Implementation Programming language: JAVA

Project Structure:

1. GUI Class:

يحوي كلاس الواجهة على واجهة البرنامج والتي تم استخدام JavaFx لبنائها والربط بينها وبين كلاس ال image processing وتتكون الواجهة من عدة ازرار و color pickerعلى اليسار و Javafx canvas ليتم عرض الصورة الرمادية عليه وتلوينها والتي كما وجدنا افضل طريقة للقيام بال image processing

2. Image processing Class:

يحوي هذا الكلاس على التوابع والخوارزمية التي تساعد على معالجة الصورة الرمادية وتحويلها الى ملونة .

Attributes	Functions
Int width عرض الصورة	Constructer() للقيام بعمل initialize لجميع ال attributes الخاصة بالكلاس.
Int height طول الصورة	RGB2bits() لتحويل قيمة ال RGB للبكسل الى bits والذي يساعد في مقارنة الحساسية .
2D array of pixels imagepixel يحوي جميع بكسلات الصورة.	pixelValid() والذي يتحقق من كون البكسل في الموقع x,y ليس خارج حدود مصفوفة الصورة ويحقق التدرج اللوني الذي يتحدد بقيمة الحساسية
2D array of pairs(x,y) pointsave يحوي احداثيات البكسلات التي سيتم تلوينها والتي وافقت الشروط .	DrawProcess() والذي يحوي خوارزمية التلوين والتي تتبع خوازرمية flood fill ولكن من اجل نقطة واحدة محددة.
Stack of pairs(pairs(x,y),color) undo لتخزين العمليات التي تتم على الصورة	ProcessImage() يقوم بمعالجة الصورة كاملة ويطبق التابع السابق على حسب عدد الخطوط (النقاط المتتالية) المأخوذة من الواجهة الرسومية.
Int sensitivity لتحديد حساسية اللون.	UndoStep() يقوم بالتراجع عن الخطوة السابقة .
	RedoStep() يقوم باعادة التراجع الذي تم اذا ما تم استدعاء التابع السابق.

Solution Approach:

1. تطبيق خوارزمية flood fill على بكسل واحد :

Parameters:

X coordinate ,Y coordinate , wanted color , visited array[][]

Algorithm: Drawprocess()

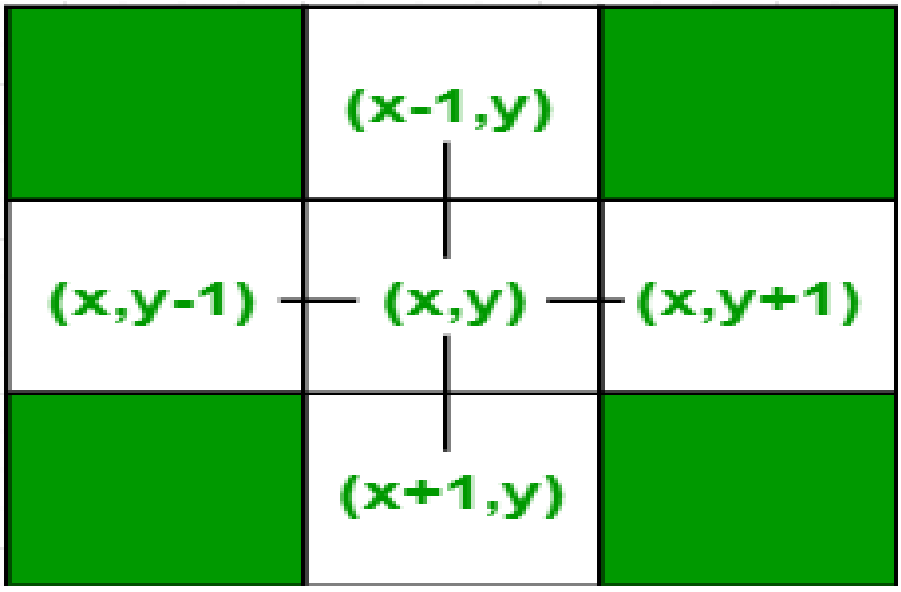
هناك طريقتين لتنفيذ الخوارزمية (عودية ,تكرارية) تم تنفيذ الطريقة التكرارية عن طريق ال BFS لكونها اسرع وتستخدم عمليات حسابية اقل وهي كالتالي:

1) قم بانشاء queue وادخل لها الاحداثيات ك اول عنصر.

2) طالما ال queue ليس فارغ :

- قم بعمل pull من الصف احتفظ بقيمة البكسل بالمتغير previous غير لونه بلون البارميتر brush
- تحقق من جيران هذا البكسل اذا كانو يحققون الشروط لكي يتم تلوينهم ايضا عن طريق التابع pixelvalid(x,y,previous,visited) والذي يتحقق من وجود البكسل الجار داخل حدود الصورة ويحقق شرط التدرج اللوني عن طريق مقارنته بمقدار الحساسية المطلوب.
- عند تحقق الشرط عند اي جار قم بادخاله لل queue وادخال الاحداثيات الى مصفوفة ال visited (لتحسين وتسريع الحل).

يتم تحديد جيران البكسل عن طريق الصورة ادناه :



2. تطبيق خوارزمية flood fill على عدة بكسلات :

نستخدم تابع imageProcess والذي يطبق الخوارزمية السابقة لكل نقطة محددة من الواجهة الرسومية واللون المحدد لها ويتم تحديد النقاط عن طريق حفظ جميع احداثيات النقاط التي يرسم عليها المستخدم في الواجهة ويتم حفظها في المصفوفة pointsave والذي يتكون من Array(array(pair<x,y>)) وكل عنصر بالمصفوفة الخارجية تحوي جميع نقاط الخط المرسوم ويتم ربط لون هذه النقاط مع الخط المناسب عن طريق المصفوفة colorsave والذي يعبر كل انديكس منها على لون الخط والذي يترجم الى جميع نقاط هذا الخط.

Results:

تم انجاز ما يلي :

1. امكانية ادخال صورة رمادية وعرضها.
2. امكانية رسم خطوط ملونة فوق الصورة.
3. حفظ الصورة بعد التلوين على القرص الصلب.
4. امكانية تلوين صورة بابعاد كبيرة بزمن مقبول.
5. اضافة ,تعديل خطوط جديدة واعادة تنفيذ التلوين مباشرة.
6. التراجع عن بعض الخطوط التي تم اضافتها
7. تنفيذ عملية التلوين مباشرة عند التعديل.

الواجهة: