

Computação Gráfica

Shaders

Uniforms padrões do Shadertoy (GLSL)

```
uniform vec3 iResolution; // Resolução da Janela  
uniform float iTime; // Float dos segundos passados  
uniform float iTimeDelta;  
uniform float iFrame;  
uniform float iChannelTime[4];  
uniform vec4 iMouse;  
uniform vec4 iDate;  
uniform float iSampleRate;  
uniform vec3 iChannelResolution[4];  
uniform samplerXX iChanneli;
```

Exemplo com Vetores

Um recurso interessante é o **swizzling**, onde você pode combinar e misturar os valores do vetor, por exemplo:

```
vec2 someVec;  
vec4 differentVec = someVec.xyxx;  
vec3 anotherVec = differentVec.zyw;  
vec4 otherVec = someVec.xxxx + anotherVec.yxzy;
```

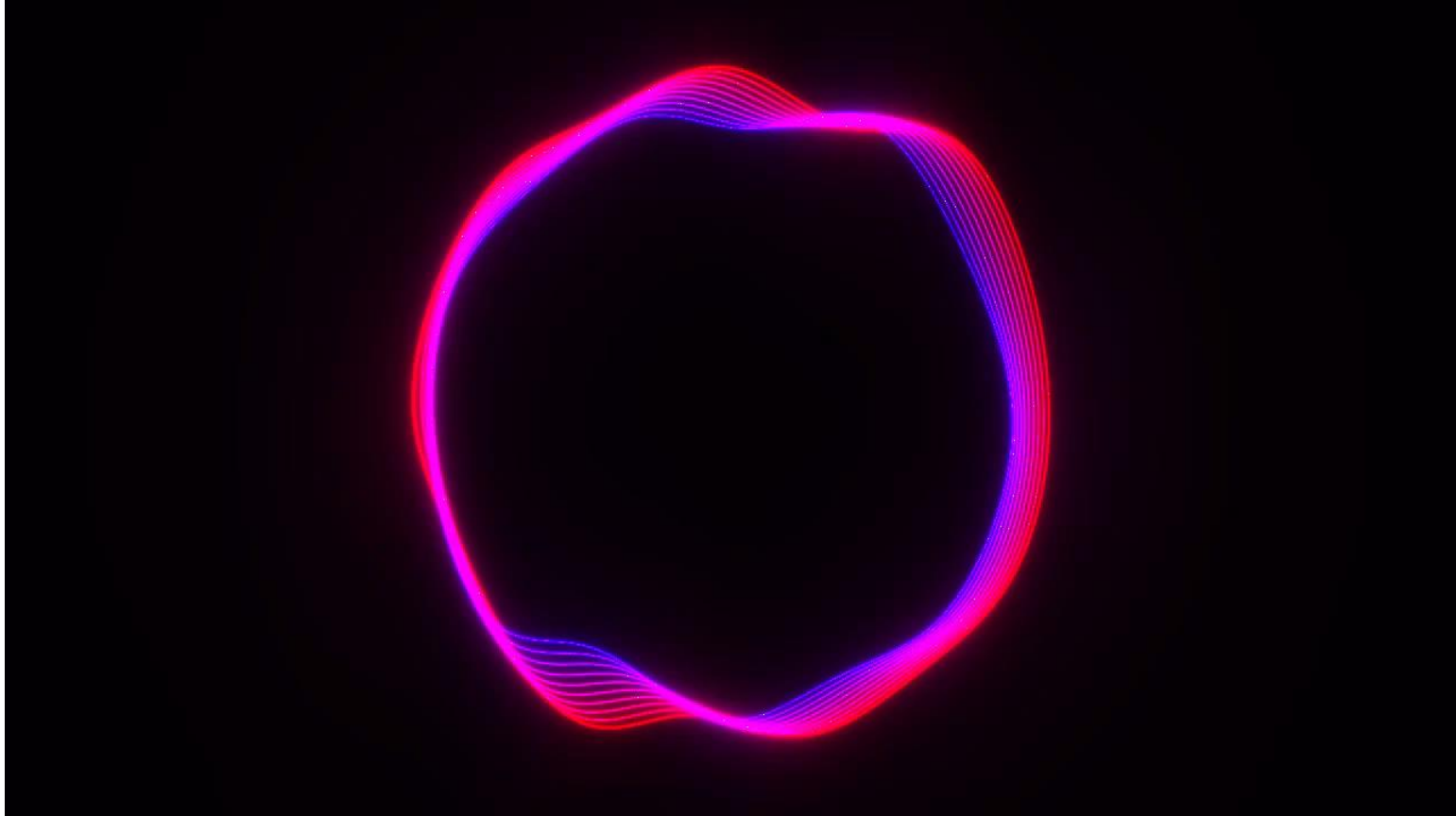
Na construção de vetores, várias formas de combinação também são viáveis, por exemplo:

```
vec2 vect = vec2(0.5, 0.7);  
vec4 result = vec4(vect, 0.0, 0.0);  
vec4 otherResult = vec4(result.xyz, 1.0);
```

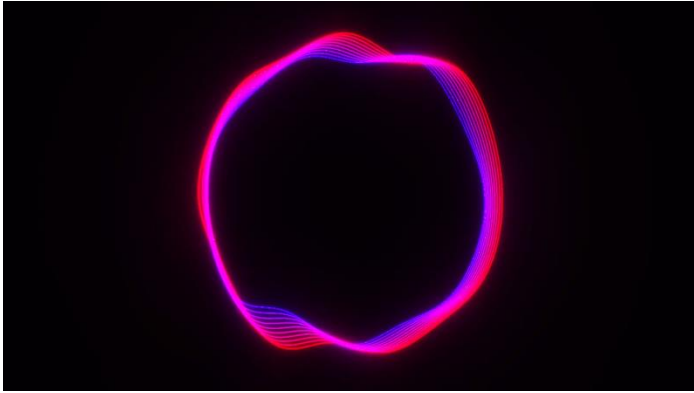
Live Coding



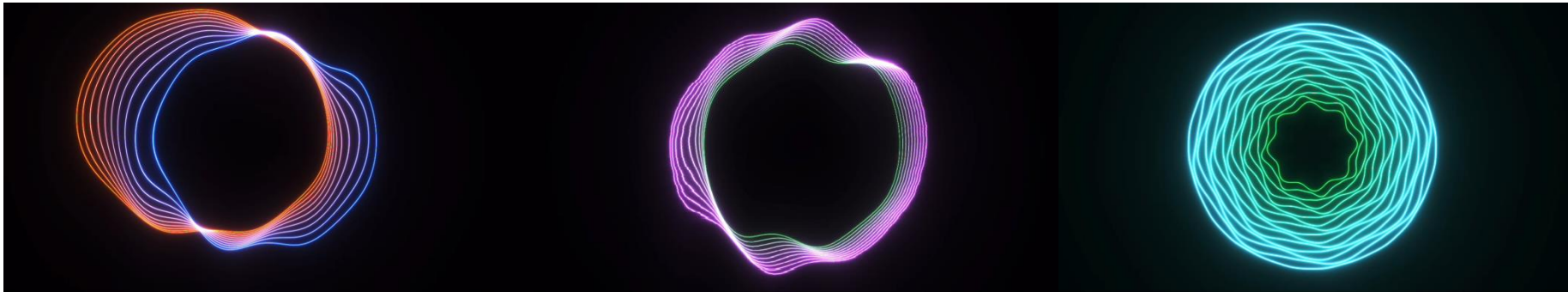
Fragment Shader - Atividade



Fragment Shader - Atividade



Desafios:



Projeto: Dicas

Links do código da aula e outros:

Aula: <https://www.shadertoy.com/view/wfXyWX#>

Funções: <https://www.shadertoy.com/view/lX2BRK>

FFT e música: <https://www.shadertoy.com/view/wcjBWt>

Referências

Baseado:

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

Usando:

<https://inspurnathan.com/posts/49-shadertoy-tutorial-part-3>

Documentações:

<https://iquilezles.org/>

<https://thebookofshaders.com/>

Computação Gráfica

Luciano Soares
<lpsoares@insper.edu.br>

Fabio Orfali
<fabioo1@insper.edu.br>

Gustavo Braga
<gustavobb1@insper.edu.br>