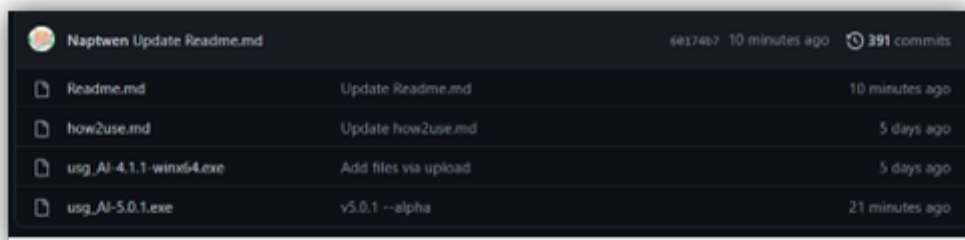


USG AI 엔진 프로그램 사용설명서

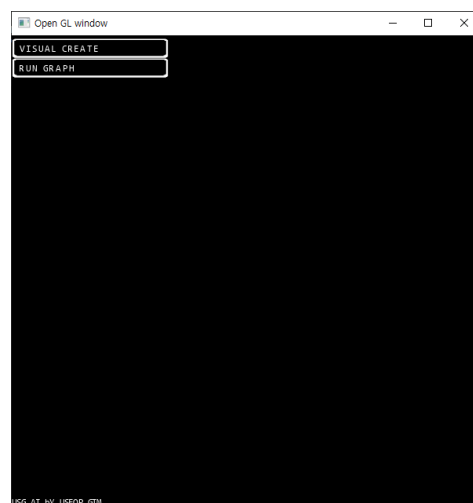
본 프로그램의 소스코드는 모두 본인이 제작하였으며, GNU AFFERO LICENSE V3 를 통해 보호됨과 동시에 특별조항을 명시합니다. 또한 특정 저작권에 속하는 OpenGL, OpenCL, OpenCV, base64 코드는 프로그램에 첨부된 저작권을 확인 하여주시기 바랍니다.

프로그램 설치법

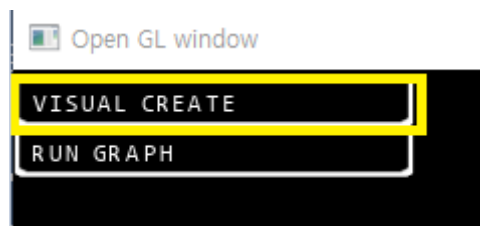
1. 프로그램의 원활한 실행을 위한 사용 소프트웨어 OpenGL, OpenCV, OpenCL을 설치합니다.
2. <https://github.com/Naptwen/usgAI> 에 접속하여 가장 최신버전의 프로그램 설치파일 받습니다.



