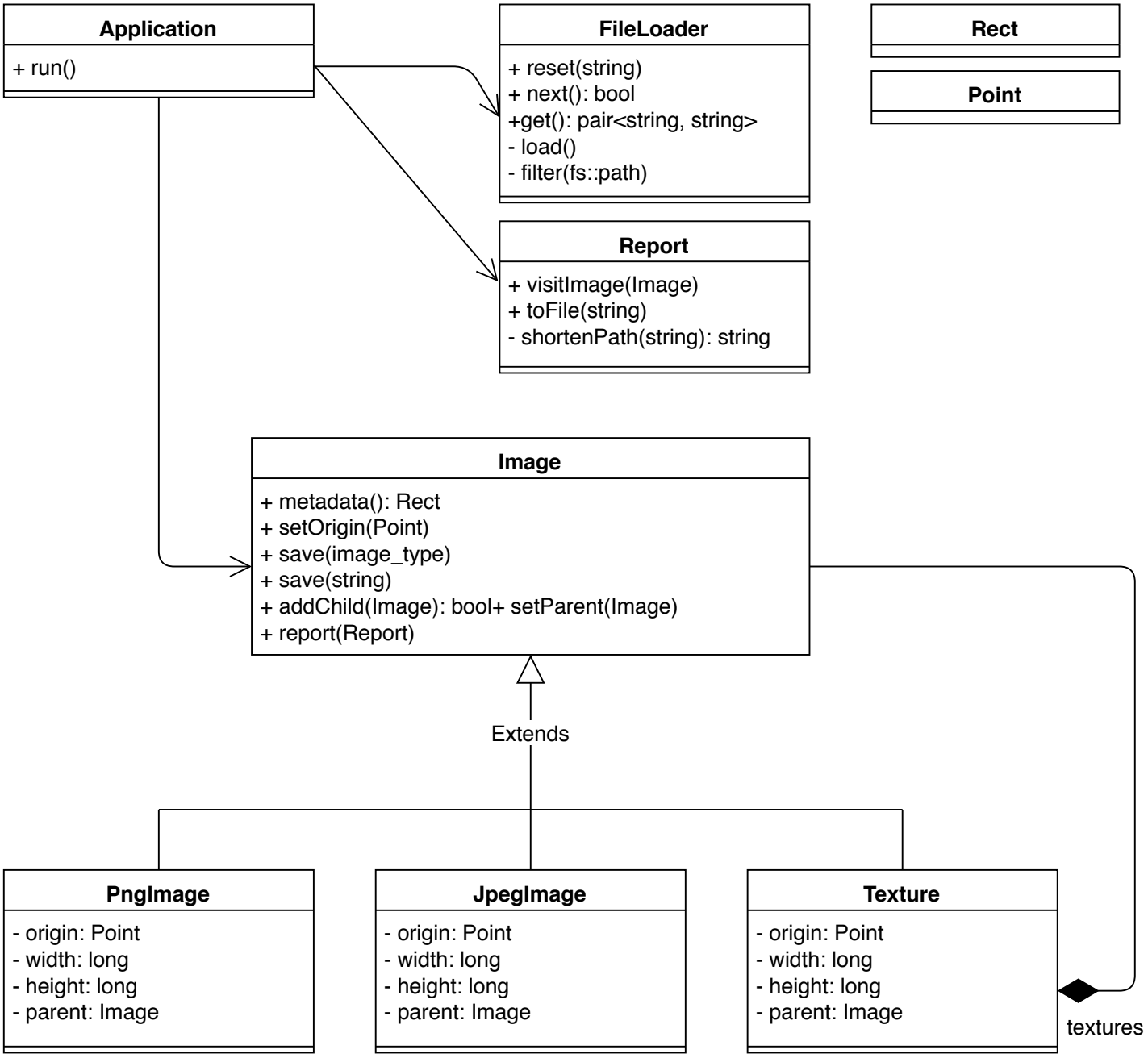
Class diagram

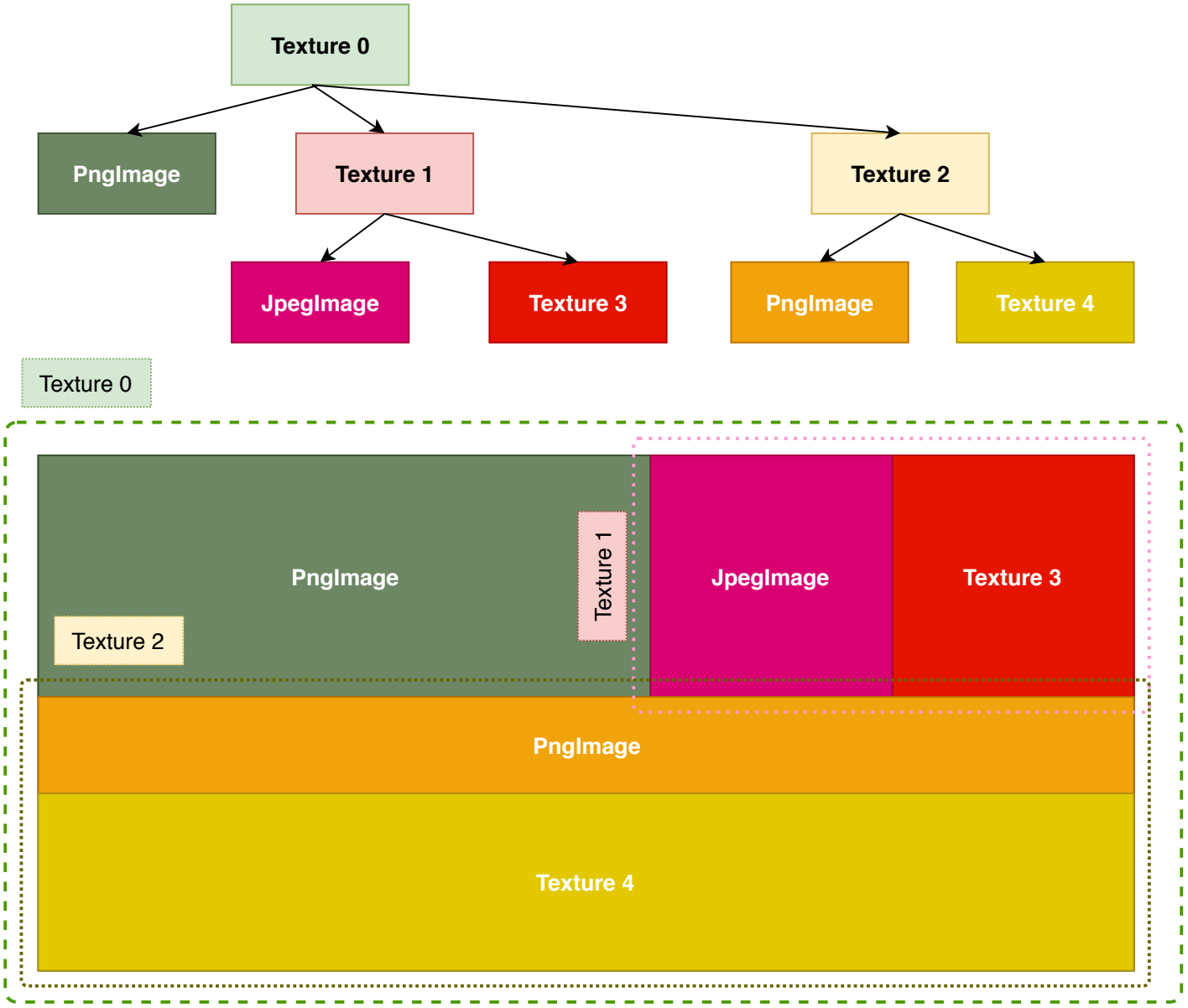


Implementation

The texture atlas is contructed following these steps:

- A FileLoader traverses the given directory and get the lists of png and jpeg images. With C++17, the stl has filesystem so that I can write platform-indenpendent code.
- The optimal size of the generated image is calculated based on the lists of images.
- Finally, an empty texture is created with the calculated size. Application traverses through the lists of png and jpeg image and add them to the texture. An png image and a metadata file (collected by a Report) are saved to the disk as the result.

Texture is the composite which store child components: PngImage, JpegImage, and other Texture. When an image is added the the texture, the root texture will find the next free space that the image can be fit. Then the texture, which has enough free space, adds the image into the list as a child and create new children textures. An example of the structure is illustrated below:



Dependencies

Catch 2: Unit test
Dlib: writing and reading image file

Build instructions

You can find here: https://github.com/tuangu/texture-atlas

Known issues

Unfortunately, I haven't found any issues yet. Please give it a try and contact me if there is any.