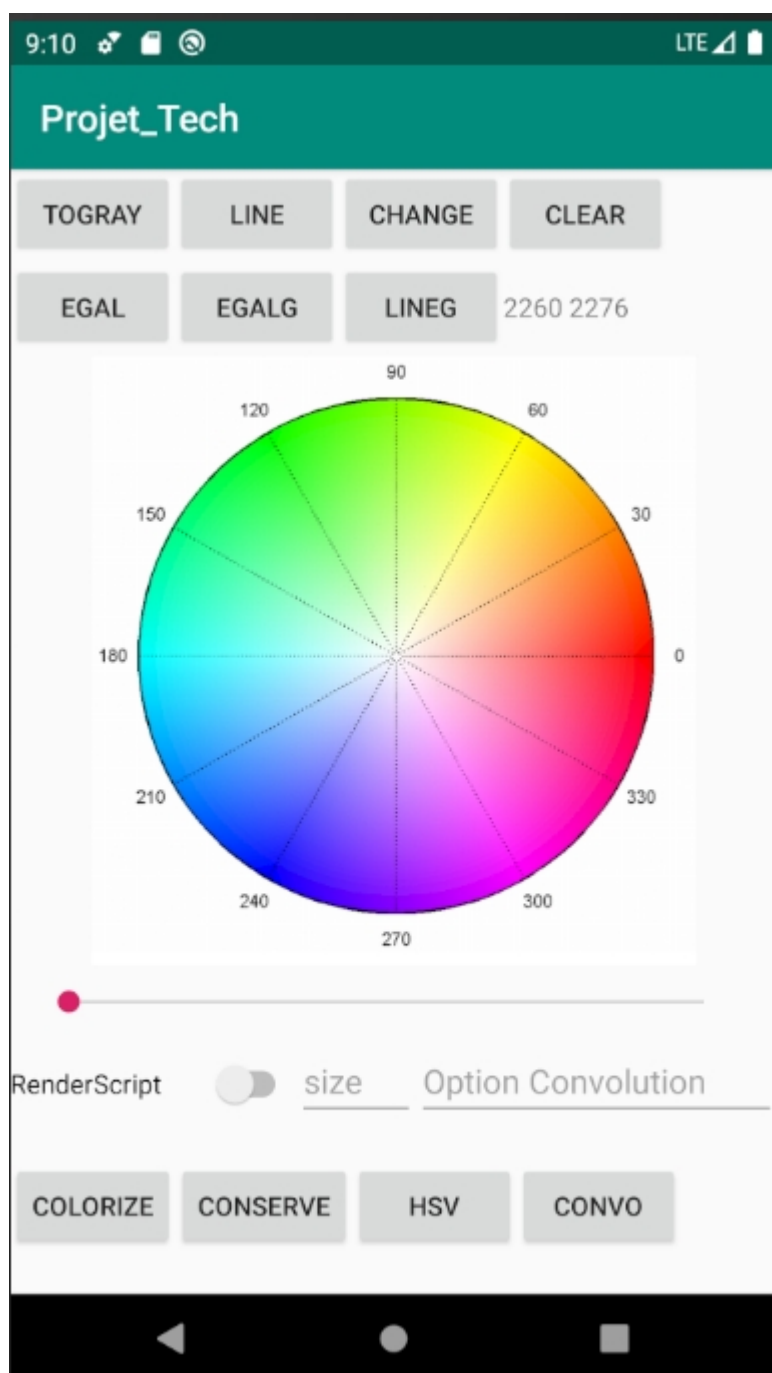


# Rapport de projet

I) Mode d'emploi de l'application.



## Fonctionnalités des boutons :

- TOGRAY : Met l'image en nuance de gris.
- LINE : Effectue un contraste linéaire. (résultat non fonctionnel)
- CHANGE : Change l'image centrale.
- CLEAR : Enlève les changements effectués sur l'image.
- EGAL : Effectue un contraste égalisé.
- EGALG : Effectue un contraste égalisé en nuance de gris.
- LINEG : Effectue un contraste linéaire en nuance de gris. (résultat non fonctionnel)
- COLORIZE : Change la teinte de l'image, la nouvelle teinte n'est pas choisissable.
- CONSERVE : Garde une seule teinte sur l'image, le reste est en nuance de gris. Couleur choisissable via la seekbar juste en-dessous de l'image (curseur rouge). Il faut presser de nouveau le bouton pour actualiser la couleur choisie.
- HSV : Effectue une double conversion rgb/hsv et hsv/rgb.
- CONVOLUTION : Lance une convolution en fonction des deux champs textuels « size » et « Option Convolution ».  
Dans le champ « Option Convolution » :
  - « contour1 » , « contour2 » , « contour3 » pour un masque qui met en valeur les contours de l'image.
  - « gauss1 » , « gauss5 » pour un filtre gaussien de taille 3\*3 ou 5\*5.
  - « identite » pour un masque représentant la matrice identité.
  - « net » pour un masque qui est censé améliorer la netteté.
  - Si vide ou erreur de saisie : utilise un filtre moyenneur, le champ « size » est ici nécessaire pour choisir une taille de ce filtre

seulement, si vide prend la valeur 1 par défaut, si un nombre pair est entré, le diminue de 1.

## Fonctionnalités des autres éléments visibles :

- Zone de texte à droite de LINEG et en-dessous de CLEAR : dimension de l'image affichée.
- Seekbar en-dessous de l'image : voir CONSERVE.
- RenderScript : switch qui permet de lancer les versions renderscript des fonctions. (seulement fonctionnel avec TOGRAY)
- « size » : voir CONVO.
- « Option Convolution » : voir Convo.

