

Developing the 3D bin packing project

Cipher Kuo

October 2021

Contents

1	Contact	2
2	Prerequisites	2
3	Architecture introduction	3
4	Deploy the frontend	4
5	Deploy the backend	5

1 Contact

shengchikuo@gmail.com
cipherk@supermicro.com.tw

2 Prerequisites

Before getting your hand dirty, you will need to setup your development environment to make sure the system you are using is able to let you follow the instruction guide rapidly without any fine tuning.

This instruction guide will take the Linux(Ubuntu20.04) as an example, and show how to setup the environment under this Linux distribution step by step, furthermore, we have successfully tested this project under the Windows system, but this document will not mention the relevant settings on Windows.

Therefore, it is strongly recommend using the same operating system to prevent you from unknown trouble shooting. If you are currently using Windows operating system, You can still use the following methods without changing the operating system, you can activate WSL2(Windows subsystem linux)to have a Linux like environment and shell, you can open the WSL2 feature by following the [Microsoft official manual.](#), and it is default installed on Windows 11

In case have unknown problems under other operating systems, please raise an issue on Github or write to contact

Download the source code

Before download the source code, make sure there is **git** on your system. following instruction will install the **git** on your Debian like(Ubuntu) system.

```
sudo apt-get update  
sudo apt-get install git -y
```

After download the git, your are able to download the source code to current work directory.

```
git clone https://github.com/N0nentity/3D-Binpacking-GUI
```

Because the font end side of this project employed the /textbfnpm as a javascript library package management tool, you will need to install the nodejs in advance. Following instructions will installed the nodejs, and let you able to use npm.

```
sudo apt-get update
```

```
sudo apt-get install nodejs -y
sudo apt-get install npm
```

Since the back end of this project is written in **Python3**, and utilized a python package Flask(a Web framework) to develop the CGI, to install a package, you will need to install the python, and python package.

```
sudo apt-get update
sudo apt-get install python3 -y
sudo apt-get install python3-pip
```

Afterward, you are able to install flask by following instruction.

```
pip3 install Flask==2.0.2
pip3 install flask_cors==3.0.10
```

comment example

3 Architecture introduction

The entire architecture can be basically separated into three modules.

1. USER INTERFACE
2. ALGORITHM
3. 3D-RENDERING & other interactive functionalities.

In the following paragraph we will use abbreviation 3D-RENDERING for short.

In order to have more user friendly functionality, we combined the USER INTERFACE and 3D-RENDERING together, so that what user need to do is type the necessary input information into the USER INTERFACE and click a send button subsequently, then render image will show up.

hypothetically, let us assume what the usage scenario would look like in case we separate this two modules. In such context, The user will use user input module to formalize the input data and let the algorithm module to generate a json or bit format data output which is definitely not human readable, then the user need to feed this output file to another application(third module) so that the output became visible.

This approach seem like cumbersome but it has its own advantage. Take our project in example, the javascript libraries and css libraries has its own dependencies, and some will conflict with each other, We chose this solution not

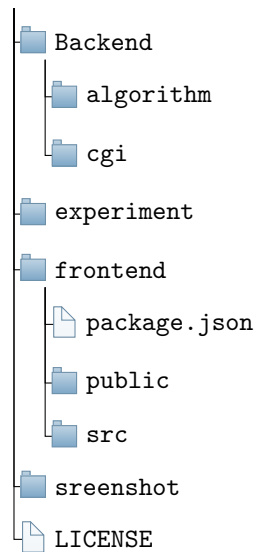
because they are the best option in their respective modules, but because they have the best compatibility when combining modules today.

So instead of three module in this project, what you will see is two module, frontend and backend,

You can take a view of architecture of the project by following instruction.

```
tree -L 2 ./3D-Binpacking-GUI/
```

3D-Binpacking-GUI



In order to keep modern front-back-separate design and under the aforementioned architecture, you can see the algorithm folder under the backend folder, algorithm module will work as a back end service which will call by a resfull API.

4 Deploy the frontend

If you have settled the environment by the previous instructions, you should have npm on your system, installing the dependencies will become quite simple, what you need to do is go to the frontend directory and

```
npm i
```

the dependencies listed in package.json will automatically install for you.

To start a developer server, type the following instruction on the terminal and enter.

```
npm run serve
```

If everything goes well, it will open the port 8080 as default port number and you should see the following output result,

```
App running at:
- Local:  http://localhost:8080/
- Network: http://192.168.88.243:8080/
```

Note that the development build is not optimized.
To create a production build, run `npm run build`.

It is worth to mention that this server is only for development and test, of course you can deploy them with Nginx or other web server like Apache httpd, but this is out of the scope.

In the front-back-separate design, the front end will need to communicate the back end service with specific url and back end port, the default is number 5000, if you want to change the back end listening port, remember to change the setting where the front end data will send to.

The configuration file for the port is under the following path,
`$project_dir/frontend/src/config/index.js`

5 Deploy the backend

Please note that, although the algorithm package `py3dbp` is a open source python package credited by this [repository](#), the same package in this project is modified with lot's of the function interface so that it can fit the need of our own project, therefore, do not install the standard package with the **pip3 install py3dbp**, install the customized package by the following command.

```
cd \"$project_dir/backend/algorithm/
sudo python3 ./setup.py install
```

To startup the backend gunicorn server, there is script name under the backend folder, what you need to do is install the gunicorn server with pip then start up the server with this script.
install gunicorn

```
pip3 install gunicorn
```

run the script to start up backend server, this script will open port number 5000 on your machine. install gunicorn

```
./run_gunicorn.sh
```