

알고리즘 교육 시각화를 위한 셀프-애니메이션 프레임워크

민수정* · 송병욱* · 김상욱*

1. 서론

원격 가상 교육이 이루어지는 과정에서 학습 자료와의 상호 작용을 제공하여 학습자의 능동적인 참여를 이끌어낼 수 있는 기능이 중요하다.[1] 특히, 프로그램의 실행 원리나 자료구조와 알고리즘의 동작 원리 등을 학습하는 경우, 학습자의 직접적인 조작에 의해서 그 원리의 동작 과정을 시각화하는 것이 보다 효과적이다[2]. 이는 학습자의 능동적인 참여를 유도하고 개념의 가시화를 통해서 학습효과를 높일 수 있다[3]. 따라서, 원격 가상 교육 환경에 적합한 알고리즘 시각화 시스템에 대한 연구가 중요하다[4,5].

알고리즘 교육 시각화는 알고리즘에서 사용되는 객체들을 시각적으로 표현하고 그것의 위치와 모양 등을 부드럽게 변화시켜 알고리즘의 실행 과정을 보이는 것이다[6]. 일반적으로 기존의 알고리즘 교육 시각화 시스템들[5,7]은 설계자에 의해 알고리즘의 동작 과정과 자료가 고정된 기성(ready-made) 애니메이션을 제공한다.

알고리즘 교육에서 학습자가 직접 작성한 알고리즘을 시각화하여 자신의 생각과 실제 동작을 비교하는 과정이 필요하다[8]. 그리고 이러한 시각화 과정은 알고리즘 소스 코드의 타이핑을 제외

한 여분의 노력 없이 직관적인 절차에 의해 이루어져야 한다. 이를 위해 알고리즘의 시각적 표현을 생성하는 과정에서 애니메이션 되는 객체의 속성과 패스의 세부적인 사항은 고려하지 않도록 투명성이 보장되어야 한다. 즉, 세부적인 사항은 시스템 자체가 제공하도록 하여 객체를 시각화하는 방법에 대해서는 고려하지 않고 시각화되는 객체들만을 고려할 수 있도록 해야 한다.

알고리즘은 사용하는 자료 구조에 따라 다른 동작 과정을 가진다. 따라서 각각의 자료 구조에 맞는 애니메이션을 구현해야 하므로 알고리즘 교육 시각화는 노동 집약적인 작업이다[9]. 이러한 작업에서 투명성을 보장하고 애니메이션 생성 과정을 자동화하기 위해서는 알고리즘 시각화를 지원하는 프레임워크가 필요하다. 프레임워크는 알고리즘 코드와 애니메이션 부분을 분리하여 처리해야 하며 다양한 자료 구조를 지원해야 한다.

본 연구에서는 알고리즘의 동작 형태를 시각적으로 표현하는 애니메이션이 자동적으로 생성되는 것을 지원하는 셀프-애니메이션 프레임워크를 제안한다. 본 프레임워크는 알고리즘 내에서 사용되는 트리나 그래프 등의 자료 구조에 대해 그 모양과 동작을 시각적으로 표현하는 클래스인 시각 자료 구조를 제공한다.

본 프레임워크는 다양한 자료 구조의 시각화를 지원하며 단일 자료 구조에 대한 다중 표현과 단

* 경북대학교 컴퓨터학과

일 알고리즘에서 다중 자료 구조의 사용을 지원한다. 또한, 마우스 이벤트를 통하여 알고리즘의 애니메이션에서 학습자가 직접 자료를 조작한다.

2장에서는 본 연구와 관련된 연구를 살펴보고 3장에서는 가상 교육 환경에 적합하도록 설계된 알고리즘 교육 시각화 시스템의 구조와 동작을 설명한다. 그리고 4장에서는 알고리즘의 시각화를 지원하는 셀프-애니메이션 프레임워크의 구성을 살펴본 후, 시각화를 관리하는 방법을 설명한다. 5장에서는 알고리즘 교육 시각화 시스템을 위해 실제 구현된 시각 자료 구조를 설명하고 시스템의 결과를 보이며, 마지막으로 6장에서 결론을 맺는다.

