

Projet de fin d'année : Outil de visualisation d'images médicales

Juliann Le Gall

Antoine Thebault

Sommaire

I.	INTRODUCTION GENERALE	3
II.	REALISATION ET DEROULEMENT DU PROJET :	3
1.	1 ^{ERE} ETAPE : INTERFACE GRAPHIQUE	3
2.	2EME ETAPE: OUVERTURE DES FICHIERS DICOM ET NIFTI	4
3.	3EME ETAPE: TRAITEMENT DES FICHIERS.....	4
4.	4EME ETAPE: VISUALISATION DES IMAGES.....	4
5.	5EME ETAPE: LES CHECKBOXS.....	4
6.	6EME ETAPE: LES SPINBOXS.....	5
7.	7EME ETAPE: LES SLIDERS.....	5
8.	8EME ETAPE: CURSEUR SUR LA FIGURE AXIALE	5
9.	TRANSFORMER LE FICHIER PY EN EXECUTABLE	5
III.	PROBLEME NON RESOLU:	6
1.	1ER PROBLEME:	6
2.	2EME PROBLEME :.....	6
IV.	TEST REALISE	6
1.	FICHIER DICOM :.....	6
2.	FENETRE ABOUT	8
3.	FICHIER NIFTI :.....	9
V.	PROJET FUTUR	9
VI.	CONCLUSION	10
VII.	ANNEXE	10
1.	TABLE DES FIGURES	10
2.	FICHIER PYTHON	10

I. INTRODUCTION GENERALE

Les progrès scientifiques dans le monde médical ne cessent de croître et sont accompagnés de technologies de plus en plus performante générant des fichiers de plus en plus complexes. En réponse à cette croissance l'outil informatique se doit de coller aux exigences des différents médecins spécialistes. C'est pour cela que nous nous sommes intéressés à programmer une application permettant d'ouvrir les fichiers générés par les différentes techniques d'imagerie médicales.

L'objectif final de ce projet est de développer un outil graphique de visualisation d'images médicales tridimensionnelles en utilisant les interfaces graphique tel que Qt. Cet outil de visualisation doit permettre d'afficher des images médicales à partir de format NIFTI(.nii) et de formats DICOM (.dcm)

