Video Laborator 5: https://youtu.be/HOv-P8QnEAA **Autor**: Florin Iancu

Spațiul Obiect

Spațiul obiect mai este denumit și **SPAȚIUL COORDONATELOR LOCALE**.

Pentru a putea lucra mai eficient și a reutiliza obiectele 3D definite, în general fiecare obiect este definit într-un sistem de coordonate propriu. Obiectele simple sau procedurale pot fi definite direct din cod însă majoritatea obiectelor utilizate în aplicațiile 3D sunt specificate în cadrul unui program de modelare precum 3D Studio Max, Maya, Blender etc. Definind independent fiecare obiect 3D, putem să îi aplicăm o serie de transformări de rotație, scalare și translație pentru a reda obiectul în scena 3D. Un obiect încărcat poate fi afișat de mai multe ori prin utilizarea unor *matrici de modelare*, câte una pentru fiecare instanță a obiectului inițial, ce mențin transformările 3D aplicate acestor instanțe.

În general, fiecare obiect 3D este definit cu centrul (sau centrul bazei ca în poza de mai jos) în originea propriului său sistem de coordonate, deoarece în acest fel pot fi aplicate mai ușor transformările de modelare. Astfel, rotația și scalarea față de centrul propriu sunt efectuate întotdeauna față de origine.

Spațiul Lume Spațiul lume sau **SPAȚIUL COORDONATELOR GLOBALE** este reprezentat prin intermediul *matricei de modelare*, aceeași despre care s-a vorbit mai sus. Matricea se obține printr-o serie de rotații, scalări și translații. Prin înmulțirea fiecărui vertex al unui obiect (mesh 3D) cu această matrice, obiectul va fi mutat din spațiul local în spațiul lume, adică se face trecerea de la coordonate locale la coordonate globale.

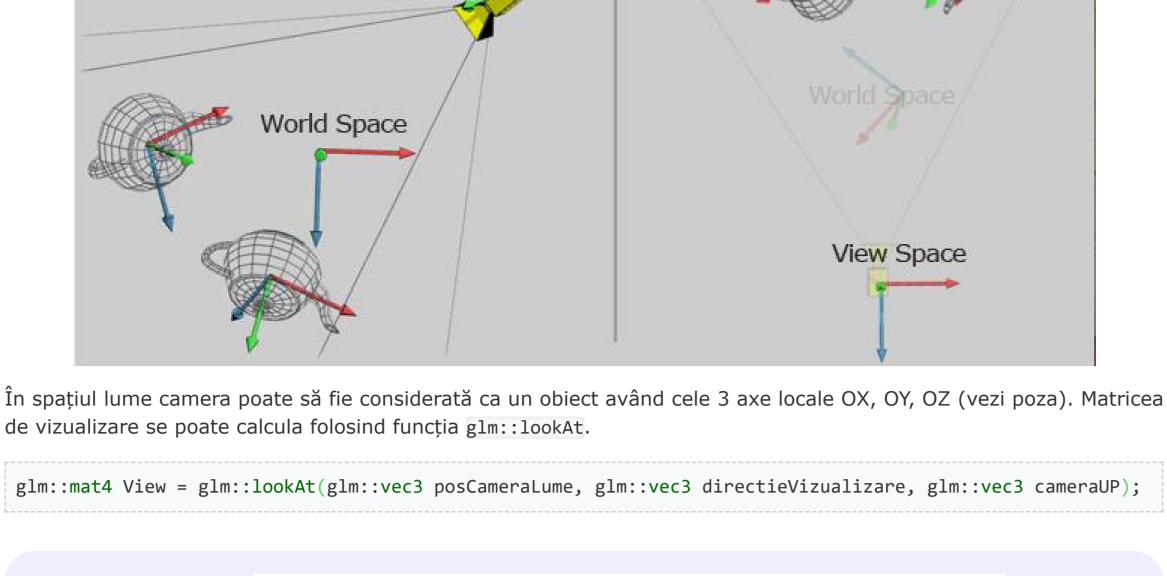
Folosind matrici de modelare diferite putem amplasa un obiect în scenă de mai multe ori, în locații diferite, cu rotație și scalare diferită dacă este necesar. Un exemplu este prezentat în scena de mai jos.



