mp_stickers

Images and Stickers

Extra credit: Feb 04, 23:59 PM
Due: Feb 11, 23:59 PM

Doxygen

Solo MP This MP, as well as all other MPs in CS 225, are to be completed without a partner.

You are welcome to get help on the MP from course staff, via open lab hours, or Piazza!

Checking Out the Code

From your CS 225 git directory, run the following on EWS:

```
$
```

```
git fetch release
git merge release/mp_stickers -m "Merging initial mp_stickers files"
```

If you're on your own machine, you may need to run:

```
$
```

```
git fetch release
git merge --allow-unrelated-histories release/mp_stickers -m "Merging initial
mp_stickers files"
```

Upon a successful merge, your mp_stickers files are now in your `mp_stickers` directory.

Part 1 (Curated): The `Image` Class

An `Image` object is a subclass of the `PNG` class. This means it inherits all the member functions of the `PNG` class; so anything you could do with a `PNG`, you can also do with an `Image`.

After creating the `Image` class, implement the methods of the `Image` class:

Open the doxygen for class Image

Testing

When you've finished this part, you can `make` and run Part 1 by running:

```
$
```

```
make testimage
./testimage
```

If execution goes smoothly, images named `lighten.png`, `saturate.png`, and `scale2x.png` will be created in your working directory.

- The files `expected-lighten.png` and `expected-saturate.png` are provided and can be `diff`ed with your output.
- The file `expected-scale2x.png` is somewhat misnamed, as there are **many** correct solutions when you scale an image. *It is not necessary to have it `diff` to the same image.* So long as your scaling algorithm creates a reasonable scale of the original image, our autograder will see it as a reasonable scaling of the image. You can verify this by running the automated tests on Part 1.

Automated Testing

To test your code using Catch, run the following:

```
$
```

```
make test
./test
```

Extra Credit Submission

For a few bonus points, you can submit the code you have implemented and tested for part one of mp_stickers. You must submit your work before the extra credit deadline (given above). Although this is optional, we encourage everyone to do this for a couple reasons:

- if you get a sufficient grade on the submission, you will receive bonus points to improve your grade.
- regardless of the quality of your work, you will get feedback that can be used to improve your grade on the required submission of mp_stickers.

Be sure to commit and push your work before the extra credit deadline to earn extra credit.

Guide: How to submit CS 225 work using git

Part 2 (Curated): The `StickerSheet` Class

Lets add stickers on top of an image!

Your goal in this part of the MP is to make a `StickerSheet` composed of a collection of `Image`s. To do so, you will create a class `StickerSheet` that will maintain an array of pointers to `Image` objects. Each `Image` in the `Scene` will have an index, an -coordinate, and a -coordinate. The member functions described below will support creating, modifying, and drawing the collection of `Image` stickers in the `StickerSheet`.

To implement the `StickerSheet` class, you will write a header file that contains a declaration of the `StickerSheet` class (`StickerSheet.h`) and a source file that contains the implementation of the `StickerSheet` class (`StickerSheet.cpp`).

To see all the required functions, check out the Doxygen:

Open the doxygen for class StickerSheet

Part 3 (Creative): Create an image with stickers!

For the last part of this MP, in the `main` function in `main.cpp` create a `StickerSheet` that contains an image and at least three stickers. Before exiting `main`, save your creation to disk as `myImage.png`.

We'll take a look at your photo filled of stickers! Keep it clean and something you're okay being shared with the class so we can show the best ones off to the whole class! :)

To generate your creative `StickerSheet`, you can use the following commands.

```
$
```

```
make
./stickers
```

Sharing Your StickerSheet

> **You just made something awesome that never existed before -- you should share your sticker sheet (but do not have to)!**
> *If you share your StickerSheet on Facebook, Twitter, or Instagram with **#cs225**, I will 👍 or ❤ the post as soon as I see it. I think many of your peers will too! — Wade*

Testing

When you've finished Part 2 and Part 3, you can `make` the full MP by running:

```
$
```

```
make test
./test
```

Automated Testing

To test your code using Catch, you will need to enable the Part 2 test cases. To do so, go into `tests/part2.cpp` and uncomment the commented section at end of the file.

As you saw when you uncommented the test case, the test case is **deliberately insufficient**. We strongly recommend augmenting these tests with your own.

Once you're ready to run the automated tests, run:

```
$
```

```
make test
./test
```

Handing in your code

You must submit your work to git for grading. We will use the following files for grading:

- `Image.cpp`
- `Image.h`
- `StickerSheet.cpp`
- `StickerSheet.h`
- `main.cpp`
- `myImage.png`

All other flies will be ignored in grading.

Guide: How to submit CS 225 work using git