**BUDAS extraction and annotator v0.1**

This program takes a floor plan image (in a jpg file) and detect the rooms and doors. The program takes a jpg file as an input – our package contains a sample file “test.jpg” for illustration purpose – and output a list of rooms its detected (represented by a polygon of lines). It also will detect the any words there are associated with the room.

Input: A floor plan image file, in jpg or png

Output: A text file BUDAS\_output.txt contains the following for each room: its coordinates, followed by a “:”, followed by the annotated text for the room, followed by another “:”, followed by the area of the room (if the user entered the area explicitly for the room it will be listed; otherwise it will be calculated based on the user entered scale)

***Installation:***

Installation procedure (for people who know how to work with python)

1. Install python 3.7.4: <https://www.python.org/downloads/release/python-374/> (Select the right operating system)
2. There are a few python packages that need to be installed, they include
3. pillow
4. pytessaract
5. tkinter
6. opencv-python
7. numpy
8. shapely

For MacOS and Linux users, you can use the pip command to install each package. For Windows user, you should download the “wheel” from <https://www.lfd.uci.edu/~gohlke/pythonlibs/#shapely> and use the pip command to install it directly.

(To run pip3, first open a terminal:

For Windows; use Windows Powershell (right click the windows logo, and select the window powershell;

For Mac, go to the Application folder, open utilities and double-click on Terminal;

In either case, a window will open where you can enter command directly.

Then you will enter “pip3 install <package name”>)

1. Install tesseract.

Use the following web site: <https://github.com/tesseract-ocr/tesseract/wiki> to install the software for the right operating system.

1. Run the program using “python3 BUDAS.py”

**Running the program**

To execute the program, open a shell and type “python3 BUDAS.py <your input file>”.

The program also allows some user flags for the extractor. You can type “python3 BUDAS.py –h” to see all the options:

-h : print list of option and exit

-b : batch processing, save detection result and exit (do not run the annotator)

-B : save the image with all the rooms blacked out (file name with "black\_" attached in front)

-W : save the image with all the rooms whited out (file name with "white\_" attached in front)

-t <number> : # of trials for detection algorithm (default 100)

-r <number> : minimum # of rooms to detect (default 100)

-s <directory> : directory to store the result files (this enable more detailed results to be stored)

-i <directory> : save intermediate images in subdirectory (used only for debugging)

If you use the –s function, it will create a directory under the name, and store the results in that directory. The result will contains an image of the rooms that are detected, also with a text file, that list each room in a line, together with the area of the room in pixels. It also shows the time taken to run the program.

**A note on the algorithm**

The algorithm is a heuristic-based method, in which each iteration may return a different plan. The program runs the same method multiple types (set by the –t option) and pick the best one.

Also, right now the program have three algorithms to select the rooms: select the result that generate the most rooms; select the result that generate rooms that cover the most area; or if the user entered a minimum number of rooms to be detected (via the –r option), then it will return the result that is closest to that room. The output files in the directory will have room/area/closest as the result. Notice that this is the result before applying the annotator.

**Note on the Annotator**

The annotator will take the result of the extraction (using the method that pick the most rooms) and present it in a GUI. The user can than choose to update the rooms via the GUI (adding, splitting and merging rooms etc.) and then hit the save button to save the results.