

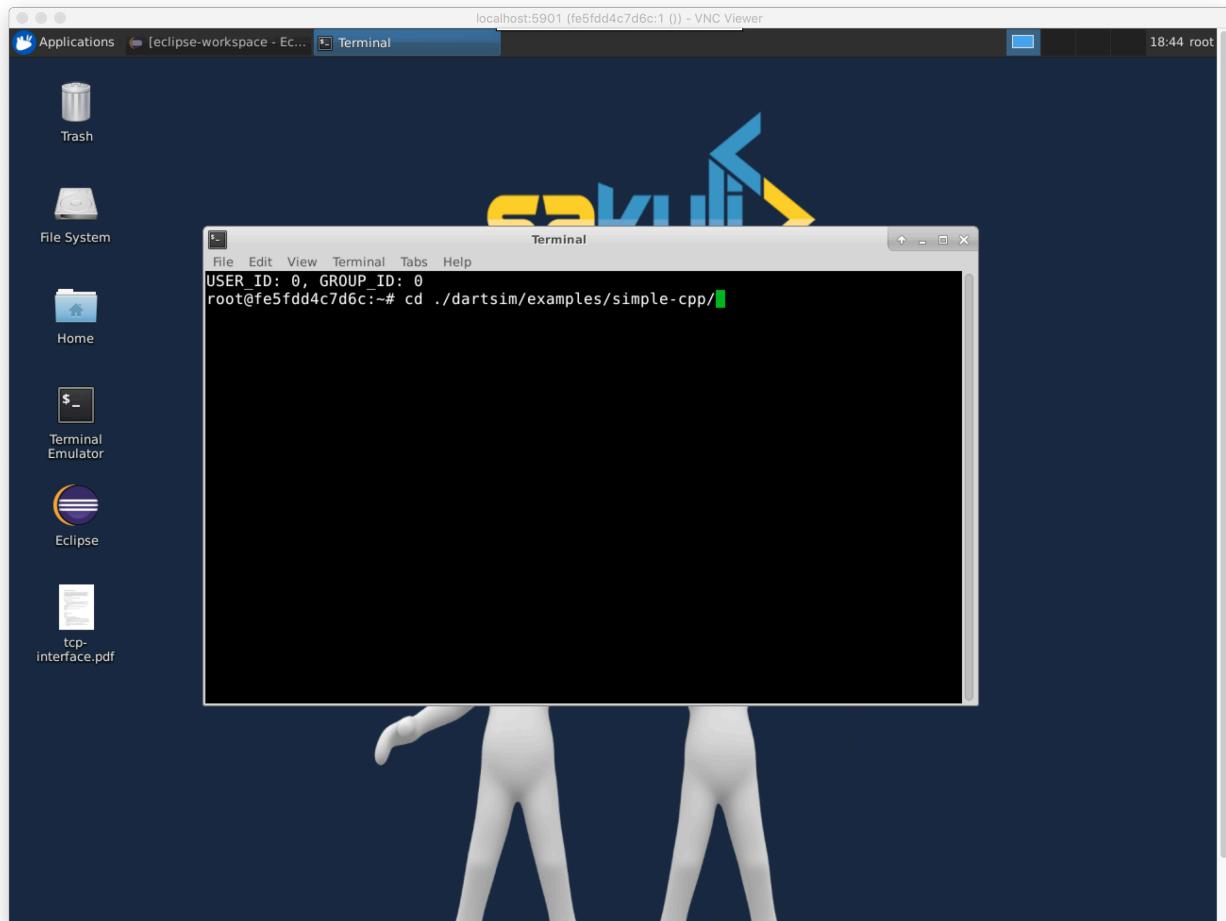
Dartsim Adaption Manager guide

This guide assumes that you have the dartsim docker app and a vnc viewer setup.

