

객체지향 프로그래밍

간단한 마인드맵 애플리케이션



과목명		객체지향 프로그래밍
담당교수		최지웅 교수님
학과		컴퓨터학부 (나반)
팀원 1		20172620 남궁혁
팀원 2		20172655 이강산
제출일		2018년 6월 13일

<< 목차 >>

1. 구현 기능	2
2. 제작 과정 및 주요 함수 설명	4
3. 실행 화면	5

1. 구현 기능

1-1. 텍스트 에디터 페인

- 한 행이 하나의 노드에 대응
- TAB으로 레벨 결정
- '적용' 버튼 클릭 시 마인드 맵 가시화

1-2. 속성 페인

- Focus 받은 노드의 속성 표시(이름, xy좌표, 길이, 높이, 색상)
- '변경' 버튼 클릭 시 수정된 노드의 속성 값을 마인드 맵 패널에 반영
 - ◆ 변경된 노드 속성, 연결선을 다시 마인드맵 페인에 반영

1-3. 마인드맵 페인

- Focus 받은 노드 색상 반전, 속성 페인에 표시
- 특정 노드 클릭 후 마우스 드래그를 통한 이동
 - ◆ 변경된 xy좌표를 속성 페인에 실시간 반영
 - ◆ 드래그를 통한 이동 시 두 노드 간 최단거리로 연결선 실시간 변화
- 선택한 노드의 모서리 드래그를 통한 노드 크기 변경
 - ◆ 변경된 xy좌표, 길이, 높이를 속성 페인에 실시간 반영
 - ◆ 변경된 좌표, 크기에 따라 연결선 실시간 변화
- 노드가 마인드맵 페인 영역을 벗어날 경우 페인 크기 자동 재조정

1-4. 노드

- 상, 하, 좌, 우 4개의 연결점 가짐

- 연결선은 부모-자식 간 최단 거리인 두 연결점에 그려짐

1-5. 메뉴 바/툴 바

- 새로만들기
 - ◆ 새로운 마인드맵 편집
 - ◆ 패널 위에 마인드맵이 존재한다면 저장 여부 질문
- 열기
 - ◆ 기존에 저장한 파일을 불러와 텍스트 에디터 페인과 마인드맵 페인에 반영
- 저장
 - ◆ 현재 텍스트 에디터 페인과 마인드맵 페인의 내용을 json파일로 저장
 - ◆ '열기'로 불러온 파일의 경우 기존 이름의 파일에 저장
- 다른 이름으로 저장
 - ◆ 현재 텍스트 에디터 페인과 마인드맵 페인의 내용을 다른 이름의 json파일로 저장
- 닫기
 - ◆ 프로그램 종료
 - ◆ 작성 중인 마인드맵 존재 시 저장 여부 질문
- 적용
 - ◆ 텍스트 편집 내용을 마인드맵 페인에 적용
- 변경
 - ◆ 속성 페인에서의 변경 내용을 마인드맵 페인에 적용

1-6. 마인드맵 모델

- 텍스트 페인의 내용을 Node 클래스의 연결리스트 방식 트리 모델로 저장

2. 제작 과정

타임라인	남궁혁	이강산
5월 3일~	GUI 공부	GUI 공부
5월 11-12일		기본 틀 제작 MyFrame 클래스 작성 Node 클래스 작성
5월 18-19일	addTree() 함수작성	적용, 변경 버튼 구현
5월 25-26일	마인드맵 페인에 노드 부착 속성 페인에 노드 속성 연결	메뉴 바/툴 바 기능 구현
6월 1-2일	LinePanel 클래스 작성	노드 원형 배치
6월 8-9일	노드 간 연결선	
6월 11일	조정, 예외처리	
6월 12일	보고서 작성	보고서 작성

