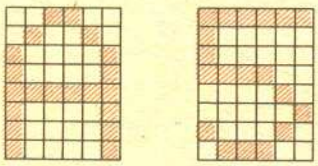
Rysowanie prostej nie jest proste

Dr Michał JANKOWSKI

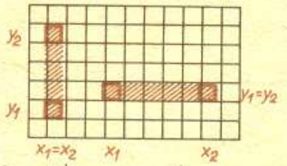
Narysowanie prostej przechodzącej przez dwa dane punkty na płaszczyźnie może sprawić kłopoty co najwyżej przedszkolakom. Czy to zadanie będzie równie łatwe, gdy kartkę papieru zamienimy na monitor (ekran) komputera?

Wszystko, co oglądamy na monitorze, a więc litery, cyfry i linie, powstaje przez wyświetlenie wyglądających jak malutkie kropki najmniejszych części ekranu nazywanych z angielskiego pikslami.

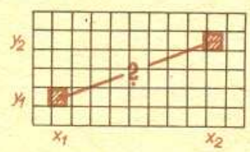


Rys. 1

Na rysunku 1 pokazujemy w powiększeniu, jak z piksli można zbudować literę *A* i cyfrę 5. Czytelnicy, którzy znają ZX Spectrum, wiedzą, że dla tego mikrokomputera ekran dzieli się na 176 wierszy i 256 kolumn, czyli składa się z 45056 piksli. Profesjonalne monitory używane w grafice komputerowej mają znacznie większą rozdzielczość, np. 4096×4096, i bardzo dużo kolorów, ale i odpowiednio wyższą cenę. Te urządzenia, tak jak i zwykłe odbiorniki TV, są przykładami tzw. rastrowych urządzeń graficznych. Rysowanie na nich możemy porównać do rysowania na kartce z zeszytu w kratkę, gdzie zamiast robienia cienkich kresek wolno nam tylko zamalowywać całe kratki. Zadanie narysowania odcinka wygląda teraz zupełnie inaczej. Dwie ustalone kratki (albo piksle ekranu) o współrzędnych całkowitych równych numerom wiersza i kolumny mamy połączyć zamalowując odpowiednie kratki (albo piksle) między nimi. Rozwiązanie jest oczywiste, jeśli oba końce odcinka leżą w tej samej kolumnie lub w tym samym wierszu (rys. 2*a* i 2*b*). W innych

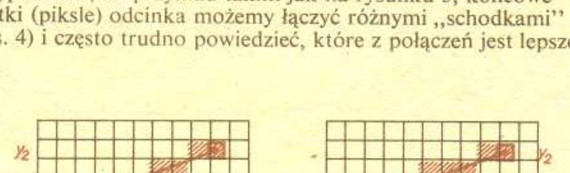


Rys. 2



Rys. 3

przypadkach, na przykład takim jak na rysunku 3, końcowe kratki (piksle) odcinka możemy łączyć różnymi „schodkami" (rys. 4) i często trudno powiedzieć, które z połączeń jest lepsze.



Rys. 4

Spróbujmy podać ogólne rozwiązanie. Niech (x_1, y_1) i (x_2, y_2) będą końcami rysowanego odcinka. Przy klasycznym rysowaniu na płaszczyźnie byłyby to punkty o tych współrzędnych, przy komputerowym rysowaniu na ekranie numery wierszy i kolumn, w których leżą odpowiednie piksle, a wreszcie w naszym modelu z kartką papieru w kratkę są to współrzędne (tak, jak w grze w okręty) końcowych kratek odcinka. Zakładamy, że $x_1 < x_2$. Równanie odcinka możemy zatem zapisać w postaci

$$y-y_1 = \frac{y_2-y_1}{x_2-x_1}(x-x_1), \qquad x \in [x_1, x_2]$$

lub

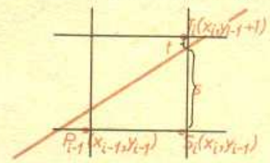
$$y = \frac{\mathrm{d}y}{\mathrm{d}x}x+b = ax+b, \qquad x \in [x_1, x_2],$$

gdzie wprowadziliśmy oznaczenia

$$\mathrm{d}y = y_2-y_1, \ \mathrm{d}x = x_2-x_1, \ a = \frac{\mathrm{d}y}{\mathrm{d}x}, \ b = y_1-\frac{\mathrm{d}y}{\mathrm{d}x}x_1.$$

Narzuca się stąd następujący algorytm rysowania odcinka:
— obliczamy wartości rzeczywiste a i b,
dla kolejnych wartości x zmieniających się co 1 od x_1 do x_2:
— obliczamy liczbę rzeczywistą $r = ax+b$,
— zaokrąglamy r do liczby całkowitej y,
— zamalowujemy kratkę (wyświetlamy piksel) o współrzędnych (x, y).

