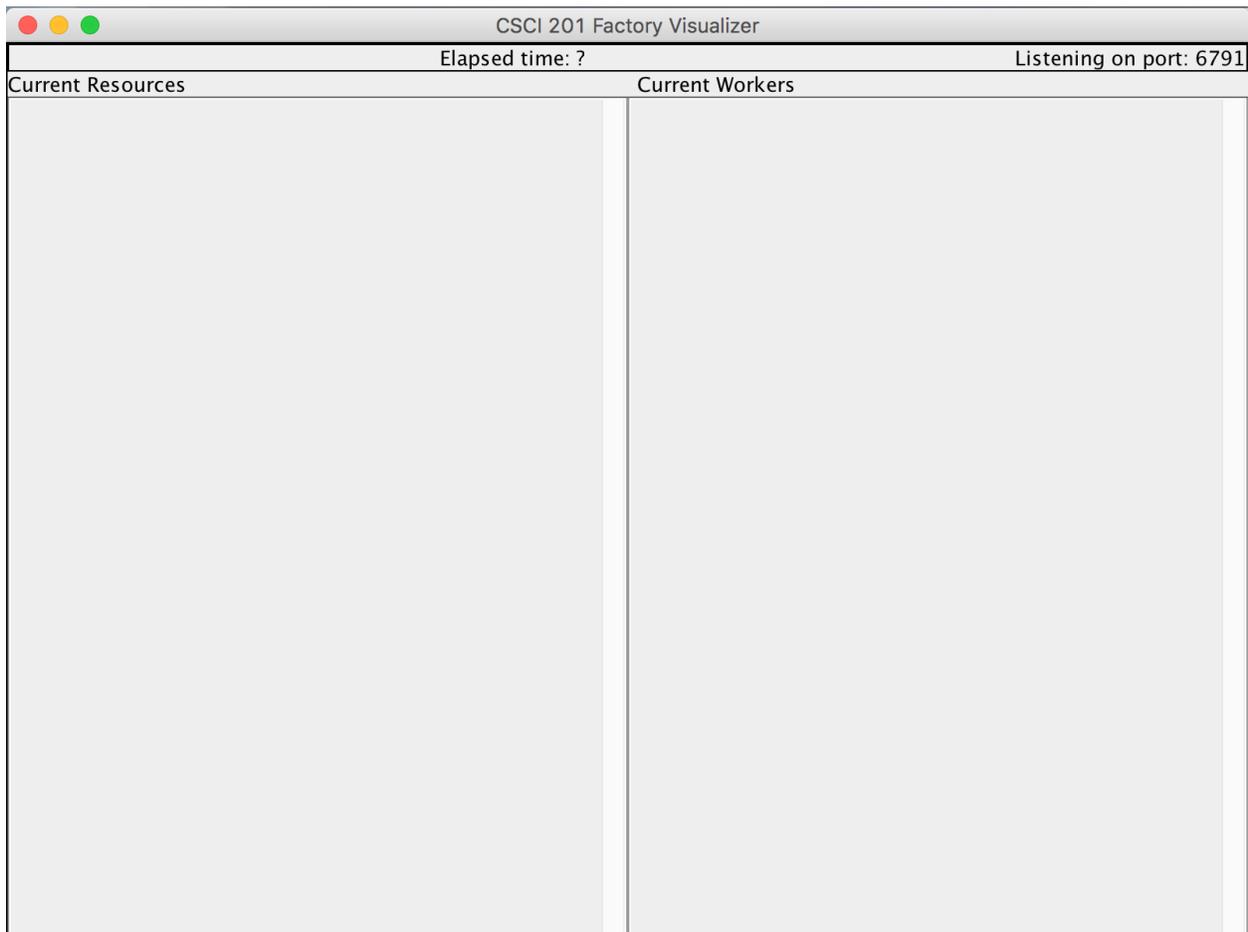


CSCI 201L - Programming Final
Summer 2016
10% of course grade

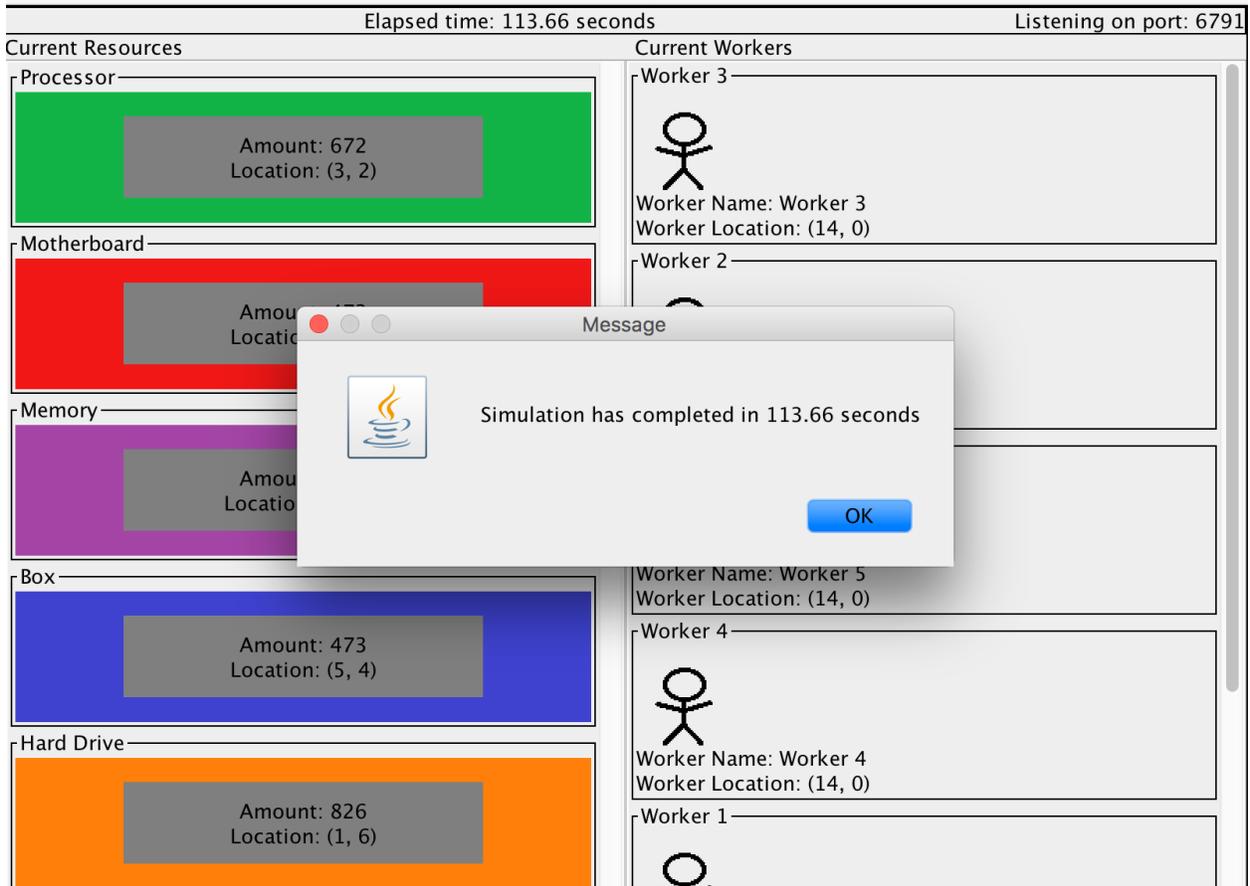
