SW 개발 환경 및 개발 방법 소개

개발 툴

A. Python version 3.6

Python은 귀도 반 로썸이 개발한 객체지향 언어이다. 1991년에 발표되었으며, 인터프리터방식의 프로그래밍 언어이다. 파이썬은 'Code Reability'라는 철학하에 작성되었으며, 이를 지원하기 위해 다양한 도구를 지원한다. Dynamic Type System,Automatic memory management에 더해 다양하고 포괄적인 standard library등도 제공한다. 이러한 특성들 덕분에 파이썬은 유지보수가 쉽고, 다양한 라이브러리가 존재하는 강력한 언어로서 기능하고 있다.

B. python-docx(python library) version 0.8.6

python-docx는 Microsoft Word(.docx)파일을 제작하고 업데이트 할 수 있는 Python library이다. 기본적으로는 Word파일을 새로 제작하는데 중점을 두고 있지만, python-docx는 이에 더해 Word파일을 XML분석할 수 있는 기능을 제공한다. Word파일의 구성요소들을 쪼갤 수 있는 XML 분석을 통해 프로그래머는 원하는 방식으로 이들을 활용할 수 있으며 이를 이용한 프로그램을 작성할 수 있다.

C. python-pptx(python library) version 0.6.9

python-pptx는 Microsoft PowerPoint(.pptx)파일을 제작하고 업데이트 할 수 있는 Python library이다. 주로 DB의 내용을 Pptx화 할 때 주로 사용된다. python-pptx를 통해 슬라이드의 레이아웃, 배치 내용, 배치 방식등을 결정할 수 있으며 이를 이용해 word파일로부터 추출한 정보를 Pptx화 하였다.

D. PyQt(python library) version 5.10.1

PyQt는 C++기반의 Qt와 Python을 연결하는 가장 널리 알려진 라이브러리이다. PyQt는 Qt 4,Qt 5버전에 대한 python binding을 지원한다. 이번 프로젝트에서는 PyQt 5를 이용하여 개발을 하였다. PyQt는 Python환경에서 GUI를 구현하기 위한 라이브러리로서, signal-slot방식을 지원하여 GUI상에서의 이벤트를 쉽게 처리할 수 있다. 사용자는 GUI를 통해 파일을 입출력하고, 옵션을 선택할 수 있다.

개발 방법

A. 프로그램 구조

프로그램은 아래의 네 부분으로 나뉜다.

- DocxParser : 입력된 Word File을 XML분석하여 각 단락들의 계층적 구조를 생성한다. 각 단락들은 Title Class의 Instance로서 기능하게 된다. 사용자는 GUI상에서 Word File에서 추출할 부분을 결정하게 되는데, 이를 바탕으로 각 단락에서 중요 내용을 추출하고 이를 Title Class의 집합인 TitleList에 저장한다.

  (환경 : Python, python-docx)

- Docx2PptxConverter : Title Class를 프레젠테이션을 만드는데 필요한 정보를 담고 있는 Pclass 클래스로 변환한다.

  (환경 : Python, python-pptx)

- PPTMaker : 전달받은 Pclass의 정보를 바탕으로 presentation객체를 생성하고 이미지와 텍스트를 입력하여 pptx 파일을 생성해낸다.

  (환경 : Python, python-pptx)

- GUI : GUI환경은 사용자가 프로그램을 구동할 때 필요한 단계들을 구현한다. 전체적인 프로그램 프레임과 단계마다 필요한 옵션들과 선택창, 만들어질 피피티 슬라이드의 개요와 순서를 보여주며 사용자가 최소한의 클릭으로 프로그램 구동에 필요한 옵션을 설정할 수 있게 한다. 사용자가 프로그램을 사용할 때의 가이드라인을 제공하기 위해서 버튼의 활성화 유무와 placeholder 등을 사용해서 프로그램이 정상적으로 구동하기 위해 필요한 조작들을 유도할 수 있도록 가이드한다.

  (환경 : Python, PyQt)

위 네 부분은 아래와 같은 구조로 동작한다.

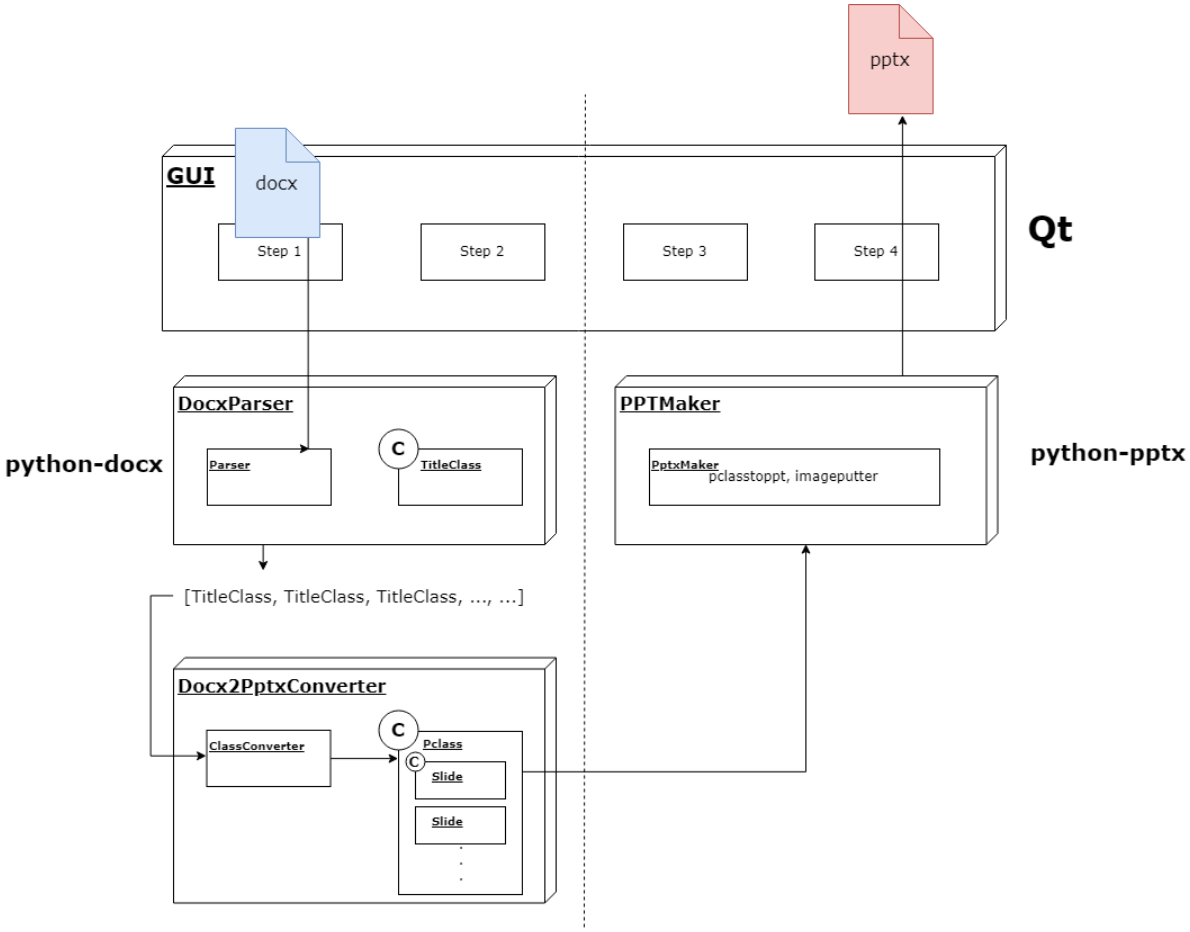

Figure 6. 프로그램 전체 구조도

프로그램은 GUI상에서의 흐름에 맞게 작동된다. GUI에는 step1, step2, step3, step4가 존재한다.