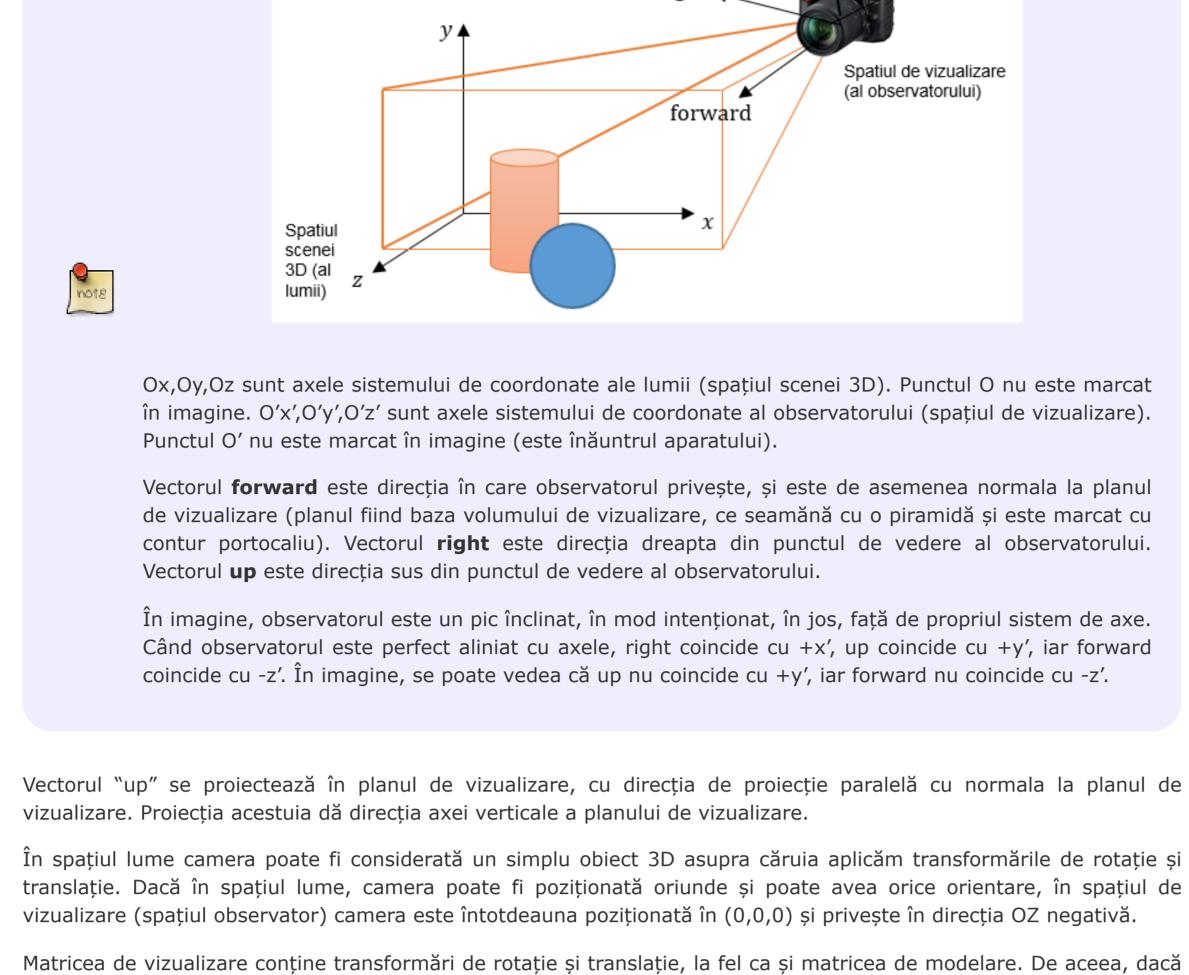
Matricea de modelare poziționează obiectele în scenă, în spațiul lume. Dar o scenă poate fi vizualizată din mai multe puncte de vedere. Pentru aceasta există transformarea de vizualizare. Dacă într-o scenă avem mai multe obiecte,

