БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

ИИТ БГУИР

Программное обеспечение информационных технологий

Отчет

По лабораторной работе №3

по курсу «Алгоритмы компьютерной графики»

Вариант 12

Выполнил студент группы 481064 Сорока А.А.

Проверил преподаватель: Коренская И.Н.

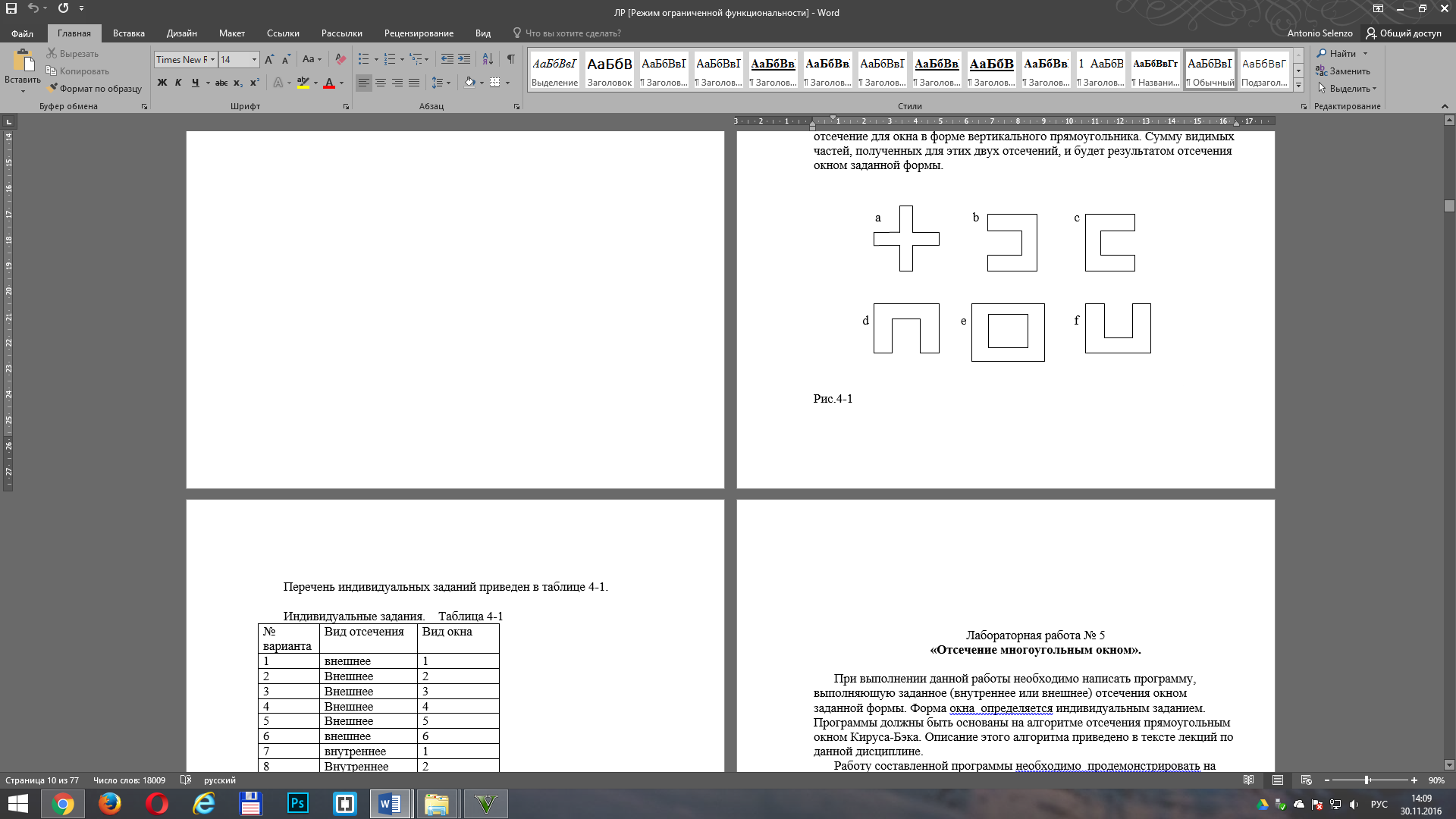
Минск 2016 г

**Цель работы**:

**Задание:**

Необходимо написать программу, выполняющую заданное (внутреннее или внешнее) отсечения окном. Форма окна определяется индивидуальным заданием. Программы должны быть основаны на алгоритме отсечения прямоугольным окном Сазерленда-Коуэна. Описание этого алгоритма приведено в тексте лекций по данной дисциплине. Работу составленной программы необходимо продемонстрировать на примере отсечения перемещающейся фигуры, полученной в результате выполнения лабораторной работы № 1, окном заданной формы.

