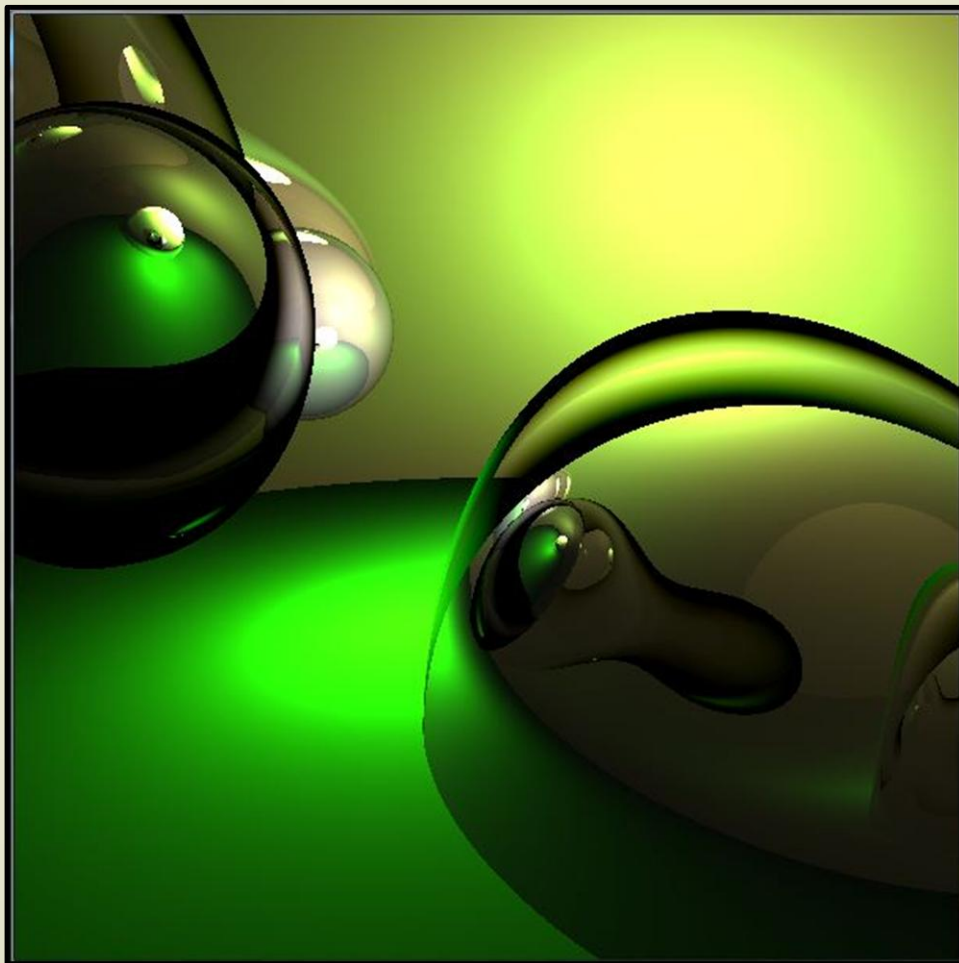


GL-raytracer

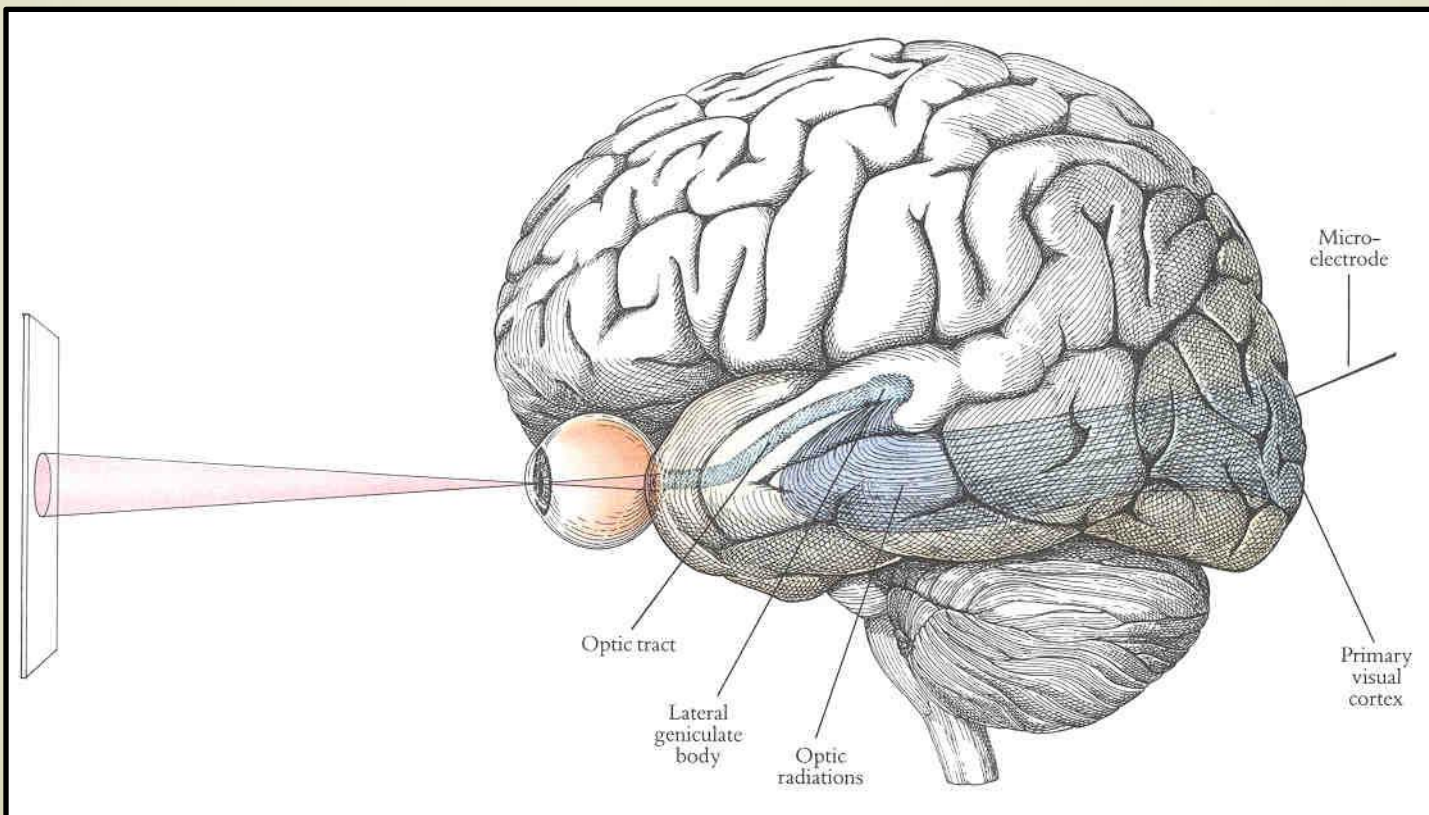
Készítette: Kovács Artúr

A sugárkövetéssel készült képek jól
néznek ki



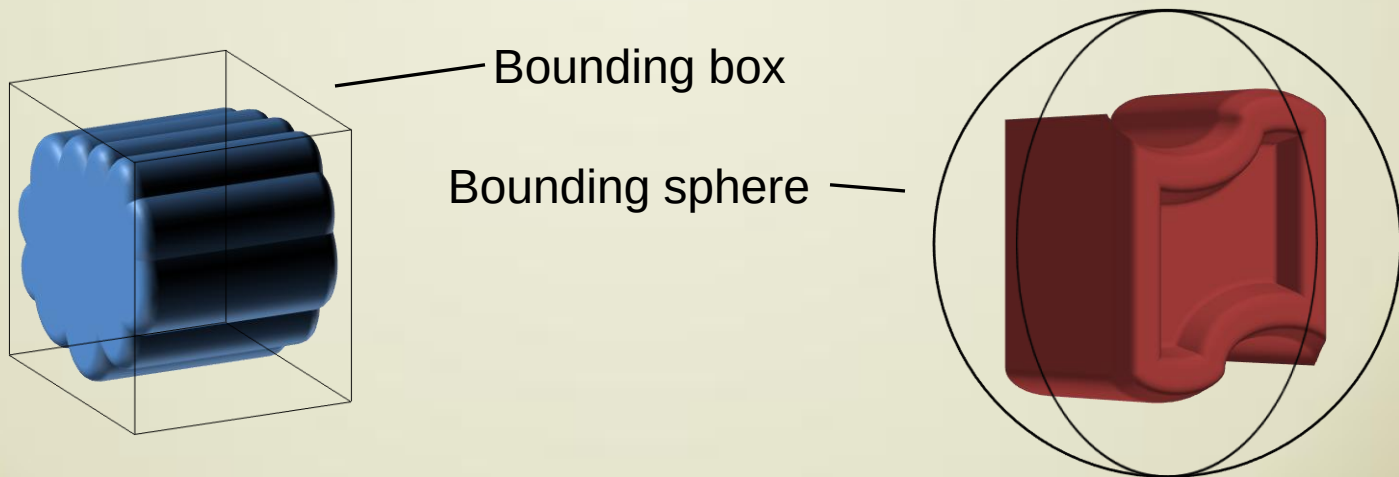
De jó lenne interaktív módon körbenézni

A valóságos interakció érzéséhez minimum 30 fps



Néhány optimalizálási lehetőség (sugárkövetéshez)

- Befogó térfogatok (bounding volumes)
- Térpartícionálás (space partitioning)
