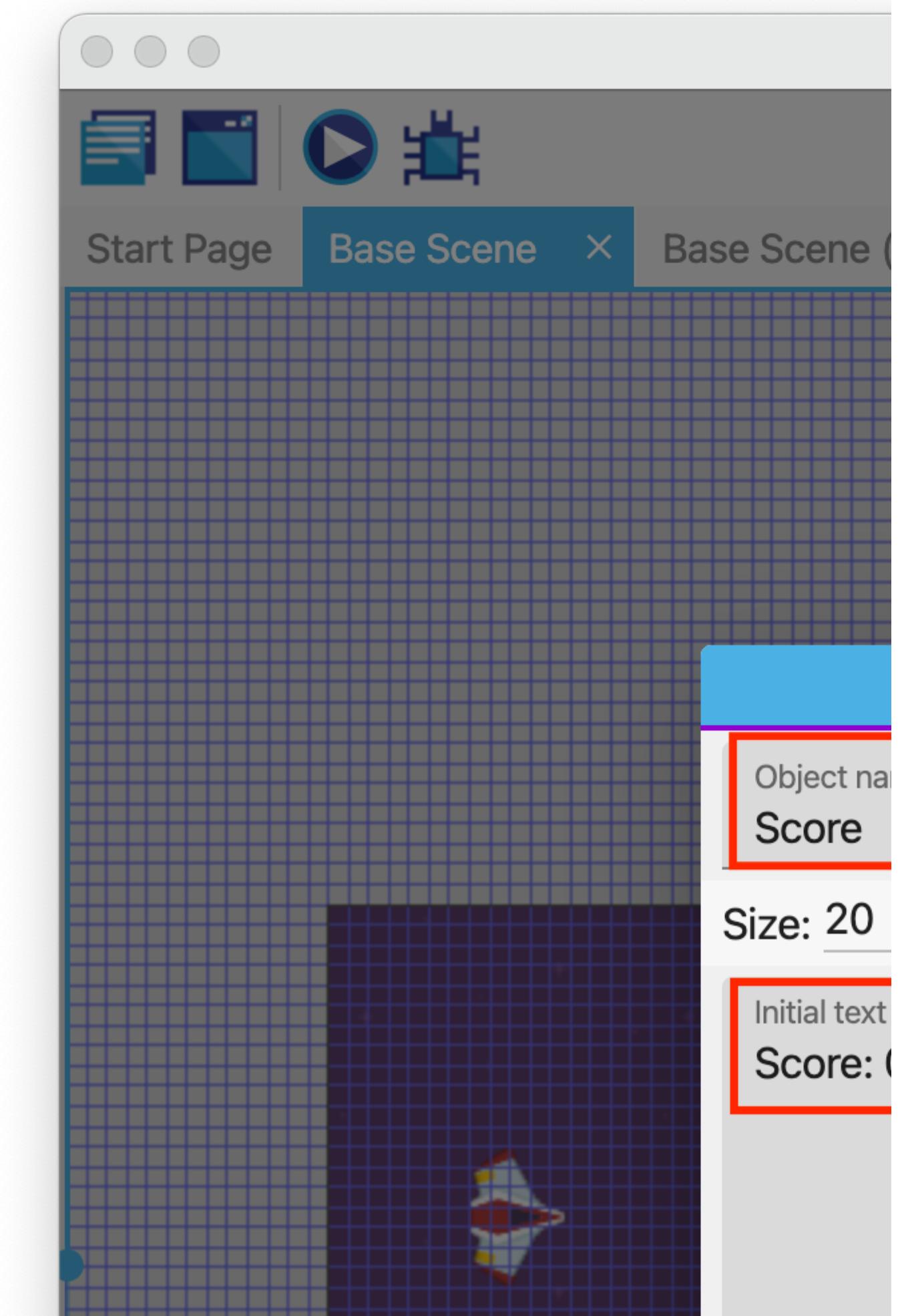
In this chapter, we will learn how to add UI elements (Text) to the game.

