***Technical University of Cluj-Napoca***

***Automation and Computer Science Faculty***

***Automation and Applied Informatics***

*CAD In AUTOMATION*

*PROJECT*

|  |  |
| --- | --- |
| **Student : Mărghitaș Răzvan-Nicolae** | **Coordinator:** |
| **Group: 30312** | Assist. Eng. Dan Gota |
|  |  |
| 2018-2019 | |

**Summary**

Contents

[Summary 2](#_Toc483133297)

**Project 1 AutoCAD** 3

[Photos](#_Toc483133298) 3

[PrintScreenDWG](#_Toc483133299) 4

[Commands](#_Toc483133300) 5

[**Project 2**](#_Toc483133300) 6

[Description](#_Toc483133300) 6

[Photos](#_Toc483133300) 6

[Code](#_Toc483133300) 7

[**Project 3**](#_Toc483133300) 8

[Desciption](#_Toc483133300) 8

[Photo](#_Toc483133300) 8

[**Project 4**](#_Toc483133300) 9

[Description](#_Toc483133300) 9

[Photo](#_Toc483133300) 9

[Code](#_Toc483133300) 10

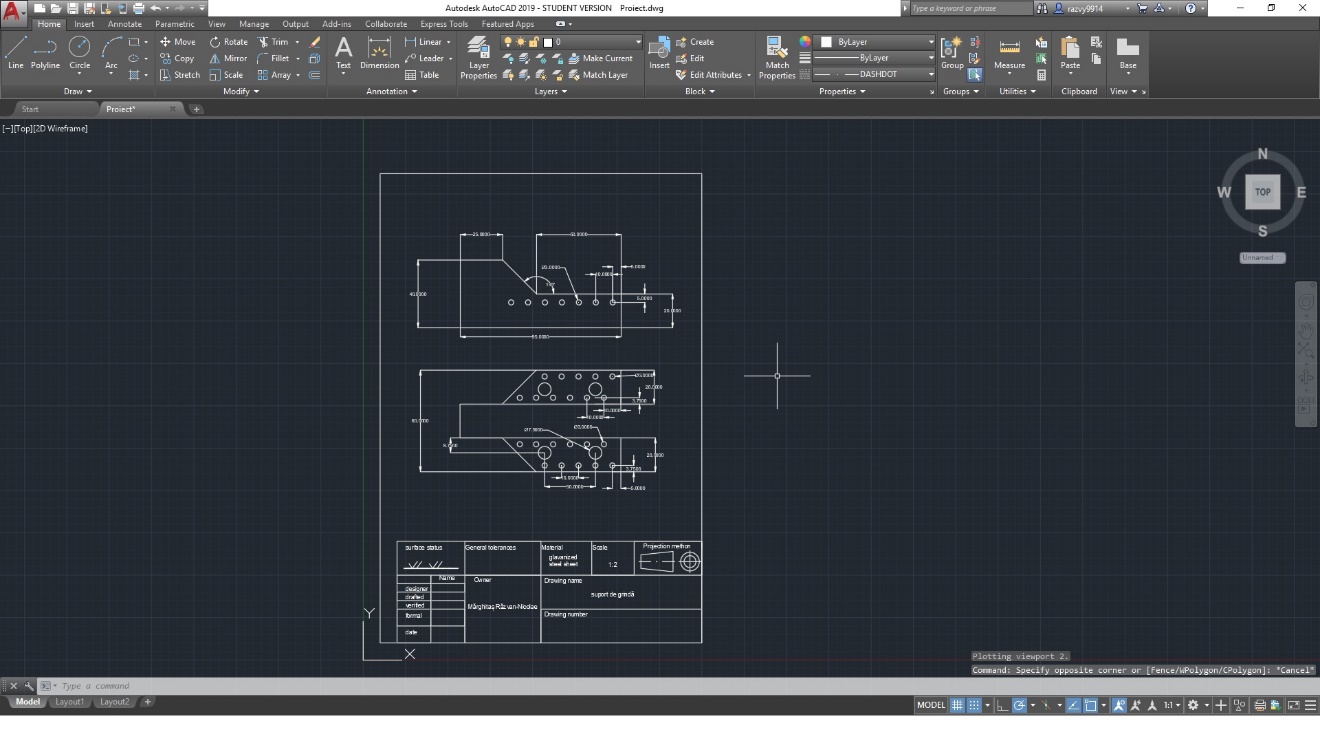
**Project1**

**Photos**

****

****

**Printscreen**

****

**Commands**

|  |  |
| --- | --- |
| Grid | Snap |
| Pline | Line (+width) |
| Rectang | Break |
| Chamfer(+distance,+angle) | Pan |
| Circle | Pedit |
| Linetype | Copy(+array) |
| Mirror | Dim |
| Dimdia | Dimang |
| Scale | Move |
| Dimstyle | Point |
| Ddtype |  |

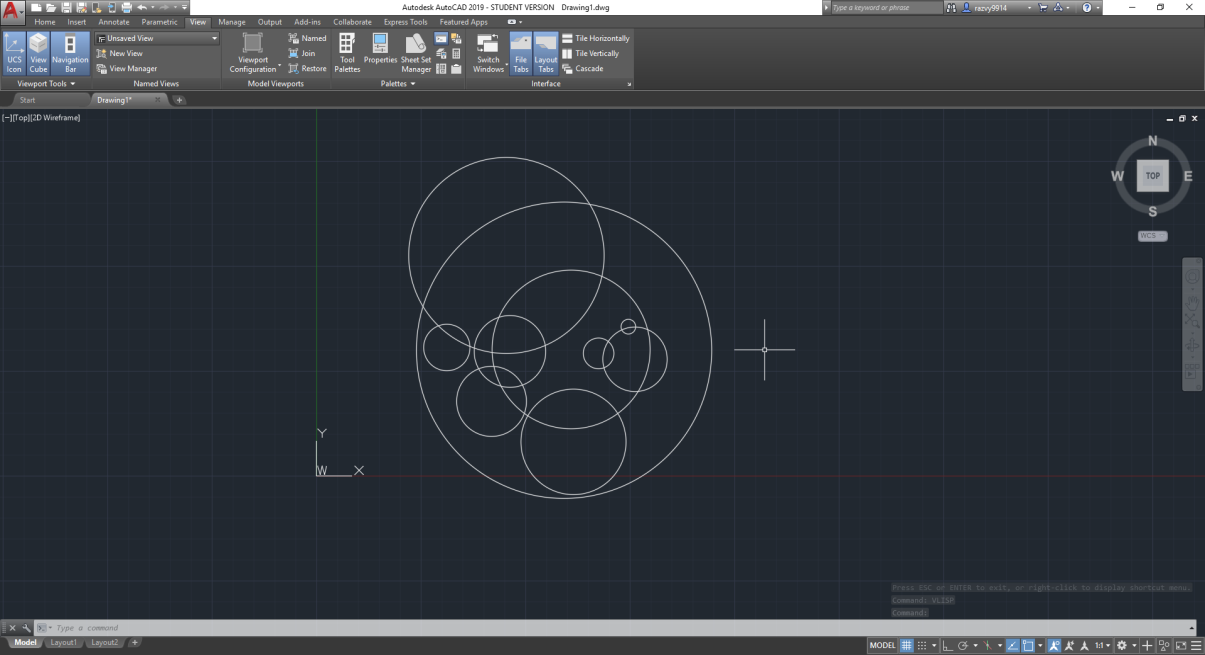
**Project2**

**Description**

The program/function resizes circles in a certain range that is given by the user to a dimension also given by the user (the dimension is measured by the circle’s radius not diameter).

