TECHNICAL
UNIVERSITY
OF CLUJ-NAPOCA
ROMANIA

Faculty of Automation and Computer Science

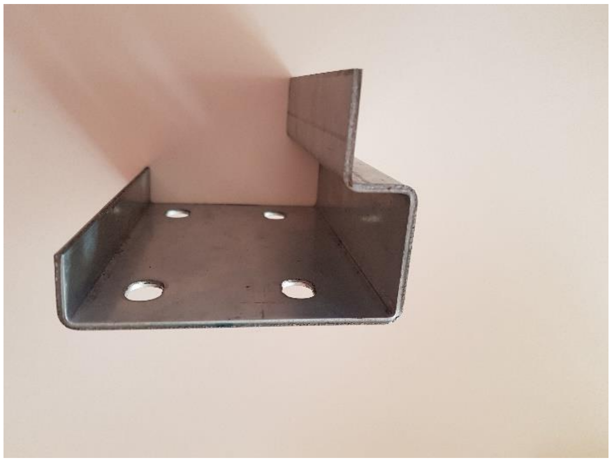
Computer Aided Graphics

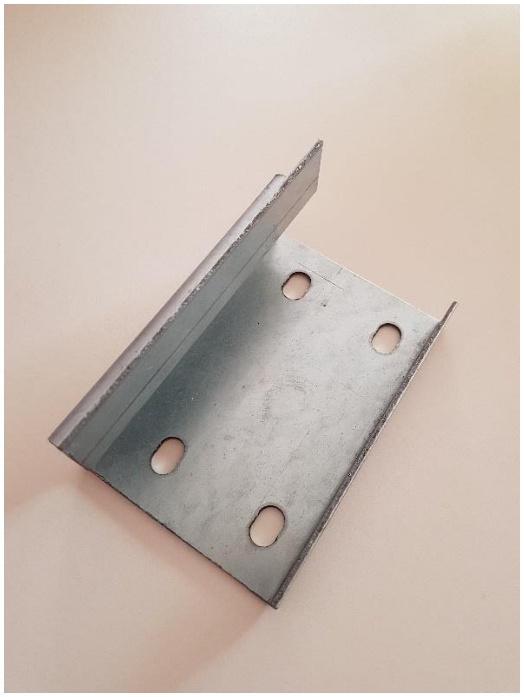
Nagy Timea

2020 - 2021

AUTOCAD PROJECT

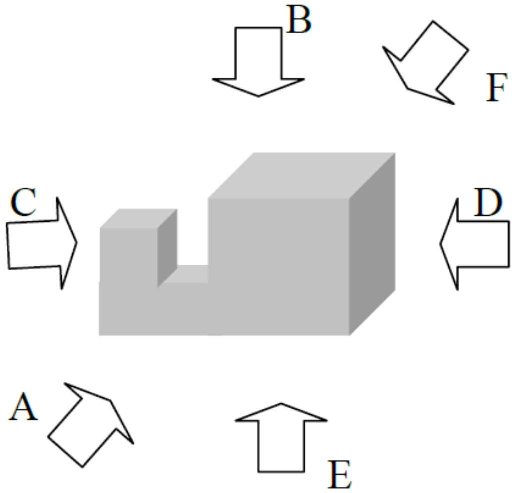
THE OBJECT



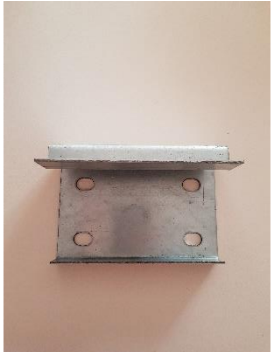


THE VIEW

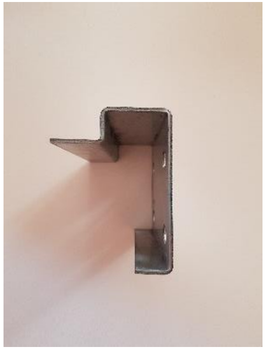
The projection method used in this project is the European projection (metoda de proiectie a primului triedru)



