Shading a Bigger, Better Sequel
Techniques in Left 4 Dead 2

BRONWEN GRIMES, VALVE

Left 4 Dead 2

- ❖ Onion AV Club's Best Games of 2009 Pick
- ❖ Gamasutra's Best Of 2009: Top 10 Games Of The Year Pick
- ❖ PC Gamer's Shooter of the Year 2009
- ❖ IGN's Best Multiplayer Game 2009
- ❖ Gamespot's Best Cooperative Multiplayer Game 2009
- ❖ GamerVision's Best Ever Multiplayer 2009
- ❖ Gamereactor's Co-Op Game of the Year and Online Game of the Year
- ❖ Planet Xbox 360's Best Co-Op Experience of 2009
- ❖ Spike TV's Video Game Awards 2009 Best Xbox 360 Game
- ❖ Ironhammers' Game of the Year

Same Platform, More Content

- ❖ Xbox 360 target
- ❖ More Content:
  - 4 hi-res boss characters that can appear anywhere
  - Maps 30% longer
  - 7800 lines of dialogue, 40% increase from L4D1
  - Melee weapons
  - Animation
  - Effects
- ❖ How do we improve visuals, add more content, but not blow our budget?

The Zombie Apocalypse

❖ Shipping in a year: pick high-impact systems

❖ Survive the zombie apocalypse: interaction with the horde

❖ Focus on improving that experience:

- Horde variation
- Weapon feedback

❖ Lots of data from first game

- What was expensive or hard to author
- What worked and what didn't

The Horde

Variation in Left 4 Dead 1

❖ Playtesters recalled these variants

- Cop is good, he's "local flavor" in some game areas
- Others are like extras, should be visible but not memorable

❖ Dedicated textures for body geometry

VALVE

❖ Limited sharing of head textures, mostly on males
  • Not all maps look good on all geo variations

❖ Color tinting by multiplication

❖ Light/dark details must be visible under all tints
   - Untinted, has medium-value base to allow contrasting detail

❖ Starts dark, only gets darker

Analysis

❖ What didn't work

- Texture variation that didn't change contrast
- Fine detail of any sort

❖ What did:

- Geometry variation that changed proportions or moved features around

- Large, different color shapes: Color blocking

Color Blocking

- ❖ Concept from traditional painting
- ❖ Under-painting of local color without applying shading or detail
- ❖ Figuring out the local color: what do you see from a distance?

What We Learned

❖ Tinting must change color blocking

❖ Best if it affects contrast between local areas of color

❖ Geometric variation must also affect color blocking

What We Learned

❖ Color blocking also works at close range

❖ Blood patterns helped get mileage out of variants playtesters mistakenly perceived as clones

Prototyping the Variation System

❖ Prototype in external app, no overhead of implementation in-engine until we're sure

❖ Choose app that allows distribution to end users

- Maya's HLSL plugin

❖ Start with biggest effect for least investment: Tinting

- Gradient mapping?

VALVE

Gradient Mapping

❖ Just like in Photoshop!

❖ Luminance values only
❖ Map every pixel with same luminance to color specified in gradient ramp

Gradient Mapping

❖ Overdoing the colors doesn't work
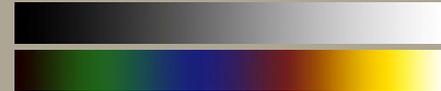
Gradient Mapping

❖ Fits well with DXT compression scheme

- Needs single channel only
- Alpha has most fidelity
- Alpha compresses independently from RGB

❖ Can't overdo the colors, but can't tint entire character with the same gradient ramp

Avoid Monochrome Results

❖ RGB can be used for masks

❖ Mask skin and clothing separately
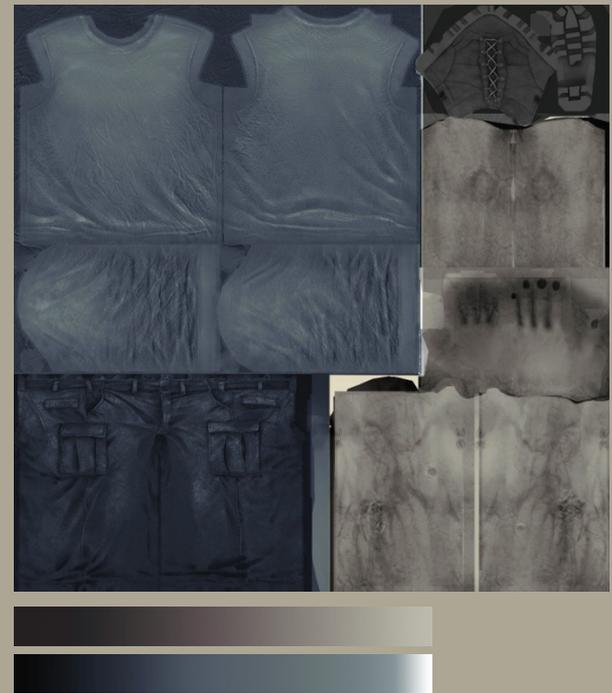
Masking Blood and Grime

❖ L4D1:

- Players identified different textures as clones
- Used blood to differentiate similar textures

❖ L4D2:

- Players should identify *same* texture as *different*
- Apply blood masking to disguise identical textures
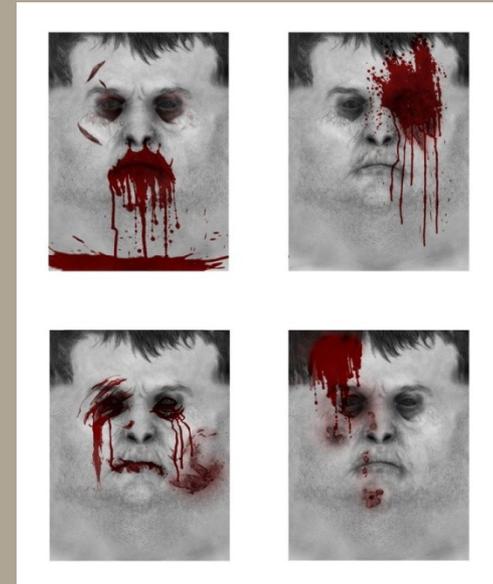
Masking Blood and Dirt

❖ Use masking to add blood
❖ Store all variants in existing texture
- Split texture into quadrants
- Store 4 masks in dedicated channel

❖ 2 texture lookups:
- ¼ size to select a single mask
- Full-size to get lum from alpha

❖ Do the same for grime

Detail Texture

❖ Blood is a solid color -- Grime doesn't have to be

❖ Use a detail texture

- Can vary depending on environment

Mud

Dirt/
Roadrash

Algae

Initial Results

Discussion and Additional Feature Requests

- ❖ Initial results promising: good overall range of luminosity
- ❖ Individuals still relatively monochrome
- ❖ Blood splats a bit blurry since masks are ¼ sized
- ❖ Unfinished goal for L4D1: retro-reflective effect of tapetum lucidum (eye-glow in headlights) signaling inhuman nature of infected
- ❖ Specular masking: important because of lack of normal maps
  - • Wait… no normal maps?
  - • Texture budget is limited: using normal maps means halving our texture budget, which means half the variation
  - • We've got to look at the fidelity of the horde as a whole, not its individual members

❖ …6 masks?! In 3 channels?!

