

# Walter - an autonomous wAlter - final report

*Mikolaj Aleksandrak*

*Kamil Beker*

*Szymon Kuleczka*

*Michal Dyczkowski*

*June 11, 2019*

## Subprojects

wAlter consists of following subprojects:

- **Walterband** by Kamil Beker - generation of background music, using neural networks and various soundtracks
- **Pizzas recommendations** by Szymon Kuleczka - pizzas recommendation system, based on customer's randomly generated order history using decision tree
- **Genetic furnishing** by Mikolaj Aleksandrak - map arrangement using genetic algorithm

All of the subprojects are contained in the *ai/* directory.

## Documentation

### Root directory

#### **walter.py**

Main file and entry point for an application. It has to be executed in order to run the project.

#### **requirements.txt**

Dependency management file, used to install all required libraries and modules.

#### **pylama.ini**

Pylama linter config file.

#### **README.md**

README file.

#### **.gitignore**

.gitignore file.

**ai/**

**pathfinding.py**

A\* algorithm implementation.

**decision\_\_tree.py**

Implementation of decision tree algorithm.

**genetic\_\_furnishing.py**

Genetic algorithm implementation.

**walterband/**

Home directory for Walterband subproject module.

**constants/**

**bananas.py**

Constants, associated with bananas (tiles with higher step cost - relevant for A\* algorithm) generation.

**colors.py**

Colors RGB/RGBA values.

**conversations.py**

Conversations between customer and waiter.

**datasets.py**

CSV files, loaded into Pandas DataFrames and associated constants.

**desirability.py**

Values used to compute scores of chromosomes in genetic algorithm and thus, how fit is every individual.

**dimensions.py**

All constants concerning dimensions (i.e. grid width, tile width).

**genetics.py**

Values used by genetic algorithm.

**images.py**

Loaded textures, sprites and images.

**maps.py**

Path to currently loaded restaurant map.

**movement.py**

Movement settings and Direction enum (used for pathfinding).

**sounds.py**

Loaded sounds and music.

**entities/****entity.py**

File with Entity class, which is a base class for all characters/agents in the project, contains path dispatching, movement logic and animation logic.

**waiter\_\_agent.py**

File with WaiterAgent class, which inherits from Entity.

**customer.py**

File with Customer class, which inherits from Entity. Contains logic for random order history generation.

**environment/****mapper.py**

Mapper class, extracting data from maps and managing occupation of tiles.

**neighborhood.py**

Neighborhood class computing tiles neighboring provided one. Used by Mapper and Genetic\_furnishing.

**restaurant.py**

Restaurant class, which encapsulates all logic for generating environment data, contains grid array, consisting of objects of class Tile.

**table.py**

Table class, containing table data.

**tile.py**

Tile class, containing tile data, logic and methods.

**gui/****drawable.py**

Contains Drawable class, which is a base class for all objects that are drawn in the application.

**animator.py**

Contains Animator class, which is responsible for animating the entities movement.

**map\_\_renderer.py**

MapRenderer class, which is used to render the map and textboxes (basically everything that is on the screen).

**spritesheet.py**

Contains helper Spritesheet class, used for effective spritesheets slicing.

**lib/**

Contains all libraries, which for some reason couldn't be included in requirements.txt.

**ptext.py**

Library for text handling in Pygame.

**reports/**

Reports directory.

**resources/**

Contains all of the resources used in the project, such as sounds, graphics, datasets in CSV format etc.