



**UNIVERSITATEA
TEHNICĂ**
DIN CLUJ-NAPOCA

PROIECT

*“GAC IN
AUTOMATICA”*

Universitatea Tehnica din Cluj-Napoca

Facultatea de Automatica si Calculatoare

*Sectia de Automatica si Informatica
Aplicata*

Student:
**Cristian-Andrei
Stoleru**

Grupa:
30122

Indrumator:
Asist. Ing. Dan Gota

Cuprins

Proiect 1: Piesa 2D – Autocad	2
Descriere produs	2
Autocad.....	4
Dwg	4
Tabel comenzi utilizate	5
Proiect 2: AutoLisp	7
Descriere program	7
Codul sursa	8
Imagini ce reprezinta functionarea functiilor	12
Functia vopsire	12
Functia lumini.....	13
Bibliografie	14
Proiect 3: Autocad P&ID	15
Descriere produs	15
Autocad P&ID.....	16
Dwg	16
Proiect 4: OpenGL	17
Obiectivul si descrierea codului	17
Codul sursa	18
PrintScreen fereasta „Cub-GAC”	24
Bibliografie	25

Proiect 1: Piesa 2D – Autocad

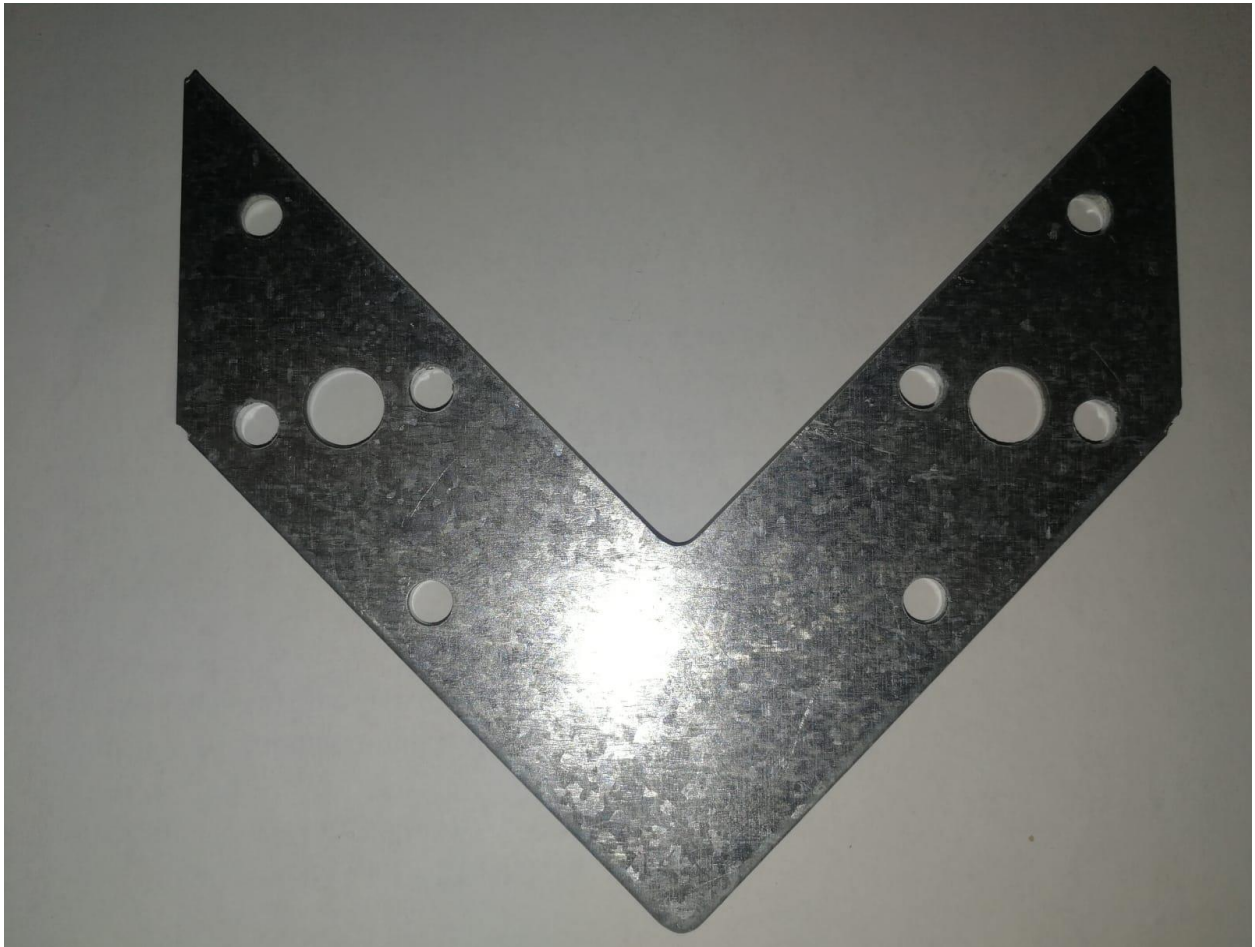
Descriere produs

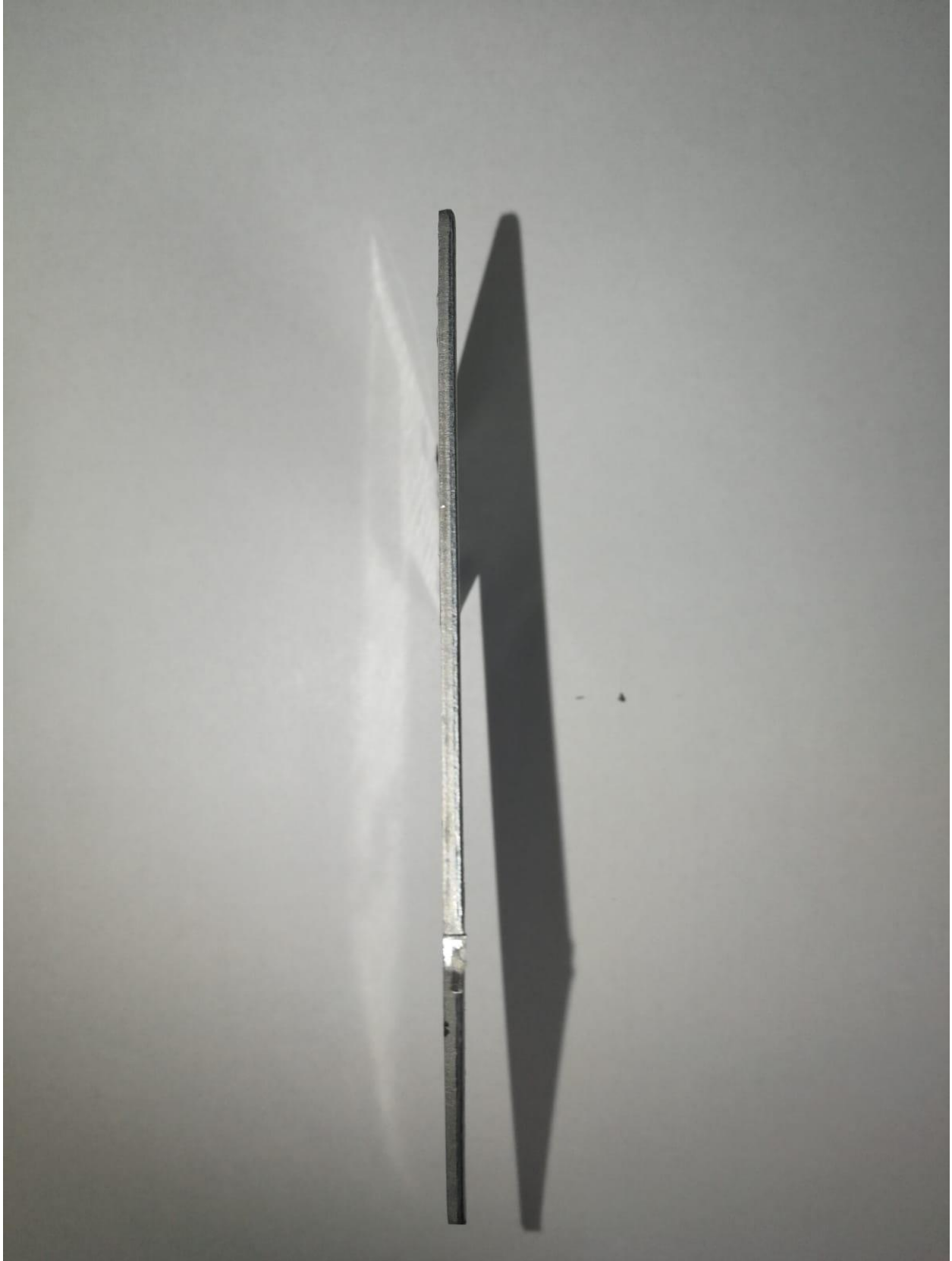
Denumire produs: Element de imbinare

Material: Otel zincat

Descriere produs: Elementul de imbinare in colt este confectionat din tabla de otel zincata. Se utilizeaza pentru constructii din lemn.

Imagini produs:

