The Game Development Process

Game Art Design and Production

Artistic Courses

• **AR 1100. ESSENTIALS OF ART.**
  This course provides an introduction to the basic principles of two and
  three-dimensional visual organization. The course focuses on graphic
  expression, idea development, and visual literacy. Students will be expected
  to master basic rendering skills, perspective drawing, concept art, and
  storyboarding through both traditional and computer-based tools.

• **AR 1101. DIGITAL IMAGING AND COMPUTER ART.**
  This course focuses on the methods, procedures and techniques of creating
  and manipulating images through electronic and digital means. Students will
  develop an understanding of image alteration. Topics may include color
  theory, displays, modeling, shading, and visual perception.

• **AR 3000. THE ART OF ANIMATION.**
  This course examines the fundamentals of computer generated 2D and 3D
  modeling and animation as they apply to creating believable characters and
  environments. Students will learn skeletal animation and traditional
  polygonal animation, giving weight and personality to characters through
  movement, environmental lighting, and changing mood and emotion. Students
  will be expected to master the tools of 3D modeling and skinning, and
  scripting of behaviors.

(Ask: Who’s taken? IMGD-Art majors?)
Introduction

• “The computer artist is modern-day alchemist”
  - (Creating the Art of the Game, by Matthew Omernick)
  - Turn polygons and pixels into wondrous worlds

• Sources of inspiration
  - Playing games! (How can make fun game if not having fun yourself?)
  - The real world (The real world is always more interesting than anything we can make up)

Introduction: Remember the Constraints

• Year 2098, Macrosoft will release FunStation 3000, 14 million terabytes of RAM, quantum-holographic drive with near infinite storage, processors at the speed of light
  - Game developers complain not fast enough

• Game artists must be creative inside confines of technology
  - All disciplines: engineering, design, sound
  - But often constraints biggest on artist
Outline

• The Pipeline
• Concept Art
• 2D Art
  – Animation, Tiles
• 3D Art
  – Modeling, Texturing, Lighting

What's a Pipeline?

• In the pipeline
  – Informal, in the process of being developed, provided, or completed; in the works; under way. (Random House)
• For our purposes
  – The sequence of operations required to move art assets from concept to the finished product
• The Art Pipeline:
  – 2D: Concept, Creation, Conversion
  – 3D: Concept, Creation (Modeling, Texturing, Lighting), Conversion
  – Asset Management
Types of 2D Art

• Created with tools:
  - User Interface (UI)
  - Sprites, Tiles, and other pixel art
  - Type and fonts
• Need a pipeline:
  - Character art
  - Scenery / worlds
  - Characters
  - Animation
  - Video

2D Asset Creation
2D Pipeline – Concept (1 of 3)

• Sketches
  - Napkin-style
  - Detailed design treatments
  - Prototypes
2D UI Prototype
designersnotebook.com

Paper UI Prototype
boxesandarrows.com
2D Pipeline – Creation (2 of 3)

- Commercial / third party tools:
  - Photoshop, The Gimp, sprite editors, HTML/browsers, Flash...
- Homegrown tools
  - Specialized animation systems
  - Tools that simulate key game features (UI layout tool, etc)
  - The game engine

Assets for a 2D animation (1 of 3)

eberlein.org/euphoria
Assets for a 2D animation (2 of 3)

Assets for a 2D animation (3 of 3)

cvrpg.com

aniway.com
2D Pipeline - Conversion (3 of 3)

- Putting the assets into the final form
- File type conversion
  - PSD to TGA / JPG, for example
  - Compression
  - Collection (zip files, pak files, etc)
- Testing in the game
- Debug / fix

3D Asset Creation
3D Pipeline – Concept (1 of 4)

- Sketches
  - Napkin-style
  - Detailed design treatments
  - Prototypes
  - Maquettes
  - Animation sketches / flipbooks
  - Mockup models
  - Texture mockups
  - Architectural layout

3D UI Prototype

lostgarden.com
3D Pipeline - Creation (2 of 4)

• Commercial / third party tools:
  - Photoshop, The Gimp, sprite editors, HTML/browsers, Flash...
  - 3D tools: 3D Studio Max, Maya, Lightwave, Blender

• Homegrown tools
  - Specialized animation systems
  - Tools that simulate key game features (UI layout tool, etc)
  - The game engine
  - Exporters / plugins
Stages of a Model

3D Pipeline (3 of 4)

- Animation systems
  - Motion capture
  - Third party tools
  - Homebuilt tools
- Texturing systems
- Shaders / surface tools
- Renderers / video systems
Texturing and Accessories

garagegames.com

A Model, Textured

zbrush
Character and a Skin (1 of 2)

secretlair.com

Character and a Skin (2 of 2)

cresswells.com
3D Pipeline – Conversion (4 of 4)

• Conversion
  - Export from modeling to custom formats
  - Putting the assets into the final form
    * File type conversion
      - PSD to TGA / JPG, for example
      - Compression
      - Collection (zip files, pak files, etc)
    * Testing in the game
    * Debug / fix

Asset Management

• How do you share the production process across time, space, and content creators?
  - Source code has many tools – "solved"
  - Data/Art is harder
    * Not easily merged
    * Dependencies not obvious
    * Relationships complex
  - Some commercial systems are trying
  - Typically a combination of:
    * Homegrown tools
    * Convention and process
Outline

• The Pipeline
• Concept Art (next)
• 2D Art
  - Animation, Tiles
• 3D Art
  - Modeling, Texturing, Lighting

Why Not Just Prototype?

• Even creating prototypes can be time consuming and expensive
• Getting it right on the first try is unlikely
• Revising instantiated work can be difficult

Thus the need for some forethought!
What is a Better Way?

Make decisions on paper, where changes and variations can be made quickly and easily.

What is Concept Drawing? (1 of 2)

• From illustration, but is a modern idea
• Main goal to convey visual representation of a design, idea, and/or mood
• Use in movies, comic books and computer games
• *Before* it is put into the final product (or even prototype)
What is Concept Drawing? (2 of 2)

It is not a full design, blueprint or specification.
It is a partial design that gives enough detail to imply a full design.

Who Is Involved?

We can think of the process as a two-part system:

The BOSS supplies the constraints that the drawing needs to fill.

The ARTIST generates drawings based on the given constraints.
The BOSS (1 of 3)

- Takes part in a higher-level design process with goals of its own

  - Attract web-goers
  - Sell more widgets
  - Win an Oscar


The BOSS (2 of 3)

- Communicates constraints inherited from this higher-level process to the ARTIST

  - We need a happy, purple dinosaur to sell more widgets!
The BOSS (3 of 3)

• Evaluates the fitness of the ARTIST’s solutions based on various heuristics

  Market research says it’s good
  Wife likes it
  Magic 8-Ball says outlook not so good

The ARTIST (1 of 4)

• Generates drawings based on the given constraints

  We need a happy, purple dinosaur to sell more widgets!
The ARTIST (2 of 4)

- Has expert knowledge of drawing materials and techniques

The ARTIST (3 of 4)

- Has aesthetic heuristics (acquired by studying style, design and master artwork)
The ARTIST (4 of 4)

- May need to cache domain-specific knowledge (dinosaur anatomy, typical dress of a noblewoman in 13th century England, etc.)

A Space of Ideas (1 of 3)

There is a space of drawings that potentially satisfy some set of constraints (could be from cache)

(Space of Evil Lizard-Monsters)
A Space of Ideas (2 of 3)

The ARTIST is capable of generating drawings that represent regions of this space.

(Remember that because a concept drawing is not a full design, there is some range of designs that each drawing represents)

A Space of Ideas (3 of 3)

The BOSS provides the ARTIST with direction in searching this space for a solution that optimizes BOSS's heuristics.

My wife says it needs more armor!

“Concepting” is like a hill-climbing search of the idea space!
Beginning with Thumbnails (1 of 3)

The exploration of possible solutions often begins with the ARTIST generating rough drawings.

These rough drawings - called "thumbnails" - are often little more than suggestive scribbles.

Thumbnails can be generated very rapidly. This allows the ARTIST to generate many points of the search space at little cost.

Beginning with Thumbnails (2 of 3)

But where does the artist come up with all these samples of the search space?

The ARTIST forms these partial solutions from domain experience and/or cached reference images!
Beginning with Thumbnails (3 of 3)

I like the one with the sunglasses. Let’s see where we can go with that!

Thumbnails enable the ARTIST and BOSS to quickly get their bearings and identify fruitful starting-points for exploration.

These rough drawings can quickly expose problems with the given constraints.

It doesn’t take highly detailed concepts to cull large portions of the search space!

Drawing the Concept (1 of 5)

There are many techniques for creating more detailed concepts (pencils, markers, watercolors, etc.)

In general, the drawing process is one of iterative refinement.

For example, when rendering a concept with markers, the ARTIST will begin by roughly sketching most of the detail with a light marker.
Next, the ARTIST does some line selection.

In this stage, the ARTIST is choosing the best of the rough details for inclusion in the final concept.

The ARTIST may also be adding some lower-level details as well, now that the higher-level details are becoming more specific.

---

Next, the ARTIST does some region coloring to separate the different elements of the drawing...
... and now a shading pass to reveal the 3D form of the concept ...

... and finally a detail pass, picking out lines to reinforce and areas to highlight.

This concept is finished ... but wait!
Back to Hill-Climbing!

Very nice! But the Magic 8-Ball doesn’t like all those spots; and the VCs think it should have big, muscley arms!

