Programming Assignment 1: Angry Birds: Part Deux (Smarter Pigs Edition)

TESTCASES DOCUMENT

Submitted by: - Rahul Raj and Olenka Dey

We inspected our program for different conditions few of the conditions are:

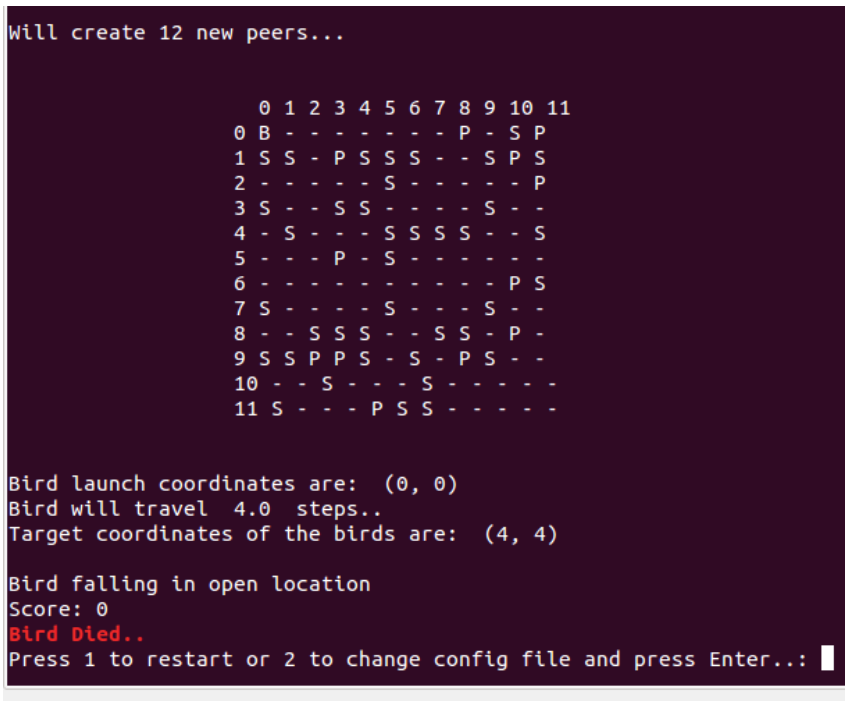
1. When the bird goes, and lands out of the grid space.
2. Changing the network architecture of how peers are connected.
3. Analyzing performance when the hop delay changes.
4. Proper detection of status and score when bird lands on:
    - Empty space
    - On a stone - how it ripples out to its neighbors
    - On a pig - how it relocates to the available location in its vicinity, how it signals other to take shelter and how it informs others if it dies or evades.

