



---

# IMGD 1001: The Game Art Pipeline

by

**Mark Claypool** (claypool@cs.wpi.edu)

**Robert W. Lindeman** (gogo@wpi.edu)

---



## 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.

## 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 art pipeline
- Concept art
- 2D Art
  - Animation
  - Tiles
- 3D Art
  - Modeling
  - Texturing
  - Lighting

## What's a Pipeline?

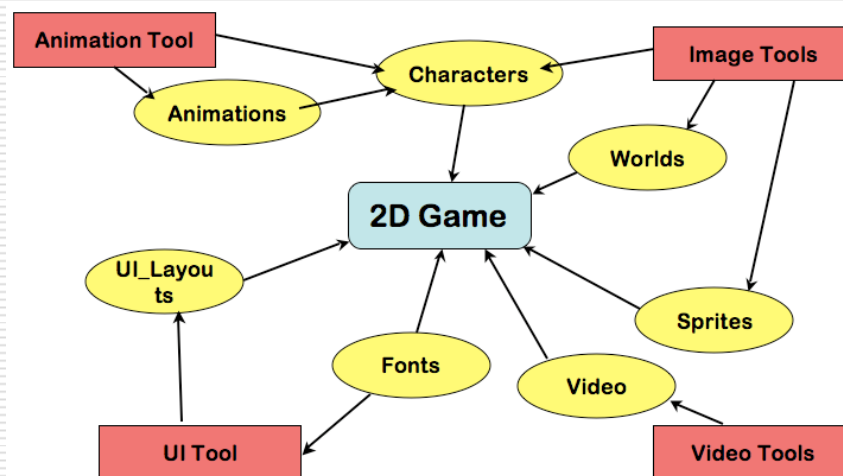
---

- *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

- These are created with tools:
  - User Interface (UI)
  - Sprites, tiles, and other pixel art
  - Type and fonts
  
- These need a pipeline:
  - Character art
  - Scenery / worlds
  - Characters
  - Animation
  - Video

## 2D Asset Creation



## 2D Pipeline (1 of 3): Concept

---

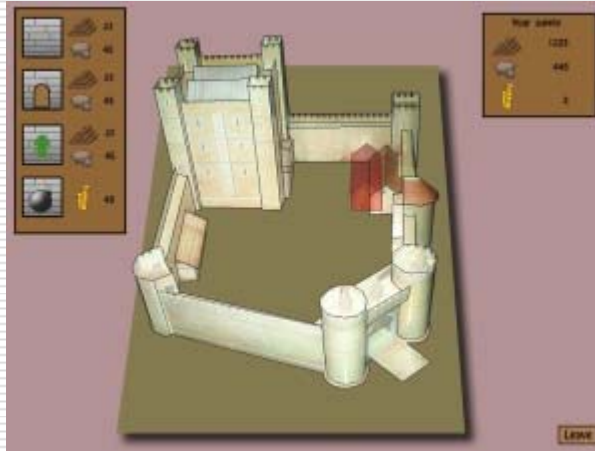
- Sketches
  - Napkin-style
  - Detailed design treatments
  - Prototypes

## "Napkin-style" Concept Art

---

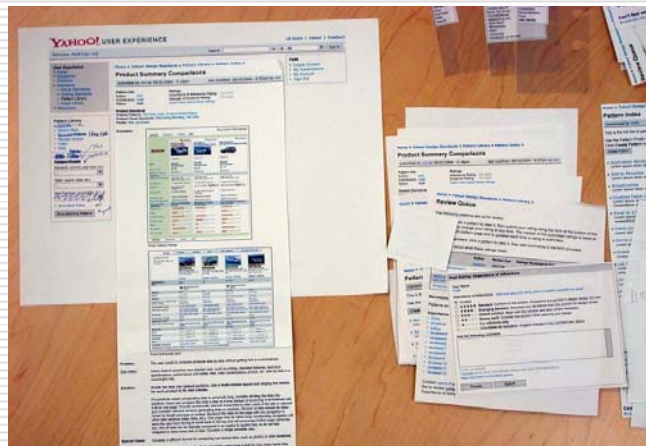


## 2D UI Prototype



[designersnotebook.com](http://designersnotebook.com)

## Paper UI Prototype



[boxesandarrows.com](http://boxesandarrows.com)