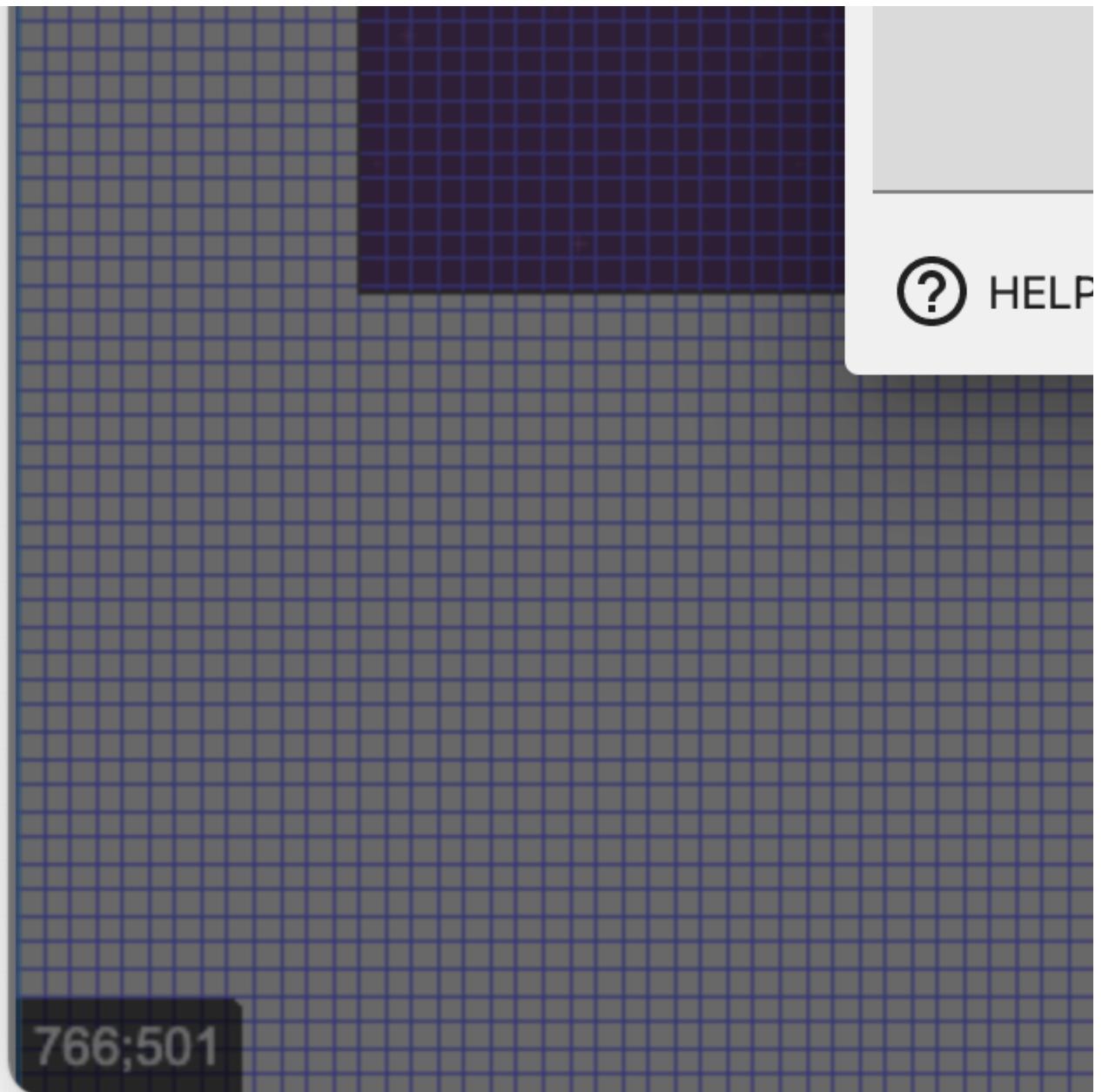
Adding UI elements

We will add two text objects to indicate the **player's health** and the **score of the game**. Firstly, create a new object, and select **Text** in the pop-up menu. Rename object as “**Score**,” type “**Score: 0**” in the text field. Then, add a custom font called “**Kenney Future Narrow.tff**” from the assets folder.

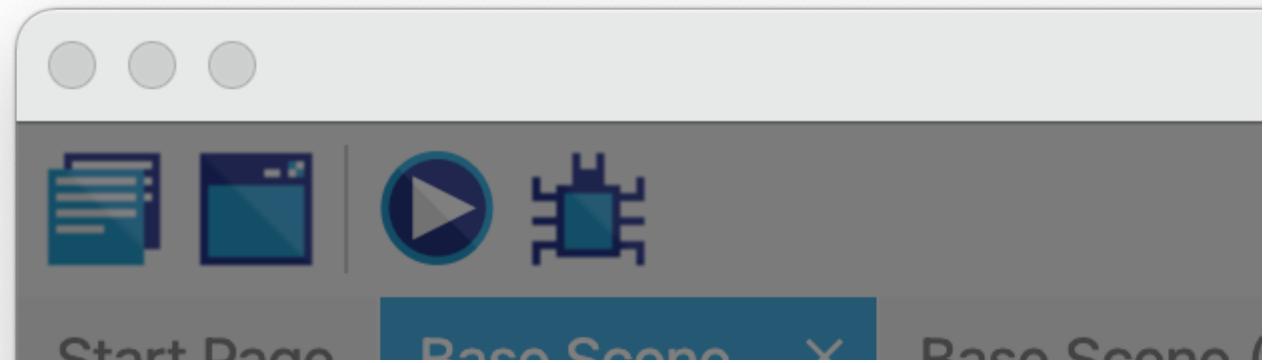


We will use yellow for the score text, but you can use any color you want.





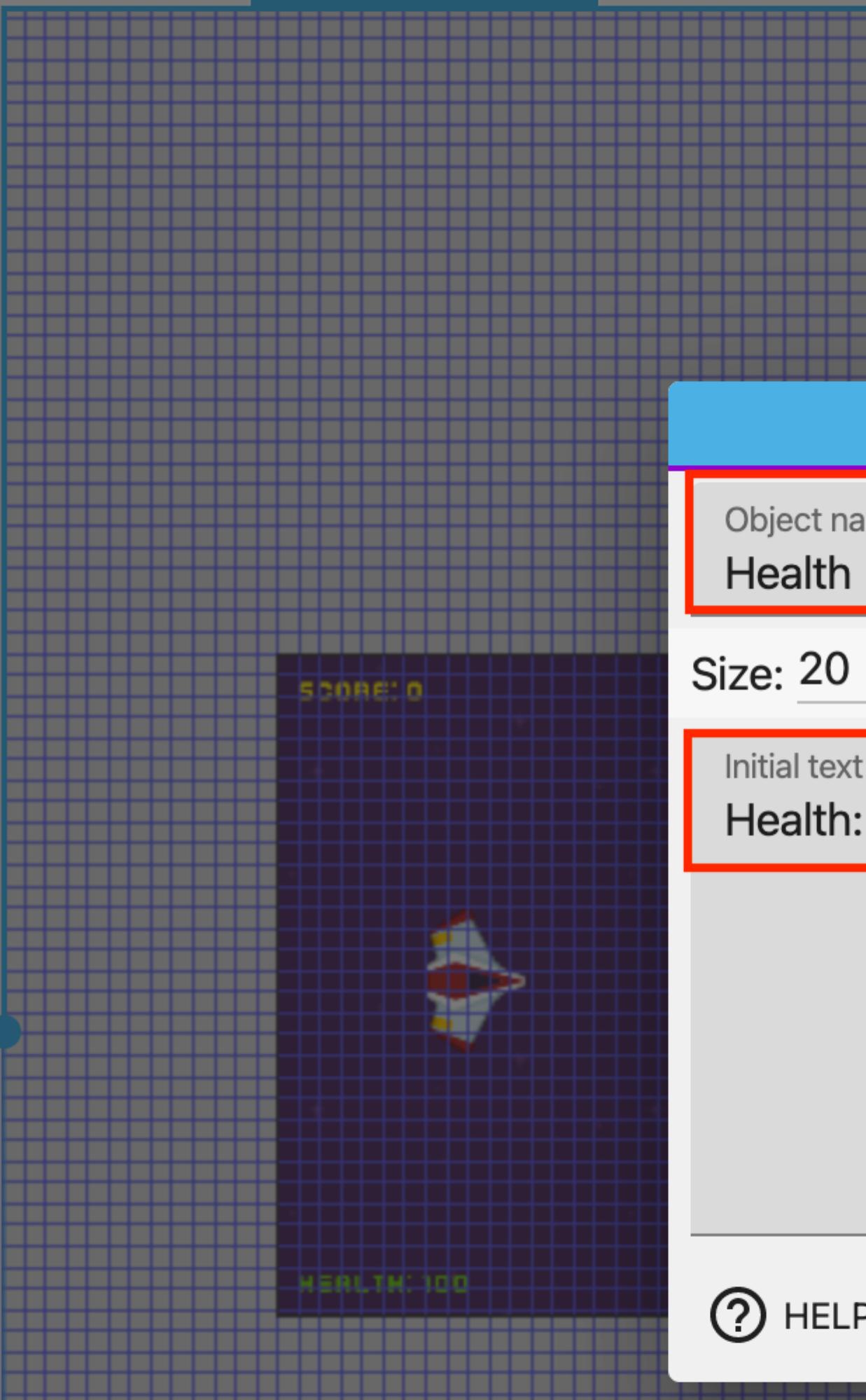
Now, we will repeat the same process to add a text for the player's health. Ensure that the object's name is "**Health**" and the font is the same as the previous step. Your settings should look like this:

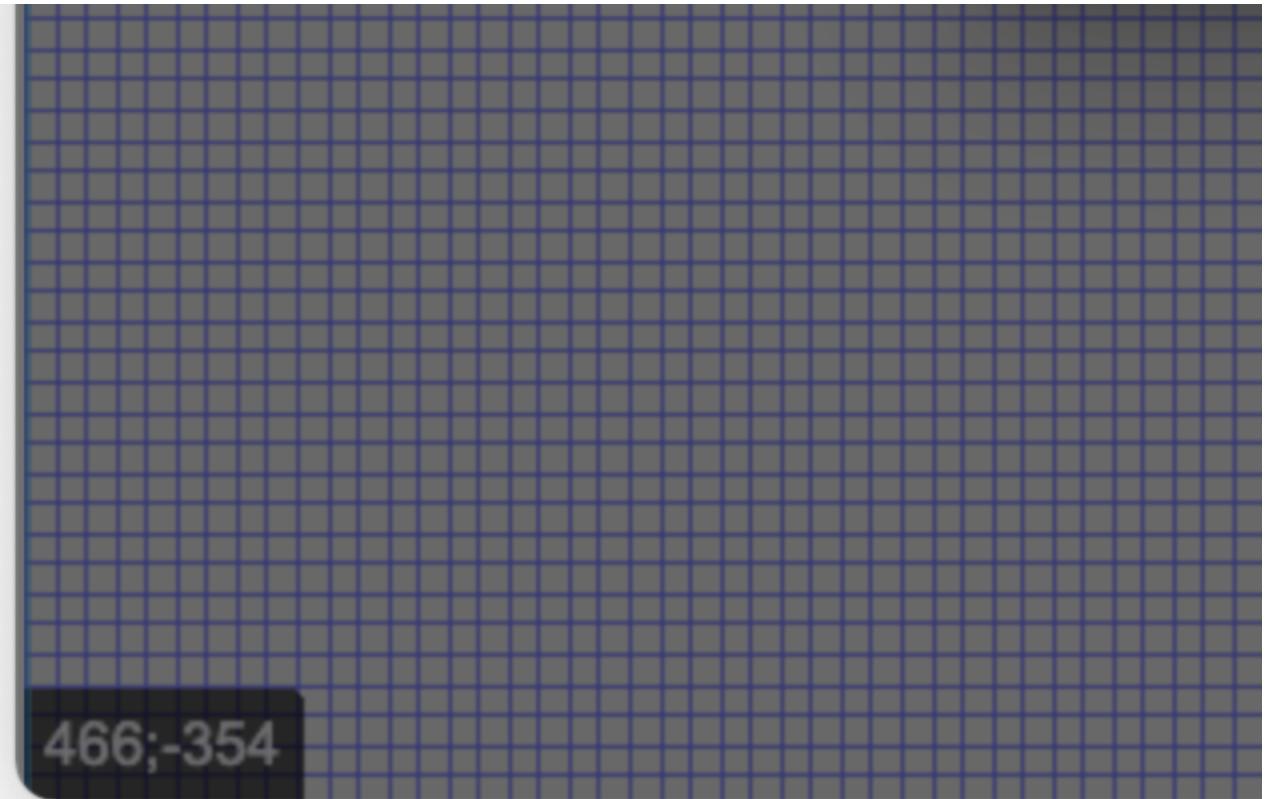


Start Page

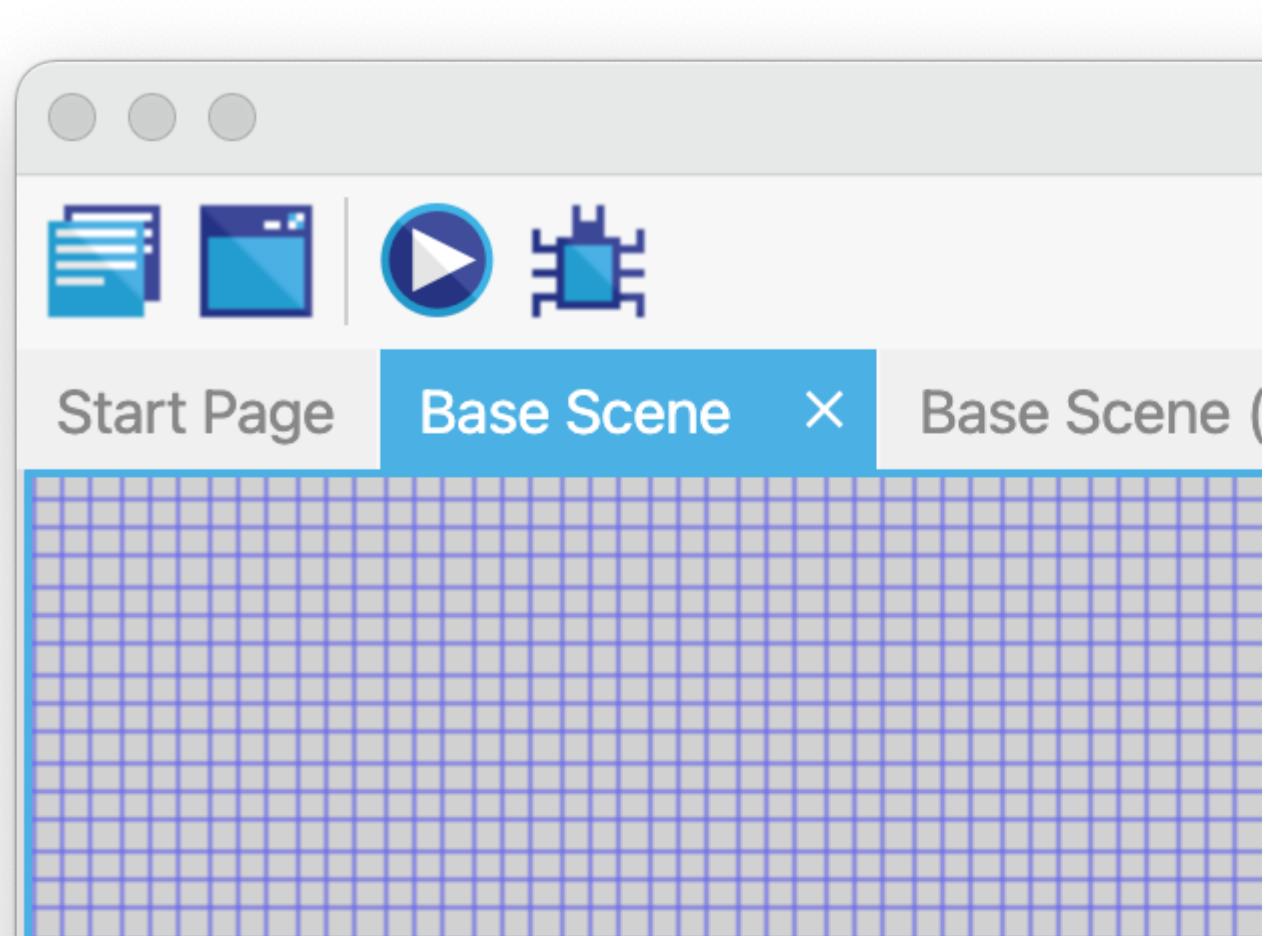
Base Scene

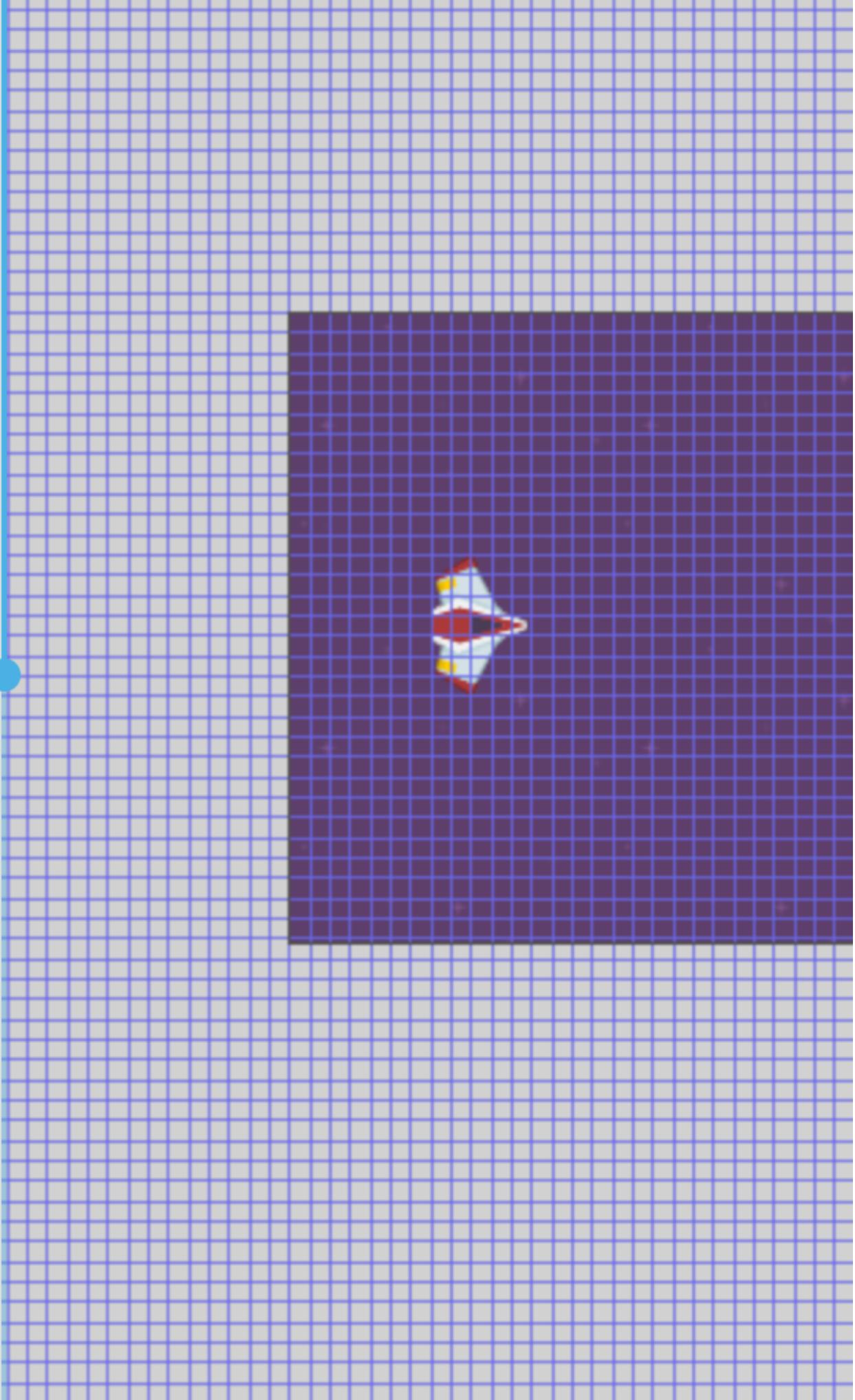
