Stephen Asunka

MSTU 5031 - JSP Programming
Spring 2006

FINAL PROJECT

Proposal

Abstract

This project aims at developing an interactive, CD-ROM/web-based, multimedia science learning resource that will assist New York State teachers integrate technology into the Grade 6 science classroom. This product can serve as a valuable knowledge base and reference point for project-based and problem solving learning activities.

Target Audience

NY State Grade 6 Students and their Teachers

Objective

The NY State standards for Grade 6 Science include the following:

4.
The Physical Setting: Students will understand and apply scientific concepts, principles, and theories pertaining to the physical setting and living environment and recognize the historical development of ideas in science. …

> 4.3.
> Matter is made up of particles whose properties determine the observable characteristics of matter and its reactivity.
>
> 4.3.1.
> Students observe and describe properties of materials, such as density, conductivity, and solubility.
>
> 4.3.2.
> Students distinguish between chemical and physical changes.
>
> 4.3.3.
> Students develop their own mental models to explain common chemical reactions and changes in states of matter.
> (source: http://www.brainpop.com/support/standards_correlation.weml)

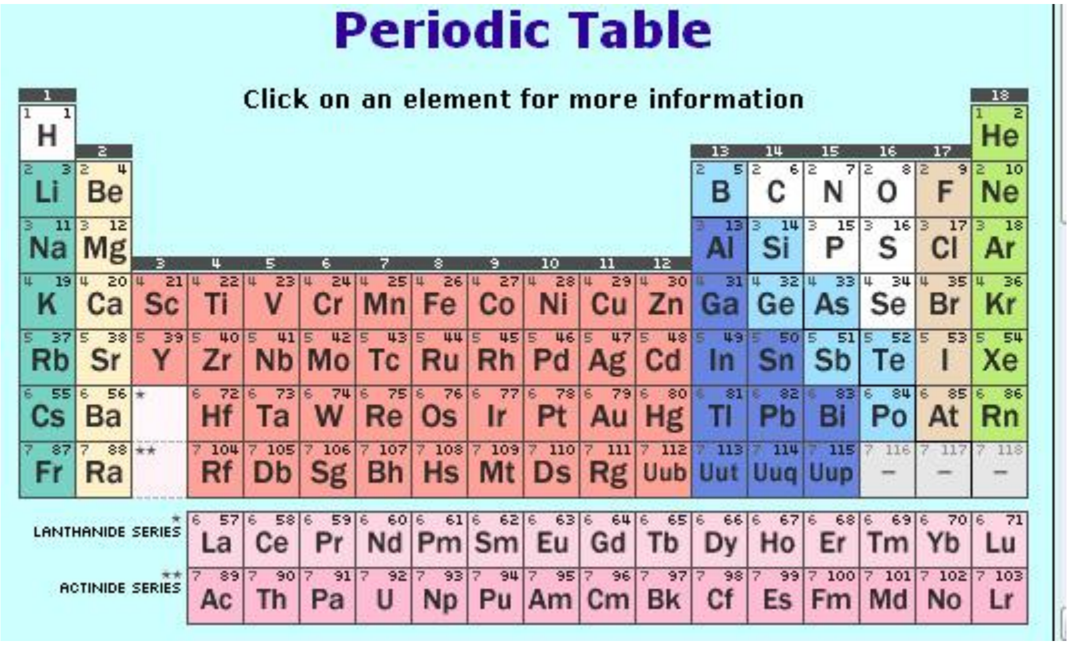
The objective of this project is to create a multimedia instructional unit that will assist New York state Grade 6 students acquire the knowledge and experiences listed above as part of their science learning activities.

Context

This application can be used as a stand-alone learning resource or can be integrated into classroom science learning activities.

Functional Use

For students to clearly understand the nature and properties of the particles (elements) that make up the world, these elements have been organized into a form of chart called the Periodic Table (see diagram below).