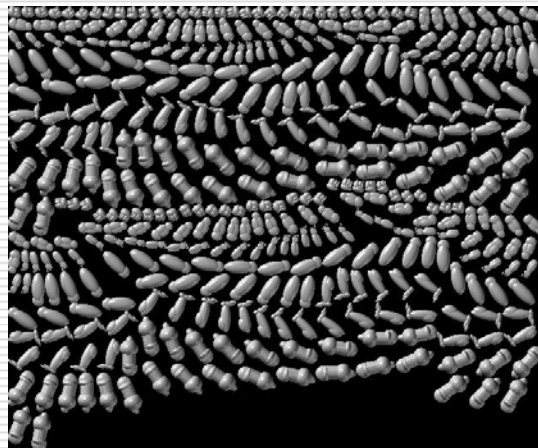
## 2D Pipeline (2 of 3): Creation

---

- 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 2D Animation (1 of 3)

---



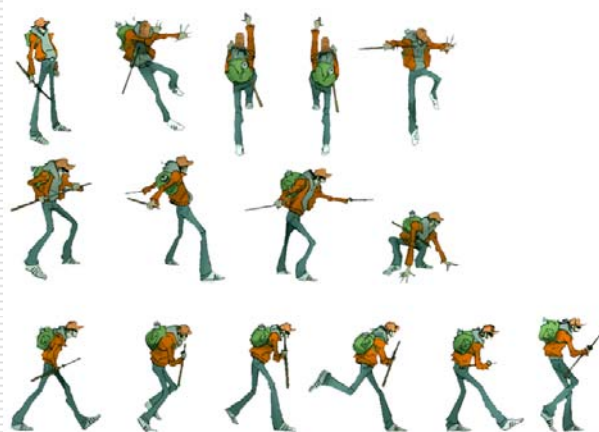
[eberlein.org/euphoria](http://eberlein.org/euphoria)

# Assets for 2D Animation (2 of 3)



cvrpg.com

# Assets for 2D Animation (3 of 3)



aniway.com



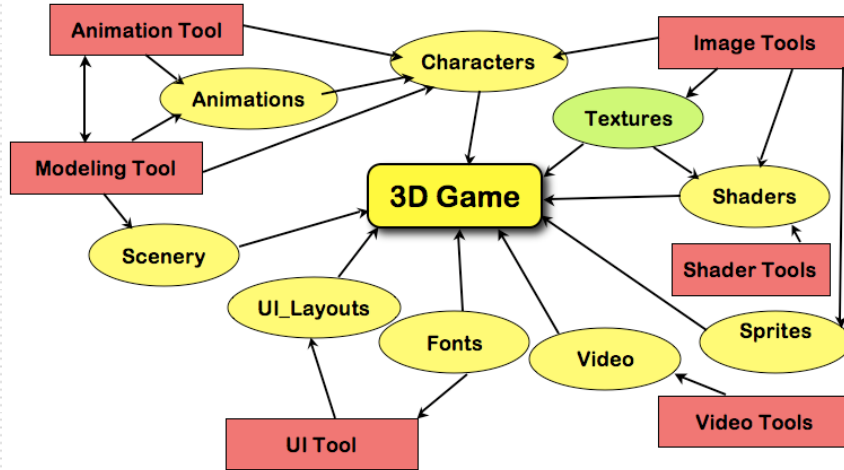
## 2D Pipeline (3 of 3): Conversion

- 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

## Group Exercise

- Break into your project groups
- At the top of a piece of paper, *each person* writes the description (text) of a visual concept for a character/item/object in your game (3-4 minutes)
  - About 1 small paragraph
- Rotate papers among group
- In the middle of the paper, draw a concept art sketch based on the text (5 minutes)
- Pass back to the original person
- Provide feedback (verbally), and briefly in text on bottom (2-3 minutes)
- (Hand in)

## 3D Asset Creation

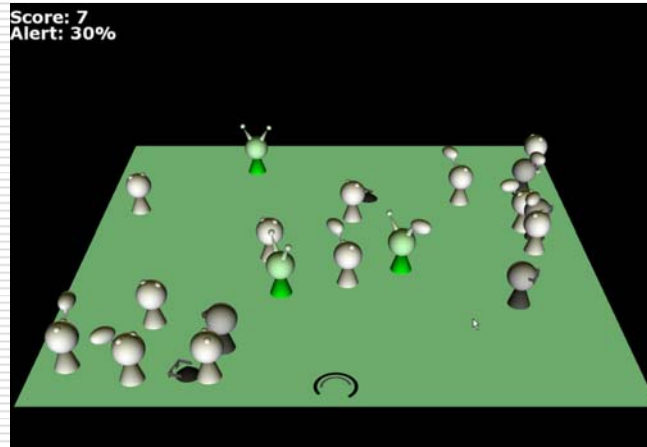


## 3D Pipeline (1 of 4): Concept

### □ Sketches

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

## 3D UI Prototype



lostgarden.com

## 3D Pipeline (2 of 4): Creation

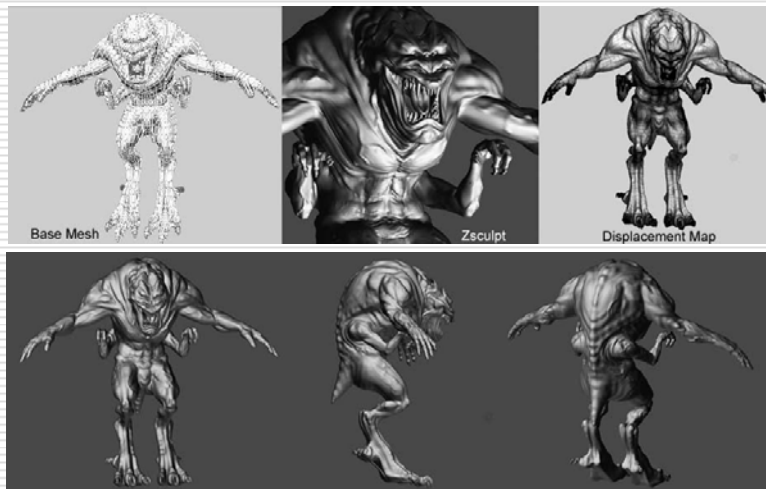
- ❑ Commercial / third party tools:
  - *Photoshop, The Gimp, sprite editors, HTML/browsers, Flash...*
  - 3D tools: 3D Studio Max, Maya, Lightwave, Blender
- ❑ Homegrown tools
  - Specialized animation systems
  - Tools that simulate key game features (UI layout tool, etc.)
  - The game engine
  - Exporters / plugins

# 3D Studio Max



gamedev.net

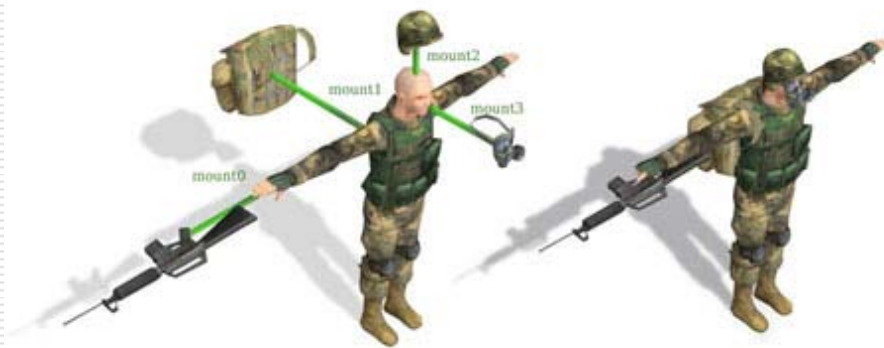
# Stages of a Model



## 3D Pipeline (3 of 4): Texturing

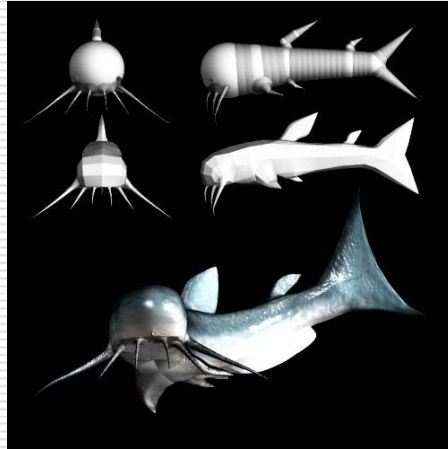
- Animation systems
  - Motion capture
  - Third party tools
  - Homebuilt tools
- Texturing systems
- Shaders / surface tools
- Renderers / video systems

## Texturing and Accessories



[garagegames.com](http://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 (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