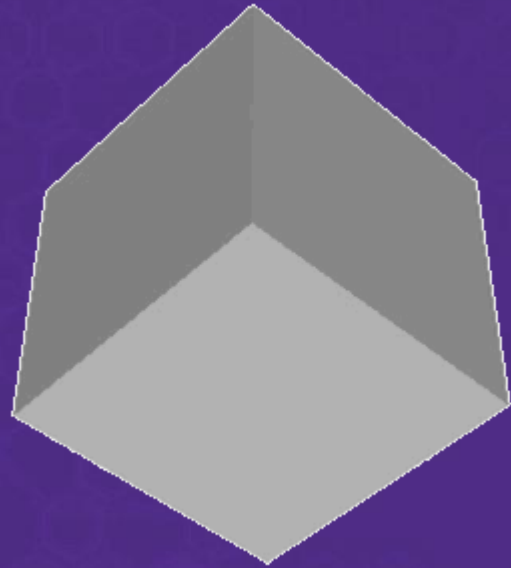


Developing Graphics Frameworks with Java and OpenGL



Day 30: Textures

Textures

- Textures: images applied to surfaces of 3D shapes
- Texture Object: data structure that stores:
 - pixel data from image
 - types of magnification/minification to use when surface and image are different sizes
- Image data stored in Texture Buffers

OpenGL Texture functions

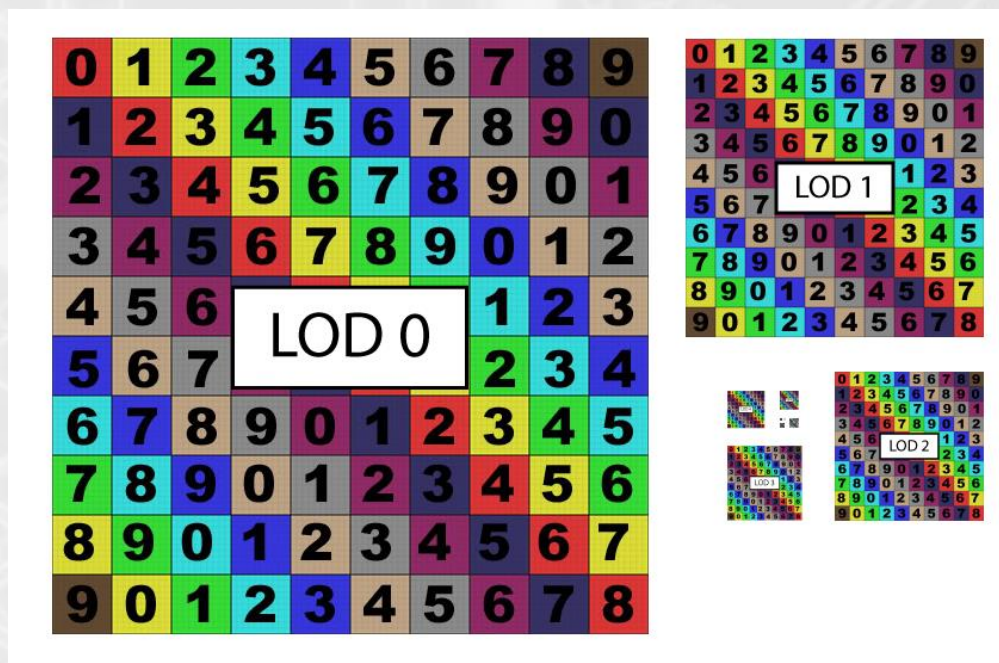
- **glGenTexture(*textureCount*)**
Returns a reference to a texture buffer
- **glBindTexture(*bindTarget, textureRef*)**
Binds a texture reference; future OpenGL commands involving textures will target the bound texture.

