GAME250: Technical Game Development (Spring 2023)

Exercise 10

Objectives

By the end of this exercise, you'll be able to:

- Add animations to UI widgets

Background

Find a detailed introduction to animations in the Unreal Motion Graphics UI Designer here:
https://docs.unrealengine.com/en-US/InteractiveExperiences/UMG/UserGuide/Animation/index.html

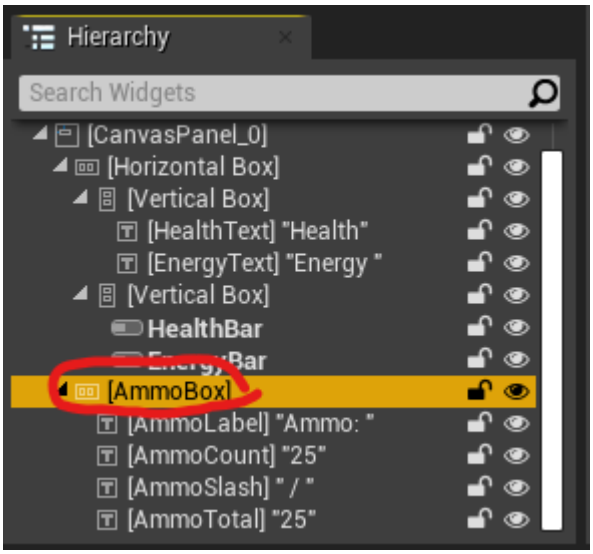
Using animations, let's add a low ammo indicator that flashes to draw the player's attention. We'll also add color to the ammo text.

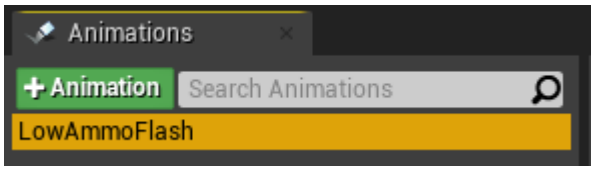
In-Lab Instructions

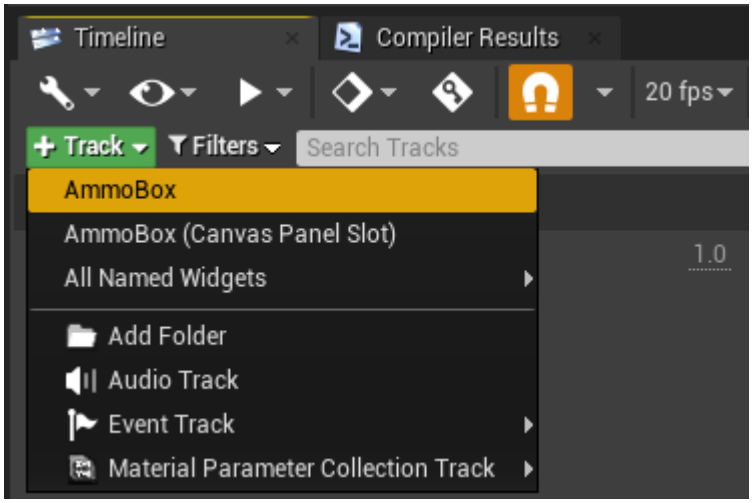
Download and open provided game project named Menus.

Our First Animation

Open the blueprint of the HUD widget. Go to the Designer section. In the Hierarchy tab, find the horizontal box that contains four text boxes and name it AmmoBox. In the Animations tab, click +Animation and name it LowAmmoFlash. Finally, in the Timeline tab, click +Track and select AmmoBox.

