

Save and Open

You need a redraw method. This one is for a 1D array, since that is what my game has.

```
public void redraw() {
    for(int i=0; i<visible.length; i++)
        pics[i].setIcon(createImageIcon(visible[i]+".jpg"));
}
```

Redraw simply takes all of the pictures stored in "visible" and prints them out.

Saving and opening requires creating a file from within you program. This file can be updated during the game and it can persist after the game is done.

First you need to open the file. After that, the files work almost exactly like System.out.println and in scanner. Instead of reading and writing to the console, they read and write to a file..

So, make a save method:

```
public void save (String filename) {
    PrintWriter out;
    try {
        out = new PrintWriter (new FileWriter (filename));
        for(int i=0; i<values.length; i++)
            out.println (""+values[i]);
        for(int i=0; i<visible.length; i++)
            out.println (""+visible[i]);
        out.close ();
    }
    catch (IOException e) {
        System.out.println ("Error opening file " + e);
    }
}
```

This code opens a file and prints out the arrays to the file.

The two highlighted loops output the two arrays which store the values of the onscreen array. They are one dimensional. You may have more or less arrays, they may be two dimensional. Adapt the code to suit your arrays.

Then, make an open method:

```
public void open(String filename){
    BufferedReader in;
    try {
        in = new BufferedReader (new FileReader (filename));
        String input = in.readLine ();
        for(int i=0; i<values.length; i++) {
            values[i]= Integer.parseInt (input);
            input = in.readLine ();
        }
        for(int i=0; i<visible.length; i++) {
            visible[i]= input;
            input = in.readLine ();
        }
    }
}
```

```

        in.close ();
    }
    catch (IOException e) {
        System.out.println ("Error opening file " + e);
    }
    redraw();
}

```

This code opens the file, reads in the information, closes the file and redraws the screen.

Again, notice that there are two loops, one for each array. The first one is an integer array, so `parseInt` is used. The second is a String array, so no `parseInt` is needed. In each line in the loop, you need to read in the number from the file (green line).

Add the buttons to the screen:

```

JButton save = new JButton("Save");
save.addActionListener(this);
save.setActionCommand("save");
add(save);

JButton open = new JButton("Open");
open.addActionListener(this);
open.setActionCommand("open");
add(open);

```

Note: you might be adding the buttons to a card instead of to the applet surface.

Add the `actionCommands` to TOP OF `actionPerformed`:

```

if(e.getActionCommand().equals("save")){
    save("file.txt");
} else if (e.getActionCommand().equals("open")){
    open("file.txt");
}

```