II. REALISATION ET DEROULEMENT DU PROJET :

1. 1^{ère} étape : Interface graphique

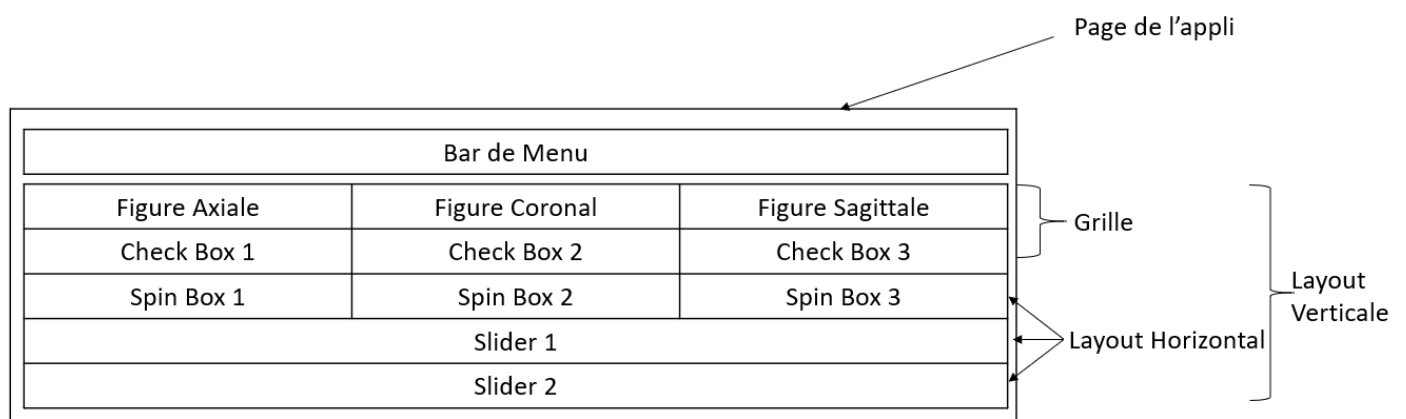


Figure 1: Schéma de l'interface graphique

L'interface graphique a été réalisé au début pour avoir une vue globale de l'application finale. Celle-ci à l'aide de nombreux widgets classé dans des lay-out ou des grilles, qui sont eux même stockés dans un widget principal. Pour faciliter l'utilisation, la page de l'application a été agrandi afin de visualiser convenablement les différentes coupes. Pour cela vous pouvez voir un schéma présenté ci-dessus qui décrit ce que nous avons créé. Dans cette continuité, il a été créé un menu et des sous menus qui permettent à l'utilisateur d'ouvrir un fichier, de quitter l'application ou obtenir de l'aide. Tout ce qui a été réalisé jusque-là n'est que l'aspect graphique.

Les étapes de créations des widgets seront détaillées ultérieurement dans chaque partie qui leurs sont dédiés.

Pour réaliser cette étape, il a fallu importer de nouveaux module dans Python comme Qt5.

2. 2ème étape: Ouverture des fichiers Dicom et Nifti

Cette application doit permettre à l'utilisateur de choisir des fichiers Dicom et Nifti qu'il possède dans son ordinateur. C'est pourquoi dans cette étape, il a fallu à partir de notre barre de menu ouvrir l'explorateur de fichier qui permet à l'utilisateur de choisir ses fichiers. Dès lors que les fichiers pouvaient être ouverts, il était nécessaire de lire ces deux types de fichiers, par conséquent il était nécessaire d'importer deux modules nibabel et pydicom.

De plus, dans notre fonction nous récupérons les données des fichiers dicom et nous vérifions que les noms des patients sont bien identiques, également nous les classons par ordre de coupe. Cette fonctionnalité permet malgré un changement de nom de fichiers et des fichiers en désordre d'afficher sans problème les 3 coupes.

L'utilisateur est obligé de choisir entre un fichier dicom et un fichier nifti, s'il fait un abandon ou la croix, il y a un message qui lui affiche qu'il n'a pas choisi de fichier.

3. 3ème étape: Traitement des fichiers

Tout d'abord, il faut savoir que les fichiers dicom sont des coupes d'une image 3D, ces fichiers sont donc en 2D, c'est pourquoi il était nécessaire de créer une fonction qui permettait de récupérer toutes les coupes de chaque fichier pour obtenir une image en 3D. Ces données sont stockées dans une liste de listes de listes. Pour faciliter son traitement avec python il faut utiliser le module numpy c'est pourquoi il était également nécessaire de l'importer. De plus ce module permettait d'utiliser plus facilement les fichiers nifti, qui sont des images 4D avec pour 4e paramètre le temps.

4. 4ème étape: Visualisation des images

Pour visualiser les 3 coupes c'est à dire axiale, coronale, et sagittale nous avons dû importer le module matplotlib. Pour faciliter leur manipulation dans le code, nous avons séparés ses trois coupes en 3 fonctions, ce qui nous permettra dans le futur de les manipuler plus simplement, et de les relier aux différentes fonctions que cette application permet de faire. Pour afficher la figure, nous avons commencé par ouvrir une figure plt, puis nous avons sélectionné les valeurs souhaitées dans le tableau numpy. Par la suite nous avons fait un imshow pour afficher l'image puis un Figure Canvas accompagné d'un draw pour que cette figure soit visualisée sur l'application. Pour que l'application soit située en haut à gauche ou au milieu ou même à droite nous utilisons notre grille que nous avons précédemment défini. De plus en fin de projet, nous avons décidé d'ajouter un titre aux figures pour que l'utilisation soit plus intuitive.

5. 5ème étape: Les checkBoxs

Pour que les checkBoxs soient parfaitement alignés aux photos, nous avons décidé de les intégrer à notre grille, voir le schéma ci-dessus. Une checkBox permet quand elle est cochée d'afficher la figure en couleur et en noir et blanc sinon. Pour obtenir une grande variété de couleur nous utilisons la map de couleur jet ce qui permet à l'utilisateur de percevoir plus facilement les nuances. Le comportement de ces checkBoxs est effectué dans la partie constructeur, celles-ci, dès qu'elles sont cochées ou décochées elles lancent la fonction qui lui est associée. Donc pour obtenir ces deux types de couleurs nous étions obligés d'ajouter une condition dans les fonctions, qui permettent de tracer les courbes, pour savoir quelle map de couleur utilisé.