- Párhuzamosítás (multithreading)



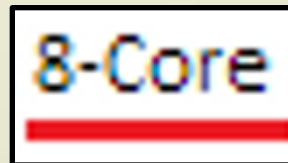
Párhuzamosítás

Többmagos processzorok...

Essentials	
Status	Launched
Launch Date	Q3'13
Processor Number	i7-4960X
# of Cores	6
# of Threads	12
Clock Speed	3.6 GHz
Max Turbo Frequency	4 GHz
Intel® Smart Cache	15 MB



Processor	AMD FX <u>8-Core</u> Black Edition
Model	FX-9590
OPN Tray	FD9590FW8KHK
OPN PIB	N/A
Operating Mode 32 Bit	Yes
Operating Mode 64 Bit	Yes
Revision	
Base Speed (MHz)	4700



Párhuzamosítás

Még több magos videokártyák...

GTX TITAN Z GPU Engine Specs:	
CUDA Cores	<u>5760</u>
Base Clock (MHz)	705
Boost Clock (MHz)	876
Texture Fill Rate (GigaTexels/sec)	338

5760



Features	Specs	Pictures	AMD R9 295X2
When developing technology			<u>5632</u>
11.5 TFLOPS <u>5632</u> stream processors 8GB memory 512-bit memory interface x 2			



