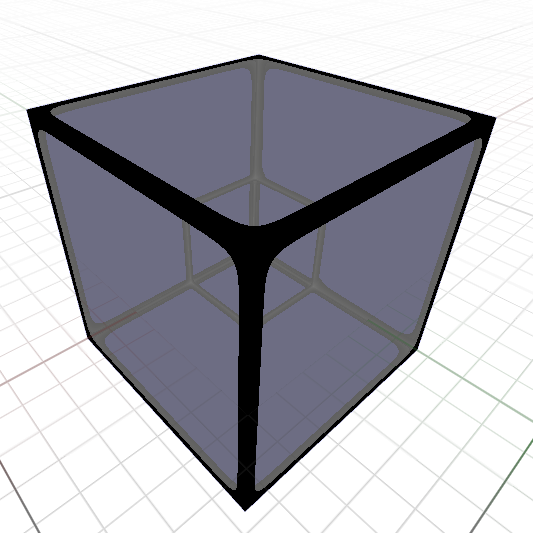
Colegiul Național “Iancu de Hunedoara”

**Tesseract**



Profesor coordonator: Student:

Manuela Diaconescu Adrian-Marian Popoviciu

2018

**Cuprins**

* Argument.............................................................................2
* Capitolul I „4 dimensiuni spațiale”......................................3
* Capitolul II „Aplicația”........................................................7
* Capitolul III „Sursa”............................................................9
* Bibliografie........................................................................15

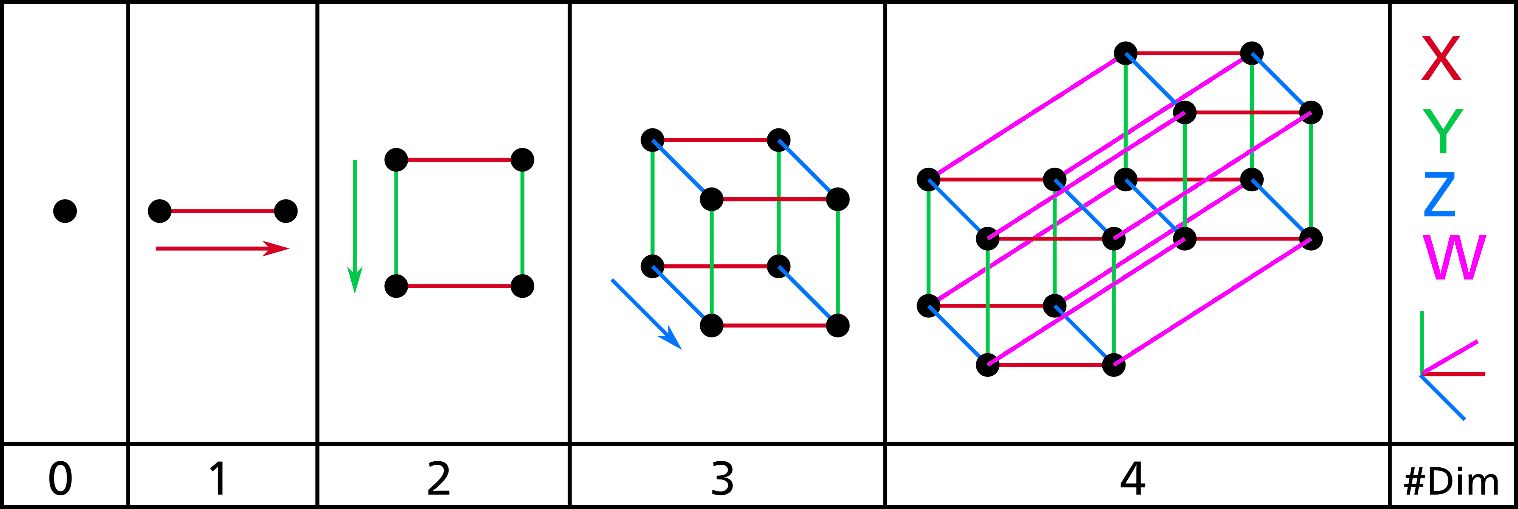
**Argument**

Am ales această temă datorita fascinației pentru generarea procedurală și geometria supra dimensională.

**Capitolul I**

**Introducere în geometria hiperspațială:**

În matematică, se poate construi o axa N perpendiculară pe celelalte N-1 axe ortogonale. Introducerea unei a patra axe perpendiculare pe spațiul tridimensional rezultă în definirea unei lungimi hiperspațiale reprezentată prin axa de coordonate W.



În spațiu 0 dimensional, figura geometrică este un punct.

În spațiu 1 dimensional, figura geometrică este un segment de dreapta obținut printr-un punct paralel.

În spațiu 2 dimensional, figura geometrică este un pătrat obținut printr-o dreaptă paralela.

În spațiu 3 dimensional, figura geometrică este un cub obținut printr-un pătrat paralel.

În spațiu 4 dimensional, figura geometrică este un tesseract obținut printr-un cub paralel.

Logica de formare a figurilor geometrice prin paralelism este aplicată și pentru restul corpurilor N dimensionale.

**Proiecții:**

Pentru a vizualiza un corp 4 dimensional in spațiu 3 dimensional, trebuie să preluăm umbra acestui corp și să o afișăm în spațiu 3 dimensional. Umbra unui obiect 4 dimensional este un obiect 3 dimensional, la fel cum prin analogie, umbra unui obiect 3 dimensional este un obiect 2 dimensional.

Folosind această matrice de proiecție, putem vizualiza obiectul 4 dimensional în spațiu 3 dimensional fără a oferi perspectiva celei de a patra dimensiuni spațiale. Prin analogie, putem vizualiza pătratul unui cub în spațiu 2 dimensional, fără influența celei de a treia dimensiuni spațiale.

**Proiecții stereografice:**

Aceasta este o matrice de proiecție stereografică. Oferă proiecția obiectului geometric în spațiu 3 dimensional cu perspectiva influențată de a patra axă dimensională.

Pentru a obține perspectivă, trebuie sa afișăm umbra acestui obiect 4 dimensional. Lw este locația punctului de lumină pe axa W din care trebuie sa scădem coordonata W al fiecărui punct proiectat.

**Rotații:**

În spațiu 3 dimensional, rotația este realizată în jurul unei axe. În spațiu 4 dimensional, rotația este realizată in jurul unui plan si o calculăm cu ajutorul a 6 matrice de rotație.

**Capitolul II**

**Cerințe de sistem:**

* OS: Windows 7 32-bit sau Android
* CPU: Quad-core, 1GHz sau mai bun
* RAM: 128MB

**Cerințe software:**

* DirectX 11
* .NET Framework 4.0
* Visual Studio 2015 C++ Redistributables

Instalarea software-urilor necesare se poate realiza individual sau prin rularea executabilului **UE4PrereqSetup\_x86** găsit în interiorul directorului \Windows x86\Engine\Extras\Redist\en-us

**Ghid de utilizare:**

În capturile de ecran avem interfața prezentată utilizatorului.

Interfața de creare (stânga) este prezentată la deschiderea aplicației sau după interacțiunea cu butonul Restart. Această interfață are ca rol preluarea dimensiunilor dorite de utilizator. Butonul Creaza transmite valorile dorite și construiește corpul geometric de dimensiunea dorită.

Interfața de utilizare (dreapta) este prezentată după crearea corpului geometric. Butoanele Culori si Proiecteaza au ca rol afișarea și ascunderea meniurilor extinse de culori, respectiv proiecție. Butonul Restart permite utilizatorului sa recreeze corpul geometric.

Meniul de culori permite utilizatorului să modifice valoarea RGB al fiecărui cub ce formează obiectul geometric pentru o vizualizare mai ușoară a rotațiilor.

Meniul de proiecție permite utilizatorul să aleagă viteza de rotație în grade per secundă, planul de rotație, tipul proiecției si locația sursei de lumină pe axa W.

**Capitolul III**

**C++:**

* **ATESSERACT**

Este apelată metoda *Construct()* care inițializează variabilele

utilizate de corpul geometric. Acestea sunt precalculate excepție pentru dimensiunile introduse de utilizator.

Metoda *Project()* realizează proiecția din 4D in 3D.



Metoda *UpdateColors()* actualizează culorile fiecărui vertex cu noul set de culori introduse de utilizator.



Metoda *AddRotation()* rotește obiectul geometric pe planul de rotație dorit și numărul de grade introdus de utilizator. Doar ramura de rotație dubla este exemplificată în codul următor!



* **UUI**

Această funcție inițializează dimensiunile obiectului și realizează proiecția.



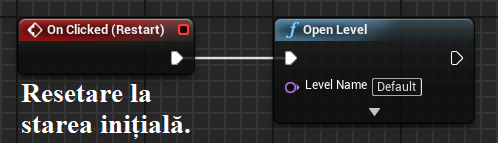
Această funcție realizează conversia din RGBA liniar in RGBA 8-bit.



**Blueprints Visual Scripting:**

**A screenshot of a computer

Description generated with very high confidence**

****

**A screen shot of a video game

Description generated with high confidence**

**A screenshot of a computer

Description generated with high confidence**

**A screenshot of a computer

Description generated with high confidence**

**A screenshot of a computer

Description generated with very high confidence**

**A screenshot of a computer

Description generated with high confidence**

**Bibliografie**

* “Computational geometry in C” de Joseph O’Rourke
* “General n-Dimensional Rotations” de Antonio Aguilera si Ricardo Pérez-Aguila
* Wikipedia: <https://en.wikipedia.org/wiki/Tesseract>