Chapter 1: Introduction

1.1 Graphics Areas

1.2 Major Applications

1.3 Graphics APIs

1.4 Graphics Pipeline

- special software/hardware subsystem that efficiently draws 3D primitives in perspective
- basic operations map the 3D vertex locations to 2D screen postions and shade the triangles
- 4D coordinates system

1.5 Numerical Issues

- IEEE *floating-point* standard
  - Three special values for real numbers
    1. Infinity(∞)
       a valid number that is larger than all other valid numbers.
    2. Minus infinity($-\infty$)
       a valid number that is smaller than all other valid numbers.
    3. Not a number(NAN)
       an invalid number
    - $\infty + \infty = +\infty$
    - $\infty - \infty = NaN$
    - $\infty \div \infty = NaN$
    - $0/0 = NaN$
    - Any aritmetic expression that includes NaN results in NaN.
    - Any Boolean expression involving NaN is false.

1.6 Efficiency

efficiency is achieved through careful tradeoffs

1.7 Designing and Coding Graphics Programs

1. **Class Design**
   some basic classes to be written include:
   - **vector2**
     A 2D vetor with indexing, vector addition, vector subtraction, dot product, cross product, scalar multiplication, scalar division.
   - **vector3**
     A 3D vector class analogous to vector2
   - **hvector**
     A homogeneous vector with four components
   - **rgb**
     An RGB color with RGB addtion, RGB substraction,RGB multiplication, scalar multiplication, scalar division
   - **transform**
     A 4*4 matrix for transformations. should include a matrix multiply
   - **image**
     A 2D array of RGB pixels with an output operation.
2. **Float vs. Double**
   - Modern architecture suggests that keeping memory use down and maintaining coherent memory access are the keys to efficiency.***this suggests using single precision data***
   - however, avoiding numerical problems suggests using double-precision arithmetic. The tradeoffs depend on the program.