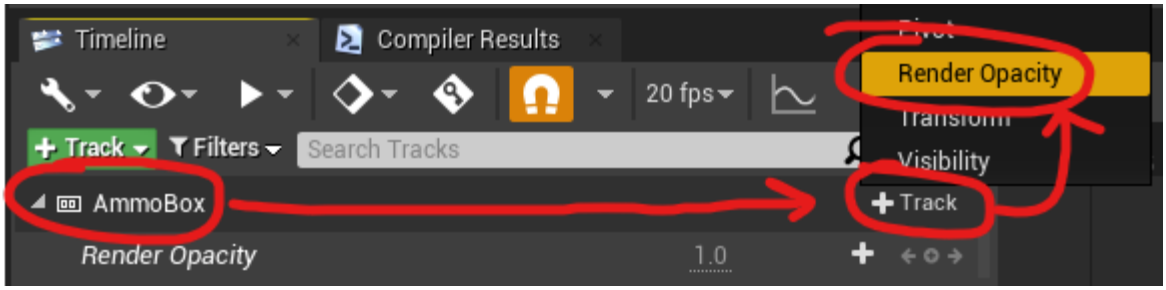




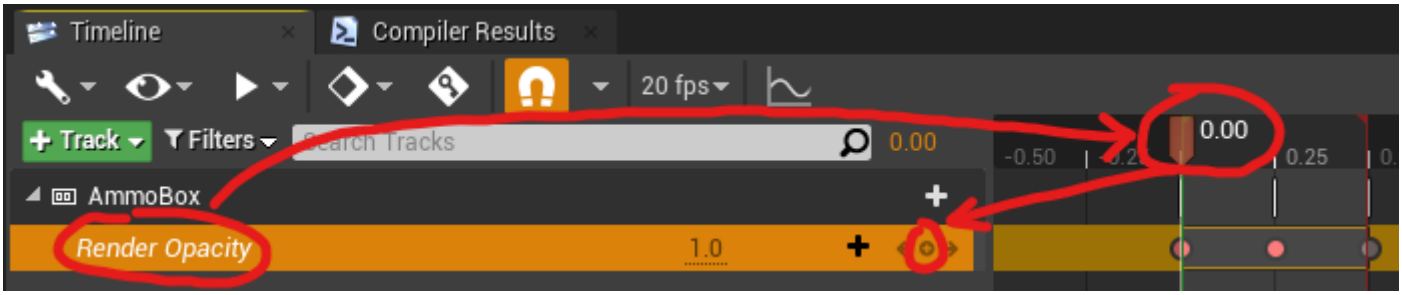


You have just created a named animation which will change the appearance of AmmoBox based on a timeline. Now let's add the flashing animation to LowAmmoFlash.

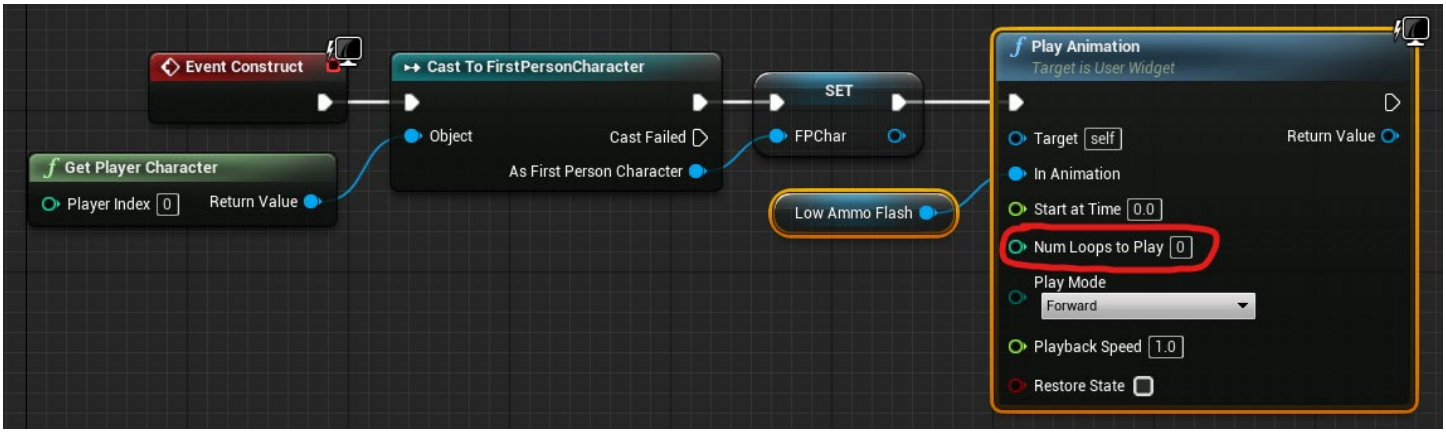
Click +Track on the right side of AmmoBox and select Render Opacity.