3. 주요 함수 설명

3-1. MyFrame 클래스

- `private void addTree(int ct, Boolean opened)`

3-2. Node 클래스

- `Public void setPaneSize()`

노드가 마인드맵 페인을 벗어날 경우 마인드맵 페인의 크기 재조정

3-3. LinePanel 클래스

- `public void setCoordinatesForPaint()`

부모, 자식 각각 4개의 부착점 좌표 할당

- `public void setShortestLine()`

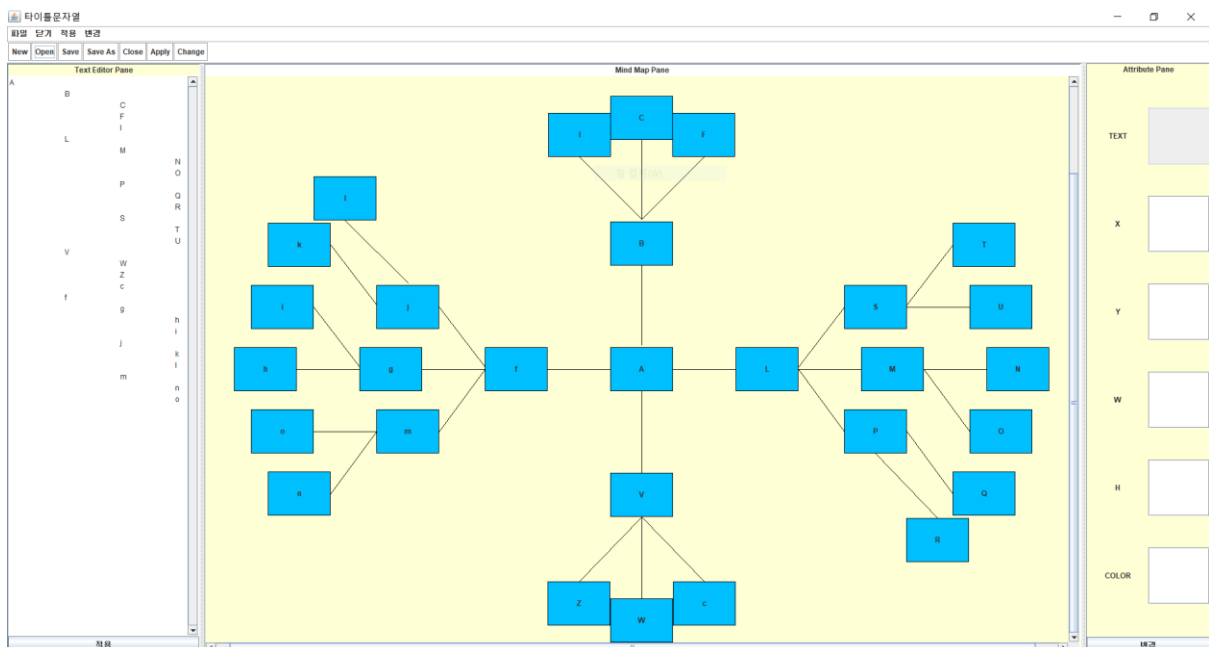
부모, 자식 간 최단 경로인 두 부착점 선택

4. 실행 화면

4-1. 초기 화면

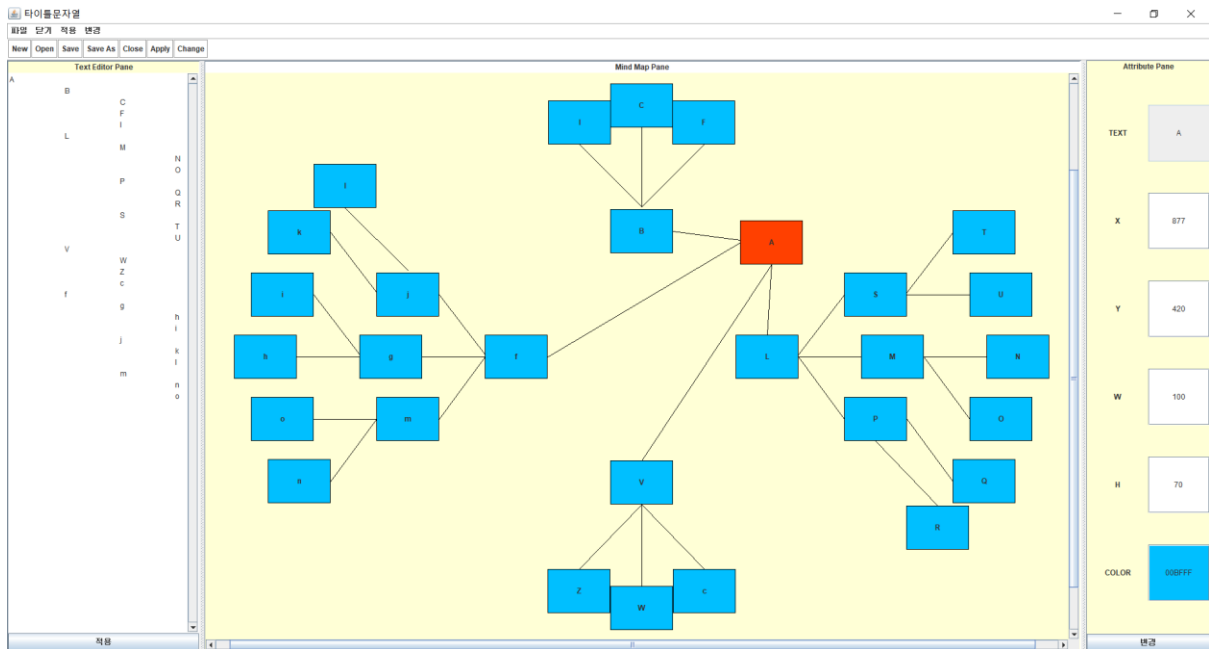


4-2. 적용 버튼 수행 (자식 수에 따른 원형 배치)



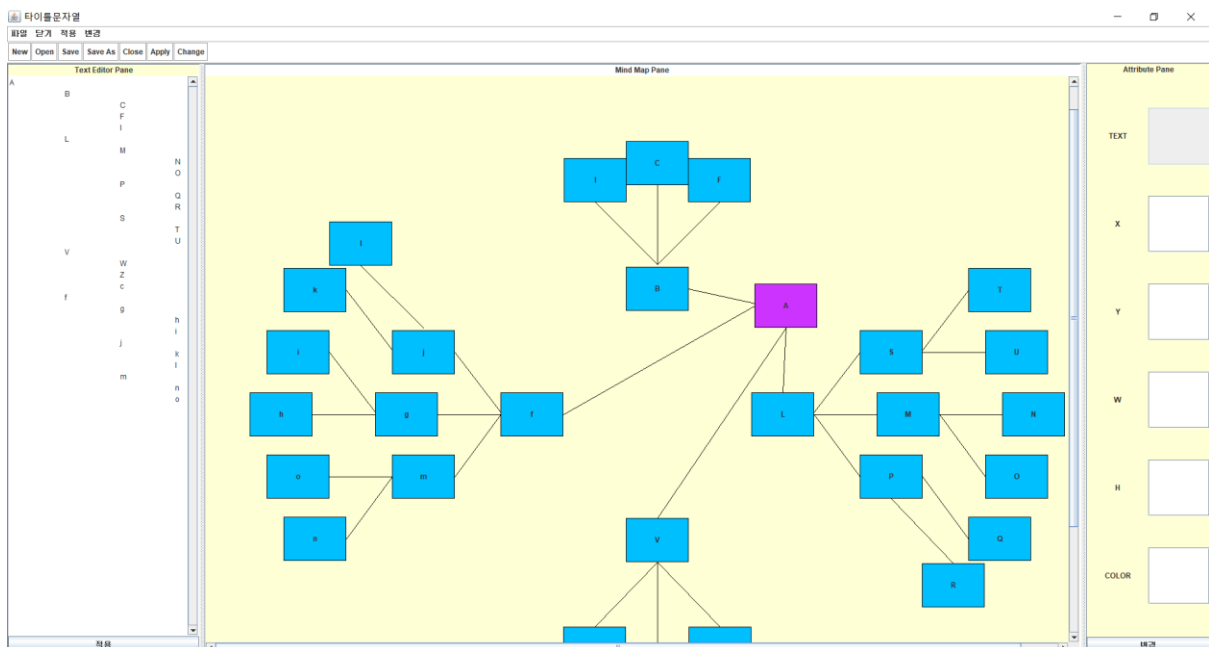
* 자식 수에 따른 노드의 원형 배치

4-3. 노드 이동 (최 상단 노드 이동, 포커스에 따른 색 반전)