- Skin tint
- Cloth tint
- Blood
- Grime
- Retro-reflectivity
- Specularity

❖ Already gave up normal maps for variation

- No way are we adding another texture just for masks!

Exclusive Masking

❖ Cloth and Skin don't overlap
❖ Can use different value ranges to mask each effect
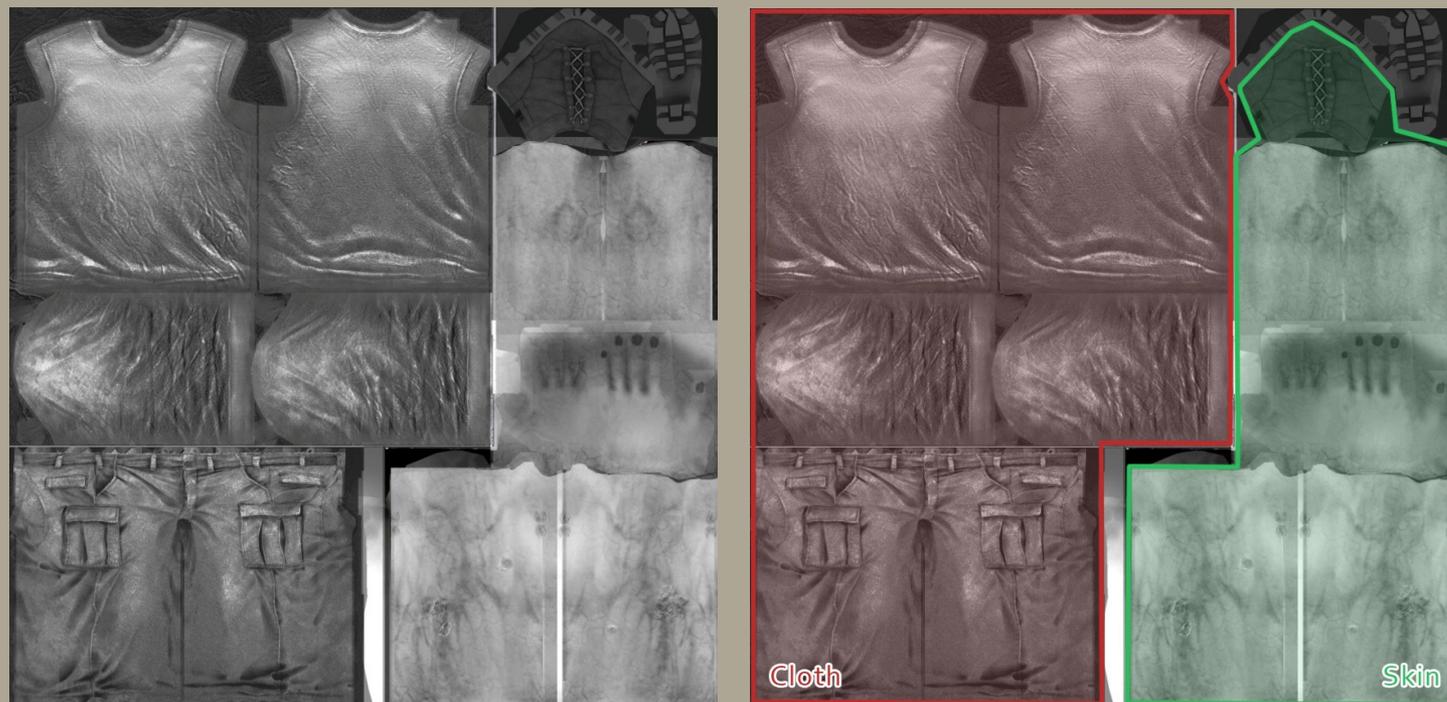
Cloth

Skin

Exclusive Masking

❖ Cloth and Skin don't overlap

❖ Can use different value ranges to mask each effect
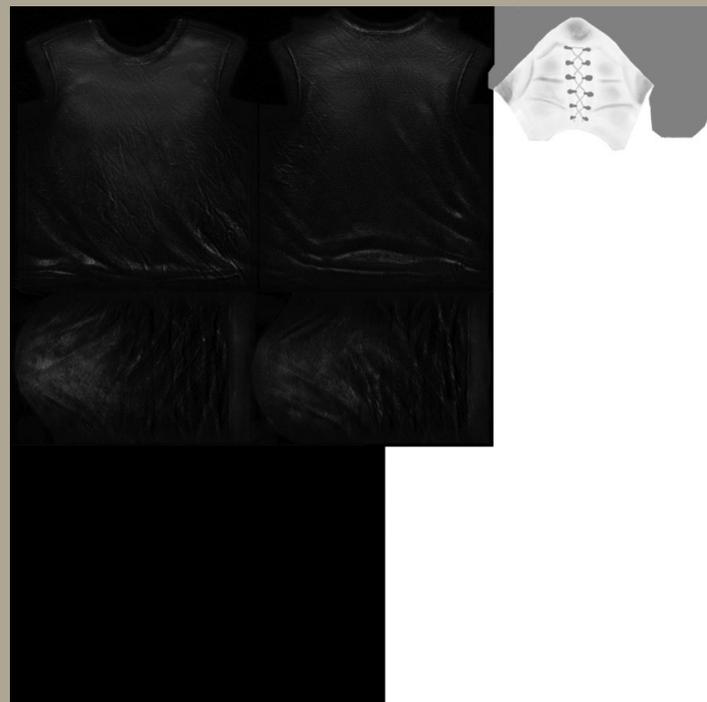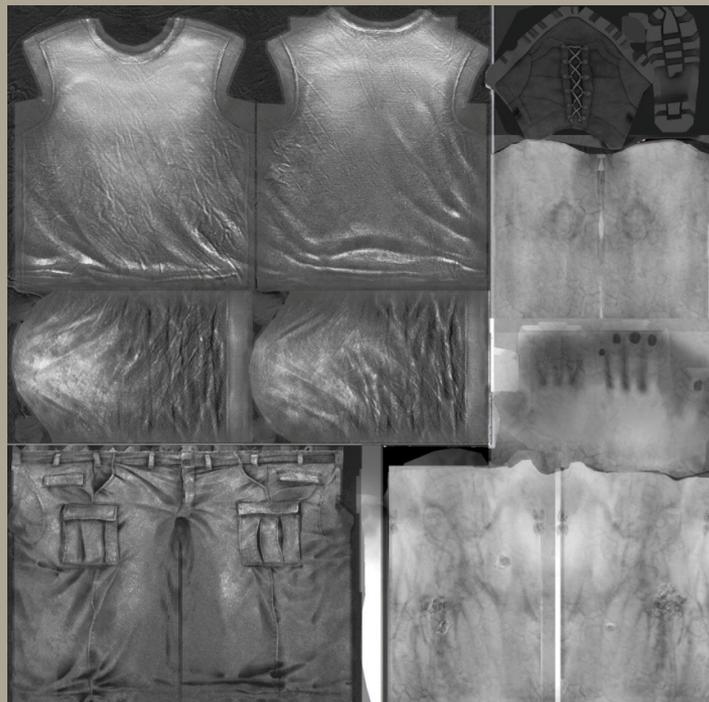
Exclusive Masking

