
Orodael Turrim Documentation

Jan Horacek

Nov 10, 2019

Contents:

1	Installation guide	3
1.1	Requirements	3
1.2	Framework installation	3
1.3	Documentation build	7
2	GUI application	9
2.1	Map widget	10
2.2	Map info tab	10
2.3	Round control tab	11
2.4	Game control tab	12
2.5	Spawn info tab	12
2.6	Action log tab	13
3	Console interface	15
4	Game mechanisms	17
4.1	Positions types	17
4.2	Unit actions	19
4.3	Sight	19
4.4	Attack range	20
4.5	Unit placing	20
4.6	Filters	20
4.7	Attacker spawn information	21
5	Game map	23
5.1	Map terrains	23
5.2	Map movement	25
6	Units	27
6.1	Defenders	27
6.2	Attackers	29
6.3	Effects	33
7	Expert system	35
7.1	Knowledge base	35
7.2	Inference	37
7.3	Action base	39

7.4	Rules file	39
7.5	Custom filters	41
8	Rules definition language	43
8.1	Grammar	43
8.2	Examples	45
9	Proxy definition	47
9.1	Map proxy	47
9.2	Game Object Proxy	48
9.3	Game Control Proxy	50
9.4	Game Uncertainty Proxy	50
10	API documentation	51
10.1	OrodaelTurrim package	51
10.2	ExpertSystem package	64
10.3	ArtificialIntelligence package	65
10.4	User package	65
11	Release Notes	67
	Python Module Index	69
	Index	71

Orodael Turrim is python based framework game for university subject Knowledge-based systems (BI-ZNS) on FTI CTU Prague.

The purpose of this project is to teach students to create their own planning expert system that can play tower defence. The expert system must prepare the defense based on information about the attackers and endure the onslaught of enemies as long as possible. The expert system is fully automatic, without external user intervention. All known information are provided by the framework itself.

Are you ready to resist the hordes of Rigor Mortis servants?

1.1 Requirements

Python

Python 3.7.* is required because of some features and better performance

External packages requirement

- PyQt5 - Graphical interface
- click - Console interface
- antlr4-python3-runtime - Knowledge base language parser
- multi_key_dict - Package for multi key dictionary support

1.2 Framework installation

1.2.1 School computers

On the school computers you should already have Python 3.7 installed. Use system Python 3.7 with virtual environment.

1. Clone framework to you directory

```
cd <your_directory>
# From GitLab for FIT CTU Prague
git clone git@gitlab.fit.cvut.cz:hajkokla/orodaelturrim.git
# or from GitHub for others
git clone git@github.com:Wilson194/OrodaelTurrim.git
```

2. Create virtual environment for framework and install all dependencies

```
cd zna_framework_python          # Change directory to framework
python3.7 -m venv __venv__         # Create virtual environment
. __venv__/bin/activate           # Activate virtual environment
pip install -r requirements.txt    # Install all requirements
```

3. Run framework (you need to run framework from console where you activate virtual environment)

```
python OrodaelTurrim
```

1.2.2 Windows

For the windows user I recommend to use [Anaconda](#) distribution. You will get whole Python installation with integration to the system and also virtual environments support with few steps in GUI installation. Also if you are using PyCharm, in the new version (2019) PyCharm support Anaconda distribution, so some features are implemented directly to IDE.

Warning: If you have some older Anaconda installation on your system, it is recommended to uninstall whole distribution and install new one with Python 3.7. If you only update the distribution, there could be some problems with PyQt dependencies.

1.2.3 Ubuntu 18

On Ubuntu you have 2 possibilities to run Python 3.7. You can use system Python interpreter or Anaconda.

System Python

1. Install Python 3.7

```
sudo apt update                # Update apt repositories
sudo apt install software-properties-common # Install program for apt adding
sudo add-apt-repository ppa:deadsnakes/ppa  # Add Python apt repository
sudo apt install python3.7              # Install Python 3.7
sudo apt install python-virtualenv      # Install virtual environments
↪support
sudo apt install python3.7-venv        # Install require packages for
↪Python 3.7
```

2. At this point, Python 3.7 is installed on your Ubuntu system and ready to be used. You can verify it by typing

```
python3.7 --version
```

3. Clone framework from the GitLab or GitHub

```
cd <your_directory>
# From GitLab
git clone https://gitlab.fit.cvut.cz/bi-zns_pracovni/zna_framework_python
# or from GitHub
git clone git@github.com:Wilson194/OrodaelTurrim.git
```

4. Create virtual environment for framework and activate


```
cd zna_framework_python          # Change directory to framework
python3.7 -m venv __venv__         # Create virtual environment
. __venv__/bin/activate            # Activate virtual environment
pip install -r requirements.txt    # Install all requirements
```

5. Run framework

```
python OrodaelTurrim              # Run Framework
```

Anaconda

1. Download Anaconda from the source page <https://www.anaconda.com/distribution/>
2. Add executable permissions and run installer from you console. You can left all options default, but it's better to disable auto activate conda. It is better to add conda bin folder to PATH.

```
cd <Downloaded_directory>
chmod +x <Downloaded_file>
./<Downloaded_file>
```

3. Edit .bashrc file

```
export PATH="/<path_to_installation>/bin:$PATH"
```

4. Now you have conda bin folder in path. You should have Python 3.7. You can verify that with

```
python --version
```

5. Clone framework from the GitLab or GitHub

```
cd <your_directory>
# From GitLab
git clone https://gitlab.fit.cvut.cz/bi-zns_pracovni/zna_framework_python
# or from GitHub
git clone git@github.com:Wilson194/OrodaelTurrim.git
```

6. Install dependencies

```
cd <cloned_repository>
pip install -r requirements.txt
```

7. Run framework

```
python OrodaelTurrim
```

1.2.4 Linux Mint

Python 3.7 is not added to apt yet. You need to install Python 3.7 from other original source. Don't worry, it is so hard.

1. Install Python 3.7

```
sudo apt install build-essential checkinstall
sudo apt install libreadline-gplv2-dev libncursesw5-dev libssl-dev libffi-dev
sudo apt install libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev

cd /usr/src
```

(continues on next page)

(continued from previous page)

```
sudo wget https://www.python.org/ftp/python/3.7.3/Python-3.7.3.tgz # Download
↳ Python

sudo tar xzf Python-3.7.3.tgz # Extract
↳ python source

cd Python-3.7.3
sudo ./configure --enable-optimizations
sudo make altinstall # Install python under python3.7 (don't replace old python
↳ version)

sudo apt install python-virtualenv # Install
↳ virtual environment support
```

2. Clone framework from the GitLab or GitHub

```
cd <your_directory>
# From GitLab
git clone https://gitlab.fit.cvut.cz/bi-zns_pracovni/zna_framework_python
# or from GitHub
git clone git@github.com:Wilson194/OrodaelTurrim.git
```

3. Create virtual environment for framework and activate

```
cd zna_framework_python # Change directory to framework
python3.7 -m venv __venv__ # Create virtual environment
. __venv__/bin/activate # Activate virtual environment
pip install -r requirements.txt # Install all requirements
```

4. Run framework

```
python OrodaelTurrim # Run Framework
```

1.2.5 Fedora

1. Install Python 3.7

```
sudo dnf install python37
```

1. Clone framework from the GitLab or GitHub

```
cd <your_directory>
# From GitLab
git clone https://gitlab.fit.cvut.cz/bi-zns_pracovni/zna_
↳ framework_python
# or from GitHub
git clone git@github.com:Wilson194/OrodaelTurrim.git
```

1. Create virtual environment for framework and activate

```
cd zna_framework_python # Change directory to framework
python3.7 -m venv __venv__ # Create virtual environment
. __venv__/bin/activate # Activate virtual environment
pip install -r requirements.txt # Install all requirements
```

2. Run framework

```
python OrodaelTurrim          # Run Framework
```

1.3 Documentation build

You can build local documentation from source files.

```
cd docs
python3.7 -m pip install -r requirements.txt
make html    # For windows make.bat html
```

Those commands will create `Index.html` file in `docs/_build` folder. This file is index page of the documentation.

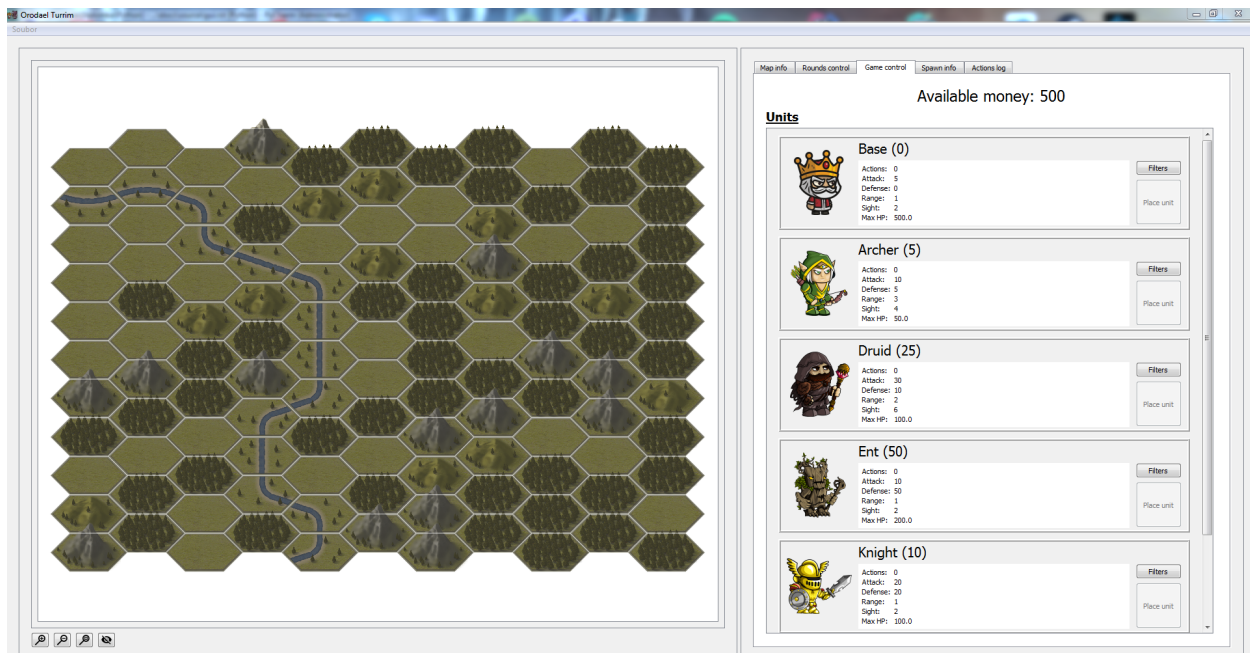
CHAPTER 2

GUI application

For the development purpose game is fully controllable from GUI application. You can start the application from the root directory

```
python OrodaelTurrim
```

This command should open game main window and the game should be initialized.







Main windows is divided to two parts, *Map widget* with the game map and *Tab widget* with all control and information panels. You can change the sizes of each parts with central slider. Or you can also hide one of the widgets by slide the widget to one side.

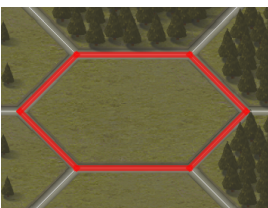


2.1 Map widget

Map widget displaying current state of the game. You can select each tile on the map and on the `Map info` tab you can get information about that tile.

On the bottom of the map, there are 4 map control buttons

-  - Zoom in the map
-  - Zoom out the map
-  - Fit the map to the map widget size
-  - Clear all borders and selections on the map

On the map there are three types of the borders

-  - Currently selected position
-  - Attack, sight or move attack of selected unit
-  - Possible tiles of enemy spawn (always on the edge)



Note: After spawn first unit you can also see that some tiles are grey and some not. Grey tiles represent positions that none of your units can see.


2.2 Map info tab

On the *Map info widget* you can get information about selected tile and unit on that tile. This tab is empty until some position is selected on the map.

