****

**MINISTERUL EDUCATIEI CULTURII SI CERCETARII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnica a Moldovei**

**Facultatea Calculatoare, Informatica si Microelectronica**

**Departamentul Ingineria Software si Automatica**

**Postoronca Dumitru FAF-233**

Report

Laboratory Work 1

**of Computer Graphics**

Checked by:

**Olga Grosu,** university assistant

DISA, FCIM, UTM

**Chisinau - 2024**

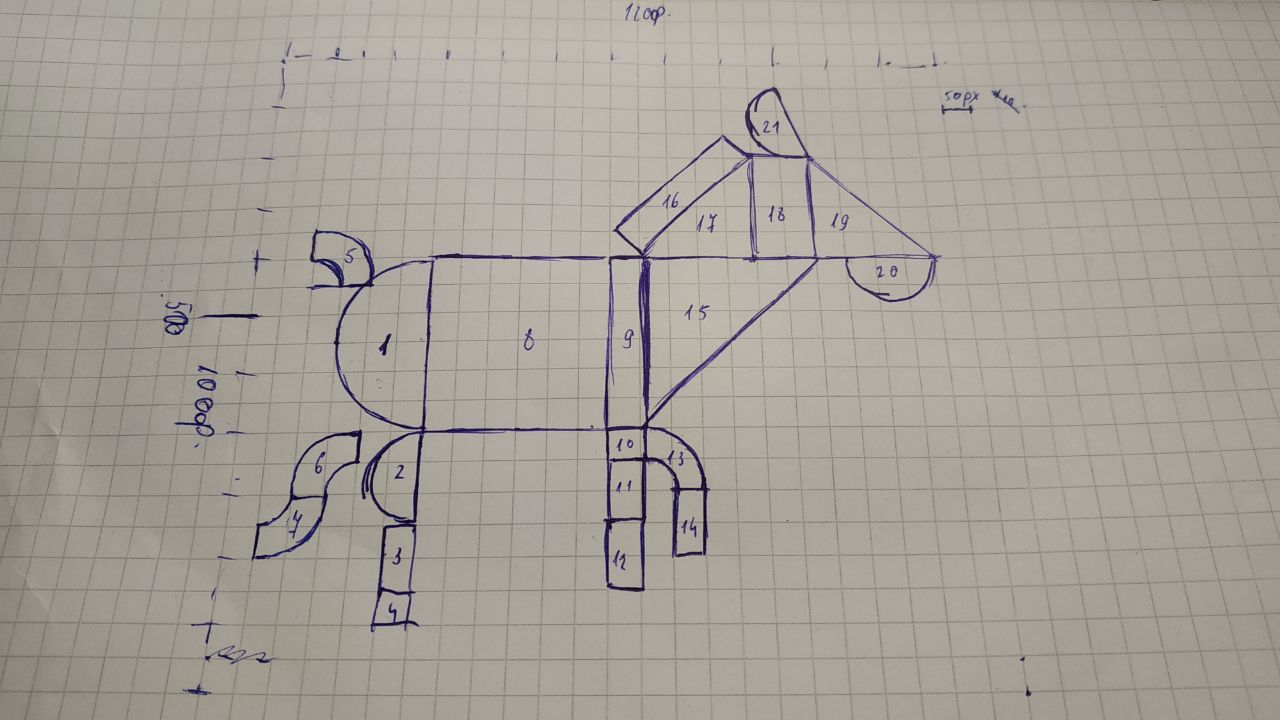
1. **The purpose of laboratory work:**

The purpose of this laboratory work is to get familiar with 2d primitives from Processing tool by drawing an animal or a creature using these primitives(e.g. arc, circle, rect)

1. **Condition:**

To use 2d primitives and especially arc, circle that require some mathematical parameters as PI

Reference picture



1. **The code of the program:**

void setup(){

size(1200, 1000);

background(250);

noStroke();

fill(255,255,255);