2. 관련 연구

알고리즘 시각화 기법에는 사전정의(predefinition) 방법, 함수 호출(annotating) 방법, 선언(declaration) 방법, 조정(manipulation) 방법 등이 있다[10]. 기존의 시스템에서는 함수 호출 방법 [7,9]과 선언 방법[11]을 주로 사용한다.

함수 호출 방법은 알고리즘 코드에서 필요한 곳마다 일반화된 시각화 함수인 애니메이션 프리미티브를 호출하여 알고리즘에서 일어나는 특정한 행위를 애니메이션으로 나타낸다. 선언 방법은 알고리즘을 시각화하는데 사용되는 그래픽 객체 집합을 제공한다. 알고리즘에서 실행되는 상태가 그래픽 객체와 매핑되어 있고 알고리즘이 실행되는 중에 변화된 상태는 대응되는 그래픽 객체를 변화시켜 시각화한다.

함수 호출 방법은 알고리즘 소스 코드를 수정해야 하며 애니메이션 장면을 위한 코드를 작성해야 한다. 또한, 알고리즘에서 애니메이션 될 부분을 직접 명시해줘야 하는 단점이 있다. 선언 방법은 알고리즘 코드와 애니메이션 코드가 분리되어

있지만, 알고리즘의 상태와 그래픽 객체를 매핑하기 알고리즘 소스 코드의 많은 부분을 수정해야 한다. 그리고, 모든 알고리즘에 일반적으로 적용하기 어렵기 때문에, 선택된 알고리즘에 적절한 그래픽 객체를 새로이 설계해야 한다. 이런 단점들은 시스템의 사용을 어렵게 만들고 새로운 자료 구조를 지원하기 위한 시스템의 확장을 제한한다.

셀프-애니메이션 기법[8]은 함수 호출 기법의 단점을 보완한 것으로 알고리즘에서 사용하는 int, 배열 등의 자료형에 그 자료형을 시각적으로 표현하기 위한 기능을 더한 시각 자료형을 제공한다. 시각 자료형 연산에 일반 자료형의 연산과 그 연산에 해당하는 애니메이션이 포함되어 있기 때문에 알고리즘이 시각 자료형을 선언하여 사용하면, 알고리즘은 자동적으로 애니메이션 된다.

Haajanen가 제안한 알고리즘 시각화 시스템인 Eliot[4]는 입력된 알고리즘을 자동적으로 애니메이션하며 원격 가상 교육에 적용되도록 발전시킨 JEliot 시스템이 있다. 셀프-애니메이션 기법을 사용하여 시각 자료형을 구현한 Eliot 클래스 라이브러리를 제공하며, 시각화를 위해 Stasko의 알고리즘 애니메이션 라이브러리인 Polka의 애니메이션 프리미티브를 사용한다. Eliot 클래스 라이브러리에 새로운 시각 자료형을 추가하기 위해서는 라이브러리를 구성하고 있는 다수의 클래스들을 수정하여야 하는 단점이 있으며, 다중 스테이지간의 동기화가 부족하고 애니메이션이 이루어질 때 자료의 직접적인 조작을 지원하지 않는다.

Lafollette은 C/C++ 프로그램 소스 코드의 간단한 신택스 변화를 통하여 C/C++ 프로그램을 시각화하는 프로토타입 소프트웨어 시각화 시스템 [10]을 제안하며 C/C++에서 사용되는 기본 자료형에 대응하는 셀프-애니메이팅 자료형의 집합을 제공한다. 이는 C/C++에서 int에 해당하는 INI 형과 pointer에 해당하는 PTR형을 제공하며 이를

이용한 리스트와 트리를 시각화한다. 하지만 역시 제한된 자료형을 지원하고 있으며 프레임워크의 개념이 없기 때문에 새로운 자료형의 추가가 어렵다. 또한 워크스테이션 환경에서 개발되어 원격 가상 교육에 사용하기는 적합하지 않다.

3. 알고리즘 교육 시각화 시스템

알고리즘 시각화는 프로그램의 구조, 제어 흐름, 자료 구조의 상태 변화 등을 그래픽으로 표현하는 기법으로, 알고리즘의 실행 과정을 그래픽 객체들의 애니메이션으로 시각화하여 사용자에게 보여주는 방법이다[11].