Step1.

step1에서 사용자는 Pptx파일로 변환할 Word파일을 선택하고 이를 입력한다. 이와 동시에 사용자는 프로그램상에서 추출할 부분을 결정하게 된다. 이때 Parser.Py의 set_parserobject가 호출되고 사용자의 선택을 추출시 반영하게 된다. set_parserobject의 결과에 맞게 Word파일을 파싱하고, 파싱된 정보를 Title instance들의 List인 TitleList에 저장한다. TitleList는 Docx2PptxConverter를 통해 실제 PPT Slide를 제작할 때 사용되는 Class인 PClass형태로 변형된다.

Step2.

step2에서 사용자는 Word 파일에서 Slide들이 어떻게 생성될 지를 볼 수 있다. Word파일에서 문서를 크롤링하고 파싱을 할 때 워드 파일에 존재하는 사진 파일도 가져온다. step2에서 사용자는 각 Word 파일에서 크롤링이 된 사진을 어떤 Slide에 삽입할 지 결정할 수 있다. 또한 각 slide에서 어떤 layout으로 생성 할 지도 선택할 수 있다. 변경하지 않아도 무방하나, 제목으로 사용할 slide는 "제목" layout을 선택하는 것을 추천한다. 마지막으로 어떤 Theme로 Pptx파일을 생성할지를 결정하고, 저장될 Pptx 파일의 제목은 어떻게 할지 입력한다.

Step3.

step3에서 사용자는 완성된 Pptx 파일의 저장 디렉토리를 결정할 수 있다. 저장 디렉토리를 선택했으면 다음 단계로 넘어갈 수 있다.

Step4.

"make"버튼을 "pptx file is created"라는 경쾌한 문구와 함께 Pptx파일이 완성된다.

프로그램 주요 기능

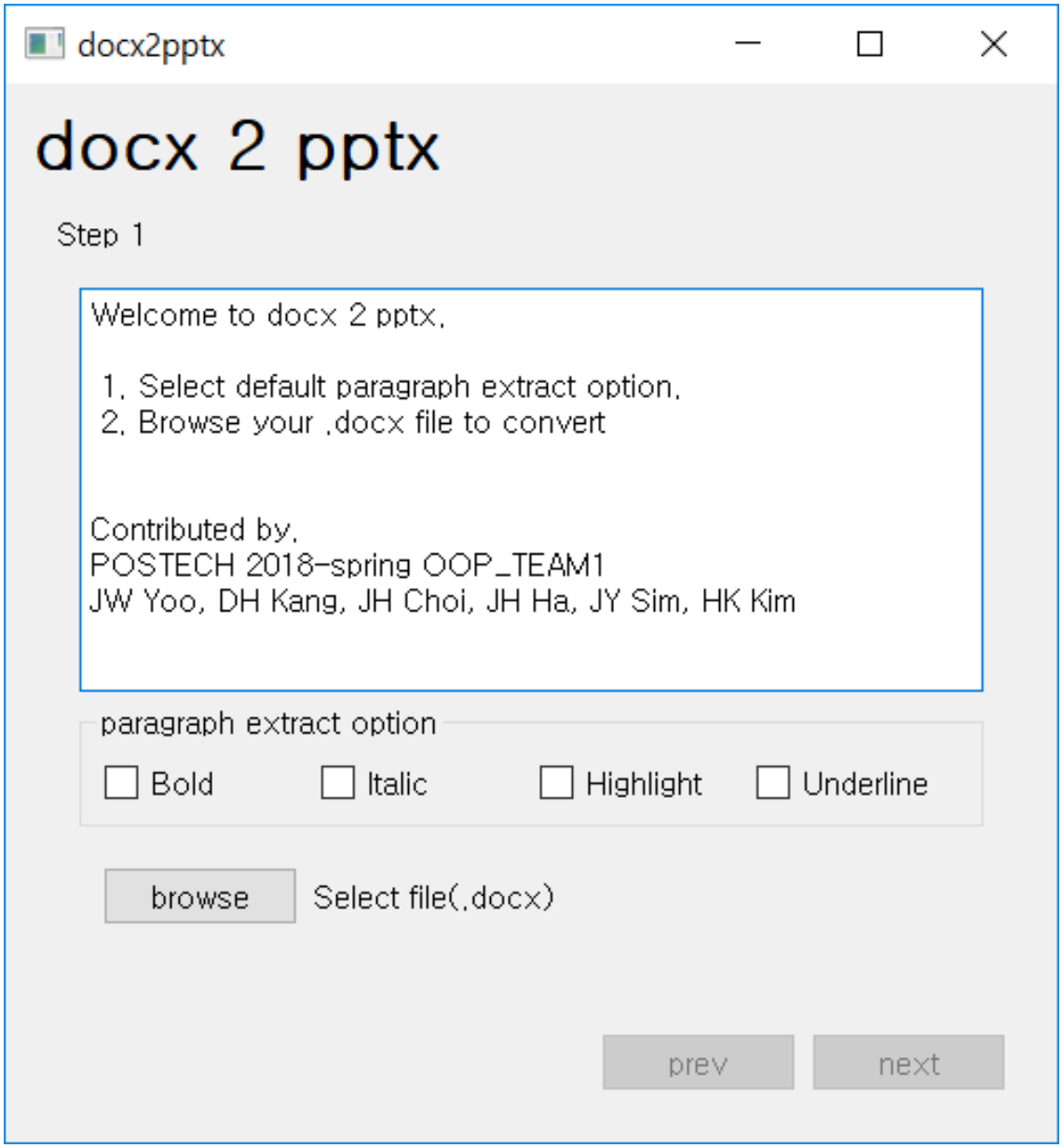

Figure 7. step1_start

우선 프로그램을 실행하면 docx2pptx 환영 문구와 step1 안내 메시지가 뜬다. 사용자는 browse 를 누르고 pptx로 변환을 원하는 docx파일을 로컬에서 선택한다. 그리고 선택한 docx파일 내용 중 추출하고 싶은 옵션을 선택한다.

