Insper

Computação Gráfica

Shaders

Tipos de Dados (GLSL)

Os tipos básicos de variáveis são:

`int, float, double, uint e bool`

Vetores podem ter 2, 3 ou 4 compomentes:

`vecn: o vetor padrão com n floats.`

`bvecn: um vetor com n booleanos.`

`ivecn: um vetor com n inteiros.`

`uvecn: um vetor com n inteiros sem sinal.`

`dvecn: um vetor com n doubles.`

Você consegue acessar os valores dos vetores com as extensões: .x .y .z .w, ou também com rgba, ou stqp

Referência: https://learnopengl.com/

Estruturas de Controle

If (validação)

```
if (v > 10.0)
{
    ...
}
```

For (inicialização ; validação; cada passo)

```
for (float i = 0.0; i < 10.0; i ++)
{
    ....
}
```

While (validação)

```
while (i < 10.0)
{
    i++;
}
```

API GLSL

https://registry.khronos.org/OpenGL-Refpages/gl4/

Uniforms padrões do Shadertoy

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);
```


Referência: https://learnopengl.com/

Fragment Shader - Exemplos

Simples:

Introdução: https://www.shadertoy.com/view/MXfBW4

Círculo: https://www.shadertoy.com/view/l3fBWr

Movimento: https://www.shadertoy.com/view/43ffW8

Blobs: https://www.shadertoy.com/view/4XXfzj

Médio:

Torus infinito: https://www.shadertoy.com/view/X32BDz

Visualização de função: https://www.shadertoy.com/view/lX2BRK

Torus musical: https://www.shadertoy.com/view/MXsBD8

Complexo:

Torus complexo: https://www.shadertoy.com/view/4XsfWN

Reaction Diffusion: https://www.shadertoy.com/view/4XSfW1

Fragment Shader - Atividade

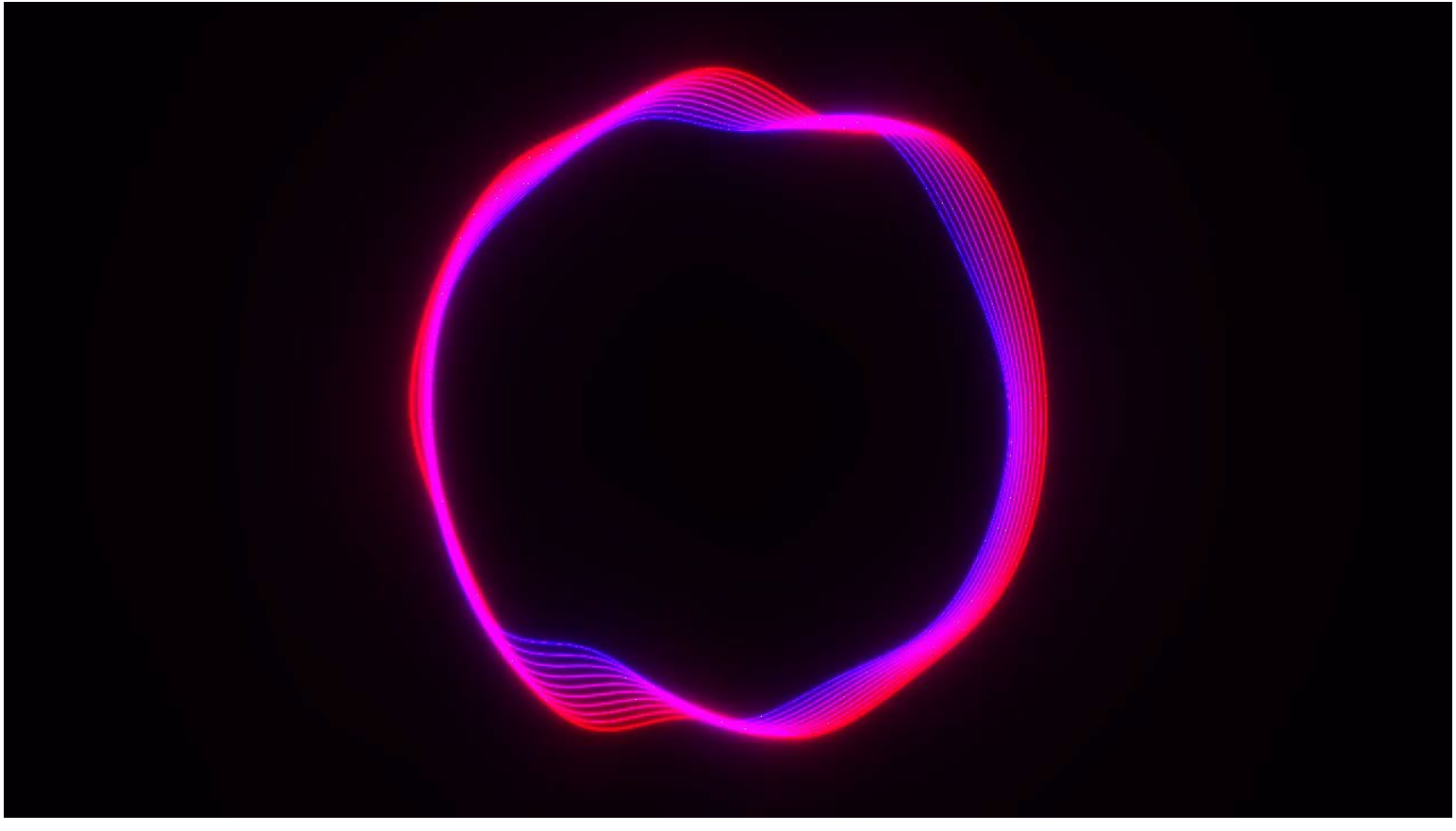
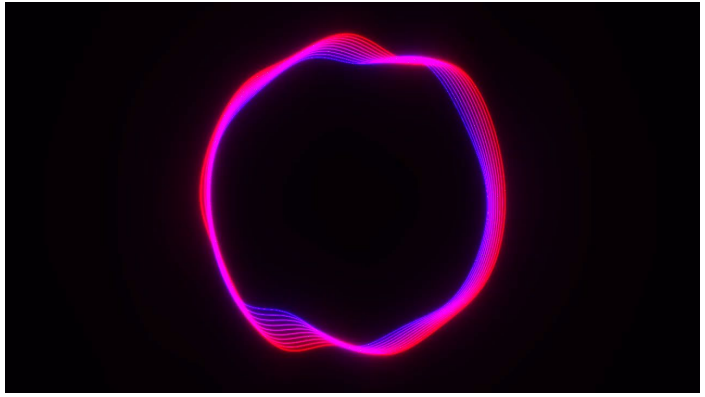


Imagem:F.Andreussi(BUW)

Fragment Shader - Atividade



Desafios:

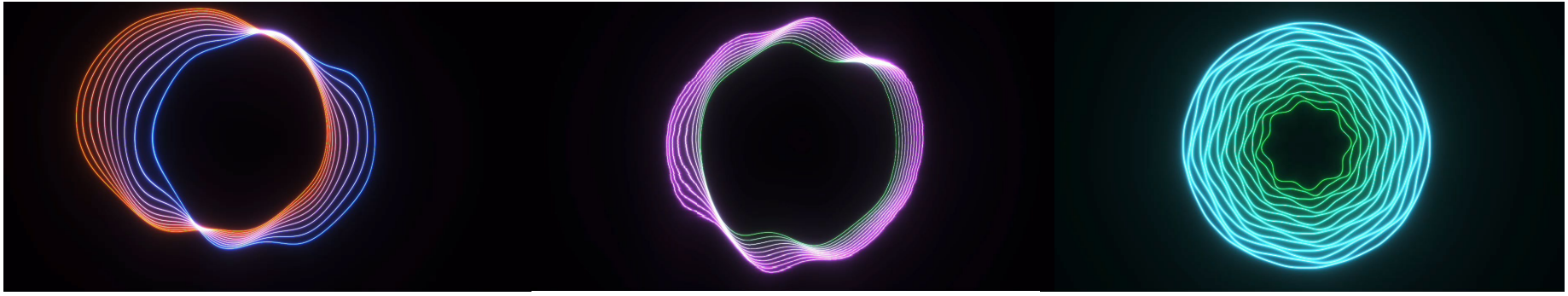


Imagem:F.Andreussi(BUW)

Insper

Fragment Shader – Gabarito atividade

https://www.shadertoy.com/view/43XfR2