In the visual timeline, add three keyframes, at 0.00, 0.25, and 0.50 seconds and set their opacities to 1.0, 0.0, and 1.0, respectively. You can add keyframes by selecting the Render Opacity row, moving the slider to the desired time, then clicking the circle / pressing Enter.



Let's take a look at how our animation looks! Hook it up in the Graph section as shown below:



You can find LowAmmoFlash as a variable in this blueprint and use Play Animation to trigger it as soon as the widget is constructed. Notice that the Num Loops to Play input pin is set to 0, which indicates infinite looping.

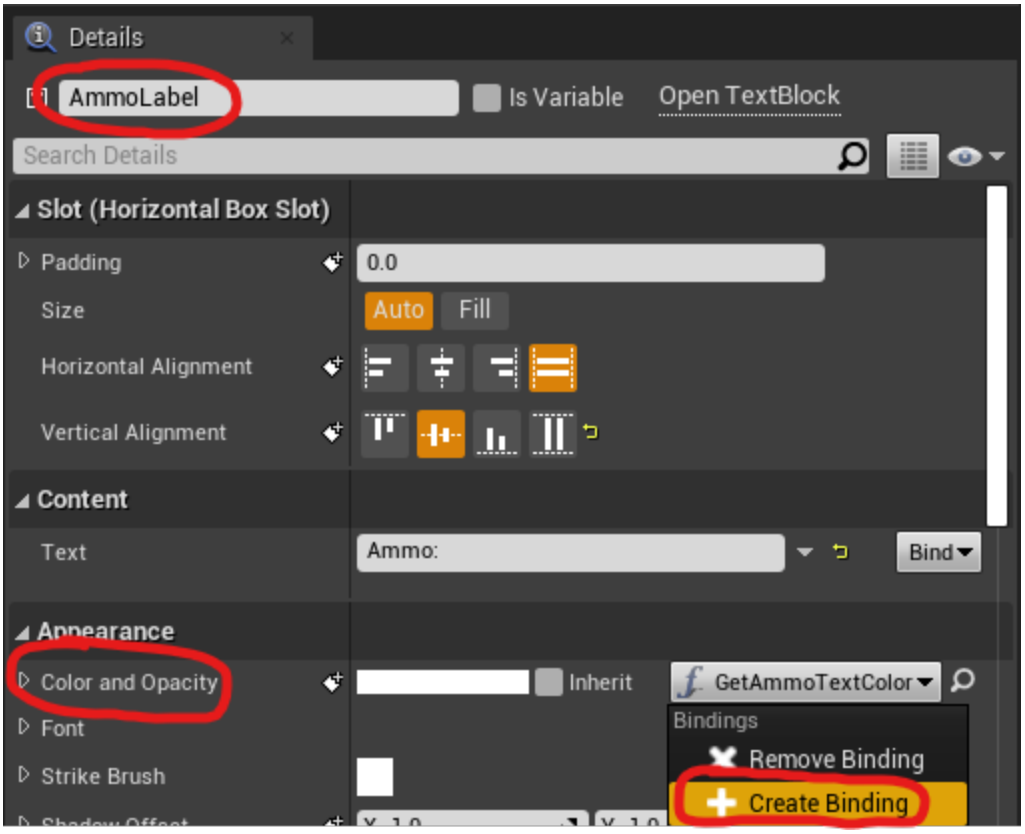
It works, but it's a bit silly since it flashes no matter what the ammo count is. Make sure to delete these two nodes to remove the constant animation.

