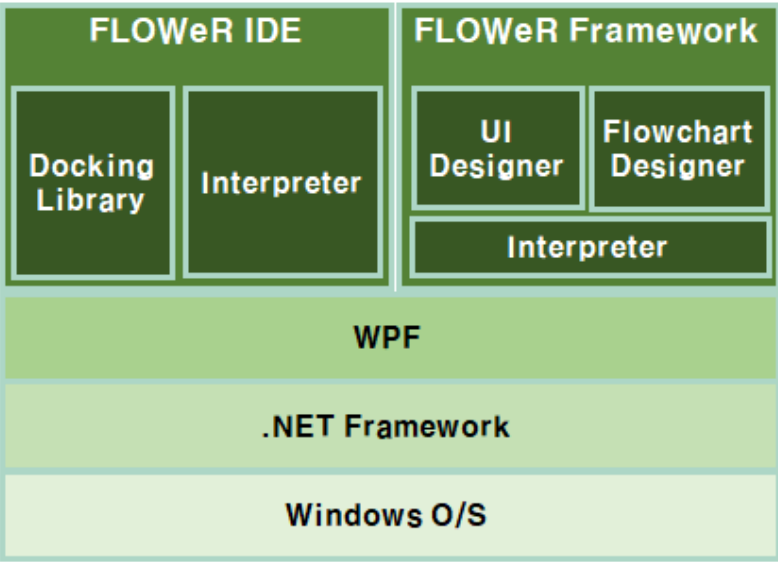
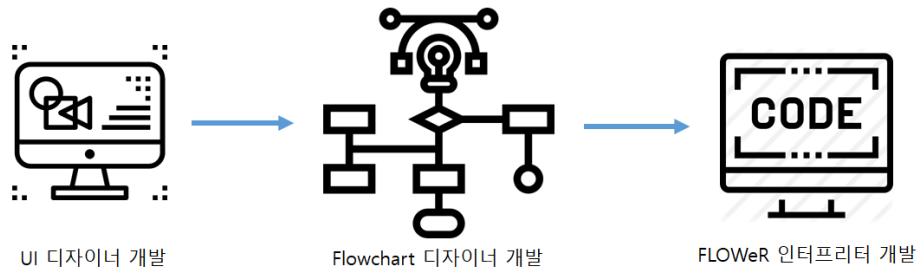