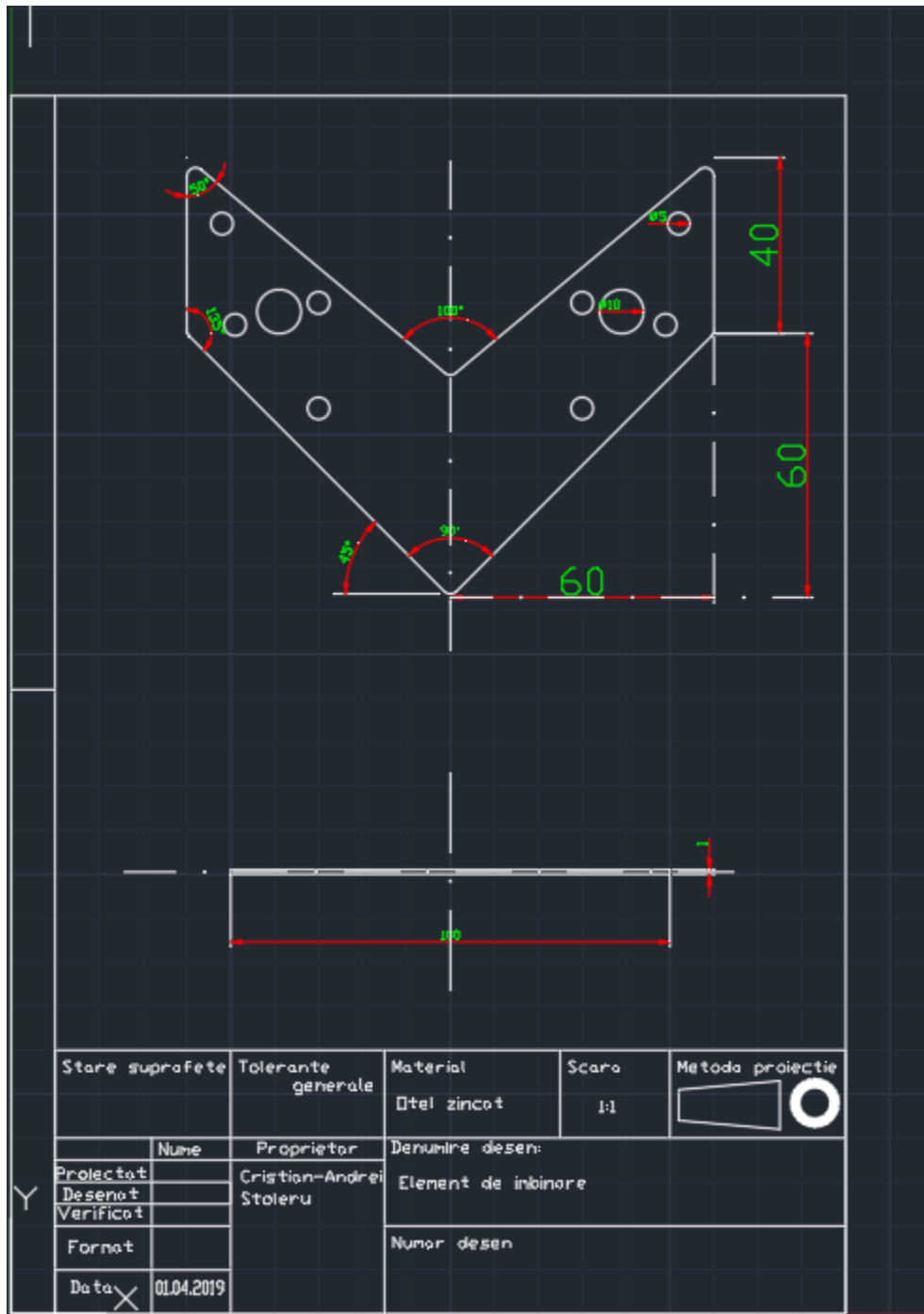






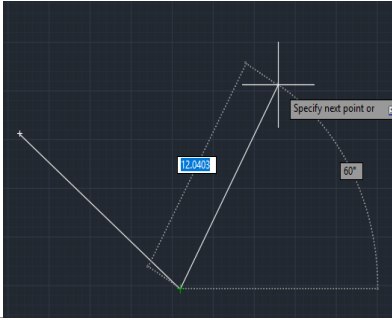
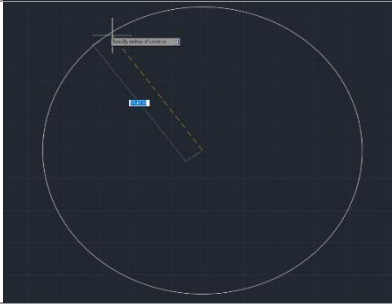
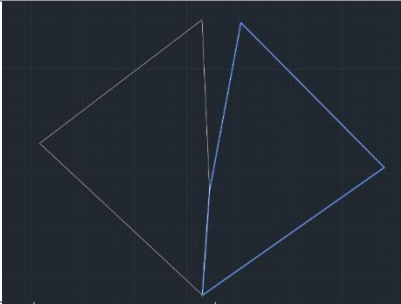

Autocad

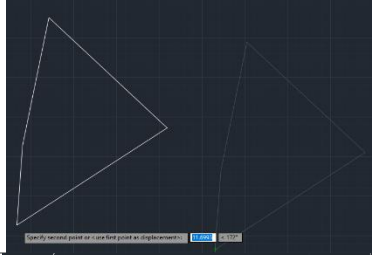
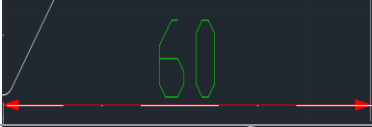
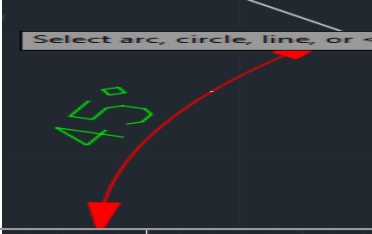

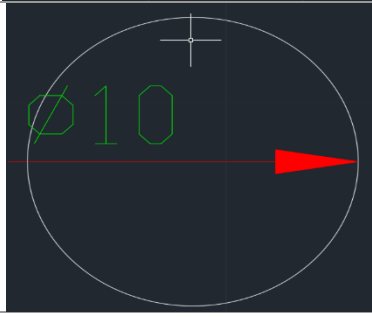
Dwg

Print Screen al [cas tema.dwg](#)



Tabel comenzi utilizate

Numar	Instructiune	Imagine
1	Grid	
2	Snap	
3	Pline	
4	Circle	
5	Explode	-
6	Mirror	
7	Fillet	

8	Move			
9	Dimlinear			
10	Dimangular			
11	Donut			
12	Dimdiameter			
13	Text		-	

Proiect 2: AutoLisp

Descriere program

Programul utilizeaza 3 functii:

1. Functia dfx:

Primește ca parametrii două liste *i* și *l* și returnează o listă compusă din asocierea celor două;

2. Functia vopsire:

Schimbă culoarea layerului pentru mai multe entități. Așteaptă numărul de entități cât și entitățile respective sunt introduse (selectate) de către utilizator;

3. Functia lumini:

Face vizibile entitățile ce aparțin layerului BECURI în funcție de valoarea lui *on* (1- invizibil / 0-vizibil).