- ❖ Modify mask in shader using levels-like operation

- ❖ Move the blackpoint to ignore all values below 127

- ❖ Result is skin-tint mask
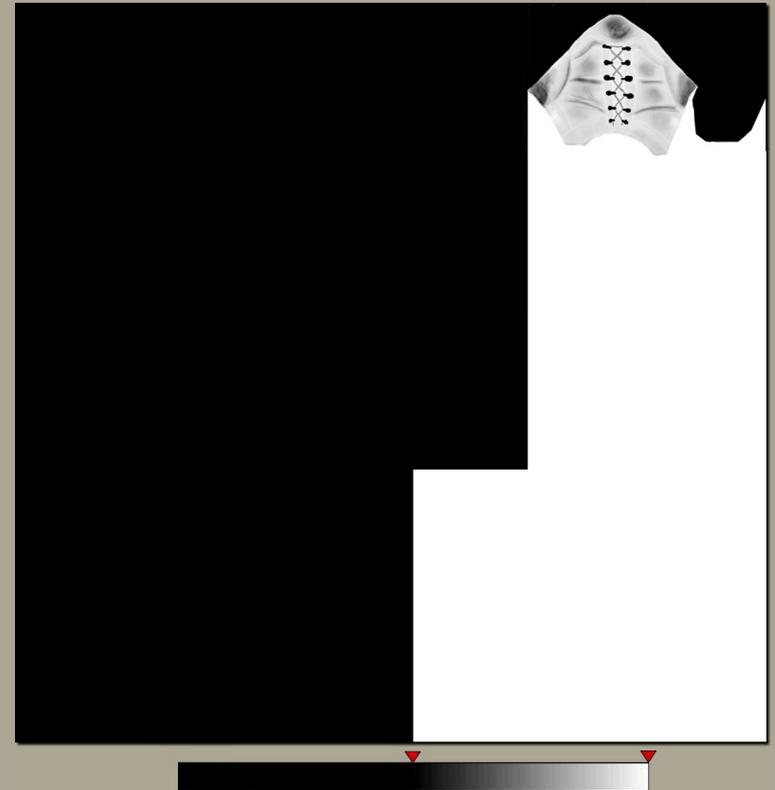
Exclusive Masking

- ❖ Do the same for cloth-tint mask, but invert first
- ❖ Ignore all values above 127
- ❖ Result is cloth-tint mask
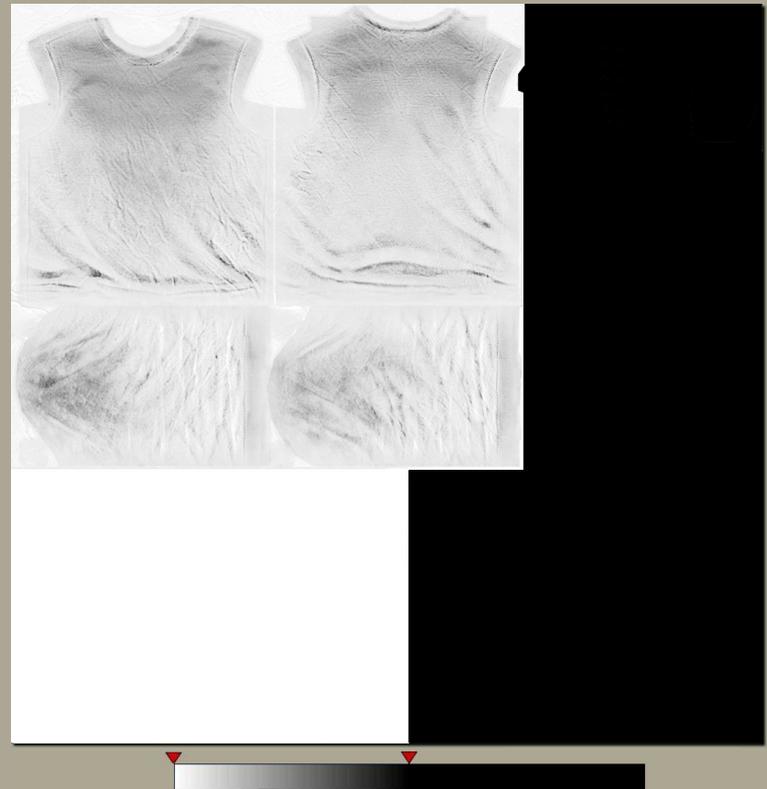
Exclusive Masking

❖ Result: can mask two separate gradient mappings using a single channel

❖ Only works because the masked areas don't overlap

Storing Information

❖ **Use exclusive masking and pair up effects**

- Spec…Detail
- Blood…Retro-reflectivity
- Clothing tint…Skin Tint
- And of course luminosity in the alpha
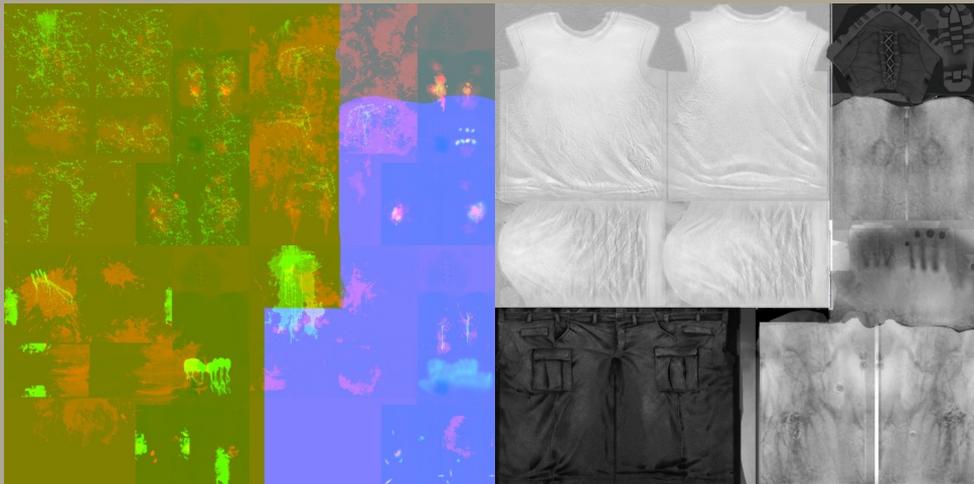
Storing Information

