

SHADERS FOR GAME DEVS

A course by [Freya Holmér](#) for students at [FutureGames](#)
shared publicly with permission 💖

[Code+Assets Download](#)

Lectures

- 1 • Shader Basics, Blending & Textures • [Video](#)
- 2 • Healthbar, SDFs & Lighting • [Video](#)
- 3 • Normal Maps, Tangent Space & IBL • [Video](#)

Assignment 1

Create a healthbar shader! You can use the Unity quad mesh as your healthbar, and set y scale to 0.125 so that it isn't as square



1a) Make a healthbar shader with:

- a slider property from 0 to 1 for the amount of health (0 = no health, 1 = max health)
- Start/End Color property
- When the health is full, the whole healthbar should be green
- When the health is 0, the whole healthbar should be red
- Any values in between should be interpolated
- Use a black color for the empty parts of the healthbar

1b) Add two thresholds for when to use the min color vs max color, so that the healthbar goes fully red when you have less than 20% health, and fully green when you have more than 80%

1c) Make the black parts transparent/invisible instead! There are two ways of doing this - either by making the shader transparent using Alpha Blending:

Blend SrcAlpha OneMinusSrcAlpha

Or you can use the [clip\(\) function](#) while still using an opaque shader!

1d) Instead of using start/end colors and start/end thresholds, use this texture instead!



<https://acegikmo.com/storage/futuregames/healthbar.png>

- The color should still be uniform across the entire healthbar
 - In other words, the x coordinate of this texture represents the amount of health, not for mapping the entire length of this texture across the entire healthbar
- The vertical changes of the texture should still be intact - the gloss and the shadows should still run across your healthbar vertically

1e) Make the healthbar pulsate/flash when you have less than 20% health

- Unity's `_Time.y` represents the current time in seconds that have passed

Optional overachieving things you can do if you want!

- Add rounded edges on the healthbar
- Add a border around the healthbar

Assignment 2

Adapt our lighting Shader we created in part 2, to support multiple light sources!

- This requires a multi-pass shader, with a base pass and an add pass
- This relates specifically to Unity's built-in forward renderer
- You might have to google around a lot and look at Unity's built in shaders!