Индивидуальное задание «внутреннее, f»



**Код программы:**

index.html

<html>

<head>

<script src="script\animation.js"></script>

</head>

<body>

<div style="position: relative;">

<canvas style="border:1px solid #ccc; position: absolute; left: 0; top: 20; z-index: 0;" id="canvas"> </canvas>

<canvas style="position: absolute; left: 0; top: 20; z-index: 1;" id="canvas1"></canvas>

</div>

<div style="position: absolute; top: 5; left: 10px;">

X: <input type="number" id="speedX" value="50" min=-500 max=500 width=30px/> Y: <input type="number" id="speedY" value="50"

min=-500 max=500 width=30px/>

<button id="startButton" onclick="startAnimation()"></button>

</div>

</body>

</html>

animation.js

var ctx, speedX, speedY, x, y, H, W, currentScale, trails, useTrail, animate;

window.onload = function () {

var canvas = document.getElementById('canvas');

ctx = canvas.getContext('2d');

var canvas1 = document.getElementById('canvas1');

W = canvas.width = canvas1.width = window.innerWidth - 25;

H = canvas.height = canvas1.height = window.innerHeight - 50;

var ctx1 = canvas1.getContext("2d");

ctx1.beginPath();

ctx1.fillStyle = "rgba(0, 255, 255, 0)";

ctx1.moveTo(565, 275);

ctx1.lineTo(630, 275);

ctx1.lineTo(630, 425);

ctx1.lineTo(695, 425);

ctx1.lineTo(695, 275);

ctx1.lineTo(760, 275);

ctx1.lineTo(760, 485);

ctx1.lineTo(565, 485);

ctx1.lineTo(565, 275);

ctx1.fill();

ctx1.strokeStyle = "rgba(0, 0, 0, 0.5)";

ctx1.lineWidth = 1;

ctx1.stroke();

var ctx2 = canvas1.getContext("2d");

ctx2.beginPath();

ctx2.fillStyle = "rgba(255, 255, 255, 1)";

ctx2.moveTo(0, 0);

ctx2.lineTo(0, H);

ctx2.lineTo(565, H);

ctx2.lineTo(565, 275);

ctx2.lineTo(630, 275);

ctx2.lineTo(630, 425);

ctx2.lineTo(695, 425);

ctx2.lineTo(695, 275);

ctx2.lineTo(760, 275);

ctx2.lineTo(760, 485);

ctx2.lineTo(565, 485);

ctx2.lineTo(565, H);

ctx2.lineTo(W, H);

ctx2.lineTo(W, 0);

ctx2.fill();

reset();

};

function reset() {

x = 590;

y = 300;

document.getElementById("startButton").textContent = "Start";

currentScale = 1;

trails = [{\_x:x, \_y:y, \_s:currentScale}];

draw(x, y, currentScale);

}

function draw(currentX, currentY, scale) {

currentScale \*= scale;

drawTriangle(currentX, currentY, currentScale, "rgba(27, 44, 245, 0.92)");

}

function drawTriangle(currentX, currentY, scale, color) {

ctx.beginPath();

ctx.fillStyle = color;

ctx.moveTo(currentX + 80 \* scale, currentY);

ctx.lineTo(currentX + 160 \* scale, currentY + 70 \* scale);

ctx.lineTo(currentX + 160 \* scale, currentY + 160 \* scale);

ctx.lineTo(currentX + 80 \* scale, currentY + 230 \* scale);

ctx.lineTo(currentX, currentY + 160 \* scale);

ctx.lineTo(currentX, currentY + 70 \* scale);

ctx.fill();

}

function startAnimation() {

if (animate) {

document.getElementById("startButton").textContent = "Start";

clearTimeout(animate);

animate = null;

}

else {

reset();

animate = true;

document.getElementById("startButton").textContent = "Stop";

speedX = document.getElementById("speedX").value;

speedY = document.getElementById("speedY").value;

useTrail = false

animation();

}

}

function animation() {

var n = 0;

var currentX = x;

var currentY = y;

do {

n = speedX \* (Math.random() \* 4 - 2);

}

while (x + n <= 25 || x + n >= W - 160);

x += n;

do {

n = speedY \* (Math.random() \* 4 - 2);

}

while (y + n <= 95 || y + n >= H - 75);

y += n;

var stepX = (x - currentX) / 50;

var stepY = (y - currentY) / 50;

var k = 0;

function moveToPoint() {

if (k != 50) {

k++;

currentX += stepX;

currentY += stepY;

ctx.clearRect(0, 0, ctx.canvas.width, ctx.canvas.height);

if (useTrail) {

trails.forEach(function(element) {

drawTrail(element.\_x, element.\_y, element.\_s);

