8. Powerups

Series

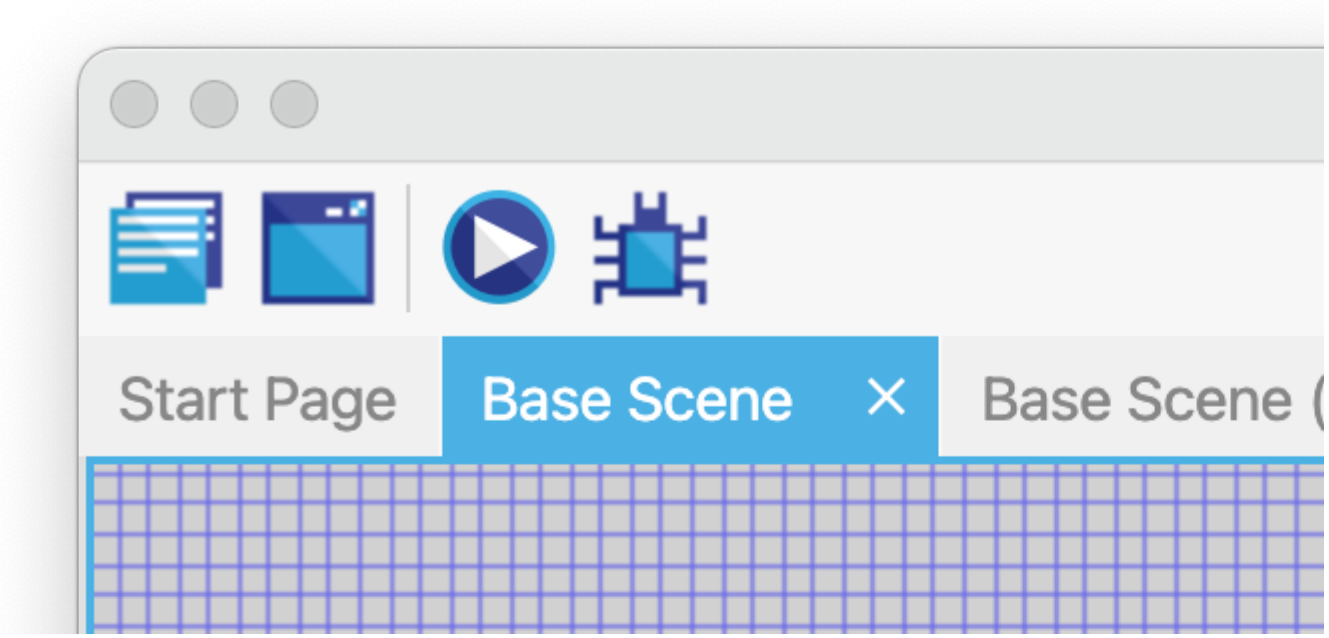
You are reading **Part 8** of the Space Shooter Tutorial.

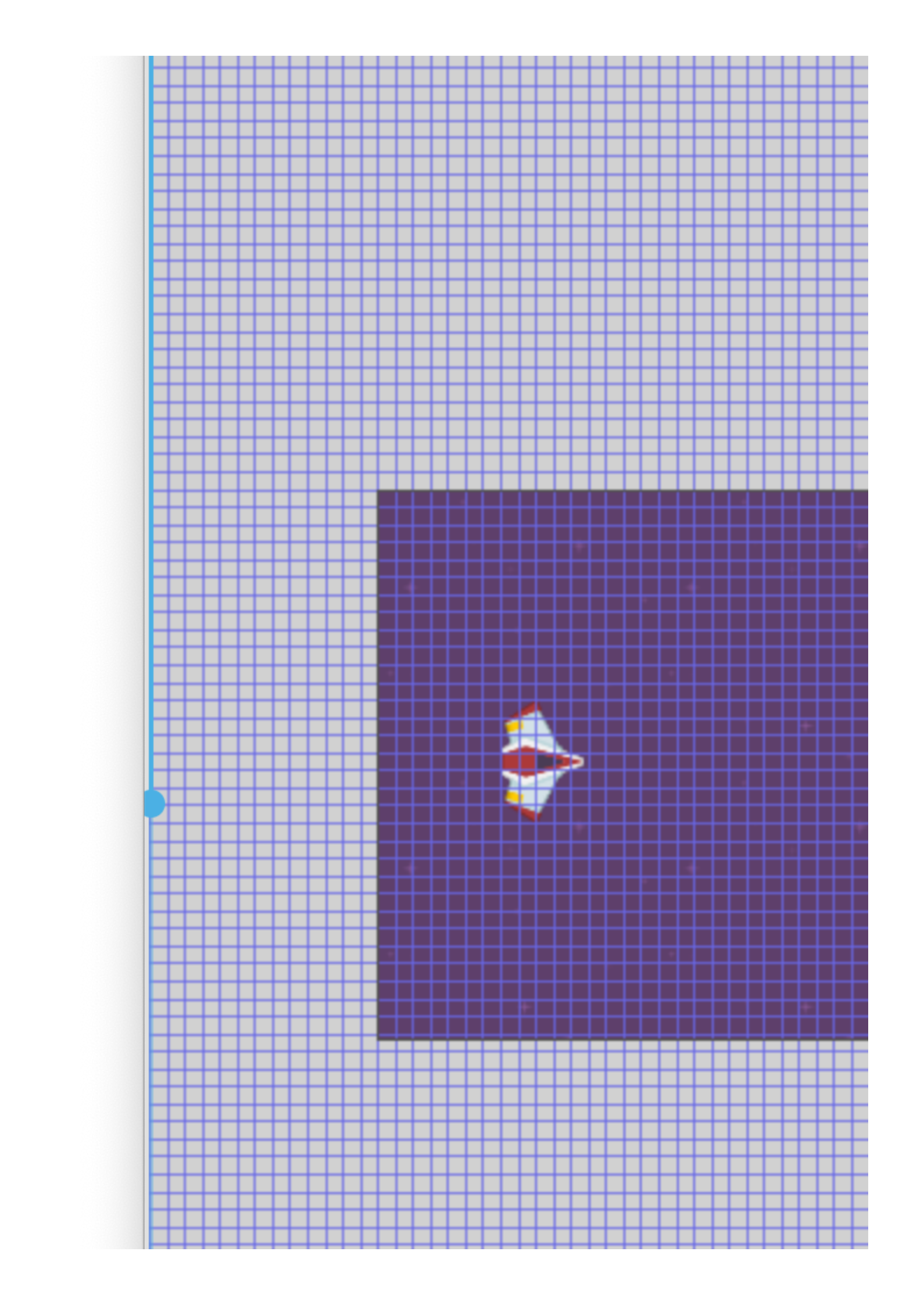
1. Space Shooter, Part 1
2. Space Shooter, Part 2
3. Space Shooter, Part 3
4. Space Shooter, Part 4
5. Space Shooter, Part 5
6. Space Shooter, Part 6
7. Space Shooter, Part 7
8. Space Shooter, Part 8
9. Space Shooter, Part 9
10. Space Shooter, Part 10
11. Space Shooter, Part 11
12. Space Shooter, Part 12
13. Space Shooter, Part 13