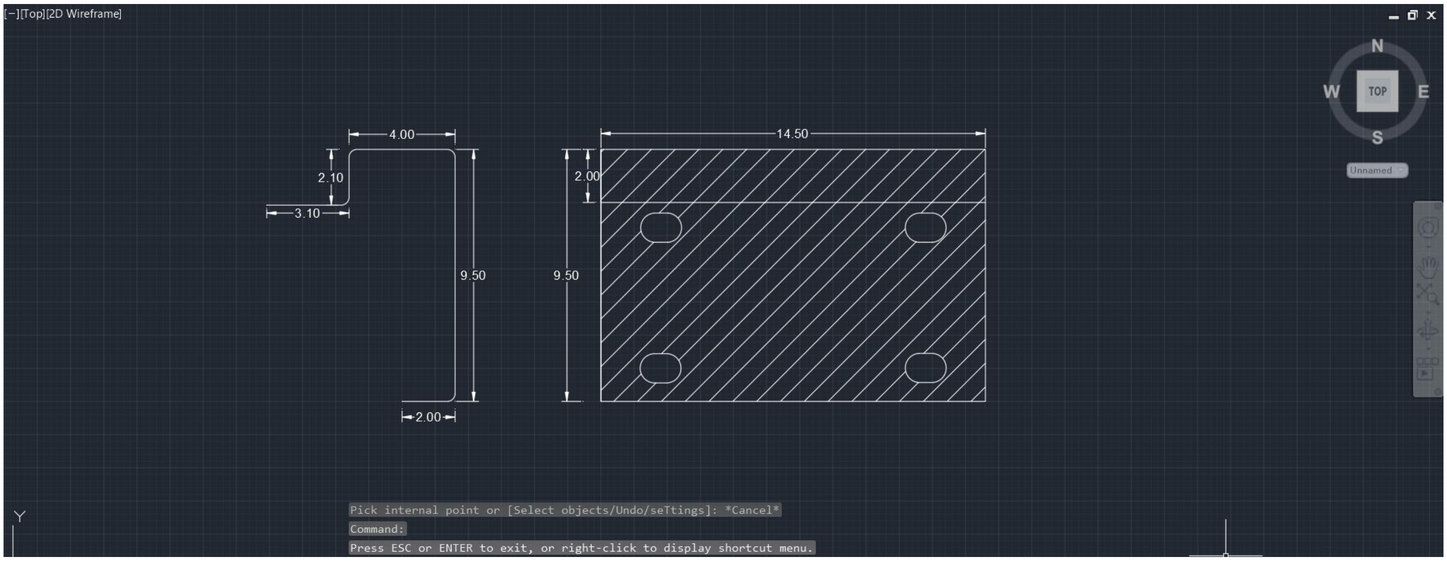
- A view (main view)



- D view (secondary view, right view)



AUTOCAD



```
Pick internal point or [Select objects/Undo/seTtings]: *Cancel*
Command:
Press ESC or ENTER to exit, or right-click to display shortcut menu.
```

COMMANDS

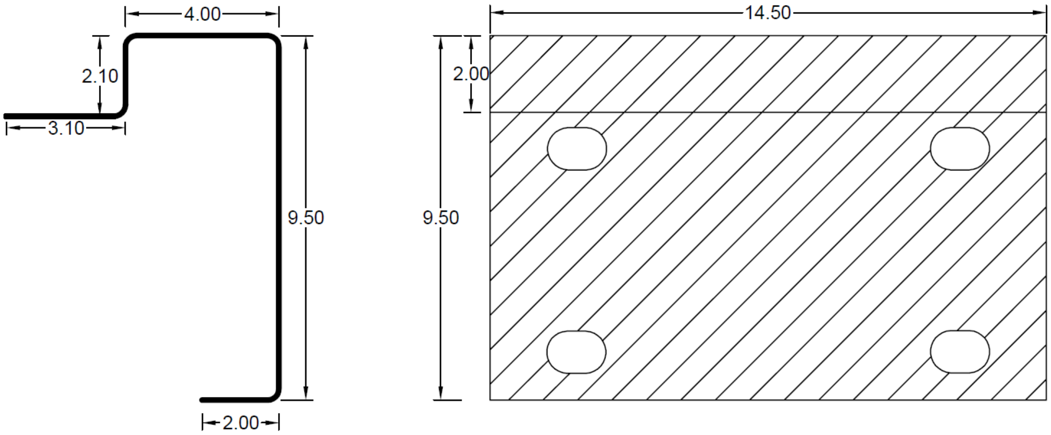
For the object

- PLINE
- LINE
- ARC (for the ellipse-like holes in the metal)
- COPY
- MOVE
- FILLET
- DIMLINEAR
- BHATCH