알고리즘 교육 시각화 시스템은 원격 가상 교육 환경에서 학습자의 알고리즘을 애니메이션으로 생성하고 학습자에게 뷰어를 통해 애니메이션에 대한 접근과 제어를 제공하는 시스템이다. 이는 시각화를 지원하는 시각 자료 구조 라이브러리, 알고리즘을 시각화시키는 애니메이션 생성기, 애니메이션을 사용자에게 보여주는 애니메이션 뷰어로 이루어진다. 시각 자료 구조 라이브러리는 시스템에서 요구하는 알고리즘의 애니메이션과 알고리즘이 애니메이션 될 때의 사용자 상호작용을 지원한다. 애니메이션 생성기는 알고리즘 입력기, 시각화 인터프리터, 컴파일러 연결기로 구성되며 사용자 인터페이스를 통해 학습자가 직접 작성한 알고리즘을 실행 가능한 애니메이션으로 변환한다. 애니메이션 뷰어는 생성된 애니메이션을 학습자의 요구에 따라 실행한다. 그림 1은 시스템의 전체 구조를 보여준다.

여기에서 시각 자료 구조 라이브러리는 여러 가지의 시각 자료 구조 클래스와 시각 자료 구조의 관리를 위한 클래스를 제공한다.

시각 자료 구조는 알고리즘 소스 코드에서 선언된 후 그것이 제공하는 메소드를 사용하도록

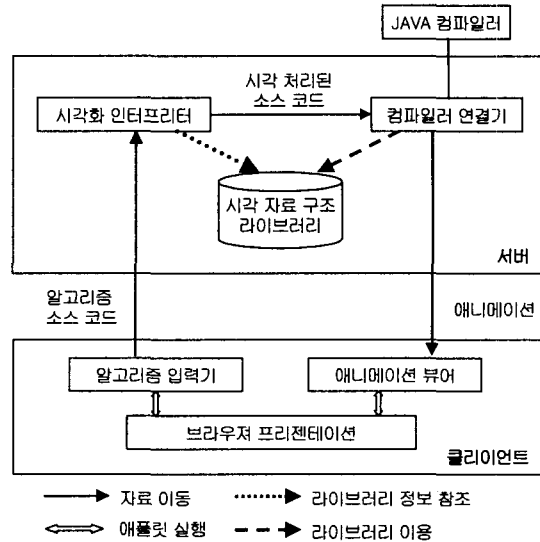


그림 1. 시스템 구조

한다. 또한 알고리즘 소스 코드는 시각 자료 구조를 관리하는 클래스의 일부가 된다. 시각 자료 구조를 알고리즘에 적용한 예는 그림 2와 같다. Visual_Tree는 트리에 대한 시각 자료 구조이며, 그것을 관리하는 클래스가 VisualTreePanel이다. 이를 일반화하여 알고리즘 패널 클래스라고 한다.

4. 셀프-애니메이션 프레임워크

4.1 프레임워크의 구성

제안한 프레임워크는 객체 지향 방법론을 기반으로 구성된다. 이는 클래스 계층 구조를 이용하여 클래스 재사용을 효율적으로 지원한다. 하위 클래스를 생성함으로써 기존의 클래스에 대한 수정 혹은 확장이 용이하다. 따라서, 다양한 시각 자료 구조를 용이하게 생성한다. 상위 계층에서 제공한 기본적인 시각화 기능에 특정 시각 자료 구조를 위한 기능을 첨가하거나 기능을 재정의함으로써 제작 과정에 독립성을 높인다.

