

Rapport SAE6.01 - MAC

Modifications apportées :

Pour compiler le projet sur macOS, nous avons rencontré un problème lié à l'utilisation de **HDRImageViewer.exe**, qui n'est pas compatible avec ce système d'exploitation. Afin de résoudre ce problème, nous avons modifié le code pour utiliser l'application « Preview », qui est intégrée à macOS.

Les modifications spécifiques ont été effectuées dans deux fonctions du fichier **controller.py** :

1. **close()** : Cette fonction a été ajustée pour s'assurer que l'application « Preview » est correctement fermée après l'affichage de l'image.
2. **displayFile()** : Cette fonction a été modifiée pour ouvrir les fichiers d'image avec l'application « Preview » au lieu de **HDRImageViewer.exe**.

Ces ajustements permettent désormais au projet de fonctionner correctement sur macOS en utilisant les outils natifs du système pour l'affichage des images.

```
def displayFile(self, HDRfilename):
    print('Starting displayFile method: ', HDRfilename)
    # check that no current display process already open
    if self.viewerProcess:
        # the display HDR process is already running
        # close current
        process_name = "Preview"
        try:
            pid = subprocess.check_output(['pgrep', '-f', process_name]).decode('utf-8').strip()
        except subprocess.CalledProcessError:
            pid = None

        if pid is not None:
            os.kill(int(pid), signal.SIGTERM)
        else:
            print(f"No process found with name {process_name}")
            time.sleep(0.05)

    # run display HDR process
    self.viewerProcess = subprocess.Popen(["open", "-a", "Preview", HDRfilename], shell=False)
    time.sleep(0.10)
    psData = subprocess.run(['ps', '-A'], capture_output=True, universal_newlines=True).stdout
    if not 'Preview' in psData:
        # re-run display HDR process
        self.viewerProcess = subprocess.Popen(["open", "-a", "Preview", HDRfilename], shell=False)
```

```
def close(self):
    # check that no current display process already open
    if self.viewerProcess:
        # the display HDR process is already running
        # close current
        process_name = "Preview"
        pid = subprocess.check_output(['pgrep', '-f', process_name]).decode('utf-8').strip()
        os.kill(int(pid), signal.SIGTERM)
        self.viewerProcess = None
```

Après avoir effectué ces changements, nous avons recompilé l'application et vérifié son fonctionnement sur macOS. L'application se lance correctement et, lors de la sélection du dossier contenant les images, l'image "grey.hdr" s'affiche sans problème dans l'application « Preview ».

Problème rencontrée :

Après avoir effectué ces changements, nous avons recompilé l'application et vérifié son fonctionnement sur macOS. L'application se lance correctement et, lors de la sélection du dossier contenant les images, l'image "grey.hdr" s'affiche bien dans l'application « Preview ».

Cependant, nous avons rencontré un problème supplémentaire. Une fois le dossier d'images choisi, les images ne s'affichaient pas dans l'application. Pourtant, le chemin affiché en bas de l'application était correct et accessible, comme nous l'avons vérifié en utilisant la commande ls sur le chemin donné.

Pour récapituler les modifications :

1. **Problème initial : HDRImageViewer.exe**, utilisé initialement, n'est pas compatible avec macOS.
2. **Solution** : Nous avons remplacé **HDRImageViewer.exe** par l'application « Preview » intégrée à macOS.
3. **Modifications spécifiques** :
 - o **close()** : Adaptée pour s'assurer que l'application « Preview » se ferme correctement après l'affichage de l'image.
 - o **displayFile()** : Modifiée pour ouvrir les fichiers d'image avec « Preview » au lieu de **HDRImageViewer.exe**.

Problème supplémentaire :

- **Affichage des images** : Malgré la sélection du dossier d'images et le chemin correct affiché, les images ne s'affichaient pas dans l'application elle-même. Nous avons confirmé que le chemin était correct et accessible en utilisant la commande ls.

```
read image: /Users/samuelcaron/dev/uHDR/images/hdr/0246_HDR.hdr
> ls /Users/samuelcaron/dev/uHDR/images/hdr
  0216_HDR.hdr      {} 0243_HDR.json      0269_HDR.hdr
 {} 0216_HDR.json   {} 0246_HDR.hdr      {} 0269_HDR.json
  0234_HDR.hdr      {} 0246_HDR.json      0275_HDR.hdr
 {} 0234_HDR.json   {} 0260_HDR.hdr      {} 0275_HDR.json
  0237_HDR.hdr      {} 0260_HDR.json      0281_HDR.hdr
 {} 0237_HDR.json   {} 0263_HDR.hdr      {} 0281_HDR.json
  0240_HDR.hdr      {} 0263_HDR.json      ➤ thumbnails/
 {} 0240_HDR.json   {} 0266_HDR.hdr
  0243_HDR.hdr      {} 0266_HDR.json
```

Malgré de nombreux tests et phases de débogage, nous n'avons pas réussi à résoudre et localiser précisément le problème. Après un point avec M. Cozot et plusieurs tests, nous avons conclu que le problème provenait du système de **multi-threading** de Qt, incompatible avec macOS. En chargeant une image sans **multi-threading** avec Qt, l'image s'affichait correctement. Cependant, avec le **multi-threading**, les fonctions de chargement d'image et de fin de thread n'étaient pas exécutées, entraînant des problèmes d'allocation de mémoire.

BERDIN Clément
CARON Samuel

Autres :

Pour afficher des images HDR :

- Fragment
- Preview
- LuminanceHDR

Écran intégré MacBook Pro M2 :

Displays:

Colour LCD:

Display Type:	Built-In Retina LCD
Resolution:	2560x1600 Retina
Main Display:	Yes
Mirror:	Off
Online:	Yes
Automatically Adjust Brightness:	Yes
Connection Type:	Internal

Écran intégré MacBook Air M2 :

Moniteurs :

LCD couleur :

Type de moniteur :	Écran Liquid Retina intégré
Résolution :	Retina 2560 × 1664
Moniteur principal :	Oui
Miroir :	Désactivé
Connecté :	Oui
Régler la luminosité automatiquement :	Oui
Type de connexion :	Interne