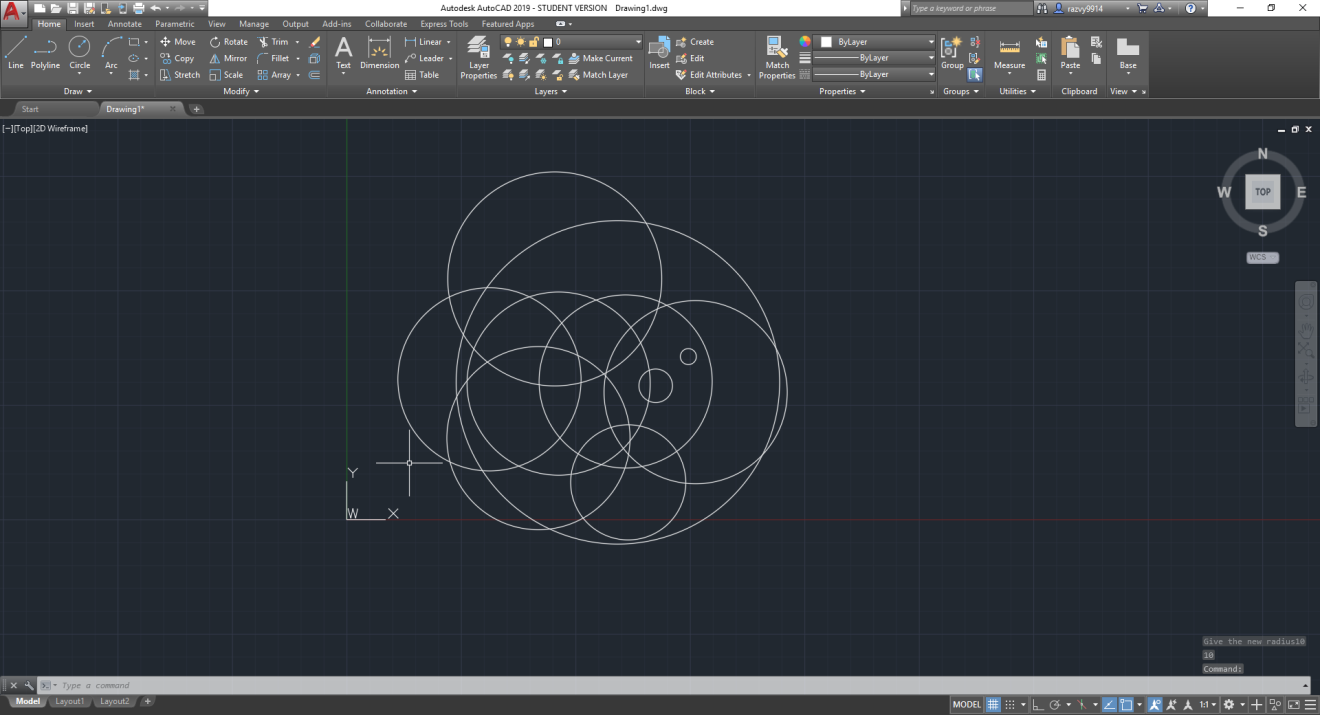
**Photos**

**before**



**after**

(after using raza with range min 2 range max 6 new radius 10)

**

**Code**

(defun c:raza()

(setq filter (list '(0 . "CIRCLE") ) circles(ssget "X" filter) ) ;filters all the circles that are in the drawing to a list circles

(setq val2 (getreal "Give the minimum value"))

(setq val1 (getreal "Give the maximum value"))

(setq newraza (getreal "Give the new radius"))

(setq n 0)

(repeat (sslength circles) ;repeats the following commands for all the circles in the list

(setq cerc (entget (ssname circles n))

raza (assoc 40 cerc)) ;sets the currect value of the currect circle's radius to raza in the format of a dotted pair

(setq valraza (cdr raza)) ;takes just the value of radius

(if (>= val1 valraza) (if (<= val2 valraza) (setq cerc (subst (cons 40 newraza) raza cerc))) ;if the current radius of the circle is in the range given, changes the current radius to a new one

)

(entmod cerc) (setq n (1+ n))