시각화 지원 클래스 계층은 알고리즘 시각화

```

class VisualTreePanel extends ViewManager {
    VisualTree visual_tree;

    VisualTreePanel() {
        visual_tree = new VisualTree(this);
        // define Other Views
        // define Control Button
    }

    private void preOrderTraverse() {
        preOrderTraverse(visual_tree.getRootNode());
    }

    private void preOrderTraverse( VisualTreeNode
        visual_treeNode ) {

        if (visual_treeNode != null) {

            visual_tree.visit(visual_treeNode);

            preOrderTraverse(visual_treeNode.getLeftChild());
            preOrderTraverse(visual_treeNode.getRightChild());
        }
    }
    // Some modules for controlling animation
}
    
```

그림 2. 시각 자료 구조의 적용 예

에 기본적인 기능을 지원하며, 시각 자료 구조들의 공통적인 속성들의 시각화를 관리한다. 자료엘리먼트 클래스는 자료 엘리먼트의 상태에 대한 정의와 위치, 크기, 모양, 상태를 관리하는 메소드, 드로잉 메소드를 가진다. 뷰 클래스는 코드뷰 클래스와 컨셉뷰 클래스가 상속하는데 코드뷰 클래스는 알고리즘 소스 코드를 알고리즘의 실행에 대응하여 보여주기 위한 것이며 컨셉뷰 클래스는 셀프-애니메이션 프레임워크에서 가장 핵심적인 클래스로써 시각화가 이루어지는 뷰와 뷰에서 시각화되어지는 자료 엘리먼트를 관리한다. 마지막으로 뷰관리자 클래스는 뷰 클래스를 시나리오를 이용하여 관리하며 여러 개의 뷰 클래스가 있을 경우, 그들 사이의 동기화를 지원한다.

시각 객체 클래스 계층은 시각화 지원 클래스 계층을 확장하여 추상적인 시각 자료 구조를 구현한다. 자료구조엘리먼트 클래스는 시각 자료 구조 내에서 사용되는 각각의 구성 요소에 해당하며 특정한 자료 구조의 특성에 알맞은 크기, 모양을

가진다. 자료구조뷰 클래스는 뷰에 자료 구조가 표현되는 모양을 설정하고 자료 구조 엘리먼트를 관리한다. 자료구조뷰 클래스를 상속한 시각자료 구조 클래스는 시각 자료 구조가 가지는 생성, 삭제, 이동, 교환, 방문 등을 시각화하는 연산을 구현한다. 알고리즘패널 클래스는 기존 자료 구조를 시각 자료 구조로 변환한 특정 알고리즘을 구현하며 시각 자료 구조의 종류에 따라 알맞은 사용자 인터페이스를 제공한다.

프레임워크를 구성하는 클래스는 이외에도 사용자 상호작용을 지원하기 위한 뷰리스너 클래스, 알고리즘의 동작 과정을 시각화 정보로 저장하는 시나리오 클래스와 시나리오 클래스의 구성요소인 시나리오 엘리먼트 클래스, 알고리즘의 코드를 시각화하는 코드뷰 클래스, 자료 구조의 실제 값을 가지는 자료구조노드 클래스 등이 있다. 프레임워크의 클래스 다이어그램은 그림 3과 같다.

프레임워크에서 알고리즘의 시각화는 시각 객체, 시나리오, 시각화 관리자의 동작에 의해 이루어진다. 시각 객체는 다양한 자료 구조의 시각적 표현을 위해 시각 자료 구조의 공통적인 속성을 추출하여 일반화한 것이며, 시나리오는 알고리즘

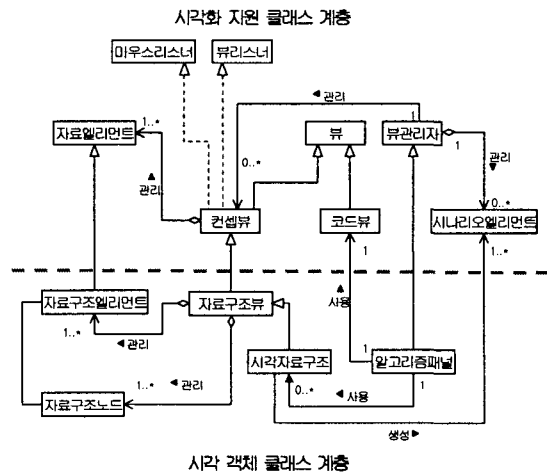


그림 3. 프레임워크의 클래스 다이어그램

에서 시각 자료 구조들이 수행할 일련의 애니메이션들에 대한 정보를 시간 순으로 저장한 것이다. 시각화 관리자는 알고리즘에서 시나리오를 생성한 후 그것을 바탕으로 어떤 자료 구조가 어떠한 애니메이션을 해야 하는가를 보고 해당 시각 객체에 알려 준다. 이 때, 시각화 관리자는 여러 개의 시각 객체를 동시에 관리하여 다중 뷰 동기화를 지원한다.