❖ Use specular mask from detail's alpha in detail-masked zones
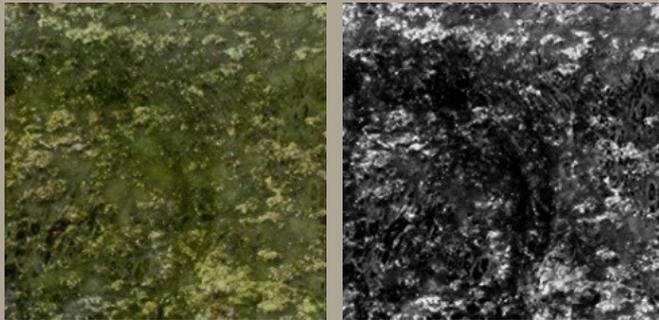
❖ Can pair blood and retro-reflectivity, if blood gets priority

- Blood on top of a retro-reflective material damps the retro-reflectivity anyway

❖ Smoothstep for blood patterns
- Lose some painted detail, get back hard edges

Storing Information

❖ Last problem: individual infected still look relatively monochrome

- Modify ranges in luminosity for further color variation
- Create more complex gradient ramp, limit luminosity in areas to map to only a portion of the ramp
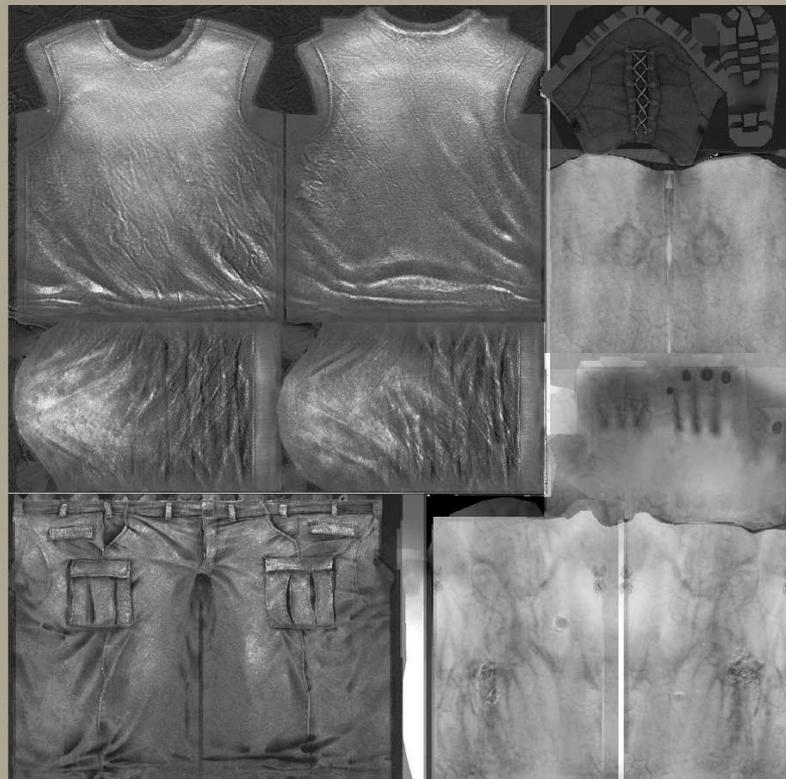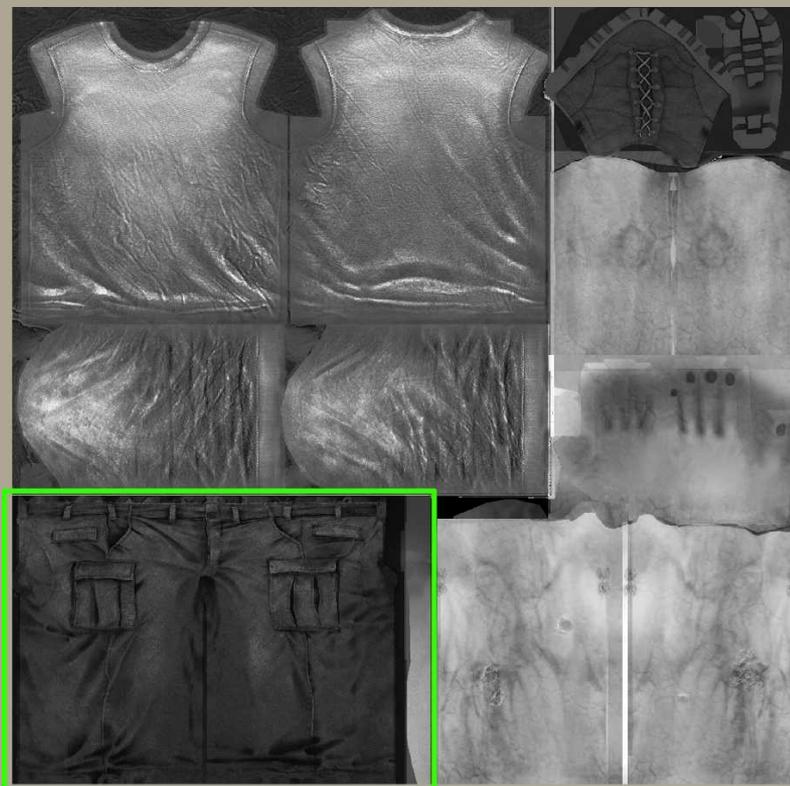
Modifying Ranges

