

Identificação: G01-03

Rita Barbosa 64925

Ricardo Gonalo 60519

Projeto:

Field-render.frag

- Uniform ufPlanets -> vec4 com as informaões de cada planeta, sendo essas: a posião do x, a posião do y, o raio e massa de cada planeta.

Field-render.vert

- Attribute vPosition -> vec2 com as posiões reais do canvas (-1.0 a 1.0, -1.0 a 1.0)

- Uniform xScale e yScale -> floats cada um com as escalas horizontais e verticais necessrias para transformar as posiões reais do canvas nas coordenadas de cada fragmento no nosso mundo, que vo ser obtidas atravs do varying fPosition.

- Varying fPosition -> vec2 com as coordenadas dos pontos j transformados nas posiões que variam de -1.5 a 1.5 (no caso do x) e de -h/2 a h/2 (no caso do y, sendo o h a altura total do canvas).

Particle-render.frag

- Varying fLeft -> float com o tempo de vida restante na partcula (em percentagem), que faz com que a mesma fique progressivamente mais branca e transparente.

Particle-render.vert

- Attribute vPosition -> vec2 com as coordenadas das partculas naquele instante.

- Attribute vAge -> float com a idade da partcula naquele instante.

-Attribute vLife -> float com a vida total da partcula.

-Attribute vVelocity -> vec2 com a velocidade da partcula naquele instante.

-Uniform uxScale e uyScale -> floats cada um com as escalas horizontais e verticais necessrias para transformar as posiões reais no canvas nas coordenadas de cada fragmento no nosso mundo.

Particle-update.vert

- Uniform uDeltaTime -> float global para todas as partículas indicando quanto tempo passou.
- Uniform randVelMin -> float com uma velocidade mínima aleatória.
- Uniform randVelMax -> float com uma velocidade máxima aleatória.
- Uniform origin -> vec2 com as coordenadas do ponto de origem das partículas.
- Uniform velDir -> float com a direção da velocidade.
- Uniform maxDirVar -> float com a variação máxima da direção.
- Uniform randLifeMax -> float com o tempo máximo de vida, sendo este aleatório.
- Uniform randLifeMin -> float com o tempo mínimo de vida, sendo este aleatório.
- Uniform uPlanets[MAX_PLANETS] -> vec4 com as informações de cada planeta, sendo essas: a posição do x, a posição do y, o raio e massa de cada planeta. Contém logo as posições máximas, que no caso do nosso projeto são 10 posições.

Atributos (Inputs que refletem o estado de cada partícula antes do update):

- vec2 vPosition -> posição atual.
- float vAge -> idade atual em segundos.
- float vLife -> vida da partícula (em que segundo é que deve morrer).
- vec2 vVelocity -> velocidade atual.

Varying (Outputs. Valores que vão ser capturados e transformados no buffer de feedback):

- vec2 vPositionOut -> Posição depois do update (tem em conta a posição atual, a velocidade e o tempo).

- float vAgeOut -> Idade da partícula depois do update (tem em conta a vida atual e o tempo).
- float vLifeOut -> Vida da partícula (constante exceto quando morre e volta a nascer).
- vec2 vVelocityOut -> Velocidade depois do update (tem em conta a velocidade atual, a aceleração e o tempo).

Implementações Extra:

- O raio do planeta varia de acordo com o tempo que se demora a clicar no botão para criar, em vez do raio ser definido através do movimento do rato.
- As partículas à medida que se aproximam do fim da sua vida vão ficando progressivamente mais transparentes e brancas.
- As partículas vão ficando progressivamente com o tempo de vida mais curto quando dentro do raio de um planeta, sendo que o processo é acelerado por gravidades mais fortes.