## II) Algorithmes.

### Algorithmes demandés

toGray et toGray2 : Deux fonctions de transformations de l'image en nuance de gris, leur différence et l'utilisation d'un buffer dans la version 2 (voir le comparatif plus bas). Il existe deux versions renderscript (voir le comparatif plus bas).

colorize : Change la teinte d'une image en une autre choisie aléatoirement. Utilisation de la fonction hsvTabs(int[], int, int) (voir la partie Algorithmes auxiliaires).

consereV1 et conserV2 : Conserve une teinte de l'image et laisse le reste en nuance de gris. La différence entre les deux versions et l'utilisation de la fonction hsvTabs(int[], int, int), qui donne un énorme gain de temps pour une augmentation de mémoire utilisée, les différences sont détailler dans le comparatif. Un commentaire sur ces deux fonction indiquent les partie différentes dans le code.

convolutionMatrice et convoAux : Fonctions marchant de paire, la première effectue le parcours de l'image et la seconde applique la matrice à une position donnée. La matrice est une classe représentant les matrices qui servent de masque, ainsi que quelques informations (voir la partie Algorithmes auxiliaires).

contrasteGrayDyn et contrasteDyn : Fonctions qui calculent et appliquent un contraste linéaire (en nuance de gris pour la première). Les résultats de ces fonctions, surtout celle en couleur laisse penser qu'elles ne fonctionnent pas comme elles le devraient.

contrasteEgalGray et contrasteEgal : fonctions qui calculent et appliquent un contraste égalisé (en nuance de gris pour la première). Elles pourraient être optimisée par un ajout d'une fonction similaire à hsvTabs(int[], int, int) qui devrai aussi calculer un maximum et un minimum.

## Algorithmes auxiliaires

Matrice et MatriceTypes : Deux classes servant à représenter des matrices qui sont utilisées en temps que masque pour la convolution. MatriceTypes sert simplement à différencier les matrices par un nom. Matrice est une classe qui a une tableau 2D représentant la matrice, une taille de matrice et un coefficient si le résultat d'un calcul avec cette matrice nécessite une multiplication. Seule la matrice « MOYENNE » est générée automatiquement.

inputTreatment : Permet d'associer un String à un MatriceType.

hsvTabs : Créer un tableau 2D représentant les valeurs HSV d'un tableau de pixels (représentant l'image), gain de temps conséquent par rapport à une conversion pendant un traitement (voir le comparatif). Amélioration possible : une version qui part tableau2D vers un tableau de pixels pourrait aussi augmenter les performances.

histoHsv : créer un histogramme à partir de valeurs de type float.

rgb\_to\_hsv et hsv\_to\_rgb : Fonctions de conversions de rgb vers hsv et hsv vers rgb, cette conversion se fait pour une valeur de rgb ou hsv à la fois.

doubleConversion : Fait une conversion rgb vers hsv puis hsv vers rgb, elle sert principalement de test.

ImageBank : Objet contenant les id des images, permet ainsi de stocker et accéder plus rapidement les images s'y trouvant.

### III) Performances et comparatifs

Les performances sont calculées sur un échantillon de deux séries de 4 appels à fonctions sur l'image la plus légère puis la plus lourde contenue dans l'application. Certaines valeurs de mémoires sont altérées par le passage du garbage collector pendant les calculs.

Tous les tests ont été effectués sur l'API 27 et sur le modèle Pixel XL.

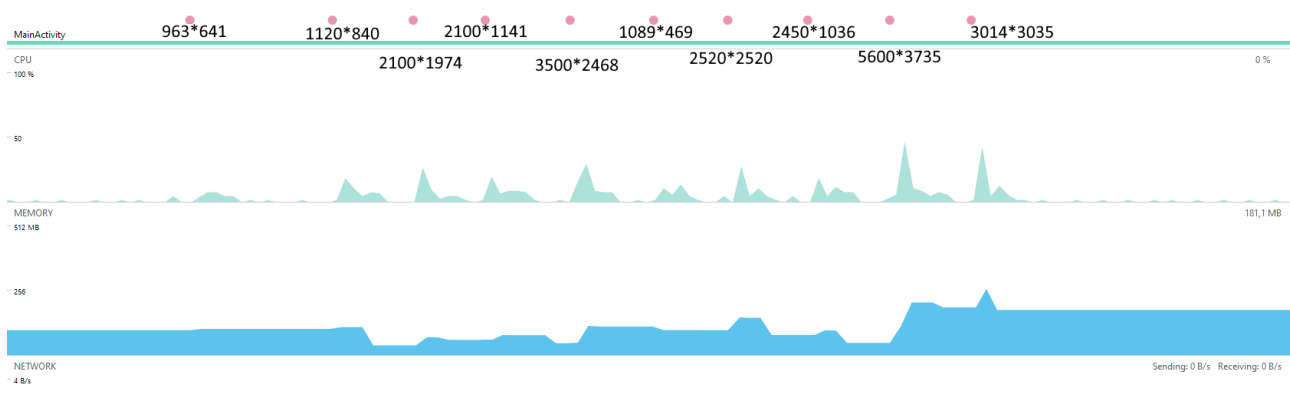


L'image la plus petite : 963\*641  
(affiché en taille réelle)