Position info

First part showing position in three types of representation. You can read about positions types at [Positions types](#)

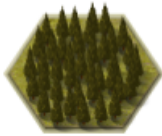
-  - Offset position [q r]
-  - Cubic position [x y z]

-  - Axial position [q r]

Terrain info

Next part showing information about terrain on the selected position. You can see there some basic attributes of the tile. You can read more about terrain types at [Map terrains](#)

Map tile: Forest






Sight cost: 3
Attack bonus: 0.2
Defence bonus: 0.1

Game object info

Last part of the widget displaying information about unit. This part is visible only if some game object is on currently selected position. You can see there information about unit attributes. If you select your own unit you can see also attack filters of this unit. You can read about attack filters at [Filters](#)

Game object info also have 3 buttons

-  - show visibility of the unit on the map with blue border
-  - show attack range of the unit on the map with blue border
-  - show move range of the unit on the map with blue border

Character: Base



Attributes

Actions: 0
Attack: 5
Defense: 0
Range: 1
Sight: 2
Max HP: 500.0
Current HP: 500.0

Effects

Filters

Random

2.3 Round control tab

On the Rounds control tab you can control game rounds and history browsing. Tab is divided to 4 parts

Game info First part displaying information about current game round (starting from 0) and current player.

Player control On the second parts there are two control buttons

- *Run inference* - Execute one round of user inference (Create knowledge base, parse rules and run inference)
- *End of round* - End your turn and execute attackers AI. After finished you will be on the start of the next round.

Auto combat The next section is for simulating game rounds. You can specify simulation options

- *Number of rounds* - How many rounds you want to simulate (you can't stop simulation until finished)
- *Display process* - If you check this option, result of each round will be displayed on the map (little bit slower)
- *Delay* - Here you can specify time delay between rounds. This is useful when you are displaying process on the map.

Browsing history Last section is for browsing game history. With buttons *Previous turn* and *Next turn* you can browse game history. Current round will be displayed in the first part. When you are in the browsing mode, you cannot do any actions.

Buttons always move to the start of the round of each player. So you can see what user can do in that round. In the log you can read what that user done. In browsing mode you can also see enemy point of view. This is allowed because of debugging. This game is not designed for competitive playing with UI, so find every information you want.

2.4 Game control tab





With the *Game control tab* you can place units manually. So you can play the game with your mouse, but it is really slow and noob method. So you should use *Game control tab* only for testing and debug purpose.

On the top you can see amount of resources you have. Under it there are your unit cards.

Unit card On the unit card you can see name of the unit with price in the brackets. Also there are basic unit attributes. If you want to know more about unit, you must look to the documentation.


With the *Filters* button you can setup attack filters for the unit. *Place unit* button is disabled until you have selected available position on the map.

Filter window When you click on the *Filters* button, new window will open. Here you can specify which filters you want to use and the order of the filters. You can read about filter system here [Filters](#). There are 4 control buttons for filters

-  - Add filter to active filters
-  - Remove filter from active filters
-  - Move current selected filter up (or you can use drag & drop)
-  - Move current selected filter down (or you can use drag & drop)

Settings of active attack filters are persistent in one game session. So when you set up filters for one unit, this unit will always born with those active filters until you change configuration. Filters are unique for each unit.

2.5 Spawn info tab

In the *Spawn info tab* you can obtain information about enemy planning spawn. You can read about your scouting at [Attacker spawn information](#). For each known round there is part with list of unit and they anticipated positions. Your scouts are not so good so there is information about position range and probability of positions. With the  you can display positions on the map (orange borders).

2.6 Action log tab

TODO

Console interface

The framework have also console interface. You can use console to test you Expert system for whole simulation. Simulate the game without GUI is faster and you don't need to click on anything. Also you can export history log from the simulation, so find out why your king die so fast.

Options

- *-gui / -nogui* - Disable or enable GUI
- *-r / -rounds* - Set maximum of rounds for simulation
- *-l / -log-output* - Save log from the game in txt format
- *-help* - Display help message

Simulation without GUI is faster and also you can automatize testing.

Game mechanisms

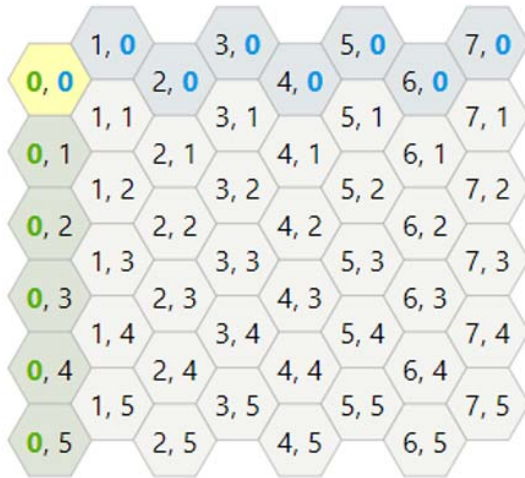
In the Orodael Turrim there are few mechanisms that are not standard for the games. Many of them are designed because of expert system control and some of them are designed for this game. Also game using hex tile, where tile indexing is not clear in the first look. So there are description of main game mechanisms.

4.1 Positions types

Hex shape maps are commonly use in many games, but usually user don't need to index positions. In Orodael Turrim you need to place units from the code so you need to understand position system. For indexing positions in hex maps there are many systems, which some of them are faster for computing, some of them are more user readable. As a compromise, I chose three types of positions that is supported by framework, Cubic position, Offset position and Axial position. You can chose any of them, which one is most understandable to you. In the code there are no different between them.

4.1.1 Offset coordinates

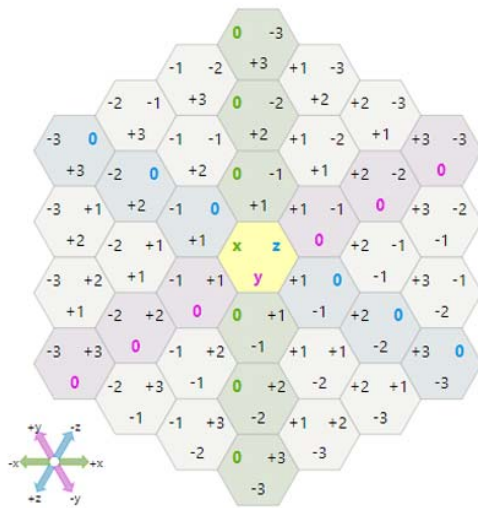
The most common approach is to offset every other column or row. Columns are named col (q). Rows are named row (r). Position [0 0] is in the center of the map, so positions on the left from the center have col (q) negative and positions above the center have negative row (r).



This is I think that offset is most understandable position system, but it is terrible for all computations.

4.1.2 Cubic coordinates

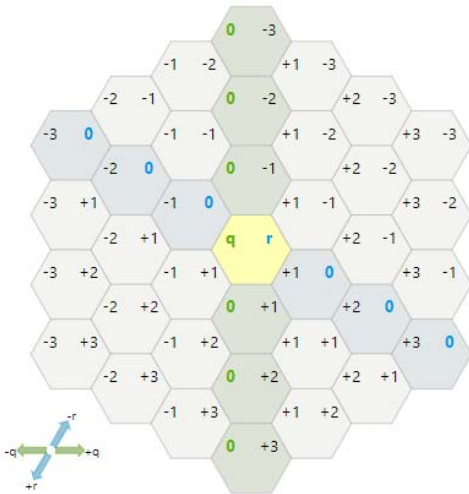
Another way to look at hexagonal grids is to see that there are three primary axes, unlike the two we have for square grids. There's an elegant symmetry with these. The cube coordinates are a reasonable choice for a hex grid coordinate system. The constraint is that $x + y + z = 0$ so the algorithms must preserve that. The constraint also ensures that there's a canonical coordinate for each hex.



This system is perfect for fast and simple operations, but could be hard to orient on the map for the user. Also you need to save 3 numbers for each position.

4.1.3 Axial coordinates

The axial coordinate system, sometimes called “trapezoidal” or “oblique” or “skewed”, is built by taking two of the three coordinates from a cube coordinate system. Since we have a constraint $x + y + z = 0$, there's some redundancy, and we don't need to store all three coordinates. This diagram is the same as the previous one, but without y axis



This system fix problem with 3 numbers in cubic coordinates system but it is more unreadable than cubic, so for user no big deal.

Note: All the information about position system is obtained from <https://www.redblobgames.com/grids/hexagons/>, so if you want to know more about hexagons systems you can read that page.

4.2 Unit actions

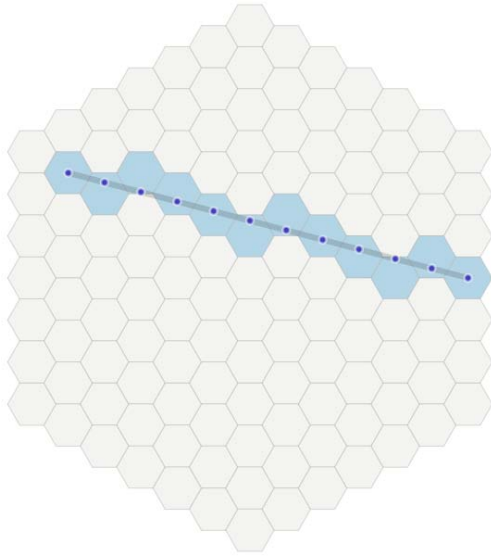
Movement over the map depends on the source terrain type, target terrain type and number of actions of the unit. Some terrains are harder to cross then others. You can find table describing movements at [Map movement](#). Actions of the units describing only how far unit can go, each unit could attack only once per round. Unit don't need to spend all the actions.

Note that defender units have 0 actions, so they can't move.

4.3 Sight

Each unit have sigh, that describe how far it can see. Sight depend on the terrain that you standing on (for example mountain gives you sight bonus) and on the tiles that you need to overlook. First constrain is that you always see all your neighbour positions.

Computing for next positions is simple, create line from the center of starting position to the center of target position. All tiles that line cross are added to sight computation. Each terrain type have own sight difficulty. Algorithm start from the neighbor position, compute sight cost and continue to the next tile. Computation not include starting position and target position. If remaining sight on the end is greater than zero, unit could see that position. If the line hitting exact border of two hex tiles, algorithm try to use both and use better result.



4.4 Attack range

Positions that unit could attack to are using same computation as sight but using attack range attribute.

4.5 Unit placing

When you are planning your defence you need to place units on the map. There are few restrictions about that

- You must place unit on visible tile (Base could be spawned anywhere)
- Spawn tile must be empty
- You must have enough resources for that unit
- You can spawn only unit of your role (attacker / defender)
- As a defender, you can't spawn units on the map edge
- *For UI control, you can't spawn unit if you are in browsing mode*

When you are place unit, you must specify filters for unit (attack filters for both roles, mover filters only for attackers). Unit filters are unchangeable after unit spawn, so think twice before you spawn your units.

Units could attack immediately after spawn, so you don't need to wait, also units can move immediately.

4.6 Filters

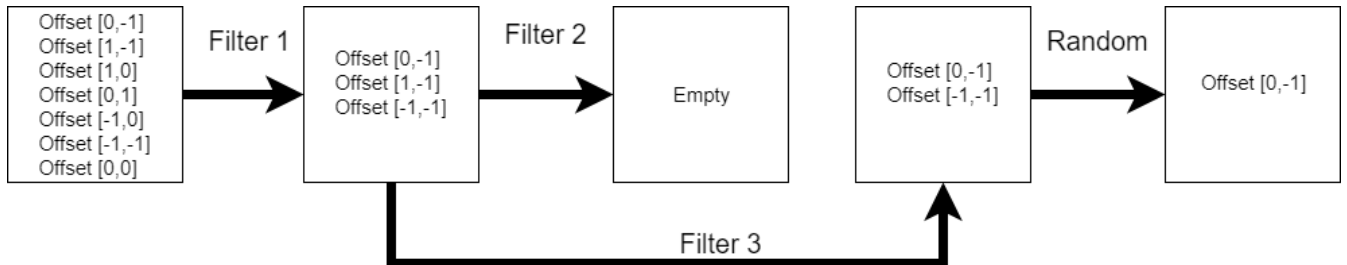
For the unit movement and attacking, Orodael Turrim using filter system. Framework using filters because expert system should define only unit spawn and don't need to decide attack strategy each turn. This is a great simplification for expert system.

