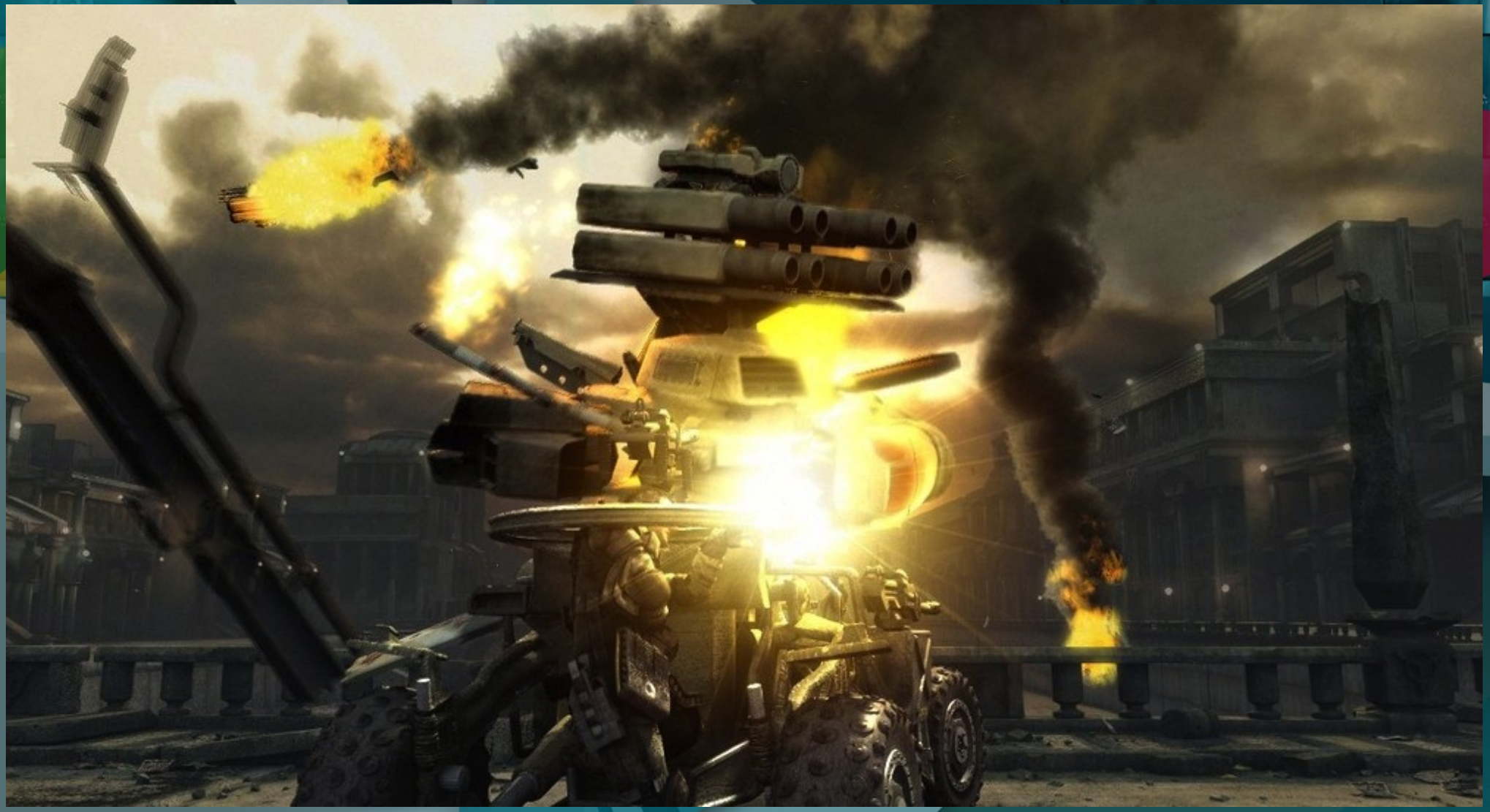


Render Engine Case Study:

Killzone 2

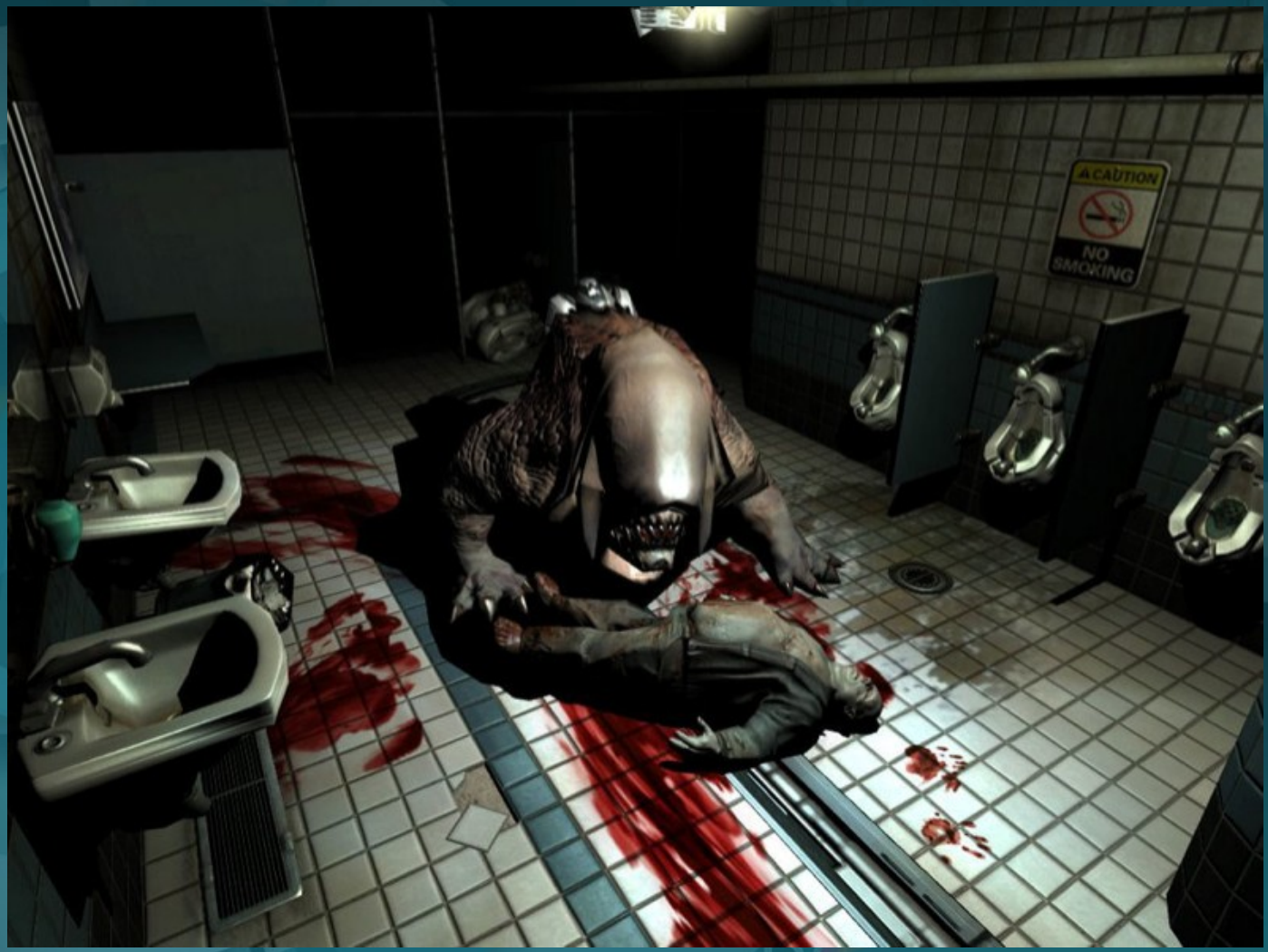
Overview

- Why is Killzone 2 interesting?
  - Takes advantage of PS3 hardware
  - Uses promising set of rendering technologies
  - Good example of supporting specific style goals with rendering
- Stuff to cover
  - Graphics in service of gameplay goals, a bit of history
  - Lighting technology
  - Particle systems
  - Post processing

