Kuncheng Feng CSC 466 progress report

## Check if game ended

## Abstract

The game is structured by a player instance vs a player instance, and the game ends if one or both players have won, a player is considered won if the other player is lost. The wording is weird here but it reflects the logic of the code.

## Code

A player is defeated if all of its ships are sunk.

```
(defmethod isPlayerDefeated((p humanPlayer) &aux ships result sunk)
  (setf ships (player-ships p))
  (setf result t)
  (loop for ship in ships do
      (setf sunk (isShipSunk ship))
      (setf result (and result sunk))
  )
  result
)
```

```
(defmethod isPlayerDefeated((p randomPlayer) &aux ships result sunk)
  (setf ships (player-ships p))
  (setf result t)
  (loop for ship in ships do
      (setf sunk (isShipSunk ship))
      (setf result (and result sunk))
  )
  result
)
```

A ship is considered sunk if all of its cells had been explored



The game over checked is actually embedded in another function, which will be explained in the next presentation.

```
(defun takeTurn(player1 player2 &aux p1Win p2Win)
  (playerOpenFire player1)
  (playerOpenFire player2)
  (setf p1Win (isPlayerDefeated player2))
  (setf p2Win (isPlayerDefeated player1))
  (cond
      ((and p1Win p2Win)
        (format t "It's a draw.~&")
      )
      (p1Win
        (format t "You won!~%")
      )
      (p2Win
        (format t "You lost!~%")
      )
      (t
        (takeTurn player1 player2)
      )
   )
)
```