

0.1 File monitoring

The ability to interact with other programs such as gimp through drag and drop is an important feature of Stopmotion. We wanted to make this feature more responsive so we decided to implement a file monitoring scheme.

Stopmotion uses a daemon service called FAM (File Alteration Monitor)[?] to monitor the image files in the opened project. If one of these files change FAM will notify Stopmotion and the file will automatically be reloaded. This result in a very responsive way of interfacing to other applications and aids the user in creating stop motion animations. As stated on the FAM webpage referred to above:

GUI tools should not mislead the user; they should display the current state of the system, even when changes to the system originate from outside of the tools themselves. FAM helps make GUI tools more usable by notifying them when the files they're interested in are created, modified, executed, and removed.

Stopmotion has a class in the Application layer called ExternalChangeMonitor which supervises the file monitoring. This class communicates with FAM by using a library called libFAM. It has functions for starting, stopping, suspending and resuming monitoring as well as a function for specifying a new working directory when the user saves a project for the first time.

When it receives a request to start monitoring it attempts to open a connection to FAM through the function `FAMOpen2(. . .)`. If this was successful it tells FAM to monitor the tmp directory with the function `FAMMonitorDirectory(. . .)`.

After the fam connection is set up the ExternalChangeMonitor creates a QSocketNotifier which monitors the fam connection for events. The socketNotifier will run in the main event loop and frees us from the complications and drawbacks involved in creating a thread for polling the connection.

When a user drags a file from Stopmotion to another program, as explained in section ?? on page ??, a URI drag is performed. That means the path to the file is passed to the retrieving application and loaded there. If the user for instance drags the file over to GIMP, GIMP will work directly on the file.

When the user saves the file it is immediately notified by FAM. Stopmotion receives many notifications while GIMP saves the file, so when it receives the first notification it starts checking every 500ms if it has received another notification. If it hasn't it means that GIMP has saved and is done with the file. The following function in the DomainFacade is then called:

```
animationChanged (const char *alteredFile)
```

This triggers the notifyAnimationChanged event and all the Observers are notified of the change through the function:

```
updateAnimationChanged(int frameNumber)
```