6. 6ème étape: Les SpinBoxs

Pour afficher les SpinBoxs au milieu de la page, nous avons d'abord dû commencé par créer un layout horizontal où nous plaçons les 3 SpinBox alignés, une fois cette étape réalisée nous avons ajouté ce layout horizontal au layout principal qui lui est vertical. Leur comportement est également effectué dans la partie constructeur, et ces derniers dès qu'ils sont modifiés ils font appel à une fonction qui leur est associé. Comme pour les checkBoxs nous avons dû rajouter dans la fonction un élément qui permet de prendre la valeur des SpinBoxs et d'afficher la coupe d'image qui lui correspond. Dans cette partie nous avons rencontré un problème quand la SpinBox pouvait aller au-delà du nombre de coupe que le fichier possède, pour contrer ce problème nous avons ajouté un maximum au SpinBox, ces valeurs maximales représentent la taille du tableau numpy.

7. 7ème étape: Les Sliders

Les Sliders sont stockés dans un layout vertical qui lui-même sera stocké dans le layout principal. Le Slider supérieur permet à l'utilisateur de choisir l'intensité minimale et le slider inférieur celui de l'intensité maximale; Les valeurs minimales et maximales de ces sliders sont mis à jour automatiquement, car par exemple la valeur du slider supérieur définit la valeur minimum du slider inférieur. Pour voir les différentes intensités nous avons ajouté une couleur barre aux 3 figures, maintenant il fallait connecter ses sliders à cette couleur barre. Nous avons décidé de connecter les fonctions dès que l'utilisateur relâche le slider car le programme ne buggait pas, puisque si nous mettions dès que la valeur du slider changeait alors les trois fonctions se lançaient en même temps et se répétaient tant que l'utilisateur ne le relâchent pas. Maintenant que la connection est établi nous rajoutons un plt.clim avec les valeurs du slider1 et du slider 2 dans les fonctions qui sont appelés dès que le slider est relâché.

8. 8ème étape: Curseur sur la figure axiale

Cette fonction permet à l'utilisateur de cliquer sur la fonction axiale à un point précis et de montrer les coupes coronale et sagittale qui correspond. Pour cela nous avons importé un nouveau module qui permet de récupérer les clics c'est à dire connaître leur position en x et en y. Une fois réalisé nous récupérons ces deux données et nous les mettons dans les 2 SpinBoxs associées, cela permet de mettre à jour directement la figure coronale et sagittale. De plus, si l'utilisateur ne souhaite plus voir le curseur il suffit à l'utilisateur de cliquer en dehors de la figure mais sur la partie axiale. Cependant, les valeurs des SpinBoxs 2 et 3 seront mis à la valeur du constructeur.

9. Transformer le fichier py en exécutable

Pour mettre le fichier py en exécutable nous avons dû installer auto py, pour cela nous avons ouvert l'invite de commande et taper pip install auto-py-to-exe. Lorsque l'installation est terminé il faut noté auto-py-to-exe pour lancer l'application puis il suffit de convertir le fichier.

III. PROBLEME NON RESOLU:

1. 1er problème:

L'utilisateur ne peut pas choisir une tranche de fichier, il est obligé de prendre tous les fichiers. Piste de résolution du problème : Il faudrait modifier la fonction coupe pour faire une liste qu'avec les numéros de coupes et de la taille de fichiers que l'on souhaite, puisque le programme a été codé de manière à avoir la première coupe avec l'instance ID = 1 et toutes les coupes les unes après les autres. De plus, il faudrait modifier les valeurs des spins box de manière à avoir la valeur minimale et maximale des coupes que l'on a sélectionné.

2. 2ème problème :

Si l'utilisateur est sur un dossier comportant des fichiers DICOM, il ne peut pas ouvrir un dossier comportant des fichiers NIfTI sans fermer et ré-ouvrir l'application. Cependant, l'inverse est possible.

