
This is the **published version** of the bachelor thesis:

Arnaus Comellas, Martí; Vanrell i Martorell, Maria Isabel, dir. Jugador d'un joc de cartes interactiu. 2023. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/272822>

under the terms of the  license

Jugador d'un joc de cartes interactiu

Martí Arnaus Comellas

Resum- L'objectiu d'aquest projecte és dissenyar i implementar diversos algoritmes d'Intel·ligència Artificial (IA) que siguin capaços de jugar a *La Podrida* enfront d'un jugador humà o d'altres agents intel·ligents. El sistema s'ha orientat a expandir la llibreria RL Card de Python. A part del disseny del jugador humà i el jugador aleatori, s'ha dissenyat un agent basat en regles heurístiques i un basat en la Cerca de Monte-Carlo (Monte-Carlo Tree Search MCTS). Per avaluar els diferents agents, s'han dissenyat i realitzat un conjunt d'experiments que han fet concloure que el MCTS és el millor algorisme i ens ha permès estudiar diferents alternatives de millores que de moment queden fora de l'abast d'aquest projecte.

Paraules Clau- Algoritmes IA, Monte-Carlo Tree Search, Reinforcement Learning, RL Card, Rule-Based, Xarxa Neuronal

Abstract- The aim of this project is to design and implement various Artificial Intelligence (AI) algorithms that are capable of playing *La Podrida* against a human player or other intelligent agents. The system is aimed at expanding the Python RL Card library. Apart from the design of the human player and the random player, an agent based on heuristic rules and one based on Monte-Carlo Tree Search (MCTS) have been designed. To evaluate the different agents, a set of experiments have been designed and carried out that have led to the conclusion that the MCTS is the best algorithm and has allowed us to envisage different alternatives for improvements that are currently outside the scope of this project.

Keywords- AI Algorithms, Monte-Carlo Tree Search, Reinforcement Learning, RL Card, Rule-Based, Neuronal Network

1 INTRODUCCIÓ - CONTEXT DEL TREBALL

El joc de La Podrida és un joc de cartes conegut a tot el país, amb diferents noms i variants. Per una de les variants de Catalunya es troba a faltar una intel·ligència artificial que sigui capaç de jugar-hi, i és per això que es va decidir d'implementar la idea de crear un agent intel·ligent que jugui a La Podrida.

La Podrida és un joc de cartes, normalment es juga amb la baralla espanyola i hi poden jugar de 3 a 12 jugadors, encara que el nombre de persones més adient per jugar una partida de manera més còmoda és de 4 a 6 jugadors.

El joc consisteix a endevinar les bases que aconseguiràs a cada ronda, sumant punts en funció de les bases guanyades i coincideixi amb el nombre predit anteriorment [1].

Actualment, hi ha molts agents per a molts jocs, ja siguin jocs de taula, videojocs, jocs de cartes, etc. però no n'he trobat cap que estigui entrenat per jugar a La Podrida. Així doncs, el treball consistirà a implementar un agent intel·ligent que pugui jugar a aquest joc.

La idea és implementar diferents agents, de menys a més complexitat, començant per un agent aleatori, seguit d'un basat en regles heurístiques i per acabar un agent basat en

l'algorisme Monte-Carlo Tree Search.

S'ha decidit treballar amb la llibreria RL Card, que se centra a aplicar diferents algorismes d'Intel·ligència Artificial a jocs de cartes. La llibreria conté varis entorns per a diferents jocs, amb interfícies gràfiques per a jugar més còmodament [2][3]. Per tant, l'objectiu principal d'aquest treball consistirà en afegir un paquet a aquesta llibreria pel joc de La Podrida.

2 OBJECTIUS

Per assolir l'objectiu global, comentat a l'apartat anterior, s'han plantejat una sèrie d'objectius més específics per tal que sigui assolible:

Objectiu 1: Adaptar la llibreria RL Card per tal que els agents implementats puguin jugar a La Podrida.

Objectiu 2: Implementar un agent basat en regles heurístiques que sigui capaç de jugar a La Podrida i que podrà servir com a base d'un agent intel·ligent que raona.

Objectiu 3: Implementar amb l'algorisme MCTS un agent que sigui capaç de jugar a La Podrida sense fer servir heurístiques i estimant probabilitats a partir de simulacions aleatòries.

- E-mail de contacte: marti.arnaus@autonoma.cat
- Menció realitzada: Computació
- Treball tutoritzat per: Maria Vanrell Martorell
- Curs 2022/23

Objectiu 4: Avaluar tots els agents, i en particular l'agent amb l'algorisme MCTS pel qual s'han de fixar els paràmetres per obtenir resultats òptims i que millori l'agent basat en regles heurístiques.

3 METODOLOGIA

El desenvolupament d'aquest treball es farà en el llenguatge Python, ja que és el llenguatge que s'utilitza en la majoria de problemes d'intel·ligència artificial, pels quals existeixen una gran quantitat de llibreries que faciliten tots els processos.

L'editor de codi que es farà servir serà el Visual Studio Code, perquè és amb el que tinc més experiència d'anteriors projectes i no m'ha donat problemes.

S'han estudiat dues llibreries diferents per fer servir com a base per aquest projecte: OpenAI Gym [3] i RL Card [4]. Hem decidit fer servir RL Card, perquè hem considerat que està més ben estructurada i oberta a noves implementacions, fins i tot, s'explica al seu web com començar a implementar un nou entorn [5].

La llibreria RL Card consta de tres mòduls principals (Veure Fig VII):

- **Mòdul Games.** Mòdul que conté l'estructura del joc, seguint un patró perquè tots els jocs de cartes tinguin la mateixa estructura. Les classes que conté aquest mòdul són *Game*, *Round*, *Dealer*, *Judger*, *Player* i *Card*. Això és, en una partida (*Game*), el *Dealer* reparteix les cartes (*Card*) a cada jugador (*Player*). A cada ronda (*Round*) de la partida el *Judger* pren les decisions sobre quines cartes es poden jugar a cada moment de la ronda.

- **Mòdul Environment.** Mòdul encarregat de generar la partida, que la partida tingui els passos corresponents i evolucioni adequadament. Aquest mòdul és essencial perquè els agents puguin jugar.

- **Mòdul Agents.** Mòdul que encarregat d'allotjar els diferents agents que es desenvolupin i posteriorment s'inicialitzin al mòdul *Environment*.

4 PLANIFICACIÓ

A continuació es descriuen les diferents tasques que s'hauran de fer pel desenvolupament del projecte, i tot seguit es fa una planificació setmanal. A l'Annex A, es dona un diagrama de Gantt que permet veure la seqüenciació de totes les tasques.

0. Revisió del programari: Revisar informació sobre els algorismes per crear agents que juguin i llibreries existents...

1. Definició de l'abast del projecte: Amb la informació trobada i les reunions amb el tutor, s'ha de definir l'abast del projecte, objectius, metodologies i realitzar la planificació.

