

# Introduction to 3D Animation with Blender

Matthew Momjian

*Student*



O'Reilly OSCON  
2010



# Welcome!

- I realize that an artistic talk at OSCON is unusual, so I appreciate your attendance
- I hope that you all enjoy the tutorial
- We will break for 30 minutes halfway in (1h 30m)
- You will want to plug in your laptop – Blender is detrimental to battery life



# About me

- Matthew Momjian, 17 years old, from Newtown Square, PA (near Philadelphia)
- Attending Franklin & Marshall College in Lancaster, PA as a freshman this fall
- Working with computers for about 15 years
- My dad is Bruce Momjian, last year's Google Open Source Awards “Database Jedi Master” winner. He works on the PostgreSQL project



# Why I Use Blender

- Working with Blender since mid 2006 (v 2.42)
- I've never been very artistic, but I'm good at Blender due to my extensive computer experience
- Blender *is* an artistic tool, but you don't have to be artistic to use it – you just have to be willing to learn
- I enjoy using Blender on my quad-core server, running Ubuntu, with a Wacom Intuos 4 tablet



# Some Housekeeping

- To get the most of this talk you will need to have Blender 2.5 Beta installed.
- Presentation files and slides can be found at <http://momjian.us/matthew/oscon>
- At that link you can find a zip of the presentation files and slides
- There are also zips of the most recent Blender daily builds, since Beta 1 isn't out yet



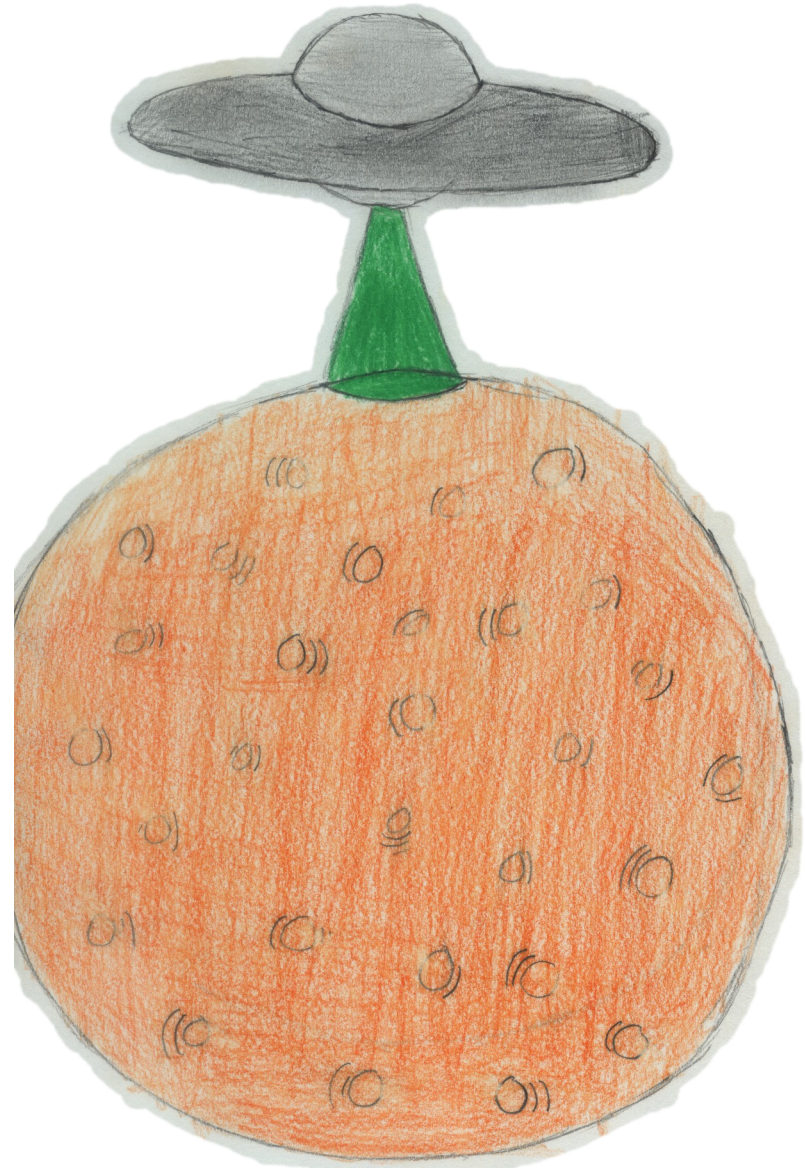
# Tutorial Structure

- The tutorial will consist of 7 sections and 7 exercises, working towards a final scene consisting of a UFO
- The first half of the tutorial will be more hands-on, whereas the latter half will be a hands-off broad technical overview
- A new file will be provided for each exercise
- Doing the exercises is highly recommended

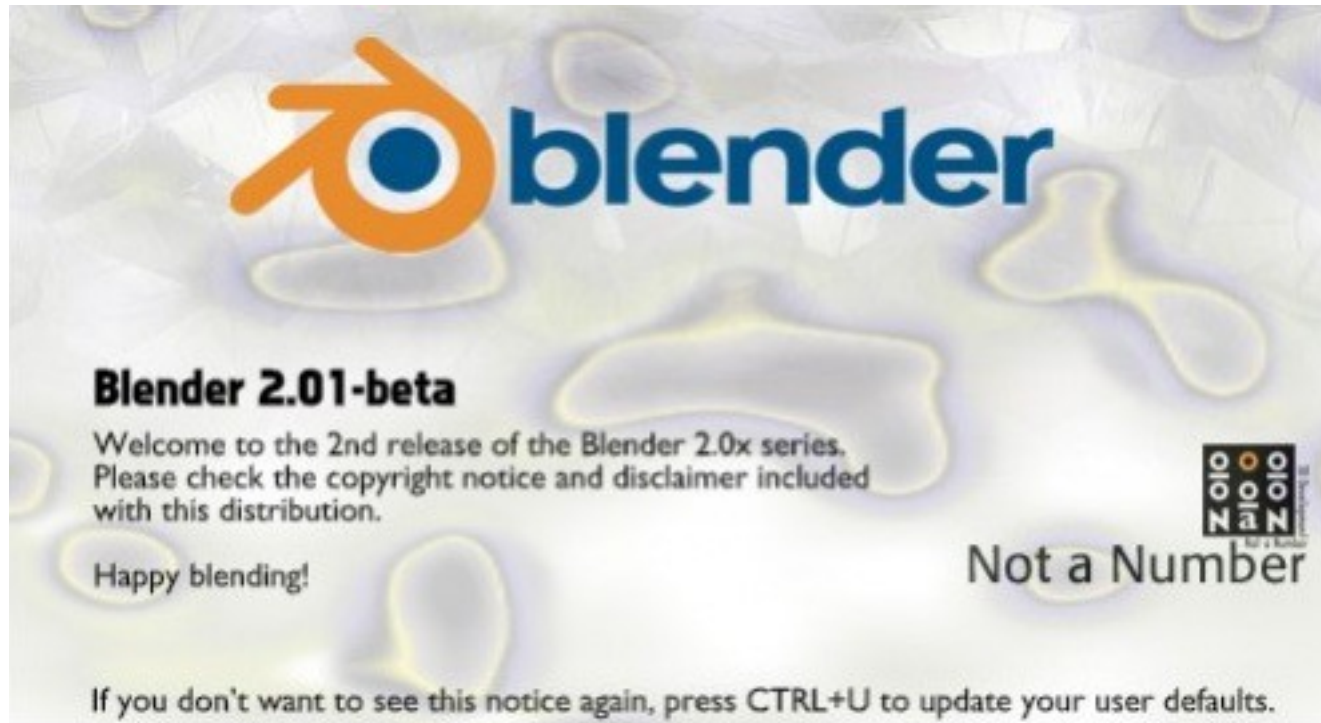


# Final Scene

- Imagine that we have been given a scene concept art by the art department
- We will be learning the skills required to make this concept into a short movie clip



# Introduction



Early 2003





# So... What's this “Blender”?

- The leading open-source 3D animation suite
- Authored in '88 by Ton Roosendaal
- Open-sourced GPL in 2002 after a €100,000 (\$100,000 at the time), 7-week fundraiser
- Currently in Beta of a major rewrite (v. 2.5)
- We will be using 2.5 Beta since there are many new features



# Blender Foundation

- Analogous to *Canonical* (Ubuntu)
- Chaired by Blender author Ton Roosendaal
- Responsible for the oversight and development of Blender
- Also oversees the production of Open Movies and Open Games
- Approximately 4 full-time staff and 6-8 developers



# Blender File Format

- Blender's scenes are stored in .blend files which are simple, compressable database files (similar to SQLite)
- This database file stores objects, meshes, materials, textures, properties, and UI layout
- It can also *pack* external resources like texture images, sound clips, or movies
- Blender can create dynamic links to the resources in another file with *linking*



# Blender vs GIMP

- Lots of people have probably used GIMP, so I think it would be helpful to compare the two programs
- Common traits
  - Both are GPL open-source software
  - Both use Python for scripting
  - Both create 2D images and deal with multimedia



# Blender vs GIMP

- Blender

- All data is stored in a database as mathematical points
- Since all content is created from scratch, projects take a large investment

- GIMP

- Data is stored as a series of color points, limiting modifiability
- Easier to pick up quickly since existing content can be edited



# What is 3D animation?

## Union of many vital components

- Content
  - Modeling
  - Materials
  - Textures
- Presentation
  - Animation
  - Lighting
  - Rendering & Compositing