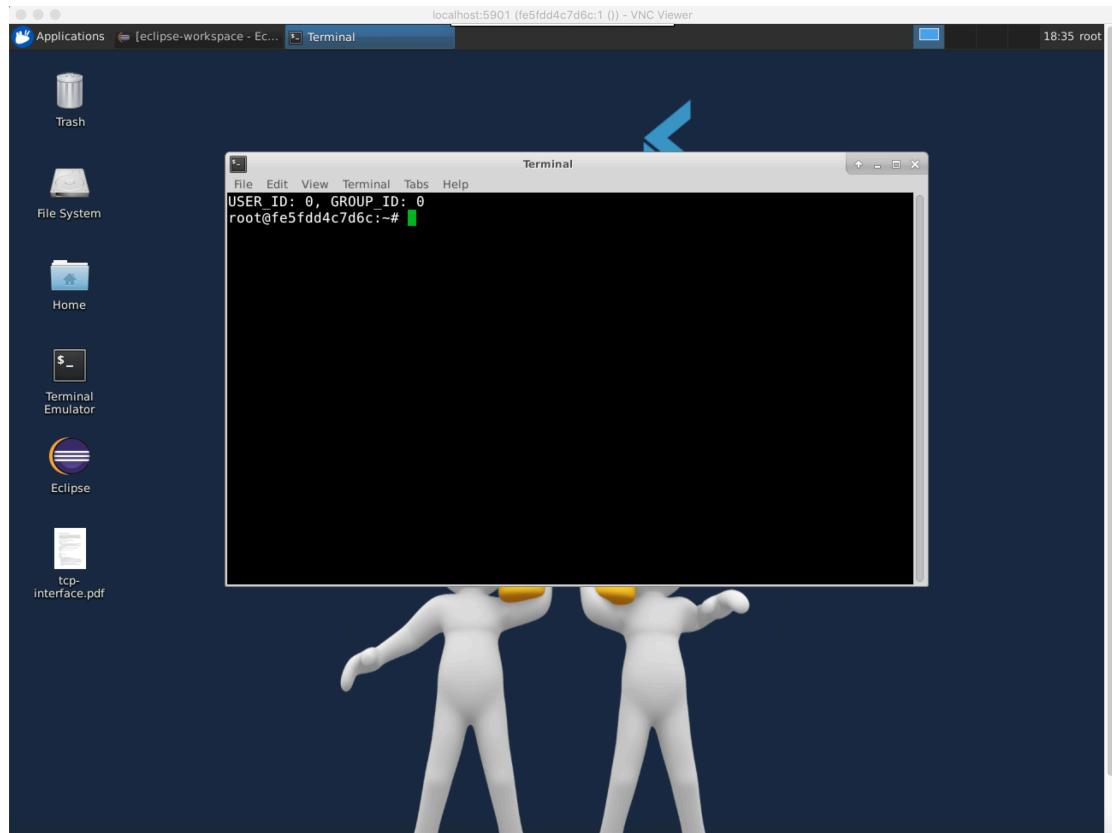
Static latency Guide

These steps show you how to set the static tactic latency of the simulator. Go to the next section to see how to change to code to implement random tactic latency values.