Istotną wadą tego algorytmu jest wykonywanie działań na liczbach rzeczywistych, co w większości komputerów jest znacznie kosztowniejsze (tzn. zajmuje dużo więcej czasu) niż działania na liczbach całkowitych. A przecież zarówno końce odcinka, jak i wszystkie zamalowywane kratki mają współrzędne całkowite. Własność tę wykorzystuje algorytm Bresenhama, który niżej opiszemy. Zakładamy dla uproszczenia, że współczynnik kierunkowy odcinka — a — jest zawarty między 0 i 1. Na rysunku 5 punkt P_{i-1} jest środkiem kolejnej kratki (piksla) tworzącej rysowany odcinek. W następnym kroku wybieramy tę kratkę,



Rys. 5

której środek leży w pionie bliżej rzeczywistego odcinka (narysowanego kolorem). Warunek ten możemy sformułować następująco: jeśli $s-t < 0$, to wybieramy kratkę o środku S_i, a w przeciwnym razie kratkę o środku T_i (inne kratki nie wchodzą w grę, bo $0 < a < 1$). Idea algorytmu jest bardzo prosta, ale czy można podejmować decyzję o wyborze S_i lub T_i wykonując tylko działania na liczbach całkowitych? Aby się o tym przekonać, zauważmy (patrz rys. 5), że

$$s = \frac{\mathrm{d}y}{\mathrm{d}x}(x_i-x_1)-(y_{i-1}-y_1), \quad t = 1-s,$$

a stąd

$$\mathrm{d}x \cdot (s-t) = 2\mathrm{d}y \cdot x_i - 2\mathrm{d}x \cdot y_{i-1} - 2\mathrm{d}y \cdot x_1 + 2\mathrm{d}x \cdot y_1 - \mathrm{d}x.$$

Oznaczmy tę wielkość przez d_i. Ponieważ $\mathrm{d}x > 0$, więc znak $\mathrm{d}_i = \mathrm{d}x \cdot (s-t)$ decyduje o wyborze następnych kratek. Między d_{i+1} i d_i zachodzi zależność

$$\mathrm{d}_{i+1}-\mathrm{d}_i = 2\mathrm{d}y \cdot (x_{i+1}-x_i) - 2\mathrm{d}x \cdot (y_i-y_{i-1}),$$

ale $x_{i+1}-x_i = 1$, a więc

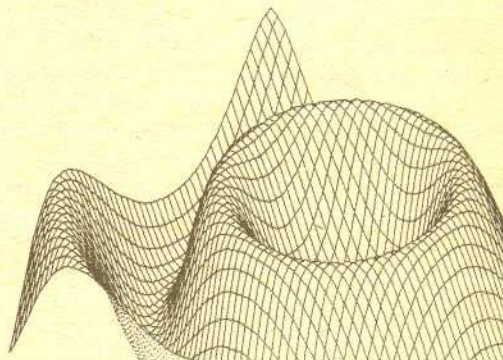
$$\mathrm{d}_{i+1} = \mathrm{d}_i + 2\mathrm{d}y - 2\mathrm{d}x \cdot (y_i-y_{i-1})$$

i, jak łatwo sprawdzić,

$$\mathrm{d}_1 = 2\mathrm{d}y-\mathrm{d}x.$$

Potrafimy zatem obliczać kolejne d_i i możemy już zapisać algorytm Bresenhama. Ma on następującą postać:
— obliczamy $\mathrm{d}_1 = 2\mathrm{d}y-\mathrm{d}x$,
dla $i = 1, 2, \ldots, x_2-x_1$:
— jeśli $\mathrm{d}_i < 0$, to {wybieramy jako następną kratkę o środku S_i, wtedy $y_i = y_{i-1}$ i obliczamy $\mathrm{d}_{i+1} = \mathrm{d}_i + 2\mathrm{d}y$},
— w przeciwnym razie {wybieramy jako następną kratkę o środku T_i, wtedy $y_i = y_{i-1}+1$ i obliczamy $\mathrm{d}_{i+1} = \mathrm{d}_i + 2(\mathrm{d}y-\mathrm{d}x)$}.

Jeśli przebrnąłeś Czytelniku przez te proste (?) rachunki dla rysowania prostych, to spróbuj się zastanowić nad rysowaniem (oczywiście na urządzeniu rastrowym) na przykład okręgu lub wykresów takich powierzchni jak przedstawiona na rysunku 6.



Rys. 6