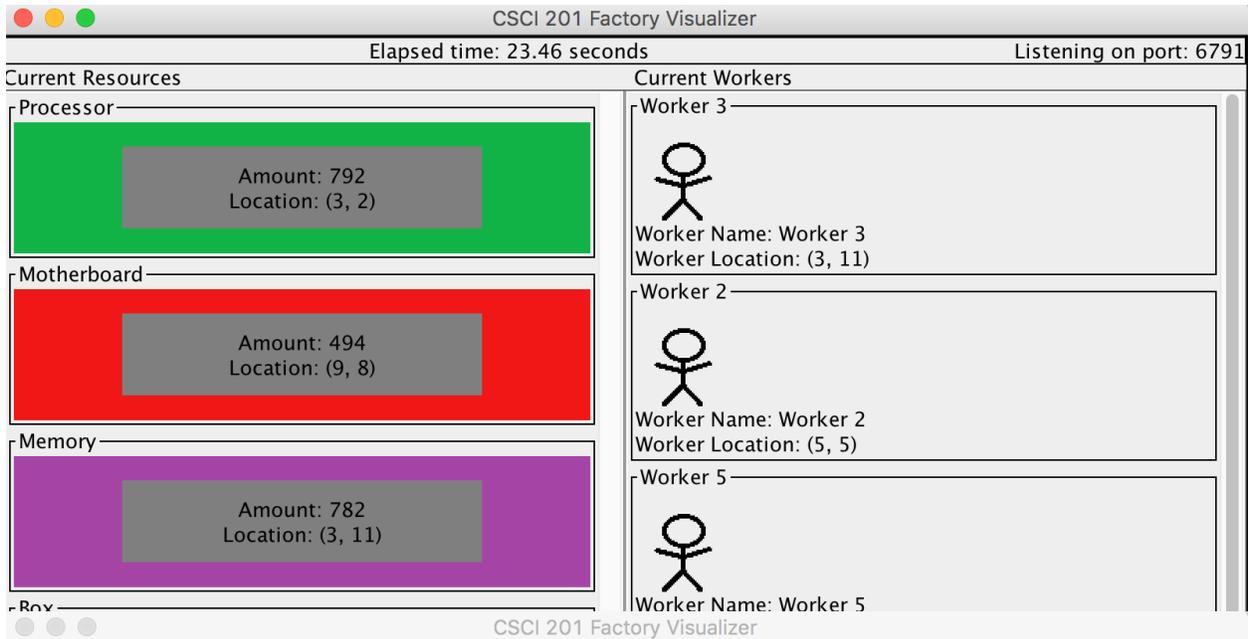
Using the factory framework code provided for this exam, you will implement a standalone visualizer for the factory simulation environment.