Periodic Table

Click on an element for more information

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1 1 H																	1 2 He
2 3 Li	2 4 Be											2 5 B	2 6 C	2 7 N	2 8 O	2 9 F	2 10 Ne
3 11 Na	3 12 Mg											3 13 Al	3 14 Si	3 15 P	3 16 S	3 17 Cl	3 18 Ar
4 19 K	4 20 Ca	4 21 Sc	4 22 Ti	4 23 V	4 24 Cr	4 25 Mn	4 26 Fe	4 27 Co	4 28 Ni	4 29 Cu	4 30 Zn	4 31 Ga	4 32 Ge	4 33 As	4 34 Se	4 35 Br	4 36 Kr
5 37 Rb	5 38 Sr	5 39 Y	5 40 Zr	5 41 Nb	5 42 Mo	5 43 Tc	5 44 Ru	5 45 Rh	5 46 Pd	5 47 Ag	5 48 Cd	5 49 In	5 50 Sn	5 51 Sb	5 52 Te	5 53 I	5 54 Xe
6 55 Cs	6 56 Ba	*	6 72 Hf	6 73 Ta	6 74 W	6 75 Re	6 76 Os	6 77 Ir	6 78 Pt	6 79 Au	6 80 Hg	6 81 Tl	6 82 Pb	6 83 Bi	6 84 Po	6 85 At	6 86 Rn
7 87 Fr	7 88 Ra	**	7 104 Rf	7 105 Db	7 106 Sg	7 107 Bh	7 108 Hs	7 109 Mt	7 110 Ds	7 111 Rg	7 112 Uub	7 113 Uut	7 114 Uuq	7 115 Uup	7 116 –	7 117 –	7 118 –
* LANTHANIDE SERIES			6 57 La	6 58 Ce	6 59 Pr	6 60 Nd	6 61 Pm	6 62 Sm	6 63 Eu	6 64 Gd	6 65 Tb	6 66 Dy	6 67 Ho	6 68 Er	6 69 Tm	6 70 Yb	6 71 Lu
** ACTINIDE SERIES			7 89 Ac	7 90 Th	7 91 Pa	7 92 U	7 93 Np	7 94 Pu	7 95 Am	7 96 Cm	7 97 Bk	7 98 Cf	7 99 Es	7 100 Fm	7 101 Md	7 102 No	7 103 Lr

A thorough understanding of the Periodic Table gives students a firm foundation that enables them understand subsequent issues pertaining to chemical reactions etc.

