

# 4 - TRANSFORMATIONS DU PLAN

## Document et code utile au démarrage

- Transformations du plan
- Transformations du plan (suite)
- Programme de démarrage pour ce tp

## 1) Ecriture matricielle d'application linéaires en coordonnées homogènes

Ecrire les matrices représentant les transformations suivantes en coordonnées homogènes:

- Identite, Similitude, Translation, Rotation, Homothetie, EtirementX, EtirementY, Projection sur une droite parallèlement à un vecteur, Symétrie glissée, Symétrie orthogonale.
- Calculer l'image d'un point : appliquer une matrice aux coordonnées homogènes du point.
- Afficher pour chaque transformation, l'image de ces figures.

## 2) Composition d'applications

1. Quelles sont les transformations qui permettent de passer d'une figure à l'autre dans cette séquence de figures?
2. Ecrire un programme qui construit ces figures:
  - en utilisant les méthodes de définition et concaténation de transformation déjà définies
  - en utilisant vos propres méthodes de calcul matriciel de l'exercice précédent
3. Afficher les figures successivement et interactivement.

## 3) Transformation bifocale

- Afficher une grille de 20x20 éléments comme dans la figure ci-dessous.
- Appliquer une transformations bifocale sur les éléments de la grille. On peut prendre une fenêtre de 800x800 et pour les paramètres de la transformation bifocale prendre pour commencer  $a = 0.5$ ,  $b = 0.7$
- Modification interactive des valeurs respectives de  $a$  et de  $b$ . Par exemple, vous pouvez :
  - utiliser la molette de la souris pour augmenter/diminuer les valeurs respectives de  $a$  et  $b$  ou/et
  - afficher les axes correspondant à  $x=a$ ,  $x=b$  et les déplacer à la souris pour modifier  $a$  et  $b$ .

## 4) Transformation fisheye

1. Afficher une grille de 20x20 éléments comme dans la figure ci-dessous.
2. Appliquer une transformations fisheye sur les éléments de la grille calculée en coordonnées cartésiennes, en prenant comme centre de la transformation les coordonnées du pointeur souris.
3. Programmer les callbacks utiles pour que l'on puisse modifier interactivement les valeurs respectives de  $d$ , paramètre de la déformation fisheye vue en cours.

## 5) Jeu de trioker

1. Afficher l'ensemble des 24 pièces du trioker dans une partie de la fenêtre.
2. Interactions - il s'agit de rendre interactives les opérations suivantes
  - translation d'une pièce pour l'amener à une position donnée en utilisant la souris
  - rotation d'une pièce par rapport à son centre et un angle variable  $\alpha$
  - rotation d'une pièce par rapport à l'un des sommets et un angle variable  $\alpha$
  - ajustement automatique d'une pièce à côté d'une autre lorsque la pièce est suffisamment proche en distance et en orientation de l'autre.
3. Afficher alternativement chacun de ces puzzles. NB : les exemples ne contiennent que les sommets du contour du puzzle.
4. En utilisant les interactions définies en 2, rendre possible la construction de l'un de ces puzzles.
5. Questions subsidiaires :
  - Calculer automatiquement la triangulation affichée pour chaque puzzle et rendre possible l'affichage/le masquage de cette triangulation.
  - Lorsque la triangulation est affichée, mettre en œuvre un mécanisme interactif d'attraction pour que lorsque le joueur déplace une pièce vers le puzzle, l'ajustement de la pièce à la maille la plus proche maille soit facilité sans pour autant que le déplacement des pièces s'en trouve entravé.
  - Proposer une esquisse de gameplay d'un jeu à  $n$  joueurs exploitant les interactions réalisées.