

Grafică pe calculator

(MLR5060)

[3D] Computer graphics

<http://www.cs.ubbcluj.ro/~per> → [Cursuri](#)



http://www.cs.ubbcluj.ro/~per/Web_Page/Cursuri.htm

Semestrul I	<p>28.09.2015 – 20.12.2015 ~ Ad 21.12.2015 – 03.01.2016 ~ Va 04.01.2016 – 17.01.2016 ~ Ad 18.01.2016 – 07.02.2016 ~ Se 08.02.2016 – 14.02.2016 ~ Va 15.02.2016 – 21.02.2016 ~ Re</p>	<p>→ Grafică pe Calculator Util. Soft. Spec. (EFS~SPM, IFR) Proiect Colectiv Fundamentele Programării (Sem., Lab.) Decision Support Systems Proiect Individual Progr. Calc.</p>
-------------	--	---

Date despre disciplină

Denumirea disciplinei		Grafică pe calculator (MLR5060)					
Titularul act. de curs		Lect. Dr. Vasile Prejmerean					
Titularul act. de sem.							
An de st.	3	Sem.	5	Tip eval.	C	Reg. disc.	Optională

Număr ore pe săpt.	4	curs	2	laborator	1 lab + 1 pr.	
Total ore	56	curs	28	laborator	28	
Distribuția fondului de timp:					ore	
Studiul după manual, suport de curs, bibliografie și notițe					7	
Doc. supl. bibl., pe platformele electr. de spec. și pe teren					7	
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					16	
Tutoriat					7	
Examinări					7	
3.7 Total ore studiu individual	44					
3.8 Total ore pe semestru	100					
3.9 Numărul de credite	4					

Obiectivele disciplinei

Obiectivul general	<ul style="list-style-type: none">• Cunoașterea elementelor necesare reprezentărilor grafice;• Înțelegerea metodelor de creștere a realismului imaginilor.
Obiectivele specifice	<ul style="list-style-type: none">• Formarea deprinderilor de realizare a aplicațiilor grafice;• Cunoașterea unor algoritmi, metode și tehnici utilizate în etapele de trecere de la modelare la reprezentare (imagine).

Evaluarea

Tip activitate	Criterii de evaluare	Evaluare	Ponderi
Curs	Noțiuni teoretice pt. realizarea aplicațiilor și în rez. de probleme	Examen scris	40% ~ 30%
Laborator si Proiect	Crearea și gestiunea obiectelor 3D	Laborator	30%
	Realizarea unui referat și a unui proiect pe o temă aleasă	Referat Proiect	10% 20% ~ 30%
<ul style="list-style-type: none">• Toate notele trebuie să fie cel puțin 5.			

Continutul cursului

1. Bazele programării în grafica interactivă.
2. Transformări geometrice uzuale în grafica 2D și 3D. Vizualizarea 3D.
3. Algoritmi raster (segmente, poligoane, zone).
4. Modelarea obiectelor (solide) 3D. Modelare geometrică și ierarhii.
5. Modelarea și vizualizarea suprafețelor și curbelor 3D.
6. Eliminarea (supra)fețelor acoperite (nevăzute, ascunse).
7. Metode de creștere a realismului imaginilor (*lumini, umbre, pr. din, culori,...*)
8. Texturi (constante, variabile, aleatoare)
9. Utilizarea culorilor în grafic. Spatii de cul. *RGB, CIE XYZ - CIE-Lab, Luv, HSV, ...*
10. Deformarea obiectelor 3D. Reprezentarea parametrică a obiectelor 3D.
11. Stereografie - Anaglife
12. Stereografie - Stereograme, ...
13. Realizarea animației.
14. Grafică Web.

Continutul laboratorului

1. Facilități grafice în C#/Java/Python

2. Transformări uzuale în grafica 2D și 3D.

3. Vizualizarea 3D.

4. Utilizarea algoritmilor de tip raster.

5. Modelarea și vizualizarea dinamică a corpurilor.

6. Modelarea și vizualizarea dinamică a curbelor și a suprafețelor.

7-9. Creșterea realismului imaginilor

- Utilizarea surselor de lumină și realizarea umbrelor,
- Aplicarea texturilor pe obiecte
- Construirea imaginilor stereografice.

