Rendering Wounds in Left 4 Dead 2

Alex Vlachos, Valve
March 9, 2010
Outline

• Goals

• Technical Constraints

• Initial Prototype

• Final Solution
Left 4 Dead 1 Wounds

- Built-in
- 5 variations only
- Requires texture support
- Always Fatal
The Pitch

Gray Horsfield lives for destruction

(Gray is a Visual Effects Artist at Valve, previously at Weta)
Goals

• Accurate location of wounds

• Wounds match weapon strength
  – Remove limbs, torso, head, half of body

• Separate wound geometry & textures

• Several active/visible wounds per model
  – Shipped up to 2 active wounds
Technical Constraints

• Already at memory limits on the Xbox 360

• Didn’t want heavy CPU setup

• Ideally wanted a GPU solution

• No additional base meshes except for wound geometry
  – Better for artists to author
  – Share wound models among many infected
Common Infected Variation

• Simplest infected has over 24,000 variations

• We didn’t want to add another variable to this
Things We Didn’t/Couldn’t Do

• Model variations of each infected with all combinations of 1 and 2 wounds

• Use different index buffers to cull polygons – not friendly with LOD and low quality wound silhouettes

• Auto-generate new polygonal meshes with holes cut for wound models

• Author different body parts/sections with different wound variations
Initial Prototype

- Use pose-space ellipsoids to cull pixels
- Fill hole with wound model
Culling Inside an Ellipsoid

- Vertex Shader calculates relative distance
- Interpolate this value and clip / texkill
Benefits

• No additional vertex buffer data
• Still only one draw call for full model
• Wounds are a separate draw call with their own textures:
Problems

• Hard cut looked unnatural

• Wound models looked strange because they required a lip around the wound border

• Lacked blood on the clothes and skin near the border of the wound

• Required an exact geometric fit with the model
Projected Texture Experiment

Try using a projected texture and use alpha to kill pixels
Abdominal Wounds

- Projected texture will affect his back
- So let’s combine the texture and ellipsoid
Blood Layer

- The texture projection is aligned with an axis of the ellipse
- We multiply the blood layer by a gradient to prevent the blood from spraying too far
// Subtract off ellipsoid center
float3 vLocalPosition = ( vPreSkinnedPosition.xyz - vEllipsoidCenter.xyz );

// Apply rotation and ellipsoid scale. Ellipsoid basis is the orthonormal basis
// of the ellipsoid divided by the per-axis ellipsoid size.
float3 vEllipsoidPosition;
vEllipsoidPosition.x = dot( vEllipsoidSide.xyz, vLocalPosition.xyz );
vEllipsoidPosition.y = dot( vEllipsoidUp.xyz, vLocalPosition.xyz );
vEllipsoidPosition.z = dot( vEllipsoidForward.xyz, vLocalPosition.xyz );

// Use the length of the position in ellipsoid space as input to texkill/clip
float fTexkillInput = length( vEllipsoidPosition.xyz );

// We use the xy of the position in ellipsoid space as the texture uv
float2 vTextureCoords = vEllipsoidPosition.xy;
Other

• Depth-only and shadow render passes
  – You don’t want phantom shadows

• Hi-Z performance issues

• Wound models are attached to base skeleton of infected model
Multiple Wounds

We limited the final solution to 2 active wounds
Upper & Lower Back
Groin
Arms & Legs
Abdomen
Head Wounds
Axe & Sword Slashes
Upper Body
Stats

• Up to 54 unique wounds per model

• Each wound is only 13% of the memory cost of the old system in Left 4 Dead 1

• Vertex shader costs 15 instructions
  – Fill-bound, so rendering perf impacted minimally

• Pixel Shader costs 7 instructions
Summary

• Wound models separate from base mesh

• Use pose-space ellipsoids for outer limiting cull volume

• Use projected texture for rough edges and blood layer

• Additional details about our rendering: http://www.valvesoftware.com/publications.html
Thank you!

Alex Vlachos, Valve
alex@valvesoftware.com