CSE4203: Computer Graphics

Lecture – 1

Introduction

Outline

- What is CG
- CG Areas
- Major Applications
- Graphics API

What is CG? (1/1)

- The term computer graphics describes any use of computers to create and manipulate images.
  - Graphics can be 2D or 3D
  - Images can be completely synthetic or can be produced by manipulating photographs.

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

CG Areas (1/4)

- **Modeling:**
  - deals with the mathematical specification:
    - shape and appearance properties in a way that can be stored on the computer.

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

CG Areas: Metaphor

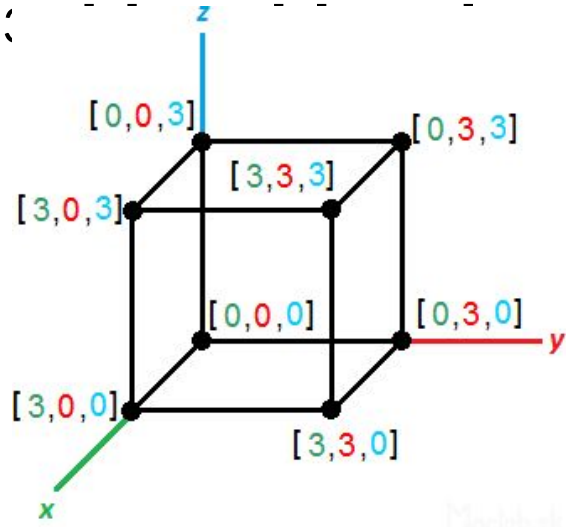
Source:
https://youtu.be/6Sv4oXSTAms

CG Areas (2/4)

- **Modeling:** Example –
  - an object can be described as 3D coordinates:

    [0, 0, 3], [0, 3, 3], [0, 3, 0], [0, 0, 0],
    [3, 0, 3], [3, 3

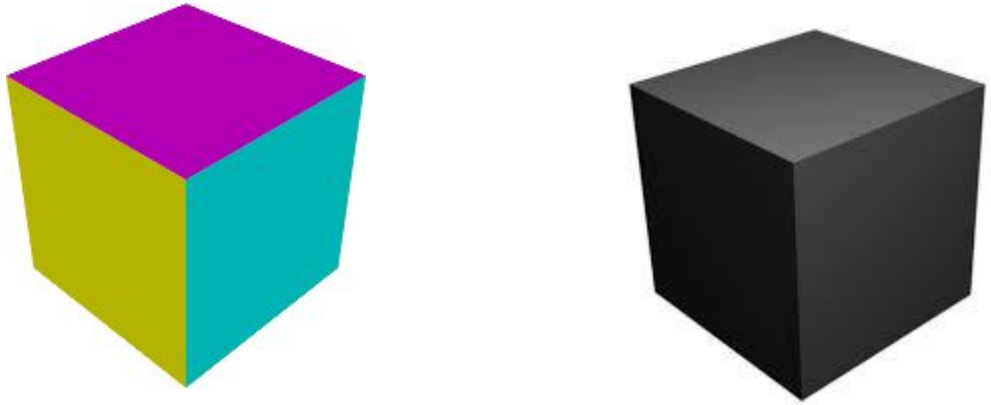
  - connect the points



Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

CG Areas (3/4)

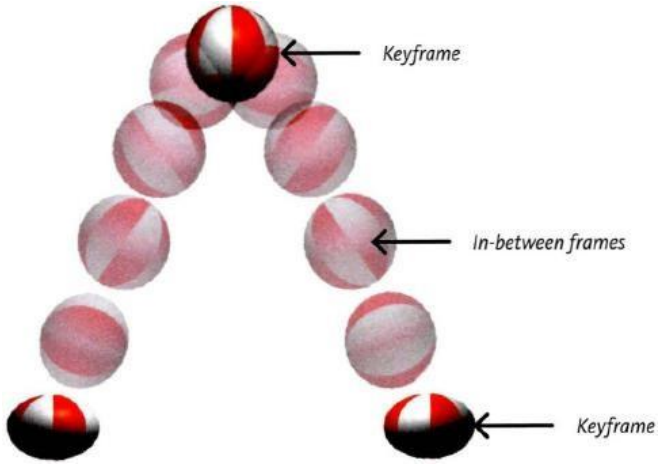
- **Rendering:**
  - a term inherited from art
  - deals with the creation of shaded images from 3D computer models.



Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

CG Areas (4/4)

- **Animation:**
  - creates an **illusion of motion** through sequences of images.
  - uses modeling and rendering but adds ***movement*** over time



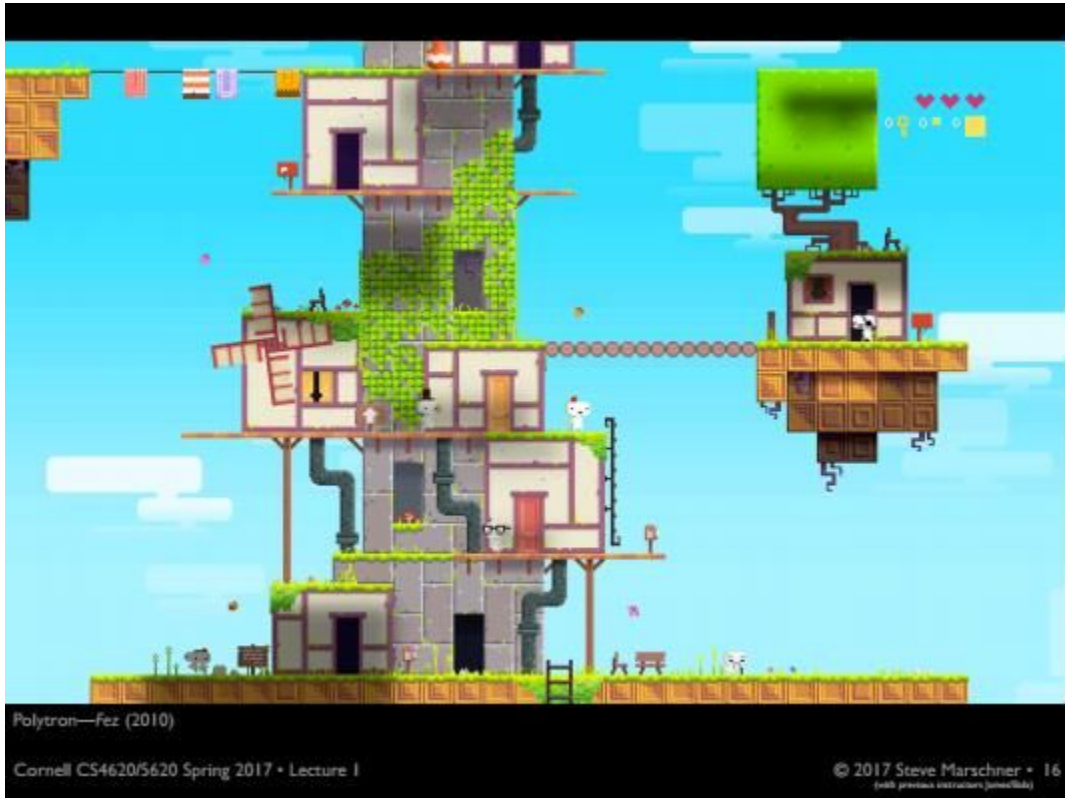
Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Major Applications (1/12)

- Video games
- Cartoons
- Visual effects
- Animated films
- CAD/CAM
- Simulation
- Mixed Reality
- Information visualization

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Major Applications (2/12)



Games (2D)

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Major Applications (3/12)

Games (3D)

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Major Applications (4/12)

Movies (VFX)

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Major Applications (5/12)

Movies (Animated)

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Major Applications (6/12)