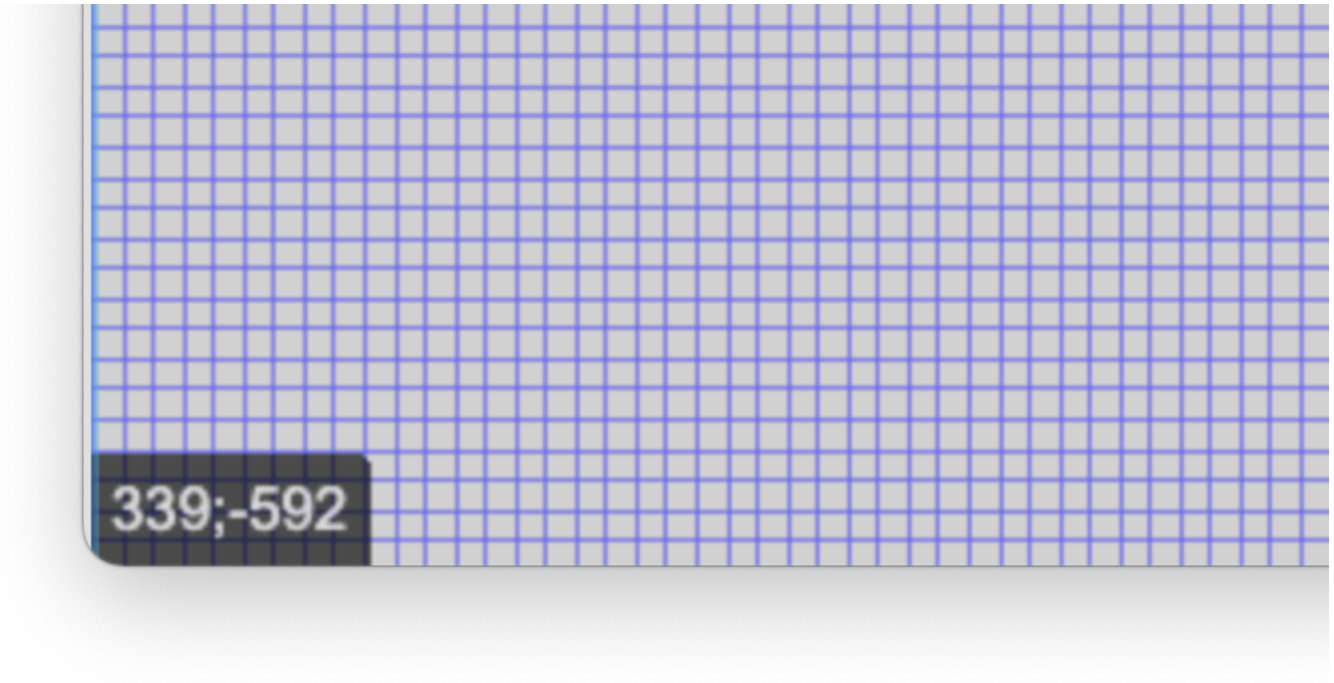
> In this chapter, we will learn how to add two different powerups to the game.

Adding powerups

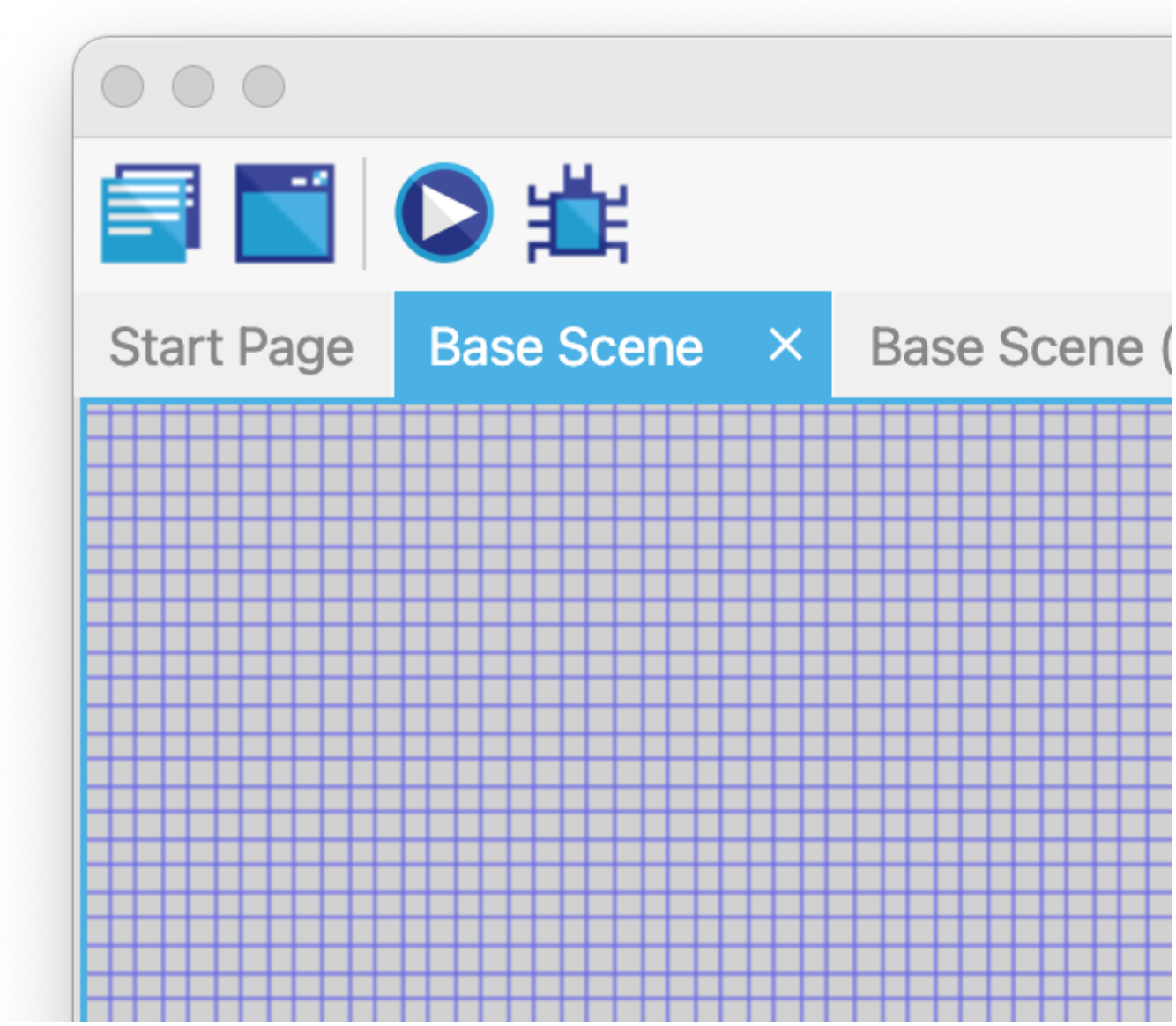
We will add two objects called **“PowerupHealth”** and **“PowerupShield.”** Select the **“powerup_health.png”** image from the assets folder for the health object. Repeat the same process for shield object by choosing the **“powerup_shield.png”** image. In the end, you should see:

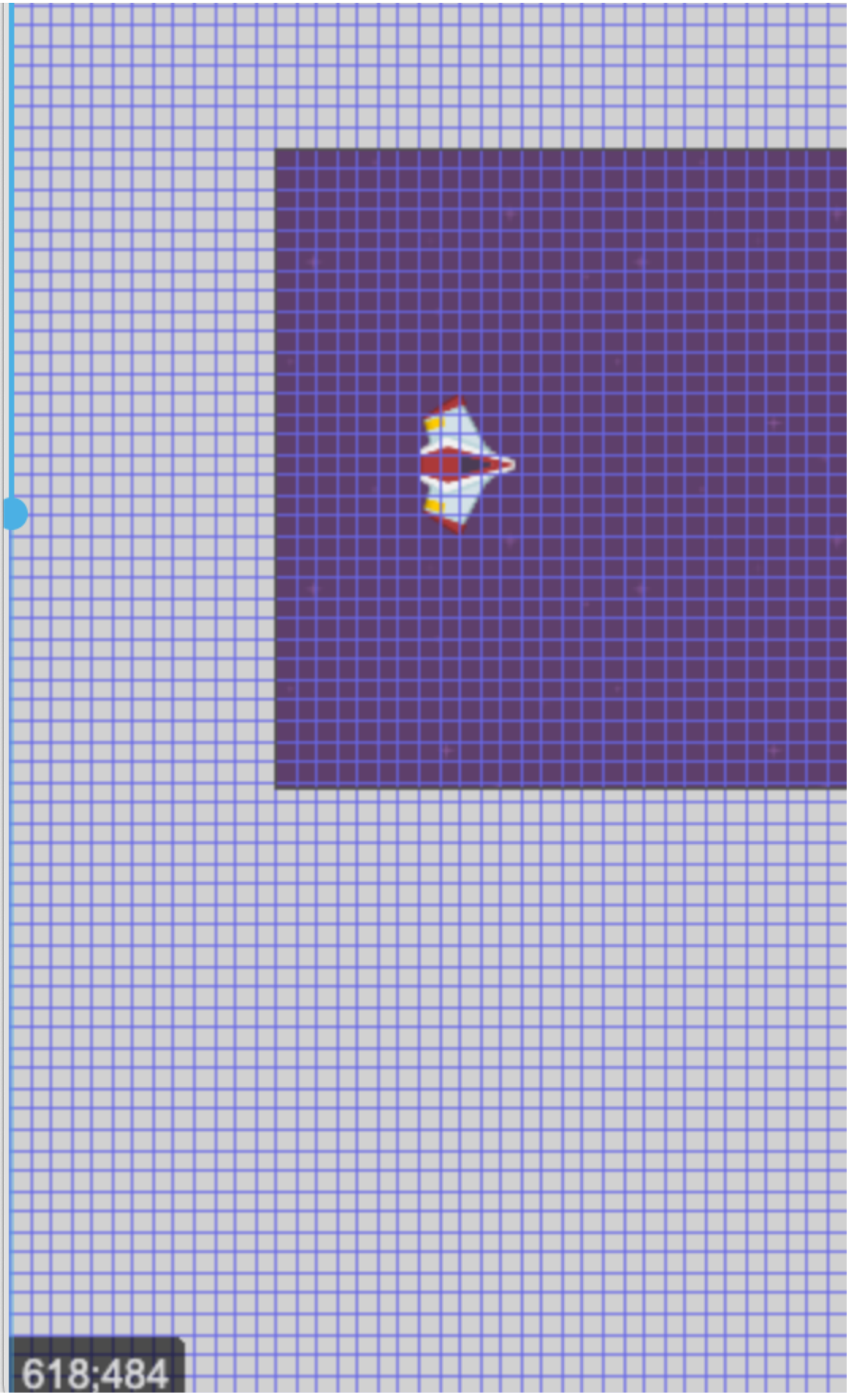






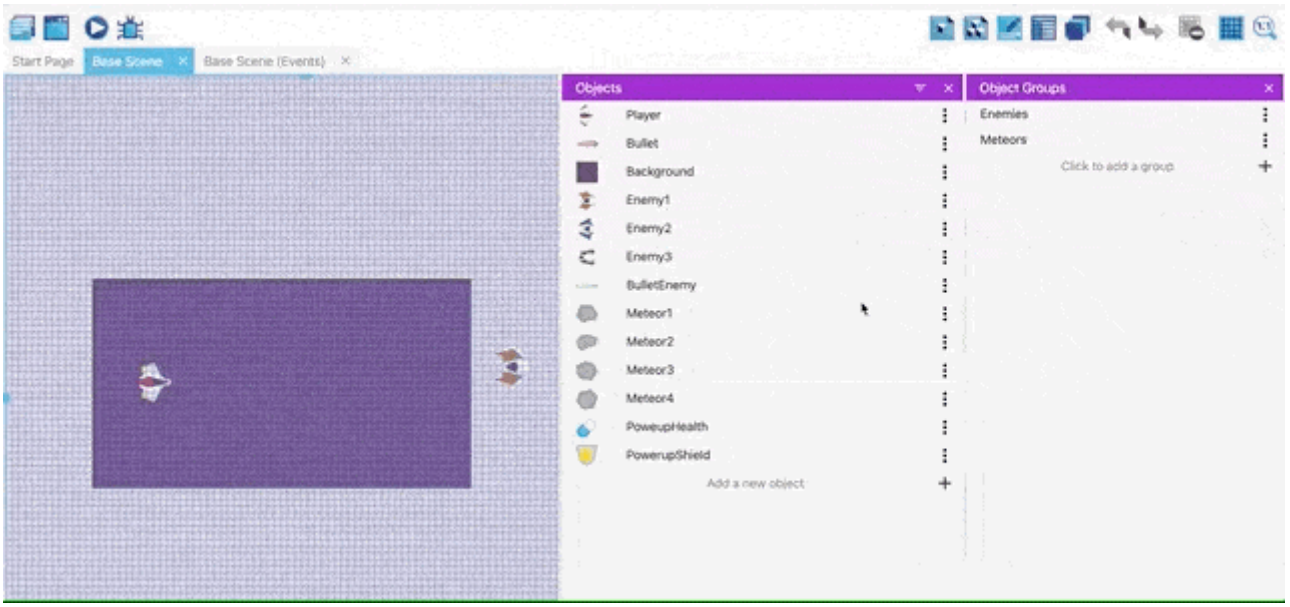
Then create an object group called **“Powerups.”**




618;484

Animating the player

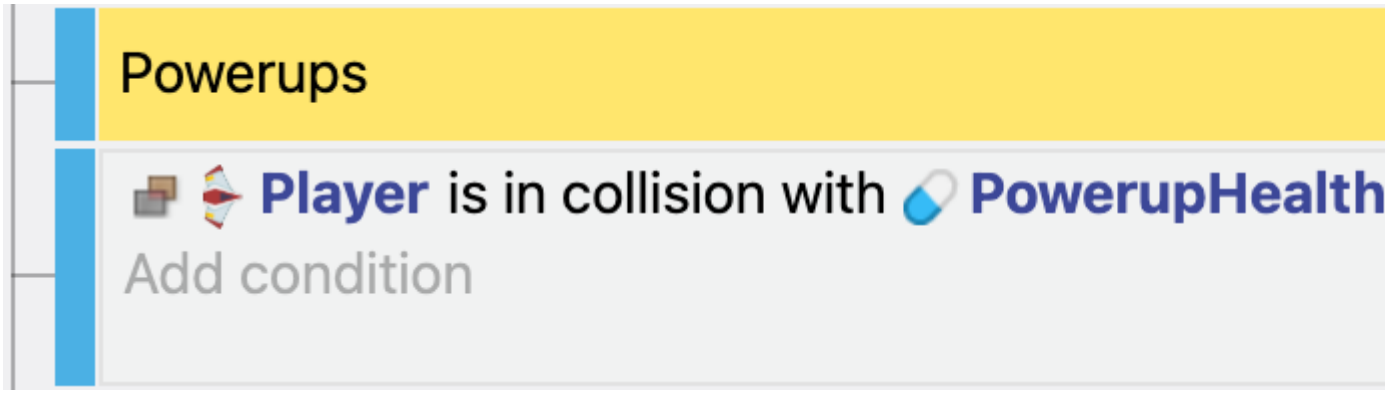
We are using animations to change the player's sprite when the player collides with **PowerupShield**. Open the player's properties panel and enter **“PlayerIdle”** in Animation#0. Then add a new animation called **“PlayerShield,”** and