Factory Visualizer GUI

The visualization GUI you are to create will allow you to dynamically track the progress of simulations from a program separate from the main factory simulator client. The **factory client** will connect to this **factory visualizer** using sockets. The top row of labels in the factory visualizer GUI will display the **elapsed time** of the simulation and the **port** on which the visualizer is listening for connections from the factory client. The GUI will initially look like this when loaded before it receives information from the factory client about resources and workers in the simulation. All messages sent from the factory client to the factory visualizer will be in the form of **serialized java objects**.



Once the resources and workers have been received from the factory client, the GUI will look like the images below, with the **resources** on the left (with the image associated with each resource in the background and a dynamically updated amount), and the workers on the right (with dynamically updated worker locations and a static worker icon). Note that the name of the worker is the title of the titled border around each worker and the name of the resource is the title of the titled border around each resource. Once the simulation has completed a dialog must pop up and inform the user that the simulation has ended along with displaying the **total runtime of the simulation**.



Factory Visualizer

This visualizer will operate as a stand alone program to which the factory simulator will need to connect via a socket. We will provide you with a template server program which listens for an arbitrary number of connections on a single hardcoded port, 6791 and which contains a skeleton of the required GUI. You may change this port number if you like. The factory client will be sending frequent messages to update the current status of the simulation in the form of **serialized objects (resource amounts, elapsed time and whether or not the simulation has completed, and worker locations)**. When it receives these update messages from the factory client, it will update its visualizer GUI shown above. More specifically, these messages will contain updated information about the amount of each resource left, the current location of each worker, the total elapsed time of the simulation thus far, and whether or not the simulation has completed. When the simulation has completed, a dialog box must be shown to the user indicating that the simulation has concluded and displaying a total runtime of the simulation. At this point, the visualizer should stop updating.