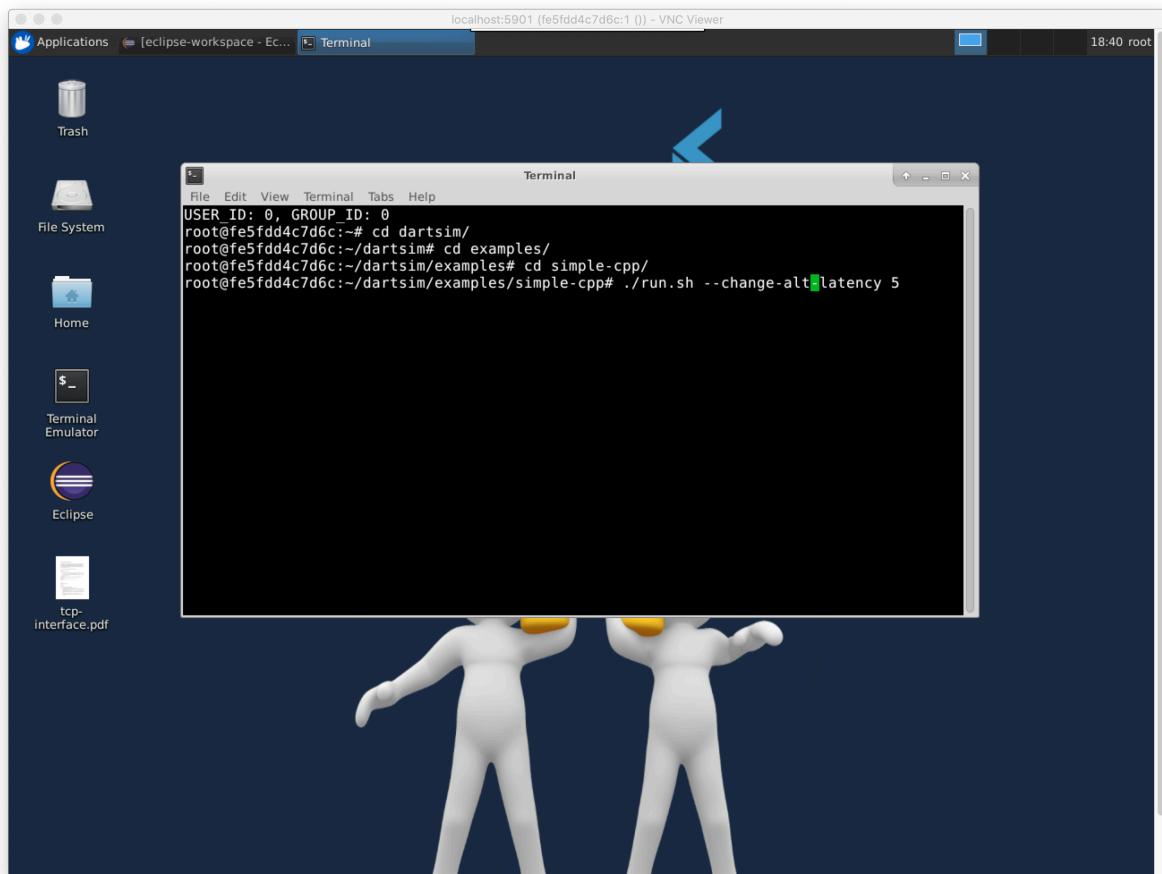
1. Open Terminal



2. Go to the directory /dartsim/examples/simple-cpp/



3. Run the simple-cpp example of the adaption manager. There are several options to run the simulation and configure the adaptation manager. Running `./run.sh --help` will list them. There are two kinds of options supported: the ones for DARTSim, and the ones for the adaptation manager. Every option that precedes the `--` argument is an option for DARTSim, the ones that follow the `--` argument are options for the adaptation manager



Random tactic latency guide

To change the value of the code to set random values. Follow the following steps

1. Open Eclipse (shortcut on the desktop)



2. Go to the file: dartsim/src/dartsimlib/SimulatorImpl.cpp

```

1*****  

2 * DARTSim Mission Simulator  

3 *  

4 * Copyright 2019 Carnegie Mellon University. All Rights Reserved.  

5 * NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING  

6 * INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON  

7 * UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS  

8 * TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE  

9 * OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE  

10 * MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND  

11 * WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.  

12 *  

13 * Released under a BSD (SEI)-style license, please see license.txt or contact  

14 * permission@sei.cmu.edu for full terms.  

15 *  

16 * [DISTRIBUTION STATEMENT A] This material has been approved for public release  

17 * and unlimited distribution. Please see Copyright notice for non-US Government  

18 * use and distribution.  

19 *  

20 * Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by  

21 * Carnegie Mellon University.  

22 *  

23 * This Software includes and/or makes use of Third-Party Software, each subject  

24 * to its own license. See license.txt.  

25 *  

26 * DM19-0045  

27 ******  

28 #include "SimulatorImpl.h"  

29 #include "DeterministicThreat.h"  

30 #include "DeterministicTargetSensor.h"  

31 #include <math.h>  

32 #include <iostream>  

33  

34 using namespace std;  

35  

36 namespace dart {  

37 namespace sim {  

38  

39 SimulatorImpl::SimulatorImpl(const SimulationParams& simParams,  

40 const RealEnvironment& threatEnv, const RealEnvironment& targetEnv,  

41 const Route& route, unsigned missionSuccessTargetThreshold)  

42 : params(simParams),  

43 threatEnv(threatEnv), targetEnv(targetEnv), route(route),  

44 screen(route.size()), vector<char>(simParams.altitudeLevels + 2, ' ')),  

45 currentConfig(simParams.altitudeLevels, TeamConfiguration::Formation::LOOSE, false, 0, 0, 0, 0),  

46 routeIt(this->route.begin()),  

47 changeAltitudeLatencyPeriods(simParams.changeAltitudeLatencyPeriods),  

48 position(*routeIt),  

49 MISSION_SUCCESS_THRESHOLD(missionSuccessTargetThreshold),  

50 SCREEN_THREATS(simParams.altitudeLevels),  

51 SCREEN_TARGETS(simParams.altitudeLevels + 1)