Questions and Discussion
Outline

• The Pipeline
• Concept Art
• 2D Art (next)
  – Animation, Tiles
• 3D Art
  – Modeling, Texturing, Lighting

2D Animation

• Animation → produces the illusion of movement
• Display a series of frames with small differences between them
• Done in rapid succession, eye blends to get motion
• Unit is Frames Per Second (fps)
  – 24-30 fps: full-motion (Game Maker does 30)
  – 15 fps: full-motion approximation
  – 7 fps: choppy
  – 3 fps: very choppy
  – Less than 3 fps: slide show
• To do successfully, need to keenly observe, focus on differences in movement
  – Apply basic principles (next)
Key Frames

- Images at extremes in movement
  - Most noticeable to observer
  - Ex: for flight wings up and wings down
  - Ex: for walking, right leg forward, legs together, left leg forward
- The more the better?
  - Smoother, yes
  - But more time to develop (tradeoffs)
  - And more prone to errors, “bugs” that interfere with the animation

In-Between Frames

- Generated to get smooth motion between key-frames
  - Can be tedious and time consuming to make
  - Most software allows duplication
  - Some does interpolation for you (Game Maker, Flash)
Frame Animation Guidelines

<table>
<thead>
<tr>
<th>Object</th>
<th>Minimum # of Frames</th>
<th>Maximum #</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-legged animal running</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>Animal biting</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Crawling</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>Explosions</td>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td>Falling</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Flying</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>Jumping</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>Kicking</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>Punching</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>Rotating/spinning</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>Running</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>Swinging (on object)</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>Throwing (on object)</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>Vehicle flying</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Vehicle moving</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>Walking</td>
<td>2</td>
<td>12</td>
</tr>
</tbody>
</table>

(See GameMaker tutorial shooter for examples of Enemy Planes, Explosions)

Motion Line

• Invisible line created by object as moves
  - Locate in center of gravity
• Straight if flying
  - Ex: bullet
• Up and down if bounces
  - Ex: rubber ball
• Depends upon speed and desire for exaggeration
  - Ex: Human sprinting versus walking
  - Ex: Warcraft III
Secondary Actions

- Animation part that does not lead movement, but follows it
  - Add extra dimension of reality
  - Ex: Hair moving in wind
  - Ex: Cape billowing backward

Steps in Creating Animation Sequences (1 of 3)

- Conceptualize - have vision (in mind or on paper) of what animation will look like
- Decide on object behavior
  1. Animated once (no looping)
  2. Animated continuously (using cycles)
    - (Ask) ... what is the difference?
    - 2nd choice means must make last key frame blend with first
- Choose a grid size - will contain and constrain object
  - Test and experiment briefly to have plenty of room
- Design key-frames - drawing the motion extremes
  - Start with simple shapes to represent main actions
    - Ex: stick figures or basic shapes (circles, squares)
Steps in Creating Animation Sequences (2 of 3)

• Estimate the in-betweens - think of how many you will need to complete the sequence smoothly
  - Be conservative. Easier to add additional transition frames than remove them
• Create object motion lines - trace the motion line and motion angles for the sequence. Make sure properties are consistent with object, else adjust
  - Use your painting program’s "line tool"
  - If not, make the appropriate adjustments to the sequence and repeat
• Apply secondary enhancements - Embellish to look convincing and enticing

Steps in Creating Animation Sequences (3 of 3)

• Test each movement
  - May have animation rendering (ex- Game Maker)
  - Can be done with 'copy' and 'undo' in tool
  - Look for flaws (movement, discolored pixels ...)
• Repeat - Do the same for all animations
General Animation Tips (1 of 3)

- Remember the relationship between frames and animation smoothness
  - More frames, more smoothness (but more time)
- Always account for color
  - Primary actions and secondary actions should be rendered in colors that make them easy to see.
  - Otherwise, the effectiveness of the animation can be compromised (ch 7 and ch 8)
- Use tempo wisely - Never too fast or too slow
  - Try to mimic nature. Observe yourself. Study the speed at which different types of objects move in different situations.

General Animation Tips (2 of 3)

- Try to individualize your objects
  - Unique and individualized touches make seem real. “Personality” that distinguishes it
  - Easiest may be to use exaggeration and embellishment (i.e., secondary actions)
- Keep it simple - Unnecessary complexity can ruin animation
  - Stick with primitives and minimal frames
  - Don’t do any more work than you have to!
General Animation Tips (3 of 3)

• Use exaggerated elements - as an animation device, adds depth
  - Especially important for short animation sequences to make convincing

• Learn from others - study how objects around you move, books on animation, observe your favorite games, look at sprites in Game Maker
  - Will give insights into animation techniques, make better animations yourself
  → Make use of Primitives (next)

Primitives

• Used in many games. If identify, can apply primitive rules and use:
  - Cylindrical primitive
  - Rotational primitive
  - Disintegration primitive
  - Color flash primitive
  - Scissors primitive
  - Growing primitive
  - Shrinking primitive
  - Minor primitives (used less often)
  - (See Chapter 9 of Feldman)
Tiles

- Needed for common backgrounds
  - Too hard to make every pixel different!
- Exploration games (especially outdoors) make heavy use
  - Grass, trees, water, sand
- Start with a grass tile to warm up

Grass is Green

- Use a basic green square
- But looks unnatural
  - Like flat, shiny metal
- No illusion of movement

← Simply a Green Box
(Ex: bkg_grass0)
Grass has Variation

• Can do a lot with simple enhancement of color shades

SIMPLY A GREEN BOX WITH LIGHTER GREEN DOTS.