System of the filters are simple. Framework take all possible positions and start applying filters from the top. Return value of the filter is sub-set of the positions. For each filter use there are described behaviour based on the return value

- If filter return empty positions set, this filter will not be used (return to previous positions and use next one)

- If filter return exactly one position, all remaining filter will not be used (not necessary)
- If in the set left more than one position after usage of all filters, framework chose on position randomly.

Filters have access to game proxy, so it can use information about terrain, unit attributes and other. User could write own filters, this functionality is described at [Custom filters](#). Filters for move actions and attack actions are same only using different starting set of positions (accessible tiles for move and tiles based on attack range for attack).



4.7 Attacker spawn information

In the game you can obtain information about incoming enemies. Unfortunately, your spies are not among the elite, so the information are not exactly accurate. Your spies trying to get the best information each round, so each round you can get better information.

The system is simple, each round spies try to get best information about incoming enemies. Each round are more or less successful. If they are successful more than previous turn, information about incoming enemies are improved. If scouts have bad day, they only send you information about unknown incoming units. Also there are some probability, that units change direction on the last moment.

Result of this is list of information for N rounds (N is defined by current AI module). Each round those information are best what you can have. This information includes the type of upcoming unit and a list of fields where they can appear in the future.

It is up to you whether you will trust your Scouts or just use the well known information.

5.1 Map terrains

The game map is rugged and you can find various types of terrain on it. Some of them gives your units bonuses but some of them will cause you hard moments in game strategy. Chose wise where you spawn your defense, you cannot change it.

Table 1: **Field**

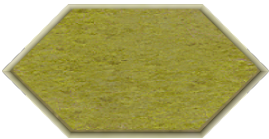
	<p>Description: This to horizon stretching plane of yellow crop provides neither bonuses nor penalties. Just an opportunity to ruin another harvest</p> <p>Bonus: None</p> <p>Penalty: None</p> <p>Sight cost: 1</p>
---	--

Table 2: **Hill**

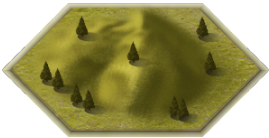
	<p>Description: Someone kept on throwing piles of dirt here and now look, there is a hill. There is a nice view.</p> <p>Bonus: Increases sight by 1, increase attack by 10% and defence by 10%</p> <p>Penalty: None</p> <p>Sight cost: half of remaining</p>
---	--

Table 3: **Forest**

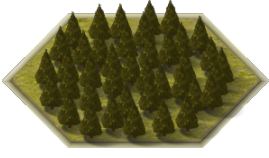
	<p>Description: The shadows of the trees provide shelter from enemy arrows and the bushes make excellent place for an ambush. However, entering the forest might prove bit exhausting.</p> <p>Bonus: Increase defence by 20%</p> <p>Penalty: None</p> <p>Sight cost: 3</p>
---	--

Table 4: **Mountain**


	<p>Description: Everyone who tried climbing those knows, it is not a piece of cake. On the other hand, they provide great place to stay safe, since nobody wants to climb them either.</p> <p>Bonus: Increase defence by 50%, increase sight by 3</p> <p>Penalty: Decrease attack by 20%, lose 5% of max HP each round</p> <p>Sight cost: All</p>
---	---

Table 5: **River**



	<p>Description: Does not matter if it's river, lake or pond, nobody wants to get wet. Especially not Larry (he cannot swim).</p> <p>Bonus: None</p> <p>Penalty: Decrease attack by 20%, decrease attack by 20%, decrease actions by 1</p> <p>Sight cost: 1</p>
---	--

Table 6: **Village**

	<p>Description: Little village in the countryside. Few huts, church and pub - everything a simple adventurer would need and even more!</p> <p>Bonus: Increase defense by 30%, increase actions by 1</p> <p>Penalty: None</p> <p>Sight cost: 1</p>
---	---

5.2 Map movement

Movement across the map could be sometimes hard. It is not easy to climb to the mountain, but when you are starting on the mountain ridge, it is not that hard. Here you can see table with movements from one type of the terrain to other one. The row describing starting terrain type and the column represent target of your journey.

	Field	Hill	Forest	River	Mountain	Village
Field	1	2	2	2	3	1
Hill	2	1	2	2	2	2
Forest	1	2	1	2	3	1
River	2	3	3	1	4	2
Mountain	3	3	3	3	2	3
Village	1	2	2	2	3	1

6.1 Defenders

You can use any of these units. Some of them are really helpful, some of them not really. But you will use them probably.

Table 1: **Knight**


	<p> Attack: 15 Defense: 12 Sight: 2 Price: 12 Lives: 80 Attack range: 1 Active ability: Large shiny shield - Blind Passive ability: Heavy armor - Root resistance </p>
---	---

Table 2: **Magician**


	<p> Attack: 40 Defense: 5 Sight: 3 Price: 30 Lives: 75 Attack range: 2 Active ability: Fireball - Burn Passive ability: Flame-proof cloak - Burn resistance </p>
---	---

Table 3: **Archer**


	<p> Attack: 10 Defense: 1 Sight: 4 Price: 5 Lives: 30 Attack range: 3 Active ability: - Passive ability: Blind resistance </p>
--	---

Table 4: **Ent**



	<p> Attack: 10 Defense: 15 Sight: 2 Price: 50 Lives: 250 Attack range: 1 Active ability: Nature harmony - Root Passive ability: Hard bark - Freeze resistance </p>
---	---

Table 5: **Druid**

	<p>Attack: 30 Defense: 5 Sight: 5 Price: 25 Lives: 100 Attack range: 2 Active ability: Solid winter - freeze Passive ability: -</p>
---	---

6.2 Attackers

Rigor Mortis servants. You need to show them who is the King here.

Table 6: **Cyclops**


	<p>Attack: 20 Defense: 10 Sight: 3 Price: 60 Lives: 80 Attack range: 2 Active ability: - Passive ability: Hard skin - Burn resistance Actions: 2</p>
---	---

Table 7: **Daemon**


	<p> Attack: 35 Defense: 5 Sight: 3 Price: 80 Lives: 150 Attack range: 3 Active ability: Fireball - burn Passive ability: Magical immune - burn and freeze resistance Actions: 2 </p>
---	---

Table 8: **Elemental**


	<p> Attack: 12 Defense: 10 Sight: 2 Price: 35 Lives: 60 Attack range: 2 Active ability: Flame body - burn Passive ability: Flame body - burn Actions: 3 </p>
--	---

Table 9: Gargoyle


	<p> Attack: 10 Defense: 10 Sight: 5 Price: 30 Lives: 60 Attack range: 2 Active ability: - Passive ability: Stone skin - freeze resistance Actions: 3 </p>
---	--

Table 10: Minotaur


	<p> Attack: 15 Defense: 20 Sight: 2 Price: 50 Lives: 150 Attack range: 1 Active ability: - Passive ability: Strong legs - root resistance Actions: 2 </p>
--	--

Table 11: Necromancer


	<p> Attack: 20 Defense: 5 Sight: 3 Price: 50 Lives: 75 Attack range: 2 Active ability: - Passive ability: Levitation - root resistance Actions: 3 </p>
---	---

Table 12: Orc



	<p> Attack: 8 Defense: 5 Sight: 3 Price: 8 Lives: 35 Attack range: 1 Active ability: - Passive ability: - Actions: 2 </p>
--	--

Table 13: Skeleton

	<p> Attack: 5 Defense: 2 Sight: 3 Price: 5 Lives: 20 Attack range: 1 Active ability: - Passive ability: No eyes - blind protection Actions: 2 </p>
---	---

6.3 Effects

In the game, there are several effects that can inflict your or enemy unit. Some unit are resistant to that effects, so pick target unit wisely when you are attacking.

Table 14: Burn


	<p> Description: Burning, effectively reducing hit points over time. Ssssmokin. Effect: Losing 5 lives per round </p>
---	--

Table 15: Freeze


	<p> Description: Freezing unit cannot move so fast. Like ... sooooo cool Effect: Reduce move actions to half </p>
---	--

Table 16: **Root**



	<p>Description: When object is rooted and therefore not able to inflict so much damage. Groot, is that you?</p> <p>Effect: Reduce attack to half</p>
---	--

Table 17: **Blind**

	<p>Description: So bright light. It is take a time until you normal sight</p> <p>Effect: Reduce sight to half</p>
---	---

CHAPTER 7

Expert system

The main module of Orodael Turrin is an expert system. Your task is to create your own planning expert system that will plan defense against Rigor Mortis. All behavior of the expert system is extracted to the `User` module, so you don't need to change anything in the program code.

Also, it is forbidden to change anything in other parts of the code. Tests of the Expert system will be executed with the original modules, only with you `User` module, so any other changes will be ignored.

Structure of the expert system is divided into 3 parts:

- `KnowledgeBase.py` - part of the module for creating knowledge based on game state
- `Inference.py` - part of the module for inference mechanism
- `ActionBase.py` - part of the module for defining conclusions and actions of conclusions

You need to implement all three parts to achieve a working expert system. As an example, there is already implemented basic version of inference, but did not support all necessary function, so use that example only for inspiration.

Warning: When you are creating an expert system, there are few restrictions for implementation. If you violate these restrictions, your solution will not be accepted.

7.1 Knowledge base

The first part that you should implement is `KnowledgeBase`. In this part of the module you should get all the necessary information from proxy and transform that information to `Facts`. Those `Facts` you will use in the inference mechanism.

7.1.1 Fact

You must use the `ExpertSystem.Structure.RuleBase.Fact` class to represent facts, otherwise you can't use some functions. Class `Fact` have 4 instance variables:

name

Represent the name of the fact. The name should be always string. The same name is then used in rules file.

eval_function

Instance variable `eval_function` is used for evaluation of rules conditions. We want to decide if the condition in the rule is satisfied or not. For basic use, when we assume that once a fact is in the base than is True, we do not need to change this variable. By default `eval_function` is True.

If we want to use operators (<, >, ==, ...), we need to set value to the fact. Also, the value of the fact could depend on the given parameters. That is the reason, why `eval_function` is defined as function (Callable). Easies way to define `eval_function` is with lambda functions.

The first option is very simple. We define a function, that always returns a number. There are no additional parameters from the rules. This could be used for compare operators (<, >, ==, ...).

```
archer_count = 5
Fact('archer_count', lambda : archer_count)
```

If you want to pass an argument from rules to the fact, you need to add arguments to the lambda function.

```
Fact('unit_count', lambda unit: get_num_unit(unit))
```

Also, you can call a function inside the lambda function. For example, if you want to have info about player money.

```
Fact('money', lambda : self.game_object_proxy.get_resources(self.player))
```

Using the lambda function is not that hard, so don't worry about that.

probability

The next instance variable is `probability`. With this parameter, you can store the probability of your fact. It using standard float numbers, so there is no magic.

data

The last instance variable is `data`. This variable is used for passing parameters from condition to conclusion. If you mark fact in your rule file with data handle mark (*) then you must specify `data` variable of that fact. TYPe of this variable is also callable (pointer to function). This function could return only a position or list of positions. It is forbidden to return any other value types. If you didn't mark fact with handle mark, you don't need to specify `data` variable. Example of usage:

```
# KnowledgeBase.py
def free_tile():
    return self.map_proxy.get_player_visible_tiles()[0]

Fact('free_tile', data=free_tile)

# rules
IF free_tile* THEN build_base free_tile;
```

Function defined by `data` variable will be evaluated when you call `build_base` from inference. So the result will be up to date with the game state.

