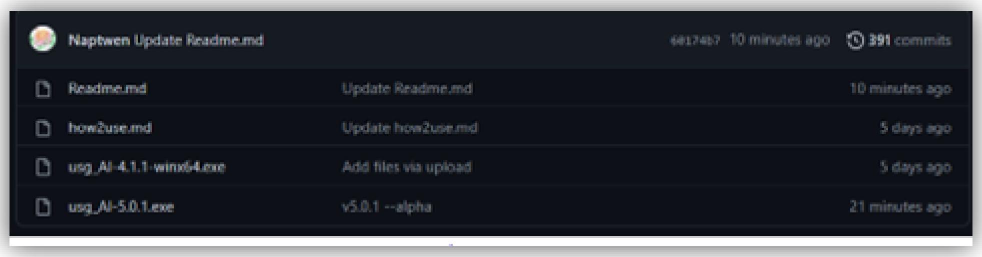
USG AI USER GUIDE

How to intall

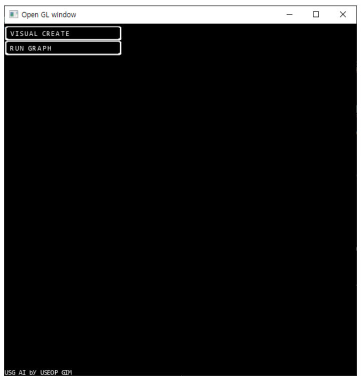
1. For working on the GUI and GPGPU. the pre-installing is required for the OpenGL 3.0, OpenCV 2.0, OpenCL 2.0
2. Access the https://github.com/Naptwen/usgAI then download the newest version



Naptwen Update Readme.md 6817467 10 minutes ago 391 commits

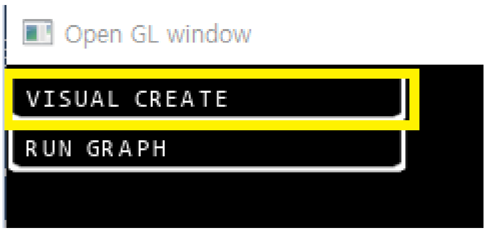
Readme.md	Update Readme.md	10 minutes ago
how2use.md	Update how2use.md	5 days ago
usg_AI-4.1.1-winx64.exe	Add files via upload	5 days ago
usg_AI-5.0.1.exe	v5.0.1 --alpha	21 minutes ago

3. After installing for the test, double click the usg_AI.exe file

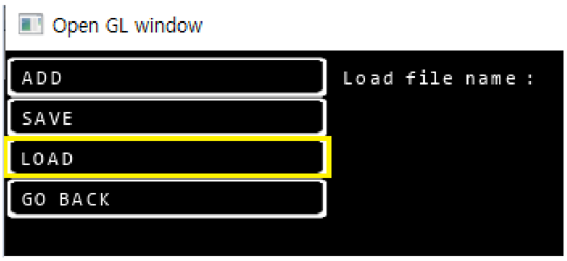


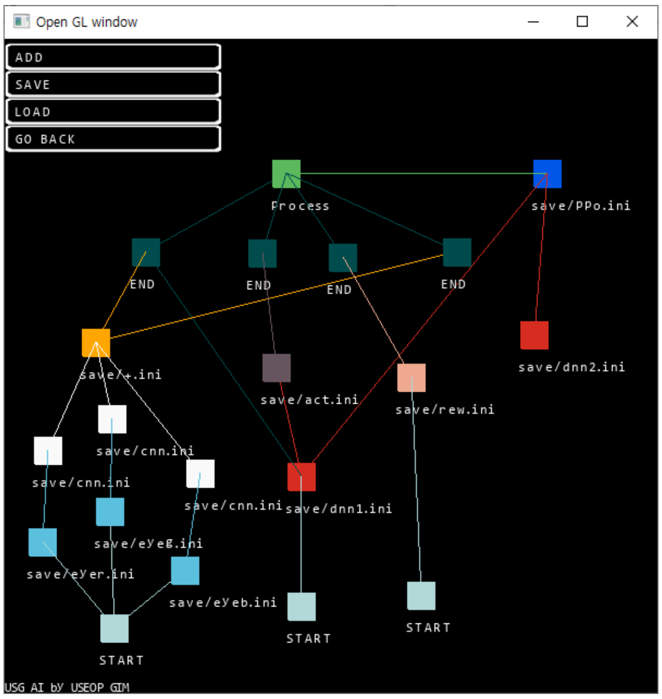
- If the program is working, you can see above image

