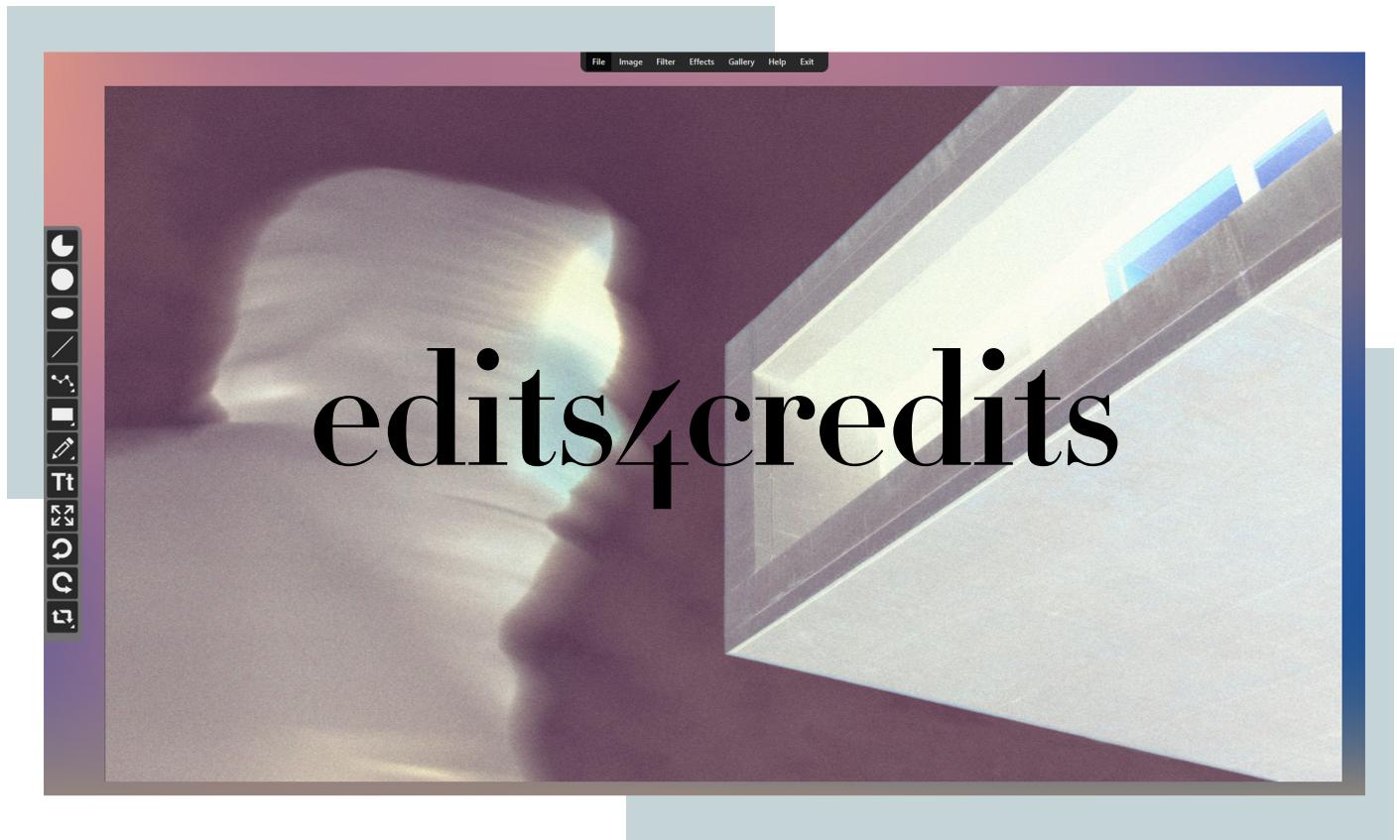


Software Entwicklung 2



Tevin Zielke, tz024@hdm-stuttgart.de
Johannes Rödel, jr125@hdm-stuttgart.de
Yannick Kebbe, yk025@hdm-stuttgart.de

<https://gitlab.mi.hdm-stuttgart.de/e4c/edits4credits>

e4c.

Kurzbeschreibung

Bei unserem Projekt handelt es sich um einen graphischen Design-Editor für das zweite Semester im Studiengang Medieninformatik an der Hochschule der Medien.