For the indicator

- PLINE
- TEXT
- MOVE
- CIRCLE
- STYLE

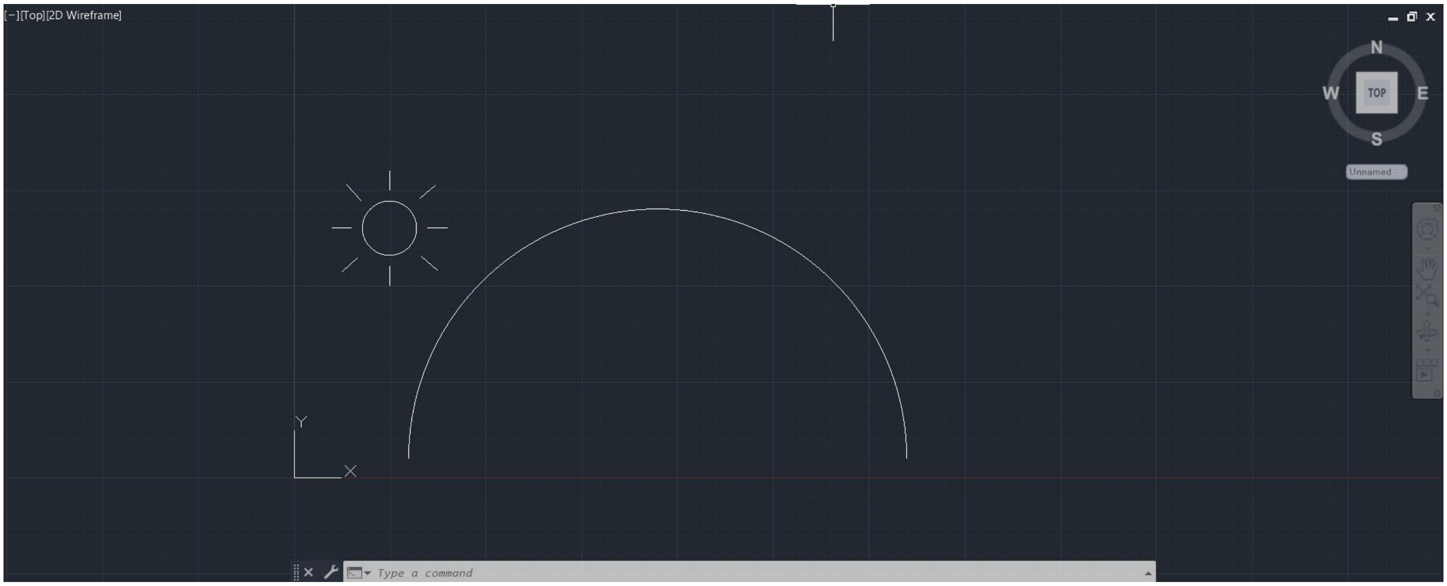
RESULT



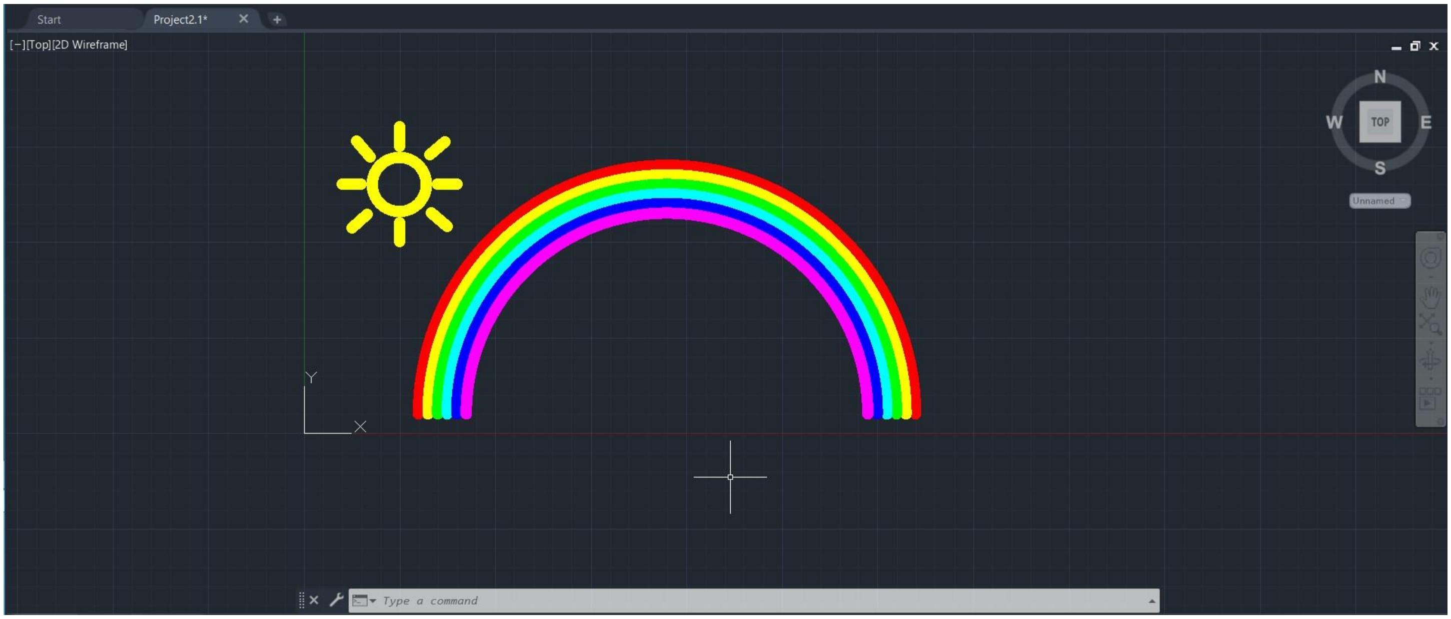
FUNCTION DESCRIPTION

The function is named RAINBOW, it selects all of the entities from the screen. The function modifies more entities, the arc and the circle with the lines. It changes their lineweight and colour, to form a rainbow and a sun. Initially, there must be only one arc, which contains the exact place and radius of the rainbow. This will be the red arc of the rainbow. The other arcs are automatically created within the function.