Base Scene (

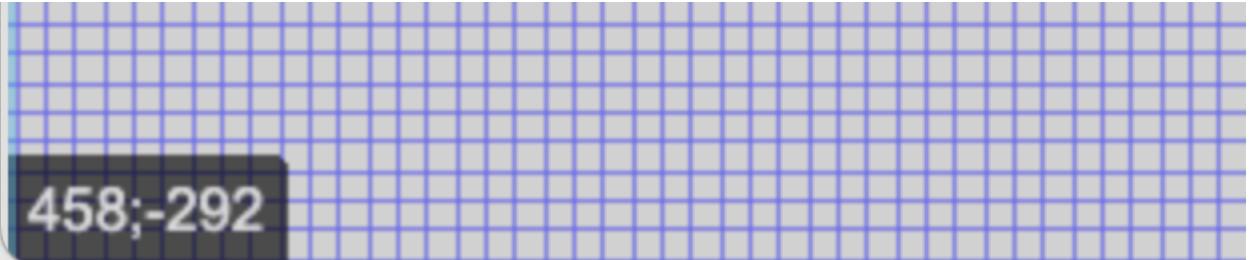




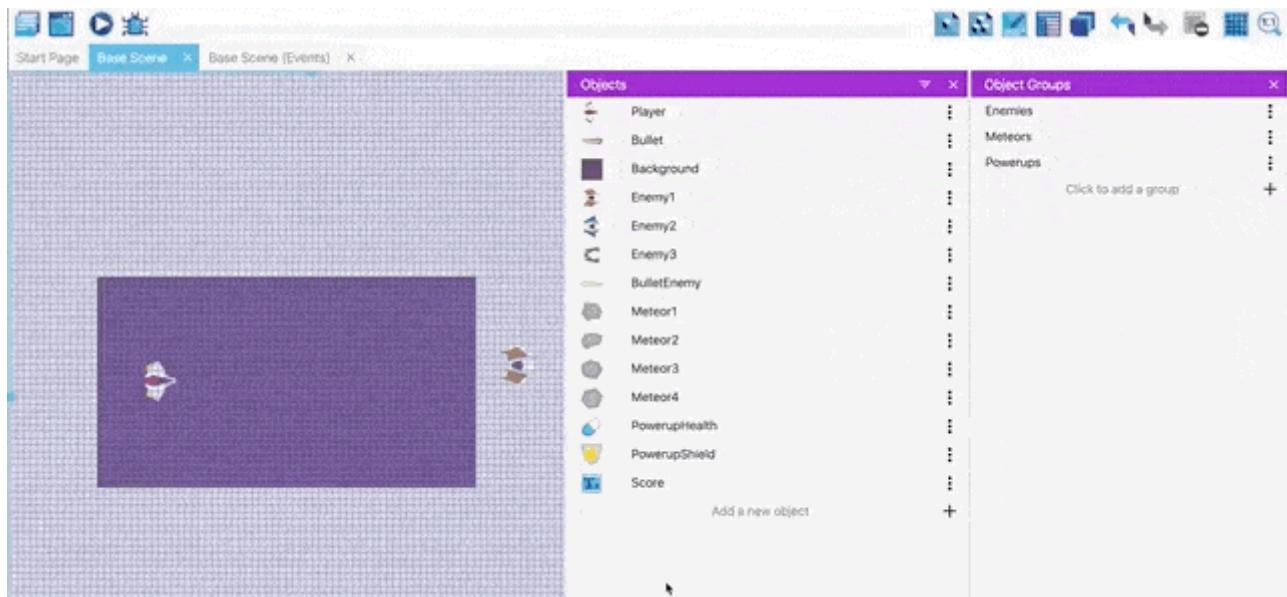
In the end, you can see your two