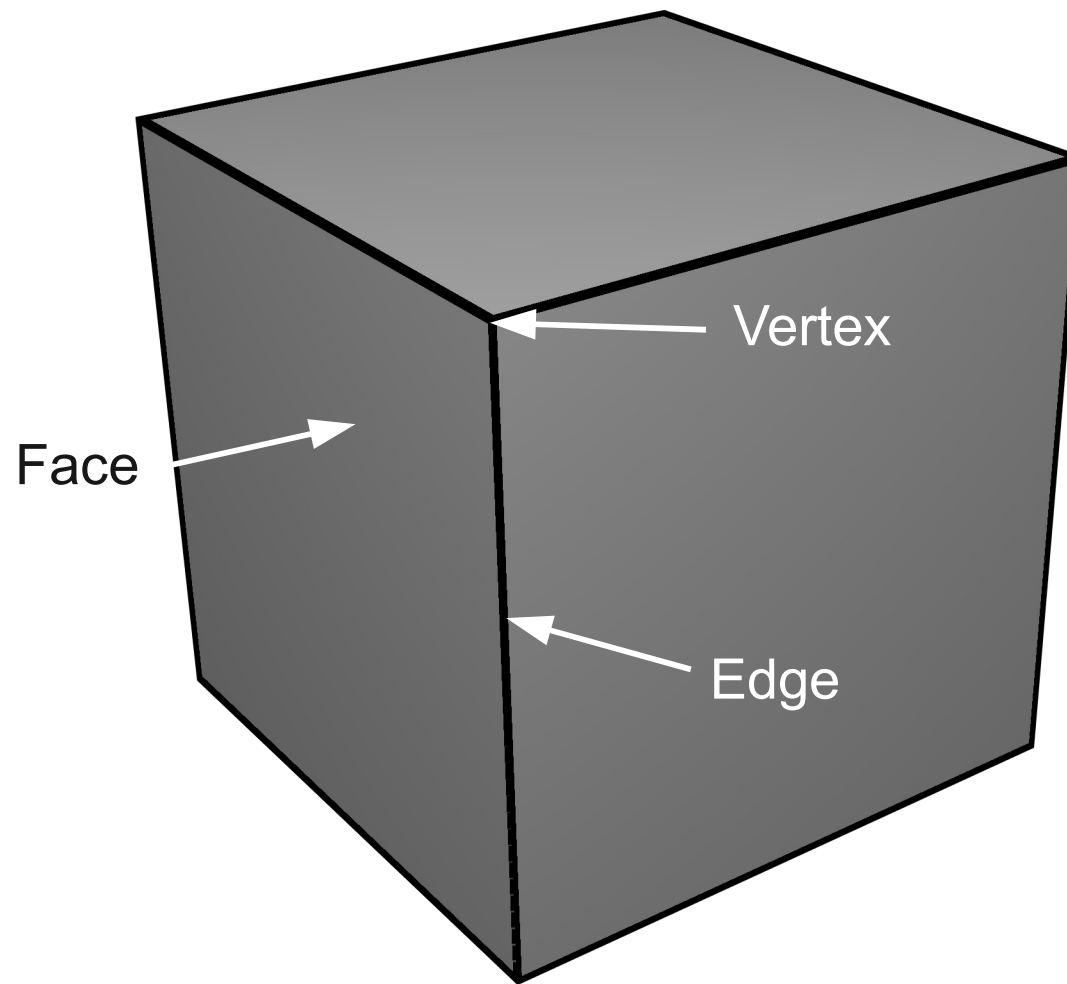


# Modeling

- Construction methods
  - Polygonal (mesh-based)
  - NURBS (curve-based)
- Elements
  - Faces are composed of  $n$ -edges, which are composed of 2 vertices



# Mesh Modeling Visual Primer





# Materials

- Defined through various reflective properties
- Represent how an object reflects light
- The values are calculated by shader models
- Materials can also be used for smoke and light effects



# Textures

- Used to dynamically modify material properties, like color, etc.
- Procedural (algorithmic) or image
- Can also be used for overlaying images



# Animation

- Keyed
  - Key points are set and interpolated
  - Good for organic objects
  - Can be controlled through armatures (skeletons which deform meshes)
- Physical
  - The physics engine calculates animation values
  - Handles collisions, soft body, cloth, fluid, smoke, particles, and hair



# Lighting

- Lighting works just like in real life – the reflection of it is what we see in renders
- One of the most common difficulties for beginners
- Lighting can make or break a scene
- The effect of lighting is dependent on material properties



# Rendering and Compositing

- Rendering

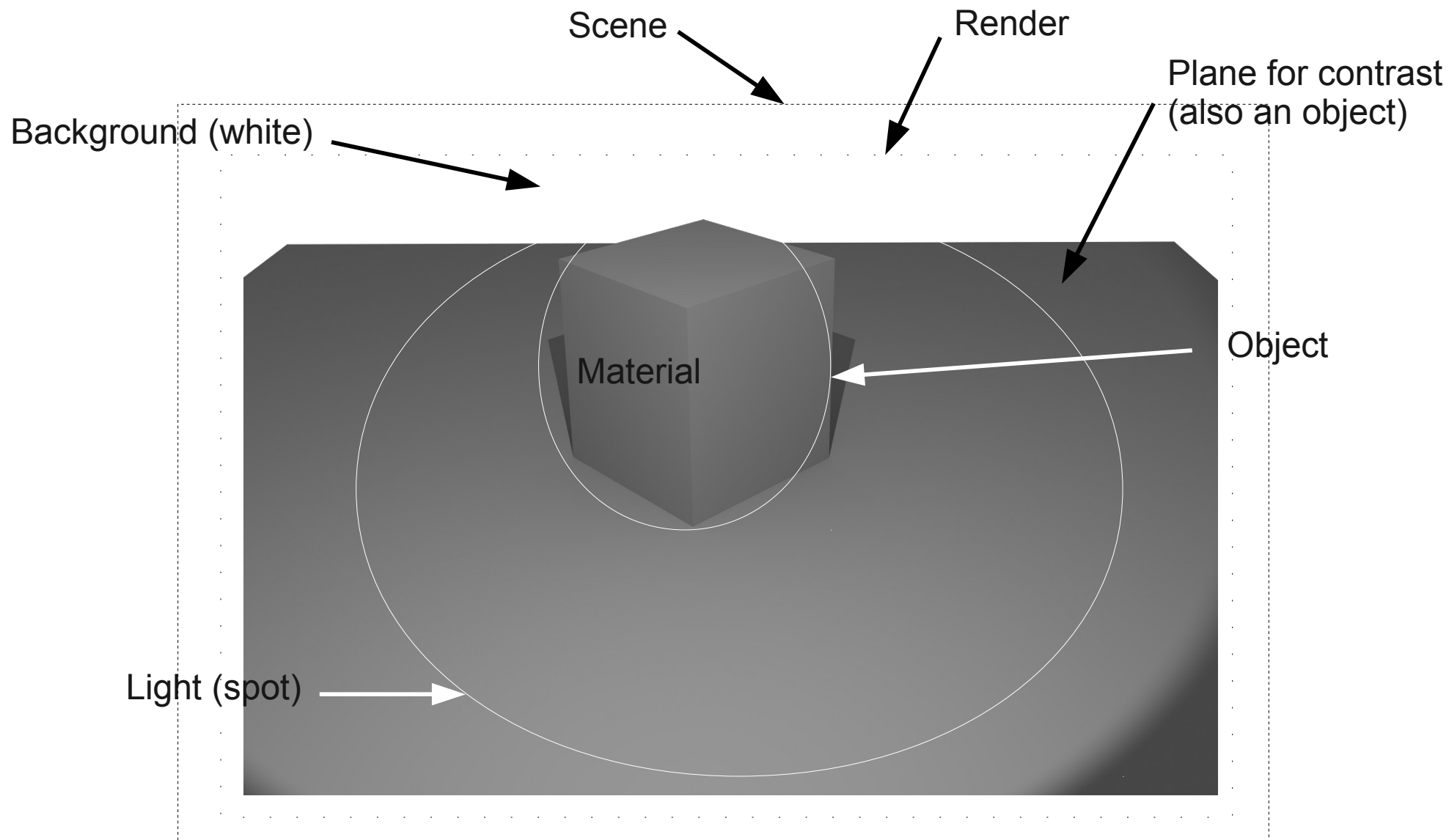
- How a 3D scene is parsed into a 2D image
- For each pixel, the renderer decides the color based on material, texture, etc
- Handles effects like halos, oversampling, motion blur

- Compositing

- Also known as post-processing
- Blender's built-in method is node-based
- Takes 2D input from the renderer and tweaks it
- Handles effects like bloom, green-screen, contrast tweaks



# Vital Components Visual Primer



# Working in Three Dimensions

- 0 dimensions – a point
- 1 dimension – a line
- 2 dimensions – a plane (think piece of paper)
- 3 dimensions – a volume (like a cube)
- In Blender we will always be working in 3 dimensions, but the 3D objects are composed of 0D, 1D, and 2D elements



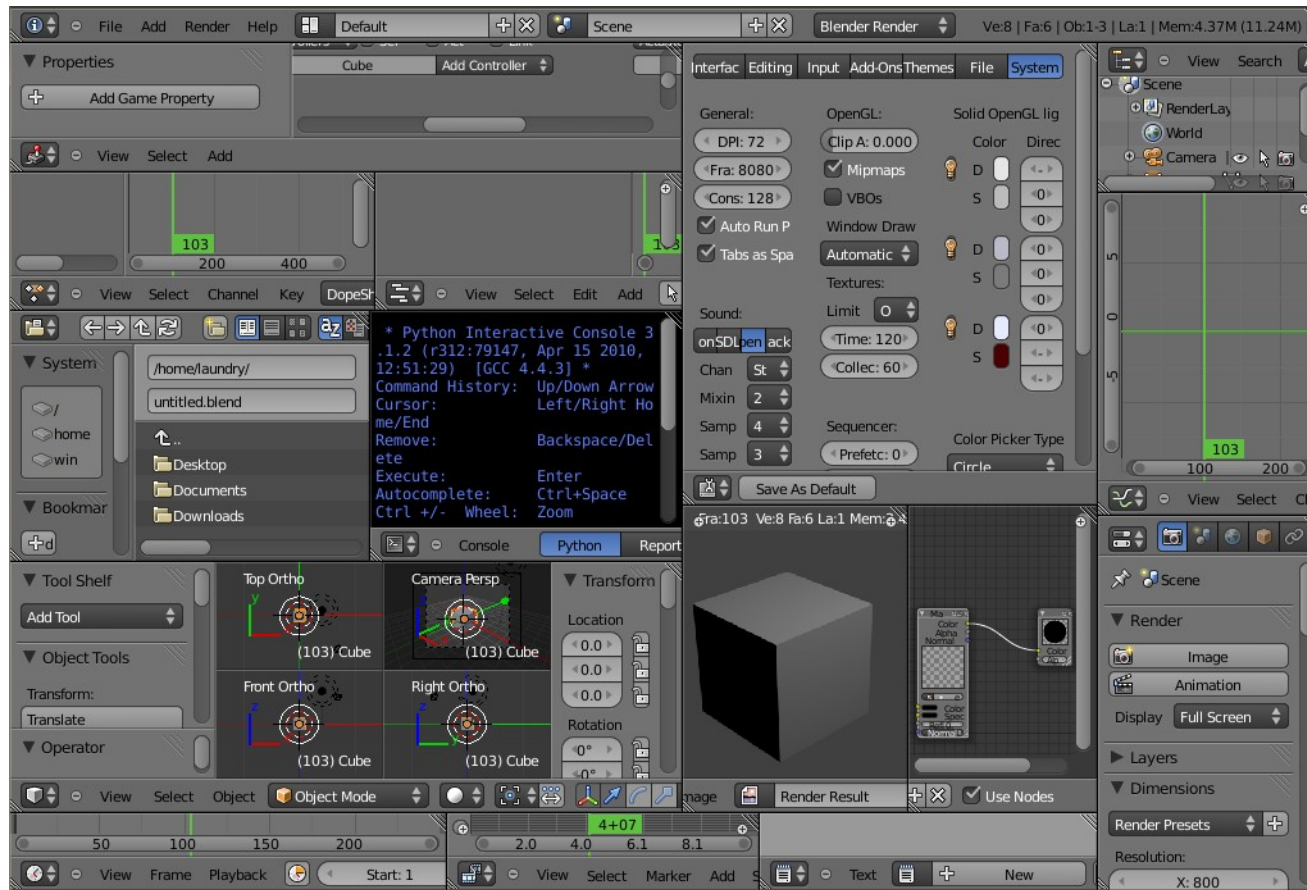
# Introduction Wrap-Up

- Any questions so far about the history of Blender or basic 3D concepts?
- Please raise your hand to ask relevant questions at any time
- More abstract questions will be taken at the end of each section
- General questions at the conclusion of the tutorial