Videokártya programozása

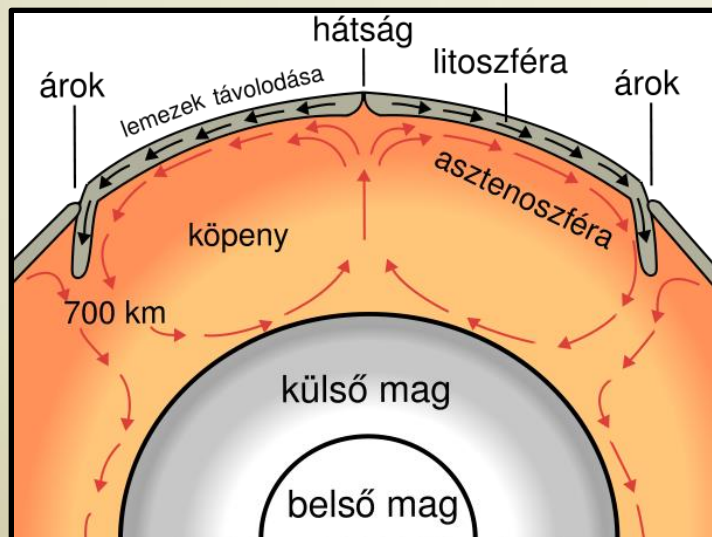
- Erősen párhuzamosítható problémákhoz kifejlesztett eszközök:
 - OpenCL
 - CUDA (csak nvidia)
 - C++ AMP
 - stb....

Videokártya programozása

– Grafikus API-val:

- DirectX
- OpenGL
- (újabbban Mantle, Metal)

Kiszámolt képet egyből
meg is jeleníthetjük



OpenGL

Pro:

- Windows, MAC, Linux

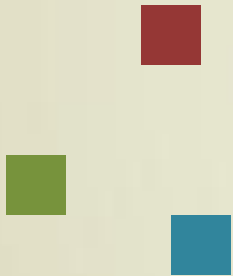
Contra:

- Ha használni akarjuk, szükségünk lesz egyéb függvénykönyvtár(ak)ra is



OpenGL primitívek

GL_POINTS



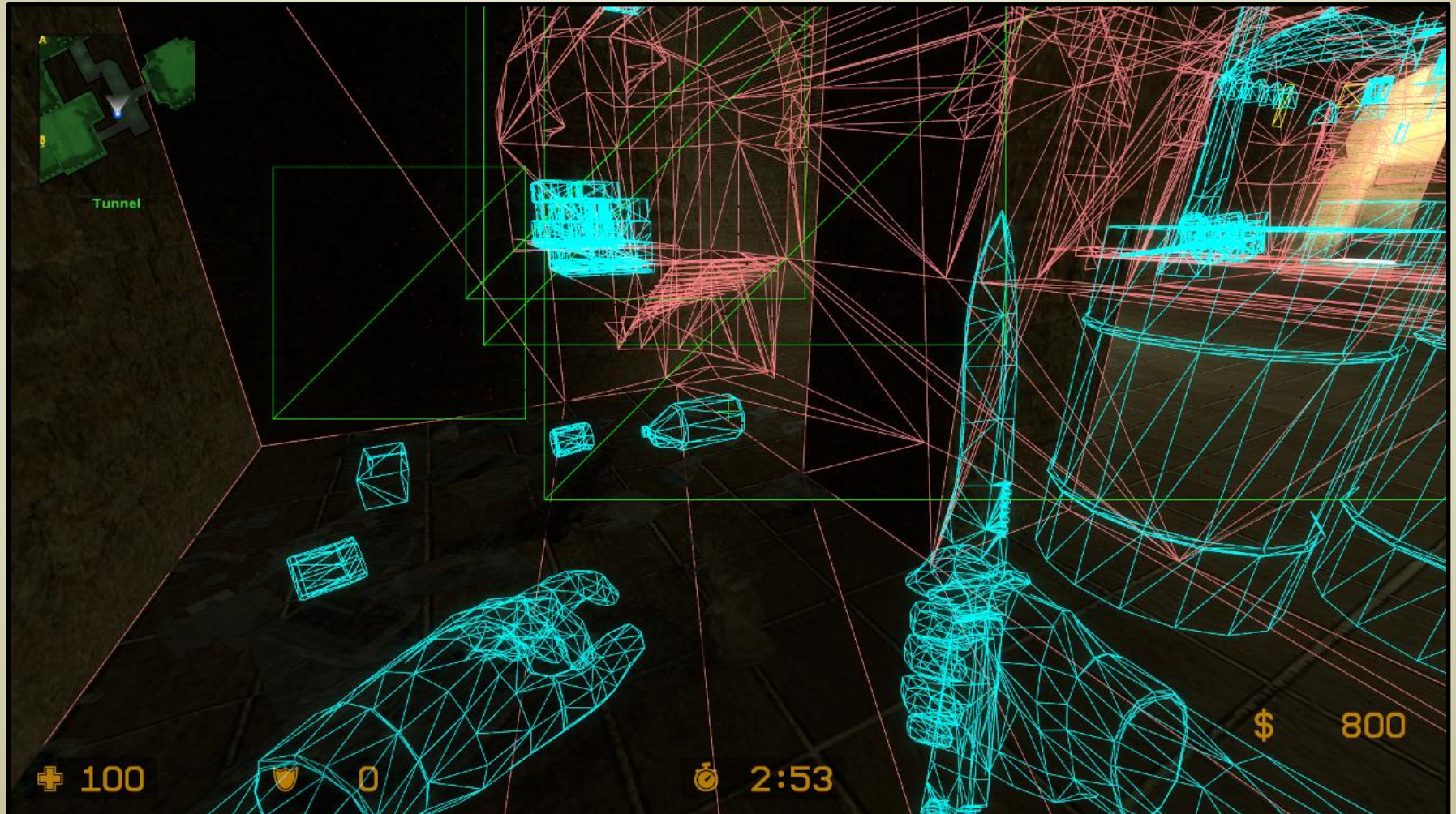
GL_LINES



GL_TRIANGLES



Igen, mindent lehet háromszögekből...



Mi határozza meg a primitív színét?

A fragment shader!

– Hatására a primitív lehet egyszínű:



– Vagy színátmenetes:



– Vagy textúrázott:

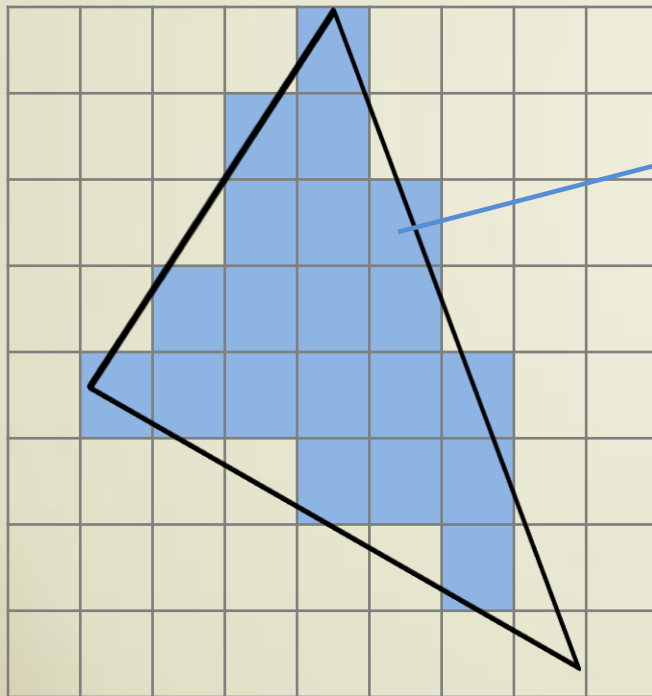


És legfőképp bármilyen.

Fragment shader - Pixel shader

Ez gyakorlatilag egy program.