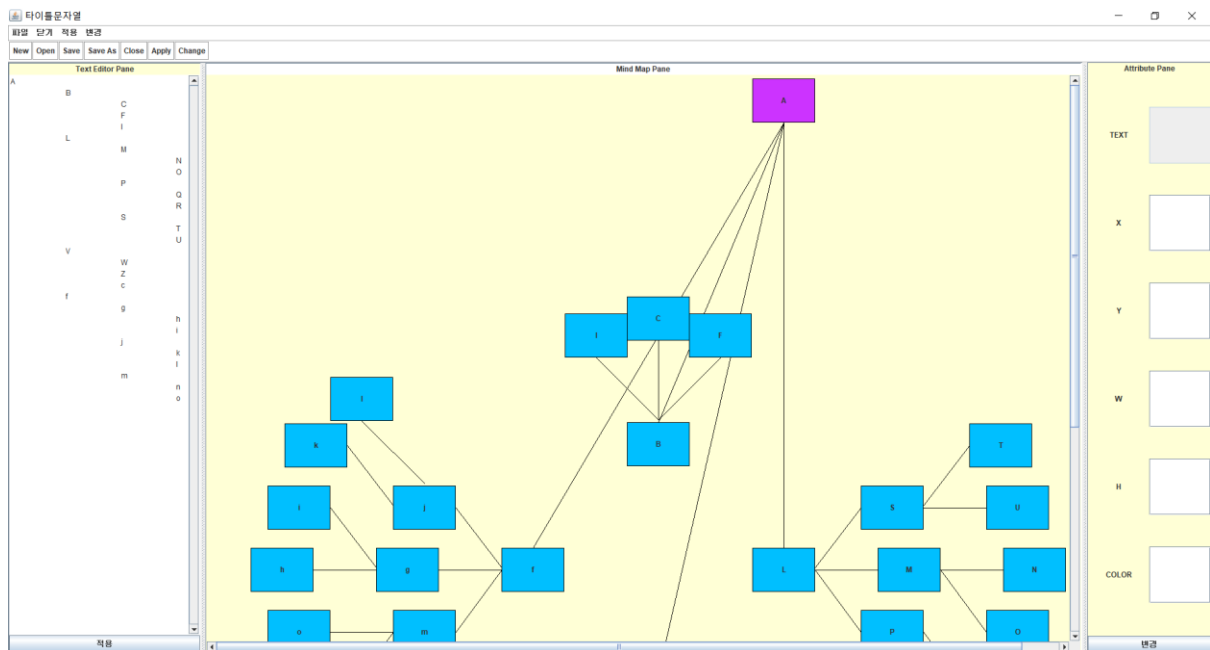
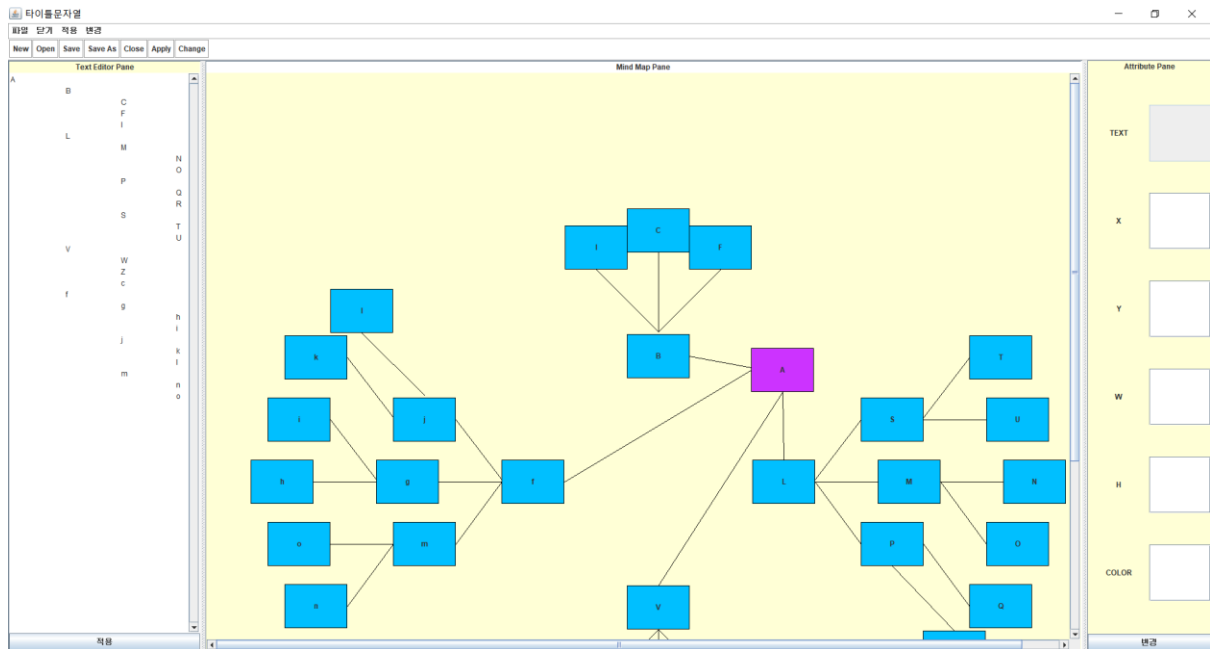
* 노드 이동 시 연결선은 두 노드간 최단거리에 실시간으로 다시 그려진다.

4-4. 노드 색상 변경 (하늘색 -> 보라색)



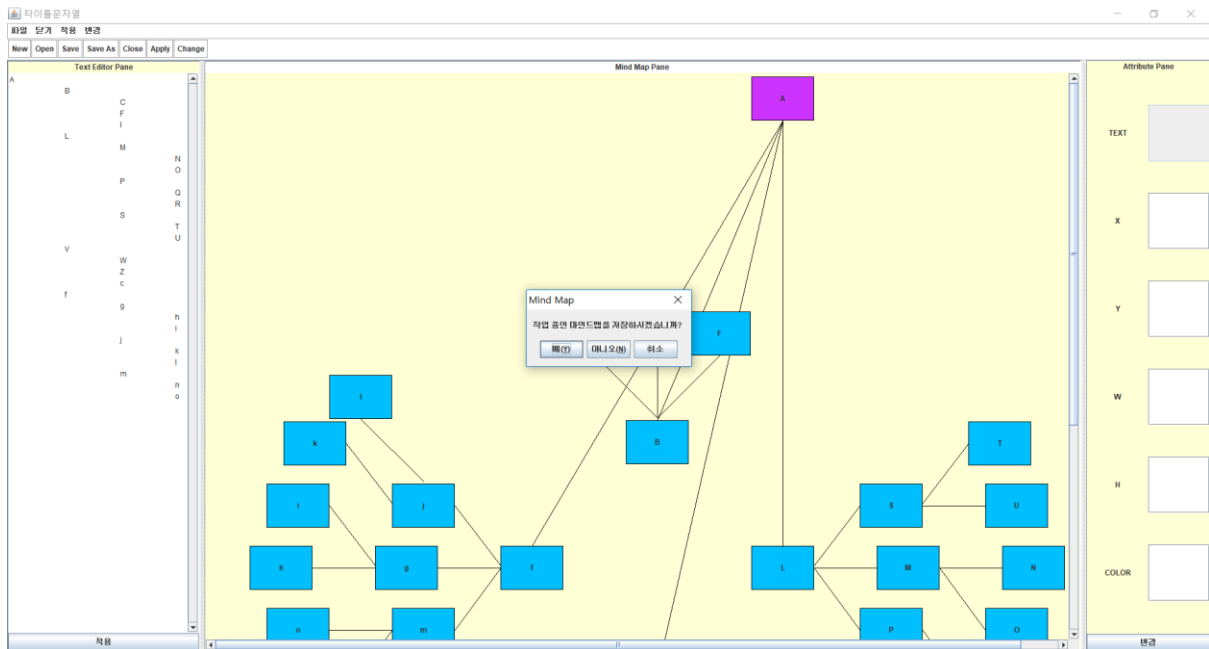
* JColorChooser 를 이용하여 색을 변경한다.