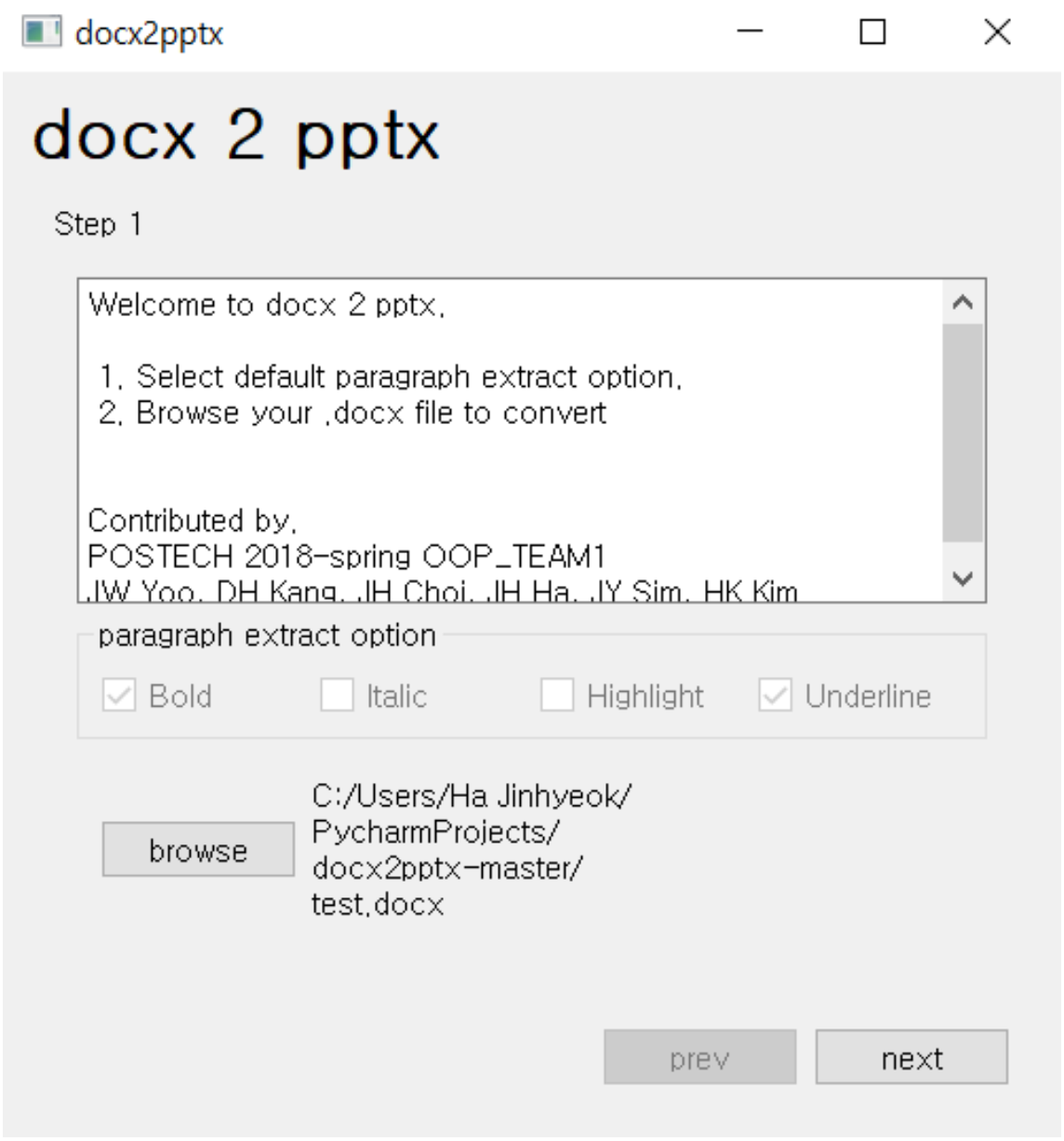

Figure 8. step1_select

예시를 보이기 위해 test.docx 파일을 불러왔다. 추출 옵션은 복수 선택이 가능하므로 Bold와 Underline을 선택했다. next를 누르면 다음 단계로 넘어간다.