# Interface



# Overview of the Blender Interface

- Blender uses a modular window system – there is no fixed interface design
- There are 16 window types that can be arranged to form a custom workspace
- Since the design is completely modular, favorite layouts can be organized into Screens for different purposes



# Objects

- An object is a data block that contains a mesh, along with materials, textures, and other properties
- A mesh is the set of vertices that actually makes the object visible
- Objects can be organized into Layers to restrict visibility / renderability



# Blender Interface

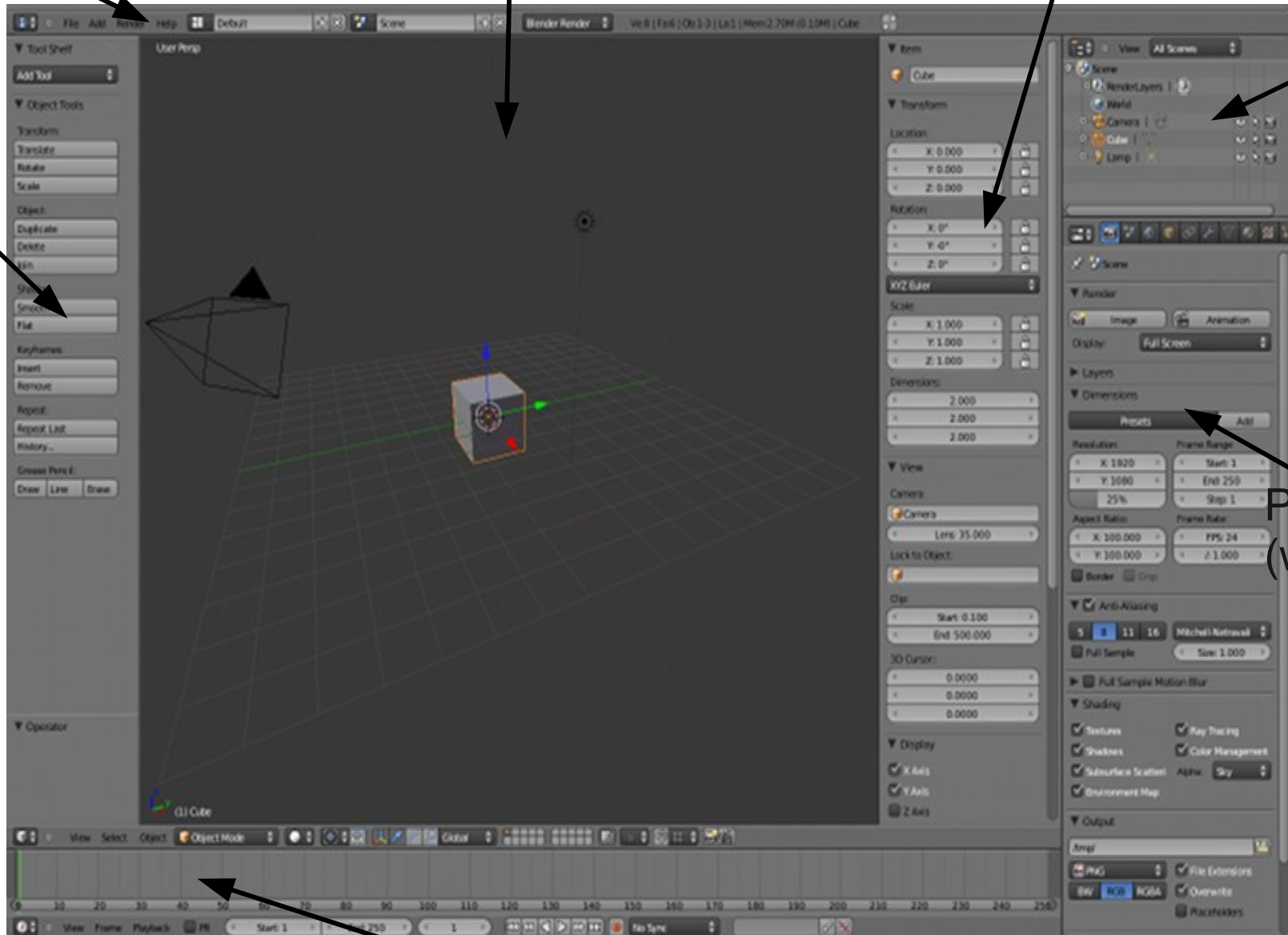
Info (window)

3D View (window)

Properties (panel)

Outliner  
(window)

Tool shelf  
(panel)

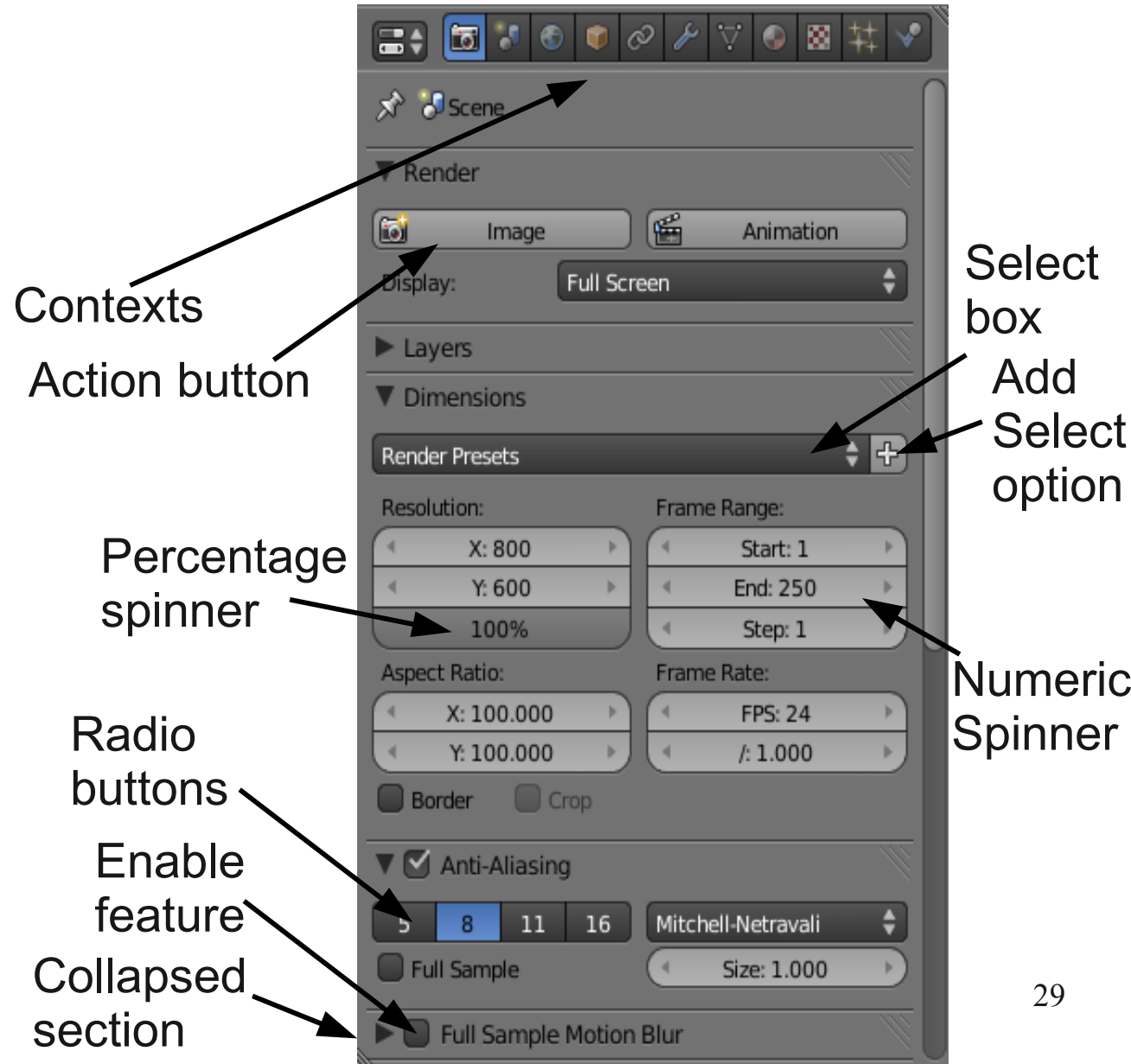


Properties  
(window)

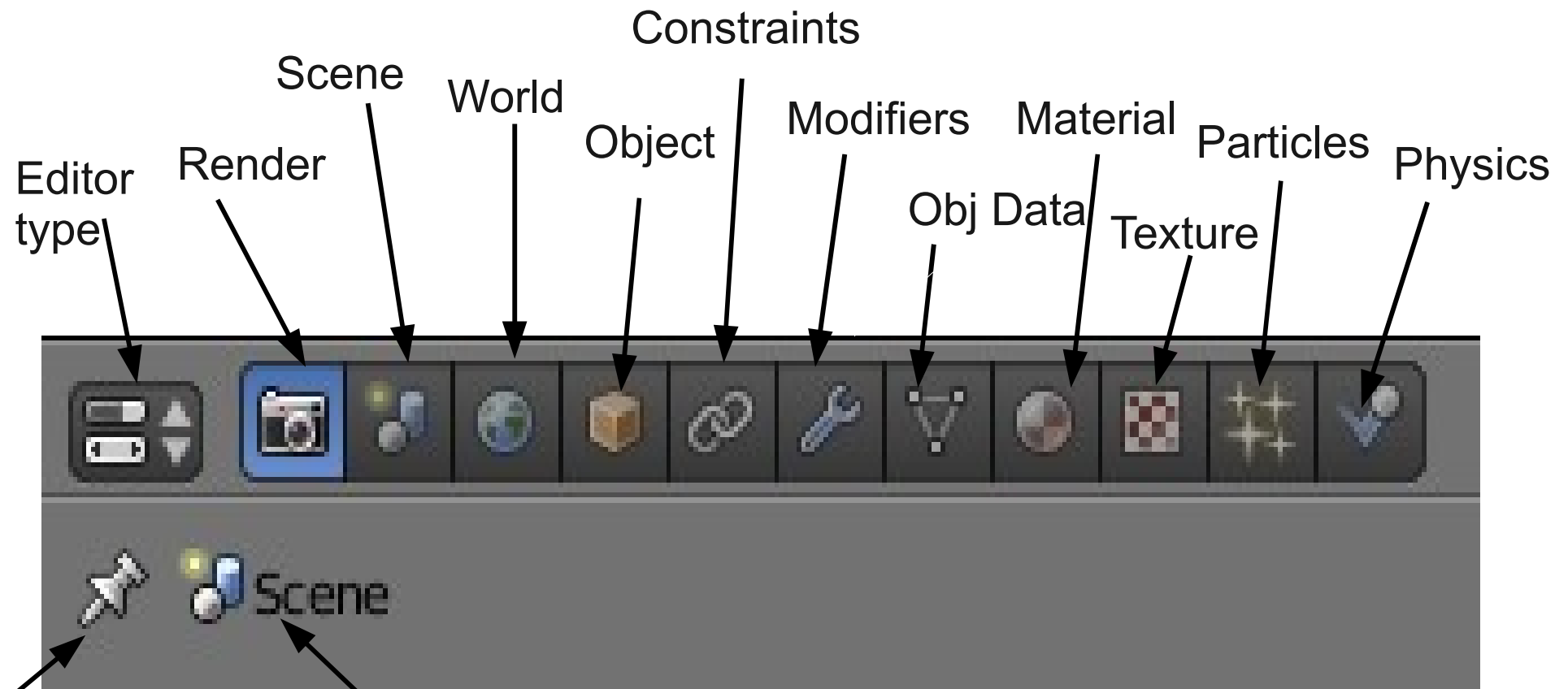


# Properties Window

The properties window contains settings for rendering, shading, textures, and other features