4-5. 노드 이동에 따른 페인 크기 재조정

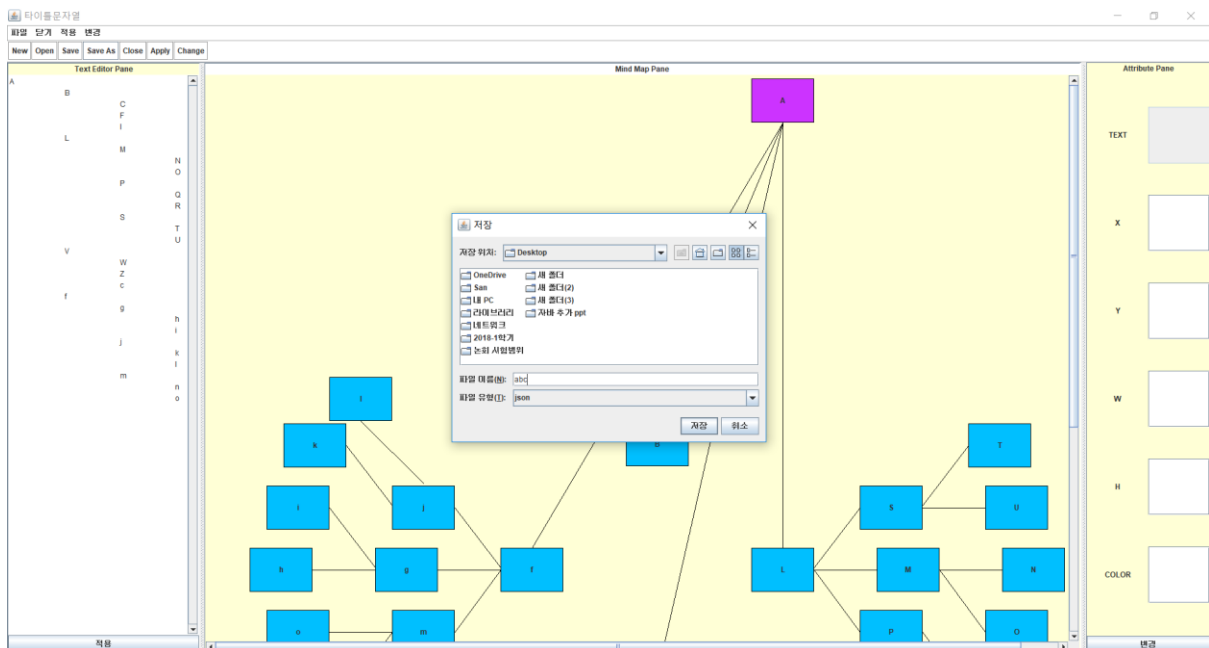


* 이동한 노드가 마인드맵 페인을 벗어나 마인드맵 페인 크기가 조정되었다.

