NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

Computação Gráfica e Interfaces
Projeto 1 - Relatório

Francisco Rodrigues | 67753
Miguel Rosalino | 68210
(p.2)

2D Parametric Curves

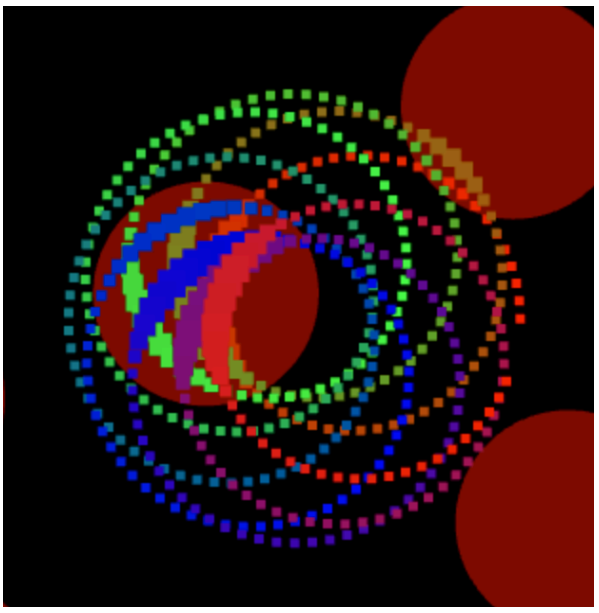
Funcionalidades adicionais

Para além da implementação base do projeto dada pelo enunciado, desenvolvemos ainda dois novos aspetos do programa.

Primeiro, é possível adicionar até 15 círculos no fundo da aplicação com a tecla ‘A’ (e remover com ‘D’). Os círculos não se intercetam e trocam de direção ao colidirem com outro, ou com os limites da janela.

Adicionalmente, os pontos da curva aumentam gradualmente de tamanho ao intersetarem um círculo, criando um novo efeito.

Nós ponderamos implementar o “bounce” e mais alguns métodos no vertex shader para ser mais eficiente. Contudo, chegamos à conclusão que não seria possível por não conseguirmos alterar valores do círculo, como por exemplo as coordenadas do centro e o vetor do movimento, dentro do shader.



Inputs do programa GLSL

Vertex shader

uniform float **u_ratio** // canvas size: width / height

uniform float **u_numberOfCirclePoints** // points per circle
uniform vec3 **u_circle** // circle coords: x,y = center, z = radius
uniform bool **u_isCircle** // if true: points are used for a circle

uniform int **u_curveFamily** // number of curve family to use
uniform float **u_numberOfPoints** // num of curve points to draw
uniform vec3 **u_coefs** // curve parameters: x = a, y = b, z = c
uniform vec2 **u_t** // parameter interval: x = t0, y = t1
uniform vec4 **u_curveLimits** // limits of curve display area
uniform int **u_numberOfCircles** // total number of circles
uniform vec2 **u_circleCenters[15]** // store circles center coords

in float **a_numberOfCurrentPoint** // number of current point

Fragment shader

in float **v_colorVariable** // varies the color of the points (values between 0.0 and 1.0), if 2.0 then set the circle color