Codul sursa

Hyperlink catre codul sursa: [Timp.lsp](#)

; returneaza ultimul element din asocierea lui i cu l

(defun dxf (i l)

 ; i- primul multime a asocierii

 ; l- a doua multime a asocierii

 (cdr (assoc i l))

);dxf

;-----

-----vopsire

;Schimba culoarea layerului pentru mai multe entitati. Atat numarul de entitati cat si entitatile respective sunt introduse(selectate) de catre utilizator

(defun c:vopsire(/ count esel ent layer olay color)

 ;**count** - Numarul de obiecte care vor fi vobiste, numar introdus de catre
 utilizator

 ;**esel** - Lista cu numele entitatii si coordonatele entitatill

 ;**ent** - Lista cu toate datele din entitatea aflata pe prima pozitie din esel

 ;**layer** - Contine denumirea layerului respectiv

 ;**olay** - Ttoate entitatile ce apartin layerului respectiv si informatiile genrale
 ale layerului

 ;**color** - Reprezinta culoarea layerului

 ;**newcolor** - Noua culoare aleasa de utilizator

(setq count (getint "\n Introdu numarul de obiecte pe care vrei sa le vopsesti: "))

(while (> count 0)

```
(if  
  (and  
    ;Selectia entitatii  
    (setq esel (entsel "Selecteaza un obiect"))  
    (setq ent (entget (car esel)))  
    ;Selectia Layerului  
    (setq layer (dxf 8 ent))  
    (setq olay (entget (tbloname "LAYER" layer)))  
    ;Selectia culorii curente si a culorii noi  
    (setq color(dxf 62 olay))  
    (setq newcolor(acad_colordlg color))  
    ;Modificarea culorii in toate entitatile ce apartin layerului  
    (setq olay(subst (cons 62 newcolor)(assoc 62 olay) olay))  
  );and  
(progn  
  ;Decrementarea count-ului  
  (setq count (- count 1))  
  ;Actualizarea listei olay  
  (entmod olay)  
  ;Refresh pentru a observa schimbarile  
  (command "REGEN")  
  ;Un mesaj pentru a arata ca schimbarile au fost efectuate cu succes  
  (alert "Layerul a fost modificat")  
);progn
```

```
;Afisarea unui mesaj in cazul in care utilizatorul nu a ales un obiect  
(alert "Nu ati selectat obiectul")  
  
);if  
  
);while  
  
); vopsire  
  
;-----lumini  
  
;face vizibile entitatile ce apartin layerului BECURI in functie de valoarea lui on (1-  
invizibil / 0-vizibil)  
  
(defun c:lumini(/ on olay )  
  
    ;on - Valoarea vizibilitatii  
  
    ;olay - Toate entitatile ce apartin layerului respectiv si informatiile genrale ale  
layerului  
  
(setq on (getint "\n Inchidem luminile (1) | Aprindem luminile (0) "))  
  
(if (or (= on 1) (= on 0))  
    (progn  
        (setq olay (entget (tbloname "LAYER" "becuri")))  
  
        ;Modificarea visibilitatii in toate entitatile ce apartin layerului  
  
(setq olay(subst (cons 70 on)(assoc 70 olay) olay))  
  
        ;Actualizarea listei olay  
  
(entmod olay)  
  
        ;Refresh pentru a observa schimbarile  
  
(command "REGEN")
```

;Afisarea mesajului

(if (= on 1) (alert "Au fost stinse luminile"))

(if (= on 0) (alert "Au fost aprinse luminile"))