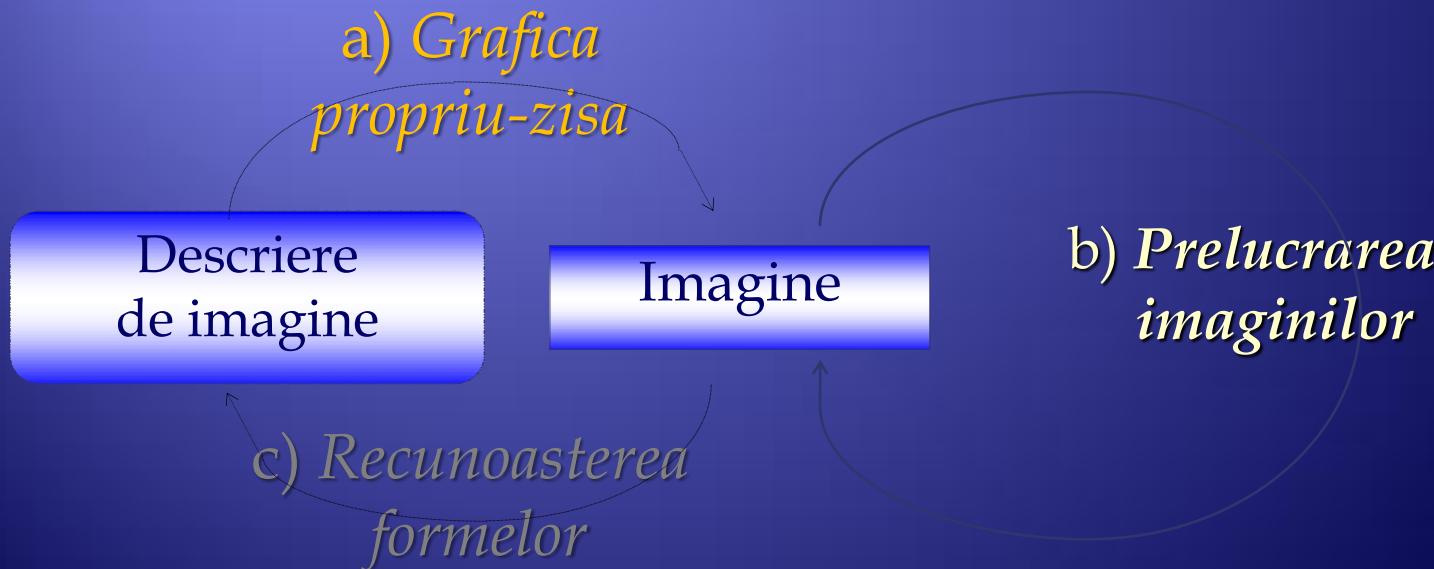
10. Realizarea animației (deformarea obiectelor), pagini Web.

11-13. Susțineri referate.