Framework also tries to pass arguments to `data` function. Evaluation tries pass all arguments to `data` function and if there is conflict, the framework tries evaluate `data` function without parameters. So if your eval function needs parameters but your `data` function not, it is no problem. Here is an example of `data` function with parameters


```
# KnowledgeBase.py
def free_tile(terrain_type):
    tiles = self.map_proxy.get_player_visible_tiles()
    border_tiles = self.map_proxy.get_border_tiles()

    for position in tiles:
        terrain = self.map_proxy.get_terrain_type(position) == TerrainType.from_
        ↪string(terrain_type)
        occupied = self.map_proxy.is_position_occupied(position)
        if terrain and not occupied and position not in border_tiles:
            return position
    return None

Fact('free_tile', data=free_tile)

# rules
IF free_tile* mountain THEN build_base free_tile;
```

For creating knowledge base you could use 3 types of proxy:

- MapProxy - access information about map
- GameObjectProxy - access information about game objects
- GameUncertaintyProxy - access information about uncertainty spawns

You can read more about proxy methods at [Proxy definition](#)

A list of the facts that you generate in `KnowledgeBase.create_knowledge_base` will be passed to inference method. You need to return a list of facts from the method.

Warning: For creating facts use only proxy! Don't try to get more information from the GameEngine itself!

7.2 Inference

In the inference part of the User module, you should define your inference method. You could use forward or backward inference. Whole inference implementation is up to you. You must implement inference in `interfere` method because this method will be executed each round in the game. Of course, you can implement other supported functions for better code structure, but the entry point must be the `interfere` method. Don't change the signature of the `interfere` method, you will never call inference directly. `interfere` method provides parameters:

- `knowledge_base` - list of facts from `KnowledgeBase.create_knowledge_base`
- `rules` - list of rules from rules files in tree representation
- `action_base` - special class used to call functions that you define in `ActionBase`

Knowledge base

Argument `knowledge_base` contains facts, that you prepare in the `KnowledgeBase` module. There is only one change, all data variables are wiped out. The passing of data variables are described below.

Rules structure

Framework will parse rules file for you. You will get rules as a list of the `ExpertSystem.Structure.RuleBase.Rule` instances. Each list item represents one rule. You can read about rule structure at [Rules file](#).

Inference method

In the `User` module you can find an example of basic inference method. This is a very simple and useless implementation of inference. Use it only as an example. As a result of the inference, you should call some action from `ActionBase`.

Action base calls

In the inference method, you have a `ActionBaseCaller` instance, that represent your action base.

If you want to call your function from `ActionBase`, you need to use the method `call` from `ActionBaseCaller`. Method `call` get parameter of type `Expression` (conclusion `Expression` node). You must use `Expression` class from given rules, don't try to create a new one. If you call method `call`, all parameters from `Expression` are passed to `ActionBase` method and also all parameters from facts are injected. So basically you don't need to worry about nothing. Just be sure, that you have a method in action base with the correct signature.

`ActionBaseCaller` has also method `has_method` that check, if correspond method exist in `ActionBase`.

Recapitulation

There are important things that need to be done. First, you need to mark your fact as a data holder with `*` in the rules file. Otherwise, the data variables will not be injected. Your `ActionBase` method must contain an argument with the same name as the fact name. There could be more than one data facts. You can combine this also with standard arguments, but data holder arguments must be after positional arguments. Last thing, don't try to pass the data holder fact (parameter) manually. Everything is done by dependency injection.

Example of usage

```
# KnowledgeBase.py
target_position = OffsetPosition(0, 0)
facts.append(Fact('free_tile', data=lambda: target_position))

# rules
IF free_tile* THEN build_base 1 1 free_tile;

# ActionBase.py
def build_base(self, position_x, position_y, free_tile):
    pass

# Inference.py
def conclusion_evaluation(self, root_node: ExpressionNode):
    if self.action_base.has_method(root_node.value):
        self.action_base.call(root_node.value)
    else:
        pass # Add to facts
```

Complex example with proxy call

```
# KnowledgeBase.py
def visible_free_tile(self, terrain_type: str):
    """ Find random free tile with given terrain type """
    tiles = self.map_proxy.get_player_visible_tiles()
    border_tiles = self.map_proxy.get_border_tiles()

    for position in tiles:
        terrain = self.map_proxy.get_terrain_type(position) == TerrainType.from_
        ↪string(terrain_type)
        occupied = self.map_proxy.is_position_occupied(position)
        if terrain and not occupied and position not in border_tiles:
            return position
    return None
```

(continues on next page)

(continued from previous page)

```

facts.append(Fact('free_tile', data=visible_free_tile, eval_function=visible_free_
↪tile))

# rules
IF free_tile* mountain THEN build_base awesome_text free_tile;

# ActionBase.py
def build_base(self, log_text, free_tile):
    pass

# Inference.py
def conclusion_evaluation(self, root_node: ExpressionNode):
    if self.action_base.has_method(root_node.value):
        self.action_base.call(root_node.value)
    else:
        pass # Add to facts

```

As you can see in the example, same function is used for data and eval_function. That because our visible_free_tiles function returns None, if there is no such position. Boolean value of None is False and boolean value of position is True. Of course, data function and eval_function could be different functions.

7.3 Action base

In the action base, you can specify your own conclusions with your own implementation. Just write new method to ActionBase. Your methods could have as many parameters as you want, but you need to provide values of the parameters in the inference. Your methods cannot start with underline, otherwise you cannot use them in inference. Also, if you are using data holder parameters, don't forget about arguments with **same name** as fact with data.

ActionBase class provides access to GameControllerProxy and instance of PlayerTag that represent your player (you need it because of identification).

7.4 Rules file

In the file rules you can specify all your rules. You must use defined language, you can read about it at [Grammar](#). Those rules will be automatically parsed and transformed to tree representation. Each rule have own tree.

Each rule is represented with ExpertSystem.Structure.RuleBase.Rule class. This class have 3 properties

- condition - tree representation of condition, root ExpressionNode
- conclusion - tree representation of conclusion, root ExpressionNode
- uncertainty - probability of whole rule

Each condition and conclusion tree is created with ExpressionNode classes for each node in the tree. ExpressionNode provides 6 properties:

- left - instance of left child node if exists, None if node don't have left child
- right - instance of right child node if exists, None if node don't have right child
- operator - if node have left and right child, there is specified operator between them (LogicalOperator)
- value - if node is leaf, there is specified expression (Expression)
- parent - instance of node parent, None if node is root

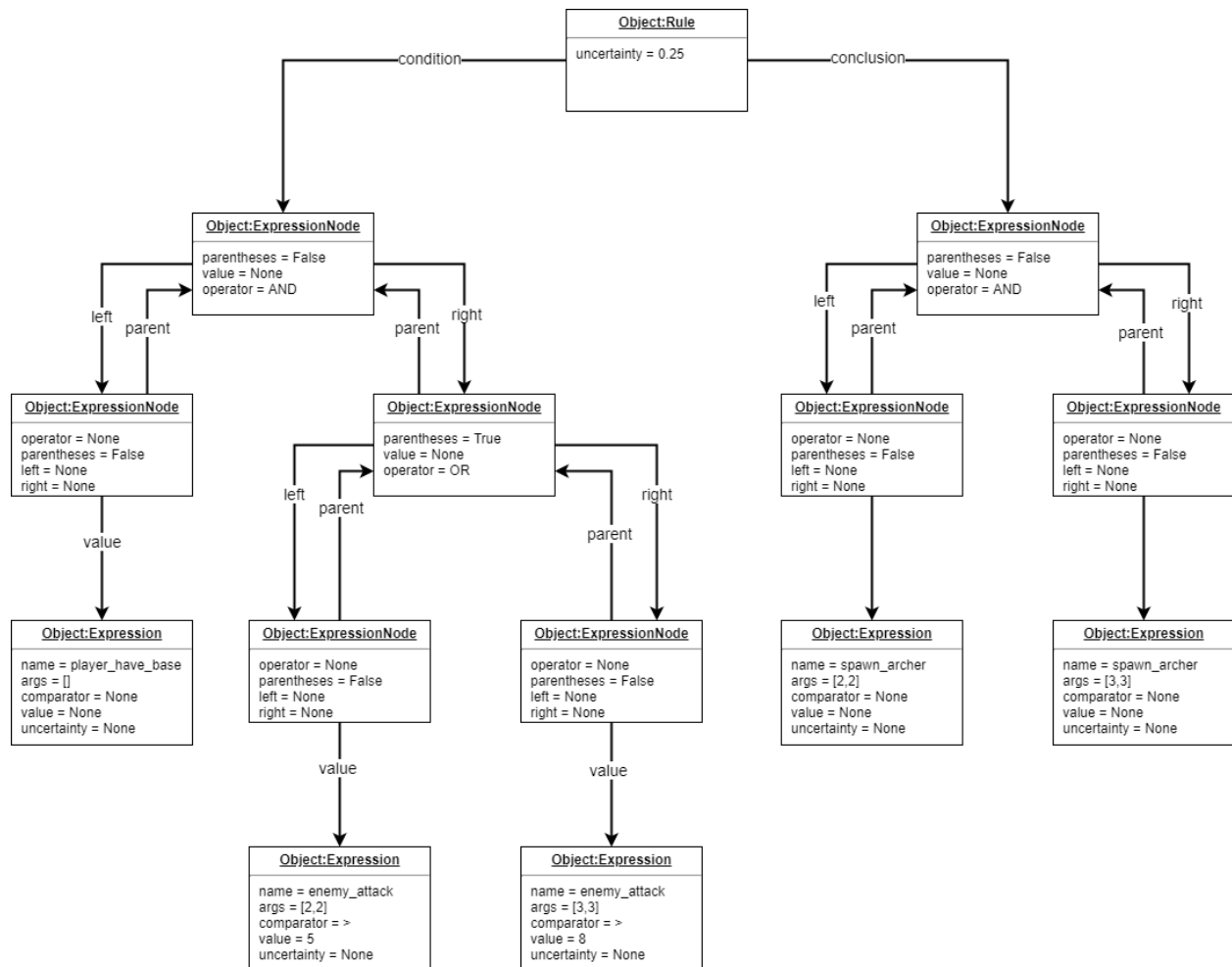
- parentheses - True if current node is in parentheses in rule, False otherwise

Leafs are Expression classes. They represent one part of the rule. Expression class provides 5 properties:

- name - name of the identifier (fact)
- args - list of arguments provided to the fact
- comparator - comparator between fact and value (Operator)
- value - value on the right side of comparator
- uncertainty - probability of this part of rule
- data_holder_mark - True, if fact is marked as data holder

Example of the tree

IF player_have_base AND (enemy_attack 2 2 > 5 OR enemy_attack 3 3 > 8) THEN spawn_↪archer 2 2 AND spawn_archer 3 3 WITH 0.25;



7.5 Custom filters

In the section [Filters](#) you can read about move and attack filter system. Now talk about how to create own custom filters. As a defender, you can use only attack filters, because your unit cannot move. But also some smart attack filters could be really handy in some cases.

If you want to define you own filter, you need to create new class that inherit from `OrodaelTurrim.Structure.Filter.FilterPattern.AttackFilter`. There are some restrictions for your filters:

- Your filter class must be in `AttackFilter.py` file in User module
- Your filter must inherit only `OrodaelTurrim.Structure.Filter.FilterPattern.AttackFilter`
- Your filter must overload `filter` method with same parameters
- `filter` method must return List of tiles and tiles must be subset of given tiles List
- You can overload `__init__` method but first two parameters must be same as in abstract class and you must call `__init__` from inherited class
- You can implement as many functions as you wont in filter class

If your class meets all requirements, you will see this filter in GUI and also you can instance your filter with `FilterFactory` (you can instance them directly but then you need to take care of initial parameters).

In the `AttackFilter.py` file you have example of custom filter.

Rules definition language

For defining expert system rules there is prepared language. This language should be universal enough for defining all your rules needs. If something missing in the language, please let me know through GIT issue in the repository.

Language is processed with parser and lexer based on antlr4, which gives you many advantages like:

- Syntax controlling
- Error messages with bad rules format
- No need to write own Python parser
- Create expression tree for easy evaluation of each rule

So when you try to write some rules with bad syntax, you don't need to check it by yourself, parser will tell you about it. Language is described with grammar. In the grammar you should find out, what you can write. Grammar definition is written in antlr4 language.