순서도 기반의 교육용 비주얼 프로그래밍 프레임워크

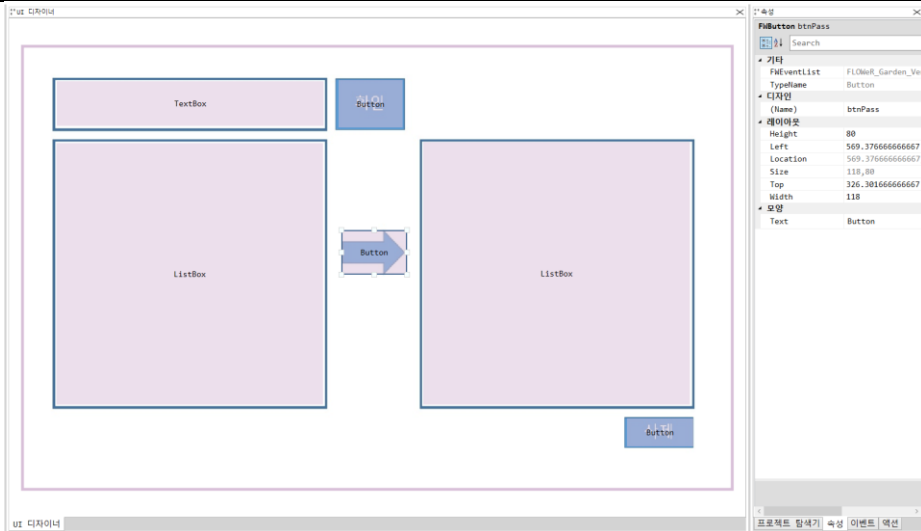
프로젝트명	순서도 기반의 교육용 비주얼 프로그래밍 프레임워크		
개발기간	2018.03~2018.06(3개월)	개발인원	4명
개발환경	Windows10, .Net Framework, WPF, Visual Studio 2017		
개발 배경	<p>4차 산업혁명 시대에서 프로그래밍이 필수과목으로 주목을 받고 우리나라 또한, 2018년도부터 초등학생, 중학생, 고등학생을 대상으로 프로그래밍 교육을 의무화 하기 시작하였다. 현재 프로그래밍 과정은 컴퓨터 관련학은 물론 건축, 의학, 생명공학, 우주항공에 이르기까지 다양한 분야에서 사용되고 이러한 프로그램을 배우기 위해서는 알고리즘 및 자료구조에 대한 이해가 필요하다. 하지만 많은 사전 지식을 필요로 하며, 수학적 사고 능력 및 프로그래밍 능력을 필요로 하므로 프로그래밍을 처음 접하는 사람들에게 많은 어려움이 있다. 이 같은 프로그래밍을 배우기 앞서 현재 초심자들을 위한 교육용 프로그래밍 언어들이 많이 있는데, 이와 같은 교육용 프로그래밍 언어들은 텍스트 코딩 방식보다 블록을 사용하는 비주얼 프로그래밍 기법을 사용한다. 하지만 흔히 사용되는 스크래치, 엔트리와 같은 교육용 프로그래밍 언어는 교육을 위해 제작되어진 자체적인 환경에서만 돌아가고 간단한 알고리즘에 대해서만 구현이 가능하다는 단점이 있다.</p> <p>이러한 문제를 해결하고자 프로그램의 UI와 이벤트의 로직에 대한 구현을 블록 프로그래밍 기법을 이용하여 윈도우 프로그램을 별도의 코딩 작업 없이 개발하여 기존 교육용 프로그래밍 언어들이 지원하지 않던 GUI와 확장성 있는 프로그램을 개발 가능하도록 유연한 개발환경을 제공하는 새로운 프레임워크를 개발하였다.</p>		
아키텍처	 <p>The diagram illustrates the system architecture of the FLOWer IDE and Framework. It is structured as follows:</p> <ul style="list-style-type: none"> FLOWer IDE (Top Left): Contains a Docking Library and an Interpreter. FLOWer Framework (Top Right): Contains a UI Designer, a Flowchart Designer, and an Interpreter. WPF (Middle): Serves as the presentation layer for both the IDE and the Framework. .NET Framework (Bottom): The core runtime environment. Windows O/S (Base): The operating system layer. <p>[시스템 아키텍처]</p>		