select **“player_shield.png”** from the assets folder.



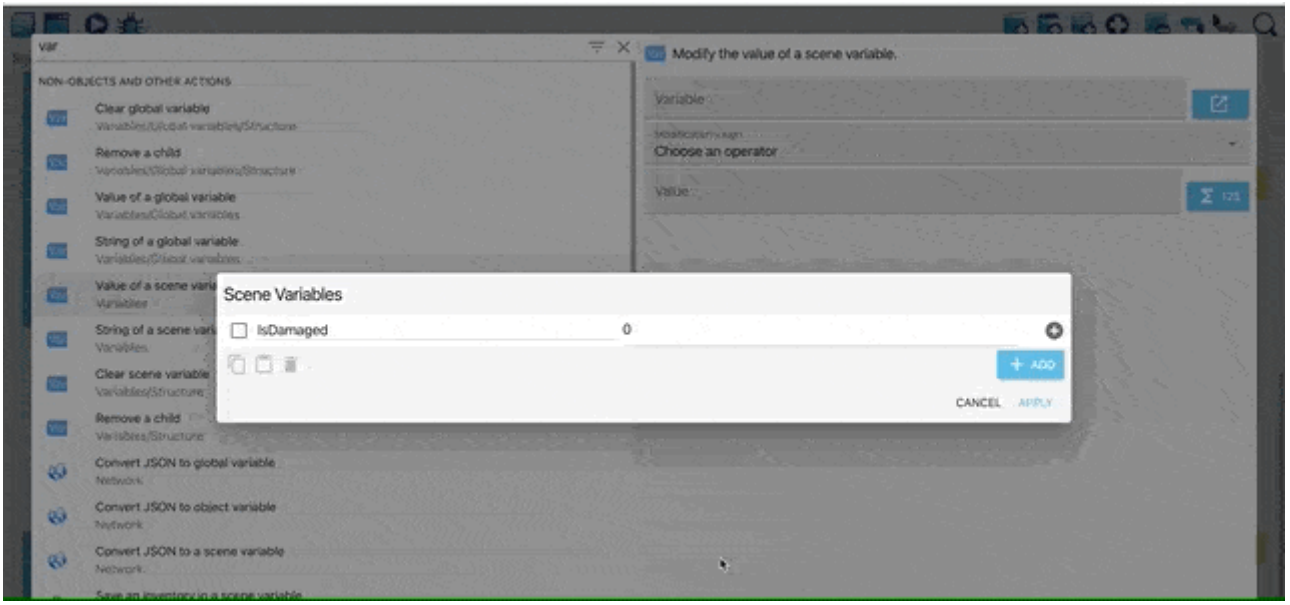
Giving extra health to the player

We are ready to open the **“Events”** page. We will add a new condition to check the collision between the **Player** and the **PowerupHealth** objects. If the collision happens, we want to **increase the player's health by 30** and **delete** the powerup object since we do not need it anymore.