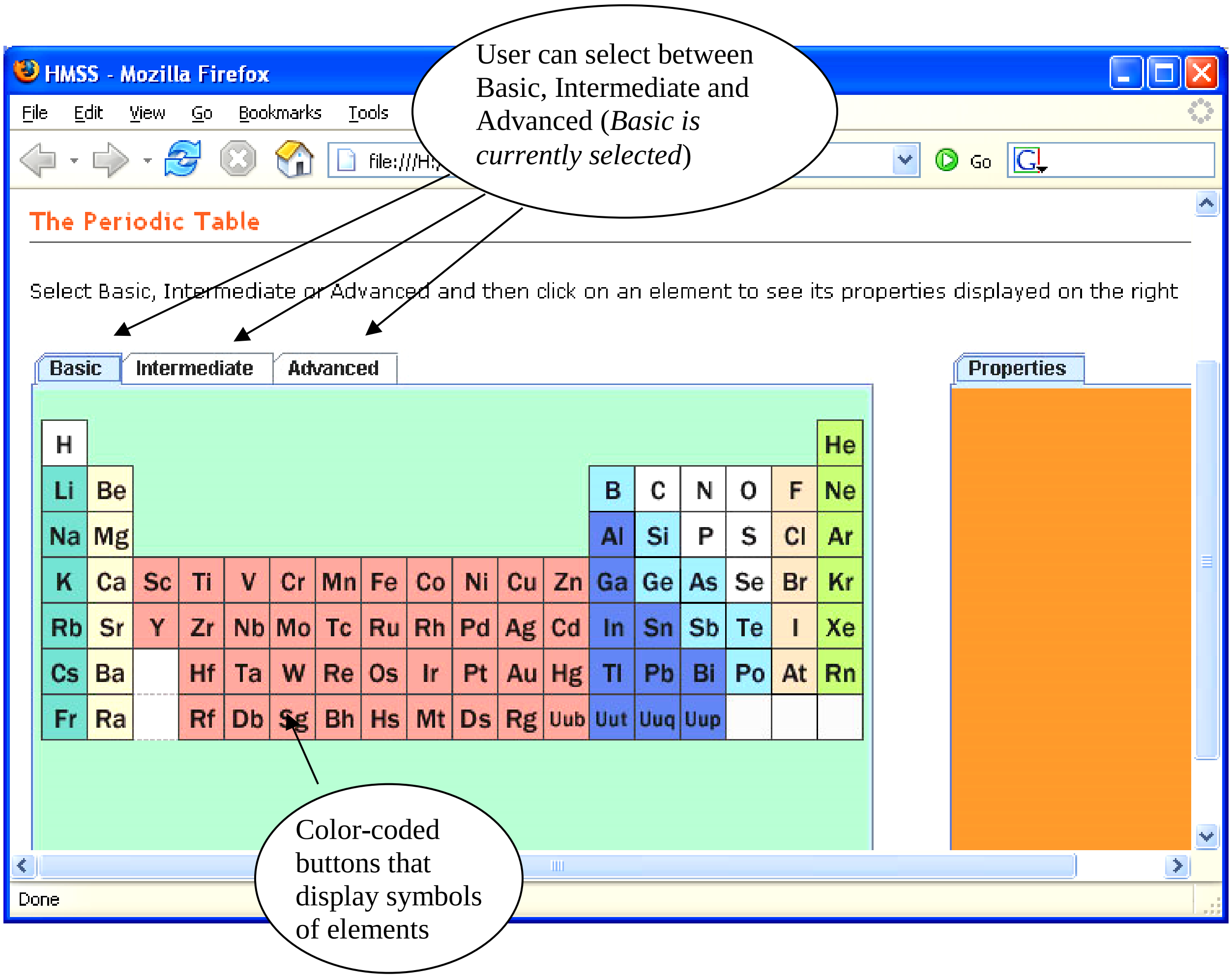
Within this project, Java will be used to create an interactive Periodic Table. Symbols of the various elements will be displayed as in the diagram shown above. However when each symbol is clicked, a pop-up window will display all the relevant information including pictures (stored in a database) about the particular element. Students will also have the opportunity to select the level of complexity of the information they need about each element, i.e. basic, intermediate or advanced.

In addition, the resource will consist of several hyperlinked pages that will guide students through an understanding of the names, symbols, appearances and basic properties of each element. At the end of the learning experience, students will create slide shows depicting their understanding of the properties and uses of the each element.

Screenshots of Interface

This screenshot is a webpage that displays two applets - Periodic Table (which is interactive) and Properties. With the Periodic Table, a user has the option of selecting between Basic, Intermediate and Advanced levels.

Symbols of the various elements are arranged (as pertains in the normal periodic table) as clickable buttons.



User can select between Basic, Intermediate and Advanced (*Basic is currently selected*)

