# GPU Programming and Visualization

## Graduate Course
## Fall 2024

# Lecture 2

- Buffers
- Uniforms
- Inter-stage variables
- Blending
- Animation loop

- Math next week.

# Gaia example

## Starting point

- https://bit.ly/3C4CnSC

Task 1: Get and run this code

# Gaia example

## Download data

- https://gea.esac.esa.int/archive/

- SELECT TOP 100
  dr3.ra, dr3.dec, dr3.phot_g_mean_mag, dr3.distance_gspphot
  FROM gaiadr3.gaia_source as dr3
  ORDER BY dr3.phot_g_mean_mag ASC

Task 2: Get GAIA data in JSON format.

# Gaia example

## Plotting stars…

```
const staticStorageValues = new Float32Array([
    [217.39232147200883,-62.67607511676666,8.9847
    …
    ].flat());
 device.queue.writeBuffer(staticStorageBuffer, 0, staticStorageValues);

let star = stars[vertexIndex/6];
let size = 0.01;
let x = size*pos[vertexIndex%6].x + sin(star.ra/180.0*3.1415) *
cos(star.dec/180.0*3.1415);
let y = size*pos[vertexIndex%6].y + cos(star.ra/180.0*3.1415) *
cos(star.dec/180.0*3.1415);
let z = sin(star.dec/180.0*3.1415);
return vec4f( x, y, z, 1.0);
```

# WebGPU

## Inter-stage variables

- Passing data from vertex to fragment shader
- Can be anything, often: colour or normals

# Gaia example

## Making stars more star like

```
struct InOut {
    @builtin(position) position: vec4f,
    @location(0) tex: vec2f,
};

@vertex fn vs( @builtin(vertex_index) vertexIndex : u32 ) -> InOut {
   ...
  var inOut: InOut;
  inOut.position = vec4f( x, y, z, 1.0);
  inOut.tex = pos[vertexIndex%6];
  return inOut;
}
@fragment fn fs(inOut: InOut) -> @location(0) vec4f {
      let r = 1.0-length(inOut.tex);
      return vec4f(r,r,r,1);
}
```