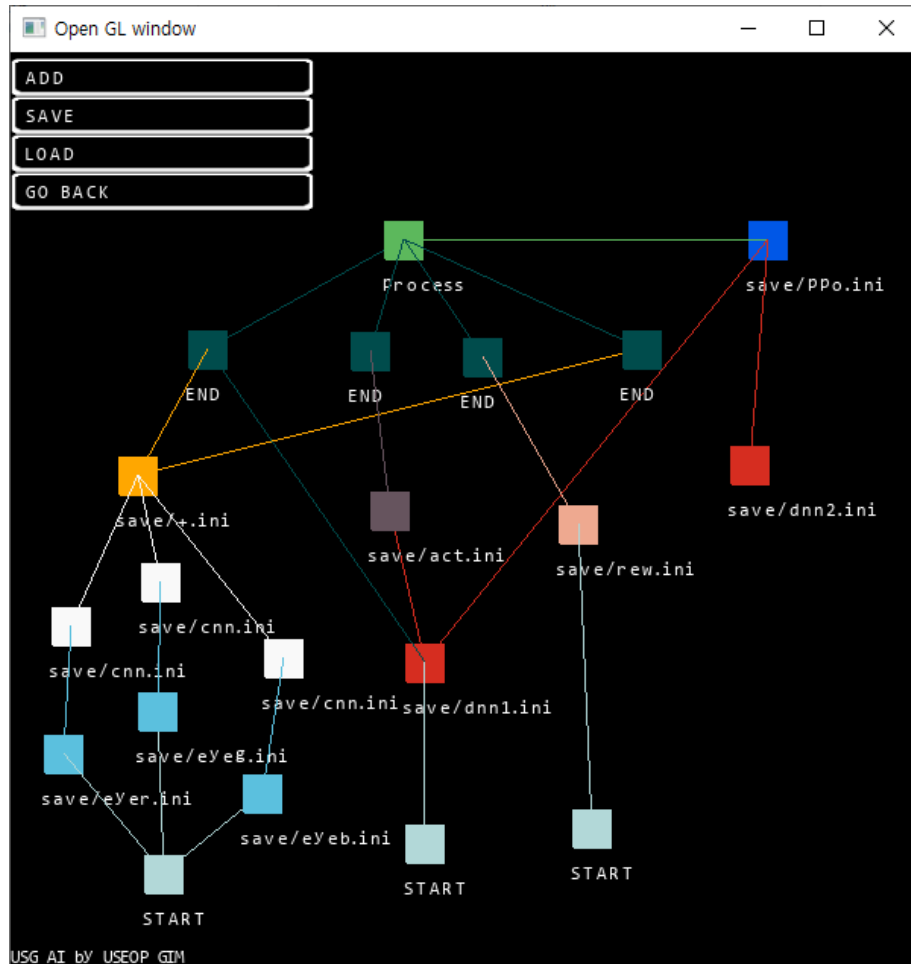
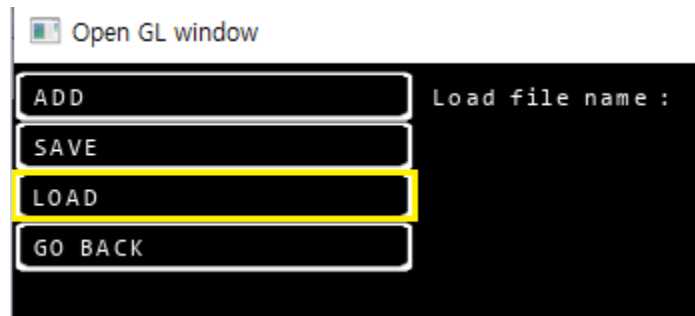
3. 설치된 폴더의 usg_AI 실행파일을 실행합니다.



- 위와 같은 화면이 실행된다면 프로그램 설치가 완료되었음을 의미합니다.
4. 프로그램의 테스트를 위하여 VISUAL CREATE 버튼을 클릭합니다.