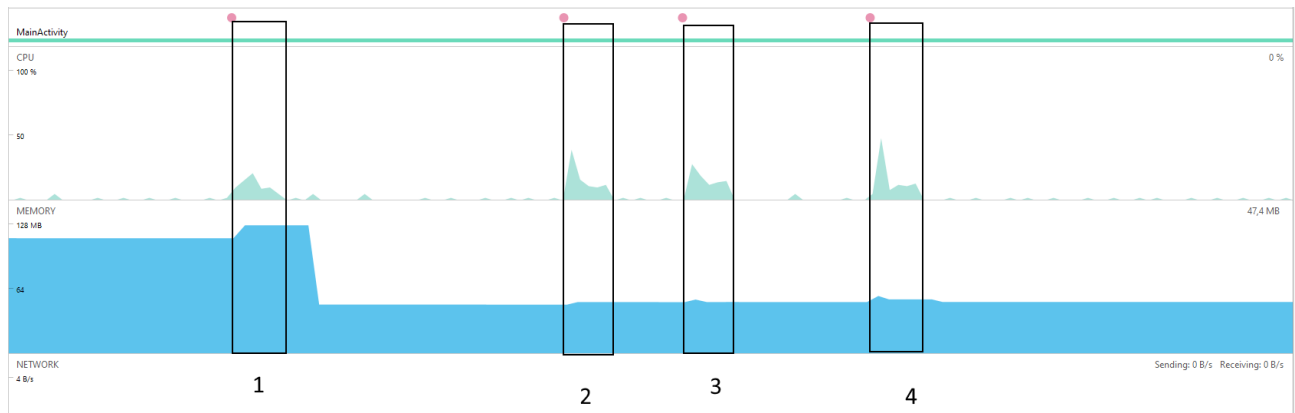
L'image la plus lourde : 5600\*3735  
(affiché en taille réduite)

- CHANGE :

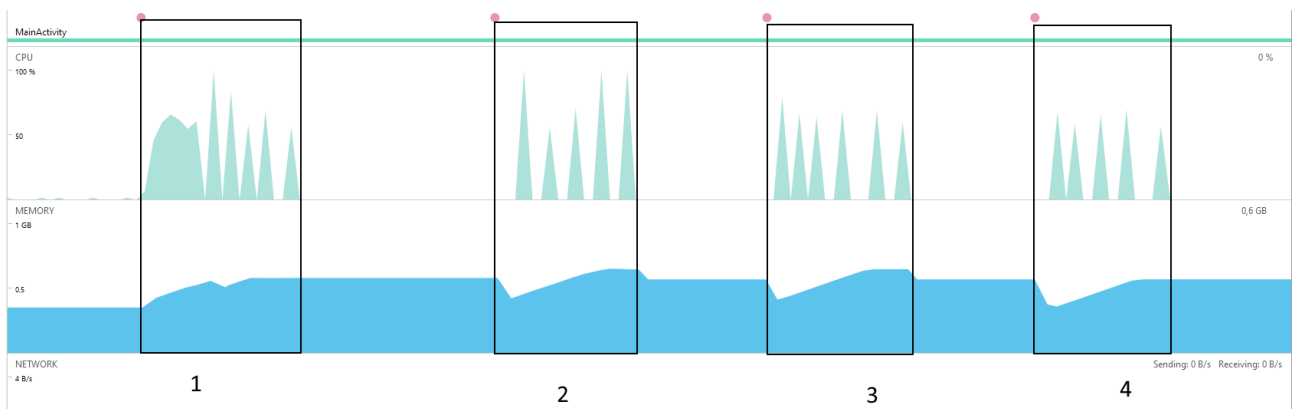


Ceci et l'appel au bouton CHANGE, les tailles des images nouvellement affichées sont indiquées en haut, on distingue bien les différences d'utilisations de mémoires et du processeur selon la taille des images.

## - COLORIZE:

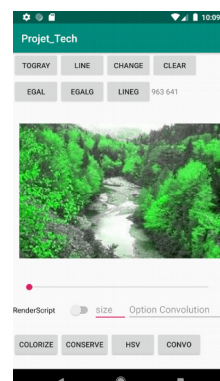
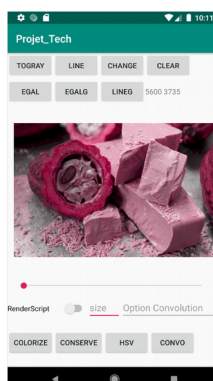
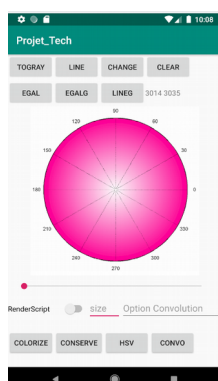


Appels de COLORIZE sur l'image la plus petite.  
Utilisations maximal moyen de la mémoires : 2,3MB.  
Temps moyen : 1,2 sec.



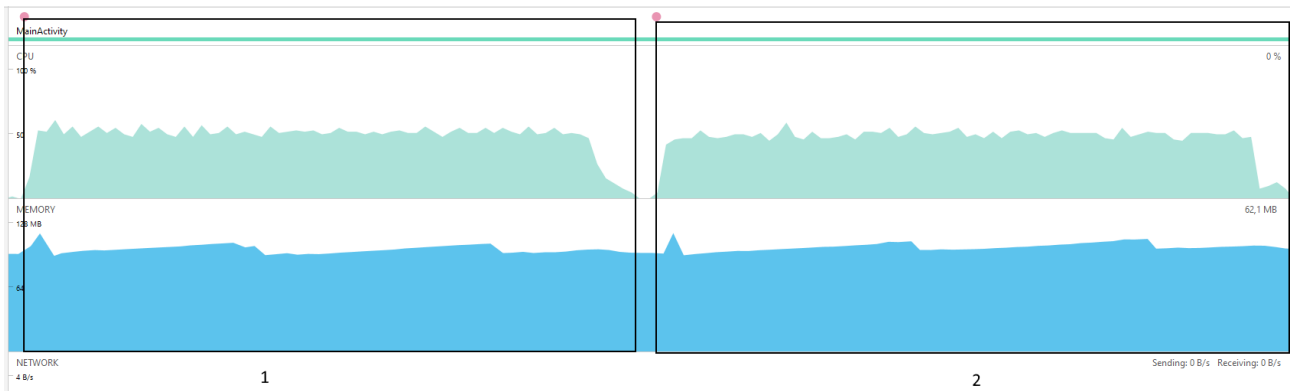
Appels de COLORIZE sur l'image la plus lourde.  
Utilisations maximal moyen de la mémoires : 0,2GB.  
Temps moyen : 3,5 sec.