4. For testing, click the VISUAL CREATE button



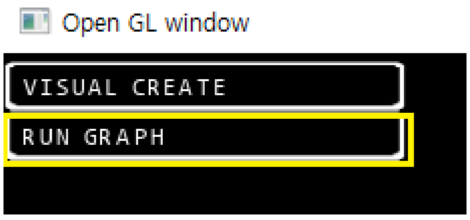
5. Next click the LOAD button then type “save/test.text’”



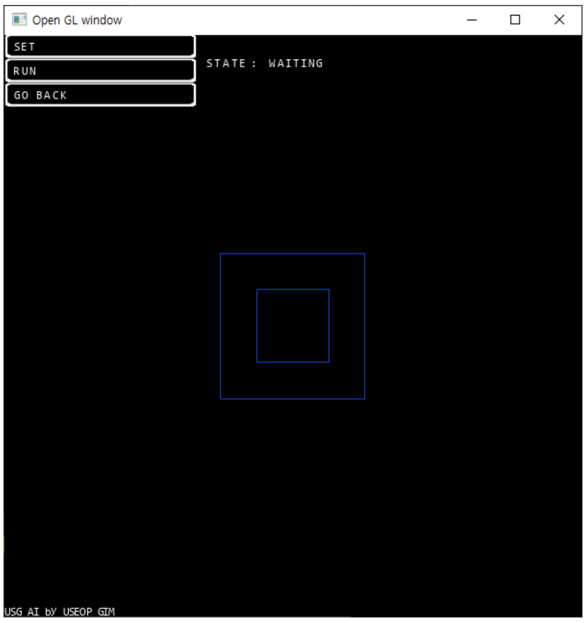


- If you can't see like above, please re-installing the file or checking the save folder in usg_AI

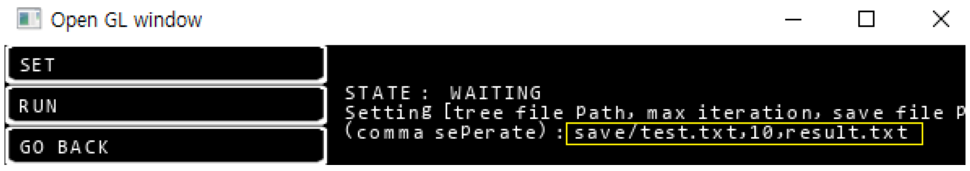
7. Click the GOBACK button then click the RUN GRAPH



8. In the RUN GRAPH screen, click the SET button



9. After click the SET button type “save/test.txt, 10, result.txt”



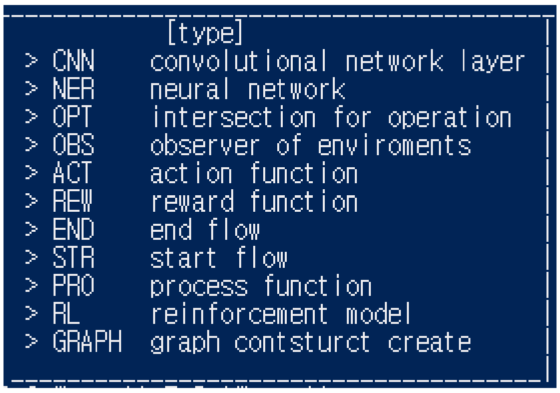
```
STATE : WAITING
Setting [tree file Path, max iteration, save file P
(comma sePerate) :save/test.txt,10,result.txt
```

10. Now click the RUN button
11. If the center double square is moving, it means that the programe is running.
12. If the moving is stopped, you can exit the program

Console Interface User Guide

Although the console interface is provided, you can write down by other third party text edit program. In this case, please reference the format that already in the save file.

