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

Reyes Marti, Pol; Bernal del Nozal, Jorge, dir. Definició de perfils d'agents intel·ligents per simular diferents estratègies per al joc Toma 6!. 2024. (Grau en Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/298941>

under the terms of the  license

Definició de perfils d'agents intel·ligents per simular diferents estratègies per al joc Toma 6!

Pol Reyes Martí

Resum— Aquest treball se centra en el desenvolupament i la informatització del joc de cartes Toma 6! i la creació d'agents intel·ligents capaços de jugar-hi. El Toma 6! és un joc de cartes que té per objectiu evitar acumular cartes amb caps de bou (punts negatius). La partida s'acaba quan un dels jugadors arriba a 66 punts negatius, i guanya qui en tingui menys. Per desenvolupar els agents intel·ligents, s'han implementat diversos algorismes automàtics per a les accions dels jugadors de tirar carta i de seleccionar fila. S'han creat algorismes que comprenen l'estat del tauler i les cartes a la mà dels jugadors i selecciona la carta adient segons el seu propòsit. Alguns d'aquests algorismes també consideren les cartes que ja s'han jugat i les que queden per jugar, oferint així una estratègia més informada. Aquestes funcionalitats han estat dissenyades per permetre la realització de partides automàtiques controlades per aquests algorismes. Finalment, s'han analitzat els algorismes en partides amb 2, 3, 4 i 5 jugadors per avaluar-ne i comparar-ne el rendiment en diverses mètriques, com poden ser el percentatge de partides guanyades i perdudes, els punts negatius aconseguits o el nombre de rondes que dura la partida.

Paraules clau— Informatització, Agent intel·ligent, Algorismes, Estratègia, Avaluació de mètriques, Comparació de resultats

Abstract— This work focuses on the development and computerization of the card game Toma 6! and the creation of intelligent agents capable of playing it. Toma 6! is a card game where the objective is to avoid accumulating cards with bull heads (negative points). The game ends when one of the players reaches 66 negative points, and the winner is the one with the fewest points. To develop the intelligent agents, various automatic algorithms have been implemented for the players' actions of playing a card and selecting a row. Algorithms have been created that understand the state of the board and the cards in the players' hands, selecting the appropriate card according to their purpose. Some of these algorithms also consider the cards that have already been played and those that remain to be played, thus offering a more informed strategy. These functionalities have been designed to enable automated games controlled by these algorithms. Finally, the algorithms have been analyzed in games with 2, 3, 4, and 5 players to evaluate and compare their performance across various metrics, such as the percentage of games won and lost, negative points accumulated, and the number of rounds the game lasts.

Index Terms—Computerization, Intelligent Agent, Algorithms, Strategy, Metrics Evaluation, Results Comparison



1 INTRODUCCIÓ

ACTUALMENT, el món dels jocs de taula continua ben arrelat en totes les franges d'edat de la societat, evidenciant una fascinació duradora per les experiències dels jocs de taula. No obstant això, els videojocs han consolidat la seva presència a la vida de molta gent, tant petits com grans. Aquesta presència ha fet que diferents jocs de taula coneguts hagin acabat adoptant també una adaptació informàtica. Aquesta adaptació digital pot ser tant a través d'una interfície en la qual puguin jugar diferents jugadors sense connexió, com una interfície on juguin en línia diferents jugadors connectats, com també un o diversos jugadors reals jugant contra un o diversos jugadors virtuals.

En aquest treball de final de grau es vol crear un agent intel·ligent del joc de taula Toma 6! [1] que sigui capaç de jugar, i guanyar, al joc de taula simulant la ment humana. I, per tant, fer possible jugar contra 1 o diversos contrincants anomenats comunament com a màquines, per continuar fent valdre els jocs de taula també en l'àmbit digital i, d'aquesta manera, donar-los un espai dins del món dels videojocs; així com intentar resoldre i millorar les estratègies del joc

1.1 Objectiu i preparació del joc

El joc de taula consta d'unes cartes numerades de l'1 al 104, on cada carta té un nombre de bous (punts negatius assignats a la carta) determinat. El joc està pensat perquè hi puguin jugar de 2 a 10 participants.