Wenn man das Programm startet gelangt man zuerst zu einem Login Screen. Aktuell kann man zwar schon Username und Passwort auslesen und überprüfen. Jedoch muss erst noch eine Datenbank implementiert werden, um eine korrekte Registrierung und Anmeldung zu gewähren. Wenn man den Login Button drückt gelangt man zu einer Auswahl, bei der man Entscheiden kann, ob man den Editor oder die Galerie öffnen möchte. Die Hauptfunktion unseres Editors besteht darin, Projekte bzw. Bilder zu erstellen und diese dann entweder in der Programm internen Galerie oder extern zu speichern. Das Programm ist auf Erweiterung mit einer User–Datenbank ausgelegt, in der Projekte online gespeichert werden und in einer Galerie zur Auktion angeboten werden können. Aktuell kann man in der Galerie Bilder speichern und ansehen.

Projekt:

Startet man nun den Editor, erscheint ein Einstellungsfenster, in der projektspezifische Daten wie Name, Breite und Höhe in Pixeln und Hintergrundfarbe festgelegt werden können. Außerdem kann man sich dazu entscheiden die Galerie zu öffnen, um von dort aus ein Bild als Hintergrund zu nehmen. Die Höhe und Breite passen sich dann automatisch an die Maße des Bildes an.

Editor:

Im Editor angelangt begegnet einem die typische Bearbeitungsfläche. Am oberen Bildschirmrand befindet sich eine Menüleiste, mit der Punkte wie Projekt Im- und Export, Transformations Tools, Filter und Zugang zur Galerie gesteuert werden können. Auf der linken Seite befinden sich verschiedene Tools zur grafischen Bearbeitung mit unterschiedlichen Formen und Pinseln, mit denen auf das Bild gemalt werden kann.

Kurzbeschreibung

Filter/ Effects:

Öffnet man die Filter- bzw Effectoptionen über den jeweiligen Buttons, in der Menüleiste, öffnet sich ein Fenster, mit den Einstellungen der jeweiligen Filter und Effekte. Die Filter werden aktuell nur auf importierte Bilder angewendet. Möchte man sie auf gezeichnete Shapes anwenden, sollte man die Shapes zunächst exportieren und als Image importieren. Unser Programm bietet die Möglichkeit eine Vorschau des Filters im Editor Fenster anzeigen zu lassen, wenn man die Attribute im Optionsfenster ändert. Möchte man den Filter anwenden muss man dies über den Apply Button bestätigen. Schließt man das Optionsfenster ohne den Apply Button zu drücken, wird das Bild im Editor wieder auf seine ursprüngliche Version zurückgesetzt.

Transformation:

Das Transformations Tool lässt sich ebenfalls über die Menüleiste öffnen. Eine Auswahl an Möglichkeiten befindet sich unter dem Menüpunkt Image. Mit dem Scale Tool kann man das importierte Bild skalieren. Ist der Scale Faktor größer als 100 wird das Bild vergrößert. Ist der Wert kleiner als 100 wird das Bild verkleinert. Die Position des Bildes kann mit dem Move Tool verändert werden. Über die X und Y Position im Bildraster kann man festlegen wo das Bild platziert werden soll. Hier kann man auch negative Werte eingeben. Auch die Transformations Tools bieten eine Preview bevor man die veränderten Werte bestätigt. Über die Enter Taste wird eine Vorschau im Editor Fenster dargestellt. Beim Positions Tool müssen ebenfalls die neuen Werte bestätigt werden.

Kurzbeschreibung

Formen und Pinsel:

Das Drawing Tool beinhaltet die Formen: Arc, Circle, Elipsis, Line, (Rounded) Rectangle, Polygon und Text.

Zudem befinden sich Buttons für ein Pinsel Tool, das Bewegen von Formen, Step Forward und Step Backward und zum leeren der gesamten Leinwand. Allgemein wird zwischen Pinselleinwand und Formleinwand unterschieden, wobei die Pinselleinwand über der Formleinwand liegt.

Sobald man auf einen der Form- oder Pinselbutton klickt öffnet sich ein Optionsfenster in dem man eine Vielzahl an Einstellungen vornehmen kann. Nach dem Schließen werden diese auf die nächsten Operationen übertragen, solange bis wieder ein Button aus der Leiste gedrückt wird.