IV. TEST REALISE

1. Fichier DICOM :

Lorsque la totalité des fichiers DICOM du dossier de test sont sélectionnés via l'open file situé dans file. L'affichage standard est celui représenté sur la Figure 2 ci-dessous. Les différentes coupes sont en nuances de gris et l'afficheur de coupe est paramétré de base à la 10e coupe pour la coupe axiale et à la 100e coupe pour les coupes coronale et sagittale. Une indicateur d'intensité est représenté à droite de chaque coupe par une barre de couleur.

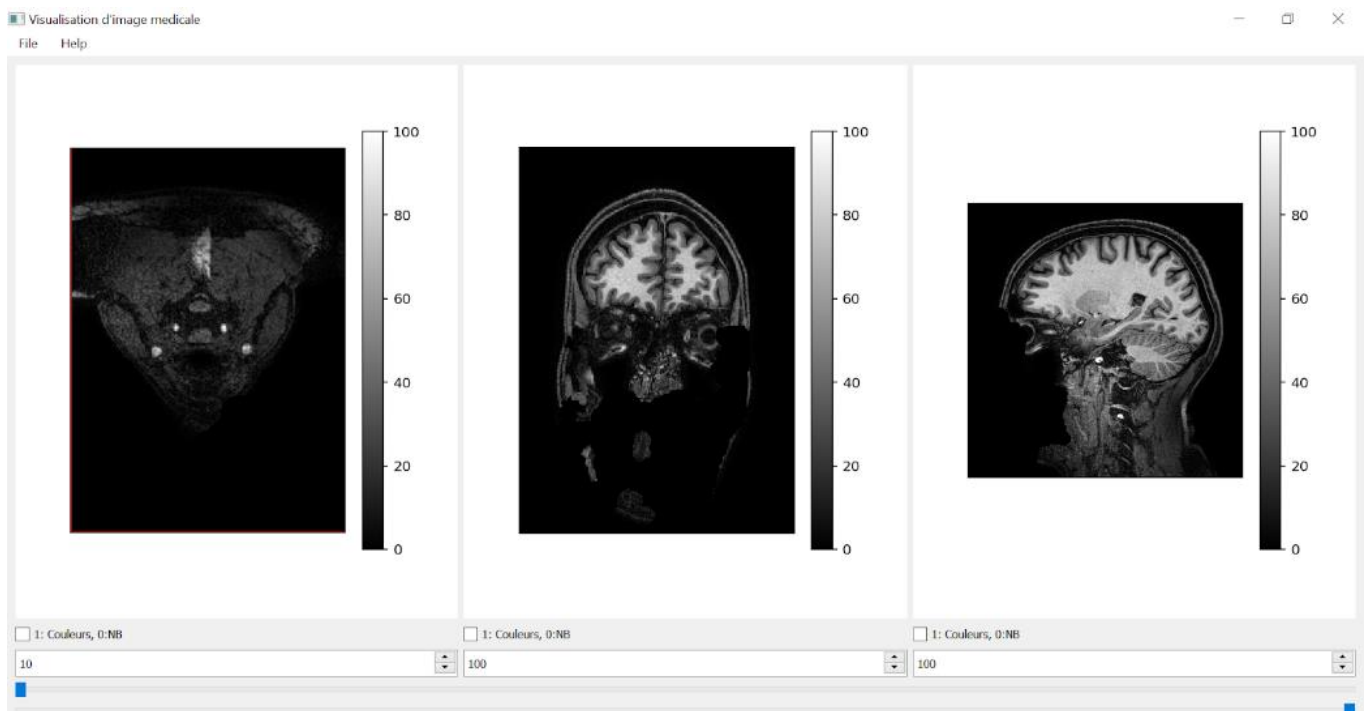


Figure 2: Ouverture d'un fichier Dicom

Lorsqu'on se sert de l'afficheur avec les flèches celui-ci va aller à la coupe d'avant ou d'après suivant les commandes effectués. Si l'utilisateur reste appuyer sur une des flèches, les coupes vont être balayé rapidement jusqu'à ce que l'utilisateur arrête d'exercer une pression sur son click. Le curseur s'arrête lorsqu'on arrête de cliquer sur une des flèche et affiche la coupe que l'afficheur demande. On peut également entrer la coupe désirée manuellement sans utiliser les flèches. Cela permet de sélectionner plus rapidement la coupe que l'on souhaite visualiser. La figure 3 ci-dessous illustre l'affichage quand l'utilisateur a demandé la coupe 200 pour la coupe axiale. Cette dernière manipulation est également réalisable pour les 2 autres coupes. L'utilisateur peut également choisir s'il veut une coupe en couleur. Pour ce faire il lui suffit de cocher la case juste en dessous de l'image se nommant : 1: Couleurs, 0 : NB. L'intensité minimale et maximale des différentes coupes peuvent également être réglées. Pour ce faire, il suffit de faire glisser le slider. L'exécution de ce dernier agit pour les 3 coupes. Les valeurs de la barre de couleur changent également selon le numéro de la coupe sélectionné. Ces différentes modifications par rapport à l'affichage standard sont illustré sur la figure 3 ci-dessous.

