Final Fantasy XIV (Patch 5.1) Crafting Simulator
Qihao Gu (qihaogu2) Moderator: Max Qian

1. Abstract

1.1. Project Purpose

A tool to simulate crafting in Final Fantasy XIV (Patch 5.1).

1.2. Background/Motivation

I'm a player of Final Fantasy XIV (an online game). The game has an interesting crafting system which various projects have been developed to simulate. Most notably are:

https://ffxivteamcraft.com/simulator/custom

https://ffxiv-beta.lokyst.net/#/simulator

In the latest patch (5.1), major changes have been applied to crafting system in game (mainly the add /removal / remake of various crafting actions), so all major crafting simulators need to update accordingly. I want to build a crafting simulator that covers upcoming changes to

1. Learn how the simulator works, so that one day I can contribute to open sourced crafting simulator projects like teamcraft.

2. Learn how to interact with the game data (there's a popular api https://xivapi.com/ developed by players, and a site of game data discovered via data mining https://github.com/xivapi/ffxiv-datamining
) so I might build other projects besides the crafting simulator.

The theory support of my simulator mainly comes from the following post, which explains the backend formula of crafting (it is written in Chinese):
https://nga.178.com/read.php?tid=18839082

2. Technical Specifications

2.1. **Platform:** Website / Java application

2.2. **Programming Languages:** Java 8

2.3. **Stylistic Conventions:**

2.4. **SDK:**

2.5. **IDE:** Intellij

2.6. **Tools/Interfaces:** Maven, OpenCSV, Java Server Page

2.7. **Target Audience:** myself, other players

3. Functional Specifications

3.1. Features

- User can search for all crafting recipes in the game, including their difficulty, required raw materials, icon and method of acquiring.

- User can perform (a sequence of) any crafting actions and see the result in the simulator. The simulated result should accurately reflect in game result. User can undo previous action(s) conveniently.

- User can make macro (i.e. a fixed sequence of crafting actions) in the simulator and test the success rate of the macro by running it many times.

- User can customize crafter related attributes, apply consumables (food / potions), apply high quality raw materials at the start of the simulation.

- The simulator should be displayed nicely on a website, and user can remotely connect to the website to use it.

3.2. Scope of project

- The mechanism behind some of the crafting actions might remain unknown throughout the development of the simulator, so in some cases the simulated result might not match in game result.

- I don't have much experience developing a website. I think displaying the simulator on a webpage should be viable (i.e. deploy the GUI to the webpage), but I don't really know how much effort is required to host the website so that remote connection is possible.

- The project is more or less similar to the chess game developed in first assignment: write a "game" composed of some rules in the backend and connect it to a GUI. Some major difference are: 1. the simulator project requires obtaining data from other party (through api or csv file), whereas the chess assignment is self-contained. 2. The simulator project requires much more documentation (potentially a need to translate the source essay which explained the mechanism) as it is less intuitively constructed, compared with chess. 3. The simulator should be deployed to a website and (hopefully) be visited remotely.

4. Timeline:

4.1. Week 1

Simulator Logic

- Learn how other simulators work (done. The basic mechanism has been implemented, now we can implement actions)

- Learn to use https://xivapi.com/ to retrieve data at runtime (the website is less convenient than expected, so most of the data retrieval is done by using the csv file provided on the data mining github: https://github.com/xivapi/ffxiv-datamining OpenCSV helped a lot in processing the data. The api will be used more in web development later.)

- Describe crafting actions programmatically (work in progress, will need to delay the expected finish date to next week; there is a lot more work than expected, mostly because of how actions interact with each other. Total amount of work is expected to be similar to chess assignment. Also need to think of a better design pattern: each action works like a function to the crafting process (i.e. it describes changes to the process), but many actions share similar parameters / methods that I tried to implement them in an object orient way. Right now, the actions are implemented as objects, and are weird to use (I have to create a new action object for every step of action, with no reuse). Might look into functional programming for this part.)

- Describe crafting recipes programmatically (done! Necessary data has been extracted as expected. Can add raw material list extraction for extra feature, which is quite trivial)

- Describe the crafting process (including random condition) programmatically (work in progress, need more testing as well)

- Simulate crafting process with implemented actions and recipes (work in progress, implement more actions first)

- User can redo / undo crafting actions (work in progress, implement more actions first, also should look back to the chess assignment to the pop stack thingie for this)

- User can create macro to record a fixed sequence of crafting actions (work in progress, this part should be trivial once all actions are implemented)