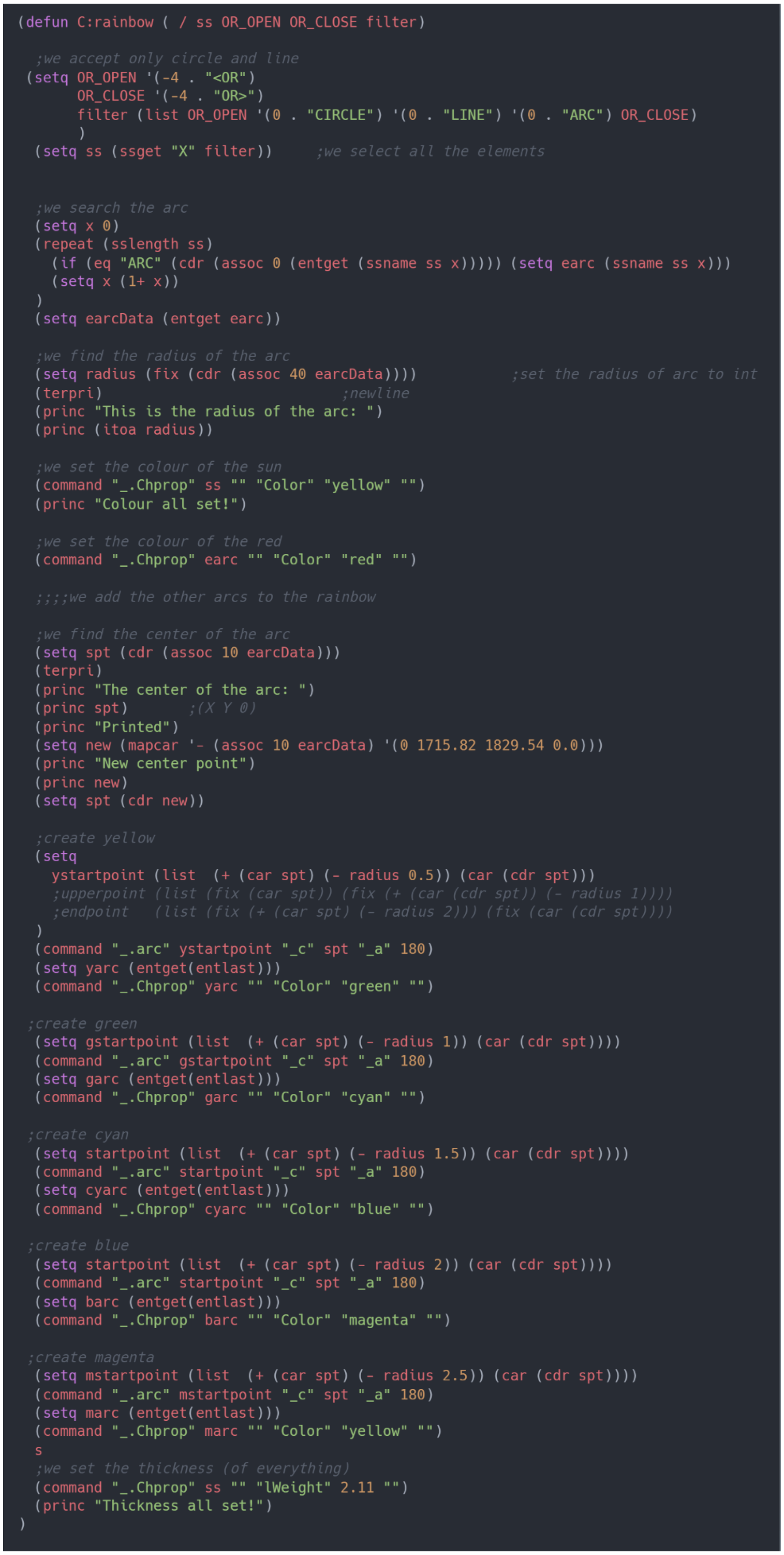
EXAMPLE BEFORE RUNNING THE FUNCTION



EXAMPLE AFTER RUNNING THE FUNCTION



THE CODE



```
(defun C:rainbow ( / ss OR_OPEN OR_CLOSE filter)

  ;we accept only circle and line
 (setq OR_OPEN '(-4 . "<OR")
       OR_CLOSE '(-4 . "OR>")
       filter (list OR_OPEN '(0 . "CIRCLE") '(0 . "LINE") '(0 . "ARC") OR_CLOSE)
       )
  (setq ss (ssget "X" filter))     ;we select all the elements


  ;we search the arc
  (setq x 0)
  (repeat (sslength ss)
    (if (eq "ARC" (cdr (assoc 0 (entget (ssname ss x))))) (setq earc (ssname ss x)))
    (setq x (1+ x))
  )
  (setq earcData (entget earc))

  ;we find the radius of the arc
  (setq radius (fix (cdr (assoc 40 earcData))))           ;set the radius of arc to int
  (terpri)                            ;newline
  (princ "This is the radius of the arc: ")
  (princ (itoa radius))

  ;we set the colour of the sun
  (command "_.Chprop" ss "" "Color" "yellow" "")
  (princ "Colour all set!")

  ;we set the colour of the red
  (command "_.Chprop" earc "" "Color" "red" "")

  ;;;;we add the other arcs to the rainbow

  ;we find the center of the arc
  (setq spt (cdr (assoc 10 earcData)))
  (terpri)
  (princ "The center of the arc: ")
  (princ spt)       ;(X Y 0)
  (princ "Printed")
  (setq new (mapcar '- (assoc 10 earcData) '(0 1715.82 1829.54 0.0)))
  (princ "New center point")
  (princ new)
  (setq spt (cdr new))

