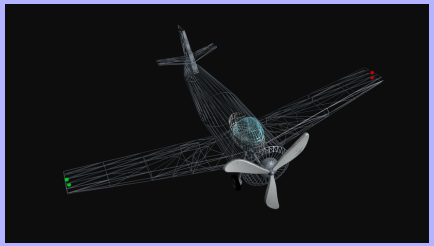
Meshes

Motivation

- We want to display more complex things than simple geometric shapes!
- Tedious to manually put vertex data into program
- So we need to be able to load meshes from external files

Information

- What goes into a mesh?
  - Vertices (2D and 3D)
  - Triangles (how vertices connect) (2D and 3D)
  - Texture coordinates (u,v values) (2D and 3D)
  - Normals (3D only)
  - Material properties (ex: Texture filename(s)) (2D and 3D)



Meshes

- Create: Many programs (Blender, Maya, 3d Studio, ...)
- Import data: Choices:
  - Native file format (.blend, .mb, ...)
    - Too complex!
  - OBJ file format
    - ASCII: Easy to read (human)
    - Larger file
    - Slower to process

ASCII

- ASCII: File is humanly readable
- To indicate number 42: character 4, then character 2
  - Character '4' has ASCII value 52
  - Character '2' has ASCII value 50
  - So we have:
    00110100 00110010
- Slower to do I/O: System must translate to/from native machine format

Binary

- Binary: Raw format used by CPU
- To indicate number 42: Store as 8, 16, or 32 bit integer
  - Ex: Value 42 as 8 bit integer:
    00101010
  - Ex: Value 42 as 16 bit integer:
    00000000 00101010
- Faster to do I/O: System can just pull raw value in from disk

OBJ

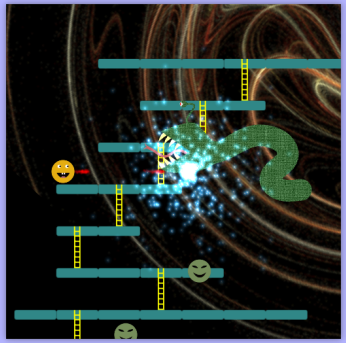
- OBJ file format: Series of lines; All ASCII
  - Each line has *keyword* telling meaning of line, then data
  - Comments indicated with # character
- Important lines:
  - o *name*
  - v x y z
  - vn x y z
  - vt x y z
  - f vspec1 vspec2 vspec3 ... vspecn
    - vi
    - vi/ti
    - vi//ni
    - vi/ti/ni
    - Note: Indices begin at ONE, not zero!
  - mtllib fname
  - usemtl name
- For now, we ignore other lines

Material

- MTL file format:
  - newmtl name
  - map_Kd textureFile
- We ignore the other lines

Assignment

- When the player comes to the end of the level, bring a boss onto the screen to fight the user. The boss should be a textured mesh. (Either find a freely licensed mesh (cite your source!) or make your own or use this cheesy mesh).
  - I wouldn't spend much time on meshes for *this* lab – Our goal is to use 3D meshes, so any 2D meshes are just temporary placeholders
- Retain all other existing functionality



Sources

- D. Brackeen. *Game Programming in Java*. New Riders Media.
- F. Luna. *Introduction to Game Programming with DirectX 10*. Wordware.
- F. Luna. *Introduction to Game Programming with DirectX 9.0*. Wordware.
- E. Lengyel. *Mathematics for 3D Game Programming and Computer Graphics*. Charles River Media.
- D. Hearn & M. P. Baker. *Computer Graphics in C*. Prentice Hall.
- Nebula image source: NASA and NSSDCA: NGC 3372: The Carina Nebula. Image Credit: NASA, The Hubble Heritage Team (AURA/STScI) https://nssdc.gsfc.nasa.gov/photo_gallery/ photogallery-astro-nebula.html https://nssdc.gsfc.nasa.gov/photo_gallery/caption/ hst_carina_ngc3372_0006.txt

Created using LaTeX.

Main font: Gentium Book Basic, by Victor Gaultney. See http://software.sil.org/gentium/
Monospace font: Source Code Pro, by Paul D. Hunt. See https://fonts.google.com/specimen/Source+Code+Pro and http://sourceforge.net/adobe
Icons by Ulisse Perusin, Steven Garrity, Lapo Calamandrei, Ryan Collier, Rodney Dawes, Andreas Nilsson, Tuomas Kuosmanen, Garrett LeSage, and Jakub Steiner. See http://tango-project.org