2. Adaptació RLCard per a la Podrida

2.1. Adaptar l'apartat Games: Són les classes més específiques del joc, les que permetran jugar a La Podrida.

2.2. Adaptar l'apartat Environment: Aquesta part, ja es comença a preparar perquè els agents siguin capaços de jugar a la Podrida, generant així un entorn i possibles

accions perquè siguin capaços de preveure els moviments dels rivals.

3. Definir jugadors de La Podrida: En aquesta part es on es desenvoluparan els diferents jugadors.

3.1. Agent Humà: Un agent que permeti que un jugador humà pugui jugar contra la resta d'agents, que en aquest cas serà per a poder fer les primeres proves i perquè es pugui jugar contra els agents intel·ligents.

3.2. Agent Random: Un agent que les seves decisions seran aleatòries dins de les possibles opcions.

3.3. Agent basat en regles (RB). Un agent que les seves decisions les pren segons unes regles predefinides.

3.4. Agent MCTS: Un agent que prengui les decisions basades en l'algorisme MCTS bàsic serà el primer agent intel·ligent a desenvolupar.

4. Avaluació dels agents programats: Aquesta tasca dissenyarà un conjunt d'experiments que permetin avaluar el rendiment de tots els agents implementats a partir de simular partides entre ells.

4.1. Avaluació del RB Agent

4.2. Avaluació inicial dels agents

4.3. Avaluació general dels paràmetres de MCTS

4.4. Refinament dels paràmetres MCTS

4.5. Avaluació de l'agent MCTS amb paràmetres òptims

5. Anàlisi dels resultats i implementació de millores: Aquesta tasca consisteix a analitzar els resultats obtinguts després de jugar les partides, i posteriorment intentar millorar el rendiment dels agents.

La planificació setmanal d'aquestes tasques es dona a la següent taula:

Tasca	Data Inicial	Estimació
0. Revisió del programari	12/09/22	3 set.
1. Definició de l'abast del projecte	28/09/22	1 set.
Entrega Informe Inicial (09/10/2022)		
2. Adaptació RLCard per a la Podrida	10/10/22	3 set.
2.1. Adaptació l'apartat Games	10/10/22	2 set.
2.2. Adaptació l'apartat Environment	24/10/22	1 set.
3. Definició jugadors de La Podrida	31/10/22	5 set.
3.1. Human Agent	31/10/22	3 dies
3.2. Random Agent	03/11/22	3 dies
3.3. RB Agent	06/11/22	1 set.
3.4. MCTS	13/11/22	4 set.
Entrega Informe Seguiment I (13/11/2022)		
4. Avaluació dels agents programats	13/11/22	10 set.
4.1. Avaluació del RB Agent	13/11/22	1 set.
4.2. Avaluació inicial dels agents	12/12/22	1 set.
4.3. Avaluació general dels	19/12/22	4 set.

paràmetres de MCTS		
4.4. Refinament dels paràmetres de MCTS	09/01/23	2 set.
4.5. Avaluació de l'agent MCTS amb paràmetres òptims	23/01/23	2 set.
5. Anàlisi dels resultats i implementació de millores	18/12/22	7 set.
Entrega Informe Seguiment I (18/12/2022)		
Proposta Informe Final (23/01/2023)		
Proposta Presentació (05/02/2023)		
Presentació (23/02/2023)		

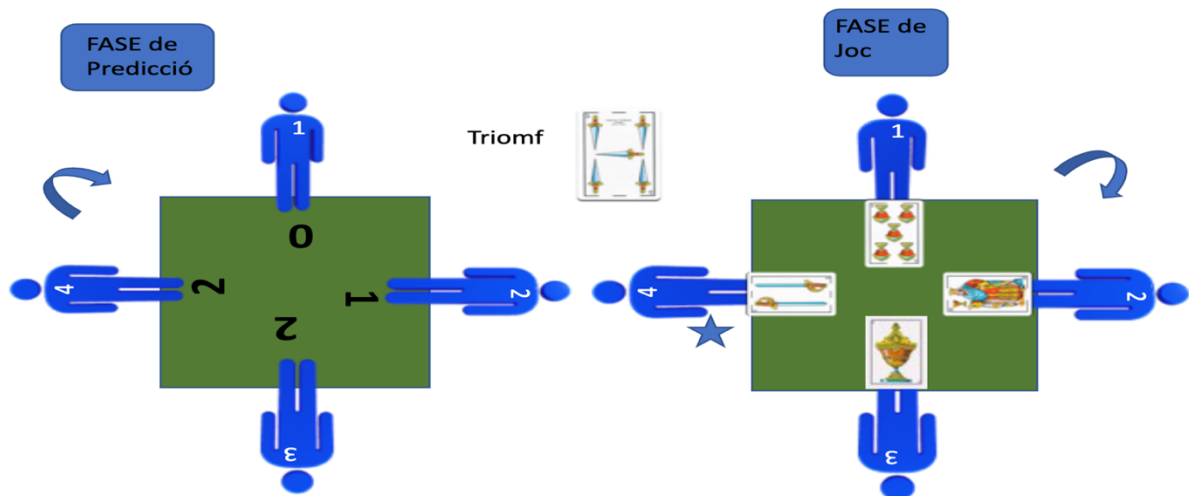
5 EL JOC DE LA PODRIDA

El Joc de La Podrida, també conegut com a “Punyetera” o “La Pocha” és un joc de cartes que habitualment es juga amb la baralla espanyola, tot i que es pot adaptar a la baralla de *poker*.

dotze rondes, si són cinc jugadors, es jugaran nou rondes, si són sis jugadors, es jugaran vuit rondes, etc.

Quan es reparteixen les cartes, també s'ha d'escollir el triomf, que serà el pal dominant de la ronda. Un cop tots els jugadors tenen les cartes corresponents a la ronda, han de predir quantes bases guanyaran de la ronda, a partir de les cartes que tenen a la mà i les prediccions de la resta de jugadors. Cada ronda comença la predicció el jugador que està a l'esquerra del qual ha repartit. A partir de la ronda tres, el nombre total de bases predites no pot coincidir amb el nombre de cartes de la ronda, així es garanteix que, com a mínim, un jugador perd a la ronda, “es podreix”.

