**TAU-Robot Exercise 1 - Design Document**

**Yaron Libman and Ilya Aizin**

Last update: March 28th 2016

Scope of this document: Explanation of components used in implementation and design rational.

* For further explanation see class and sequence diagrams.

**Introduction**

*we decided to dived the project to few logic class, each class is responsible for a defined part of the simulator work.*

***House***

*Uses: Direction*

* Represent a single House, contain the house information and map
* Responsible for creating the house object, has simple constructor that creating the house by info and the map by a hard coded house (for exercise 1)
* Implement a copy constructor
* Responsible for moving the vacuum inside the house by direction
* Responsible to update the house dirt level in the vacuum location by request
* Checks validity of the house

**OneSimulation**

*Inner class of Simulator*

*Used by:* Simulator

*Uses:* House, Abstract Algorithm, Sensor

* Represents a single simulation of a single abstract algorithm running on a single house.
* Responsible for updating house state when abstract algorithm makes a move.
* Checks validity of Abstract algorithms moves and updates simulation status accordingly (i.e. Abstract algorithms made an illegal move, out of battery etc.)
* Calculates single simulation score using winner information provided by Simulator

**Simulator**

*Uses:* House, OneSimulation, Naïve Algorithm, Abstract Algorithm, Sensor, Abstract Sensor

* Loads Abstract Algorithm implementations, houses and configuration information.
* For each house, runs all simulation in parallel.
* Coordinates multiple simulations by executing them step by step in round robin fashion.
* Keeps track of the winner and simulation finishing conditions (Max\_Steps).
* When simulation finishes initiates score calculation by OneSimulation’s by passing winner and global simulation information.
* Prints calculated scores.

**Naïve Algorithm**

*Simple implementation of Abstract Algorithm*

*Used by:* Simulator, OneSimulation

*Uses*: Sensor only

* Chooses next step randomly between staying or moving to any non-wall cell.

**Sensor**

*Simple implementation of Abstract Sensor*

Used by: OneSimulation

Uses: SensorInformation, House, Direction

* Represents a single Robot Sensor.
* Responsible to tell the robot environment, return a Sensor Information object .

**Error Handling**

In exercise 1 we handle

* the program arguments.
* Config file open an exits.

Each error is printed to the user, and the program will exit after the message print