}, this);

}

draw(currentX, currentY, 0.9994);

if (animate) {

window.requestAnimationFrame(moveToPoint);

}

}

else {

trails.push({\_x:currentX, \_y:currentY, \_s:currentScale});

animate = setTimeout(animation, 300);

}

}

moveToPoint();

}

**Скриншот выполнения программы:**

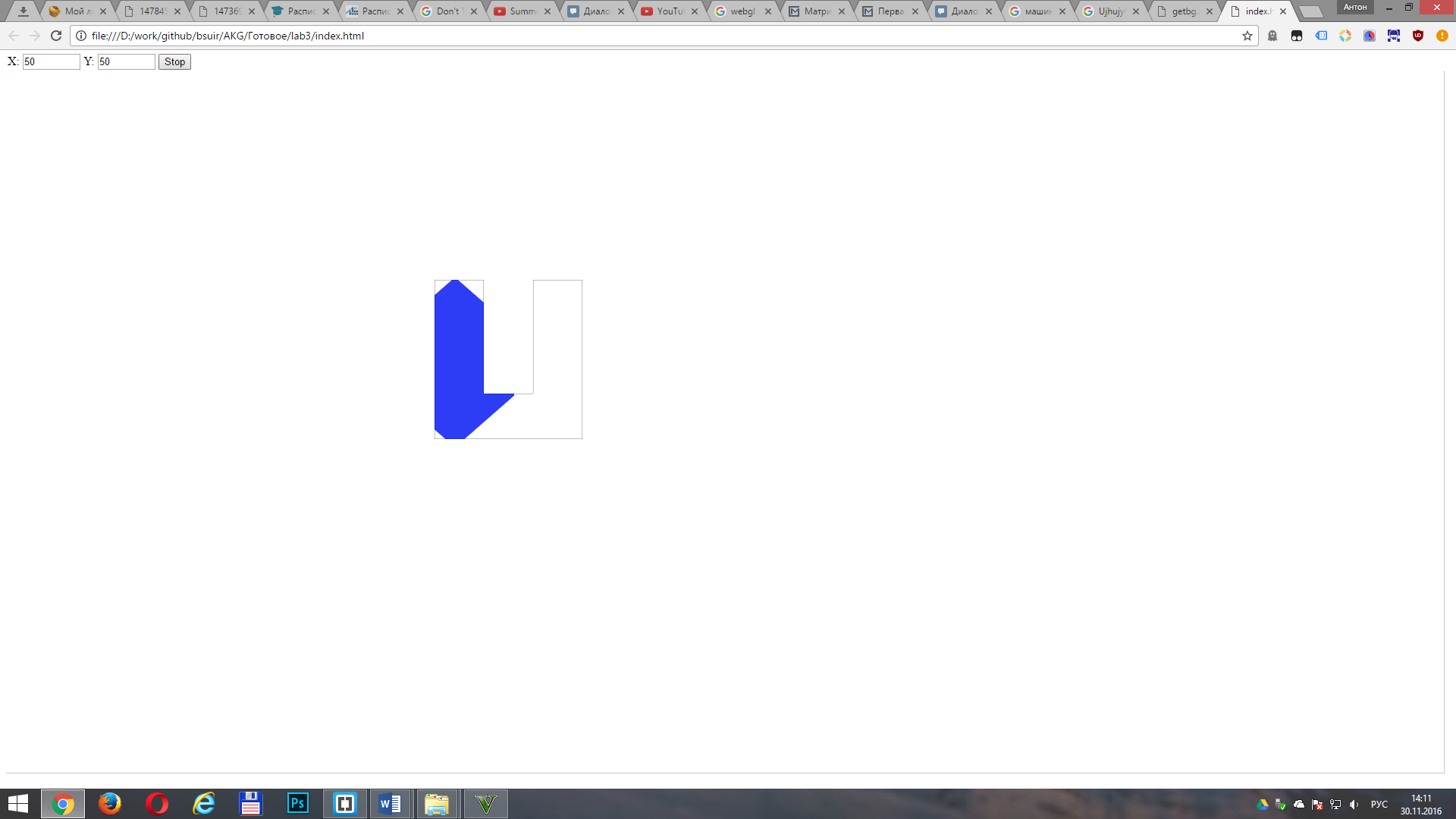


Рисунок 1 – Рабочая область приложения в ходе его работы

**Вывод**: в ходе выполнения лабораторной работы была разработано программное средство на языке JavaScript, служащее для создания анимации движущегося шестиугольника изменяющегося по ходу движения, а так же выполняющее внутреннее отсечение.