Després de les prediccions, es comença a jugar les cartes. El primer jugador a obrir la ronda serà el jugador de la dreta del que ha repartit, i és el que amb la carta que tiri, escollirà el pal a seguir de la basa. Quan un jugador no disposa de cap carta del pal a seguir, ha de fallar amb el triomf de la ronda, si en té, que supera qualsevol carta que no sigui del triomf. Cada jugador està obligat a superar la carta guanyadora de la base en el moment que li toqui tirar, sempre que pugui, excepte si ja hi ha ‘un fallo’ amb triomf i pot



I Diagrama Fases d'una ronda

Tal com s'ha introduït abans, el joc consisteix a encertar el nombre de bases que faràs a cada ronda, de manera que si ho encertes obtindràs una puntuació positiva i si no ho encertes una puntuació negativa. La suma d'aquestes puntuacions és la que determinarà el guanyador de la partida. Cada carta té un valor associat al nombre que conté. L'ordre de major a menor és As, 12, 11, 10..., 3, 2 tot i que hi poden haver variants que li donin més valor alguna carta. El joc està dividit en rondes, on cada ronda se sumaran o restaran punts. A cada ronda es juguen un nombre de bases i en finalitzar es recompten les puntuacions. Una basa és una tirada de cada jugador, sempre seguint les regles del joc. Una ronda consta de dues fases, la fase de la predicció (*Bid*) i la fase de jugar (*Play Card*). El nombre de cartes per ronda va augmentant fins a arribar al nombre màxim que es pugui i després disminueix fins a acabar amb una carta cada jugador. El nombre de rondes que es jugaran es determina amb el nombre de jugadors, es jugaran tantes rondes com les cartes permetin, si són quatre jugadors, es jugaran

seguir el pal, però en general és obligat superar la carta guanyadora. Un cop s'ha acabat la basa, el jugador que ha tirat la carta guanyadora, s'emporta la basa i comença obrint la nova.

Quan han acabat totes les bases de la ronda, es fa el recompte de punts, si has predit correctament sumes deu punts més el nombre de bases que has guanyat, en canvi, si no has encertat la predicció, restaràs deu punts.

Es pot veure un exemple de les dues fases d'una ronda, Predicció i Jugar Carta, a la imatge I.

Al final de la partida guanya el jugador que ha obtingut més punts en total [1].

6 MÈTODE

Com que l'objectiu d'aquest treball és implementar agents intel·ligents que juguin a La Podrida, s'han implementat diversos agents per a poder comparar els resultats i acabar trobant l'agent òptim. S'han implementat aquests quatre agents:

- Jugador Random
- Jugador humà
- Jugador basat en un sistema basat en regles heurístiques
- Jugador basat en una estratègia de MCTS amb Determinització amb diferents paràmetres

Es va considerar d'implementar un jugador basat en l'algorisme MiniMax, però es va descartar perquè el MiniMax requereix la definició d'una heurística que és bastant inviable per a un joc estocàstic i incomplet com és "La Podrida", és a dir que es donen una sèrie de cartes a l'atzar i amagades. El MiniMax és ideal per jocs de taulell amb tota la informació visible, i per a dos adversaris, en què la funció heurística és simètrica a l'hora d'avaluar el jugador i l'adversari [7].

La idea inicial era poder desenvolupar el MCTS amb Reinforcement Learning (RL) per tal d'obtenir millors resultats, de manera més ràpida i jugant millor, però ho vam descartar de seguida perquè el MCTS sense RL tenia suficient complexitat per a ocupar tot el temps del projecte. Es va deixar la millora del RL per a treballs futurs.

6.1 JUGADOR RANDOM

Aquest agent, com el seu nom indica, serà un agent que prendrà les decisions de manera aleatòria.

Servirà bàsicament per poder fer les primeres proves, ja que és un agent molt senzill d'implementar, però també serà el que pitjors resultats s'esperen al no tenir cap mena de criteri per prendre les decisions.

6.2 JUGADOR HUMÀ

El jugador humà és un agent on les decisions sol·licita que les prengui un humà a través d'una interfície.

És l'agent que permet que es puguin jugar partides contra els altres jugadors implementats i valorar els moviments dels contrincants.

6.3 JUGADOR BASAT EN REGLES HEURÍSTIQUES

L'agent basat en regles, es pot veure com un sistema expert, ja que és un sistema que emula la capacitat de prendre decisions d'un humà basant-se en el raonament sobre un coneixement que està representat amb una sèrie de regles basades en l'experiència d'un expert en el joc.

Per tant, és un agent que prendrà la decisió que correspongui fent servir un raonament lògic basant en el Modus Ponens generalitzat [6], a partir de l'estat actual del joc i de les regles que s'han definit i que es reproduïen a l'Annex 3. Aquestes regles han estat dissenyades basant-me en la meua pròpia experiència. Les regles que s'han deduït estan a l'Annex 3.

6.4 JUGADOR BASAT EN MCTS

Aquest agent prendrà les decisions en funció de la probabilitat de guanyar que calculi l'algorisme MCTS adaptat per aquest joc.

El MCTS és un algorisme de cerca per prendre decisions sobre un conjunt finit d'alternatives, a partir de simulacions aleatòries.

Construeix un arbre que a partir de l'estat actual, estima la probabilitat de guanyar de cada un dels fills, és a dir, de cada una de les possibles jugades que es poden fer des de l'estat actual.

Per estimar la probabilitat de guanyar es basa en el resultat de les diferents simulacions aleatòries que farà fins al final del joc a les fulles que es van construint durant la cerca.

El MCTS no té un criteri per aturar les simulacions. Com més simulacions més bona és l'estimació, i per tant més bona és la decisió, però més temps trigarà a tenir una decisió. Per això normalment es limita el nombre d'iteracions que farà a cada presa de decisió.

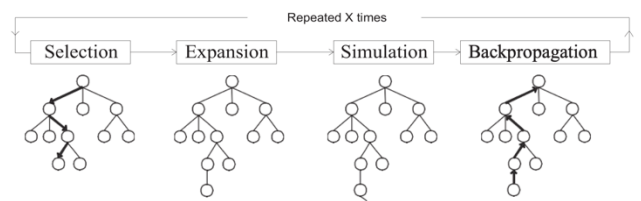
Cada iteració es compon de quatre passos:

-Selecció (*Selection*). A partir de l'arrel, se selecciona el node fill que la política de selecció (*Tree Policy*) li retorni. Aquesta política pren la decisió avaluant entre dos criteris d'explotació i explotació segons una funció de selecció que determina si aprofundim en les millors jugades (explotació) o donem marge a anar a buscar-ne de noves (exploració).

-Expansió (*Expansion*). Quan s'ha seleccionat un node fulla, que ja ha estat visitat, aleshores s'expandeix.

-Simulació (*Simulation*). Quan s'ha seleccionat un node fulla no visitat, aleshores es fa una simulació per inicialitzar la seva probabilitat de guanyar, se simula fins a arribar a un estat final. En les simulacions, les decisions dels altres jugadors s'escullen de manera aleatòria, però també es poden aplicar heurístiques.

-Actualització (*Backpropagation*). Els resultats de la simulació es propaguen des de la fulla fins al node arrel.



II Iteració Monte-Carlo Tree Search

Aquest procés es repetirà tantes vegades com iteracions es vulguin realitzar, com més iteracions es facin, l'algorisme podrà estimar millor els resultats i així escollir la decisió més correcta, però també tardarà més temps a generar l'arbre i prendre una decisió, per això s'ha de trobar un nombre que trobi el balanç entre ser prou gran perquè pugui simular i prou petit perquè sigui viable utilitzar l'algorisme.

