ID2010VT201 – Programming of Interactive Systems

Lab 2 – The Tag Assignment

Carried out by:

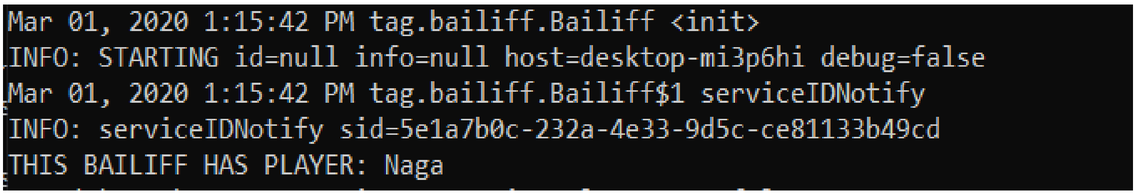
Nagasudeep Vemula- vemula@kth.se

Akhil Yerrapragada - akhily@kth.se

1/03/2020

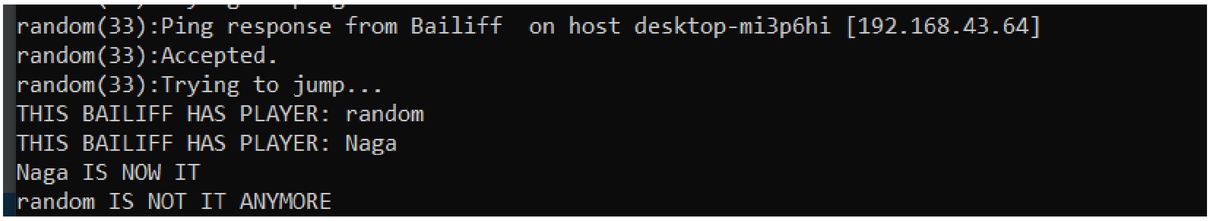
We have carried out the following mandatory tasks from the provided question file :-

I. Enabling the game of tag starting from provided software and designing and implementing the same for 3 mobile agents.



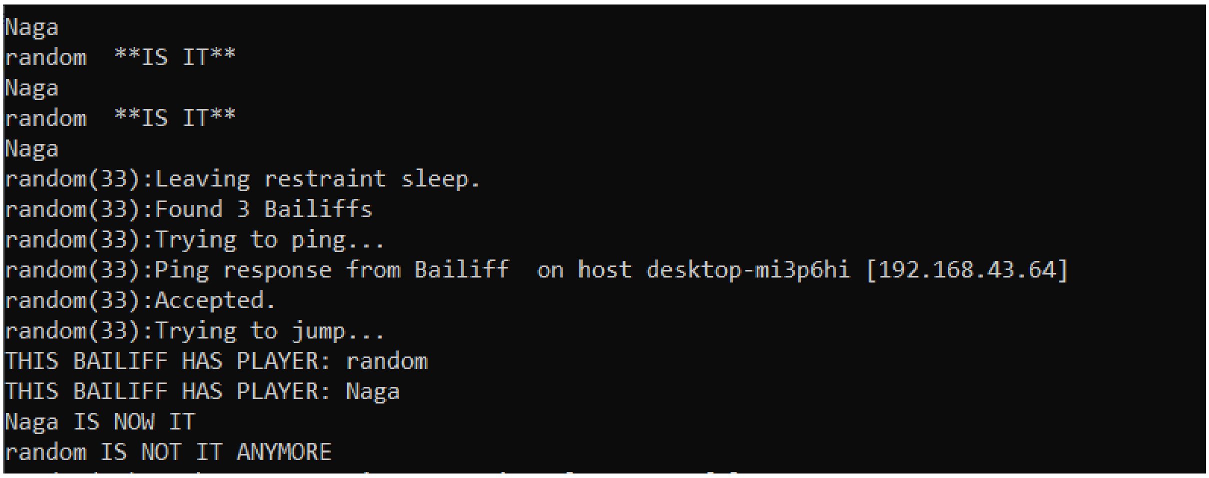
```
Mar 01, 2020 1:15:42 PM tag.bailiff.Bailiff <init>
INFO: STARTING id=null info=null host=desktop-mi3p6hi debug=false
Mar 01, 2020 1:15:42 PM tag.bailiff.Bailiff$1 serviceIDNotify
INFO: serviceIDNotify sid=5e1a7b0c-232a-4e33-9d5c-ce81133b49cd
THIS BAILIFF HAS PLAYER: Naga
```