OpenGL Texture functions

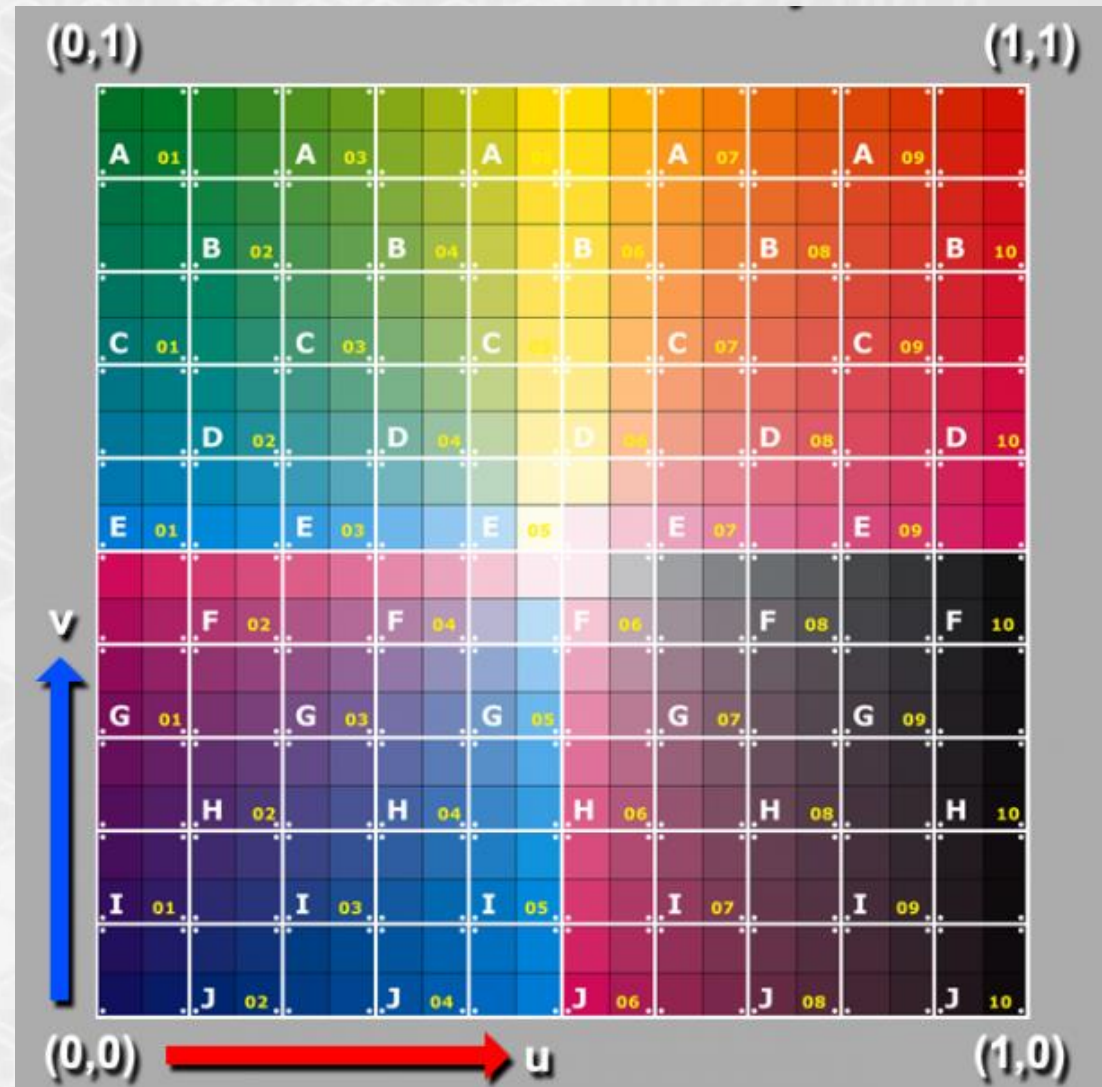
- **glTexImage2D**(*bindTarget, level, internalFormat, width, height, border, format, type, pixelData*)
 - Uploads texture data to GPU.
 - internalFormat, format: usually GL_RGBA
 - type (precision): usually GL_UNSIGNED_BYTE (8 bits)