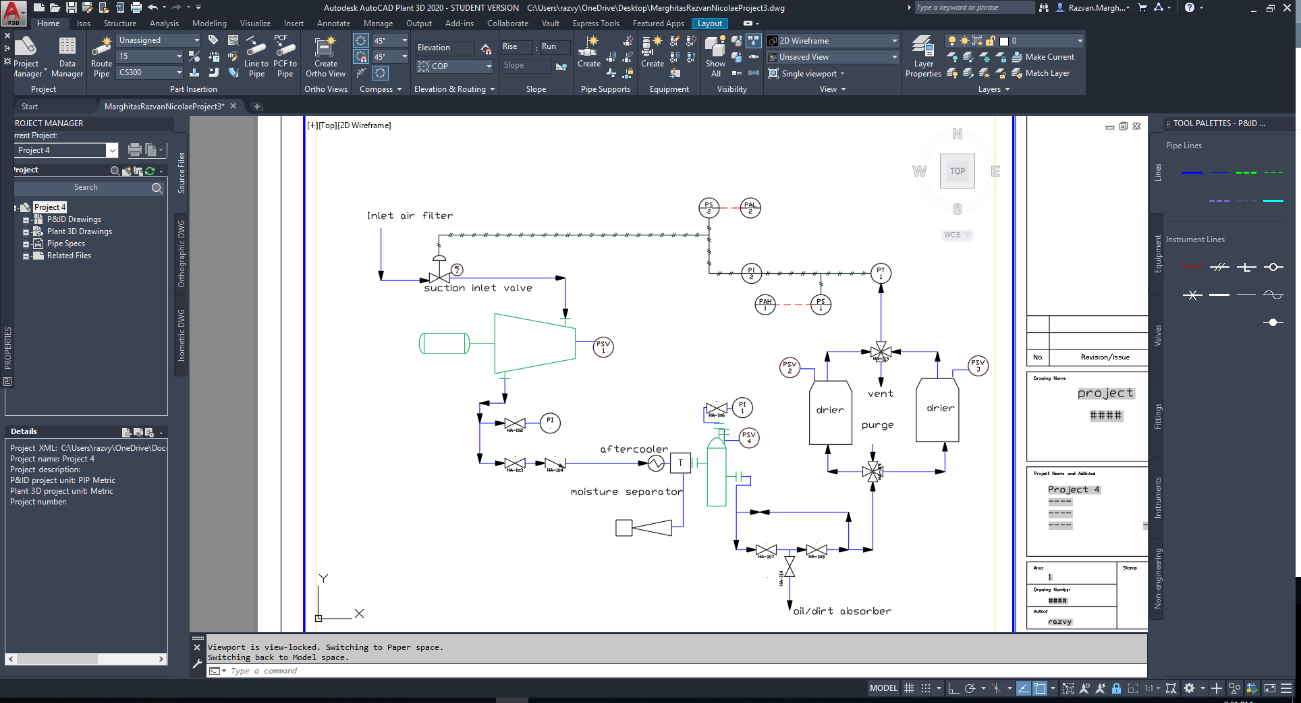
))

**Project3**

**Description**

The compressor takes air from the atmosphere through a filter and compressor compresses the air. The filter/vent removes the big dirt in the air. Always the compressed air will be hot, the hot air is cooled using coolers. Hot air is passed through aftercoolers. Moisture developed in the air due to cooling is separated using moisture separator. Now the air is sent to a receiver. A receiver removes the dirt and oil content in the air supply. Then the air is sent to the dryers. Dryers remove the remaining moisture content in the air. The dryers use desiccants to absorb moisture from the air. The regulator regulates the air pressure to the required air supply pressure for plant instruments.