Exemples :



## - CONVO :

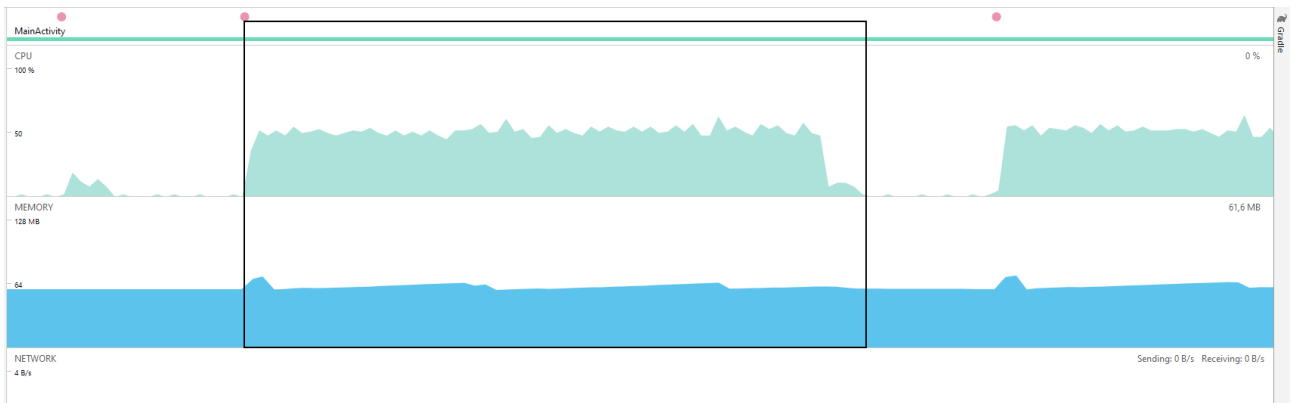
Pour CONVO les test sont effectués sur l'image la plus petite uniquement.



Masque de taille 3\*3.

Utilisations maximal moyen de la mémoires : 11,5MB.

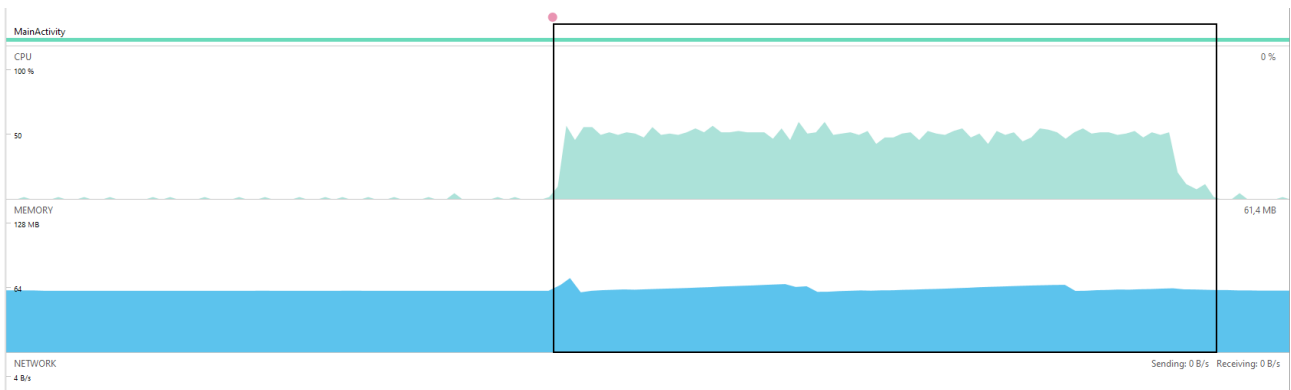
Temps moyen : 15,1sec.



Masque de taille 5\*5.

Utilisations maximal moyen de la mémoires : 11,1MB.

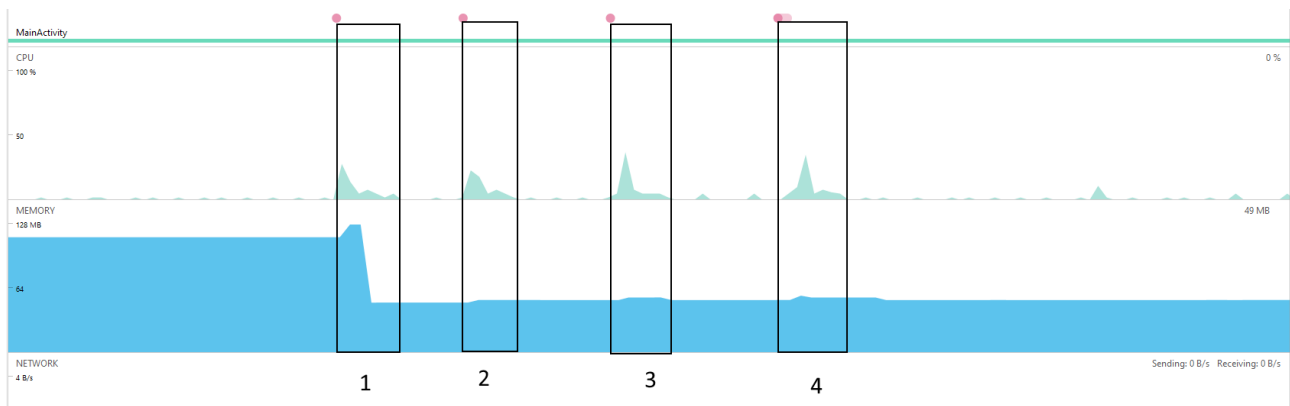
Temps moyen : 14,7sec.



