TEAM INFINITE TUPLES

http://www.cs.rit.edu/~bdi8241/adhoc/

Jacob Hays

Brad Israel

TEAM INFINITE TUPLES

Agenda

- Project Overview
- Review of design
- GUI
- Problem loading and solving logic
- Questions

TEAM INFINITE TUPLES

Project Overview

- Adhoc collaborative problem solving framework.
- Focus is on fractals.
- Allow users to create fractal problems and distribute workload to other users within the adhoc network

TEAM INFINITE TUPLES

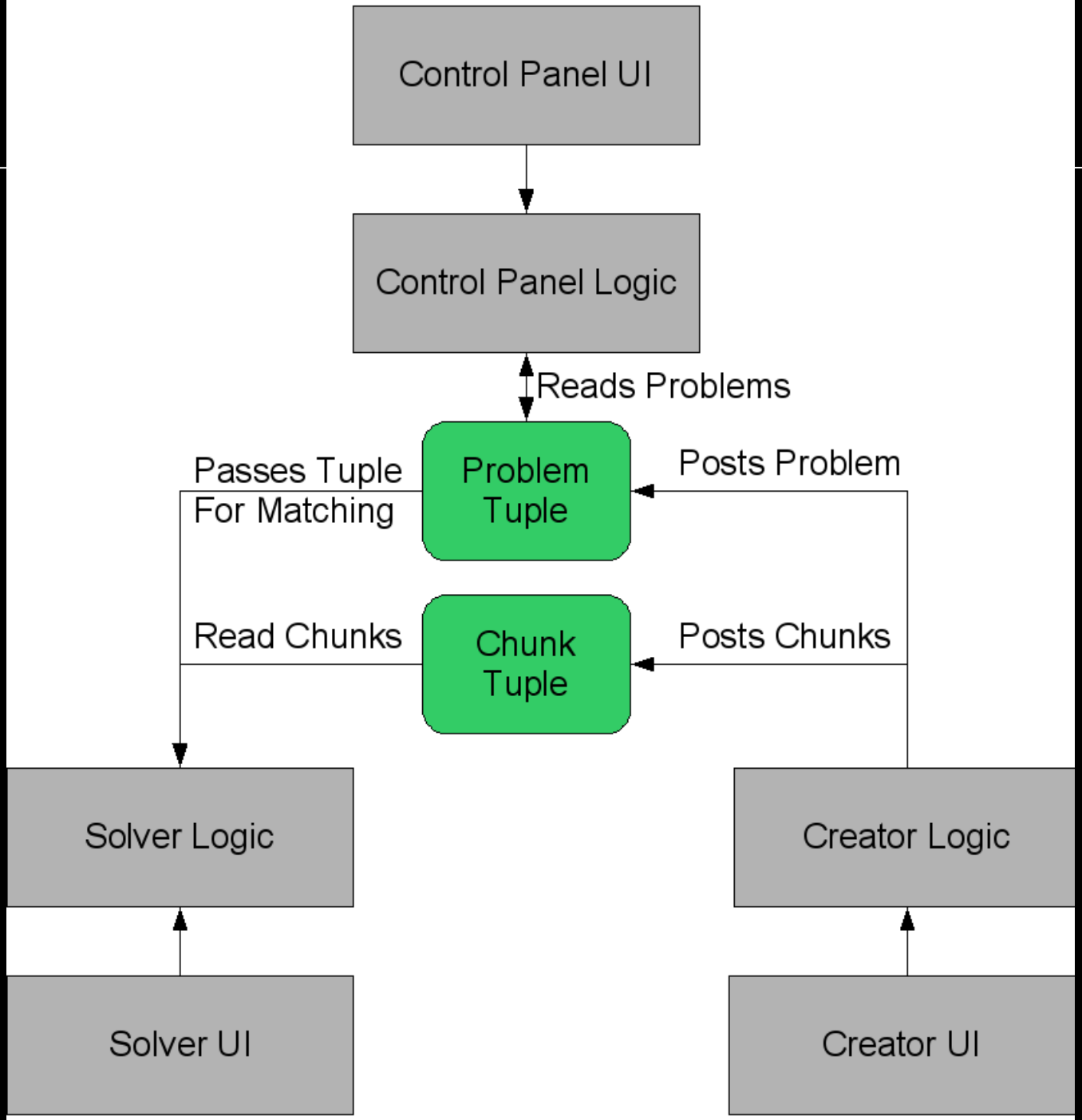
<u>Product Design Review:</u>

- A application that runs on multiple computers, that communicate together with Tupleboard.
- Problem Creator: Load compiled java classes that implement a fractal problem. User sets parameters.
- Problem Viewer: View completed and in progress problems. View performance statistics.
- Control Panel: Set which problems to work on, display statistics, etc.
- Problem Solver: Decides what chunks of problems to work on. Will create threads to work on individual chunks of problems.

