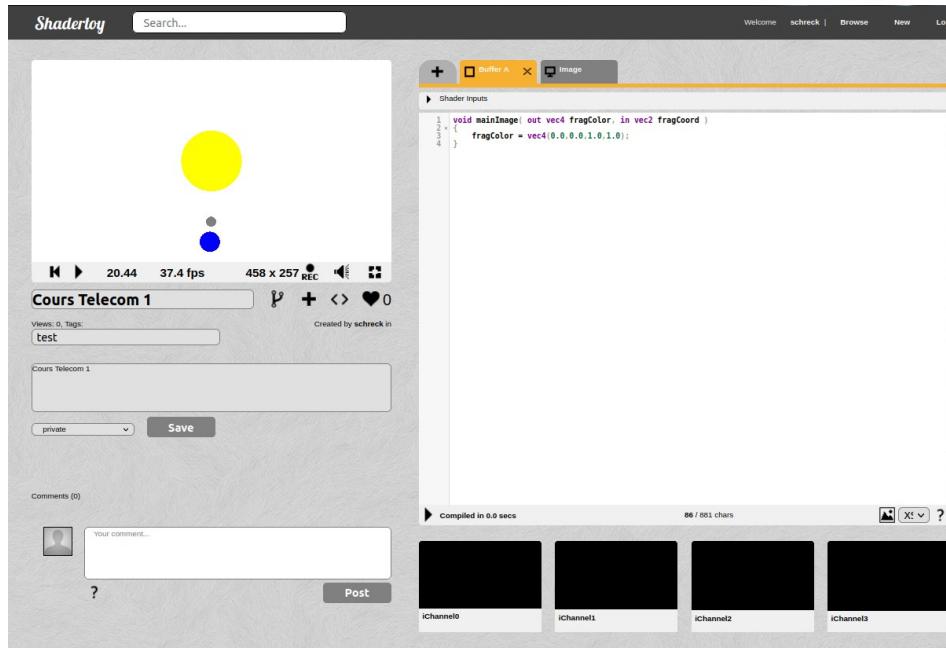


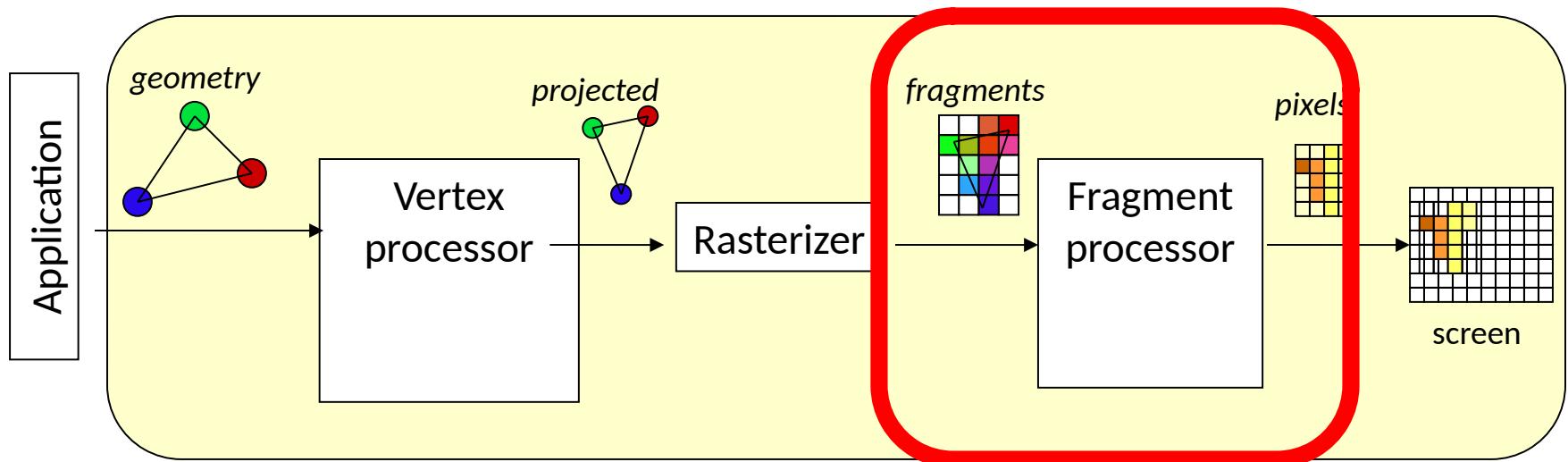
Intro Shadertoy

Fragment shader: language GLSL

<https://www.shadertoy.com/>
<https://shaderoo.org>



GPU pipeline in a nutshell



Shadertoy: fragment shader

The image shows the Shadertoy interface with a color gradient preview on the left and the GLSL code editor on the right.

Preview: A vertical color gradient from red at the top to blue at the bottom, displayed in a 458x257 resolution window. The interface includes playback controls (rewind, play, frame rate 7.07, 34.2 fps), a recording button, and volume controls.

