Module II

Working with Ghosts Team

Game Programming and Robotics - Fall’ 2013
IBA, Karachi

Ghost-team Controller

– Ms Pac-Man's quest is opposed by the four ghosts: Blinky (red), Pinky (pink), Inky (green) and Sue (brown).

– The goal of a ghost-team controller is to minimize the score obtained against it by the Ms Pac-Man controllers. The winning ghost team will be the team with the lowest average score against it.

Rules of the game

- There are no restrictions regarding the actions of Ms Pac-Man: movement in any direction not blocked by a wall is allowed at all times. For the ghost team, on the other hand, three restrictions apply:
  – Ghosts may not reverse direction. Subsequently, a ghost may only choose its direction at a junction, choosing any of the paths available except the one the ghost used to approach the junction.
  – Occasionally there is a global reversal event when all the ghosts suddenly change direction. This may happen in any game tick with a small probability of 0.0015.
  – To prevent the ghosts from spoiling the game by constantly blocking the power pills (in which case the game would result in a stalemate), each level finishes after 3000 time steps; half the score of all remaining pills are awarded to Ms Pac-Man.

Ghosts AI

“To give the game some tension, I wanted the monsters to surround Pac Man at some stage of the game. But I felt it would be too stressful for a human being like Pac Man to be continually surrounded and hunted down. So I created the monsters’ invasions to come in waves. They’d attack and then they’d retreat. As time went by they would regroup, attack, and disperse again. It seemed more natural than having constant attack.”- Toru Iwatani, Pac-Man creator

Ghosts AI

Modes of behavior

Ghosts have three mutually-exclusive modes of behavior they can be in during play: *chase*, *scatter*, and *frightened*. Each mode has a different objective/goal to be carried out:

- **CHASE**—A ghost's objective in *chase* mode is to find and capture Pac-Man by hunting him down through the maze.
- **SCATTER**—In *scatter* mode, the ghosts give up the chase for a few seconds and head for their respective home corners. It is a welcome but brief rest—soon enough, they will revert to chase mode and be after Pac-Man again.
- **FRIGHTENED**—Ghosts enter *frightened* mode whenever Pac-Man eats one of the four energizers located in the far corners of the maze. During the early levels, the ghosts will all turn dark blue (meaning they are vulnerable) and aimlessly wander the maze for a few seconds. They will flash moments before returning to their previous mode of behavior.

Ghosts AI

Different Personalities

- Each ghost exhibits unique behavior when chasing Pac-Man, giving them their different personalities:
  - Blinky (red) is very aggressive and hard to shake once he gets behind you.
  - Pinky (pink) tends to get in front of you and cut you off
  - Inky (light blue) is the least predictable of the bunch
  - Clyde (orange) seems to do his own thing and stay out of the way.

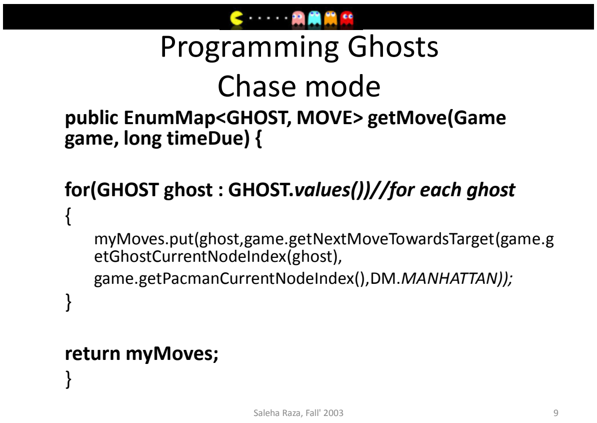
Ghosts AI

Appearance of Cooperation

- Because of the way the individual chase rules were crafted, there seemed to be an implicit cooperation between the ghosts.
- For example, with Blinky always following Pac-Man and Pinky mostly trying to get in front, the player often felt "boxed in".
- In reality, the only ghost that used the position of another ghost in its calculations was Inky who only indirectly used the location of Blinky to determine its target tile.

Path Planning in ghosts

- A* can be used to calculate the move that will bring the ghost closest to a Pacman agent **without interfering with other ghost agents**, **using a heuristic in the weighting of the path**.
- The heuristic is simple: starting at the current position, moves between adjacent squares are weighted one, and the weight is increased by a significant amount if a ghost agent already resides in that square.
- Weights can be adjusted based on the distance of the that point from the starting point
- This helps cooperation between the ghosts as they will not all follow a single path to the Pacman if multiple paths are available.