  ;create yellow
  (setq
    ystartpoint (list  (+ (car spt) (- radius 0.5)) (car (cdr spt)))
    ;upperpoint (list (fix (car spt)) (fix (+ (car (cdr spt)) (- radius 1))))
    ;endpoint   (list (fix (+ (car spt) (- radius 2))) (fix (car (cdr spt))))
  )
  (command "_.arc" ystartpoint "_c" spt "_a" 180)
  (setq yarc (entget(entlast)))
  (command "_.Chprop" yarc "" "Color" "green" "")

 ;create green
  (setq gstartpoint (list  (+ (car spt) (- radius 1)) (car (cdr spt))))
  (command "_.arc" gstartpoint "_c" spt "_a" 180)
  (setq garc (entget(entlast)))
  (command "_.Chprop" garc "" "Color" "cyan" "")

 ;create cyan
  (setq startpoint (list  (+ (car spt) (- radius 1.5)) (car (cdr spt))))
  (command "_.arc" startpoint "_c" spt "_a" 180)
  (setq cyarc (entget(entlast)))
  (command "_.Chprop" cyarc "" "Color" "blue" "")

 ;create blue
  (setq startpoint (list  (+ (car spt) (- radius 2)) (car (cdr spt))))
  (command "_.arc" startpoint "_c" spt "_a" 180)
  (setq barc (entget(entlast)))
  (command "_.Chprop" barc "" "Color" "magenta" "")

 ;create magenta
  (setq mstartpoint (list  (+ (car spt) (- radius 2.5)) (car (cdr spt))))
  (command "_.arc" mstartpoint "_c" spt "_a" 180)
  (setq marc (entget(entlast)))
  (command "_.Chprop" marc "" "Color" "yellow" "")
  s
  ;we set the thickness (of everything)
  (command "_.Chprop" ss "" "lWeight" 2.11 "")
  (princ "Thickness all set!")
)
```