# Properties Contexts



Pin window  
to context

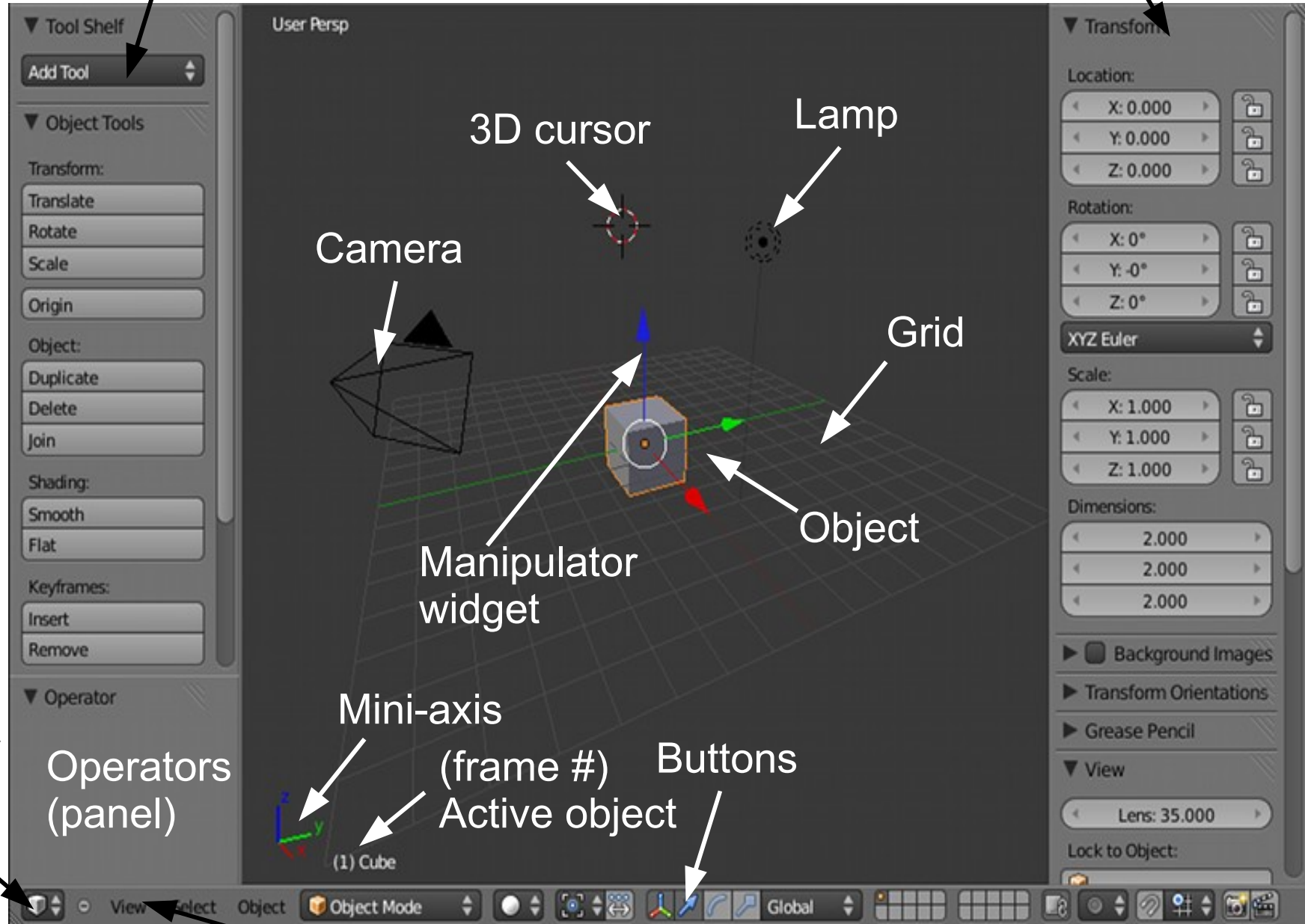
Breadcrumb trail  
Drag-drop to apply  
to another object



# The 3D View

Tool shelf  
(panel)

Properties  
(panel)



# Interface Exercise

- To get the most out of Blender, you will need to know how to use it
- At this time please start Blender 2.5 Beta





# Configuration

- Once you have Blender running, go to File > User Preferences
- Under the Input tab, enable “Emulate 3 Button Mouse”
- Also enable the option “Emulate Numpad” if your laptop doesn't have a dedicated numpad



# Moving in the 3D World

- Blender is designed with a “One hand on mouse, one hand on keyboard” philosophy
- The mouse handles 3D movement, whereas the keyboard controls actions



# Mouse Controls

- Middle mouse button (MMB) rotates the 3D view
- Shift+MMB pans the view
- Ctrl+MMB or scroll wheel zooms the view
- If you have a 2-button mouse, use Alt+LMB instead of MMB. E.g., Shift+Alt+LMB



# Keyboard Controls

- Blender has 1,182 default keybindings and gestures
- Don't panic! There are GUI methods to do almost everything
- However, the keyboard is usually faster than using the GUI
- We will start out by the tools to move, rotate, and scale objects in the 3D world



# Keyboard Controls

- Grab (G), is used to move objects in the 3D world.
- Rotate (R), is used to rotate objects
- Scale (S), is used to change the size of objects
- X, Y, Z are used to restrict the axis on the above three tools, and you can use Shift+ to lock (disable) an axis



# View Controls

- There are a number of preset view positions in Blender
  - NUM 1 – Front view
  - NUM 3 – Side view (right)
  - NUM 7 – Top view
  - Press Ctrl+number to view the opposing side
  - NUM 0 – camera view toggle
  - NUM 5 – perspective / orthographic view toggle



# Look Around

- We will now break for 5 minutes
- During this time, please acquaint yourself with Blender
- I will come around to see if anyone needs help



# Interface Wrap-Up

- Was everyone able to navigate in Blender without any problems?
- Any questions about the Blender interface?
- New window types
  - Properties – changing settings and properties
  - 3D View – manipulation of objects
  - Timeline – frame handling
  - Outliner – lists all assets in a scene





# Modeling



# Faces

- Faces were traditionally only composed of 3 edges. This forms triangles called *tri's*
- Modern modeling allows >3-sided faces, commonly 4-sided (*quads*).
- Faces have only one side – the *normal*
- Therefore faces alone are not meant to be a final product – another step is required



# The Next Step – Volumes

- Volumes are composed of  $n$ -faces – the least number possible is 4, composing a tetrahedron
- All volumes are three-dimensional



# Smooth Mesh Volumes

- Since there are a finite number of elements, it is difficult to render naturally smooth edges
- 3 possible solutions to this:
  - Faces can be lighted as smooth instead of solid, eliminating hard edges
  - Solids can be *subdivided*, causing the computer to interpolate between elements
  - NURBS meshes (vector)



# NURBS “meshes”

- Stands for “non-uniform rational basis spline”
- Comparable to vector graphics
- Defined by mathematical formulae
- Ideal for curved objects like:
  - Spheres
  - Organic amoeba-like organisms
  - Camera tracks and animation paths



# Modifiers

- Blender has a system of *Modifiers* that is used to apply real-time changes to meshes
  - Subdivision Surface – divides mesh to smooth hard edges, useful with Smooth Shading
  - Solidify – converts a 2D skin into a 3D volume
  - Mirror – allows you to model only half of an object, helpful for human heads and vehicles
  - Multiresolution – maintains a mesh at multiple subdivision levels, used with sculpting



# Modeling Exercise

- In this exercise we will be modeling:
  - UFO and cockpit (2 UV spheres)
  - The energy beam (cone)
  - The planet (an cosphere)
- Follow along as we start with a new scene (Ctrl-N), and raise your hand if you have a question or problem
- Don't panic about following exactly, as we'll start with a new file for each section



# Modeling Wrap-Up

- Any questions about the exercise?
- Any modeling questions?
- New window types
  - Properties (Modifiers context) – used to add modifiers to a mesh





# Materials



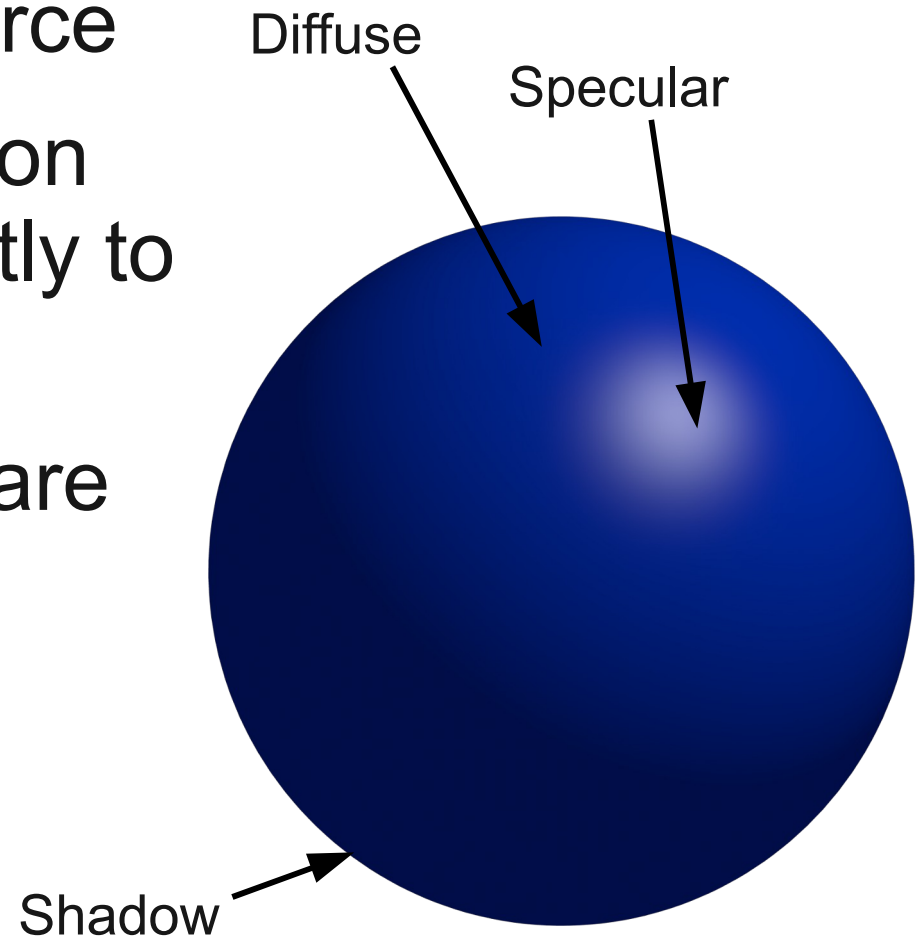
# Materials

- So far we haven't added any materials, so everything is pure gray
- *Materials* are used to provide color to an object
- A *material* in Blender is technically defined as a set of properties that represents how something reflects, emits, and transmits light



