

Algorithm Visualization Final Report

Casey Davis, Johnathan Mell, Di Mu, Federico Nusymowicz

`{davisca, jmell, dimu, fnusy}@seas.upenn.edu`

May 4, 2012

1 Overview

The Android application developed in this project is an algorithm visualization application aimed at instructing young students interested in learning about computer science algorithms through an interactive graphical user interface presented as a competitive game. Specifically, the application focuses on the bin-packing problem (and the more specific knapsack problem), which involves optimally placing valued items of certain weights into a finite number of bins of limited capacity, maximizing the total value of the objects inside the bins. The user interface includes the ability to touch and drag objects into bins, notifying the user when the optimal solution has been found.

As we completed the final iteration of our project, we were able to make several improvements to the program, mainly centered around the GUI. By recalibrating the main screen and adjusting the font sizes, we were able to optimize the program for tablet use. Furthermore, a high score screen was added and level names were incorporated so users could see their progress as they completed levels.

2 Features

2.1 User stories

Below is a list of the user stories we have completed as of the end of the final iteration, along with the point values assigned to each.

1. As a user who selected a difficulty level, I should see a paginated panel that displays all corresponding objects (3 points).
2. As a user, I should be able to remove objects from bins one at a time using the paginator (2 points).
3. As a user, I should be able to move objects from bin to bin (1 point).
4. As a user, I should be confronted with tricky problems on harder difficulties, wherein greedy algorithms will not work (1 point).
5. As a user, I would like a metric of how optimal my solution is compared to the algorithm solution (1 point).
6. As a user, I should see a timer whenever I am solving a problem (1 point).

7. As a user, I should see my high score, as well as a high score screen (2 points).
8. As a user, I would prefer graphically larger sizes of bins and bin objects available on a tablet screen, which is larger (2 points).
9. As a user, I would prefer a cleaner user interface, particularly making the font size of text displayed on the bins and bin objects bigger, with different colors for objects, and a background that has depth (1 point).
10. As a user, I would pagination to be removed if there is only one page of objects (1 point).

Total points completed: 15

With four people working on the project for three weeks, we should have achieved at least a total point value of 12, if a point is defined as one person working for one week. Given that our point value was 15, this is more than our goal, with a **project velocity of 3.75 points/person**.

2.2 Unfinished user stories

Most of our work was centered around the GUI during the last iteration, though a few were focused on the high score screen. Planned user stories include:

- As a user, I would like to see the optimal solution displayed once I give up.
- As a user, I would like to see the objects look more 3D.

Improvements to the GUI were ongoing, and given more time, one-page pagination and 3D objects would likely have been achieved. The optimal solution we did not have time to display, though it is calculated using our optimization algorithm.

3 Known bugs, untested code, etc.

3.1 Bugs

We have found two significant bugs in the code during this iteration:

- Currently, if the high score is reset, sometimes an exception is thrown, causing the application to stop. This will be resolved in the final modifications to the code.
- When traversing the levels in reverse order (using the 'Previous' button), their names are replaced instead with numbers.

3.2 Untested code

Although most of the non-trivial, complex code of the project includes a complete test suite, much of the code involves output that is fully graphical, given that the application's primary function is to provide a GUI for users to learn about algorithms.

This part of the code includes files such as `BinPackingView.java` and `BinPackingActivity.java`, which are used to develop the GUI for our application. Although many of the simple calculations used to position objects on the screen can potentially be tested for proper values, ultimately the best tests for the methods in such classes are simply to observe the results displayed

on the screen of the device or emulator used for testing. As such, we did not write test cases for this code.

We did, however, include a full test suite for the underlying infrastructure upon which the GUI is built, including the Java classes representing bins and objects, and the output of the optimization algorithm for the bin-packing problem. Tests for the paginator (which displays objects on multiple pages when there are too many to fit on the screen) were also written.

4 Potential changes

4.1 Proposed additional features

There are some features that, given more time, we would like have seen implemented in the application. Ideally, we would like to expand the platform to include other educational algorithm tools, such as a max-flow graph. This was something we discussed with the client, but did not plan to complete due to our limited time frame.

Furthermore, we would like to add support for objects with continuous-valued weights for the bin-packing aspect of the problem. This would allow for more complicated problems with difficult solutions, though it would necessitate a refactoring of our optimization algorithm.

Finally, a more immediately useful feature would be a level-select screen and buttons that return the user to the main menu (although the latter can technically be accomplished with the back button on the majority of Android devices).

4.2 Developmental changes

If we were to restart the project, there are some changes to be made. For the most part, the initial GUI we set up worked adequately, and the functionality for the actual bin-packing was set up earlier. It would, however, have been beneficial to place more focus on planning the GUI earlier, so that perhaps we could have explored other options such as selecting options and tapping the bins, rather than using drag and drop (which drained a significant amount of time in implementing its functionality).

5 Final comments

5.1 Notes on the customer

Circumstances necessitated that Dr. Murphy work with us on behalf of the customer for the development of this application, the purpose of which was so to see how effective virtual tools and games were on educating burgeoning computer scientists. Working with him was a pleasure, with feedback being given at each stage, and flexible deadlines being established so as to ensure the group could deliver a quality product.

5.2 Copyright

We would like to release the code for this project to the customer, through Dr. Murphy, for non-profit use.

5.3 Final code modifications

Code modifications discussed for Homework 6 are ongoing, and will be sent to the TA when completed.