Mipmapping

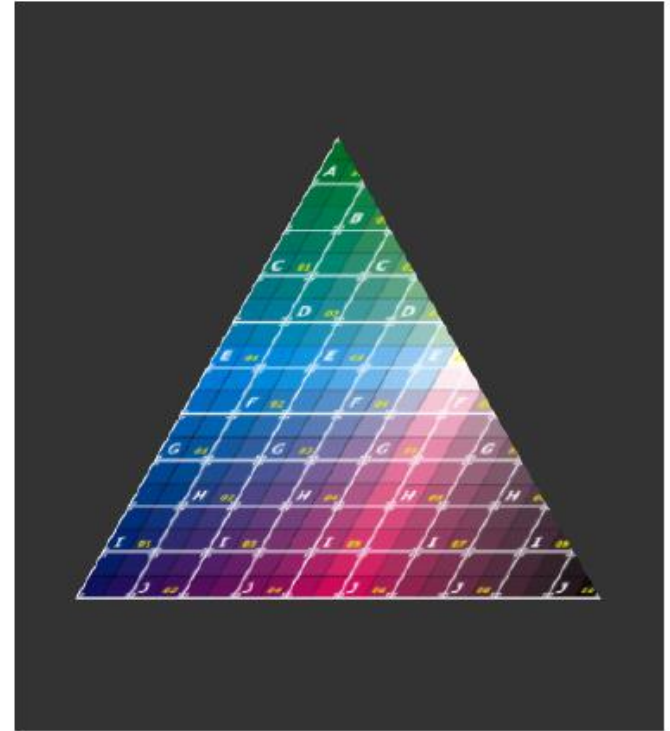
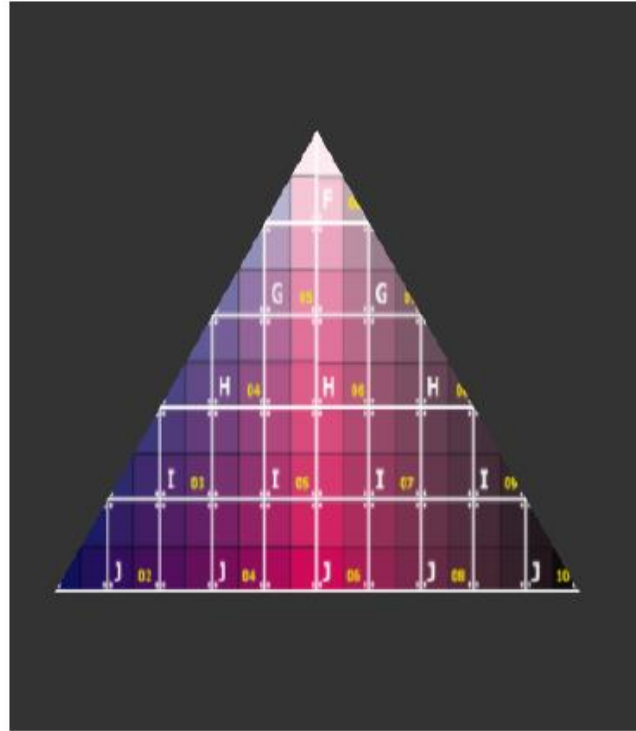
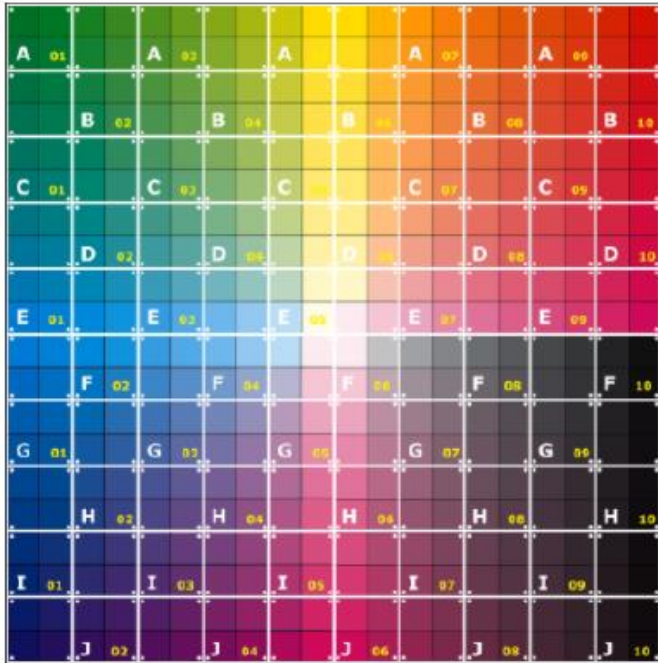
- “Zooming out” can be a long operation;
requires averaging many pixels to determine final color
- Mipmap: sequence of pre-calculated image reductions
 - LOD: Level of Detail
- OpenGL command:
glGenerateMipmap



Texture coordinates



Texture coordinates



Using Textures in Shaders

- *Sampling*: the process of calculating a color based on data stored in a texture object
- Texture objects are assigned to a *Texture Unit*, which performs the sampling calculations
- Shader programs use uniform variables of type **sampler2D** to store texture unit references
- **texture(sampler2DRef, uvCoord)**
Returns **vec4** containing color data at given point