Le miniprojet est à faire par groupes de deux ou individuellement. Les groupes peuvent s'échanger des idées ou des approches générales, mais pas du code directement.

Partie 1. Visualisation de *seams* simples

(a) Créez une image de 5×5 pixels dont vous indiquez vous-même directement toutes les valeurs (à choisir librement entre 0 et 255) avec la fonction **`new_image_grey_with_data`** de **`miniprojectutils`**.

(b) Enregistrez cette image au format PNG avec la fonction **`save_image`**. Enregistrez-la également au format SVG avec les détails de chaque pixel avec la fonction **`save_image_greyscale_details`**. Comparez les deux images pour vérifier que les niveaux de gris correspondent et que les valeurs indiquées soient bien celles que vous avez

(c) Créez un *seam* tout simple manuellement, par exemple comme ceci: **`seam = [3, 3, 2, 3, 4]`**. Enregistrez ensuite ce *seam* avec l'image générée en le passante comme argument supplémentaire à la fonction **`save_image_greyscale_details`**. Vérifiez que le résultat correspond à vos attentes.

Partie 2. Recherche du meilleur *seam*

Implémentez les quatre fonctions que la fonction préimplémentée **`find_seam`** appelle selon les instructions ci-dessous.

Au fur et à mesure que vous avancez, testez votre fonction sur de très petits exemples. Pour cela, en amont, mettez en commentaire l'appel à la fonction **`seam_carving`** après la condition **`if __name__ == "__main__"`** et ajoutez dessous du code qui crée une image aléatoire de taille 3×3. Si nécessaire, ajouter des **`print`** et utilisez le débuggeur pour voir si tout se passe comme prévu.

(a) Implémentez **`find_seam__init_pixel_data`**.

Cette fonction reçoit une image (qui sera le résultat du filtre Sobel). Utilisez-la pour créer une liste de listes de **`PixelData`** afin d'avoir exactement un **`PixelData`** par pixel de l'image originale. (Les valeurs des champs seront pour l'instant bien sûr les valeurs par défaut.)

Parcourez ensuite la première liste de **`PixelData`** pour faire en sorte de stocker le coût initial de chaque pixel (via le champ **`min_energy`**) de la première ligne à la valeur de gris correspondante dans l'image. En effet, pour la première ligne, pas de comme ou de calcul à faire: comme c'est le pixel de départ, le coût cumulé jusqu'à ce pixel est par définition la valeur en termes de niveau de gris de ce pixel.

(b) Implémentez **`find_seam__fill_pixel_data`**.

Parcourez les lignes suivantes en passant par chaque pixel. Déterminez les valeurs de champs **`min_energy`** et **`parent`** selon l'algorithme discuté au cours: il s'agit de regarder lequel des trois (ou deux, dans les bords) pixels potentiels de la rangée précédentes fournissent le meilleur parcours.

Cette fonction ne retourne rien, elle se contente de modifier le contenu de **`cells`**.

(c) Implémentez **`find_seam__find_min_column_on_last_row`**.

Dans cette fonction, il faut chercher sur la dernière ligne l'index de colonne du pixel qui a la plus petite **`min_energy`** de sa ligne.

(d) Implémentez **`find_seam__backtrack_seam`**.

Pour terminer le travail, cette fonction doit faire le *backtracking*: remontez les pixels de bas en haut en suivant les indications des champs **`parent`** et créez le *seam* final en tant que liste de **`int`**s comme décrits au cours.

Attention: vous parcourez les lignes de bas en haut pour le *backtracking*, mais le *seam* doit indiquer les index de colonnes de haut en bas de l'image.

(e) Finalement, comme dernière étape, décommentez dans la fonction **`seam_carving`** le code des parties 4 à 6 et décommentez l'appel à cette fonction tout en bas du fichier, après la condition **`if __name__ == "__main__"`**. Les images ci-dessous vous montrent ce que vous devriez obtenir après réduction de la largeur de 200 unités pour chacune des images tests.

(Notez que suivant l'ordre dans lequel vous testez les pixels prédécesseurs et selon la compression JPEG utilisée, c'est possible que votre code soit correct même si vous n'avez pas exactement les mêmes valeurs pour chaque pixel.)

Les instructions pour soumettre votre fichier Python seront communiquées via Moodle en cours de semaine prochaine.