TEAM INFINITE TUPLES

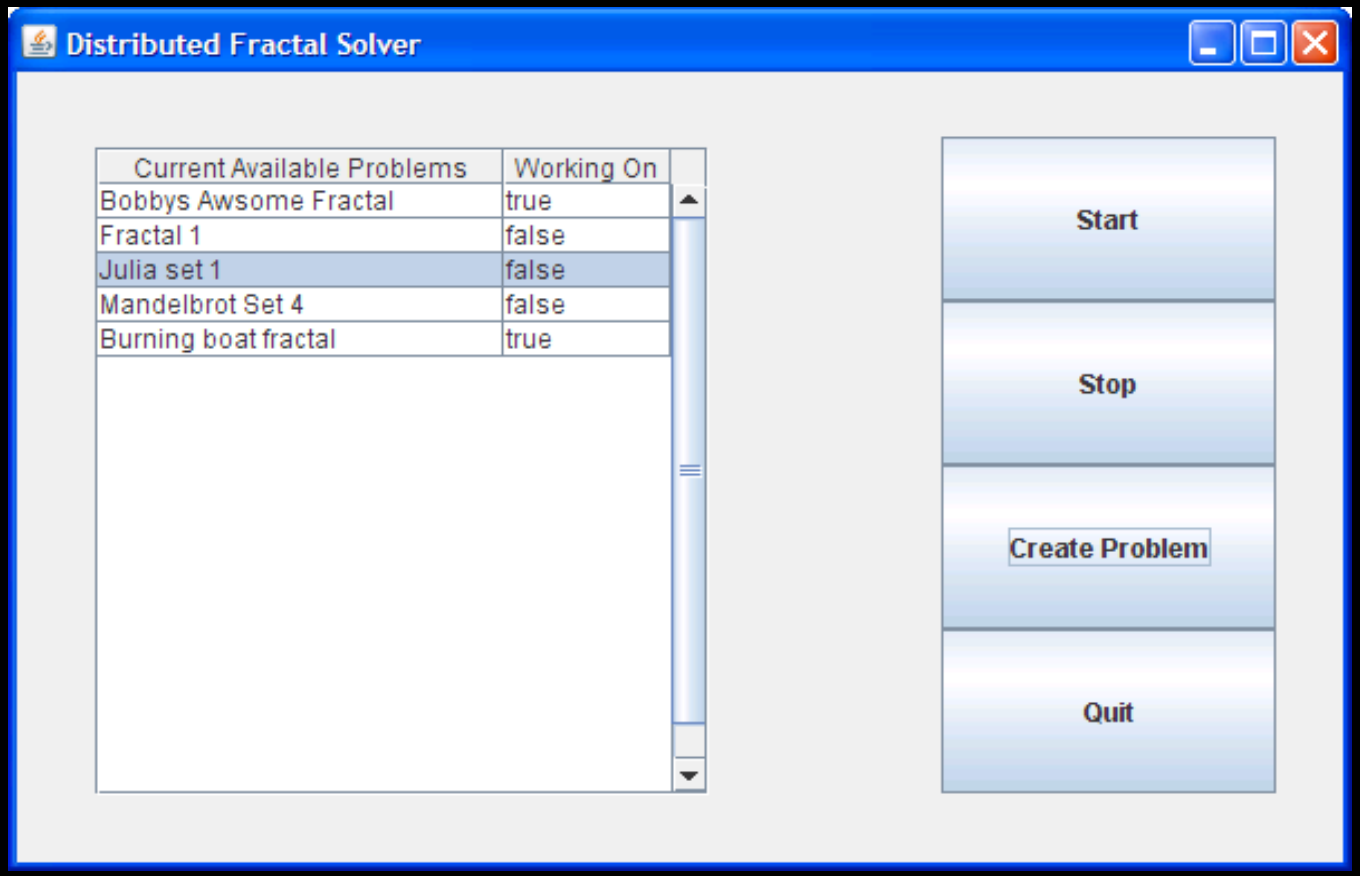
<u>Product Design Review:</u>

- Communication through 2 tuple types
- Problem Tuple: Contains formula, parameters, Name, poster, number of chunks.
- Chunk tuple: Contains copy of Problem tuple, Chunk number and parameters, Data (Empty or complete), metric.
- The Chunk tuples act as the state of the problem. They are initially empty, but will be updated as more chunks of the problem are completed.