--help : show the basic options
--license : show license
--create “type”: writing the model using console interface



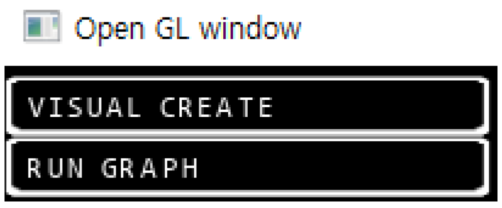
```
 ----------------------------------------------------
               [type]                               |
 > CNN    convolutional network layer               |
 > NER    neural network                            |
 > OPT    intersection for operation                |
 > OBS    observer of enviroments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

CNN : Convolutional Neural Network block
NER : Deep Neural Network block
OPT : Operation block (version 5 provides only summation and multiplication)
OBS : The observation block (reading the state function inshared library)
ACT : The activation block (reading the action function in shared library)
REW: The reward block (reading the reward function in shared library)
(ADVANCE : if you want to make flexible function, please use shared library as function)
END: The end of one cycle of flow graph block
STR: The start of one cycle of flow graph block
PRO: The control the cycle(process) of the flow graph block
RL: The reinforcement model block
GRAPH: The graph flow chart file

--setting : Changing the setting options
verboseon(off) : Show the detail of the program runtime message
gpgpuon(off) : Using OpenCL GPGPU

--run "graph file path" "max iteration" "save file name" : running the program

GUI User Guide



VISUAL CREATE : Creating the graphical flow chart
RUN GRAPH : Running the graphical flow chart

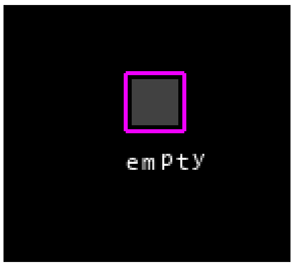
VISUAL CREATE