# WebGPU

## Blending

https://www.andersriggelsen.dk/glblendfunc.php

# WebGPU

## Blending

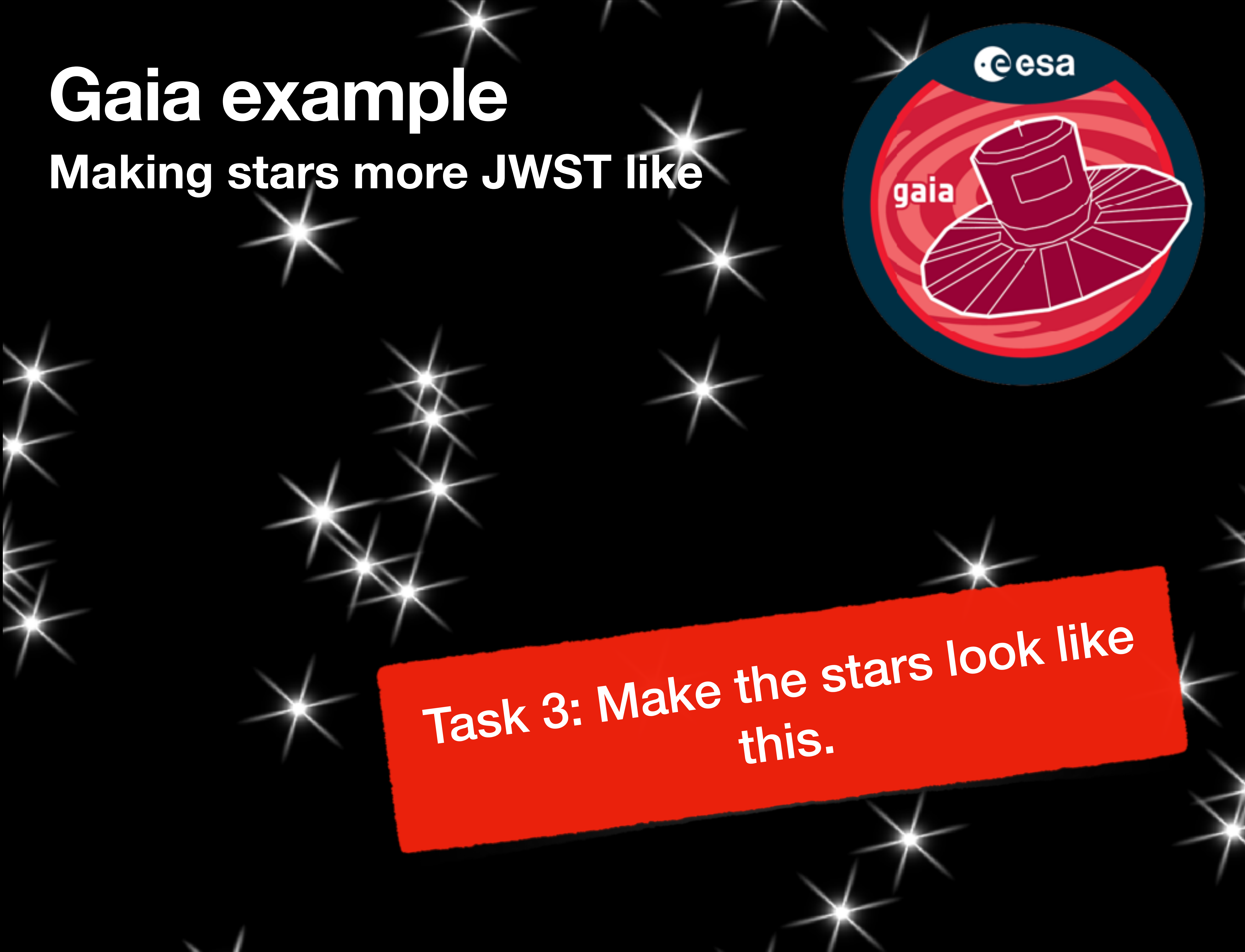
```
fragment: {
   module,
   targets: [
    { format: presentationFormat,
     blend: {
      color: {
       srcFactor: 'src-alpha',
       dstFactor: 'one-minus-src-alpha'
      },
      alpha: {
       srcFactor: 'src-alpha',
       dstFactor: 'one-minus-src-alpha'
      },
     },
    }
   ],
  },
```

# Gaia example

## Making stars more JWST like

# Gaia example

## Making stars more JWST like



Task 3: Make the stars look like this.

# Gaia example

## Making stars more JWST like

```
@fragment fn fs(inOut: InOut) -> @location(0) vec4f {
        let phi = atan2(inOut.tex.y,inOut.tex.x);
        var r = 1.0-length(inOut.tex);
        let a = pow(clamp(sin(phi*6.0),0.0,1.0),5.0/r);
        var f = pow(1.1*r,8.0) + r*a;
        return vec4f(1,1,1,f);
}
```

# Gaia example

## Animation

```
struct UniformData {        time: f32,    };
@group(0) @binding(1) var<uniform> uniformData: UniformData;

const uniformBufferSize = 4;
const uniformBuffer = device.createBuffer({
    size: uniformBufferSize,
    usage: GPUBufferUsage.UNIFORM |GPUBufferUsage.COPY_DST,
  });
const uniformValues = new Float32Array(uniformBufferSize / 4);

{ binding: 1, resource: { buffer: uniformBuffer }},
```

# Gaia example

## Animation

```
var time = 0.0;
function render() {
   uniformValues.set([time], 0);
   time += 0.1;
   device.queue.writeBuffer(uniformBuffer, 0, uniformValues);
   …
   requestAnimationFrame(render);
}
requestAnimationFrame(render);

var star = stars[vertexIndex/6];
star.ra += uniformData.time;
```

# Gaia example

## Magnitude

Task 4: Change stars size based on their magnitude

$$m_1 - m_{\mathrm{ref}} = -2.5 \log_{10}\left(\frac{F_1}{F_{\mathrm{ref}}}\right)$$

# Gaia example

## Magnitude

```
let flux = pow(-star.m/2.5,10);
let size = max(0.02,0.0002*sqrt(flux));
```

# Gaia example

## Final code

https://gist.github.com/hannorein/4e4304c2af10bdfba39c164b3dc0f142

https://bit.ly/3YpoRjZ