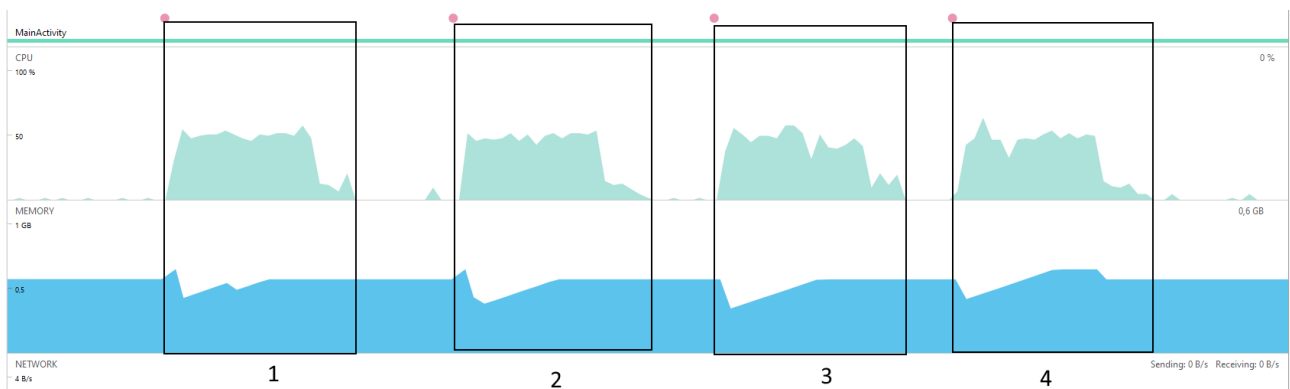
Masque de taille 9\*9.

Utilisations maximal moyen de la mémoires : 15,5MB.  
Temps moyen : 12,3sec.

- EGAL :

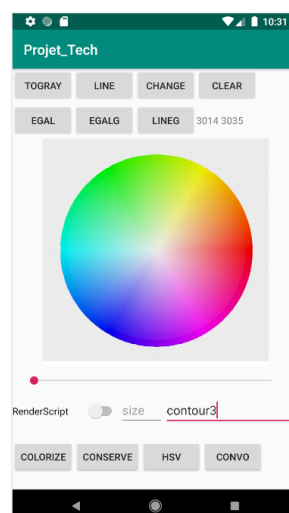


Appels de EGAL sur l'image la plus légère.  
Utilisations maximal moyen de la mémoires : 2,2MB.  
Temps moyen : 1,4 sec.



Appels de EGAL sur l'image la plus lourde.  
Utilisations maximal moyen de la mémoires : 0,2GB.  
Temps moyen : 5,5 sec.

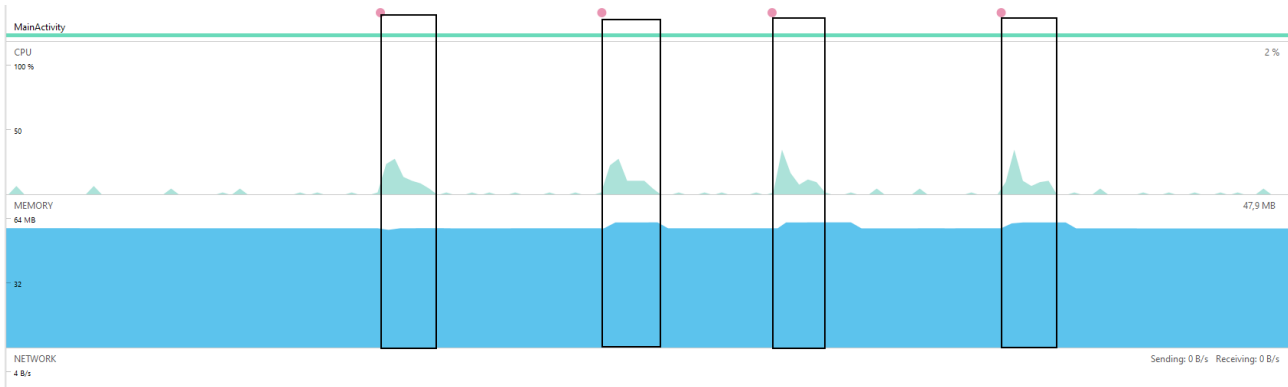
Exemples :