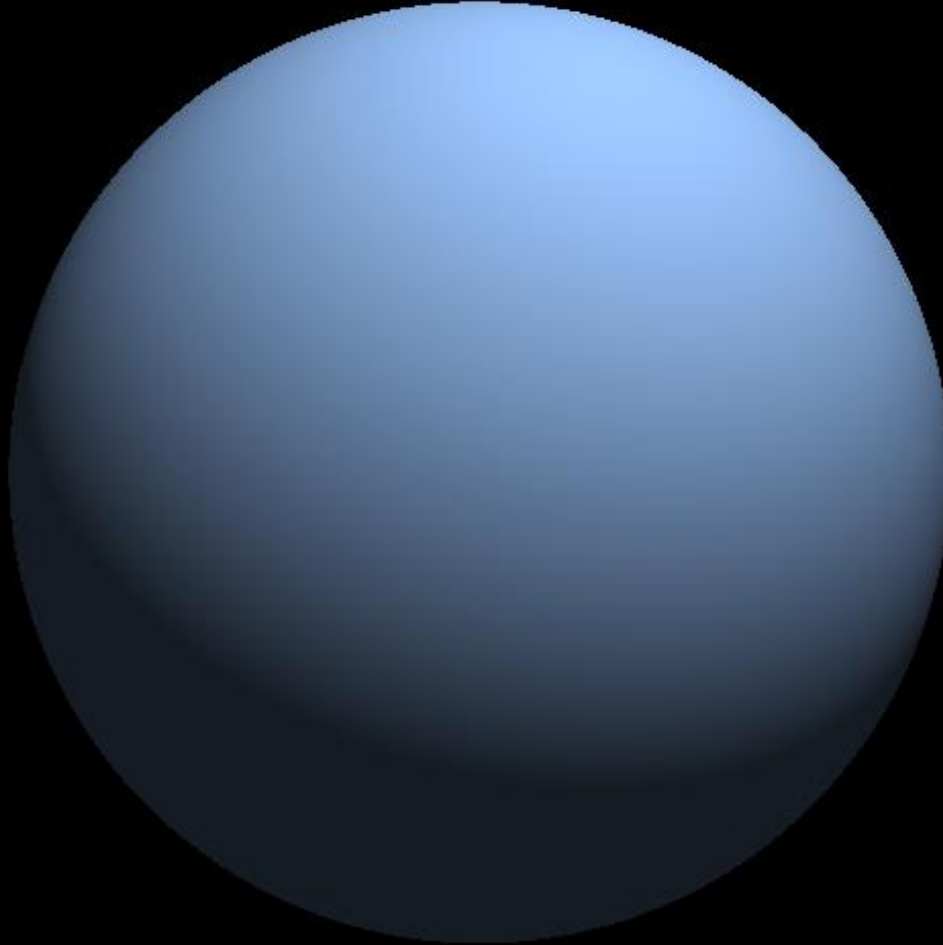
Egy kimenete van: a pixel színe.



Mindegyik kék pixelre
lefutott a fragment
shader. Párhuzamosan.

OpenGL magyarázása eddig tartott. Itt az a lényeg, hogy úgy oldjuk meg a sugárkövetést, hogy kirajzolunk egy teljes képernyőt lefedő téglalapot, aminek a fragment shader-ébe írjuk az egy sugárnak a sugárkövetését számoló kódot. (Így néz ki egy teljes képernyőt lefedő téglalap aminek a fragment shadere a kék színnel tér vissza.) Tehát itt kell felidézni, hogy is működik a sugárkövetés.

Itt pedig már azt látjuk hogy hogy néz ki amikor a fragment shaderbe leírtuk a sugárkövető kódot, és meghatároztunk egy gömböt.



Konklúzió

Valósídejű

- Valósídejű megjelenítés

Offline

- Komplex jelenetek
(rengeteg fényforrás,
objektum)
- Összetett, valóságű anyagok
- Soft shadows
- Anti-aliasing
- Post processing

Ajánlott mindefféle

- OpenGL-hez „OpenGL 3.0 and above”! :
http://www.opengl.org/wiki/Getting_started#Tutorials_and_How_To_Guides
- John Carmack, The Physics of Light and Rendering:
<https://www.youtube.com/watch?v=MG4QuTe8aUw>