You can add a comment to classify your events.

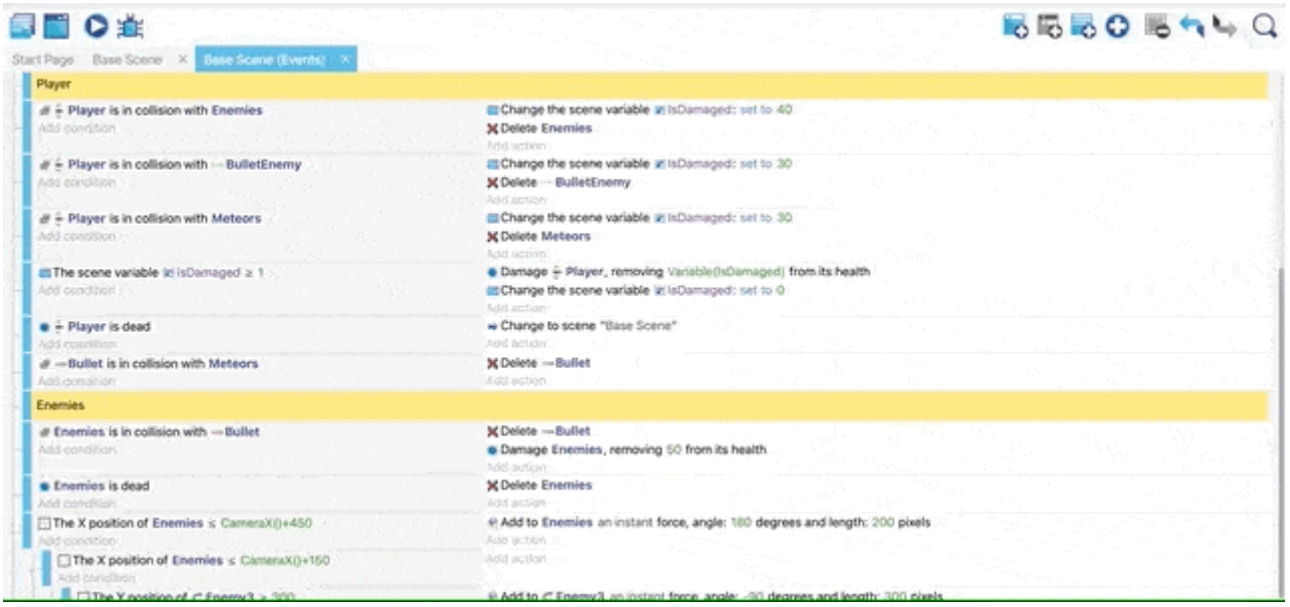


Now, we are going you use a new scene variable for the shield object. Create a new scene variable called **“PowerupShield”** to prevent getting any damage while using this powerup. Enter **0** as a default value.

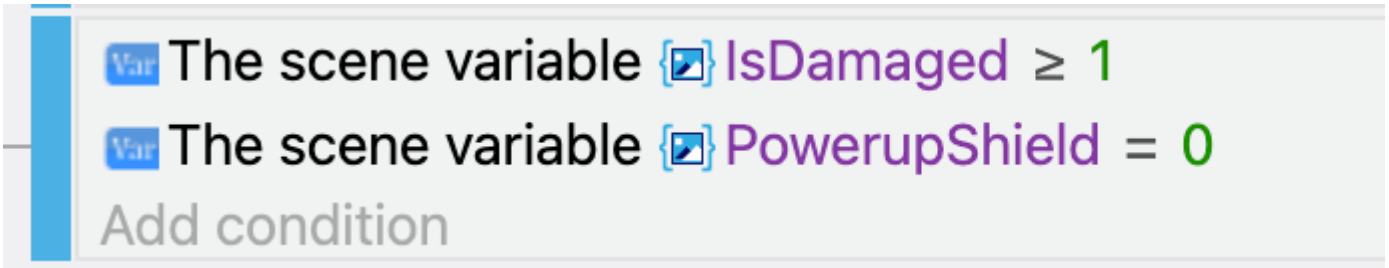


Go to the **“IsDamaged”** condition and add a new condition to check if the **PowerupShield** variable is **equal to 0**.