5. By increasing number of peers in the network.

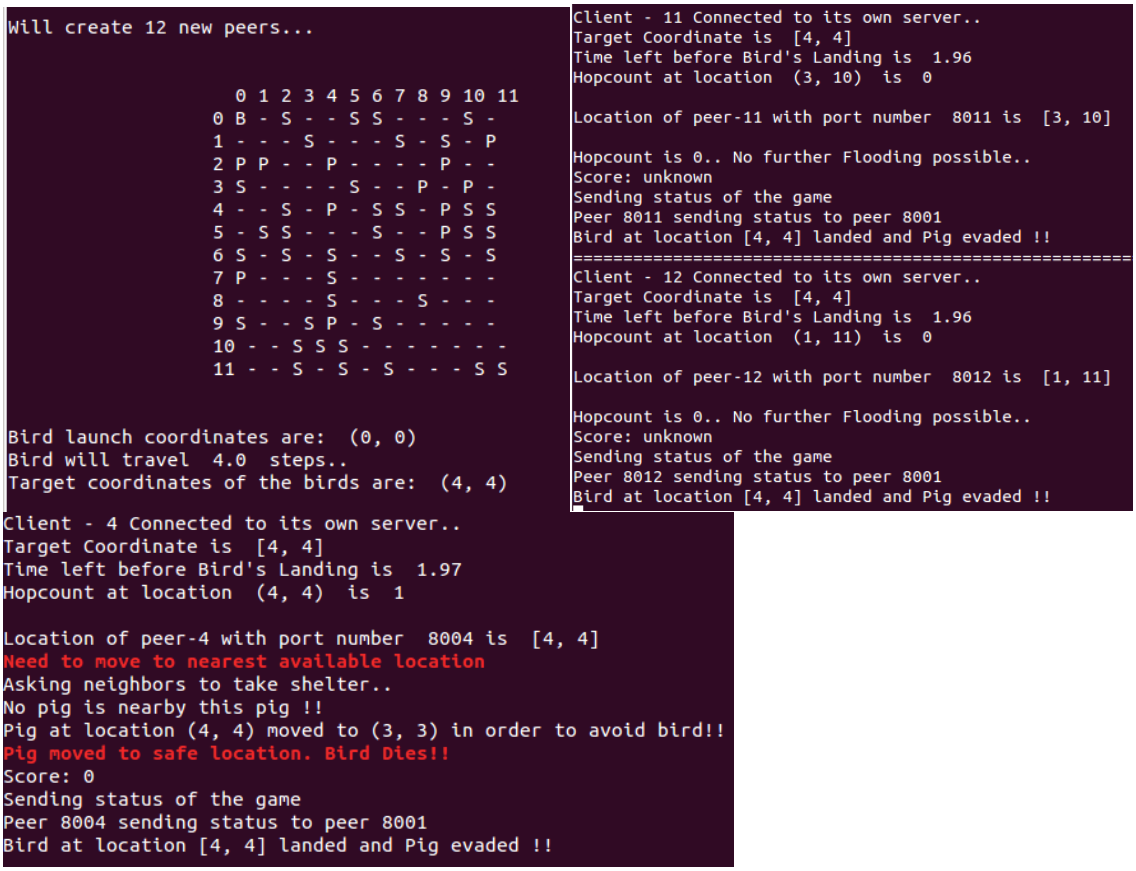
We basically tested our code by inspection these scenarios. We let the program to generate random assignments to the peers and stones, based on our input of heavily dense or sparse network. We have used random generation because it becomes cumbersome to allocate so many locations manually for around 40-50 stone and 20 pigs.

Few of the above cases are shown below:

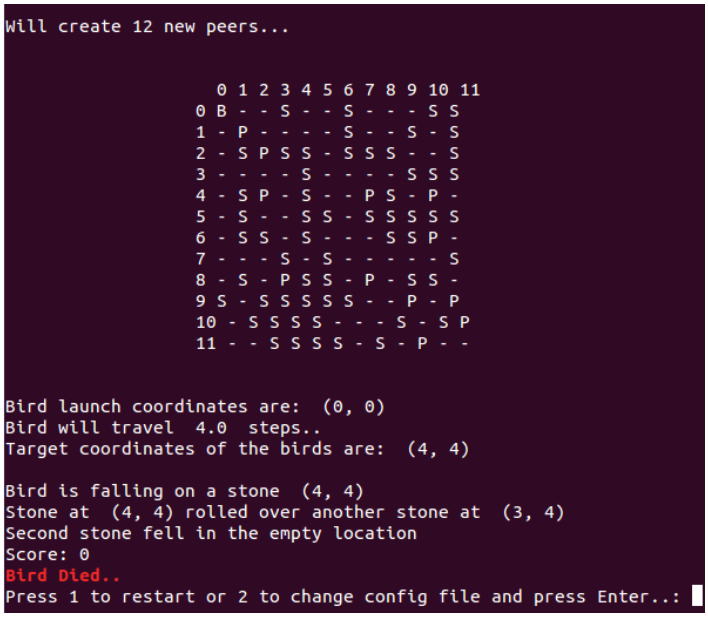
1. Bird falls into an empty location:



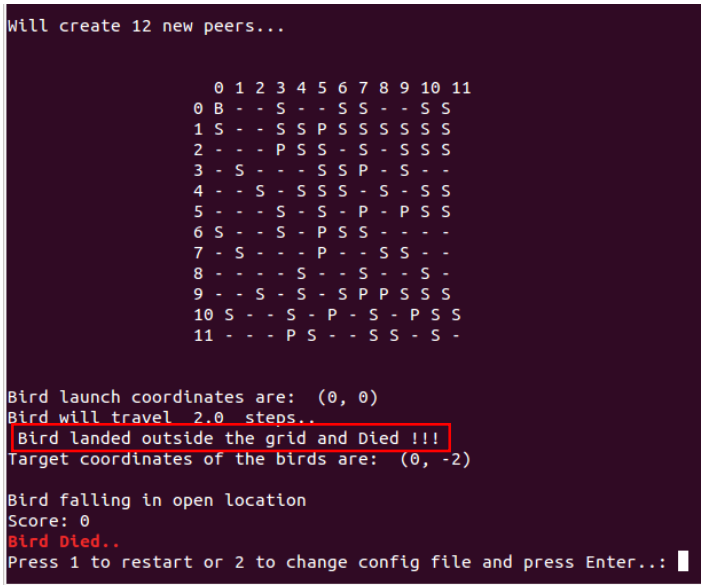
2. Pig evaded the bird and how the status message propagated till the last peer, i.e from Pig4 to pig12.



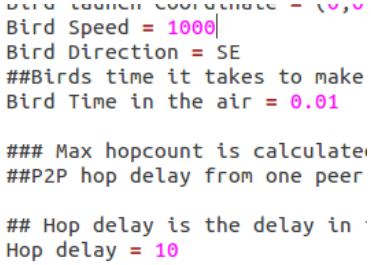
3. When the grid is densely populated, how falling stone affects its neighbors.



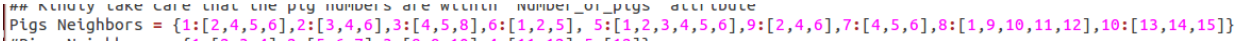
4. When the direction of launch changed and time in the air is increased in the config file. The bird is supposed to go out of the grid. The result that we got shows that our implementation is correct.



5. When Pig could not save itself. This happens when the flight time of the bird is very less (means the bird lands very fast) and the pigs network is not able to reach to a target pig during that time.
Following configuration is used for it:



It is being tested using 15 pigs and 100 stones. The pigs relation used was:



We can see from below output that the status message propagated throughout the network about the bird killed a particular pig. (Hopcount=-1 is just to indicate the user that it's a status message)

```
olenka@olenka-VirtualBox:~/DOS$ ==============================
Client - 1 Connected to its own server..
Target Coordinate is  (8, 9)
Time left before bird's landing is  0.01
Hopcount at location  (3, 2)  is  10.0

Location of peer-1 with port number  8001 is  [3, 2]
sending location to my next peer
Client 8001 Established connection with the server 8002
Client 8001 Established connection with the server 8004
Client 8001 Established connection with the server 8005
Client 8001 Established connection with the server 8006
============================================================
Client - 2 Connected to its own server..
Target Coordinate is  [8, 9]
Time left before Bird's Landing is  0.0
Hopcount at location  (4, 4)  is  9

Location of peer-2 with port number  8002 is  [4, 4]

pig at location  [8, 9]  died !!
Score: 1
Sending status of the game
Peer 8002 sending status to peer 8003
Bird killed pig at location [8, 9]
Peer 8002 sending status to peer 8004
Bird killed pig at location [8, 9]
Peer 8002 sending status to peer 8006
Bird killed pig at location [8, 9]
============================================================
Client - 3 Connected to its own server..
Target Coordinate is  [8, 9]
Time left before Bird's Landing is  0
Hopcount at location  (5, 3)  is  -1

Location of peer-3 with port number  8003 is  [5, 3]
Sending status of the game
Peer 8003 sending status to peer 8004
Bird killed pig at location [8, 9]
Peer 8003 sending status to peer 8005
Bird killed pig at location [8, 9]
Peer 8003 sending status to peer 8008
Bird killed pig at location [8, 9]
============================================================
Client - 4 Connected to its own server..
Target Coordinate is  [8, 9]
Time left before Bird's Landing is  0
Hopcount at location  (0, 4)  is  -1
```