fiecare obiect are o matrice de modelare diferită (care l-a mutat din spațiul obiect în spațiul lume), însă toate obiectele au aceeași matrice de vizualizare. Transformarea de vizualizare este definită pentru întreaga scenă.

Spațiul de vizualizare sau **SPAȚIUL CAMEREI** este reprezentat de **matricea de vizualizare**.



right $\stackrel{\chi'}{\blacktriangleleft}$



around it." - Futurama

pentru vizualizarea întregii scene din punctul de vedere al camerei.

Totuși, cele două matrici au scopuri diferite. Una este folosită pentru poziționarea obiectelor în scenă, iar cealaltă

ținem scena pe loc și mutăm camera, sau dacă ținem camera pe loc și rotim/translatăm scena, obținem același efect:

"The engines don't move the ship at all. The ship stays where it is and the engines move the universe

Exemplu: Dacă vrem să ne uităm pe axa OX(lume) din poziția (3, 5, 7) codul corespunzător pentru funcția glm::lookAt este: glm::lookAt(glm::vec3(3, 5, 7), glm::vec3(1, 0, 0), glm::vec3(0, 1, 0));

Spațiul de Proiecție După aplicarea transformării de vizualizare, în spațiul de vizualizare, camera se află în origine și privește înspre -OZ. Pentru a putea vizualiza pe ecran această informație este necesar să se facă proiecția spațiului vizualizat de cameră într-un spațiu 2D. Cum spațiul vizibil al camerei poate fi de diferite feluri, cel mai adesea trunchi de piramida (proiecție perspectivă) sau paralelipiped (proiecție ortografică), în OpenGL este necesară trecerea într-un spațiu final numit spațiu de proiecție ce reprezintă un *cub* centrat în origine cu dimensiunea 2, deci coordonatele X, Y, Z între -1 și +1.

Din spațiul de proiecție este foarte ușor matematic să obținem proiecția finală 2D pe viewport fiind nevoie doar să

mapăm informația din cubul [-1,1] scalată corespunzător pe viewport-ul definit de aplicație. Matricea de Proiecție

ecran.

proiecție.

proiecție asupra geometriei din scenă

-1,-1,1)

perspectivă (dreapta)

viewport (**dreapta**)

din scenă cu cele 3 matrici calculate.

