

Assignment 3 Esercizio 2 PCD 20/21

Christ Valtes Medjouwo Diagong

Christ.medjoumo@studio.unibo.it

Idene Prisca Magadjou Tchendjou

Ideneprisca.@studio.unibo.it

1 -Analisi del problema

Il programma richiesto consiste in una versione multigiocatore distribuita via rete, dinamica e collaborativa di un puzzle semplificato. Ogni giocatore deve poter entrare e uscire liberamente dalla partita e collaborare alla composizione dell'immagine originale. L'evolversi dei progressi nella risoluzione del gioco deve essere in tempo reale e visibile per tutti. Viene richiesta la possibilità di vedere le azioni degli altri giocatori partecipanti per un'esperienza collaborativa migliore.

2- Soluzione 1 basata su paradigma ad attori P2P

Il sistema è implementato con akka e sfrutta il Artery **Remoting** come meccanismo attraverso il quale gli attori su nodi diversi parlano tra loro internamente.

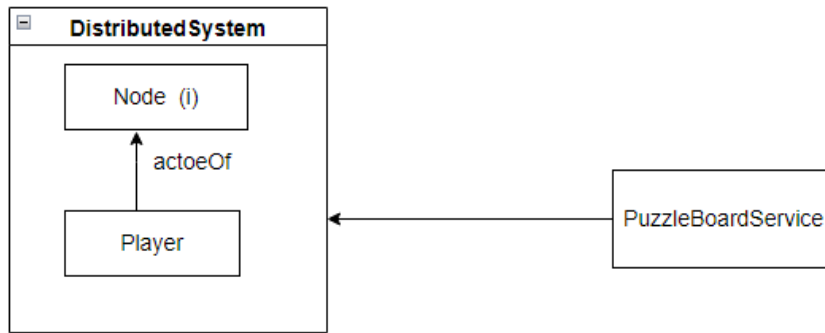


Fig1: Nodo principale insieme alla Board condivisa

Le regole che definiscono il sistema sono le seguenti:

- È definito un **ActorSystem** abilitato ad essere remote connesso su un host e una porta ben specificato dal **applicationConf**. L'attore **Player** figlio suo
- Questo attore gioca il ruolo di **EntryPoint** per tutti gli altri attori del sistema fornendo una porta diversa ogni volta che si presenta un nuovo giocatore.
- Visto che si tratta di un'architettura peer To peer, non c'è un server centrale il sistema è completamente decentralizzato quindi ogni nuovo giocatore è esposto su una porta diversa fornita dal giocatore precedente .la comunicazione dai diversi nodi di ogni giocatore si fa con scambio di Messaggio. **L'actorSelector** di **Akka** ci permette di ottenere il riferimento esatto di ogni oggetto che è esposto in un nodo remoto.
- I principali messaggi scambiati sono le seguenti.

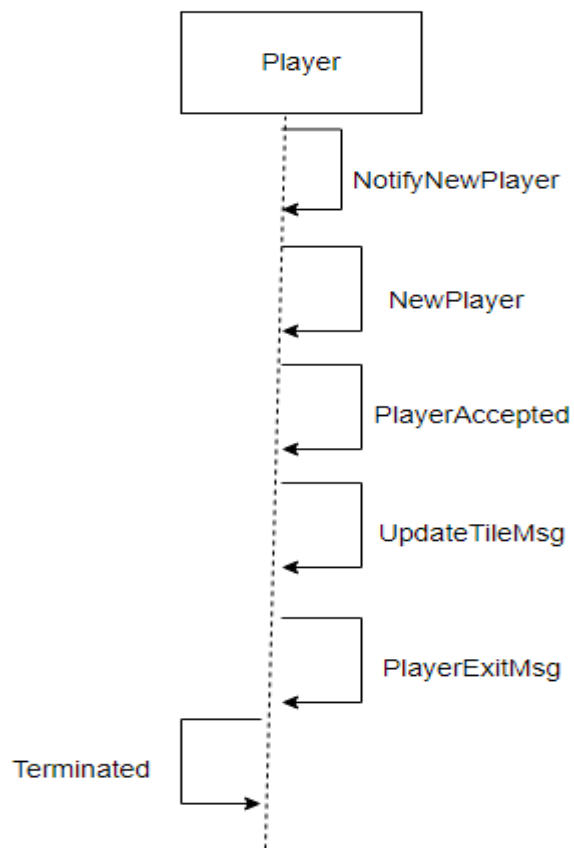


Fig2: cambio di messaggi dalle differenti istanze dell'attore Player.

3 – Soluzione 3 basata su java RMI

La soluzione proposta utilizza RMI come tecnologia attraverso la quale i differenti nodi comunicano.

La BoordService premete di creare un board per ogni giocatore, di gestire gli aggiornamenti di stato da parte di ogni giocatore. Consente di registrare tutte le diverse board, e di verificare lo stato completo del gioco cioè verificare se un giocatore ha completato o è riuscito a combinare l'immagine giusta.

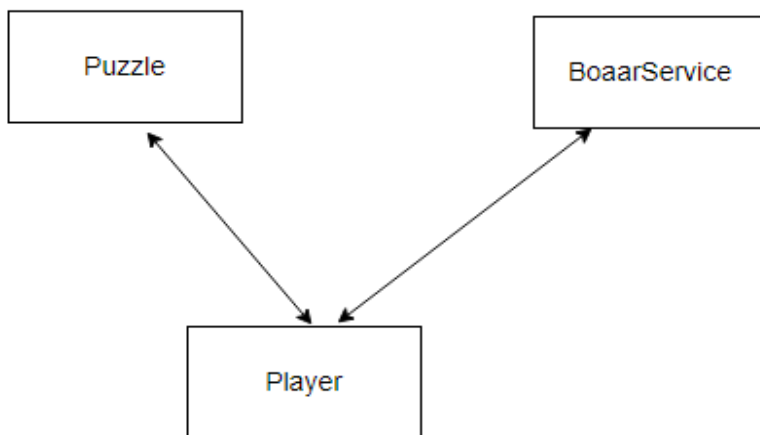


Fig3: Architettura del sistema.