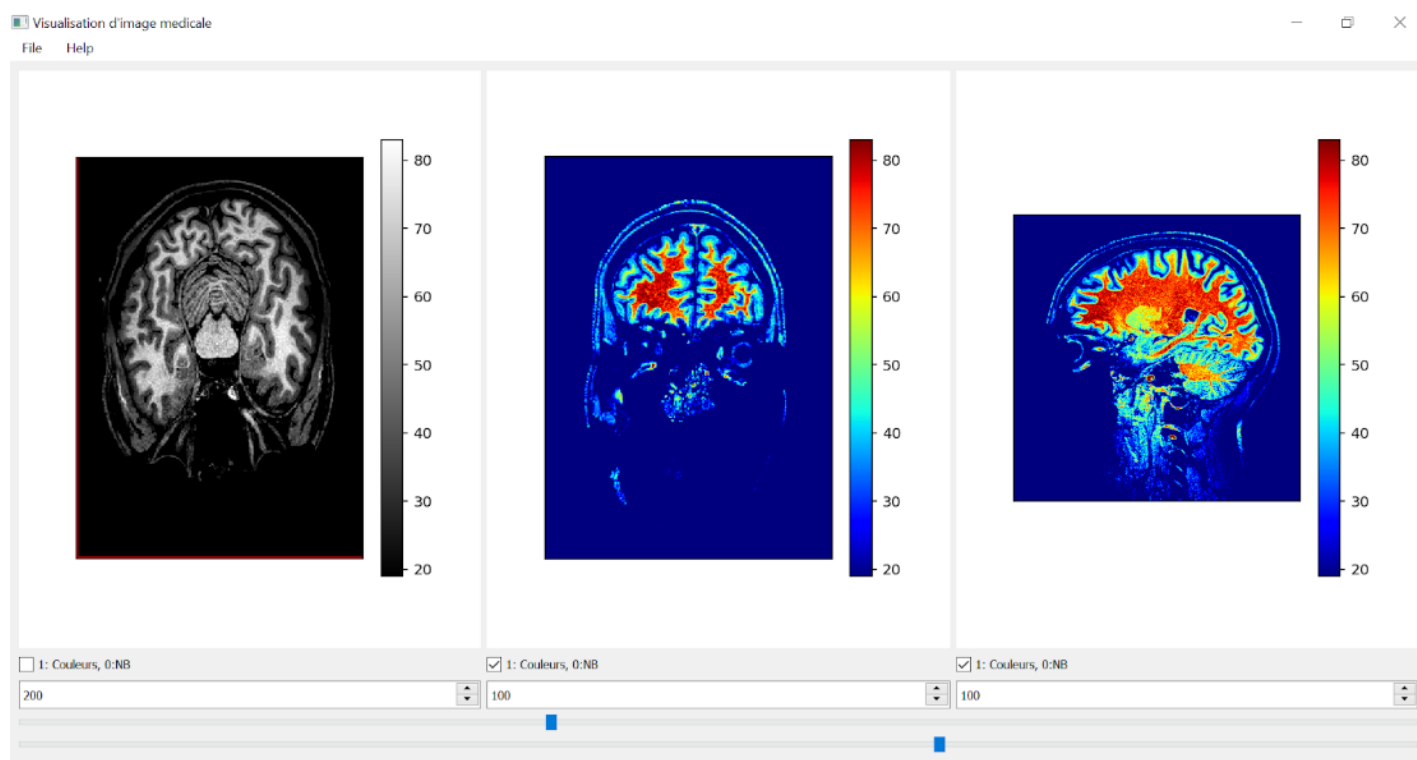


Figure 3: Utilisation des widgets sur un fichier Dicom

Un curseur indiquant la position sur l'encéphale est également disponible pour l'utilisateur. En effet, en cliquant sur une position sur la coupe axiale, les 2 autres coupes sont modifiés et représente l'endroit concerné mais suivant les 2 autres orientations. L'affichage de cette fonctionnalité est représenté sur la figure 4 ci-dessous. Si l'utilisateur veut réinitialiser son curseur il lui suffit de cliquer en dehors de l'image.