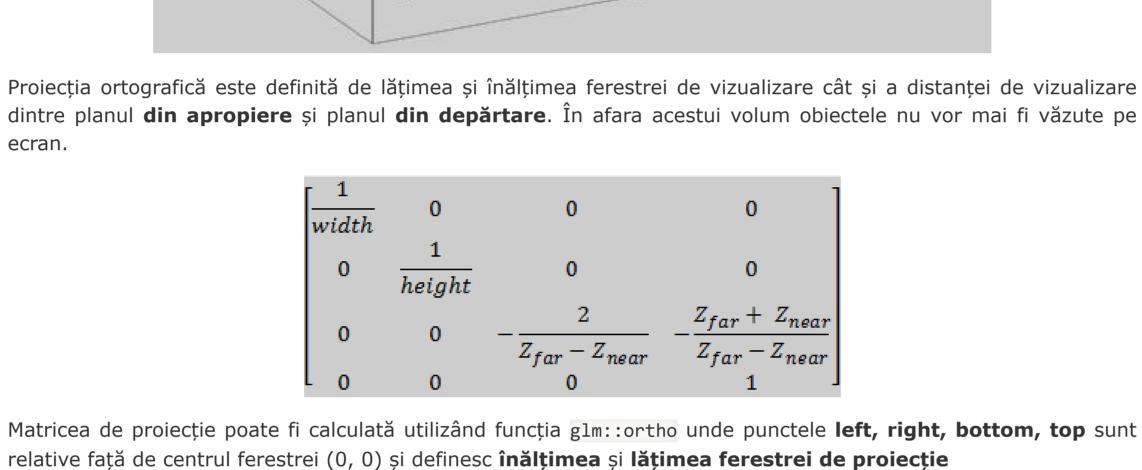
Trecerea din spațiul de vizualizare în spațiul de proiecție se face tot utilizând o matrice, denumită matrice de proiecție, calculată în funcție de tipul de proiecție definit. Biblioteca GLM oferă funcții de calcul pentru cele mai utilizate 2 metode de proiecție în aplicațiile 3D, anume: proiecția perspectivă și ortografică Datele (vertecșii din spațiul de vizualizare) sunt înmulțite cu **matricea de proiecție** pentru a se obține pozițiile corespunzătoare din spațiul de proiecție. Proiecția Ortografică

În proiecția ortografică observatorul este plasat la infinit. Distanța până la geometrie nu influențează proiecția și

deci nu se poate determina vizibil din proiecție. Proiecția ortografică păstrează paralelismul liniilor din scenă.

width

height



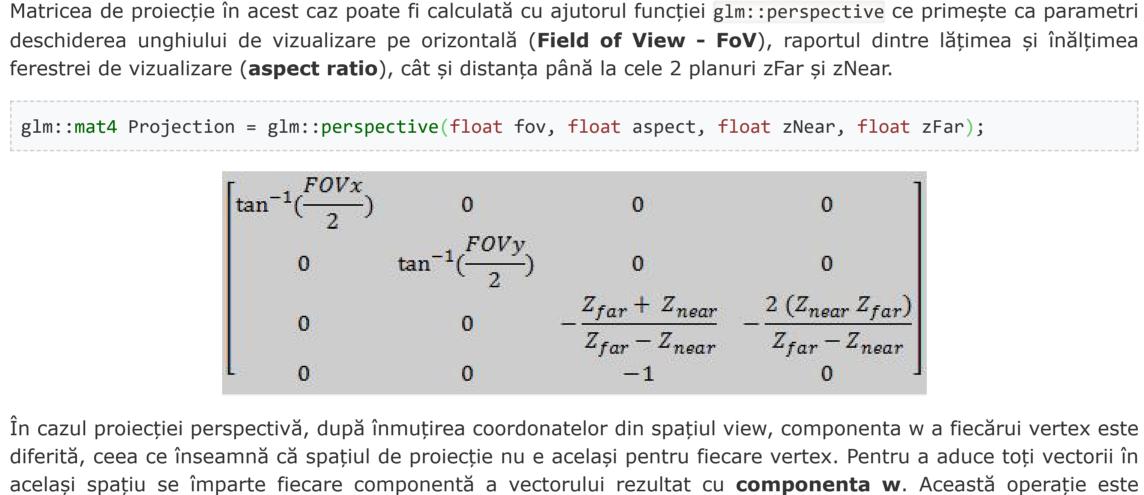
Proiecția Perspectivă

Proiecția perspectivă este reprezentată de un trunchi de piramidă (frustum) definit prin cele 2 planuri, cel din

apropiere și cel din depărtare, cât și de deschiderea unghiurilor de vizualizare pe cele 2 axe, OX și OY. În

proiecția perspectivă distanța până la un punct din volumul de vizualizare influențează proiecția.

glm::mat4 Projection = glm::ortho(float left, float right, float bottom, float top, float zNear, float z



realizată automat de procesorul grafic, în cadrul unei aplicații fiind nevoie doar de înmulțirea cu matricea de