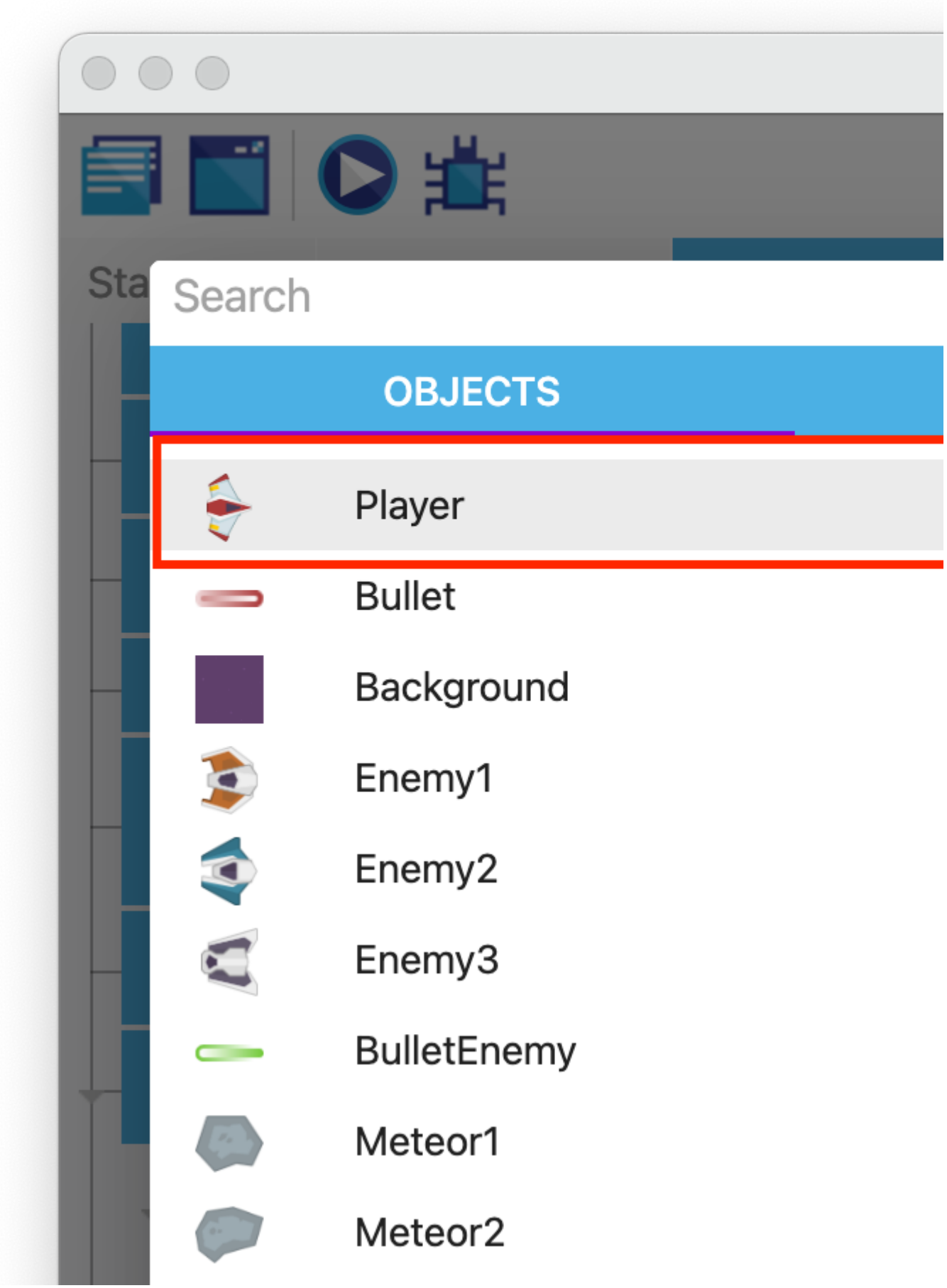
> We are going to use this condition to prevent getting damaged while using the powerup.

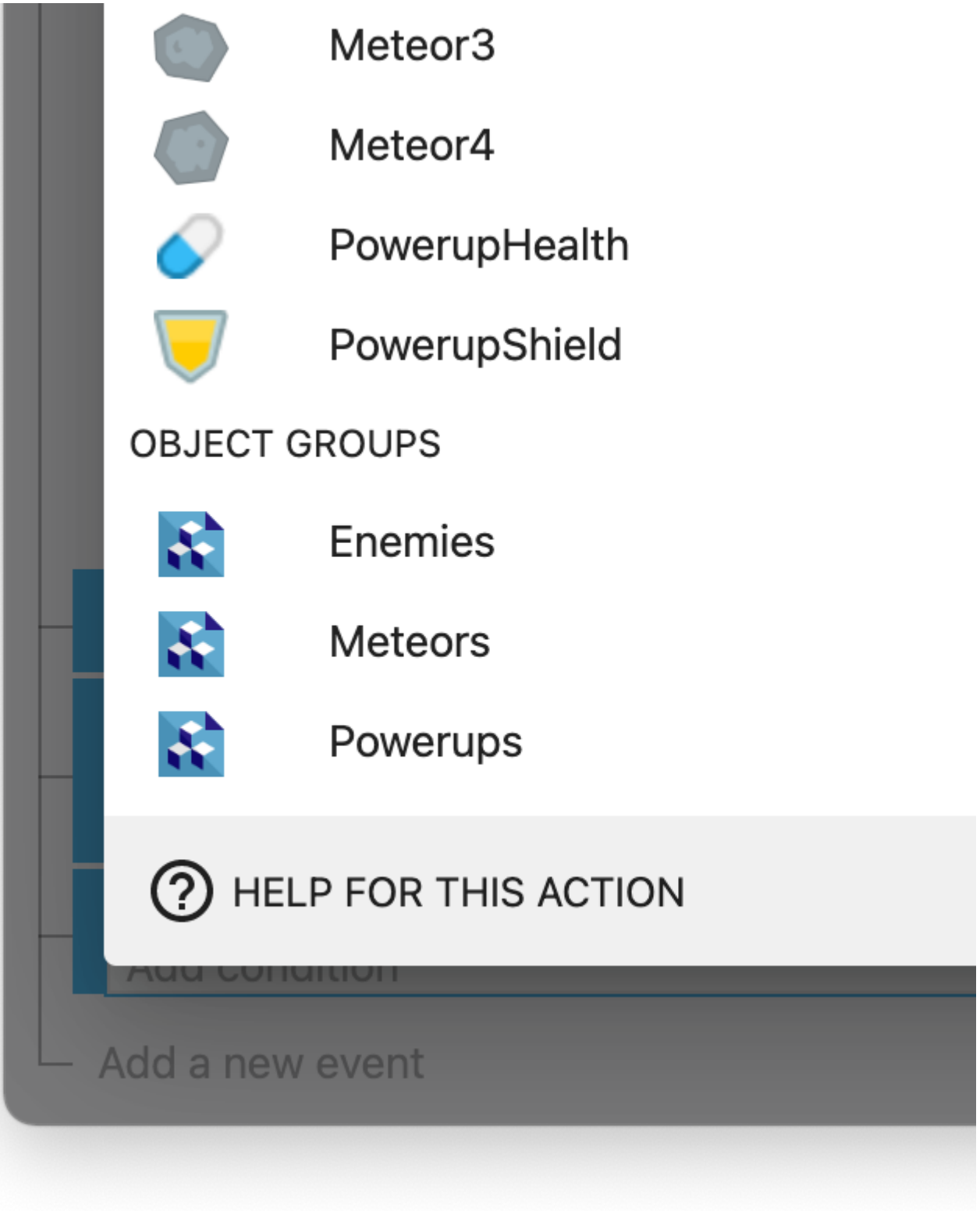


In the end, you should see:



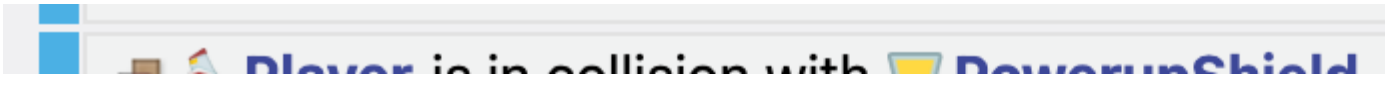
Create a new condition for detecting the collision between the **Player** and the **PowerupShield** objects. Then, add an action to change the animation of the player. Select Player, and choose **“Change the animation (by name)“** and enter **“PlayerShield”** in the name field.

