**RAY TRACING ÎN JAVASCRIPT**

Autor: Bratan Radu-George

Cadru didactic coordonator: Bocăneală Corina

Facultatea de Științe și Mediu, Matematică Informatică

**ABSTRACT**

Ideea de „ray tracing” a fost explorată de nenumărați matematicieni și fizicieni încă din secolul al șaisprezecelea și mai ales în ultimii 4 ani, datorită avansurilor tehnologice majore care au permis din ce în ce mai multor oameni să obțină acces la hardware capabil de a interpreta astfel de algoritmi. Atât profesioniștii care lucrează în domenii precum dezvoltarea de jocuri video sau post-producția de filme cinematografice, cât și programatorii sau artiștii obișnuiți sunt atrași de astfel de tehnologii incitante, deoarece le oferă oportunitatea de a crea o lume virtuală detaliată, vie și asemănătoare cu cea reală. Cu toate acestea, tehnica ray tracing încă nu este accesibilă tuturor, datorită prețului ridicat al procesoarelor capabile să o interpreteze în timp real, fapt care i-a determinat pe programatori să exploreze metode prin care pot eficientiza acești algoritmi.

La nivelul de bază, tehnica ray tracing simulează interacțiunea luminii cu suprafețele aflate într-un spațiu. La un nivel mai avansat, această tehnologie combină diferite metode de simulare a razelor primitive de lumină cu tehnica iluminării globale și așa numitele „shadere”.

Scopul acestei lucrări este să prezinte un exemplu de algoritm creat să funcționeze pe toate dispozitivele cu acces la Internet și să explice ce optimizări pot fi implementate pentru a mări viteza de „randare”.

**CE ÎNSEAMNĂ RAY TRACING**

În termeni simpli, „ray tracing”-ul este o tehnică de simulare a luminii folosită pentru a genera imagini digitale. Ideea de trasare a unor raze pentru a genera o imagine a fost inventată încă din secolul al șaisprezecelea de către pictorul și teoreticianul Albrecht Dürer, care, folosindu-se de un fir de ață atașat unui băț din lemn, a trasat conturul unui obiect din viața reală pe o bucată de pânză.

**CINE FOLOSEȘTE ACEASTĂ TEHNICĂ**

Tehnica ray tracing este folosită, de cele mai multe ori, de artiștii digitali, dezvoltatorii de jocuri video și oamenii din industria producției filmelor. În prezent, există o cerere, aflată în continuă creștere, pentru efecte speciale fotorealiste în jocuri video și filme.

**AVANTAJE ȘI DEZAVANTAJE RAY TRACING**

Unul dintre avantajele acestei tehnologii este libertatea artiștilor de a explora mai multe concepte creative, atunci când au la dispoziție toate resursele necesare să își îndeplinească viziunea artistică.

Cu toate acestea, unul dintre cele mai mari dezavantaje este costul ridicat, atât la nivelul timpului de procesare al interacțiunii dintre razele de lumină și suprafețele obiectelor, cât și la nivelul prețului componentelor capabile să îndeplinească astfel de procese.

**CUM FUNCȚIONEAZĂ TEHNICA RAY TRACING**

Să presupunem că avem o scenă 3D care conține o sferă, o sursă de lumină și o cameră foto virtuală care generează o imagine pe ecranul calculatorului. La un nivel de bază, algoritmul ray tracing calculează direcțiile tuturor razelor care provin de la sursa de lumină, indiferent dacă acestea interacționează cu sfera sau nu. Ulterior, razele care au lovit sfera ricoșează în diferite direcții, și doar acelea care ajung în obiectivul camerei vor fi reprezentate pe ecranul calculatorului. Metoda aceasta, numită „light based ray tracing”, este mult prea costisitoare din punct de vedere al timpului de procesare, deoarece algoritmul calculează direcțiile tuturor razelor de lumină, nu doar ale celor care sunt reprezentate pe ecran.

O versiune îmbunătățită a acestui algoritm se numește „eye based ray tracing” și funcționează invers: razele pornesc din centrul camerei virtuale și ricoșează spre sursa de lumină. Această metodă este mult mai eficientă deoarece numărul de raze generate de camera virtuală este un număr finit, egal cu numărul de pixeli ai ecranului calculatorului, comparativ cu infinitatea (în limitele memoriei computerului) de raze transportate de sursa de lumină din metoda anterioară.

**IMPLEMENTAREA ÎN JAVASCRIPT/TYPESCRIPT**

Pentru a calcula direcțiile razelor de lumină, pozițiile obiectelor în spațiu și interacțiunile dintre acestea, m-am folosit de vectori, pe care i-am definit într-o clasă numită Vector3, alături de un set de operații cu aceștia, precum suma și produsul scalar. Pentru a efectua calcule cu vectorii, am creat trei variabile, x, y, și z, reprezentând coordonatele lor. Pornind de la această clasă, am definit ulterior razele de lumină (formate din vectori), punctul de intersecție dintre o rază și un obiect (al cărui normală este un vector), o suprafață plană (formată din doi vectori perpendiculari) și o sferă (a cărui centru este, practic, un vector nul), folosindu-mă tot de clase.