}

void fig1(){

pushMatrix();//saves the current transformation state

fill(#B02229);

translate(300, 550);//move origin to these coordinates

rotate(PI);

arc(0, 0, 300, 300, -PI/2, PI/2, CLOSE);

popMatrix();//restore previous transformation state

}

void fig2(){

pushMatrix();

fill(#D4452A);

translate(300, 775);

rotate(PI);

arc(0, 0, 150, 150, -PI/2, PI/2, CLOSE);

popMatrix();

}

void fig3(){

fill(#B02229);

rect(250,850, 50, 100);

}

void fig4(){

fill(#DDC626);

rect(250,950, 50, 100);

}

void fig5(){

pushMatrix();

translate(100, 450);

rotate(-PI/2);

fill(#D85226);

arc(0, 0, 200, 200, 0, HALF\_PI, PIE);

fill(255);

arc(0, 0, 100, 100, 0, HALF\_PI, PIE);

popMatrix();

}

void fig6(){

pushMatrix();

translate(200, 800);

rotate(PI);

fill(#A82126);

arc(0, 0, 200, 200, 0, HALF\_PI, PIE);

fill(255);

arc(0, 0, 100, 100, 0, HALF\_PI, PIE);

popMatrix();

}

void fig7(){

pushMatrix();

translate(50, 800);

fill(#E5D01B);

arc(0, 0, 200, 200, 0, HALF\_PI, PIE);

fill(255);

arc(0, 0, 100, 100, 0, HALF\_PI, PIE);

popMatrix();

}

void fig8(){

fill(#3178BE);

rect(300, 400, 300, 300);

}

void fig9(){

fill(#D64B2C);

rect(600, 400, 50, 300);

}

void fig10(){

fill(#E5D01B);

rect(600, 700, 50, 50);

}

void fig11(){

fill(#3178BE);

rect(600, 750, 50, 100);

}

void fig12(){

fill(#E5D01B);

rect(600, 850, 50, 100);

}

void fig13(){

pushMatrix();

translate(650, 800);

rotate(-PI/2);

fill(#3178BE);

arc(0, 0, 200, 200, 0, HALF\_PI, PIE);

fill(255);

arc(0, 0, 100, 100, 0, HALF\_PI, PIE);

popMatrix();

}

void fig14(){

fill(#E5D01B);

rect(700, 800, 50, 100);

}

void fig15(){

fill(#B9222A);

triangle(650, 400, 650, 700, 950, 400);

}

void fig16(){

pushMatrix();

translate(600, 350);

fill(#B32029);

rotate(5\*PI/4);

rect(0, 0, -(50\*sqrt(2)), 200\*sqrt(2));

popMatrix();

}

void fig17(){

fill(#3178BE);

triangle(650, 400, 850, 400, 850, 200);

}

void fig18(){

fill(#66A040);

rect(850, 200, 100, 200);

}

void fig19(){

fill(#D1422B);

triangle(950, 400, 1150, 400, 950, 200);

}

void fig20(){

pushMatrix();

fill(#E5CE1F);

translate(1075, 400);

arc(0, 0,150, 150, 0, PI, CLOSE);

popMatrix();

}

void fig21(){

pushMatrix();

fill(#DD4E2B);

translate(900, 125);

rotate(PI/3);

arc(0, 0,150, 150, 0, PI, CLOSE);

popMatrix();

}

void draw(){

fig1();

fig2();

fig3();

fig4();

fig5();

fig6();

fig7();

fig8();

fig9();

fig10();

fig11();

fig12();

fig13();

fig14();

fig15();

fig16();

fig17();

fig18();