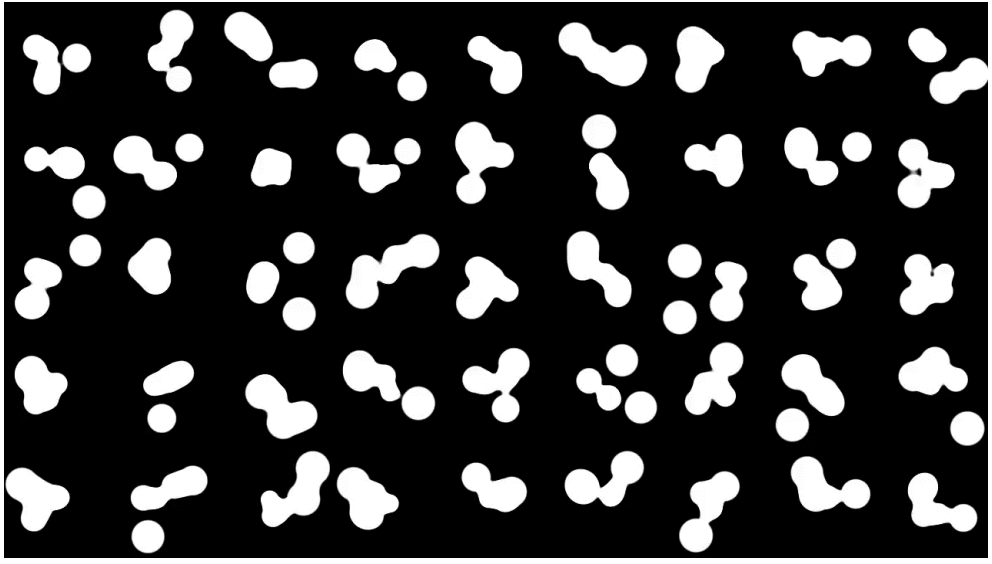
Insper

Correção da atividade na Unity

Próximo projeto

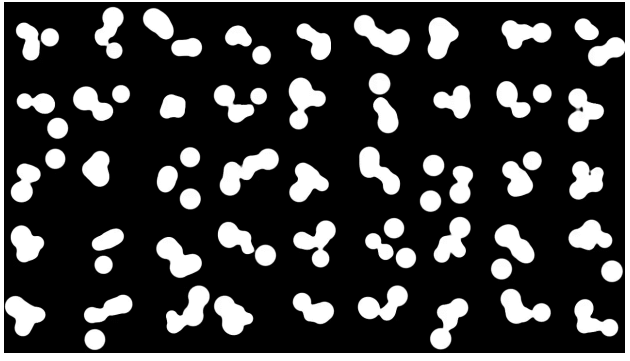
Dicas:

- Crie um círculo.
- Crie um *blob*.
- Crie múltiplos *blobs* com um for().
- Use fract().
- Gere um ruído em bloco.
- Brinque com os parâmetros pra gerar diferenças significativas entre as células e *blobs*.

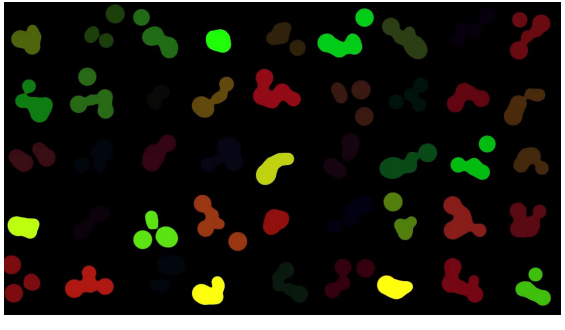


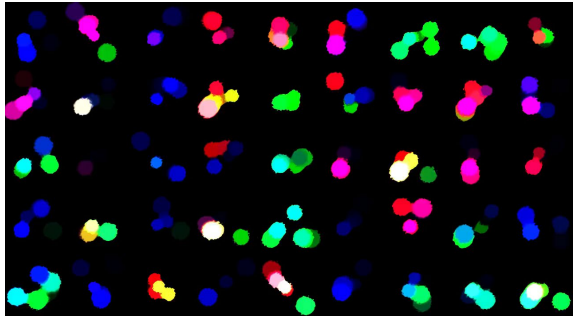
Rubrica

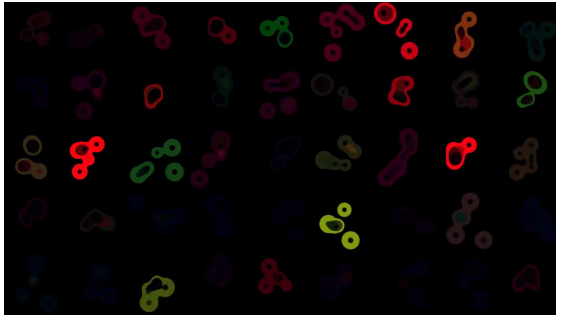
C: Fazer até aqui:



C+ até A+: Colocar algo extra. Exemplos:







Referências

Baseado:
https://www.shadertoy.com/

Usando:
https://inspirnathan.com/posts/49-shadertoy-tutorial-part-3

Documentações:
https://iquilezles.org/
https://thebookofshaders.com/

Insper

Computação Gráfica

Luciano Soares
<lpsoares@insper.edu.br>

Fabio Orfali
<fabioo1@insper.edu.br>

Gustavo Braga
<gustavobb1@insper.edu.br>