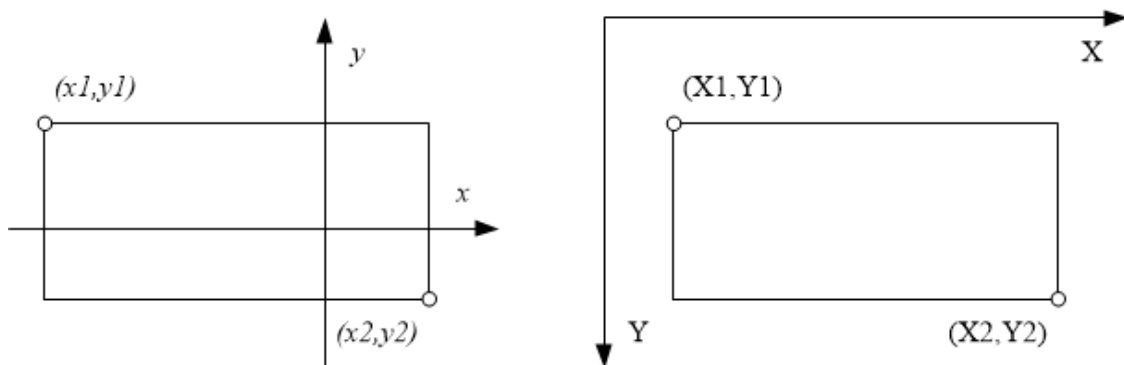


JUL: Фрактальное множество Жюлиа

Математическое моделирование: преобразование координат.

Рассмотрим комплексную плоскость и точки $z = (x, y)$ на ней. Изучим простейший итерационный процесс вида $z \rightarrow z^2 + c$, где $c = (p, q)$ комплексный параметр. Итерационный процесс в зависимости от значения параметра может а) иметь конечный предел, б) просто быть ограниченным, но не сходящимся, либо в) стремиться к бесконечности. Ваша задача — для конкретного c отобразить множество точек z , для которых итерационный процесс не расходится, т.е. реализуется случай а или б. Оно называется множеством Жюлиа.



Координатные системы

Для того чтобы отобразить часть комплексной плоскости на дискретном экране, необходимо установить взаимно-однозначное соответствие между двумя прямоугольниками в этих системах координат: $(x_1, y_1) \dots (x_2, y_2)$ и $(X_1, Y_1) \dots (X_2, Y_2)$. Делается это преобразование исходя из простейшей пропорции (для Y аналогично):

$$\frac{X - X_1}{X_2 - X_1} = \frac{x - x_1}{x_2 - x_1}.$$

Важно не запутаться и сопоставить одинаковые точки этих прямоугольников, например, левой верхней точке прямоугольника на математической плоскости должна соответствовать левая верхняя точка экранного прямоугольника. Тот факт, что экранная ось Y направлена вниз, не должен приводить к перевероту картинки на экране.

Работа по шагам:

1. **Нарисуйте оси (1 балл).** Установите соответствие между **левой половиной экрана** и математической областью $-2 \dots 2$ по x и $-3 \dots 3$ по y . Для этого заведите структуры `point_t` для вещественной точки, `rect_t` для прямоугольной области, а также функцию

```
point_t Transform(point_t p, rect_t const* from, rect_t const* to);
```

которая преобразует координаты точки из одной произвольной прямоугольной области в другую, возвращая новые координаты. Напишите функцию

```
void DrawAxes(rect_t const* math, rect_t const* screen);
```

в которой возьмите центр математических координат (ноль) и переведите его в экранные, а затем в полученной точке нарисуйте оси. Убедитесь, что они на месте, так вы проверите преобразование из математических координат на экран.

2. **Закрасьте круг (+1 балл).** Нарисуйте множество точек — внутренность единичного диска, — следующим образом. Пробегитесь **по всем точкам левой половины экрана** и поставьте только те, которые в комплексной плоскости лежат внутри окружности радиуса $R = 1$. Оформите цикл в виде функции

```
void DrawDisk(rect_t const* math, rect_t const* screen);
```

которая в цикле преобразует экранные точки в математические и производит проверку принадлежности единичному кругу в отдельной функции

```
labbool_t IsInsideDisk(point_t p);
```

Убедитесь, что круг рисуется в нужном месте и нужного размера, так вы проверите преобразование из экранных координат в математические.

3. **Закрасьте множество Жюлиа (+1 балл).** На **правой половине экрана** отобразите ту же часть комплексной плоскости, но для определения принадлежности множеству используйте более сложную функцию, получив тем самым функцию рисования множества Жюлиа

```
void DrawJulia(rect_t const* math, rect_t const* screen);  
labbool_t IsInsideJulia(point_t p);
```

Для характерного изображения возьмите константу $c = (-0.12375, 0.56508)$. Напоминаю, что точка принадлежит множеству Жюлиа в том случае, если итерационный процесс не расходится, т.е. точка бесконечно долго остаётся в ограниченной области:

- Задайтесь предельным числом итераций $N_{\max} = 100$ и радиусом $R_{\max} = 2$, возьмите интересующую вас точку $z = (x, y)$.
- Организуйте цикл $z \rightarrow z^2 + c$, который прерывается по одному из двух условий: либо превышено число итераций N_{\max} (тогда процесс считаем ограниченным), либо точка z вышла за пределы радиуса R_{\max} (и тогда процесс считаем расходящимся). Помните о правилах умножения комплексных чисел!

4. (*) **Добавьте анимацию (+1 бонус).** Оберните вызов DrawJulia в цикл, на каждой итерации которого используйте своё значение параметра. Меняйте его по закону $c = (r \sin \alpha, r - r \cos \alpha)$, где $r = 0.32$, а α постоянно увеличивается. Вам придется передавать этот параметр c в функцию рисования.