8.1 Grammar

```
grammar Rules;

/* Lexical rules */

// Language symbols definition
IF    : 'IF';
THEN  : 'THEN';
WITH  : 'WITH';

AND   : 'AND' ;
OR    : 'OR';

TRUE  : 'TRUE';
FALSE : 'FALSE';
```

(continues on next page)

(continued from previous page)

```

// Assign operator
ASSIGN : ':=';

// Comparison Operators
GE : '>=';
GT : '>';
LE : '<=';
LT : '<';
EQ : '==';
NE : '!=';

// Parenthese for operator order definition
LPAREN: '(';
RPAREN : ')';

// Parenthese for uncertainty
LSPAREN : '[';
RSPAREN : ']';

// Data holder mark
DHM : '*';

// Float number
DECIMAL : '-'?[0-9]+('.'[0-9]+)? ;

// Identifier (Characters, underscore and number)
IDENTIFIER : [a-zA-Z_][a-zA-Z_0-9]* ;

// Each rule ending with semicolon
SEMI : ';';

// Hashtag comments
COMMENT : '#' .+? ('\n'|EOF) -> skip ;

// Skip processing white spaces
WS : [ \r\t\u000C\n]+ -> skip ;

/* Grammar rules */

rules_set : single_rule* EOF;

// Single rule format
single_rule: IF condition THEN conclusion SEMI | IF condition THEN conclusion WITH_
↳DECIMAL SEMI;

condition: left_logical_expr;
conclusion: right_logical_expr;

// Condition format
left_logical_expr
: left_logical_expr AND left_logical_expr # LogicalExpressionAnd
| left_logical_expr OR left_logical_expr # LogicalExpressionOr
| function_expr # ComparisonExpression
| function_expr LSPAREN DECIMAL RSPAREN # ComparisonExpression
| LPAREN left_logical_expr RPAREN # LogicalExpressionInParen
;

```

(continues on next page)

(continued from previous page)

```
// One condition expression format
function_expr
    : IDENTIFIER DHM? args
    | IDENTIFIER DHM? args comp_operator DECIMAL
    | IDENTIFIER DHM?
    | IDENTIFIER DHM? comp_operator DECIMAL
    | IDENTIFIER DHM? comp_operator IDENTIFIER
    | IDENTIFIER DHM? args comp_operator IDENTIFIER
    | (TRUE | FALSE);

args : arg args | arg;
arg : DECIMAL | IDENTIFIER;
comp_operator : GT | GE | LT | LE | EQ | NE;

// Conclusion format
right_logical_expr
    : right_logical_expr AND right_logical_expr      # RLogicalExpressionAnd
    | LPAREN right_logical_expr RPAREN                # RLogicalExpressionInParen
    | r_function_expr                                # RLogicalExpression
    ;

// One conclusion expression format
r_function_expr
    : IDENTIFIER
    | IDENTIFIER args
    | IDENTIFIER args ASSIGN DECIMAL
    | IDENTIFIER args ASSIGN IDENTIFIER
    | IDENTIFIER ASSIGN DECIMAL
    | IDENTIFIER ASSIGN IDENTIFIER;
```

8.2 Examples

Here you can find some examples, about what can be written in the language. It is make no sense to write all possibilities, because there are infinity number of possibilities. It is recommended to look at grammar definition and learn what is possible by that. Here you can find some inspiration. All examples are correct so you can copy them to rule file and execute inference, but you probably don't get any output, because of missing action definition.

Rule with no condition

```
IF TRUE THEN build_base_random;
```

Rule with simple condition

```
IF player_dont_have_base THEN build_base_random;
```

Parametrized expressions

```
IF player_dont_have_base THEN build_base 2 2;
```

Expression with comparision

```
IF base_count == 0 THEN build_base 2 2;
```

Expression with comparision and arguments

```
IF object_count base == 0 THEN build_base 2 2;
```

Expression with data holder mark and arguments

```
IF free_position* mountain THAT build_archer free_position;
```

Logical operators in condition

```
IF object_count base == 0 AND terrain 2 2 == forest THEN build base 2 2;  
IF object_count archer < 2 OR object_count knight < 2 THEN spawn_archer 1 3;
```

Logical operators in conclusion

```
IF object_count archer < 2 THEN spawn_archer 1 3 AND spawn_archer 5 5;
```

Parentheses for logical operators order

```
IF player_have_base AND ( object_count archer < 2 OR object_count knight < 2 ) THEN_  
↪spawn_archer 1 3;
```

Defining value in conclusion

```
IF object_count archer > 10 THEN strength := 10;  
IF object_count archer > 10 THEN strength := high;
```

Probability of whole rule

```
IF object_count archer > 10 THEN strength := 10 WITH 0.50;
```

Probability of condition expressions

```
IF object_count archer > 10 [0.25] AND object_count knight > 5 [0.8] THEN strength :=_  
↪10;
```

9.1 Map proxy

```
class OrodraelTurrim.Business.Proxy.MapProxy (game_engine: GameEngine)
```

```
compute_accessible_tiles (position: Position, actions: int) → Optional[Dict[Position, int]]
```

Computes map with accessible tiles as keys and remaining action points as values from specified position and number of remaining action points

Parameters

- **position** – Position to use as base point of computation
- **actions** – Number of action points to consider for computation

Returns Dict with accessible tiles as keys and remaining action points as values None if position is not on map

```
compute_visible_tiles (position: Position, sight: int) → Optional[Set[Position]]
```

Computes set of visible tiles in sight radius from given position.

Parameters

- **position** – Position to use as base point of computation
- **sight** – Value of sight to consider for computation

Returns Set of visible tiles of specified game object. None if position is not on map

```
get_bases_positions () → Set[Position]
```

Retrieves positions of defenders' bases

Returns Positions of defenders' bases

```
get_border_tiles () → Set[Position]
```

Retrieves set of tiles on the edge of game map

get_inner_tiles () → Set[Position]

Retrieves set of tiles which are not on the map edge

get_map_height () → int

Retrieves number of tiles in each column of game map

get_map_width () → int

Retrieves number of tiles in each row of game map

get_player_visible_tiles () → Set[Position]

Retrieves set of visible tiles for player.

Returns Set of visible tiles

get_terrain_type (position: Position) → Optional[TerrainType]

Retrieves terrain type of given position Return None if Positions is not on map

Parameters **position** – Position to get terrain type for

Returns Terrain type of given position

is_position_occupied (position: Position) → bool

Checks whether given position is occupied or not. You can check only visible positions

Parameters **position** – Position to be checked

Returns True in case there is game object on given position, False otherwise, None if user did not see the position

is_position_on_map (position: Position) → bool

Checks whether given position is on map or not

Parameters **position** – Position to be checked

Returns True in case position is within map bounds, False otherwise

player_have_base (player: PlayerTag) → bool

Check if player already have a base

Parameters **player** – Target player to be checked

Returns True if player have base, False otherwise

9.2 Game Object Proxy

class OrodaelTurrim.Business.Proxy.GameObjectProxy (game_engine: GameEngine)

get_active_effects (position: Position) → Optional[Dict[EffectType, int]]

Retrieves types of currently active effects and their durations on game object on specified position

Parameters **position** – Position of queried game object

Returns Dict of types of active effects and associated remaining durations None if there is no unit at the position or you don't see that position

get_attack_effects (position: Position) → Optional[Set[EffectType]]

Retrieves the types of effect to be applied to the target of attack of game object on specified position

Parameters **position** – Position of queried game object

Returns Set of types of effect to be applied upon attacking None if there is no unit at the position or you don't see that position

get_attribute (*position: Position, attribute_type: AttributeType*) → Optional[float]

Retrieves value of specified attribute of game object on specified position

Parameters

- **position** – Position of queried game object
- **attribute_type** – Type of attribute to be retrieved

Returns Value of specified attribute None if there is no unit at the position

get_current_hit_points (*position: Position*) → Optional[float]

Retrieves amount of currently remaining hit points of game object on specified position

Parameters **position** – Position of queried game object

Returns Amount of currently remaining hit points None if there is no unit at the position or you don't see that position

get_current_round () → int

Returns

get_income (*player: PlayerTag*) → int

Retrieves income of given player

Parameters **player** – Player whose income should be obtained

Returns Current income of given player None if player not registered

get_object_type (*position: Position*) → Optional[GameObjectType]

Retrieves the type of game object on the specified position. The player must see that position. This function could be used to get enemy types.

Parameters **position** – Position of queried game object

Returns Type of game object on specified position GameObjectType.NONE if there is no unit at the position None if player don't see that position

get_resistances (*position: Position*) → Optional[Set[EffectType]]

Retrieves the types of effect which will NOT affect game object on specified position

Parameters **position** – Position of queried game object

Returns Set of resistances of game object on specified position None if there is no unit at the position or player don't see that position

get_resources (*player: PlayerTag*) → int

Retrieves current resources of given player

Parameters **player** – Player whose resources should be obtained

Returns Current resources of given player None if player not registered

get_role (*position: Position*) → GameRole

Retrieves the role of game object on specified position

Parameters **position** – Position of queried game object

Returns Role of game object on specified position GameRole.NEUTRAL if there is no unit at the position, None if you don't see that position

get_visible_enemies (*position: Position*) → Optional[Dict[Position, int]]

Retrieves map of distances to currently visible enemies by game object on specified position

Parameters **position** – Position of queried game object

Returns Dictionary of visible position with enemy as a Key and distance as a value Return None if there is no unit at the position, None if you don't see target position

get_visible_tiles (*position: Position*) → Optional[Set[Position]]
Retrieves set of currently visible tiles of game object on specified position

Parameters **position** – Position of queried game object

Returns Set of currently visible tiles None if there is no unit at the position, None if you don't see target position

9.3 Game Control Proxy

class OrodaelTurrim.Business.Proxy.**GameControlProxy** (*game_engine: GameEngine*)

spawn_unit (*information: SpawnInformation*) → None
Attempts to spawn unit based on given spawn information

Parameters **information** – Information bundle describing spawned unit

Raise `IllegalActionException` if invalid spawn attempt

9.4 Game Uncertainty Proxy

class OrodaelTurrim.Business.Proxy.**GameUncertaintyProxy** (*game_engine: GameEngine*)

spawn_information () → List[List[UncertaintySpawn]]

Get spawn information from uncertainty module.

First level is rounds, where 0 is the nearest round

Second level is list of `UncertaintySpawn` classes

Returns Spawn information from Uncertainty module

10.1 OrodaelTurrin package

10.1.1 Subpackages

OrodaelTurrin.Business package

Subpackages

OrodaelTurrin.Business.Interface package

Submodules

OrodaelTurrin.Business.Interface.Player module

Module contents

Submodules

OrodaelTurrin.Business.Factory module

OrodaelTurrin.Business.GameEngine module

OrodaelTurrin.Business.GameMap module

OrodaelTurrin.Business.History module

OrodaelTurrim.Business.Logger module

OrodaelTurrim.Business.MapGenerator module

OrodaelTurrim.Business.Proxy module

class OrodaelTurrim.Business.Proxy.**GameControlProxy** (*game_engine: GameEngine*)

Bases: object

spawn_unit (*information: SpawnInformation*) → None

Attempts to spawn unit based on given spawn information

Parameters **information** – Information bundle describing spawned unit

Raise **IllegalActionException** if invalid spawn attempt

class OrodaelTurrim.Business.Proxy.**GameObjectProxy** (*game_engine: GameEngine*)

Bases: object

get_active_effects (*position: Position*) → Optional[Dict[EffectType, int]]

Retrieves types of currently active effects and their durations on game object on specified position

Parameters **position** – Position of queried game object

Returns Dict of types of active effects and associated remaining durations None if there is no unit at the position or you don't see that position

get_attack_effects (*position: Position*) → Optional[Set[EffectType]]

Retrieves the types of effect to be applied to the target of attack of game object on specified position

Parameters **position** – Position of queried game object

Returns Set of types of effect to be applied upon attacking None if there is no unit at the position or you don't see that position

get_attribute (*position: Position, attribute_type: AttributeType*) → Optional[float]

Retrieves value of specified attribute of game object on specified position

Parameters

- **position** – Position of queried game object
- **attribute_type** – Type of attribute to be retrieved

Returns Value of specified attribute None if there is no unit at the position

get_current_hit_points (*position: Position*) → Optional[float]

Retrieves amount of currently remaining hit points of game object on specified position

Parameters **position** – Position of queried game object

Returns Amount of currently remaining hit points None if there is no unit at the position or you don't see that position

get_current_round () → int

Returns

get_income (*player: PlayerTag*) → int

Retrieves income of given player

Parameters **player** – Player whose income should be obtained

Returns Current income of given player None if player not registered

get_object_type (*position: Position*) → Optional[GameObjectType]

Retrieves the type of game object on the specified position. The player must see that position. This function could be used to get enemy types.

