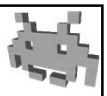
Course Plan



lec. 1: **Introduction**
lec. 2: **Mathematics** for 3D Games
lec. 3: **Scene Graph**
lec. 4: Game **3D Physics** + **Particle Systems**
lec. 5: Game **Particle Systems**
lec. 6: Game **3D Models**
lec. 7: Game **Textures**
lec. 8: Game **3D Animations**
lec. 9: Game **Materials** ADDENDUM
lec. 10: **Networking** for 3D Games
lec. 11: **Artificial Intelligence** for 3D Games
lec. 12: **3D Audio** for 3D Games
lec. 13: **Rendering Techniques** for 3D Games

46

Choice of material models: Chapter 2 ('00s)

Main reasons:
1. more GPU processing power affords us more realism.
2. Programmable shaders.
3. More GPU RAM to store textures for parameters

- The Lighting Equation becomes more complex
  - more terms are added
- It feeds on more material parameters…
  - Factors for: Fresnel effect, Anisotropic effect, Reflectivity – with environment maps, …
- Authoring materials becomes an increasingly complex, and *ad-hoc,* task
  - Difficult to port one material …
    - …from one engine to another, …from one game to another, …from one asset to another
  - Difficult to guess the right parameters for a given object
    - especially if it has to look good under widely different lighting conditions

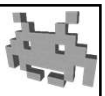
the task of the "material artist"

47

Much wider expressiveness is needd

48

Material models improving



49

Material models are improving

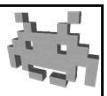
50

Choice of material models: Chapter 3 ('10s to today)

- Physically Based Materials (PBM)
  - an ongoing trend!
- General characteristics and objectives:
  - increased intuitiveness:
    - provide Material Artist with a higher-level material description
    - eases the Material Authoring task
  - increased standardization:
    - makes materials more cross-engine / portable (almost)
  - increased generality:
    - accommodates for more lighting effects / types of materials, such as Fresnel or anisotropic materials…
  - increased realism / quality:
    - more faithful, physically justified model of real-world materials
    - it's possible to capture materials from real-world samples
    - rendering results look better under widely different lighting env

51

«Physically Based Lighting» (PBL)



- A lighting model more inspired more by physical reality
  - For example, energy conservation
  - And less by tricks that just follow some intuition, see specular Phone reflections
- A lighting model accepting, as input, a PBM
- Also, a lighting model taking fewer shortcuts than otherwise typical
  - For example, use
    - diffuse color: *one* texture
    - baked AO: *a separate* texture
  - Instead of:
    - diffuse color × baked AO : one texture (cheaper!)
- Warning: PBM & PBL are, basically, buzzwords ☺

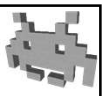
52

PBM and PBL: objectives

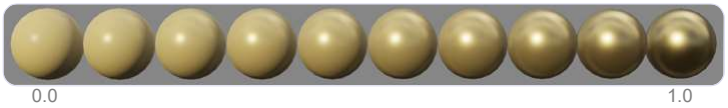
- Make realistic materials easier to design (by material artists)
- Make it possible to *capture* materials from real world samples
- Make it easier to make lighting look reasonably realistic… under many different lighting environment
- Standardize, and make it easier to share material description across different project applications
- Ideally: most real-world materials can be described accurately
- Ideally: no combination of Material parameters looks bad or wrong
- Use few parameters (ease storage, authoring), every parameter describe in a 0 to 1 range

53

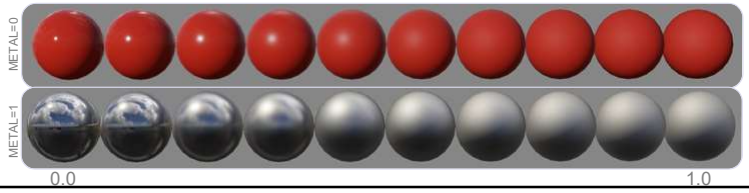
Physically Based Materials (PBM). A good choice of parameters



- Base color (rgb – or "diffuse", same as old school)
- Specularity (scalar – or rgb sometimes)
- "Metallicity" (scalar)



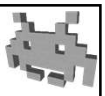
- Roughness (scalar)

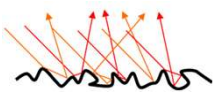


images: unreal engine 4

54

Physically Based Materials (PBM)



- Base color: (a color)
  - Same as in base lighting model
- Specularity: (a scalar, in 0 to 1)
  - Total amount of light bouncing off the surface with reflections (regardless of how).
  - Very rarely it's 0 or close. Although in different ways, "everything is shiny."
- Metallicity (or metallosity): (a scalar, in 0 to 1)
  - Is the surface a (conductive, dielectric) material or not?
  - In theory, either 0 or 1, but it's possible to interpolate results.
- Roughness: (a scalar, in 0 to 1)
  - Low = 
  - High = 

55