L'algorisme *Upper Confidence bounds applied to Trees* (UTC) determina la política de selecció per equilibrar entre explotació i exploració, per decidir si explora nous camins o bé explota els camins ja explorats que han donat millors rendiments. S'ha de trobar un valor òptim que equilibri les exploracions i les explotacions. Els nodes s'avaluen amb la següent fórmula:

$$UCB = \frac{V_i}{n_i} + c * \sqrt{\frac{2 * \ln(N)}{n_i}}$$

on V_i és el nombre de victòries del node que s'està avaluant, n_i és el nombre de vegades que s'ha explorat aquest node i N és el nombre de vegades que s'ha explorat el node pare del qual s'està avaluant. El paràmetre c està predefinit i es pot calcular un valor adequat amb experiments previs. És el que determina si s'exploren més nodes o s'exploten els nodes ja explorats. El valor de la c és entre 0 i 1, com més pròxim a 1 més es prioritza a explorar nous nodes en amplada, és a dir, nodes mai simulats; i com més proper a 0 prioritza les estimacions en profunditat, és a dir, augmentant el nombre de simulacions en els millors nodes.

6.4.1 DETERMINITZACIÓ

El principal problema per aplicar l'algorisme MCTS als jocs de cartes ve del fet que no són jocs d'informació perfecta, no es pot saber les cartes que queden per sortir, ja que estan ocultes a les mans de cada jugador, per tant, no es pot predir de manera realista les accions que prendran els altres jugadors. Perquè el MCTS es pugui aplicar en jocs d'informació imperfecta s'introdueix el terme de la *Determinització*.

La *determinització* complementa l'estat actual amb la informació del conjunt d'estats possibles, convertint el joc en un d'informació perfecta i així poder aplicar algorismes d'informació perfecta, com el MCTS.

En el cas de La Podrida, la *determinització* generarà un cert nombre de diferents alternatives sobre les mans dels oponents. Aquest nombre de determinitzacions el representarem amb el paràmetre X , i les determinitzacions les representarem amb el conjunt $\{D_1, D_2, \dots, D_X\}$, amb això ja tindrem la informació completa i aleshores ja es pot aplicar el MCTS a cada una de les alternatives d'informació completa. Per a cada determinització el MCTS retornarà una estimació de la probabilitat de guanyar de cada una de les possibles jugades, que representem com $\{J_1, \dots, J_K\}$, per tant, el retornarà la millor jugada d'acord amb totes les estimacions:

$$MCTS(E, D_j) = \operatorname{argmax}_{i:1..K} \{Prob_{guanyar}(J_i)\}$$

Per a tenir una millor aproximació de l'estat real, el que es fa és reduir el domini de possibles cartes per a cada jugador, fixant-se amb les cartes que ja han sortit sobre el taulell, o si un jugador ha fallat d'un pal, o fins i tot si un jugador no ha pogut superar una carta en algun moment de la ronda.

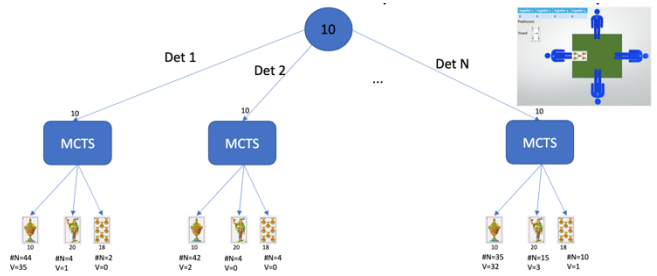
Un cop s'ha aplicat el MCTS a totes les determinitzacions cal combinar els resultats i seleccionar la millor jugada de totes, això es fa prioritzant la jugada, J_i que ha estat seleccionada més vegades ple MCTS, això és

$$J = \operatorname{argmax}_{m:1..K} \left\{ \sum_{j=1}^N \partial(J_m, MCTS(E, D_j)) \right\}$$

$$\text{on } \partial(J_i, J_j) = \begin{cases} 1 & \text{si } J_i = J_j \\ 0 & \text{en altre cas} \end{cases}$$

S'han implementat tres diferents mètodes de determinització:

- Un primer mètode que completa l'estat de manera aleatòria, sense tenir en compte res sobre l'estat actual del joc. ('*SampleRandomHands*')
- El segon mètode, completa l'estat a partir de les cartes que encara no estan sobre el taulell. ('*NotUsedCards*')
- L'última variant es limita les possibles cartes per a cada jugador, a partir de les cartes que ja han aparegut i també fixant-se amb les cartes que ha jugat cada jugador, és a dir, es té en compte els pals que ha fallat el jugador, essent la variant de la determinització més realista. ('*FailedSuitsHands*')



III Esquema MCTS + Determinització

7 DESENVOLUPAMENT

En aquest apartat s'exposaran els diferents problemes apareguts durant el desenvolupament i solucions amb els retards sobre la planificació inicial que han implicat.

El primer inconvenient que es va trobar va ser a l'hora d'importar els mòduls per poder utilitzar-los entre ells. Va ser una qüestió amb les rutes de cada carpeta, perquè l'editor Visual Studio Code no les identificava correctament i no trobava les classes, però es va resoldre amb la llibreria 'sys', inserint cada ruta abans d'importar els mòduls. Va tenir un impacte a la planificació amb un retard d'una setmana.

A l'hora de dissenyar el jugador humà també hi va haver una sèrie d'inconvenients, perquè es va haver de dissenyar una interfície gràfica que es mostrés la informació de manera clara, però que a la vegada fos senzilla d'implementar. Al no tenir-ho previst a la planificació, va tenir un impacte d'una setmana més de retard.

Un cop es van implementar el jugador humà i aleatori, es va decidir modificar els objectius i afegir un agent basat en regles heurístiques, cosa que lògicament va afegir una reestructuració a la planificació inicial, ja que no s'havia tingut en compte, disminuint el temps per implementar millores dels agents.

Es va decidir implementar aquest jugador per diversos motius, el motiu principal va ser que creiem que ens podria ser útil més endavant a l'hora d'implementar el MCTS, també per familiaritzar-se més amb l'entorn de la llibreria i per poder tenir un agent més amb el que fer proves. Un cop introduïts els canvis esmentats, la reestructuració final de la planificació es va continuar sense retards i sense cap problema.

8 RESULTATS

Per avaluar els agents que s'han anat desenvolupant s'han dissenyat i fent diferents grups d'experiments:

- Experiment 1. Avaluació de l'agent basat en regles
- Experiment 2. Avaluació inicial de tots els agents: human, aleatori, RB i MCTS
- Experiment 3. Avaluació general dels paràmetres del MCTS
- Experiment 4. Refinament dels paràmetres del MCTS.
- Experiment 5. Avaluació final de l'agent MCTS amb paràmetres òptims.

Els resultats per a cada un d'aquests experiments s'expliquen a les següents seccions.

