

SISTEMA MULTIAGENT APLICAT A L'EXTRACCIÓ DE CONEIXEMENT DE VIDEOJOCs MOBA

Pol Major Munich

**Grau d'Enginyeria Informàtica
Intel·ligència Artificial**

**Consultor: David Isern Alarcón
7 de gener del 2018**



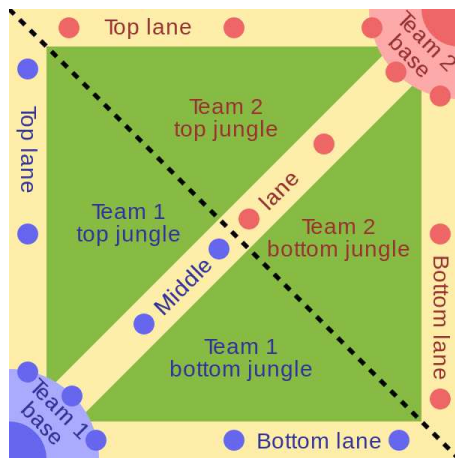
ÍNDEX

- Introducció del projecte:
 - *Multiplayer Online Battle Arena* (MOBA).
 - API de Riot Games.
 - Disseny de la solució: eines utilitzades.
- Implementació:
 - Extracció de les dades més rellevants.
 - Tractament de les dades.
 - Anàlisi estadística.
 - Anàlisi descriptiva: regles d'associació.
- Resultats.
- Conclusions.



1.MOBA

- En els videojocs MOBA dos equips s'enfronten en batalles de curta duració.
- Cada jugador utilitza un personatge i sol tenir com a objectiu destruir la base enemiga.



- Videojoc: *League Of Legends* (LoL)
- Duració de la partida: 15-60minuts.
- Número de jugadors: 5 contra 5.
- Campions a elegir: actualment, 139.



1.1 PERQUÈ ELS MOBA?

- Es juguen milers de partides diàries.
- Cada partida té multitud de variables i factors determinants de la victòria o la derrota.
- Per tant, hi ha una gran quantitat de patrons i coneixement ocult entre les dades que generen.
- S'espera trobar coneixement que millori l'experiència de joc dels usuaris.



2. L'API DE *RIOT GAMES*

- Permet extreure dades del LoL en format JSON.
 - Dels personatges.
 - De les partides.
- Utilitza una clau provisional.
 - Té certes limitacions: només 100 crides cada 2minuts.
- Sempre per millorar l'experiència de joc.

```
{
  "title": "the Barbarian King",
  "name": "Tryndamere",
  "key": "Tryndamere",
  "id": 23
}
{
  "stats": {
    "armorperlevel": 3.1,
    "attackdamage": 69,
    "mpperlevel": 0,
    "attackspeedoffset": -0.0672,
    "mp": 100,
    "armor": 33,
    "hp": 625.64,
    "hpregenperlevel": 0.9,
    "attackspeedperlevel": 2.9,
    "attackrange": 125,
    "movespeed": 345,
    "attackdamageperlevel": 3.2,
    "mpregenperlevel": 0,
    "critperlevel": 0,
    "spellblockperlevel": 1.25,
    "crit": 0,
    "mpregen": 0,
    "spellblock": 32.1,
    "hpregen": 8.512,
    "hpperlevel": 98
  },
}
```



3. DISSENY DE LA SOLUCIÓ

- Què es vol aconseguir?
 - Ajudar als jugadors a prendre millors decisions de joc.
 - Facilitar la tasca d'anivellament del joc als seus desenvolupadors.
- Com es vol fer?
 - Extraient, processant i analitzant automàticament **noves dades** cada dia.



3.1 EINES UTILITZADES

- R, per crear models de regles d'associació a través d'*scripts*.



- Splunk, per automatitzar l'anàlisi estadística de les dades.



- Agents intel·ligents, capaços d'interactuar entre ells i amb l'API de Riot Games, R i Splunk.

