SOLUZIONE ULTIMO ESERCIZIO PROVA IN ITINERE

```java
public class PlayerTask implements Runnable {
    private final Game game;
    private final int playerId;
    private final int rounds;
    public PlayerTask(TurnSynchronization game, int playerId, int rounds) {
        this.game = game;
        this.playerId = playerId;
        this.rounds = rounds;
    }
    @Override
    public void run() {
        for (int i = 0; i < rounds; i++) {
            try {
                game.play(playerId);
            } catch (InterruptedException e) {
                e.printStackTrace();
                Thread.currentThread().interrupt();
            }
        }
    }
```

```java
public class Game {
  private int turn = 1; // 1 = Player1, 2 = Player2, 3 = Player3
  public synchronized void play(int playerId) throws InterruptedException {
      // Aspetta finché non è il tuo turno
      while (turn != playerId) {
          wait();
      }
      // Gioca
      System.out.println("Player " + playerId + " gioca!");
      // Passa turno al prossimo player
      turn = (playerId % 3) + 1;
      // Risveglia tutti i giocatori in attesa del proprio turno
      notifyAll();
  }}
```

```java
public class Main {
public static void main(String[] args) {
    Game game = new Game();
    Thread p1 = new Thread(new PlayerTask(game, 1, 10));
    Thread p2 = new Thread(new PlayerTask(game, 2, 10));
```

```
    Thread p3 = new Thread(new PlayerTask(game, 3, 10));
    p1.start();
    p2.start();
    p3.start();
}}
```