8.1 AVALUACIÓ DE L'AGENT BASAT EN REGLES

En aquesta avaluació es van realitzar 100 partides entre un agent basat en regles i 3 agents randoms. Els resultats d'aquesta avaluació serviran per comprovar que l'agent basat en regles jugui millor que els agents aleatoris, tot i que els resultats poden no ser molt bons.

% Victòria	Punts Mitjana	Temps per partida
96%	-2,51	0.038s

Taula 1 Avaluació RB Agent

A la Taula 1 s'observen els resultats obtinguts en aquest experiment. Tal com es pot observar, l'agent basat en regles heurístiques va obtenir un percentatge de victòries molt elevat, però una mitjana de punts en negatiu, cosa que vol dir que va guanyar moltes partides amb la puntuació negativa. Aquest fet és normal perquè els agents aleatoris no tenen en compte l'estat de la partida i adulteren els criteris de les regles en què es basa l'agent.

En resum, l'agent basat en regles funciona correctament perquè és capaç de guanyar en un 96% de les vegades a agents aleatoris.

8.2 AVALUACIÓ INICIAL DE TOTS ELS AGENTS: HUMAN, ALEATORI, RB I MCTS

Un cop es va finalitzar la implementació del mètode MCTS es van avaluar els quatre agents implementats de manera senzilla per a poder obtenir les primeres conclusions i comprovar que estaven correctament desenvolupats.

	Human vs 3 Random		RB vs 3 Random		MCTS vs 3 Random		MCTS vs RB vs 2 Random	
%Victòria	100 %		96%		60%		20 %	80%
Pts. Mitjana	120	-121	-2.5	-137	-45	-156	-78	-34
Temps per partida	10 min		0.038s		5min		5 min	

Taula 2 Avaluació Inicial de tots els agents

A la Taula 2 s'hi veuen reflectits els resultats. Després de realitzar les primeres avaluacions dels agents la primera reflexió que es pot treure és que el millor agent dels 4 és el RB, que com havíem vist va guanyar un 96% de les partides contra 3 agents Random i el 80% contra el MCTS i 2 agents Random.

Les proves que es van realitzar amb el Human Agent eren bàsicament per comprovar que el joc funcionava correctament i que no es podien fer il·legalitats, però que es guanyessin el 100% de les partides no és interessant.

Pel jugador MCTS, s'han utilitzat els paràmetres (iteracions = 20, determinitzacions=10, c=1), però aquests paràmetres necessitaven ser ajustats per obtenir resultats i temps més òptims..

Per a totes les avaluacions del MCTS, es va utilitzar la determinització de 'FailedSuitsHands', ja que és la més realista.

Els altres dos mètodes de determinització no s'han avaluat perquè es considera que els resultats serien massa irrealistes, ja que les simulacions no fan servir informació disponible pel jugador MCTS, i per tant, degrada el seu rendiment.

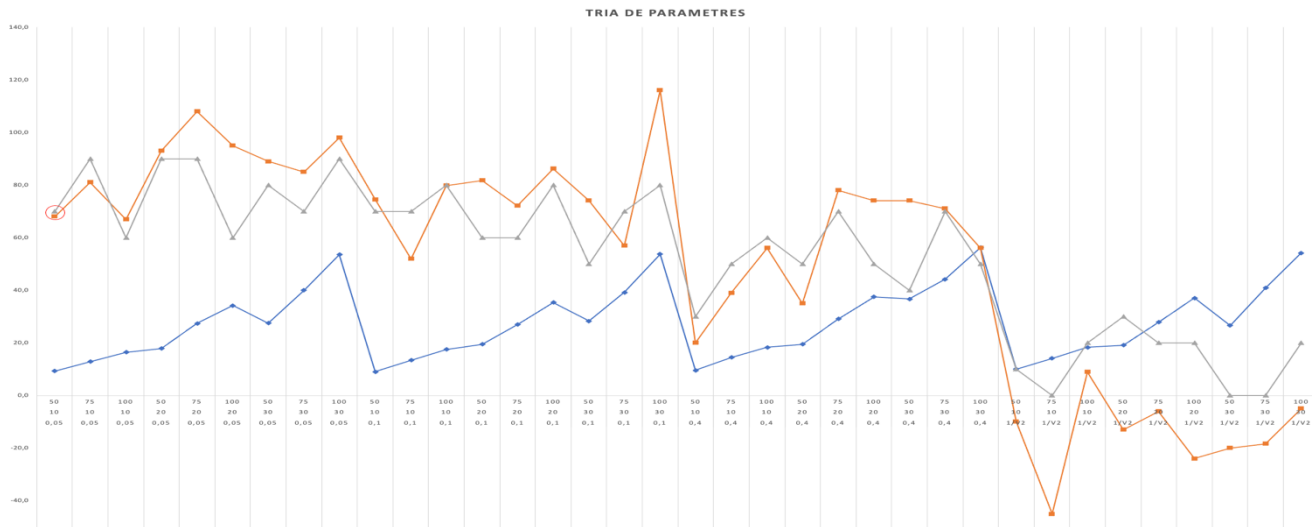
Com es veu a la Taula 2, els resultats del MCTS són molt pobres per estar parlant d'un algorisme com el MCTS, això és perquè no s'han ajustat els paràmetres, es van agafar uns valors a l'atzar. Els experiments de les següents seccions van dirigits a trobar els millors paràmetres.

8.3 AVALUACIÓ GENERAL DELS PARÀMETRES DEL MCTS

Calia fer primer una avaluació general de les diferents combinacions de paràmetres per l'algorisme MCTS per tal de millorar els resultats de l'agent.

Per a realitzar aquestes proves s'ha fet jugar l'agent MCTS contra 3 agents basats en regles.

A continuació, a la gràfica IV, es mostra un gràfic amb les diferents combinacions de paràmetres que s'han provat. A l'eix Y posem el % de victòries de l'agent MCTS (calculat sobre 10 partides) per la combinació de paràmetres indicats a l'eix de les X. En aquest gràfic es mostra el temps (blau), el % de victòries (gris) i la puntuació mitjana (taronja) que s'ha obtingut amb cada combinació de paràmetres.



IV Gràfica Refinament dels paràmetres de l'MCTS

Es pot apreciar que a mesura que s'augmenta el paràmetre *c* els resultats van empitjorant. Augmentar aquest paràmetre significa donar més pes a l'expansió nous nodes. També es va variant els paràmetres de nombre d'iteracions i de nombre de determinitzacions, on es pot apreciar que els resultats milloren, a mesura que augmentem el temps. Per tant caldrà trobar un compromís entre aquests dos paràmetres.

Es pot veure com la línia blava té una evolució repetida quan augmenta el nombre d'iteracions i determinacions. Les línies taronja i gris comencen amb uns valors elevats, que es consideren molt bons resultats, i es veu com van disminuint, relacionat amb l'augment del paràmetre *c*.

