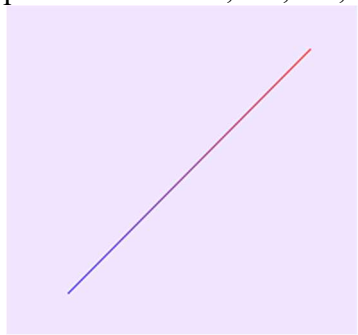
LIFAMI – TP : Interpolation

Objectifs : Notion d'interpolation linéaire, bilinéaire, etc.

La base

1. Ecrivez une fonction qui prend en paramètres deux valeurs et un coefficient d'interpolation entre 0 et 1 et qui calcule l'interpolation (la moyenne pondérée) des deux valeurs.

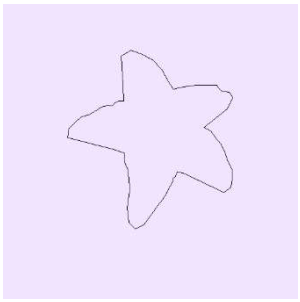
2. Ecrivez une fonction qui dessine une droite entre A et B avec un dégradé du bleu au rouge.
   Indications : la droite D paramétrique est
   $$D(t) = \vec{A} + t.\overrightarrow{AB}.$$
   Quand t=0, la fonction donne le point A.
   Quand t=1, la fonction donne le point B.
   Quand t=0.5, la fonction donne le point milieu entre A et B
   Etc.
   On découpera la droite en 10 portions avec t=0, 0.1, 0.2, 0.3, … 0.9, 1



3. Ecrivez la procédure qui affiche un carré dont les côtés sont alignés avec les axes, et qui sera rempli par interpolation bilinéaire des couleurs des 4 sommets.

Morphing ou comment changer de forme

Un polygone 2D est défini par un nombre *n* de points et un tableau *p* de points. Un point est défini par deux coordonnées réelles.



4. Ecrivez la structure *Point* et la structure *Polygon.* Pour la structure *Point* vous pouvez réutiliser la structure complexe du *TP-Complexe* en la renommant. Copiez/Collez également tous les opérateurs +, -, *. Les champs de la structure *Polygon* doivent s'appeler *p* pour le tableau et *n* pour le nombre de points.

5. Sur la page de l'UE, récupérez les fichiers *poly1.h*, *poly2.h*, etc. Rangez les dans le même dossier que votre fichier *TP_interpolation.cpp*. Dans votre code, après avoir déclaré les structures ajoutez

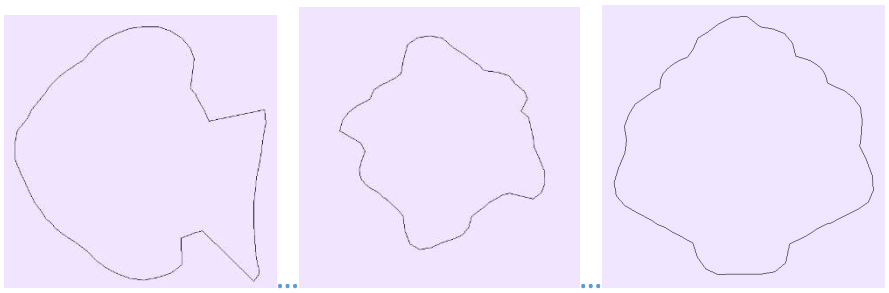
```
#include "poly1.h"
```

   Vérifiez qu'elles marchent bien avec votre code. Sinon renommez vos champs.

6. Ecrivez la procédure *draw_polygon* qui affiche un polygone avec des lignes. Vous pouvez utiliser la fonction *line* qui trace une ligne.

7. Ecrivez la procédure *interpolation_polygon* qui calcule un nouveau polygone interpolé à partir de 2 polygones en entrée et d'un poids.

8. Utilisez la procédure précédente pour afficher une animation déformant le polygone 1 vers le polygone 2. Indication : utilisez la fonction *elapsedTime()* qui renvoie le temps passé depuis le lancement du programme et trouvez une fonction périodique qui pourrait faire osciller l'animation d'un polygone à l'autre.