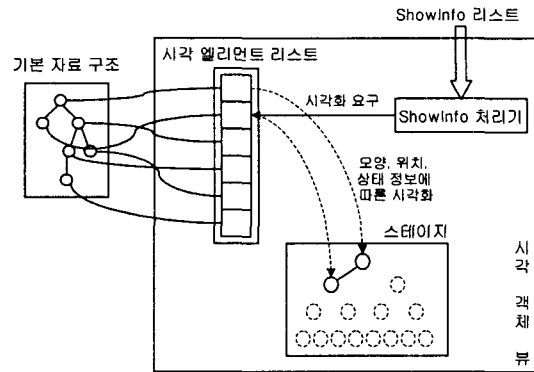


그림 4. 시각 객체 뷰의 동작

4.2 시각 객체의 구성 및 관리

여러 종류의 자료 구조들 사이에 공통적인 속성이 존재하듯이 시각적인 면에서도 그러한 공통 속성이 나타난다. 각 자료 구조의 시각적 표현을 위한 시각 자료 구조의 공통적 속성을 추출하여 일반화한 것이 시각 객체이다. 시각 객체는 다음과 같이 구성된다.

- 기본 자료 구조 : 알고리즘에서 사용하는 자료 구조이다.
- 시각 엘리먼트의 집합 : 시각 엘리먼트는 자료 구조의 개별 요소나 요소들간의 링크를 시각적으로 나타내는 객체이며, 시각 객체에서 개별적으로 시각화되는 최소 단위이다.
- 시각 객체 뷰 : 애니메이션을 직접적으로 표현하며 시각 엘리먼트를 관리하고 시각 엘리먼트 객체가 시각화되는 스테이지를 가진다.
- 시나리오 생성 연산의 집합 : 자료 구조의 특정 연산에 대응하는 시각화 정보를 생성하여 시나리오에 등록하는 연산들의 집합이다.

시각 객체 뷰는 시각 엘리먼트 리스트를 이용하여 시각 엘리먼트들을 관리한다. 그림 4에서 메모리의 실제 자료 구조와 시각 객체 뷰에 시각화된 자료 구조가 시각 엘리먼트 리스트를 통해 어떻게 매핑되는가를 보여준다.

실제 자료 구조의 요소는 자신을 시각화하는 시각 엘리먼트를 링크하고 있다. 시각 엘리먼트 역시 실제 자료 구조의 요소를 링크하고 있으며 스테이지 상의 그래픽에 대한 링크도 가진다. 엘리먼트 리스트에 시각 엘리먼트가 등록되면 스테이지에 바로 시각화된다. 실제 자료 구조에서 변화가 일어나면 그에 대응하도록 시각 엘리먼트의 위치 정보를 변화시켜 스테이지의 그래픽을 애니메이션 한다. 변화되는 위치 정보를 ShowInfo 리스트이라고 하며, 이는 시나리오로 이미 만들어져 있는 상태이다.

4.3 시나리오의 생성 및 관리

시나리오란 단일 알고리즘에서 일어나는 모든 행동들을 시간 순서에 맞게 정렬해 놓은 시각화 정보들의 집합이며, 시각 객체 뷰에서 일어나는 모든 애니메이션은 시나리오에 따라 이루어진다.