# Material Lighting Terminology

- *Specular* light is the direct reflection from a light source
- *Diffuse* light is the reflection that does not return directly to the viewer
- *Shadows* are points that are not reached by the light



# Material Properties

- *Hardness* controls the size of the specular reflection – a higher number means a smaller, more defined specular reflection
- *Alpha*, also known as opacity, controls how visible the material is, with zero being invisible. Controlled in the *Transparency* tab
- *Mirror* allows a material to reflect its surroundings. Controlled in the *Mirror* tab

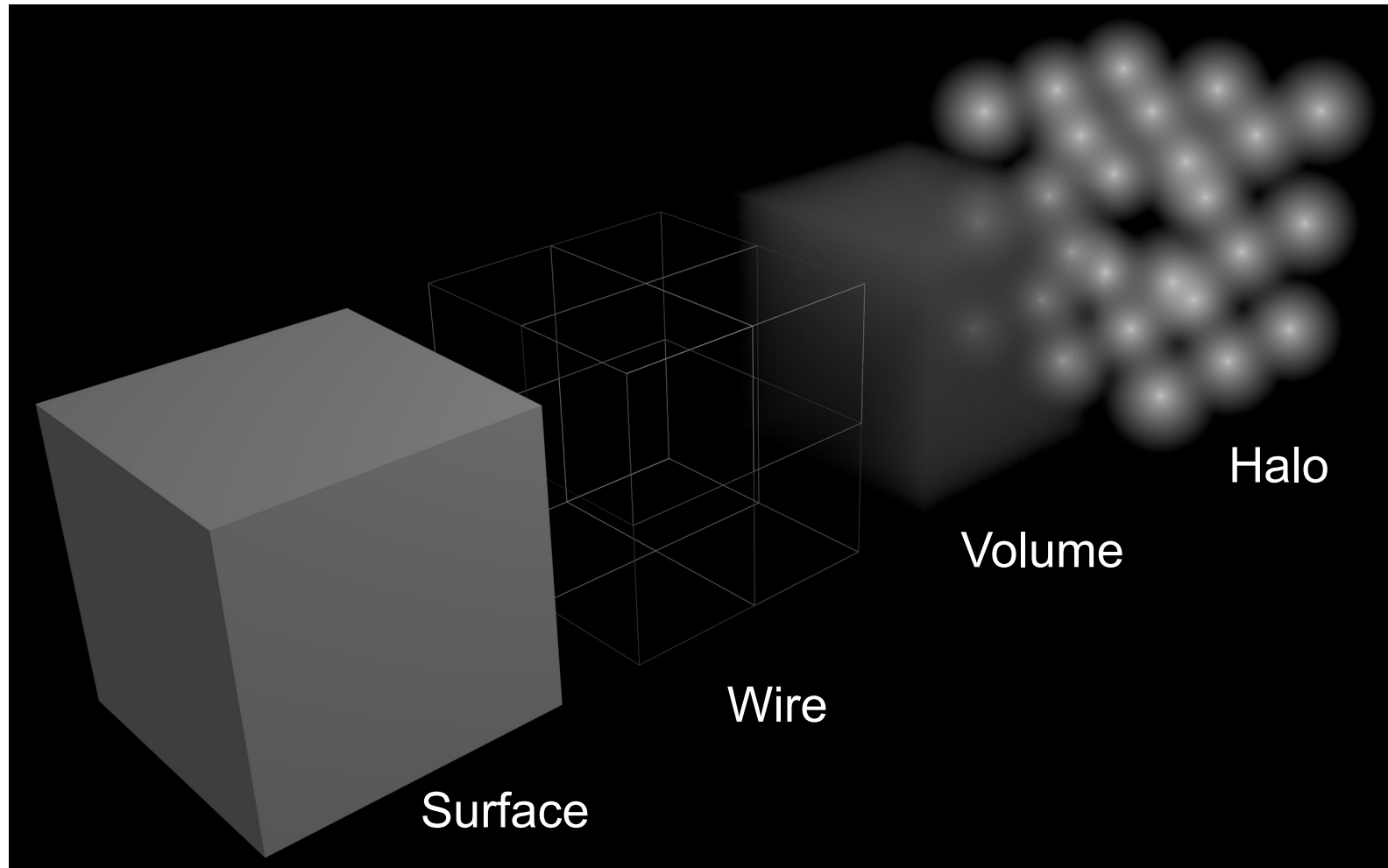


# Non-Surface Material Models

- So far we have been talking about surface, or solid, rendering
- However, there are other render types
- *Wire* renders only the edges, not the faces
- *Volume* renders the object as a volumetric gas, like smoke
- *Halo* renders the points as light sources. Good for lens flare and laser effects



# Material Models



# Materials Exercise

- For the exercise we will add a mirror material for the UFO
- We will also add a volumetric material to the energy beam
- Please load the file “1-Materials.blend”
- The loaded file will look similar to what we finished with in Modeling



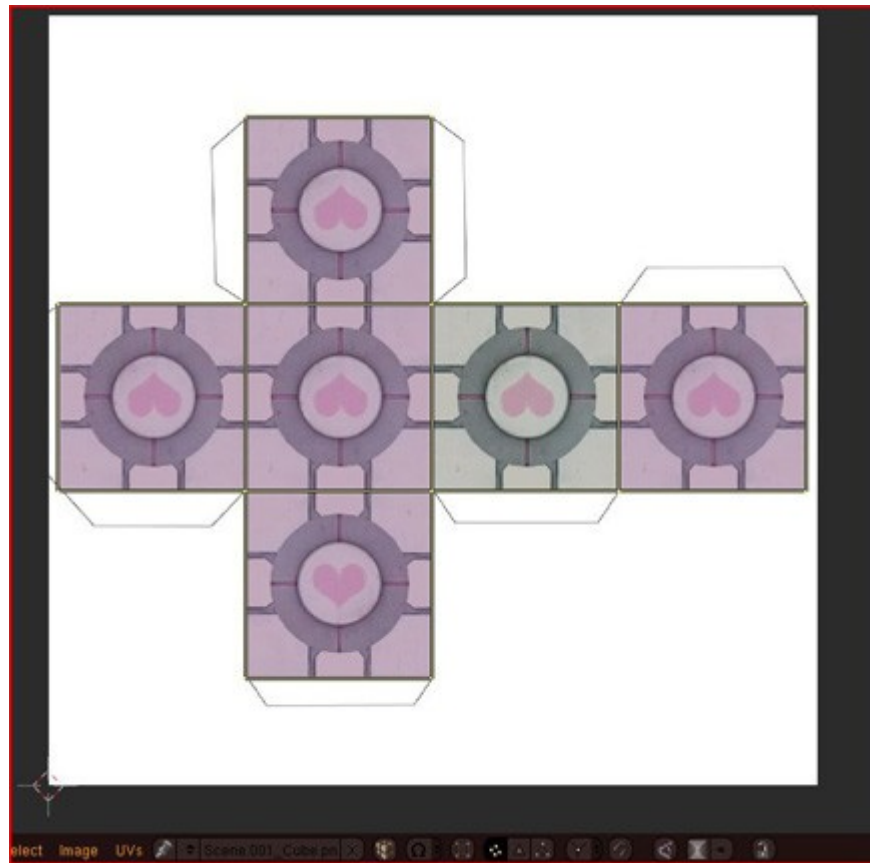
# Materials Wrap-Up

- Any questions about the exercise?
- Any materials questions?
- New window types
  - Properties (Material context) – options for modifying material and shader properties





# Textures



# Textures

- Textures are applied to various material properties
- They can modify many material properties, like Color, Alpha, and Normal
- There are two types of textures
- Procedural textures are mathematically generated
- Image textures are based on external images



# Procedural Textures

- Procedural textures are generated through mathematical algorithms
- Procedural textures create values between 0 and 1
- The values are mapped to influence – a value of 1 means maximum change, whereas 0 has no change
- They are useful for rough surfaces, static, clouds, wood, and other natural patterns



# Image Textures

- Are often used for decals and embellishments
- They can be applied as a procedural texture where they influence attributes
- Or, the image can be mapped right onto the mesh using *UV mapping*

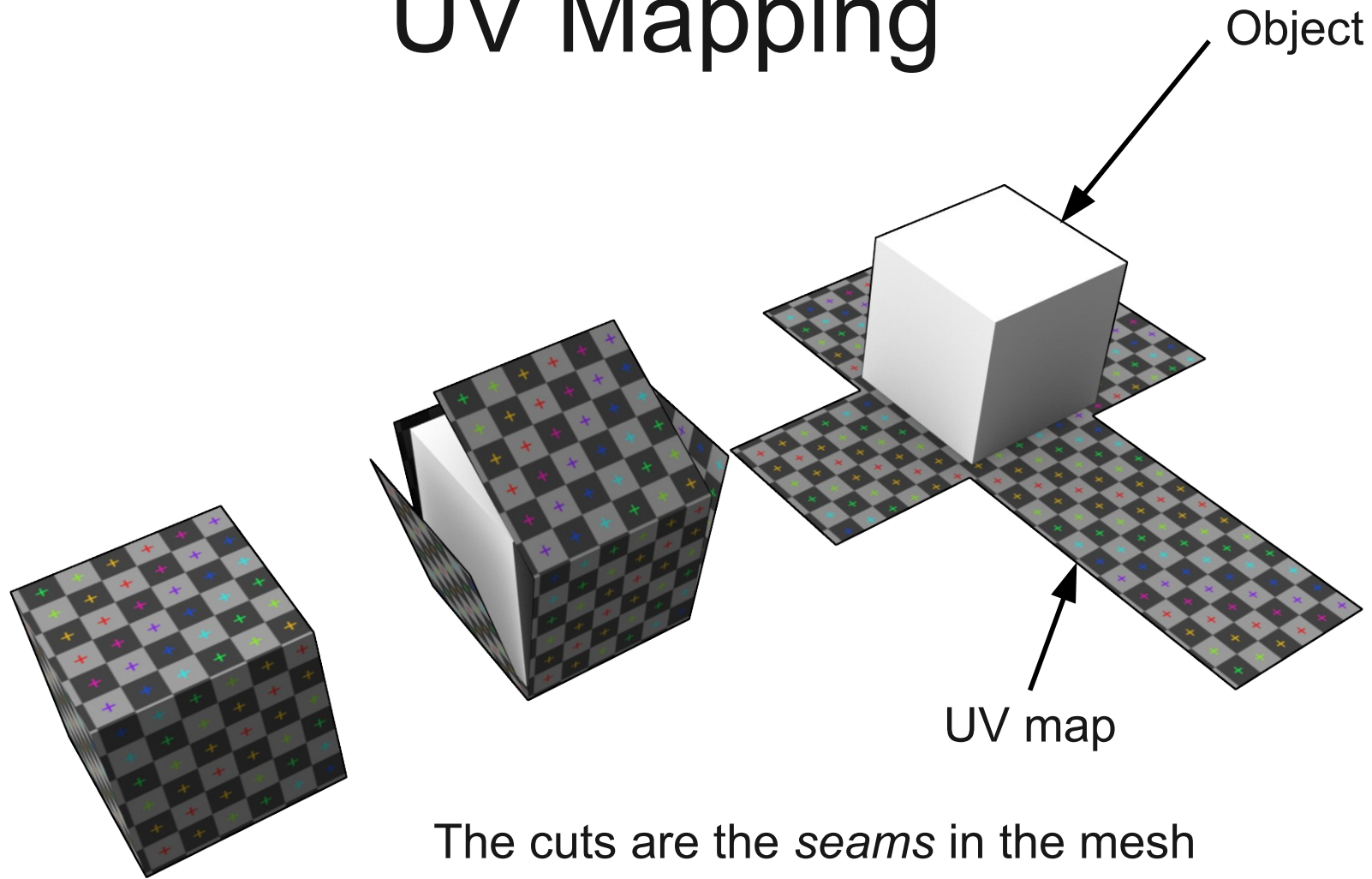


# UV Mapping

- UV mapping stands for the coordinates used –  $U = X$ , and  $V = Y$
- It is the process of flattening a 3D mesh into the (2D) UV space to overlay a 2D image
- The software can automatically flatten some common shapes like spheres and cubes at predefined *seams*