Graphics in service of gameplay

- Want to choose graphics techniques that supports game design decisions
  - This used to be a much harder problem
  - In early days of 3D, technology limited game design decisions
    - Doom 1 - Couldn’t render real 3D, levels need to be flat
    - Quake - Leap to real 3D makes characters too expensive

Doom 3

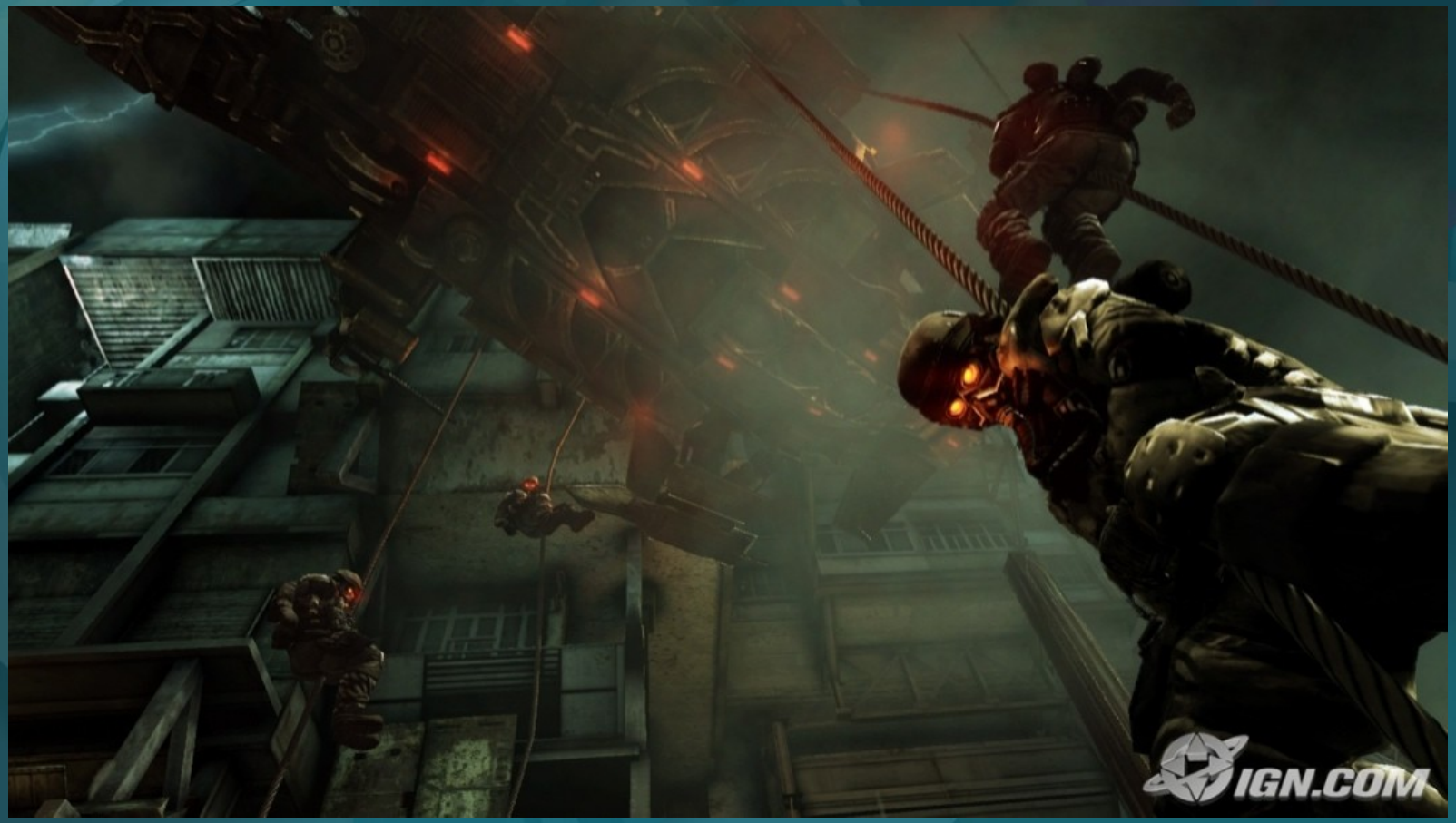


- Used to talk about Doom 3 in the equivalent presentation
  - GREAT example of choosing tech based on specific goals
  - Needed good shadows to create horror atmosphere

Graphics in service of gameplay

- Wanted to find equivalent modern game that made tradeoffs as interesting as Doom 3, but can't
  - Rendering tech is getting good, compromise is less needed
- Now we can start to focus a lot more on meeting artistic goals
  - You hear the term "cinematic" applied to a lot of features
  - Still tradeoffs though, lots of techniques are incompatible
  - And you only have so much GPU time

Lighting



Gamer : Oh, Crap! Look at all those bad guys!

Rendering Programmer : Oh, crap! Look at all those dynamic lights!

