Design Document

Version X.Y – yyyy.mm.dd

Created yyyy.mm.dd

<u>Project Name</u>

Name 1

Name 2

…

Name N

SECTION 1: PROJECT IDENTIFICATION

This section should include a concise statement that address such questions as:

- Why are you doing this project (i.e. what is the motivation?)
- How will it enhance or add to functionality that already exists?

SECTION 2: USER STORIES

User Stories help translate your users' needs into requirements. To create a list of user stories, start by considering your users' needs. Each user story will translate to a development task that will be owned by a member of your development team. As you write your user stories, keep in mind you that the user who owns the story will need to create not only an implementation that satisfies the story but also the associated tests.

Some examples of user stories are below.

Name	ID	Owner	Description	Implementation Details	Priority	Effort
Save File	1.1	Jane	As a user, I want a UI that allows me to save my progress so that I can continue where I left off when I return to use the software	Create a UI so that users can save progress in whatever folder they would like to on their hard drive.	1	2
Undo/ Redo	1.2	Fred	As a user, I want options to undo and redo my moves so that I can fix mistakes easily if and when I make them	Create an Iterable data structure to store historical moves and actions so these can be easily undone	1	1

Note that you may also want include some user stories that refactor existing code in order to make it more efficient or usable in some way. To include refactoring items in your list, you can use the following syntax:

"As a developer, I want … so that …. "

An example might be:

> *"As a developer, I want to change the process by which my Tree Objects are indexed so that the application will load 2-3 times more quickly."*

Note also that you may include as many user stories as you like. We will expect you to complete all of those that you mark as high priority (i.e. with a number 1) and we expect each team member to have a suitable amount of work to complete (10-15 hours worth). But if stories with lower priorities remain once the term is complete, that's ok; you can always continue to iterate on your product as long as it interests you.

SECTION 3: SOFTWARE DESIGN

This section should describe the design patterns you will use to realize your user stories. For each design pattern, provide a draft of UML diagrams to show how you'll implement the pattern as well as a written description that details how it will be used in the context of your user stories. An example of a pattern description is provided below. Note that you can adjust these patterns as you progress with your implementation, as this is necessary.

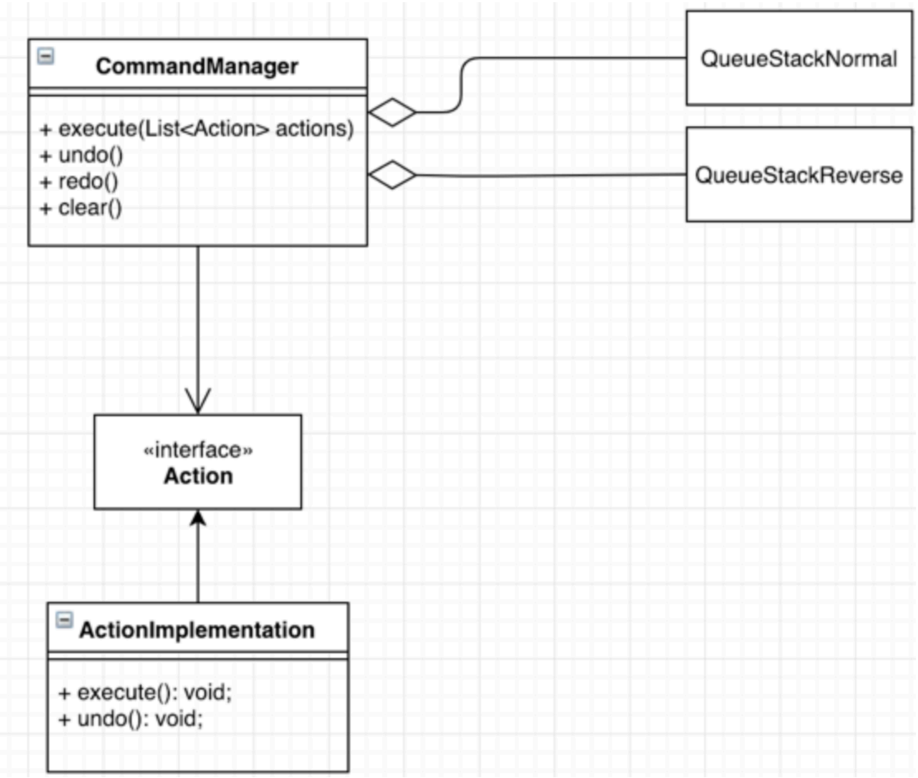
Design Pattern #1: Command Pattern

Overview: This pattern will be used to implement User Story 1.2 (Undo/Redo).

UML Diagram:

Implementation Details: The UML diagram outlines these main components:

- The *Action* interface, which includes two methods: execute and undo.
- Two modified linked lists ('*QueueStacks*'). These will push like a stack and pop like a queue.
- The *CommandManager*, which will execute actions and perform undo/redo.



Architecture

The *execute* method of the *CommandManager* requires an array of software actions, which are Objects. Every executed action will be registered in the *QueueStackNormal*. When performing undo, the action will be popped from this data structure, the undo method will be called and then the action will be pushed to *QueueStackReverse*. The opposite will happen when executing the redo operation. The *clear* method will be used to clear all registered actions in any of the *QueueStacks*.

SECTION 4: EXPECTED OVERALL PROJECT TIMELINE

This section should address such questions as:

- What is the current team's assessment of the project timeline?
- What are the major milestones?
- The most common tool for project planning in industry is the Gantt Chart
  - Gantt Chart Template: https://asq.org/quality-resources/gantt-chart