#### C++ arrays

Tessellation

Vertex array

Index array

Circle ring

Sphere

Sphere

Interpolation

Cubic

Assignment 2

Tessellation Interpolation

# Tessellation & Interpolation

**EDAF80: Computer Graphics** 

Rikard Olajos





# **AGENDA**

1 C++ arrays

Tessellation

Interpolation

Assignment 2

## **RAW ARRAYS: STACK & HEAP ALLOCATION**

### C++ arrays

Vertex array
Index array
Parametric surfaces
Circle ring
Sphere
Torus

Interpolation Linear Cubic

Assignment .
Tessellation

Tessellation Interpolation Demo

## Stack

```
float numbers[3];
numbers[0] = 1.0f;
...
```

## Stack: direct initialization

```
float numbers[3] = { 1.0f, 2.0f, 3.0f };
```

# Heap

```
float* numbers = new float[3];
number[0] = 1.0f;
...
delete[] numbers;
```

# **STL ARRAYS: VECTOR & ARRAY**

### C++ arrays

Vertex array Index array Parametric surfaces Circle ring Sphere

Interpolation Linear Cubic

Assignment 2
Tessellation
Interpolation

### **Includes**

```
#include <array>
#include <vector>
```

### Initialization

```
std::array<int, 3> arr;  // Static array with 3 integers
std::vector<float> vec;  // Dynamic array with floats (on the heap)
```

### Element access & size

```
arr[0] = 1;  // Set first element to 1
vec.push_back(1.0f);  // Add 1.0f to end of vector
std::cout << vec[0];  // Print first element of 'vec'
std::cout << vec.size();  // Print number of elements in 'vec'</pre>
```

## **TESSELLATION**

# Tessellation

Vertex array
Index array
Parametric surfa

Sphere
Torus

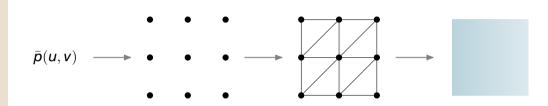
iterpolation

Linear

A --!----

Tossellation

Interpolation



- Setup vertex array
- Setup index array (triangulate)

## **CREATE VERTEX ARRAY**

#### C++ arrays

Vertex array Index array Parametric surfaces

Circle ring Sphere

Interpolation Linear

Assignment

Tessellation Interpolation • Create vertex array (e.g. 3×3 vertices)

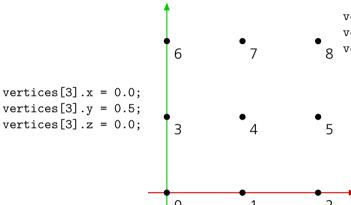
```
auto vertices = std::vector<glm::vec3>(9);
```

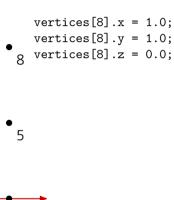
Assign vertex

```
vertices[index] = glm::vec3(x, y, z);
```

### **VERTEX ARRAY LAYOUT**

#### Vertex array





### TRIANGULATION

Index array

• Create index array  $(2 \times (3-1)(3-1))$  triangles)

```
auto indices = std::vector<glm::uvec3>(8);
```

Define triangle (indices for the three vertices)

```
indices[index] = glm::uvec3(v0, v1, v2);
```



#### C++ arrays

### Tossollatio

#### Vertex arra

Index array
Parametric surface
Circle ring

Sphere Torus

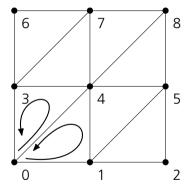
### nterpolatio

Cubic

Assignme

Tessellation

Demo



### **INDEX ARRAY LAYOUT**

- Indices in counter-clockwise order (CCW)
- Backface culling is off by default, turn it on to improve performance!

```
indices[0].x = 0;
indices[0].y = 1;
indices[0].z = 4;
indices[1].x = 0;
indices[1].y = 4;
indices[1].z = 3;
```