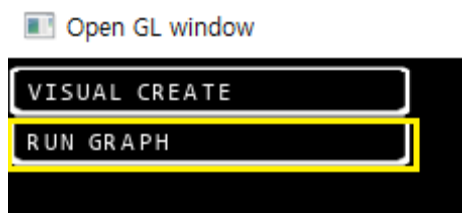


5. LOAD 버튼을 클릭후 “save/test.txt” 를 입력한후 엔터키를 누릅니다.

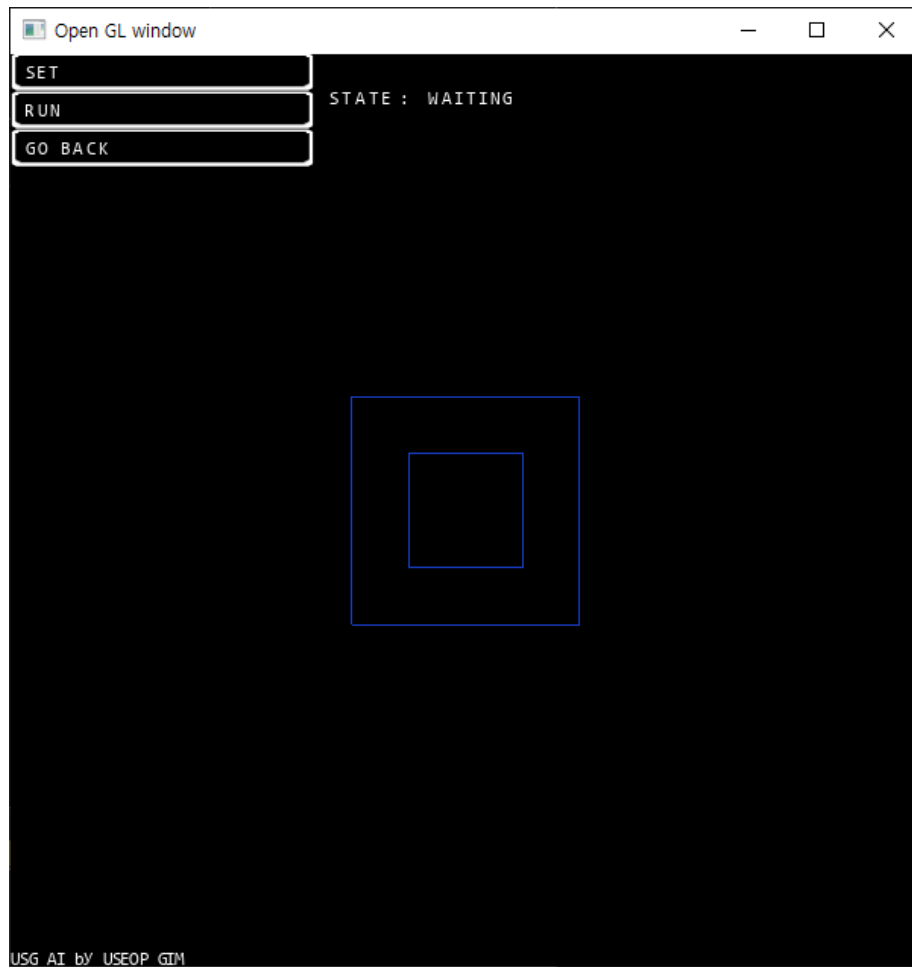


- 위와 같은 화면이 나오지 않는다면, save 폴더안에 예제 파일들이 존재하는지 확인합니다.

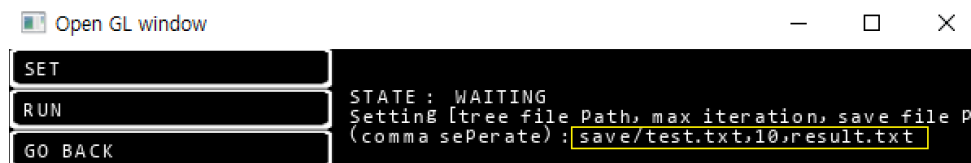
7. RUN GRAPH 버튼을 클릭합니다.



8. 밑의 화면과 같은 화면이 나오면 SET버튼을 클릭합니다.



9. 버튼을 클릭한후 “save/test.txt, 10, result.txt” 를 입력합니다.



10. 다음으로 RUN 버튼을 클릭합니다.
11. 중앙의 사각형이 움직이면서 방향키가 자동으로 움직이면 프로그램이 작동하는 것입니다.
12. 중앙의 사각형의 움직임이 멈추면 프로그램을 종료하셔도 됩니다.

기본 콘솔 인터페이스 설명

각모델들은 편집파일을 통해 직접 작성 및 편집 혹은 기본 제공되는 콘솔 인터페이스 프로그램을 통해서 작성 할 수 있습니다.

--help : 기본 제공되는 기능을 살펴 볼 수 있습니다.

--license : 라이선스 정보를 표시합니다.

--create "type": 콘솔인터페이스를 통해서 모델들을 작성할 수 있습니다.

```
[type]
> CNN      convolutional network layer
> NER      neural network
> OPT      intersection for operation
> OBS      observer of enviroments
> ACT      action function
> REW      reward function
> END      end flow
> STR      start flow
> PRO      process function
> RL       reinforcement model
> GRAPH    graph consturct create
```

CNN : Convolutional Neural Network 블록을 작성합니다.

NER : Deep Neural Network 블록을 작성합니다.

OPT : 사칙연산 블록을 작성합니다. (version 5 는 합과 곱만을 제공합니다.)

OBS : 화면에 관한 공유 라이브러리 블록을 작성합니다.

ACT : 행동에 관한 공유 라이브러리 블록을 작성합니다.

