**Team Members**

Qi Zeng UFID: 8363-2136 Chen Cao UFID: 8154-4258

**What is working**

There are three scala files: *example\_bonus.scala,* and *ActorLogging\_bonus.scala.*

*ActorLogging\_bonus.scala* is a class to provide logging function

*example\_bonus.scala* is a fixed version of project3 to generate logs.

COMPLIE and RUN: In the file containing *example.scala* and *actorlog.scala*,

scalac \*.scala // compile

scala example\_bonus //logging example

The log files will be generated in a logfile\_bonus folder with each actor one log.

**How you integrated logging with the actors**

we use trait in scala to implement the actor with logging function as a abstract class ActorLogging. So if we want to add log to ordinary scala programs with actors, we can add this function by inherit the class ( use *import acl.ActorLogging*).

Besides, we have to slightly modify the actor definition sentence in source code

**original:**

*class actor\_name( variable) extends Actor*

**actor with logging function :**

*class actor\_name( variable) extends ActorLogging*

It's a elegant way to deal with logging because it would not litter the codes and it's quite recoverable. I believe no one can find the difference between the them without special attention and comparison.

**About the ActorLogging class**

it has three function: *react, logFile* and *getTime*.

1. Override react is the most important one and it's designed to replace the original react function in Actor to deal with message, so we can automatically log whenever we receive a message. In other words, it's related to when to log.
2. logFile function adapt the file write function to deal with log format: it is related to how to log and what to log.3
3. getTime can provide the system time and we can write it in log file. As we all know, time is critical for a log.

**What log files are produced**

The log files are contained in the ./logfile\_bonus folder, with each actor one file.

The log file format is like the following

[INFO] type: log type, name: msg receiver, number: item id

[TIME] yyyy-mm-dd hh:mm:ss

[MESSAGE] msg type

[FROM] msg sender

**Example:**

“ [INFO] type: Got, name: Actor\_example\_bonus$pastryNode@229ec9cd, number: 2

[TIME] 2012-11-20 11:22:33

[MESSAGE] join(c81e728d,0,2)

[FROM] [Actor\_example\_bonus$pastryNode@66d2c37c](mailto:Actor_example_bonus$pastryNode@66d2c37c) "

The format is human readable. [INFO] contains the basic information of log type, message receiver actor and log sequence number. [TIME] is the machine time when log generates. [MESSAGE] is specified the format as “msg type” [FROM]can tell us the “msg sender actor” to better trace the message source.