La finalitat del joc és obtenir el menor nombre de bous possibles. La partida finalitza quan un dels participants hagi acumulat 66 o més bous.

- E-mail de contacte: pol.reyes@autonoma.cat
- Menció realitzada: Computació
- Treball tutoritzat per: Jorge Bernal del Nozal (Ciències de la computació)
- Curs 2023/24

Guanya el jugador que al final de la partida tingui menys bous en la seva possessió, és a dir, qui tingui menys punts negatius.

Per iniciar el joc es barregen les cartes. Depenent del nombre de jugadors s'utilitzaran més o menys cartes. Per exemple: Si a la partida hi ha 2 jugadors, s'utilitzaran les cartes de l'1 al 24; si a la partida hi ha 5 jugadors, s'utilitzaran les cartes de l'1 al 54; Sempre seguint l'interval de cartes $[1, \text{numJugadors} \cdot 10 + 4]$ en els nombres enters.

Seguidament, es reparteixen 10 cartes aleatòriament a cada jugador, i es posen les 4 cartes restants cara amunt a sobre la taula. Cadascuna de les 4 cartes formarà part del principi d'una fila, les quals podran tenir com a màxim 5 cartes cadascuna. Les 4 files conformaran el taulell del joc. Aquestes files sempre seguiran un ordre numèric ascendent i les cartes noves sempre es col·locaran a la fila que contingui el valor més proper a la carta que s'està jugant.

1.2 Idea i Estat de l'Art

Aquest projecte neix de la idea d'informatitzar els diferents jocs clàssics o jocs de taula per poder-hi accedir digitalment, i també de la idea d'intentar programar agents informàtics que siguin capaços de competir, i fins i tot guanyar, als diferents jocs àmpliament coneguts com ho podria fer un humà.

El joc del Toma 6! és un joc de taula en el qual juguen dos o més jugadors, que implica informació oculta (les cartes dels altres jugadors), i que és difícil de dominar pels humans perquè té un 50% de sort i un 50% d'estratègia.

Introduïdes aquestes característiques del joc de taula, com comenten els autors [2], es poden diferenciar dues branques en els jocs com són els Jocs d'Informació Perfecta (PIG en anglès) - on els jugadors coneixen en cada moment l'estat del joc - i els Jocs d'Informació Imperfeta (IIG en anglès) - on els jugadors tenen informació desconeguda-.

Els autors comenten que en la primera branca, la intel·ligència artificial guanya per sobre dels jugadors, ja que pot obtenir més informació tant actual com de partides anteriors, com pot ser en el cas dels escacs o el GO. Per altra banda, a la branca de l'IIG, actualment, encara hi continua havent una petita bretxa entre la capacitat per guanyar de la IA envers els humans; tot i que cada vegada hi ha més avenços que indiquen que es podria millorar la situació de la IA en aquest tipus de jocs.

De fet, es poden extrapolar aquests tipus de jocs que tenen una certa informació desconeguda a tota mena de situacions, sigui en negociacions, negocis o operacions quirúrgiques, com també en jocs de cartes, que poden ser molt diversos i amb necessitats ben diferents. Per això, proposen diversos mètodes més complexos per resoldre aquests problemes.

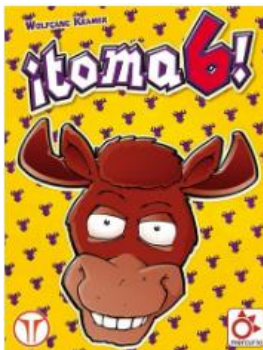


Fig. 1. Portada del joc Toma 6!

Segons [2] es poden considerar dos tipus de joc: jocs de forma extensiva, que implica una seqüència d'estats del joc, i els jocs de coordinació, que implica una coordinació entre els jugadors per completar una acció. En el cas del joc Toma 6!, es considera del tipus de joc de forma extensiva, però també cal tenir en compte que hi ha una certa coordinació/competència entre els jugadors per les cartes que es comparteixen a sobre el taulell.

