Lecture 16

Perspective in 2D Games

Take Away for Today

- What is the game "camera"?

  - How does it relate to screen space? Object space?
  - How does the camera work in a 2D game? 3D?

- How do we give 2D games depth?

  - Advantages, disadvantages of *orthographic view*
  - Advantages, disadvantages of *axonometric view*

- How does "tileability" affect art in games?

the gamedesigninitiative
at cornell university

The Game Camera

- What makes a game 3-D?
  - Everything is shown on a 2-D screen (mostly)

- 3D game have a **user controlled** "camera"
  - Position camera to look at art from all sides
  - 3-D art has enough information to allow this

- CS/INFO 3152 limits you to a 2-D game
  - The game camera has a *fixed perspective*
  - You render all art to one visible side

Camera in 2D Games

World origin

World

Camera origin

Camera

Perspective
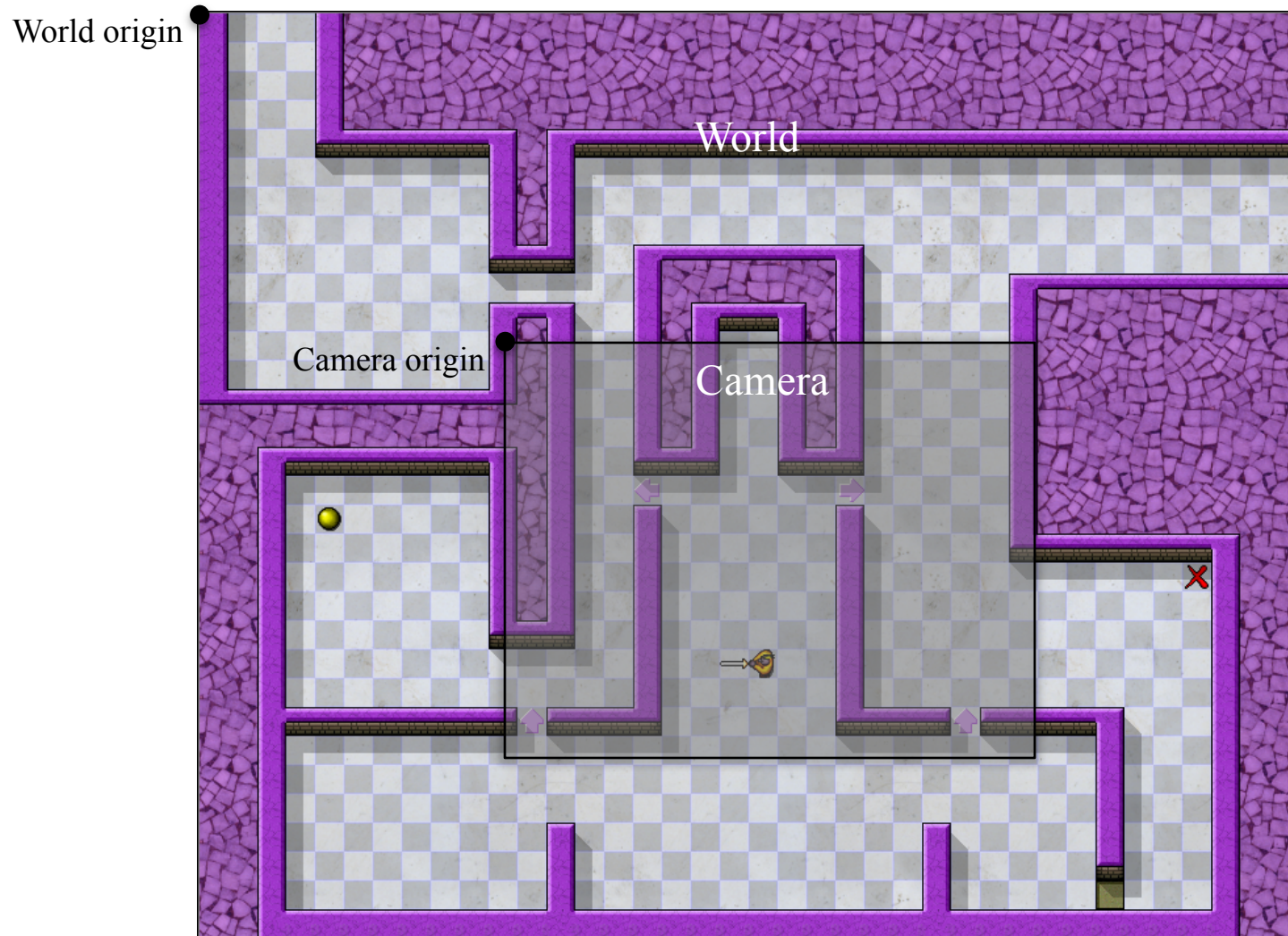
Specifying the Camera

- Camera is a **coord space**
  - Called "eye space"
  - Eye position at origin

- How to move camera?
  - Transforms again!

