Algorithm Visualization First Iteration Report

Casey Davis, Johnathan Mell, Di Mu, Federico Nusymowicz

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

March 20, 2012

1 Overview

This project is an Android application that is used as a tool for students to interactively learn about algorithms in computer science. Specifically, this tool allows users to solve bin-packing and max-flow problems in the form of a game.

In this iteration, the bin-packing problem is implemented graphically in the form of a click-and-drag setup where users drag objects with certain weights and values into bins with a specified capacity. The goal of this game is to find the optimal way of placing the objects into bins such that their total value is maximized and the capacities of the bins are not exceeded.

The application allows users to choose a difficulty level, each of which includes different numbers of objects and bins, specified in an editable XML file. Users may drag objects on the screen (represented by blue squares) into the different bins (represented by red squares). When the optimal solution has been achieved the user is automatically notified.

Below we describe in more detail the different features of the application in the form of user stories, which were specified before the beginning of this iteration.

2 Features

2.1 User stories

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

1. As a user, I should be able to drag objects into bins (3 points).
2. As a user, I should be able to pick amongst easy, medium, and hard difficulty levels (1 point).
3. An an app manager, I should be able to configure objects and bins for different problem difficulties from within an XML file (3 points).
4. As a user, I should see a home screen with a title and difficulty selection buttons (2 points).
5. As a user, I should see bins corresponding to the difficulty I selected, along with their remaining capacity (1 point).
6. As a user, I should see each object's weight and value (1 point).

7. As a user, I should not be able to exceed a bin's capacity when dragging an object into it (1 point).
8. As a user, I should be able to remove objects from bins (1 point).
9. As a user, I should be notified when I solve the algorithm correctly (1 point).
10. As an app manager, I should be able to get the value of an approximate optimal solution for a given problem (2 points).

Total points completed: 16

There are no user stories that we planned to complete before this iteration demo that we have not completed.

2.2 Project Velocity

With four people working on this project for several weeks, and having completed a total of 16 points, our project velocity is calculated to be 4, which is above the target velocity we hoped to work at.

3 Testing, bugs, issues, etc.

Rather than computing the optimal solutions to the problems in the XML in advance, we implemented an algorithm for computing the optimal solution within the app, so that an app manager can change the configurations for the different difficulty levels.

Because we allow real-valued weights for the objects, rather than just integers, the bin-packing problem is NP-hard. Thus, it was necessary for us to round object weights and bin capacities to integers before calculating the optimal solutions. Therefore, the optimal solutions are only approximations to the actual optimal solutions. We could have implemented a brute-force solution, but the computation of the optimal solution would take $O((b+1)^n)$ time, where b is the number of bins and n is the number of objects. In configurations with large numbers of objects, this would significantly impact the running time of the app.

Other than this caveat, there are no other known bugs or issues with the application.