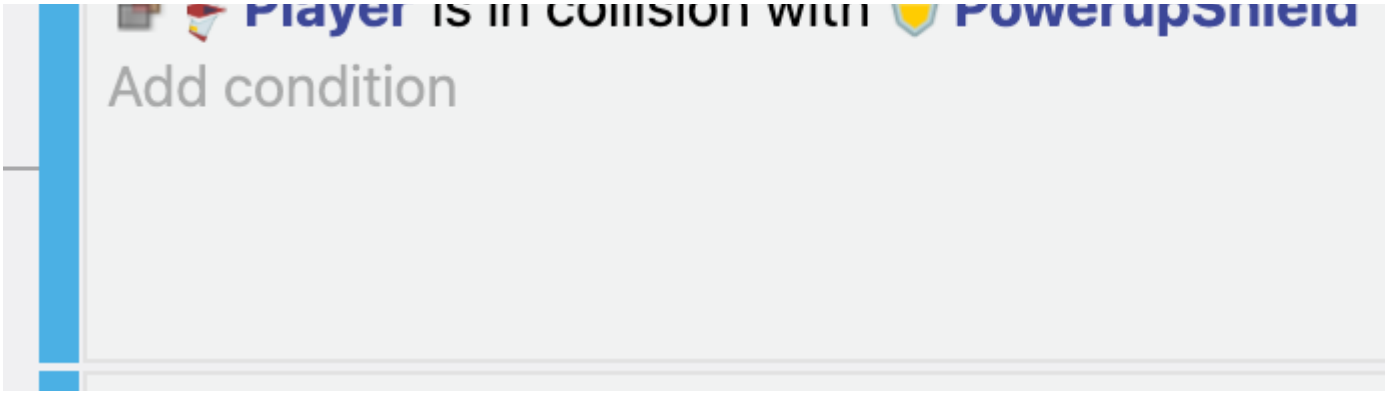




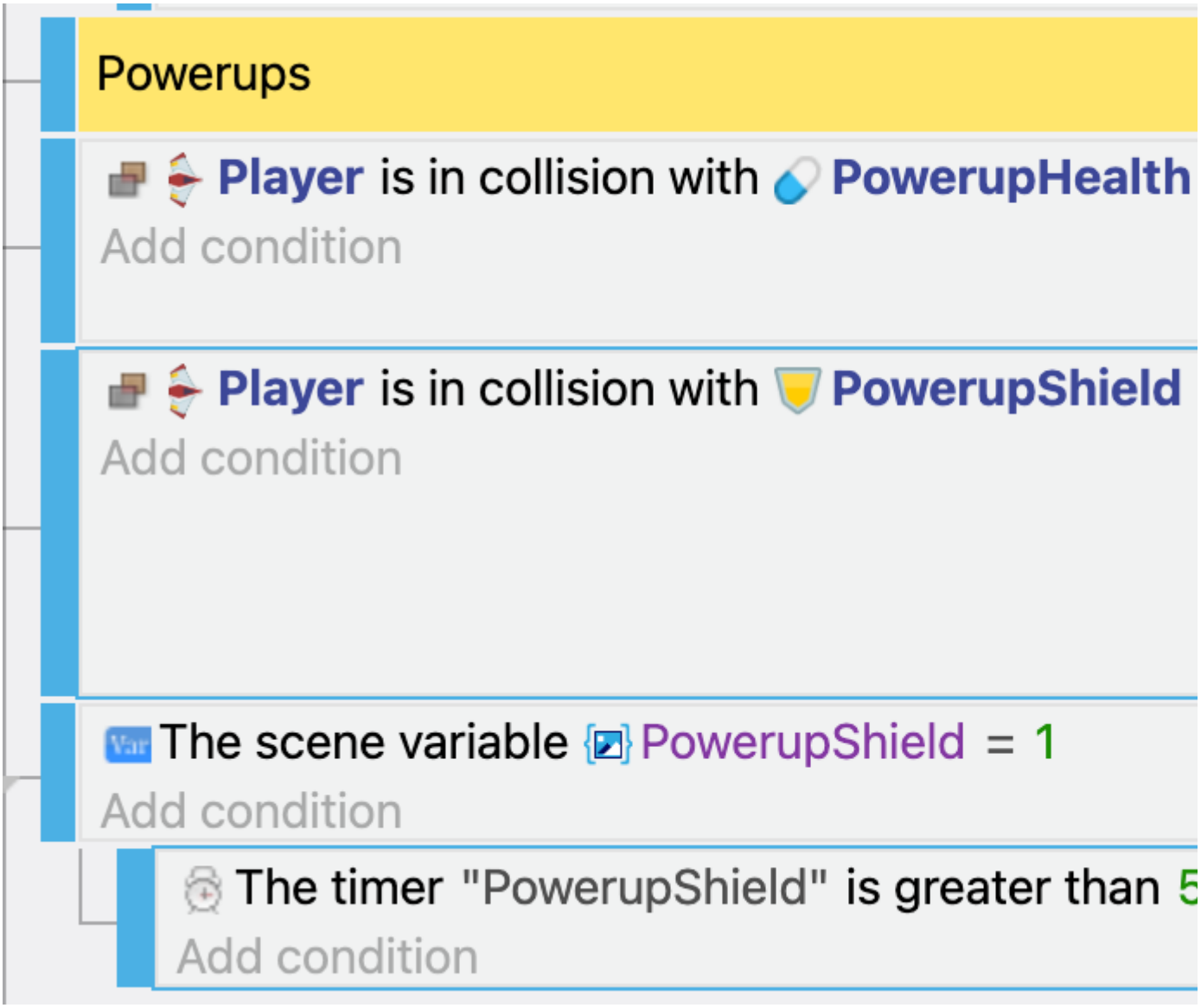
Then, add some further actions to **set “PowerupShield” to 1**, **delete the “PowerupShield”** object, and reset a timer called “PowerupShield.” In the end, you should see these events:

We will use this timer to determine the usage of powerup in the next steps.