El cercle vermell mostra el punt dels paràmetres que s'han escollit com a definitius, els quals donen uns resultats molt bons i el temps no s'excedeix. (*c*=0,05; *it*=50; *det*=10)

8.4 REFINAMENT DELS PARÀMETRES DEL MCTS

En aquesta avaluació l'objectiu era refinar encara més els paràmetres del MCTS, per això es va comparar com els agents MCTS competien entre ells, cada un amb diferents paràmetres.

8.4.1 REFINAMENT PARÀMETRE 'c'

Com ja s'ha comentat anteriorment, el paràmetre '*c*' és el que té més impacte sobre els resultats de l'agent MCTS, per això és el que es va refinar primer.

c	0,05	0,25	0,5	0,75
% victòria	63%	31%	6%	0%
Pts Mitjana	74	47	-16	-45

Taula 3 1er Refinament paràmetre 'c'

En aquesta taula es pot veure el que ja es va veure en l'anterior experiment, com més petit és el paràmetre '*c*', millors resultats dona, és a dir, prioritant l'explotació. Els temps d'execució de cada partida rondaven els 40 minuts.

Veient els resultats de la Taula 3, es va voler refinar més el paràmetre, posant com a valors límits el 0,05 i el 0,025. A la Taula 4 es mostren els resultats obtinguts.

c	0,05	0,1	0,15	0,25
% victòria	19%	56%	12,5	12,5%
Pts. Mitjana	64	73	45	25

Taula 4 2n Refinament paràmetre 'c'

En aquesta última taula els resultats estan més ajustats pel que fa a puntuacions mitjanes, però pel que fa al percentatge de victòries, el que es pot veure és que el 0,1 és el paràmetre que més partides ha guanyat.

Els altres dos paràmetres utilitzats per aquestes proves han sigut els escollits a l'apartat anterior.

Per tant, deixem fixat el *c*=0,1 per a les següents avaluacions.

8.4.2 REFINAMENT PARÀMETRE IT

Aquest paràmetre és el que fa alentir el temps de les partides perquè com més gran sigui, més iteracions farà l'algorisme MCTS, per això és important que sigui el més petit dins els que donen resultats acceptables.

it	20	30	40	50
% victòria	10%	30%	40%	20%
Pts. Mitjana	78	87	92	67

Taula 5 Refinament paràmetre it

A la Taula 5 es poden observar els resultats. D'aquesta taula es pot extreure que amb 40 iteracions en tenim prou per obtenir uns resultats correctes, no vol dir que sigui millor que fer-ne 50, però sí que es suficient i així també disminueix el temps d'execució de les partides.

Cal comentar que els guanyadors de les partides estan força repartits, així es pot veure la importància del paràmetre '*c*', que és el mateix en els quatre agents.

8.4.3 REFINAMENT PARÀMETRE DET

L'últim paràmetre a fixar és el nombre de determinitzacions, que és el paràmetre que menys impacte té sobre els resultats finals, ja que només afegeix aleatorietat, com més gran sigui més determinitzacions aleatòries es faran en cada decisió. També fa que augmenti el temps.

ndet	10	20	30	40
% victòria	10%	0%	45%	45%
Pts Mitjana	44	65	103	93

Taula 6 Refinament paràmetre ndet

Així doncs, a la Taula 6 s'hi poden veure els resultats. En aquesta taula es pot veure que els resultats són millors si el nombre de determinitzacions és més elevat, però al no disposar de suficients recursos, es considera que amb 10 determinitzacions és suficient per obtenir bons resultats.

En resum, s'ha pogut veure i concloure que la combinació de paràmetres definitiva és ($c=0.1$; $it=40$; $ndet=10$).

8.5 AVALUACIÓ DE L'AGENT MCTS AMB PARÀMETRES ÒPTIMS

Una vegada fixats els paràmetres pel MCTS, s'ha avaluat l'agent contra tres agents RB i contra un jugador humà. Per avaluar l'agent MCTS amb els paràmetres fixats, s'han fet una sèrie de partides, on es poden veure els resultats a la següent taula.

	MCTS	RB
% victòria	74%	26%
Pts. Mitjana	92	42

Taula 7 MCTS òptim vs RB

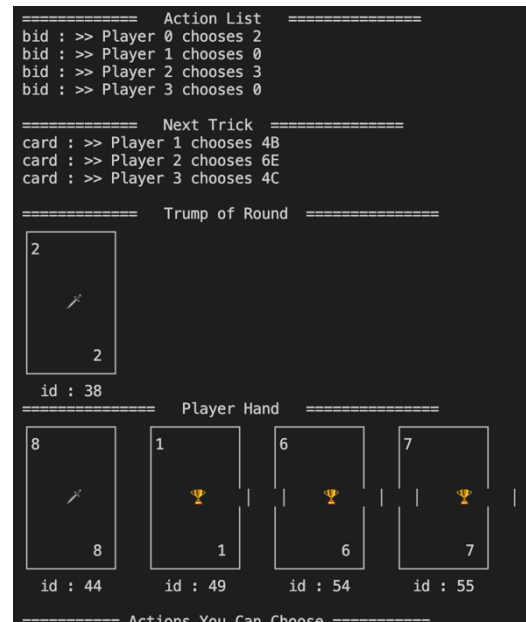
Com es pot veure a la taula, després de jugar multitud de partides de l'agent MCTS enfront d'agents basats en regles, el que guanya la gran part de partides és l'agent MCTS, fet que reforça les hipòtesis que l'algorisme MCTS era una bona opció per a La Podrida.

S'ha fet una partida d'un jugador humà en vers tres agents MCTS amb els paràmetres fixats anteriorment. El resultat final és que el jugador humà ha guanyat la partida [90 63 63 84].

A la il·lustració V, es pot veure la interfície gràfica de l'agent humà, on es mostren les cartes que resten a la mà, les accions possibles i la llista de jugades de cada basa.

Com es pot veure ha estat molt igualada, i les puntuacions que s'han obtingut han sigut força altes. Pel que fa al temps, els agents MCTS tarden 5 segons de mitjana en prendre una decisió, essent a les rondes amb més cartes més lentes i les rondes amb menys cartes més ràpides.

En conclusió, s'ha assolit una partida realista i de nivell.



V Interfície Agent Humà

9 CONCLUSIONS

Es dividiran les conclusions en tres parts: sobre la metodologia, sobre els resultats, i sobre les competències adquirides.

Com a conclusions sobre la metodologia escollida s'ha de dir que va ser un gran encert escollir la llibreria RL Card, ja que m'ha donat moltes facilitats a l'hora de desenvolupar el joc i posteriorment poder crear els agents i que sigui senzill de fer-los jugar.