그림 5은 시나리오의 구조이다. 시나리오를 구성하고 있는 가장 작은 단위인 시나리오 엘리먼트는 하나의 ShowInfo 리스트와 그것이 적용되는 시각 객체 뷰 지시자로 이루어진 집합이다. 또한, 이것들의 집합이 시나리오 엘리먼트 리스트를 구성한다. 시나리오 엘리먼트의 상위 구조로 시나리오 엘리먼트 리스트를 둔 이유는 하나의 자료

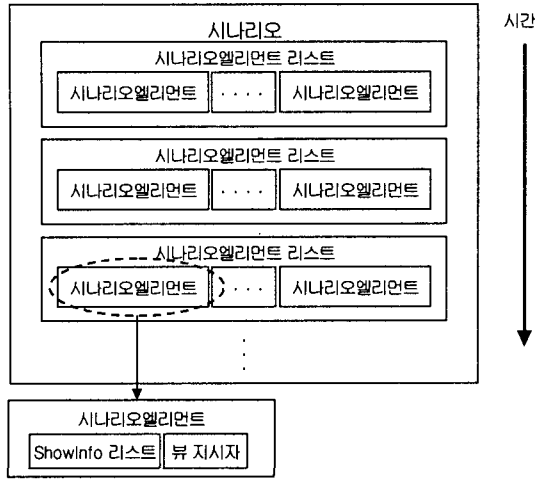


그림 5. 시나리오의 구조

구조를 한 가지 이상의 표현으로 나타낼 경우를 위해서이다. 시나리오 엘리먼트 리스트에 속한 모든 시나리오 엘리먼트는 동시에 처리되므로 여러 개의 자료 객체 뷰가 시각화되는 것을 지원한다. 따라서 하나의 자료 구조를 한 가지 이상의 표현으로 나타내어 동시에 시각화하는 것이 가능하다.

5. 구현 결과 및 평가

제시한 셀프-애니메이션 프레임워크는 JAVA를 이용하여 구현하였으며 프레임워크를 이용하여 생성된 시각 자료 구조 클래스들과 이를 관리하는 클래스들은 시각 자료 구조 라이브러리에 포함된다.

현재 시각 자료 구조 라이브러리가 지원하는 시각 자료 구조로는 일차원 배열, 이차원 배열, 단일 연결 리스트, 이중 연결 리스트, 스택, 큐, 이진 트리, 일반 그래프, 방향 그래프, 가중(Weight) 그래프가 있으며 그림 6은 이들의 관계를 나타내는 클래스 계층 구조이다.

시각 자료 구조는 기본적으로 컨셉뷰를 상속받는다. 추상 자료 구조들 중에 기본적인 것들에 대

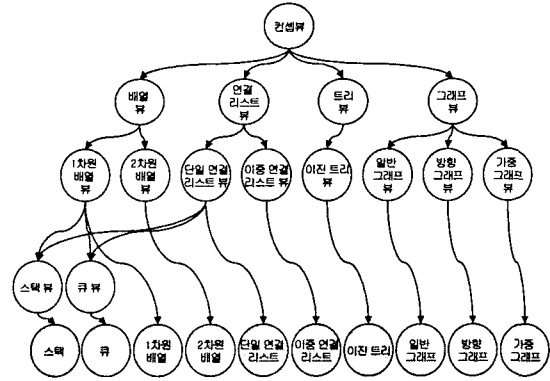


그림 6. 시각 자료의 클래스 계층 구조

한 뷰 클래스를 생성한 다음 각각의 뷰 클래스들에서 다시 세분화한 자료 구조 뷰 클래스를 상속하여 정의한다. 마지막으로 뷰 클래스를 상속하여 시나리오 생성 연산을 첨가하여 시각 자료 구조 클래스를 만든다.

알고리즘 교육 시각화 시스템 역시 JAVA를 사용하여 서버/클라이언트 구조로 구현되었다. 클라이언트 측은 원격 가상 교육 환경에 적합하고 JAVA를 지원하는 모든 웹 브라우저에서 실행되는 자바 애플릿으로 이루어진다.

클라이언트는 알고리즘을 입력받아 웹 서버로 전송하고 서버에서 전송된 애니메이션 애플릿을 실행한다. 서버에서는 알고리즘을 실행가능한 클래스 파일들로 구성된 애플릿으로 만들어서 웹 문서에 삽입한 다음 문서의 URL을 사용자 측에 전송한다.

URL을 전달받은 클라이언트는 웹 브라우저를 실행시켜 해당 URL의 웹 문서를 불러와서 알고리즘이 애니메이션 되는 애플릿을 실행한다. 그림 7과 그림 8은 애니메이션 애플릿의 예이다.

