



Урок 20. C++: координаты и пиксели

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 20 из 34 · 45 минут · Модуль «Компьютерная графика на C++»

Уровень: начинающий программист; C++17.

Программа: <https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

У сетки 6 столбцов, но ячейки с номером 6 нет. Создадим изображение, где каждая координата проверяется по клеткам.

Результат урока: Построить пиксельную модель 6×5 из заданных клеток и объяснить границы координат, оси и увеличение.

01. Проблемная ситуация

Пиксельную модель 6×5 предлагают завершить чёрной клеткой(6,5). Другой вариант —(5,4). Начертите сетку, подпишите нулевые индексы и отметьте правильную клетку. Объясните, почему увеличение картинки не добавляет новых исходных клеток.

02. Понятия и разобранный пример

5–14 минут: 4 минуты на понятия, 3 на пример, 2 на объяснение способа партнёру.

Пиксельная модель и векторный файл SVG

В растровом изображении пиксели образуют сетку. Для модели 6×5 допустимы столбцы $x=0\dots5$, строки $y=0\dots4$; начало вверху слева, y растёт вниз. На этом уроке C++ записывает SVG: каждую клетку изображаем прямоугольником 1×1. Это векторный файл с пиксельной моделью, не растровый PNG и не библиотечная команда `putpixel`. Различие формата и изображённой сетки сохраняем явно.

`viewBox="0 0 6 5"` задаёт внутренние координаты; `width=300`, `height=250` увеличивают каждую учебную клетку до квадрата 50×50 экранных единиц. Белый фон рисуется первым, цветные клетки — после него. У `rect` координаты x , y задают левый верхний угол, `width` и `height` — размеры.

Прямоугольник с $x=6$ целиком вне внутренней области.

Заголовок `fstream` и `std::ofstream` позволяют записать текст в `pixels.svg` в рабочей папке. `R"SVG(...)"` — многострочный строковый литерал C++; содержимое между разделителями — данные SVG. Не заменяйте прямые кавычки типографскими. После сборки и запуска откройте файл в обычном браузере локально, например через «Открыть файл». Сеть, аккаунты и отдельная графическая библиотека не нужны. Терминал ничего не печатает: продукт находится в файле.

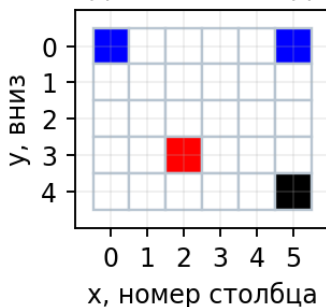
Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

Четыре `rect` задают клетки(0,0),(5,0),(2,3),(5,4). Первые две синие, третья красная, последняя чёрная. Исходных клеток 30, цветных 4, белых 26. Имя файла находится в строке `std::ofstream`; SVG не является C++-кодом внутри литерала.

```
#include <fstream>
int main() {
    std::ofstream out("pixels.svg");
    out << R"SVG(<svg xmlns="http://www.w3.org/2000/svg" width="300" height="250"
        viewBox="0 0 6 5" shape-rendering="crispEdges">
<rect width="6" height="5" fill="white"/>
<rect x="0" y="0" width="1" height="1" fill="blue"/>
<rect x="5" y="0" width="1" height="1" fill="blue"/>
<rect x="2" y="3" width="1" height="1" fill="red"/>
<rect x="5" y="4" width="1" height="1" fill="black"/>
</svg>
)SVG";
    return 0;
}
```



Пиксельная модель 6×5; индексы 0...5 и 0...4



Пиксельная модель 6×5: начало(0,0) сверху слева; синие клетки(0,0),(5,0), красная(2,3), чёрная(5,4).

03. Практика 1. Проверяем материал

Новая карта четырёх цветных клеток

14–25 минут. Сохраните новый исходник для practice.svg. Измените четыре rect по таблице, не меняя размеры фоновой сетки.

Цвет	Клетка новой карты
blue	(1,1) и(4,1)
red	(2,4)
black	(0,3)

2 минуты — клеточный прогноз, 4 — изменение и сборка, 3 — открытие SVG и сравнение, 2 — журнал. На бумаге подпишите минимум один край и направление y. Продукт — исходник, SVG и отмеченные четыре клетки. Критерии: правильный формат файла, четыре цвета и координаты, границы сетки, объяснение увеличения.

04. Практика 2. Перенос способа

Горизонтальное отражение сетки

25–33 минуты. Перенос: отразите вторую карту горизонтально. В сетке из 6 столбцов формула $x' = 5 - x$, $y' = y$. Сначала заполните четыре новые координаты, затем самостоятельно создайте mirror.svg. Критерии: отражение, а не сдвиг; сохранены цвета и строки; все клетки допустимы.

05. Обсуждение и границы вывода

33–40 минут. Можно ли назвать файл pixels.svg растровым только потому, что в нём показана сетка? Сравните формат хранения, исходную модель и размер показа. Объясните, что изменится при увеличении width до 600 и height до 500.