);progn

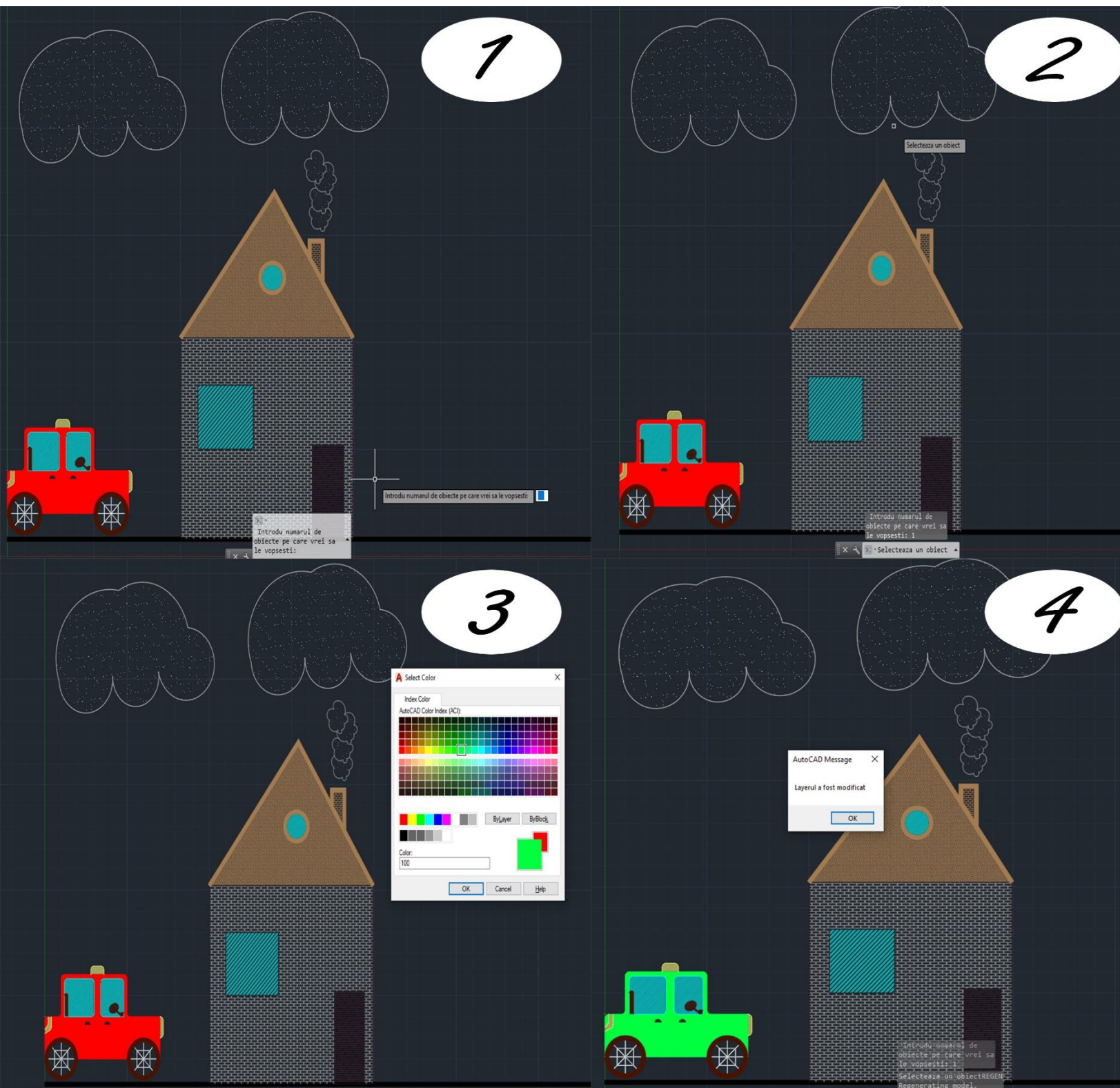
(alert "Valoare eronata")

