

공학석사 학위논문

상호작용 애니메이션을 이용한 자료구조 알고리즘의 학습 시스템



2006년 2월

부경대학교 산업대학원

전 산 정 보 학 과

장 수 미

장수미의 공학석사 학위논문으로 인준함

2005 년 12 월 9 일

주 심 공학박사 박 승 섭



위 원 공학박사 김 영 봉



위 원 공학박사 정 순 호 (인)

목 차

표 목차	iii
그림 목차	iv
Abstract	vi
1. 서론	1
2. 관련연구	3
2.1 기존 알고리즘 학습 시스템	3
2.2 애니메이션 구현 유틸리티	8
2.2.1 액션스크립트	10
3. 학습 시스템	12
3.1 상호작용 애니메이션을 이용한 자기주도적 학습	12
3.1.1 상호작용 애니메이션의 표현	14
3.2 자료구조 학습	17
3.2.1 학습 내용	17
4. 설계 및 구현	19
4.1 화면 설계	19
4.2 개발 환경	22

4.3 구현	23
5. 평가	36
6. 결론	37
참고문헌	38

표 목차

<표 1> 기존 알고리즘 학습 시스템의 특징	6
<표 2> 애니메이션 구현 유틸리티의 비교	9
<표 3> 소프트웨어 및 하드웨어 환경	22
<표 4> 기존 학습 시스템과의 비교	36

그림 목차

(그림 1) BLASA	7
(그림 2) XTANGO	7
(그림 3) POLKA	7
(그림 4) JSAMBA	7
(그림 5) JAWWA	7
(그림 6) 3개의 원판 이동 애니메이션	7
(그림 7) 스택 동작 동영상 화면	7
(그림 8) 플래시를 이용한 저작과정	11
(그림 9) 학습 시스템	12
(그림 10) 애니메이션 과정	13
(그림 11) 상호작용 애니메이션	13
(그림 12) 학습 내용	17
(그림 13) 학습 단계	18
(그림 14) 화면 구성	18
(그림 15) 초기화면 설계	20
(그림 16) 기본학습화면 설계	23
(그림 17) 심화학습화면 설계	21

(그림 18) 초기 화면	23
(그림 19) 메인 화면	24
(그림 20) 기본학습 화면	25
(그림 21) 심화학습 화면	26
(그림 22) 스택	27
(그림 23) 선형 큐	28
(그림 24) 환형 큐	28
(그림 25) 삽입정렬(1)	29
(그림 26) 삽입정렬(2)	30
(그림 27) 선택정렬	30
(그림 28) 버블정렬	31
(그림 29) 퀵정렬	32
(그림 30) 병합정렬(1)	33
(그림 31) 병합정렬(2)	33
(그림 32) 힙정렬(1)	34
(그림 33) 힙정렬(2)	35
(그림 34) 힙정렬(3)	35

Learning System of Data Structure Algorithms using Interactive Animation

Soo-Mi Jang

*Dept. of Computer and Information science
Pukyong National University*

Abstract

This paper implements the learning system of data structure algorithms available according to the level of learners under a remote education environment.

The existing learning systems of the data structure are deficient in the active participation of learners and interactions between learners and the systems, because the animations of them using fixed numeric data are not presented interactively.

In systems, it which it is possible for learners to control data size, instruction execution rate and tracking code in the course of the execution of the algorithms, enable self-directed learning. And the effective learning is provided by implementing interactive animations of Flash Action Script on web.

1. 서론

자료구조 알고리즘에 대한 학습은 프로그래밍에 익숙해지고자 하는 모든 사람들에게 필수적이면서, 처음 이 부분을 학습하는 학생들에게는 이해하기에 손쉽지 않은 부분이다. 따라서 이러한 애로점을 고려하면 학습자의 이해를 쉽게 진작시킬 수 있는 학습 시스템의 개발이 필요하다. 그러나 1980년대까지는 알고리즘 학습 시스템이 개인용 컴퓨터나 워크스테이션 상에서 인스톨되어 사용되었으므로 장소적 한계를 극복할 수 없었다. 또한 전통적인 교육환경에서는 공급자인 교사가 수요자인 학생에게 제한된 공간 내에서 제한된 시간에 수업내용을 전달하는 것이 일반적이므로 공간적 제약뿐 아니라 교사 학생 모두에게 시간적 불리함 (time penalty)을 초래했다[1]. 1990년대 이후 인터넷의 보급 및 확산으로 소프트웨어 시각화에 대한 연구가 웹 기반 하에 이루어지면서 장소적, 시간적 불리함을 해결할 수 있게 되었다. 웹 기반 교육은 교육의 효율성과 더불어 전통적 학습에 비해 많은 장점을 지닌다. 일반적인 장점으로선 전달위주의 단방향식 학습 방법에서 벗어나 상호작용을 통해 적극적으로 학습을 유도하여 학업 성취도를 높일 수 있으며, 학생 개인의 특성을 고려한 차별화된 교과과정과 학습내용의 제공으로 개인의 학습자로 하여금 원하는 학습목표 도달을 용이하게 해 준다는 점이다. 또한 장소 독립적 교육을 가능하게 해 준다는 장점이 있다 [2]. 그러나 현재 개발되어 있는 웹기반 시스템은 단순히 자료 제시 형에 그치거나 간단한 애니메이션으로 설명을 대신한 것에 불과하여 학습자 입장에서 보면 동적인 요소보다 문서위주의 단순한 정적인 요소가 많고, 단순한 지식 전달로, 현실감이 부족하기 때문에 학습효과가 미흡하다.[3]

따라서 자료구조 알고리즘의 학습 시스템을 상호작용 애니메이션 구현이 가능한 플래시 액션스크립트(Flash Action Script) 기반의 교수자가 아닌 학습자 중심으로 설계한다. 이는 학습자가 여러 요인들을 직접 조작해서 결과를 확인하고, 학습 능력 향상 및 학습자들이 이해하기 힘든 부분을 상호작용 애니메이션으로 처리하여 학습자들이 직접 눈으로 확인하면서 학습할 수 있도록 하여 흥미를 유발시키고 이해를 도와 학습 효과를 증대 시켜 줄 수 있다.

본 논문에서는 Action Script의 애니메이션을 이용하여 자료구조 알고리즘 학습을 위한 적극적인 상호작용을 제공하는 시스템을 웹에서 제한 없이 자유롭게 시뮬레이션 해 볼 수 있도록 구현 한다.

본 논문은 모두 5장으로 구성되어 있으며 그 내용은 다음과 같다. 2장에서는 기존 알고리즘 시스템에 대한 연구와 플래시와 플래시 액션스크립트에 대해 알아보고, 3장에서는 제안하는 자료구조 알고리즘 학습 시스템에 대한 전반적인 내용을 기술한다. 4장에서는 시스템의 설계 및 구현을 하고, 마지막 5장에서는 결론을 맺는다.

2. 관련연구

2.1 기존 알고리즘 학습 시스템

알고리즘의 시각화에 대한 많은 연구가 행해지고 있는데, 대표적인 시스템으로는 Balsa[4], XTANGO[5], POLKA[6], SAMBA[7], JAWWA[8] 등이 있고, 현재에도 연구가 계속되고 있다.

Balsa는 1984년 시각화 시스템 가운데 처음으로 상호작용을 구현한 것으로 알고리즘 애니메이션 분야에 있어 선구자적인 역할을 하였다. 이후 개발되는 소프트웨어 시각화를 평가하는 지표로 사용된 Balsa는 Brown 대학의 전산학 수업시간에 이용되었고 그 외에 실제 알고리즘 설계와 분석에 이용되었다. 이 시스템은 주로 C언어로 작성되었고, 애니메이션 부분은 PASCAL로 작성되었다. 하나의 디스플레이 상에서 동시에 다중 뷰를 제공한 첫번째 시스템으로, 다중 뷰를 보여줌과 동시에 다중 알고리즘의 실행을 보여주었다. 사용자는 완전한 애니메이션을 실행시킬 수 있고, 이미 실행했던 애니메이션을 재생할 수 있도록 액션을 기록할 수 있다.

TANGO는 1990년 Brown 대학에서 알고리즘 애니메이션 프레임워크로 개발되었다. 대부분의 소프트웨어 시각화는 애니메이션 표현을 위해 간단한 지우기 - 그리기 기법을 사용하여 왔다. 그러나 TANGO는 시각적으로 부드러운 밑그림을 개발하여, 각 단계별 이행에 대한 특정한 코드에 대해 높은 수준의 애니메이션을 보여준다. TANGO의 초기 버전은 유닉스 워크스테이션 상에서 동작하였고, 다양한 프로그램에 대한 이차원 배열의 칼라 애니메이션만을 제공하였다. XTANGO라고 불리는 현재 버전은 C를 컴파

일 할 수 있는 대부분의 UNIX 워크스테이션과 X11 윈도우시스템 상에서 동작한다. XTANGO는 다수의 알고리즘 애니메이션라이브러리를 가지고 있어 사용하기에 쉽다. 프로그래머는 그래픽 전문가일 필요가 없으며, 이러한 점이 다른 소프트웨어 시각화와 다른 점이다.

1993년에 개발된 POLKA는 XTANGO에서 유래된 범용 애니메이션 시스템이며, 다른 시스템보다 더 쉽고, 빠르게 애니메이션을 구현할 수 있고, 특히 병렬 프로그램이나 계산을 애니메이션 하는데 사용 될 수 있다. 색, 실시간, 2차원과 부드러운 애니메이션을 지원하며, 사용이 간편하고, 인터랙티브를 지원한다. POLKA의 원시 버전은 UNIX와 X11 윈도우 상에서 C++에 의해 수행되었으며, 새로운 버전인 POLKAW는 window 95나 window NT 상에서 실행되며 C++로 수행된다. POLKAW는 ftp.cc.gatech.edu로 부터 ftp를 이용하여 다운 받을 수 있다.

POLKA로부터 유래한 SAMBA는 POLKA의 특징을 포함하면서 POLKA보다 많은 면에서 개선되었다. SAMBA는 어떤 언어의 프로그램이든 실행할 수 있다. 그 까닭은 언어에 상관없이 ASCII 명령어를 한 줄씩 읽어 들여 번역한 후 애니메이션을 수행하기 때문이다. 이러한 명령어는 배우고 사용하기 쉽도록 설계되었다. JSAMBA라 불리는 SAMBA의 Java 버전이 개발되면서 웹상에서 기능을 충분히 시험할 수 있게 되었다.