Per avaluar el rendiment de la intel·ligència artificial es pot considerar que quan dos o més jugadors tenen una estratègia i es desvien per fer un altre moviment no contemplat per l'estratègia, aquest fet comporta un resultat relatiu negatiu pel jugador que ha efectuat el moviment; aquest resultat també pot ser donat al joc Toma 6!, ja que a vegades pot ser imprevisible el moviment dels altres adversaris.

Si es vol aconseguir que una estratègia sigui considerada com a positiva s'ha de realitzar seguint sempre aquesta estratègia, però no només això, sinó que tampoc no es pot avaluar el valor de l'estratègia fins disputades moltes partides; ja que el fet de tenir un component atzars fa que no es puguin guanyar totes i cadascuna de les partides. Altrament, però, no només cal seguir sempre l'estratègia i no voler cometre cap error, sinó que també pot considerar-se positiu per guanyar desviar-se per derrotar l'oponent si les condicions ho requereixen; per exemple en el cas del Toma 6! tirar una carta que no segueixi l'estratègia si al jugador li queden dos cartes i preveu que canviar l'estratègia li farà no guanyar punts amb les dos tirades restants.

Diversos estudis demostren que les estratègies basades en retroalimentació aconsegueixen millors resultats que estratègies basades en raonaments purament racionals.

Hi ha dos algorismes prometedors per usar en jocs IIG. Per una banda, el Counterfactual Regret Minimization (CFR) és un algorisme que fa valdre les accions dutes a terme amb anterioritat de manera errònia pel futur intentar reduir els errors el màxim possible. Aquest algorisme requereix una memòria lineal d'informació.

Per altra banda, l'algorisme Monte Carlo (MC) utilitza l'aleatorietat per resoldre problemes que poden ser deterministes, en principi. Del Monte Carlo Tree Search trobem diferents variants com: Monte Carlo Simulation que utilitza experiments aleatoris per resoldre moltes variables aleatòries, el Flat Monte Carlo que utilitza el MC Simulation amb les accions d'un estat concret, el Monte Carlo Tree Search (MCTS) que consisteix en ordenar i avaluar les millors opcions de l'arbre creat per dur a terme el moviment i el Monte Carlo Sampling for Regret Minimization (MCCFR) que redueix el temps de convergència de CFR mitjançant MC Sampling a través de blocs de camins des de l'arrel fins a un node terminal i llavors en calcula els errors.

Els primers passos en el rendiment dels jocs programats informàticament va ser amb el joc d'escacs, on primerament es va concentrar en la força bruta, però posteriorment es va demostrar que hi havia una forta correlació entre la velocitat de cerca i el rendiment del programa d'escacs per jugar [3]. Més endavant, es van anar publi-

cant algorismes més innovadors.

En el joc de IIG el fet de no poder contemplar tots els escenaris possibles respecte als de PIG canvia l'espai de cerca donant com a resultat un gran factor de ramificació i la necessitat de tenir en compte les probabilitats.

Des del punt de vista de la IA, crear programes per jugar bé a un joc com el pòquer és molt difícil, ja que ha de tenir en compte coneixements incomplets (targetes d'oponents ocultes), múltiples agents competidors i engany; com també trobem en el joc Toma 6!

La investigació dels jocs de taula i de cartes està motivada històricament perquè plantegen reptes interessants computacionalment en intel·ligència artificial des dels primers anys de la informàtica; a diferència de noves formes de joc d'ordinador més interactives que posteriorment es van anar introduint i que estaven més basats en uns gràfics computacionalment més potents i no tant en agents intel·ligents.

El treball en jocs d'ordinador ha servit com avenç, amb proves experimentals, en moltes àrees de la informàtica i de la intel·ligència artificial.

Actualment, és factible recollir quantitats de dades molt detallades de jocs digitals i aquesta facilitat d'accés a les dades de joc ha donat lloc a més ús en l'enfocament de mineria de dades durant el desenvolupament [4]. També els arbres de cerca de MTCS s'han convertit en una solució important per la intel·ligència artificial, encara que no necessàriament l'estil del joc utilitzant aquesta estratègia ha de ser semblant a la de l'ésser humà o a la que s'espera comunament.

