Lab 2 – Game of Tag

Rafat Khan <rafatk@kth.se>

March 11, 2022

1 Introduction

This assignment is implementation of the game of Tag for mobile agents. The distributed system with basic functionalities provided for this assignment consists of several rooms (Bailiff) and some players (Dexter).

The game of tag is a child's game, played minimally by two players. It is very simple. One player is choosen to be 'it'. That player's task is to run up to one of the other players and touch him or her, saying "You're it!", whereupon the 'it' property and the associated task is transferred to the other player. The players who are currently not 'it', try to evade being tagged, usually by running and hiding.

2 The Assignment

2.1 Player behaviours

A player can be 'it' or 'not it'. When it is in the 'it' state, it hunts down victims and tries to tag them. As this is a simple heuristic to implement, a successful predator will probably look for the Bailiff with the most players.

On the other hand, an untagged player has more freedom in how they behave. Naturally, it needs to watch out for the player who is 'it', and attempt to escape if it ends up in the current Bailiff. The Bailiff has the option of staying put, or moving over to another safe Bailiff, if not immediately threatened. Depending on whether it chooses to hop on, it can choose from a random Bailiff, a Bailiff with few players, or a Bailiff with many players.

2.2 Implementation

2.2.1 Discovering execution environments (Bailiffs)

The players finds the bailiffs using the jini lookup servers. In my implementation, each player enter a loop in which it randomly pick one bailiff, pings it to see if it is alive, and migrate to it, or try another one.

```
svcItems = SDM.lookup (bailiffTemplate, maxResults, null);
```

2.2.2 Query properties of the execution environment

Players query for the properties.

```
  public String getProperty (String key)
{
  log.fine(String.format("getProperty key=%s", key));
  return propertyMap.get (key.toLowerCase ());
}
```

2.2.3 Negotiate for status (being 'it' or not)

Here, I have used a command '-t' to tag an agent as it or not.

```
//define command
String[] msg = {
    "-t      Set the current Dexter as IT"
};
if (av.equals("-t"))
    dx.setTagged(true);
//define function
private boolean isTagged = false;
public void setTagged(boolean isTagged_) {isTagged =
isTagged_;}
public boolean isTagged() {return isTagged;}
```

2.2.4 Move to other Bailiff

A player can jump from a bailiff to another. In my settings, the players will keep jumping from one bailiff to another. If a player is 'IT' and it can't find another player in the bailiff, it will also keep jumping to another bailiff.

```
public void migrate (Object obj, String cb, Object [] args)
throws
  java.rmi.RemoteException, NoSuchMethodException
{
  log.fine(String.format("migrate obj=%s cb=%s args=%s",
                         obj.toString(),
                         cb,
                         Arrays.toString(args)));
  Agitator agt = new Agitator (obj, cb, args);
  System.out.println("Show the number of clients " + NumberOfClients);
  agt.initialize ();
  agt.start ();
}
```

2.2.5 Tag another player

Tag another player to make that 'IT'. This function is being called when an 'IT' player finds another player in the same bailiff.

```
public void setTagged_(Dexter dex, Dexter dex1)
{
        List<Dexter> ld = listDexters;
        if(ld.removeIf(s -> s.getUUID().equals(dex1.getUUID())))
        {
          Dexter tmp = dex1;
          tmp.setTagged(true);
      ld.add(tmp);
          if(ld.removeIf(s ->
          s.getUUID().equals(dex.getUUID()))) {
            dex.setTagged(false);
            ld.add(dex);
          }
          setDexter(ld);
        }
```

2.3 Interface

```
public String getProperty (String key)
  throws
    java.rmi.RemoteException;
public void migrate (Object obj, String cb, Object [] args)
  throws
    java.rmi.RemoteException,
    java.lang.NoSuchMethodException;

public void setTagged_(Dexter dex, Dexter dex1)
  throws
    java.rmi.RemoteException;
```

2.4 Result

The system is running without any issues. The players are able to find the bailiffs and can jump from one another. The tah command is working and the tagged player is behaving as expected.

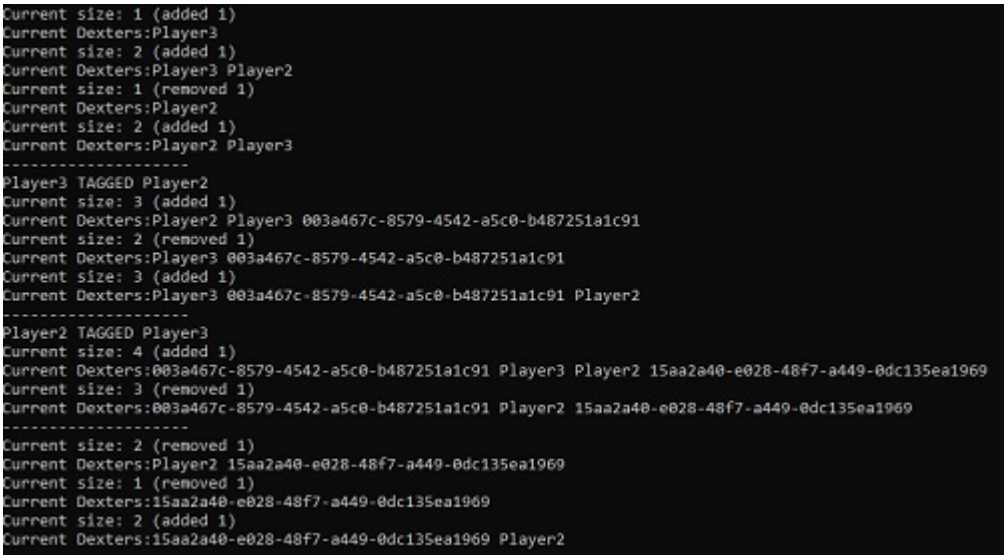

Figure 1: Player.

2.5 Conclusion

Overall, this assignment was challenging for me. I had fun learning how to use the codebase server and had a deeper understanding of distributed systems while solving the tasks.