## **PARAMETRIC SURFACES**

#### C++ arrays

#### Taccallation

Vertex array

#### Parametric surfaces

Circle rin

Spnere

#### nterpolatio

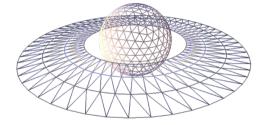
Cubic

A ---!-----

#### , toolgrilleric

ressellation

Demo



#### C++ arrays

#### Tossallation

Vertex array

#### Parametric surfaces

Circle ring Sphere

iorus

### Interpolation

Linear

#### Assignmen

Tessellation Interpolation

### **PARAMETRIC SURFACE & TANGENT SPACE**

• Map surface from 2D:

$$\bar{p}(x,y,z) = \bar{p}(u,v)$$

•  $\mathbb{R}^2 \mapsto \mathbb{R}^3$ 



## **PARAMETRIC SURFACE & TANGENT SPACE**

#### C++ arrays

#### Toscollation

Vertex array

### Parametric surfaces

Circle ring

Sphere

lakawa a laki

### Linear

Cubic

#### Assignmen

Tessellation Interpolation Map surface from 2D:

$$\bar{p}(x,y,z) = \bar{p}(u,v)$$

- $\mathbb{R}^2 \mapsto \mathbb{R}^3$ 
  - Tangent  $t = \frac{\partial \bar{p}}{\partial u}$



### PARAMETRIC SURFACE & TANGENT SPACE

Parametric surfaces

• Map surface from 2D:

$$\bar{p}(x,y,z) = \bar{p}(u,v)$$

- $\mathbb{R}^2 \mapsto \mathbb{R}^3$ 
  - Tangent  $t = \frac{\partial \bar{p}}{\partial u}$  Binormal  $b = \frac{\partial \bar{p}}{\partial v}$



### PARAMETRIC SURFACE & TANGENT SPACE

### Parametric surfaces

• Map surface from 2D:

$$\bar{p}(x,y,z) = \bar{p}(u,v)$$

- $\mathbb{R}^2 \mapsto \mathbb{R}^3$ 
  - Tangent  $t = \frac{\partial \bar{p}}{\partial u}$

  - Binormal  $b = \frac{\partial \bar{p}}{\partial v}$  Normal  $n = \frac{\partial \bar{p}}{\partial u} \times \frac{\partial \bar{p}}{\partial v}$



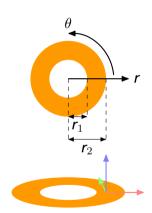
# **CIRCLE RING**

Circle ring

$$\bar{\pmb{\rho}}(\pmb{r},\theta) = \left\{\begin{matrix} \pmb{r}\cos(\theta) \\ \pmb{r}\sin(\theta) \\ 0 \end{matrix}\right\} \text{ for } \begin{matrix} \pmb{r}_1 \leq \pmb{r} \leq \pmb{r}_2 \\ 0 \leq \theta < 2\pi \end{matrix}$$

$$t = rac{\partial ar{p}}{\partial r} = egin{cases} \cos( heta) \ \sin( heta) \ 0 \end{cases}$$

$$b = \frac{\partial \bar{p}}{\partial \theta} = \begin{cases} -r\sin(\theta) \\ r\cos(\theta) \\ 0 \end{cases}$$



## **SPHERE**

C++ arrays

Tassallatio

Vertex array Index array

Parametric surface

Circle ring Sphere

Torus

nterpolation

Assignment

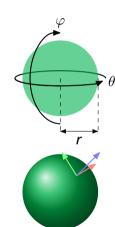
Tessellation

Interpolation

$$\bar{\boldsymbol{\rho}}(\theta,\varphi) = \begin{cases} r\sin(\theta)\sin(\varphi) \\ -r\cos(\varphi) \\ r\cos(\theta)\sin(\varphi) \end{cases} \text{ for } \begin{array}{l} 0 \leq \theta \leq 2\pi \\ 0 \leq \varphi \leq \pi \end{array}$$

$$egin{aligned} oldsymbol{t} &= rac{\partial ar{p}}{\partial heta} = \left\{ egin{aligned} oldsymbol{r}\cos( heta)\sin(arphi) \ -oldsymbol{r}\sin( heta)\sin(arphi) \end{array} 
ight\} \end{aligned}$$

$$b = \frac{\partial \bar{p}}{\partial \varphi} = \begin{cases} r \sin(\theta) \cos(\varphi) \\ r \sin(\varphi) \\ r \cos(\theta) \cos(\varphi) \end{cases}$$