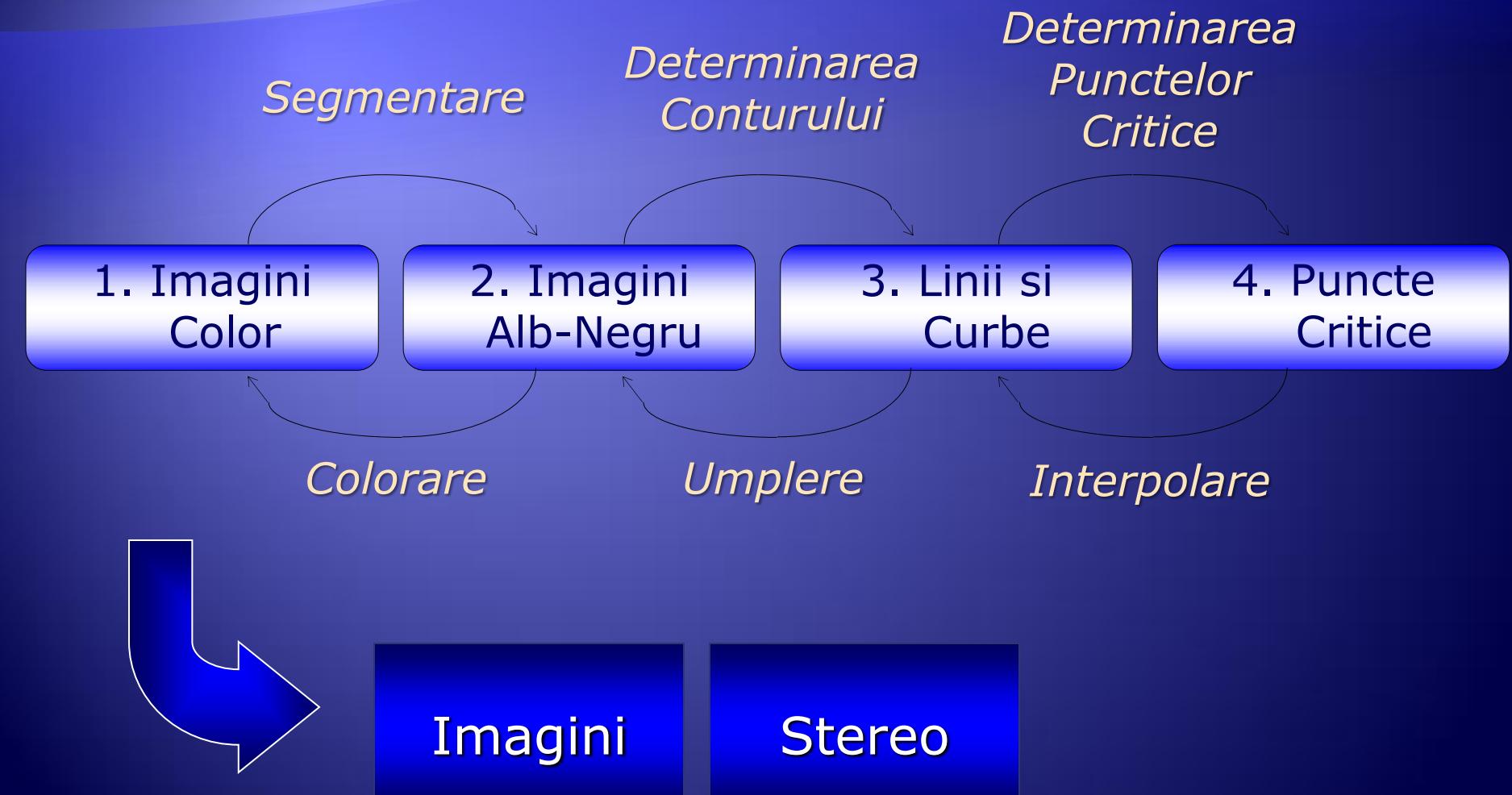
14. Predare proiecte.

Grafică:

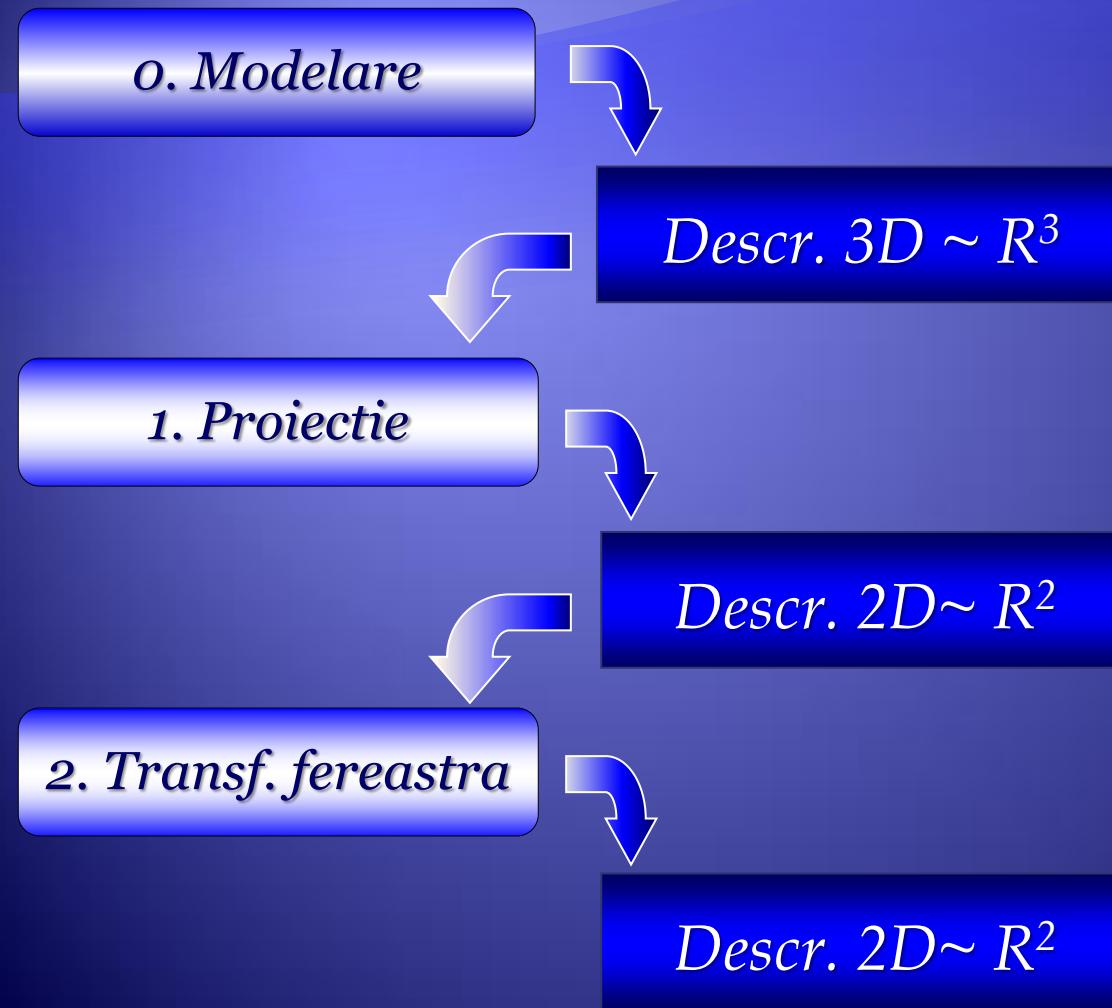
- a) Grafica propriu-zisa ~ 3D
- b) Prelucrarea imaginilor -2D
- c) Recunoasterea formelor -?