text objects under the other objects.



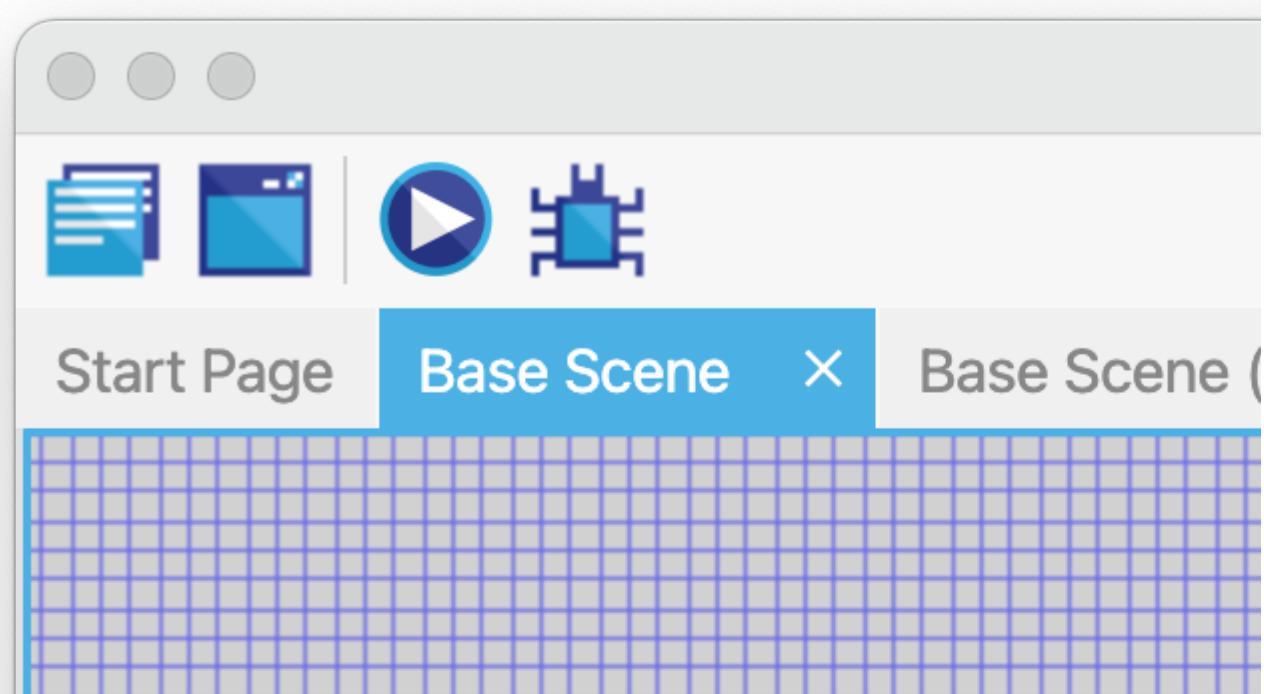




You can drag and drop the text objects to the scene.



