Modern Technologies

Jaanus Jaggo
Why is useful to know modern technologies?

I’m making an indie 2D game, why should I care?
Why is useful to know modern technologies?

I’m making an indie 2D game, why should I care? Because indiepocalypse
Evolution of PC game graphics

Then vs Now

Not just the number of polygons
Importance of light

Simulated light behaviours:

- Shadows
- Reflections
- Ambient light
- Atmospheric scattering
- Physically correct materials

History - 2.5D

Pitstop 2 - 1984

Wolfenstain 3D 1992

Doom 1993


Looks like 3D but not made of polygons
History - Early 3D games

Quake 1996

Voodoo1 (3dfx) 1996
First pure 3D GPU

GeForce 256 (Nvidia) 1999

GTA 3 - 2001

Pure textures with baked lights - polycount was important
History - 3D with real-time lighting

Doom 3
2004
Stencil shadows

Elder’s Scrolls Oblivion
2006
History - Post effects

Battlefield 2
2005

Fallout 3
2008

Hitman Absolution
2012
History - Deferred rendering

- Cheaper lighting
- Even more post-effects

Crysis 2
2011
History - Deferred rendering

http://www.adriancourreges.com/blog/2016/09/09/doom-2016-graphics-study/

http://www.adriancourreges.com/blog/2015/11/02/gta-v-graphics-study/
Future - raytracing?
Introduction to lighting in 3D games

Real world light is a complex thing

John Carmack explaining light in games:
https://www.youtube.com/watch?v=IyUgHPs86XM&feature=youtu.be&t=18m38s
Rendering process

Window -> Analysis -> Frame Debugger
Light

Significant role in 3D games.

**RYSE: Son of Rome** scene with and without lighting:
Light baking

Lightsources

1. Directional light:
   No falloff

2. Point light:
   Has falloff

3. Spot light:
   Has falloff, funnel shape

4. Ambient light:
Spot Light source effects

IES profiles for spot lights - provided by real world light manufacturers

Volumetric light beams

Point Light source effects

Tube and sphere lights

Free alloy shaders:
https://assetstore.unity.com/packages/vfx/shaders/alloy-physical-shader-framework-11978
Lightsources

Area lights - real world light sources, have shape and size.
https://www.youtube.com/watch?v=ZLRgEN7AQgM

Article: https://eheitzresearch.wordpress.com/415-2/
Other light technologies

- Shading models
- Shadows
- Real time Global Illumination
- Reflections
Shading

Early games:
- Baked
- Simplistic light model (eg. Phong)

Ambient + Diffuse + Specular = Phong Reflection
Shading

Surfaces tend to look more plastic compared to modern games
Shading

Physically-Based rendering (PBR) - more physically correct computations
Shading

**Metalness** - is the material metal or nonmetal

![Metalness examples](image.png)

Metalness = 0

Metalness = 1
Shading

**Energy conservation** - object can not reflect more light than it receives.
Shading

**Fresnel** - the percentage of light that a surface reflects at grazing angles.
Shadows
Shadows

**Shadow mapping** - the scene is rendered from the perspective of the light source (shadow map).
Shadows

The quality depends on the resolution of the shadow map.
Shadows

Cascaded Shadow Maps -> multiple shadow maps with different resolutions.
(More precision close to camera)
Global Illumination

Global Illumination (GI) - ambient light bouncing off the surfaces
Global Illumination

Static lightmaps (Raycasting)

It can be done in real-time but it is very very slow.
Global Illumination

Unity -> Enlighten

Baked static geometry + dynamic light sources
Global Illumination

Unity -> Enlighten  P.A.M.E.L.A

Player Controlled Lighting
Global Illumination

Unity -> **Enlighten**

Only static geometry can bounce light, to receive it **light probes** have to be added.
Global Illumination

Both Unreal and Cryengine have real-time voxel cone-tracing GI solutions. -> too slow for older computers

Unreal engine elemental demo https://www.youtube.com/watch?v=MOvfn1p92_8
Reflections
Reflections

**Planar reflections** - camera renders scene below the surface

Performance expensive, only works with planar surfaces (not so good for wavy water).
Reflections

**Reflection probes** - the surrounding is rendered to a cubemap.

Good results near the reflection probe.