JAWWA는 1996년 Duke 대학의 자료구조 강의에 처음으로 사용되었다. 이는 별도의 절차 없이 웹에서 쉽게 애니메이션을 생성하기 위한 스크립트 언어이다. Java로 작성되었기 때문에 사용자가 애니메이션을 작성할 수 있는 인터페이스를 제공하며, Java를 지원하는 모든 웹 브라우저에서 동작한다. 원, 사각형, 다각형, 선, 텍스트와 같은 간단한 프리미티브(primitive) 객체들을 생성할 수 있고, 각각 또는 그룹으로 객체들을 움직일 수 있다. 사용자는 JAWWA 명령어를 파일로 생성하고, 간단한 HTML파일을 수정함

으로써 웹에서 동작할 수 있는 애니메이션을 쉽게 생성 할 수 있다.

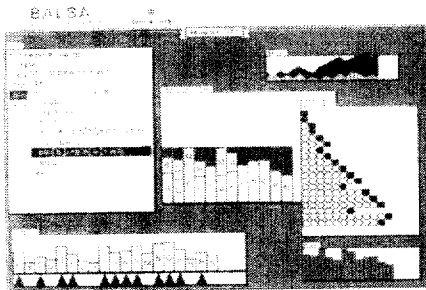
이런 시스템들은 기본적인 기능들을 제공하는 데에는 성공했지만, 디스플레이가 제한적이고 특정 환경이나 애플리케이션을 필요로 한다. 국내 시스템으로는 '알고리즘시각화를 위한 SVG코드 생성기의 설계 및 구현(2004)[9]'은 XML을 기반으로 2차원 그래픽을 표현하기 위해 만들어진 언어로서 W3C(World Wide Web Consortium)에 의해 제안된 XML 그래픽 표준인 SVG를 이용한다. 이미 작성되어 있는 소스 프로그램에 간단한 시각화 스크립트를 삽입하고 이에 프리프로세서 라이브러리를 포함하여 프리프로세서 과정을 거쳐 컴파일하고 실행하게 되면, 프로그램의 실행결과와 함께 알고리즘을 시각화하기 위한 SVG 애니메이션 코드를 자동 생성하게 된다.

'자료구조 학습을 위한 자기 주도적 코스웨어 설계 및 구현(2004)[10]'은 실업계 고등학교 자료구조 교과에서 자료구조의 이론학습 및 알고리즘 수행과정을 실습을 할 수 있도록 설계하였다. 자바스크립트로 구현한 프로그램을 이용하여 학습자가 임의 데이터를 입력하여 상호학습 (Feedback Learning)을 지원하였고, 각 단원의 내용을 동영상 (Flash Animation)으로 표현하여 학습자가 각내용에 대한 제어 버튼을 제공하였다. 그러나 사용자와의 상호작용이 미비하거나 애니메이션이 단순하다는 문제가 있다.

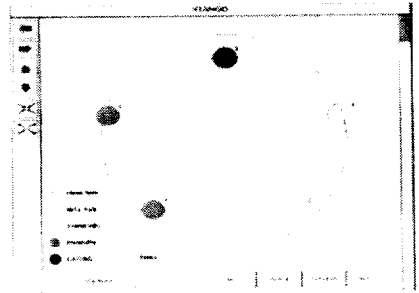
알고리즘 학습 시스템의 특징은 <표 1>과 같이 정리된다.

<표 1> 기존 알고리즘 학습 시스템의 특징

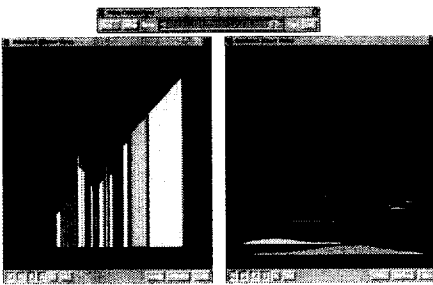
제목	특징	연도	구현언어
BLASA	- 각각의 알고리즘을 실행하는 다중뷰 지원 - 다양한 라이브러리 제공 - 뷰 윈도우 내용이 제공자에 의해 미리 정해져 있음	1984	C
XTANGO	- 다양한 라이브러리 루틴을 제공하여 쉽고 빠르게 활용 - 한번에 하나의 이미지만 작업 - 자료구조 표현하기에 부족	1990	C
POLKA	- 병렬 프로그램이나 계산을 시각화	1993	C++
SAMBA	- 프로그램 언어에 대한 제한 없이 시각화 가능	1994	C++
JAWWA	- 애니메이션의 빠른 구현 - Java 기반으로 웹에서 동작가능	1996	Java
알고리즘시각화를 위한 SVG코드 생성기의 설계 및 구현	- XML 그래픽 표준인 SVG 애니메이션 코드를 생성하여 시각화 - 웹브라우저에서 실행	2004	C
자료구조 학습을 위한 자기 주도적 코스웨어 설계 및 구현	- 동영상(Flash Animation)과 심화학습(Feedback Learning) 제공 - 제어 버튼(처음, 다음, 이전, Show, stop)을 이용한 상호작용 학습	2004	Flash



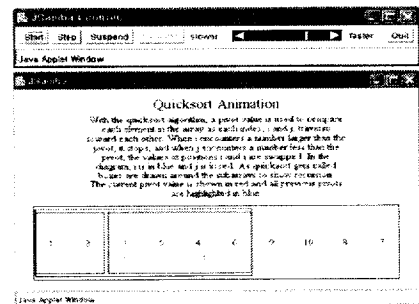
(그림 1) BLASA



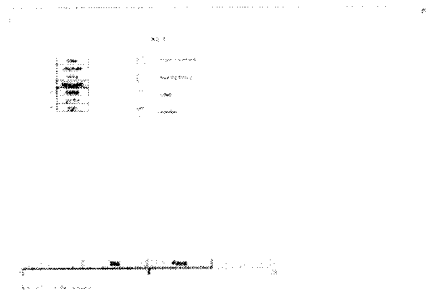
(그림 2) XTANGO



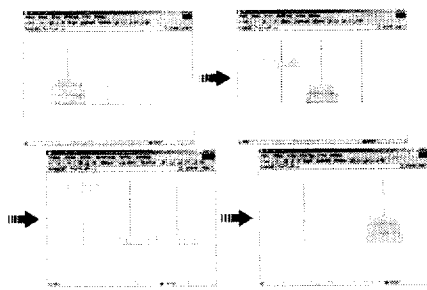
(그림 3) POLKA



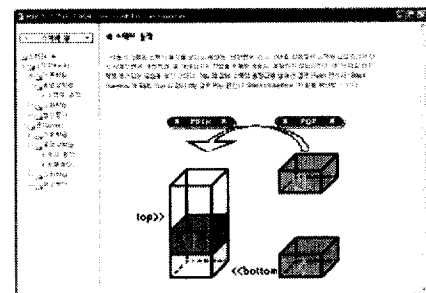
(그림 4) JSAMBA



(그림 5) JAWWA



(그림 6) 3개의 원판 이동애니메이션



(그림 7) 스택 동작 동영상 화면

2.2 애니메이션 구현 유틸리티

애니메이션을 생성하기 위한 접근법은 첫째, 파워포인트와 같은 전형적인 프레젠테이션 툴을 사용하는 것, 둘째, Macromedia Dreamweaver, Flash, Animal 등의 드래그 앤드 드롭이나 옵션 선택으로 시각요소를 편집하는 것, 셋째, Jeliot와 AlgAE 등 소스 코드로부터 직접 애니메이션을 생성하는 것, JAWWA, JSAMBA, Animal, Flash와 같이 스크립트 언어에 의해 애니메이션을 생성하는 것 등으로 구분된다[11].

프레젠테이션 툴은 사용하는 방법은 비교적 쉽고 장비의 전환 없이 강의에 통합될 수 있으나 애니메이션 효과의 확장이 제한적이다.

드래그 앤드 드롭이나 옵션선택으로 시각 요소를 편집하는 방법은 효과를 즉시 확인 할 수 있으므로 애니메이션을 생성하는 가장 쉬운 방법 중 하나이며, 별도의 프로그래밍 경험을 필요로 하지 않는다는 장점이 있다.

소스코드로부터 시스템에 의해 애니메이션을 생성하는 방법은 자동으로 생성되므로 가장 쉬운 애니메이션 생성을 제공한다. 동일 주제에 대해 다양한 애니메이션을 생성하는 것도 추가적인 노력 없이 행해질 수 있으며, 소스 코드가 변경되어도 알고리즘 인수를 변경하거나 알고리즘을 다시 시작하는 것만으로도 새로운 애니메이션을 생성 할 수 있으나, 애니메이션 툴에 의해 지원되는 프로그래밍 언어가 제한적이며, 소스 자체가 간단명료하지 않거나 긴 실행시간을 요하는 문제인 경우 적절한 애니메이션을 기술하는 것이 쉽지 않고, 애니메이션 효과가 소스 코드 인터프리터 고정코드로 삽입되므로 코드가 표현되는 방법을 사용자가 설정할 수 없다.

스크립트 언어에 의해 애니메이션을 생성하는 방법은 일반적으로 명령어는 직관적으로 정의되며, 애니메이션 정의가 자명하므로, 프로그래밍 지식

없이도 애니메이션 파일을 쉽게 읽을 수 있고, 애니메이션 정의가 자동으로 생성되며, 어떤 프로그래밍 언어에도 사용될 수 있다. 이 방법에서 코드 각각에 대한 이해는 용이한 반면, 원하는 효과를 생성하거나, 이를 알고리즘 코드에 삽입하는 과정은 어려울 수 있다.

지금 까지 살펴본 애니메이션 구현 유틸리티에 대해 <표 2>와 같이 정리하였다.