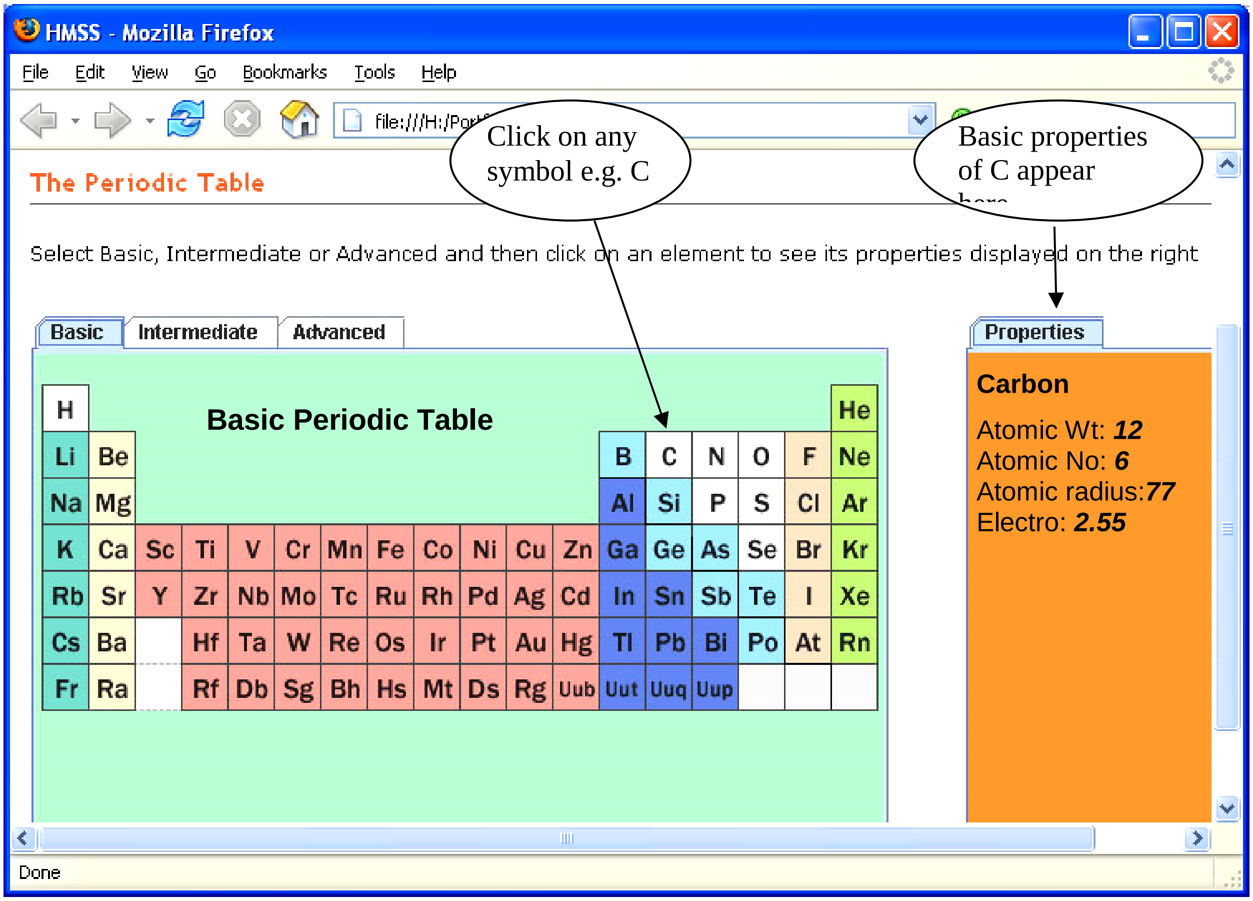
The Periodic Table

Select Basic, Intermediate or Advanced and then click on an element to see its properties displayed on the right

H																	He
Li	Be											B	C	N	O	F	Ne
Na	Mg											Al	Si	P	S	Cl	Ar
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe
Cs	Ba		Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn
Fr	Ra		Rf	Db	Sg	Bh	Hs	Mt	Ds	Rg	Uub	Uut	Uuq	Uup			

Color-coded buttons that display symbols of elements

On clicking on a particular button, the properties of the element that the button represents are displayed in the properties applet. The details of the properties displayed depend on the level of the Periodic Table that is currently being used.