Alle Drawing Tools lassen sich mit der Maus steuern. Das Malen mit dem Pinsel und das Erzeugen von Formen wird mit der Maus bedient.

Zudem lassen sich mit dem Move Tool Formen über die Leinwand ziehen. Diese legen sich dabei immer auf oberste Ebene. Es sollte beachtet werden, dass sich die Pinselstriche nicht bewegen lassen.

Alle Operationen der Formen werden in einer Historie gespeichert. In dieser kann mit dem Step Forward- und Step Backward Tool zwischen den Versionen gewechselt werden Jedoch sollte man beachten, dass wenn man zurück geht und dann eine weitere „Draw“- oder „Move“-Operation durchführt, die restliche Historie gelöscht wird und ab diesem Punkt weiter gearbeitet wird.

Als Hinweis sind bei den Tools, bei denen ein Kontextmenü beigelegt ist, mit einem kleinen Dreieck unten rechts versehen (i.d.R. von oben: Polygon -> Draw, Rectangle -> Rounded Rect., Brush -> Eraser, Clear Shapes -> Clear Brush).

Kurzbeschreibung

Gallery:

Die Galerie selbst ist eine Oberfläche um einfache Bilder oder vergangene Projekte darzustellen. Das Programm erstellt beim Start des Programms, sofern nicht vorhanden, ein verstecktes Directory unter dem user:home Verzeichnis mit dem Namen /.edits4creditsgallery Sollten sich in dieser schon vorher Bilder befinden, werden diese in die Galerie geladen. Zudem werden Projekte, welche im Editor unter save to Gallery gespeichert werden automatisch in dem Verzeichnis gespeichert. Wenn man auf ein Bild klickt öffnet sich ein Popup mit dem vergrößerten Bild und einem Button. Dieser zeigt je nachdem die Worte "Use" oder "Delete" an. "Use" und "Delete" wird nur angezeigt, wenn man vorher oben in der Leiste unter Gallery delete auswählt. Klickt man auf "Use" wird das Bild entweder im Editor importiert oder als Basis vom Projekt genommen, je nachdem von wo aus die Galerie geöffnet wird. Zudem kann auch durch Gallery > Import Bilder in die Galerie importiert werden.

Startklasse &

Besonderheiten

Startklasse:

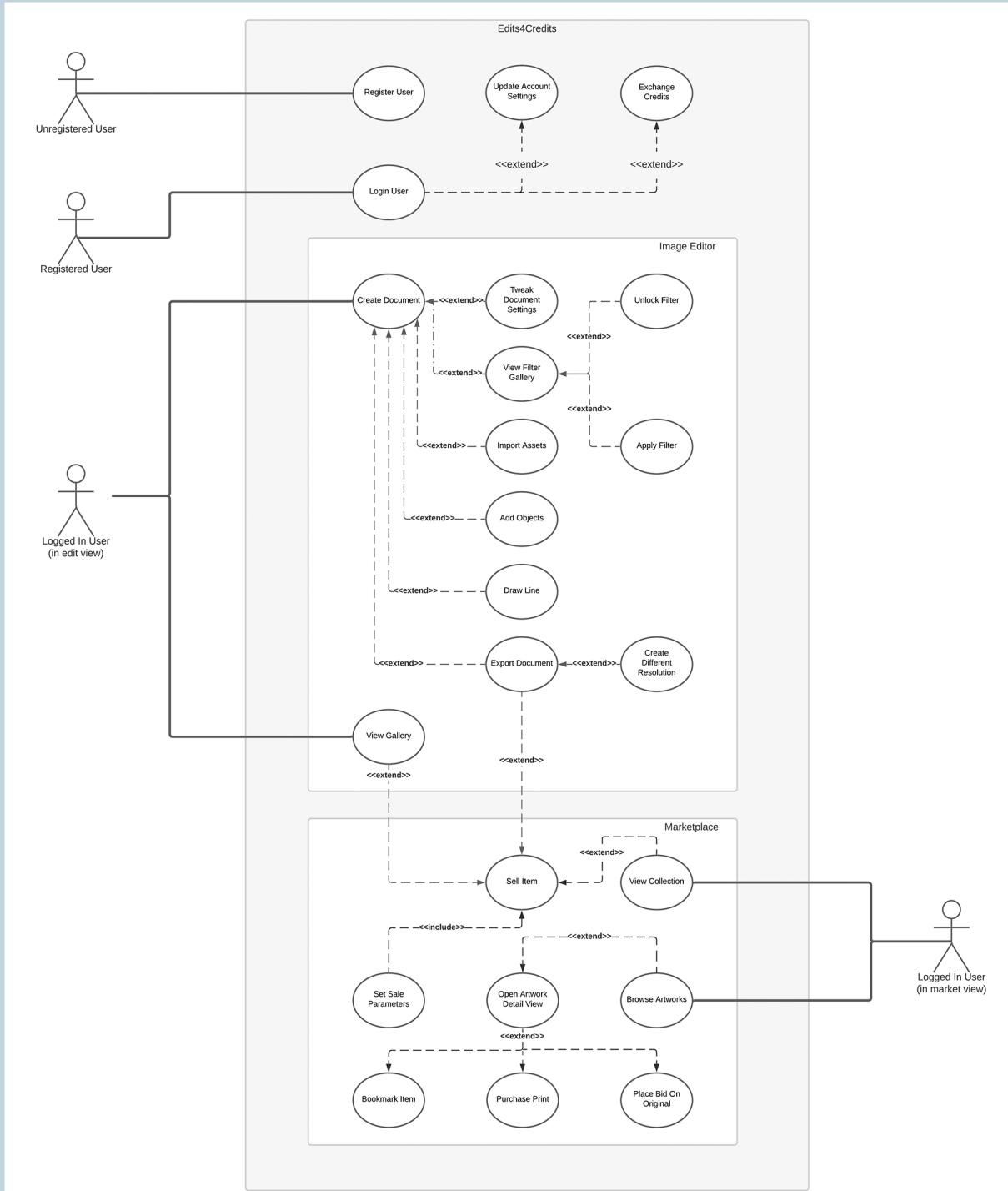
Die Main- bzw. Start-Methode befindet sich in der Klasse GuiDriver im Package org/launcher/GuiDriver.java.

Besonderheiten:

Beim Login muss man keine Daten eingeben. Man kommt zum nächsten Screen, wenn man den Login Button betätigt. Aktuell ist läuft das Programm noch lokal und noch nicht über eine Online Datenbank, wie Anfangs geplant.