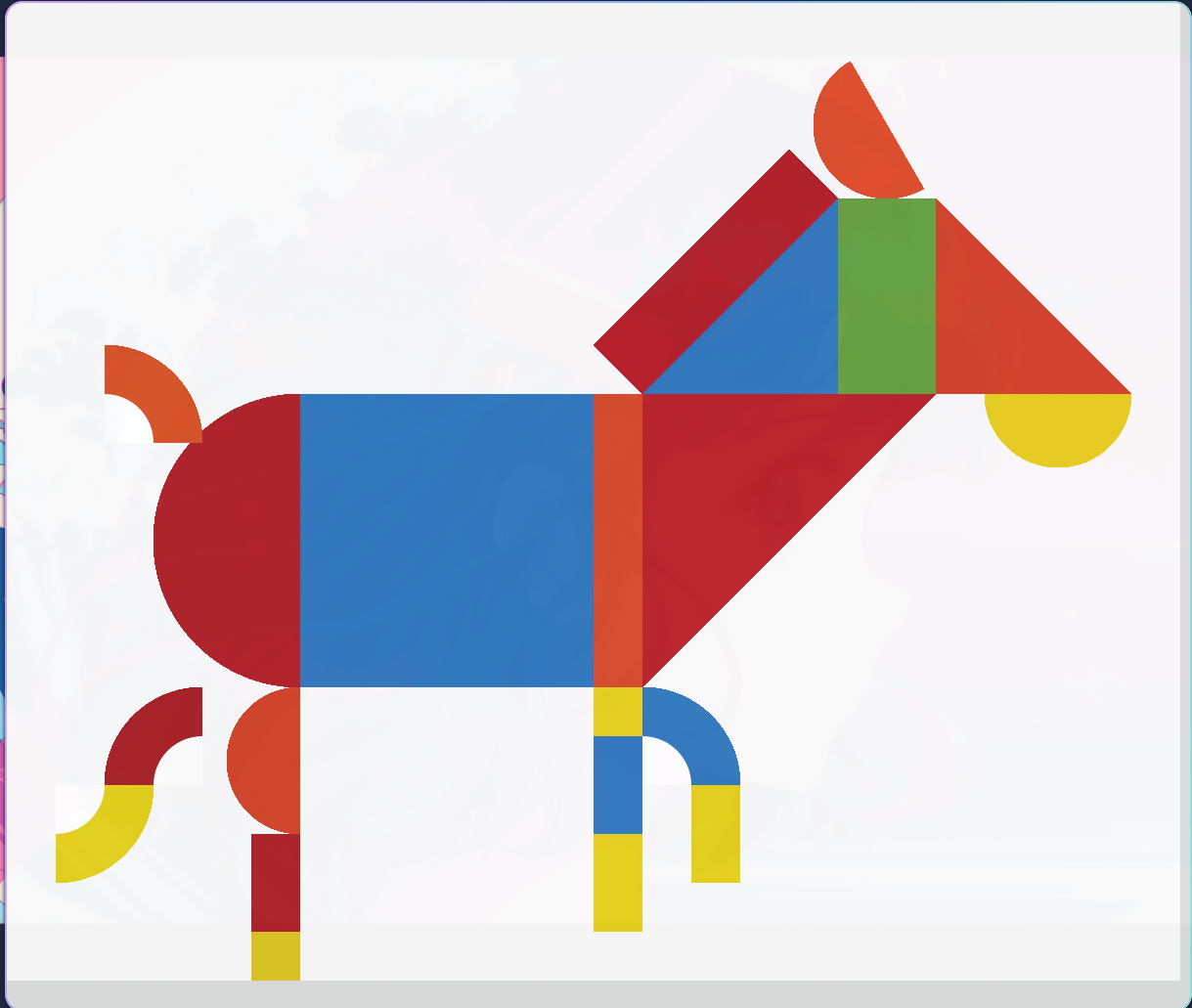
fig19();

fig20();

fig21();

}

1. **Screen printing of the program execution:**

****

1. **Conclusions:**

I can say that this laboratory work introduced me well in 2d primitives from the Processing tool. I implemented some functions to rotate the shapes, some mathematical operations to calculate the length and height of the shape 16.

I got some problems on figures like 5, 6, 7, because I tried to find an already implemented solution in the documentation page where all 2d primitives were listed, but I figured it out on my own. I just combined 2 arcs one on another to get the visual effect. Later, when this option won’t be good, I think I will try to use another functions like vertex that could bend the rect shape.

On some figures I used pushMatrix and popMatrix functions, because they allowed me to draw the figures isolately. Some of the properties from previous figures like fill color could extend on other shapes, so I found this as a suitable solution for me

Reference: <https://www.kidiko.ro/jucarie-din-lemn-tabla-magnetica-cu-forme-geometrice/>