Modifying Ranges
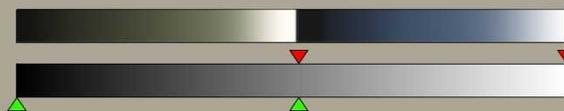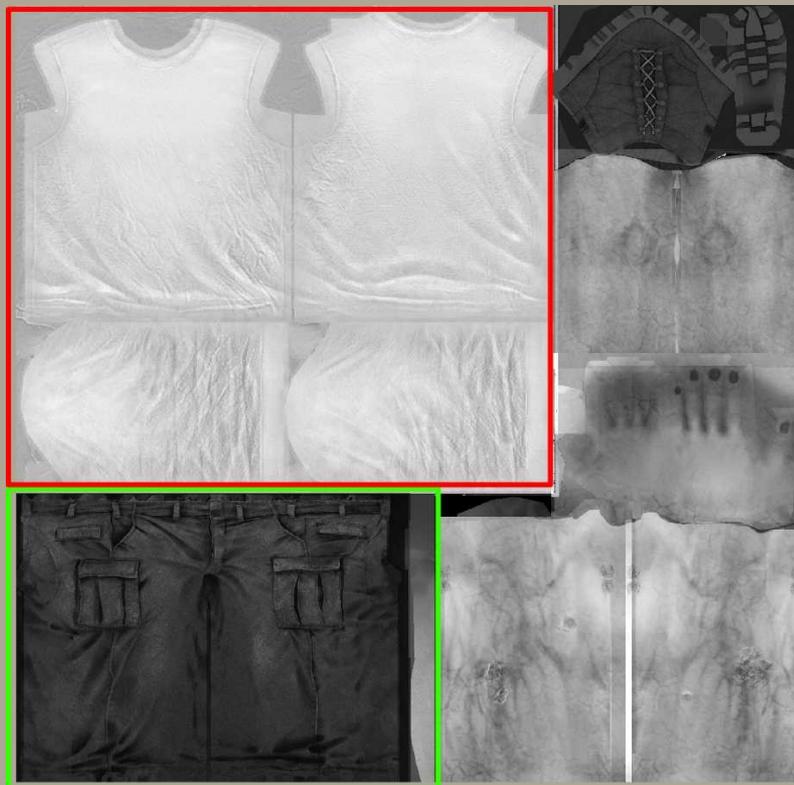
Modifying Ranges
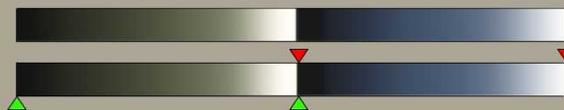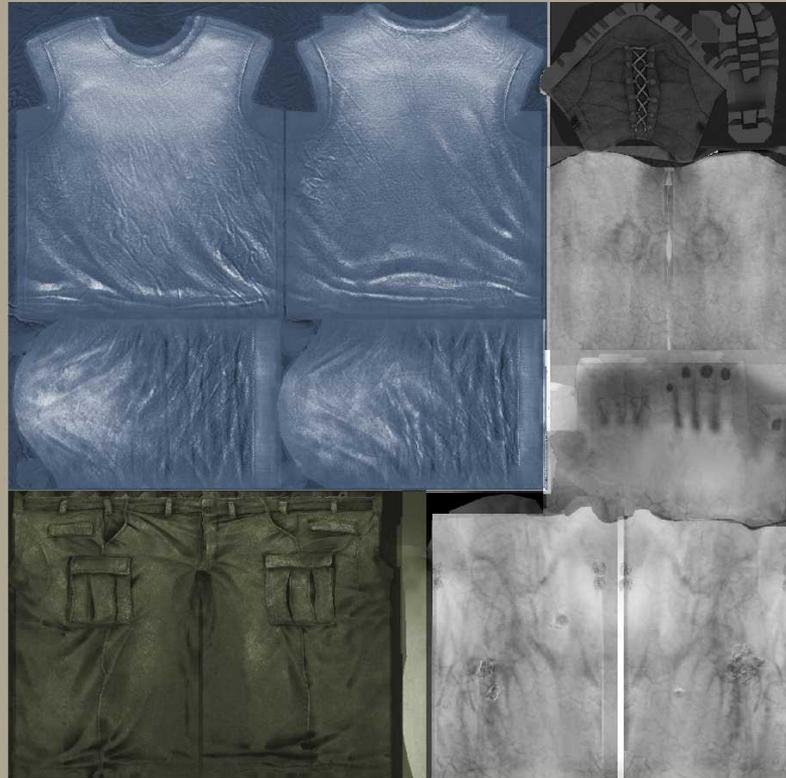
Modifying Ranges

Modifying Ranges
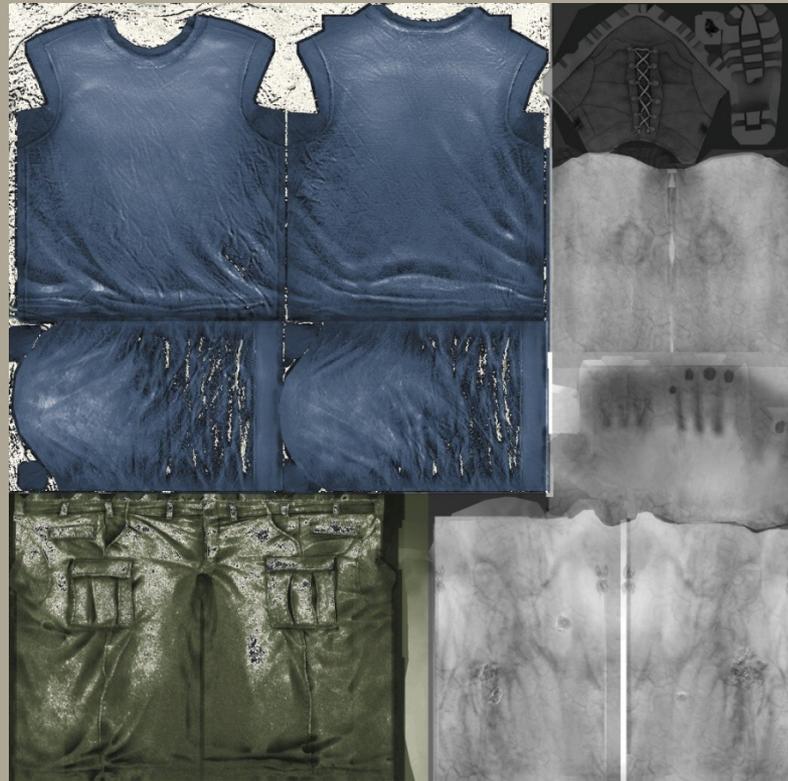
❖ Need buffer between ranges because of compression

Results

Geometric Variation

❖ Much of texture is shared, helps render batching

❖ Texel density in areas players focus on:

- Torso: Center of gravity, direction of motion, intent to move
- Head: AI has spotted a target
- Hands: Attack

❖ All geo vars made first, unwrapped together

❖ Lots of steps, high probability of user error

- Obvious candidate for scripting

❖ Let texture artists see the final result while working

❖ Shader does a lot of compositing with the masks

- Photoshop is pretty good at compositing too, hmm?

❖ Let's review what the shader does

Luminosity

Skin Mask

Blood Mask

Final Diffuse

Luminosity

Cloth Mask

Detail

Detail Mask

Spec Mask

Final Spec Mask

Detail Spec

Detail Mask
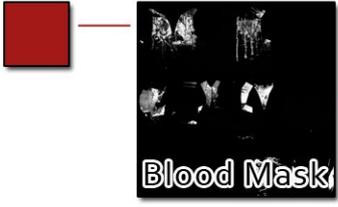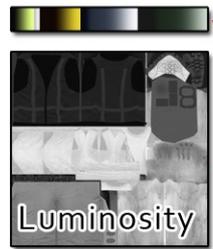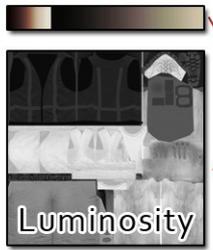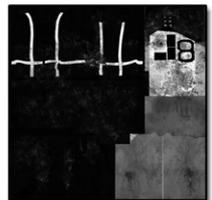
Retro-Reflect

❖ Create standard configuration with named layer sets

❖ Script setup and reconstruction

- Use gradient adjustment layers, pattern layers, and solid layers for masked effects

- Blood and detail painted at full size, one at a time, and hidden when not needed

- Luminosity painted at full range, levels adjustment layers push values into correct ranges for gradient mapping

- Specular mask painted in same file, hidden when not needed

❖ Result: Photoshop looks remarkably congruent with in-game result

Authoring Textures

Authoring Textures

Authoring Textures

Authoring Textures

Zombie Recipe

❖ Each infected contains:

- 2-3 head textures with 4 blood patterns each
- 4-7 head geometry variations
- 1 body texture with 4 blood patterns
- 3-8 body geometry variations
- Detail from shared texture
- 8 skin tints and 8 clothing tints from shared 16x256 texture
  - "Uncommon" common infected like construction guy in previous slide have their own dedicated palette

Zombie Recipe

❖ Simplest infected has over 24,000 variations
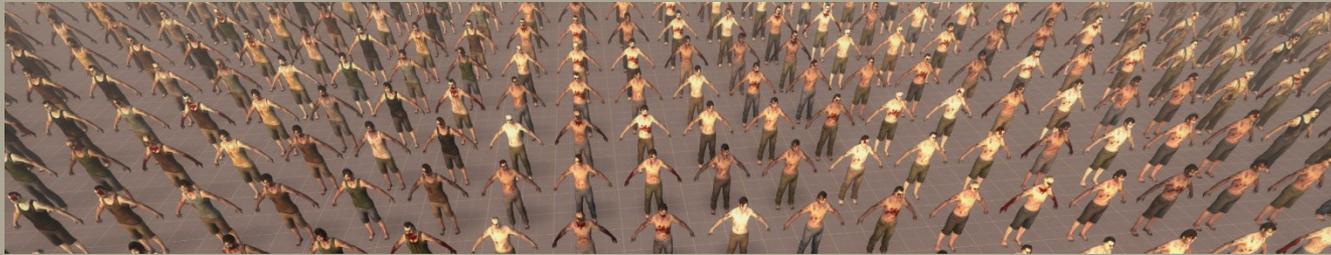
❖ Levels use as many as 6 models, as few as 2

- Depends on memory, costuming

❖ Creation time is less: made fewer, more effective textures

Final Results

Measuring Success

- ❖ 50% less memory
- ❖ 10x variation
- ❖ Lighting is done per vert instead of per pixel
  - Vertex shader instructions increased by ~100
  - Pixel shader comparable with L4D1
- ❖ Only "uncommon commons" stand out

The Wound System

Game-Level Goals

❖ Player experiences a zombie apocalypse horror film with their friends

- Zombies are endless, oblivious to hurt

❖ Provide feedback appropriate to type/level of weapon

- Communicate power of weapon
- Easily identify hurt or dispatched targets

Wounds in Left 4 Dead 1

- ❖ Built-in, expensive for vert mem
- ❖ 5 variations only, all hand-authored for each zombie
- ❖ Requires texture support, expensive for texture mem
- ❖ Always Fatal, doesn't support "oblivious to hurt"

VALVᴱ

❖ Place instanced wound object

❖ Deform or cut

❖ Geo level not good enough

- Deformation boundaries too dissimilar if mesh tesselation is not the same

❖ Place instanced wound object

❖ Deform or cut

❖ Geo level not good enough

❖ Pixel Level has own problems
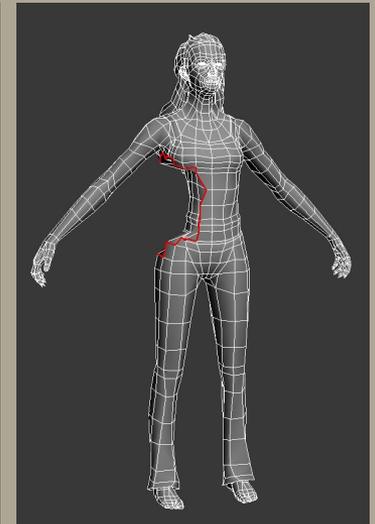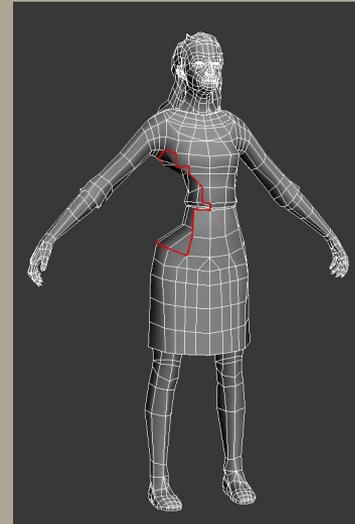
- Cut too uniform, doesn't look like damage

First Attempts

- ❖ Place instanced wound object
- ❖ Deform or cut
- ❖ Geo level not good enough
- ❖ Pixel Level has own problems
- ❖ Meat flowers not the way to go

❖ Place instanced wound object

❖ Deform or cut

❖ Geo level not good enough

❖ Pixel Level has own problems

❖ Meat flowers not the way to go

❖ Place geo inside: seams have to look messier
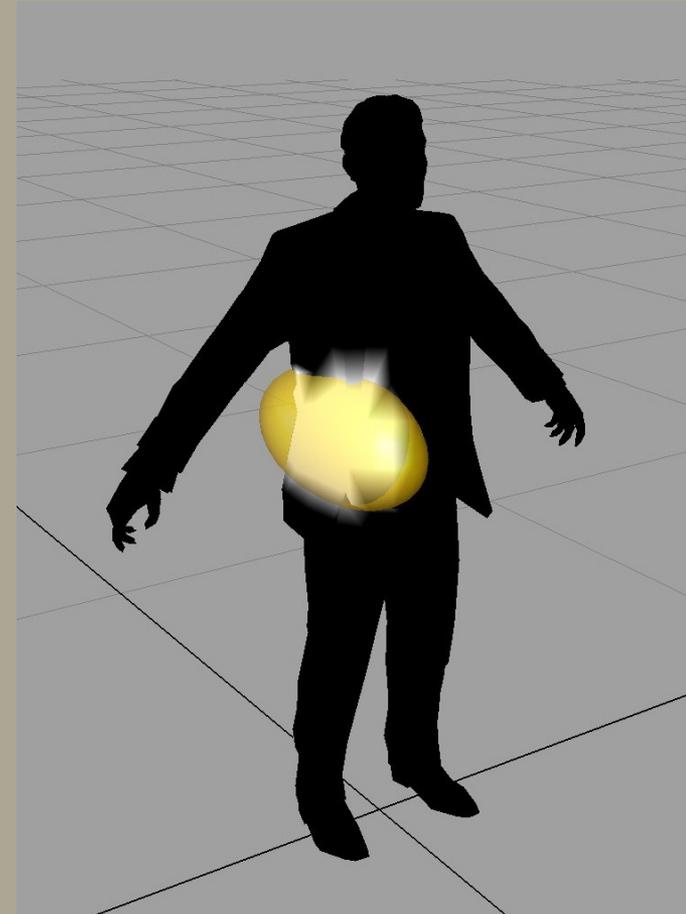
❖ Ellipsoid defines affected area