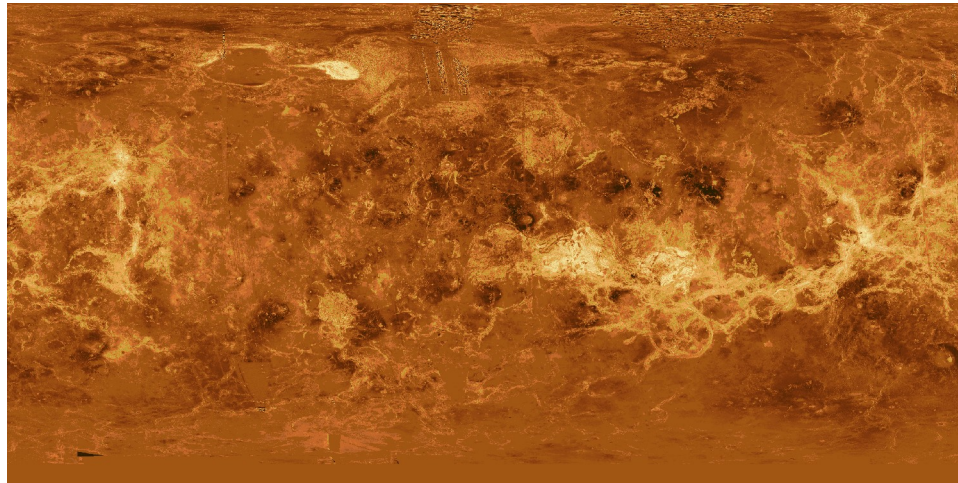


# UV Mapping



# Textures Exercise

- For texturing, the exercise will be UV mapping a picture of mars onto the planet
- Please load the file “2-Textures.blend”



# Textures Wrap-Up

- Any questions about the exercise?
- Any texturing questions?
- New window types
  - Properties (Texture context) – provides options for modifying texture properties, mapping, and influences
  - UV/Image Editor – used to map UV nodes to an image. Can also be used to view render result



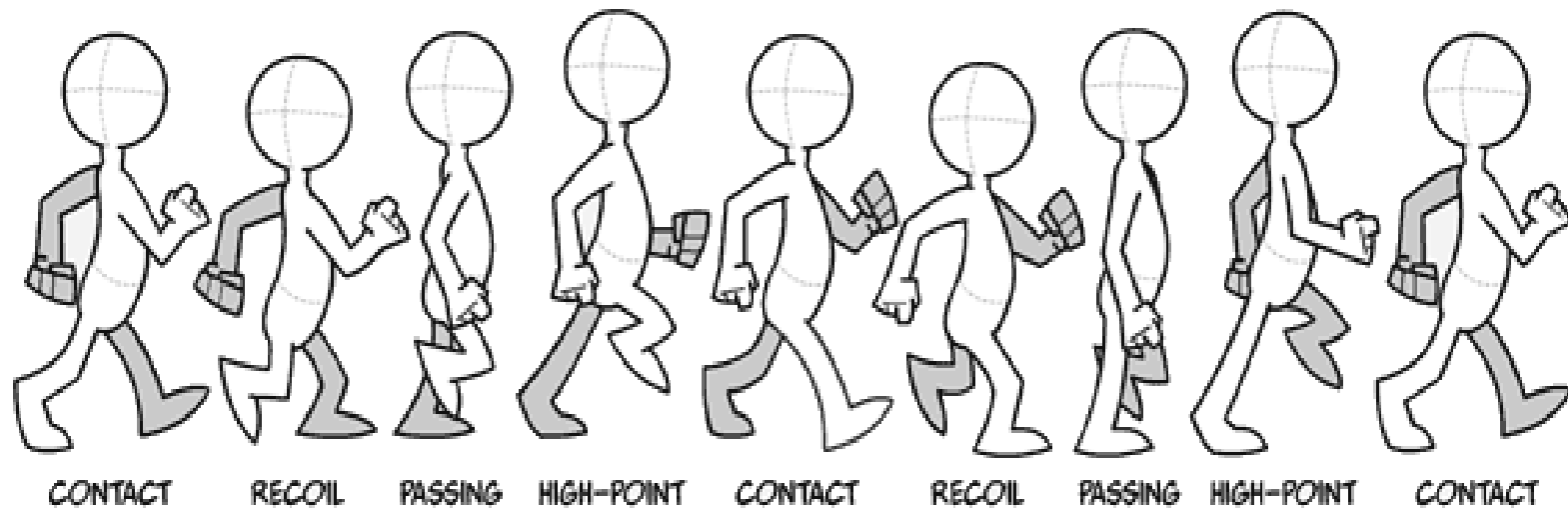


# Break

- Now that we have covered Content, we will be taking a 30 minute break
- When we reconvene we will continue on to Presentation
  - Animation
  - Lighting
  - Rendering and Compositing



# Animation



# Animation Basics

- So far we have been making stills – but 3D animation is about movies
- The way most animations are done is by *keyframing*, which involves setting keys (values) for the desired frames
- Not all animation is manual, as there are a variety of external engines for physical simulations



# Animation Theory

- In Blender, animations are handled as graphs – the X axis is time, and the Y axis is value
- Values can be things like Location or Rotation
- The value channels are called F-Curves, and there is one F-Curve per value per axis
- F-Curves can also have modifiers as simple as noise, or as complex as a Python script



# Keyframing Basics

- In keyframing, the desired frame is selected, then by pressing the I key, you can insert keys for that frame
- The mid-frame values are interpolated to provide a smooth transition
- There are various interpolation and extrapolation options



# Keyframing Options

- In addition to animating objects, keyframes can be used to animate properties
- To animate a property, you place the cursor over the property and press I. Then, advance to another frame, change the value, and insert another keyframe
- This can be used to animate properties like Lamp Energy, Color, or Camera Lens Angle



# Animation Workflow

- So far we have talked about F-Curves
- However, to ease development, a set of F-Curves can be stored as Actions to be reused
- These Actions can then be arranged in the Non-Linear Action Editor (NLA) to create complex animations from a common set of base curves



# Simulation Engines

- Sometimes manual animation is too difficult or not believable. These automatic simulation engines can be enabled in the Physics context
  - Collision
  - Fluid
  - Smoke
  - Force Field
  - Soft Body
  - Cloth





# Collision

- Physics collisions are handled by Bullet, an open-source physics library used in many programs
- To enable collisions, switch the render engine from “Blender Render” to “Blender Game” in the header
- In the Physics context, enable the Actor checkbox
- If the object should be moving, change Physics Type to Rigid Body. Otherwise, the object will be static but still collide
- Then, start the simulation with the P key over the 3D view



# Fluid

- The Fluid simulation engine is called El'Beem
- To add a fluid simulation, click “Add” in the Physics context. Then, select a type – You will need a domain (a cube), and an Icosphere as a Fluid
- Then, press Bake Fluid Simulation. It will take a while.
- Now, you can apply a Shade Smooth and Subdivision Surface to the fluid object



# Smoke and Force Field

- Smoke
  - New in 2.5, useful for campfires, etc.
  - Fluid-based
  - Can be influenced by Force Fields
  - Advanced color ramps based on density, temperature
- Force Field
  - New in 2.5
  - Can influence Smoke and Cloth physics
  - Types include generic Force, Wind, and Vortex



# Soft Body and Cloth

- Soft Body
  - Provides a bouncing effect like jello
  - Depending on settings, it can also be used to simulate things like skin and claymation
  - Goals can be specified to tweak the effect
- Cloth
  - Used for, obviously, cloth
  - There are presets for many common cloths like Cotton, Leather, or Silk
  - Can be influenced by Force Fields



# Particle Effects

- Blender can also automate particle effects, available in the Particles context
- Used for effects like hair and sand
- Can also be programmed with advanced goal calculations to simulate things like swarms of insects



# Animation Exercise

- For the animation exercise, we will be animating the UFO to fly into the scene and over the planet.
- Please load the file “3-Animation.blend”

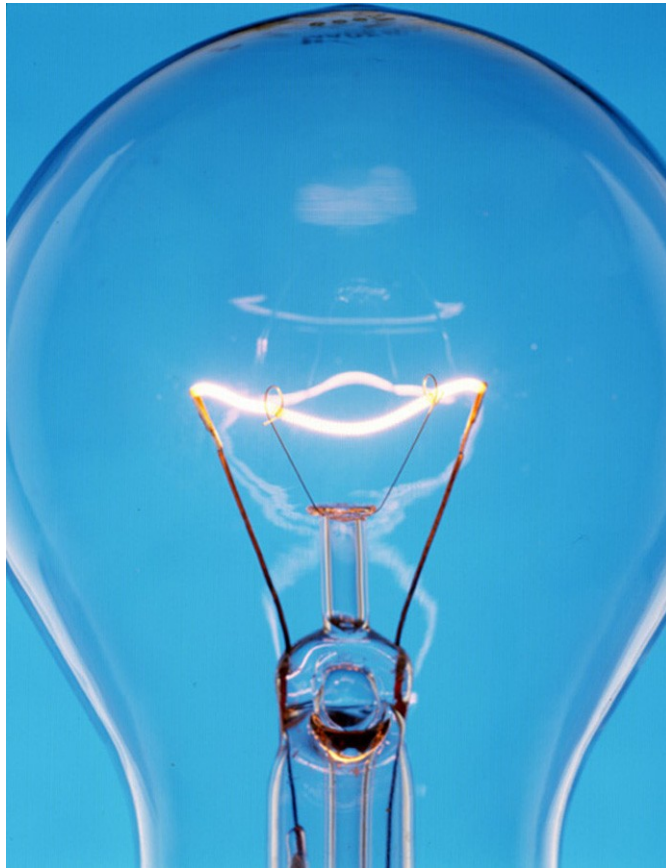


# Animation Wrap-Up

- Any questions about the exercise?
- Any animation questions?
- New window types
  - Graph Editor – provides options to modify F-Curves
  - Properties (Particles context) – used to enable particle systems
  - Properties (Physics context) – used to add physics simulations
  - NLA Editor and Dopesheet – used for Action clips



