**프로젝트 보고서**

**20170288 박예빈**

**20170294 박해영**

|  |  |  |  |
| --- | --- | --- | --- |
| **0. 마인드맵 애플리케이션 구현** | | | |
|  | | | |
|  | **public** **class** **DrawMindMap**  **class** **BasicFramework** **extends** **JFrame**  **class** **Elements** **extends** **JMenuBar**  **class** **Bar** **extends** **Elements**  **class** **Mindmap** **extends** **Elements**  **class** **Attribute** **extends** **Elements**  **class** **Text** **extends** **Elements**  **class** **Data**  **class** **MakeToLabel** **extends** **Elements**  **class** **Tree** **extends** **MakeToLabel**  **class** **JDrawPanel** **extends** **JPanel**  **class** **NewButtonListener** **implements** ActionListener  **class** **ButtonListener** **implements** ActionListener  **class** **updateLine**  **class** **JLabelListener** **extends** **MouseAdapter**  **public** **class** **DotListener** **extends** **JLabelListener**  **class** **SaveButtonListener** **implements** ActionListener  **class** **OpenButtonListener** **implements** ActionListener  **class** **CloseButtonListener** **implements** ActionListener  **class** **WriteXMLFile**  **class** **ReadXMLFile** | | |
| **1. 사용자 인터페이스 요구 사항** | | | |
|  | | | |
| - Attribute Editor는 스크롤이 존재하지 않는다.  - Text Editor 는 세로 스크롤이 기본으로 존재한다.  ( 단, 글자 길이가 가로 길이보다 넘어갈 시 스크롤이 생성된다. )  - 프레임의 사이즈는 고정되어 있다.  - JSplitPane의 분리대는 고정되어 있다. | | | |
|  | | | |
| - Mind Map Panel의 스크롤 바는 노드의 레벨(height)에 따라 활성화의 유무가 결정된다. | | | |
|  | | - MenuBar의 Menu가 존재한다.  - Menu에 MenuItem이 존재한다 | |
| **2. 텍스트 에디터 페인의 요구 사항** | | | |
|  | | | |
| - 트리가 형성될 때 각 노드는 Data 클래스로 저장된다  - 접근의 용이성을 높이기 위해 Data 클래스의 필드로 parent, child, height, label 등을 가진다.  - Data 클래스는 parent, child, sibling 값을 각 하나씩만 할당을 받는다. 따라서 여러 자식이 있을 경우 getChild()를 한 후, 다시 getSibling()함수로 여러 자식들에 접근한다.  - 이와 달리 모든 노드는 자신의 부모 노드와 연결 되어 있다.  - Tree는 적용 버튼을 통해 생성된다.  - 생성된 Data 값들은 ArrayList<Data> Datas에 저장되어 TextEditor, MindMap, Attribute Editor에서 관리가 가능해진다. | | | |
|  | | - 트리의 형성을 위해 입력된 문자열을 \n을 기준으로 토큰을 만든다.  - 첫번째 토큰부터 차례로 검사한다.  - 무조건 첫번째가 루트이므로 바로 트리의 start가 된다.  - 이후에 추가되는 값의 레벨을 정하기 위해 최근 저장한 값의 \t(Tab)의 최대 인덱스 값과 현재 추가하고자 하는 \t(Tab)의 최대 인덱스 값을 서로 비교한다.  - 사용자가 입력을 할 때 \t(Tab)으로 계층을 나누는데, 현재 데이터의 최대 인덱스가 최근 데이터의 인덱스보다 클 경우 무조건 그 차이는 1이어야 한다.  - 이 때 추가되는 값은 이전 노드의 Child이다.  - 현재 최대 인덱스 값이 최근 인덱스 값과 같을 경우 최근 데이터의 Sibling이 된다.  - 마지막으로 최근 인덱스 값이 작을 경우 자신 노드의 부모 세대의 Sibling이 되기 때문에 이 노드의 위치를 찾기 위해 getParent()로 위 세대로 올라가며 인덱스를 비교한다. 같은 인덱스를 발견했을 때 추가된 노드는 그 노드의 Sibling이 된다. | |
|  | | - input form에 어긋난 경우 : 팝업 창 띄우기 | |
|  | | - root node : MindMap Panel의 중심에 띄운다. | |
|  | | - 이 외 : Parent node의 중심을 기점으로 직교좌표계를 그리고 사분면을 기준으로 띄운다.  🡺 Parent node가 위치한 사분면과 자신이 몇 번째 형제(=getSiblingIndex())인지를 통해 모든 경우의 수 구현 | |
| **3. 속성 페인의 요구 사항** | | | |
|  | | - color : 6자리 16진수(hex)  ( user input form : 0x\_\_\_\_\_\_ ) | |
|  | | - 어떠한 Label을 띄우지 않았을 경우, 변경 버튼을 눌렀다면 팝업창을 띄움으로써 사용자에게 알려준다. | |
|  | | - 속성 값을 변경할 Label을 선택하지 않았을 경우 팝업창을 띄움으로써 사용자에게 알려준다. | |
| **4. 마인드 맵 페인의 요구 사항** | | | |
|  | | - node를 클릭하면 Attribute Editor Panel에 node의 속성 값을 띄워준다. | |
|  | | - node를 클릭하면 연결점 및 크기 변환 좌표를 표시해준다. | |
|  | | - node를 클릭 시 color의 속성 값은 반전되기 전의 컬러 16진수를 표현해준다. | |
|  | | | |
| - node의 위치 이동  🡺 parent와 child의 x, y 좌표를 기준으로 그려지는 공간 내부로만 이동 가능  🡺 위치 이동 시 연결선 실시간 draw  - node의 크기 변경  🡺 parent와 child의 x, y좌표를 기준으로 그려지는 공간 내부로만 크기변경 가능 | | | |
| **5. 메뉴 바, 툴 바의 요구 사항** | | | |
|  | | | - 다른 이름으로 저장 시, 중도 포기할 경우 팝업창을 띄움으로써 사용자에게 알려준다. |
|  | | | - Save (저장) : 사용자의 DeskTop에 (root의 String).xml로 저장된다 |
|  | | | - Save As…(다른 이름으로 저장) : 사용자가 지정한 공간에 지정한 파일로 저장한다  (단, 같은 이름의 xml 파일이 존재할 경우, 덮어쓰기를 진행한다.) |
|  | | | - Save As…의 경우 : 지정된 파일이 해당 공간에 없을 경우 해당 팝업창을 띄워 사용자에게 알린다. |
|  | | | - 애초에 사용자가 적용버튼을 실행한 적이 없다면 Save & Save As 버튼은 활성화 되어 실행되지 않는다. |
|  | | | - menu Bar에 help(도움말) menuItem을 추가하였다. |
|  | | | - help의 TextEditor Form을 누르면 TextEditor의 입력 형식에 대한 팝업창이 띄워진다. |
| **6. 파일 생성에 대한 요구 사항** | | | |
|  | | | - XML file 형식  >> 모든 node에 담긴 Data를 xml 파일로 저장한다. |
|  | | | - element <R>은 node의 height를 의미한다  >> 이를 이용하여 TextEditor에 적힐 String 값을 구한다. (height == \t의 개수) |