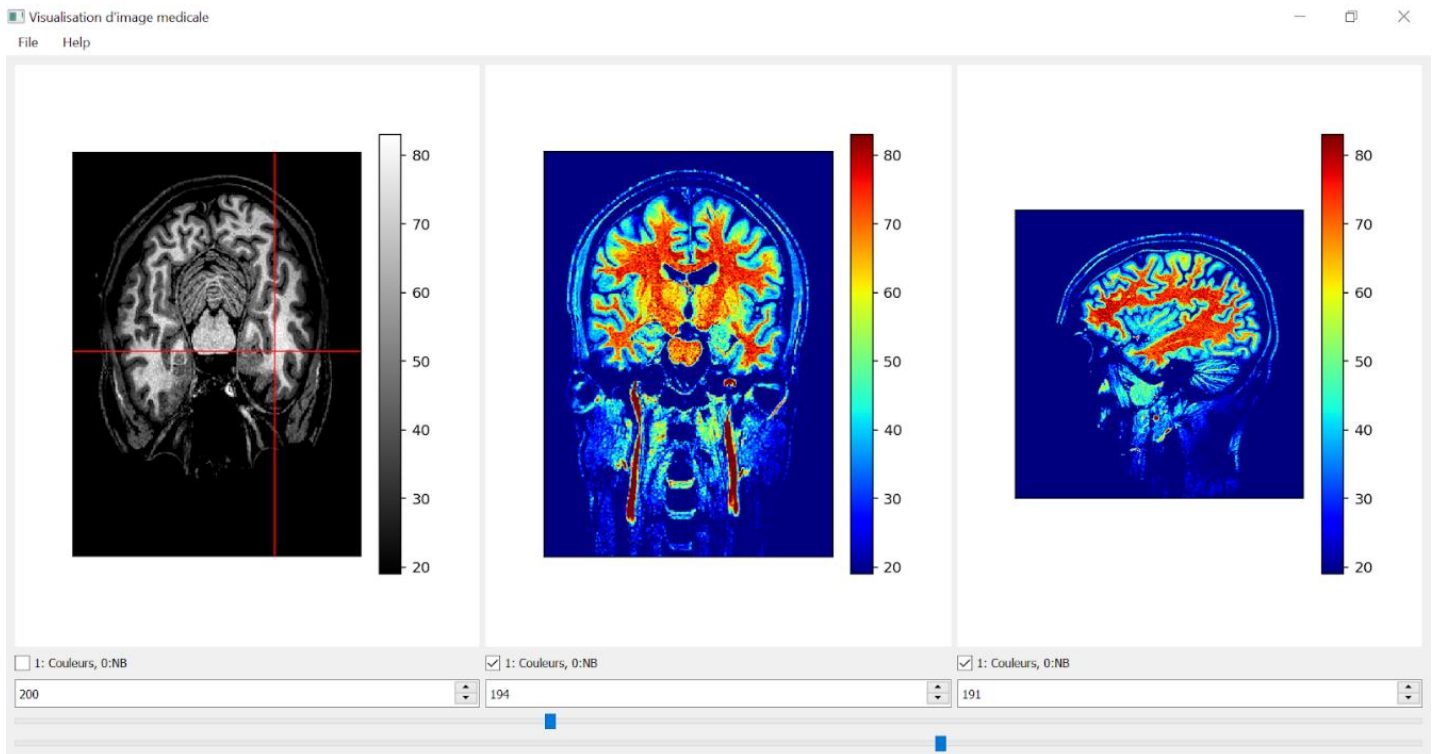


Figure 4: Utilisation du curseur sur un fichier Dicom

2. Fenêtre About

Une fenêtre comportant des informations sur l'application et son utilisation sont disponibles pour l'utilisateur. Pour afficher celle-ci il suffit de cliquer sur Help puis About. Celui-ci propose une description succincte de l'application. La fenêtre affichée est représentée sur la figure 5 ci-dessous.

