**Szakaszrajzoló algoritmusok**

**DDA**

**ELJÁRÁS DDA(SZIN S, EGÉSZ X0, EGÉSZ Y0, EGÉSZ X1, EGÉSZ Y1);**

**VÁLTOZÓK**

**VALÓS: DX, DY, HOSSZ, NX, NY, X, Y;**

**EGÉSZ: I;**

**ALGORITMUS**

**DX <- X1 - X0;**

**DY <- Y1 - Y0;**

**HOSSZ <- ABS(DX);**

**HA (HOSSZ < ABS(DY)) AKKOR**

**HOSSZ <- ABS(DY);**

**HA\_VÉGE;**

**NX <- DX / HOSSZ;**

**NY <- DY / HOSSZ;**

**X <- X0;**

**Y <- Y0;**

**PIXEL(KEREKÍT(X), KEREKÍT(Y), S);**

**CIKLUS I <- 1..HOSSZ**

**X <- X + XN;**

**Y <- Y + YN;**

**PIXEL(KEREKÍT(X), KEREKÍT(Y), S);**

**CIKLUS\_VÉGE;**

**ELJÁRÁS\_VÉGE;**

**Midpoint (0° <= alfa <= 45°)**

**ELJÁRÁS MP\_SZAKASZ\_V1(SZIN S, EGÉSZ X0, EGÉSZ Y0, EGÉSZ X1, EGÉSZ Y1);**

**VÁLTOZÓK**

**EGÉSZ: D, DY, DX, X, Y, I;**

**ALGORITMUS**

**DX <- X1 - X0;**

**DY <- Y1 - Y0;**

**D <- 2 \* DY - DX;**

**X <- X1;**

**Y <- Y1;**

**CIKLUS I <- 1..DX**

**PIXEL(X, Y, S);**

**HA (D > 0) AKKOR**

**Y <- Y + 1;**

**D <- D + 2 \* (DY - DX);**

**KÜLÖNBEN**

**D <- D + 2 \* DY;**

**HA\_VÉGE;**

**X <- X + 1;**

**CIKLUS\_VÉGE;**

**ELJÁRÁS\_VÉGE;**

**Midpoint**

**ELJÁRÁS MP\_SZAKASZ\_V2(SZIN S, EGÉSZ X0, EGÉSZ Y0, EGÉSZ X1, EGÉSZ Y1);**

**VÁLTOZÓK**

**EGÉSZ: D, DY, DX, R, SX, SY I, X, Y;**

**LOGIKAI: T;**

**ALGORITMUS**

**DX <- ABS(X1 - X0);**

**DY <- ABS(Y1 - Y0);**

**SX <- SGN(X1 - X0);**

**SY <- SGN(Y1 - Y0);**

**HA (DX < DY) AKKOR**

**R <- DX;**

**DX <- DY;**

**DY <- R;**

**T <- IGAZ;**

**KÜLÖNBEN**

**T <- HAMIS;**

**HA\_VÉGE;**

**D <- 2 \* DY - DX;**

**X <- X0;**

**Y <- Y0;**

**PIXEL( X, Y, S);**

**CIKLUS\_AMÍG ((X <> X1) VAGY (Y <> Y1))**

**HA (D > 0) AKKOR**

**HA (T = IGAZ) AKKOR**

**X <- X + SX;**

**KÜLÖNBEN**

**Y <- Y + SY;**

**HA\_VÉGE;**

**D <- D - 2 \* DX;**

**HA\_VÉGE;**

**HA (T = IGAZ) AKKOR**

**Y <- Y + SY;**

**KÜLÖNBEN**

**X <- X + SX;**

**HA\_VÉGE;**

**D <- D + 2 \* DY;**

**PIXEL( X, Y, S);**

**CIKLUS\_VÉGE;**

**ELJÁRÁS\_VÉGE;**

**Körrajzoló algoritmus**

**Körpontok tükrözése**