**<프로젝트 일지>**

|  |  |  |
| --- | --- | --- |
| **일정** | **구현 사항** | **담당자** |
| 5/10 - 5/12 | 개인 9장 공부 |  |
| 5/12 - 5/13 | 기본 틀 구현 | 박해영 |
| 5/16 - 5/19 | 기본 틀 🡪 Event & elements file 분할 | 박예빈 |
| 5/22 | Menu 구현방법 수정 & menuItem event 수정 | 박해영 |
| 5/22 – 5/25 | Tree 구현 & addNode 초기 틀 작성 | 박예빈 |
| Xml 구현 방법 공부 | 박해영 |
| 5/27 – 5/28  - 5/29 | menuItem event 추가 & Tree 예외 처리 |  |
| mindmapPanel에 JLabel 띄우기 – 연결 |  |
| 5/30 - 6/1 | JLabel 띄우기 – 알고리즘 구현 | 박해영 |
| 6/2 | New Button 구현 & JLabel 이동 구현 | 박예빈 |
| 6/3 | JLabel 띄우기 – 알고리즘 error 수정 | 박해영 |
| 6/3 – 6/5  – 6/6 | JLabel 알고리즘 완성 | 박해영 |
| 선 그리기 – 알고리즘 구현 | 박예빈 |
| 6/6 – 6/7 | 선 그리기 – event에 따른 case 구현 | 박예빈 |
| XML – save & open 구현 | 박해영 |
| 6/8 | Change button listener 구현 | 박해영 |
| 6/9 | Node 이동 제한 설정 | 박예빈 |
| 6/10 | Color hex 계산 & change 적용 | 박해영 |
| 6/10 – 6/11 | Node 연결점으로 크기 조절 | 박예빈 |
| 6/11 | Close button listener 구현 | 박해영 |
| Node click 선택 & 해제 구현 | 박예빈 |
| 6/12 | 최종 예외 처리 및 확인 작업 |  |

\* 담당자 빈칸은 같이 진행