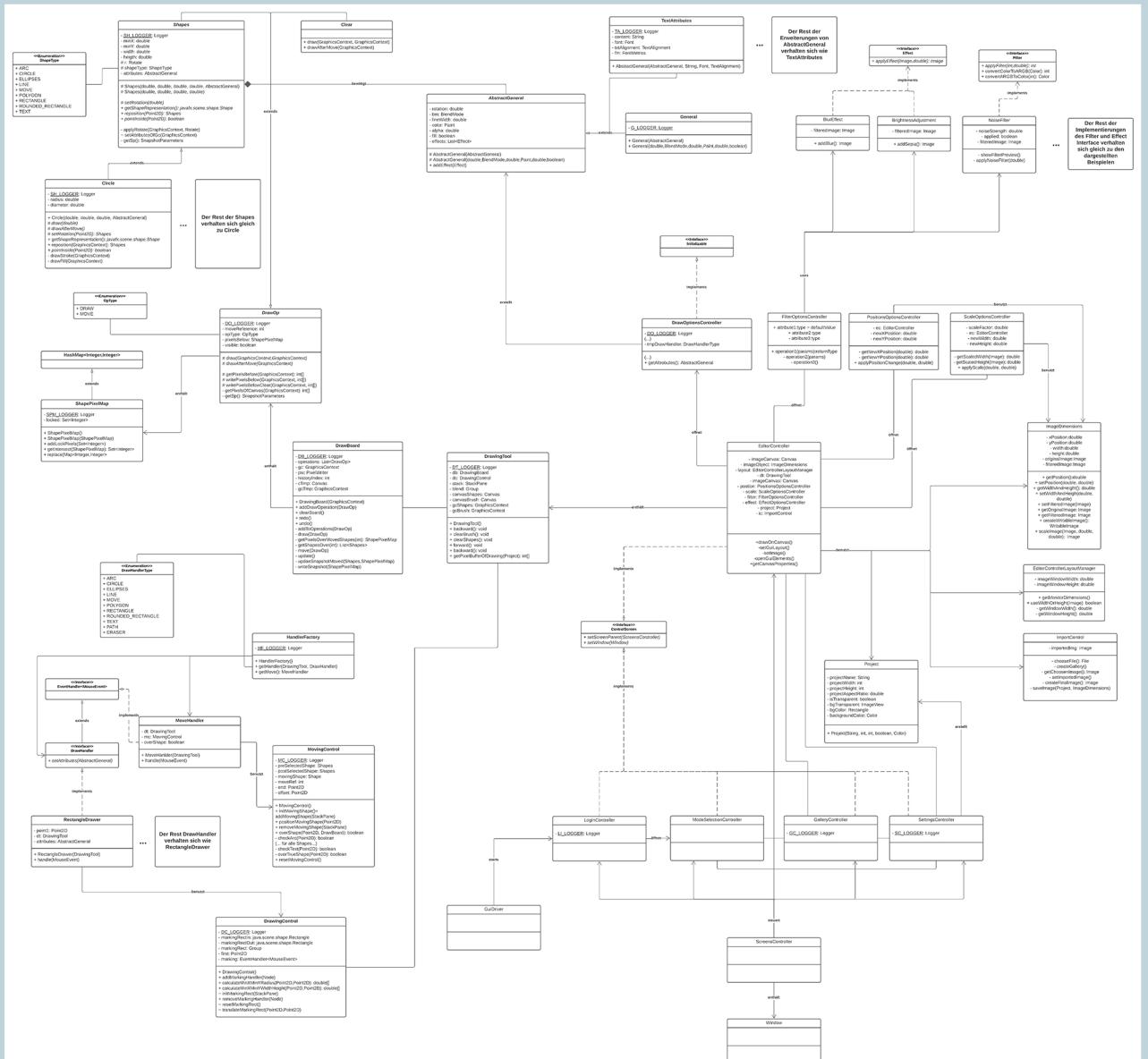
UML Diagramme

Use-Case-Diagramm



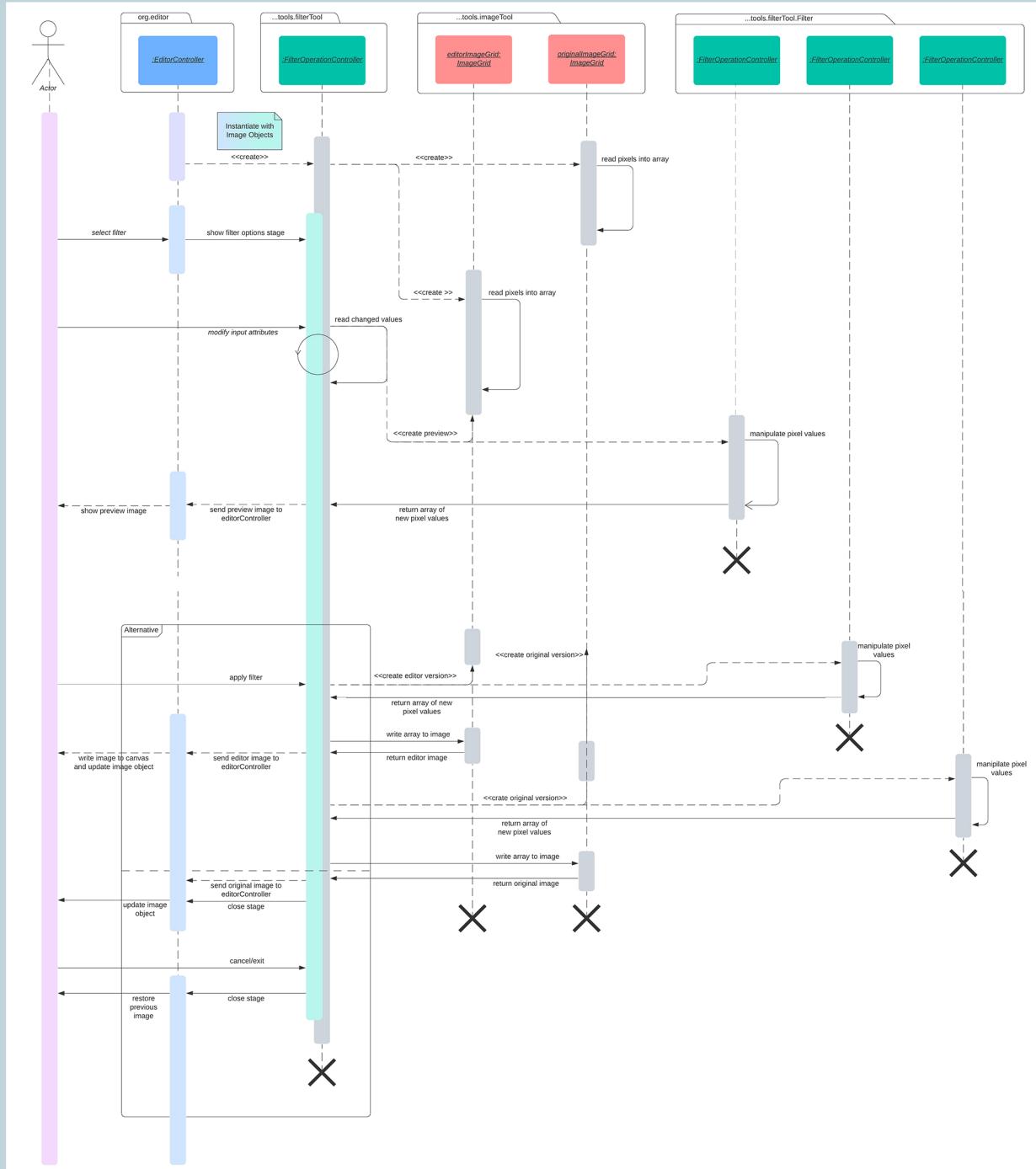
UML Diagramme

Class Diagram



UML Diagramme

Interaction Sequence Diagram



Stellungnahme

Architektur

Interfaces und Factory:

1. ControlScreen wird implementiert von LogInController, ModeSelectionController, SettingsController, GalleryController und EditorController.
2. DrawHandler wird implementiert von ArcDrawer, CircleDrawer, EllipsesDrawer, EraserDrawer, LineDrawer, PathDrawer, PolygonDrawer, RectangleDrawer, RoundedRectangle and TextDrawer es wird nur über die Factory HandlerFactory auf die verschiedenen Drawer zugegriffen.
3. Effect wird implementiert von BrightnessAdjustment, ContrastAdjustment, SaturationAdjustment, BlurEffect, DisplacementEffect und SepiaEffect.
4. Filter wird implementiert von GlitchFilter, GrayscaleFilter, InvertedFilter, NoiseFilter und Original.
5. HandlerFactory liefert Objekte, welche das DrawHandler interface implementiert haben. Als Parameter werden in die getHandler() Methode Enums der einzelnen Handler übergeben, welche die korrespondierenden dann zurückliefern.

Stellungnahme

Architektur

Abstract Classes:

1. DrawOP wird extended von der abstrakten Klasse Shapes und der Klasse Clear.
2. Shapes wird extended durch alle Klassen im Package /org/editor/tools/drawingtool/objects

(1) InvalidNameException beinhaltet unsere eigene checked Exception für die Vergabe des Projektnamens.

(2) Das Projekt ist nach dem Package-Konzept „Package by Feature“ geordnet, also jedes Package beinhaltet alle Klassen, die mit dem Feature zu tun haben.

Clean Code:

In keinen Kernklassen werden public member verwendet und statische Methoden waren nur in der GuiDriver: getIcon() notwendig. Und Referenzen werden meist kopiert, außer da wo es nötig ist direkte Referenzen zu bekommen, diese sind dann aber package only.

Exceptions:

Die checked Exception InvalidNameException (/org/editor/project/InvalidNameException) wird an Aufrufer zur Fehlerbehandlung weitergegeben (/org/editor/project/SettingsController)

Stellungnahme

GUI:

Die graphische Oberfläche wurde mit Hilfe von fxml-Dateien erstellt und mit css-Dateien gestyled. Beide lassen sich im resources Ordner unter /fxml und /styles finden. Die Controller sind jeweils in den verschiedenen Packages der Features unter gebracht.