Volum de vizualizare perspectivă (stânga) și rezultatul obținut (dreapta) în urma aplicării transformării de

-1,-1,1)

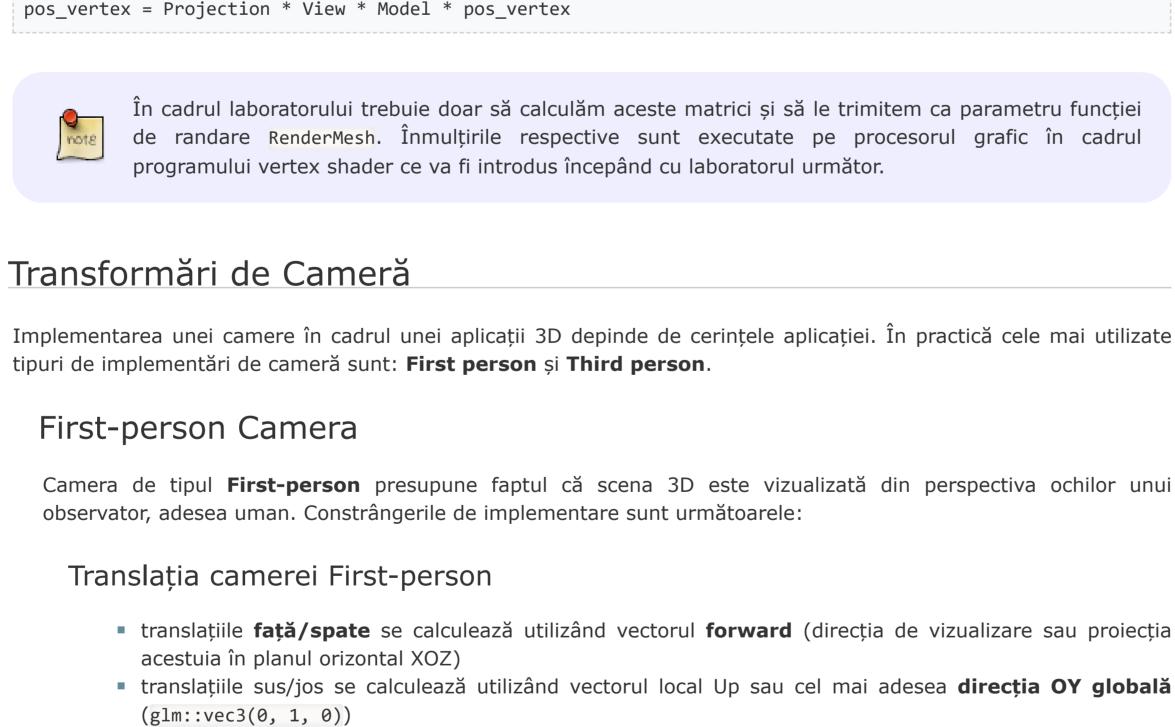
Spațiul Coordonatelor de Dispozitiv Normalizate (NDC) După aplicarea transformărilor de Modelare, Vizualizare și Proiecție iar apoi divizarea cu W a vectorilor, se obține spațiul de coordonate normalizate (NDC) reprezentat de un CUB centrat în origine (0, 0, 0) cu latura 2. Informația din acest cub se poate proiecta foarte ușor pe orice suprafață 2D de desenare definită de utilizator.

