Transformations II
Week 2, Fri Jan 15

13
2D Translation

2D Translation

15
Linear Transformations

Challenge

Readings for Transformations I-IV

Review: Event-Driven Programming

• transformation matrices

- • linear transformations are combinations of

- • shear

- • scale

- • rotate

- • reflect

- • properties of linear transformations

- • matrix multiplication

- • for everything except translation

- • how to do everything with multiplication?

- • then just do composition, no special cases

- • homogeneous coordinates trick

- • represent 2D coordinates (x,y) with 3-vector (x,y,1)

- • matrix multiplication

- • vs. procedural

- • control flow through event callbacks

- • key was pressed

- • mouse moved

- • callback functions called from main loop when events occur

- • mouse/keyboard state setting vs. redrawing

News

• prereq letters

2D Rotation

Shear

Reflection

Shear

Review: 2D Rotations

Review: 2D Transformations

2D Translation

2D Translation

2D Translation

Challenge

News

Readings for Transformations I-IV

Review: Event-Driven Programming

• transformation matrices

- • linear transformations are combinations of

- • shear

- • scale

- • rotate

- • reflect

- • properties of linear transformations

- • matrix multiplication

- • for everything except translation

- • how to do everything with multiplication?

- • then just do composition, no special cases

- • homogeneous coordinates trick

- • represent 2D coordinates (x,y) with 3-vector (x,y,1)

- • matrix multiplication

- • vs. procedural

- • control flow through event callbacks

- • key was pressed

- • mouse moved

- • callback functions called from main loop when events occur

- • mouse/keyboard state setting vs. redrawing

13
2D Translation

2D Translation

15
Linear Transformations

Challenge

Readings for Transformations I-IV

Review: Event-Driven Programming

• transformation matrices

- • linear transformations are combinations of

- • shear

- • scale

- • rotate

- • reflect

- • properties of linear transformations

- • matrix multiplication

- • for everything except translation

- • how to do everything with multiplication?

- • then just do composition, no special cases

- • homogeneous coordinates trick

- • represent 2D coordinates (x,y) with 3-vector (x,y,1)

- • matrix multiplication

- • vs. procedural

- • control flow through event callbacks

- • key was pressed

- • mouse moved

- • callback functions called from main loop when events occur

- • mouse/keyboard state setting vs. redrawing

News

• prereq letters

2D Rotation

Shear

Reflection

Shear

Review: 2D Rotations

Review: 2D Transformations

2D Translation

2D Translation

2D Translation

Challenge

News

Readings for Transformations I-IV

Review: Event-Driven Programming

• transformation matrices

- • linear transformations are combinations of

- • shear

- • scale

- • rotate

- • reflect

- • properties of linear transformations

- • matrix multiplication

- • for everything except translation

- • how to do everything with multiplication?

- • then just do composition, no special cases

- • homogeneous coordinates trick

- • represent 2D coordinates (x,y) with 3-vector (x,y,1)

- • matrix multiplication

- • vs. procedural

- • control flow through event callbacks

- • key was pressed

- • mouse moved

- • callback functions called from main loop when events occur

- • mouse/keyboard state setting vs. redrawing

News

• prereq letters

2D Rotation

Shear

Reflection

Shear

Review: 2D Rotations

Review: 2D Transformations

2D Translation

2D Translation

2D Translation

Challenge
**Homogeneous Coordinates**
- our 2D transformation matrices are now 3x3:
  - point in 2D cartesian
  - use rightmost column!