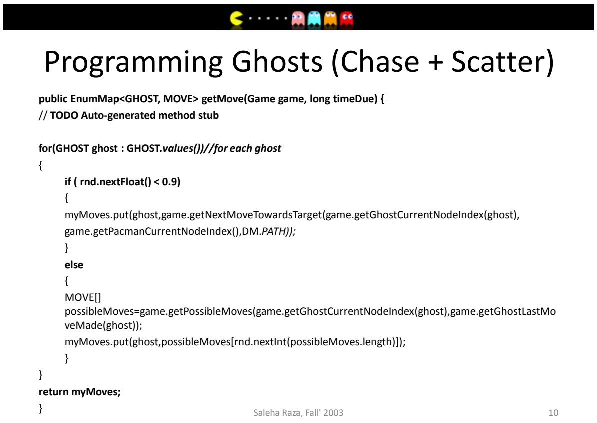
Programming Ghosts Chase mode

```
public EnumMap<GHOST, MOVE> getMove(Game game, long timeDue) {

for(GHOST ghost : GHOST.values())//for each ghost
{
    myMoves.put(ghost,game.getNextMoveTowardsTarget(game.getGhostCurrentNodeIndex(ghost),
    game.getPacmanCurrentNodeIndex(),DM.MANHATTAN));
}

return myMoves;
}
```

Saleha Raza, Fall' 2003 9

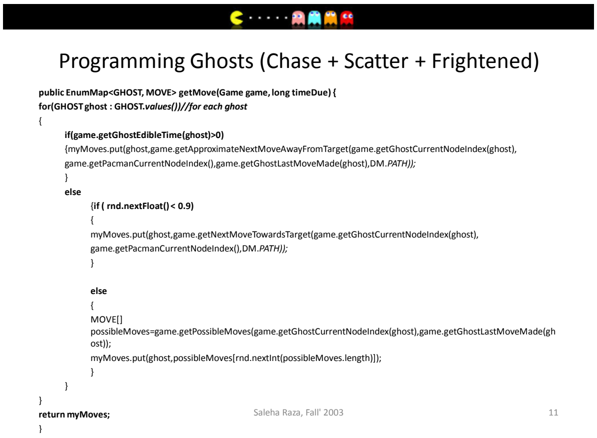


Programming Ghosts (Chase + Scatter)

```
public EnumMap<GHOST, MOVE> getMove(Game game, long timeDue) {
// TODO Auto-generated method stub

for(GHOST ghost : GHOST.values())//for each ghost
{
    if ( rnd.nextFloat() < 0.9)
    {
    myMoves.put(ghost,game.getNextMoveTowardsTarget(game.getGhostCurrentNodeIndex(ghost),
    game.getPacmanCurrentNodeIndex(),DM.PATH));
    }
    else
    {
    MOVE[]
    possibleMoves=game.getPossibleMoves(game.getGhostCurrentNodeIndex(ghost),game.getGhostLastMoveMade(ghost));
    myMoves.put(ghost,possibleMoves[rnd.nextInt(possibleMoves.length)]);
    }
}
return myMoves;
}
```

Saleha Raza, Fall' 2003 10

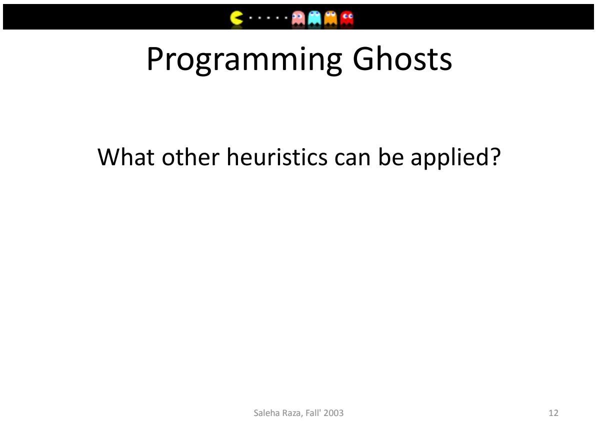


Programming Ghosts (Chase + Scatter + Frightened)

```
public EnumMap<GHOST, MOVE> getMove(Game game, long timeDue) {
for(GHOST ghost : GHOST.values())//for each ghost
{
    if(game.getGhostEdibleTime(ghost)>0)
    {myMoves.put(ghost,game.getApproximateNextMoveAwayFromTarget(game.getGhostCurrentNodeIndex(ghost),
    game.getPacmanCurrentNodeIndex(),game.getGhostLastMoveMade(ghost),DM.PATH));
    }
    else
        {if ( rnd.nextFloat() < 0.9)
        {
        myMoves.put(ghost,game.getNextMoveTowardsTarget(game.getGhostCurrentNodeIndex(ghost),
        game.getPacmanCurrentNodeIndex(),DM.PATH));
        }

        else
        {
        MOVE[]
        possibleMoves=game.getPossibleMoves(game.getGhostCurrentNodeIndex(ghost),game.getGhostLastMoveMade(ghost));
        myMoves.put(ghost,possibleMoves[rnd.nextInt(possibleMoves.length)]);
        }
    }
}
return myMoves;
}
```

Saleha Raza, Fall' 2003 11



Programming Ghosts

What other heuristics can be applied?

Saleha Raza, Fall' 2003 12