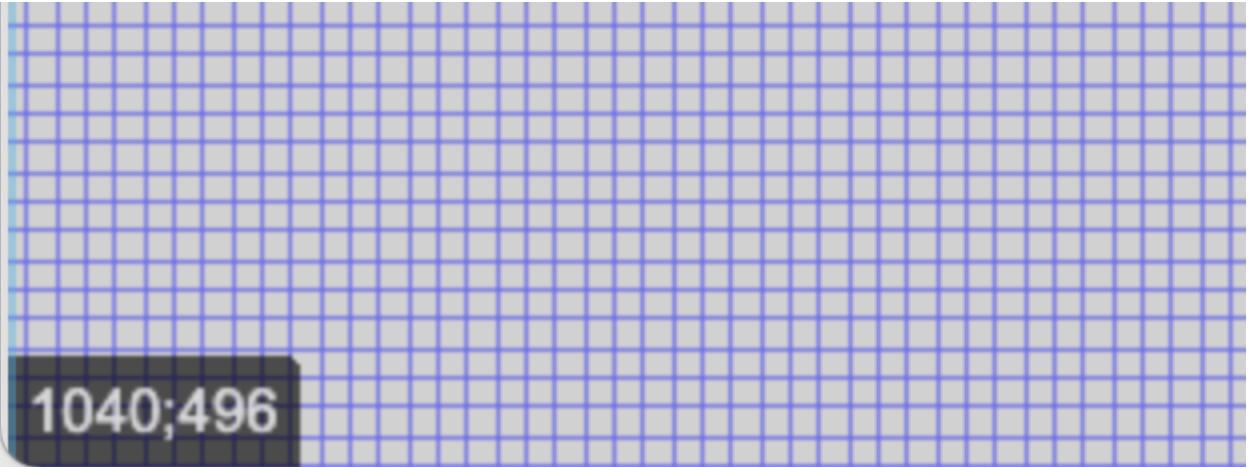
We are putting **Score** to the **left upper corner** and **Health** to the **left lower corner** of the scene.



SCORE: 0



HEALTH: 100



Positioning text objects

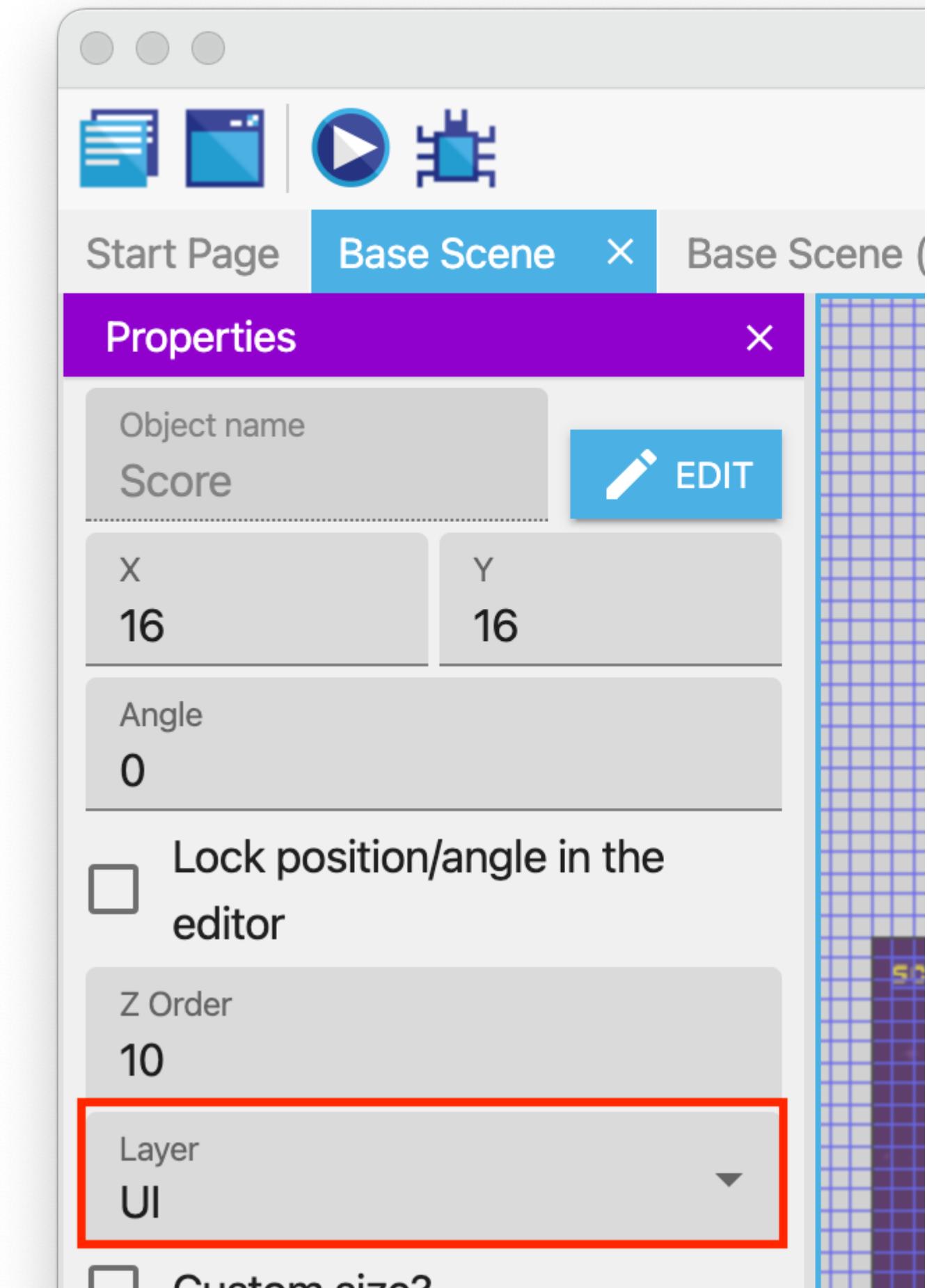
We'll use a new layer where this object will be positioned, so that they are always visible and stay at the same place on the screen. Create a new layer called **UI** and put this layer at the top of the list.