## **SPHERE**

C++ arrays

Vertex array Index array

Circle ring

Sphere

Interpolation

Cubic

Assignment

Tessellation Interpolation Demo

$$\bar{\pmb{\rho}}(\theta,\varphi) = \begin{cases} r\sin(\theta)\sin(\varphi) \\ -r\cos(\varphi) \\ r\cos(\theta)\sin(\varphi) \end{cases} \text{ for } \begin{array}{l} 0 \leq \theta \leq 2\pi \\ 0 \leq \varphi \leq \pi \end{array}$$

$$egin{aligned} oldsymbol{t} = rac{\partial ar{oldsymbol{p}}}{\partial heta} = \left\{ egin{aligned} oldsymbol{r}\cos( heta)\sin(arphi) \ -oldsymbol{r}\sin( heta)\sin(arphi) \end{array} 
ight\} \end{aligned}$$

$$b = \frac{\partial \bar{p}}{\partial \varphi} = \begin{cases} r \sin(\theta) \cos(\varphi) \\ r \sin(\varphi) \\ r \cos(\theta) \cos(\varphi) \end{cases}$$

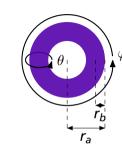
- t and b can be simplified since only direction is important
- t needs to be simplified, or it will be undefined for  $\varphi = 0$

$$\bar{p}(\theta,\varphi) = \begin{cases} (r_{a} + r_{b}\cos(\theta))\cos(\varphi) \\ (r_{a} + r_{b}\cos(\theta))\sin(\varphi) \\ -r_{b}\sin(\theta) \end{cases} \text{ for } \begin{array}{l} 0 \leq \theta \leq 2\pi \\ 0 \leq \varphi \leq 2\pi \end{array}$$

$$t = \frac{\partial \bar{p}}{\partial \theta} = \begin{cases} -r_b \sin(\theta) \cos(\varphi) \\ -r_b \sin(\theta) \sin(\varphi) \\ -r_b \cos(\theta) \end{cases}$$

$$b = \frac{\partial \bar{p}}{\partial \varphi} = \begin{cases} -(r_a + r_b \cos(\theta)) \sin(\varphi) \\ (r_a + r_b \cos(\theta)) \cos(\varphi) \\ 0 \end{cases}$$

### **TORUS**





#### C++ arrays

#### Toccollation

Vertex array

Index array

Circle ring

Sphere

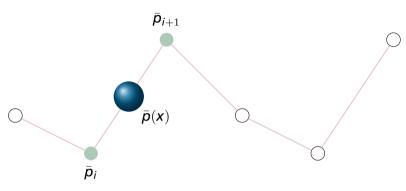
#### nterpolation

Linear

Assignmen

Tessellation

Demo



$$\bar{\boldsymbol{\rho}}(\boldsymbol{x}) = \begin{bmatrix} 1 & \boldsymbol{x} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{\rho}}_i \\ \bar{\boldsymbol{\rho}}_{i+1} \end{bmatrix} \text{ for } \boldsymbol{x} \in [0,1]$$

#### C++ arrays

#### Toccollation

Vertex array

Index array

Circle ring

Sphere

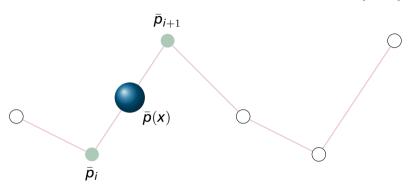
#### nterpolation

Linear

Assignmen

Tessellation

Demo



$$\bar{\pmb{p}}(\pmb{x}) = \begin{bmatrix} 1 & \pmb{x} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \bar{\pmb{p}}_i \\ \bar{\pmb{p}}_{i+1} \end{bmatrix}$$
 for  $\pmb{x} \in [0,1]$ 

$$\bar{\boldsymbol{p}}(\boldsymbol{x}) = (1 - \boldsymbol{x})\bar{\boldsymbol{p}}_i + \boldsymbol{x}\bar{\boldsymbol{p}}_{i+1}$$

#### C++ arrays

#### Tanadlatian

Vertex array

Index array

Circle ring

Sphere

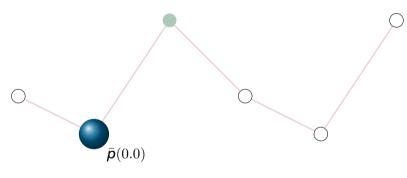
#### nterpolation

Linear

Assianmen:

Tessellation

Interpolation



$$\bar{\pmb{\rho}}(\pmb{x}) = \begin{bmatrix} 1 & \pmb{x} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \bar{\pmb{\rho}}_i \\ \bar{\pmb{\rho}}_{i+1} \end{bmatrix} \text{ for } \pmb{x} \in [0,1]$$

$$\bar{\boldsymbol{p}}(\boldsymbol{x}) = (1 - \boldsymbol{x})\bar{\boldsymbol{p}}_i + \boldsymbol{x}\bar{\boldsymbol{p}}_{i+1}$$

#### C++ arrays

#### Tossallation

Vertex array Index array Parametric surfaces Circle ring

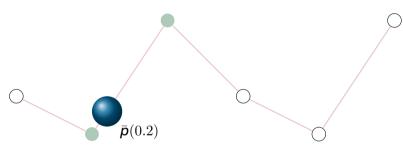
#### nternolatio

Linear

Assignm

Tessellation

Demo



$$\bar{\boldsymbol{p}}(\boldsymbol{x}) = \begin{bmatrix} 1 & \boldsymbol{x} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{p}}_i \\ \bar{\boldsymbol{p}}_{i+1} \end{bmatrix}$$
 for  $\boldsymbol{x} \in [0, 1]$ 

$$\bar{\boldsymbol{p}}(\boldsymbol{x}) = (1 - \boldsymbol{x})\bar{\boldsymbol{p}}_i + \boldsymbol{x}\bar{\boldsymbol{p}}_{i+1}$$

#### C++ arrays

#### Tossollation

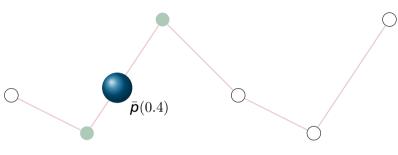
Vertex array Index array Parametric surfaces Circle ring

#### nternolation

Linear

### Assignment

Tessellation Interpolation



$$ar{m{p}}(m{x}) = egin{bmatrix} 1 & m{x} \end{bmatrix} egin{bmatrix} 1 & 0 \ -1 & 1 \end{bmatrix} egin{bmatrix} ar{m{p}}_i \ ar{m{p}}_{i+1} \end{bmatrix}$$
 for  $m{x} \in [0,1]$ 

$$\bar{\boldsymbol{p}}(\boldsymbol{x}) = (1 - \boldsymbol{x})\bar{\boldsymbol{p}}_i + \boldsymbol{x}\bar{\boldsymbol{p}}_{i+1}$$

#### C++ arrays

#### Tossollation

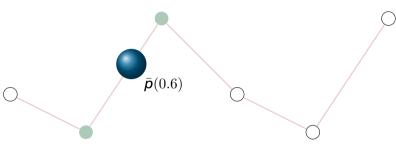
Vertex array Index array Parametric surface: Circle ring

#### nternolatio

Linear

### Assignmen

Tessellation Interpolation



$$\bar{\pmb{p}}(\pmb{x}) = \begin{bmatrix} 1 & \pmb{x} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \bar{\pmb{p}}_i \\ \bar{\pmb{p}}_{i+1} \end{bmatrix}$$
 for  $\pmb{x} \in [0,1]$ 

$$\bar{\boldsymbol{p}}(\boldsymbol{x}) = (1 - \boldsymbol{x})\bar{\boldsymbol{p}}_i + \boldsymbol{x}\bar{\boldsymbol{p}}_{i+1}$$

#### C++ arrays

#### Tossallation

Vertex array Index array Parametric surface

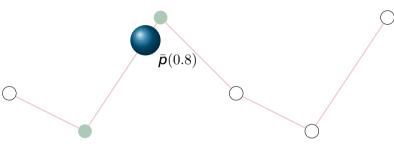
Sphere Torus

#### nterpolation

Linear

Assignmen

Tessellation Interpolation



$$\bar{\pmb{\rho}}(\pmb{x}) = \begin{bmatrix} 1 & \pmb{x} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \bar{\pmb{\rho}}_i \\ \bar{\pmb{\rho}}_{i+1} \end{bmatrix} \text{ for } \pmb{x} \in [0,1]$$

$$\bar{\boldsymbol{p}}(\boldsymbol{x}) = (1 - \boldsymbol{x})\bar{\boldsymbol{p}}_i + \boldsymbol{x}\bar{\boldsymbol{p}}_{i+1}$$

