Real Time Hair Preview for
Monsters University using
NVIDIA GPUs

David G Yu
Sullivan and Art

- Hair is an essential part of their Personality and Appearance
- Not the same without hair
Sullivan and Art

- Hair is an essential part of their Personality and Appearance
- Not the same without hair
Working in Context

- Framing is different without hair
Working in Context

- Framing is different without hair
Sullivan in Presto

- Pixar’s Presto Animation System
- NVIDIA GeForce GTX TITAN GPU
- GPU Compute
- Modern OpenGL
Hair for One Face

- Camera Facing Triangle Strips
- Generated with a Geometry Shader
- Four Control Vertices per Hair
- Width per Control Vertex
- Clamping, Blending, Randomized Draw Order
- ~1M hairs on Sullivan (~10K faces)
Hair Geometry

- Character’s body mesh is the scalp
- Root control vertex CV0 is on the scalp
- Located using barycentric coords
- CV1, CV2, CV3 tangent space offsets
- Connected with Linear Triangle Strip Segments
Posing Hair

- Load Colors, Widths, Tangent Space CVs, scalp face indices into GPU memory once
- Update Scalp Vertices (character mesh) when the character’s pose changes
- Run a GPU kernel to compute hair Root CVs to update hair for the new character pose
- All hair data remains on the GPU

```glsl
in vec3 hairInput;
uniform isamplerBuffer scalpFaces;
uniform samplerBuffer scalpVertices;

out vec4 hairVertex;
out vec4 hairNormal;
out vec4 hairTangent;

void main()
{
    int faceIndex = int(hairInput.x);
    vec2 faceUV = hairInput.yz;
    ivec3 scalpFace = texelFetch(scalfFaces, faceIndex).xyz;
    int v0 = scalpFace[0];
    vec4 P0 = texelFetch(scalfVertices, v0 * 2);
    vec4 N0 = texelFetch(scalfVertices, v0 * 2 + 1);
    int v1 = scalpFace[1];
    vec4 P1 = texelFetch(scalfVertices, v1 * 2);
    vec4 N1 = texelFetch(scalfVertices, v1 * 2 + 1);
    int v2 = scalpFace[2];
    vec4 P2 = texelFetch(scalfVertices, v2 * 2);
    vec4 N2 = texelFetch(scalfVertices, v2 * 2 + 1);

    hairVertex = P0 * faceUV[0] + P1 * faceUV[1] + P2 * (1.0 - faceUV[0] - faceUV[1]);
    hairNormal = N0 * faceUV[0] + N1 * faceUV[1] + N2 * (1.0 - faceUV[0] - faceUV[1]);
    hairTangent = P1 - P0;
}
```
Interaction

- Animation rig moves the character's body mesh
- Associate subsets of faces on the body mesh with specific animation controls
- Highlight the associated faces when interacting with an animation direct manipulation widget
Thanks!

- Rob Russ - Animation Lead
- Jacob Brooks and Laurence Emms - Character TDs
- Dirk Van Gelder, Evan Wallace, David G Yu - Software R & D