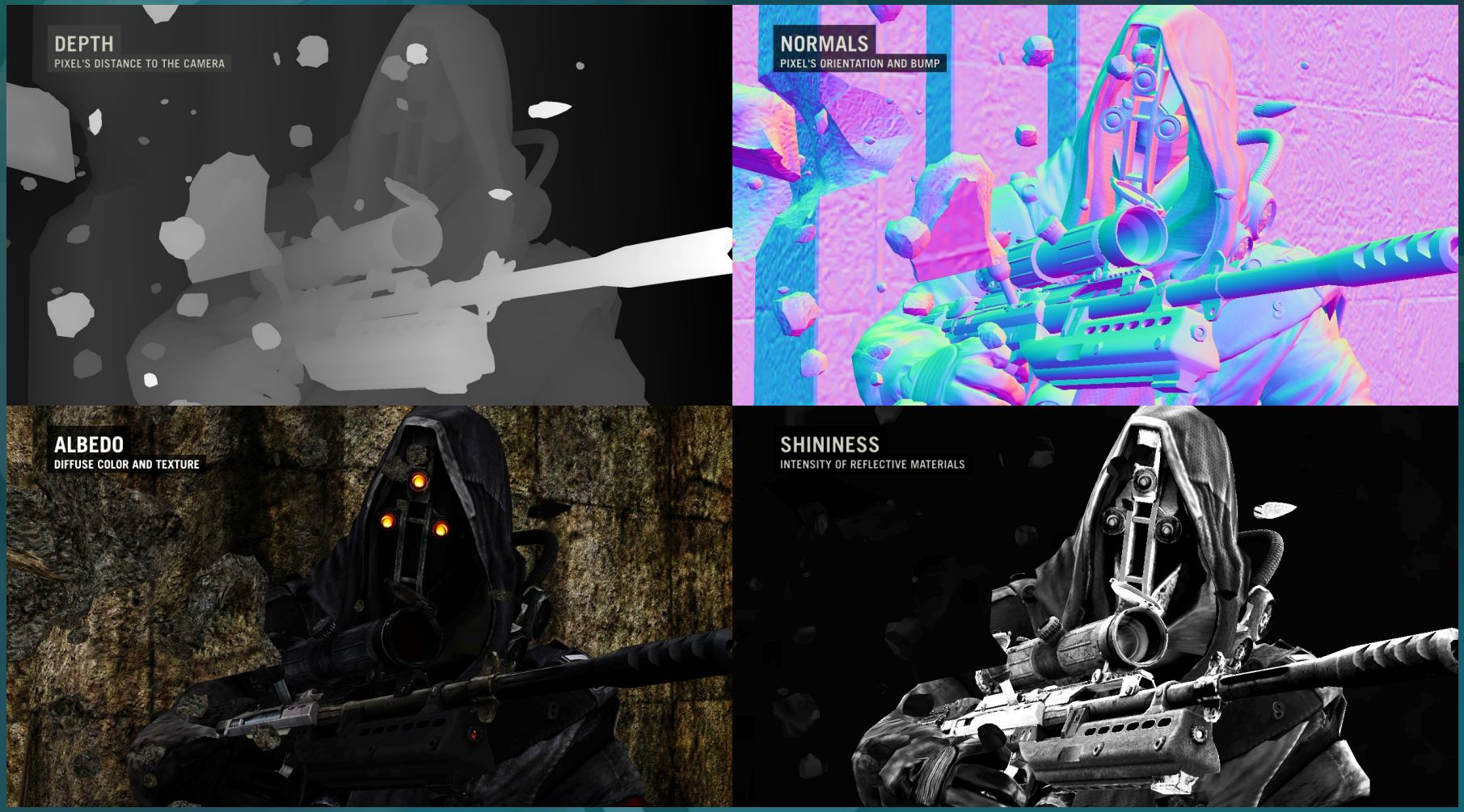
Lighting

- Real world has lots of lights in it
  - Lightmaps let you have as many STATIC lights as you want
    - You really want dynamic lights though
- Various other tricks to up light count
  - Doom 3 used extensive polygon preprocessing
  - Prototype uses light selection, light decals and effect lights
- Hard to get lots of dynamic lighting with conventional techniques
  - You tend to end up paying for it even when not benefiting
- But, when there are lots of lights they tend to be small
  - If we only render pixels touched by given light, cost is managable

Deferred shading

- The way to do this is to seperate surface property calculation and lighting
- This is known as deferred shading