Factory Simulator Changes

In order to complete this exam, you will need to modify the current simulator code to update the external factory visualizer every **5.0** seconds. The updates must take place on separate, designated threads spawned from the factory simulation client. One thread will be designated to update worker locations, one thread will be designated to update resource amounts and one thread will be designated to update the elapsed time of the simulation (and whether or not it has completed, for displaying the simulation complete message dialog). These 3 threads will initiate separate connections (using separate sockets) to the visualizer and send it an update message every 5.0 seconds. Note that, you can use the same set of threads throughout the execution of the simulation (you don't need to spawn new threads every 5 seconds), but, to receive full credit, you **MUST** implement the updating on separate threads. All three of these threads must be new threads. **You cannot use the main simulation thread to do any connecting to the visualizer or send any messages. All threads that communicate with the visualizer must be spawned for that purpose alone by extending the Thread class. All messages sent to the visualizer must be in the form of serialized objects which are de-serialized at the visualizer.** Two such serializable classes, Resource and Worker, are provided in the resource package. The visualizer is currently configured to use objects of these classes to expedite the dynamic GUI updating.

Hints

1. The FactorySimulator class is a good place to spawn these update threads, as it contains or has access to all the required information. You should be able to implement most required functionality (outside of classes you create) in this file.
2. Note that FactoryWorker is not serializable. It is recommended you use the provided Worker class to update the visualizer.
3. An acceptable and easy way to achieve updating the visualizer every 5.0 seconds is to put each update thread to sleep in between sending messages.
4. Please read through the grading criteria (below) very carefully as it more precisely explains what we will be looking for during grading.
5. You will need to implement a method for monitoring the completion of the simulation in the factory client and for determining elapsed simulation time. These functionalities were

created in later labs (this package is based on the initial factory code), so you must re-implement these changes or import them from your previous labs.

Grading Criteria

% of Final Grade	Criteria
Visualizer GUI: port number - 1.0%	Port number to which visualizer is attached is displayed as shown in GUI
Visualizer GUI: elapsed time - 1.0%	Elapsed time of factory simulation displayed as shown in GUI and updated dynamically every 5.0 seconds
Visualizer GUI: resources display - 0.5%	Resources displayed as shown in GUI with name of resource, image of resource, and location (x, y) of resource
Visualizer GUI: resources update - 0.5%	Resources updated dynamically throughout simulation to reflect current remaining amount every 5.0 seconds
Visualizer GUI: message response - 1.0%	The visualizer GUI accepts connections from the factory simulation client and reads in the objects sent to it using de-serialization
Visualizer GUI: workers display - 0.5%	Workers displayed as shown in GUI with name of worker, icon, and location of worker.
Visualizer GUI: workers update - 0.5%	Worker locations updated dynamically throughout simulation to reflect current remaining amount every 5.0 seconds
Visualizer GUI: end of simulation: 1.0%	Visualizer GUI correctly indicates that the simulation has ended using a message dialog that displays the total runtime of the simulation.
Visualizer GUI: on-the-fly GUI update - 1.0%	The visualizer GUI must update itself automatically without the user having to resize the window, click on it, etc.
Factory Simulation: connect to visualizer - 0.5%	Separate threads (3, one responsible for updating resources, one responsible for updating workers, and one responsible for updating elapsed time and whether the simulation is complete or not) are created and individually connect to the visualizer using a separate socket.
Factory Simulation: message timing - 0.5%	The 3 updater threads send messages every 5.0 seconds to the visualizer server using serialization of objects.
Factory simulation: worker messages - 0.5%	A thread is spawned to update worker locations by sending serialized objects to the visualizer
Factory simulation: resource messages - 0.5%	A thread is spawned to update resource locations by sending serialized objects to the visualizer
Factory simulation: simulation data messages - 0.5%	A thread is spawned to send a serialized object(s) to the visualizer that represent the current elapsed simulation time and whether or not the simulation has been completed.
Factory Simulation: message format - 0.5%	All messages sent to the factory visualizer are serialized objects