**Photo**



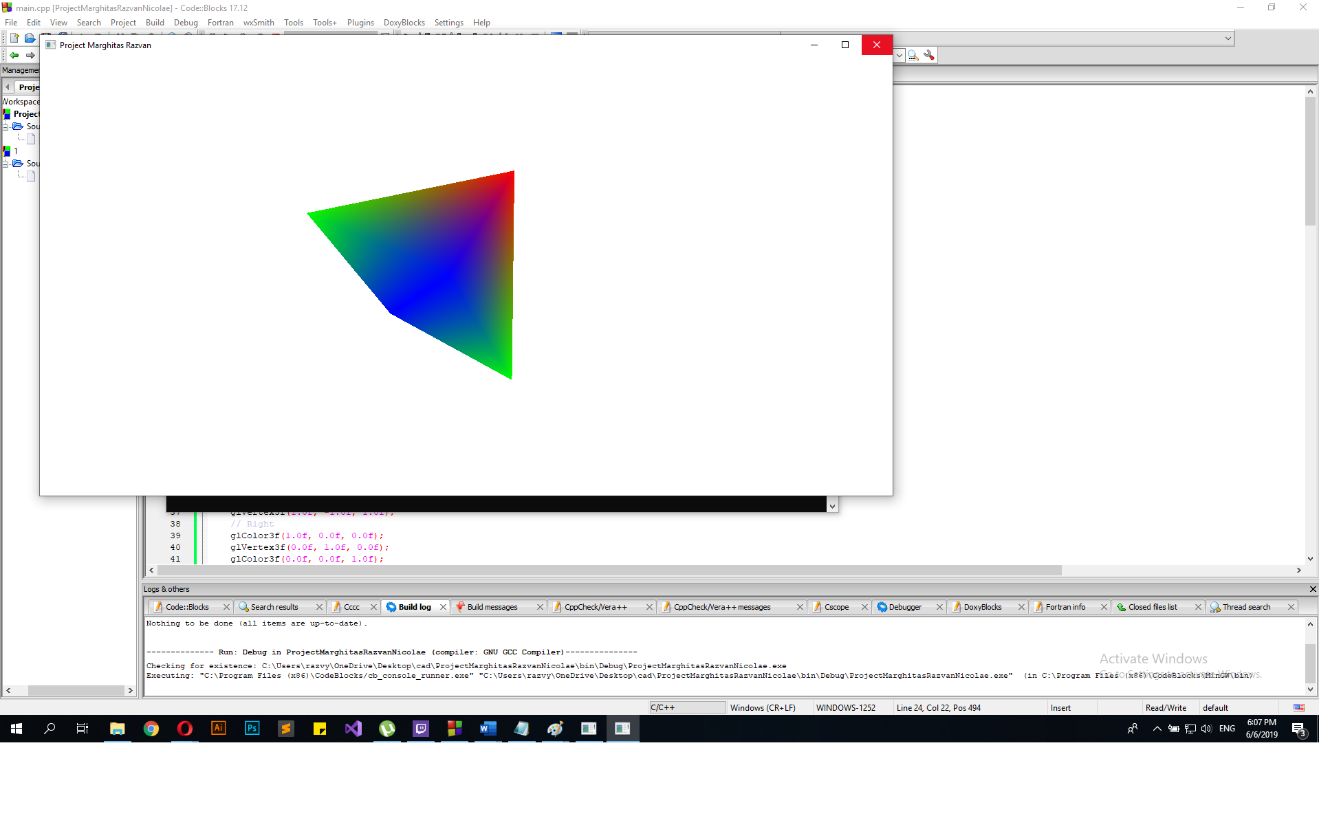
**Project 4**

**Description**

The objective of the program is to observe a pyramid and for me to learn more about opengl.

It construct a pyramid and rotates it with an angle pyraang and if we want to give the user an illusion that the pyramid rotates faster we encrease the rotation angle with + or - and use q to exit.