Parameters **position** – Position of queried game object

Returns Type of game object on specified position GameObjectType.NONE if there is no unit at the position None if player don't see that position

get_resistances (*position: Position*) → Optional[Set[EffectType]]

Retrieves the types of effect which will NOT affect game object on specified position

Parameters **position** – Position of queried game object

Returns Set of resistances of game object on specified position None if there is no unit at the position or player don't see that position

get_resources (*player: PlayerTag*) → int

Retrieves current resources of given player

Parameters **player** – Player whose resources should be obtained

Returns Current resources of given player None if player not registered

get_role (*position: Position*) → GameRole

Retrieves the role of game object on specified position

Parameters **position** – Position of queried game object

Returns Role of game object on specified position GameRole.NEUTRAL if there is no unit at the position, None if you don't see that position

get_visible_enemies (*position: Position*) → Optional[Dict[Position, int]]

Retrieves map of distances to currently visible enemies by game object on specified position

Parameters **position** – Position of queried game object

Returns Dictionary of visible position with enemy as a Key and distance as a value Return None if there is no unit at the position, None if you don't see target position

get_visible_tiles (*position: Position*) → Optional[Set[Position]]

Retrieves set of currently visible tiles of game object on specified position

Parameters **position** – Position of queried game object

Returns Set of currently visible tiles None if there is no unit at the position, None if you don't see target position

class OrodaelTurrim.Business.Proxy.**GameUncertaintyProxy** (*game_engine: GameEngine*)

Bases: object

spawn_information () → List[List[UncertaintySpawn]]

Get spawn information from uncertainty module.

First level is rounds, where 0 is the nearest round

Second level is list of UncertaintySpawn classes

Returns Spawn information from Uncertainty module

class OrodaelTurrim.Business.Proxy.**MapProxy** (*game_engine: GameEngine*)

Bases: object

compute_accessible_tiles (*position: Position, actions: int*) → Optional[Dict[Position, int]]
 Computes map with accessible tiles as keys and remaining action points as values from specified position and number of remaining action points

Parameters

- **position** – Position to use as base point of computation
- **actions** – Number of action points to consider for computation

Returns Dict with accessible tiles as keys and remaining action points as values None if position is not on map

compute_visible_tiles (*position: Position, sight: int*) → Optional[Set[Position]]
 Computes set of visible tiles in sight radius from given position.

Parameters

- **position** – Position to use as base point of computation
- **sight** – Value of sight to consider for computation

Returns Set of visible tiles of specified game object. None if position is not on map

get_bases_positions () → Set[Position]
 Retrieves positions of defenders' bases

Returns Positions of defenders' bases

get_border_tiles () → Set[Position]
 Retrieves set of tiles on the edge of game map

get_inner_tiles () → Set[Position]
 Retrieves set of tiles which are not on the map edge

get_map_height () → int
 Retrieves number of tiles in each column of game map

get_map_width () → int
 Retrieves number of tiles in each row of game map

get_player_visible_tiles () → Set[Position]
 Retrieves set of visible tiles for player.

Returns Set of visible tiles

get_terrain_type (*position: Position*) → Optional[TerrainType]
 Retrieves terrain type of given position Return None if Position is not on map

Parameters **position** – Position to get terrain type for

Returns Terrain type of given position

is_position_occupied (*position: Position*) → bool
 Checks whether given position is occupied or not. You can check only visible positions

Parameters **position** – Position to be checked

Returns True in case there is game object on given position, False otherwise, None if user did not see the position

is_position_on_map (*position: Position*) → bool
 Checks whether given position is on map or not

Parameters **position** – Position to be checked

Returns True in case position is within map bounds, False otherwise

player_have_base (*player: PlayerTag*) → bool

Check if player already have a base

Parameters **player** – Target player to be checked

Returns True if player have base, False otherwise

OrodaeITurrim.Business.Thread module

OrodaeITurrim.Business.Uncertainty module

Module contents

OrodaeITurrim.Presenter package

Subpackages

OrodaeITurrim.Presenter.Widgets package

Submodules

OrodaeITurrim.Presenter.Widgets.ControlWidget module

OrodaeITurrim.Presenter.Widgets.GameControlWidget module

OrodaeITurrim.Presenter.Widgets.LogWidget module

OrodaeITurrim.Presenter.Widgets.MapInfoWidget module

OrodaeITurrim.Presenter.Widgets.MapWidget module

OrodaeITurrim.Presenter.Widgets.RoundControlWidget module

OrodaeITurrim.Presenter.Widgets.SpawnInfoWidget module

OrodaeITurrim.Presenter.Widgets.UnitSpawnInfoWidget module

OrodaeITurrim.Presenter.Widgets.UnitWidget module

Module contents

Submodules

OrodaeITurrim.Presenter.Connector module

OrodaeITurrim.Presenter.Main module

OrodaelTurrim.Presenter.Utils module

Module contents

OrodaelTurrim.Structure package

Subpackages

OrodaelTurrim.Structure.Actions package

Submodules

OrodaelTurrim.Structure.Actions.Abstract module

```
class OrodaelTurrim.Structure.Actions.Abstract.GameAction (game_engine:  
                                                         GameEngine)  
    Bases: abc.ABC  
    Core class defining methods for game actions (history marks)  
    TIME_STAMP_FORMAT = '%H:%M:%S'  
    execute () → None  
        Executes given game action  
    text  
        Returns text representation of this game action to show, what happened  
    undo () → None  
        Undoes any effects, which this game action caused effectively returning game to the state before this action  
        happened  
    xml (parent: xml.etree.ElementTree.Element) → _elementtree.SubElement  
        Convert action to XML tag
```

OrodaelTurrim.Structure.Actions.Combat module

OrodaelTurrim.Structure.Actions.Effect module

OrodaelTurrim.Structure.Actions.Log module

```
class OrodaelTurrim.Structure.Actions.Log.LogAction (game_engine:      GameEngine,  
                                                         log_message: str)  
    Bases: OrodaelTurrim.Structure.Actions.Abstract.GameAction  
    User custom log action  
    execute () → None  
        Executes given game action  
    text  
        Returns text representation of this game action to show, what happened  
    undo () → None  
        Undoes any effects, which this game action caused effectively returning game to the state before this action  
        happened
```

xml (*parent*) → `_elementtree.SubElement`
Convert action to XML tag

OrodaelTurrim.Structure.Actions.Placement module

OrodaelTurrim.Structure.Actions.Resources module

OrodaelTurrim.Structure.Actions.Terrain module

OrodaelTurrim.Structure.Actions.Turn module

Module contents

OrodaelTurrim.Structure.GameObjects package

Subpackages

OrodaelTurrim.Structure.GameObjects.Prototypes package

Submodules

OrodaelTurrim.Structure.GameObjects.Prototypes.Attackers module

OrodaelTurrim.Structure.GameObjects.Prototypes.Defenders module

OrodaelTurrim.Structure.GameObjects.Prototypes.Prototype module

Module contents

Submodules

OrodaelTurrim.Structure.GameObjects.Attributes module

OrodaelTurrim.Structure.GameObjects.Effect module

OrodaelTurrim.Structure.GameObjects.GameObject module

Module contents

Submodules

OrodaelTurrim.Structure.Enums module

OrodaelTurrim.Structure.Exceptions module

exception OrodaelTurrim.Structure.Exceptions.**BadActionBaseParameters**

Bases: *OrodaelTurrim.Structure.Exceptions.OrodaelTurrimException*

ActionBase method has bad parameters

exception OrodaelTurrim.Structure.Exceptions.**BadFactDataValue**

Bases: *OrodaelTurrim.Structure.Exceptions.OrodaelTurrimException*

Fact contain bad value type

exception OrodaelTurrim.Structure.Exceptions.**IllegalActionException**

Bases: *OrodaelTurrim.Structure.Exceptions.OrodaelTurrimException*

You try to use illegal action on the game engine

exception OrodaelTurrim.Structure.Exceptions.**IllegalArgumentException**

Bases: *OrodaelTurrim.Structure.Exceptions.OrodaelTurrimException*

Illegal argument passed to object initialization

exception OrodaelTurrim.Structure.Exceptions.**IllegalConfigState**

Bases: *OrodaelTurrim.Structure.Exceptions.OrodaelTurrimException*

Something missing in game config file

exception OrodaelTurrim.Structure.Exceptions.**IllegalHistoryOperation**

Bases: *OrodaelTurrim.Structure.Exceptions.OrodaelTurrimException*

Trying to do illegal operation when in Browsing mode

exception OrodaelTurrim.Structure.Exceptions.**IllegalLogMessage**

Bases: *OrodaelTurrim.Structure.Exceptions.OrodaelTurrimException*

You are trying to log message which is not correct type

exception OrodaelTurrim.Structure.Exceptions.**IllegalRulesFormat**

Bases: *OrodaelTurrim.Structure.Exceptions.OrodaelTurrimException*

Problem with parsing rules file

exception OrodaelTurrim.Structure.Exceptions.**OrodaelTurrimException**

Bases: Exception

Base framework exception. All framework exception inherit from this exception

OrodaelTurrim.Structure.Map module

OrodaelTurrim.Structure.Position module

OrodaelTurrim.Structure.Resources module

class OrodaelTurrim.Structure.Resources.**PlayerResources** (*resources: int, income: int, income_increase=0*)

Bases: object

Structure that hold information about player resources

add_resources (*amount: int*) → None

Add amount of resources

```

income
income_increase
increase_income (value: int) → None
remove_resources (amount: int) → None
    Remove amount of resources
resources

```

OrodaelTurrim.Structure.Terrain module

```
class OrodaelTurrim.Structure.Terrain.Field
```

Bases: *OrodaelTurrim.Structure.Terrain.Terrain*

Class representing field on map. This to horizon stretching plane of yellow crop provides neither bonuses nor penalties. Just an opportunity to ruin another harvest.

```
char () → str
    Return character that represent this terrain type for string map definition
```

```
get_move_cost (target: TerrainType) → int
    Get move cost of the terrain based current terrain type and target terrain type. Move cost have value based
    on target and source terrain type
```

Parameters *target* – target terrain type

Returns action cost

```
get_remaining_sigh (current_sight: int) → int
    Get remaining sight after current terrain type
```

Parameters *current_sight* – current sight number

Returns remaining sight

```
info_text ()
    Return text information of the terrain
```

```
terrain_type
```

```
class OrodaelTurrim.Structure.Terrain.Forest
```

Bases: *OrodaelTurrim.Structure.Terrain.Terrain*

Class representing ever green forest. The shadows of the trees provide shelter from enemy arrows and the bushes make excellent place for an ambush. However, entering the forest might prove bit exhausting.

```
affect_defense (original_value: float)
```

```
char () → str
    Return character that represent this terrain type for string map definition
```

```
get_move_cost (target: OrodaelTurrim.Structure.Terrain.Terrain) → int
    Get move cost of the terrain based current terrain type and target terrain type. Move cost have value based
    on target and source terrain type
```

Parameters *target* – target terrain type

Returns action cost

```
get_remaining_sigh (current_sight: int) → int
    Get remaining sight after current terrain type
```

Parameters `current_sight` – current sight number

Returns remaining sight

`info_text()`

Return text information of the terrain

`terrain_type`

class `OrodaelTurrim.Structure.Terrain.Hill`

Bases: `OrodaelTurrim.Structure.Terrain.Terrain`

Class representing little hill. Someone kept on throwing piles of dirt here and now look, there is a hill.