<표 2> 애니메이션 구현 유틸리티의 비교

애니메이션 방법	장점	단점	예
프레젠테이션 툴이용	·사용이 용이	·애니메이션 효과의 확장 제한적	Powerpoint
시각화면에서 직접 시각요소 편집	·애니메이션 생성이 용이 ·별도의 프로그래밍 불필요	·지원되는 프로그래밍 언어가 제한적 ·애니메이션 효과를 사용자가 결정할 수 없음	Macromedia Dreamweaver, Flash, Animal
소스코드로부터 직접 애니메이션 생성	·애니메이션의 자동생성 ·소스 코드 변경에 따른 애니메이션 생성 용이	·지원되는 프로그래밍 언어가 제한적 ·애니메이션 효과를 사용자가 결정할 수 없음	JAL, AlgAE
스크립트 언어에 의한 애니메이션 생성	·명령어의 이해 및 사용 용이	·프로그램 내에 코드를 삽입하는 과정이 어려울 수 있음	JAWWA, JSAMBA, Flash, Animal

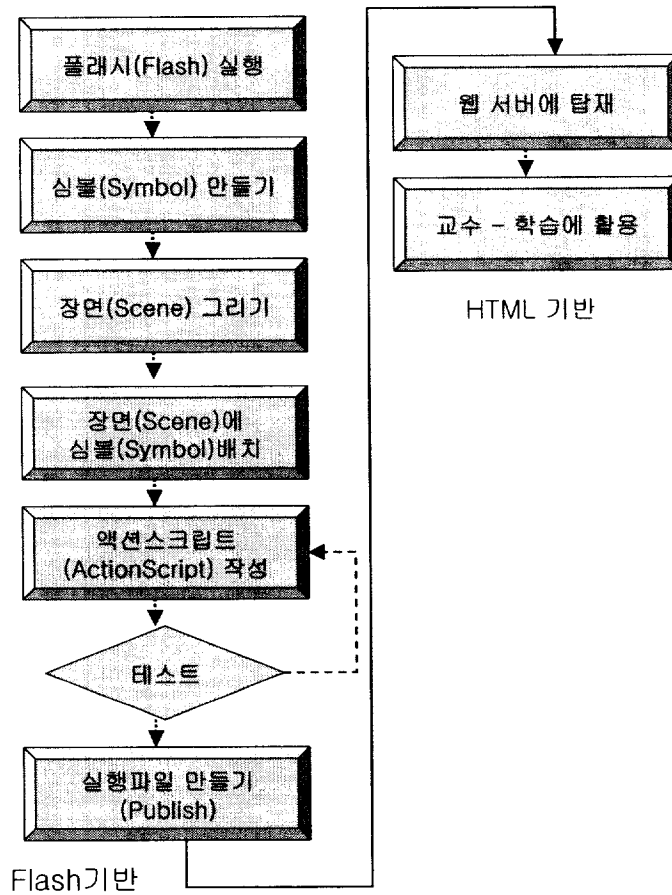
2.2.1 액션스크립트(ActionScript)

액션스크립트(ActionScript)는 플래시(Flash) 5.0부터 탑재된 자바스크립트(JavaScript)와 유사한 프로그램 내에서 사용되는 스크립팅 언어(Scripting Language)로 객체 지향언어이다. 플래시에서는 타임라인을 이용하여 애니메이션을 제작할 수 있는 것은 물론, 액션스크립트를 이용하여 강력하고 풍부한 상호작용이 포함된 멀티미디어 시스템을 제작 할 수 있다 [12].

액션스크립트를 이용하면 플래시 무비 내의 시각적, 청각적 요소들을 명령어를 이용해서 제어하게 되는데, 이러한 명령어들 하나하나를 가리켜 액션이라고 하며, 액션스크립트(ActionScript)는 액션들이 모여서 구성되는 코드(Code)를 의미한다.

플래시 무비 안에 액션스크립트가 사용되면 무비가 플래시플레이어(Flash Player)로 재생될 때 이 액션스크립트를 해석하여 실행한다. 플래시 액션스크립트(ActionScript)는 스크립트 언어이므로 상대적으로 짧은 기간 안에 개발 가능하고, 운영체제의 영향을 받지 않으며, 다른 웹 프로그램언어와 연동이 가능하다.

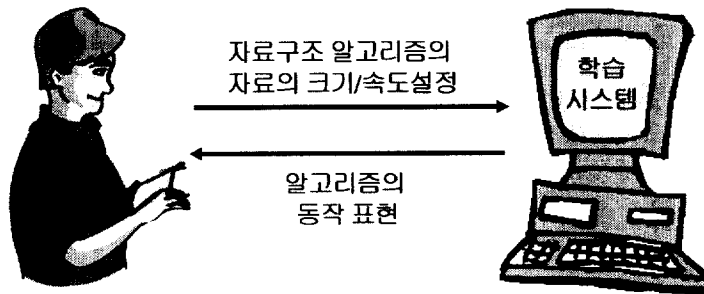
플래시 액션스크립트 기반의 시스템은 저작도구나 다른 원격 학습 시스템에 비해 다음과 같은 장점이 있다. HTML(Hypertext Makeup Language) 기반이므로 트래픽을 감소시킬 수 있으며 웹 브라우저만 있다면 강의를 들을 수 있다는 범용성과, 시스템을ダイナ믹하게 만들 수 있어 학습시의 지루함을 효과적으로 상쇄시킬 수 있고, 시스템의 업데이트와 수정이 용이하다는 것이다. 저작 과정은 (그림 8)과 같이 2단계를 거친다.



(그림 8) 플래시(Flash)를 이용한 저작과정

3. 학습 시스템

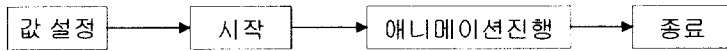
본 시스템은 상호작용 애니메이션을 통한 자기 주도적 학습과 자료구조 알고리즘 정의를 텍스트를 통한 기본학습과, 심화학습을 통한 알고리즘의 특징을 파악하고 분석한다.



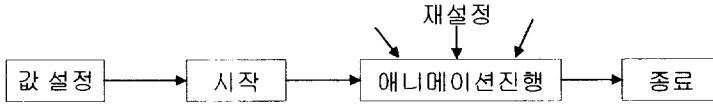
(그림 9) 학습 시스템

3.1 상호작용 애니메이션을 이용한 자기 주도적 학습

기존의 시스템에서는 교수자에 의해 정해진 값의 애니메이션이 진행되었으나 본 학습 시스템에서는 (그림 10)과 같이 애니메이션 진행 중에라도 학습자가 여러 가지 요인들을 제어할 수 있는 상호작용 애니메이션을 이용하여 자기 주도적 학습을 지원한다.



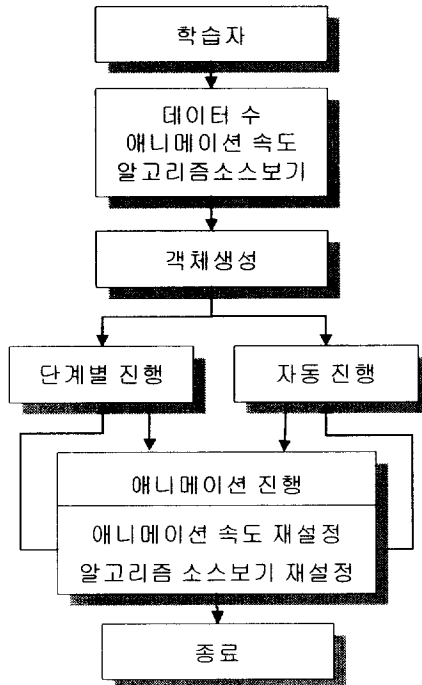
(a)정해진 값에 의한 애니메이션



(b)상호작용 애니메이션

(그림 10) 애니메이션과정

상호작용 애니메이션은 학습자가 데이터 수, 애니메이션 속도, 동기화된 알고리즘 소스 보기, 단계별 및 자동 진행 등의 값을 설정하여 학습을 시작하여 학습도중이라도 이러한 요인들 값을 재설정하여 애니메이션을 제어할 수 있다.



(그림 11) 상호작용 애니메이션

3.1.1 상호작용 애니메이션의 표현

각 자료구조 알고리즘의 시각 표현에서 요구되는 애니메이션의 공통된 기능을 지원하는 객체들의 표현을 다음과 같이 통일한다.

(1) 개체 그리기

MovieClip 클래스의 메서드를 사용하여 스테이지에 선과 칠을 그린다. 사용자를 위한 드로잉 도구를 만들고, 이벤트에 대한 응답으로 무비에 모양을 그려서 모든 무비 클립에 드로잉 메서드를 사용한다. 또한 무비 클립을 복제해서 사용한다, 이는 SWF 파일이 재생되는 동안 무비 클립의 인스턴스를 작성하여, 복제된 무비 클립의 재생 헤드는 원본(또는“부모) 무비 클립에서 재생 헤드가 위치하는 지점과 상관없이 항상 프레임 1에서 시작한다. 부모 무비 클립의 변수는 복제된 무비 클립으로 복사되지 않으므로 각각의 변수를 지정한다.

[무비클립 복제의 예]

```
duplicateMovieClip(target, newname, depth)
```

target은 복제하려는 무비 클립의 대상 경로이고, newname 복제된 무비 클립의 고유한 식별자이다. Depth는 복제된 무비 클립의 고유한 심도 레벨이다. 심도 레벨이란 복제된 무비 클립이 쌓이는 순서를 말한다. 이미 사용 중인 심도에 SWF 파일이 쌓이는 것을 막으려면 복제된 각 무비 클립에 고유한 심도 레벨을 지정한다.

(2) 텍스트 표현

동적 또는 입력 텍스트 필드는 TextField 만들 때 속성 관리자에서 이 필드의 인스턴스 이름을 지정한다. ActionScript 명령문에서 이 인스턴스 이름을 사용하여 텍스트 필드 및 필드의 내용을 설정하거나 변경하고 그 서식을 지정하여 사용자와 상호작용을 제공한다.

(3) 동적애니메이션을 위한 무비 클립

무비 클립은 서로 독립적으로 실행되는 내포된 소형 SWF 파일이며, 무비 클립을 포함하는 타임라인에 대해서도 독립적으로 실행된다. 모든 무비 클립 인스턴스는 인스턴스 이름을 가지고 있다. ActionScript에서는 이 이름을 통하여 무비 클립 인스턴스를 ActionScript로 제어 가능한 객체로 고유하게 식별한다. 애니메이션을 위한 무비 클립에 객체 속성의 prototype이란 속성을 통하여 메소드를 추가하여 다이나믹한 움직임을 구현한다. 다이나믹한 움직임을 구현하기 위해 cos 함수를 이용한다.

$$y = \cos(x) \quad (-1 \leq y \leq 1)$$

$$y = 1 + \cos(x) \quad (0 \leq y \leq 2)$$

$$y = (\text{distance}/2) * (1 + \cos(x)) \quad (0 \leq y \leq \text{distance})$$

즉, cos함수+1 의 최대값은 2가 되고, 거리/2 를 곱하므로, 최종값은 거리가 된다

x가 0부터 출발할 때, y는 최대값부터 시작하므로,

$$(\cos(0) + 1) * (\text{distance}/2) = \text{distance} \text{ 가 된다.}$$

여기서 x를 일정하게 증가시킬 때, 원하는 움직임을 얻기 위해서는 다음과 같은 식을 쓴다.

$$\text{현재위치} = \text{목표위치} - (\text{distance}/2) * (1 + \cos(x)) \quad (0 \leq y \leq \text{distance})$$

[실제 가속도가 붙은 것처럼 움직이는 애니메이션 만들기의 예]

```
MovieClip.prototype.Move = function(x,y){
    this.firstX = this._x;
    this.firstY = this._y;
    this.distX = x-this.firstX;
    this.distY = y-this.firstY;
    this.degree = 0;
    this.onEnterFrame = function(){
        this.degree+=this.speed;
        this.rad = (this.degree)*Math.PI/180;
        if(this.degree>=180){
            this._x = x;
            this._y = y;
            this.onEnterFrame=null;
            return;
        }
        this._x = x -
(this.distX/2)*(1+Math.cos(this.rad));
        this._y = y -
(this.distY/2)*(1+Math.cos(this.rad));
    }
}
```

(4) 색상값 설정

내장 Color 클래스의 메서드를 사용하여 무비 클립의 색상을 조절하여 선택값, 비교값, 범위 등을 시각적으로 구별하여 표현한다. setRGB() 메서드는 16진수 RGB(빨강, 녹색, 파랑) 값을 무비 클립에 지정한다.