**Photo**

**

**Code**

*#include <windows.h>*

*#include <GL/glut.h>*

*char title[] = "Project Marghitas Razvan";*

*GLfloat pyraang = 0.0f;*

*int refresh = 15;*

*int speed = 1;*

*void initial()*

*{*

*glClearColor(1.0f, 1.0f, 1.0f, 1.0f);*

*glClearDepth(1.0f);*

*glEnable(GL\_DEPTH\_TEST);*

*glDepthFunc(GL\_LEQUAL);*

*glHint(GL\_PERSPECTIVE\_CORRECTION\_HINT, GL\_NICEST);*

*}*

*void displayp()*

*{*

*glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);*

*glMatrixMode(GL\_MODELVIEW);*

*glLoadIdentity();*

*glTranslatef(0.0f, 0.5f, -6.0f);*

*glRotatef(pyraang, 1.0f, 1.0f, 1.0f);*

*glBegin(GL\_TRIANGLES); //beggins rotation*

*// Front*

*glColor3f(1.0f, 0.0f, 0.0f);*

*glVertex3f( 0.0f, 1.0f, 0.0f);*

*glColor3f(0.0f, 1.0f, 0.0f);*

*glVertex3f(-1.0f, -1.0f, 1.0f);*

*glColor3f(0.0f, 0.0f, 1.0f);*

*glVertex3f(1.0f, -1.0f, 1.0f);*

*// Right*

*glColor3f(1.0f, 0.0f, 0.0f);*

*glVertex3f(0.0f, 1.0f, 0.0f);*

*glColor3f(0.0f, 0.0f, 1.0f);*

*glVertex3f(1.0f, -1.0f, 1.0f);*

*glColor3f(0.0f, 1.0f, 0.0f);*

*glVertex3f(1.0f, -1.0f, -1.0f);*

*// Back*

*glColor3f(1.0f, 0.0f, 0.0f);*

*glVertex3f(0.0f, 1.0f, 0.0f);*

*glColor3f(0.0f, 1.0f, 0.0f);*

*glVertex3f(1.0f, -1.0f, -1.0f);*

*glColor3f(0.0f, 0.0f, 1.0f);*

*glVertex3f(-1.0f, -1.0f, -1.0f);*

*// Left*

*glColor3f(1.0f,0.0f,0.0f);*

*glVertex3f( 0.0f, 1.0f, 0.0f);*

*glColor3f(0.0f,0.0f,1.0f);*

*glVertex3f(-1.0f,-1.0f,-1.0f);*

*glColor3f(0.0f,1.0f,0.0f);*

*glVertex3f(-1.0f,-1.0f, 1.0f);*

*// Bottom*

*glColor3f(0.0f,0.0f,1.0f);*

*glVertex3f( 1.0f, -1.0f, 1.0f);*

*glColor3f(0.0f,1.0f,0.0f);*

*glVertex3f(-1.0f, -1.0f, 1.0f);*

*glColor3f(0.0f,0.0f,1.0f);*

*glVertex3f(-1.0f, -1.0f, -1.0f);*

*// Bottom*

*glColor3f(0.0f,0.0f,1.0f);*

*glVertex3f( 1.0f, -1.0f, 1.0f);*

*glColor3f(0.0f,0.0f,1.0f);*

*glVertex3f(-1.0f, -1.0f, -1.0f);*

*glColor3f(0.0f,1.0f,0.0f);*

*glVertex3f( 1.0f, -1.0f, -1.0f);*

*glEnd();*

*glutSwapBuffers();*

*pyraang += 0.15f\*speed; //incresease the angle of rotation/speed*

*}*

*void timer(int value)*

*{*

*glutPostRedisplay();*

*glutTimerFunc(refresh, timer, 0);*

*}*

*//key bindings for exiting, increasing speed and decreasing it*

*static void key(unsigned char key, int x, int y)*

*{*

*switch (key)*

*{*

*case 'q':*

*exit(0);*

*break;*

*case '+':*

*speed++;*

*break;*

*case '-':*

*if (speed>1)*

*{*

*speed--;*

*}*

*break;*

*}*

*glutPostRedisplay();*

*}*

*void reshape(GLsizei width, GLsizei height)*

*{*

*if(height == 0) height = 1;*

*GLfloat aspect = (GLfloat)width/(GLfloat)height;*

*glViewport(0,0,width,height);*

*glMatrixMode(GL\_PROJECTION);*

*glLoadIdentity();*

*gluPerspective(45.0f, aspect, 0.1f, 100.0f);*

*}*

*int main(int argc, char\*\* argv)*

*{*

*glutInit(&argc,argv);*

*glutInitDisplayMode(GLUT\_DOUBLE);*

*glutInitWindowSize(1240,640);*

*glutInitWindowPosition(50,50);*

*glutCreateWindow(title);*

*glutDisplayFunc(displayp);*

*glutKeyboardFunc(key);*

*glutReshapeFunc(reshape);*

*initial();*

*glutTimerFunc(0, timer, 0);*

*glutMainLoop(); return 0; }*