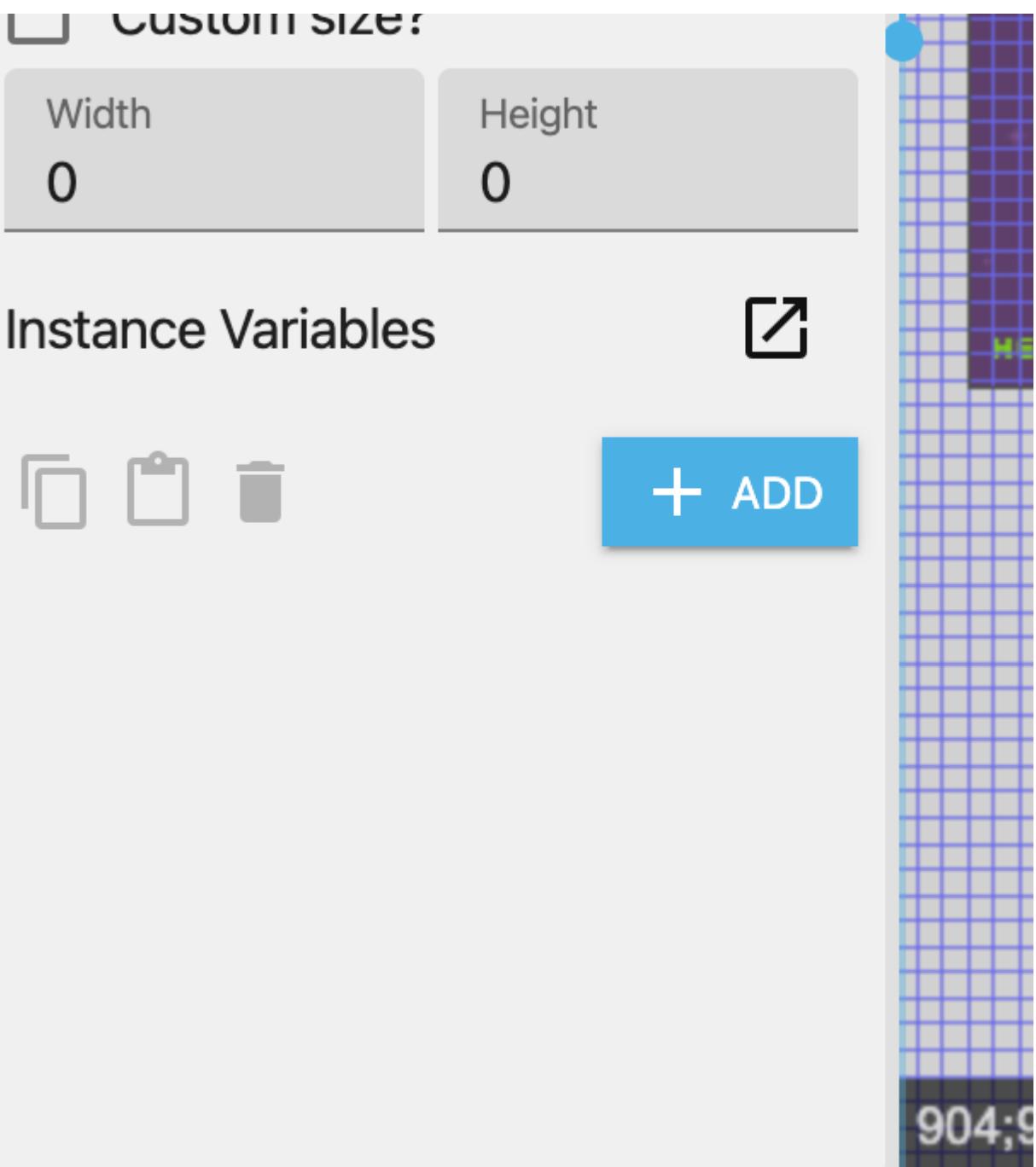
If you forgot how to add a layer, you can check this [chapter](#).

Layer	Status
UI	Selected
Base layer	
Background	Selected
Background color	

We should then select the text objects on the scene and open the properties panel from the icon at the upper right corner. **Change its layer to UI**. By putting these objects on the UI layer, they will always be shown on top of other game objects

(which are on the base layer) and will stay visible on the screen even when the camera of the base layer is moved.