- **Inverse** of scrolling
  - Scrolling: move obj to eye
  - Camera: move eye to obj
  - Two matrices are *inverses*
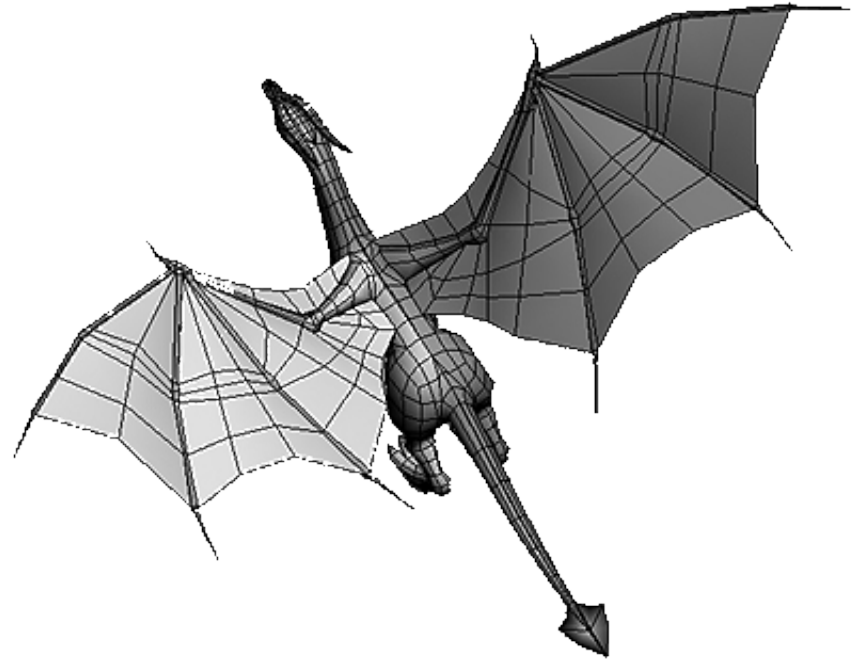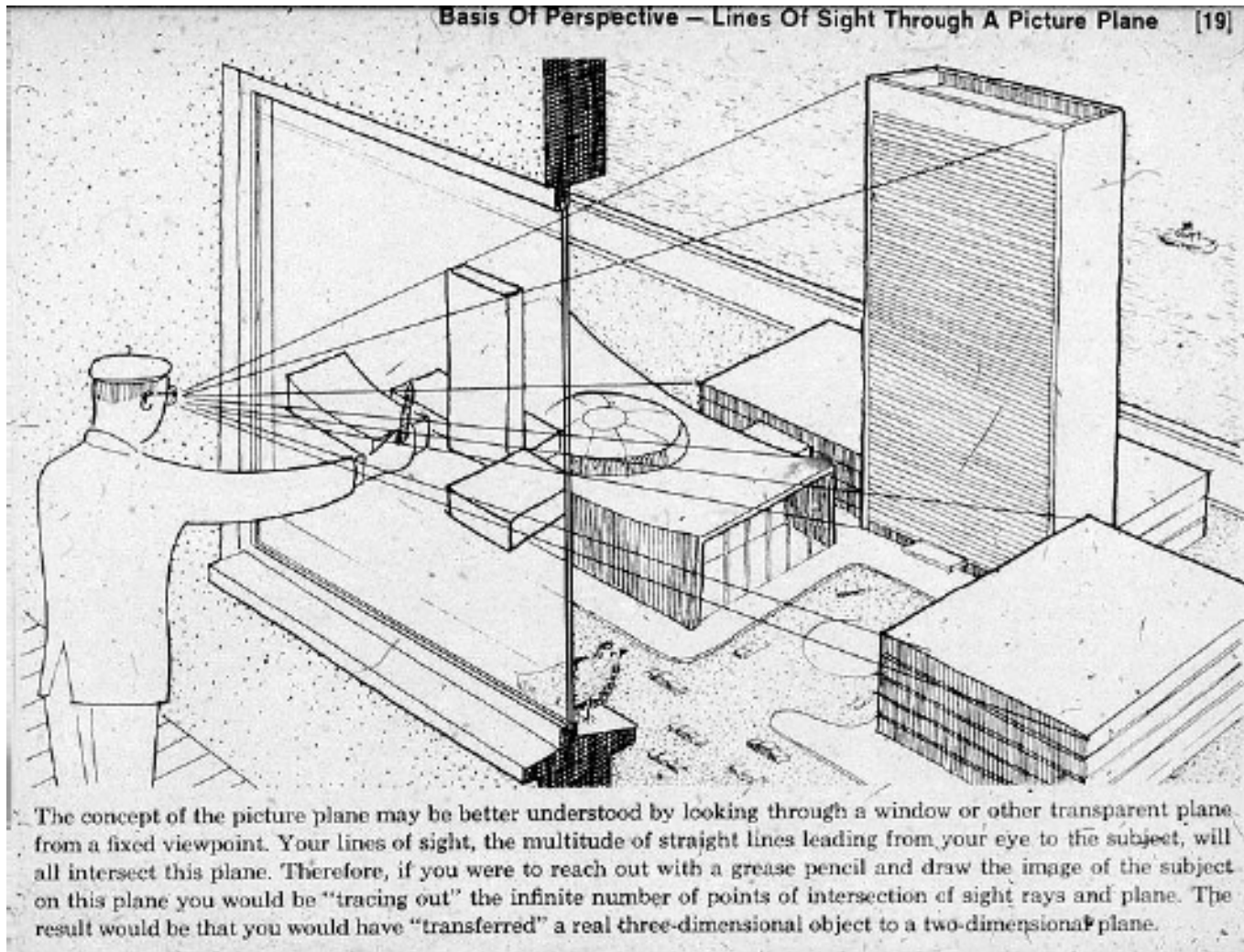
Perspective
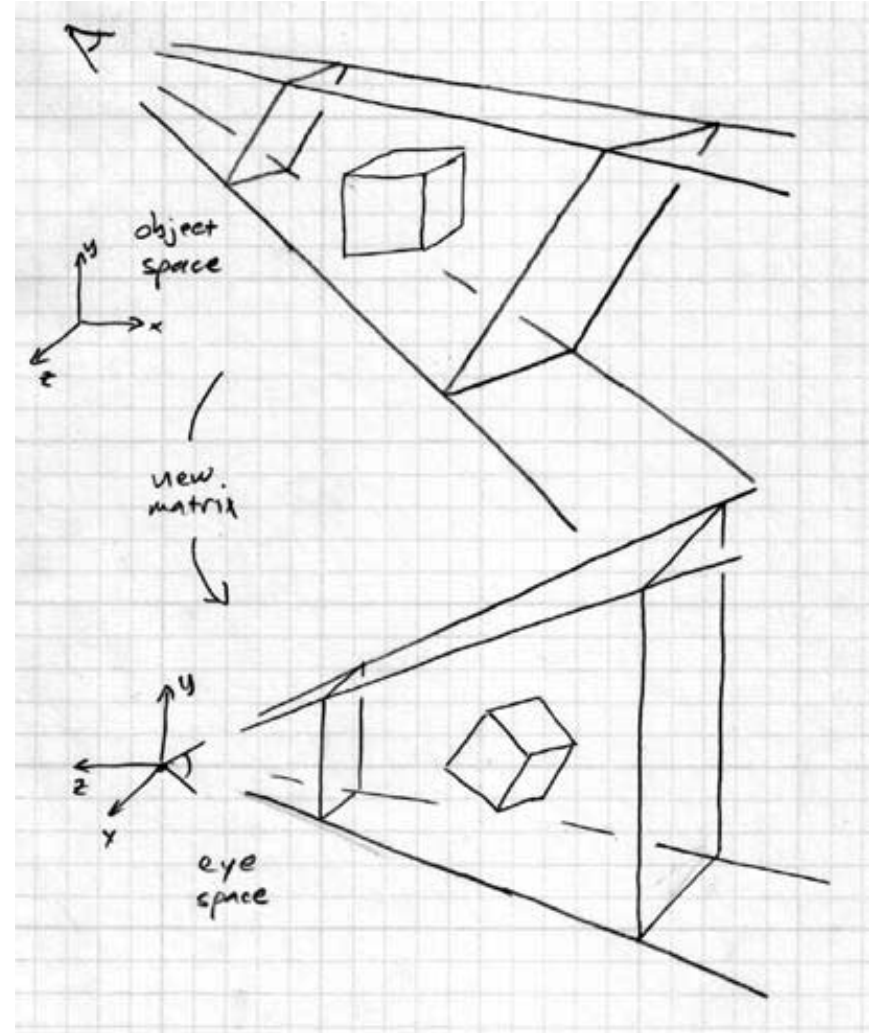
Camera in XNA

- Purpose of `BasicEffects` class
  - `View` property holds **eye space transform**
  - … also, `Texture` property holds current texture

- Set camera with an **effect pass**
  - `effect.CurrentTechnique.Passes`
  - Call `pass.Apply()` before you draw
  - Works with both sprites and triangles
  - See `GameCanvas.cs` in *Programming Lab 2*

Perspective

Drawing for a Perspective

- 3D Models make it easy
  - Rotate model to position
  - Flatten to jpeg, tiff, etc…

- But 3D modeling is hard
  - Very technical programs
  - Cannot draw "by hand"