Mixed Reality

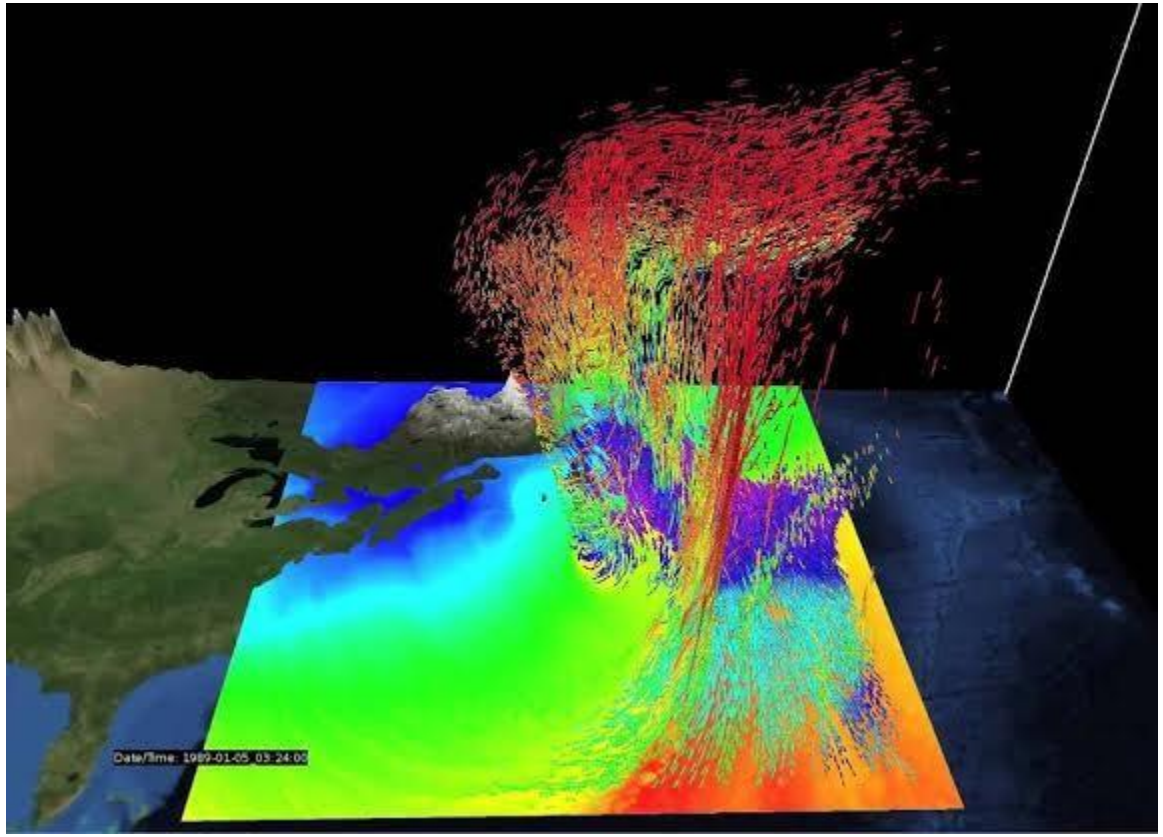
Source: https://media-cldnry.s-nbcnews.com/image/upload/newscms/2018_11/2362571/180314-virtual-reality-headset-ew-1243p.jpg
https://images.theconversation.com/files/245627/original/file-20181114-194494-1p82jkx.jpg?ixlib=rb-1.1.0&q=45&auto=format&w=1200&h=1200.0&fit=crop

Major Applications (7/12)

