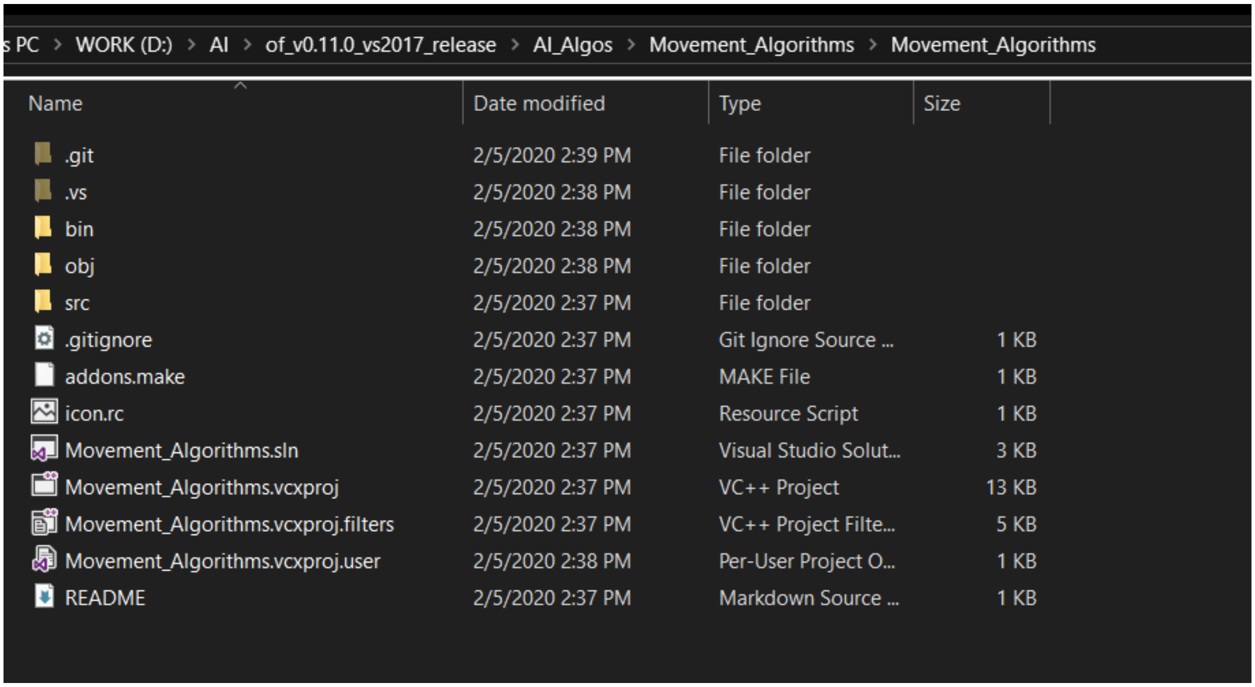
IDE : Visual Studio 2017

I've setup the project at two levels below the root folder of the OpenFrameworks. It is ideal to put in the same location to avoid path mismatches while the project looks for OpenFrameworks library.



s PC > WORK (D:) > AI > of_v0.11.0_vs2017_release > AI_Algos > Movement_Algorithms > Movement_Algorithms

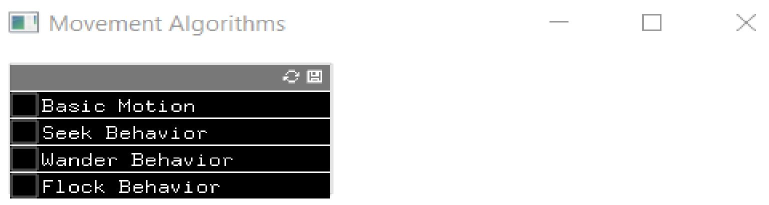
Name	Date modified	Type	Size
.git	2/5/2020 2:39 PM	File folder	
.vs	2/5/2020 2:38 PM	File folder	
bin	2/5/2020 2:38 PM	File folder	
obj	2/5/2020 2:38 PM	File folder	
src	2/5/2020 2:37 PM	File folder	
.gitignore	2/5/2020 2:37 PM	Git Ignore Source ...	1 KB
addons.make	2/5/2020 2:37 PM	MAKE File	1 KB
icon.rc	2/5/2020 2:37 PM	Resource Script	1 KB
Movement_Algorithms.sln	2/5/2020 2:37 PM	Visual Studio Solut...	3 KB
Movement_Algorithms.vcxproj	2/5/2020 2:37 PM	VC++ Project	13 KB
Movement_Algorithms.vcxproj.filters	2/5/2020 2:37 PM	VC++ Project Filte...	5 KB
Movement_Algorithms.vcxproj.user	2/5/2020 2:38 PM	Per-User Project O...	1 KB
README	2/5/2020 2:37 PM	Markdown Source ...	1 KB

This is where my project resides. openFrameWorksLibrary/AI_Algos/Movement_Algorithms/Movement_Algorithms/ "Solution file is here"

How To Use:

Open the Movement_Algos.sln file using Visual Studio 2017 (This is important because using a lower version will result in errors because of the mismatched C++ toolsets and there is no openframeworks plugin for visual studio 19)

Build the solution then run the app. If everything goes well, a window will open up and look like this.



Instead of creating different files for different behaviors, I've created an interactive UI where you can chose the behavior you want to execute.

After choosing a particular behavior (by clicking the button (square box) associated with it), the algorithm for that particular behavior will execute and the UI panel disappears. If you want to run another behavior, you will need to close the existing window and run the app again.

In order to change the underlying parameters for any behavior, you will need to open the OfApp.cpp file and look for the function named "update()". There is a switch statement with all the behaviors I've implemented, you can look for the required behavior and change the underlying parameters.

I've implemented several other behaviors (Look up the Behaviors enum defined in the OfApp.h file to know the exact behaviors), which can also be tested or run by making a single line change to the code. At line 37 of OfApp.cpp file (function is setup()), uncomment the line specified below and pass the enum for the behavior you want to test as the parameter to the function named StartBehavior

```
//StartBehavior(AI::Behaviors::Behaviors::DYNAMIC_WANDER);
```

If you run into any problems, please contact me @ hitheshwar.m@gmail.com

Thanks,

Hitesh Mittapelly