- How to draw perspective?
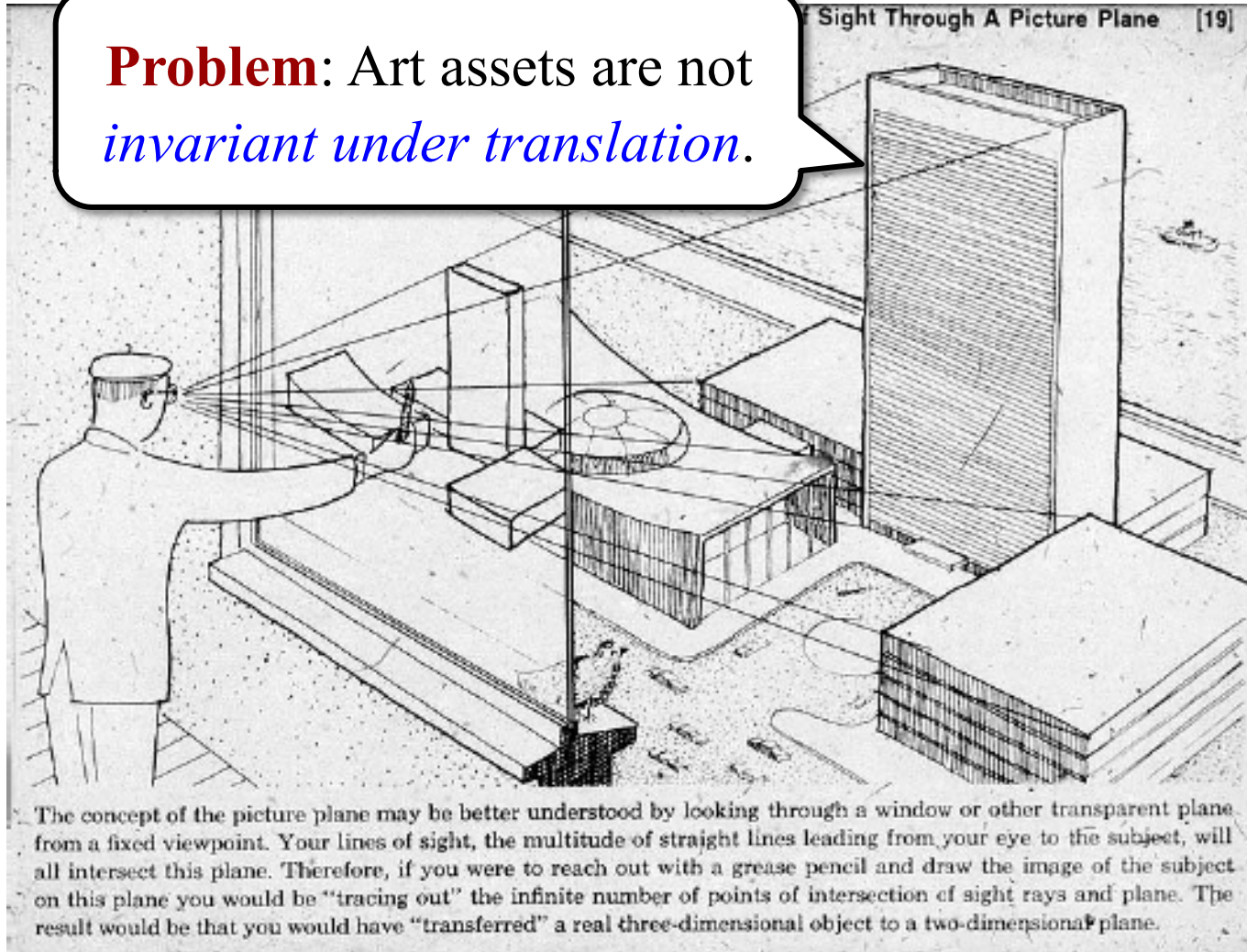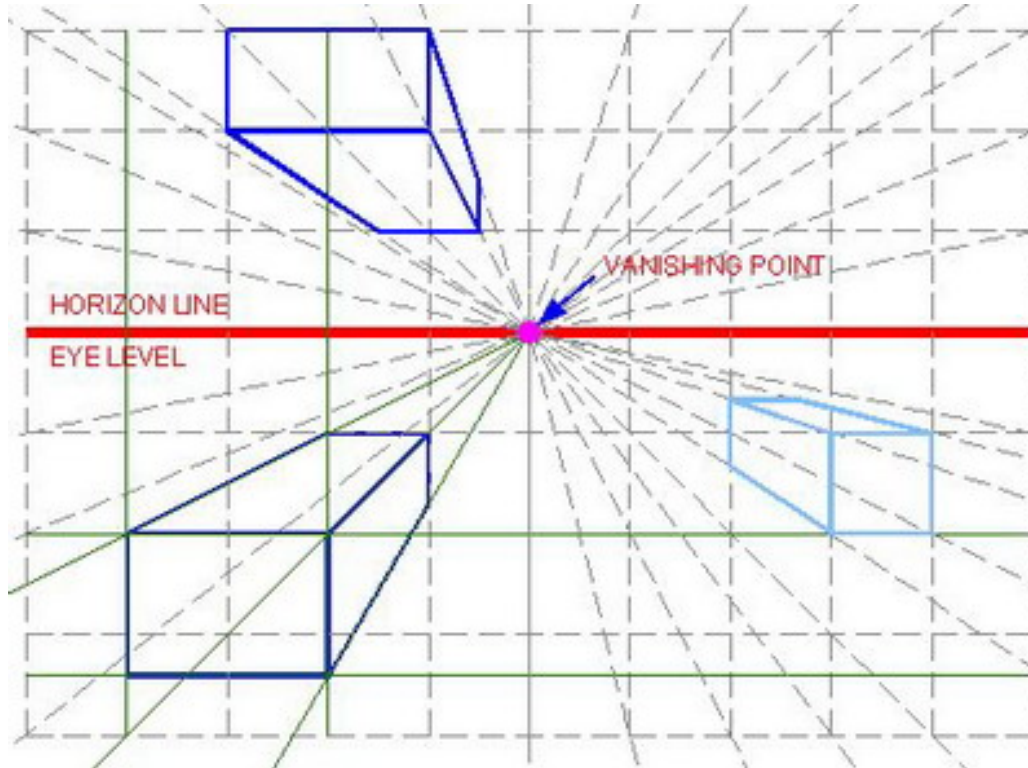 - Artist "captures" camera

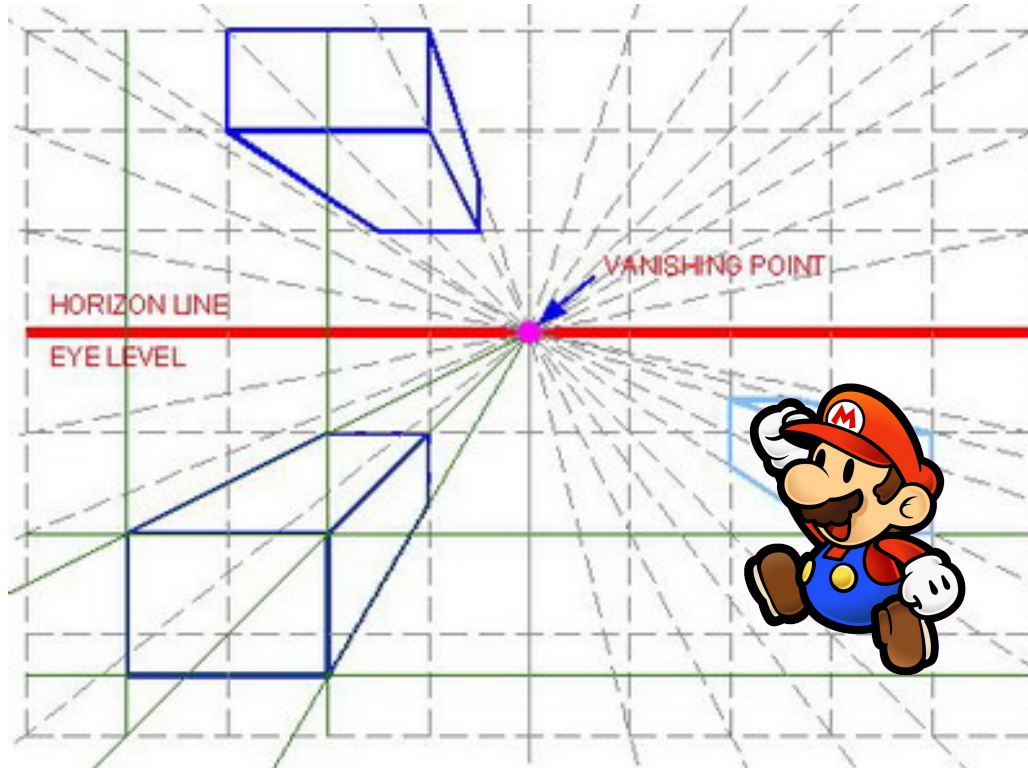- **Realism creates problems**

Perspective

Plane Projection in Drawing

Basis Of Perspective — Lines Of Sight Through A Picture Plane [19]

The concept of the picture plane may be better understood by looking through a window or other transparent plane from a fixed viewpoint. Your lines of sight, the multitude of straight lines leading from your eye to the subject, will all intersect this plane. Therefore, if you were to reach out with a grease pencil and draw the image of the subject on this plane you would be "tracing out" the infinite number of points of intersection of sight rays and plane. The result would be that you would have "transferred" a real three-dimensional object to a two-dimensional plane.

Perspective

Plane Projection in Drawing

Problem: Art assets are not *invariant under translation*.

The concept of the picture plane may be better understood by looking through a window or other transparent plane from a fixed viewpoint. Your lines of sight, the multitude of straight lines leading from your eye to the subject, will all intersect this plane. Therefore, if you were to reach out with a grease pencil and draw the image of the subject on this plane you would be "tracing out" the infinite number of points of intersection of sight rays and plane. The result would be that you would have "transferred" a real three-dimensional object to a two-dimensional plane.

the gamedesigninitiative
at cornell university

Vanishing Points are **Not** Our Friend

Perspective

the gamedesigninitiative
at cornell university

Vanishing Points are **Not** Our Friend

Perspective

Vanishing Points are **Not** Our Friend

Should not be same

Perspective

the gamedesigninitiative
at cornell university

Vanishing Points are **Not** Our Friend

2D games rely on **distortional** perspectives

the gamedesigninitiative
at cornell university

Orthographic Projection

- *Project perpendicular* to an axis
  - **Top-down**: perpendicular to z-axis
  - **Side scrolling**: perpendicular to y-axis

- Very easy to do artistically
  - Art objects are flat tiles
  - Layer tiles via compositing