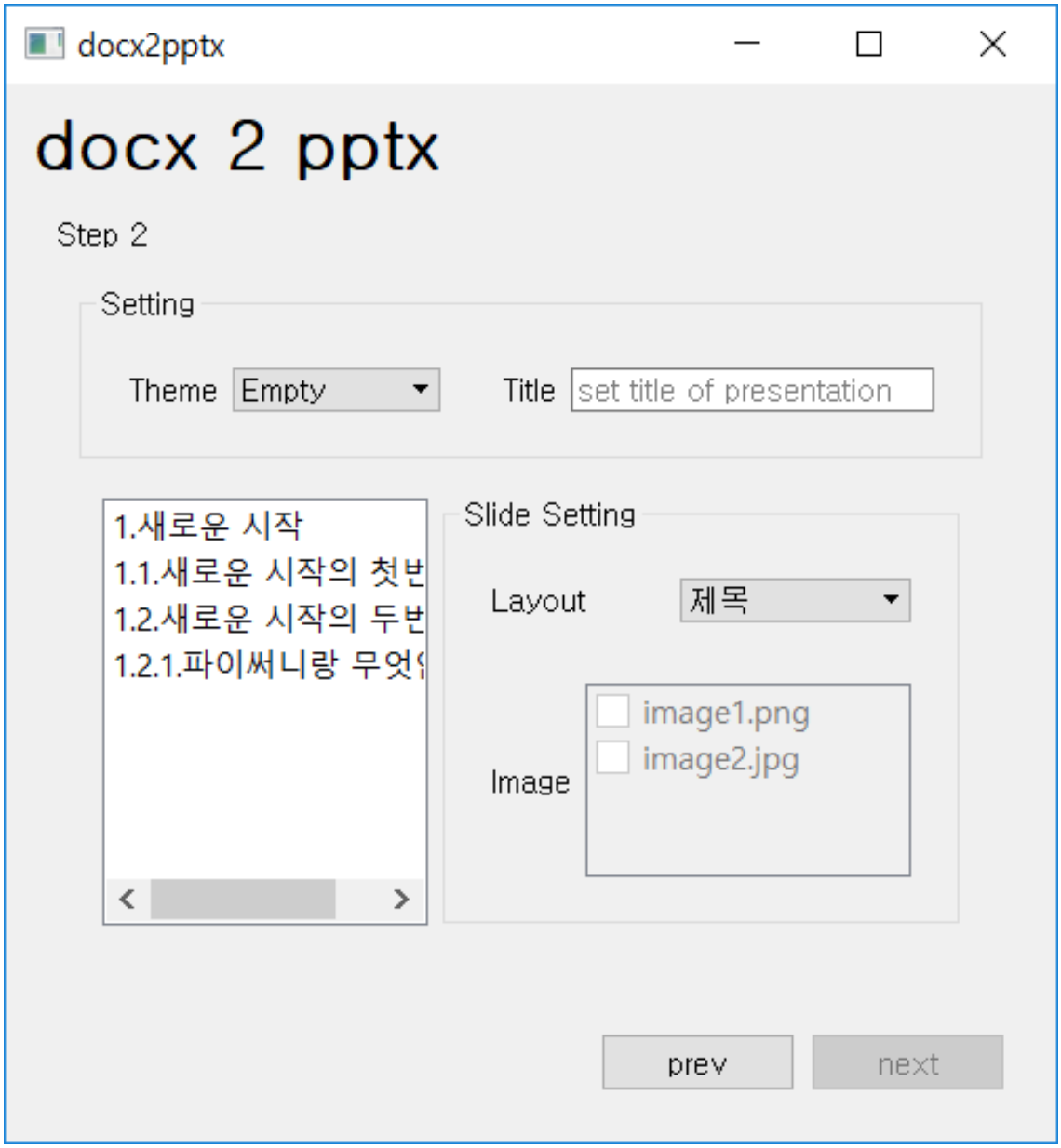

Figure 9. step2_basic

Step2로 넘어오면 템플릿으로 쓰일 theme을 선택하고 title을 입력 받도록 한다.

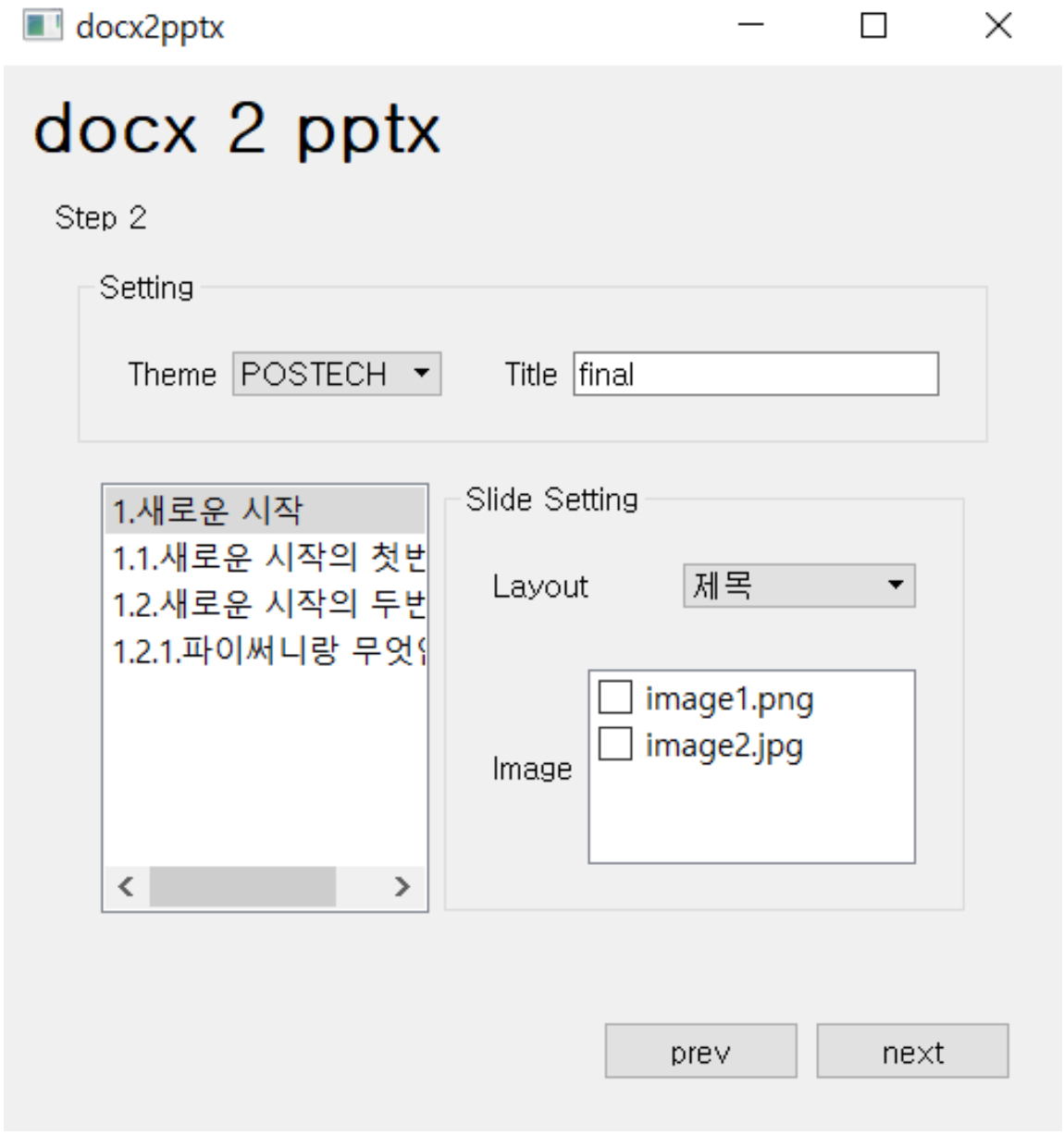

Figure 10. step2_title

템플릿 theme과 title을 입력하면 그 아래에 제목별로 클릭이 가능하고 각각의 제목마다 만들어지는 slide의 layout과 image도 선택한다.

