Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Кафедра ИПМ

**«Программирование человеко-машинных интерфейсов»**

**Лабораторная работа № 1**

Выполнила: Сорокин Ю.Б.

Группа: Р3317

Преподаватель: Зинчик А.А.

Санкт-Петербург

2016 год

**Цель работы**

Работа с большими массивами пикселов.

**Задание**

1. Написать программу, выполняющую следующее задание. Для произвольного

графического файла формата BMP размером не менее чем 1000х1000 выполнить

следующие действия:

1)Загрузить и отобразить файл.

2) Применить действие, согласно варианту задания. Пользоваться функциями GetPixel

и SetPixel запрещено.

3)Отобразить результат и сохранить новый файл.

2. Оценить отношение скорости работы функции BitBlt и пары функций GetPixel и

SetPixel при решении вашего варианта задания и построить график зависимости этого

отношения от размеров изображения. Размеры изображения от 100х100 до 1000х1000 с

шагом увеличения каждого размера на 100 (200х200, 300х300 и т.д.)

Вариант: Убрать красный и зеленый каналы изображения

**Код программы**

**public class** ImgArea **extends** JPanel {  
  
 **private** BufferedImage image;  
 **private** BufferedImage **oldImage**;  
 **private boolean changed**=**false**;  
 **private boolean saved**=**false**;  
  
 **public** ImgArea(){  
 **this**.setBackground(Color.***WHITE***);  
  
 **try** {  
 image = ImageIO.*read*(**new** File(**"src/img/1000x1000.bmp"**));  
 **oldImage** = ImageIO.*read*(**new** File(**"src/img/1000x1000.bmp"**));  
 } **catch** (IOException ex) {}  
  
 **this**.addMouseListener(**new** MouseAdapter() {  
 @Override  
 **public void** mouseClicked(MouseEvent e) {  
 **super**.mouseClicked(e);  
 **if**(**changed**)  
 repaint();  
 **else** ChangeRGB();  
 }  
 });  
  
 **for**(**int** i = 100;i<=1000;i+=100){  
 BufferedImage TestImage;  
 **try** {  
 TestImage = ImageIO.*read*(**new** File(**"src/img/"**+String.*valueOf*(i)+**"x"**+String.*valueOf*(i)+**".bmp"**));  
 System.***out***.println(i+**"x"**+i+**" "**+ChangeRGBwithGetSet(TestImage)+**" "**+Change(TestImage));  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
  
 }  
 }  
  
 @Override  
 **public void** paintComponent(Graphics g) {  
 **super**.paintComponent(g);  
 **if**(**changed**) {  
 g.drawImage(**oldImage**, getWidth() / 2 - image.getWidth() / 2, getHeight() / 2 - image.getHeight() / 2, **this**);  
 **changed**=**false**;  
 }  
 **else** {  
 g.drawImage(image, getWidth() / 2 - image.getWidth() / 2, getHeight() / 2 - image.getHeight() / 2, **this**);  
 **changed**=**true**;  
 }  
 }  
  
 **private long** ChangeRGBwithGetSet(BufferedImage TestImage){  
 **long** start = System.*currentTimeMillis*();  
 **for** (**int** i=0; i<TestImage.getWidth();i++)  
 **for** (**int** j = 0; j < TestImage.getHeight(); j++) {  
 **int** rgb = TestImage.getRGB(i, j);  
 Color oldColor = **new** Color(rgb);  
 **int** b = oldColor.getBlue();  
 Color newColor = **new** Color(0,0,b);  
 TestImage.setRGB(i, j, newColor.getRGB());  
 }  
 **long** end = System.*currentTimeMillis*();  
 **long** time = end-start;  
 **return** time;  
 }  
  
  
 **private long** Change(BufferedImage TestImage){  
 **long** start = System.*currentTimeMillis*();  
 **int**[] argb = **new int**[TestImage.getWidth()\*TestImage.getHeight()];  
 image.getRGB(0,0,TestImage.getWidth(),TestImage.getHeight(),argb,0,TestImage.getWidth());  
 **for** (**int** i=0; i<argb.**length**;i++)  
 argb[i]=argb[i]&0xFF;  
 TestImage.setRGB(0,0,TestImage.getWidth(),TestImage.getHeight(),argb,0,TestImage.getWidth());  
 **long** end = System.*currentTimeMillis*();  
 **long** time = end-start;  
 **return** time;  
 }  
  
 **private void** ChangeRGB(){  
 **int**[] argb = **new int**[image.getWidth()\*image.getHeight()];  
 image.getRGB(0,0,image.getWidth(),image.getHeight(),argb,0,image.getWidth());  
 **for** (**int** i=0; i<argb.**length**;i++){  
 argb[i]=argb[i]&0xFF;  
 }  
 image.setRGB(0,0,image.getWidth(),image.getHeight(),argb,0,image.getWidth());  
 **if**(!**saved**){  
 **saved** = **true**;  
 **try** {  
 ImageIO.*write*(image, **"PNG"**, **new** File(**"src/img/new.bmp"**));  
 } **catch** (IOException e) {}  
 }  
 repaint();

**Вывод**

При выполнении данной лаб работы, я ознакомился с работой с изображениями, работой с RGB.