);if

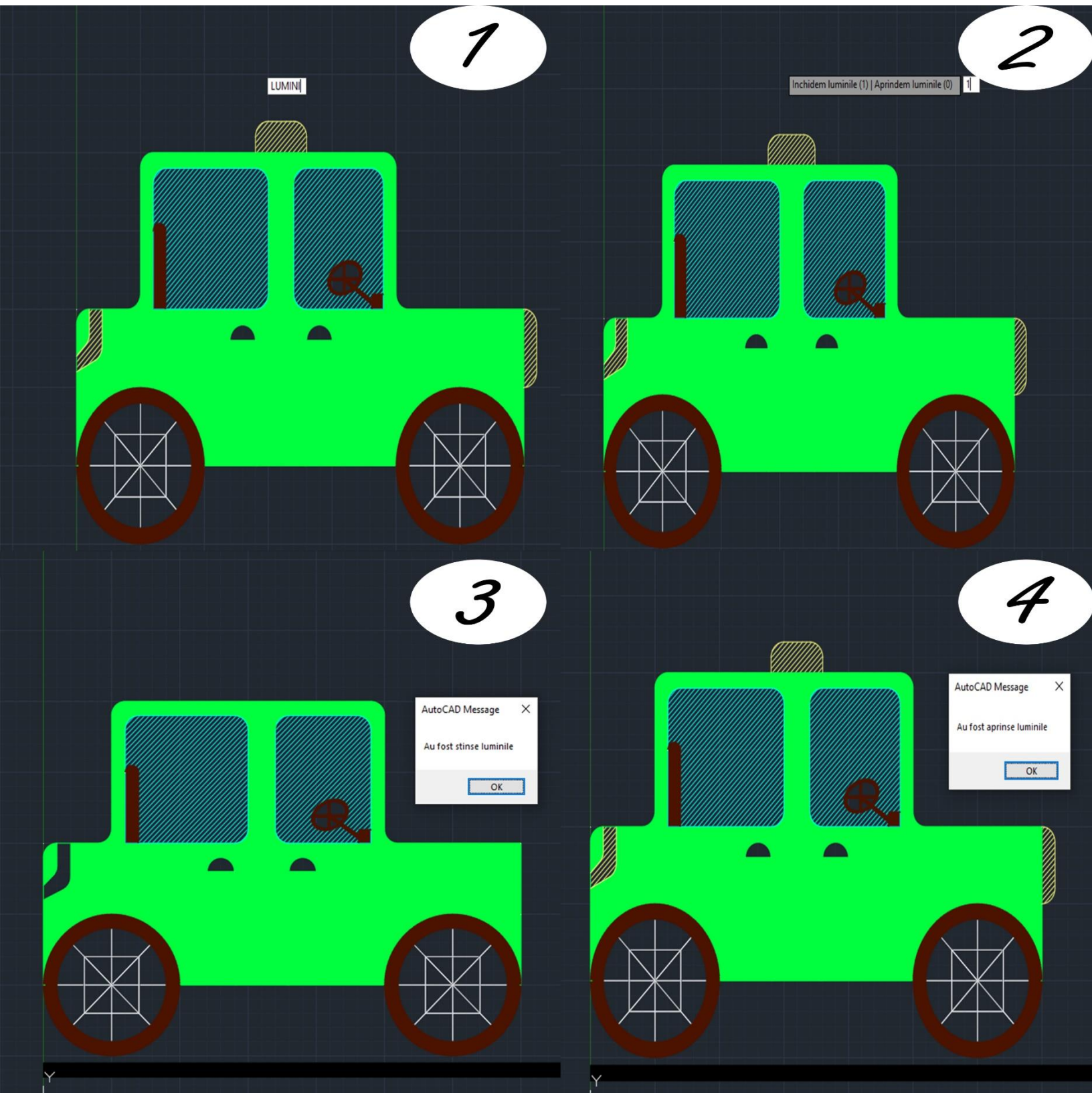
); lumini

Imagini ce reprezinta functionarea functiilor

Funcția vopsire



Funcția lumini



Bibliografie

[AutoLISP Basics de Land F/X \(Tutorial made by Jeremiah Farmer\)](#)

[Autodesk Knowledge Network](#)

Proiect 3: Autocad P&ID

Descriere produs

Denumire produs: Automat de facut pizza.

Mai jos este prezentata schema unui aparat portabil, asemanator cu cel de cafea, de facut pizza.

Funcțiile panoului de control:

1. Porneste/Opreste transferul din recipientul ce contine aluatul intr-un model care sa formeze forma rotunda a pizzei. Aceasta va fi mutata apoi in recipientul utilizat pentru incalzire;