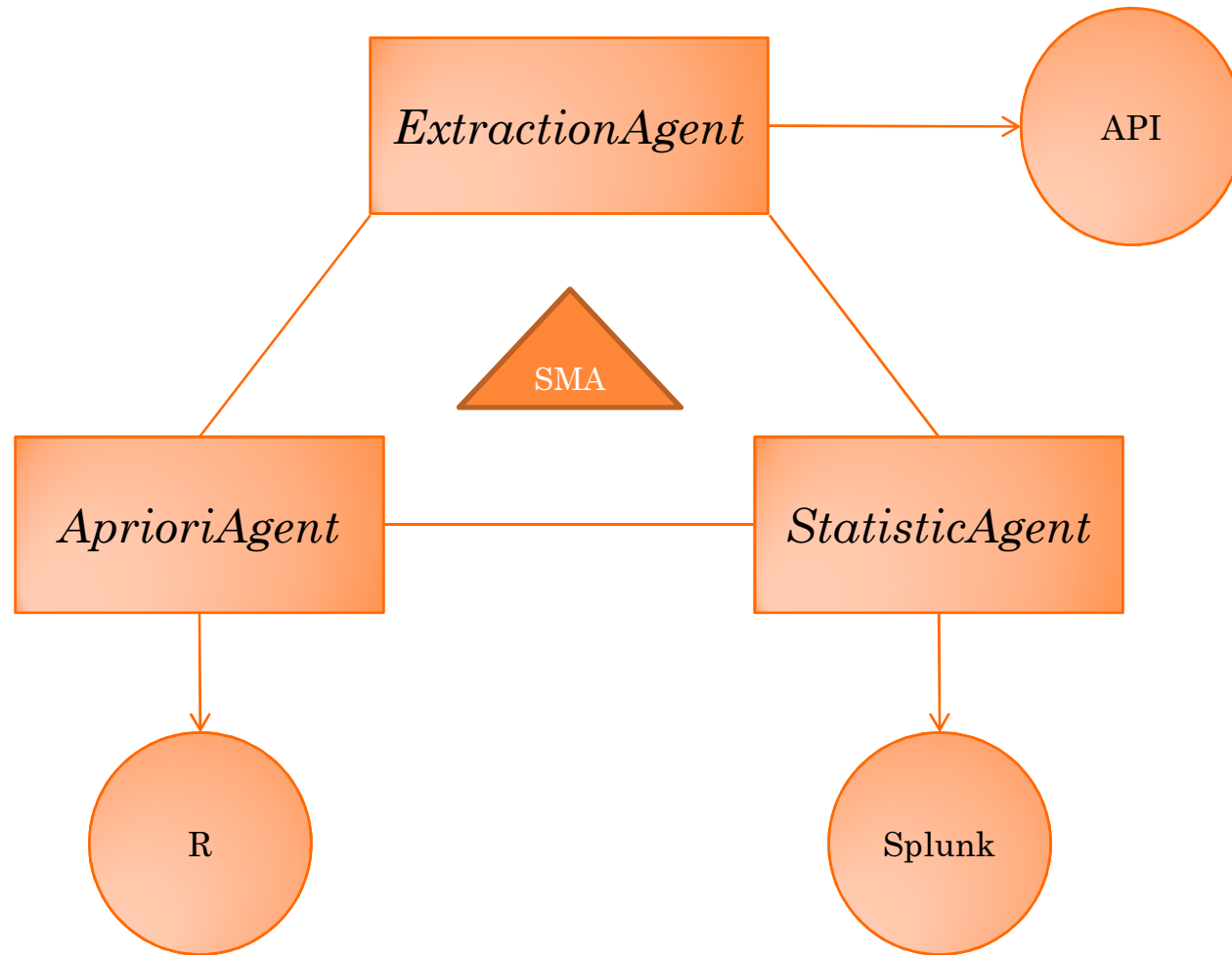


3.2 PER QUÈ UN SMA?

- Desacoblament.
- Automatització dels processos.
- Comunicació en temps real.
- Descentralització.
- Eficiència.
- Estudi dels SMA en aquest àmbit.



3.2.1 EL SISTEMA MULTIAGENT



4. EXTRACCIÓ DE LES DADES

- L'*ExtractionAgent* interactua amb l'API de Riot Games.
- Per gestionar les limitacions de la clau provisional, s'espera 2 minuts cada 100 peticions.
- Hi ha diferents tipus de dades:
 - Estàtiques, que varien poc o gens en mesos.
 - Dinàmiques, que s'actualitzen cada pocs minuts.
 - Les més interessant són les referents a les partides que es juguen constantment.



4.1 ALGORISME D'EXTRACCIÓ

- No totes les partides són igual de rellevants.
 - S'utilitza un codi únic per cada tipus de partida.
 - Existeix un *timestamp* general.
- S'agafen només les partides classificatòries més recents:

```
Algorisme ExtreurePartides()
inici
obtenirPartidesActuals()
guardarMarcaTemps()
per cada partidaActual fer
    obtenirJugadorsActius()
fiper
per cada jugadorActiu fer
    filtrarHistorialPartides(temps, codi)
    descartarPartidesRepetides()
    guardarIdentificacióPartides()
fiper
fi
```



4.2 TRACTAMENT DE LES DADES

- A partir de l'identificador de les partides, l'agent es centra, de moment, en escriure dues llistes:
 1. Conjunt de personatges d'equips guanyadors.
 2. Conjunt de personatges d'equips perdedors.

```
Algorisme llegirGuanyadors()
```

```
inici
```

```
llegirIdentificadorsDelFitxer()
```

```
per cada identificador fer
```

```
    obtenirPartidaDeAPI(identificador)
```

```
    consultarEquipGuanyador()
```

```
    consultarCampions()
```

```
    escriureLlistes()
```

```
    si ((cridesAPI % 100) == 0) llavors
```

```
        esperar(120s)
```

```
    fisi
```

```
fiper
```

```
fi
```

```
id,champ
```

```
3463363495,Jhin
```

```
3463363495,Leona
```

```
3463363495,Katarina
```

```
3463363495,Maokai
```

```
3462026830,Nidalee
```

```
3462026830,Vladimir
```

```
3462026830,Urgot
```

```
3462026830,Taric
```

```
3462026830,MissFortune
```