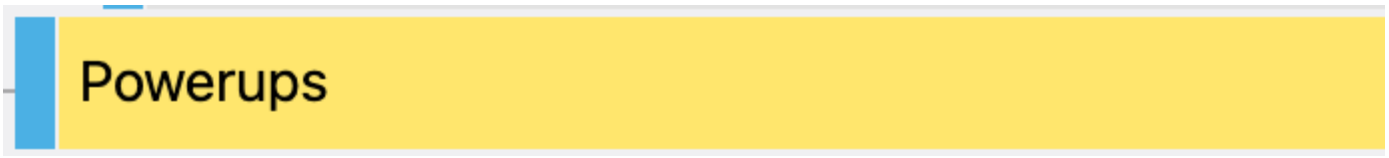


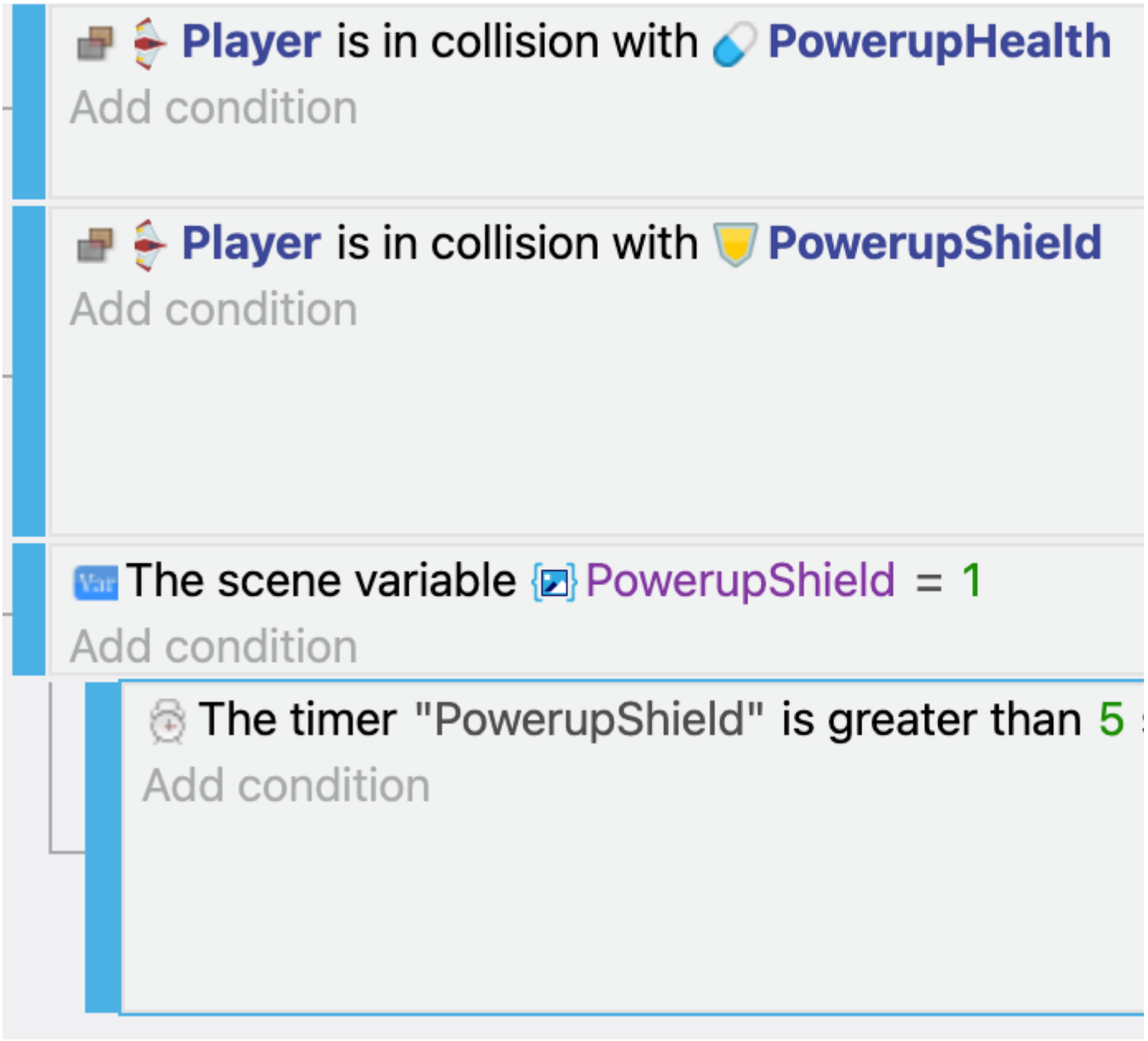


Add a new condition to check whether the **“PowerupShield”** variable is **equal to 1** or not. Then, add a sub-event to check the **timer called “PowerupShield” is greater than 5 seconds**.



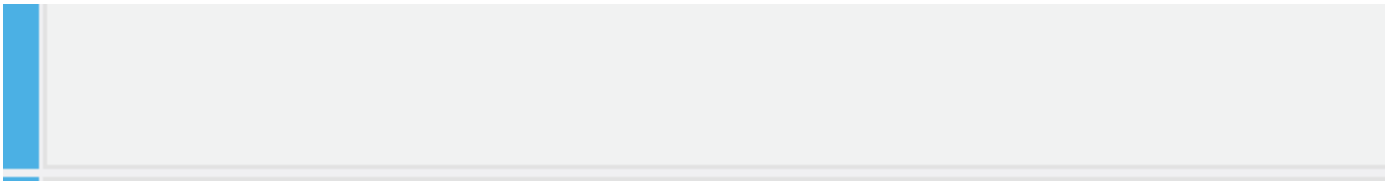
Finally, we are going to set “**PowerupShield”** and **“IsDamaged”** variables to **0** and **change the animation** of the player to **“PlayerIdle.”** We also need to reset the timer called **“PowerupShield.”** In the end, your events should look like this:





Testing out

Before testing powerups, we should change the scale of powerup objects just like in the previous cases. Select **Powerups object group** and set scale factor to **1.5**:



Then, open the scene view and drag and drop the powerup objects to the scene. We are ready to run the game. Now you can see the powerups on the scene. If you collect **health's powerup**, **the player's health will be increased by 30**. If you collect the **shield's powerup**, the **player's sprite will be changed**, and you will not get damaged for 5 seconds.

Next step

Space Shooter, Part 9