/\* Fonction setQuadPerspective \*/

/\*La fonction "setQuadPerspective" vérifie si un quadrilatère est convexe en utilisant une méthode par force brute : en calculant les produits croisés des vecteurs formés par ses côtes pour chaque combinaison de quatre points d'un tableau "monQuad". Elle verifie egalement que les quatre points du quadrilatère sont distincts et que le point en haut à gauche a une valeur y plus élevée que le point en bas à gauche, que le point en haut à droite a une valeur y plus élevée que le point en bas à droite, que le point en haut à gauche a une valeur x plus basse que le point en haut à droite et que le point en bas à gauche a une valeur x plus basse que le point en bas à droite. Elle crée plusieurs boucles et vérifie un grand nombre de combinaisons.\*/

function setQuadPerspective() {

let i, j, k, l;

let topLeftX = -1,

topLeftY = -1,

bottomLeftX = -1,

bottomLeftY = -1,

topRightX = -1,

topRightY = -1,

bottomRightX = -1,

bottomRightY = -1;

compoint = 0; //On vérifie que chaque point est bien distinct

/\*Pour vérifier que les diagonales d'un quadrilatère ABCD se coupent en son sein ( === quadrilatère convexe) on compare les deux valeurs

[(xC-xA)(yB-yA)-(yC-yA)(xB-xA)]\*[(xD-xA)(yB-yA)-(yD-yA)(xB-xA)] et

[(xA-xC)(yD-yC)-(yA-yC)(xD-xC)]\*[(xB-xC)(yD-yC)-(yB-yC)(xD-xC)]

\*/

for (i = 0; i < 7; i = i + 2) {

for (j = 0; j < 7; j = j + 2) {

if (j != i) {

for (k = 0; k < 7; k = k + 2) {

if (k != j && k != i) {

for (l = 0; l < 7; l = l + 2) {

if (l != i && l != j) {

if (l != k) {

var cond1 =

((monQuad[j] - monQuad[i]) \*

(monQuad[k + 1] - monQuad[i + 1]) -

(monQuad[j + 1] - monQuad[i + 1]) \*

(monQuad[k] - monQuad[i])) \*

((monQuad[l] - monQuad[i]) \*

(monQuad[k + 1] - monQuad[i + 1]) -

(monQuad[l + 1] - monQuad[i + 1]) \*

(monQuad[k] - monQuad[i]));

var cond2 =

((monQuad[i] - monQuad[j]) \*

(monQuad[l + 1] - monQuad[j + 1]) -

(monQuad[i + 1] - monQuad[j + 1]) \*

(monQuad[l] - monQuad[j])) \*

((monQuad[k] - monQuad[j]) \*

(monQuad[l + 1] - monQuad[j + 1]) -

(monQuad[k + 1] - monQuad[j + 1]) \*

(monQuad[l] - monQuad[j]));

if (cond1 < 0 && cond2 < 0) {

/\*En prenant TL(TopLeft), TR(TopRight), BL(BottomLeft) et BR(BottomRight)

Voici les conditions à remplir pour dessiner un quadrilatère sur le Canvas

-le point TL doit être plus haut que le point BL

-le point TR doit être plus haut que le point BR

-le point TL doit être à gauche du point TR

-le point BL doit être à gauche du point BR\*/

if (

monQuad[i + 1] < monQuad[l + 1] &&

//le point TL doit être plus haut que le point BL

monQuad[j + 1] < monQuad[k + 1]

//le point TR doit être plus haut que le point BR

) {

if (

monQuad[i] < monQuad[j] &&

//le point TL doit être à gauche du point TR

monQuad[l] < monQuad[k]

//le point BL doit être à gauche du point BR

) {

if (

monQuad[i + 1] < monQuad[k + 1] &&

//le point TL doit être à gauche du point TR

monQuad[j + 1] < monQuad[l + 1]

//le point BL doit être à gauche du point BR

) {

/\* Assignation des valeurs triées\*/

bottomRightX = monQuad[k];

bottomRightY = monQuad[k + 1];

topRightX = monQuad[j];

topRightY = monQuad[j + 1];

bottomLeftX = monQuad[l];

bottomLeftY = monQuad[l + 1];

topLeftX = monQuad[i];

topLeftY = monQuad[i + 1];

}

}

}

}

}

}

}

}

}

}

}

}

//Purge et réaffectation du conteneur monQuad

monQuad = [];

monQuad.push(topLeftX);

monQuad.push(topLeftY);

monQuad.push(topRightX);

monQuad.push(topRightY);

monQuad.push(bottomRightX);

monQuad.push(bottomRightY);

monQuad.push(bottomLeftX);

monQuad.push(bottomLeftY);

for (var xy in monQuad) {

if (monQuad[xy] === -1) {

return false;

}

}

return true;

}