**ELJÁRÁS KORPONT(SZIN S, EGÉSZ X, EGÉSZ Y);**

**ALGORITMUS**

**PIXEL( X, Y, S);**

**PIXEL( X, -Y, S);**

**PIXEL(-X, Y, S);**

**PIXEL(-X, -Y, S);**

**PIXEL( X, Y, S);**

**PIXEL( X, -Y, S);**

**PIXEL(-X, Y, S);**

**PIXEL(-X, -Y, S);**

**ELJÁRÁS\_VÉGE;**

**Midpoint (valós aritmetikával)**

**ELJÁRÁS MP\_KOR\_V1(EGÉSZ R, SZIN S);**

**VÁLTOZÓK**

**EGÉSZ: X, Y;**

**VALÓS: D;**

**ALGORITMUS**

**X <- 0;**

**Y <- R;**

**D <- 5 / 4 - R;**

**KORPONT(X, Y, S);**

**CIKLUS\_AMÍG (Y > X)**

**HA (D < 0) AKKOR**

**D <- D + 2 \*X + 3;**

**KÜLÖNBEN**

**D <- D + 2 \* (X - Y) + 5;**

**Y <- Y - 1;**

**HA\_VÉGE;**

**X <- X + 1;**

**KORPONT(X, Y, S);**

**CIKLUS\_VÉGE;**

**ELJÁRÁS\_VÉGE;**

**Midpoint**

**ELJÁRÁS MP\_KOR\_V2(EGÉSZ R, SZIN S);**

**VÁLTOZÓK**

**EGÉSZ: X, Y, H;**

**ALGORITMUS**

**X <- 0;**

**Y <- R;**

**H <- 1 - R;**

**KORPONT(X, Y, S);**

**CIKLUS\_AMÍG (Y > X)**

**HA (H < 0) AKKOR**

**H <- H + 2 \*X + 3;**

**KÜLÖNBEN**

**H <- H + 2 \* (X - Y) + 5;**

**Y <- Y - 1;**

**HA\_VÉGE;**

**X <- X + 1;**

**KORPONT(X, Y, S);**

**CIKLUS\_VÉGE;**

**ELJÁRÁS\_VÉGE;**

**Vágó algoritmusok**

**Cohen-Sutherland: Végpontokhoz rendelt kód**

**FÜGGVÉNY VEGPONT\_KOD(PONT P): EGÉSZ;**

**VÁLTOZÓK**

**EGÉSZ: KOD;**

**ALGORITMUS**

**KOD <- 0;**

**HA (P.X < ABLAK.BAL) AKKOR**

**KOD <- KOD VAGY 1;**

**KÜLÖNBEN**

**HA (P.X > ABLAK.JOBB) AKKOR**

**KOD <- KOD VAGY 2;**

**HA\_VÉGE;**

**HA\_VÉGE;**

**HA (P.Y < ABLAK.FONT) AKKOR**

**KOD <- KOD VAGY 4;**

**KÜLÖNBEN**

**HA (P.Y > ABLAK.LENT) AKKOR**

**KOD <- KOD VAGY 8;**

**HA\_VÉGE;**

**HA\_VÉGE;**

**VEGPONT\_KOD <- KOD;**

**FÜGGVÉNY\_VÉGE;**

**Cohen-Sutherland (vázlatos)**

**ELJÁRÁS COHEN\_SUTH\_VAZL(PONT P0, PONT P1)**

**VÁLTOZÓK**

**EGÉSZ: KOD0, KOD1;**

**LOGIKAI: ELFOGAD;**

**ALGORITMUS**

**KOD0 <- VEGPONT\_KOD(P0);**

**KOD1 <- VEGPONT\_KOD(P1);**

**ELFOGAD <- HAMIS;**

**CIKLUS\_AMÍG (IGAZ)**

**HA ((KOD0 VAGY KOD1) = HAMIS) AKKOR**

**ELFOGAD = IGAZ;**

**CIKLUS\_KILÉP;**

**KÜLÖNBEN**

**HA ((KOD0 ÉS KOD1) = IGAZ) AKKOR**

**CIKLUS\_KILÉP;**

**KÜLÖNBEN**

**AZ\_EGYIK\_KINTLÉVŐ\_PONT\_VÁGÁSA;**

**A\_PONTHOZ\_TARTOZÓ\_KÓD\_ÚJRASZÁMÍTÁSA;**

**HA\_VÉGE**

**HA\_VÉGE;**

**CIKLUS\_VÉGE;**

**HA (ELFOGAD = IGAZ) AKKOR**

**SZAKASZ(P0, P1);**

**HA\_VÉGE;**

**ELJÁRÁS\_VÉGE;**

**Cohen-Sutherland**

**ELJÁRÁS COHEN\_SUTH\_VAZL(PONT P0, PONT P1)**

**VÁLTOZÓK**

**EGÉSZ: KOD0, KOD1, KOD\_KINN, X, Y;**

**LOGIKAI: ELFOGAD;**

**ALGORITMUS**

**KOD0 <- VEGPONT\_KOD(P0);**

**KOD1 <- VEGPONT\_KOD(P1);**

**ELFOGAD <- HAMIS;**

**CIKLUS\_AMÍG (IGAZ)**

**HA ((KOD0 VAGY KOD1) = HAMIS) AKKOR**

**ELFOGAD = IGAZ;**

**CIKLUS\_KILÉP;**

**KÜLÖNBEN**

**HA ((KOD0 ÉS KOD1) = IGAZ) AKKOR**

**CIKLUS\_KILÉP;**

**KÜLÖNBEN**

**HA (KOD\_0 = IGAZ) AKKOR**

**KOD\_KINN = KOD\_0;**

**KÜLÖNEBEN**

**KOD\_KINN = KOD\_1;**

**HA\_VÉGE;**

**HA ((KOD\_KINN ÉS 1) = IGAZ) AKKOR**

**X = ABLAK.BAL;**

**Y = P0.Y + (P1.Y – P0.Y) \* (ABLAK.BAL – P0.X) / (P0.X – P0.X);**

**KÜLÖNBEN**

**HA ((KOD\_KINN ÉS 2) = IGAZ) AKKOR**

**X = ABLAK.JOBB;**

**Y = P0.Y + (P1.Y – P0.Y) \* (ABLAK.JOBB – P0.X) / (P0.X – P0.X);**

**KÜLÖNBEN**

**HA ((KOD\_KINN ÉS 4) = IGAZ) AKKOR**

**Y = ABLAK.FONT;**

**X = P0.X + (P1.X – P0.X) \* (ABLAK.FONT – P0.Y) / (P0.y – P0.Y);**

**KÜLÖNBEN**

**HA ((KOD\_KINN ÉS 8) = IGAZ) AKKOR**

**Y = ABLAK.LENT;**

**X = P0.X + (P1.X – P0.X) \* (ABLAK.LENT – P0.Y) / (P0.Y – P0.Y);**

**HA\_VÉGE;**

**HA\_VÉGE;**

**HA\_VÉGE;**

**HA\_VÉGE;**

**HA (KOD\_0 = IGAZ) AKKOR**

**P0.X = X;**

**P0.Y = Y;**

**KOD0 <- VEGPONT\_KOD(P0);**

**KÜLÖNBEN**

**P1.X = X;**

**P1.Y = Y;**

**KOD1 <- VEGPONT\_KOD(P1);**

**HA\_VÉGE;**

**HA\_VÉGE**

**HA\_VÉGE;**

**CIKLUS\_VÉGE;**

**HA (ELFOGAD = IGAZ) AKKOR**

**SZAKASZ(P0, P1);**

**HA\_VÉGE;**

**ELJÁRÁS\_VÉGE;**

**Alakzat kitöltő algoritmusok**

**Él flag algoritmus**

**ELJÁRÁS KITÖLT\_ÉL\_FLAG(SZIN: HÁTTÉR, SZÍN SZ);**

**VÁLTOZÓK**

**EGÉSZ: X, Y;**