Clasificarea imaginilor



Etape in reprezentarea Ob. 3D



Spatiul 3D

*Fereastra Reală
Window*

*Fereastra Ecran
Viewport*

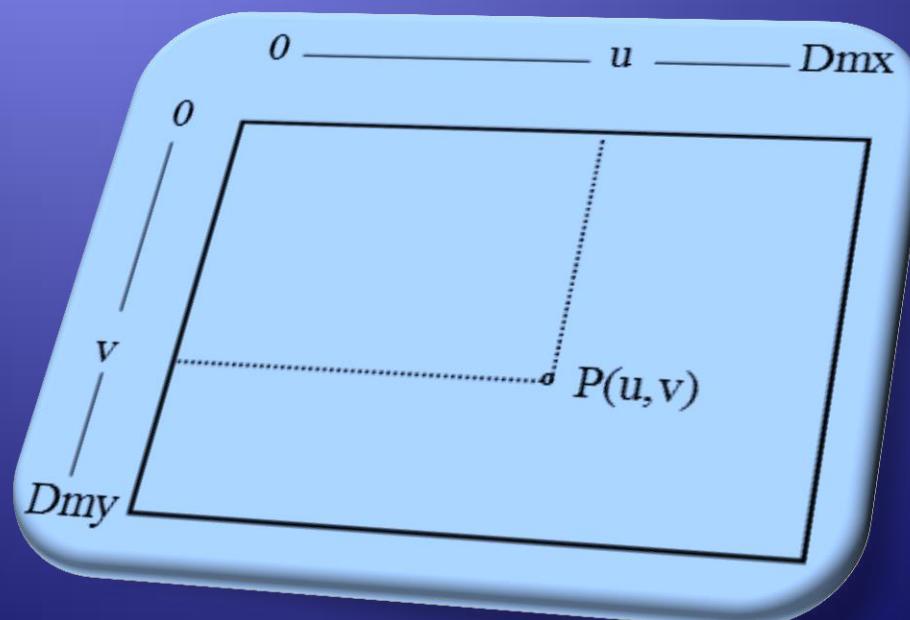
Ferestre

Ecranul/Fereastra poate fi considerat ca fiind format dintr-o matrice de puncte (pixeli).

Un punct de pe ecran poate fi aprins (colorat) într-o anumită culoare iar poziția sa este definită prin coordonatele sale u și v reprezentând coloana, respectiv linia, unde:

$0 \leq u < Dmx$, Dmx = dimensiunea maximă pe coloană, iar

$0 \leq v < Dmy$, Dmy = dimensiunea maxima pe linie

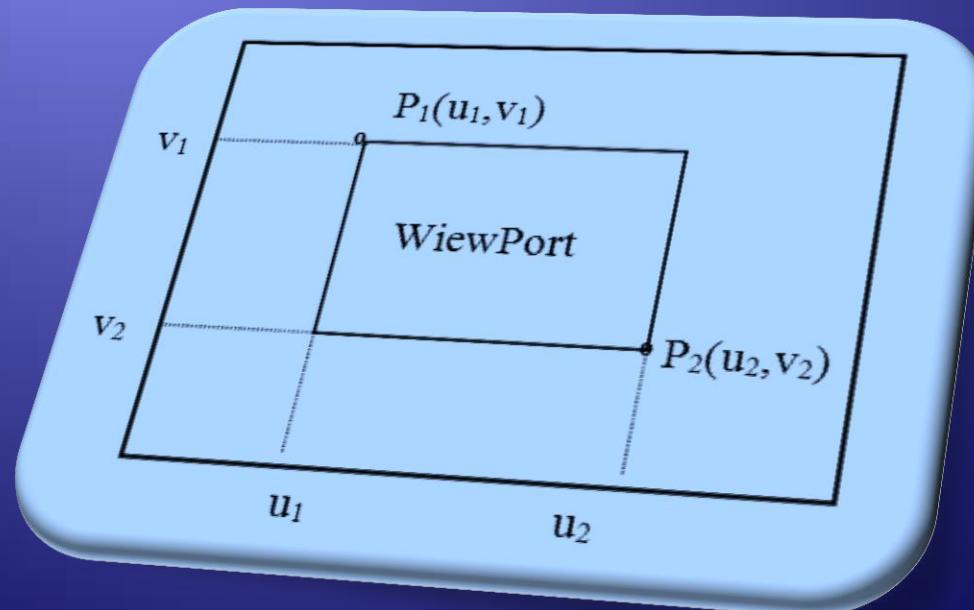


Fereastra fizică ~ ViewPort

Dacă dorim să trasăm pe ecran primitive grafice referite prin coordonate aflate într-un domeniu real, va trebui să aplicam o transformare a acestor coordonate, pentru că instrucțiunile grafice referă puncte de coordonate întregi din domeniul $[0, D_{mx}] \times [0, D_{my}]$.

Există posibilitatea de a realiza un desen doar pe o portiunea ecranului, pe un subdomeniu $[u_1, u_2] \times [v_1, v_2]$ al domeniului maxim $[0, D_{mx}] \times [0, D_{my}]$.

Acest subdomeniu îl vom numi **fereastra fizică (ViewPort)** și va fi precizat prin coordonatele ecran a două puncte diagonale opuse $P_1(u_1, v_1)$ și $P_2(u_2, v_2)$.