```

3. Go to function SimulatorImpl::executeTactic (line 291)

```

268     currentConfig.altitudeLevel = currentConfig.altitudeLevel - 1;
269   }
270 }
271 auto ttcIncAlt2 = currentConfig.ttcIncAlt2;
272 if (ttcIncAlt2 > 0) {
273   currentConfig.ttcIncAlt2 = -ttcIncAlt2;
274   if (ttcIncAlt2 == 0) {
275     currentConfig.altitudeLevel = currentConfig.altitudeLevel + 2;
276   }
277 }
278
279 auto ttcDecAlt2 = currentConfig.ttcDecAlt2;
280 if (ttcDecAlt2 > 0) {
281   currentConfig.ttcDecAlt2 = -ttcDecAlt2;
282   if (ttcDecAlt2 == 0) {
283     currentConfig.altitudeLevel = currentConfig.altitudeLevel - 2;
284   }
285 }
286
287 return targetDetectedInThisStep;
288 }
289
290 TeamConfiguration SimulatorImpl::executeTactic(string tactic, const TeamConfiguration& config) {
291   auto newConfig = config;
292   if (tactic == INC_ALTITUDE) {
293     if (changeAltitudeLatencyPeriods > 0) {
295       newConfig.ttcIncAlt = changeAltitudeLatencyPeriods;
296     } else {
297       newConfig.altitudeLevel = newConfig.altitudeLevel + 1;
298     }
299   } else if (tactic == DEC_ALTITUDE) {
300     if (changeAltitudeLatencyPeriods > 0) {
301       newConfig.ttcDecAlt = changeAltitudeLatencyPeriods;
302     } else {
303       newConfig.altitudeLevel = newConfig.altitudeLevel - 1;
304     }
305   } else if (tactic == INC_ALTITUDE2) {
306     if (changeAltitudeLatencyPeriods > 0) {
307       newConfig.ttcIncAlt2 = changeAltitudeLatencyPeriods;
308     } else {
309       newConfig.altitudeLevel = newConfig.altitudeLevel + 2;
310     }
311   } else if (tactic == DEC_ALTITUDE2) {
312     if (changeAltitudeLatencyPeriods > 0) {
313       newConfig.ttcDecAlt2 = changeAltitudeLatencyPeriods;
314     } else {
315       newConfig.altitudeLevel = newConfig.altitudeLevel - 2;
316     }
317   } else if (tactic == GO_TIGHT) {
318     newConfig.formation = TeamConfiguration::Formation::TIGHT;

