Change Request 3

Due date: 4/10/2018

1. Histogram

Add a new capability that can calculate the current image’s histogram (<https://en.wikipedia.org/wiki/Image_histogram>) and show it on a new canvas. For simplicity, you could just do the calculation on a grey-scale image.

1. Apply Effect On Selected Region

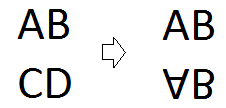
Currently you can apply an effect on the entire canvas, like blur, sharp, etc. Add a new capability that can apply the effects on a certain selected region.

1. Apply Effect Outside of Selected Region

Contrary to the task 2, apply the effects to the outside of the selected region.

1. Mirror Mode (horizontal)

Add a new capability named horizontal mirror mode. In this mirror mode, the canvas is divided by upper and lower half. All the figures that are drawn in one half should be mirrored to the other half each time. E.g. mirror the upper half:



Note the users must be allowed to decide which half to mirror.

1. Mirror Mode (vertical).

Add a new capability named vertical mirror mode. In this mirror mode, the canvas is divided by left and right half. All the figures that are drawn in one half should be mirrored to the other half each time. E.g. mirror the left half:



Note the users must be allowed to decide which side to mirror.

1. Signature tool

Add a new signature tool. EasyPaint should be able to load a predefined signature from image files. Users stamp the signatures onto the canvas by mouse clicking on any position they want.

|  |  |
| --- | --- |
| Name | Change Number |
| Leng | 5 |
| Varmitha | 6 |
| John | 4 |
| Amanda | 2 |
|  |  |
|  |  |

NOTE: Fill out the above form and push to your group folder on the Git repository. Each request can be selected at most by one member of the team.

**List of deliverables:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **What** | **Where & When** | **Description** | **Points** |
| **1** | **Git push** | Git server, 4/10/2018, before the beginning of the lab | Before the due time, you are required to correctly push your source code to the assigned directory of the Git repository. Pushing to a wrong directory or ruining the baseline due to incorrect operations will result in a failing grade (0 points) for this part. Make sure you resolve any conflict prior to your push. | 50 |
| **2** | **Report,**  **1st version** | Blackboard,  3/27/2018,  before the beginning of the lab | You are required to submit an electronic copy of the report describing how you used the incremental change process to implement the change request, in the format defined by *Report\_Format.docx*.  The 1st version of your report must complete sections 1-3. | 40 |
| **Final Report\*** | Blackboard ,  4/10/2018,  before the end of the lab | Complete the rest of your report. It should also contain the fragments of the source code files that were modified/added, with the code highlighted as specified. | 35 |
| File Name: **LastName\_FirstName\_Change#.docx** |
| **3** | **Refactoring** | Make a plan,  3/27/2018,  before the beginning of the lab | During the impact analysis, make a plan for doing a prefactoring in your inspected classes. You can extract a method, create a superclass, merge several entities, etc., but it cannot be just renaming entities or making a new method which simply calls the existing one. You should change your plan after the group meeting if more than one person refactor the same code snippet in the team. | 0 |
| Git server,  4/3/2018,  before the beginning of the lab | When implementing the refactoring, the original code must stay as comments in the source code with your individual information (e.g. name, other necessary comments). You also need to leave comments to distinguish your code after refactoring (i.e. refactored code).  If you remove the original code completely from the baseline or forget to leave personal information for refactored code, your refactoring will be considered as invalid and lose all points for this part. | 25 |
| **4** | **Group Meeting** | In lab,  3/27/2018 | As required in the report part, everyone must complete the impact analysis before the meeting. Discuss your estimated impact set as well as your refactoring with your teammates to find out all possible conflicting files. Then negotiate some deadlines before the final due time for committing those files individually and submit the schedule to the repository (clarify the final individual prefactoring plan as the team record). Anyone missing the group meeting will be considered as admitting the schedule decided by his/her teammates. | 0 |
| **5** | **Refactoring Demo\*\*** | In Lab,  4/3/2018 | Each student will do a demonstration of the prefactoring. Individual demo. | 25 |
| **6** | **Team Demo\*\*** | In Lab, 4/10/2018 | Each team will do a demonstration of the team project. | 25 |

\* You are required to test your code by running it with some cases (functional testing). You should specify and record the cases you use in the verification section of your report (E.g. After implementing the change request X, I drew a dashed line out of the canvas and failed to do so. Then I drew a dashed line inside the canvas successfully. So this capability worked correctly for my test cases.).

\*\* Fail to demo your change in the class time will get 0 for that part. In other words, later demo will lose all credit of the demo part. Repository and report parts still follow the late policy in the syllabus.

Do NOT push any redundant file to the repository (DLLs, binary files folder, etc.). Move the new source files in correct folders if you add them (inspect the folders in the 1st baseline of easyPaint to make sure).