(Ex: bkg_grass1)

Make Variation More Random

• Can use the “spray” tool

3 GREEN SHADES RANDOMLY "SPRAYED" AND A 16X16 CHUNK TAKEN FROM IT.

(Ex: bkg_grass2)
Make Look Random but with Control

• Draw by hand for more control
  - 4 pixel line strokes

(Building up "strands" from the 3 shades of green to get a "direction" flow.

(Ex: bkg_grass3)

The "Grid" (1 of 3)

• Looks too much like tiles
• "Large" blank is problem, so remove
The “Grid” (2 of 3)

- Still, some “lines” are visible when repeated
- Break up with more color

The “Grid” (3 of 3)

- Much better!

(Ex: bkg_grass4)
Don’t Try This at Home

• Don’t use the same texture for all, else not much better than just colors

| OKAY, I HAVE MY GRASS TILE... |
| AND NOW IT’S A DIRT TILE! |
| AND WATER! HA HA HA! |
| AND DESERT! I’M A GENIUS! |

Don’t Try This at Home

• When the rubber hits the road?
Outline

- The Pipeline
- Concept Art
- 2D Art
  - Animation, Tiles
- 3D Art (next)
  - Modeling, Texturing, Lighting

3D Art - Sub-Outline

- Preparing to Create
- Modeling Theory
  - Example
- Texturing
- Lighting
Preparing to Create

• Using reference essential
  - Difference between mediocre and exceptional game
  - Gives you goals, direction, clues, motivation
• Ex: portrait of friend
  - Could: sit down, imagine friend, draw
  - Or, could: use photo and draw
  - Latter will include details didn’t think of
  - Same holds for buildings, cars, etc.
• Reference is not “cheating”
  - Yeah, many want to create directly in minds, but
    using the right reference a skill in itself!

How and Where to Collect Reference

• Search ’net – Google image search (show demo)
  - Ex: sofa, couch, lazy-boy, lounge chair
  - 2 minutes can provide a lot of details
• Books
  - Ex: on submarines for U-boat
• Movies
  - Ex: U-571
• Physical location
  - Ex: visit U-boat tour, tour country/climate of game.
    Even fantasy world has trees, etc.
**Concept Art**

- Pre-visualize art for communication, color, inspiration
  - Establish “look and feel”, like storyboarding for film
  - Saves time and money since iterate before rendering
- Even if company has concept artists, digital artists should still do their own
  - Remember, computer is just another tool
  - Figure drawing helps understand shape and line
    * And often required for portfolio!

---

**Blocking Out Your Scene (1 of 3)**

- **Mantra: "Broad strokes, then add detail"**
  - Ex: painting mountain scene.
    * Start with blue sky, define brown mountains, lake. Finer brush for trees, clouds. Finer for rocks, birds in sky …
    * Start in corner. Paint all details and move over. You’d go crazy! Would be skewed.
  - Ex: animating a character.
    * Set two keyframes, point A and point B. Get speed right, basic idea. Add frames for up and down. Then legs and arms swing …
    * When done, smooth walk
Blocking Out Your Scene (2 of 3)

- For scene, use simple primitives to define scale and layout
- Ex: create village.
  - Working with designer, create plane (crudely drawn map) of layout
    - Scan and import into 3D tool (say, Maya)
  - Import 5'11” character (just shape)
    - Use to decide how tall building or how wide door
  - Add objects in right scale
  - Quickly → basic, functional scene of right size and scale. Broad stroke number 1!

Tip: get artists & designers to agree upon measurement units & heights of characters

Based on Chapter 1, Creating the Art of the Game, by Matthew Omernick

Blocking Out Your Scene (3 of 3)

- If game engine working, can export into game and run around
  - Often designer will do this, anyway, but artist should have input
- Can throw in some lighting (later) and colors (later)
- Add a few textures (not final ones, but canned that show right feel)
  - Can even add text saying “brick”

Based on Chapter 1, Creating the Art of the Game, by Matthew Omernick
3D Art - Sub-Outline

• Preparing to Create
• Modeling Theory (next)
  - Example
• Texturing
• Lighting

Intro to Modeling Theory

• Understand core philosophy of 3d modeling for games
• Want to do it fast and efficiently
  - Allow "broad strokes" in model early
  - More time (and polygon resources) for refinement later
• If quick, but sloppy, end up with stray vertices, overlapping faces ...
  - Wasted resources
  - Plus bugs! For collision detection
• Modeling Types (talk about each a bit, next)
  - NURBS
  - Subdivision Surfaces
  - Polygon (is king in game development)

