

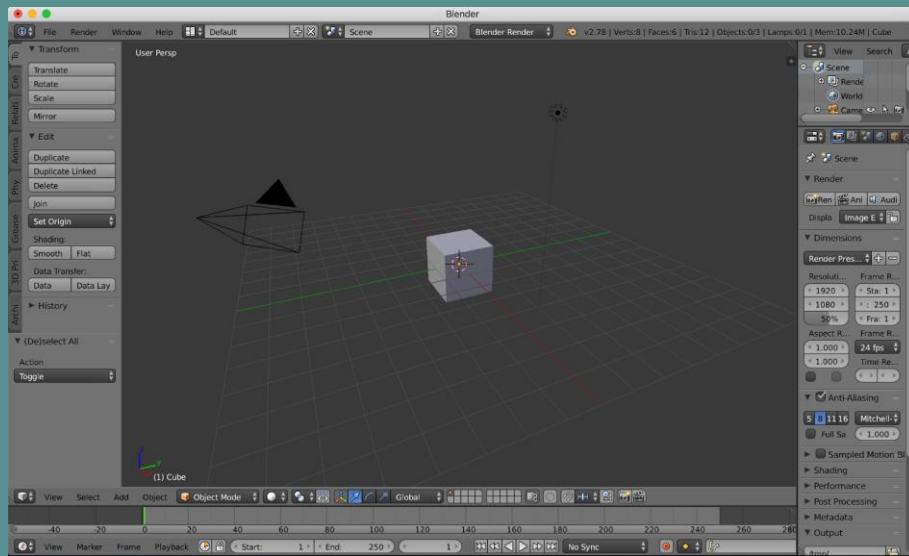
3D Graphics

By Chaya and Gabriela



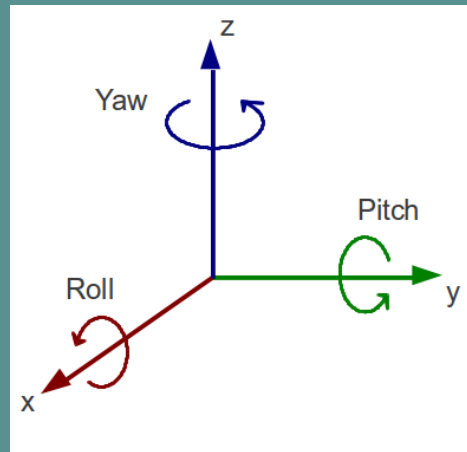
Background

- Purpose:
 - 3D Printing
 - Computer Aided Design
 - Video Games
 - 3D Simulations/Testing
- Programs/Modules:
 - Panda3D
 - VPython (needs python 2)
 - Pyglet
 - CAD
 - Blender



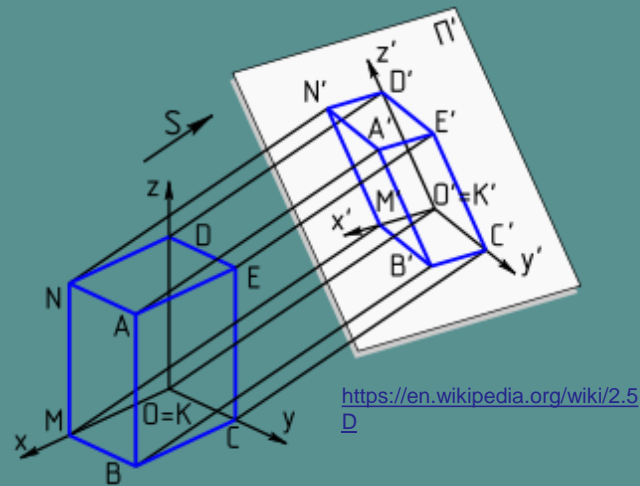
2D vs. 3D Graphics

- (x, y, z) vs, (x, y)
- $(h, p, r) = \text{yaw, pitch roll}$
- Rendering 3D models
- Lighting/Shadows
- Camera/Perspective
- Don't always have a sense of location



2.5D Graphics (pseudo 3D)