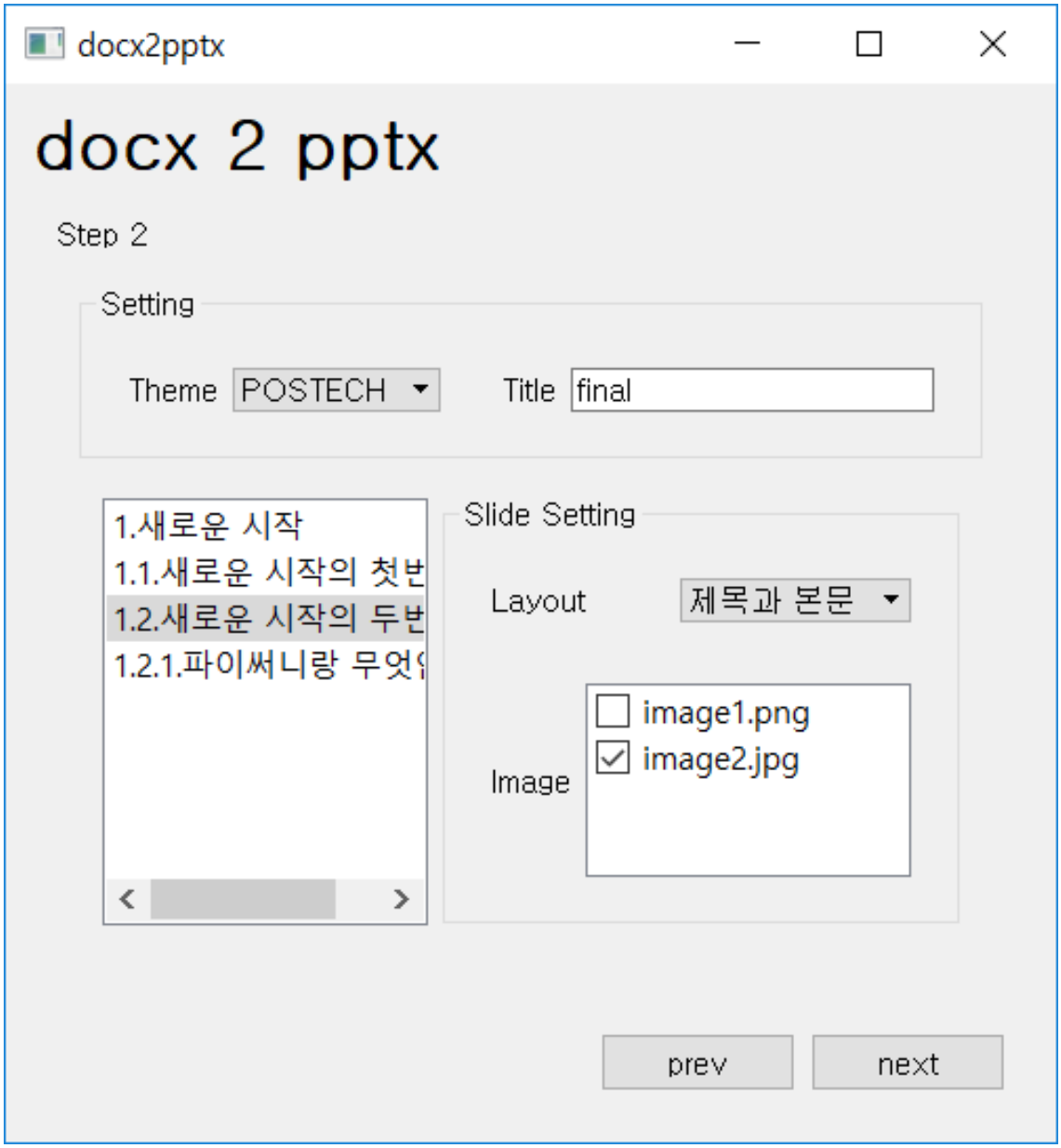

Figure 11. step2_title_image

각 슬라이드마다 layout을 선택해줄 수 있고 선택하지 않으면 default로 '제목과 본문' layout이 설정된다. image의 목록은 사용자가 등록한 docx파일에서 추출된 이미지들이 순서대로 쓰여있다. next버튼을 누르면 다음 단계로 넘어간다.

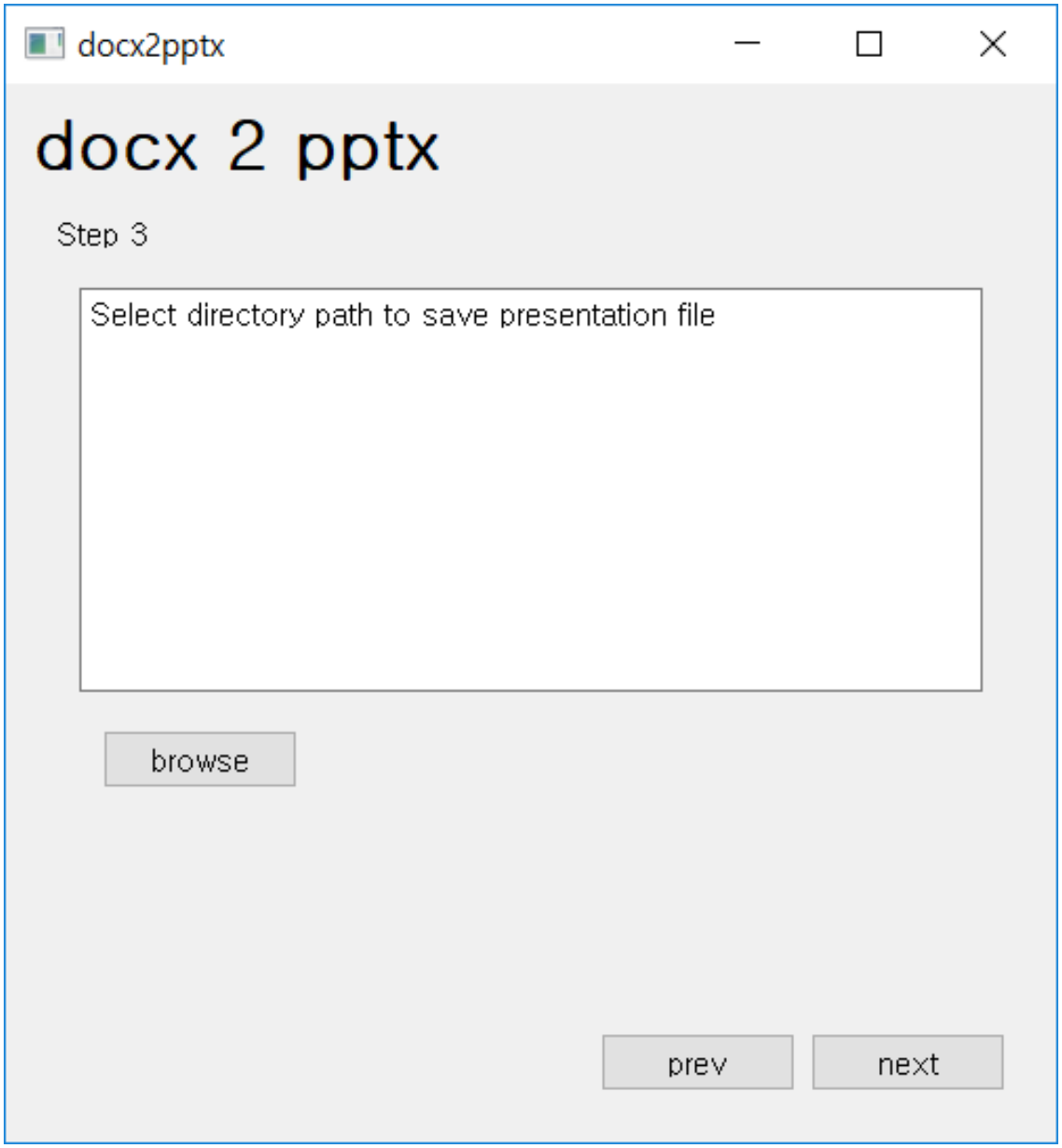

Figure 12. step3_browse

Step3로 넘어오면 완성된 pptx 파일을 저장할 directory를 선택하도록 한다. browse버튼을 누르고 저장할 로컬 공간을 선택한다.

