Optimizing Social Distance Keeping in Indoor Environments via a Public Display Navigation Support System - Investigations with R

Christian Terbeck (christian.terbeck@uni-muenster.de)

January 27, 2022

Preparation

Use NLRX library.

```
library(nlrx)
```

Set NetLogo settings according to NLRX documentation

```
netlogopath <- file.path("C:/Program Files/NetLogo 6.1.0")
modelpath <- file.path("C:/Users/chris/github/master/model6-1.nlogo")
outpath <- file.path("C:/Users/chris/github/master/output/r")
```

Set nl object (NetLogo 6.1 is currently the highest supported version).

```
nl <- nl(nlversion = "6.1.0",
         nlpath = netlogopath,
         modelpath = modelpath,
         jvmmem = 1024)
```

Check if nl object can initialize without errors.

```
print(nl)
```

Set up the experiment based on the NetLogo model.

```
nl@experiment <- experiment(expname="hospital",
                            outpath=outpath,
                            repetition=1,
                            tickmetrics="false",
                            idsetup="setup",
                            idgo="simulate",
                            runtime=30000,
                            evalticks=seq(40,50),
                            metrics=c("time", "overall-contacts / 2"),
                            variables = list('familiarity-rate' = list(min=0, max=1, qfun="qunif")),
                            constants = list("scenario" = "\"hospital\"",
                                             "dt" = 0.5,
                                             "initial-number-of-visitors" = 0,
                                             "staff-members-per-level" = 4,
                                             "spawn-rate" = 120,
                                             "mean-visiting-time" = 30,
                                             "max-visiting-time" = 60,
                                             "max-capacity" = 120,
                                             "area-of-awareness" = 10,
                                             "angle-of-awareness" = 15,
                                             "show-areas-of-awareness?" = "false"))
```

Check if all variables and constants are assigned correctly:

```
eval_variables_constants(nl)
```

Set up simulation design:

```
nl@simdesign <- simdesign_lhs(nl=nl,
                              samples=1,
                              nseeds=3,
                              precision=3)
```

Check for errors again:

```
print(nl)
```

Running Simulation(s)

Compute results:

```
results <- run_nl_all(nl = nl)
```

Evaluation

Attach simulation, write output and analyze data:

```
setsim(nl, "simoutput") <- results
write_simoutput(nl)
analyze_nl(nl)
```