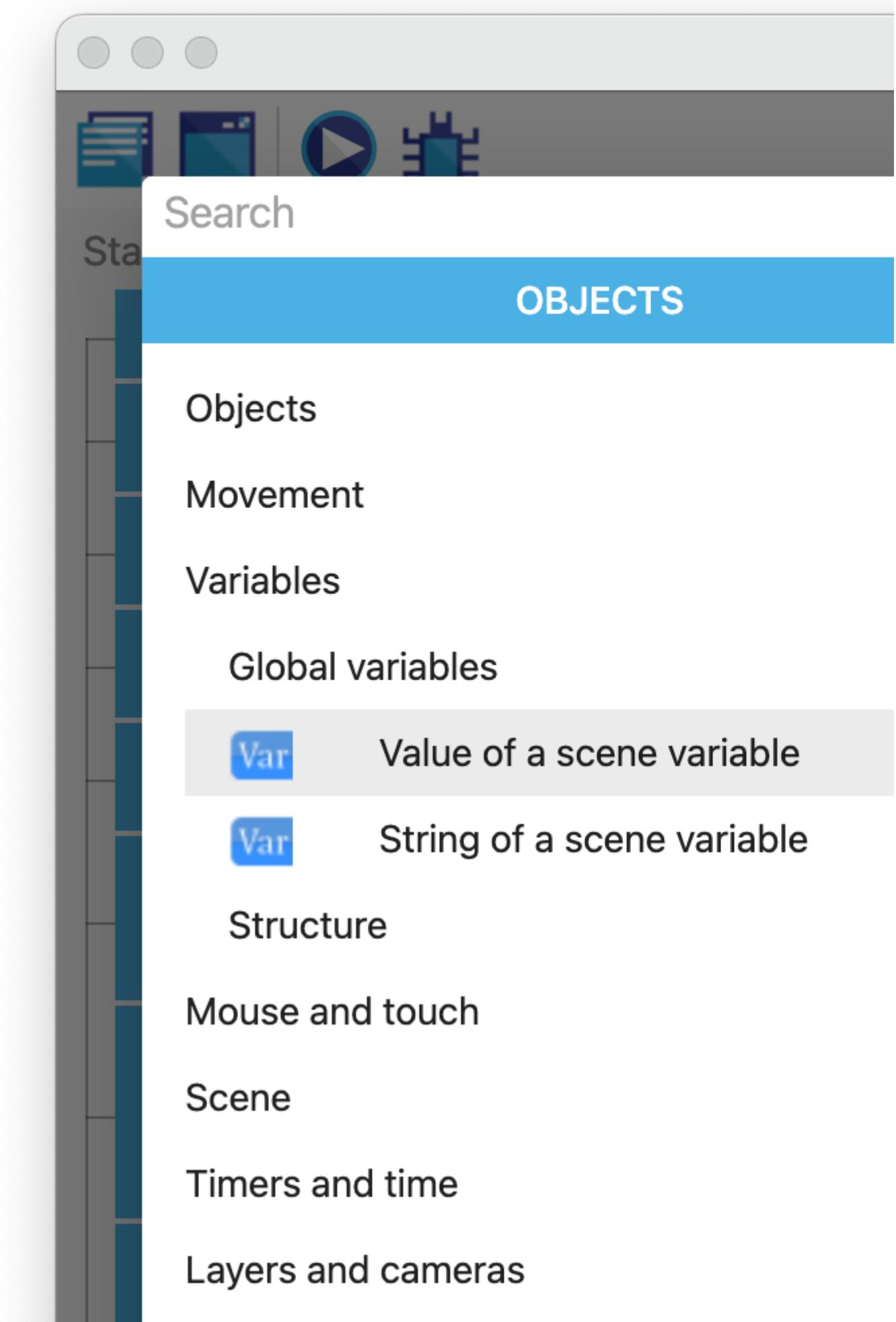


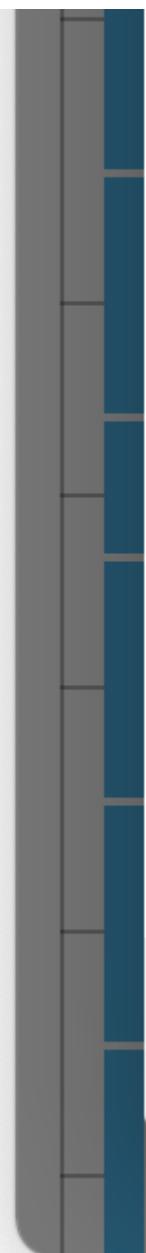


Updating values of text objects

We are ready to open the “Events” page. We will create a new scene variable called “Score” with an **initial value of 0**. Then, add an action to change the score with time. We are going to use the TimeDelta function to change the score according to time.

Ensure that you selected **Add** as a modification sign.





Audio

Storage

Files

Network

Game's window and resolution

Functions

External layouts

① HELP FOR THIS ACTION

Add condition

We can also increase the score of the player when the player explodes an enemy. Hence, we will add an action to increase the score by 100 under the **Enemies dead condition**.



Enemies is dead

Add condition

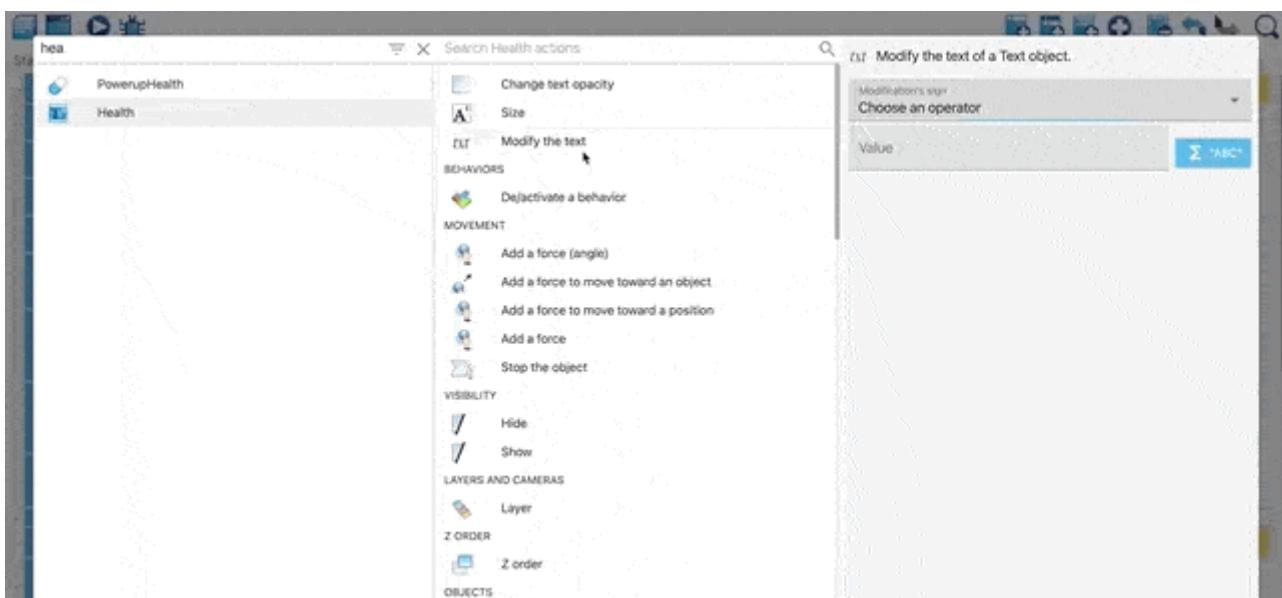
Then, we are ready to update the score of the text object. Add a new action, and select the score text object. Afterward, choose to modify the text and type “**Score** + **VariableString(Score)**”. You should see these actions:

Ensure that the modification sign is set to =.

Add condition

We will add one last action to modify the health text. We will use the benefits of auto-completing GDevelop here: Select **Health** text and choose to modify text. Fill the value with “**Health:** ” + **LargeNumberToString(Player.Health::Health())**.

You can use auto-completion while writing your actions.



In the end, your actions should look like this:

Add condition

Add condition

Testing out

We are ready to test our text objects. You can see the change in health and score.

Next step

[Space Shooter, Part 10](#)