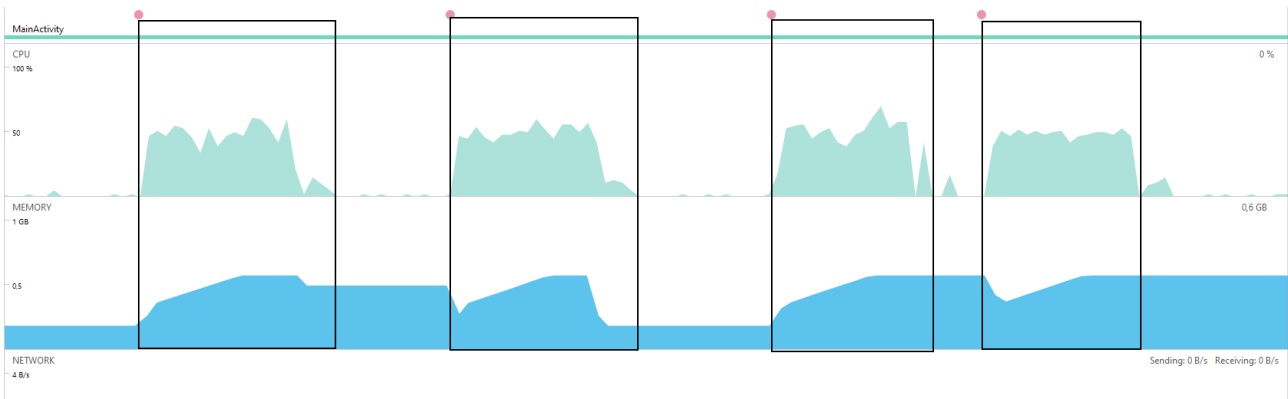


- CONSERVEV1 et CONSERVEV2 :

## CONSERVEV2 :

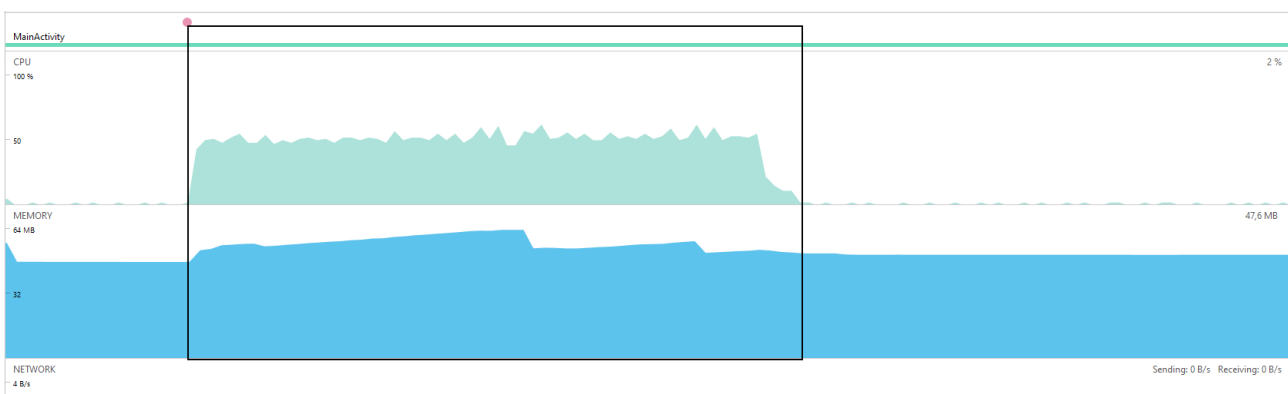


Appels de CONSERVE sur l'image la plus légère.  
Utilisations maximal moyen de la mémoire : 2,3MB.  
Temps moyen : 1,3 sec.



Appels de CONSERVE sur l'image la plus lourde.  
Utilisations maximal moyen de la mémoire : 0,3GB.  
Temps moyen : 4,3 sec.

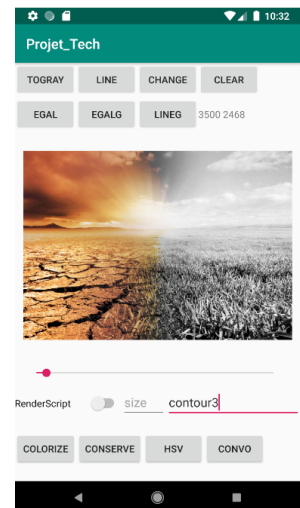
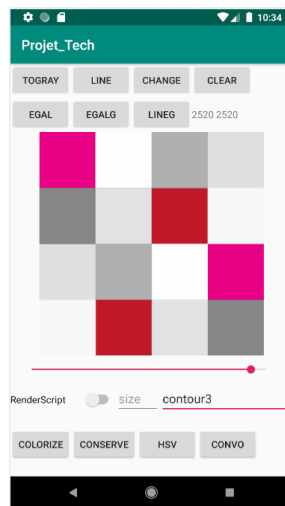
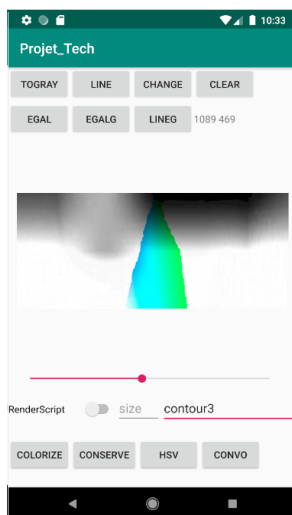
Comparatif avec CONSERVEV1 :



CONSERVEV1 sur l'image la plus légère.  
Utilisations maximal moyen de la mémoires : 14,7MB.  
Temps moyen : 14,3 sec.