Click on any symbol e.g. C

Basic properties of C appear here

The Periodic Table

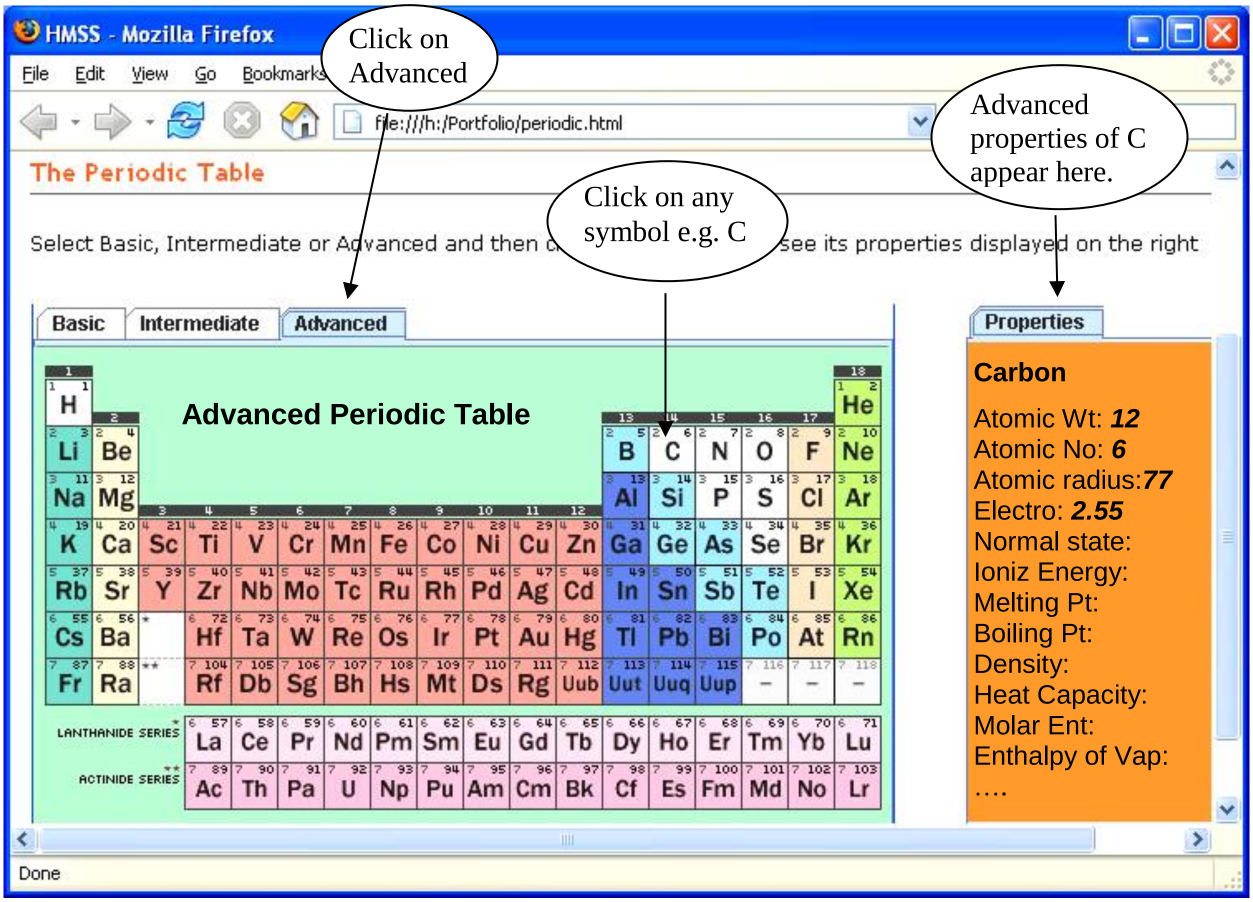
Select Basic, Intermediate or Advanced and then click on an element to see its properties displayed on the right

Basic Periodic Table

H																	He
Li	Be											B	C	N	O	F	Ne
Na	Mg											Al	Si	P	S	Cl	Ar
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe
Cs	Ba		Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn
Fr	Ra		Rf	Db	Sg	Bh	Hs	Mt	Ds	Rg	Uub	Uut	Uuq	Uup			

Carbon

Atomic Wt: ***12***
Atomic No: ***6***
Atomic radius:***77***
Electro: ***2.55***


HMSS - Mozilla Firefox
File Edit View Go Bookmarks
file:///h:/Portfolio/periodic.html
Basic
Intermediate
Advanced
Properties
Done

Click on Advanced

Advanced properties of C appear here.

Click on any symbol e.g. C

The Periodic Table

Select Basic, Intermediate or Advanced and then c … see its properties displayed on the right

Advanced Periodic Table

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
H																	He
Li	Be											B	C	N	O	F	Ne
Na	Mg											Al	Si	P	S	Cl	Ar
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe
Cs	Ba	*	Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn
Fr	Ra	**	Rf	Db	Sg	Bh	Hs	Mt	Ds	Rg	Uub	Uut	Uuq	Uup	–	–	–

LANTHANIDE SERIES *	La	Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu
ACTINIDE SERIES **	Ac	Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr

Carbon

Atomic Wt: ***12***
Atomic No: ***6***
Atomic radius:***77***
Electro: ***2.55***
Normal state:
Ioniz Energy:
Melting Pt:
Boiling Pt:
Density:
Heat Capacity:
Molar Ent:
Enthalpy of Vap:
….