Control Panel UI
Control Panel Logic
Reads Problems
Passes Tuple
For Matching
Problem
Tuple
Posts Problem
Read Chunks
Chunk
Tuple
Posts Chunks
Solver Logic
Creator Logic
Solver UI
Creator UI

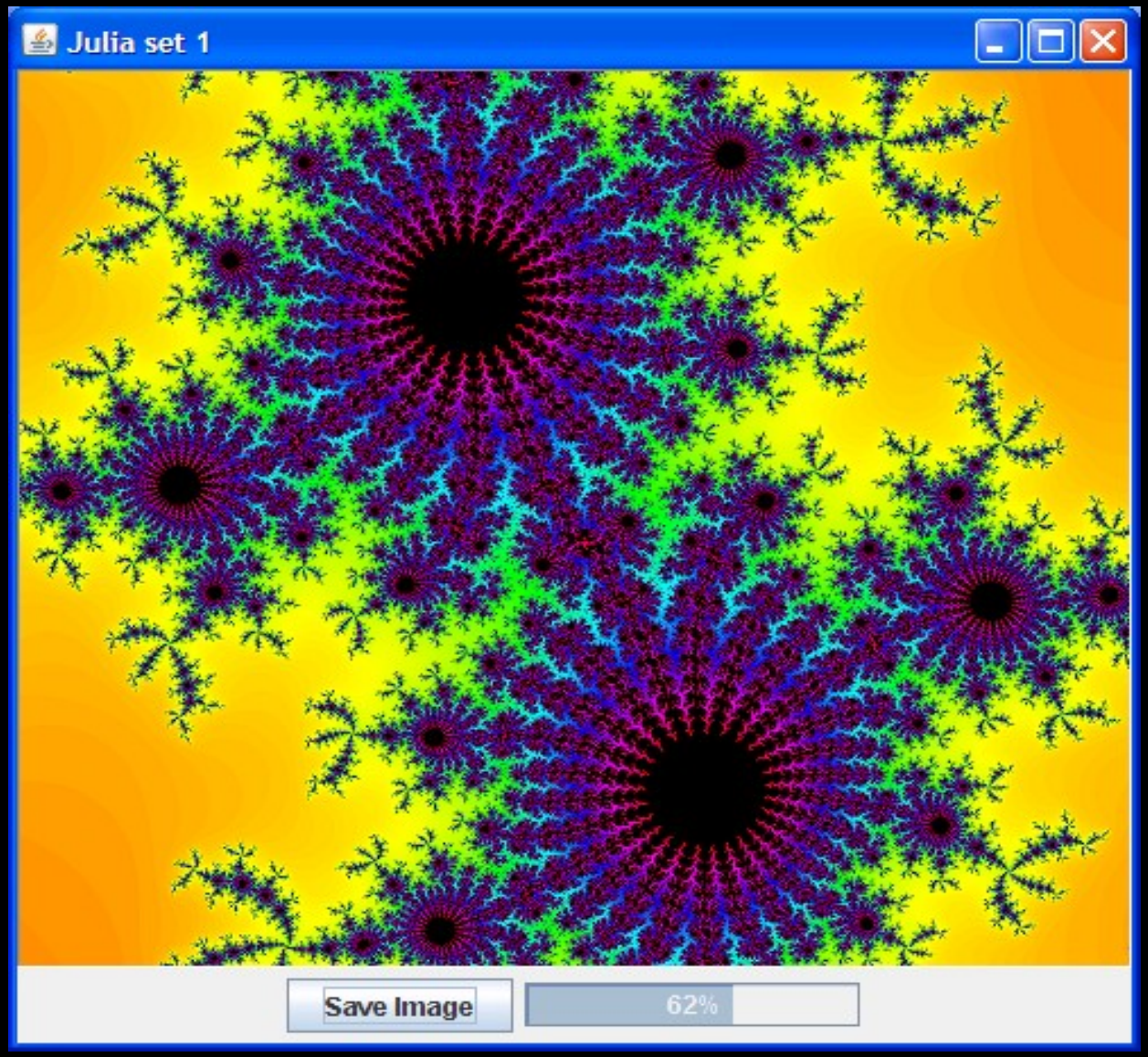
TEAM INFINITE TUPLES

GUI:



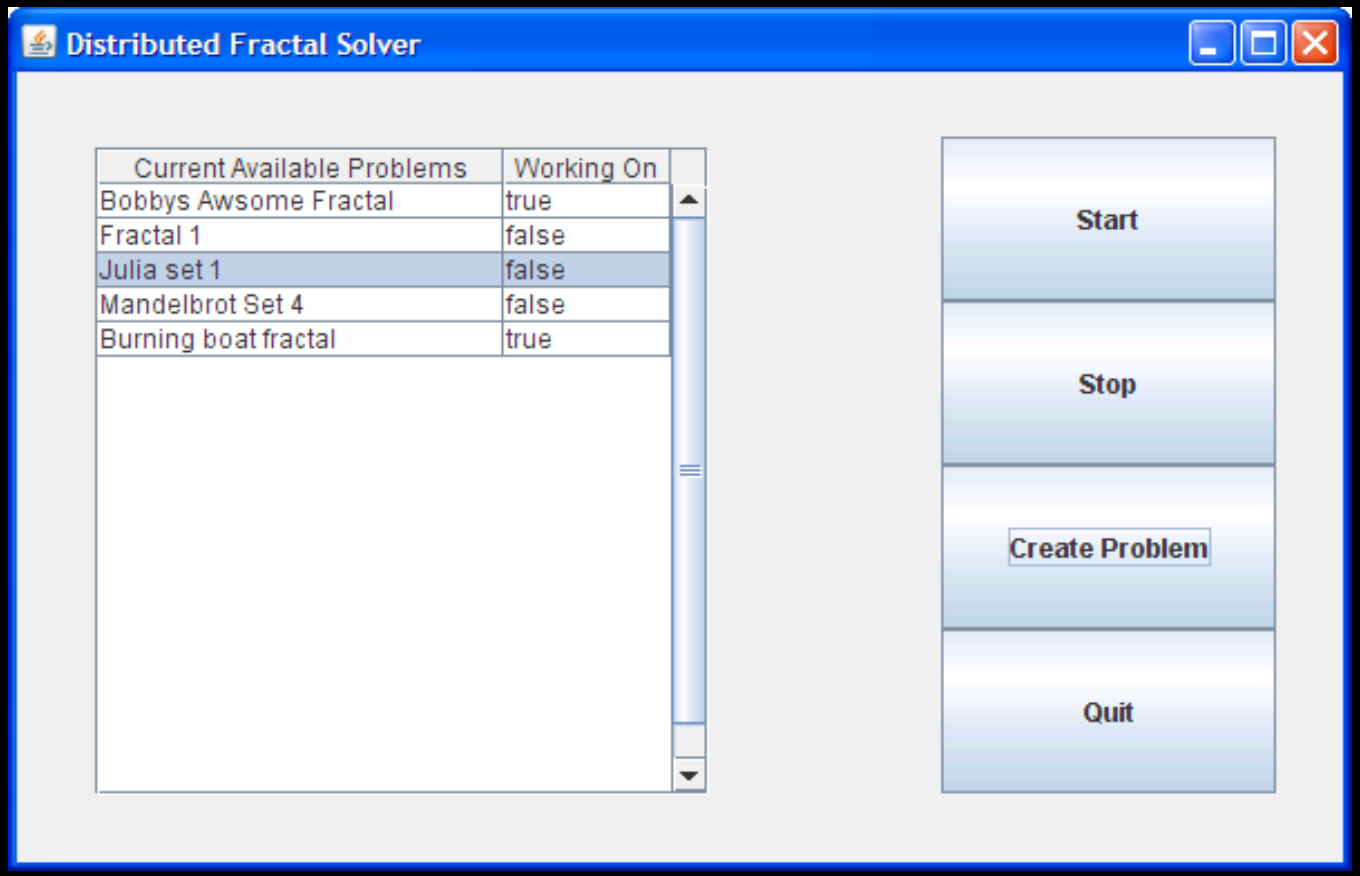
TEAM INFINITE TUPLES

GUI:



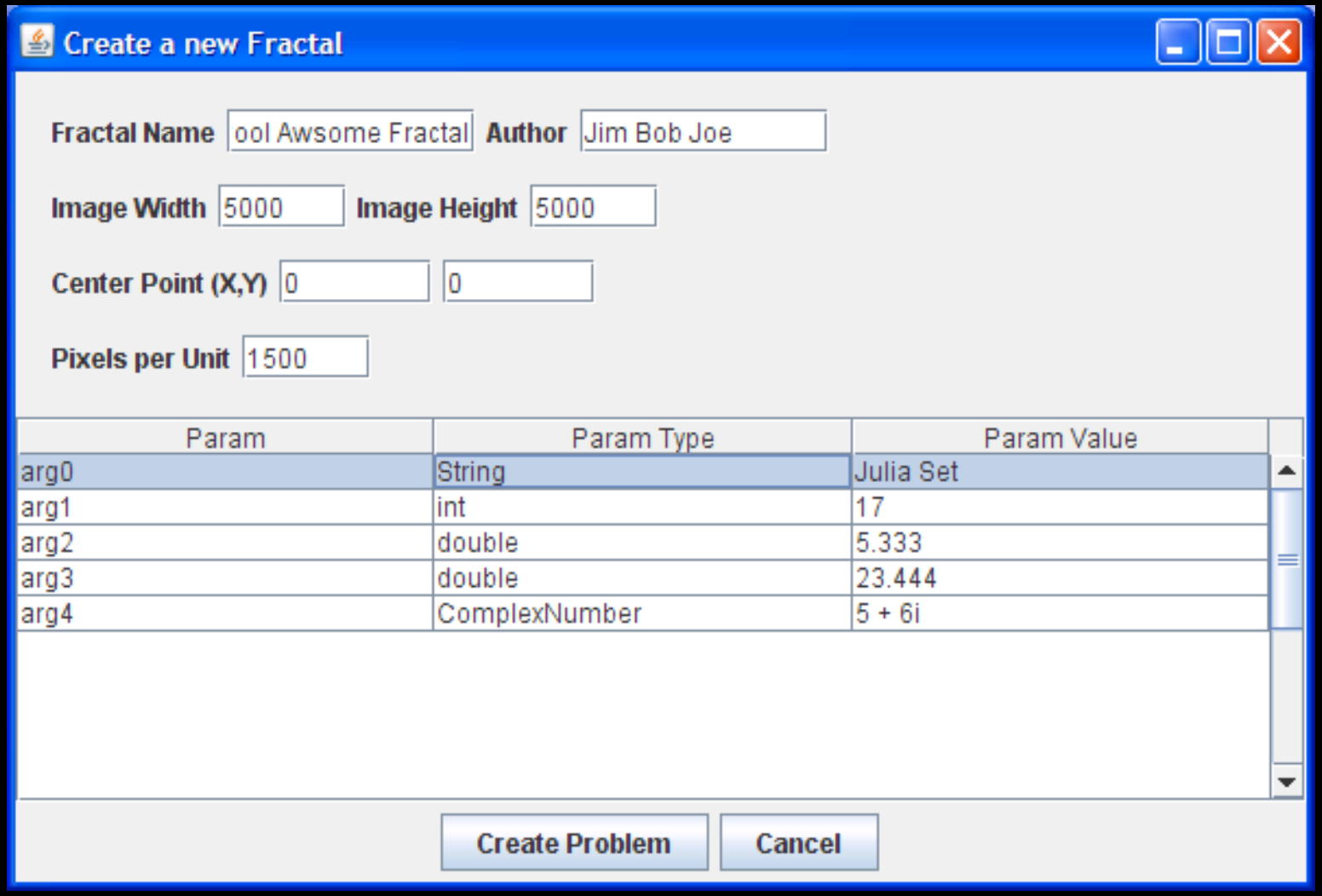
TEAM INFINITE TUPLES

GUI:



TEAM INFINITE TUPLES

GUI:



TEAM INFINITE TUPLES

Problem Creation:

- Implement and compile the fractalImplementation interface
- Load the class file using a ClassLoader
- Use Java Reflections to find the constructor and allow the user to enter in any constructor arguments
- Enter picture parameters, such as height, width, resolution, etc
- Create the needed tuples

TEAM INFINITE TUPLES

Problem Example:

```
public class myTestFractal implements fractalImplementation {
        private int maxIter;
        private double breakOut;

        public myTestFractal(String maxIter, String breakOut){
                this.maxIter = Integer.parseInt(maxIter);
                this.breakOut = Double.parseDouble(breakOut);
        }

        public Color getPixelColor(double x, double y) {
                Random chooser = new Random();
                Color pixel = new Color( chooser.nextInt(255), chooser.nextInt(255),
                        chooser.nextInt(255));
                return pixel;
        }
}
```