- But enforces *2-D gameplay*
  - 3rd dimension lost; cannot be seen
  - **Distorted**: All rays to eye are parallel

the gamedesigninitiative
at cornell university

Orthographic Projection
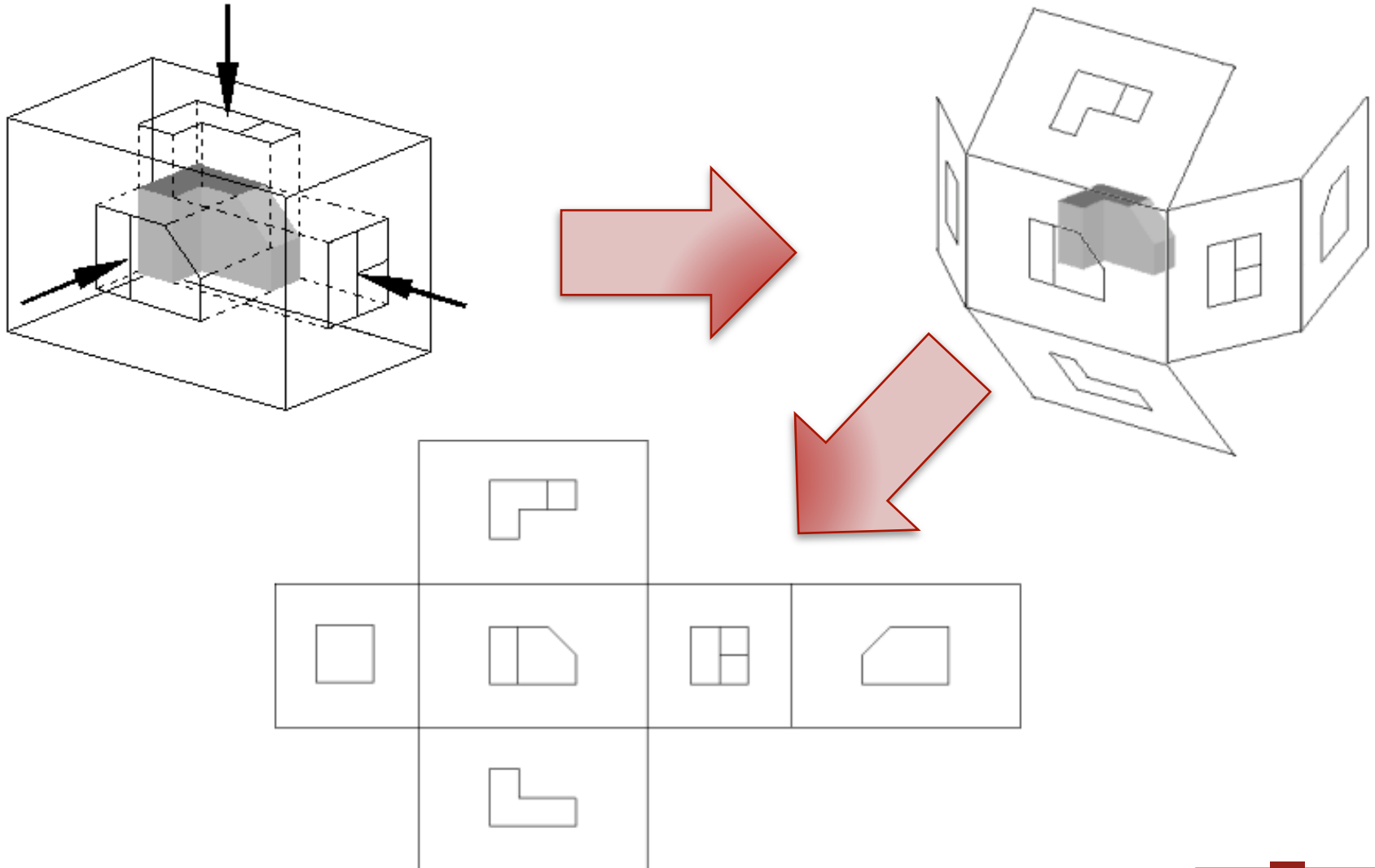
Perspective

Parallax Scrolling

- Gives depth to orthographic projection
  - Objects in background have distance
  - Rate of scrolling depends on distance

- Implement with multiple background layers
  - Each layer scrolls at a different rate
  - See course website for sample code

- Often requires some degree of **transparency**
  - *Composite* front layers with back layers

Perspective

the game**design**initiative
at cornell university

Parallax Scrolling

Perspective

the gamedesigninitiative
at cornell university
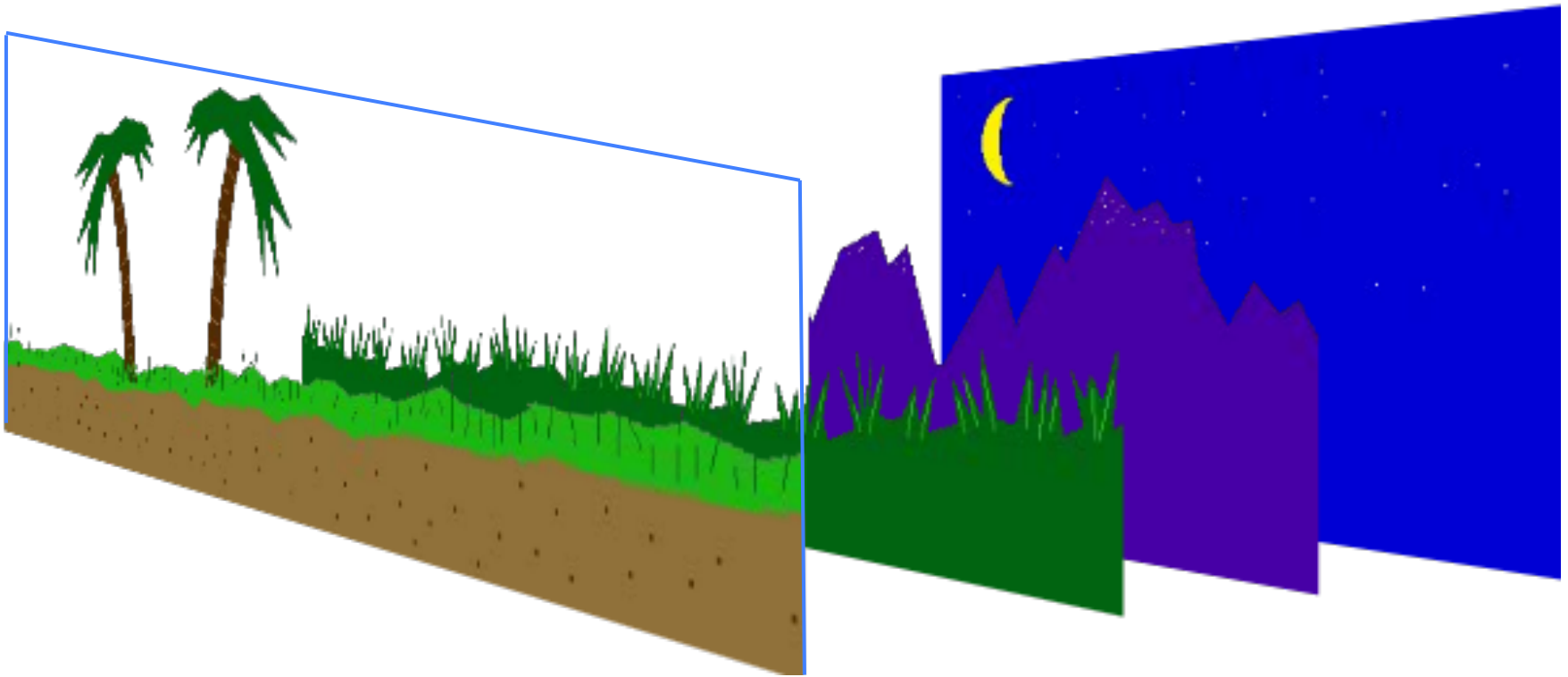
Axonometric Projection

- ## Off axis view of object
  - ### View along all 3-axes

- ## Once again: **distorted**
  - ### Not a true projection
  - ### No *vanishing point*
  - ### Axes are "foreshortened"

- ## Allows 3-D gameplay
  - ### "Cliffs" are visible
  - ### May also **hide objects**!