[무비클립의 color 적용 예]

```
MovieClip.prototype.ChangeColor = function(col){
    var myColor = new Color(this);
    myColor.setRGB("0x"+col);
    delete myColor;
}
```

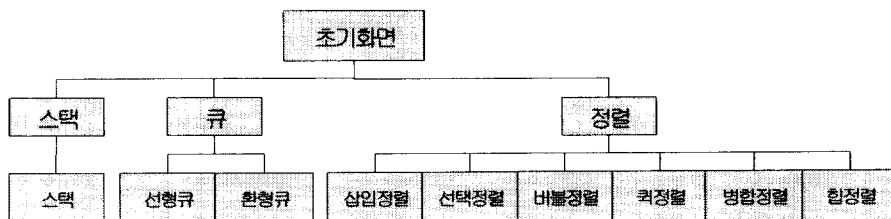

3.2 자료구조 학습

자료 구조는 일반적으로 자료를 일련의 순서를 갖고 나열하는 선형구조와 자료의 계층적 구조나 망형 구조 등으로 표현하는 비선형 구조, 그리고 컴퓨터의 주기억 공간 내에 저장된 레코드로 구성된 파일에서 레코드의 특정 부분을 키(KEY)로 하여 키값 따라 오름차순과 내림차순으로 레코드를 재배열하는 정렬로 구성된다.

선형구조는 자료를 구성하고 있는 자료들 상호간에 일대일 대응 관계가 성립하는 선형 형태의 자료구조로서 선형리스크, 연결리스트, 스택, 큐, 테크 등으로 구분된다. 비선형구조는 자료들 상호간에 일대다 또는 다대다의 대응 관계에 있는 자료 구조로서 계층적 구조의 트리나 망형 구조의 그래프 등으로 구분된다. 정렬은 정렬 장소에 따라 내부정렬과 외부정렬로 구분한다.

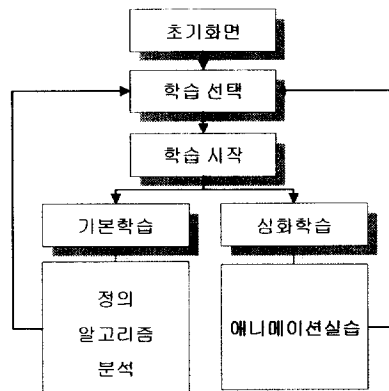
3.2.1 학습 내용

본 시스템의 학습 내용은 (그림12)와 같이 선형구조의 스택과 선형큐, 환형큐 그리고 정렬의 삽입정렬, 버블정렬, 선택정렬, 퀵정렬, 병합정렬, 힙정렬로 구성되었다.



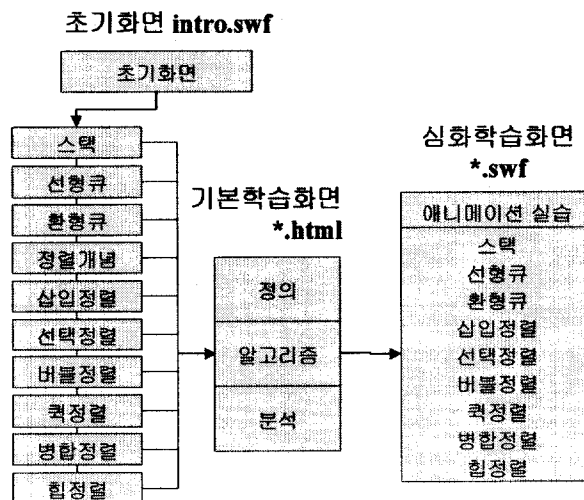
(그림 12) 학습내용

학습단계는 기본학습과 심화 학습으로 (그림13)과 같이 구성되어 기본학습에서는 해당 내용의 정의, 알고리즘, 분석을 텍스트를 통해 개념을 학습한다. 심화학습에서는 각 알고리즘의 상호작용 애니메이션을 통해 학습한다.



(그림 13) 학습단계

화면구성은 (그림14)와 같이 초기화면은 swf로 구성되고 기본학습화면은 html 문서로 심화학습은 swf파일로 구성된다.



(그림 14) 화면구성

4. 설계 및 구현

4.1 화면 설계

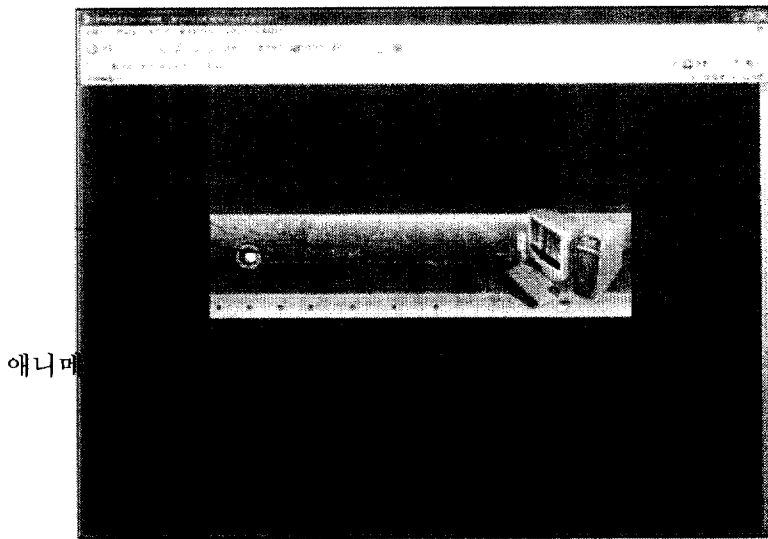
전체 화면의 디자인과 인터페이스를 통일하고 일관성 있는 레이아웃으로 학습자가 안정감을 가지고 학습에 임할 수 있도록 한다. 학습화면은 1024*768의 화면 해상도에서 인터넷 익스플로러의 표준 화면을 이용하고 다음과 같은 사항을 고려하여 학습자의 편의를 도모한다.

첫째, 부드럽고 조화된 색의 포맷을 통해 학습에 대한 집중력을 높인다.

둘째, 가능한 한 학습 내용을 한 화면 단위로 설계하여 스크롤 사용을 배제하고 학습자의 학습에 대한 인지적 부담감을 줄일 수 있도록 한다.

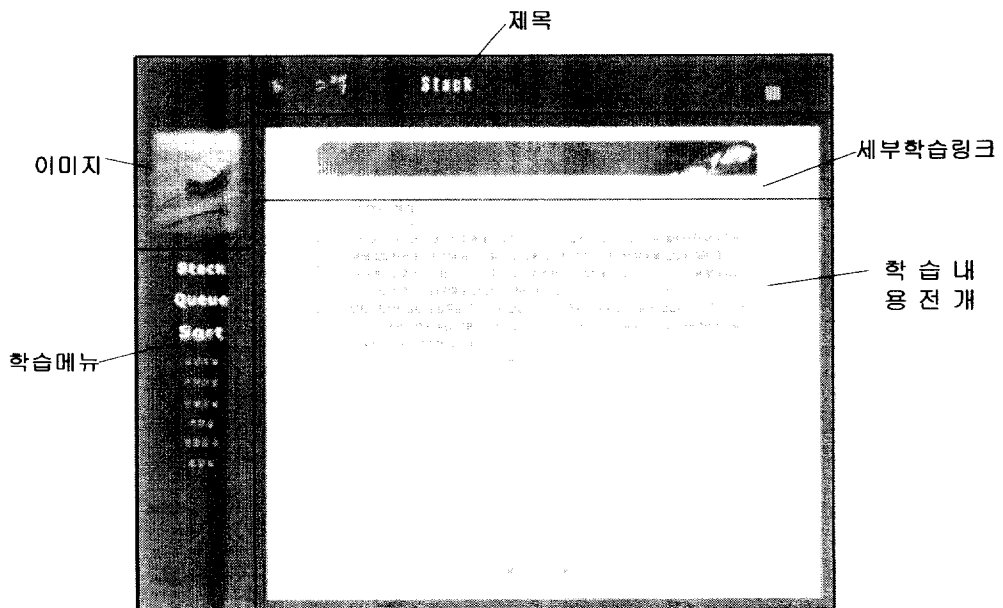
셋째, 버튼과 이미지를 가시화 하고 학습 화면을 구조화 하여 시각적인 혼돈을 없애고 현재의 학습 위치를 파악하는데 어려움이 없도록 한다.