First of all, the block is the unit for saving and running each algorithm
This program's one of main point is that using the connections of blocks for easy to create the machine learning algorithm.

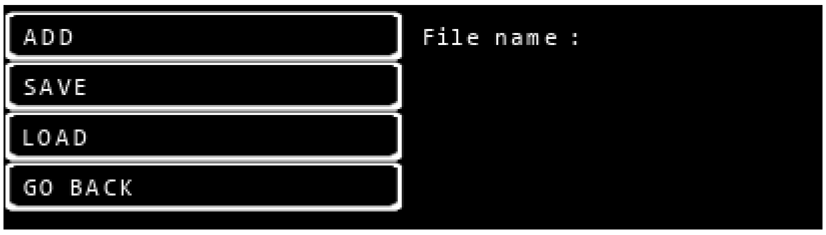
ADD : Adding a new block
SAVE : Save current selected block and its child blocks
LOAD : Loading the graph flow chart

Guide for using graphical interface

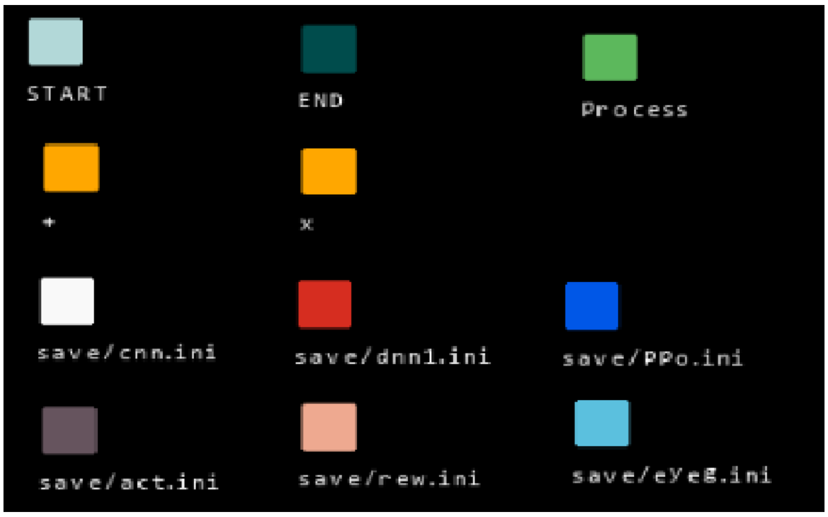
1. When click the ADD button, empty block is created.
   The total node is limited in 2^32-1
2. When click the block, the selected block is highlighted.



3. When right click the block, you can link the block with file by file name

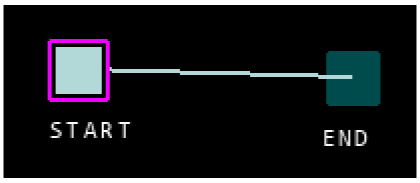


4. For each file is colored by its type(model)

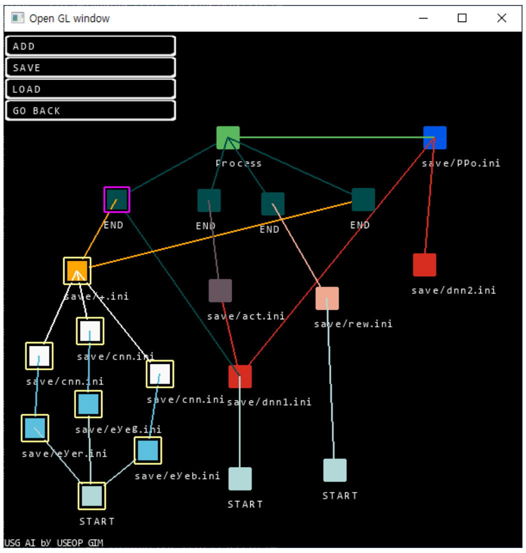


Especially, file name as “+”,”x”, “END”,” START” are automatically created even there are no such file.

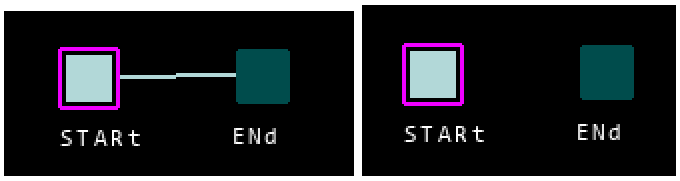
5. From selected block to another, the first block be child and the second be parent for the selected block



6. When click the connected block, it also highlighted the selected blocks children.

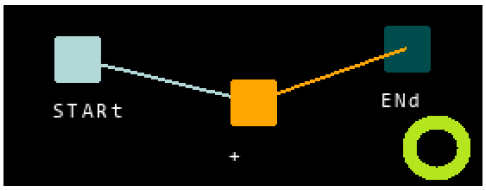


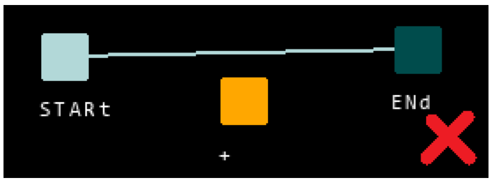
7. When double click the selected block, its connections are removed.



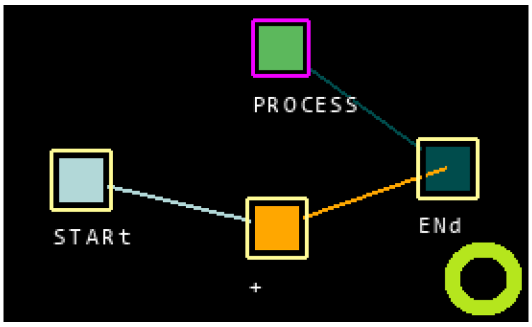
User Guide for connection

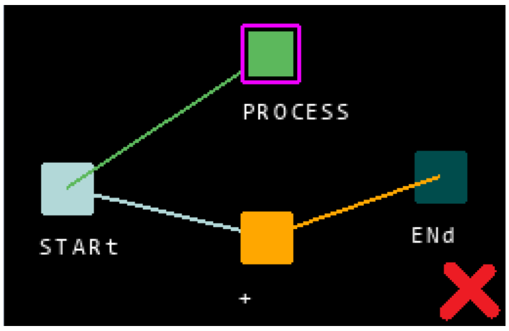
1. The START and END block must be exist at the end of both side in the block that want to be run.



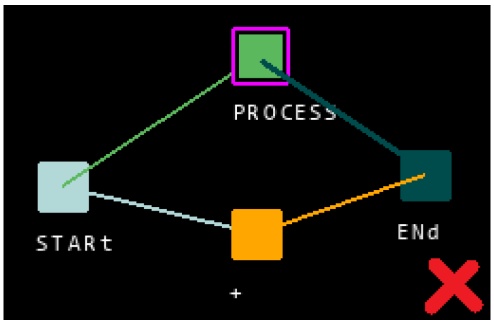


2. The PROCESS block must have the END block as child.

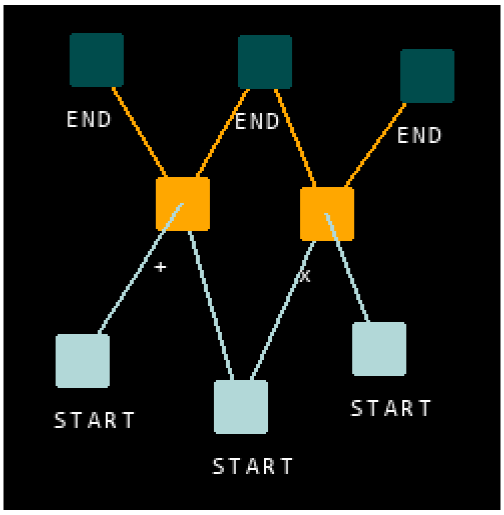