Programming

1. Periodic Table

Classes and Methods (using JFC/Swing)

Classes
public class PeriodicTable extends JApplet implements ActionListener
This is the main class of the applet using Swing

public class BasicPanel extends JPanel

public class IntermediatePanel extends JPanel

public class AdvancedPanel extends JPanel

The three main panels that will constitute a tabbed pane

Methods in PeriodicTable Class
init()
This method initializes the applet

getContentPane()
Gets contents

createGUI()
creates the GUI

setLayout()
Creates layout of panels

addTab()
Adds the three panels to create a tabbed pane

add()
Adds panel to display properties of elements

getAppletContext().getApplet("Applet2")
Establishes link with other applet

Methods in Basic Panel Class (same for the other panel classes)
BasicPanel()
Constructor

setLayout ()
Creates grid layout of element buttons

JButton
Button class

JButton b1 = new JButton("H")
Instance of JButton - creates a button representing the element with symbol H

> setBorder()
> *Creates preferred border of buttons*
>
> setBackground()
> *Sets background color of element buttons*
>
> setPreferredSize()
> *Sets size of element buttons*
>
> setToolTipText(“Hydrogen”)
> *Displays element name as tooltip text when mouse rolls over corresponding button*
>
> setAllowMultipleSelection()
> *Sets whether multiple selection of elements is allowed. Default: false*
>
> addActionListener(new ActionListener())
> *Register an event handler that puts the text into the properties pane*
>
> add()
> *Adds button to grid*

public void actionPerformed(ActionEvent e)
Handles the button click event

setActionCommand()
Sets the action command for this button
(This will involve connecting to database of element properties)

getActionCommand()
Returns properties of selected element and displays them in a table

2. Web Pages (using JSP)

i. Home page that contains a form requiring a user to log in

public class Login
This class handles the logging in of a user

public boolean userlogin (String username, String password)
This method takes a user's information (name & password), checks in database and then redirects user to either a "confirmation" page or a try again screen

ii. A Register page that contains a form requiring new users to register

public class Register
This class handles the registration of new users to the website

public Boolean registration()
This method takes a new user's information, checks members database and if the new information is unique, adds the information and returns a confirmation or otherwise

3. Database (Microsoft Access)

i. Members Table
Table with only 3 columns - Firstname, Password, Email

ii. Properties Table
Table with over 20 columns - Name, Atomic_Number, Atomic_weight, Melting_point etc etc.

Reflections

Creating an interactive periodic table using java applets turned out to be a much more complicated and mind boggling issue than I had previously envisaged. Two main issues had to be considered - layout and functionality.

Layout

Placing two applets (a *sender* that contains the periodic table and a *receiver* that contains a text area to output text) on the same HTML page was not much of a problem. Also, placing three different panels as tabs in one container in the *sender* applet was quite easy to accomplish. The difficulty however was arranging the buttons that represent each element in a way as pertains in standard periodic table. I promptly realized that I could not achieve this with a simple one or two-line code. Each button had to be created separately. There are 103 elements but I had to create over 140 buttons, put them in a grid layout and then make some of them invisible.

Functionality

Placing the getAppletContext().getApplet("Receiver"); line of code in the *sender* applet gets it to communicate with the *receiver* applet.

I then envisaged that when a periodic table button in the *sender* applet is clicked:

- That particular button changes to a unique color
- Information about the element represented by that button is pulled from a database
- The formatted information is displayed in the text area of the *receiver* applet

When another button is clicked, the previous button goes back to its normal color, the present button changes to the unique color, and the information in the text area is instantly replaced with new information that pertains to the present button.

The first difficulty I encountered was that I could not figure out how to implement the methods (either within or without) each panel so as to establish the desired functionality. I converted the panels into stand alone applets, but in trying to get them arranged as tabs in a container in the *sender* applet ruined the entire layout. I therefore dropped the idea of 3 panels in one container and resorted to having just a single *sender* applet. After doing this, I succeeded in getting (hard-coded) information displayed in *receiver* applet when a button in the *sender* is clicked, but I again realized that most code will have to be added to each and every button separately. I am still in the process of testing and fine-tuning the functionality before I will consider the database connectivity, text formatting etc.