Stride Project report

Andrei Bondarenko, Mathias Ooms, Stan Schepers, Laurens Van Damme

May 2019

1 Daycare & PreSchool

Implemented:

- Generator for Daycare's
  Can be found on Daycare&PreSchool branch:
  - main/cpp/geopop/generators/DaycareGenerator.cpp
- Generator for PreSchools
  Can be found on Daycare&PreSchool branch:
  - main/cpp/geopop/generators/PreSchoolGenerator.cpp
- Populator for Daycare's
  Can be found on Daycare&PreSchool branch:
  - main/cpp/geopop/populators/DaycarePopulator.cpp
- Populator for PreSchools
  Can be found on Daycare&PreSchool branch:
  - main/cpp/geopop/populators/PreSchoolPopulator.cpp
- Extended GeoGridConfig
  Can be found on Daycare&PreSchool branch:
  - main/cpp/geopop/GeoGridConfig.h + .cpp

Use:
The use of Daycare's and PreSchools is part of the simulation itself and happens automatically (therefore there's no user manual created for this feature). Depending on the given population, the Daycare's and PreSchools are generated and populated. This is displayed in the info-output.

Tests:
The description of the tests can be found in our test plan in section: "Testing Plan - Daycare & PreSchool". Tests can be found on the "Daycare&PreSchool" branch in following files:

- test/cpp/gtester/geopop/generators/DaycareGeneratorTest.cpp
- test/cpp/gtester/geopop/generators/PreSchoolGeneratorTest.cpp
- test/cpp/gtester/geopop/populators/DaycarePopulatorTest.cpp
- test/cpp/gtester/geopop/populators/PreSchoolPopulatorTest.cpp
- main/cpp/geopop/io/GeoGridJSONReaderTest.cpp (extended)
- main/cpp/geopop/io/GeoGridJSONWriterTest.cpp (extended)

2 Data formats

Implemented:

- JSON reader for household files
  Can be found on DataFormatsJSON branch:
  - main/cpp/geopop/io/HouseholdJSONReader.h + .cpp
- JSON reader and writer for GeoGrid files
  Can be found on DataFormatsJSON branch:
  - main/cpp/geopop/io/GeoGridJSONReader.h + .cpp
  - main/cpp/geopop/io/GeoGridJSONWriter.h + .cpp
- HDF5 reader and writer for GeoGrid files
  Can be found on DataFormatsHDF5 branch:
  - main/cpp/geopop/io/GeoGridHDF5Reader.h + .cpp
  - main/cpp/geopop/io/GeoGridHDF5Writer.h + .cpp
  - main/cpp/geopop/io/GeoGridHDF5Utils.h + .cpp

  Following files were also introduced/altered:
  - main/cpp/geopop/io/GeoGridWriter.h + .cpp
  - main/cpp/geopop/io/GeoGridStreamWriter.h
  - main/cpp/geopop/io/GeoGridFileWriter.h
  - main/cpp/geopop/io/GeoGridReader.h + .cpp
  - main/cpp/geopop/io/GeoGridStreamReader.h + .cpp
  - main/cpp/geopop/io/GeoGridFileReader.h + .cpp

Use:
In the configuration XML set population_file parameter to file name with desired extension (.json or .h5). On respective branches formats were also added to file formats and results sections in user manual.
Tests:
The description of the tests can be found in our test plan in section: "Testing Plan - Data Formats". Tests can be found on respective branches for each format in following files:

- test/cpp/gtester/geopop/io/GeoGridXXXXWriterTest.cpp
- test/cpp/gtester/geopop/io/GeoGridXXXXReaderTest.cpp
- test/cpp/gtester/geopop/io/HouseholdJSONReaderTest.cpp

3 Data visualization

Implemented:

- Location as template library with get<0/1> function to get latitude/longitude
- JSON/hdf5/protobuf reader/writer for epi-output
- Viewer for epi-output integrated in sim controller
- GUI
  - Location circles with center point
  - HSV colors with selection of health category
  - Selection of Circle and Rectangle
  - Data bar showing epi-output of location/selection
  - Separate executable
  - Automatic simulation

Files:
Epi-output writer can be found in the geopop/io folder. The reader is part of the visualization so it can be found together with the GUI files in the visualization folder. The epi-output viewer is together with all other viewers in the folder viewers.

Use:
How to use the data visualization is explained in the user manual of the data visualization branch. The section can be found between the "Running the simulator" and "Python wrapper" sections and is called: "Running the visualization".

Tests:

The description of the tests can be found in our test plan in section: "Testing Plan - Data visualization". The test files for epi-output generation and writers can be found in the gtester/geopop folder in generators folder and io folder. The epi-output reader test can be found in the gtester/visualization folder.

4 Demographic profile

Implemented:

- Reading in multiple household files in $main/cpp/pop/GeoPopBuilder.cpp$
- Added Young and Old age brackets in $main/cpp/contact/AgeBrackets.h$
- Storing the reference household per household type and a average reference household in $main/cpp/geopop/GeoGridConfig.cpp$.
- Assigning a household type to cities in $main/cpp/geopop/io/LocationsCSVReader.cpp$.
- Assigning and calculating the young/old fraction for a Location in $main/cpp/geopop/Location.h$ and $main/cpp/geopop/io/LocationsCSVReader.cpp$.
- Use the reference household according its household type for a populating households in $main/cpp/geopop/populators/HouseholdPopulator.cpp$
- Generating more daycares, preschools and K12 schools according to the city's young old fraction in $main/cpp/geopop/generators/\{Daycare|PreSchool|k12School\}Generator.cpp$

Use:
The use of Demographic Profile can be obtained by removing the household_file in the XML and assigning different household types to household files as shown in $main/resources/config/run\_generate\_default_provinces\_flanders.xml$. In $main/resources/data/flanders_cities.csv$ we assign every location its respective household type which household file it should use.
Tests:
The description of the tests and the location in the project can be found in our test plan section "Test Plan - Demographic Profile".

5 Big class personnel of companies

Implemented:

- Generator for Workplaces (altered)
  Can be found on WorkplaceSizeDistribution branch:
  - main/cpp/geopop/generators/WorkplaceGenerator.cpp

- Populator for Workplaces (altered)
  Can be found on WorkplaceSizeDistribution branch:
  - main/cpp/geopop/populators/WorkplacePopulator.cpp
- Soft limit for Contactpool (altered)
  Can be found on WorkplaceSizeDistribution branch:
  - main/cpp/contact/Contactpool.h + .cpp

Use:
The use of an workplace-size distribution is determined by the XML configuration file. The parameter "distribution_file" must be set to a valid .csv-file, the trace-output will give extra information of the use of this file. The user-manual contains more details.

Tests:
The description of the tests can be found in our test plan in section: "Testing Plan - Big class personnel of companies". Tests can be found on the "WorkplaceSizeDistribution" branch in following files:

- test/cpp/gtester/geopop/generators/WorkplaceGeneratorTest.cpp
- test/cpp/gtester/geopop/populators/WorkplacePopulatorTest.cpp
- test/cpp/gtester/geopop/io/WorkplaceFormatReaderTest.cpp
- test/cpp/gtester/ScenarioRuns.cpp
- test/cpp/gtester/ScenarioData.cpp
- main/cpp/util/RunConfigManager.h + .cpp