그림 7의 힙 정렬 애니메이션의 경우, 힙을 트리와 배열 두 가지 형태로 나타내며, 이들은 정렬이 이루어지는 과정에서 요소의 이동 동작이 동시에 이루어진다. 반면에 그림 8의 깊이 우선 탐색

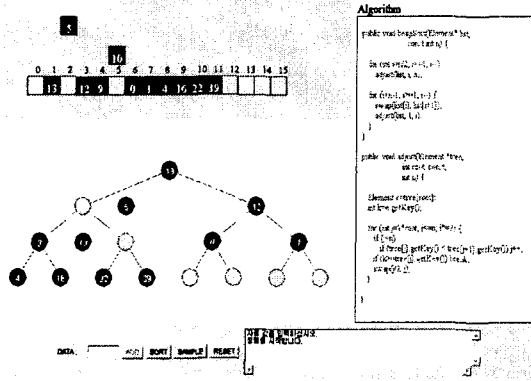


그림 7. 힙 정렬 애니메이션

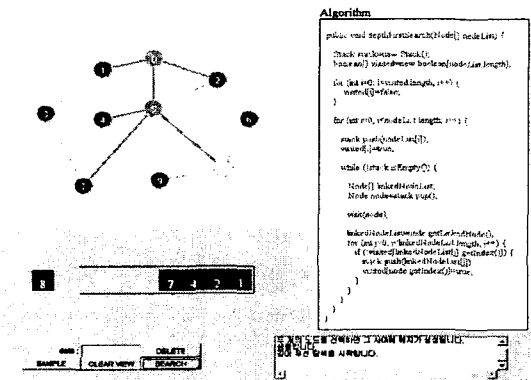


그림 8. 그래프의 깊이 우선 탐색 애니메이션

애니메이션은 알고리즘에서 사용되는 그래프와 스택의 동작을 알고리즘의 실행에 맞추어 애니메이션 한다. 또한, 화면에 있는 알고리즘의 소스 코드 역시 애니메이션에 맞추어 현재 실행되고 있는 부분이 표시가 된다.

셀프-애니메이션 프레임워크를 이용한 알고리즘 교육 시각화 시스템을 평가하기 위하여 자료 구조나 알고리즘 과목을 수강한 컴퓨터과학과 학생 40명을 대상으로 만족도를 측정하였다. 그 결과는 그림 9와 같다. 시스템의 사용자 인터페이스에 대해서는 만족하는 편이며 사용자 상호작용에 대해서도 긍정적인 평가를 했다. 특히 알고리즘 학습 과정에서 알고리즘 교육 시각화 시스템이

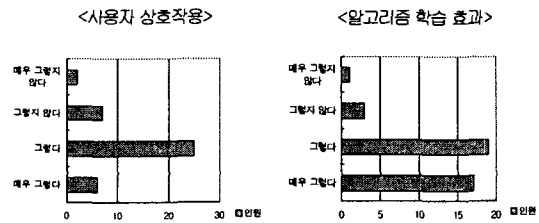


그림 9. 학습자 만족도 조사

효과적인 학습 환경을 제공한 것으로 평가되었다.

6. 결론

본 연구에서는 알고리즘의 시각적 표현을 생성하는 과정에서 투명성이 보장되는 알고리즘 교육 시각화 시스템을 개발하고, 알고리즘의 동작 형태를 시각적으로 표현하는 애니메이션이 자동적으로 생성되는 것을 지원하는 셀프-애니메이션 프레임워크를 제안하였다. 이는 별도의 노력을 들이지 않고 알고리즘을 시각화하기 위하여 자료 구조의 모양과 동작을 시각적으로 표현하는 클래스인 시각 자료 구조와 이를 관리하기 위한 클래스를 제공한다.

본 프레임워크는 용이한 자료 구조의 생성을 위해 계층 구조로 이루어진다. 상위 계층에서 제공한 기본적인 시각화 기능에 특정 시각 자료 구조를 위한 기능을 추가하거나 기능을 재정의 함으로써 제작 과정에 독립성을 높이며 다양한 자료 구조의 생성을 지원한다.