G-Buffer generation

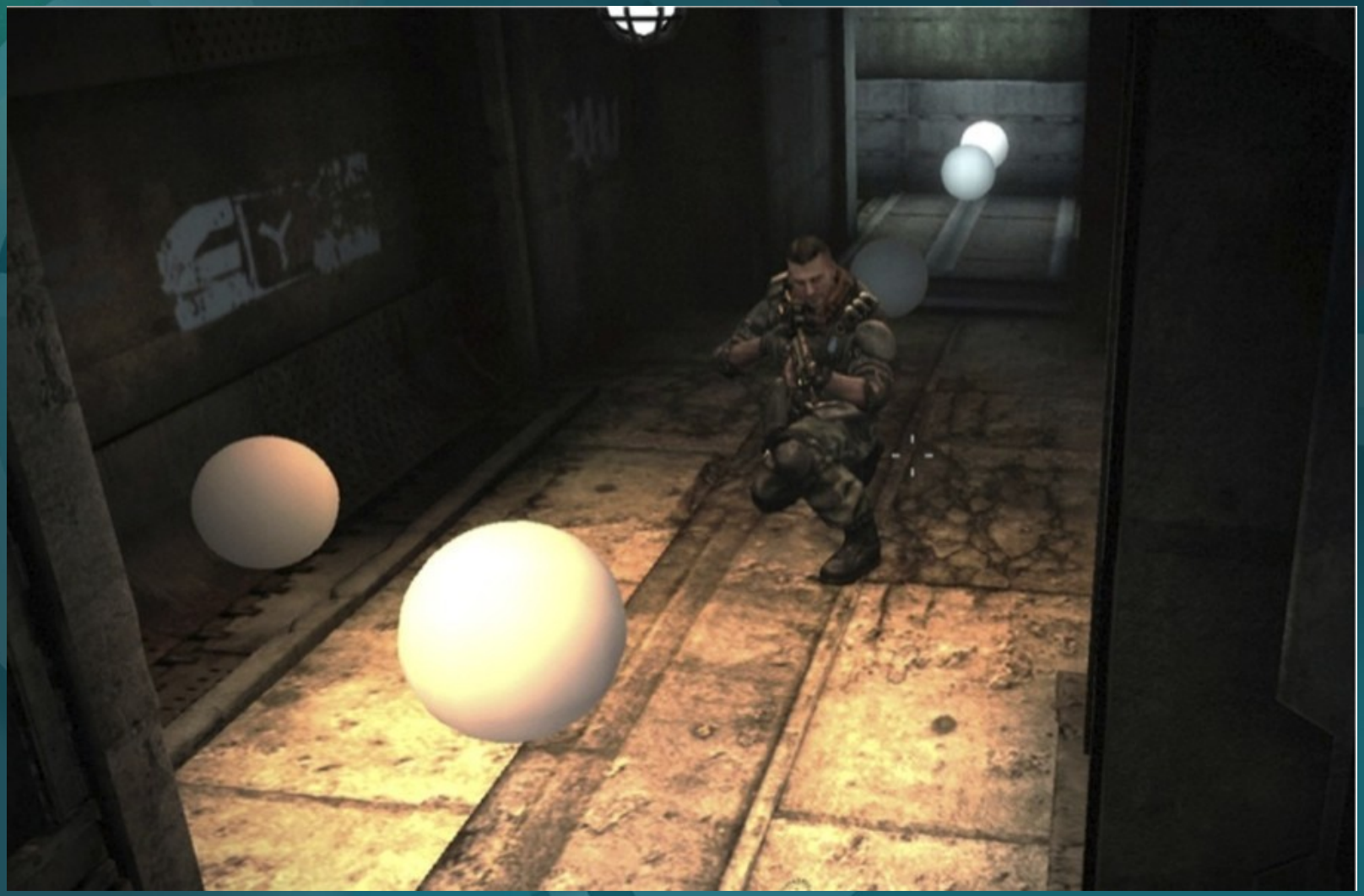


Render surface properties to temporary buffers

Lighting pass