Based on Chapter 6.2, Introduction to Game Development
Non-Uniform Rational Basis Spline (NURBS)

• Uses curved surfaces based on a few points
• Strengths:
  – Great for cut-scenes
  – Resolution independent
  – Inherent mapping coordinates
• Weaknesses:
  – More difficult to learn
  – Difficult transitioning between high and low density areas
  – Seams are complicated to overcome
  – Not supported by many game engines

Subdivision Surfaces

• Strengths
  – Has polygonal ease of editing (can manipulate points) with NURBS smoothness
  – Very efficient way to work
  – Great for cut-scenes or as the basis for high resolution normal map source models
• Weaknesses
  – Almost no game engines support this geometry type
Polygonal

• **Strengths:**
  - Very straightforward, easy to troubleshoot, easy to modify
  - Supported by all 3D game engines

• **Weaknesses:**
  - A technical process
    - Constantly manipulating topology
  - Faceting
    - Rough around the edges
  - Fixed Resolution
    - Unless level of detail models are created

• Polygons preferred since most used (talked about rest of section)
  - By polygons, we mean triangles
  - Face may have triangles that share vertices (Ex: square down middle)
  - Software may hide shared edge for cleaner look

Based on Chapter 6.2, Introduction to Game Development

---

Polygonal Modeling Basics: Primitives

• **Primitives** are basic shapes
• **Most 3d packages have same primitives:**
  - Sphere, Cube, Cylinder, Plane
  - Use for “broad strokes”
• **Concentrate on primitives within object**
  - Ex: human body (ovals for shoulders, cylinders for legs, sphere for head...)
• **Components** are parts that make up primitive
  - Ex: vertices, edges, triangles, faces, elements
  - Similar across all packages but terminology can vary
• **Transformation** allows moving, rotating, scaling object or component
Polygonal Modeling Basics:
Normals

- Face normals are at right angle to polygon
  - Tell what direction if facing, how to render, how light will react
- Viewed from other side, is invisible
  - Fine if on inside (say, of solid cube)
- When debugging, pay attention to normals as well as polygons

Polygonal Modeling Basics:
Backface Culling

- Toggles display of faces that point away from view
  - When on, see through wireframe
  - When off, looks solid (not drawn)
- Makes look less cluttered
Polygons and Limits

- 3d Software renders scene of triangles like game
  - But 3d software slow (Toy Story 1 frame / 15 hrs)
  - Game is real time (30 frames / second)
- Need to limit polygons. How spent depends upon world size and where needed.
  - Ex: Medal of Honor versus Soul Caliber 2. MH details spread across world, less on avatars. SC can have detailed avatars since only 2 in one ring.
- Think of how many polygons each item needs. Estimates, educated guesses. Then, make pass. (Tools will often give count)
  - Used wisely, can make detailed scenes with few (Ex: 2.5, page 24)
  - Ch 6.2 assumes 4000 (typical for PS2 street fighting game or hero in third-person action game)

Based on Chapter 2, Creating the Art of the Game, by Matthew Omernick

Polygon Reduction

- Being able to model without wasting polygons important → takes practice
- Ask if a player will see face?
  - Ex: oil barrel as cylinder. Will see bottom? Nope, then delete.
- Are all faces necessary? Looks great, yeah, but some can be removed.
  - Ex: 12-sided cylinder still looks “round” with 8 sides? Then do it.
- (Example exercise p30-31)

Based on Chapter 2, Creating the Art of the Game, by Matthew Omernick
Modeling Tools

- Certain tools and techniques used 80-90% of the time
  - (Bottom 3 used for next example)

  - **Line Tool**:
    - Draw outline of object and extrude to get 3-d shape
      - Ex: profile of car. Use line tool. Then, extrude outward to get shape. "Broad stroke"
      - Some risk in may have vertices and faces you don't need, but careful planning and practice helps
  
  - **Extrude**:
    - Take component (often face), duplicating it, pulling pushing or scaling to refine model
      - Ex: take cube. Extrude face outward and smaller
      - Ex: take cube. Extrude part of face to make window
  
  - **Cut**:
    - Subdivides faces and adds new faces
  
  - **Adjust**:
    - The artistic part of modeling. Try to capture form, profile and character by moving vertices
      - "Vertex surgery", part of the technical manipulation

---

3D Art - Sub-Outline

- Preparing to Create
- Modeling Theory
  - Example (next)
- Texturing
- Lighting
Box Modeling: Reference

- Decide on polygon limits
- Posed and turnaround sketches of a character
  - Can often be imported into 3d tool

Box Modeling: Start With A Box

- Begin with a box
- Cut it in half
- Approximate the torso shape
- Cut it in half (will do half well, then mirror)
Box Modeling: Extrude The Torso And Neck

- Extrude the box several times
  - 3 times for the top, 2x for the bottom
- Adjust to simulate a rough torso (with bulge)
- Do the same for the neck