Logging:

Allgemein werden bei uns Exceptions, welche abgefangen werden können, jedoch nicht repariert werden können, mit der Stufe error geloggt. Diejenigen die abgefangen werden können oder nicht sehr wichtig für den Programmverlauf sind, werden mit warn geloggt.

Zudem ist in jeder zentralen Klasse debug logging vorhanden, sowohl als auch info logging für den allgemeinen Programmverlauf.

Als Fatal werden nur schwerwiegende Exceptions geloggt, welche das Programm zum Abstürzen oder nicht Starten bringen.

Es werden in speziellen Files für die einzelnen Funktionen und in einer File für die Gesamtheit geloggt.

Stellungnahme

Threads

Das sequentielle Bearbeiten der Bilder auf Pixelebene würde in einer single-threaded Applikation für gewaltigen Stau sorgen, gerade wenn man Image Objekte behandelt, deren Pixelanzahl in die hohen Millionen geht.

In der Klasse FilterOperation werden die Pixel-Arrays zunächst in Blöcke gevierteilt, für die jeweils eigene Threads erstellt werden. Die Basiseinstellung von vier Blöcken kann dynamisch erhöht werden, wie zum Beispiel beim Checkerboard Effekt, bei dem durch “+” und “-“ Buttons die Anzahl der Partitionen verdoppelt, beziehungsweise halbiert werden kann. Pro Reihe herrscht jeweils ein ExecutorService über einen fixedThreadPool der Menge der angegebenen Blockzahl. Des Weiteren findet innerhalb jedes Blocks eine weitere durch jeweils eigene ExecutorServices gemanagete Verteilung in Panels statt. Diese Delegierung von grob nach fein lässt die Bearbeitung auf Pixelebene reibungslos stattfinden und die Vielzahl an Threads wird nach erledigter Arbeit von unten nach oben eingeholt. Durch den Einsatz CountDownLatches wird geprüft, wann Threads ihre Arbeit erledigt haben, beziehungsweise geregelt, wann andere Threads/Tasks ihre Arbeit starten dürfen.

Ein Hintergrund Thread wird im GalleryController ausgeführt, der Bilder aus dem ./edits4creditsgallery Ordner lädt. Die Galerie kann nur geöffnet werden, wenn alle Bilder geladen sind oder er zumindest das ganze Directory durchsucht hat (wird im LogInController durch die init() methode des GalleryControllers geladen).

Stellungnahme

Streams und Lambda-Funktionen:

Lambda-Funktionen werden überall genutzt, wo es möglich ist (z.B. bei Erstellung von EventHandlern, dem Delegieren von Methodenaufrufen bei Animationen, oder bei der Thread Erstellung).

Parallel Streams sind in der Klasse ImportControl vorhanden. In der Methode getFinalImage(...) werden die Pixel der unterschiedlichen Layers des finalen Bilds (Hintergrund, Imagelayer, DrawLayer) parallel mit dem SRC_OVER Algorithmus zusammengeführt.

Dokumentation und Testfälle:

Alle wichtigen Methoden sind kommentiert und es befinden sich Tests und negativ Tests zu den meisten zentralen Klassen im test directory. Da die meisten Methoden bei uns auf Basis der GUI laufen und die Komplexität des testen von GUI Elementen relativ hoch ist, wurde uns auf Nachfrage gesagt, dass wir diese nicht testen brauchen, somit befinden sich nur Methoden von nicht GUI-Elementen in den Tests.

Vorname	Nachname	Kürzel	Matrikelnummer	Projekt
Tevin	Zielke	tz024	42450	Edits4Credits
Johannes	Rödel	jr125	41959	Edits4Credits
Yannick	Kebbe	yk025	42360	Edits4Credits

Arc.	Clean Code	Doku	Tests	GUI	Logging/Except.	UML	Threads	Streams	Profiling	Summe - Projekt	Kommentar	Projekt-Note
3	3	3	3	3		3	3	3	3	30,00		1,00
3	3	3	3	3		3	3	3	3	30,00		1,00
3	3	3	3	3		3	3	3	3	30,00		1,00