Image Annotation Software

Introduction

In this project, we had to implement an image annotation software. It can be used to annotate images, using a minimalistic but effective UI. The UI associate labels defined by users to specific zones on a custom imported image. It can be used in the context of machine learning to train a neural network to recognize images.

Choice of API

For this project, the main difficulty was to choose an API to design the UI. At first, we looked for information about Kivy, a Python API. We found the API rather complex and, coupled by our inexperience in Python, we looked for a Java API.

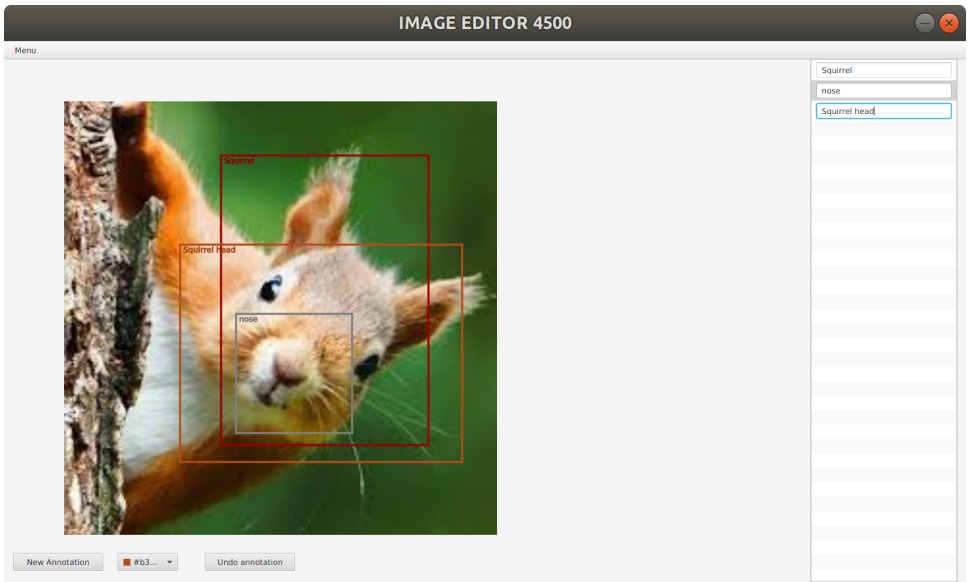
At this time, we tried to use Swing, which we had already used for a previous project. We began the developpement but soon we felt the limitations of Swing. After the first few components, we felt like organizing the UI was an overwhelming task in Swing so we looked for JavaFX in association with SceneBuilder and found the perfect mixup between our existing skills and learning a new API.

Functionalities

Our image annotation software implement multiples functionalities:

1. The ability to import images at any time.
2. The ability to draw a selection to highlight a zone of the image.
3. The ability to choose the color of the selection frame.
4. The user can label the selections using a list of text fields.
5. The user can undo the last annotation.
6. The user can exit the application.