Name of your shader: Nameless

Tags, comma separated. For example: Tags

Description: Describe your shader

Visibility: private

Submit: Submit button

Code Editor:

```
1 void mainImage( out vec4 fragColor, in vec2 fragCoord )
2 {
3     // Normalized pixel coordinates (from 0 to 1)
4     vec2 uv = fragCoord/iResolution.xy;
5
6     // Time varying pixel color
7     vec3 col = 0.5 + 0.5*cos(iTime+uv.yxy+vec3(0,2,4));
8
9     // Output to screen
10    fragColor = vec4(col,1.0);
11 }
```

Build Status: Compiled in 0.0 secs

Character Count: 158 chars

Help: ?

Shadertoy: fragment shader

A screenshot of the Shadertoy interface showing the code editor and preview window. The code editor on the right contains the following GLSL fragment shader:

```
1 void mainImage( out vec4 fragColor, in vec2 fragCoord )
2 {
3     // Normalized pixel coordinates (from 0 to 1)
4     vec2 uv = fragCoord/iResolution.xy;
5
6     // Time varying pixel color
7     vec3 col = 0.5 + 0.5*cos(iTime+uv.yxy+vec3(0,2,4));
8
9     // Output to screen
10    fragColor = vec4(col,1.0);
11 }
```

The preview window on the left shows a red square on a purple gradient background. The status bar at the bottom indicates "Compiled in 0.0 secs" and "158 chars".

Shadertoy: fragment shader

The image shows the Shadertoy interface. On the left is a preview window displaying a purple gradient background with a dashed red rectangle. Below the preview are controls for playback (rewind, play, frame rate 7.07, 34.2 fps), resolution (458 x 257), and recording. To the right is the main code editor area. The title bar says "Image". Under "Shader Inputs", there is a single input labeled "iResolution". The code itself is:

```
1 void mainImage( out vec4 fragColor, in vec2 fragCoord )
2 {
3     // Normalized pixel coordinates (from 0 to 1)
4     vec2 uv = fragCoord/iResolution.xy;
5
6     // Time varying pixel color
7     vec3 col = 0.5 + 0.5*cos(iTime+uv.yxy+vec3(0,2,4));
8
9     // Output to screen
10    fragColor = vec4(col,1.0);
11 }
```

Below the code editor are fields for "Name of your shader", "Tags, comma separated. For example:", "Describe your shader", a dropdown menu set to "private", and a "Submit" button. At the bottom of the editor are status messages: "Compiled in 0.0 secs" and "158 chars". A help icon (?) is in the bottom right corner.

Shadertoy: fragment shader

The image shows the Shadertoy interface with a preview window on the left and a code editor on the right.

Preview Window: Displays a purple square with a red wavy line starting from the bottom-left corner. A dashed line indicates the camera's field of view. The interface includes controls for playback (rewind, play, frame rate 7.07, 34.2 fps), resolution (458 x 257), and recording.

Code Editor: Shows the GLSL code for a fragment shader named "Image". A red box highlights the input parameter `in vec2 fragCoord`.

```
1 void mainImage( out vec4 fragColor ) in vec2 fragCoord
2 {
3     // Normalized pixel coordinates (from 0 to 1)
4     vec2 uv = fragCoord/iResolution.xy;
5
6     // Time varying pixel color
7     vec3 col = 0.5 + 0.5*cos(iTime+uv.yxy+vec3(0,2,4));
8
9     // Output to screen
10    fragColor = vec4(col,1.0);
11 }
```

Input Fields: Name of your shader, Tags, comma separated. For example, Describe your shader.

Buttons: private, Submit, Compiled in 0.0 secs, 158 chars, X, ?

Shadertoy: fragment shader

The image shows the Shadertoy interface. On the left is a preview window displaying a pink-to-purple gradient background with a vertical dashed red line. Below the preview are controls for playback (rewind, play, stop), frame rate (7.07, 34.2 fps), resolution (458 x 257), and recording. To the right is the GLSL code editor. A red box highlights the declaration of the output variable `fragColor`. A red arrow points from this highlighted line to the word "RGBA" followed by a purple square, indicating that the shader outputs an RGBA color. The code itself is as follows:

```
1 void mainImage( out vec4 fragColor, in vec2 fragCoord )  
2 {  
3     // Normalized pixel coordinates (from 0 to 1)  
4     vec2 uv = fragCoord/iResolution.xy;  
5  
6     // Time varying pixel color  
7     vec3 col = 0.5 + 0.5*cos(iTime+uv.yxy+vec3(0,2,4));  
8  
9     // Output to screen  
10    fragColor = vec4(col,1.0);  
11 }
```

Below the code, status information indicates it was compiled in 0.0 secs and contains 158 chars.

Shadertoy: fragment shader



<https://www.youtube.com/watch?v=8--5LwHRhjk>

<https://www.shadertoy.com/view/WsSBzh>