Desenarea figurilor din planul real

În fereastra ecran $[u_1, u_2] \times [v_1, v_2]$ (*ViewPort*) putem reprezenta puncte $P(x,y)$ dintr-un domeniul $[a,b] \times [c,d] \subset R^2$ numit fereastră reală (*Window*). Pentru a putea referi în instrucțiunile grafice un astfel de punct, va trebui să aplicăm o transformare de coordonate $P(x,y) \rightarrow M(u,v)$, care verifică egalitățile :

$$\frac{x-a}{b-a} = \frac{u - u_1}{u_2 - u_1}$$

$$\frac{y-d}{c-d} = \frac{v - v_1}{v_2 - v_1}$$

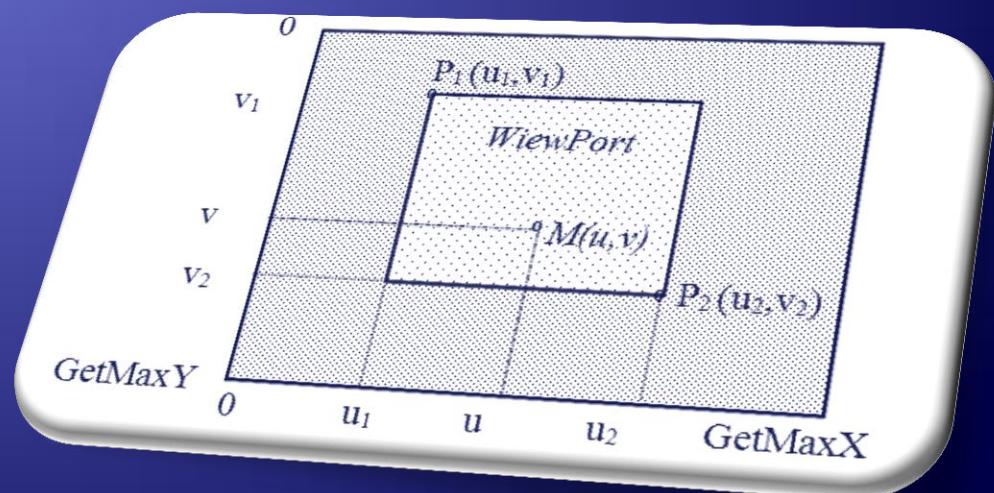
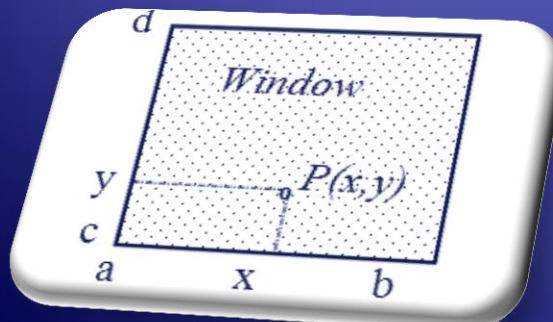
$$\frac{y-d}{c-d} = \frac{v - v_1}{v_2 - v_1}$$

$$\frac{y-d}{c-d} = \frac{v - v_1}{v_2 - v_1}$$

de unde:

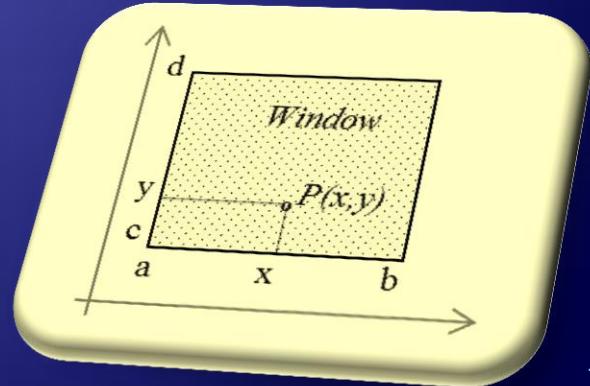
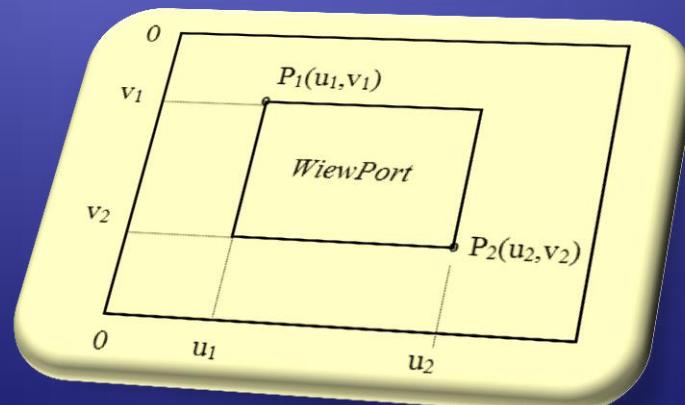
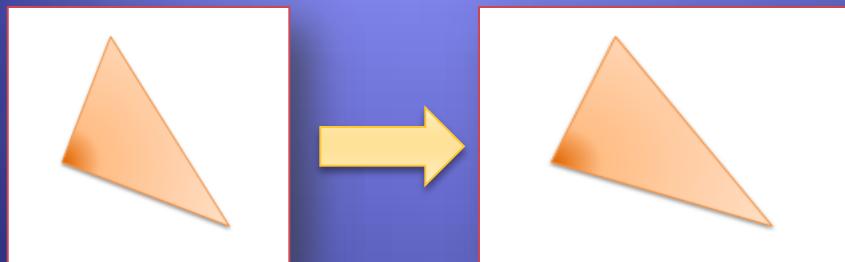
$$u := Round((x-a)*(u_2-u_1)/(b-a)) + u_1$$

$$v := Round((y-d)*(v_2-v_1)/(c-d)) + v_1.$$



Etape în reprezentări 2D

- a) Definire Fereastra Ecran : *Viewport* (u_1, v_1, u_2, v_2);
 - b) Definire Fereastra Reală : *Window* (a, d, b, c); // Dupa determinare!
// Atentie la Deformări! Geometrie, Proiectare, ...
 - c) Referire pixeli: $u(x), v(y)$;
 - *Line* ($u(x_1), v(y_1), u(x_2), v(y_2)$) → *Linie* (x_1, y_1, x_2, y_2);
sau *Segment* (P_1, P_2); // $P_i(x_i, y_i)$
 - ... *Pixel* (x, y, cul);
 - ... *Dreptunghi* (x_1, y_1, x_2, y_2);