Screenshots



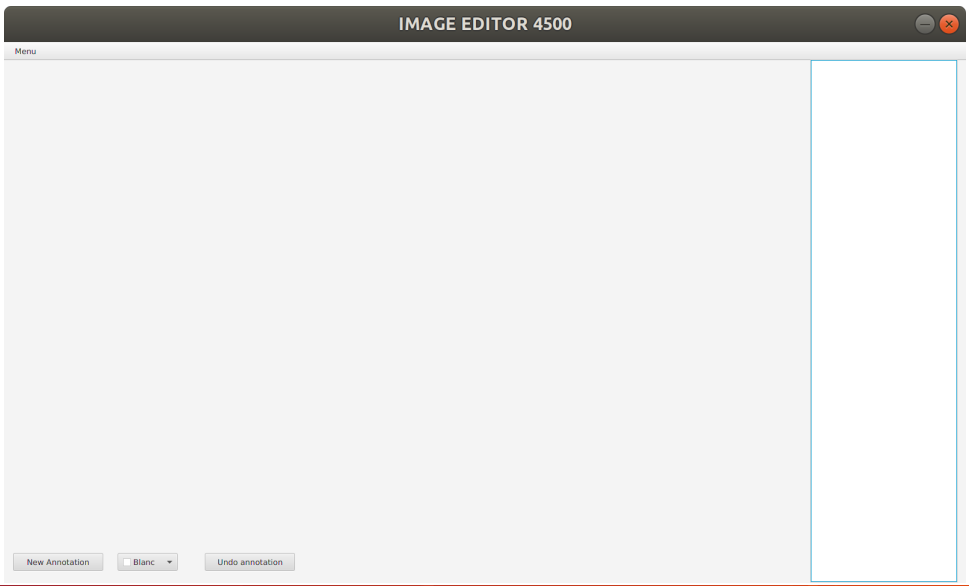


IMAGE EDITOR 4500
Menu
New Annotation
Blanc
Undo annotation

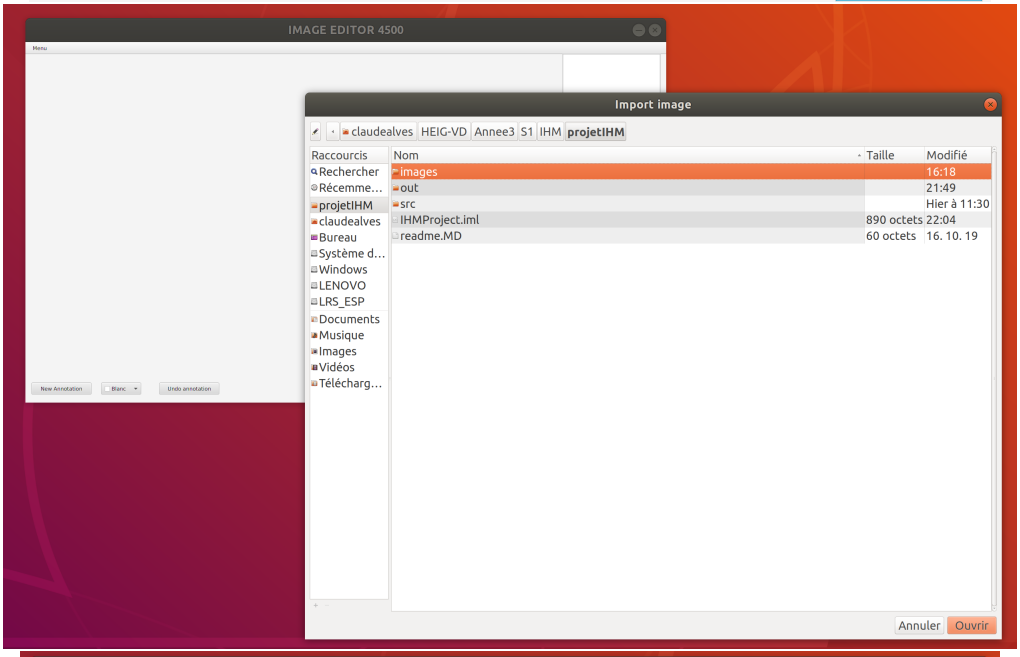


IMAGE EDITOR 4500
Import image
claudealves
HEIG-VD
Annee3
S1
IHM
projetIHM
Raccourcis
Rechercher
Récemme…
projetIHM
claudealves
Bureau
Système d…
Windows
LENOVO
LRS_ESP
Documents
Musique
Images
Vidéos
Télécharg…
Nom
Taille
Modifié
images
16:18
out
21:49
src
Hier à 11:30
IHMProject.iml
890 octets
22:04
readme.MD
60 octets
16. 10. 19
Annuler
Ouvrir

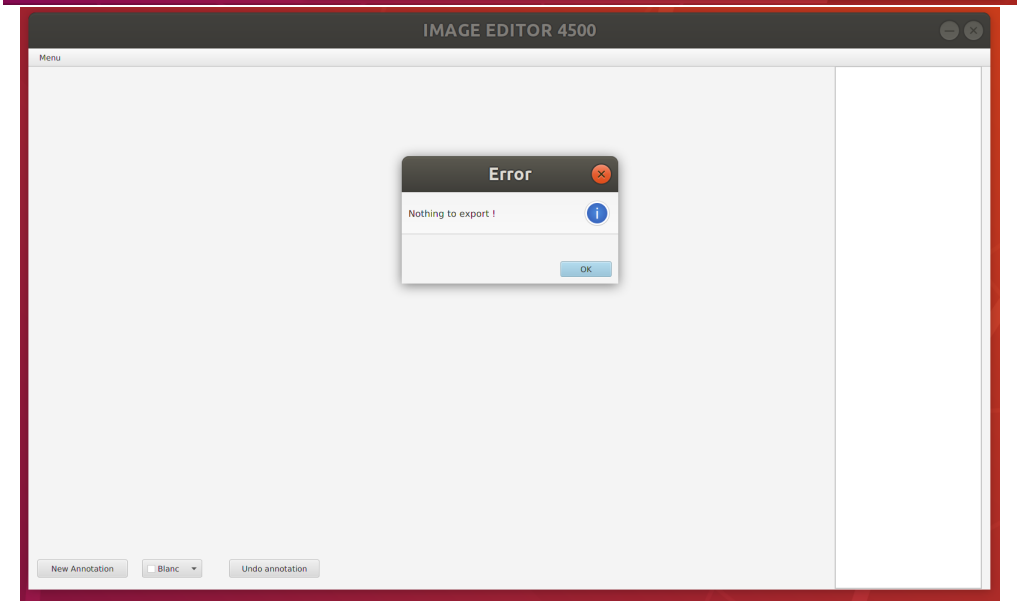


IMAGE EDITOR 4500
Menu
Error
Nothing to export !
OK
New Annotation
Blanc
Undo annotation

Conclusion

This laboratory was great for us to improve at designing a UI. The downside was that we spent more time on the features implementation than we spent on designing. Also, the ability to save the state was way too challenging for us. We implemented the save system, but at some point we had to save annotation in a JSON format, but building a custom parser was a long task we did not have to time to do. Nevertheless, we learned a lot about JavaFX, which was also the goal of the laboratory.