SPiral Model Reinforced in Teaching Introductory Programming Course

Mirela Djordjevic
Manhattanville College (UNITED STATES)

Abstract

How to reconcile difficulties in learning and teaching complex subject of Computer Science (CS). The need for introductory CS courses for all college students is well established, as they benefit by instilling critical reasoning and making rigorous computer code. During the course students are overwhelmed by programming language syntax and strictness in coding. How to make these students enjoy the struggle of finding syntax errors, looking for paths to solve problems, and rewriting program codes. Spiral model in teaching is well established but this model must be reinforced as students struggle with accepting new notions and combining them with the previous knowledge especially in introductory programming courses. Teaching introductory Visual Basic course with this methodology shows positive effects on students learning.

Keywords: Spiral model, Progressive Assignments, Introductory Visual Basic Course.

1 INTRODUCTION

Manhattanville College (Mville) is a liberal arts school that offers major in CS, but many non CS major students take introductory programming courses to satisfy quantitative reasoning. During lectures and labs, students use computers to actively follow the work of an instructor. The non CS major students need different approach in teaching introductory programming courses.

Understanding Object-Oriented terms is essential to fully understand VB programming language, as VB is built around methods, objects, instantiations and such. According to our experience, the recommendation is to teach the features that are basic to all programming languages such as program syntax, arithmetic operations and control structures, “mixed” with the specifics of the object oriented terms and visual controls. Thus we teach objects and classes, and event-driven programming later in the course, but use them at the very beginning of the course. How to “combine” these topics? We try to incorporate the spiral model of teaching and also spiral model of grouping these concepts.

2 SPIRAL MODEL

Spiral model in teaching Visual Basic with progressive assignments is a good methodology in teaching introductory programming course. Important issues of reinforcing spiral model are to develop more tracks in the model that can establish links with the previous layers in the spiral model. These tracks are supported by the additional requirements in the progressive assignments.

Topics in a course are divided up into fragments and the fragments introduced in an order that facilitates student problem solving. Many of the fragments introduce a topic, but do not cover it in detail. Just enough detail is given initially so as to form a basic understanding that can be applied to problem solving. Additional cycles contain reinforcing fragments that go into more detail on the topic. How to implement this spiral model in the introductory programming course? How to combine diverse components of the CS curriculum?

3 MVILLE SPIRAL MODEL REINFORCED

Topics in a course are divided into three fragments: visual controls and events, fundamental concepts, and object oriented concepts represented in Appendix 3. These groups show with different colors: visual controls as pink, fundamental concepts as violet and object oriented concepts as blue. Each cycle introduces a new element in the fragment, and continues to combine it with all previously defined concepts.
Tracks of reinforcement are developed to establish links with the previous layers in the spiral model. These tracks are supported by posing questions and requirements in the progressive assignments. They are distinguished from the rest with the "(R)".

4 PROGRESSIVE ASSIGNMENTS

Progressive assignments are set as a base for practicing new notions introduced each week. Assignments progress by answering questions and building new projects.

5 STUDENTS’ RESPONDS

Some students’ responds “This class was the first introduction I’d ever had to computer science and programming”. “I feel that your teaching was effective in that I now have a far greater understanding and good deal of knowledge of the content in addition to learning how to apply it.” “I think you did an excellent job in teaching such a difficult subject with limited time to a class with no prior knowledge of the content.”

6 CONCLUSION

We try to reconcile some difficulties in learning and teaching complex subject of introductory programming in Visual Basic by making tracks in the spiral model of concepts that remind, refresh, and reinforce the important notions. Students evaluate the course as hard and demanding, but also enjoy-
able. They recommend adding vocabulary to define basic terms introduced in each lecture which will be added in the following school year.

REFERENCES


APPENDIX - GRAPHICS PROGRESSIVE ASSIGNMENTS

1. Draw some shape on a panel when a button is clicked.
   **Goals:** Visual Studio (basic constructs: Form, Button, simple library defined draw methods)
   **Concepts:** Button/Button click; Sequential structure; library defined Classes/Objects.
   **Questions:** How to specify position of a shape?
   Note: Objects and classes are mentioned but not explored in detail.

2. Draw a shape on a panel when a side of the shape is specified in the text box/input box.
   **Goals:** Variables and types (int and double
   **Concepts:** Textbox/Label, InputBox/MessageBox; Variable/ Type
   **Reinforce:** Sequential structure
   **Questions:** How to change a side of the shape?

3. Draw a shape on a panel when a side of the shape is specified by the track bar.
   **Goals:** Library defined classes (Track bar, Button, Text box)
   **Concepts:** Variable/Type; Track bar class, Track bar property value, Track bar scroll event
   **Reinforce:** Panel class, Panel class properties width and height
   **Questions:** How to modify a position of the shape?