TEAM INFINITE TUPLES

Problem Solving:

- Get the class and constructor arguments from the Problem tuple
- Load the class using a ClassLoader
- Use Java Reflections to initialize the class using the constructor with arguments
- Loop through every pixel in the image and use reflections to invoke the getPixelColor method
- Output the computed pixel data

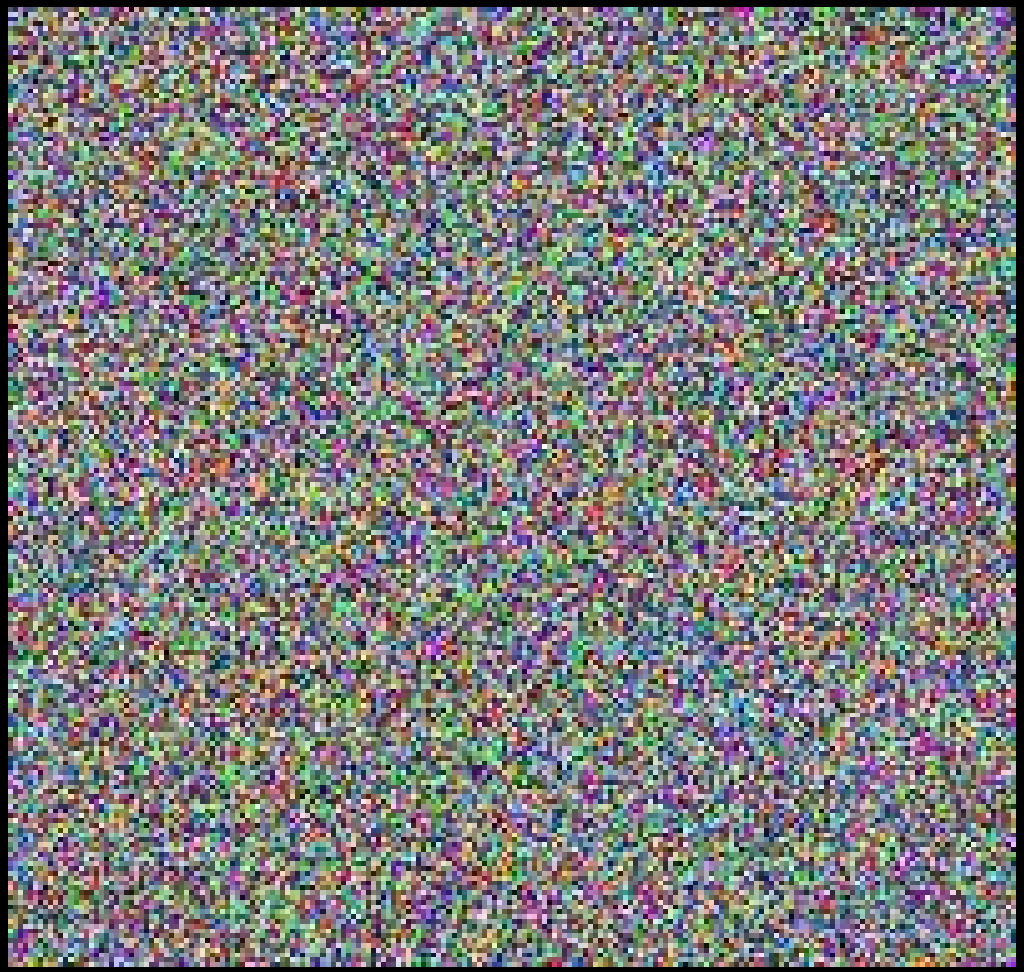
TEAM INFINITE TUPLES

Solve Example:

```
for(int row = 0; row < probTuple.height; row++){
        methArgs[0] = new Double(probTuple.ycenter + (yoffset-row) / probTuple.resolution);
        for(int col = 0; col < probTuple.width; col++){
                methArgs[1] = new Double(probTuple.xcenter + (xoffset+col) /
                probTuple.resolution);
                Object ret = null;
                try{
                        ret = iterMeth.invoke(probInstance, methArgs);
                }catch(IllegalAccessException iae){
                        iae.toString();
                }catch(InvocationTargetException ite){
                        ite.toString();
                }
                image.setPixelColor(row, col, (Color)ret);
        }
}
```

TEAM INFINITE TUPLES

<u>Output:</u>



http://www.cs.rit.edu/~bdi8241/adhoc/