A part d'usar cerca d'arbres de Monte Carlo, també s'han utilitzat per a diverses formes de joc en els últims anys les xarxes neuronals, amb l'objectiu de crear els agents més forts possibles dins dels seus dominis. Per exemple, en el GO es va utilitzar l'aprenentatge amb xarxes neuronals que van servir per guiar el MCTS i millorar-ne els resultats. Tanmateix, també es van utilitzar en jocs aleatoris per proporcionar estimacions del valor de mantenir cada combinació de cartes en el pòquer.

Si tractem la possibilitat d'emular el joc humà, tenim dos tipus d'imitacions: la imitació directa i indirecta.

La imitació directa utilitza un mètode d'aprenentatge supervisat en un conjunt de dades que conté una llista d'estats en un joc, juntament amb l'acció que ha fet un jugador humà en aquest estat.

La imitació indirecta, que és la imitació humana que s'usa principalment en el desenvolupament d'aquest treball, significa que l'agent utilitza un predictor dels moviments humà per fer un algorisme de presa de decisions cap a un comportament semblant a un humà, en lloc d'especificar-lo completament.

En treballs fets anteriorment es plasma la dificultat entre maximitzar el rendiment d'una IA i fer-la semblant a un humà.

2 OBJECTIUS DEL PROJECTE

2.1 Principal

- Dissenyar un sistema algorítmic que permeti jugar i guanyar partides en una adaptació informàtica del joc de taula Toma 6!

2.2 Secundaris

- Definir i dissenyar diferents perfils de jugadors.
- Desenvolupar una versió de prova del joc per poder validar els algorismes.
- Preparar els diferents algorismes i enfrontar-los entre ells en diferents variants per obtenir els seus resultats en diferents mètriques.

3 REQUISITS SOFTWARE/HARDWARE

Pel que fa a les eines software s'ha utilitzat com a IDE el Visual Studio Code [5] per simplicitat amb diferents tipus d'arxius, perquè no requereix crear un projecte d'un tipus específic de Visual Studio Code i perquè es considera més flexible pel que fa a la usabilitat. També, però, es pot usar Pycharm [6] o Spyder [7] creant un projecte específic i transferint-hi els arxius.

Respecte al llenguatge de programació s'ha utilitzat Python [8], ja que ens permet tenir més eines i llibreries a l'abast, i també estructures de dades que ens permet fer càlculs més ràpids i poder visualitzar els resultats obtinguts de manera més gràfica.

Aspectes	Hardware VSCode	Requisits Actuals
Sistema Operatiu	Windows 10	Windows 11, 64-bit
Processador	1.6 GHz	11th Gen Intel(R) Core(TM) i7-1195G7 @ 2.90GHz
Memòria RAM	1GB RAM	16 GB RAM
Gràfics		Intel(R) Iris(X) Xe Graphics
Versió Python	Python 2.X (dependrà de les llibreries emprades)	Python 3.10

4 RISCOS DEL PROJECTE

Cas	Risc	Pla	Probabilitat
Objectius molt ambiciosos	Alt	Modificar els objectius fixats per assoliments més realistes i assolibles	Mitjana
Terminis molt justos	Mig	Reduir la complexitat de les metes	Mitjana
Ordinador s'espantlli	Alt	Tenir penjats al Google Drive els documents escrits i els de codi per evitar perdre el progrés	Baix

No aconseguir dissenyar un agent intel·ligent	Crític	Continuar observant diferents estratègies senzilles que permetin a l'agent poder tenir un estil de joc senzill	Baix
No aconseguir dissenyar agents intel·ligents diferents	Mig	Jugar al joc amb diferents persones i amb diferents quantitats de persones per observar diferents estratègies de joc i fer agents que combinin les diferents estratègies i permutant les propietats dels moviments en cadascun d'ells	Mitjana
Els agents intel·ligents no guanyen mai	Alt	Buscar noves estratègies, siguin més complexes, o inclòs més senzilles en ser un joc amb una part de sort	Baix

5 METODOLOGIA

Per tal de garantir i mantenir un flux de treball adequat i eficient, s'optarà per seguir la metodologia àgil Kanban [9], la qual s'integrarà amb l'eina Trello [10]. Mitjançant aquesta metodologia i plataforma, es podrà obtenir una visió clara en qualsevol moment de l'estat actual del projecte, permetent avaluar si es desenvolupa dins dels terminis establerts o si es detecten retards.