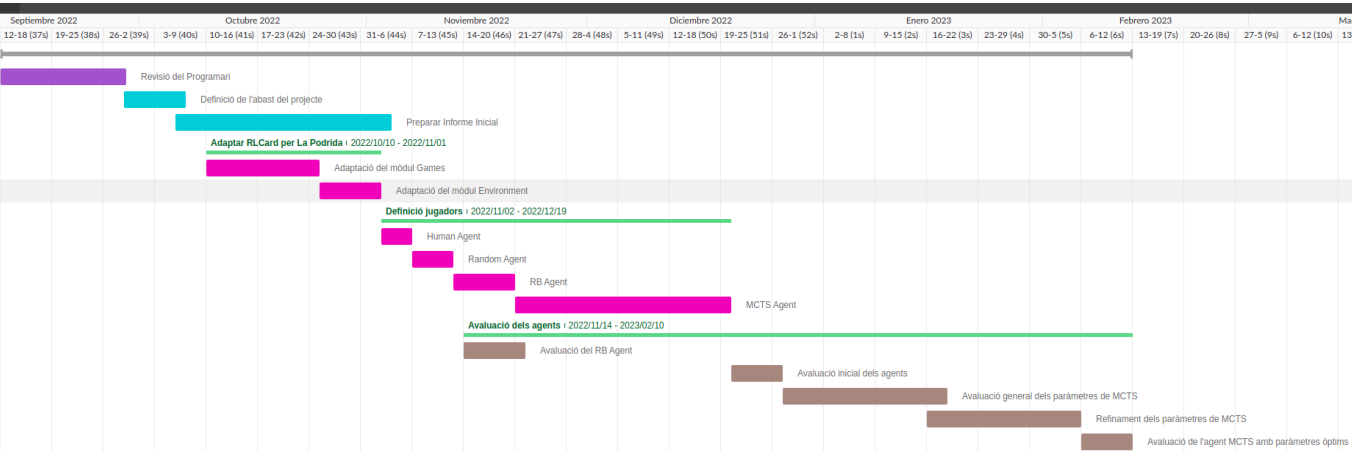
Pel que fa als resultats obtinguts, es pot dir que s'està molt orgullós de l'agent basat en regles, perquè ha obtingut molt bons resultats i juga les partides de manera molt ràpida. Per la part de l'agent MCTS, s'ha vist que els paràmetres influeixen molt en els resultats, perquè al principi es veia que aquest agent no donava els resultats esperats, però una vegada es van fixar els paràmetres, es podria dir que els resultats assolits són excel·lents. Com a conclusió final podem dir que els agents MCTS juguen d'una manera molt realista, complicant les decisions a prendre.

Per acabar, vull concloure que aquest projecte m'ha servit per adquirir experiència a l'hora de gestionar i desenvolupar un projecte individual, així com prendre decisions en moments crítics per falta de temps o de recursos. Sobretot la part de l'RL Card, que ha sigut una molt bona experiència i penso que em pot servir per a poder desplegar algun entorn d'algun altre joc o bé afegir algun agent per a La Podrida.

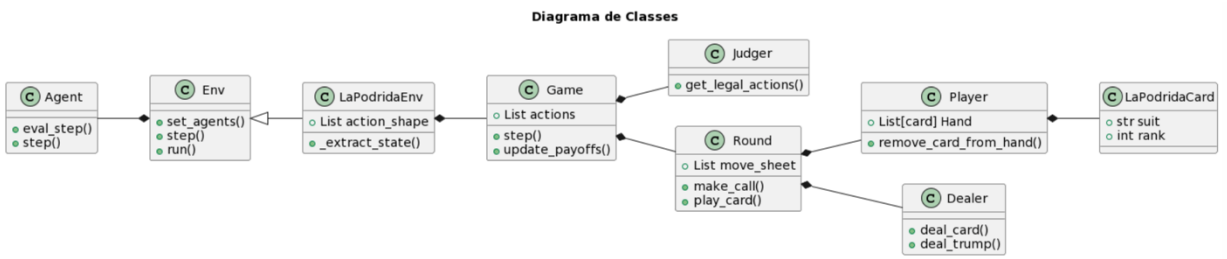
BIBLIOGRAFIA

- [1] ConectaGames "Reglas de La Podrida". <https://conecta-gamesblog.wordpress.com/category/la-podrida/>
- [2] Pàgina web RLCard, "RL Card: A Toolkit for Reinforcement Learning in Card Games". <https://rlcard.org/index.html>
- [3] OpenAI GYM. "Gym documentation". <https://www.gymnasium.dev>
- [4] Repositori GitHub RL Card. <https://github.com/damianlab/rlcard>
- [5] RL Card vs GYM. "Compare RL Card vs Gym and see what are their differences". <https://www.libhunt.com/compare-rlcard-vs-gym>
- [6] Reglas de Inferencia. Universidad Politécnica Madrid. <https://dit.upm.es/~gfer/ssii/rcsi/rcsisu28.html>
- [7] Determinization. "Determinization with MCTS for the card game Hearts." https://studenttheses.uu.nl/bitstream/handle/20.500.12932/37736/Thesis_draft.pdf?sequence=1&isAllowed=y
- [8] Inteligencia Artificial. Departamento de Lenguajes y Sistemas Informáticos UPC. "IA – Algoritmos de Juegos" https://www.cs.upc.edu/~bejar/ia/material/trabajos/Algoritmos_Juegos.pdf

10 ANNEXOS



VI Diagrama de Gantt amb les tasques



VII Diagrama de classes

BASE DE REGLES.

FETS sobre l'estat del jugador:

- f1: `type_action('bid')` : el jugador està a la fase de predicció de la ronda.
- f2: `have_trumps`: el jugador té un o més cartes del pal del triomf.
- f3: ronda (1-6): el jugador està en una ronda entre la 1 i la 6 (incloses).
- f4: `legal_bid(num_trumps)`: el jugador pot fer la predicció coincidint amb el nombre de triomfs que té.
- f5: `have_better_trump_cards`: el jugador 1 o 12 del pal del triomf.
- f6: `have_good_cards`: el jugador té 1 o 12 d'algun pal que no és el triomf.
- f7: `legal_bid(num(good_cards)+num(good_trump_cards))`: el jugador pot fer la predicció coincidint amb el nombre de triomfs bons que té més amb el nombre de cartes bones que té.
- f8: ronda(5-7): el jugador està en una ronda entre la 5 i la 7 (incloses).
- f9: `bids_at_moment(<2)`: Fins al moment, només s'han predit 1 o cap basa.
- f10: `legal_bid(num(good_cards)+num(good_trump_cards)+1)`: el jugador pot fer la predicció coincidint amb el nombre de triomfs bons que té més amb el nombre de cartes bones que té més un.
- f11: `legal_bid(num(good_cards)+num(good_trump_cards)-1)`: el jugador pot fer la predicció coincidint amb el nombre de triomfs bons que té més amb el nombre de cartes bones que té menys un.
- f12: ronda (1-4): el jugador està en una ronda entre la 1 i la 4 (incloses).
- f13: `legal_bid(num(good_cards))`: El jugador pot fer la predicció coincidint nombre de cartes bones que té.
- f14: ronda (8-12): el jugador està en una ronda entre la 8 i la 12 (incloses).
- f15: `bids_at_moment(< (num_round - 4))`: Fins al moment, s'han predit menys bases que el nombre de la ronda menys quatre.
- f16: `num_bids(>2)`: Ja han predit mínim dos jugadors.
- f17: `legal_bid(2)`: El jugador pot predir 2 bases.
- f18: `legal_bid(1)`: El jugador pot predir 1 basa.
- f19: `type_action('card')`: el jugador està a la fase de jugar carta de la ronda.
- f20: `num_legal_actions(1)`: el jugador només té una acció possible a realitzar.
- f21: `want_win_trick`: el jugador necessita guanyar mínim una basa més per encertar la predicció.
- f22: `can_win_trick`: el jugador encara pot guanyar la basa amb les cartes que pot tirar.
- f23: `bids_correct`: el jugador ha guanyat el nombre de bases que havia predit.