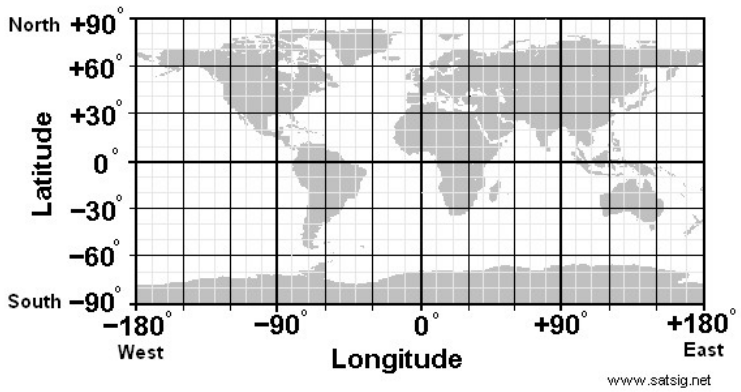


9. Ecrivez une procédure qui déplace une soucoupe volante (un cercle) le long d'un polygone.

Froid et chaud (Pour aller plus loin)

Une station météo est représentée par une position et une mesure de température à un temps donné. La position est une coordonnée longitude et latitude en degré. La température est mesurée en degré Celcius.

Les données de cet exercice sont issues de la NASA et indiquent un écart par rapport à la température moyenne de la station, cette mesure est appelée l'anomalie et permet de voir si la zone se réchauffe ou se refroidit :
http://data.giss.nasa.gov/gistemp/maps/



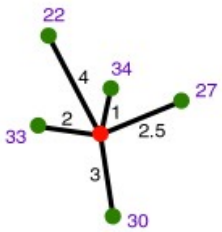
10. Ecrivez une structure *WeatherStation* qui comprend la longitude/latitude en degré, les coordonnées en pixel et la température ; ainsi que la structure stockant toutes les stations météo dans un tableau.

    Sur la page de l'UE, vous trouverez la fonction *init* qui initialise toutes les stations, ainsi que l'image du monde pour la question suivante.

11. Ecrivez une fonction qui convertit une coordonnée longitude/latitude en une coordonnée pixel.

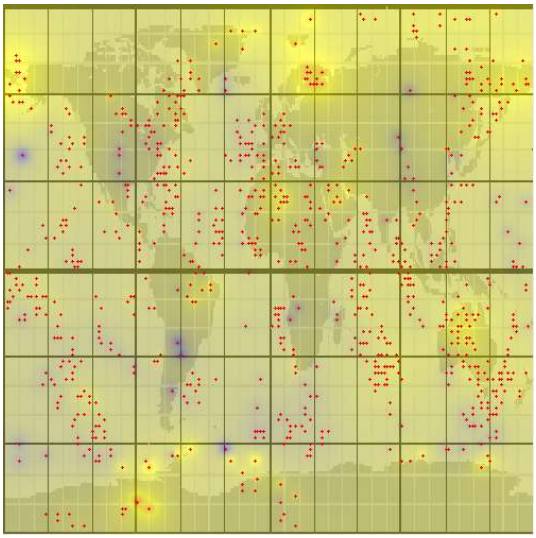
12. Ecrivez une procédure qui affiche la carte du monde, ainsi que toutes les stations météo, par exemple avec un cercle.

13. Ecrivez une fonction qui calcule par interpolation la température de n'importe quel point du monde repéré par ses coordonnées. Pour ca vous allez calculer la moyenne pondérée de toutes les stations du monde. Le poids devra dépendre de la distance, une station proche du point devra avoir un poids élevé, une station éloignée un poids faible, voire très faible. Une approche classique est de donner comme poids l'inverse de la distance : 1/d. Attention quand d=0 !! Cette méthode s'appelle *Inverse distance weighting*.



$$Z(x) = \frac{\Sigma w_i z_i}{\Sigma w_i} = \frac{\frac{34}{1^2} + \frac{33}{2^2} + \frac{27}{2.5^2} + \frac{30}{3^2} + \frac{22}{4^2}}{\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{2.5^2} + \frac{1}{3^2} + \frac{1}{4^2}} = 32.38$$

14. Ecrivez la procédure qui affiche la température de tous les pixels en surimpression sur toute la carte du monde. Utilisez le champ A de la fonction *color(R,G,B,A)*.



Rechauffement de la planète : en bleu la température moyenne a baissé de maximum 4°C jusqu'à rouge où la température a augmenté de 10°C en passant par le jaune avec une augmentation de 3°C

15. (Pour aller plus loin) Calculez et tracez la courbe des températures moyenne en fonction de la latitude.

Pour aller au bout du monde …

16. Ecrivez une fonction qui interpole chaque pixel de deux images pour passer d'une image à l'autre avec un fondu. Ceci n'interpole que la couleur, pas la forme de l'objet.

17. Agrandir une image de taille (L,H) vers (n x L, m x H) par interpolation linéaire