II. Tagging can be done only between players present in the same bailiff.



```
random(33):Ping response from Bailiff  on host desktop-mi3p6hi [192.168.43.64]
random(33):Accepted.
random(33):Trying to jump...
THIS BAILIFF HAS PLAYER: random
THIS BAILIFF HAS PLAYER: Naga
Naga IS NOW IT
random IS NOT IT ANYMORE
```

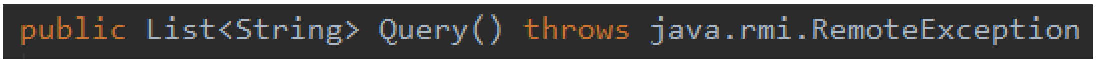
III. The tag(the 'it' property) must be passed reliably from one player to another. It must not be lost or duplicated during the transaction.



```
Naga
random  **IS IT**
Naga
random  **IS IT**
Naga
random(33):Leaving restraint sleep.
random(33):Found 3 Bailiffs
random(33):Trying to ping...
random(33):Ping response from Bailiff  on host desktop-mi3p6hi [192.168.43.64]
random(33):Accepted.
random(33):Trying to jump...
THIS BAILIFF HAS PLAYER: random
THIS BAILIFF HAS PLAYER: Naga
Naga IS NOW IT
random IS NOT IT ANYMORE
```

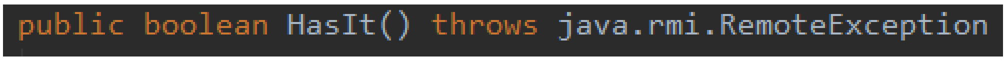
Design Decisions

1. The player needs cooperation from the bailiff so that it can get list of players currently located in a bailiff. We have used the following method to enable this:-



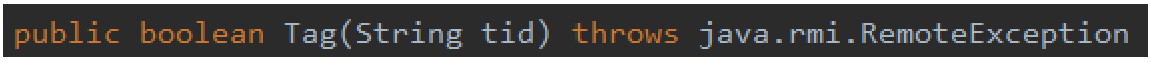
```
public List<String> Query() throws java.rmi.RemoteException
```

2. Players need to query other players to check if they are 'it' or not. Method HasIt() has been implemented to parse through the player map and return true on finding a player that has 'it'.



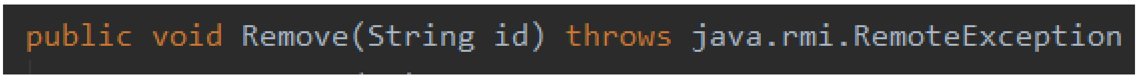
```
public boolean HasIt() throws java.rmi.RemoteException
```

3. Player that has 'it' should be able to tag other players. We have implemented the function Tag with the target id being passed as parameter.



```
public boolean Tag(String tid) throws java.rmi.RemoteException
```

4. Remove the players from the current bailiff on them jumping to another bailiff, this is implemented using the remove method by passing the id as parameter.