Google Maps

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

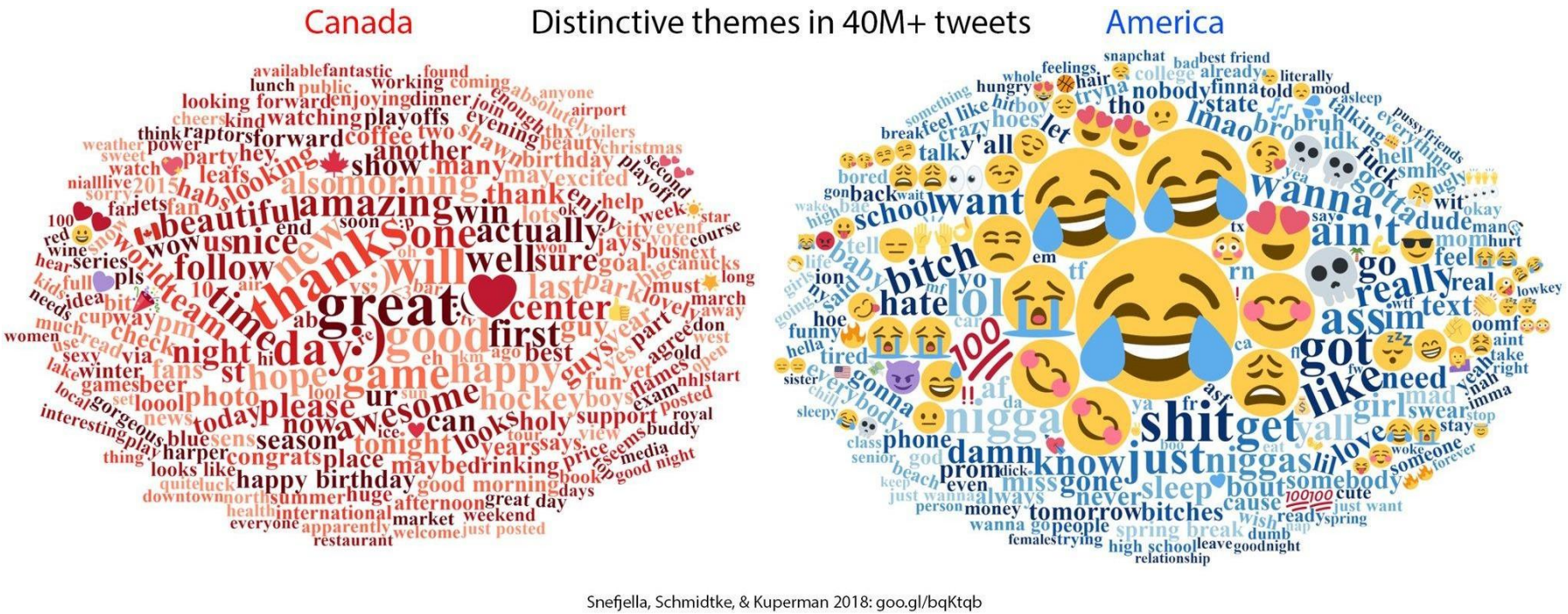
Major Applications (8/12)



Scientific Visualization (SciVis)

Source: https://youtu.be/eJy5dHMY-S4

Major Applications (9/12)



Word Cloud

Major Applications (9/12)

Major Applications (10/12)

CAD (3D modeling)

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Major Applications (11/12)

Simulation

Source: https://www.aircharterservice.com/about-us/news-features/blog/are-vr-flight-simulators-the-future-of-pilot-training

Major Applications (12/12)

Simulation

Source: https://www.financialexpress.com/sports/what-is-drs-all-the-the-rules-number-of-chances-and-components-explained/578996/

Graphics API (1/2)

- A ***graphics API*** is a set of functions that perform basic operations such as –
  - drawing images and 3D surfaces into windows on 2D screen.

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Graphics API (2/2)

Every ***graphics program*** needs to be able to use two related APIs

- **Graphics API** for visual output.
  - Ex:
    - i.e. command for drawing lines, circles etc.
- **User-interface API** to get input from the user.
  - Ex:
    - Windows API (WinAPI), UIKit, Android SDK etc.

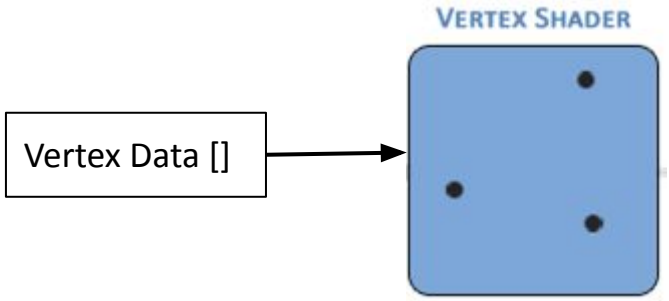
Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Graphics Pipeline (1/9)

- Special software/hardware subsystem that maps the **3D vertex** locations to **2D screen**.
- From modeling to rendering.
  - Shade the triangles –
    - Realistic
    - Proper back-to-front order.

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

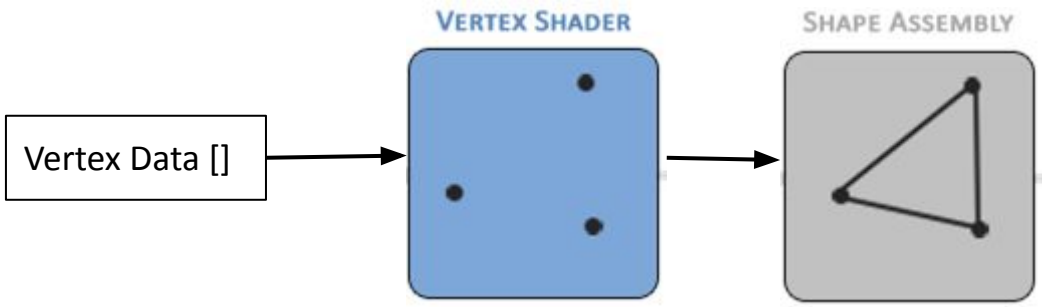
Graphics Pipeline (2/9)