본 시스템의 초기화면의 설계는 (그림 15), 기본학습화면 설계는(그림16), 심화학습화면의 설계는 (그림17)과 같다

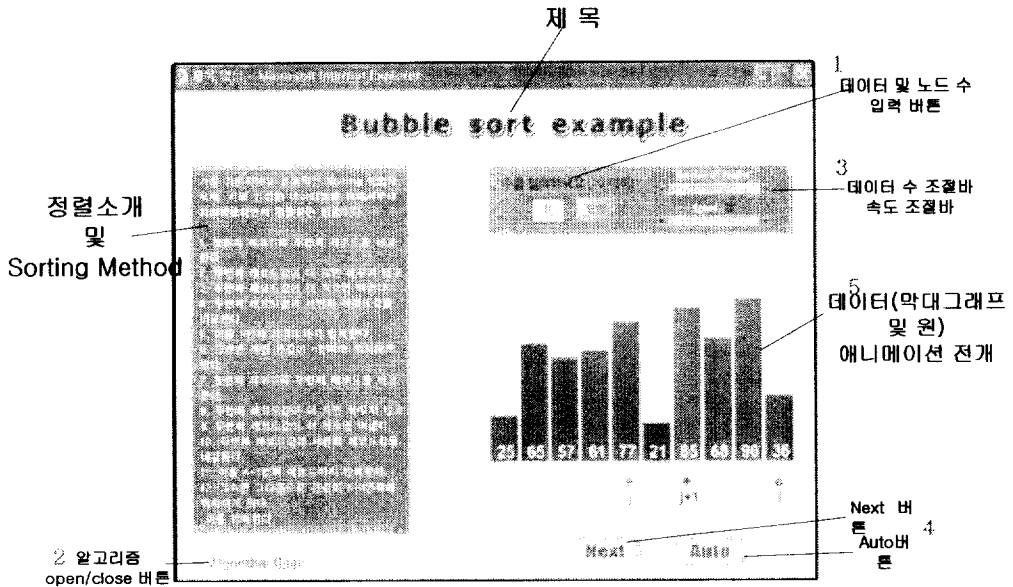


애니메이션

(그림 15) 초기 화면 설계



(그림 16) 기본학습 화면 설계



(그림 17) 심화학습 화면 설계

① 학습자가 데이터 또는 노드의 수를 입력해서 랜덤하게 다양한 값을 가지고 학습하게 한다.

② 알고리즘 소스의 on/off 버튼을 눌렀을 때 실제로 애니메이션 되는 부분과 동기화 되어 코드가 수행되는 단계별 진행과정과 데이터 변수 값을 직접 확인 할 수 있도록 설계하여 학습자가 선택할 수 있게 한다.

③ 학습자의 학습속도에 따라 애니메이션의 속도를 조절한다.

④ 자기 주도적 학습이 가능하도록 단계별(next)또는 자동진행(auto)을 선택하게 제작한다.

⑤ 칼라를 이용하여 알고리즘에서 선택된 데이터와 비교된 데이터의 값을 칼라로 구분하고 정렬된 데이터도 칼라로 구분하여, 선택범위도 box로 표현하여 시각적으로 쉽게 구별할 수 있게 한다.

4.2 개발 환경

본 연구에서 사용한 소프트웨어와 하드웨어의 프로그램 개발환경은 <표 2>와 같다.

운영체제는 WindowsXP이고 저작 언어는 Flash MX, HTML과 자바스크립트를 이용하였다. 하드웨어 환경은 CPU는 Pentium III, 주기억 장치는 512MB, 보조기억장치는40GB를 사용한다.

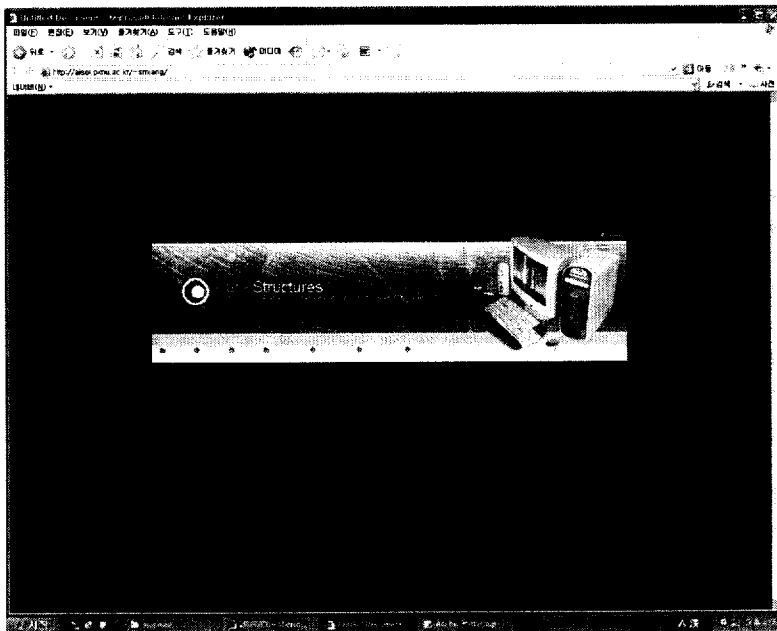
<표 3> 소프트웨어 및 하드웨어 환경

구분		사양
소프트웨어	운영체제	WindowsXP
	저작언어	Flash MX, HTML과 자바스크립트
	웹브라우저	Explorer 6.0
하드웨어	중앙처리장치	Pentium III
	주기억장치	512MB
	보조기억장치	40GB

4.3 구현

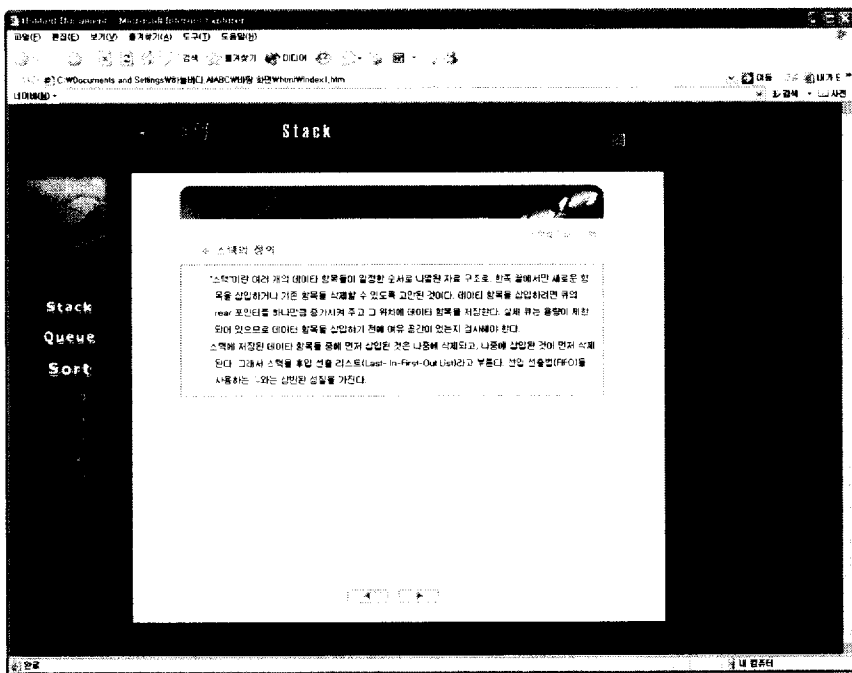
본 시스템의 구현은 해당 내용을 데이터 값의 실제 이동을 동적 애니메이션으로 구현하였고, 알고리즘의 단계적 진행과정을 시각적으로 표현하였다. 또한 학습도중이라도 학습자가 언제든지 재설정하여 학습할 수 있도록 하였다. 중요한 부분은 칼라나 아이콘을 사용하여 강조하며, 버튼은 시각적으로 쉽게 인식되도록 단순 명료한 것을 사용하였다.

본 시스템에 접속하게 되면 (그림 18)와 같이 인트로 화면을 볼 수 있으며 플래시로 역동적이고 경쾌한 사운드로 제작하여 학습자의 적극적인 참여를 유도할 수 있다.



(그림 18) 초기 화면

인트로 화면이 끝나면 자동으로 (그림 19)과 같이 메인 화면으로 연결된다. 메인화면은 크게 세개의 프레임으로 나누어 구성된다. 왼쪽 프레임 상단은 이미지를 나타내고 하단은 학습메뉴로 구성되어 해당 학습 내용으로 이동할 수 있다. 오른쪽 상단은 제목을 플래시 텍스트 애니메이션으로 보여준다. 메인 프레임에서는 학습 화면으로 구성되어 학습내용이 전개된다.



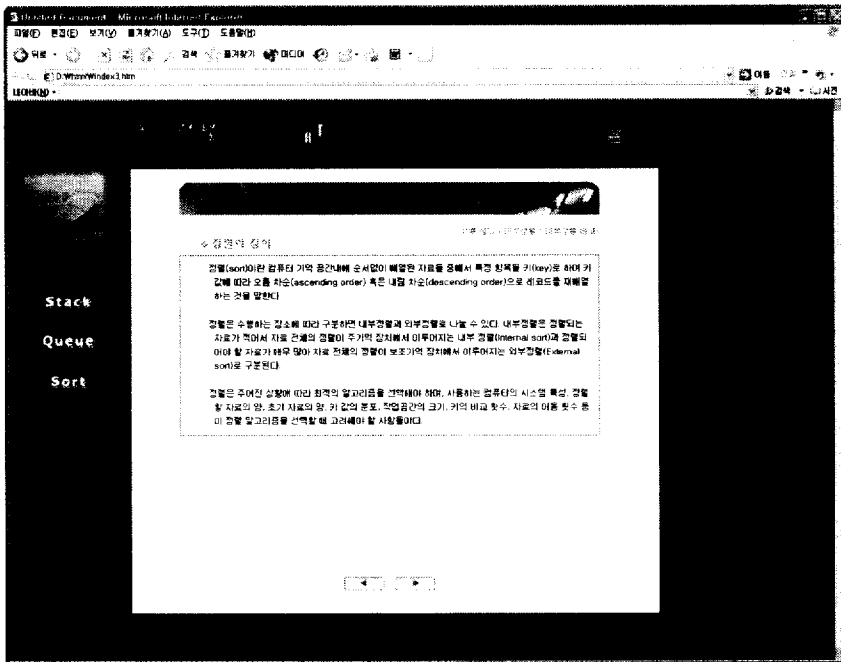
(그림 19) 메인 화면

오른쪽 프레임 학습 메뉴에서 버튼을 선택하면 해당 학습이 (그림 20)과 같이 나타난다.

기본학습화면의 오른쪽 상단에는 각각의 정의, 알고리즘, 분석이 서브메뉴로 링크되어 학습 도중 원하는 학습 내용으로 보다 쉽게 이동 할 수 있게 하였고, 아래쪽에는 이전과 다음의 이미지 버튼을 이용하여 한 화면 단위로 이동하면서 학습할 수 있고 세부학습 사이도 연결시켜서 순차적으로

이동할 수 있도록 하였다.

학습 화면은 가능한 한 스크롤을 배제하고 학습자가 편안하게 학습할 수 있도록 하나의 화면 단위로 구성하였다.