❖ Ellipsoid defines affected area
❖ Per-vert values for affected area
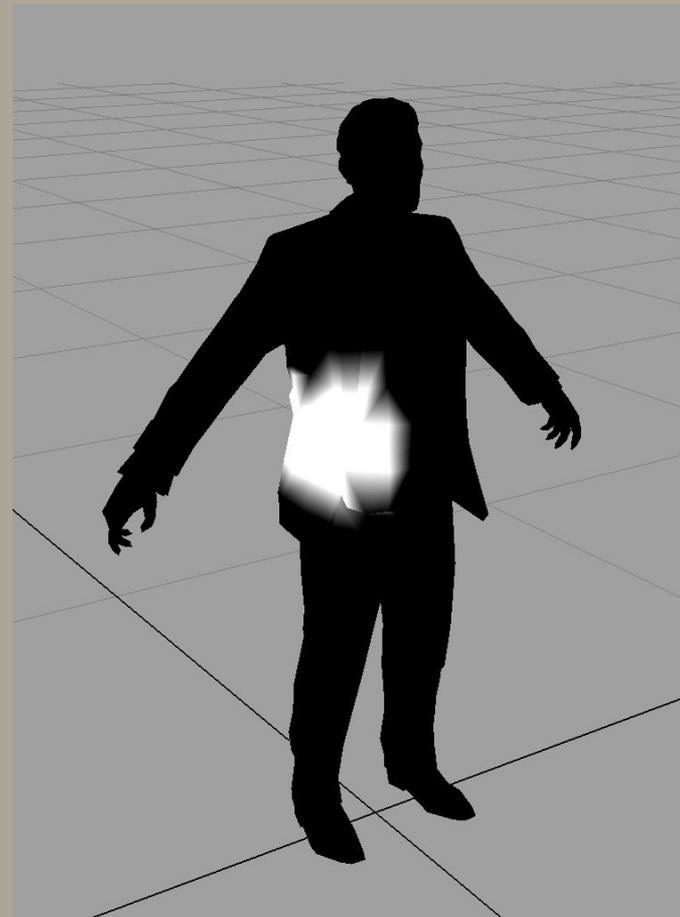
Blowing a Hole

- ❖ Ellipsoid defines affected area
- ❖ Per-vert values for selected area
- ❖ Determine falloff

Blowing a Hole

- ❖ Ellipsoid defines affected area
- ❖ Per-vert values for selected area
- ❖ Determine falloff
- ❖ Overlap planar projection
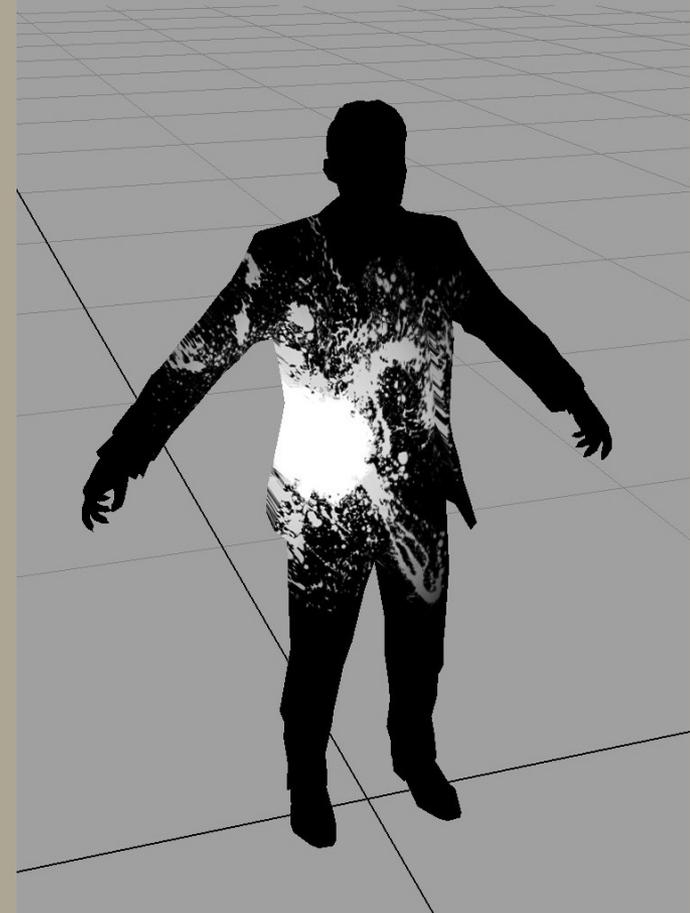
Blowing a Hole

❖ Ellipsoid defines affected area

❖ Per-vert values for selected area

❖ Determine falloff

❖ Overlap planar projection

❖ Only brightest area culls

Blowing a Hole

- ❖ Ellipsoid defines affected area
- ❖ Per-vert values for selected area
- ❖ Determine falloff
- ❖ Overlap planar projection
- ❖ Only brightest area culls
- ❖ Other non-black pixels contribute to blood masking

The Insides

perform boolean | internal anatomy model | dynamically wrap nurb mesh | extract textures, paint and sculpt.
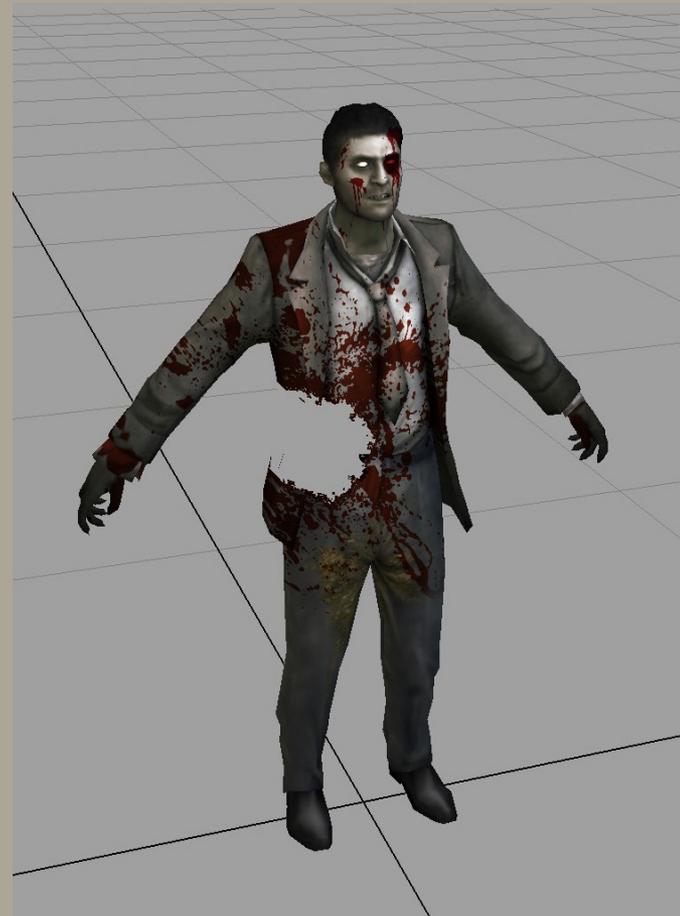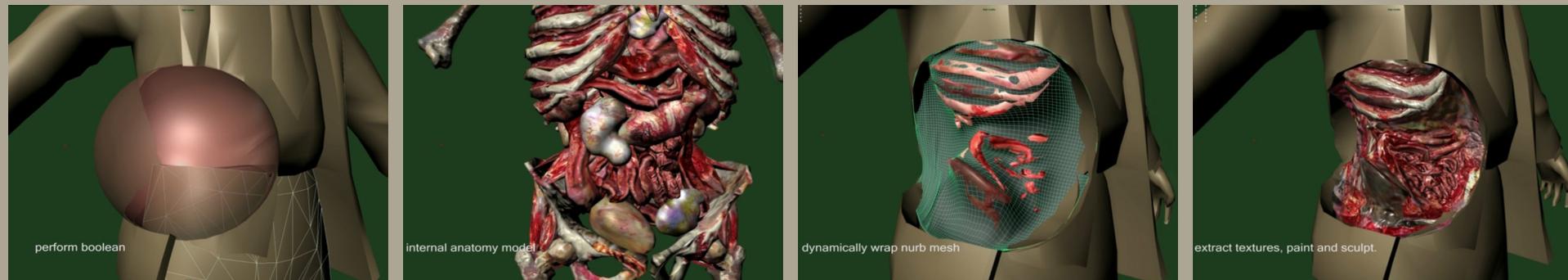
❖ Boolean with stretched sphere = ellipsoid cull