6 PLANIFICACIÓ TEMPORAL

Tasca	Durada
Estudi d'estat de l'art i familiarització amb el joc Toma 6!	23 febrer - 10 març
Desenvolupament de la primera versió del joc (sense IA) i creació del banc de proves	11 març - 10 abril
Lliurament informe de progrés 1	21 abril
Disseny, implementació i avaluació d'agents intel·ligents pel joc	11 abril - 10 maig
Experimentació, anàlisi de resultats i refinaments	10 maig - 31 maig
Lliurament informe de progrés 2	26 maig
Proves d'usabilitat amb gent i recopilació de feedback	1 juny - 16 juny
Proposta informe final	16 juny
Redacció de la documentació, article final i preparació de la defensa oral	16 juny - 30 juny
Proposta presentació	29 juny
Lliurament del dossier	1 juliol
Presentació	5 juliol

7 DESENVOLUPAMENT

El desenvolupament d'aquest projecte consta principalment de dues parts: software i computació.

La primera, la part de software, està formada per l'algorísmica del flux normal del joc i les seves classes.

La segona, la part de computació, està formada per l'algorísmica que conformarà els agents intel·ligents capaços de jugar al joc de certes determinades maneres.

7.1 Software

El flux del joc i les classes del desenvolupament són els que controlen que se segueixin les regles del joc del Toma 6! i que, per tant, se'n segueixin les normes.

Les classes del joc són les següents: la classe Game, que controla totes les funcionalitats del joc, i la classe Player que és una classe abstracta de la qual sorgeixen 2 tipus de classes concretes que són el jugador real i jugadors algorísmics. Els jugadors tenen la capacitat per:

- Poder seleccionar una de les seves cartes de la mà.
- Poder seleccionar una de les files en el cas d'haver tirat una carta més baixa que l'última carta de les 4 files.
- Comptar els bous (punts negatius) de les cartes que han anat acumulant a cada una de les rondes.

Més endavant s'explicarà com es duran a terme aquestes accions en el cas de l'agent intel·ligent.

El joc està format per rondes, on cada ronda els jugadors tindran 10 cartes. Un cop finalitzades les 10 cartes, es començarà una nova ronda. A cada ronda s'aniran acumulant els bous (punts negatius) aconseguits i així durant totes les rondes fins que un dels jugadors arribi a tenir 66 bous o més.

En primer lloc, s'iniciarà el taulell (initBoard/reinitBoard) on se seleccionaran les cartes del joc de la carta número 1 fins a la carta numJugadors*10+4, i es repartiran 10 cartes aleatòriament (deal) a cadascun dels jugadors i les 4 restants formaran el taulell inicial.

Seguidament, es mostrarà el taulell del joc (showGameBoard) i tots els jugadors jugaran una carta de la seva mà sense descobrir-la a la resta de contrincants (allPlayersThrowCard). Dins d'aquesta funcionalitat tindrà lloc l'estratègia dels jugadors, tant dels jugadors agents intel·ligents com dels jugadors reals, els quals analitzaran quina carta és més adient.

Una vegada tots els jugadors tinguin la carta seleccionada, es mostraran aquestes cartes perquè tots els participants les puguin veure (showThisTurnCards).