Box Modeling: Extrude The Head

- Extrude from the neck
  - First to eye level, then to top of head
- Extrude the head
  - Adds volume to the head
- Edit into aroughed out head
  - Cuts above eye line for brow and under for nose

Based on Chapter 6.2, Introduction to Game Development
Box Modeling: Create Rough Arms

- Here, only one of two arms
- Extrude the upper side of the torso for the shoulder area
- Extrude several times for the arm
- Manipulate into rough arm shapes
  - Bend at elbow

Box Modeling: Create Rough Hands

- Extrude a few times for basic hand volume
  - 3, in this example
- Cut and extrude the thumb volume
- Note: refer to own hands for proportions

Based on Chapter 6.2, Introduction to Game Development
Box Modeling: Time For Legs

- Again, create only one of the legs
- Extrude and edit
- Extrude feet forward from stump

Box Modeling: Mirror

- Delete half of the model
- Mirror the other side
- Attach and weld the seam

Next up, refining the model!
Box Modeling: Proportions Match

• Bring the turnaround sketches into the viewport on a textured plane
• Manipulate until the model matches up
  – Important things: head right size, extremity lengths, eye level.

Box Modeling: Begin Adding Detail

• Square legs and shoes are especially prominent
  – Will look blocky in game engine
• Add a few more segments to support more curvature for the legs
Box Modeling: Cuffs

- Cut faces into feet to get curvature, adjust as necessary
  - Constantly compare to sketches
- A simple extrude to create the cuffs of the pants

Box Modeling: Gimme A Hand (1 of 2)

- Often the most difficult
  - Use own hand constantly for proportions
    - Slightly curved, so natural, middle finger higher
- Cut where fingers begin
- Extrude outward for 4 fingers
- Fingers will need joints if animated
Box Modeling: Gimme A Hand (2 of 2)

- Same buildup, but from underneath

Box Modeling: More Arm Please

- Cut in more detail for the arm, and manipulate for better form and curvature
Box Modeling: The Torso

- Cut in to support relevant detail
- Additional polygons at shoulder to support deformation

Based on Chapter 6.2, Introduction to Game Development

Box Modeling: The Back

- Add polygons for the back, and the bevel of the shirt

Based on Chapter 6.2, Introduction to Game Development
Box Modeling: The Face, Head And Hair

- Phases
  - Major structures: brow, eye, cheekbone, mouth nose hairline
  - Extrude volume for hair
  - Adjust bottom for extruding ponytails
  - Cut in polygons around eyes, mouth, nose
  - Once done, add some asymmetry (part off center)

Box Modeling: Done

- The completed model in wireframe and shaded

Images courtesy of WildTangent, modeled by David Johnson.

Based on Chapter 6.2, Introduction to Game Development
Box Modeling: Summary

• Done for character, but can apply to other things
• General idea:
  - Start with box, cylinder or other primitive
  - Extrude, Cut, Adjust...
  - Get topology, proportions right
  - Once happy, refine until details complete

Other Modeling Techniques: 3D Sculpting

• A low resolution model can be sculpted into a very detailed mesh
• This can be used in game via normal maps
  - (Calculate lighting on each pixel, gives illusion of more polygons of fidelity)

Images courtesy of Pixologic.

Based on Chapter 6.2, Introduction to Game Development
Other Modeling Techniques:
Reverse Engineering

- Real world objects or sculptures can be scanned or digitized
- This may not save time because of complicated polygon cleanup, but will ensure high fidelity

Image courtesy of FARO Technologies, Inc.

Based on Chapter 6.2, Introduction to Game Development

Other Modeling Techniques:
BSP

- BSP stands for *Binary Space Partition*
- A coding term that is also method for organizing data
- Like cutting away a mineshaft
  - Start inside solid room
  - Cut away chunks with primitives
- Satisfying since can make space quickly
- BSP Editors come with many games like *Quake, Unreal and Half-Life*

Based on Chapter 6.2, Introduction to Game Development
Low Poly Modeling (1 of 3)

• Again, too many polygons results in lower frame rates
• To keep frame rates consistent, use level-of-detail (LOD) meshes
  - Multiple versions of object, progressively lower levels
• When far away, use low level
  - Assume more objects in Field of View
• When close, use higher level
  - Assume fewer objects in Field of View

Based on Chapter 6.2, Introduction to Game Development

Low Poly Modeling (2 of 3)

• For entire level (ie- map with environment), entire polygon count matters
  - Impacts amount of memory needed
• But only visible polygons rendered
  - Rest are “culled” and not computed

Images courtesy of WildTangent

Based on Chapter 6.2, Introduction to Game Development
Low Poly Modeling (3 of 3)

• With low polygon modeling, much of the detail is painted into the texture (next topic!)

Images courtesy of WildTangent, model and texture by David Johnson.