Step 1: Vertex shader takes vertex data (position, texture coordinates, normals) as input and transform it

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

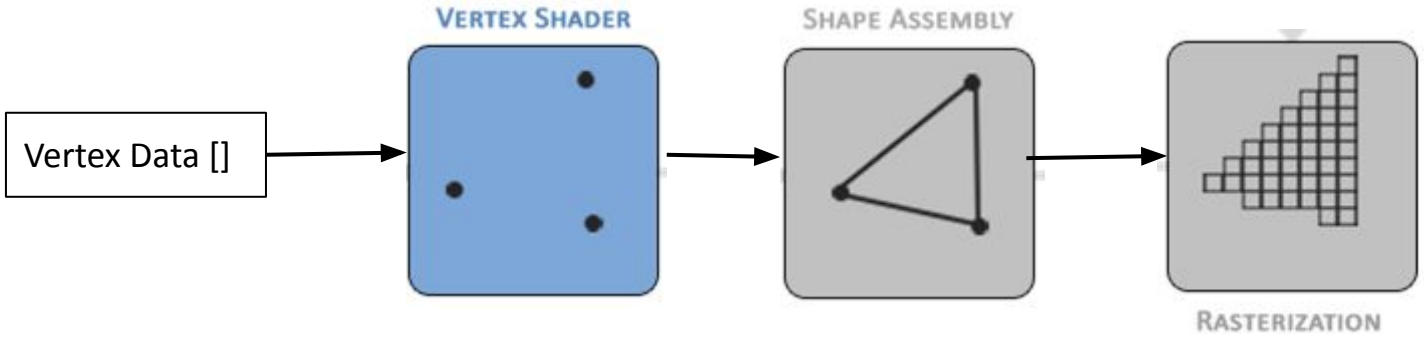
Graphics Pipeline (3/9)



Step 2: Shape assembly gets the transformed vertices and group them into geometric primitives

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

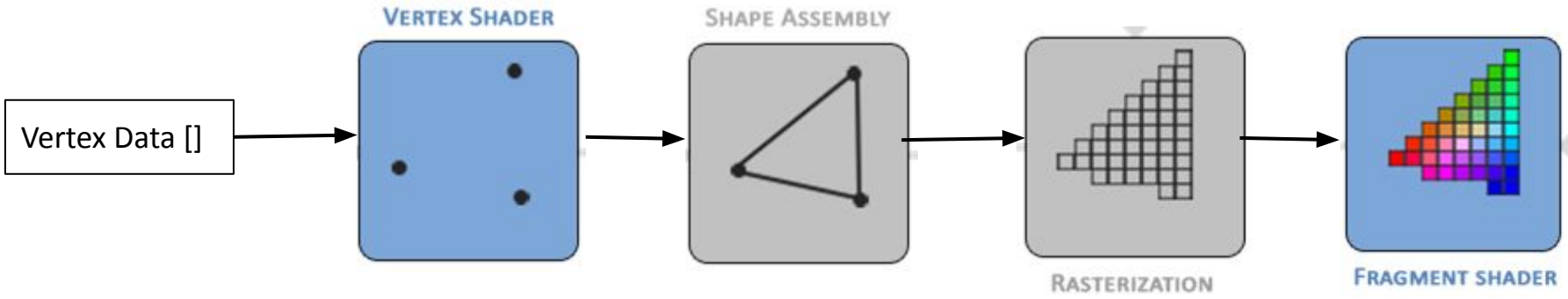
Graphics Pipeline (4/9)



