

SE 3XA3: Development Plan

Group #10

Lab: L03

Aamina Hussain, hussaa54

Jessica Dawson, dawsor1

Fady Morcos, morcof2

Table 1: Revision History

Date	Developer(s)	Change
Feb 4, 2022	Jessica Dawson, Aamina Hussain, and Fady Morcos	First version
Apr 11, 2022	Aamina Hussain	Updated Section 6, Added Section 9 Project Review

This document contains the development plan for recreating a random abstract art generator. This plan will ensure the developers communicate properly, are consistent, and follow the project schedule.

1 Team Meeting Plan

The team shall meet twice a week during labs:

- Mondays, 9:30 am to 11:30 am
- Wednesdays, 9:30 am to 11:30 am

The team shall also meet at least once a week outside of labs. The location, day, time, and duration of this meeting will be decided weekly based on convenience and necessity.

2 Team Communication Plan

The team will communicate using the Microsoft Teams chat. This chat will be used to schedule weekly meetings and any further discussions that were not finalized during the in-person labs.

3 Team Member Roles

The team member roles are:

- Developer: Aamina Hussain, Fady Morcos, Jessica Dawson
- Scribe (records/transcribes what occurs during meetings): Aamina Hussain
- Communicator (asks the supervisors (the TAs) questions that the team has): Fady Morcos
- Submitter (makes sure documents are finalized and submitted before the deadline): Jessica Dawson

4 Git Workflow Plan

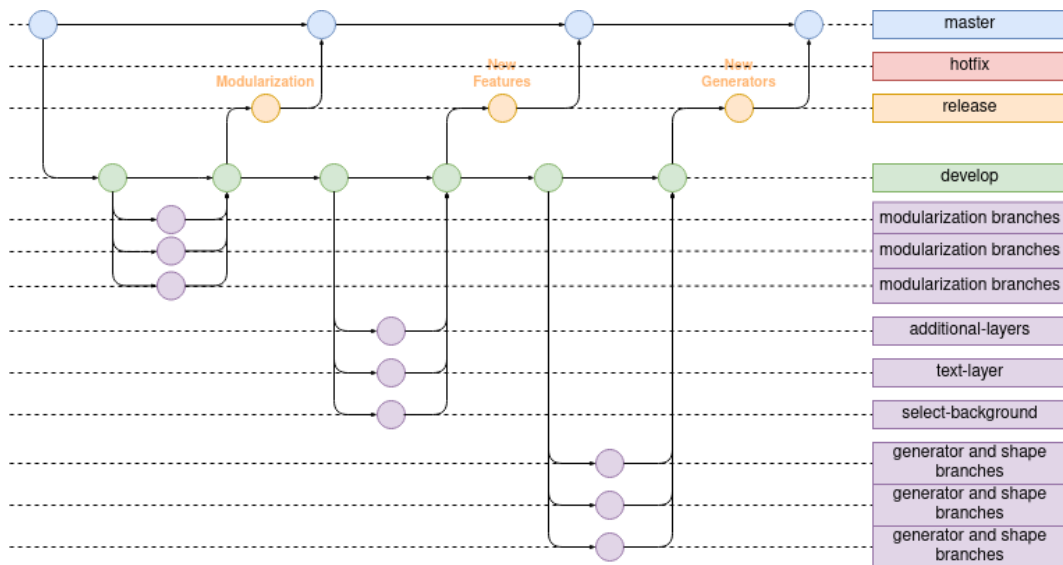


Figure 1: git workflow

Git Branches:

- master: for production ready releases, each release will be tagged with a version number
- develop: main development branch, for unstable nightly commits
- release: for cleanup before a release
- hotfix: for fixing any bugs found in a production release
- modularization branches: a set of feature branches for modularizing the code architecture, what these will look like will depend on a more indepth analysis of the code and what needs to be reworked
- addition-layers: feature branch for adding extra drawing layers to the program
- text-layer: feature branch for adding a layer that writes text on top of the art
- select-background: feature branch for allowing the user to pick the background color
- generator and shape branches: a set of feature branches each one adding a new generator or shape, what these will look like will depend on what new generators and shapes we decide to add

We will use a gitlab issue board to keep track of tasks with an open list for unassigned tasks, a board for each team member for their assigned tasks, and a closed board for finished tasks. Labels will be used to mark these tasks and associated merge requests.

Task Labels:

- bug: for bugfixes
- enhancement: for reworking code without adding features
- feature: for adding a feature to the code
- available: for an unassigned issue
- todo: for an assigned issue that has yet to be started
- in-progress: for an issue currently being worked on
- merge-ready: for an issue ready to be merged
- completed: for a fully merged and completed issue

Milestones will be used to track the completion of and group issues related to different section of our workflow plan:

- modularization: for tracking the completion of the first set of tasks related to the modularization of the code
- additional layers: for tracking the development of the additional layers features
- text layer: for tracking the development of the text layer features
- select background: for tracking the development of the select background color feature
- generators and shapes: for tracking the development of new generators and shapes

5 Proof of Concept Demonstration Plan

None of the required python library are difficult to install for this project. All of the modules can easily be installed from a command line using pip install quickly. However, this leads to portability being a bit concerning since the user will have to install these libraries in order to run the program. The proof of concept demonstration will consist of us running the Python program where we modularize the original code and add at least one new feature. This will create an interface where the user can choose different options for the abstract art they want generated. This demo will prove that we are capable of creating the core or bulk of the project.

6 Technology

The program will be written using the Python programming language. One of the main Python libraries we will be using is Pygame. The code will be written using an IDE called PyCharm. ~~The team will use the testing framework Pytest to test our program.~~ Latex will be used for generating most of the documentation. The MIS document will be generated using Doxygen.

~~##COMMENT: We realized that it would not be logical to test our code using an automated or unit testing framework due to the nature of the program. This program can only be tested by actually viewing the generated output, so for this reason we decided to forgo automated testing and use a manual testing approach to test our program.~~

7 Coding Style

The Google python style guide will be used: <https://google.github.io/styleguide/pyguide.html>

8 Project Schedule

Please refer to [this link](#).

9 Project Review

After the Revision 1 Demonstration, we are considering this project to be successfully completed. Despite the short period of time we had for this project, we completed all the objectives we had set at the start. This includes all objectives we had pertaining to the project's documentation, as well as objectives for the program code.

The success of the project can be attributed to the planning and constant communication of the team members. This allowed each member to be aware of their exact tasks and when those tasks had to be completed. We planned for lengthy tasks to be completed ahead of time in case those tasks ended up taking longer to complete than planned. We were also fully open to meeting outside of our usual meet-up times if necessary.

An area of improvement for our team is using git to its full extent. There was a bit of a learning curve for us when it came to using git, and instead of each working on separate branches, we usually messaged and let everyone know when we were pushing an update to the main branch. The workflow would have been smoother if we had full knowledge of the functionalities of git and actually used them.