Trimetric

2 axes equal

Dimetric

all axes equal

Isometric

Perspective

Axonometric Projection

Perspective

the
game**design**initiative
at cornell university

Axonometric Projection

Perspective

the gamedesigninitiative
at cornell university
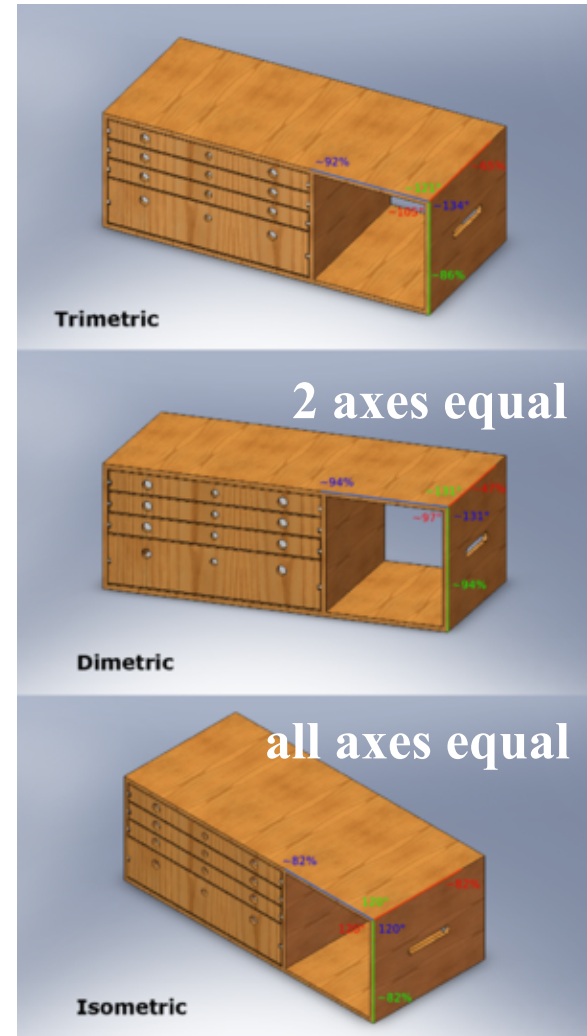
Projection Types

- **Isometric**
  - All axes are equal
  - If need all dimensions
  - Used in classic RPGs

- **Dimetric**
  - *z*-axis is very short
  - *x*, *y* axes are equal
  - Orthographic+depth
  - For aesthetic reasons only

2 axes equal

all axes equal

Perspective
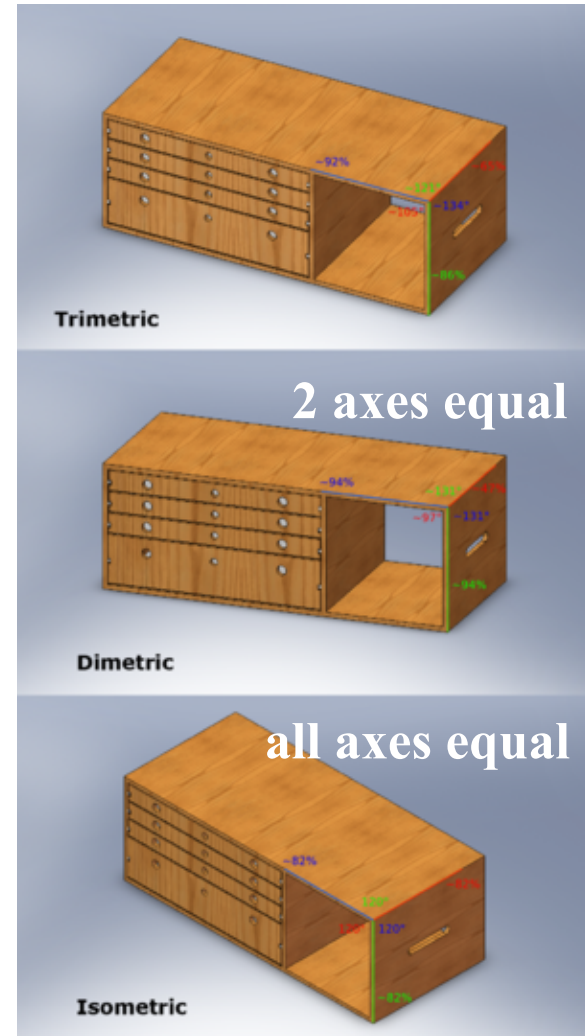
Projection Geometry

- Axes relative to screen
  - z goes "into" the screen
  - x, y are in screen plane

- Axonometric coodinates
  - May not be "true" coords
  - "Meaning" of x, y, z?

- Orthographic substitutes
  - **Side-scroller**: y is height
  - **Top-down**: z is height

Top Down

Side-Scroller

Perspective
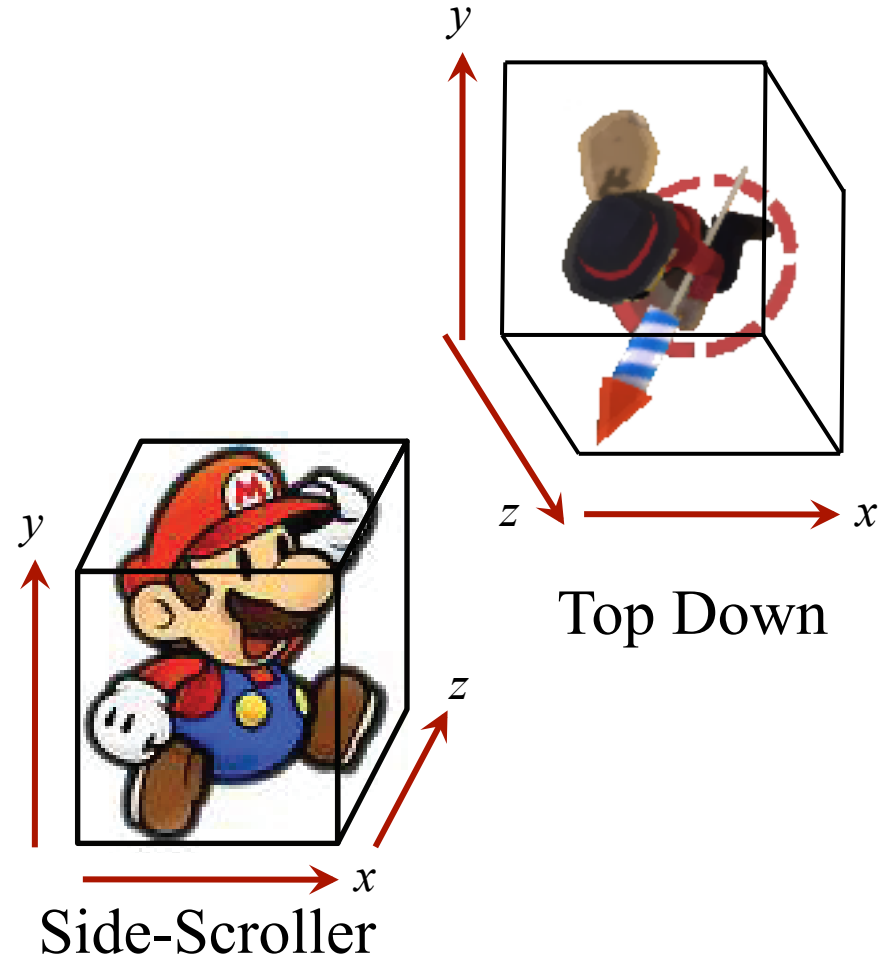
Isometric View

$$\frac{h}{w} = \frac{1}{\sqrt{3}}$$

- If need all axes visible
- Used in classic RPGs

Game View

Top View

Side View

Perspective

the gamedesigninitiative
at cornell university

Isometric View

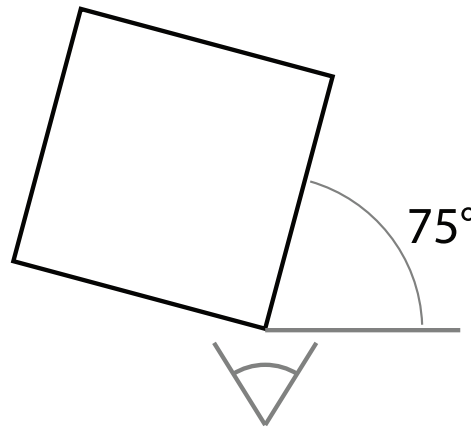- x, y, z = Axonometric Coords

- x', y' = Screen Coordinates

$$x' = x - z$$
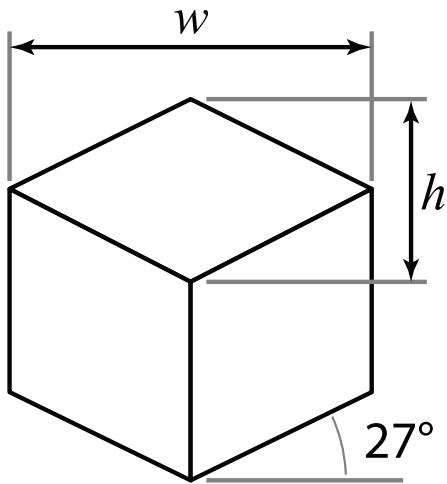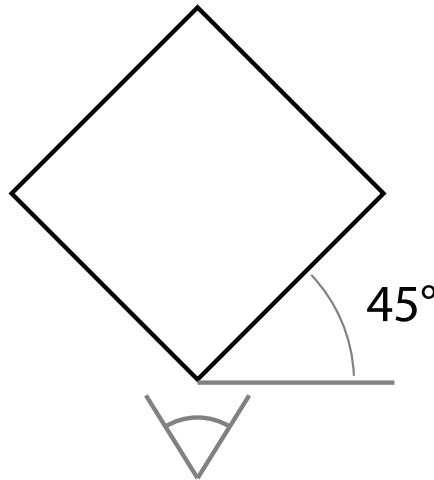
$$y' = y + \tfrac{1}{2}(x+z)$$
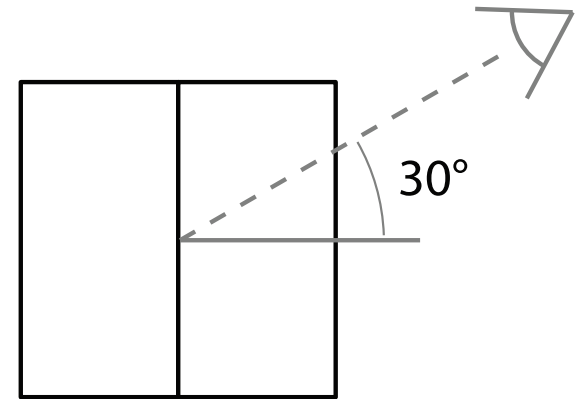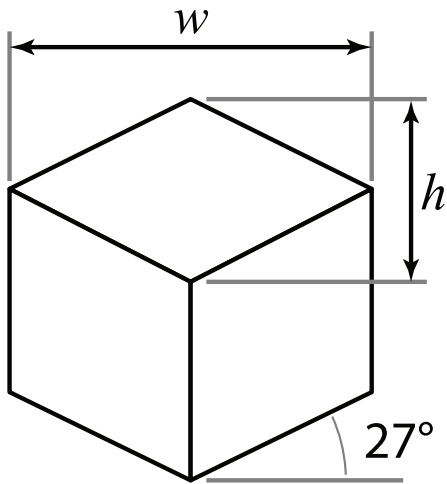
Game View
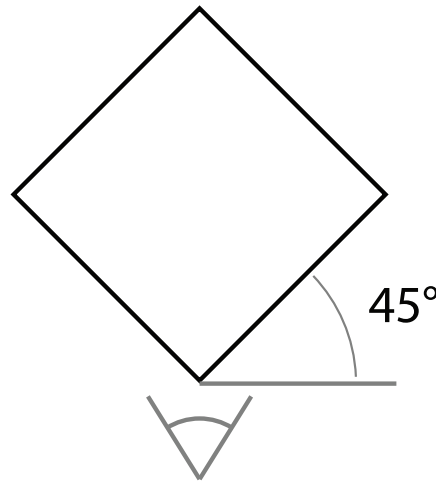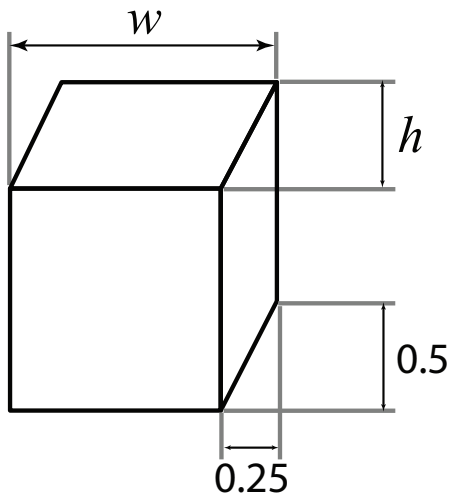

Top View
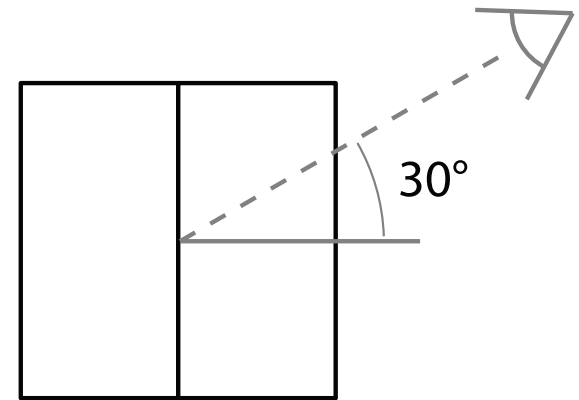
Side View

Isometric View: Zaxxon

Perspective

Dimetric View (Side-Depth)

$$\frac{h}{w} = \frac{1}{6}$$

- For horizontal movement
- Gives depth to side
- **Example**: platformer

w

h

0.25

0.5

Game View

75°

Top View

10°

Side View

Perspective
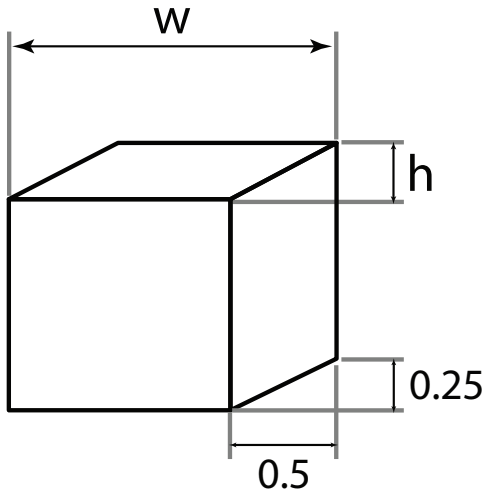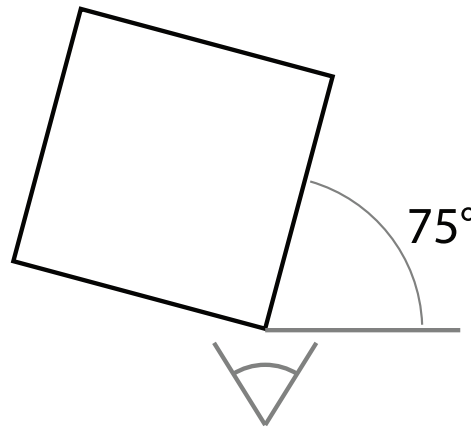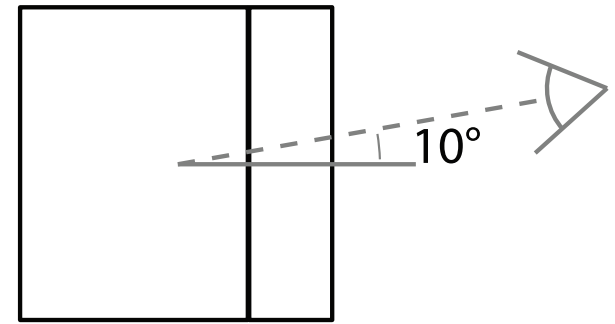
Dimetric View (Side-Depth)