❖ Full interior model used as reference to keep wounds aligned

❖ Use fields and nurbs soft-bodies in Maya to wrap section of interior model

❖ Additional sculpting and painting in Mudbox: normal-mapping

❖ Skin to infected skeleton

  • Spawned wound will attach to infected and deform with it

The Insides
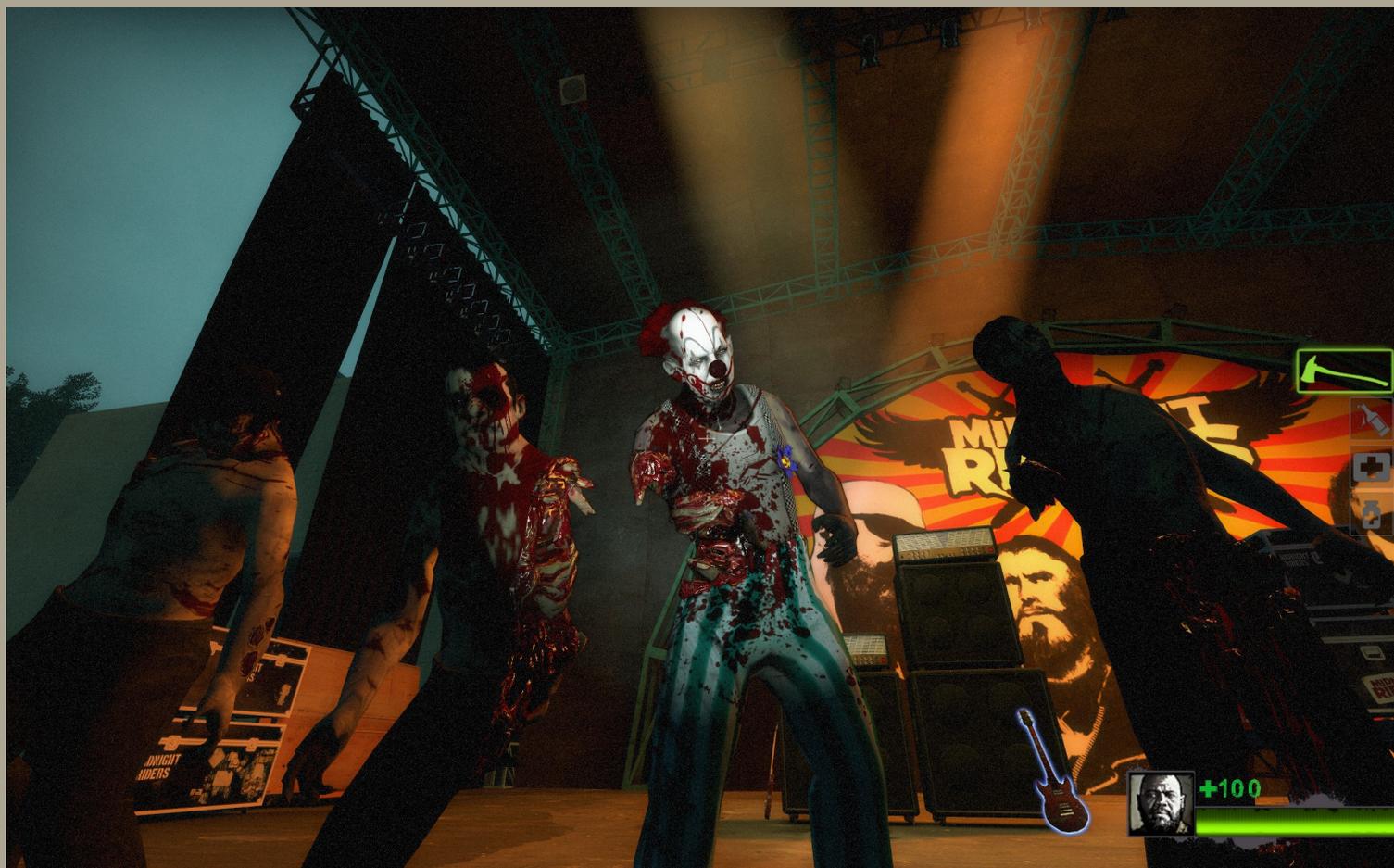
Slashing Damage

- ❖ Melee weapons bring players in range of common infected attacks

- ❖ Without wounds, power of weapons not visible: playtesters only saw drawbacks

- ❖ Adding wounds helped playtesters understand one-hit kills, ability to hit multiple infected with a single swing

- ❖ Adoption of the melee weapons increased

Slashing Damage

- ❖ Collapsing ellipsoid into disk resulted in stretched textures

- ❖ Added second shape for cull to texture for slash

- ❖ Directionality a factor in selecting a wound

❖ Drawbacks

- Texture Stretching

❖ Advantages

- Add multiple wounds before infected dies!
- Easy to iterate
- No extra mesh data to store

❖ Next steps

- Improve pipe bomb, grenade launcher and chainsaw with massive damage

Massive Damage

- ❖ Chainsaw, pipe bomb, grenade launcher: Communicate power of most destructive weapons
- ❖ Culling a torso as easy as culling an arm

Measuring Success

- ❖ 54 wounds each for males, females
  - Multiple wounds increase variety
- ❖ Each wound only 13% of the cost from old system
- ❖ Vertex shader uses another 15 instructions
  - Fill-bound, so rendering perf impacted minimally
- ❖ Pixel Shader uses 7 more instructions
  - Big whoop
- ❖ Horde and wound system together:
  - 1.5x as expensive
  - 10x variation, 156x the number of ways to die
- ❖ Visually a big hit with playtesters
  - "Loved pipe bomb gibbage"
  - "Big step above L4D…loved details like ribs showing"
- ❖ Improved player satisfaction with melee weapons

How do we improve visuals, add more content, but not blow our budget?

❖ Trade memory for computation
- Re-arrange content for small footprint
- Reconstitute/remix in shader

❖ Produce less content, more variation, iterate faster