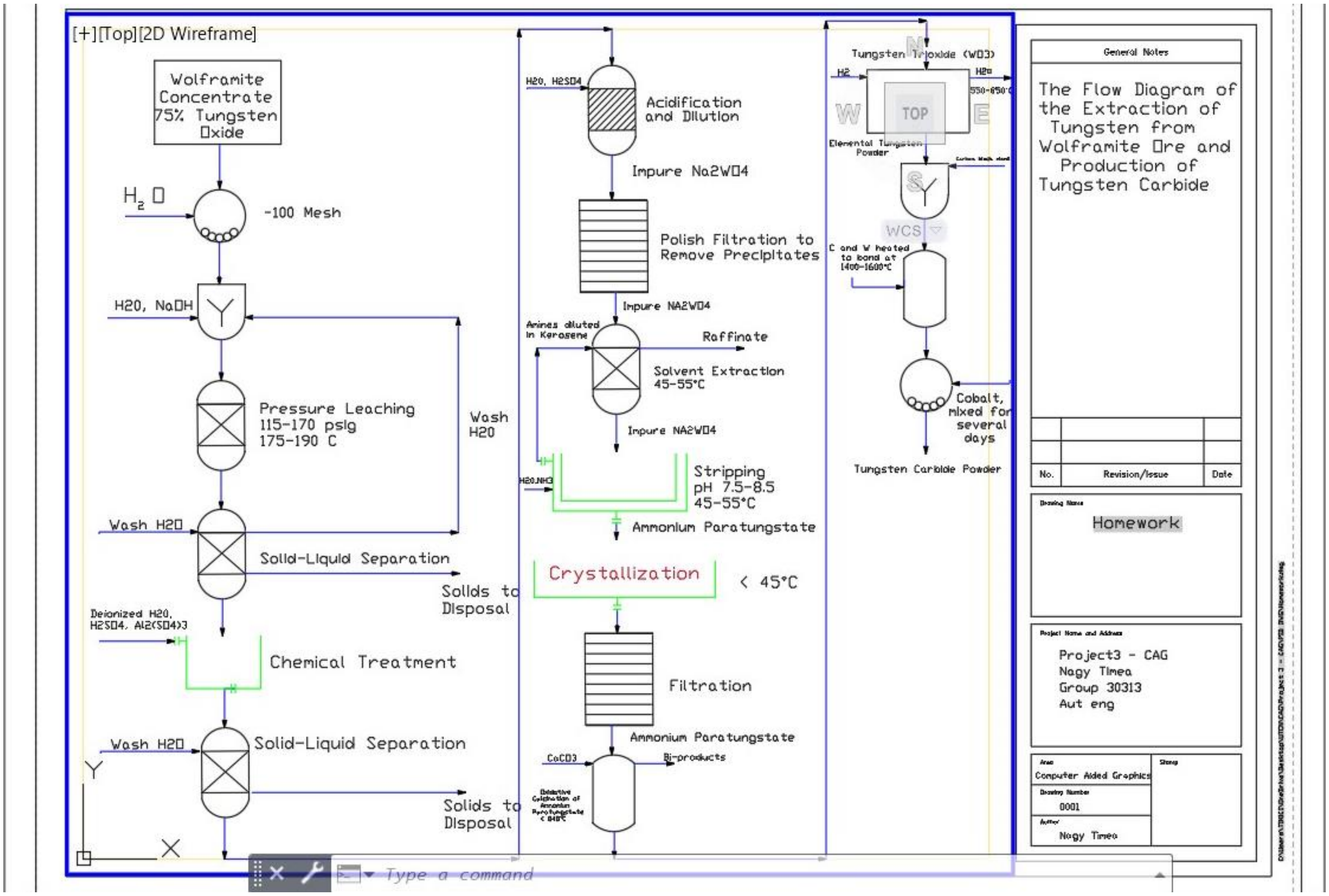
PIPING & INSTRUMENTATION DIAGRAM

DESCRIPTION OF THE PROCESS

This drawing is a detailed diagram in the process industry which shows the piping and process equipment together with the instrumentation and control devices of a chemical extraction, more precisely the extraction of Tungsten from Wolframite Ore and the Production of Tungsten Carbide.

We can see that, firstly 75% Wolframite Concentrate Tungsten Oxide is introduced in the system, then, through the process we shall use/intoroduce water (H_2O), sodium hydroxide (NaOH), we separate liquids from solids and dispose the solid substances, we add deionized water, sulfuric acid and aluminium sulfate. The mixture goes under chemical treatment, then we separate again solids from liquids, acidificate and dillute, then the impure sodium tungstate is polish filtrated to remove precipitates, the we extract the solvent from impure sodium around 45-55 °C. After a crystallization process on less than 45 °C, the ammonium paratungstate is filtrated, then after the oxidative calcination of the ammonium paratungstate we obtain Tungsten Trioxide (WO_3), to which we add H_2 and extract H_2O under 550-650 °C. We obtain the material called elemental tungsten powder to which we add black carbon and mix it and then heat to bond at around 1400-1600°C. By adding cobalt, we get the desired material, the Tungsten Carbide Powder.