3D Art - Sub-Outline

• Preparing to Create
• Modeling Theory
  - Example
• Texturing (next)
• Lighting
Texturing

• **Motivation**
  - Games rely heavily for realism
  - Important to compensate for low geometry
  - Challenging, yet rewarding

• **Distinction between texture and shader**
  - *Shader* - define surface property of object
    - how shiny, bumpy, how light effects
  - *Texture* - bitmap plugged into shader that defines image we want to appear on object

Based on Chapter 6.4, Introduction to Game Development

Detail in Texture

• **Add depth, lines, etc. without polygons**
• **Box is 12 polygons, bricks would take many more**

(Taken from http://www.mostert.org/3d/3dpdscenem.html)

Based on Chapter 6.4, Introduction to Game Development
Breaking Down Mesh Object for Mapping

- Evaluate the 3D model for common areas
  - Simplifies the work
  - Saving valuable texture space
  - Reduce the amount of texture borders
    (Example on Book CD-ROM for 3DS max .. case study)

A Brief Word on Alpha Channels

- The embedded extra 8 bits of 32-bit image
  - 24 bits gives true color, \(2^{24} \sim 16\) million colors)
- Use for:
  - Transparency
  - Reflection
  - Bump maps
**Alpha Channel - Transparency**

- Used to create transparency
- White means opaque, black means transparent, grey are values of opacity (semi-transparency)

![Image of tree with alpha channel applied](image1.png)

Images courtesy of WildTangent.

Based on Chapter 6.5, Introduction to Game Development

---

**Alpha Channel - Reflection**

- Define what areas reflect light most - human face shiny where oil, water ripples
- Three common types of reflection
  - Camera projected - always the same, but can be unrealistic
  - Cubemap - 6 sides, but predefined
  - Dynamic - sides computed on the fly

![Image of reflective surfaces](image2.png)

Images courtesy of WildTangent.

Based on Chapter 6.5, Introduction to Game Development
**Alpha Channel - Bump Map**

- Use to create illusion of varying heights
- Light is protrusion, dark is recession
- Tweaks each pixel based on grayscale value

---

**Alpha Channel - Normal map**

- A variant of bump mapping
- Uses color images (RGB) instead of grayscale
  - Still tweak each pixel

Images courtesy of WildTangent.
3D Art - Sub-Outline

• Preparing to Create
• Modeling Theory
  - Example
• Texturing
• Lighting (next)

Lighting

• Can conjure feelings, emotions, even change what you are seeing
  - Reveal (or hide) depth
  - (Many books on traditional lighting)
  - AR/ID 3150. LIGHT, VISION AND UNDERSTANDING
• Remember, when see things is really reflection of light
• Sub-outline
  - Color
  - Mood
  - Setup
  - 3-D lights

Based on Chapter 6.6, Introduction to Game Development
Color

- Powerful in setting mood
- Typical ok,
  - Green is ok, Red is danger
  - But feel free to move beyond cliché
- Culture specific
  - Sure, Red danger, but in China Red happy
  - White purity, but in China White death
- Powerful associations
  - Ex: The Matrix
    * Green is in Matrix
    * Blue is in real-world
- Balance
  - Too many and chaotic, over-stimulation
  - Too little and drab and boring
  - (Color theory classes can help)

Based on Chapter 6.6, Introduction to Game Development

Color Indicates Danger

RTX Red Rock

Pleasing Colors

Star Wars: Knights of the Old Republic


Mood

- Intensity, direction, angle, number of lights, and shadows all affect mood
- Even humidity, dust, air quality

Based on Chapter 6.6, Introduction to Game Development
Mood by Lighting Example (1 of 3)

A calming outdoor scene using simple, yet effective, lighting.


Mood by Lighting Example (2 of 3)

Long shadows not only add to the atmosphere, but also help break up repetition.

Mood by Lighting Example (3 of 3)

Light beams and rays give clues as to the humidity, dust, and air quality in a scene


Lighting Setup (1 of 3)

• Traditional lighting
  - Key light, Fill light, Back light
• Key light – main light source. Most intense and majority. Put at angle to define 3-D forms.
Lighting Setup Example

A sphere lit only by a key light positioned at an angle. The detail and form of the sphere are not as clear as if we added another light source.


The Key light is placed next to the camera, about 35-45 degree angle to the subject. The angle is determined by what kind of mood that you want the scene to have.

http://www.3dtotal.com/team/Tutorials/Jenns3pt_tut/3plighting.asp
Lighting Setup (2 of 3)

- Fill light - Brings out some details out of shadow. Works well at angle.

Lighting Setup Example

A fill light brings out more form, and softens the shadows from a fill light. Notice the point light has been added to the left of the sphere.

Place the Fill Light at a 90 degree angle from the Key Light, usually slightly higher or lower than the Key Light.

Lighting Setup Example

http://www.3dtotal.com/team/Tutorials/Jenns3pt_tut/3ptlighting.asp

Lighting Setup (3 of 3)

• **Backlight** - Placed behind and slightly above or below object to help define shape. Highlights edges, pulls away from background.
  - (Also called the Rim Light or Hair Light)

Based on Chapter 6.6, Introduction to Game Development
Lighting Setup Example

The addition of the third light highlights the edge, helping give the sphere more dimension.