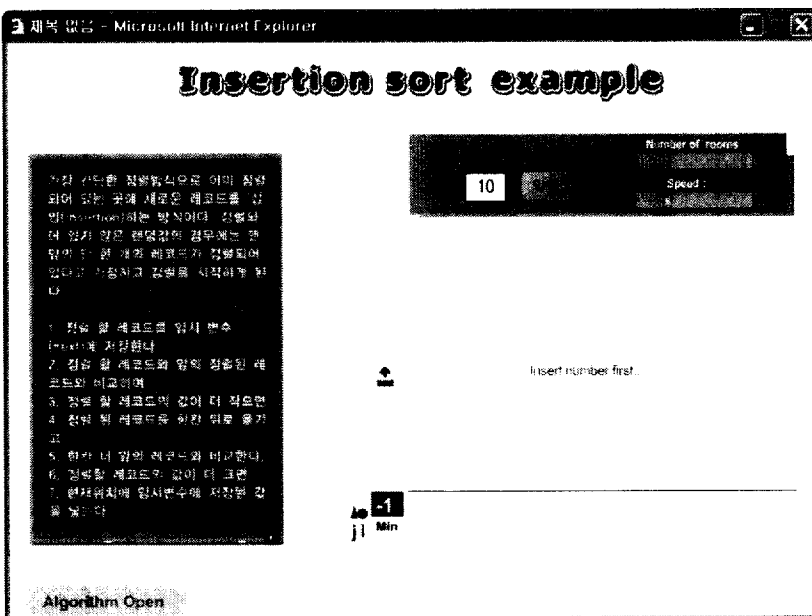


(그림 20) 기본학습 화면

자료구조 알고리즘을 상호작용 애니메이션을 이용하여 쉽게 이해 할 수 있도록 팝업창으로 구성하여 심화학습 화면을 (그림 21) 과 같이 구현하였다

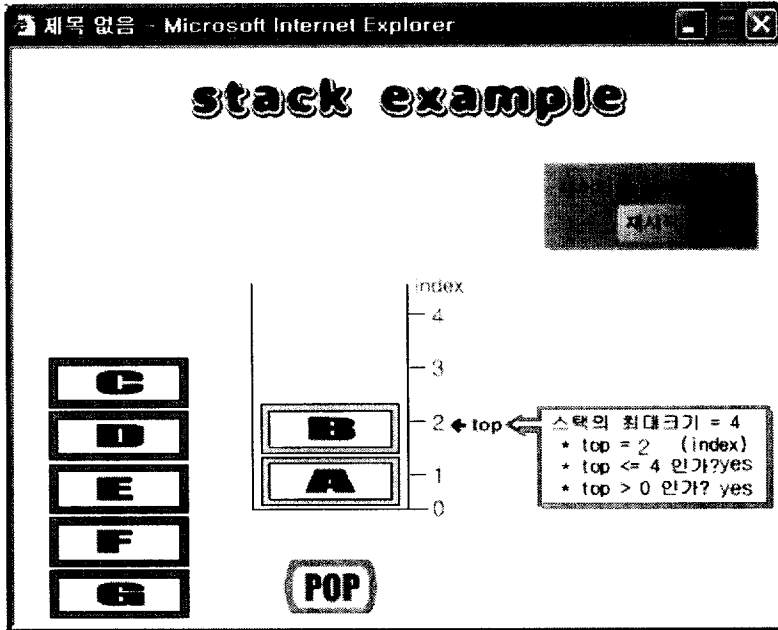
애니메이션 자료는 해당 내용을 데이터를 이용해서 데이터 값의 실제 이동과 단계적 진행과정을 알고리즘 소스의 진행과정과 함께 동기화하여 시각적으로 표현하였고, 학습도중이라도 학습자가 언제든지 초기화 하여 설정을 재설정하여 재학습 할 수 있도록 하였다. 데이터 및 노드수는 학습자

가 주관적으로 설정하고, 동적 애니메이션의 속도도 학습자의 학습속도에 맞게 설정할 수 있다. 각 화면은 일관된 서식을 적용하여 작성하고 개념과 원리를 이해하기 쉽도록 이미지와 동적 애니메이션을 이용해서 표현하였고, 중요한 부분은 색상을 달리하거나 아이콘을 사용하여 강조하며, 버튼은 시각적으로 쉽게 인식되도록 단순 명료한 것을 사용하였다.



(그림 21) 심화학습 화면

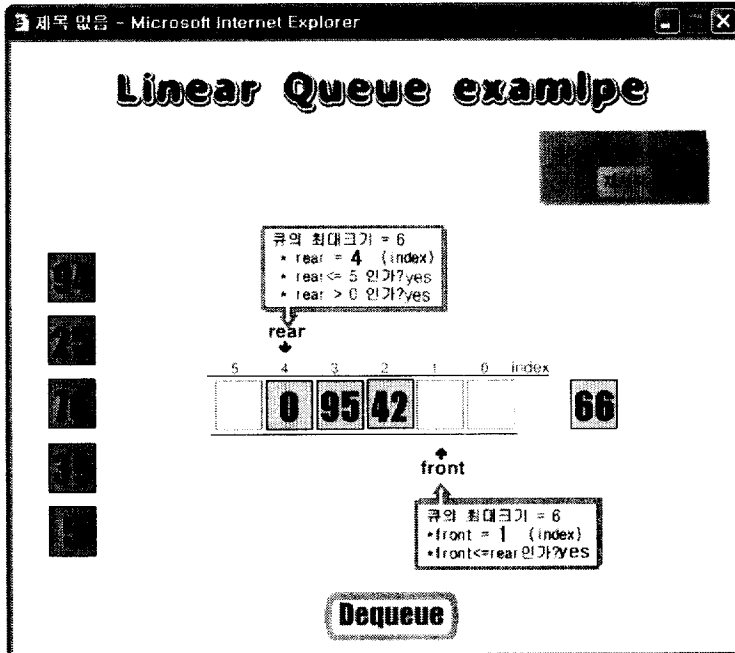
스택은 (그림 22)와 같이 기저(base)로부터 데이터 항목들을 차례로 쌓아올린 모양을 가지며, 이를 클릭하면 삽입이 되고, pop 버튼을 클릭하면 삭제가 된다. 현재 저장된 최상위 항목을 스택 포인터라는 지시자 top이 위치를 가리킨다.



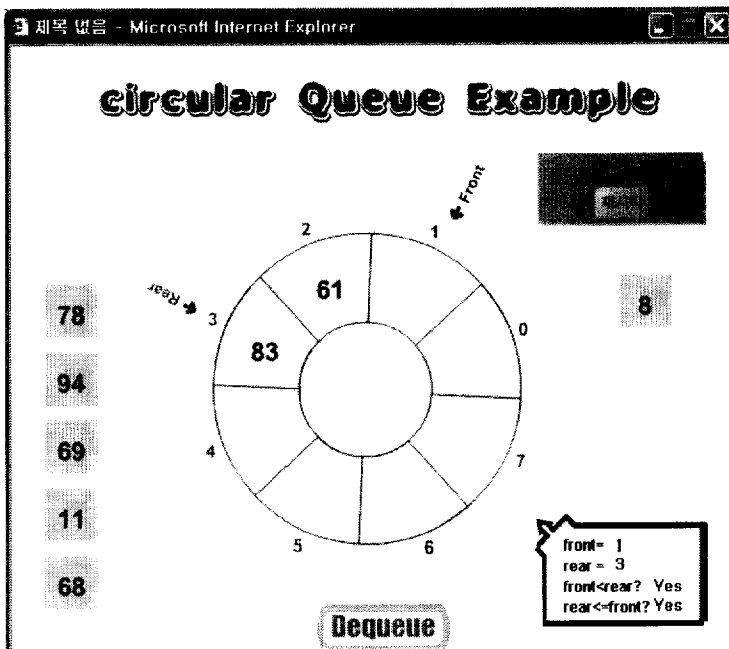
(그림 22) 스택

선형 큐는 (그림 23)과 같이 여러 개의 데이터 항목들이 차례로 저장되는 모양을 가진다. 새로운 데이터 항목의 삽입은 rear에서 일어나고, 이 위치는 rear 포인터라는 지시자가 가리킨다. 데이터 항목의 삭제는 front에서 일어나며, 이 위치는 front 포인터라는 지시자가 가리킨다.

환형 큐는 큐 구조가 (그림 24)과 같이 원형으로 구성되어 rear 포인터 값과 front 포인터를 변화시켜 시각적으로 표현한다.



(그림 23) 선형 큐

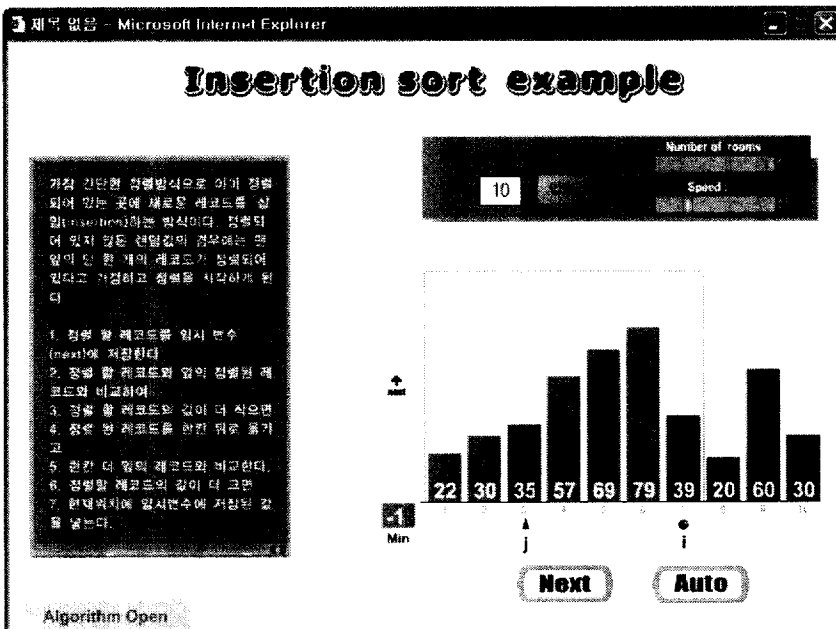


(그림 24) 환형 큐

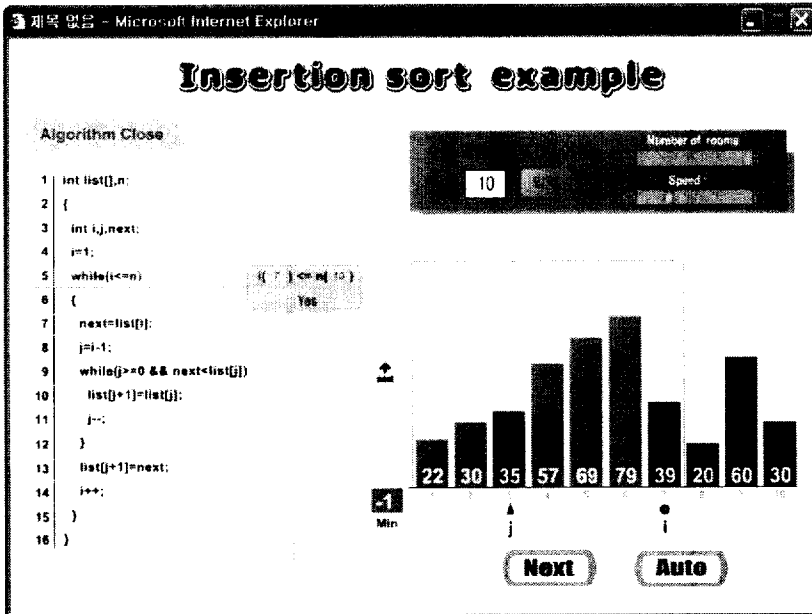
삽입정렬 알고리즘에서 사용된 변수값을 i, j 의 화살표 아이콘과 색상으로 구분하여 표현하였고, 선택된 데이터와 비교된 데이터의 값을 화살표와 같은 색상으로 표현하였다. 그리고 정렬된 데이터도 색상으로 구분하고, 선택 범위도 box로 표현하여 시각적으로 쉽게 구별할 수 있다. 또한 데이터의 크기를 시각적으로도 확인할 수 있도록 (그림 25)와 같이 구현했다.

