

## Robot Brain Thread - Documentatie

---

Metoda `run()` este **bucloa principala** a unui jucator controlat de AI. Responsabilitatile sale sunt:

- **Ascultarea** evenimentelor/mesajelor de la server.
- **Actualizarea cunostintelor interne** despre starea jocului.
- **Luarea deciziilor** atunci cand este randul robotului (constructii, schimburi, dat zarul, mutarea hotului, folosirea cartilor de dezvoltare etc.).
- **Trimiterea actiunilor** catre server.
- **Gestionarea timeout-urilor**, erorilor si conditiilor speciale (constructie speciala, atacuri cu pirati).
- **Parasirea** corecta a jocului cand se termina sau cand este eliminat.

Ruleaza intr-o bucla continua (`while (alive)`), procesand evenimente si planificand actiuni.

---

## Ce informatii sunt folosite si modificate

### 1. Starea Jocului (`game`, `SOCGame`)

Actualizeaza starea jocului la mesajele `GAMESTATE` si `STARTGAME`:

```
handleGAMESTATE(((SOCGameState) mes).getState());
game.setCurrentPlayerNumber(((SOCTurn) mes).getPlayerNumber());
game.updateAtTurn();
```

### 2. Informatii despre Jucatorul Curent (`ourTurn`, `ourPlayerNumber`)

Verifica daca este randul robotului:

```
if (game.getCurrentPlayerNumber() == ourPlayerNumber)
    ourTurn = true;
else
    ourTurn = false;
```

### 3. Resurse (**ourPlayerData.getResources()**)

Monitorizeaza numarul de resurse pentru a planifica constructii sau a gestiona discard-ul:

```
if (ourPlayerData.getResources().getTotal() > 1)
    planBuilding();

if (ourPlayerData.getResources().getTotal() > 7)
    expectDISCARD = true;
```

### 4. Plan de Constructie (**buildingPlan**)

Pastreaza un plan pentru ce va construi robotul:

```
if (buildingPlan.isEmpty())
    planBuilding();

if (!buildingPlan.isEmpty())
    client.buildRequest(game, -1);
```

### 5. Schimburi si Raspunsuri

#### (**waitingForTradeResponse, handleMAKEOFFER**)

Gestioneaza oferirea, acceptarea sau respingerea schimburilor:

```
handleMAKEOFFER((SOCMakeOffer) mes);
handleTradeResponse(acceptingPN, true);
```

### 6. Mutarea Hotului (**moveRobberOnSeven, expectPLACING\_ROBBER**)

Muta hotul cand se da cu 7 sau se foloseste o carte de cavaler:

```
if (((SOCDiceResult) mes).getResult() == 7)
    moveRobberOnSeven = true;
```

```
if ((game.getGameState() == SOCGame.PLACING_ROBBER) && !waitingForGameState)
    moveRobber();
```

## 7. Cronometru / Timeout (**counter**)

Foloseste un contor pentru a evita blocarile:

```
counter++;
if (counter > 15000)
    client.leaveGame(game, "counter 15000", true, false);
```

---

# Cum Reactioneaza Robotul la Evenimentele Jocului

Tip Mesaj	Actiune Robot
GAMESTATE	Actualizare stare joc
STARTGAME	Initializare joc si pozitionari initiale
TURN	Resetare variabile pe tura, verificare daca e randul nostru
PLAYERELEMENT, RESOURCECOUNT	Actualizare resurse jucatori
DICERESULT, DICERESULTRESOURCES	Tratare rezultate zar, mutare hot
PUTPIECE, MOVEPIECE	Actualizare pozitii constructii/nave
MAKEOFFER, ACCEPTOFFER, REJECTOFFER	Gestionare schimburi
DEVCARDACTION	Folosire carti de dezvoltare
SIMPLEACTION, SIMPLEREQUEST	Gestionare evenimente speciale minore
TIMINGPING	Incrementare contor pentru timeout
DISCARDREQUEST	Alegere carti pentru discard
CHOOSEPLAYERREQUEST	Alegerea unui jucator de jefuit sau pentru schimb

---

# Variable Cheie in Robot Brain

Variabila	Scop
<code>alive</code>	Daca este <code>false</code> , robotul opreste bucla
<code>counter</code>	Tine timpul, detecteaza blocari
<code>ourTurn</code>	Daca este randul robotului
<code>waitingForGameState</code>	Asteptare raspuns server
<code>expectPLAY1</code>	Pregatit pentru actiuni normale pe tura
<code>expectPLACING_ROBBER</code>	Asteptare plasare hot
<code>buildingPlan</code>	Ce vrea sa construiasca robotul
<code>waitingForTradeResponse</code>	Asteptare raspuns schimb
<code>waitingForDevCard</code>	Asteptare cumparare carte de dezvoltare

---

## Fluxul General

1. Asteapta un nou eveniment (SOCMessage) de la server.
2. Actualizeaza cunostintele interne despre joc.
3. Daca este randul robotului si nu asteapta ceva:
  - Decide actiunea (dau cu zarul, construiesc, schimb, joc carte).
4. Trimite actiunea prin `client`.
5. Continua in bucla pana la sfarsitul jocului sau eliminarea robotului.