`affect_attack(original_value: float)`

`affect_defense(original_value: float)`

`affect_sight(original_value: int) → int`

`char()` → str

Return character that represent this terrain type for string map definition

`get_move_cost(target: TerrainType) → int`

Get move cost of the terrain based current terrain type and target terrain type. Move cost have value based on target and source terrain type

Parameters `target` – target terrain type

Returns action cost

`get_remaining_sigh(current_sight: int) → int`

Get remaining sight after current terrain type

Parameters `current_sight` – current sight number

Returns remaining sight

`info_text()`

Return text information of the terrain

`terrain_type`

class `OrodaelTurrim.Structure.Terrain.Mountain`

Bases: `OrodaelTurrim.Structure.Terrain.Terrain`

Class representing pointy rock giants. Everyone who tried climbing those knows, it is not a piece of cake. On the other hand, they provide great place to stay safe, since nobody wants to climb them either.

`affect_attack(original_value: float)`

`affect_defense(original_value: float)`

`affect_sight(original_value: int)`

`char()` → str

Return character that represent this terrain type for string map definition

`compute_damage(hit_points: float)`

Computes, how much damage will this terrain inflict on start of each turn

Parameters `hit_points` – Previous value of hit points of game object

Returns Amount of damage to be inflicted to game object

get_move_cost (*target: TerrainType*) → int

Get move cost of the terrain based current terrain type and target terrain type. Move cost have value based on target and source terrain type

Parameters **target** – target terrain type

Returns action cost

get_remaining_sigh (*current_sight: int*) → int

Get remaining sight after current terrain type

Parameters **current_sight** – current sight number

Returns remaining sight

info_text ()

Return text information of the terrain

terrain_type

class OrodaelTurrim.Structure.Terrain.**River**

Bases: *OrodaelTurrim.Structure.Terrain.Terrain*

Class representing mass of water. Does not matter if it's river, lake or pond, nobody wants to get wet. Especially not Larry (he cannot swim).

affect_actions (*original_value: float*)

affect_attack (*original_value: float*)

affect_defense (*original_value: float*)

char () → str

Return character that represent this terrain type for string map definition

get_move_cost (*target: TerrainType*) → int

Get move cost of the terrain based current terrain type and target terrain type. Move cost have value based on target and source terrain type

Parameters **target** – target terrain type

Returns action cost

get_remaining_sigh (*current_sight: int*) → int

Get remaining sight after current terrain type

Parameters **current_sight** – current sight number

Returns remaining sight

info_text ()

Return text information of the terrain

terrain_type

class OrodaelTurrim.Structure.Terrain.**Terrain**

Bases: *abc.ABC*

Abstract class for terrain types.

affect_actions (*original_value: float*) → float

affect_attack (*original_value: float*) → float

affect_attribute (*attribute: AttributeType, original_value: Union[int, float]*) → Union[float, int]

Provides affected value of specified attribute by this terrain

Parameters

- **attribute** – Type of attribute, which should be affected
- **original_value** – Original value of affected attribute

Returns Affected value of specified attribute by this terrain

affect_defense (*original_value: float*) → float

affect_max_hit_points (*original_value: int*) → int

affect_range (*original_value: int*) → int

affect_sight (*original_value: int*) → int

char () → str

Return character that represent this terrain type for string map definition

compute_damage (*hit_points: float*) → float

Computes, how much damage will this terrain inflict on start of each turn

Parameters **hit_points** – Previous value of hit points of game object

Returns Amount of damage to be inflicted to game object

get_move_cost (*target: TerrainType*) → int

Get move cost of the terrain based current terrain type and target terrain type. Move cost have value based on target and source terrain type

Parameters **target** – target terrain type

Returns action cost

get_remaining_sigh (*current_sight: int*) → int

Get remaining sight after current terrain type

Parameters **current_sight** – current sight number

Returns remaining sight

info_text () → str

Return text information of the terrain

terrain_type

class OrodaelTurrim.Structure.Terrain.Village

Bases: *OrodaelTurrim.Structure.Terrain.Terrain*

Class representing little village in the countryside. Few huts, church and pub - everything a simple adventurer would need and even more!

affect_actions (*original_value: float*)

affect_attack (*original_value: float*)

affect_defense (*original_value: float*)

char () → str

Return character that represent this terrain type for string map definition

get_move_cost (*target: OrodaelTurrim.Structure.Terrain.Terrain*) → int

Get move cost of the terrain based current terrain type and target terrain type. Move cost have value based on target and source terrain type

Parameters **target** – target terrain type

Returns action cost

get_remaining_sigh (*current_sight: int*) → int
 Get remaining sight after current terrain type

Parameters **current_sight** – current sight number

Returns remaining sight

info_text ()
 Return text information of the terrain

terrain_type

OrodaelTurrim.Structure.TypeStrucutre module

class OrodaelTurrim.Structure.TypeStrucutre.**DoubleLinkedList**
 Bases: object

Structure for store double linked list

empty () → bool
 Determinate if linked list is empty

head
 Get reference to head Node

next ()
 Move pointer to the next Node

pointer
 Get pointed Node reference

previous ()
 Move pointer to the previous Node

push_back (*value: Any*) → None
 Push value to the end of the linked list

push_front (*value: Any*) → None
 Push value to the start of the linked list

tail
 Get reference to tail Node

value
 Return value of the pointer Node. Raise exception if pointer is not set

class OrodaelTurrim.Structure.TypeStrucutre.**Node** (*data: Any = None*)
 Bases: object

next

previous

value

OrodaelTurrim.Structure.Utills module

Module contents

10.1.2 Submodules

10.1.3 OrodaelTurrim.Main module

10.1.4 OrodaelTurrim.config module

10.1.5 Module contents

10.2 ExpertSystem package

10.2.1 Subpackages

ExpertSystem.Business package

Subpackages

ExpertSystem.Business.Parser package

Subpackages

ExpertSystem.Business.Parser.KnowledgeBase package

Submodules

ExpertSystem.Business.Parser.KnowledgeBase.ErrorListener module

ExpertSystem.Business.Parser.KnowledgeBase.RulesLexer module

ExpertSystem.Business.Parser.KnowledgeBase.RulesListener module

ExpertSystem.Business.Parser.KnowledgeBase.RulesListenerImplementation module

ExpertSystem.Business.Parser.KnowledgeBase.RulesParser module

Module contents

Module contents

Submodules

ExpertSystem.Business.Player module

ExpertSystem.Business.UserFramework module

Module contents

10.2.2 Module contents

10.3 ArtificialIntelligence package

10.3.1 Submodules

10.3.2 ArtificialIntelligence.Filter module

10.3.3 ArtificialIntelligence.Main module

10.3.4 Module contents

10.4 User package

10.4.1 Submodules

10.4.2 User.ActionBase module

10.4.3 User.AttackFilter module

10.4.4 User.Inference module

10.4.5 User.KnowledgeBase module

10.4.6 Module contents

- **1.2.3**
 - Fix bug with fact at conclusion in rules
 - Update documentation based on unit balance
 - Update default rules files
- **1.2.2**
 - Implement bss units to AI player
 - Add `get_current_turn` to `GameObjectProxy`
- **1.2.1**
 - Data variable in `Fact` change from variable to callable
 - Remove option to call actions from action base directly (you need to you call method)
 - Remove option to call actions from action base by string (you need to use `Expression`)
 - Add method `get_inner_tiles` to map proxy. Method returns all non border tiles
 - Documentation update
 - Extended example implementation
- **1.2.0**

Warning: This version is incompatible with 1.1.* versions

- **Implement variable passing from knowledge base to action base**
- Change method, how to call `ActionBase` methods from inference
- Removed `MapProxy` from `ActionBase`
- Rename `Interference.py` to `Inference.py` in `User` module

- Add more details and examples to User module documentation
 - Update documentation of proxy classes
 - Replace `Player` object with `PlayerTag` object in User module
 - Fix bug with console simulation works without base
 - Improve bug reports - add user implementation to report
 - Unit balancing
 - Add option to disable AI console output (`AI_CONSOLE_OUTPUT`)
- **1.1.8**
 - Fix bug with more than two arguments in rules (more arguments were squeezed)
 - Fix bug with bad definition of `LEFT_LOWER` constant for `OffsetPosition`
 - Add some example implementation to User module
 - Fix random exception at operation with gui log widget
 - Some changes in documentation
- **1.1.7**
 - **Add** `MapProxy` to `ActionBase`
 - Updated class doc in `ActionBase`
 - Add console option `-x / -log-output-xml` for XML log output
- **1.1.6**
 - Rise max number in GUI for round simulation to 9999
- **1.1.5**
 - Fix bug with problem of using other position types in `ActionBase`
- **1.1.4**
 - Fix bug in the example evaluation in `Interference` class
- **1.1.3**
 - Add bug reporting feature

a

[ArtificialIntelligence](#), 65
[ArtificialIntelligence.Filter](#), 65

e

[ExpertSystem](#), 65
[ExpertSystem.Business](#), 65
[ExpertSystem.Business.Parser](#), 64
[ExpertSystem.Business.Parser.KnowledgeBase](#),
64

O

[OrodaelTurrin](#), 64
[OrodaelTurrin.Business](#), 55
[OrodaelTurrin.Business.Interface](#), 51
[OrodaelTurrin.Business.Proxy](#), 52
[OrodaelTurrin.Presenter](#), 56
[OrodaelTurrin.Presenter.Widgets](#), 55
[OrodaelTurrin.Structure](#), 64
[OrodaelTurrin.Structure.Actions](#), 57
[OrodaelTurrin.Structure.Actions.Abstract](#),
56
[OrodaelTurrin.Structure.Actions.Log](#), 56
[OrodaelTurrin.Structure.Exceptions](#), 58
[OrodaelTurrin.Structure.GameObjects](#), 57
[OrodaelTurrin.Structure.GameObjects.Prototypes](#),
57
[OrodaelTurrin.Structure.Resources](#), 58
[OrodaelTurrin.Structure.Terrain](#), 59
[OrodaelTurrin.Structure.TypeStrucutre](#),
63

U

[User](#), 65

A

[add_resources\(\)](#) (*OrodaelTurrim.Structure.Resources.PlayerResources method*), 58
[affect_actions\(\)](#) (*OrodaelTurrim.Structure.Terrain.River method*), 61
[affect_actions\(\)](#) (*OrodaelTurrim.Structure.Terrain.Terrain method*), 61
[affect_actions\(\)](#) (*OrodaelTurrim.Structure.Terrain.Village method*), 62
[affect_attack\(\)](#) (*OrodaelTurrim.Structure.Terrain.Hill method*), 60
[affect_attack\(\)](#) (*OrodaelTurrim.Structure.Terrain.Mountain method*), 60
[affect_attack\(\)](#) (*OrodaelTurrim.Structure.Terrain.River method*), 61
[affect_attack\(\)](#) (*OrodaelTurrim.Structure.Terrain.Terrain method*), 61
[affect_attack\(\)](#) (*OrodaelTurrim.Structure.Terrain.Village method*), 62
[affect_attribute\(\)](#) (*OrodaelTurrim.Structure.Terrain.Terrain method*), 61
[affect_defense\(\)](#) (*OrodaelTurrim.Structure.Terrain.Forest method*), 59
[affect_defense\(\)](#) (*OrodaelTurrim.Structure.Terrain.Hill method*), 60
[affect_defense\(\)](#) (*OrodaelTurrim.Structure.Terrain.Mountain method*), 60
[affect_defense\(\)](#) (*OrodaelTurrim.Structure.Terrain.River method*), 61
[affect_defense\(\)](#) (*OrodaelTurrim.Structure.Terrain.Terrain method*), 62
[affect_defense\(\)](#) (*OrodaelTurrim.Structure.Terrain.Village method*), 62
[affect_max_hit_points\(\)](#) (*OrodaelTurrim.Structure.Terrain.Terrain method*), 62
[affect_range\(\)](#) (*OrodaelTur-*

rim.Structure.Terrain.Terrain method), 62
[affect_sight\(\)](#) (*OrodaelTurrim.Structure.Terrain.Hill method*), 60
[affect_sight\(\)](#) (*OrodaelTurrim.Structure.Terrain.Mountain method*), 60
[affect_sight\(\)](#) (*OrodaelTurrim.Structure.Terrain.Terrain method*), 62
[ArtificialIntelligence \(module\)](#), 65
[ArtificialIntelligence.Filter \(module\)](#), 65