4-6. 닫기 / 저장



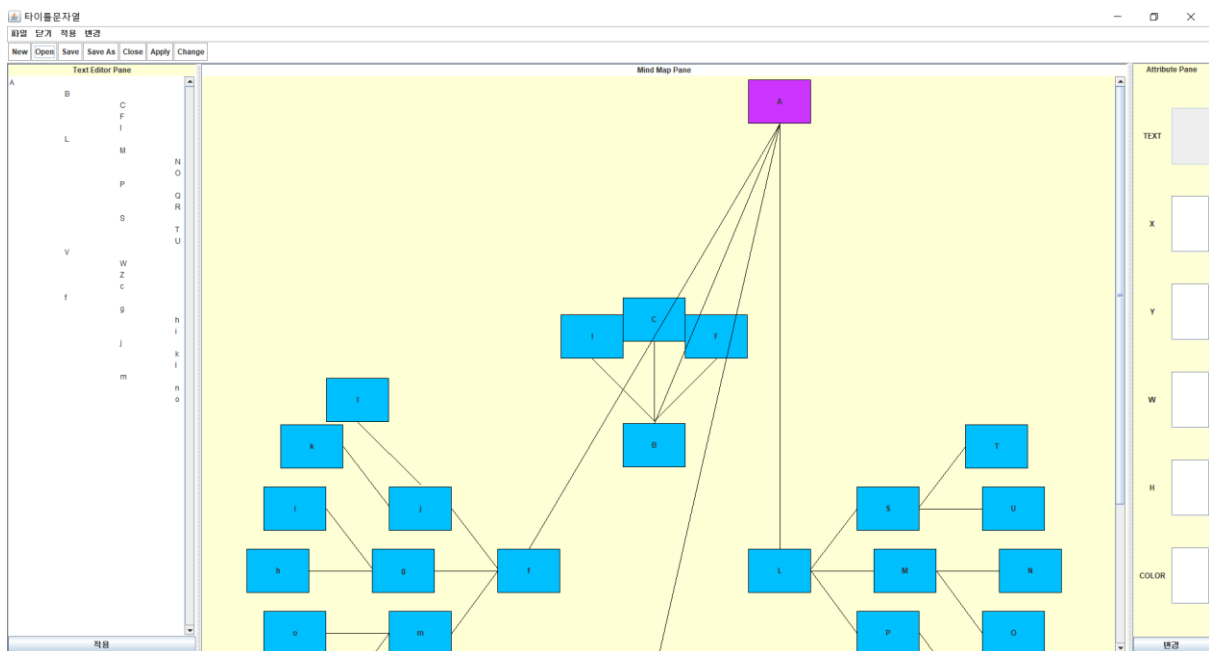
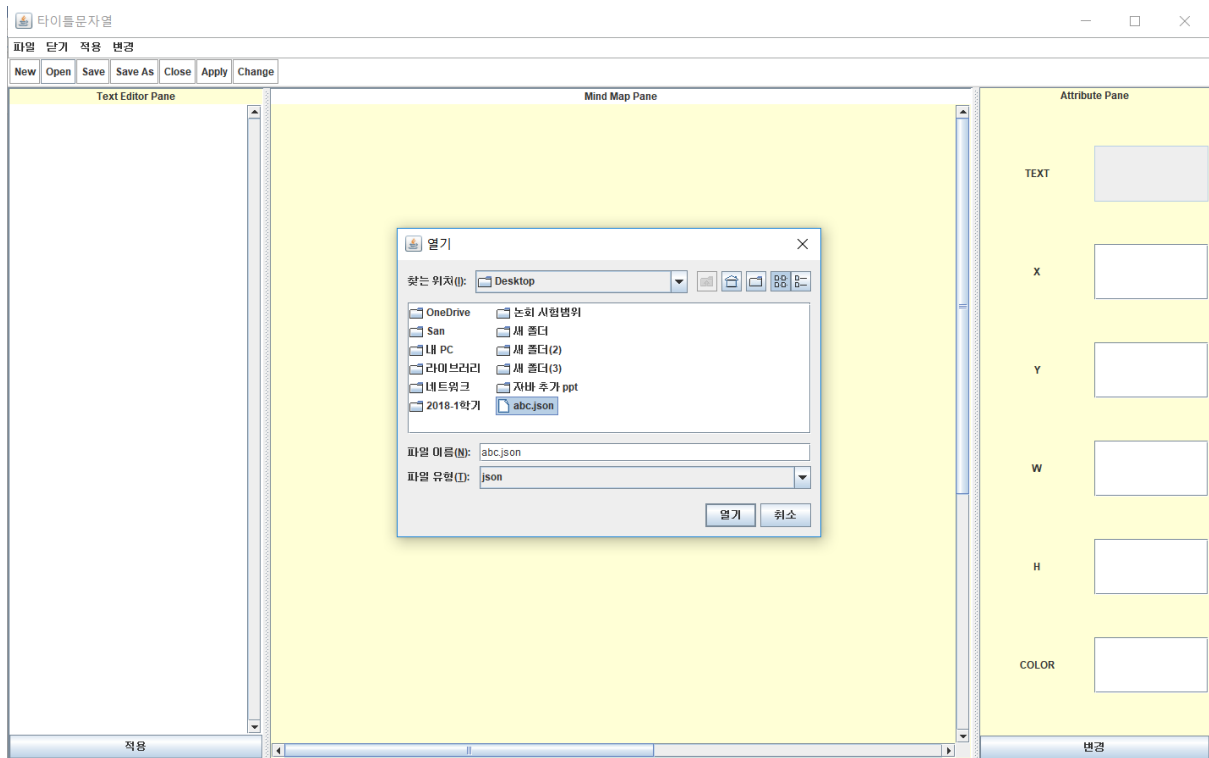
* 마인드맵이 있을 때 '새로만들기' 또는 '닫기' 시 저장할지를 확인한다.



* json파일로 저장된다

(첫 저장 이후에는 현재 파일의 경로를 이용하여 저장하므로 저장 다이얼로그를 띄우지 않는다.)

4-7. 불러오기



* 불러오기 시 json 파일에 저장된 노드관계, 좌표, 색상을 불러온다.