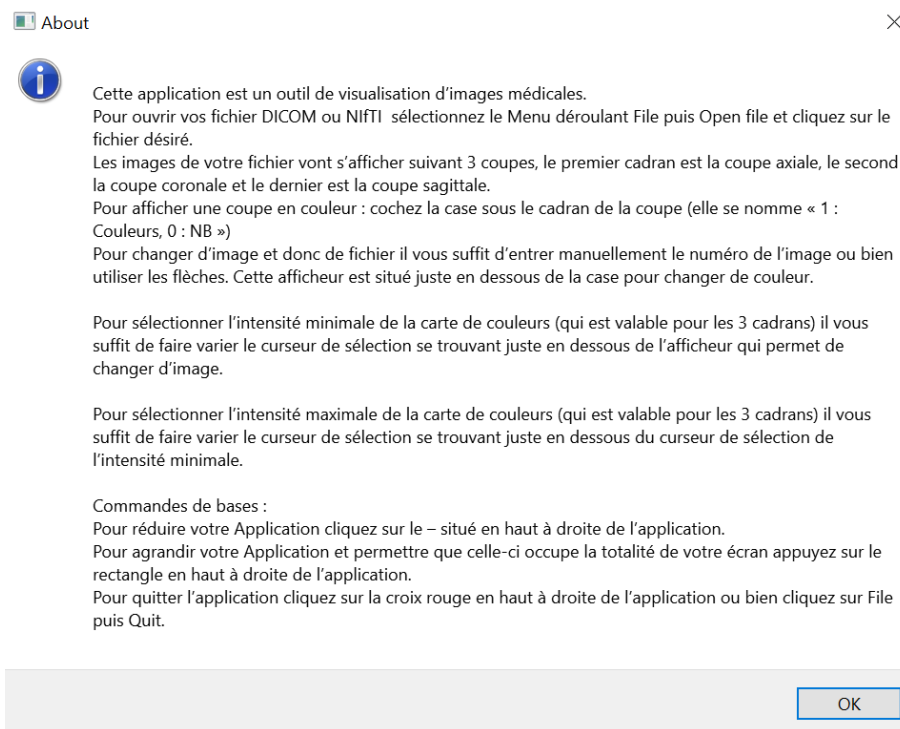


Figure 5: Fenêtre About

3. Fichier NiftI :

L'application fonctionne également pour des fichiers NiftI et délivre les mêmes fonctionnalités que pour les fichiers DICOM. La figure 6 ci-dessous illustre ceci. La seule différence entre ces deux types de fichiers est qu'une image NiftI a une dimension de plus qui est le temps.