REW : 보상에 관한 공유 라이브러리 블록을 작성합니다.

END : 그래프 흐름의 끝을 의미하는 블록을 작성합니다.

STR : 그래프 흐름의 시작을 의미하는 블록을 작성합니다.

PRO : 그래프 흐름을 제어할 블록을 작성합니다.

RL : 강화학습 모델 블록을 작성합니다.

GRAPH : 그래프를 작성합니다.

프로그램에 실제 사용되는 것은 CNN, NEW, OBS, ACT, REW 들입니다.

GRAPH, OPT, END, STR 은 GUI 프로그램을 통해서 작성가능합니다.

--setting : 각종 설정을 조절합니다.

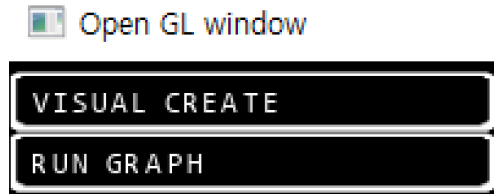
--run “불러올 파일 이름” “최대 실행 횟수” “저장 파일이름” : 프로그램을 실행합니다.

불러올 파일 이름 : 반드시 그래프 흐름 파일이어야합니다.

최대 실행 횟수 : 프로그램의 최대 실행 횟수입니다.

저장 파일 이름 : 저장할 파일 이름을 의미합니다.

그래픽 인터페이스 설명



VISUAL CREATE : 제어흐름도를 작성하기 위한 프로그램을 실행합니다.

RUN GRAPH : 그래프 파일을 동작하기 위한 프로그램을 실행합니다.

VISUAL CREATE

먼저 블록이란 각 알고리즘을 실행할 단위입니다.

본 머신러닝 엔진프로그램은 이러한 블록들의 관계를 통해 제어흐름을 조절하고 보다 쉽게 인공지능의 다양한 알고리즘 구현에 있어서 관계도를 통하여 작성하기 쉽도록 구성되어있습니다.

ADD : 블록을 추가합니다.

SAVE : 현재 선택된 블록과 그 블록의 자식블록들을 모두 저장합니다.

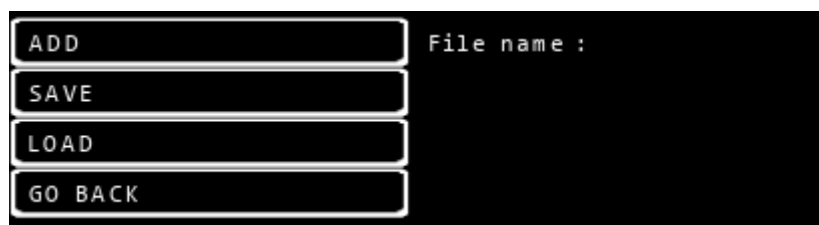
LOAD : 그래프 파일을 불러옵니다.

사용법

1. ADD를 클릭하면 화면에 **empty**라는 블록이 생성됩니다.
2. 블록을 왼쪽 클릭하면 블록주변에 테두리가 나타나며 현재 선택된 블록을 나타냅니다.



3. 블록 오른쪽 클릭하면 파일이름을 입력할 수 있습니다.



4. 파일의 종류에 따라 블록의 색이 적용되며 아래와 같습니다.

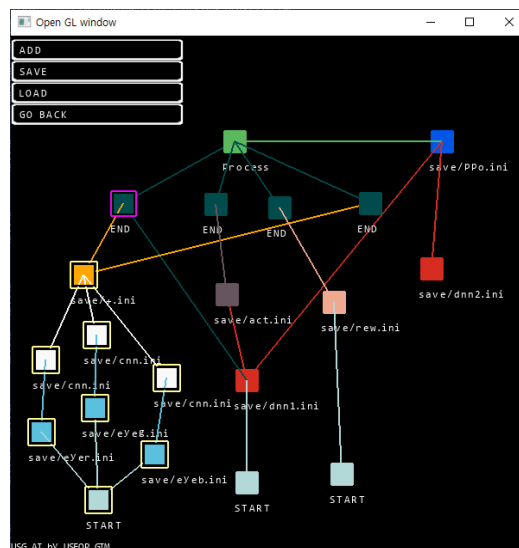


파일명 입력시 “+”, “x”, “END”, “START” 들은 파일이 없어도 각각의 블록들로 자동으 로 변하게 됩니다.

