Dans ce mini projet, nous devions créer des effets sur des images en un temps limité. Trois effets étaient demandés, les deux premiers étaient imposés alors que le troisième était libre. Le premier effet consistait à réaliser un fondu progressif de l’image vers le blanc. Pour le deuxième qui s’intitulait « inversion mouvante » nous devions mettre en mouvement une barre de 30 pixels de la gauche de l’image vers la droite à l’intérieur de laquelle une permutation locale du bleu et du rouge des pixels devait apparaitre. Dans le troisième et dernier effet nous avons pris l’initiative de faire apparaitre 50 carrés faisant office de filtres modifiant localement la couleur de l’image. La position des carrés est strictement aléatoire et la couleur du filtre également. La fin de cet effet est une forme de balayage de la gauche vers la droite de l’image qui fait disparaitre les carrés en réaffichant l’image d’origine.

**Algorithmes**

**Début du programme principal**

Déclarer les images 1, 2 et 3 ;

Charger l’image 1 ;

Demander à l’utilisateur s’il veut « commencer l’effet 1 ? » ;

Si oui : appeler la procédure de l’effet 1 ;

Si non : passer à la ligne suivante ;

Charger l’image 2 ;

Demander à l’utilisateur s’il veut « colorier les fleurs ? » ;

Si oui : appeler la procédure de l’effet 2 ;

Si non : passer à la ligne suivante ;

Charger l’image 3 ;

Demander à l’utilisateur s’il veut « finir le DM ? » ;

Si oui : appeler la procédure de l’effet 1 ;

Si non : passer à la ligne suivante ;

**Fin du programme principal**

**Début du programme (procédure effet 1)**

Pour La luminositéAjoutée allant de 0 à 25

Attendre 50 ms

Pour le nombre entier Hauteur allant de bas en haut

Pour le nombre entier Largeur allant de gauche à droite

tableauPixel = valeur du pixel aux positions (Largeur, Hauteur)

Pixel aux positions (Largeur, Hauteur) = tableauPixel + luminosité

**Fin de la première procédure**

**Début du programme (procédure effet 2)**

numéroPixel = 0

TableauRecoloration est un tableau qui contient pour chaque pixel sa valeur pour le rouge, le vert et le bleu.

Pour le nombre entier Largeur allant de Gauche à Droite

Attendre 15 ms

Pour le nombre entier Hauteur allant de bas en Haut

Tableau d’entiers tabPix récupère la valeur du pixel aux positions (Largeur, Hauteur)

Tableau recoloration à la position numéroPixel récupère les 3 valeurs de tabPix

numéroPixel = numéroPixel+1

La valeur du pixel à la position (Largeur, Hauteur) = tabPix(bleu : 2), tabPix(vert : 1), tabPix(rouge : 0).

Si Hauteur >= 30 –ObtenirValeurLargeur

Le pixel à la position (Largeur -30, Hauteur) vaut TableauRecoloration à la position (numéroPixel – 30\* obtenirHauteurImage)

**Fin de la deuxième procédure**

**Début du programme (procédure effet 3)**

L’entier numPixel = 0

Le tableau d’entiers tabRecoloration contient les valeurs du rouge, vert et bleu de chaque pixel

Pour Largeur allant de gauche à droite

Pour Hauteur allant de bas vers le haut

tabRecoloration à la place numPixel récupère les valeurs rouge, vert bleu du pixel aux positions (Largeur, Hauteur)

numPixel = numPixel + 1

**Fin des 2 boucles pour**

Pour l’entier numCarré allant de 0 à 50

L’entier LargeurAlea est généré aléatoirement entre (–largeurdel’Image +25 et largeurdel’Image – 25)

L’entier HauteurAlea est généré aléatoirement entre (-hauteurdel’Image + 25 et hauteurdel’Image – 25)

Le tableau tabRandom de 3 entiers voit ses valeurs générées aléatoirement entre 1 et 4

Pour l’entier abscisse allant de -25 à 25

Attendre 2 ms

Pour l’entier ordonne allant de -25 à 25

Le tableau d’entiers tabPix récupère la valeur du pixel aux positions (abscisse + LargeurAlea, ordonne + HauteurAlea)

Le pixel aux positions (abscisse + LargeurAlea, ordonne + HauteurAlea) voit ses couleurs changées en : ( tabPix(rouge)/tabRandom(0), tabPix(vert)/tabRandom(1), tabPix(bleu)/tabRandom(2) )

**Fin des 3 boucles pour**

numPixel = 0

Pour largeur allant de gauche à droite

Attendre 10 ms

Pour hauteur allant de bas vers le haut

Le pixel aux positions (largeur, hauteur) voit ses valeurs changées en : tabRecoloration( (numPixel)(0 : rouge), (numPixel)(1 : vert), (numPixel)(2 : bleu) )

numPixel = numPixel + 1

**Fin des 2 boucles pour et fin de la troisième procédure**

**Sources :**

<http://jsbeautifier.org/> Site internet qui nous a permis de réorganiser notre code.

Les TD d’ISN en lien avec les effets : (tableaux, effets sur image, génération de carrés..)

Tout au long de ce mini projet, nous avons essayé de faire preuve de clarté ainsi que de rigueur notamment dans l’élaboration et la conception de notre programme. Nous avons également fait en sorte de réutiliser exclusivement de nombreux éléments vus en cours. Nous avons cherché à expliquer de manière simple le code à l’aide des commentaires. L’utilisation d’algorithmes pour chaque partie du programme a était extrêmement pratique en particulier pour celles étant plus compliquées. Par exemple, lors de la réalisation d’un tableau permettant de récupérer les données des pixels de l’image d’origine, nous sommes revenus à une réflexion sur papier nous permettant de prendre un certain recul sur ce que nous voulions faire.  Nous avons essayé de choisir des images relativement appropriées aux effets réalisés. En espérant que cela vous plaira. BONNE FÊTES !!