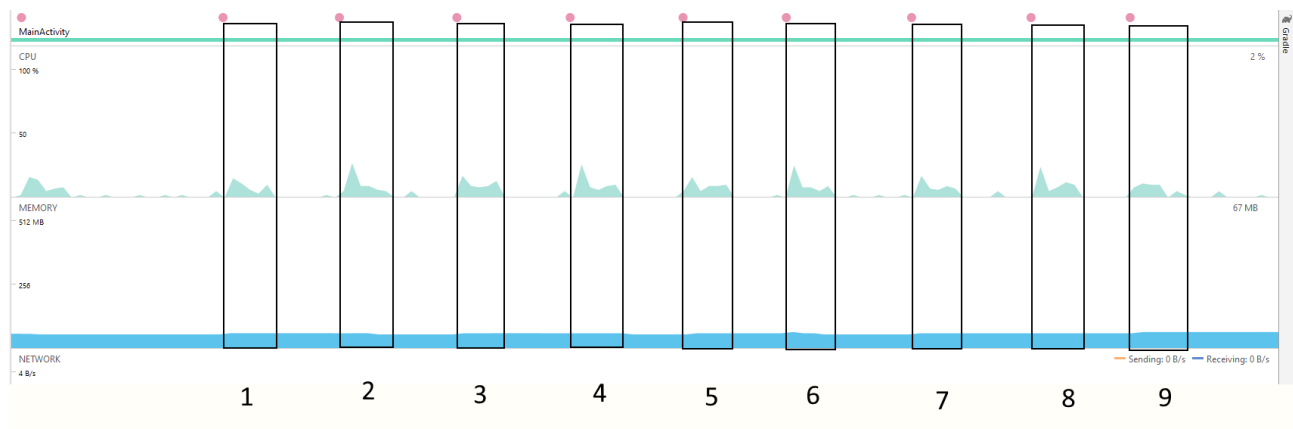
Sur l'image lourde, la fonction n'était pas fini après 3min5.

Exemple :

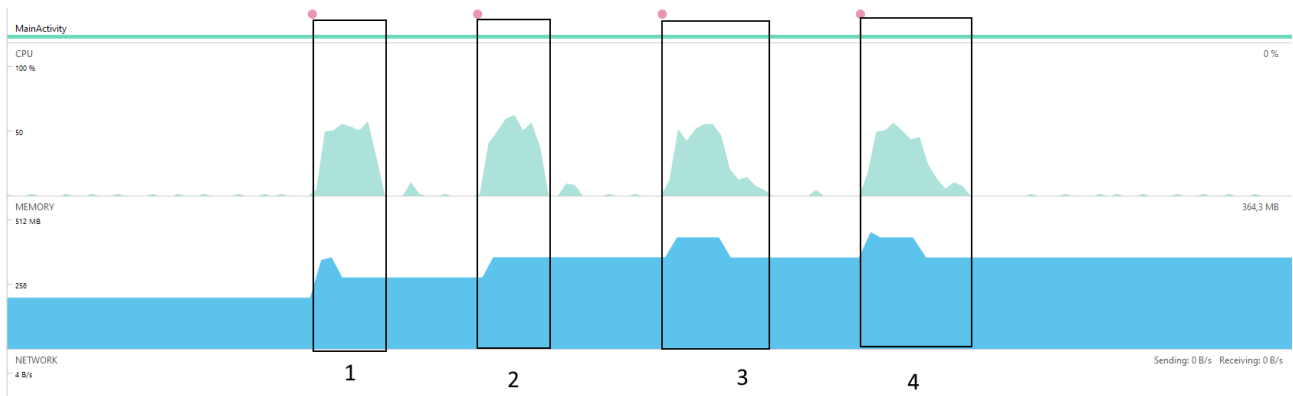


TOGRAY et TOGRAY2:

TOGRAY2



TOGRAY2 sur l'image la plus légère.  
Utilisations maximal moyen de la mémoires : 5,6MB.  
Temps moyen : 1,2 sec.



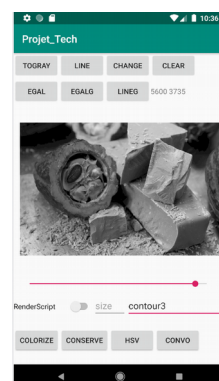
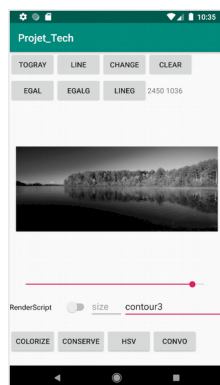
TOGRAY2 sur l'image la plus lourde.  
 Utilisations maximal moyen de la mémoires : 80,3MB.  
 Temps moyen : 2,1 sec.

Comparatif avec TOGRAY:



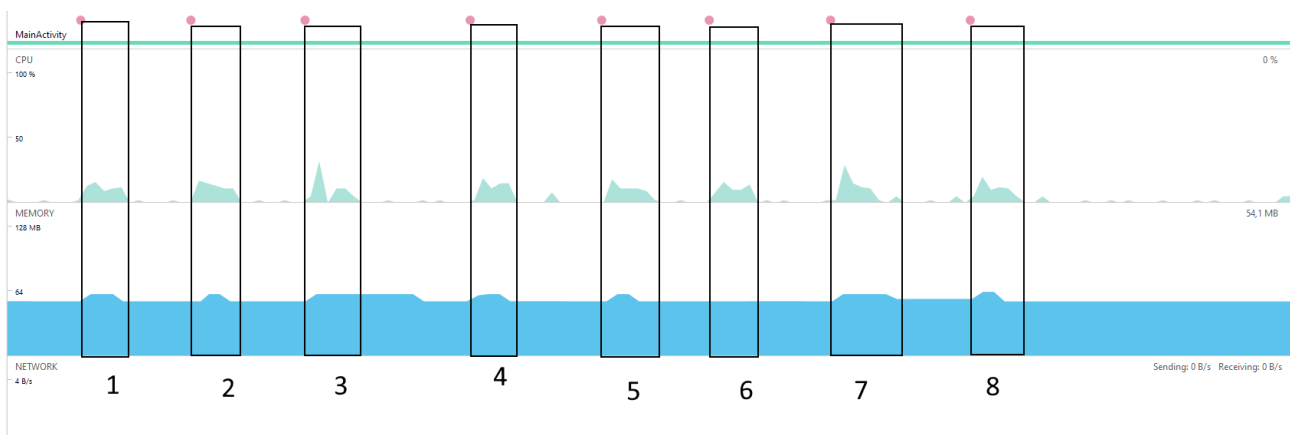
TOGRAY sur l'image la plus lourde.  
 Utilisations maximal moyen de la mémoires : 79,8MB.  
 Temps moyen : 37,1 sec.

