Last Updated: 3/5/15

Topic Name: UML State and Sequence Diagram – Not Yet Graded

Brief Description:

State Diagram:
When writing a State Diagram it is best practice to display the objects in a particular class per diagram.
The idea behind a State Diagram is that each system has a behavior that needs to be described to understand how the system "thinks". Each block on the diagram itself represents a "state" of the system component being observed. Each component/block will have one of more behaviors that may lead to different outcomes that are dependent on the state the machine is in at that specific moment in the process. Here in my example of a simple etch-a-sketch program I give two different types of state diagrams to better understand this system.

Sequence Diagram:
Sequence diagrams are a way to describe interactions of classes and objects in relation to the other classes and objects that they communicate with. They show in a sequence of time how each object or class communicates through messages, and when those messages are sent or received. This helps us understand and describe the logical portion of the system. The sequence diagram works complementary of the use case diagram and is sometimes called "event driven". As each event occurs we can follow the path of communication of the objects as we move through time of execution to completion. In my example I show how a simple etch-a-sketch program might look as it sends signals to the other classes and objects.

Teaching Examples:

None

Files to View:

- State Diagram - StateDiagram.pdf
- State Diagram - EventStateDiagrams.pdf
- Sequence Diagram - Sequence Diagram.pdf

Last Updated: 2/21/15

Topic Name: UML Use Case – Already Graded and Completed

Brief Description:
This week I looked at a couple of the UML diagrams and refreshed on a couple of them. I used the links that were provided on I-Learn to study up on them and I have a couple that I've done to show how they are used in a practical example of software design. The Use Case is essentially a system that describes other systems or "cases" that are involved in my example of a program. There are two key subjects in a Use

Case diagram: Actors and Use Cases. Actors are anything that will have interactions with the program such as a person or even a database. The Use Cases are simply everything that the user can **see**, **touch**, or **change** in the program. These will mostly just be buttons in my example, as well as the canvas area of the screen.

Teaching Examples:

None

Files to View:

Use Case - Turtle One And Six.pdf