4.2. Week 2

Simulator Logic

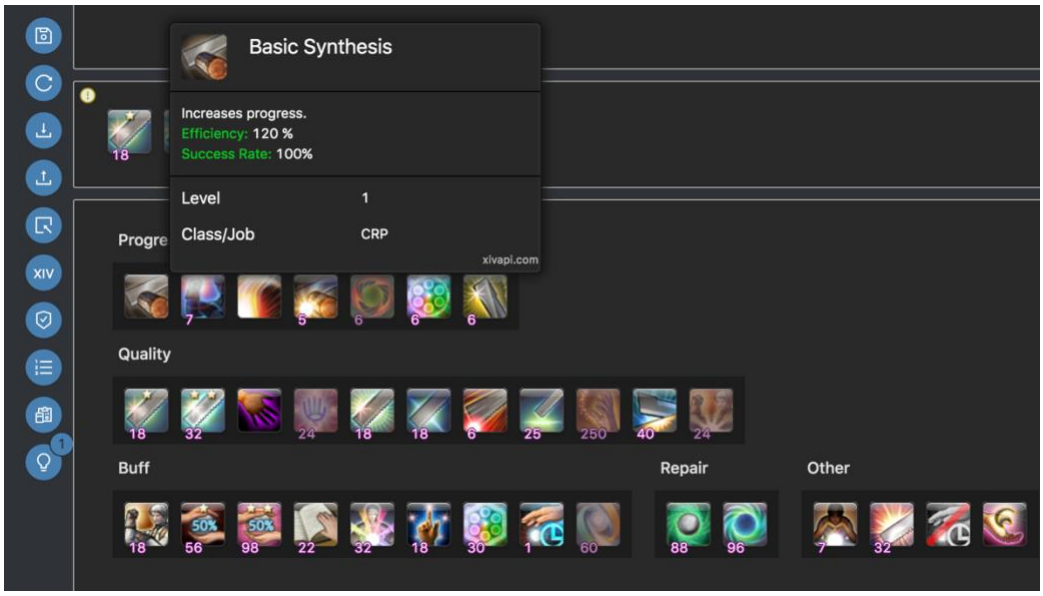
- Finish what has been left from week 1 (requires non trivial time so do it early)

Web application (Java Server Page)

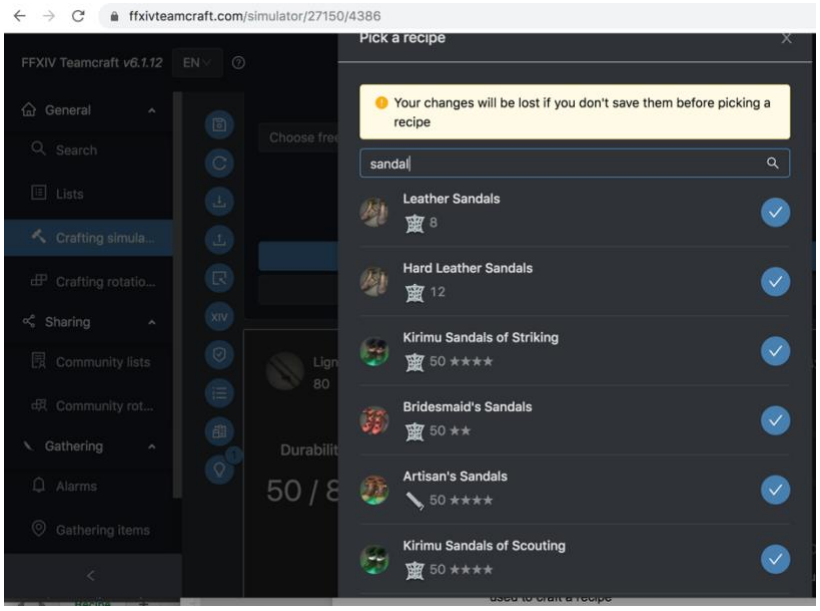
- Use JSP to build a static website for the simulator and deploy online. Example simulator websites are:
  https://ffxivteamcraft.com/simulator/27150/4386
  https://ffxiv-beta.lokyst.net/#/simulator

- Icons related to actions / items / recipes should be displayed on the website

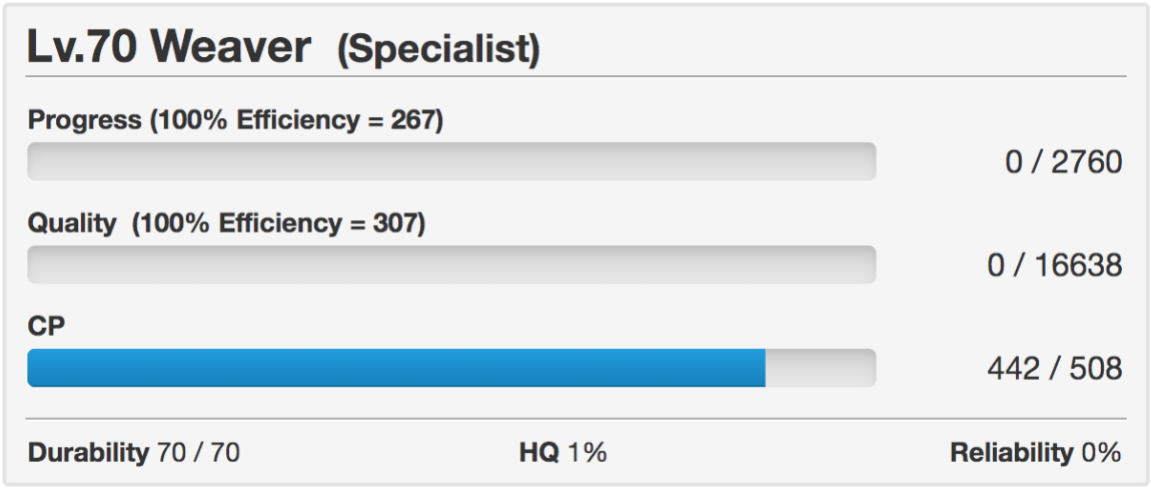
- Moving mouse over icon should display tooltip with related information



