Design Decisions – Milestone 4

Written by: Monica Ruttle, Reid Cain-Mondoux , Justin Fleming, and Noah Segal

This document provides detail on design decisions made on the SocialSim project, explaining the rational behind all design decision. The project follows following java beans convention, allowing our team members to save with the XML encoder and decoder. The project is currently separated into 5 main sections:

* Simulation
* Document
* User: Consumer & Producer
* Searching
* Main Window
* MainWindowSave / SimSave

# User: Consumer & Producer

The primary change made to the implementation of Consumer and Producer was the introduction of an abstract User class. The Consumer and Producer now inherit from the abstract User class. The User class has a list of followers, a list of Users they are following, the User’s payoff, their tag, ID, preferred search method, and the list of documents they like. It also contains shared methods to follow users and like documents (as well as common assessors & modifiers). When the Consumer and the Producer class inherit from User they implement two abstract methods responsible for: how they take their turn (takeTurn) and how they calculate their payoff (caluclatePayoff). This allows the Consumer to behave differently from the Producer. One example of such behaviour is the Producer’s payoff calculation. The Producer’s payoff is calculated by examining the top k documents. The Producer’s payoff is incremented after every search by the number of its own Documents returned by the User’s search method. This design change was discussed with Professor Babak Esfandiari.

# Document

The Document class represent a document. It includes data to be used as metrics for the simulation. The Document is a representation of real documents but only include relevant simulation data. It stores a list of who likes that document, and a method to add to this list.

# Simulation

The simulation currently keeps track of all documents, consumers and producers. Since consumer and producers inherit from User, they are contained in a single User list which makes selecting whose turn it is independent of User type. This was modified from Milestone 1, as before it contained a list of Consumers. It is the central hub of the emulation software. It currently keeps track of all documents and the simulation state. The user interaction is handled by a Main Window, which gets updated by the Simulation. The simulation handles starting the game and controlling whose turn it is. It also updates the users each time a turn is taken and passing that information to the Main Window.

# Searching

Searching interface is using the strategy design paradigm to use different methods of selecting documents to discover what searching method is the most beneficial to the consumer and the producer. The Consumer and the Producer store which search method they use, as it may differ per user. The search methods described in the Project Outline were also implemented.

# Main Window

The Main Window was implemented using the Grid Bag Constraints layout. This was chosen as it allowed for organization of all of the window’s components. The top of the window consists of five TextFields paired with five JLabels, which are used to allow a user to easily enter all of the required information to start a simulation. This information includes the number of Turns, Tags, Producers, Consumers, and Search Results. The Producers and Consumers are displayed within their own JTables. This allows each User’s information to be easily organized and displayed. One important design decision when displaying the Producers and Consumers was to only show the number of followers and number of Users they followed. A popup window will be added in Milestone 3, which will display the name and ID of the selected User’s following and followers list. The generated Documents are also displayed within their own JTable, which is located to the left of the Consumer and Producer JTables.

The simulation starts by pressing the “Start” button in the top-right corner. When it begins the Start button will change to a “Step” button, and the Turns field will decrement with each turn/step. Aside from the Number of Search Results, all TextFields in the top bar cannot be edited once the simulation begins. The number of Search Results can be edited before each turn.

When the Simulation starts, a bar graph of User Payoff will appear at the bottom of the window. The x-axis displays the User ID and the y-axis displays the payoff as an integer value. The graph updates after each turn. JCommon and JFreeChart libraries were to implement the graph.

# Double Click Window

A mouse listener was added to each table, which detects when a user double-clicks on an cell in one of the three tables (Documents, Producers, and Consumers). A different constructor is called depending on the table that the user clicks on (Document or User). Double-clicking on a Document shows a list of Users that like the selected Document. Double-clicking on a Producer or Consumer displays two lists (Followers and Following) as well as a drop-down menu used to change the User’s search method. In addition, the double-click window for Producers includes an option to change how the Producer acts (either as a Consumer or as a Producer) as well as a drop-down menu that is used to set the Producer’s alternate tag. In order to save altered settings, the user must select the “Ok” button. Pressing “Cancel” will ignore any changes to the User’s settings.

# MainWindowSave / SimSave

The purpose of the MainWindowSave and SimSave classes were to save the state of the simulation and properly restore it at a later time. We wanted to ensure that all components and listeners in the GUI were retained when saving. In addition, the Simulation has an instantiation of the MainWindow. The purpose of the SimSave final was saving, we did not want to save a second copy of MainWindow (and end up with two copies of the GUI). We ensure that we save only relevant information.

This allowed for us to easily Undo and Redo by using our Save/Load features. Saved turns are deleted at the end of the Simulation. However, saved simulations are not.