[More information about cubemaps.](#)
Reflections

Parallax-corrected cubemaps: 
https://www.youtube.com/watch?time_continue=72&v=Bvar6X0dUGs

Same cubemaps projected differently.
Reflections

**Screen space reflections** - reflections sampled from already rendered scene.

Artifacts near the edges
Other technologies

- Geometry manipulation - Chamfered edges
- Volumetric effects - Fog
- Particles
- Post-effects
Limited texture size

UV layout starts simple and innocent

https://www.reddit.com/r/starcitizen/comments/3ogi3o/im_an_tech_artist_in_the_industry_and_id_love_to/
Limited texture size

Problem:

texture space requirement grows depending on:

- Model scale
- Model complexity
- How close the player can get

https://www.reddit.com/r/starcitizen/comments/3ogi3o/im_an_tech_artist_in_the_industry_and_id_love_to/
Limited texture size

Ok, what about the Gladiator ship from Star Citizen

How big texture would you need?

https://www.reddit.com/r/starcitizen/comments/3ogi3o/im_an_tech_artist_in_the_industry_and_id_love_to/
Limited texture size

Solution: tiling textures + decals

https://www.reddit.com/r/starcitizen/comments/3ogi3o/im_an_tech_artist_in_the_industry_and_id_love_to/
Result

https://www.reddit.com/r/starcitizen/comments/3ogi3o/im_an_tech_artist_in_the_industry_and_id_love_to/
Texture sampling

Sampling texture based on world coordinates (triplanar sampling):

- **Faster workflow** - Unwrapping and texture painting can be avoided
- **More fun**

https://www.facebook.com/groups/IndieGameDevs/permalink/10154835397656573/
Fog

Historically used to hide the limited render distance

Adds sense of depth
fog

Fog particles
Fog

**Volumetric fog** - scatters light, creates light rays
Fog

**Volumetric fog** - scatters light, creates light rays

Unity implementation: [https://github.com/Unity-Technologies/VolumetricLighting](https://github.com/Unity-Technologies/VolumetricLighting)
Particles

Large number of tiny sprites or objects - simulate “fuzzy” phenomena
Particles

Can be used in game logic - Zombie boids
Particles

Modern particles:
- Emit light
- Collide with room geometry
- Distort background

GPU particles -> only available in Unreal Engine
Post effects

Postprocess effects - applied after the whole screen has been drawn.
Post effects

Postprocess effects - applied after the whole screen has been drawn.
Deferred shading - light calculations are done in post effect.
Post effects

**Resulting image with post effects**
Other post effects

**Depth of field** - blurs objects in the distance
Other post effects

Ambient occlusion - darkens corners and cavities
Other post effects

**Ambient occlusion** - darkens corners and cavities

[Image of Ambient Occlusion Effect]
Other post effects

**Bloom** - makes light pixels bleed out of their boundaries

[Image: Unity natural bloom]
Other post effects

**Distortion** - distorts and curves the whole image

[Distortion shader]
More technologies I don’t have enough time to talk about
Conclusion

This was just a small selection of modern 3D techniques.

Learn more how GTA 5 effects and optimizations:
http://www.adriancourreges.com/blog/2015/11/02/gta-v-graphics-study/
GC effects researched in ATI

Diana Algma MSc

Siim Raudsepp BSc

Jaagup Kuhi BSc

Marko Täht MSc
What to study next?

MTAT.03.015 Computer Graphics

MTAT.03.305 Computer Graphics Seminar

MTAT.03.328 Computer Graphics Project
Next lecture procedural generation