**Issue in the demo:**

While giving the demo we had an issue where the **collision between the Pacman and the Ghost was not detected**. This was due to the logic we used for collision detection and not a concurrency related bug.

Previously the code used was

**if** **(**gx **/** PAC\_SIZE **==** px **/** PAC\_SIZE**)** **{**

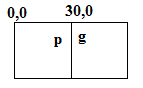
**if** **(**gy **/** PAC\_SIZE **==** py **/** PAC\_SIZE**)** **{**

gameState **=** 2**;**

gameOverMessage **=** Client**.**GHOSTWINS**;**

**}**

Here px, py are the x and y coordinates of the Pacman and gx, gy are the x and y coordinates of the Ghost. The operation gx / PAC\_SIZE would give the horizontal grid index of the ghost (The maze can be thought of as a 13x13 grid ). Similarly px / PAC\_SIZE gives the horizontal grid index for the pacman, gy/ PAC\_SIZE gives the vertical grid index of the ghost and py / PAC\_SIZE gives the vertical grid index of the pacman (PAC\_SIZE is 30). A collision was detected when the ghost and pacman were in the same grid. However as were using the division operation and the GUI increments the grid position by 5 pixels in the movement direction, it was possible in rare cases for the ghost and pacman to intersect but belong to different grids



The diagram above illustrates this. Px/PAC\_SIZE would be 0. Gx/PAC\_SIZE would be 30. Pacman is moving right and Ghost is moving left. If px is in the range (25,30) and gx is in the range (30,35) then after a movement they would change position and interchange grid positions also. (Move by 5 pixels at a time.)

We replaced this logic with the following code:

**if** **(**Math**.**abs**(**gx **-** px**)** **<** PAC\_SIZE **&&** Math**.**abs**(**gy **-** py**)** **<** PAC\_SIZE**)** **{**

gameState **=** 2**;**

gameOverMessage **=** Client**.**GHOSTWINS**;**

**}**

We now don’t have the same problem as we check for the absolute difference in the grid coordinates now instead.

In the demo, we saw that the **game was over but the players were still moving**. This was because of a small bug in the code. The thread which continuosly updated the GUI and made sure the players were moving in the current direction continues to run until it detects that there is a collision or the pacman has eaten all the pellets. In our case, the collision was not detected for the above reason and hence, the GUI continued to run. But the thread listening to server received the "game over" message. So, the message box for game over appeard but the other thread was still running. In order to fix this we changed the variable 'gameState' in the board class, when we received message from the server. This solved the problem.