알고리즘 open/close 버튼 선택 여하에 따라 학습도중이라도 언제든지 학습자가 알고리즘 소스의 진행 과정과 데이터 값의 변화를 애니메이션 진행과 동기화하여 (그림 26)과 같이 직접 눈으로 한 단계씩 확인하며 진행할 수 있다. 선택정렬, 버블정렬도 삽입정렬과 같은 원리로 표현했다.

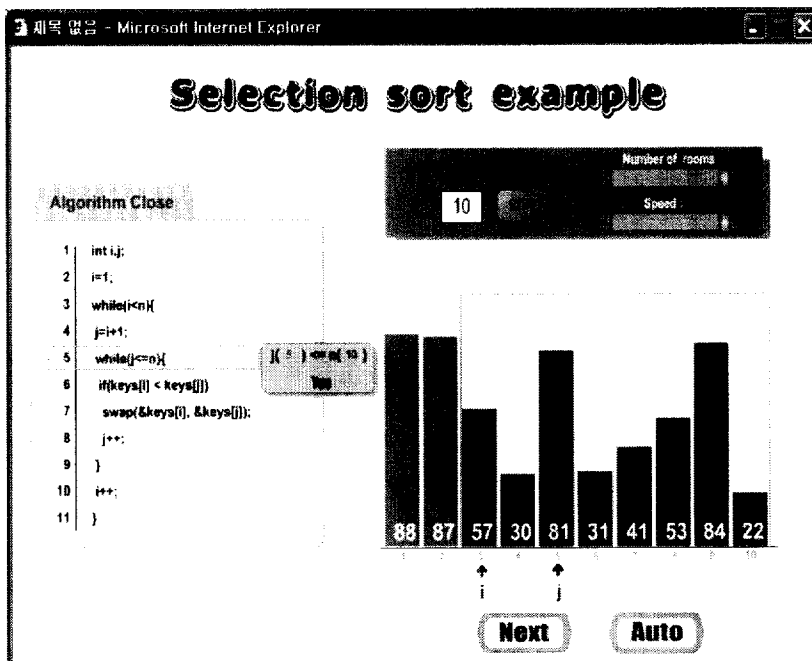
선택정렬은 (그림 27)과 같고, 버블정렬은 (그림 28)과 같다.



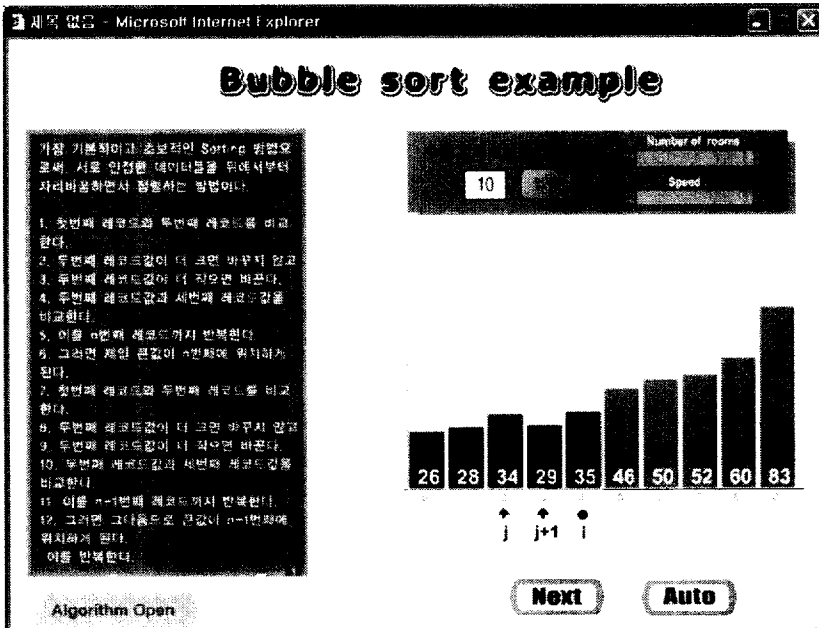
(그림 25) 삽입정렬(1)



(그림 26) 삽입정렬(2)



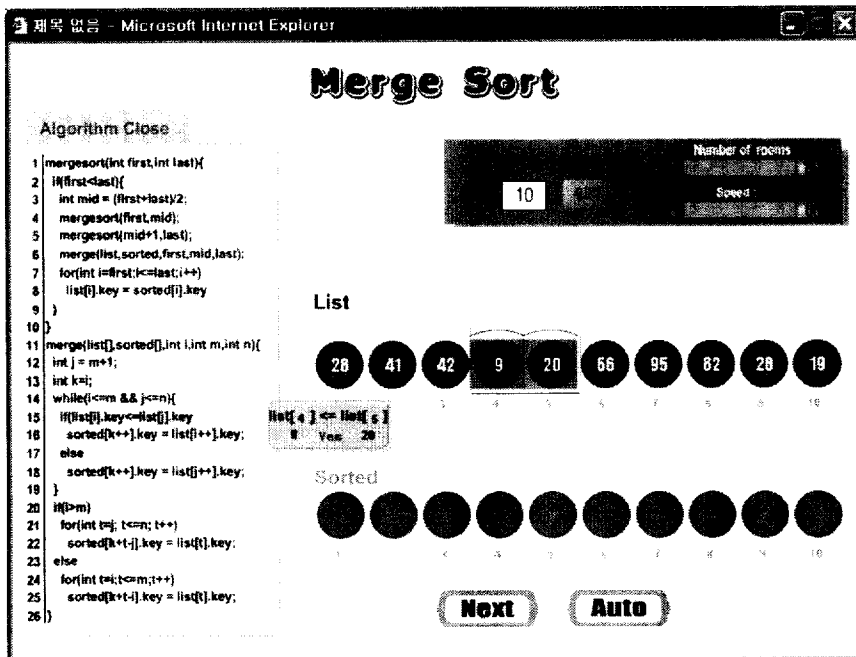
(그림 27) 선택정렬



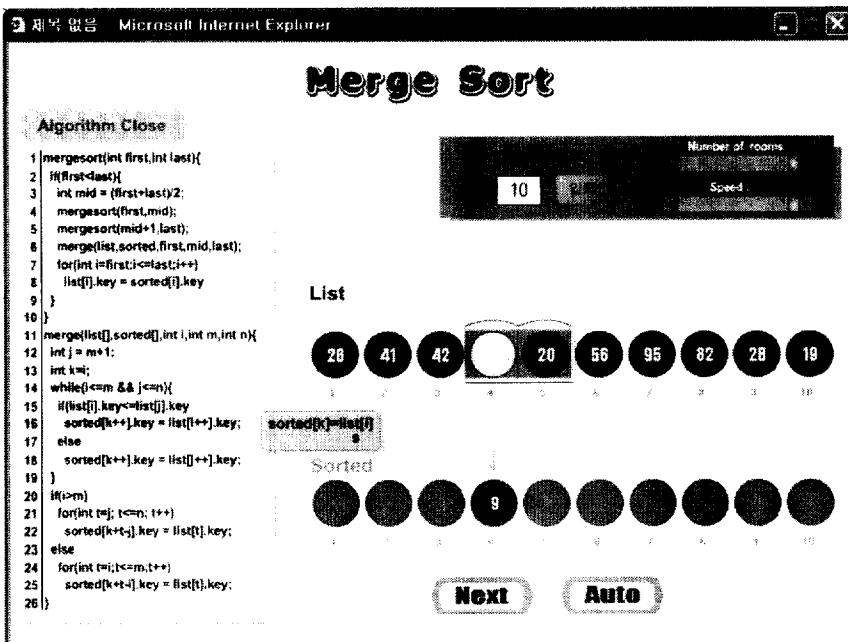
(그림 28) 버블정렬

퀵정렬은 재귀호출을 하는 과정을 동적 애니메이션으로 표현하였기에 선택정렬, 삽입정렬, 버블정렬 애니메이션을 표현한 과정에 추가적으로 재귀호출하는 과정을 depth를 이용하여 표현하였다.

재귀호출을 할 때마다 depth가 한 단계씩 내려가고, 지나간 단계의 범위를 표시하고 현재 진행 중인 범위를 색상으로 구분해서 표현하여 전체 몇 번의 재귀호출이 어느 범위에서 발생하였는지를 시각적으로 (그림 29)과 같이 표현하였다.