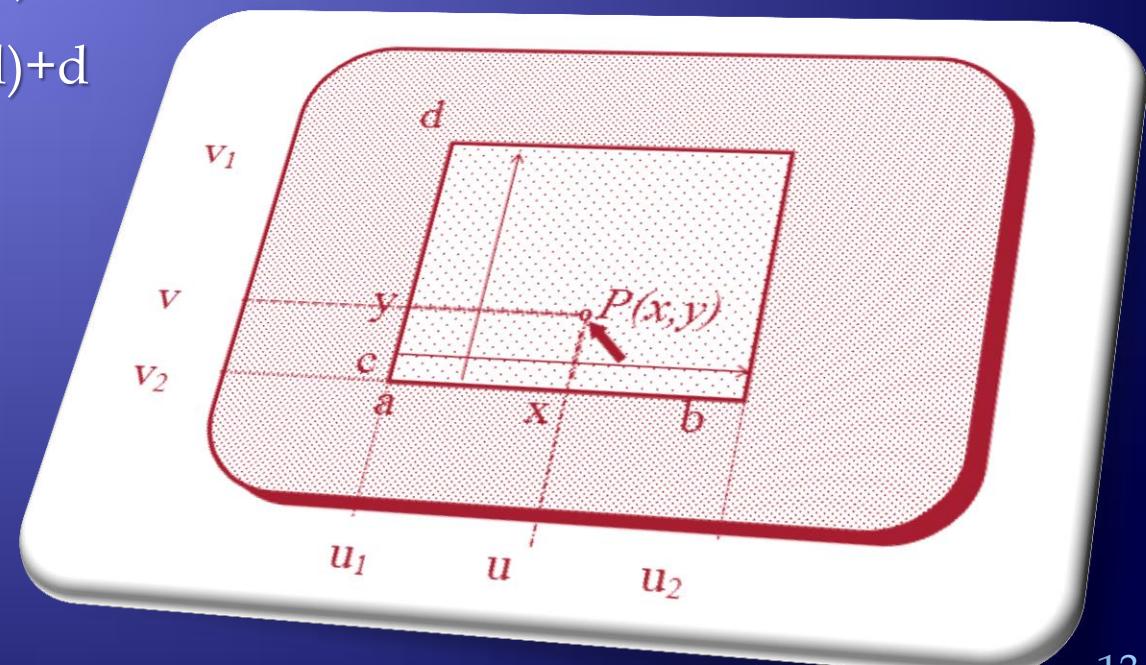


Interpretarea imaginilor de pe ecran

Uneori ne interesează cum putem afla coordonatele reale ale unui punct precizat printr-un pixel de pe ecran, deci problema inversă și anume transformarea din fereastra ecran în cea reală ($M(u,v) \rightarrow P(x,y)$). Acest lucru este necesar în situația în care avem reprezentat pe ecran un desen și ne interesează, de exemplu, coordonatele aproximative ale unui punct de pe ecran. Aceste transformări sunt următoarele :

$$u^{-1}: x := (u - u_1) / (u_2 - u_1) * (b - a) + a$$

$$v^{-1}: y := (v - v_1) / (v_2 - v_1) * (c - d) + d$$



Temă

1.  Să se realizeze o aplicație practică pentru o problemă (la alegere) de analiză matematică sau de geometrie plană (de exemplu o problemă de loc geometric) cu reprezentare grafică și interpretarea rezultatelor de pe ecran cu ajutorul mouse-lui.
2.  Să se realizeze o aplicație pentru a desena o histogramă a temperaturilor medii lunare dintr-un an (realizată cu dreptunghiuri).
3.  Să se realizeze reprezentarea grafică a unor curbe remarcabile cum ar fi astroïda care are ca ecuație:
$$x^{2/3} + y^{2/3} = a^{2/3},$$
Se poate realiza simplu dacă această curbă este descrisă de forma parametrică :
$$x = a \cdot \sin^3 t, \quad y = a \cdot \cos^3 t$$

... animație!

Bibliografie

1. ALBEANU, Gr., Grafica Fundamentali, Editura Universitatii Bucuresti, 200
2. DOGARU, D., Metode noi de lucru cu computer, Enciclopedica Bucuresti, 198
3. FOLEY, J.D., VAN DAM, A., Computer Graphics, Addison Wesley Publishing Comp., 1982
4. FOLEY, J. D., VAN DAM, A. Computer Graphics - Principles and Practice, Second Edition in C, Addison Wesley, 1996
5. HAWKINS, K., Astle, D. "OpenGL Programming Guide", Addison Wesley, 2004
6. HEARN, D., BAKER P., Computer Graphics - Principles and Practice, Addison Wesley, 2004
7. HEDINSON, R., PAVLIDIS, T Algorithms for Graphics, Springer-Verlag, 1982.
8. PETCU, D., CUCU, L., Principii de Grafica 3D, Editura Excelsior, Timisoara 1995.
9. POP, D., PETCU, D., Modelarea si simularea in informatica, Editura Eubaea, 2004, Colectia Matematica si Informatica
10. WOLFRAM, S., Mathematica: A System for Doing Mathematics by Computer. Addison Wesley Publ.Comp., 1992.
11. OpenGL Programming Guide, <http://www.opengl-tutorial.org/>
12. OpenGL Programming Guide, <http://www.opengl-tutorial.org/>
13. NeHe's OpenGL Tutorials, http://www.polytech.unice.fr/~buffa/cours/synthese_image/DOCS/Tutoriaux/Nehe/opengl.htm
14. A. Watt, 3D Computer Graphics, Addison-Wesley, Great Britain, 1993
15. A. Watt, 3D Computer Graphics, (3rd Edition), Amazon, 1999