```

4. Change the code as shown in the screenshot. (or like the code script below)

```

283     if (ttcDecAlt2 == 0) {
284         currentConfig.altitudeLevel = currentConfig.altitudeLevel - 2;
285     }
286 }
287
288 return targetDetectedInThisStep;
289 }
290
291=TeamConfiguration SimulatorImpl::executeTactic(string tactic, const TeamConfiguration& config) {
292     auto newConfig = config;
293     cout << "executing tactic " << tactic << endl;
294
295     float randomLatency = rand() % 5;
296 // cout<<"with latency = "<< randomLatency<<endl<<endl;
297
298     if (tactic == INC_ALTITUDE) {
299         if (changeAltitudeLatencyPeriods > 0) {
300             newConfig.ttcIncAlt = randomLatency;
301             cout<<changeAltitudeLatencyPeriods<<endl;
302             cout<< newConfig.ttcIncAlt << endl;
303             cout<<"with latency = "<< randomLatency<<endl<<endl;
304
305         } else {
306             newConfig.altitudeLevel = newConfig.altitudeLevel + 1;
307         }
308     } else if (tactic == DEC_ALTITUDE) {
309         if (changeAltitudeLatencyPeriods > 0) {
310             newConfig.ttcDecAlt = randomLatency;
311             cout<<changeAltitudeLatencyPeriods<<endl;
312             cout<< newConfig.ttcDecAlt << endl;
313             cout<<"with latency = "<< randomLatency<<endl<<endl;
314
315         } else {
316             newConfig.altitudeLevel = newConfig.altitudeLevel - 1;
317         }
318     } else if (tactic == INC_ALTITUDE2) {
319         if (changeAltitudeLatencyPeriods > 0) {
320             newConfig.ttcIncAlt2 = randomLatency;
321             cout<<changeAltitudeLatencyPeriods<<endl;
322             cout<< newConfig.ttcIncAlt2 << endl;
323             cout<<"with latency = "<< randomLatency<<endl<<endl;
324
325         } else {
326             newConfig.altitudeLevel = newConfig.altitudeLevel + 2;
327         }
328     } else if (tactic == DEC_ALTITUDE2) {
329         if (changeAltitudeLatencyPeriods > 0) {
330             newConfig.ttcDecAlt2 = randomLatency;
331             cout<<changeAltitudeLatencyPeriods<<endl;
332             cout<< newConfig.ttcDecAlt2 << endl;
333             cout<<"with latency = "<< randomLatency<<endl<<endl;
334
335         } else {
336             newConfig.altitudeLevel = newConfig.altitudeLevel - 2;
337         }
338     }
339 }
```

```

TeamConfiguration SimulatorImpl::executeTactic(string tactic, const
TeamConfiguration& config) {
    auto newConfig = config;

    float randomLatency = rand() % 5;

    if (tactic == INC_ALTITUDE) {
        if (changeAltitudeLatencyPeriods > 0) {
            newConfig.ttcIncAlt = randomLatency;
            cout<<changeAltitudeLatencyPeriods<<endl;
            cout<< newConfig.ttcIncAlt << endl;
            cout<<"with latency = "<< randomLatency<<endl<<endl;

        } else {
            newConfig.altitudeLevel = newConfig.altitudeLevel + 1;
        }
    } else if (tactic == DEC_ALTITUDE) {
        if (changeAltitudeLatencyPeriods > 0) {
            newConfig.ttcDecAlt = randomLatency;
            cout<<changeAltitudeLatencyPeriods<<endl;
            cout<< newConfig.ttcDecAlt << endl;
            cout<<"with latency = "<< randomLatency<<endl<<endl;

        } else {
            newConfig.altitudeLevel = newConfig.altitudeLevel - 1;
        }
    } else if (tactic == INC_ALTITUDE2) {
        if (changeAltitudeLatencyPeriods > 0) {
            newConfig.ttcIncAlt2 = randomLatency;
            cout<<changeAltitudeLatencyPeriods<<endl;
            cout<< newConfig.ttcIncAlt2 << endl;
            cout<<"with latency = "<< randomLatency<<endl<<endl;

        } else {
            newConfig.altitudeLevel = newConfig.altitudeLevel + 2;
        }
    }
}
```

```

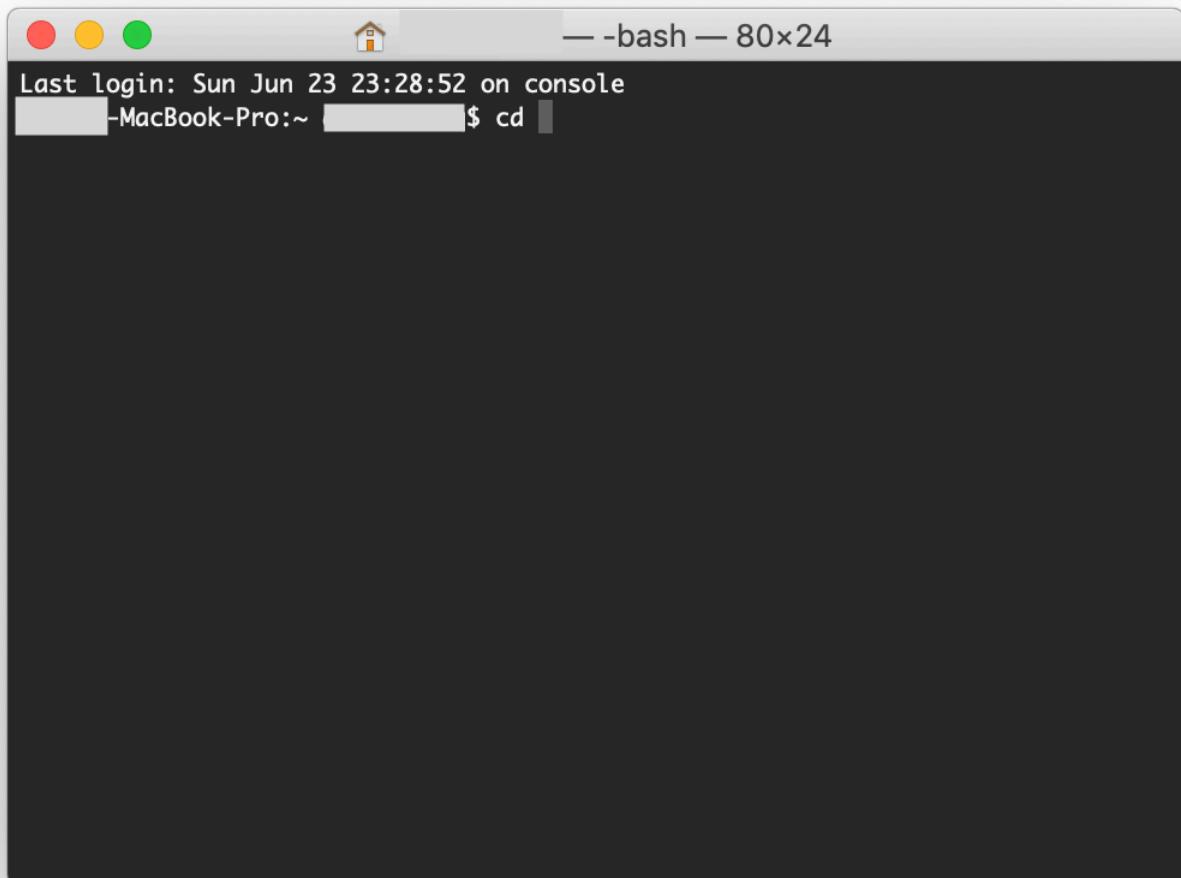
} else if (tactic == DEC_ALTITUDE2) {
    if (changeAltitudeLatencyPeriods > 0) {
        newConfig.ttcDecAlt2 = randomLatency;
        cout<<changeAltitudeLatencyPeriods<<endl;
        cout<< newConfig.ttcDecAlt2 << endl;
        cout<<"with latency = "<< randomLatency<<endl<<endl;
    } else {
        newConfig.altitudeLevel = newConfig.altitudeLevel - 2;
    }
} else if (tactic == GO_TIGHT) {
    newConfig.formation = TeamConfiguration::Formation::TIGHT;
} else if (tactic == GO_LOOSE) {
    newConfig.formation = TeamConfiguration::Formation::LOOSE;
} else if (tactic == ECM_ON) {
    newConfig.ecm = true;
} else if (tactic == ECM_OFF) {
    newConfig.ecm = false;
} else {
    throw std::runtime_error(string("unknown tactic ") + tactic);
}
return newConfig;
}

