Universidade Nova de Lisboa
School of Science and Technology

Electric Fields

Computer Graphics and Interfaces

Authors
António Duarte - 58278
Manuel Pereira - 57973

October 24, 2021

Contents

Chapter 1

GLSL Shaders and their Inputs

Two GLSL programs were implemented for this project, 'gridProgram' and 'chargeProgram', either fully independent from the other, neither sharing buffers nor shaders.

1.1 Grid Point Program and Shaders

Tasked with drawing the two points for each position, and the line connecting them. The visual effect created by shifting one of the points draws the equivalent to electric field lines.

1.1.1 Vertex Shader

Attributes:

vec3 vPosition: World Coordinates for a specific point.

Uniforms:

- **float uTableWidth**: World width limit, used for world-coordinate transmortion to clip coordinates.
- **float uTableHeight**: World height limit, used for world-coordinate transformation to clip coordinates.
- **vec3 uChargePosition[MAX_CHARGES]**: Vector matrix containing the position of all charges, and their effective charge on the z field.
- **int uChargeAmount**: Amount of charges present in the afore-mentioned matrix (serves as loop upper bound).
- **float uLineLength**: Upper limit of line length, affected by a slider.
- **float uFieldScale**: Vector scale modifier, affected by a slider.

Varying:

vec4 fColor: Passes the RGBA color vector from the vertex shader to the fragment shader. Value is given by the colorize function.

1.1.2 Fragment Shader

Varying:

varying vec4 fColor: Receives RGBA color vector from vertex shader and applies it to gl_fragColor.

1.2 Charge Point Program and Shaders

Solely tasked with drawing charges.

1.2.1 Vertex Shader

Attributes:

- **vec2 vPosition**: World coordinates for a specific charge.
- **float vCharge**: The effective charge value of the charge.

Uniforms:

- **float uTableWidth**: World width limit, used for world-coordinate transformation to clip coordinates.
- **float uTableHeight**: World height limit, used for world-coordinate transformation to clip coordinates.

Varying:

float fCharge: Passes the charge signal from the vertex shader to the fragment shader. Value is received through the z-field of the world coordinates -1; 1.

1.2.2 Fragment Shader

Uses procedural shading to apply complex shapes to a sole draw call. Point size is set to 20.0.

BAse procedural shading creates a circle from the point, and discards a horizontal rectangle to form a "-". If the charge received is positive, a vertical central rectangle is also discarded as to shape out "+", and the charge is drawn green. Otherwise, if the charge is negative, it is drawn red.

Varying:

float fCharge: Receives the charge signal from the vertex shader, used for procedural shading technique.

Chapter 2

Extra Functionalities

- *Modifying max line length*: Changes the maximum line length. Accessed through the side panel, in the form of a slider.
- *Modifying field scale*: Changes the scale of the electric force applied on grid points. Accessed through the side panel, in the form of a slider.
- *Modifying rotational speed*: Changes the speed of charge rotation. Accessed through the side panel, in the form of a slider.
- *Charge reset*: Removes all charges on the field, applied by pressing "Backspace".
- *Charge rotation inversion*: Inverts the rotation of the charges, applied by pressing "I".
- *Toggle the UI elements*: Toggles the visibility of the sidebar, applied by pressing "U".