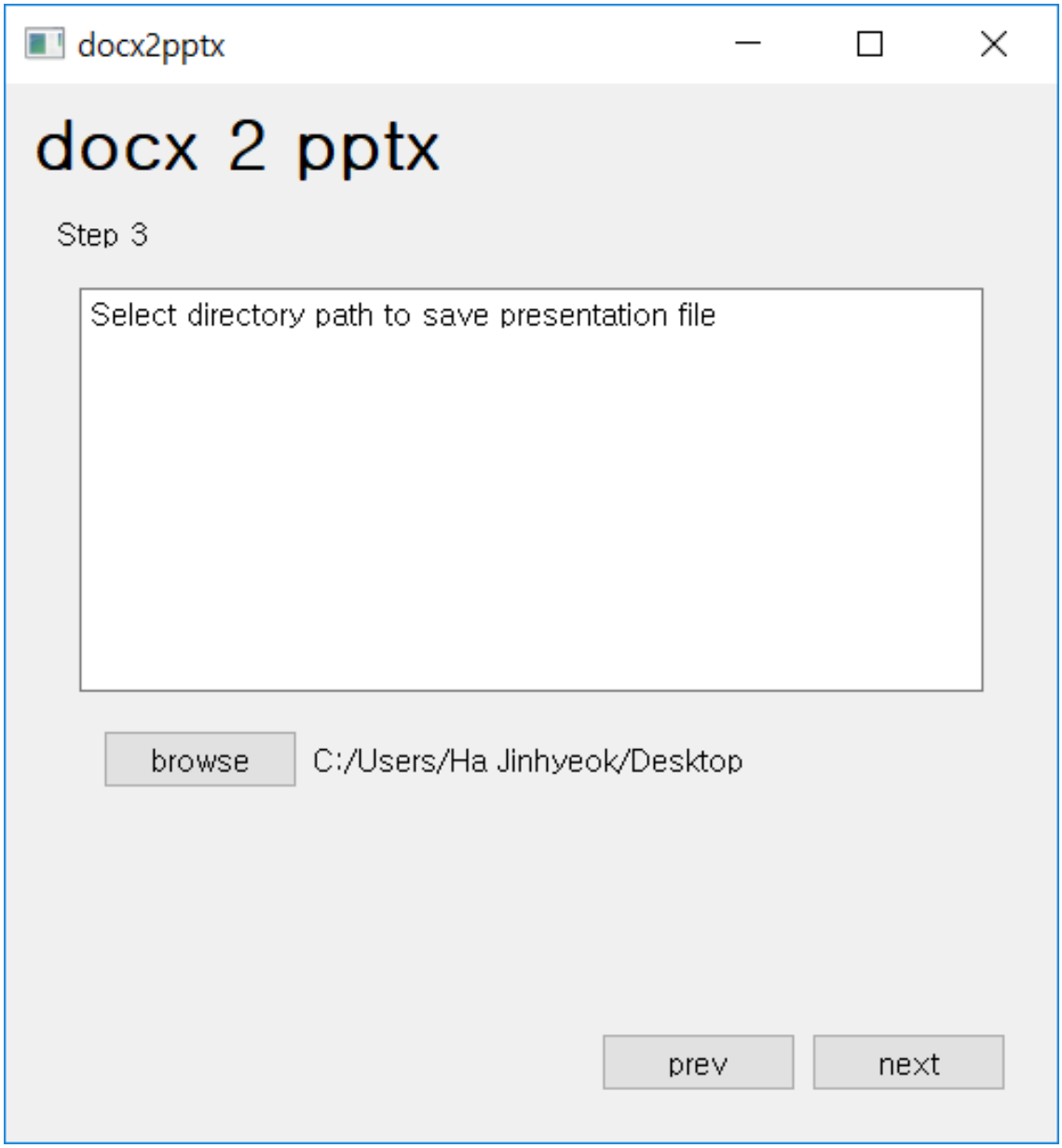

Figure 13. step3_select

Directory를 선택하면 next버튼이 활성화된다. 예시는 편의를 위해 바탕화면을 directory로 지정했다. next를 누르면 다음 step으로 넘어간다.