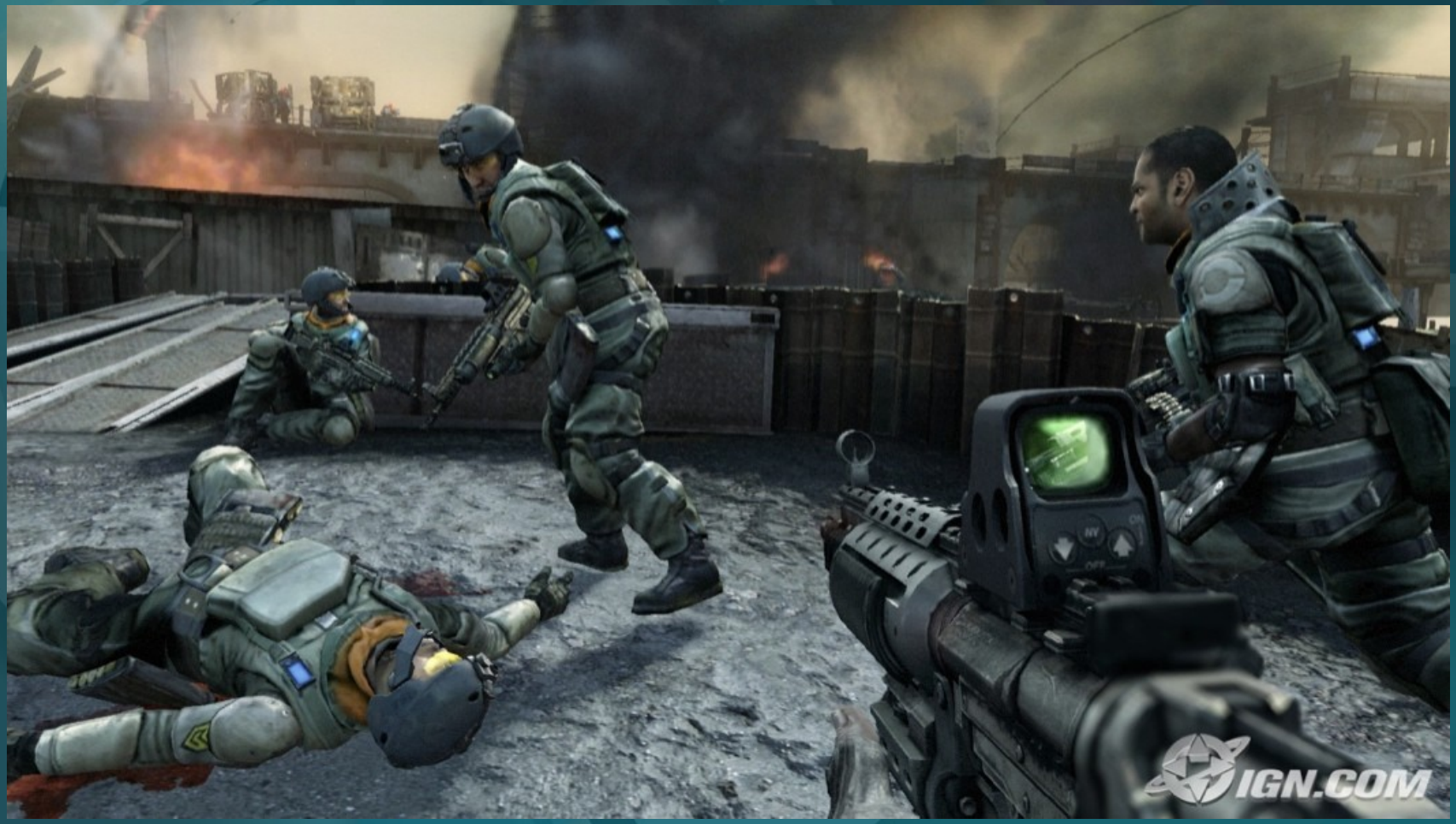
- Lighting pass
  Just render area affected by each light, and accumulate