**Homogeneous Coordinates Geometrically**
- point in 2D cartesian + weight w = point P in 3D homog. coords
- multiply by w to get (x/w, y/w, 1)
- projects line to point onto w=1 plane
- like normalizing, one dimension up when w=0, consider it as direction
- points at infinity
- these points cannot be homogenized
- lies on w=0 plane

**Summary: Transformations**
- may seem unintuitive, but they make graphics operations much easier
- allow all affine transformations to be expressed through matrix multiplication
- origin does not necessarily map to origin
- lines map to lines
- parallel lines remain parallel
- ratios are preserved
- closed under composition

**3D Rotation About Z Axis**
- around z axis:
  - \( x' = x \cos \theta - y \sin \theta \)
  - \( y' = x \sin \theta + y \cos \theta \)
  - \( z' = z \)

**Homogeneous Coordinates Summary**
- may seem unintuitive, but they make graphics operations much easier
- allow all affine transformations to be expressed through matrix multiplication
- origin does not necessarily map to origin
- lines map to lines
- parallel lines remain parallel
- ratios are preserved
- closed under composition

**3D Translation**
- \( T(x, y, z) = T(x - x_0, y - y_0, z - z_0) \)

**3D Scaling**
- \( S(x, y, z) = S(x/k, y/k, z/k) \) (R is orthogonal)

**3D Shear**
- \( S(x, y, z) = S(x, y + ax, z + bx) \)

**Undoing Transformations: Inverses**
- \( T^{-1}(x, y) = T^{T}(x, y) \)
- \( R^{-1}(x, y) = R^{T}(x, y) \) (R is orthogonal)

**Composing Transformations**
- translation
  - \( T_1 = T_1(x_1, y_1, z_1) \)
  - \( T_2 = T_2(x_2, y_2, z_2) \)
  - \( T_1 \circ T_2 = T_2 \circ T_1 \)

- scaling
  - \( S \times S \)

- rotation
  - \( R \times R \)
Composing Transformations

Ta Tb = Tb Ta, but Ra Tb = Tb Ra, but Ra and Ta
• rotations around same axis commute
• rotations around different axes do not commute
• rotations and translations do not commute

Interpreting Transformations

Matrix Composition
• matrices are convenient, efficient way to represent series of
  • general purpose representation
  • hardware matrix multiply
  • matrix multiplication is associative
    • p' = (T'R'S')p
    • p' = (T'R)'S'p
• procedure
  • correctly order your matrices!
  • multiply matrices together
  • result is one matrix, multiply vertices by this matrix
  • all vertices easily transformed with one matrix multiply

Rotation About a Point: Moving Object

p' = TRp
• which direction to read?
  • right to left
    • interpret operations wrt fixed coordinates
    • moving object
  • left to right
    • interpret operations wrt local coordinates
    • changing coordinate system

OpenGL pipeline ordering!
  • same relative position between object and basis vectors

Rotation: Changing Coordinate Systems

• same example: rotation around arbitrary center
  • rotate about p by θ
  • translate p to origin
  • rotate about origin
  • translate p back

General Transform Composition
• transformation of geometry into coordinate system where operation becomes simpler
  • typically translate to origin
  • perform operation
  • transform geometry back to original coordinate system
Rotation About an Arbitrary Axis

- axis defined by two points
- translate point to the origin
- rotate to align axis with z-axis (or x or y)
- perform rotation
- undo aligning rotations
- undo translation

Arbitrary Rotation

- arbitrary rotation: change of basis
- given two orthonormal coordinate systems \(XYZ\) and \(ABC\)
- if A's location in the \(XYZ\) coordinate system is \((a_x, a_y, a_z, 1)\), ...
- transformation from one to the other is matrix \(R\) whose columns are \(A, B, C\):

\[
R(X) = \begin{bmatrix}
    a_x & b_x & c_x & 0 \\
    a_y & b_y & c_y & 0 \\
    a_z & b_z & c_z & 0 \\
    0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
    a_x \\
    a_y \\
    a_z \\
    1
\end{bmatrix} = A
\]

\[
\begin{bmatrix}
    b_x \\
    b_y \\
    b_z \\
    1
\end{bmatrix} = B
\]

\[
\begin{bmatrix}
    c_x \\
    c_y \\
    c_z \\
    1
\end{bmatrix} = C
\]