```

Running Dartsim locally

To run the darstsim locally instead of running on the docker container (This guide assumes you are using a Unix machine e.g. Ubuntu or MacOs). Follow the following steps:

1. Open the terminal.



2. Go to the directory you wish to copy your dartsim application.
3. Run the following command: `sudo docker cp dartsim:/headless/dartsim/ ./`

```
Last login: Sun Jun 23 23:28:52 on console
-MacBook-Pro:~ [REDACTED] $ cd Downloads
-MacBook-Pro:Downloads [REDACTED] $ mkdir [REDACTED] Dartsim; cd [REDACTED] Dartsim
-MacBook-Pro:[REDACTED] Dartsim [REDACTED] $ sudo docker cp dartsim:/headless/dartsim/ ./
```

Now that you have a copy of the dartsim app folder on your local machine. But hold on! There is still more to do! The reason this application is packaged in the docker container is to relieve you of the responsibility of setting up the appropriate environment for the application. We need to install multiple libraries for the app to work properly. If you are using an Ubuntu machine (or similar), just run the following command: `sudo apt-get install libboost-all-dev libyaml-cpp-dev make automake autoconf g++ default-jdk ant wget libtool`. If you are using a Mac (or another Unix machine without apt), use a package manager (e.g. Homebrew) to get these libraries easily.

Note: The appropriate libraries and their homebrew names are provided below in the table.

Library	Name
Boost	boost
Yaml-Cpp	yaml-cpp
Make	make
Automake	automake
G++	(Install through Xcode)
JDK (required if you want to run the TCP version of the adaption manager)	(Go to the oracle website to install JDK)
Ant	ant
Wget	wget
Libtool	libtool

Once these libraries are installed, the dartsim code can be run same way as in the docker container. If you have made any changes to the code, it will need to be recompiled. Go to the readme in the dartsimlib folder (*dartsim/src/dartsimlib/*) or the simple-cpp example folder (*dartsim/examples/simple-cpp/*) for the appropriate commands to build the dartsim or the adaption manager. Note: delete the “Build” folder in these directories to compile without any issue. Some versions of *make* have trouble compiling it properly if the build folder is present.