**LOGIKAI: BENN;**

**ALGORITMUS**

**CIKLUS Y <- Y\_MIN..Y\_MAX**

**CIKLUS X <- X\_MIN..X\_MAX**

**BENN <- HAMIS;**

**HA (SZÍN(X, Y) <> HÁTTÉR) AKKOR**

**BENN <- NEM BENN;**

**HA\_VÉGE;**

**HA (BENN = IGAZ) AKKOR**

**PIXEL(X, Y, SZ);**

**HA\_VÉGE;**

**CIKLUS\_VÉGE;**

**CIKLUS\_VÉGE;**

**ELJÁRÁS\_VÉGE;**

**Rekurzív 4 algoritmus**

**ELJÁRÁS KITÖLT\_REK4(SZÍN: HÁTTÉR, SZÍN SZ, EGÉSZ X, EGÉSZ Y);**

**ALGORITMUS**

**HA (SZÍN(X, Y) = HÁTTÉR) AKKOR**

**PIXEL(X, Y, SZ);**

**KITÖLT\_REK( X, +Y, SZ);**

**KITÖLT\_REK( X, -Y, SZ);**

**KITÖLT\_REK(+X, Y, SZ);**

**KITÖLT\_REK(-X, Y, SZ);**

**HA\_VÉGE;**

**ELJÁRÁS\_VÉGE;**

**Rekurzív 8 algoritmus**

**ELJÁRÁS KITÖLT\_REK8(SZÍN: HÁTTÉR, SZÍN SZ, EGÉSZ X, EGÉSZ Y);**

**ALGORITMUS**

**HA (SZÍN(X, Y) = HÁTTÉR) AKKOR**

**PIXEL(X, Y, SZ);**

**KITÖLT\_REK( X, +Y, SZ);**

**KITÖLT\_REK( X, -Y, SZ);**

**KITÖLT\_REK(+X, Y, SZ);**

**KITÖLT\_REK(-X, Y, SZ);**

**KITÖLT\_REK(+X, +Y, SZ);**

**KITÖLT\_REK(+X, -Y, SZ);**

**KITÖLT\_REK(-X, -Y, SZ);**

**KITÖLT\_REK(-X, +Y, SZ);**

**HA\_VÉGE;**

**ELJÁRÁS\_VÉGE;**

**4 irányú verem algoritmus**

**ELJÁRÁS KITÖLT\_verem4(SZÍN: HÁTTÉR, SZÍN SZ, EGÉSZ X, EGÉSZ Y);**

**VÁLTOZÓK**

**EGÉSZ[]: DX, DY, I, NX, NY;**

**VEREM[EGÉSZ, EGÉSZ]: V;**

**ALGORITMUS**

**DX <- [0, 1, 0, -1];**

**DY <- [-1, 0, 1, 0];**

**V.PUSH(X, Y);**

**CIKLUS\_AMÍG(V.POP(X, Y))**

**PIXEL(X, Y, SZ);**

**CIKLUS I <- 1..4**

**NX <- X + DX[I];**

**NY <- Y + DY[I];**

**HA (SZÍN(X, Y) = HÁTTÉR) AKKOR**

**V.PUSH(NX, NY);**

**HA\_VÉGE;**

**CIKLUS\_VÉGE;**

**CIKLUS\_VÉGE;**

**ELJÁRÁS\_VÉGE;**

**Poligon algoritmus**

**STRUKTÚRA \_ET //Éltábla**

**EGÉSZ: X;**

**\_ETL: ETL;**

**\_ET: KÖVETKEZŐ;**

**STRUKTÚRA\_VÉGE;**

**STRUKTÚRA \_ET //Éltáblához tartozó éllista**

**EGÉSZ: X1, X2;**

**VALÓS: M;**

**\_ETL: KÖVETKEZŐ;**

**STRUKTÚRA\_VÉGE;**

**STRUKTÚRA \_AET //Aktív éltábla**

**VALÓS: X, M;**

**EGÉSZ: Y2;**

**\_AETL: KÖVETKEZŐ;**

**STRUKTÚRA\_VÉGE;**