If the connection be recursive, the program is forced to be exited!

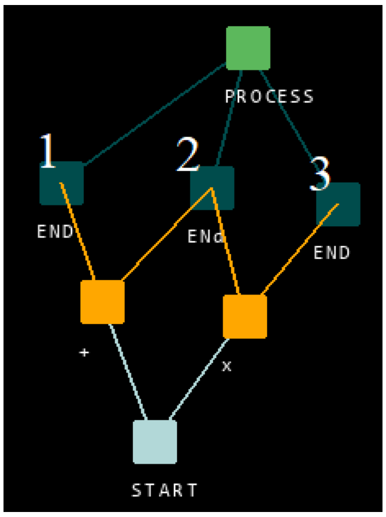


3. There are unlimited for the number of child and parents

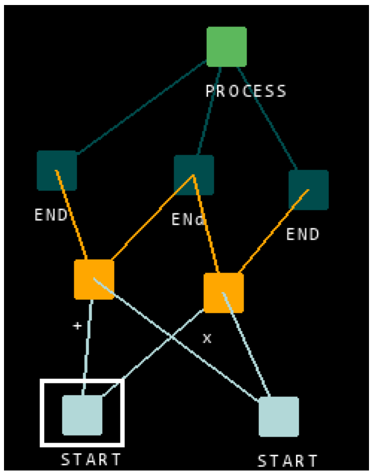


However, when save the block, only the selected block and its children are saved.

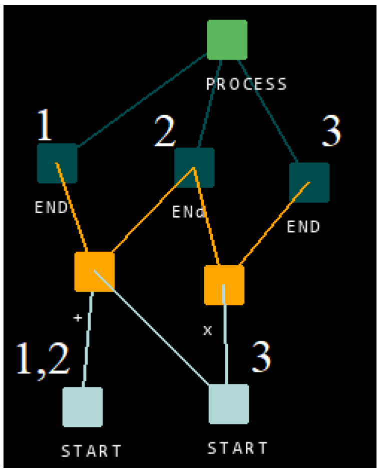
4. The order of running the program is base on the position of end block connecting PROCESS block. The most left upside block be front.



5. As same as above, the order of the START block is also the most left upside block be first.



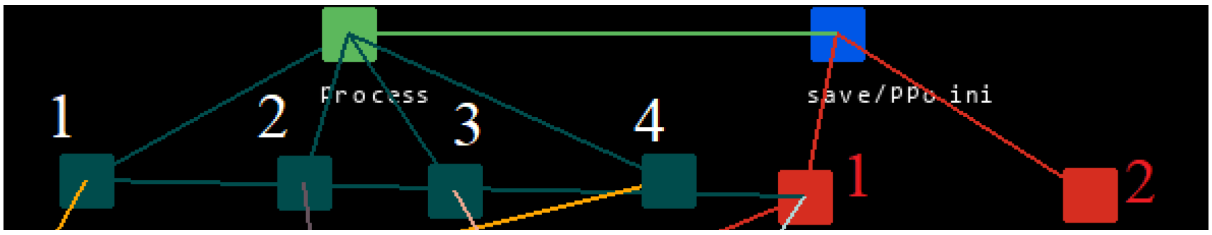
For example, above connection shows only the left START block be the start line for all END blocks.



Beside, above connection shows the 1,2 END block is start from the left START block and 3 END block is start from the right START block.

Reinforcement block

The reinforcement block must have a single PROCESS block with four END block and multiple neural network block (the number of neural network block depeneds on the reinforcement learning model)



- END [1] : the first end block recieve the enviroment state.
- END [2] : the second end block recieve the action state.
- END [3] : the third end block recieve the reward state.
- END [4] : the last end block recieve the enviroment state after the action.
- The most left side neural network block be the main agent block for the learning, action(or policy) in model. In other words, it is the main neural network decides the action.

Shared library shape

```
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);
```

Please see the src/extrafn.cpp file.