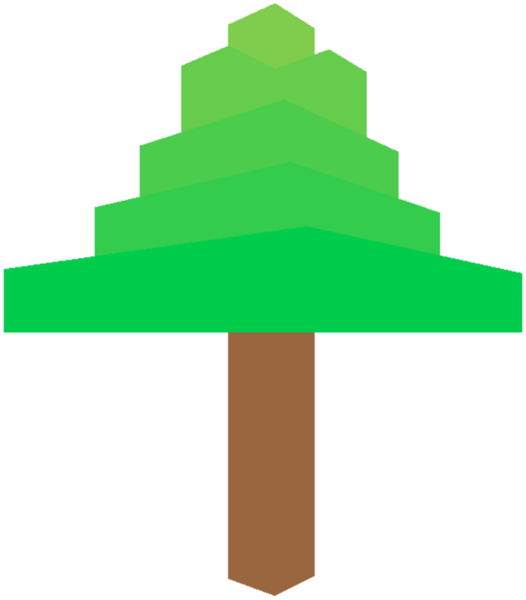
RESULT



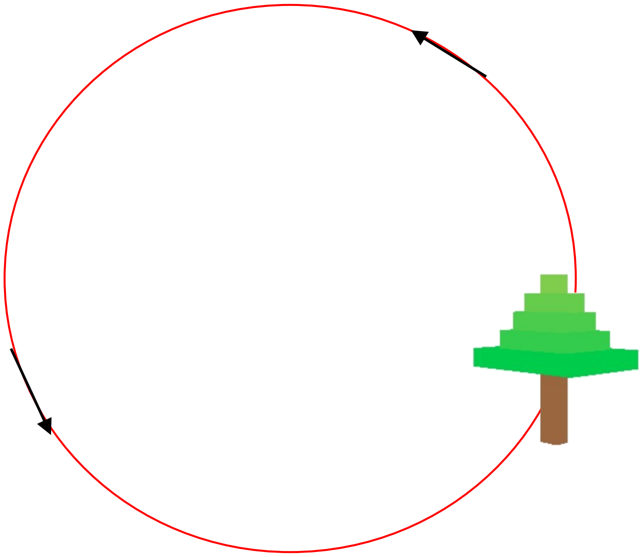
OPENGL PROJECT

DESCRIPTION

My OpenGL project is written in codeblocks using C++ and glut extension. I created a 3D tree out of cubes which is rotating around the Y axis and also orbiting on a circle. The tree I designed is inspired from a minecraft tree like the first picture and looks like the second picture in my project.

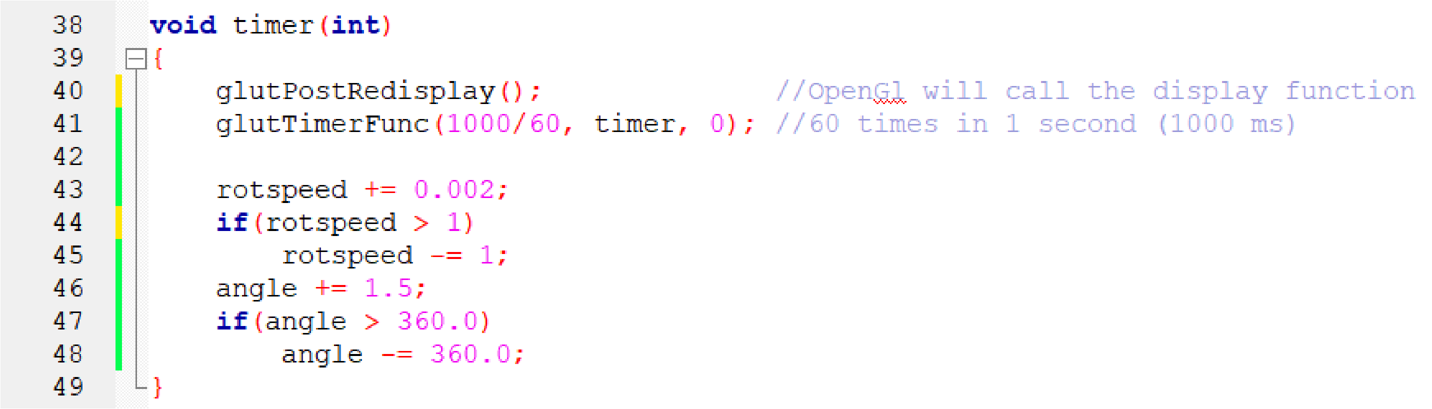


The tree is rotating around the red circle, but the circle is not visible in the project.



THE CODE

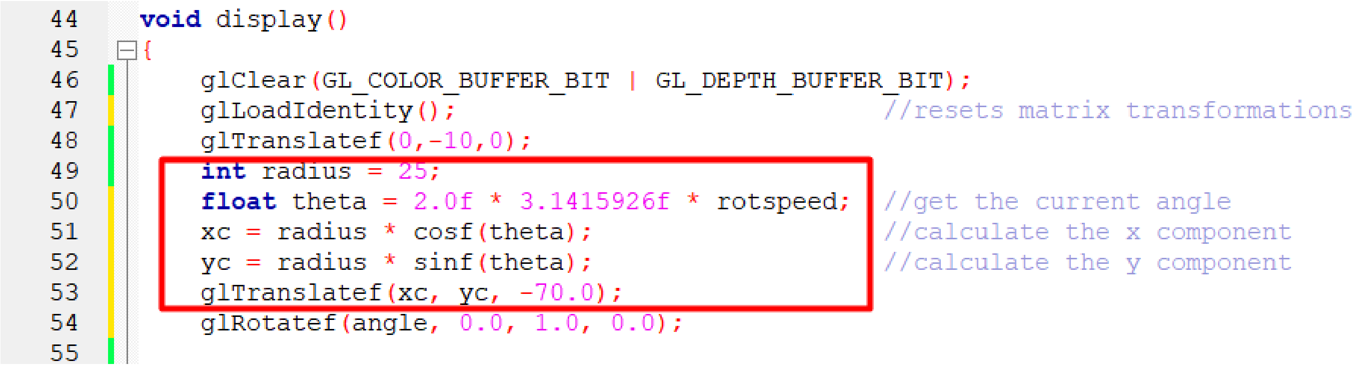
I have a timer function which calls itself 60 times every second and calls the display function.



```
void timer(int)
{
    glutPostRedisplay();                //OpenGL will call the display function
    glutTimerFunc(1000/60, timer, 0); //60 times in 1 second (1000 ms)

    rotspeed += 0.002;
    if(rotspeed > 1)
        rotspeed -= 1;
    angle += 1.5;
    if(angle > 360.0)
        angle -= 360.0;
}
```

The display function firstly erases the screen with white, and then displays the new content. This way is the animation created. In the display function the red part handles the tree's circular motion. It calculates the angle of the rotation. The rotation speed variable is changed in the timer function.