# Lighting





# Lighting

- Lighting is what brings a scene together
- In a render, we see the light “reflecting” off of materials as defined by the material settings
- Lighting can be pure white or colored to provide emotion to a scene
- There are 5 main types of lamps, and all are useful in their element



# Lamp Types

- Point – similar to an incandescent lightbulb
- Sun – like the sun, all shadows are parallel
- Spot – creates a circular directed light like a spotlight, can create volumetric halos. Can render buffer shadows, which create quick soft shadows
- Hemi – creates global lighting in a hemispheric shape with no shadows. Useful for color tone
- Area – a flat panel light similar to a diffuser panel with soft raytraced shadows. Most realistic light type, but also most CPU intensive



# Lighting Mood Colors

- White – while you may think that all light is white, it often looks unnatural. Good for studio lighting or sterility
- Red – conveys intensity, aggression, warmth
- Blue – conveys coldness
- Green – ugly and creepy indoors, but conveys warmth and vegetation outdoors
- Yellow – conveys incandescent light indoors, and can also simulate sunlight

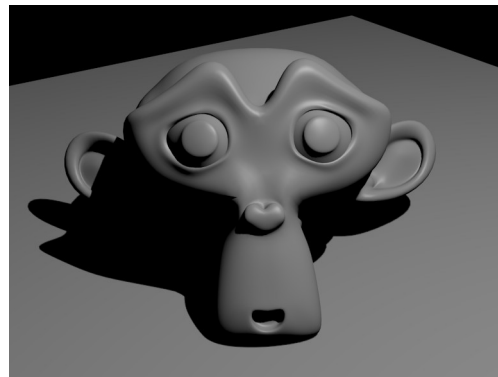


# Ambient Occlusion

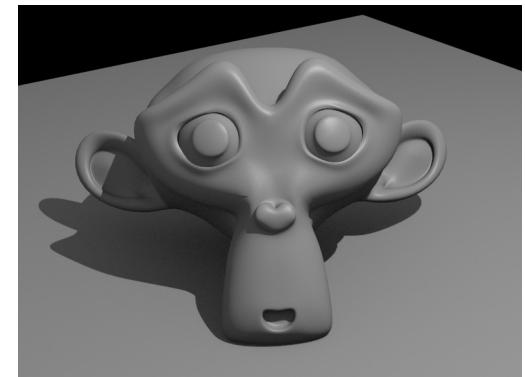
- Ambient occlusion is a type of global lighting that creates soft shadows, but is hard on CPU
- Simulates the natural reflection of light from clouds, walls, and other non-emitting surfaces
- Enabled in World context of properties

Note how the back side is now properly lit

No Ambient Occlusion



With Ambient Occlusion



# Lamp Options

- Energy – controls the brightness of a lamp
- Negative – causes the lamp to subtract light from where it would normally add light
- Specular – enable specular highlights from a lamp. If you have many lamps, only a few should have this enabled, or it looks too busy
- Diffuse – enable diffuse shading. Disable if you want a very controlled (specular) light



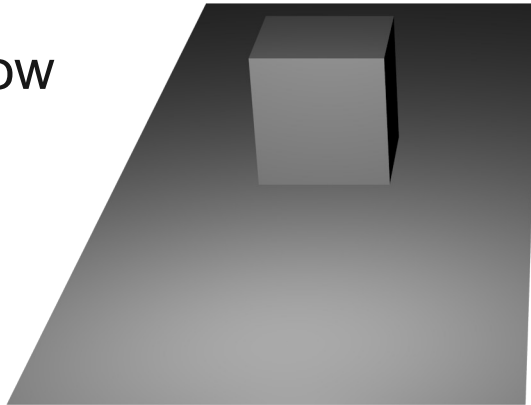
# Shadow Types

- No Shadow – disables shadow calculation for a lamp. All Hemis are No Shadow
- Ray Shadow (all except Hemi) – photons are traced from the light to each object like in real life. Ray shadows have hard edges, except when used with an Area Lamp
- Buffer Shadow (only Spot) – a faster method than ray shadowing that can provide quick, soft shadows



# Shadow Types

No Shadow



Area lamp with ray shadow  
Note the lack of definition,  
which is fairly realistic



Ray Shadow



Buffer Shadow  
Note the  
softer edges



# Lighting Exercise

- We will be lighting the scene we have been working on with Ambient Occlusion and a Sun lamp
- Please load the file “4-Lighting.blend”



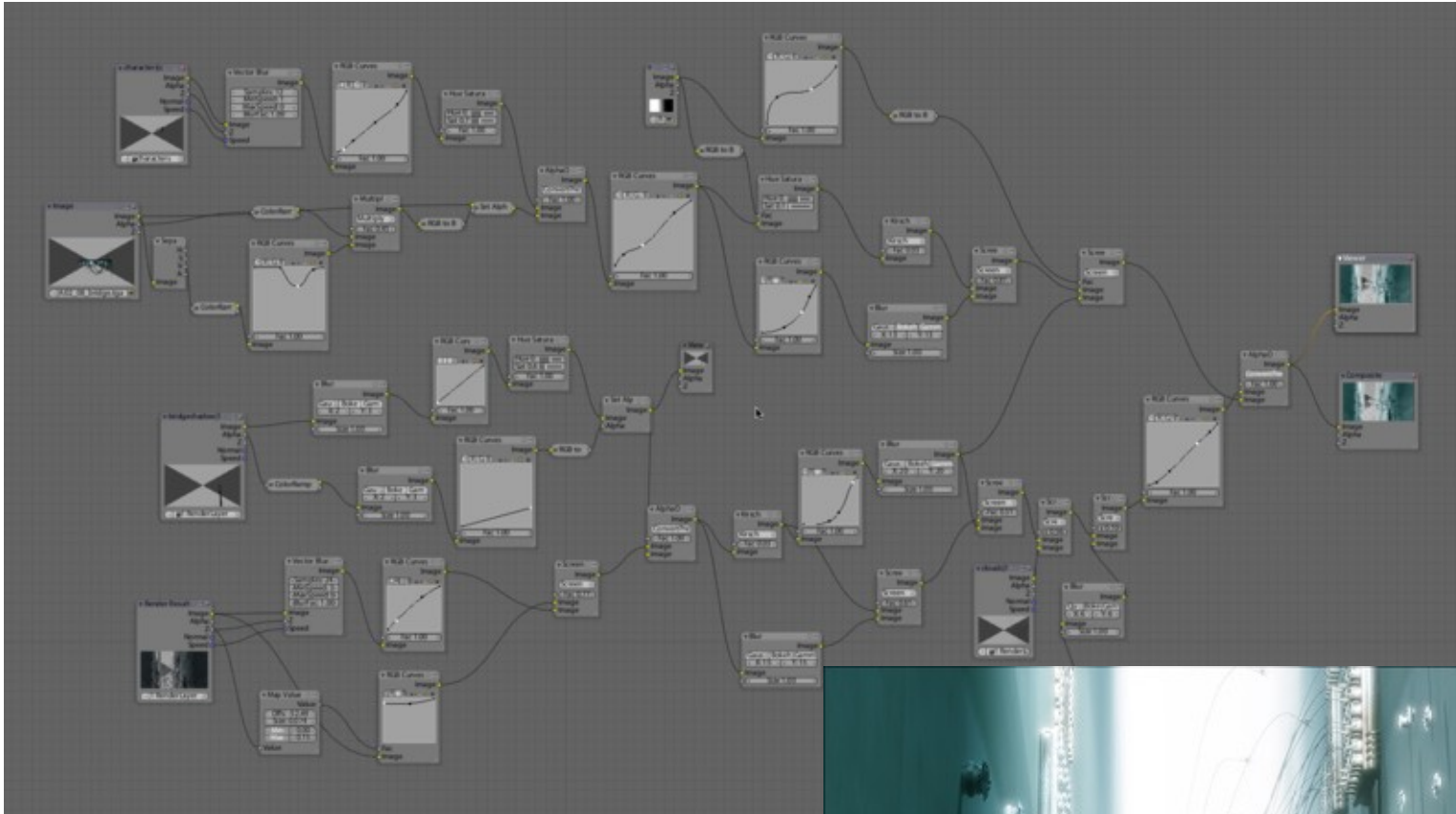


# Lighting Wrap-Up

- Any questions about the exercise?
- Any questions about lighting?



# Rendering and Compositing



# Rendering

- While Blender is cool, it is just a means to an end – the goal is to create 2D stills or movies
- The process of converting 3D scene data to a 2D image is called *rendering*
- Rendering is done from the point of view of the active camera
- The renderer can also apply many effects



# Render Output Options

- The render options are controlled in the Render context
- Here you can control resolution, anti-aliasing, motion blur, frame rate, and output format and location
- You can also control render layers, which are a compositing tool



# Render Engines

- While we are using the default Blender render engine, there are many other render engines
- Yafray is a ray-tracer that provides for more advanced water and glass distortions than Blender
- Luxrender is an unbiased renderer, which means that it continues tracing light rays forever (until you tell it to stop)

