Lab 3

Goutham Devarayasamudram and Siddharth Udayappan Chidambaram

Introduction

As before, our world is again a grid with each position occupied by a Pig, a stone column or left empty. We have 3 programs, MainGame,Pig and database. MainGame is in-charge of the overall game mechanics. The Pig program is a Python class for representing a Pig in our system. The database class implements our database server. Our grid is again a 1-Dimensional world.

Setup

Initially, the user is requested for the number of Pigs. The main Game then proceeds to create that many Pigs and assigns random ports to them. The Maingame also creates an object of database class and gives it a random port to listen on. Our co-ordinator Pigs will communicate to the database using this port.

The Maingame then chooses 2 random Pigs for co-ordinators. To each of these co-ordinators, it assigns roughly half of the remaining pigs and this assignment is mutually exclusive (and exhaustive in that each Pig in the system is assigned to one or the other leader). Each of these co-ordinators notify the Pigs they are incharge of that they are the co-ordinators. The Pigs will note down who their co-ordinator is (this is the co-ordinator they will send their status to). All communication in our system is done using network calls which have some random delay.

Bird Approaching

The MainGame generates random values for bird landing time and position. It then gives these values to the 2 co-ordinators. The Maingame will then wait until the 2 co-ordinators return with the score (All communication with the co-ordinator from the miangame is threaded).

After receiving the bird information from the Main Game, Each of the co-ordinators will see which of the Pigs they are incharge of is getting affected by the Bird. How the leader will select the affected Pigs is given below (similar to how it was in the previous lab).They will then proceed to warn these Pigs to change their position. Note that if one of the co-ordinators is asleep, then the other co-ordinator will take up the task of warning all the Pigs. Once they have sent the warning, the co-ordinators will wait until they have received the status(saying whether the pig survived the bird attack or not) from all Pigs they are incharge of.

Bird Landing

At the affected Pigs' end, when the bird approaching message is received, the affected Pig checks if it has already received the bird landing message before this. If it hasn't, it proceeds to take evasive action by changing its position. When the bird landing message is received, it checks if the bird approaching message has already been received. If it has received the warning, then the Pig has already taken evasive action. The Pig then sends a message to its respective co-ordinator saying its not been hit. If the bird landing message was received first, then the Pig has been hit and it notifes the co-ordinator that its been hit.

When the co-ordinators receive status from the Pigs,they will update the pig status in the database. Once the co-ordinators have received scores from all the Pigs, the control is returned to the Main Game which then displays the score for that game. The score for that iteration is just the sum of the scores from the 2 Co-ordinators. We also display the cumulative score which is the total number of Pigs hit across all iterations. This completes one iteration of our game.

Fault Tolerance

At the end of each game iteration, we randomly select one of the co-ordinators to fall asleep. We ensure that only one of the co-ordinators can be asleep at a given time (i.e. there is 1 co-ordinator who is always up). At the start of each iteration, we let the 2 co-ordinators send messages to each other to check if the other one is awake. If one of the co-ordinators is asleep, then the other co-ordinator(which is awake) proceeds to assume responsibility for all the Pigs. It first queries the database about the existing status of all the Pigs. Once it has this information, it then notifies all the Pigs that it is their co-ordinator now. The rest is as before. This co-ordinator will be responsible for giving the score to the Main Game and updating the database for all the Pigs.

Its also possible for the co-ordinator asleep to wake up with some probability. Once the co-ordinator(lets call it Oink) is awake, it tells the other co-ordinator that it is up. The other co-ordinator (say Doink) will then select a random group of Pigs from the list it is incharge of and hands over their responsibility to Oink. Oink will notifies these Pigs that is their co-ordinator now. Doink will only give the Pig IDs to Oink. Oink will then query the database to obtain more information about the Pigs it is incharge of (specifically their position).

Note that the probability for a Co-ordinator falling asleep (assuming no other Co-ordinator is sleeping) and for a Co-ordinator waking up is $\frac{1}{2}$ in both cases.

Database and Consistency

The database server is incharge of updating the Pig status in the town register (which is a file called townrecord.txt). At the end of each iteration, both the co-ordinators(or 1 of them if the other is asleep) will send the Pig status for that iteration to the database. The database will then proceed to update this information in the town register. We ensure that both the co-ordinators cannot update the file at the same time by using locks. A co-ordinator keeps waiting until it has access to the lock and can update the file. Cache consistency is enforced since the co-ordinators push things onto the database immediately after the Pig statuses are received. So, the database always has the most updated status for all Pigs at the end of every iteration.

Message Passing

Like last time, we introduce random delays in our network calls. This delay is (in seconds) 0.1 multiplied by some random number between 1 and the number of Pigs in our system. The sending Pig sleeps for the duration of the random delay and then sends the message. This part is threaded and hence doesn't block the sending Pig.

We use 11 message IDs for communication. Their details are given in the table in the next page.

Assumptions

Given below are the assumptions we made:

- Only one Co-ordinator can be asleep at a time
- If a Co-ordinator is asleep, then it is effectively out of the game till it wakes up. It wont be affected by the bird for the duration of its sleep.
- The Co-ordinators chosen at the start of the game are never changed. As in, we have the same 2 co-ordinators for the entire game (ofcourse its possible one of them might goto sleep).

Table 1: Message IDs

Message IDs	Name	Description
1	Bird Approaching	Sent by the leader Pig to the affected Pigs telling them they might get hit by the bird
2	Co-ordinator Message	Sent by the Co-ordinator to the Pigs it is incharge of telling them that it is their Co-ordinator. On receiving this message, the Pigs will store the Co-ordinator ID.
3	Bird Landing Message	Received by the affected Pigs telling them the bird has landed at their location. The Pigs will then send their status to the leader
4	Status Message	Received by the leader from the affected Pigs telling their status after the Bird landed
5	Pig Acknowledgement	Received by the Co-ordinator from the Pigs it is incharge of. This is an acknowledgement message saying that the Pigs accept the other Pig as the Co-ordinator.
6	Co-ordinator status	Sent by a Co-ordinator to another checking if it is up.
7	Co-ordinator Acknowledgement	Sent by one Co-ordinator to another saying that its awake. If a Co-ordinator doesn't receive this message within some time, it will assume the other Co-ordinator is asleep.
8	Database Info	Received from the Database process by a Co-ordinator giving the latest Pig information.
9	Co-ordinator Awake	Received by one Co-ordinator from the other Co-ordinator saying the other Co-ordinator is awake. The receiving Co-ordinator will then assign some Pigs to the now-awake Co-ordinator and send it as a message.
10	Co-ordinator Reply	Received by the now-awake Co-ordinator from another giving the list of Pigs it is incharge of
11	Database Acknowledgement	Sent by the Database process telling the co-ordinator that its done with updating the town register.