

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 17.12.2021 09:42:31

Уникальный идентификатор документа:

0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра вычислительной техники

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Локтионова
«17» декабря 2017 г.



ПОСТРОЕНИЕ ДВУХМЕРНЫХ ФИГУР В PROCESSING.

Методические указания по выполнению лабораторной работы
по дисциплине «Инженерная и компьютерная графика»
для студентов специальности 09.03.01 «Информатика
и вычислительная техника»

Курск 2017

УДК 621.37(075)

Составители: М.В. Бобырь, С.А. Кулабухов

Рецензент

Доктор технических наук, профессор *И.В. Зотов*

1. Построение двумерных фигур в Processing: методические указания по выполнению лабораторной работы по дисциплине «Инженерная и компьютерная графика» / Юго-Зап. гос. ун-т; сост.: М.В. Бобырь, С.А. Кулабухов. – Курск, 2017. – 12 с.: ил. 10, Табл. 2. – Библиогр.: с.12.

Рассмотрены базовые понятия компьютерной графики на основе программирования в Processing. В учебно-методической работе содержатся задания для выполнения практических работ.

Методические указания соответствуют требованиям программы дисциплины «Компьютерная графика».

Предназначены для студентов специальности 09.03.01 «Информатика и вычислительная техника» дневной и заочной форм обучения.

Текст печатается в авторской редакции

Подписано в печать 10.11.18 Формат 60x84 1/16.

Усл.печ. л. 0,6 Уч.-изд. л. 0,5 Тираж 50 экз. Заказ 1839 Бесплатно.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

ПОСТРОЕНИЕ ДВУХМЕРНЫХ ФИГУР В PROCESSING.

1. Цель работы

Изучить основы программирования. Понять основы компьютерной графики и познакомиться со средой разработки Processing. Научиться создавать интерактивную графику на простых примерах.

2. Виды компьютерной графики.

Компьютерная графика, в общем случае, это процесс генерации изображения при помощи компьютера. Кроме того, компьютерной графикой так же называют результат, полученный этой самой «генерацией», то есть все, что мы видим на экране монитора можно отнести к компьютерной графике.

Графика бывает двумерная (2D) и трехмерная (3D). Двумерная графика — это «плоские» изображения, такие как фотографии или рисунки. Трехмерная графика — это уже работа с объектами в трехмерном пространстве.

Двумерная графика, в свою очередь делится на растровую и векторную. Растровая графика — это классический способ представления компьютерной графики, по своей сути — это массив разноцветных пикселей, каждый из которых требует хранения для него цвета и координат (прообразом растровой графики можно считать вышивку крестом, где изображение складывается из цветных крестиков). Векторная графика — это способ представления изображения при помощи математических символов


и примитивов. Растровая и векторная графики в силу своих различий не являются взаимозаменяемыми, так как при помощи растровой графики удобнее всего представлять фотореалистичные изображения, например, фотографии, а при помощи векторной — рисунки и схемы. С одной стороны — несомненным преимуществом векторной графики является возможность масштабирования без потери качества и небольшой вес файла с простыми объектами, а с другой — реалистичное изображение с мелкими деталями будет невероятно большим за счет сложности фигур, с помощью которых оно создано.

Подробнее о каждом виде графики вы можете прочитать в статьях Растровая графика и Векторная графика.

3. Ход выполнения лабораторной работы

Запустите программу Processing. В текстовом редакторе наберите следующее:

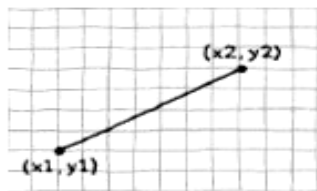
```
ellipse(50, 50, 80, 80);
```

Затем нажмите кнопку Run . В результате выполнения откроется новое окно с изображением круга посередине. Числа в этой команде означают следующее:

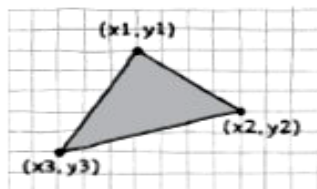
- центр эллипса находится на расстоянии 50 пикселей от левой границы и 50 пикселей от верхней границе;
- ширина и длина эллипса 80 пикселей.

! Нарисуйте эллипс с произвольными координатами и шириной/длиной.