2. Porneste/Opreste mixerul ce amesteca sosul pentru a fi mai putin dens. Realizeaza transferul sosului in dispozitivul de imprastiere pentru a putea fi impartit sos pe intregul aluat;
3. Porneste/Opreste transferul din recipientul ce contine salam in cel pentru incalzire;
4. Porneste/Opreste transferul din recipientul ce contine cascaval in cel pentru incalzire;
5. Deschide/Inchide capacul cuptorului(recipientul de incalzire);
6. Porneste/Opreste dispozitivul de rotire (Faciliteaza imprastierea salamului si a cascavalului pe intreaga suprafata a aluatului);
7. Porneste/Opreste dispozitivul de incalzire;
8. Porneste conveierul pentru a realiza transferul pizzei din cuptor in recipientul de servire.

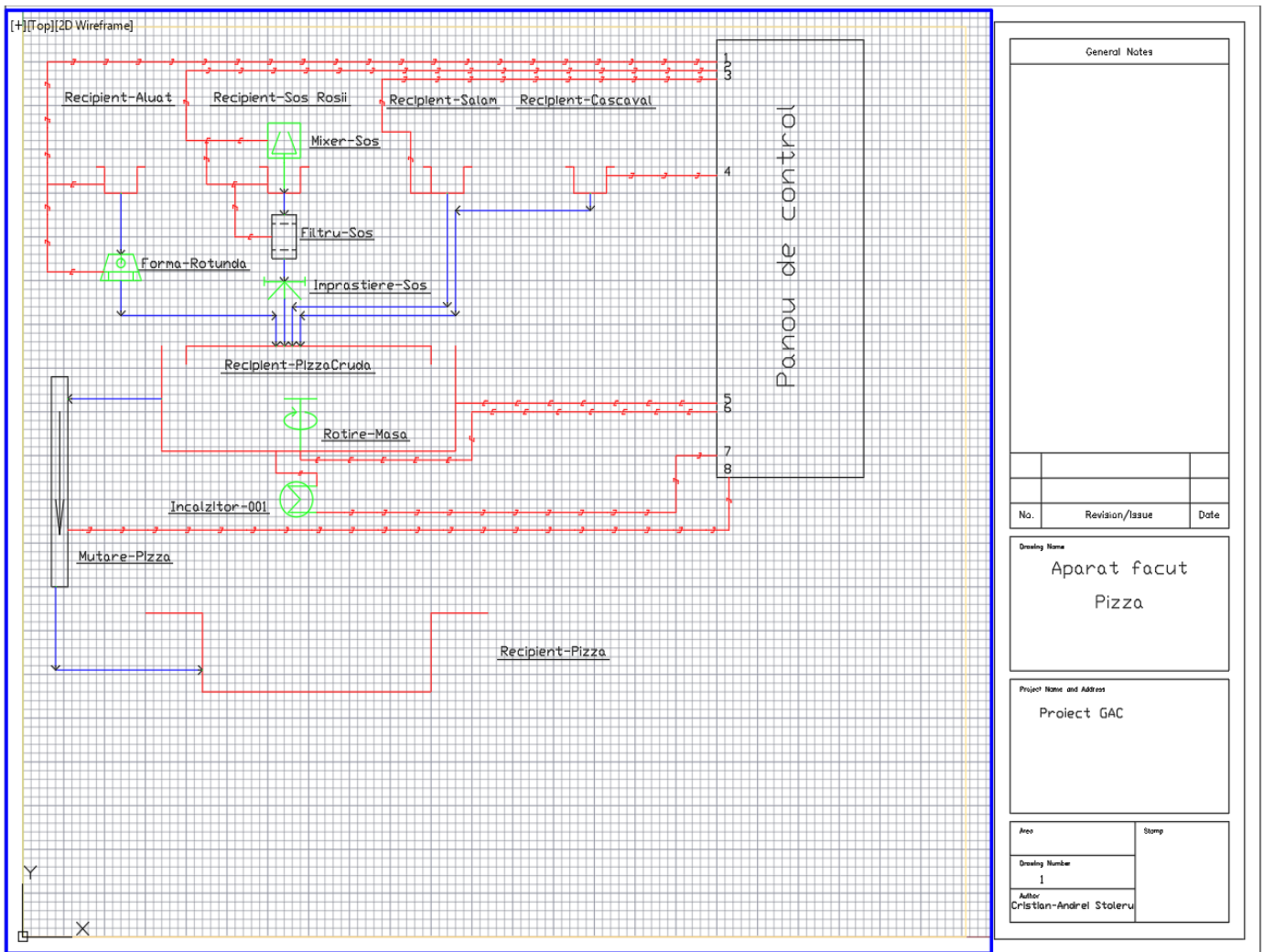
Modul de functiionare:

5(D)→1→2→6(P)→3→4→6(O)→5(I)→7→8

Autocad P&ID

Dwg

Print Screen al fisierului [Proiect3GACDWG.dwg](#)



Proiect 4: OpenGL

Obiectivul si descrierea codului

Obiectivul: Crearea unei obiect 3D și o animație pentru acesta utilizând pachetul de biblioteci OpenGL ale limbajului C++.

Descrierea codului:

Programul utilizează o funcție cu numele `display()`, unde se desenează 6 pătrate, de culori diferite, astfel încât acestea să formeze un cub de latură $l=0.5$. Această funcție modifică vederea utilizatorului rotind cu ajutorul unghiului format din `rotate_x` și `rotate_y` si translatând vederea cu ajutorul variabilei `count` creând astfel o animație plăcută vizual.

În funcția `main()` se creează o fereastră cu numele „Cub-Gac” în cadrul căreia se apelează funcția `display`.

Codul sursa

```
#include <stdio.h>

#include <stdarg.h>

#include <math.h>

#define GL_GLEXT_PROTOTYPES

#ifdef __APPLE__

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

double rotate_y=0;

double rotate_x=0;

int counter=-220;

void reshape( int width, int height)

{

    // setarea vederea

    // aceasta functie este apelata atunci cand fereastra este modificata

    glViewport(0,0,width,height);

    glMatrixMode(GL_PROJECTION);

    glLoadIdentity();

    gluPerspective(60,(float)width/(float)height, 1.0, 10.0);

    glMatrixMode(GL_MODELVIEW);

}
```

```
void display(){  
  
    // Stergem ecranul si bufferul z  
  
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);  
  
    //Resetam transformarile  
  
    glLoadIdentity();  
  
    // Realizam animatiile  
  
    glTranslatef(counter/150.0, 0.0,-3.0);  
  
    glRotatef( rotate_x, 1.0, 0.0, 0.0 );  
  
    glRotatef( rotate_y, 0.0, 1.0, 0.0 );  
  
    rotate_y += 2.5;  
  
    rotate_x += 2;  
  
    counter+=1;  
  
    if(counter>220)  
    {  
        counter=-220;  
  
        rotate_x=0;  
  
        rotate_y=0;  
    }  
}
```

// Realizam Patratele care formeaza cubul

glBegin(GL_POLYGON);//patrat 1

glColor3f(255,140,0);

glVertex3f(0.5, -0.5, -0.5);//p1

glVertex3f(0.5, 0.5, -0.5);//p2

glVertex3f(-0.5, 0.5, -0.5);//p3

glVertex3f(-0.5, -0.5, -0.5);//p4

glEnd();

glBegin(GL_POLYGON);//patrat2

glColor3f(0.5, 0.5, 1.0);

glVertex3f(0.5, -0.5, 0.5);

glVertex3f(0.5, 0.5, 0.5);

glVertex3f(-0.5, 0.5, 0.5);

glVertex3f(-0.5, -0.5, 0.5);

glEnd();

glBegin(GL_POLYGON);//patrat3

glColor3f(1.0, 0.4, 1.0);

glVertex3f(0.5, -0.5, -0.5);

glVertex3f(0.5, 0.5, -0.5);

glVertex3f(0.5, 0.5, 0.5);

glVertex3f(0.5, -0.5, 0.5);

glEnd();

```
glBegin(GL_POLYGON);//patrat4  
glColor3f( 0.3, 1.0, 0.3 );  
glVertex3f( -0.5, -0.5, 0.5 );  
glVertex3f( -0.5, 0.5, 0.5 );  
glVertex3f( -0.5, 0.5, -0.5 );  
glVertex3f( -0.5, -0.5, -0.5 );  
glEnd();
```