Lighting Setup Example

The Back Light is placed directly opposite the camera and behind the subject.

http://www.3dtotal.com/team/Tutorials/Jenns/3pt_tut/3ptlighting.asp
Working with 3D lights (1 of 3)

- 3-D lighting different than traditional lighting
  - Start with traditional and modify until you get desired affect (broad strokes)
- Tools give different kinds of lights
  - (next)
- A few effective practices
  - (after)

Based on Chapter 6.6, Introduction to Game Development

Working with 3D lights (2 of 3)

- Directional Lights - used for sunlight or moonlight. Often as key light. Predictable.

By the time the sun’s rays reach the earth, they are nearly parallel to one another.

Based on Chapter 6.6, Introduction to Game Development
Working with 3D lights (3 of 3)

• Ambient Lights - spread everywhere, equally. Uniform diffuse lights.
  - Can skip by creative placement of the Fill Light, but gives more precise control over illumination
• Spot Lights - focus beam on single location. Great control.
• Point Lights - single point in all directions. Light bulbs, candles, etc.
• Background Light - soften the areas of the background that Key Light doesn't illuminate

Example of Working with 3D lights

A room lit without radiosity. Bottom The same room with a radiosity solution.
Effective Lighting Practices (1 of 3)

**Pools of light**
- Don't always try to light evenly.
- Gives sense of mystery

Pools of light in Indiana Jones: The Emperor’s Tomb

Based on Chapter 6.6, Introduction to Game Development

Effective Lighting Practices (2 of 3)

**Guide lights**
- Use light to guide the player.
- Helps highlight areas that are accessible and important to the objectives.

Based on Chapter 6.6, Introduction to Game Development
Effective Lighting Practices (3 of 3)

• Be Creative
  – Try not to stick to the standard solutions
  – Tell a story with your lights
  – Talk to level designer about scenes, even
  – Ex: Maybe your level harder than last, convey that tension

• Experiment
  – Start simple, add detail.
  – Experiment at early stages.
  – Try crazy combinations of color, reverse the intensities, or reposition lights in unorthodox places.

Lighting Summary

• Study real-world light carefully to understand 3D light
  – 3D is at best only an approximation

• Study different conditions - rain, sunny, indoor, outdoor....

• Study lights from photos

• The key to developing skills as lighting artist → observe and re-create what you see
Bit Bucket

• The rest of the topics are to be covered on students own time
• Or, possibly in class, as time allows

Make Interesting Textures

• Consider story behind object
• Consider door (contoured, so could do geometry, but cheaper to put picture up)
• Could just take one on Internet and put up
• But can make more believable
  - How old? Who uses it?
    Repainted? How long ago?
• Add grunge around knob, show nicks at bottom, flecks of color where repainted …

Based on Chapter 3, Creating the Art of the Game, by Matthew Omernick
Textures are Their Own Artwork

• Rarely ready to go ... spend time in Photoshop massaging, customizing
• Think of each texture as custom artwork
• Before and after page 49
  - Wood → with coffee mug stain, nicks and scratches
  - Window → depth in reflections, uneven opacity
  - Concrete → cracks, discoloration
• Need to be aware if tiled and reused
  - Interesting textures harder to re-use since noticeable

Resolution

• Analogy:
  - Smiley face with 15 rocks
    • Hard to make out
  - Smiley face with 30 rocks
    • Looks Better
  - Smiley face with sand
    • Looks great
• So, always use high resolution for textures? Not necessarily. Takes more video memory.
**Where To Use Pixels?**

- Think about
  - Physical size - actual size of object relative to character
  - Distance - how far away and how close can character get to it
- Consider: room with box, window, clock
  - Each has a different resolution texture applied to it
  - Box not much (on floor and can't crawl) 128x128
  - Wall more since big (but still uninteresting) 512x512
  - Clock small and high, but numbers so 64x64
  - Window has picture of lighthouse but far so 32x32

Based on Chapter 3, *Creating the Art of the Game*, by Matthew Omernick

---

**Color Depth**

- How many bits to use to color each pixel
  - Ex: 16 colors (4-bit) lot less memory than 65,536 colors (16-bit)
  - Recommendation, try low and see if holds
- Sometimes low-bit gives “washed out” look that can be desired
- In fact, T.V. and real-world have lower color depth than most computer monitors
  - Try for yourself
  - Vibrant on computer may not be realistic
  - Worse, if port to T.V. reds bleed together

Based on Chapter 3, *Creating the Art of the Game*, by Matthew Omernick
Sprites

- Graphic objects that can move separately from background
- Often animated
- Topics:
  - Grid Squares
  - Primitives

Grid Squares

- “Mini-Screen” to depict
  - Individual pixel modifications
- Help observe animation progression
- (Show Game Maker image editor example)
- Strips for tools

http://www.flyingyogi.com/fun/download.cgi?spritelib