셀프-애니메이션 프레임워크를 이용한 알고리즘 교육 시각화 시스템은 다양한 시각 자료 구조를 지원하여 컴퓨터과학의 교육 과정에서 다루는 대부분의 알고리즘을 시각화한다. 또한, 단일 알고리즘에서 다수의 자료 구조의 사용을 지원하고 단일 자료 구조를 다양하게 표현하여 효과적인 알고리즘 학습 환경을 제공하며, 마우스 이벤트를 통하여 학습자가 뷰에서의 직접적으로 자료를 조

작하도록 함으로써 직관적인 학습이 이루어지도록 한다.

참 고 문 헌

[1] J. Korsh, P. LaFollette and R. Sangwan, "SAnimating programs and students in the laboratory," *Frontiers in Education Conference*, 1998. FIE '98. 28th Annual, Vol. 3, pp. 1139 - 1144, 1998.

[2] Kehoe, Colleen and J. T. Stasko, "Using Animations to Learn about Algorithms: An Ethnographic Case Study," *Graphics, Visualization, and Usability Center, Georgia Institute of Technology, Atlanta, GA, Technical Report GIT-GVU-96-20, September 1996.*

[3] 김미영, 민수정, 김상욱, "ADVA: 자료구조의 애니메이션과 알고리즘의 시각화", *HCI 2000 학술대회 발표논문집, 한국정보과학회, 2000.1.*

[4] J. Haajanen, M. Pesonius, E. Sutinen, J. Tarhio, T. Terasvirta and P. Vanninen, "Animation of user algorithms on the Web," *Proceedings of VL '97, Symposium on Visual Languages, IEEE, pp. 356-363, 1997.*

[5] M. A. Najork, M. Brown and R. Raisamo, "A Java-based implementation of collaborative active textbooks," *Visual Languages, 1997. Proceedings of VL '97, Symposium on Visual Languages, IEEE, pp. 372-379, 1997.*

[6] Gruia-Catalin Roman and Kenneth C. Cox, "A Taxonomy of Program Visualization System," *IEEE Computer, Vol. 26, No. 12, pp. 11-24, Jan. 1993.*

[7] M. H. Brown, "Exploring Algorithms Using Balsa-II," *IEEE Computer, Vol. 21, No 5, pp. 14-36, May 1988*

[8] S.-P. Lahtinen, E. sutinen, J. Tarhio and A.-P. Tuovinen, "Object-Oriented Visualization of Program Logic," *Proceedings of Technology of Object-Oriented Languages and Systems,*

1998, pp. 76-88, 1998

[9] J. T. Stasko, "Tango: a framework and system for algorithm animation," *IEEE Computer, Vol. 23, No. 9, pp. 27-39, Sept. 1990.*

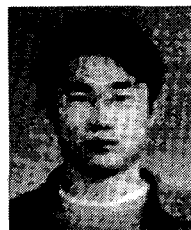
[10] P. Lafollete, J. Korsh and R. Sangwan "A Visual Interface for Effortless Animation of C/C++ Programs," *Journal of Visual Languages and Computing, 2000, pp. 27-48, 11, 2000.*

[11] E. Heltulla, A. Hyrskykari and K. Raiha, "Graphical Specification of Algorithm Animations with Aladdin," *Proceedings of the International Conference on System Sciences, Vol. 2, pp. 40-54. 1989*



민 수 정

- 경북대학교 컴퓨터과학 학사
- 경북대학교 컴퓨터과학 석사
- 현재, 경북대학교 시간강사
- 관심분야 : 인간과 컴퓨터 상호작용, 가상 교육



송 병 옥

- 경북대학교 컴퓨터과학 학사
- 경북대학교 컴퓨터과학 석사
- 현재, 경북대학교 대학원 컴퓨터과학과 박사과정
- 관심분야 : 인간과 컴퓨터 상호작용, 네트워크 보안



김 상 옥

- 경북대학교 컴퓨터공학 학사
 - 서울대학교 컴퓨터과학 석사
 - 서울대학교 컴퓨터과학 박사
 - 현재, 경북대학교 컴퓨터과학과 교수
 - 관심분야 : 인간과 컴퓨터 상호작용, 컴퓨터언어, 분산 멀티미디어 컴퓨팅
-