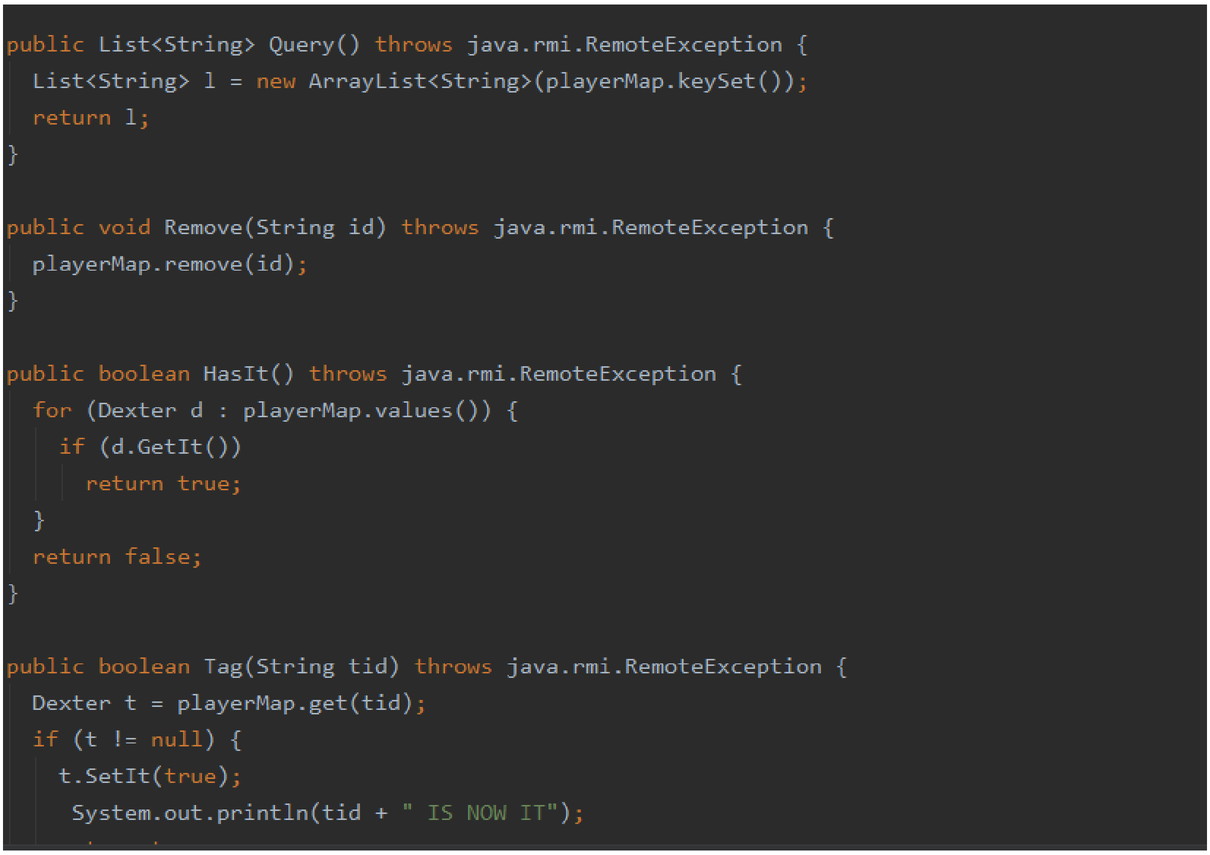


```
public void Remove(String id) throws java.rmi.RemoteException
```

5. Each player must have a unique id so that it can recognise itself in a list of players or be able to specify some other player.

Implementation Strategy

Interface:



```
public List<String> Query() throws java.rmi.RemoteException {
  List<String> l = new ArrayList<String>(playerMap.keySet());
  return l;
}

public void Remove(String id) throws java.rmi.RemoteException {
  playerMap.remove(id);
}

public boolean HasIt() throws java.rmi.RemoteException {
  for (Dexter d : playerMap.values()) {
    if (d.GetIt())
      return true;
  }
  return false;
}

public boolean Tag(String tid) throws java.rmi.RemoteException {
  Dexter t = playerMap.get(tid);
  if (t != null) {
    t.SetIt(true);
     System.out.println(tid + " IS NOW IT");
```

1. Query()-Enables player to get list of players currently present in the bailiff. Accesses the player map in the form of a list and it returns the same.

2. Remove(String id)- For the removal of a player from a bailiff when it jumps to the next bailiff. Takes the id as parameter for referencing.

3. HasIt()- To check if a player has the ‘It’ property set to true or not. The dexter parses through the player map values to find the player with the property and returns true on finding.

4. Tag(String tid)- To enable the game of tag to be played,searches for players on the map and does a check to see if the player is tagged or not. Transfers the tag property now to the chosen player with a display message for the same.