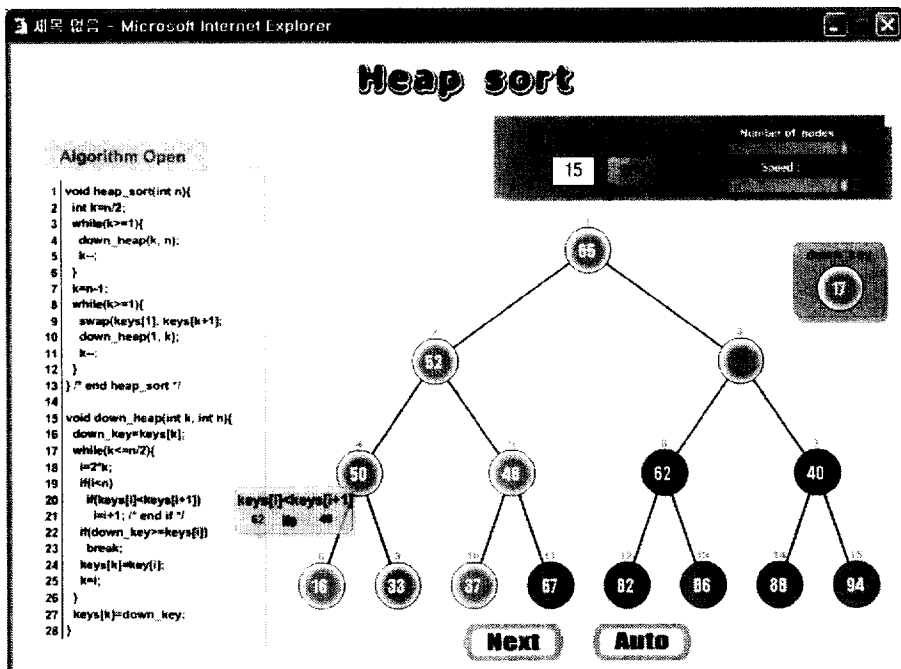


(그림 30) 병합정렬(1)

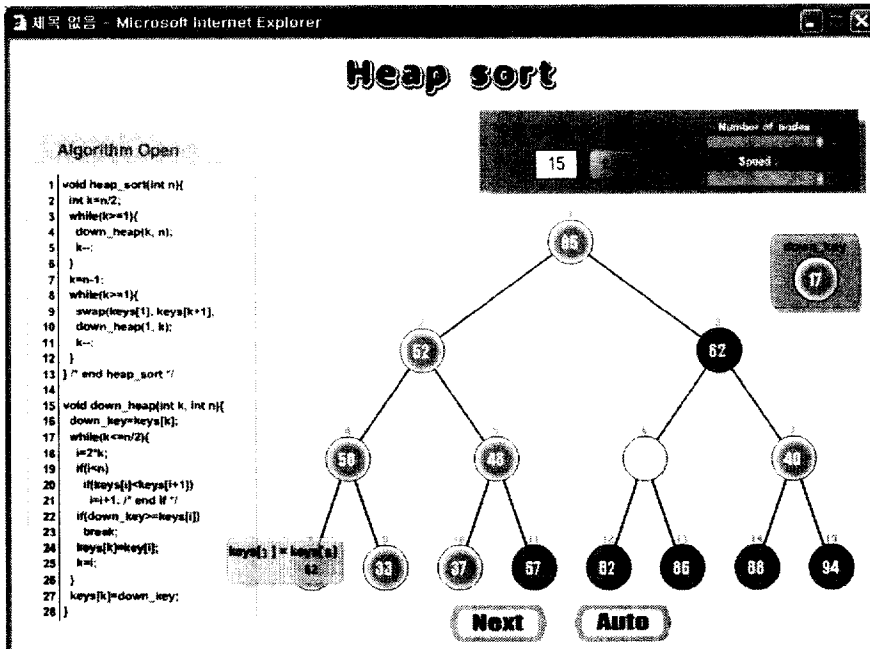


(그림 31) 병합정렬(2)

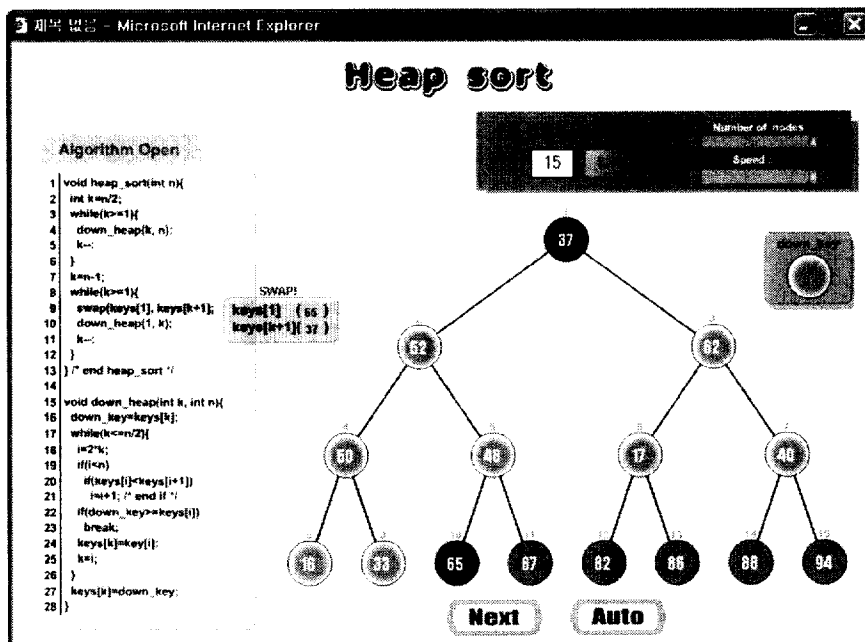
힅정렬은 완전이진트리인 (그림 32)과 같이 표현하였다. 병합정렬과 같이 비교되는 두 데이터는 파란색으로 표시하고, 비교 후 선택된 데이터가 넘겨주는 값을 노란색으로 그 값을 받는 데이터는 분홍색으로 (그림 33)과 같이 표현하였다. 서로 데이터를 swap할 때는 보라색으로 표현하였고 정렬이 끝난 데이터는 회색으로 (그림 34)과 같이 표현하여 시각적으로 쉽게 구분할 수 있게 표현하였다.



(그림 32) 힅정렬 (1)



(그림 33) 힙정렬 (2)



(그림 34) 힙정렬 (3)

5. 평가

본 학습 시스템과 같은 액션스크립트로 구현한 기존 학습시스템 ‘2004 자료구조학습을 위한 자기주도적 코스웨어 설계 및 구현’과의 비교를 <표 4>와 같이 한다.

상호작용 애니메이션 이용한 본 학습 시스템은 알고리즘 소스 코드를 분석하고 수행되는 과정을 단계별 진행 과정과 그에 따른 데이터 값의 변화를 동기화 시키고 칼라로 구분하여 시각적으로 알기 쉽게 구현하여 학습자가 직접 소스 코드를 보면서 사고할 수 있고 이해할 수 있도록 하여 학습에 대한 흥미 유발 및 학습 이해력에 도움을 주어 자기 주도적 학습이 가능하다.

<표 4> 기존 학습 시스템과의 비교

본 학습시스템		2004 자료구조학습을위한자기주 도적코스웨어설계및구현
프로그래밍언어	액션스크립트	액션스크립트
알고리즘소스와의 동기화	애니메이션 진행과 알고리즘소스 진행의 동기화	애니메이션만 진행
데이터 수	학습자가 설정	교수자가 미리 설정
데이터 값	랜덤	고정
애니메이션표현	다이나믹	단순
애니메이션 진행중 속도조절	가능	불가능
칼라 이용한 데이터 구분	가능	불가능

6. 결론

교육 환경의 변화로 초고속 정보통신망과 멀티미디어를 활용한 교수 학습 환경이 요구 되고 있으며 컴퓨터의 교육적 활용에 있어서 가장 뛰어난 장점중의 하나는 학습의 개별화를 촉진시킬 수 있다는 것이다. 이러한 시대적 요구에 따라서 웹 기반 학습 시스템은 필수적이라 할 수 있다. 컴퓨터 자료구조 알고리즘은 프로그래밍에 익숙해지고자 하는 모든 학생들에게 필수적인 과목이다. 따라서 자료구조는 개념적이고 추상적인 내용으로 텍스트만으로는 이해하기 어렵다는 점을 고려하여 시간과 공간의 제약에서 벗어나 학습자의 요구에 따라 자신의 능력에 맞는 개별적인 자료구조를 학습할 수 있고, 이해를 쉽게 진작시킬 수 있는 자기 주도적 학습 시스템의 개발이 필요하다.

본 연구는 자료구조 알고리즘의 학습을 위한 액션스크립트(ActionScript) 기반의 상호작용 애니메이션을 이용한 학습 시스템으로 학습자에게 자료구조 알고리즘 내용을 시각적으로 쉽게 이해할 수 있도록 하였으며, 일관된 서식을 제공하고, 학습자의 학습 속도에 맞는 학습 진행에 효율적인 학습 환경을 제공하여 주어진 데이터를 이용해서 과제를 해결하는 것뿐만 아니라 학습자가 원하는 데이터를 직접 입력해서 결과를 얻을 수 있는 형태로 다양한 값에 대한 실습이 가능하도록 하여 보다 적극적인 학습자의 참여를 이끌어 내어 자기주도적 학습이 가능하게 하였다.

본 연구를 바탕으로 다음과 같이 제언을 하고자 한다. 학습자의 수준에 따라 알고리즘을 보다 쉽게 이해할 수 있도록 음성이나 소리를 활용한 구현이 이루어져야 하겠다.

참고문헌

- [1] Porter, Brandon. Educational Fusion : An Instructional, Web-based Software Development Platform, MIT Thesis, 1998
- [2] 추교흠 외 2 , 교수중심의 웹 기반 평가시스템 설계 및 구현, 한국 정보처리학회 '99춘계 학술발표논문집, pp737-740
- [3] 인영아, "Flash 학습을 위한 웹 기반 시뮬레이션형 학습 모형 설계 및 구현", 전남대학교 교육대학원 석사학위논문, 2004
- [4] Brown, M. H. & Sedgewick, R.(1985). Techniques for Algorithm Animation. IEEE S of tware, 2(1) :28- 39.
- [5] Stasko, J.T.(1990). Tango : A Framework and System for Algorithm Animation. IEEE Computer, 23(9): 27- 39.
- [6] <http://www.cc.gtech.edu/gvu/softviz/arviz/polka.html>
- [7] <http://www.cc.gatechedu/gvu/softviz/algoanim/samba.htm> l
- [8] Susan H. Rodger, "Using Hands-On Visualizations to Teach Computer Science from Beginnings Courses to Advanced Courses, Second Program Visualization Workshop, Hornstrup Center, Denmark. 2002.6
- [9] 이향숙, 이수현, "알고리즘 시각화를 위한 SVG 코드 생성기의 설계 및 구현", 한국정보처리학회 추계학술대회 논문집 제 11권 제2호, 2004
- [10] 민경혜, "자료구조 학습을 위한 자기 주도적 코스웨어 설계 및 구현", 한국정보과학회 봄 학술발표논문집(B), p.661-663, 2004
- [11] Guido RoBling and Bernd Freisleben, "Approaches for Generating Animations for Lectures," AACE 11th International Society for

Information Technology and Teacher Education Conference, pp.
809-814, 2000. 2.

- [12] Tooru Ueno 지음, 최용석·IT사업팀 옮김, “FLASH 액션스크립트
바이블”, 성안당, 2001.