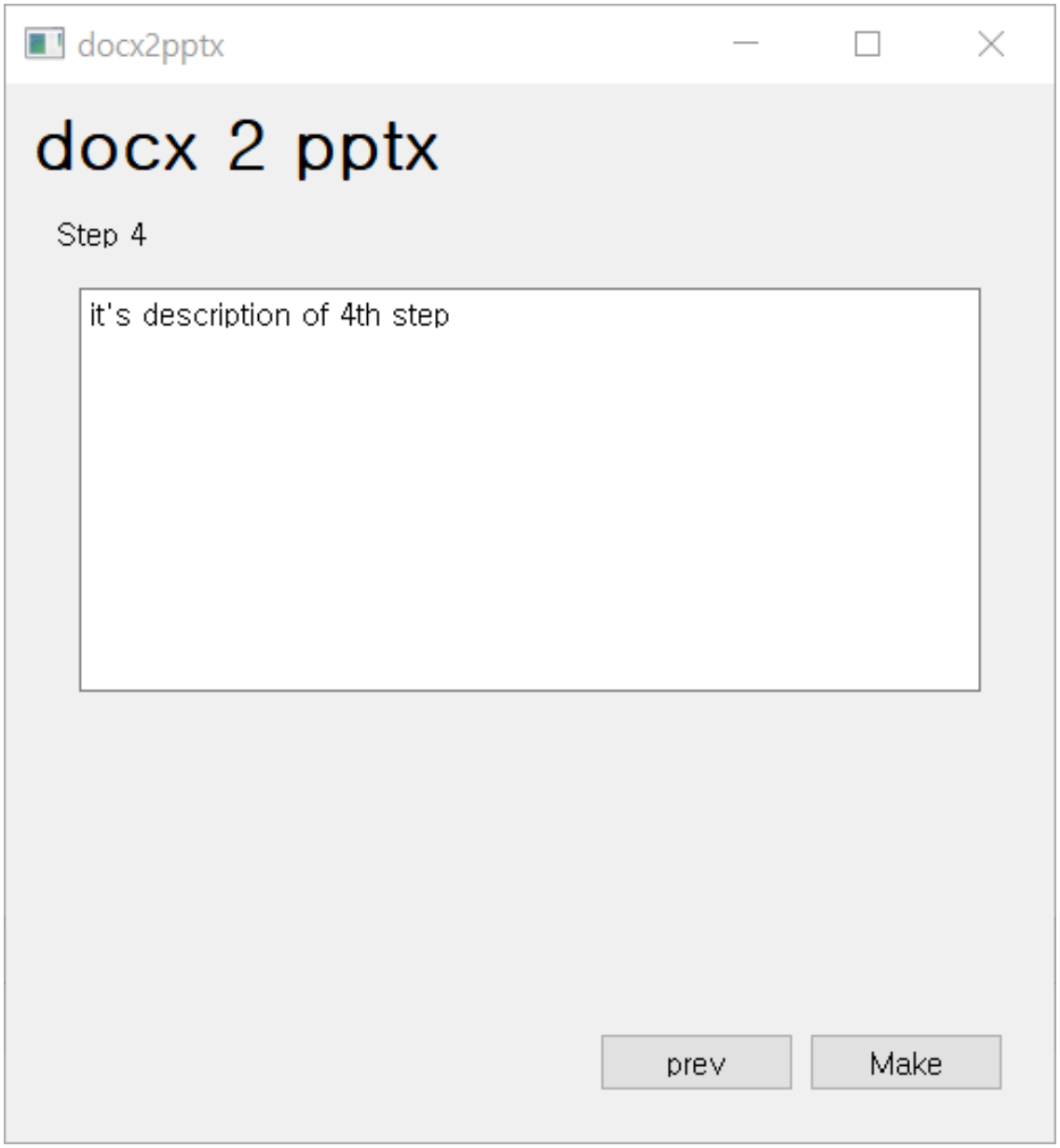

Figure 14. step4_description

Step4에서는 최종적으로 pptx파일을 생성해낸다. make 버튼을 누르면 step3에서 설정했던 directory에 step2에서 쓴 title을 가진 pptx파일이 생성된다.

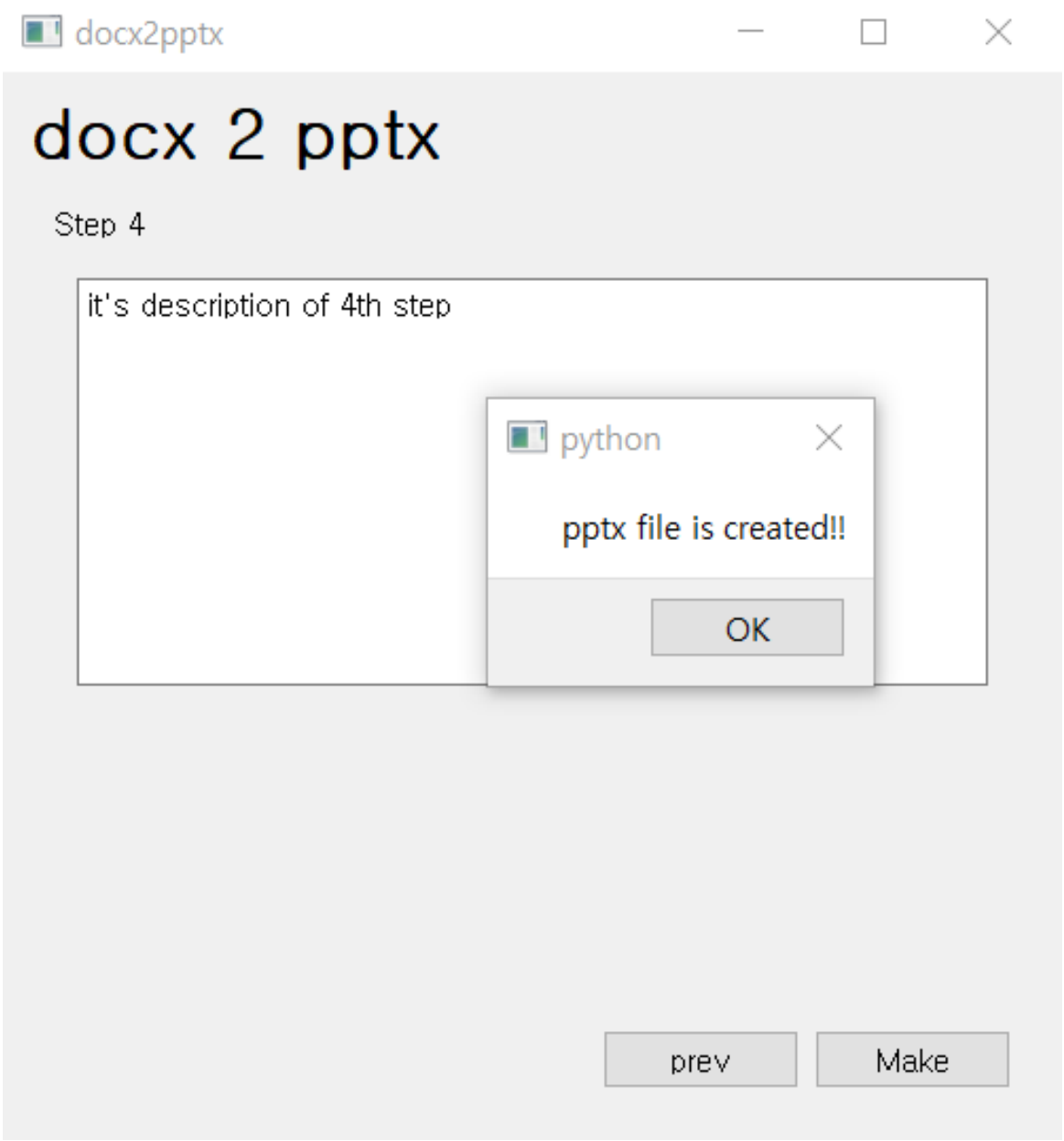

Figure 15. step4_make

Make 버튼을 누르면 사진과 같은 문구가 뜬다. OK를 누르면 바탕화면에 만들어진 final.pptx파일을 확인할 수 있다.

Figure 16. 생성된 pptx

위와 같이 생성된 것을 확인할 수가 있다.

Figure 17. final_pptx

최종적으로 만들어진 pptx 파일이다.