Exemplu vizualizare spațiu NDC din direcția camerei (stânga) și proiecția corespunzătoare pentru un anumit

Aplicarea trasformărilor de Modelare, Vizualizare și Proiecție se face prin înmulțirea fiecărui vertex al geometriei

Aplicarea Transformărilor de Modelare, Vizualizare și Proiecție

Exemplu rezultat al proiecției în coordonate dispozitiv normalizate (NDC). Proiecție ortografică (stânga),



• translațiile dreapta/stânga se calculează folosind vectorul local right (ce se poate obține și prin operația

de cross product între vectorii forward și up) sau folosind proiecția acestuia pe planul orizontal XOZ

rotația sus/jos se poate face rotind vectorii forward respectiv up în jurul vectorului axei OX adică

rotațiile se fac păstrând observatorul pe loc și modificând direcția în care privește acesta

pentru rotația **stânga/dreapta**, vectorii **forward** respectiv **right** se pot calcula prin aplicarea transformării de rotație în jului axei OY globale. Se poate roti și în jurul axei OY locale (vectorul up), însă în general nu prea are aplicabilitate practică vectorul up se poate recalcula folosind cross product între right și forward forward = RotateWorldOY(angle) * forward;

posCamera = posCamera + glm::normalize(direction) * distance;

Rotația camerei First-person

right = RotateWorldOY(angle) * right;

vectorul right (right rămâne constant)

forward = RotateLocalOX(angle) * forward;

astfel:

glm::vec3 forward = ...

// get the rotate vec4 vector

up = glm::cross(right, forward);

up = glm::cross(right, forward);

• rotationAxis este axa față de care rotim. În cazul nostru pentru rotația față de OX este vectorul right, pentru rotația față de OZ este vectorul forward, sau glm::vec3(0, 1, 0) pentru rotația față de **OY global** • întrucât vectorii utilizați sunt glm::vec3 când facem înmulțirea va trebui să construim un vector de 4 componente ca să putem înmulți cu matricea de 4×4. Puteți construi vectorul

glm::mat4 = glm::rotate(glm::mat4 model, float angle, glm::vec3 rotationAxis);

primul parametru reprezintă o matrice de modelare asupra căreia aplicăm

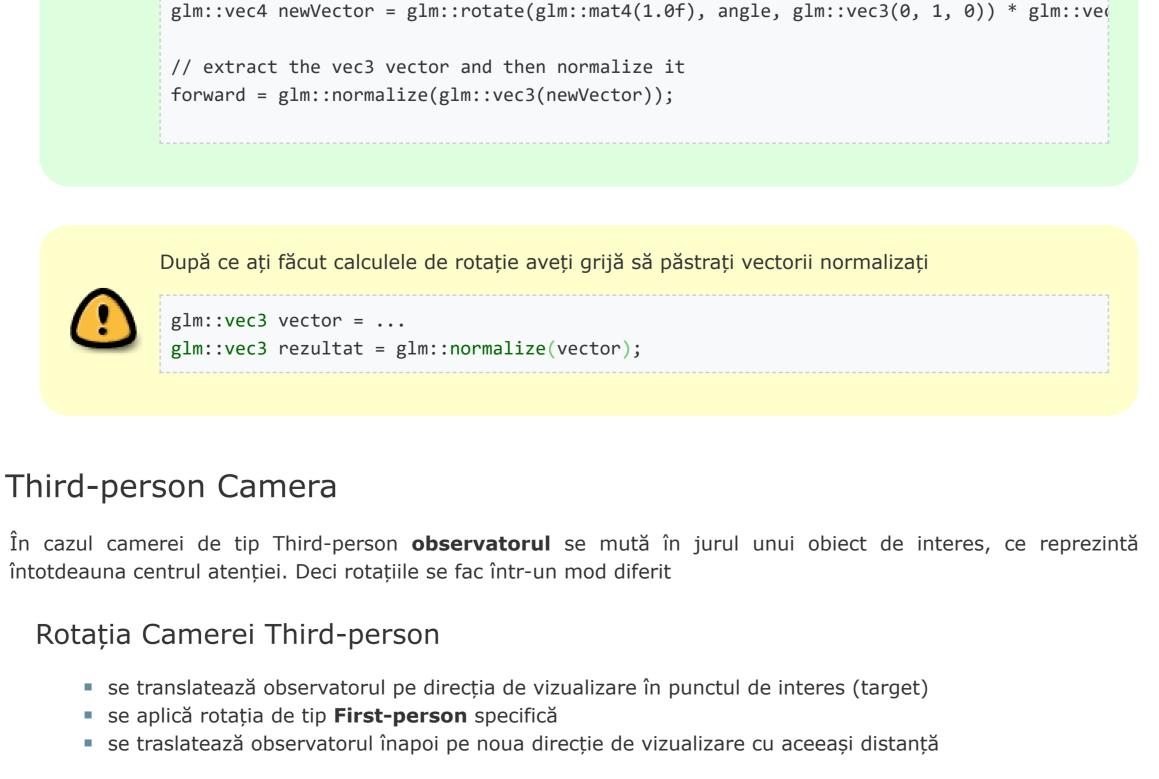
Dacă vrem să rotim vectorul "forward" în jurul axei OY globale atunci facem astfel:

transformarea specificată. Atunci când nu avem o transformare precedentă se pornește

Matricile de rotație necesare se pot calcula folosind funcția glm::rotate

de la matricea identitate glm::mat4(1.0f)

glm::vec4 newVec = glm::vec4(forward, 1.0);



În laborator aveți variabila distanceToTarget care reține distanța până la punctul față de care rotim

translația camerei în mod corespunzător.

Translația Camerei Third-person

Poziția camerei depinde de poziția punctului de interes. Astfel, mișcarea punctului de interes va determina și

1. Descarcăți framework-ul de laborator 2. Să se implementeze camera de tip First Person (fișierul <u>LabCamera.h</u>) 3. Să se implementeze camera de tip Third Person (fișierul <u>LabCamera.h</u>) 4. Să se completeze funcțiile de translație ale camerei din Laborator5::0nInputUpdate()

7. Schimbare proiecție perspectivă/ortografică

• tasta O face trecerea în proiecție ortografică

tasta P face trecerea în proiecție perspectivă

8. Să se modifice FoV-ul camerei în cazul proiecției persepective

• folosiți 2 taste pentru a modifica pozitiv și negativ FoV-ul

Cerințe laborator

Old revisions

- 5. Să se completeze funcțiile de rotație ale camerei din Laborator5::0nMouseMove() 6. Să se deseneze încă 2 obiecte în scena 3D având rotația/scalarea/translația diferite aveți grijă să setați matricea de modelare de fiecare dată înainte de desenare utilizați glm::translate(), glm::rotate() și glm::scale() pentru a construi o matrice de modelare pentru fiecare object
- se va folosi OnInputUpdate() 9. Să se modifice lățimea și/sau înălțimea ferestrei de proiecție în cazul proiecției ortografice se va folosi OnInputUpdate()

(CC) BY-SA CHIMERIC DE WSC CSS DOKUWIKI OF GET FIREFOX RSS XML FEED WSC XHTML 1.0

egc/laboratoare/05.txt · Last modified: 2020/11/12 14:19 by victor.asavei

Media Manager Back to top

Recent changes Nogin Search **Info curs**

Elemente de Grafică pe

Calculator Infographie

Laboratoare Laboratorul 01 Laboratorul 02

TBA

- Laboratorul 03 Laboratorul 04 Laboratorul 05
- Laboratorul 06 Laboratorul 07 Prezentare Tema 1 Laboratorul 08
- Prezentare Tema 2 Recuperări laborator Prezentare Tema 3

Laboratorul 09

Vacanţă

- Resurse: Redare text Teme Regulament General
- Tema 1 Bow and Arrow ■ Tema 2 - Skyroads
 - Dispozitiv Normalizate (NDC) Aplicarea Transformărilor de Modelare, Vizualizare și Proiecție
 - First-person Camera Translaţia camerei First-person Rotația camerei
 - Third-person Camera Rotația Camerei Third-person Translaţia Camerei
 - Third-person