Animation Binding

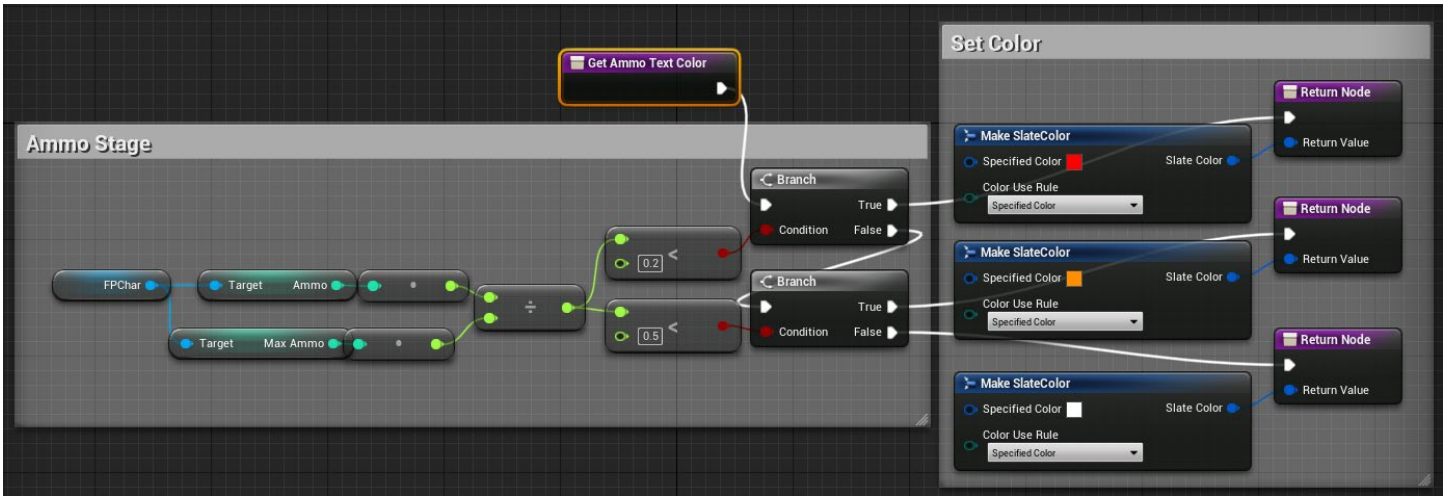
When should the LowAmmoFlash animation start? When the ammo is low, of course! Let's add some code to make this happen. Along the way, we'll also update the color of the text. This will occur in three stages:

- 50% ~ 100% full: white text
- 20% ~ 50% full: orange text
- 0% ~ 20% full: red text and flashing

Select one of the text boxes in AmmoBox and create a binding function for its color and opacity. Since we already have an animation for the opacity, this function will only update the color.