4. Draw a shape on a panel when a mouse is clicked on the panel.
   **Goals:** Methods and parameters
   **Concepts:** Panel, Mouse click on the Panel; Methods/Parameters; library defined Classes/Objects
   **Reinforce:** InputBox/MessageBox, TextBox/Label
   **Questions:** How to specify color of a shape?

5. Change a color of the shape when the radio button is selected.
   **Goals:** nonlocal variables vs local variables, selection statement
   **Concepts:** Radio Buttons, Radio Button selected; Selection/If statement;
   **Reinforce:** TrackBar
   **Questions:** How to be more specific about the shape color?

6. Draw shapes with the random colors
   **Goals:** Library defined class Random
   **Concepts:** Random class, Random color
   **Reinforce:** nonlocal variables vs local variables
Questions: How to be more specific about the shape?
7. Define a shape by defining a class.
Goals: Use Library defined classes
Concepts: User defined class
Reinforce: Mouse click event, Radio button, Track bar
8. Redo projects 1-5 with different shape objects.
Goals: Simple circle class, square class, shape class.
Concepts: User defined class
Reinforce: local/non-local variables
Question: How to display information about shapes that are displayed on the panel?
9. Store the information about all shapes displayed on the panel in the text box.
Goals: Practice loops, Textbook with multiline property set as true.
Concepts: text box with multiline property; for loops; collections of object
Reinforce: Text box properties and methods
Questions: How to color all shapes on the panel?
10. Store all shapes as a collection and display string information in the list box.
Goals: Loops and collections
Concepts: List box, click/double click on the list box element; while loops; collections of object;
Reinforce: ListBox class, ListBox property items
Question: How to select a specific shape from the collection?
11. Select a particular shape in a collection and color/remove it.
Goals: Loops and Listbox
Concepts: List box and its events; loops; ArrayList
Reinforce: Loops, Collections, Random class, InputBox/MessageBox
12. Draw 100 shapes at random positions onto the panel. String information about shapes in the Listbox.
Goals: Loops ListBox
Concepts: List box and events; collections of objects;
Reinforce: Loops, Collections, Random class, InputBox/MessageBox

<table>
<thead>
<tr>
<th>Project</th>
<th>Visual</th>
<th>Fundamental</th>
<th>OOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Panel, Button, Label</td>
<td>Statement Sequence</td>
<td>Graphics class, Drawing methods</td>
</tr>
<tr>
<td>2</td>
<td>InputBox/MessageBox; Text-Box/Label</td>
<td>Variables/TypesAssignment statement</td>
<td>Graphics object</td>
</tr>
<tr>
<td>3</td>
<td>Mouse click onto a panel</td>
<td>Methods/Parameters</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Functions/Parameters</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Track bar, Radio buttons, Checkboxes</td>
<td>If statements</td>
<td>Exceptions</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>User defined classes</td>
</tr>
<tr>
<td>7</td>
<td>Textbox / multiline property</td>
<td>If statements</td>
<td>Classes, properties</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>For Loops</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ListBox</td>
<td>While Loops</td>
<td>ArrayLists</td>
</tr>
<tr>
<td>10</td>
<td>ListBox events</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1. Progressive Assignments – Concepts.**
**Table 2. Progressive Assignments - Reinforced Spiral model.**

<table>
<thead>
<tr>
<th>Project</th>
<th>Visual</th>
<th>Fundamental</th>
<th>OOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Panel, Button, Label</td>
<td>Statement Sequence</td>
<td>Graphics class, Drawing methods</td>
</tr>
<tr>
<td>2</td>
<td>InputBox/MessageBox; Text-Box/Label</td>
<td>Variables/Types Statement Sequence</td>
<td>Graphics object</td>
</tr>
<tr>
<td>3</td>
<td>Mouse click onto a panel vs. Button click event</td>
<td>Methods/Parameters Assignment statement</td>
<td>Panel class</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Functions/Parameters</td>
<td>Random class</td>
</tr>
<tr>
<td>5</td>
<td>Track bar, Radio buttons, Check boxes</td>
<td>If statements</td>
<td>Exceptions</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Methods/Parameters</td>
<td>User defined classes Panel class</td>
</tr>
<tr>
<td>7</td>
<td>Textbox / multiline property</td>
<td>If statements</td>
<td>Classes, properties</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>For Loops</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ListBox</td>
<td>While Loops</td>
<td>ArrayLists, Exceptions</td>
</tr>
<tr>
<td>10</td>
<td>Track bar, Radio buttons, Check boxes</td>
<td>Loops</td>
<td>Random class, Array List class</td>
</tr>
</tbody>
</table>