

## Appendix C

# Maintenance Manual

To use the developed system, it is necessary to fulfil the following requirements for the system:

1. PC with installed Fedora 29<sup>1</sup> or Ubuntu 18.04<sup>2</sup>. It may be also later versions too.
2. Installed a modern web browser of your choice. For example Firefox<sup>3</sup>, Chromium<sup>4</sup> or Chrome<sup>5</sup>
3. Access to the Internet.

### C.1 Installation

Open a terminal window and copy and paste the following commands. For Fedora OS follow Listing C.1 and for Ubuntu OS follow Listing C.2.

```
1 sudo dnf install git python3 gcc
2 git clone git@github.com:gandalf15/analogy.git
3 cd analogy
4 make
5 chmod +x analogy.py
```

**Listing C.1:** Terminal commands for Fedora 29

```
1 sudo apt update
2 sudo apt install git python3 gcc
3 git clone git@github.com:gandalf15/analogy.git
4 cd analogy
5 make
6 chmod +x analogy.py
```

**Listing C.2:** Terminal commands for Ubuntu 18.04

At this point the installation is done.

---

<sup>1</sup><https://getfedora.org/>

<sup>2</sup><https://www.ubuntu.com/>

<sup>3</sup><https://www.mozilla.org/en-US/firefox/>

<sup>4</sup><https://www.chromium.org/>

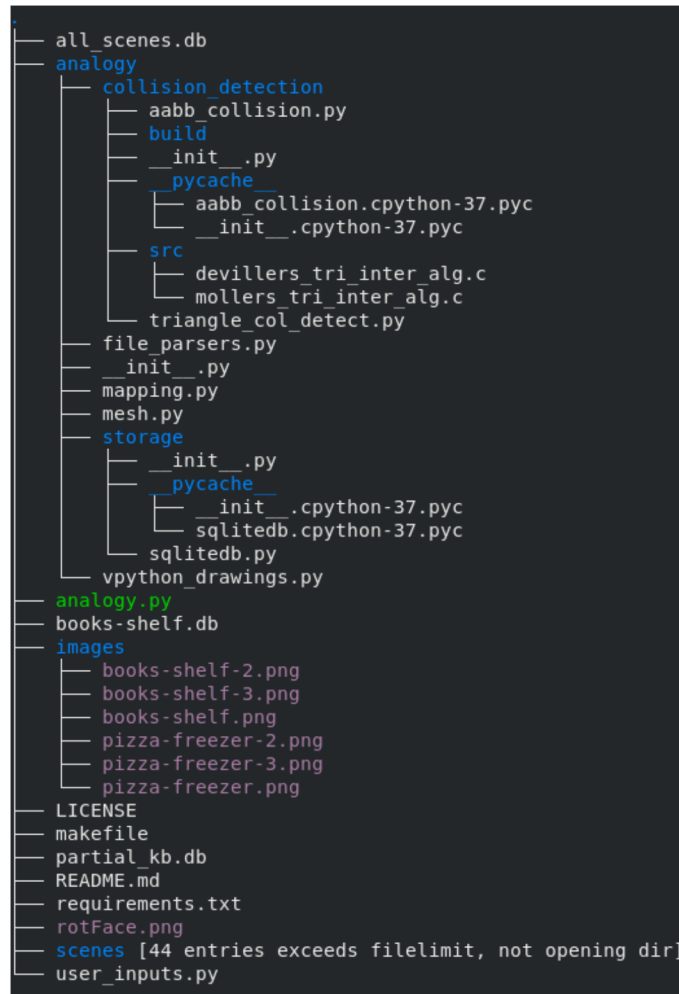
<sup>5</sup><https://www.google.co.uk/chrome/>

## C.2 Software Dependencies

There is only a single software dependency. It is VPython library and it can be downloaded from this URL: <https://vpython.org/>. However, there is no need to do this because I created a make file that automates the whole process.

## C.3 Description Of Files

Figure C.1 shows the directory structure of the project.



**Figure C.1:** File and directory structure of the project.

As you can see the names of the directories and files are self explanatory. Therefore, I am not going to describe every single file but only the most important parts.

- `all_scenes.db` - It contains knowledge about all scenes that are in `scenes/` directory.
- `analogy/` - This directory is the analogy package that can be reused in a different project. It contains all the necessary algorithms and data structures.
- `analogy/collision_detection/` - This directory algorithms for collision detection. This currently includes triangle-triangle collision detection using two different algorithms implemented in C programming language. The file `triangle_col_detect.py` is a wrapper for

these algorithms. The file `aabb_collision.py` contains the axis-aligned bounding box collision check algorithm.

- `analogy/file_parsers.py` - It contains file parsers. Currently supported only OBJ file format.
- `analogy/mapping.py` - It contains the analogy scoring algorithm and mapping algorithm for AABBs.
- `analogy/mesh.py` - It contains Classes for Mesh, Surface, Vertex and AABB.
- `analogy/storage/` - This directory should hold all connectors for storage options. Currently, there is implemented support only for SQLite database.
- `analogy/vpython_drawings.py` - It includes functions that use vpython library for drawing.
- `analogy.py` - It is the main file that puts all together.
- `books-shelf.db` - It contains knowledge only about a one scene and that is the `scenes/books-shelf.obj`.
- `requirements.txt` - List of all the python3 requirements.
- `scenes/` - This directory contains example scenes. This means OBJ files and their `.mtl` files.
- `user_input.py` - It contains functions for getting manipulation points and force vectors from the user.