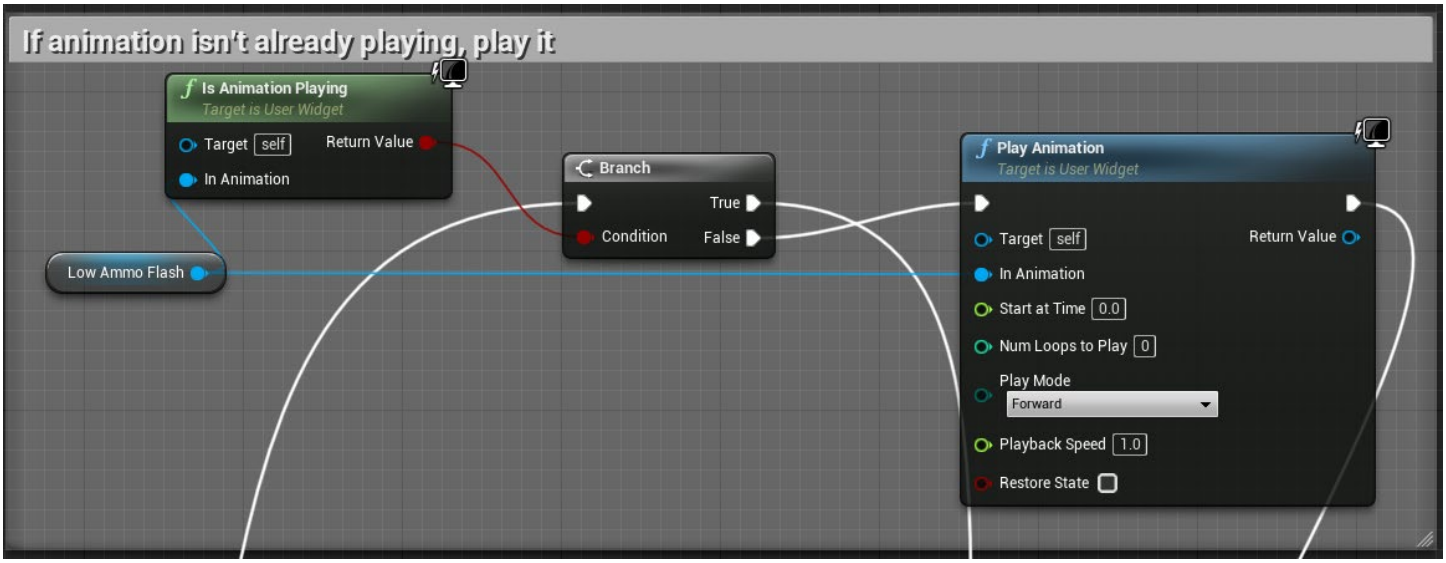
Name the function GetAmmoTextColor and add in the following nodes. The left side is to determine the stages and the right side is for setting the colors.



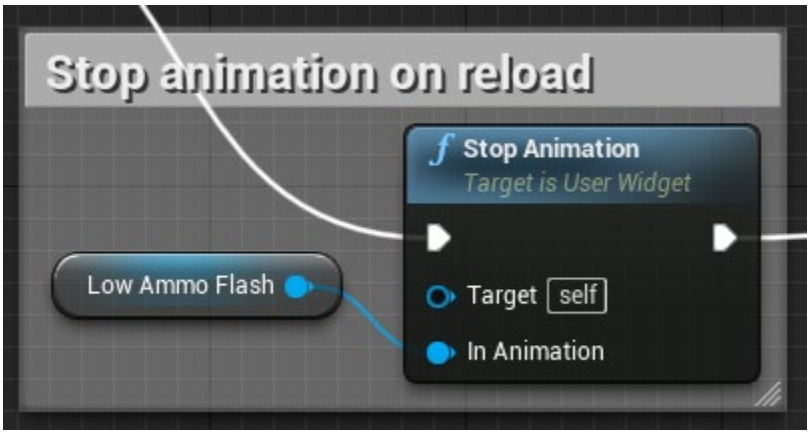
Run the game to confirm that the text box updates properly based on the remaining ammo. Now apply the binding to each of the other text boxes in AmmoBox.

Finally, let's add the animation. There are a few new nodes of interest: Is Animation Playing and Stop Animation.

When the ammo count is below 20% and the animation wasn't already playing, play it.



When the ammo count replenishes to 50% or above (reload), stop the animation.



Now It's Your Turn!

Add the same kind of animation to the HealthBar and Energy bar: if the they drop below 25%, add a flashing animation. You will need to create a new animation and bind them to new functions for each of the bars.

You can optionally add color changes to the bars.

Lab Submission

Take a screenshot of the new animation timelines and functions in the Now It's Your Turn! section.

Submit the screenshot to Canvas before the deadline.