Exemple :

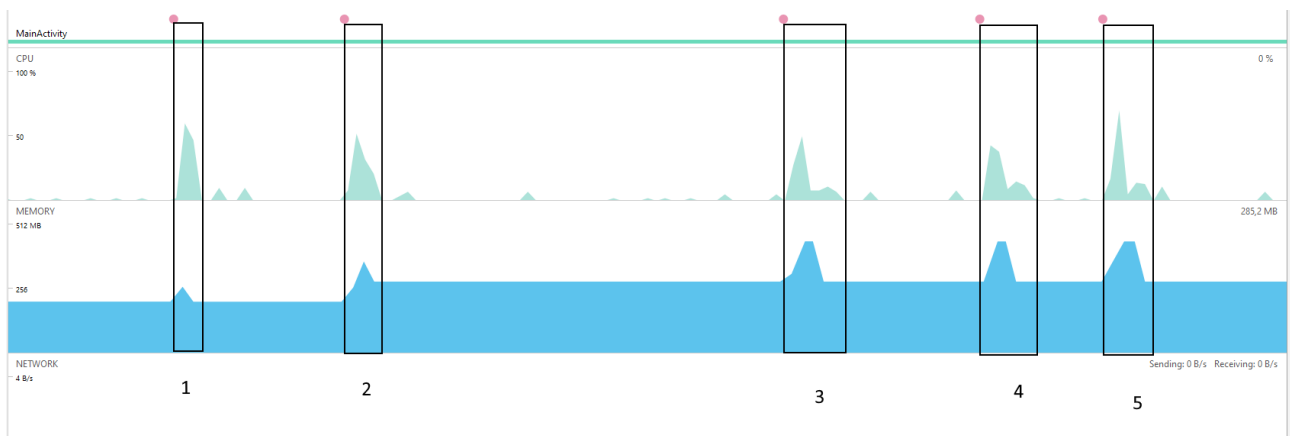


- TOGRAYRS et TOGRAYRS2:

## TOGRAYRS2

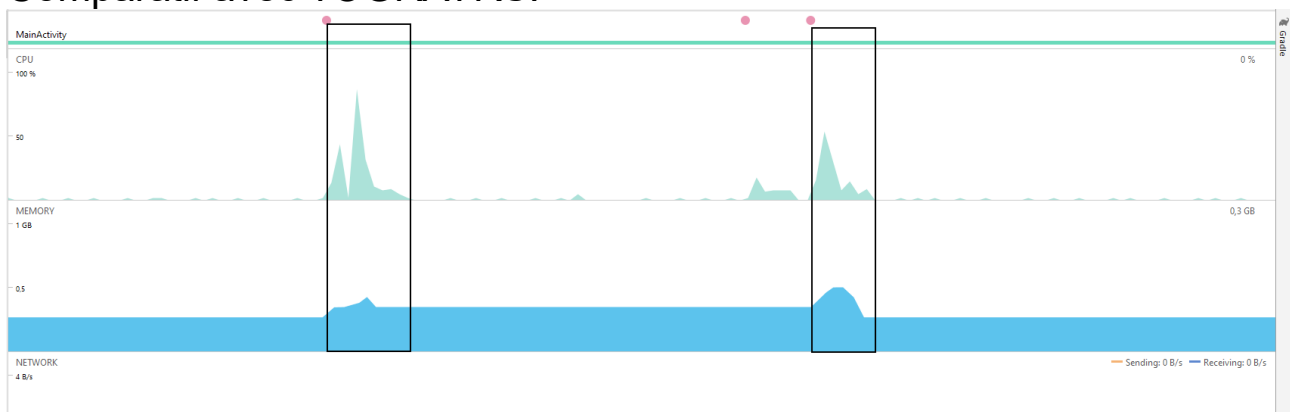


TOGRAYRS2 sur l'image la plus légère.  
Utilisations maximal moyen de la mémoires : 7,2MB.  
Temps moyen : 1,1 sec.



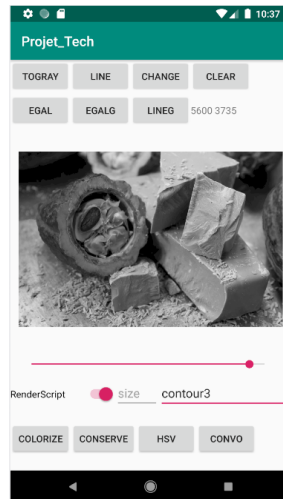
TOGRAYRS2 sur l'image la plus lourd.  
Utilisations maximal moyen de la mémoires : 160MB.  
Temps moyen : 0,9 sec.

## Comparatif avec TOGRAYRS:



TOGRAYRS sur l'image la plus lourde.  
Utilisations maximal moyen de la mémoires : 0,1GB.  
Temps moyen : 2 sec.

Exemple :



## IV) Limites

Les problèmes de ce projet sont le manque de renderscript, le contraste linéaire non fonctionnel et certaines fonctions pourraient être améliorées.