#### C++ arrays

#### Tossallation

Vertex array Index array

Parametric surfac

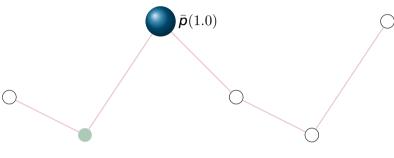
Sphere Torus

#### nterpolation

Linear

Assignmen

Tessellation Interpolation



$$ar{m{p}}(m{x}) = egin{bmatrix} 1 & m{x} \end{bmatrix} egin{bmatrix} 1 & 0 \ -1 & 1 \end{bmatrix} egin{bmatrix} ar{m{p}}_i \ ar{m{p}}_{i+1} \end{bmatrix}$$
 for  $m{x} \in [0,1]$ 

$$\bar{\boldsymbol{p}}(\boldsymbol{x}) = (1 - \boldsymbol{x})\bar{\boldsymbol{p}}_i + \boldsymbol{x}\bar{\boldsymbol{p}}_{i+1}$$

#### C++ arrays

Toccollation

Vertex array Index array

Circle ring
Sphere

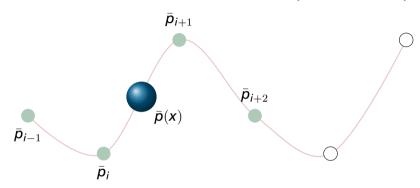
Interpolation

Cubic

Assiann

Tessellation Interpolation

## **CUBIC INTERPOLATION (CATMULL-ROM)**



$$\bar{q}(\mathbf{x}) = \begin{bmatrix} 1 & \mathbf{x} & \mathbf{x}^2 & \mathbf{x}^3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\tau & 0 & \tau & 0 \\ 2\tau & \tau - 3 & 3 - 2\tau & -\tau \\ -\tau & 2 - \tau & \tau - 2 & \tau \end{bmatrix} \begin{bmatrix} \bar{p}_{i-1} \\ \bar{p}_{i} \\ \bar{p}_{i+1} \\ \bar{p}_{i+2} \end{bmatrix} \text{ for } \mathbf{x} \in [0, 1]$$

#### C++ arrays

Tossallation

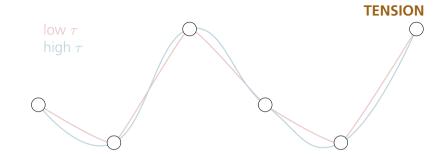
Index array
Parametric surface
Circle ring
Sphere

Interpolation

Cubic

# Assignmen

Tessellation Interpolation Demo



$$\bar{q}(\mathbf{x}) = \begin{bmatrix} 1 & \mathbf{x} & \mathbf{x}^2 & \mathbf{x}^3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\tau & 0 & \tau & 0 \\ 2\tau & \tau - 3 & 3 - 2\tau & -\tau \\ -\tau & 2 - \tau & \tau - 2 & \tau \end{bmatrix} \begin{bmatrix} \bar{\mathbf{p}}_{i-1} \\ \bar{\mathbf{p}}_{i} \\ \bar{\mathbf{p}}_{i+1} \\ \bar{\mathbf{p}}_{i+2} \end{bmatrix} \text{ for } \mathbf{x} \in [0, 1]$$

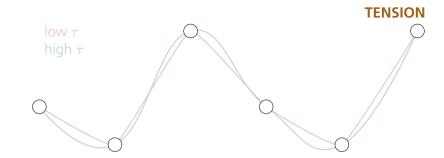
#### C++ arrays

lessellation
Vertex array
Index array
Parametric surface
Circle ring
Sphere

Interpolation Linear

Linear Cubic

Assignment 2
Tessellation
Interpolation



$$\bar{q}(\mathbf{x}) = \begin{bmatrix} 1 & \mathbf{x} & \mathbf{x}^2 & \mathbf{x}^3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\tau & 0 & \tau & 0 \\ 2\tau & \tau - 3 & 3 - 2\tau & -\tau \\ -\tau & 2 - \tau & \tau - 2 & \tau \end{bmatrix} \begin{bmatrix} \bar{\mathbf{p}}_{i-1} \\ \bar{\mathbf{p}}_{i} \\ \bar{\mathbf{p}}_{i+1} \\ \bar{\mathbf{p}}_{i+2} \end{bmatrix} \text{ for } \mathbf{x} \in [0, 1]$$

- ullet au= tension, how "stiff" the curve is at the control points
- Keep within [0, 1]
- Good initial value: 0.5