FETS sobre accions que pot fer el jugador:

- a1: `return (num_trumps)`: el jugador retorna el nombre de triomfs que té.
- a2: `return (num_trumps + num(better_trump_cards))`: el jugador retorna el nombre de triomfs que té més el nombre dels millors triomfs.

- a3: `return (num_trumps - 1)`: el jugador retorna el nombre de triomfs que té menys un.
- a4: `return (num(good_trump_cards) + num(good_cards))`: el jugador retorna el nombre de triomfs bons que té més el nombre de cartes bones.
- a5: `return (num(good_cards)+num(good_trump_cards)+1)`: el jugador retorna el nombre de triomfs bons que té més el nombre de cartes bones més el nombre de triomfs bons que té més un.
- a6: `return (num(good_cards)+num(good_trump_cards) - 1)`: el jugador retorna el nombre de triomfs bons que té més el nombre de cartes bones més el nombre de triomfs bons que té menys un.
- a7: `return (num(good_cards)+num(good_trump_cards) + num(better_trump_cards))`: el jugador retorna el nombre de triomfs bons que té més el nombre de cartes bones més el nombre de triomfs bons que té.
- a8: `return 0`: el jugador retorna 0 com a predicció.
- a9: `return (num(good_cards))`: el jugador retorna el nombre de cartes bones que té.
- a10: `return (num(good_cards) + 1)`: el jugador retorna el nombre de cartes bones que té més un.
- a11: `return (num(good_cards) - 1)`: el jugador retorna el nombre de cartes bones que té menys un.
- a12: `return 2`: el jugador retorna 2 com a predicció.
- a13: `return 1`: el jugador retorna 1 com a predicció.
- a14: `return(legal_actions[0])`: retorna l'única acció possible a realitzar.
- a15: `return(best_card(legal_actions))`: retorna la millor acció possible a realitzar.
- a16: `return(worst_card(legal_actions))`: retorna la pitjor acció possible a realitzar.

REGLES

Mòdul 1: Regles referents a la fase de predicció

R1: $f1 \wedge f2 \wedge f3 \wedge f4 \rightarrow a1$

R2: $f1 \wedge f2 \wedge f3 \wedge \text{no}(f4) \wedge f5 \rightarrow a2$

R3: $f1 \wedge f2 \wedge f3 \wedge \text{no}(f4) \wedge \text{no}(f5) \rightarrow a3$

R4: $f1 \wedge f2 \wedge \text{no}(f3) \wedge f6 \wedge f7 \rightarrow a4$

R5: $f1 \wedge f2 \wedge \text{no}(f3) \wedge f6 \wedge \text{no}(f7) \wedge f8 \wedge f9 \wedge f10 \rightarrow a5$

R6: $f1 \wedge f2 \wedge \text{no}(f3) \wedge f6 \wedge \text{no}(f7) \wedge f8 \wedge \text{no}(f9) \wedge f11 \rightarrow a6$

R7: $f1 \wedge f2 \wedge \text{no}(f3) \wedge f5 \wedge f6 \wedge \text{no}(f7) \wedge f8 \wedge \text{no}(f9) \wedge \text{no}(f11) \rightarrow a7$

R8: $f1 \wedge f2 \wedge \text{no}(f3) \wedge \text{no}(f5) \wedge f6 \wedge \text{no}(f7) \wedge f8 \wedge \text{no}(f9) \wedge \text{no}(f11) \rightarrow a6$

R9: $f1 \wedge \text{no}(f2) \wedge f12 \rightarrow a8$

R10: $f1 \wedge \text{no}(f2) \wedge f6 \wedge \text{no}(f12) \wedge f13 \rightarrow a9$

R11: $f1 \wedge \text{no}(f2) \wedge f6 \wedge f8 \wedge f9 \wedge \text{no}(f12) \wedge \text{no}(f13) \rightarrow a10$

R12: $f1 \wedge \text{no}(f2) \wedge f6 \wedge f8 \wedge \text{no}(f9) \wedge \text{no}(f12) \wedge \text{no}(f13) \rightarrow a11$

R13: $f1 \wedge \text{no}(f2) \wedge f6 \wedge \text{no}(f8) \wedge \text{no}(f9) \wedge \text{no}(f12) \wedge \text{no}(f13) \rightarrow a10$

R14: $f1 \wedge \text{no}(f2) \wedge \text{no}(f6) \wedge \text{no}(f12) \wedge f14 \wedge f15 \wedge f16 \wedge f17 \rightarrow a12$

R15: $f1 \wedge \text{no}(f2) \wedge \text{no}(f6) \wedge \text{no}(f12) \wedge f14 \wedge f15 \wedge f16 \wedge \text{no}(f17) \rightarrow a13$

R16: $f1 \wedge \text{no}(f2) \wedge \text{no}(f6) \wedge \text{no}(f12) \wedge f14 \wedge \text{no}(f15) \wedge f16 \wedge f18 \rightarrow a13$

R17: $f1 \wedge \text{no}(f2) \wedge \text{no}(f6) \wedge \text{no}(f12) \wedge f14 \wedge \text{no}(f15) \wedge f16 \wedge f18 \rightarrow a8$

R18: $f1 \wedge \text{no}(f2) \wedge \text{no}(f6) \wedge \text{no}(f12) \wedge \text{no}(f14) \rightarrow a8$

Módul 2: Regles referents a la fase de jugar carta

R19: $f19 \wedge f20 \rightarrow a14$

R20: $f19 \wedge \text{no}(f20) \wedge f21 \wedge f22 \rightarrow a15$

R21: $f19 \wedge \text{no}(f20) \wedge f21 \wedge \text{no}(f22) \rightarrow a16$

R22: $f19 \wedge \text{no}(f20) \wedge \text{no}(f21) \wedge f22 \wedge f23 \rightarrow a16$

R23: $f19 \wedge \text{no}(f20) \wedge \text{no}(f21) \wedge \text{no}(f22) \wedge f23 \rightarrow a15$

R24: $f19 \wedge \text{no}(f20) \wedge \text{no}(f21) \wedge f22 \wedge \text{no}(f23) \rightarrow a15$

R25: $f19 \wedge \text{no}(f20) \wedge \text{no}(f21) \wedge \text{no}(f22) \wedge \text{no}(f23) \rightarrow a16$