MAN: Множество Мандельброта

Математическое моделирование: интерполяция.

Возьмём за основу предыдущую работу. Множеством Жюлиа для комплексного параметра с называлось множество таких точек комплексной плоскости z, для которых итерационный процесс $z \to z^2 + c$ не расходился к бесконечности. Множеством Мандельброта называется множество **комплексных параметров** c, для которых итерационный процесс, начинающийся с $z = 0$, не расходится (остаётся ограниченным). Оно изображается не в плоскости $z = (x, y)$, а в плоскости параметров $c = (p, q)$.

- В множестве Жюлиа фиксировали c, рисовали плоскость значений z_0.
- В множестве Мандельброта фиксируем z_0, рисуем плоскость значений c.

Этапы работы:

0. **Изучите и поправьте решение предыдущей задачи (0 баллов)**. Вам предлагается готовое решение (шаблон) предыдущей задачи JUL. При возникновении ошибок компиляции (из-за несоответствия стандарту Си) внесите необходимые исправления. Отключите анимацию. Переместите отображение Жюлиа в левую часть экрана. Видим два одинаковых множества. Как и в прошлой работе пока для Жюлиа возьмите $c = (-0.12375,\ 0.56508)$, и область $X \times Y = [-2,\ 2] \times [-3,\ 3]$.

1. **Нарисуйте множество Мандельброта (+1 балл).** Измените программу так, чтобы в левой половине экрана отображалась **внешность** множества Жюлиа, а в правой — внешность множества Мандельброта.

   - Функцию `IsInside` заменитт на `IsOutside` (исправьте и название, и наполнение). Снаружи ставьте белую точку, внутри **не трогайте** цвет по умолчанию (чёрный).

   - Скопируйте функцию в `IsOutsideMandelbrot`, которую вызовите для правой половины экрана. Для Мандельброта возьмите область $P \times Q = [-2.75,\ 1.25] \times [-3,\ 3]$, внесите необходимые изменения, в остальном код такой же, что в Жюлиа (при $z = 0$).

2. **Добавьте полосатую раскраску (+1 балл).** Раскрасьте внешность множества в зависимости от количества итераций, потребовавшихся для того, чтобы добраться до “бесконечности”.

   - Положите $N_{\max} = 1023$, $R_{\max} = 2$.

   - Из функции `IsOutside` возвращайте не просто `TRUE`/`FALSE`, а в случае `TRUE` характеристику того, как близко точка находится к множеству — количество шагов $N_{\max} - i$ (`>0`, `!=FALSE`).

   - Нечётные шаги закрашивайте белым, чётные **синим** (`LABCOLOR_DARK_CYAN`). Внутренность остаётся неизменно **чёрной**.

   - После того, как всё заработает, возьмите для Жюлиа параметр $c = (-0.835,\ 0.2321)$, а для Мандельброта отобразите область

$[-0.7454356,\ -0.7454215] \times [0.1129986,\ 0.113019]$. И слева, и справа будут видны интересные узоры.

3. **Реализуйте плавную раскраску (+1 балл).**

   – Вычислите безразмерный параметр $t = n/(N_{\max} + 1) \in [0,1)$, поправьте его, возведя в куб: $t \to t^3 \in [0,1)$, чтобы добавить нужную нам нелинейность (подобрано эмпирически для данной задачи).

   – С помощью параметра t получите цвет из палитры из $K_{\max}$ цветов, заданной преподавателем, для чего отобразить интервал $[0,1)$ на $[0, K_{\max} - 1)$. При этом параметру t будет соответствовать целый номер цвета k в палитре и вещественный остаток $\alpha \in [0,1)$. Этот остаток используйте для интерполяции между соседними цветами в палитре: $\mathrm{RGB} = (1 - \alpha)\mathrm{RGB}[k] + \alpha\mathrm{RGB}[k + 1]$.

   – Используйте функцию `LabSetColorRGB(r, g, b)` для выбора цвета, чтобы увидеть красивое сине-голубое множество Жюлиа и желто-оранжевое множество Мандельброта.

```
static color_t s_palette[] = {
  {0x00, 0x00, 0xFF}, {0x00, 0xFF, 0xFF}, {0xFF, 0xFF, 0x00},
  {0xFF, 0x00, 0x00}, {0xFF, 0xFF, 0x00}, {0x00, 0xFF, 0xFF},
  {0x00, 0x00, 0xFF},
};
```