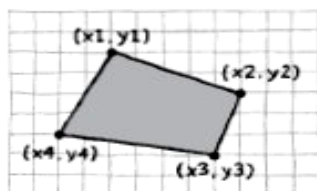
Для рисования простых фигур возможно использование следующих функций:



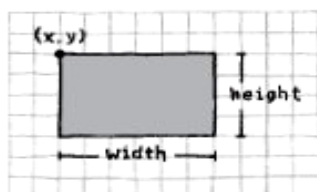
`line(x1, y1, x2, y2)`



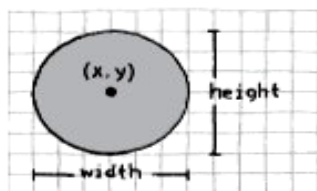
`triangle(x1, y1, x2, y2, x3, y3)`



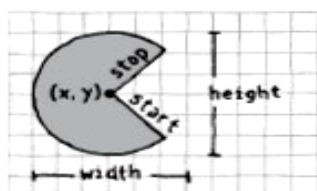
`quad(x1, y1, x2, y2, x3, y3, x4, y4)`



`rect(x, y, width, height)`



`ellipse(x, y, width, height)`



`arc(x, y, width, height, start, stop)`

Рисунок 1 - Способы построения двумерных фигур в processing

Переходим к более сложному тексту:

Введите следующий программный код:

```

void setup()
{
  size(480, 120); // размер окна 500 на 300 пикселей
  smooth();
}
void draw()
{
  if (mousePressed)
  {
    fill(0);      // если мышка нажата, то заливка белый цвет
  }
  else
  {
    fill(255);   // иначе, то заливка черный цвет
  }
  ellipse (mouseX, mouseY, 30, 30); // где курсор мыши
                                     //(координата mouseX, mouseY),
                                     //отобразится круг, если мышь не нажата,
                                     //заливка белый цвет, иначе черный
}
}

```

! Поменяйте цвет заливки в диапазоне от 0 до 255 и поэкспериментируйте в программном коде с длиной/шириной эллипса.

Рисуем точку в системе координат с привязкой к верхнему углу картинки

```
size (200, 100); // размер окна
```

```
point (140, 60); // координаты точки
```

Рисуем геометрические фигуры: линию, треугольник, многоугольник, прямоугольник, круг и арку круга.

! Составьте программный код с помощью которого можно нарисовать 5 фигур с произвольными координатами. Например, что бы нарисовать прямую линию, введите:

Линия:

```

size(500, 300);           // размер окна
line(20, 50, 300, 100); // (20, 50) координата X; (300, 100) координата Y;
                        //начало координат в верхнем левом углу.

```

Эллипс:

```

size (500, 300); // размер окна
ellipse(250, -50, 350, 450);
ellipse(150, 150, 90, 90);
ellipse(430, 50, 15, 15);

```

! Подберите координаты, что бы эллипсы соприкасались по касательной

Часть эллипса:

```

size (500, 300); // размер окна
arc(90, 60, 80, 80, 0, HALF_PI);
arc(190, 60, 80, 80, 0, PI+HALF_PI);
arc(290, 60, 80, 80, PI, TWO_PI+HALF_PI);
arc(390, 60, 80, 80, QUARTER_PI, PI+QUARTER_PI);

```

Введите программный код:

```
size(500, 300);
```

```
ellipse(150, 30, 150, 150);
```

```
rect(190, 50, 300, 20); // Прямоугольник расположен поверх
```

эллипса потому, что в программе он стоит после него.

Если поменять последовательность, то прямоугольник будет расположен под эллипсом:

```
size(500, 300);
```

```
rect(190, 50, 300, 20);
```

```
ellipse(150, 30, 150, 150);
```

Знакомство со свойствами фигур

а) Сглаживание линии:

```

size(500, 300);
smooth(); // Включаем сглаживание
ellipse(150, 50, 100, 100);
noSmooth(); // Выключаем сглаживание
ellipse(250, 50, 100, 100);

```

б) Устанавливаем толщину линий с помощью команды strokeWeight(8):

```

size(500, 300);
smooth();
ellipse(75, 60, 90, 90);
strokeWeight(8); // Расширяем линию до 8 пикселей
ellipse(175, 60, 90, 90);
ellipse(279, 60, 90, 90);
strokeWeight(20); // Расширяем линию до 20 пикселей
ellipse(389, 60, 90, 90);

```

в) Заливка фигур с помощью функции fill(204);

- Фигуры с градацией серого цвета

```

size(500, 300);
smooth();
background(240); // цвет фона светло-серый, если нужно установить цвет фона черный введите 0.
fill(204); // Светло-серый
ellipse(132, 82, 200, 200); // Светло-серый круг
fill(153); // Серый
ellipse(228, -16, 200, 200); // Серый круг
fill(102); // Темно-серый
ellipse(268, 118, 200, 200); // Темно-серый круг

