OpenGL “Mount Evans”

- **Assumes “modern” GPU for full support**
  - GPUs that only support OpenGL 2.x feature set might not fully support “Mount Evans”
- **Adds new features relative to OpenGL 2.x using new OpenGL 3 API**
  - Geometry Shaders
  - Integer Pipeline
  - New uses for buffer object storage:
    - transformed vertex data, textures, & shader uniforms
  - Texture Arrays
  - New formats:
    - Floating point depth renderbuffers and textures
    - 1 & 2 component compressed textures
    - sRGB renderbuffers
    - Shared exponent textures
    - Packed floating point renderbuffers and textures
  - Instanced rendering
  - Per-DrawBuffer blending controls

“Mount Evans” - Geometry Shaders

- Adds a new GLSL programmable stage to the pipeline
- After vertex shader, before fragment shader
  - before clamping, flat shading, clipping
- Can generate new vertices and/or change the primitive type
  - i.e., points can become triangles
- Shader has access to all vertices of a primitive
- Can get access to adjacent vertices if supplied by application
- Inspired on existing specification:
  - http://www.opengl.org/registry/specs/EXT/geometry_shader4.txt

“Mount Evans” - Integer Pipeline

- **Adds Integer shader data types and operations**
  - attributes, varyings, temporaries, samplers
  - integer vector data types
  - bitwise operators
- **Adds Integer internal texture formats**
  - 16bit, 32bit
- **Adds Integer render formats**
  - drawables, textures, renderbuffers
- **Removes “normalization” stage from various parts of pipeline**
- **Inspired by existing specifications:**
  - http://www.opengl.org/registry/specs/EXT/texture_integer.txt
  - http://www.opengl.org/registry/specs/EXT/gpu_shader4.txt

“Mount Evans” - New Buffer Objects Uses

- **Textures**
  - Texel data (for certain targets) can be stored in buffer objects
  - Useful for large 1D lookup tables
- **Transformed Vertex Data**
  - Buffer objects can be used to capture output of vertex/geometry transformation
  - Useful for capturing “non-image” data processed by GPU
- **Shader Uniforms**
  - Shader uniforms can be stored in buffer objects
  - Useful for rapidly switching between sets of shader uniforms
- **Inspired by existing specifications:**
  - http://www.opengl.org/registry/specs/EXT/texture_buffer_object.txt
  - http://www.opengl.org/registry/specs/EXT/bindable_uniform.txt
  - http://www.opengl.org/registry/specs/NV/transform_feedback.txt

“Mount Evans” - Texture Arrays

- **Adds new texture targets to represent arrays of images of same size within one texture object**
  - Useful for efficiently switching rendering input/output between multiple images
- **Supports arrays of 1D and 2D images**
- **Individual images addressable by one texture coordinate: [0, layer-1]**
- **Can attach layers to FBO attachment points**
- **Some limits**
  - No filtering between layers
  - Mipmaps are per-layer
- **Inspired by existing specification:**
  - http://www.opengl.org/registry/specs/EXT/texture_array.txt

Status

- **Participating companies**
  - Apple, AMD, NVIDIA, Intel, S3, Blizzard + individual contributors (Jon Leech et al)
- **Group meetings roughly weekly**
  - expected to pick up as Long's Peak settles
- **Mount Evans is layered on top of Long's Peak**
  - Expected specification(s): approximately 3-5 months after Long's Peak finalized
- **For more info, contact:**
  - OpenGL ARB "nextgen" TSG chair
    - Jeremy Sandmel, **jsandmel@apple.com**