06. Индивидуальная проверка

40–45 минут. Для сетки 9×4 назовите правую нижнюю клетку, число клеток и формулу отражения x. Объясните, почему $x=9$ недопустим как индекс. Четыре критерия по 1 баллу.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Красная клетка во второй карте находится в последней строке, но не в последнем столбце; чёрная — в первом столбце строки 3. После изменения имени в ofstream новый файл practice.svg нужно открыть отдельно, а не смотреть старый pixels.svg. Полное решение:

```
#include <fstream>
int main() {
    std::ofstream out("practice.svg");
    out << R"SVG(<svg xmlns="http://www.w3.org/2000/svg" width="300" height="250"
        viewBox="0 0 6 5" shape-rendering="crispEdges">
<rect width="6" height="5" fill="white"/>
<rect x="1" y="1" width="1" height="1" fill="blue"/>
<rect x="4" y="1" width="1" height="1" fill="blue"/>
<rect x="2" y="4" width="1" height="1" fill="red"/>
<rect x="0" y="3" width="1" height="1" fill="black"/>
</svg>
)SVG";
    return 0;
}
```

Практика 2: разбор и критерии

Синие(4,1) и(1,1), красная(3,4), чёрная(5,3). Последний столбец 5 переходит в 0, а 0 — в 5. Формула $6-x$ ошибочна: при $x=0$ даёт недопустимый 6. Код:

```
#include <fstream>
int main() {
    std::ofstream out("mirror.svg");
    out << R"SVG(<svg xmlns="http://www.w3.org/2000/svg" width="300" height="250"
        viewBox="0 0 6 5" shape-rendering="crispEdges">
<rect width="6" height="5" fill="white"/>
<rect x="4" y="1" width="1" height="1" fill="blue"/>
<rect x="1" y="1" width="1" height="1" fill="blue"/>
<rect x="3" y="4" width="1" height="1" fill="red"/>
<rect x="5" y="3" width="1" height="1" fill="black"/>
</svg>
)SVG";
    return 0;
}
```

Обсуждение: возможный вывод

Файл остаётся векторным, содержит описания прямоугольников. Модель остаётся 6×5 , каждая клетка будет показана как 100×100 ; новых исходных клеток не появится. Пиксель экрана, учебная клетка и элемент SVG — разные понятия, даже когда рисунок выглядит клеточным.

Индивидуальная проверка: ответ и критерии

(8,3), 36 клеток, $x'=8-x$. Индексы начинаются с 0, девятый столбец имеет номер 8, а не 9.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, рабочую папку с правом записи, редактор, распечатку ученической части PDF, бумагу и ручку. Локальный компилятор G++ с C++17 или настроенная учителем IDE; сборка `g++ -std=c++17 -Wall -Wextra -pedantic main.cpp -o lesson`. В Windows задайте lesson.exe. Запуск .lesson в Linux/macOS, .lesson.exe в Windows. Для 20–22 дополнительно обычный браузер: открывать локальный SVG, не C++-исходник. Все входы и ожидаемые результаты даны. Установку и проверку запуска учитель выполняет до занятия; детские аккаунты и интернет не нужны.

Уровень — начинающий программист, 7 класс. Предпосылки: структура C++, сборка программы и вывод текста. В паре автор вводит и объясняет, проверяющий сопоставляет с прогнозом, затем роли меняются. Без компьютера трассируйте полный код на бумаге, отмечайте координаты по схеме и проверяйте готовые результаты; учитель демонстрирует заранее собранный пример. Бумажная трассировка не выдаётся за реальную компиляцию или запуск ученика. План 45 минут включает чтение, ввод, сборку и обсуждение. В графических задачах сначала прогноз, затем сравнение. Матрица программы предназначена учителю.



Минуты	Работа	Результат
0–5	Проблема	Вопрос и первое решение
5–14	Понятия и пример	Способ проверки
14–25	Практика 1	Готовый продукт и проверка
25–33	Практика 2	Перенос способа
33–40	Обсуждение	Аргумент и ограничение
40–45	Индивидуальная проверка	Самостоятельный ответ

07. Соответствие исходной программе

Пункт программы	Действие ученика	Проверяемый результат
Система координат, управление пикселями	Задаёт цветные клетки в пиксельной модели SVG	Изображение 6×5 и проверенные индексы
Изменение координат	Отражает клеточную карту	Сохранены строки и допустимые границы

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

W3C SVG2: координатные системы и viewBox

<https://www.w3.org/TR/SVG2/coords.html>

C++: строковые литералы

<https://eel.is/c++draft/lex.string>

C++: файловые потоки

<https://eel.is/c++draft/input.output>

Тексты, учебные данные, реплики, программы и схемы подготовлены для уроков портала «Школы России». Числовые серии — учебные модели, не результаты проведённых классом опытов. Схемы авторские, сторонние фотографии не используются. Методические ссылки приведены для учителя; ученик получает все материалы здесь. Проверка первичных источников: 17.10.2026.