```

- Цветные фигуры

```

size(500, 300);
noStroke();
smooth();
background(0, 26, 51); // Темно-синий цвет
fill(255, 0, 0); // Красный
ellipse(132, 82, 200, 200); // Красный круг
fill(0, 255, 0); // Зеленый
ellipse(228, -16, 200, 200); // Зеленый круг
fill(0, 0, 255); // Синий
ellipse(268, 118, 200, 200); // Синий круг

```


Чтобы установить прозрачность фигуры, в функции fill необходимо ввести 4 параметр в диапазоне от 0 до 255.

4. Задания для самостоятельного выполнения

1. Построить 5 простых двумерных объектов. Объекты выбирать из таблицы 1 согласно варианту. Номера фигур брать из таблицы 2.

Таблица 1

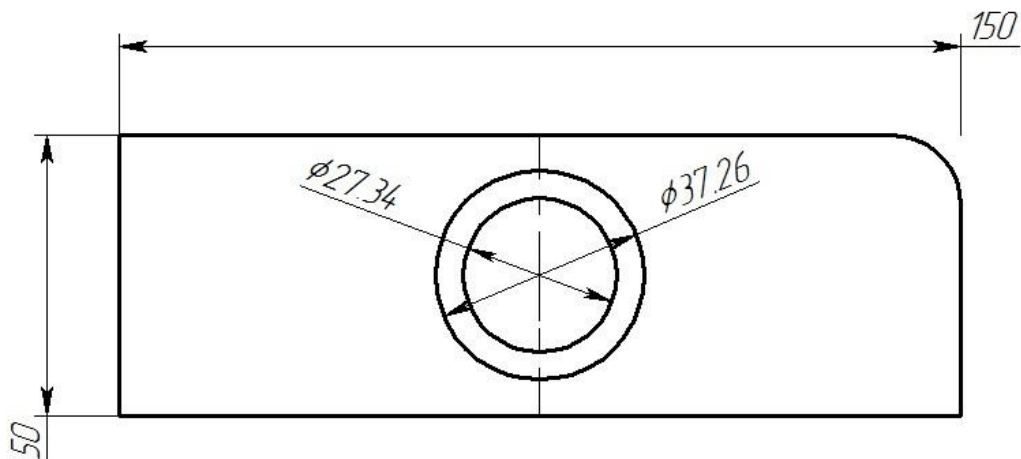
№ варианта	Номера фигур				
1	1	1	2	4	5
2	6	4	1	4	2
3	4	1	5	3	3
4	3	4	1	2	6
5	2	1	2	2	5
6	4	1	4	4	1
7	6	6	2	2	1
8	2	5	1	4	6
9	5	4	2	2	5
10	4	5	3	3	1
11	5	6	6	3	1
12	5	2	6	5	2
13	5	3	2	2	2
14	3	2	3	4	5
15	1	4	2	6	1
16	3	4	1	5	2
17	1	6	2	2	4
18	4	1	3	2	2
19	1	3	4	2	4
20	6	1	3	5	6
21	3	6	6	1	1
22	6	1	1	3	4
23	2	5	5	4	1
24	3	3	5	1	1
25	1	6	6	4	4

Таблица 2

№ фигуры	Наименование фигуры
1	Треугольник
2	Квадрат
3	Четырехугольник
4	Прямоугольная трапеция
5	Круг
6	Неполный круг (рис. 1 фигура 6)

Размеры и положение фигур выбирать произвольно.

2. Нарисуй следующую картину:



! Для рисунка выбрать произвольные размеры ширины и высоты. Диаметр кругов выбрать по номеру варианта.

5. Контрольные вопросы

1. Расскажите о видах компьютерной графики?
2. Какие задачи решает Processing?
3. Какие двумерные объекты позволяет создавать Processing?
4. Как реализован механизм управления цветом в Processing?
5. Как работает механизм сглаживания?

6. Содержание отчёта

Отчёт должен содержать:

- 1) титульный лист;
- 2) наименование работы и цель исследований;
- 3) описание хода выполнения задания;
- 4) изображение построенной фигуры и код программы для ее построения.

7. Библиографический список

1. Кейси Риз и Бен Фрай «Учимся программировать вместе с Processing» перевод с английского Издательская группа ВHV, 2012. - 194 с., ил.