More Lighting



- Also layer on other techniques
  - Static lights - lightmaps for world, spherical harmonics for dynamics

Particle systems



Goal - Make it look like a warzone

Solution - Set everything on fire

Particles

- LOTS of active particle systems in KZ2
  - 250+ systems
  - 3000+ particles
  - Prototype has more like 30 / 500
- Particles update on SPU
  - We update particle systems on SPU as well but have two problems
    - Did it later in development than they did
    - Cross platform game, can't depend on SPUs as heavily
- Fill rate probably still an issue
  - Not sure how they solved this but possibilities include
    - Lots of systems are in the distance
    - More particles may actually lower fill-rate due to less overlap
    - May do rendering tricks (lower res buffers)

Post processing effects

Post processing effects

- Problem : KZ2 heavily GPU limited
  - Deferred rendering very GPU intensive
  - Already had obvious stuff (vertex related) moved from GPU to SPU
- Solution : Move some pixel related code to SPUs
  - Note : This is insane

Post processing effects

- Why are pixel ops off GPU insane?
  - Stalls pipe, need to wait
- How did they get away with it?
  - Single, fixed hardware platform you can use more tricks
  - PS3 has just enough tricks availible to make it feasible
  - Couldn’t possibly do this on XBox 360 or PC
    - Might not need to though, graphics hardware is faster
- Big advantage of PS3, you can do stunts like this
  - Downside, sometimes you HAVE to

Post processing effects

- How do they do it?
  - Only works on effects done low res
    - Bloom, motion blur, DOF
    - SPUs to slow for high res (VRAM access probably is too)
  - GPU does downsampling
  - GPU and SPUs coordinate using interrupts
  - Processing on next frame can start while waiting
- It's quite slow
  - Much slower then doing on the GPU
    - Better quality though
  - Interrupts mean CPU doesn't have to baby sit
  - Being able to start next frame means RSX doesn't lose time
  - Had available SPU time, so slowness doesn't matter

Wrap up

- Deferred shading / lightmaps / spherical harmonics
  - Probably best all around lighting solution
- Pick rendering tech based on art goals
  - That many particles might not be needed in some games
- Know your hardware
  - It may have crazy trick up it’s sleeve
  - Multi-platform game are always a half-step behind

A RADICAL ENTERTAINMENT PRESENTATION

RADICAL ENTERTAINMENT™

RADICAL