5. ANÀLISI ESTADÍSTICA

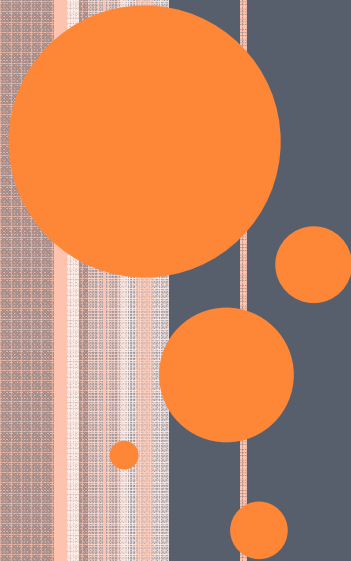
- L'*StatisticAgent* s'encarrega d'aquesta part. Cada cert temps:
 - Demana noves dades a l'*ExtractionAgent*.
 - Si n'obté, les puja a la plataforma Splunk.
 - Actualitza els resultats dels anàlisis automàtics.
 - Si detecta campions amb estadístiques massa positives, en demana un anàlisi més a fons a l'*AprioriAgent*.

```
WinRatio, PickRation, Champion  
56.986900, 17.442303, Khazix  
55.717256, 18.318227, Camille  
54.107981, 32.447254, Ezreal  
53.770492, 11.615508, Evelynn  
53.498871, 16.871049, Maokai  
53.364269, 16.414045, Janna  
53.246753, 20.527077, Xerath  
52.592593, 20.565161, Alistar
```



ANÀLISI ESTADÍSTICA:

SPLUNK



6. ANÀLISI DESCRIPTIVA

- L'*AprioriAgent* interactua amb el programa R per a crear models descriptius, executant *scripts* creats dinàmicament.
- Utilitza l'algorisme *apriori*, que resulta en un conjunt de regles d'associació del tipus:
 - Si (a) llavors (b)
- Per mesurar la rellevància de cada regla, s'utilitzen tres factors anomenats suport, confiança i pes.



6.1 FACTORS DE RELLEVÀNCIA

(Rakan) => (Xayah)

```
[3]
{Rakan}
=> {Xayah}
support: 0.0182
lift: 8.08
confidence: 0.535
```

- Support = $P(\text{Rakan}, \text{Xayah}) = 0.0182$
- Confiança = $\text{suport} / P(\text{Rakan}) = 0.535$
- Pes = $\text{confiança} / P(\text{Xayah}) = 8.08$
- És un cas de forta relació entre dos personatges. En presència d'un, l'altre apareix 8.08 vegades més sovint del que és habitual.



7. RESULTATS

- Es generen diferents fitxers resultants dels anàlisis. La idea és poder fer-los públics, en un futur, a través d'un servidor Web.

- Fitxers html:

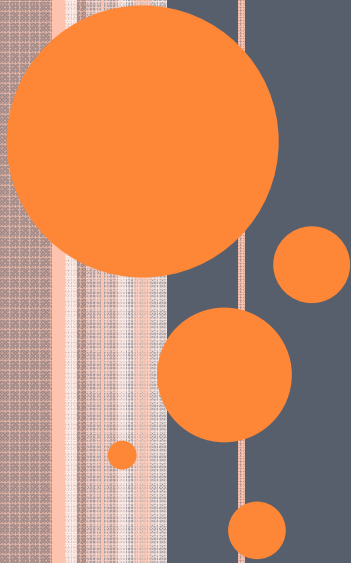
- Gràfic **de dispersió** de les regles guanyadores i perdedores.
- Gràfic **descriptiu interactiu** de les regles guanyadores i perdedores.

- Fitxer de text:

```
Synergy found: Rakan=>Xayah with a lift value of: 6.591903
Janna is too strong! No synergy detected.
Synergy found: Lulu=>KogMaw with a lift value of: 7.4913405
Synergy found: Lulu=>Twitch with a lift value of: 3.2504197
Malzahar is too strong! No synergy detected.
Taliyah is too strong! No synergy detected.
Vladimir is too strong! No synergy detected.
```



RESULTATS



8. CONCLUSIONS

- El SMA:
 - Compleix les seves funcions sense necessitat d'intervenció humana.
 - A més, extreu coneixement valuós tan per els jugadors com per als desenvolupadors del joc.
 - Presenta els resultats automàticament en forma de fitxer web *html*.
- En general:
 - S'han complert els objectius establerts.
 - El projecte pot ajudar a millorar l'experiència de joc dels jugadors.
 - No obstant, hi ha moltes línies de futur per explorar.



8.1 LÍNIES DE FUTUR

- El SMA és, encara, limitat en quant a funcionalitats.
- Hi ha moltes més dades, a part dels camps, que poden ser analitzades.
- Han quedat pendents altres tipus d'anàlisi:
 - Per exemple, utilitzar altres algorismes de mineria de dades, a part de l'*apriori*.
 - També, automatitzar el càlcul del percentatge de victòria per parelles.
- Adaptar el sistema a altres videojocs MOBA.
- Crear una interfície Web que permeti fer peticions concretes als agents.
- Obtenir una clau permanent presentant el projecte a Riot Games.





FI