1 Project description

You will create an multithread application to simulate life of autonomous worms. All the worms will be placed within the same environment and will interact with each other. The goal is to simulate an ecosystem with diverse types of individuals living inside.

TODO: picture of worms / screenshot of app?

All worms will live on a ractangular ground, where hitting an edge results in appearing on the other side. Similar to the *Snake* game available on old Nokia phones. In other words, the environment is a torus projection on a 2D screen.

TODO: picture of scrossing the edge / series of pictures showing edge crossing.

The design of the core functionality with 2 types of worms is been prepared, together with an implementation with a few missing functionalities. You will need to implement them to make the application run and to satisfy predefined test cases and later you will prepare your own design for the extensions of the application.

1.1 Worms movements

Board is a square grid where all the worms are placed. Each worm has its head in one of the fields (i.e. $(1, 1)$) and possibly a tail which is along the recently visited fields. This is a natural behaviour of a snake, where head moves into a new position and the tail follows that movements. See the example where a worm started at position $(3, 0)$ then went through fields $(2, 0)$, $(2, 1)$ and finished at field $(1, 1)$.

TODO: Drawing of a worm and its tail

1.2 Interactions between worms

There will be two types of worms (you will make additional types in *extended functionality*):

- *Lazy Worm*
- *Hunter Worm*

1.2.1 *Lazy Worm*

This worm randomly travels around until other worm or a bonus is close enough, then it goes directly to the desired goal. At each field it will randomly choose one of available fields (by default there will be three fields available – no steps back).

TODO: no going back picture

1.2.2 *Hunter Worm*

TODO: description

2 Technical specification

Project package contains a directory with source codes, makefile and tests. It should follow best practices of a multithread application and a style guidelines.

Source files with main logic are available in `sources/`, all tests are within `tests/`. Your first modifications will need to be made in `sources/TODO_FILENAME`.

2.1 Basic functionality

List of files that need to be modified for the whole application to run:

- TODO...

2.2 Extended functionality

When finished with basic functionality you will need to extend your application to become interesting to watch. It will engage you to shape your code in a way that fits the core design, but extension's design will be entirely up to you.

Extensions (detailed explanation in a separate section of this document):

- Implement bonuses (small items like cherries) that appear randomly on the board,
- Save function that allows to dump current state into a file (and load it during the next run),
- (big) Implement behaviour schema for an worm, basic artificial intelligence,