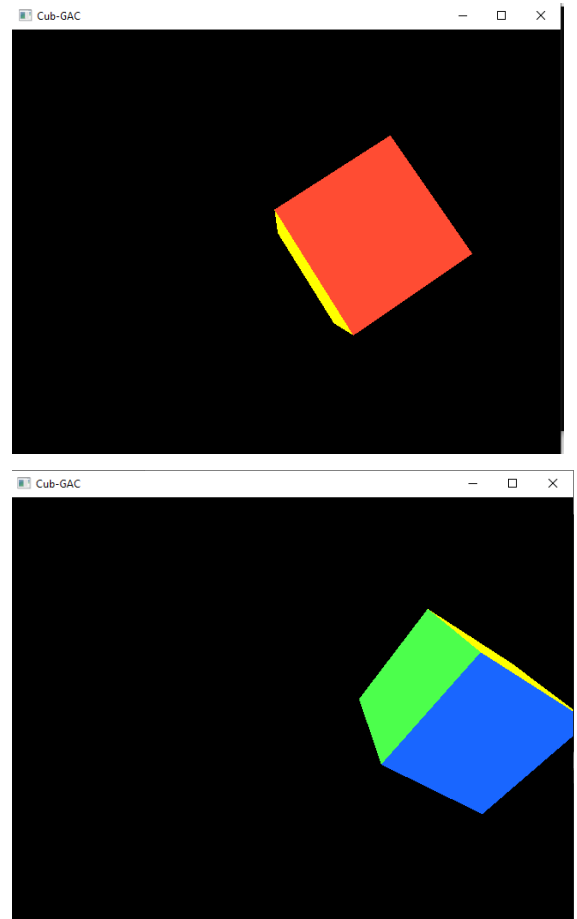
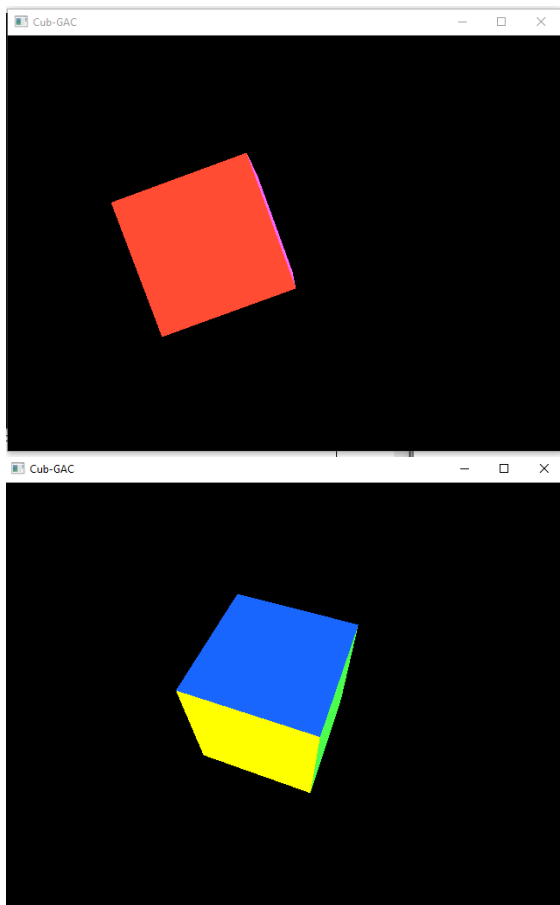
```
glBegin(GL_POLYGON);//patrat5  
glColor3f( 0.1, 0.4, 1.0 );  
glVertex3f( 0.5, 0.5, 0.5 );  
glVertex3f( 0.5, 0.5, -0.5 );  
glVertex3f( -0.5, 0.5, -0.5 );  
glVertex3f( -0.5, 0.5, 0.5 );  
glEnd();
```

```
glBegin(GL_POLYGON);//patrat6  
glColor3f( 1.0, 0.3, 0.2 );  
glVertex3f( 0.5, -0.5, -0.5 );  
glVertex3f( 0.5, -0.5, 0.5 );  
glVertex3f( -0.5, -0.5, 0.5 );  
glVertex3f( -0.5, -0.5, -0.5 );  
glEnd();
```

```
//Goleste toate bufferele astfel incat comenzile sa fie executate mai rapid  
glFlush();  
  
//Foloseste bufferul din layerul folosit  
glutSwapBuffers();  
  
}  
  
  
// -----  
int main(int argc, char* argv[]){  
  
// Porneste si initializeaza GLUT-ul  
glutInit(&argc,argv);  
  
// Seteaza modul de display, ce are nevoie de un buffer dublu de culoare pentru fereastra cu buffer Z  
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);  
  
//Seteaza dimensiunile ferestrei  
glutInitWindowSize(640,480);  
  
// Facem fereastra  
glutCreateWindow("Cub-GAC");  
  
// Activeaza bufferul Z pentru andancime  
glEnable(GL_DEPTH_TEST);
```

```
// Apelările funcției display  
glutDisplayFunc( display );  
glutIdleFunc( display ); // reapelează funcția în modul idle  
glutReshapeFunc( reshape );  
  
// Evenimentele sunt controlate de GLUT  
glutMainLoop();  
  
return 0;  
}
```


PrintScreen fereastra „Cub-GAC”



Bibliografie

Laboratoare GAC → OpenGL;

<https://www.wikihow.com/Make-a-Cube-in-OpenGL>