개발내용

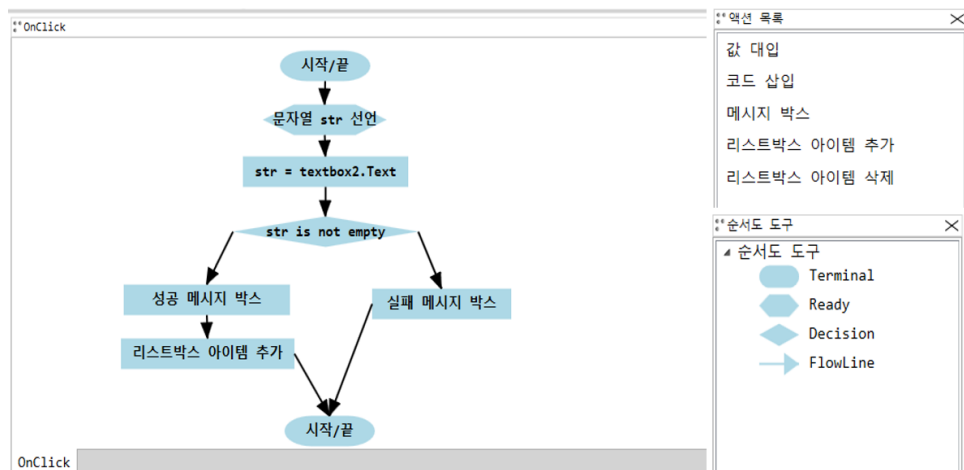


- 프레임워크의 효율적인 개발을 위한 통합개발환경
- IDE의 유연한 화면 배치를 위한도킹 라이브러리
- 프로그램의 UI를 작성하기 위한 UI 디자이너
- 프로그램의 Event를 작성하기 위한 Flowchart 디자이너
- 디자이너를 해석하여 코드를 생성하기 위한 인터프리터

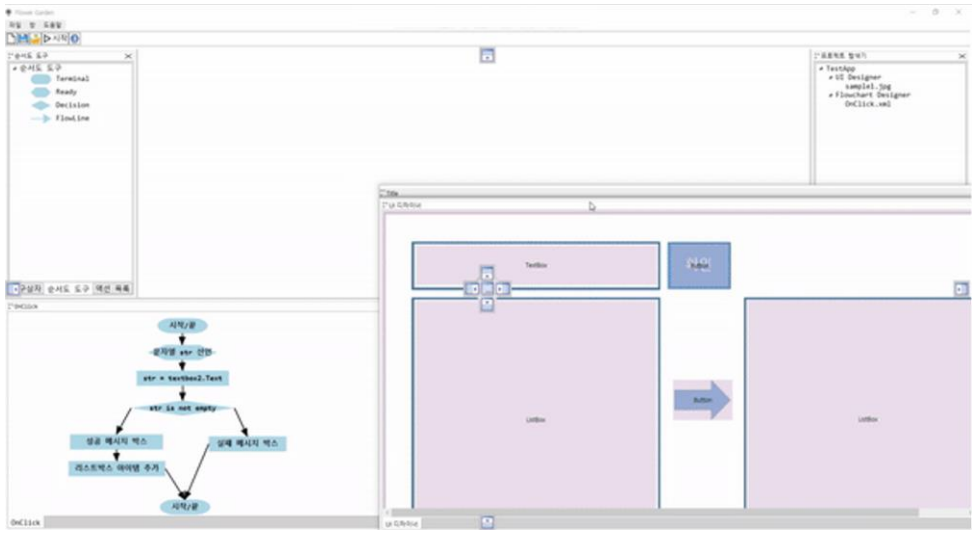
담당업무



UI 디자이너



Flowchart 디자이너

	<div><pre><?xml version="1.0" encoding="utf-8"?> <Terminal> <Ready Type="문자열" Name="str" /> <Process ProcessName="InputValue" VarName="str" VarValue="textbox2.Text" /> <Decision Condition="str != string.Empty"> <Process ProcessName="MessageBox" Message="\""추가되었습니다."\" /> <Process ProcessName="ListBoxAdd" ListBoxName="lboxFirst" ListBoxItem="str" /> </Decision> <Decision Condition="!(str != string.Empty)"> <Process ProcessName="MessageBox" Message="\""값을 입력해주세요."\" /> </Decision> </Terminal></pre><div>→</div><pre>private void OnClick(object sender, EventArgs e) { string str; str = textbox2.Text; if (str != string.Empty) { System.Windows.Forms.MessageBox.Show("추가되었습니다."); lboxFirst.Items.Add(str); } else { System.Windows.Forms.MessageBox.Show("값을 입력해주세요."); } }</pre></div> <p>인터프리터</p>  <p>도킹 라이브러리 및 통합개발환경</p>
고찰	<p>프로젝트를 진행하면서 WPF와 MVVM패턴, 데이터 바인딩에 학습이 가능.</p> <p>WPF의 사용이 상대적으로 미흡하여 그래픽 엔진에 대한 퍼포먼스가 부족.</p>