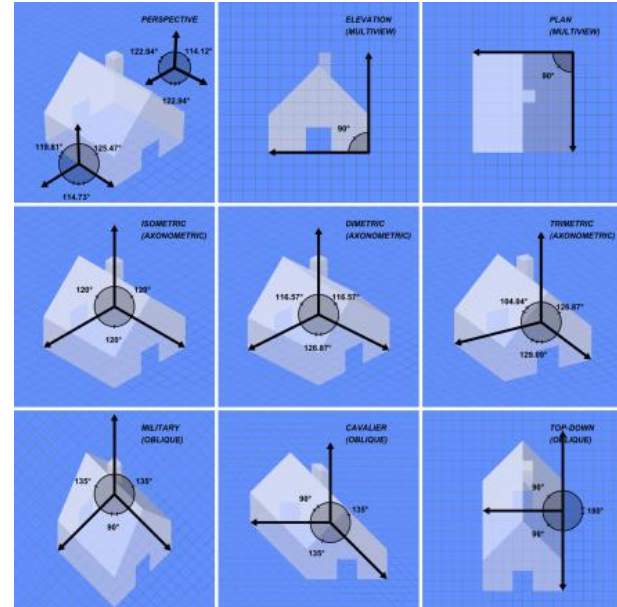
- Incorporates more geometry
- Side scrolling
- Oblique projection
- Axonometric projection





2.5D Tkinter Demo - Cube Demo

- Utilizes geometry
- Based on 120 degree angles





2.5D Tkinter Demo - Racecar Demo

- Utilize existing ~3D~ images
- Diagonal movement == perspective

Panda3D

- Module for 3D graphics
- Used for 3D rendering/load 3D models
- Create a 3D environment
- Create animations



Demos

- Ball-in-maze - `disableMouse()`, `base.mouseWatcherNode.hasMouse()`
- Culling- `self.camera.setPos()`, `self.camera.setHpr()`
- Shadows- `render.setShaderInput()`, `NodePath()`, `setShader()`
- Roaming Ralph- `CollisionTraverser()`, `collisionRay()`

Getting Started

- Showbase Class
 - imports most Panda3D models
 - Causes 3D window to appear
- Run()
 - Renders a frame
 - Handles background tasks (your code!!)
 - Repeats

```
from direct.showbase.ShowBase import ShowBase

class MyApp(ShowBase):

    def __init__(self):
        ShowBase.__init__(self)

app = MyApp()
app.run()
```

Loading and Rendering Models

- Connect models to your Showbase
- Load model - tells Panda3D that there is a model you wanna use!
- Render model
 - Self.render - top level node
 - Need to *reparent to* self.render



Setting up your Scene

Lights

- Point lights
- Directional Lights
- Ambient Lights

Background

- Just another model!

Working with Models

Position

- X, Y, Z
- `setPos()`, `getPos()`

Rotation

- H, P, R
- `setHpr()`, `getHpr()`

Intervals

Aka animation!!

Definition - “the playback of scripted actions”

- Setting up the interval
- Playing the interval

```
1 interval.start()  
2 interval.start(startT, endT, playRate)  
3 interval.loop()  
4 interval.loop(startT, endT, playRate)  
5 interval.finish()  
6 interval.pause()  
7 interval.resume()
```

Intervals

- PosInterval vs HprInterval
- Sequence vs Parallel


Key Presses - User Interaction

- Setting up the keypress
 - `self.accept`
- Activating the keypress
Binding a method to a keypress

Key Presses - Smooth Directional Movement

- KeyMap
 - Which keys are currently being pressed
- Self.accept
 - Will update the keyMap
- Task Manager (doMethodLater)
 - Similar to timerFired!
 - Performs actions based on the values of keyMap

Other Panda3D Features

- Impose 2D text/images on top of the screen
 - Splash Screens
 - Collision Detection
 - Timing
 - Etc. (check out the Panda3D manual)
- 
- A decorative pattern at the bottom of the slide consisting of a series of vertical bars of varying heights, each composed of three overlapping circles in shades of teal and light blue.

Demo

Back to the Other Demos



Questions?





Reminder to take attendance:
<https://tinyurl.com/3dgraphicsarecool>