Un cop totes les cartes estiguin ensenyades, els jugadors podran tirar la seva carta seguint l'ordre numèric ascendent de les cartes tirades en aquest torn (sortOrderOfTheCards). Per fer-ho, s'haurà de comprovar la posició del taulell de cadascuna de les cartes (checkPositionBoard) que es realitza de forma automàtica sempre que no hi hagi carta baixa. Hi ha dues casuístiques per tirar la carta:

1. D'una banda, la carta que ha escollit el jugador és més petita que l'última carta de les 4 files que hi ha al taulell i no la pot col·locar en cap de les files. En aquest cas, el jugador té la possibilitat

d'escollir una fila del taulell (chooseLine) i quedar-se amb les cartes que hi ha, i en conseqüència amb els bous de totes les cartes de la fila ("eat" the cards line). Posteriorment, posarà la seva carta com la primera de la nova fila (put the card in the new empty line). En aquesta funcionalitat, es tornarà a donar lloc a l'estratègia dels jugadors, tant dels agents intel·ligents com dels reals, que analitzaran quina fila prefereixen escollir.

Ex: el jugador haurà d'escollir una de les 4 files, ja que la carta introduïda és més petita que totes les últimes cartes de les files (en vermell). Si el jugador escull la fila 3, com podem veure a la Fig. 1, es quedarà amb les cartes 19 i 20 i introduirà la carta 16 com a primera de la fila 3.

f1	30					nova carta: 16	f1	30				
f2	22	25	28				f2	22	25	28		
f3	19	20					f3	16				
f4	10	13	15	18			f4	10	13	15	18	

Fig. 2. Exemple tirar carta baixa

2. D'altra banda, en cas de no haver tirat una carta baixa, el jugador haurà de col·locar la seva carta a la fila que li correspongui. Les cartes de les files s'ordenen d'esquerra a dreta de més petites a més grans, és a dir, en ordre ascendent.

També, la carta tirada s'haurà de posar a la fila en la qual tingui menys diferència amb l'últim número de les 4 files.

Ex: com que el 20 és amb el que més s'aproxima al 27 en ordre ascendent, la carta s'introduirà a la fila 3.

f1	30					nova carta: 27
f2	22	25	28			
f3	19	20	27			
f4	10	13	15	18		

Fig. 3. Exemple de tirar carta més gran que l'últim d'alguna la fila

Si la carta tirada pel jugador és la sisena de la fila, el jugador es quedarà amb les cartes de la fila ("eat" the cards line) i la seva carta passarà a ser la primera carta de la fila (put the card in the new empty line).

Ex: com que el 24 és la sisena carta de la fila 3, ja que amb el número que té menor diferència és el 23, el jugador que ha tirat la carta es queda amb les cartes de la fila 3 i introdueix el 24 com a primer de la fila 3.

f1	30					6ena	f1	30				
f2	22	25	28			nova carta: 24	f2	22	25	28		
f3	19	20	21	22	23		f3	24				
f4	10	13	15	18			f4	10	13	15	18	

Fig. 4. Exemple de tirar la carta que correspon a la sisena carta de la fila

Quan tots els jugadors hagin comprovat la posició del taulell a la seva carta i hagin efectuat el moviment, tindran una carta menys a la seva mà. Seguidament, tornaran a fer tots una tirada, i així successivament fins que no

els hi quedin cartes a la mà, que serà quan es donarà per finalitzada la ronda. Aleshores, s'anotaran tots els bous de cadascun dels participants. Seguidament, es tornarà a iniciar el taulell (reinitBoard), es repartiran les cartes i es tornarà a seguir el mateix procediment del joc.

El joc s'acaba quan un dels jugadors tingui 66 o més bous, llavors, el jugador que tingui menys bous serà el guanyador de la partida.

A la Fig. 5 podem veure el diagrama de flux:

Les accions en vermell corresponen a les accions on intervenen les estratègies dels jugadors i moviments que efectuaran, tant els jugadors reals com els jugadors intel·ligents. En aquestes accions hi entra en joc la part d'algorísmica per decidir quina carta tirar per part dels agents intel·ligents.