Alături de definițiile acestor obiecte 3D, am implementat câte o funcție pentru calculul poziției unei drepte (vectorul rază de lumină) față de o sferă și calculul poziției unei drepte față de un plan.





La final, algoritmul se folosește de două subprograme pentru a calcula iterativ culoarea fiecărui pixel de pe ecran, în funcție de cum se intersectează razele cu obiectele din scenă.

**BIBLIOGRAFIE – IMPLEMENTARE ALGORITM**

* <https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-ray-tracing/how-does-it-work> [18 Aprilie 2022]
* <https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-ray-tracing/raytracing-algorithm-in-a-nutshell> [18 Aprilie 2022]
* <https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-ray-tracing/implementing-the-raytracing-algorithm> [18 Aprilie 2022]
* <https://github.com/ercang/raytracer-js> [22 Aprilie 2022]
* <https://www.scratchapixel.com/lessons/3d-basic-rendering/minimal-ray-tracer-rendering-simple-shapes/minimal-ray-tracer-rendering-spheres> [2 Mai 2022]
* <https://www.scratchapixel.com/lessons/3d-basic-rendering/minimal-ray-tracer-rendering-simple-shapes/ray-plane-and-ray-disk-intersection> [2 Mai 2022]
* <https://github.com/iechevarria/ray-tracer-js> [8 Mai 2022]

**BIBLIOGRAFIE – IMAGINI PREZENTARE POWERPOINT**

* <https://images.anandtech.com/doci/15013/14%20-%20On_678x452.jpg>
* <https://upload.wikimedia.org/wikipedia/commons/thumb/8/83/Ray_trace_diagram.svg/1200px-Ray_trace_diagram.svg.png>
* <https://upload.wikimedia.org/wikipedia/commons/1/1a/D%C3%BCrer_-_Man_Drawing_a_Lute.jpg>
* <https://preview.redd.it/1j4y3hz9mr131.png?width=960&crop=smart&auto=webp&s=1da400830ad38402a84e6c763d7fcdc473041196>
* <https://logolook.net/wp-content/uploads/2021/06/Minecraft-Logo.png>
* <https://assets2.rockpapershotgun.com/control1.jpg/BROK/resize/880%3E/format/jpg/quality/80/control1.jpg>
* <https://cutewallpaper.org/24/control-png/control-remedy.png>
* <https://images5.alphacoders.com/638/thumb-1920-638820.jpg>
* <https://image.api.playstation.com/vulcan/img/rnd/202010/2122/yGwOrNzL0tM7ENOWPVXKPghP.png>
* <https://cdnb.artstation.com/p/assets/images/images/013/585/075/large/abdelrahman-kubisi-main-final.jpg?1540281069>
* <https://cdna.artstation.com/p/assets/images/images/013/585/076/original/abdelrahman-kubisi-process.gif?1540280812>
* <http://assets.sbnation.com/assets/2820593/pixar-before-after.jpg>
* <https://www.candyrific.com/res/uploads/brands/logos/color/Monsters-U-logo.png>
* <https://i.pcmag.com/imagery/articles/04Er6LW8YPsfh2pmHEghbK7-5.fit_lim.size_960x.jpg>
* <https://upload.wikimedia.org/wikipedia/commons/3/32/Shadow_of_the_Tomb_Raider.png>
* <https://www.scratchapixel.com/images/upload/ray-tracing-refresher/rt-setup2.png>?
* <https://www.mauriciopoppe.com/images/ray-tracing!orthographic.jpg>
* <https://www.scratchapixel.com/images/upload/shading-intro2/shad2-globalillum5.png>?
* <https://static.techspot.com/articles-info/1888/images/2019-08-11-image-8-p_1100.webp>
* <https://www.scratchapixel.com/images/upload/introduction-to-ray-tracing/lighttoeyebounce.png>
* <https://www.scratchapixel.com/images/upload/introduction-to-ray-tracing/tracefromeyetolight.gif>
* <https://mathinsight.org/media/image/image/vector_parallelogram_law.png>
* <https://www.pngall.com/wp-content/uploads/8/Metal-Ball-PNG-Free-Download.png>
* <https://cdn.picpng.com/arrow/arrow-left-back-backwards-sign-106696.png>
* <https://upload.wikimedia.org/wikipedia/commons/thumb/f/f9/Plane_equation_qtl3.svg/1200px-Plane_equation_qtl3.svg.png>
* <https://www.scratchapixel.com/images/upload/ray-simple-shapes/plane.png>?
* <https://upload.wikimedia.org/wikipedia/commons/thumb/7/7e/Sphere_-_monochrome_simple.svg/1200px-Sphere_-_monochrome_simple.svg.png>
* <https://upload.wikimedia.org/wikipedia/commons/6/67/Line-Sphere_Intersection_Cropped.png>