- Search bar for recipe: user can type in key word and find all recipes with names containing the keyword. A drop down menu displaying potential search results should load, like



- Have visuals (i.e. bars) to present the crafting process! Like this:



- Click an action button to perform the action, click it again from the list of action history to undo it.
- If somehow managed to get all the stuffs done, look into how to remote connection.

4.3. Week 3

Web application

- Finish what's left over
- Improve the look of simulator
- Add functionality of searching for item (not just items with recipe), display retrieved information on a custom webpage (or as a last resort redirect to the database page with item introduction)
- Add functionality of displaying the raw material list required for a recipe. Clicking the item icons on the raw material list should open up / redirect to a page with related information
- Add functionality of applying consumables (food / potion) to the crafting process. This requires implementation both in the back end and the frontend.
- Make remote connection possible, if it's too hard or unfeasible go read the source code of other simulators and see what can be learnt.
- Try to implement an algorithm that solves a recipe with given crafter attributes. This feature has not been fully implemented even in major simulator projects. Look into some popular solution for patterns.
- Add functionality of finding the minimum crafter attribute requirement to complete a target recipe with a fixed sequence of crafting actions.

5. Future Enhancements

6. Functional Rubric

Week 1 Functional Requirements		
Requirement – Describe crafting actions	4	1.5: An Action class that provides implementation of all crafting actions in game. 2: Provides additional information for all crafting actions besides their mechanic: icon, level of acquiring.
Requirement – Describe recipe	3	1: A Recipe class that provides necessary information for all crafting recipes in game: recipe name, recipe difficulty. 1.5: Provides additional information for all crafting recipes: icon, material list, class restriction, level of acquiring.
Requirement – Describe crafting process	5	2: A Process class that describes a "crafting process" in game: given specified crafter attributes and target recipe, simulate the result of performing a sequence of crafting actions. 2.5: Provide a simple GUI to show the crafting process.
Requirement – Data retrieval	3	1.5: Retrieve actual game data using https://xivapi.com/
Testing Requirements		
Testing – Unit Tests	5	0: no test 1.5: 80% coverage 2.5: 95% coverage
Testing – Manual Tests	5	0: no test 2.5: take some in game screenshot to show what crafting is; compare data in the game's crafting UI to the data you computed to show your calculation is correct.
Week 2 Functional Requirements		
Requirement – Undo	3	1.5: Crafting actions can be undone in a crafting process, so the crafting process can roll back to the state before target action(s) were applied.

Requirement – GUI on webpage	5	2.5: A webpage that displays major components of the simulator. User can interact with the webpage to use the simulator. See the sample simulator websites for inspiration.
Requirement – Search (recipes)	4	1.5: A search functionality for user to search for recipes to load into simulator. 2: Search results are displayed nicely
Requirement – Icons and Tooltips	3	1.5: Crafting actions are displayed with icon and tooltips on the website.
Testing Requirements		
Testing – Unit Tests	5	0: no test 1.5: 80% coverage 2.5: 95% coverage
Testing – Manual Tests	5	0: no test 2.5: Take many screenshots of your cool looking website to show all the functionality you added
Week 3 Functional Requirements		
Requirement – Search (items)	4	1.5: A search functionality for user to search for items in game 2: Search results are displayed nicely
Requirement – Raw Material List	3	1.5: The raw materials required in a recipe should be displayed on the website
Requirement – Crafting Solver	5	2.5: Implement an algorithm that finds a viable sequence of crafting actions that "solves" a recipe. Note that even major simulator project like teamcraft doesn't have this feature fully implemented.

Requirement – Find Minimum Crafter Attributes	3	1.5: Given a sequence of crafting actions and recipe, find the minimum crafter attribute that makes the sequence of actions workable.
Testing Requirements		
Testing – Unit Tests	5	0: no test 1.5: 80% coverage 2.5: 95% coverage
Testing - Manual	5	0: no test 2: Have screenshots for all the functionality you added 2.5: Live test remote connection if it's remotely implemented