Step 3: Rasterization transforms the primitives into fragments

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

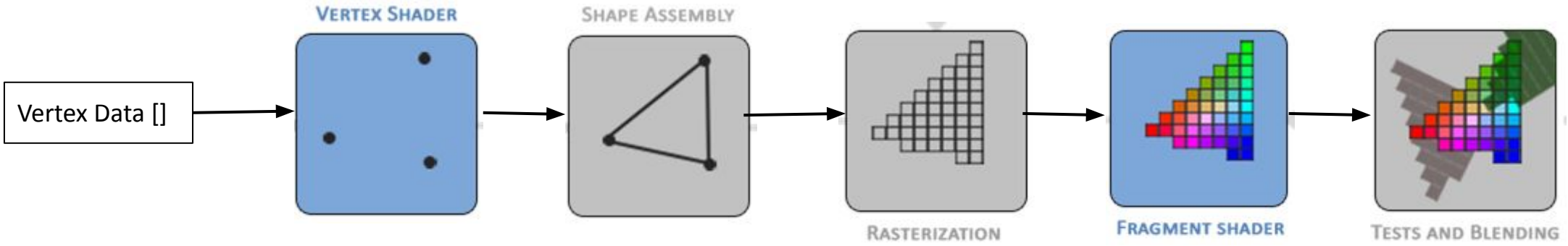
Graphics Pipeline (5/9)



Step 4: Fragment shader gets rasterized fragments and apply additional operations such as coloring, interpolation and texture mapping

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Graphics Pipeline (6/9)



Step 5: Some additional operations such as depth testing (to determine visibility) and blending are performed at the final step.

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Graphics Pipeline (7/9)

- Why triangles?
  - It is the **simplest** universal surface element
  - it is the **convex hull of three points**.
    - A line or a point are even simpler, but do not create surfaces.
    - it isn't possible to use only a finite number of them without having cracks.

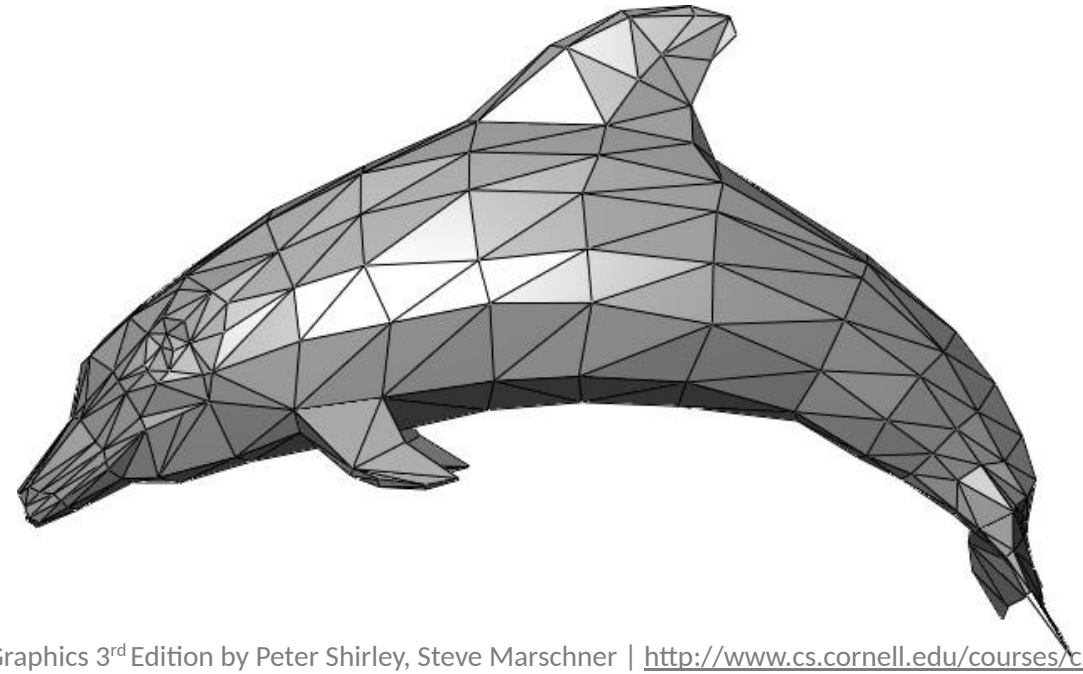
Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

Graphics Pipeline (8/9)

- Mesh:

  A polygon mesh is a collection of vertices, edges and faces that defines the shape of a polyhedral object.

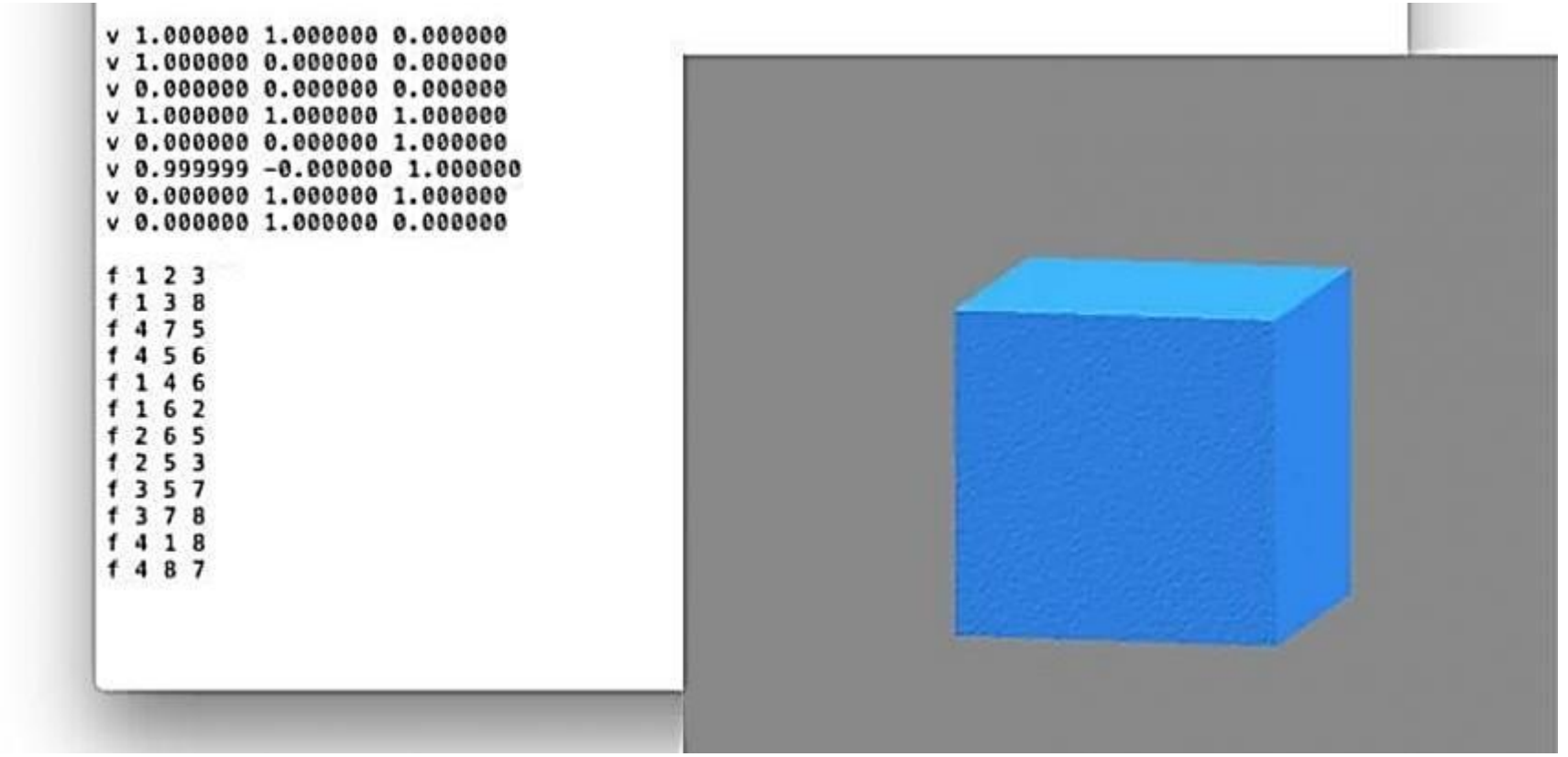
  – Ex. *Quad* mesh, Triangle mesh.



Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/ | https://en.wikipedia.org/wiki/Triangle_mesh

Graphics Pipeline (9/9)

.obj file



Source:
https://www.sculpteo.com/en/glossary/obj-file-3d-printing-file-format/

LoD (1/3)

- Rendering speed ↔ number of triangles being drawn
  - More triangles: more storage.

