

```

/* This class encapsulates Monster position and direction
 * It also keeps a reference to the player it is tracking
 * The canView attribute can be used to limit monster visibility
 */

public class Monster extends Moveable {
    private boolean canView = true; // allows
    private Player player;
    boolean isBaby = false;
    int bornTime = 0;

    public Monster(Grid g, Player p, int row, int col) throws Exception {
        super(g);
        player = p;
        setCell(grid.getCell(row, col));
    }

    // check if it's baby monster
    public boolean isBaby() {
        return isBaby;
    }

    public Cell move(int presses) {
        boolean gotStoped = false;
        if (bornTime > 0) {
            for (int i = 0; i < player.getBlock().size(); ++i) {
                if (player.getBlock().get(i).getHealth() == 0) {
                    player.getBlock().remove(i);
                } else if (currentCell == player.getBlock().get(i).getCell()) {
                    player.getBlock().get(i).healthDec();
                    gotStoped = true;
                }
            }
            for (int i = 0; i < player.getTrap().size(); ++i) {
                if (currentCell == player.getTrap().get(i).getCell()) {
                    gotStoped = true;
                }
            }
            if (gotStoped == false) {
                if (currentCell.row % 5 == 0 && currentCell.col % 5 == 0
                    && (currentCell.row == player.getCell().row ||
currentCell.col == player.getCell().col)) {
                    currentCell = player.getCell();
                } else {
                    currentDirection = grid.getBestDirection(currentCell,
player.getCell());
                    currentCell = grid.getCell(getCell(), getDirection(), 1);
                }
            }
            bornTime++;
            if (bornTime >= 10)
                isBaby = false;
            return currentCell;
        }

        // check if monster needs to be hidden
        public boolean viewable() {
            Cell p = player.getCell();
            Cell m = currentCell;
            if (bornTime > 0 && (Math.abs(p.col - m.col) > 4 || Math.abs(p.row - m.row) > 4)) {
                canView = false;
            } else {
                canView = true;
            }
            return canView;
        }
    }
}

```