- x, y, z = Axonometric Coords

- x', y' = Screen Coordinates

$$x' = x + \tfrac{1}{2}(z)$$

$$y' = y + \tfrac{1}{4}(z)$$
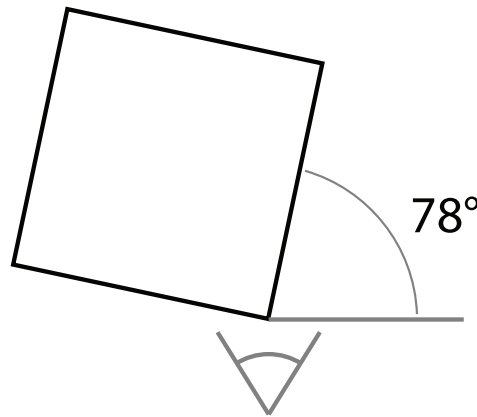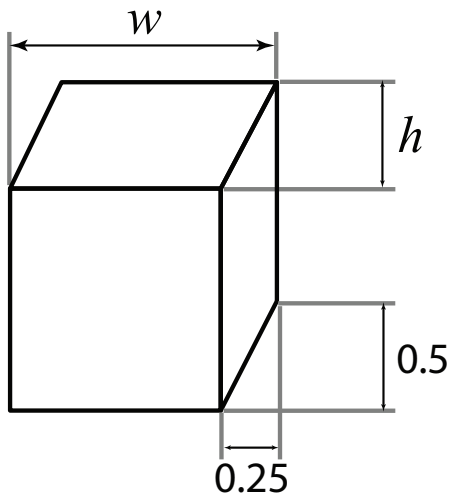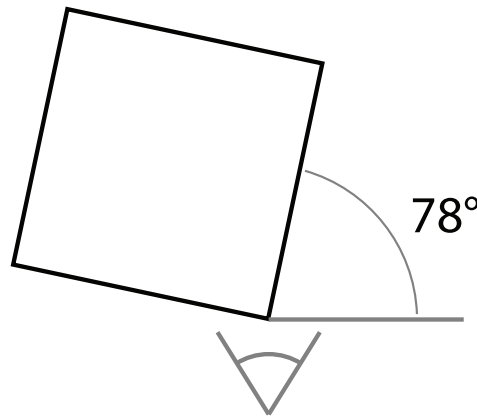

Game View

Top View

Side View

Perspective

the gamedesigninitiative
at cornell university
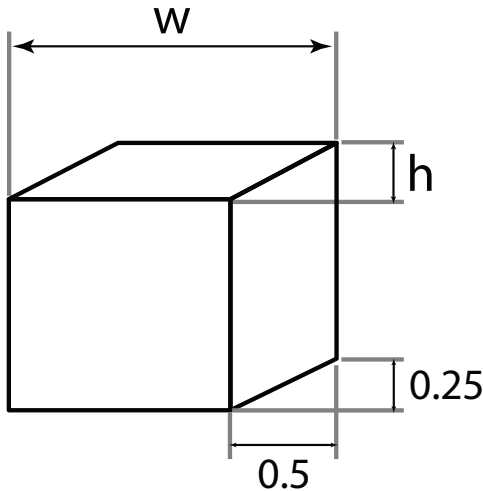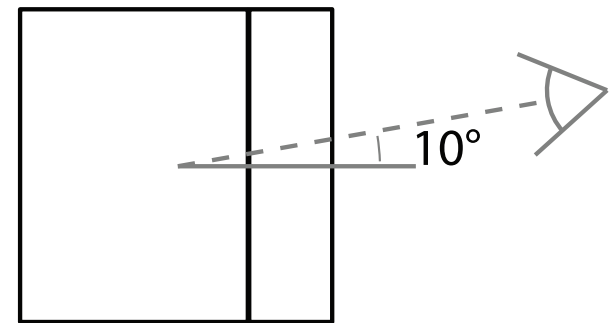
Dimetric View (Top-Depth)

$$\frac{h}{w} = \frac{2}{5}$$

- For full 2D movement
- z still goes into screen
- **Example**: stealth games

Game View

w

h

0.5

0.25

Top View

78°

Side View

~24°
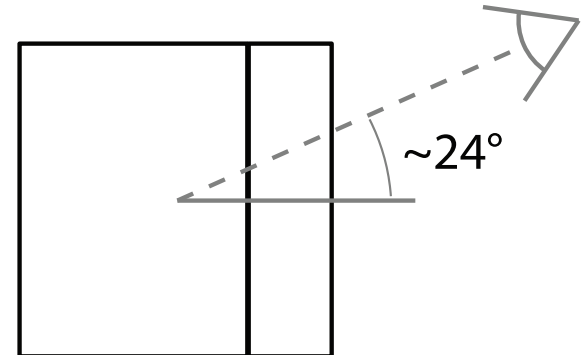
Dimetric View (Top-Depth)

- x, y, z = Axonometric Coords

- x', y' = Screen Coordinates
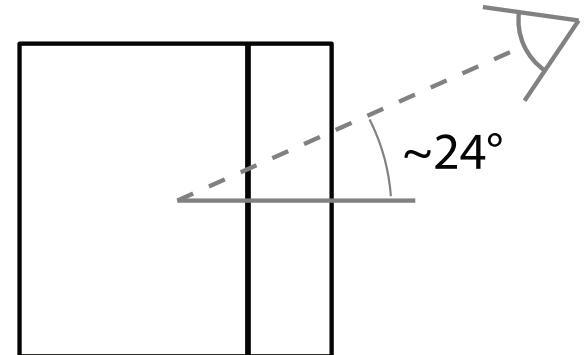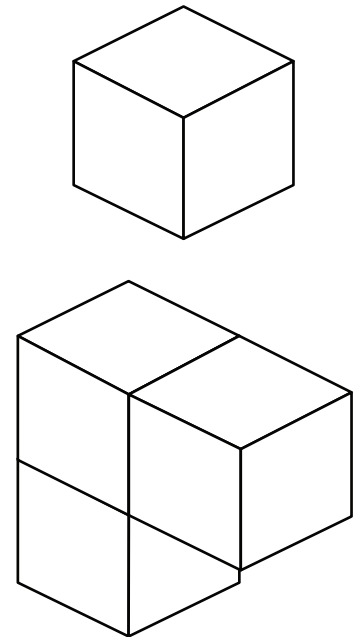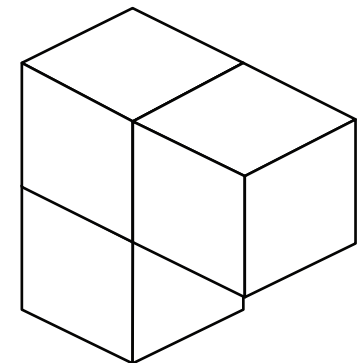
$$x' = x + \tfrac{1}{4}(z)$$

$$y' = y + \tfrac{1}{2}(z)$$

w

h

0.5

0.25

Game View

78°

Top View

~24°

Side View

Perspective

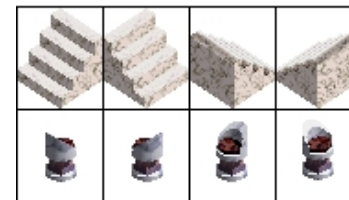# Dimetric View (Top Down)

Perspective

Drawing for Axonometric View

- Use boxes shown on slide
  - Tiling boxes is easy
  - Draw shape inside box

- Complex, large shapes?
  - Glue together boxes
  - Draw inside box group

- Objects need many angles
  - Transparency is tricky
  - Standard: 8 compass points

- **Example**: LakeHills.ai

Perspective

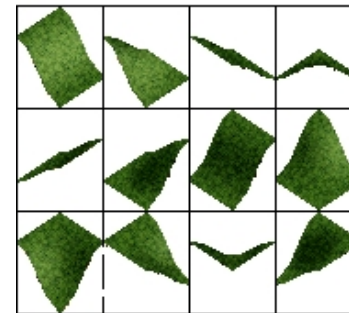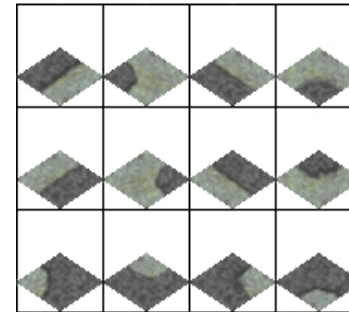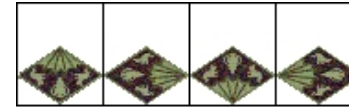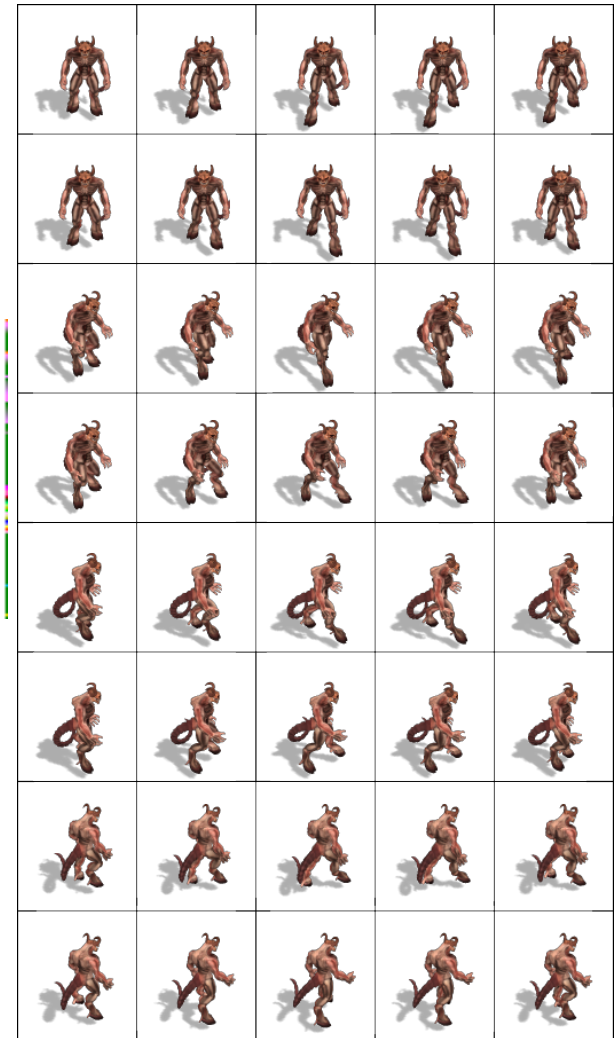the game**design**initiative
at cornell university