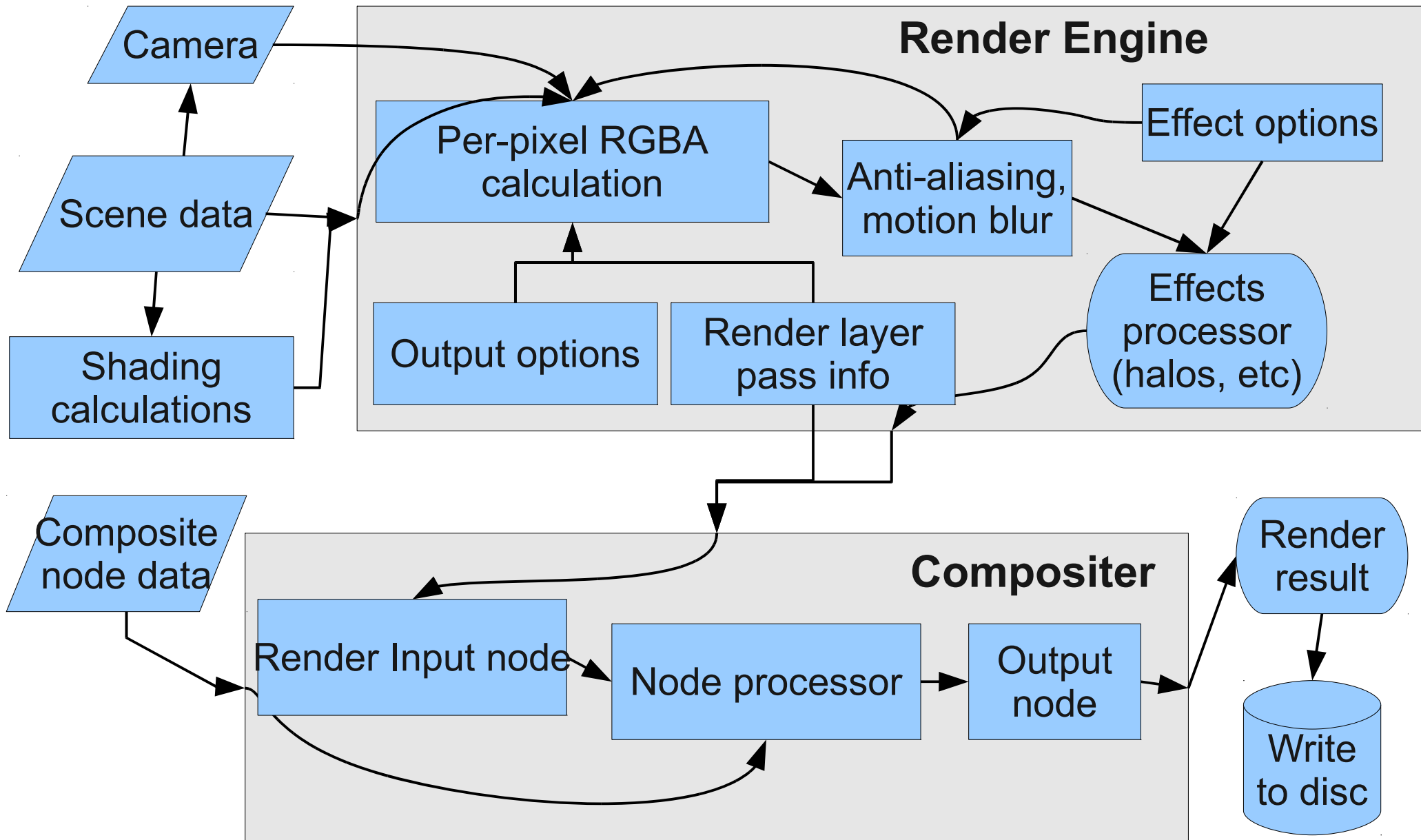


# Compositing

- Blender has a powerful node-based compositing engine that allows for advanced effects like blurs, brightness/contrast tweaks, and green-screening
- The render engine can split output into layers to allow the compositor to apply effects to individual objects



# Render / Composite Still Flowchart



# World Options

- When rendering a scene, there are some preset settings that aren't part of the scene
- These can be found in the properties, world context
- Here you can control effects like sky color, sky gradient, stars, mist, global lighting, and ambient occlusion





# Rendering and Compositing Exercise

- We will be adding a star background to our scene and compositing to add lens distortion
- Please load the file “5-Render.blend”

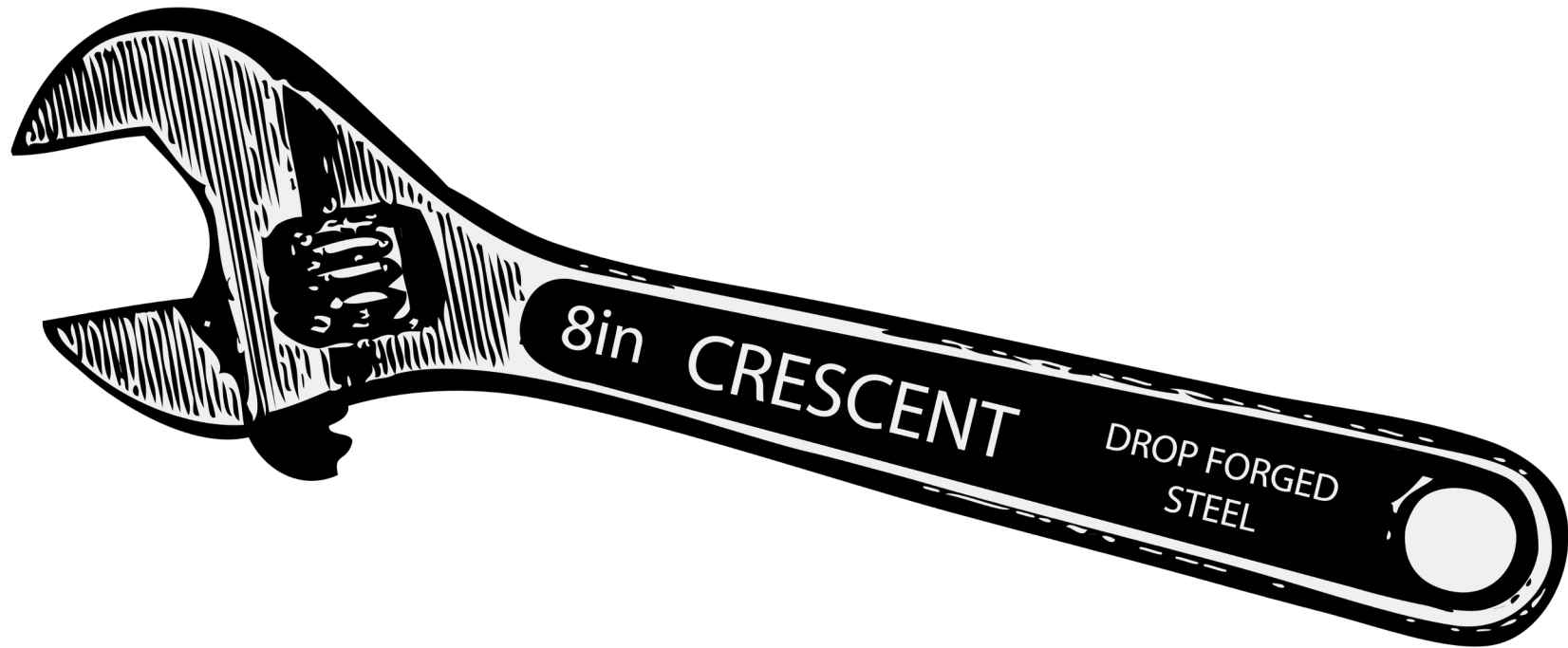


# Rendering and Compositing Wrap-Up

- Any questions about the exercise?
- Any questions about rendering, compositing, or post-processing?



# Miscellaneous Cool Blender Features



# Video Sequencer

- Blender has a built-in video sequencing module, accessible through the Video Sequence Editor window type
- This can be used to cut existing Blender scenes, or with external video clips
- The video sequencer can also edit existing sound clips
- When paired with the compositor, many effects are possible, like green-screening



# Game Engine

- We touched on the game engine in animation, where it can be used for collision physics
- However, the game engine can be used for, namely, games
- Blender assets can be combined with Logic Editor nodes and Python scripts
- Blender has promoted the development of games through the Open Game Project



# Python Scripting

- Blender 2.5 comes with an all-new, improved Python API
- With this API, you can write plugins to automate any Blender function you want
- The Python API can also be used to code logic for games

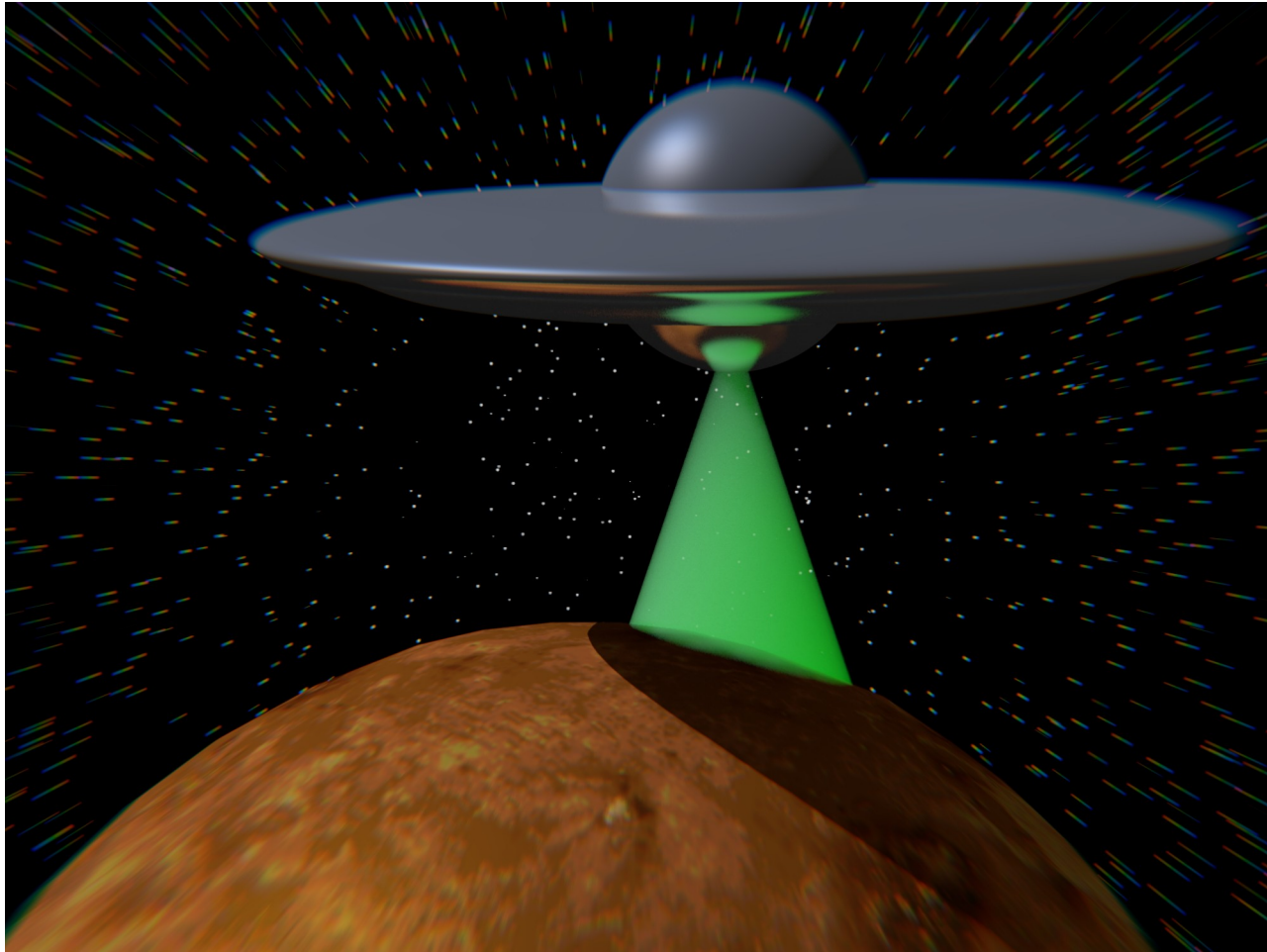


# Plugins

- Written in Python
- Can augment any current Blender function
- Many plugins add new primitives to the Shift+A toolbox, like gemstones or f(x) plotters



# Conclusion





# Furthering Your Blender Education

- IRC - #blender on [irc.freenode.net](http://irc.freenode.net)
- The official forums at <http://blenderartists.org>.  
This is a great place to post Works in Progress for (friendly) critique
- The official manual at <http://wiki.blender.org>



# Contact me!

- If you have any questions, I would be glad to help
- IRC: mmomjian, I'm often on #blender and #blendercoders
- Twitter: @mpmomjian
- E-mail: [matthew@momjian.us](mailto:matthew@momjian.us)



# Thanks for coming!

- I hope you enjoyed the tutorial
- Please rate / comment
- There will be a Blender Birds of a Feather session *tonight* at 8:00p in D137/138. Please come if you want support or would just like to talk about Blender
- Enjoy the rest of the week at OSCON!
- Feel free to talk to me now or anytime this week if you have questions / comments