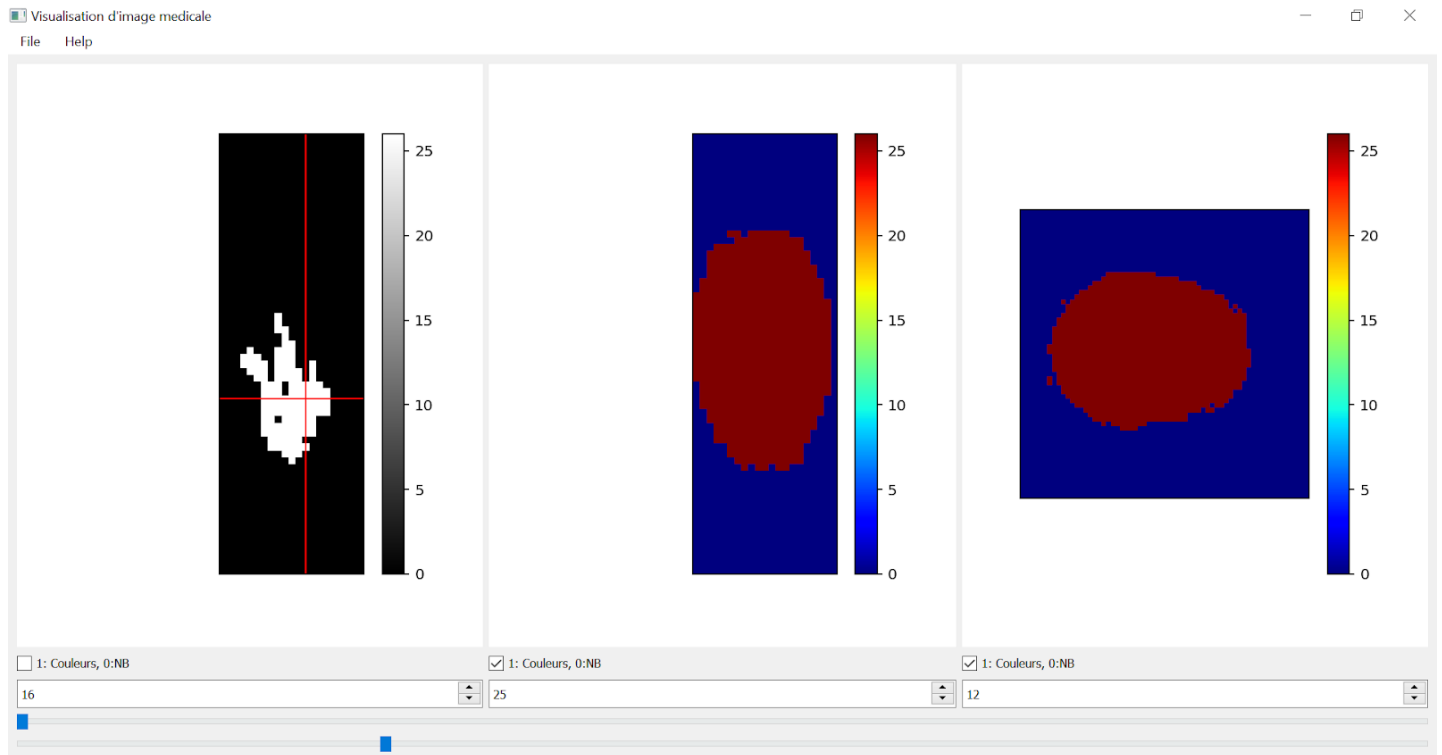


Figure 6: Ouverture d'un fichier Nifti

V. PROJET FUTUR

Dans le futur, nous pourrions essayer de visualiser le fichier en 3D tout en le bougeant selon l'axe x y ou z. L'application pourrait également être convertie sur Android pour pouvoir visualiser les différentes coupes sur tablette et ainsi rendre l'application encore plus intuitive et facile à utiliser.

VI. CONCLUSION

La réalisation de ce projet nous a permis d'introduire un aspect nouveau de programmation en python à savoir la Programmation Orientée Objet (POO). En effet, celle-ci nous a permis de créer une interface graphique via Python. Dans notre cas, nous avons réalisé une application permettant de faciliter le travail des radiologues qui croulent sous les fichiers générés par les différents appareils d'imagerie médicale.

En effet notre application, permet de sélectionner et d'ouvrir tous les fichiers présent dans un dossier qui sont générés lors d'un scanner ou IRM d'un patient. Après ouverture les fichiers peuvent être analysés sous 3 coupes en même temps (axiale, coronale et sagittale). Cela permet de gagner en rapidité pour le scientifique qui doit analyser les différentes coupes. De plus, l'application est intuitive et facile d'utilisation, une fenêtre About peut être générée en cas de difficulté pour comprendre l'application.

VII. ANNEXE

1. Table des Figures

Figure 1: Schéma de l'interface graphique.....	3
Figure 2: Ouverture d'un fichier Dicom.....	6
Figure 3: Utilisation des widgets sur un fichier Dicom.....	7
Figure 4: Utilisation du curseur sur un fichier Dicom.....	8
Figure 5: Fenêtre About	8
Figure 6: Ouverture d'un fichier Nifti	9

2. Fichier python

Le fichier python est en annexe de ce dossier.