5. 왼쪽 클릭 후 다른 블록을 클릭하게 되면 각 블록끼리 연결되며 선택했던 블록이 자식 블록이 되고 마지막에 클릭한 블록이 부모 블록이 됩니다.



6. 부모 블록을 클릭시 연결된 자식 블록은 자동으로 표시해줍니다.



7. 같은 블록을 두 번 왼쪽 클릭시 해당 블록에 연결된 모든 관계가 제거됩니다.

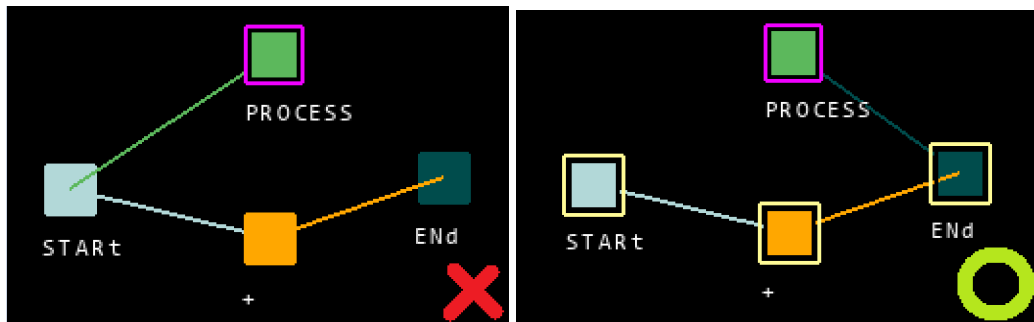


관계도 설명

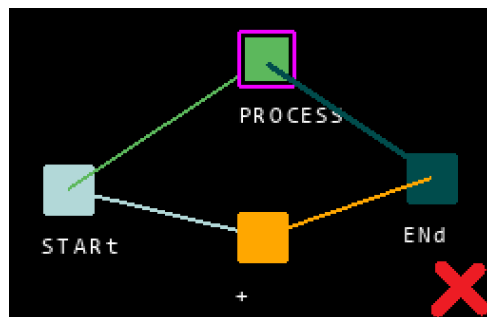
1. 프로그램의 실행과 끝을 의미하는 **START**와 **END** 블록은 반드시 각 블록관계의 처음과 끝에 존재하며, 쌍을 이루어야 합니다.



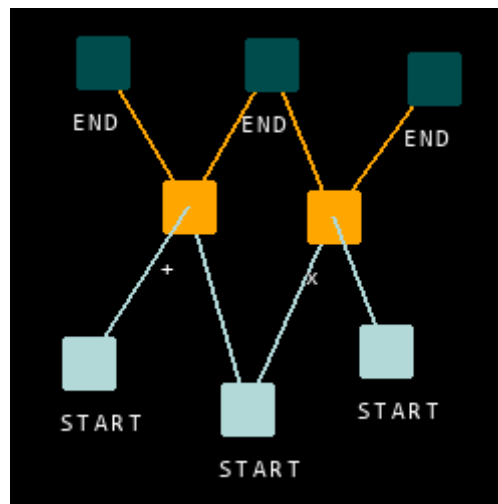
2. 프로그램의 실행순서를 제어할 **PROCESS**블록이 **END**블록의 부모블록으로 존재해야 합니다.



주의 : 부모가 다시 자식으로 연결되는 순환구조는 **프로그램을 강제종료**시킵니다.

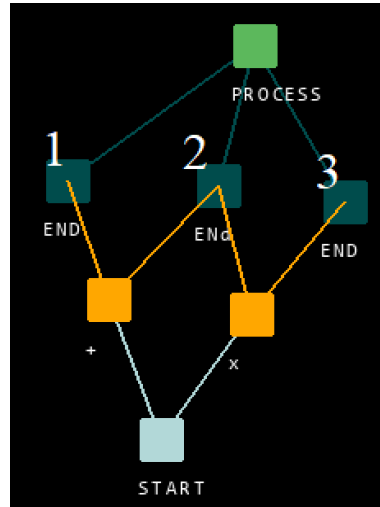


3. 한블록이 가질 수 있는 부모와 자식의 수는 제한이 없습니다.