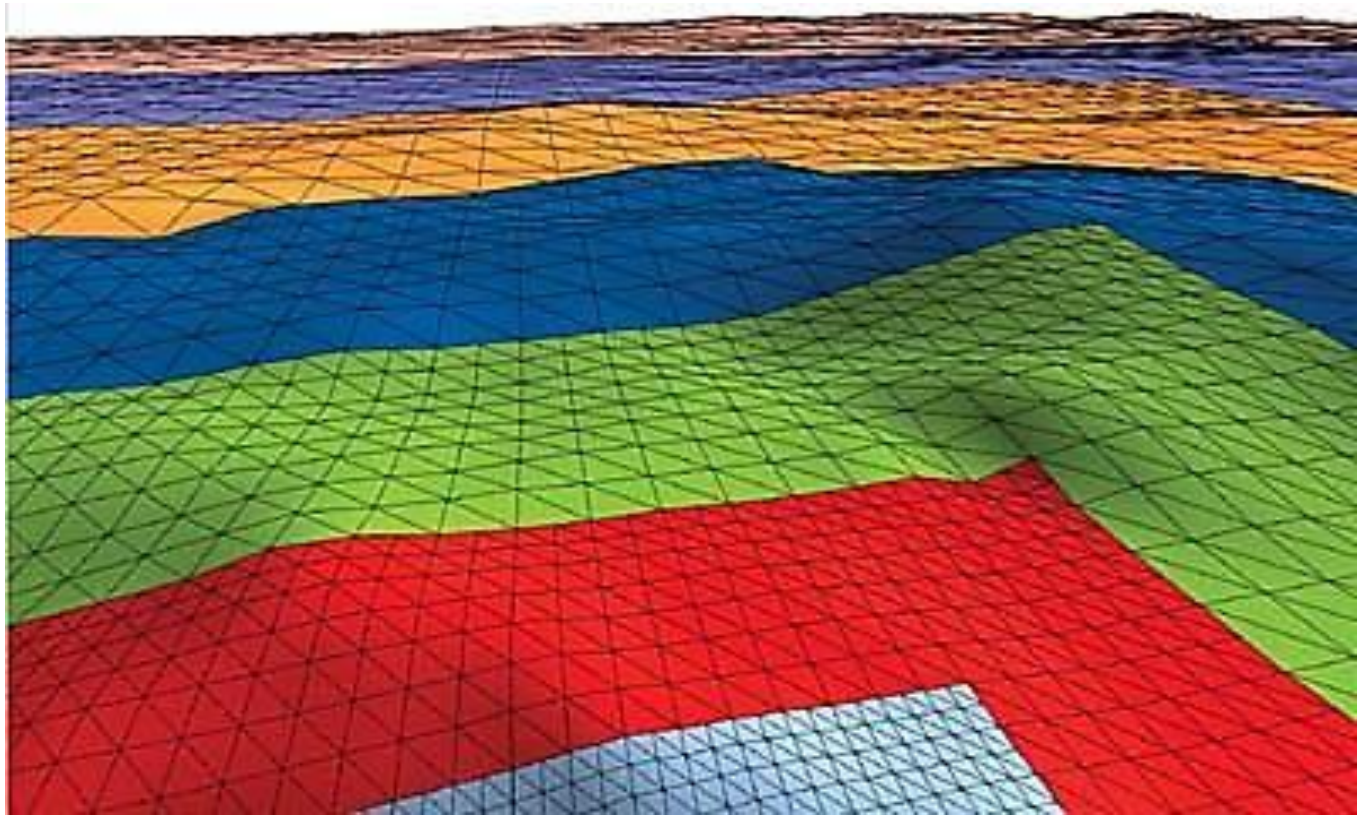
[0, 0, 3], [0, 3, 3], [0, 3, 0], [0, 0, 0],
[3, 0, 3], [3, 3, 3], [3, 3, 0], [3, 0, 0]

- It is worthwhile to minimize the number of triangles used to represent a model.
  - *Level of detail or LoD* optimizes the rendering of complex models by varying level of detail
  - If the model is viewed in the distance, fewer triangles needed and vice versa

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/ | https://en.wikipedia.org/wiki/Triangle_mesh

LoD (2/3)

- Example of *LoD*:



Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/
Source: https://developer.nvidia.com/gpugems/gpugems2/part-i-geometric-complexity/chapter-2-terrain-rendering-using-gpu-based-geometry

LoD (3/3)

Source:The Last of Us Part 1

Further Reading

- Fundamentals of Computer Graphics, 4th Edition - Chapter 1
- Real-Time Rendering, Fourth Edition - Section 19.9

Thank You