### **ASSIGNMENT 2**

#### C++ array

Vertex array Index array Parametric surfaces Circle ring Sphere

Linear Cubic

Assignment 2

Tessellation Interpolation Demo

- Tessellate objects from parametric equations
- Linear and cubic interpolation
- Files you have to modify
  - src/EDAF80/assignment2.cpp
  - src/EDAF80/parametric\_shapes.cpp
  - src/EDAF80/interpolation.cpp

## **TESSELLATION**

#### C++ array

Vertex array Index array Parametric surface Circle ring

Torus

Linear

Assignment

Tessellation Interpolation • Implement function bodies in src/EDAF80/parametric\_shapes.cpp

```
bonobo::mesh_data parametric_shapes::createQuad(...);
bonobo::mesh_data parametric_shapes::createSphere(...);
bonobo::mesh_data parametric_shapes::createTorus(...); // Optional
```

### **TESSELLATION**

#### C++ arrays

Vertex array Index array Parametric surfaces Circle ring Sphere

Interpolation Linear

Assignment :

Tessellation

Implement function bodies in src/EDAF80/parametric\_shapes.cpp

```
bonobo::mesh_data parametric_shapes::createQuad(...);
bonobo::mesh_data parametric_shapes::createSphere(...);
bonobo::mesh_data parametric_shapes::createTorus(...); // Optional
```

- Look at createCircleRing(...) in the same file for guidance
- Make sure parameter definitions and ranges are correct
  - Circle ring:  $0 \le \theta < 2\pi, r_1 \le r \le r_2$
  - Sphere:  $0 \le \theta \le 2\pi, 0 \le \varphi \le \pi$

## **DEBUGGING NORMALS**

#### C++ arrays

#### Vortov array

Index array
Parametric surface
Circle ring
Sphere

### Interpolation

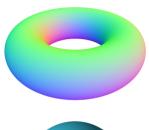
Cubic

Assignmen

#### Tessellation

Interpolation Demo

- Colourize, use the "Normals" shader to represent normals as RGB values
- Inspect illumination, is illumination consistent with the location of the light source?





## **COLOURIZING NORMALS**

### C++ arrays

• Map from [-1,1] to [0,1]

•  $(N \cdot 0.5) + 0.5$ 

• Example: Z axis (0,0,1) becomes (0.5,0.5,1)

• Values are normalized:  $(1,1,1)\mapsto (\frac{1}{\sqrt{3}},\frac{1}{\sqrt{3}},\frac{1}{\sqrt{3}})$ 

Interpolation

Assianment

Assignment . Tessellation

Tessellation Interpolation



$$(1,1,1) \mapsto (1,1,1)$$



$$(-1, -1, -1) \mapsto (0, 0, 0)$$



$$(1,0,0) \mapsto (1,0.5,0.5)$$

### **TOOLS IN GUI**

Tessellation

- Change cull mode: Disabled, Back faces, Front faces
- Change polygon mode: Fill, Line, Point
- Change shaders: Fallback, Diffuse, Normal, Tangent, Bitangent, Texture coords

### **INTERPOLATION**

#### C++ arrays

Vertex array Index array Parametric surfaces Circle ring

Interpolation

Cubic

Tessellation

Interpolation

- Implement linear and cubic interpolation
- Implement function bodies in src/EDAF80/interpolation.cpp

```
glm::vec3 interpolation::evalLERP(...);
glm::vec3 interpolation::evalCatmullRom(...);
```

- Test with just 2 (LERP) or 4 (cubic) points first
- Animate an object along the path using both function and the predefined control points
- use\_linear and catmull\_rom\_tension variables are bound to the GUI and should be used

### **IMPLEMENTATION SKETCH**

Tessellation
Vertex array
Index array
Parametric surfaces
Circle ring
Sphere

Interpolation Linear

Assignment

Tessellation Interpolation

```
// Tnit:
std::array<glm::vec3, N> control points = { ... };
float path_pos = 0.0f;
float velocity = ...
// Main loop:
    int i = floor(path pos);
    // Pick indices for interpolation: i-1, i, i+1, i+2
    // Make sure indices wrap: 0, 1, .... N-1. 0, 1. ...
    // Call interpolation function with points from control points
    path_pos += velocity;
```

# **DEMO**

#### C++ arrays

#### Tessellation

Vertex array

Index array

Circle ring

Sphere

#### taka wa a bakta i

#### riccipolatio

Cubic

#### Assignment 2

Tessellation

Interpolation

Demo