Drawing for Axonometric View

- Use boxes shown on slide
  - Tiling boxes is easy
  - Draw shape inside box

- Complex, large shapes?
  - Glue together boxes
  - Draw inside box group

- Objects need many angles
  - Transparency is tricky
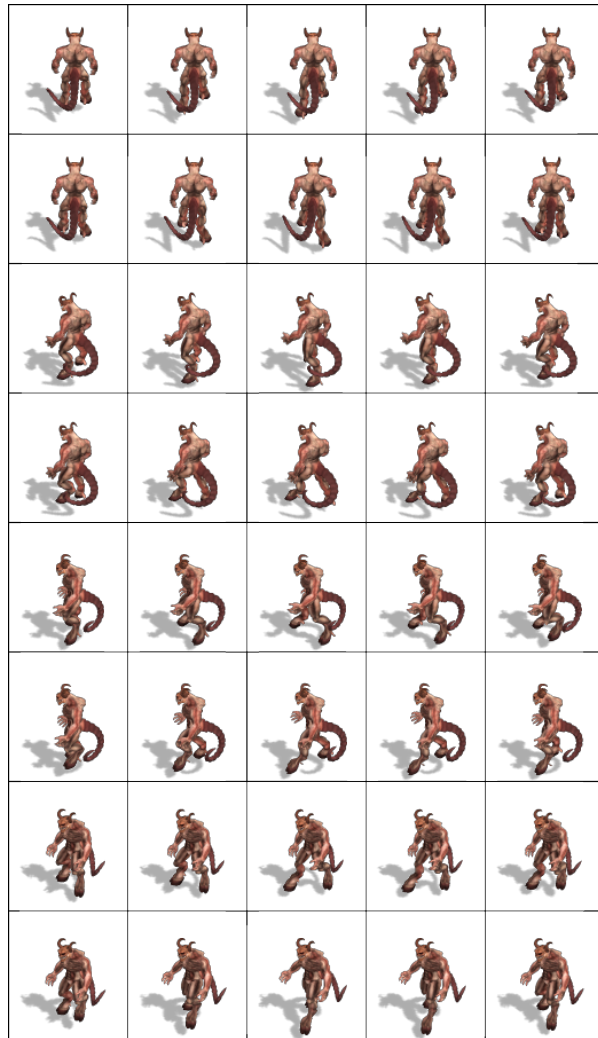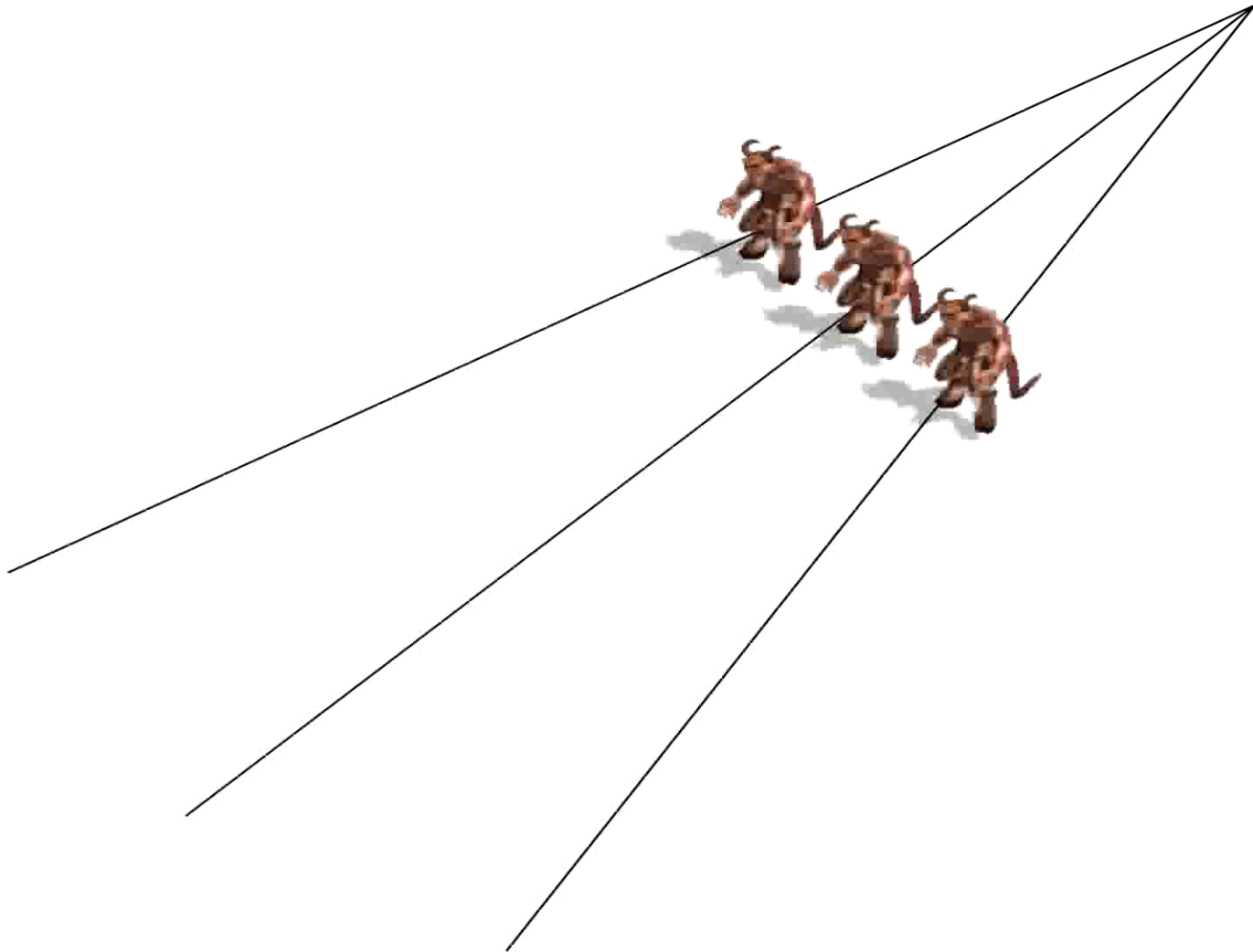 - Standard: 8 compass points

- **Example**: LakeHills.ai

Perspective

the game**design**initiative
at cornell university

Isometric Walking Animation

Perspective

Isometric Walking Animation

Perspective

Which Style to Use?

Orthographic

- **Advantages**
  - Easy to make tiles
  - Easy to composite

- **Disadvantages**
  - Movement is 2D
  - Game feels flat

- Common in this class

Axonometric

- **Advantages**
  - Sort of easy to tile
  - Some 3-D movement

- **Disadvantages**
  - Harder to composite
  - Objects may be hidden

- Lot of work for artist

the **gamedesign**initiative
at cornell university

Combining the Perspectives

Perspective

Combining the Perspectives

Perspective

the gamedesigninitiative
at cornell university

Summary

- Camera represents "eye space" coordinates
  - 3D games have arbitrary camera movement
  - 2D games are limited to scrolling movement

- 2-D art requires you chose a projection
  - **Orthographic** is easy, but limits gameplay
  - **Axonometric** has better gameplay, but harder to draw

- Axonmetric type depends on style of game
  - Isometric common to classic RPGs
  - Dimetric gives depth to traditional orthographic

Perspective

the gamedesigninitiative
at cornell university