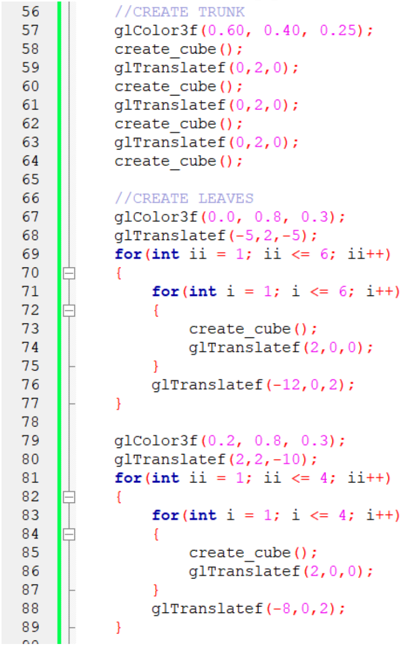


```
void display()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();                               //resets matrix transformations
    glTranslatef(0,-10,0);
    int radius = 25;
    float theta = 2.0f * 3.1415926f * rotspeed;  //get the current angle
    xc = radius * cosf(theta);                      //calculate the x component
    yc = radius * sinf(theta);                      //calculate the y component
    glTranslatef(xc, yc, -70.0);
    glRotatef(angle, 0.0, 1.0, 0.0);

```

I created a create_cube function which creates 1 cube at the origin point. It creates the cube's front, left, right, back, top and bottom parts.

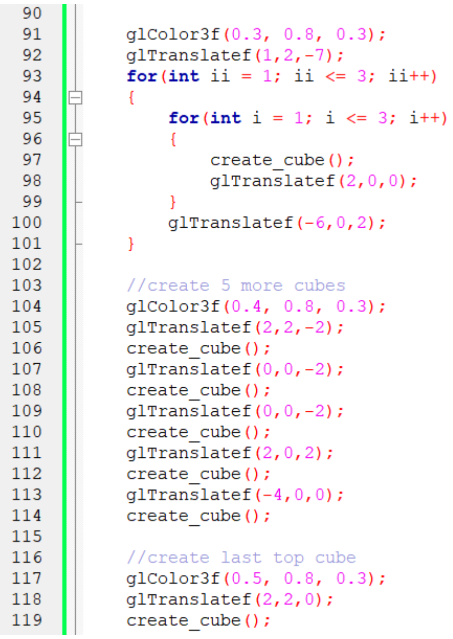
The creation of the tree happens here:



```
    //CREATE TRUNK
    glColor3f(0.60, 0.40, 0.25);
    create_cube();
    glTranslatef(0,2,0);
    create_cube();
    glTranslatef(0,2,0);
    create_cube();
    glTranslatef(0,2,0);
    create_cube();

    //CREATE LEAVES
    glColor3f(0.0, 0.8, 0.3);
    glTranslatef(-5,2,-5);
    for(int ii = 1; ii <= 6; ii++)
    {
        for(int i = 1; i <= 6; i++)
        {
            create_cube();
            glTranslatef(2,0,0);
        }
        glTranslatef(-12,0,2);
    }

    glColor3f(0.2, 0.8, 0.3);
    glTranslatef(2,2,-10);
    for(int ii = 1; ii <= 4; ii++)
    {
        for(int i = 1; i <= 4; i++)
        {
            create_cube();
            glTranslatef(2,0,0);
        }
        glTranslatef(-8,0,2);
    }
```



```

    glColor3f(0.3, 0.8, 0.3);
    glTranslatef(1,2,-7);
    for(int ii = 1; ii <= 3; ii++)
    {
        for(int i = 1; i <= 3; i++)
        {
            create_cube();
            glTranslatef(2,0,0);
        }
        glTranslatef(-6,0,2);
    }

    //create 5 more cubes
    glColor3f(0.4, 0.8, 0.3);
    glTranslatef(2,2,-2);
    create_cube();
    glTranslatef(0,0,-2);
    create_cube();
    glTranslatef(0,0,-2);
    create_cube();
    glTranslatef(2,0,2);
    create_cube();
    glTranslatef(-4,0,0);
    create_cube();

    //create last top cube
    glColor3f(0.5, 0.8, 0.3);
    glTranslatef(2,2,0);
    create_cube();
```

TABLE OF CONTENTS