B

[BadActionBaseParameters](#), 58
[BadFactDataValue](#), 58

C

[char\(\)](#) (*OrodaelTurrim.Structure.Terrain.Field method*), 59
[char\(\)](#) (*OrodaelTurrim.Structure.Terrain.Forest method*), 59
[char\(\)](#) (*OrodaelTurrim.Structure.Terrain.Hill method*), 60
[char\(\)](#) (*OrodaelTurrim.Structure.Terrain.Mountain method*), 60
[char\(\)](#) (*OrodaelTurrim.Structure.Terrain.River method*), 61
[char\(\)](#) (*OrodaelTurrim.Structure.Terrain.Terrain method*), 62
[char\(\)](#) (*OrodaelTurrim.Structure.Terrain.Village method*), 62
[compute_accessible_tiles\(\)](#) (*OrodaelTurrim.Business.Proxy.MapProxy method*), 53
[compute_damage\(\)](#) (*OrodaelTurrim.Structure.Terrain.Mountain method*), 60
[compute_damage\(\)](#) (*OrodaelTurrim.Structure.Terrain.Terrain method*), 62
[compute_visible_tiles\(\)](#) (*OrodaelTurrim.Business.Proxy.MapProxy method*), 54

D

`DoubleLinkedList` (class in `OrodaelTurrim.Structure.TypeStrucutre`), 63

E

`empty()` (`OrodaelTurrim.Structure.TypeStrucutre.DoubleLinkedList` method), 63

`execute()` (`OrodaelTurrim.Structure.Actions.Abstract.GameAction` method), 56

`execute()` (`OrodaelTurrim.Structure.Actions.Log.LogAction` method), 56

`ExpertSystem` (module), 65

`ExpertSystem.Business` (module), 65

`ExpertSystem.Business.Parser` (module), 64

`ExpertSystem.Business.Parser.KnowledgeBase` (module), 64

F

`Field` (class in `OrodaelTurrim.Structure.Terrain`), 59

`Forest` (class in `OrodaelTurrim.Structure.Terrain`), 59

G

`GameAction` (class in `OrodaelTurrim.Structure.Actions.Abstract`), 56

`GameControlProxy` (class in `OrodaelTurrim.Business.Proxy`), 52

`GameObjectProxy` (class in `OrodaelTurrim.Business.Proxy`), 52

`GameUncertaintyProxy` (class in `OrodaelTurrim.Business.Proxy`), 53

`get_active_effects()` (`OrodaelTurrim.Business.Proxy.GameObjectProxy` method), 52

`get_attack_effects()` (`OrodaelTurrim.Business.Proxy.GameObjectProxy` method), 52

`get_attribute()` (`OrodaelTurrim.Business.Proxy.GameObjectProxy` method), 52

`get_bases_positions()` (`OrodaelTurrim.Business.Proxy.MapProxy` method), 54

`get_border_tiles()` (`OrodaelTurrim.Business.Proxy.MapProxy` method), 54

`get_current_hit_points()` (`OrodaelTurrim.Business.Proxy.GameObjectProxy` method), 52

`get_current_round()` (`OrodaelTurrim.Business.Proxy.GameObjectProxy` method), 52

`get_income()` (`OrodaelTurrim.Business.Proxy.GameObjectProxy` method), 52

`get_inner_tiles()` (`OrodaelTurrim.Business.Proxy.MapProxy` method), 54

`get_map_height()` (`OrodaelTurrim.Business.Proxy.MapProxy` method), 54

`get_map_width()` (`OrodaelTurrim.Business.Proxy.MapProxy` method), 54

`get_move_cost()` (`OrodaelTurrim.Structure.Terrain.Field` method), 59

`get_move_cost()` (`OrodaelTurrim.Structure.Terrain.Forest` method), 59

`get_move_cost()` (`OrodaelTurrim.Structure.Terrain.Hill` method), 60

`get_move_cost()` (`OrodaelTurrim.Structure.Terrain.Mountain` method), 60

`get_move_cost()` (`OrodaelTurrim.Structure.Terrain.River` method), 61

`get_move_cost()` (`OrodaelTurrim.Structure.Terrain.Terrain` method), 62

`get_move_cost()` (`OrodaelTurrim.Structure.Terrain.Village` method), 62

`get_object_type()` (`OrodaelTurrim.Business.Proxy.GameObjectProxy` method), 52

`get_player_visible_tiles()` (`OrodaelTurrim.Business.Proxy.MapProxy` method), 54

`get_remaining_sigh()` (`OrodaelTurrim.Structure.Terrain.Field` method), 59

`get_remaining_sigh()` (`OrodaelTurrim.Structure.Terrain.Forest` method), 59

`get_remaining_sigh()` (`OrodaelTurrim.Structure.Terrain.Hill` method), 60

`get_remaining_sigh()` (`OrodaelTurrim.Structure.Terrain.Mountain` method), 61

`get_remaining_sigh()` (`OrodaelTurrim.Structure.Terrain.River` method), 61

`get_remaining_sigh()` (`OrodaelTurrim.Structure.Terrain.Terrain` method), 62

`get_remaining_sigh()` (`OrodaelTurrim.Structure.Terrain.Village` method), 62

`get_resistances()` (`OrodaelTurrim.Business.Proxy.GameObjectProxy` method), 53

`get_resources()` (`OrodaelTurrim.Business.Proxy.GameObjectProxy` method), 53

`get_role()` (`OrodaelTurrim`

rim.Business.Proxy.GameObjectProxy
method), 53

get_terrain_type() (*OrodaelTurrim.Business.Proxy.MapProxy* method), 54

get_visible_enemies() (*OrodaelTurrim.Business.Proxy.GameObjectProxy* method), 53

get_visible_tiles() (*OrodaelTurrim.Business.Proxy.GameObjectProxy* method), 53

H

head() (*OrodaelTurrim.Structure.TypeStrucutre.DoubleLinkedList* attribute), 63

Hill (class in *OrodaelTurrim.Structure.Terrain*), 60

I

IllegalActionException, 58

IllegalArgumentException, 58

IllegalConfigState, 58

IllegalHistoryOperation, 58

IllegalLogMessage, 58

IllegalRulesFormat, 58

income (*OrodaelTurrim.Structure.Resources.PlayerResources* attribute), 58

income_increase (*OrodaelTurrim.Structure.Resources.PlayerResources* attribute), 59

increase_income() (*OrodaelTurrim.Structure.Resources.PlayerResources* method), 59

info_text() (*OrodaelTurrim.Structure.Terrain.Field* method), 59

info_text() (*OrodaelTurrim.Structure.Terrain.Forest* method), 60

info_text() (*OrodaelTurrim.Structure.Terrain.Hill* method), 60

info_text() (*OrodaelTurrim.Structure.Terrain.Mountain* method), 61

info_text() (*OrodaelTurrim.Structure.Terrain.River* method), 61

info_text() (*OrodaelTurrim.Structure.Terrain.Terrain* method), 62

info_text() (*OrodaelTurrim.Structure.Terrain.Village* method), 63

is_position_occupied() (*OrodaelTurrim.Business.Proxy.MapProxy* method), 54

is_position_on_map() (*OrodaelTurrim.Business.Proxy.MapProxy* method), 54

L

LogAction (class in *OrodaelTurrim.Structure.Actions.Log*), 56

M

MapProxy (class in *OrodaelTurrim.Business.Proxy*), 53

Mountain (class in *OrodaelTurrim.Structure.Terrain*), 60

N

next (*OrodaelTurrim.Structure.TypeStrucutre.Node* attribute), 63

next() (*OrodaelTurrim.Structure.TypeStrucutre.DoubleLinkedList* method), 63

Node (class in *OrodaelTurrim.Structure.TypeStrucutre*), 63

O

OrodaelTurrim (module), 64

OrodaelTurrim.Business (module), 55

OrodaelTurrim.Business.Interface (module), 51

OrodaelTurrim.Business.Proxy (module), 52

OrodaelTurrim.Presenter (module), 56

OrodaelTurrim.Presenter.Widgets (module), 55

OrodaelTurrim.Structure (module), 64

OrodaelTurrim.Structure.Actions (module), 57

OrodaelTurrim.Structure.Actions.Abstract (module), 56

OrodaelTurrim.Structure.Actions.Log (module), 56

OrodaelTurrim.Structure.Exceptions (module), 58

OrodaelTurrim.Structure.GameObjects (module), 57

OrodaelTurrim.Structure.GameObjects.Prototypes (module), 57

OrodaelTurrim.Structure.Resources (module), 58

OrodaelTurrim.Structure.Terrain (module), 59

OrodaelTurrim.Structure.TypeStrucutre (module), 63

OrodaelTurrimException, 58

P

player_have_base() (*OrodaelTurrim.Business.Proxy.MapProxy* method), 54

PlayerResources (class in *OrodaelTurrim.Structure.Resources*), 58

pointer (OrodaealTurrim.Structure.TypeStrucutre.DoubleLinkedList attribute), 63

previous (OrodaealTurrim.Structure.TypeStrucutre.Node attribute), 63

previous () (OrodaealTurrim.Structure.TypeStrucutre.DoubleLinkedList method), 63

push_back () (OrodaealTurrim.Structure.TypeStrucutre.DoubleLinkedList method), 63

push_front () (OrodaealTurrim.Structure.TypeStrucutre.DoubleLinkedList method), 63

R

remove_resources () (OrodaealTurrim.Structure.Resources.PlayerResources method), 59

resources (OrodaealTurrim.Structure.Resources.PlayerResources attribute), 59

River (class in OrodaealTurrim.Structure.Terrain), 61

S

spawn_information () (OrodaealTurrim.Business.Proxy.GameUncertaintyProxy method), 53

spawn_unit () (OrodaealTurrim.Business.Proxy.GameControlProxy method), 52

T

tail (OrodaealTurrim.Structure.TypeStrucutre.DoubleLinkedList attribute), 63

Terrain (class in OrodaealTurrim.Structure.Terrain), 61

terrain_type (OrodaealTurrim.Structure.Terrain.Field attribute), 59

terrain_type (OrodaealTurrim.Structure.Terrain.Forest attribute), 60

terrain_type (OrodaealTurrim.Structure.Terrain.Hill attribute), 60

terrain_type (OrodaealTurrim.Structure.Terrain.Mountain attribute), 61

terrain_type (OrodaealTurrim.Structure.Terrain.River attribute), 61

terrain_type (OrodaealTurrim.Structure.Terrain.Terrain attribute), 62

terrain_type (OrodaealTurrim.Structure.Terrain.Village attribute), 63

text (OrodaealTurrim.Structure.Actions.Abstract.GameAction attribute), 56

text (OrodaealTurrim.Structure.Actions.Log.LogAction attribute), 56

TIME_STAMP_FORMAT (OrodaealTurrim.Structure.Actions.Abstract.GameAction attribute), 56

U

undo () (OrodaealTurrim.Structure.Actions.Abstract.GameAction method), 56

undo () (OrodaealTurrim.Structure.Actions.Log.LogAction method), 56

User (module), 65

V

value (OrodaealTurrim.Structure.TypeStrucutre.DoubleLinkedList attribute), 63

value (OrodaealTurrim.Structure.TypeStrucutre.Node attribute), 63

Village (class in OrodaealTurrim.Structure.Terrain), 62

X

xml () (OrodaealTurrim.Structure.Actions.Abstract.GameAction method), 56

xml () (OrodaealTurrim.Structure.Actions.Log.LogAction method), 57