프로그램 전체 구조도 및 클래스 설계도

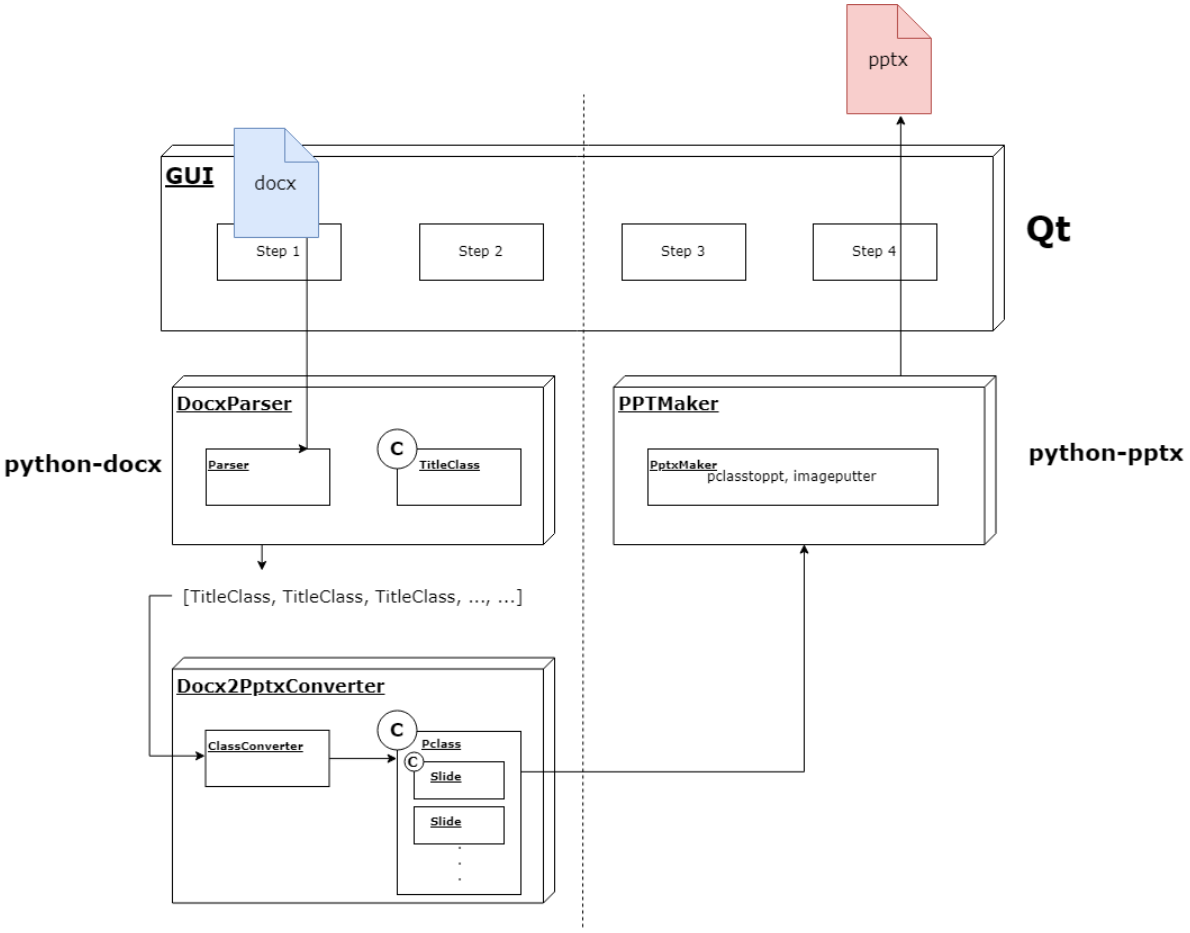

Figure 18. 프로그램 전체 구조도

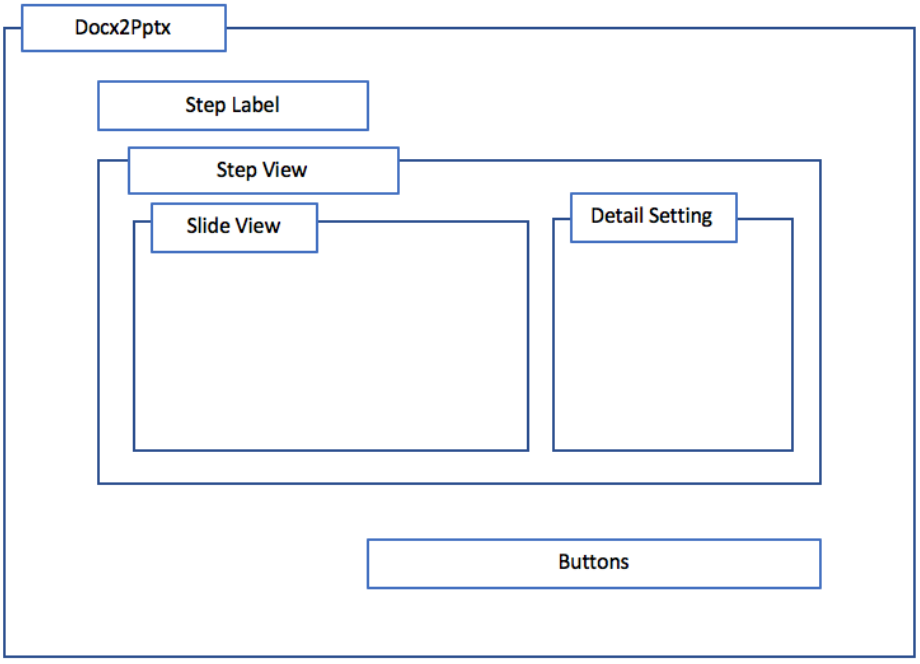

Figure 19. UI diagram

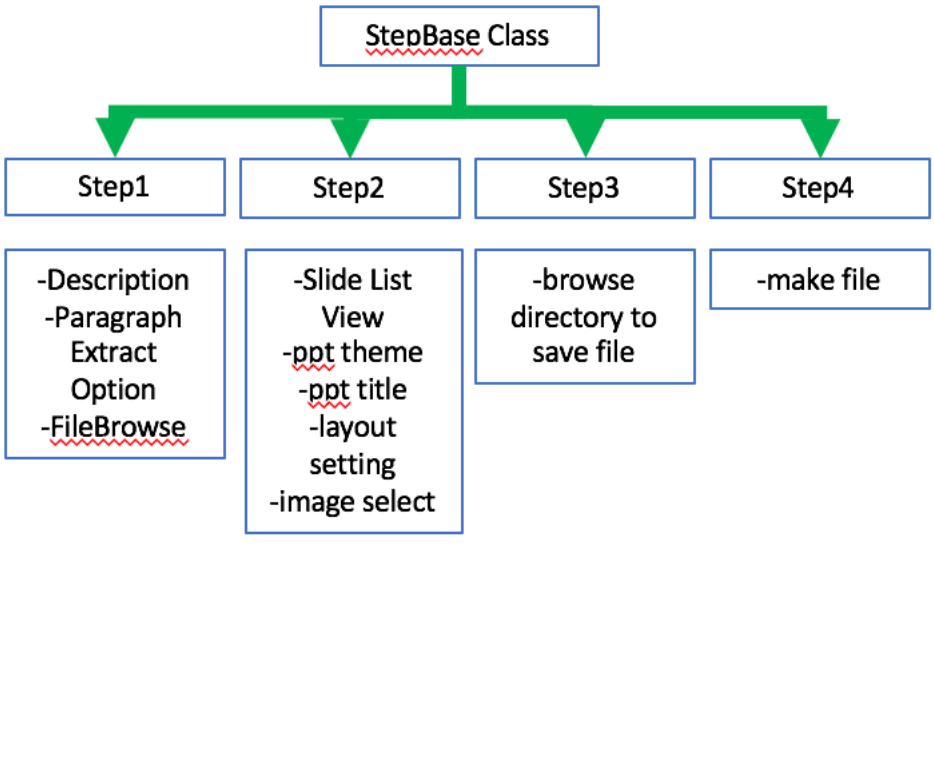

Figure 20. UI 설계도