```java
import java.util.ArrayList;

/*  This class encapsulates player position and direction
 */
public class Player extends Moveable {
    private boolean readyToStart = false;
    private int presses = 0, energy = 200;
    private int roadblockPut = 0;
    private ArrayList<Trap> traps = new ArrayList<Trap>();
    private ArrayList<Roadblock> roadblocks = new
 ArrayList<Roadblock>();
    private boolean canPutTrap = true, canPutBlock = true;

    public Player(Grid g, int row, int col) throws Exception {
        super(g);
        currentCell = grid.getCell(row, col);
        currentDirection = ' ';
    }

    public Cell move(int presses) {
        boolean canMove = false;
        if ((presses == 1 && energy >= 2) || (presses == 2 &&
 energy >= 6) || (presses == 3 && energy >= 14)) {
            canMove = true;
        }
        if (getTrap().size() > 0)
            for (int i = 0; i < getTrap().size(); ++i) {
                getTrap().get(i).decTime();
                if (getTrap().get(i).getTime() == 0) {
                    getTrap().remove(i);
                }
            }
        if (canMove) {
            currentCell = grid.getCell(currentCell,
 currentDirection, presses);
            if (currentCell.gotGold) {
                energy += 6;
                currentCell.gotGold = false;
            }
            if (presses == 1) {
                energy -= 2;
            } else if (presses == 2) {
```

```java
                  energy -= 6;
            } else if (presses == 3) {
                  energy -= 14;
            }
        }

        if (presses != 0) {
            allowPut();
        }

        clearPress();
        return currentCell;
    }

    // keypress count for moving multiple cells
    public int getPresses() {
        return presses;
    }

    public void setEnergy(int energy) {
        this.energy = energy;
    }

    public int getEnergy() {
        return energy;
    }

    public ArrayList<Roadblock> getBlock() {
        return roadblocks;
    }

    public ArrayList<Trap> getTrap() {
        return traps;
    }

    // eating nougats
    public void eat() {
        energy += 6;
    }

    public void addPress() {
        if (presses < 3) {
            presses += 1;
```

```java
82          }
83      }
84
85      public void clearPress() {
86          presses = 0;
87      }
88
89      public void putTrap() {
90          if (canPutTrap && energy >= 50) {
91              traps.add(new Trap(grid, currentCell));
92              energy -= 50;
93              canPutTrap = false;
94          }
95      }
96
97      public void putBlock() {
98          if (canPutBlock && roadblockPut < 3) {
99              roadblocks.add(new Roadblock(grid, currentCell));
100             roadblockPut += 1;
101             canPutBlock = false;
102         }
103     }
104
105     public void allowPut() {
106         canPutTrap = true;
107         canPutBlock = true;
108     }
109
110     public void setReady(boolean val) {
111         readyToStart = val;
112     }
113
114     public boolean isReady() {
115         return readyToStart;
116     }
117 }
118
```