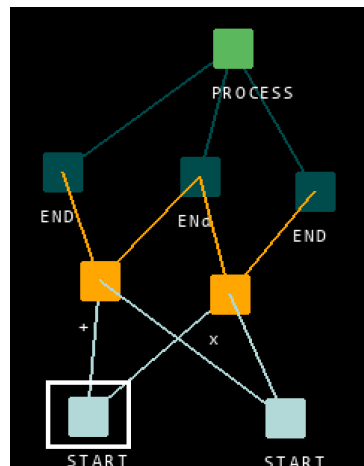


하지만 저장시 반드시 선택된 블록과 자식블록들만 저장됩니다.

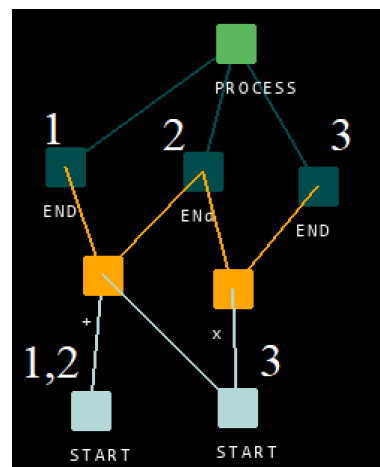
4. 프로그램의 실행 순서는 **PROCESS**에 연결된 **END**블록중 가장 왼쪽부터 실행됩니다.



5. 마찬가지로 **END**블록은 자식 블록중 가장 왼쪽의 **START**블록으로부터 시작합니다



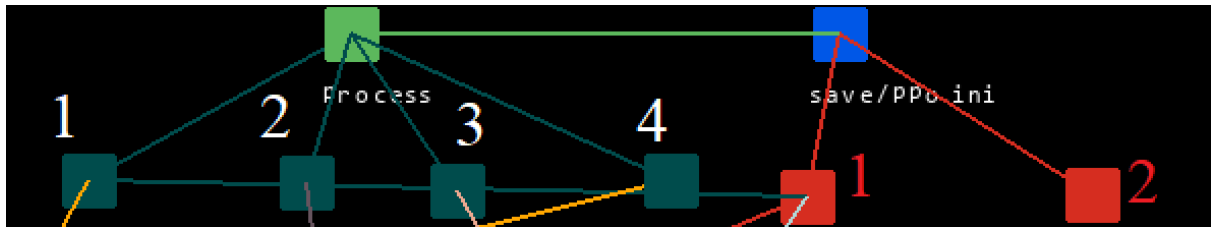
위와 같은경우는 가장 왼쪽의 **START**만 작동합니다.



위와 같은경우는 1번째 2번째 **END** 블록은 가장 왼쪽의 **START**와 연결되며 3번째 가장 오른쪽 **START**와 연결됩니다.

강화학습 블록

강화학습 블록은 반드시 4개의 END를 가진 PROCESS를 자식노드로 가지며, 서로다른 뉴럴네트워크 블록을 강화학습 방법에 따라 자식 노드를 가집니다.



- END 블록의 1번은 과거 상태를 나타냅니다. 2번은 행동을 나타냅니다. 3번은 보상을 나타냅니다. 4번은 행동후 변한 상태를 나타냅니다.
- 강화학습 블록의 뉴럴네트워크 중 가장 왼쪽의 (그림의 1번) 이 모든 뉴럴 네트워크의 메인모델이됩니다. (실제 행동을 결정하는 네트워크)

공유 라이브러리 함수 모양

```
extern "C" __declspec(dllexport) void statefn(std::vector<int> &src, std::vector<float> &dist);
extern "C" __declspec(dllexport) float rewardfn();
extern "C" __declspec(dllexport) float actionfn(const std::vector<float> &src);
```

설치된 파일의 src/extrafn.cpp파일을 참고하세요