

Nebula – Persistence Documentation

Michał Biszczanik, Miłosz Chojecki

Introduction

Storage of created designs is essential, as recreating them from scratch each time would be deterring to say the least. Persistence is capable of serialising the model to store it externally and reconstructing the model from serialised data.

How it works

User specifies input and output files as command-line arguments:

```
./nebula [-l in_file.neb] [-s out_file.neb]
```

If no input file is specified, Nebula loads a sample project consisting of two logic gates.

If no output file is specified, the project is discarded when exiting Nebula.

Structure of a .neb file

An example .neb file looks like this:

```
gate 0.6 0.5 -2.76016 2.21604 1
gate 0.6 0.5 -0.978603 1.93038 0
endgates
port -2.76016 2.34104 0
port -2.76016 2.59104 0
port -2.16016 2.46604 1
port -0.978603 2.05538 0
port -0.978603 2.30538 0
port -0.3786 2.18038 1
endports
connection -2.16016 2.46604 -0.978603 2.05538
connection -2.16016 2.46604 -0.978603 2.30538
connection -0.978603 2.05538 -2.16016 2.46604
connection -0.978603 2.30538 -2.16016 2.46604
endconnections
```

The file consists of three sections: gates, ports and connections.

Each gate is stored as follows:

```
gate <x_size> <y_size> <x_coord> <y_coord> <gate_type>
```

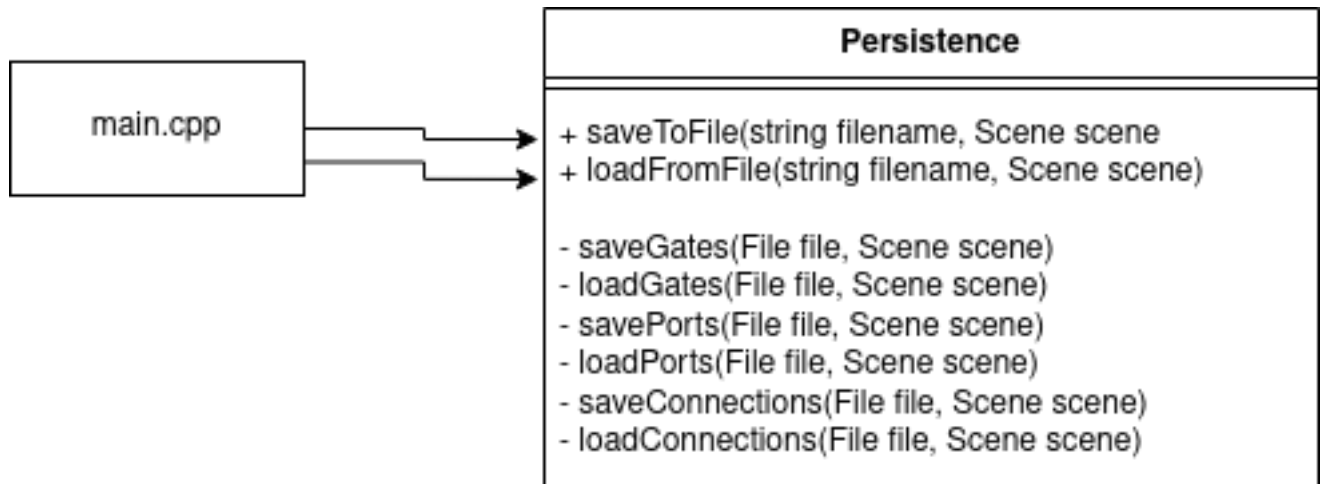
Each port is stored as follows:

```
port <x_coord> <y_coord> <type>
```

Each connection is stored as follows:

```
connection <first_x_coord> <first_y_coord> <second_x_coord> <second_y_coord>.
```

Class diagram of persistence module



Swimlane diagram of persistence module