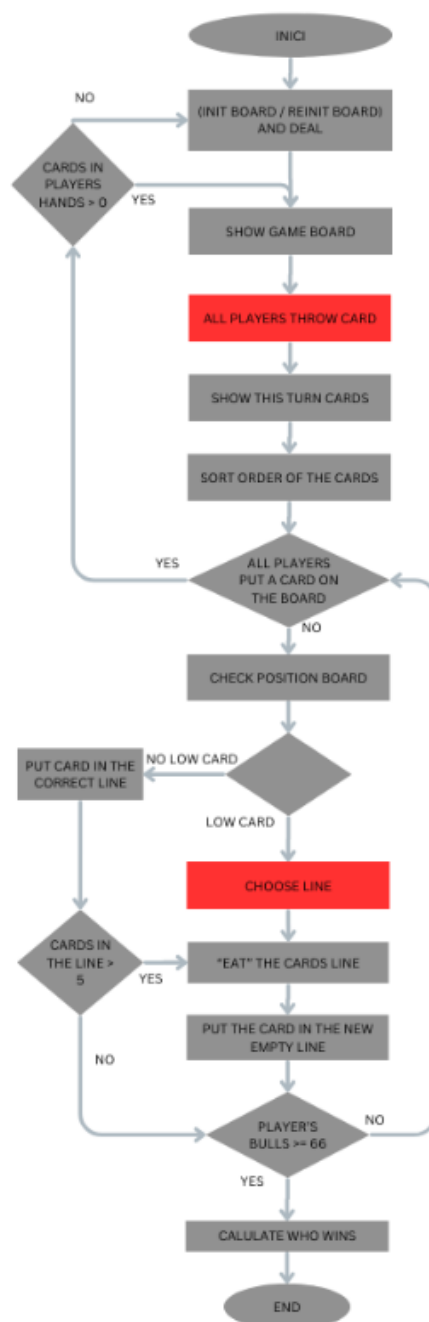


Fig. 5. Diagrama de flux del joc

7.2 Computació

Hi ha diversos algorismes que podran tenir cadascun dels agents intel·ligents, i d'aquesta manera també jugaran de diferent forma. Cal recordar que tots juguen alhora la seva carta seleccionada, i que, per tant, pot ser que un jugador es pensi que tirarà una carta a una fila i l'acabi tirant a una altra.

Per l'acció de All Players Throw Card els agents intel·ligents podran tenir les següents funcionalitats:

- **Tirar la carta de valor més baix que no sigui carta baixa:** l'agent intel·ligent selecciona la carta més baixa de la seva mà, que no sigui una carta baixa, és a dir, que sigui una carta que la pugui col·locar, a priori, en una de les files.
- **Tirar la carta que porti al límit la fila:** l'agent intel·ligent selecciona una carta que segur que estarà en una de les files. Aquesta ocuparà la posició 5 de la fila, suposant que es tiraran les cartes entremig de l'última carta actual de la fila i la carta escollida per l'agent intel·ligent. Ex: En la Fig. 6 podem veure que seleccionem la carta 34 que aniria al límit de la fila, suposant que els altres participants tirin la carta 32 i 33. Per obtenir el número ens caldrà saber quina és la llargada actual de la fila ($f1=len(2)$), l'últim número de la fila (31) i mirar si tenim la carta a la nostra mà: $31 + 5 (len(fila) - 2 (len(f1)) = 34$. Com que tenim la carta 34, aquest algorisme faria tirar aquesta carta.

f1	30	31			34
f2	22	25	28		
f3	19	20			
f4	10	13	15	18	

mà jugador:
1 8 21 29 34

Fig. 6. Exemple de tirar carta amb l'algorisme que porta al límit la fila

Aquest algorisme tindrà aquesta variant i una altra que serà sabent les cartes que falten per tirar, i per tant, si la carta 32 ja s'ha tirat, la carta que buscaria l'algorisme en aquest mateix exemple seria la carta 35. Com que en l'exemple de la Fig. 6 no la tenim a la nostra mà, no seria aquest l'algorisme seleccionat per jugar la carta en aquest torn.

- **Tirar la carta que més s'acosti a alguna de les cartes de la fila:** aquest algorisme busca quina de les cartes de la mà s'acosta més (menor diferència) a alguna de les cartes que hi ha a la fila. Per exemple, si tenim el taulell i la mà corresponents de la Fig. 7

f1	30	31			
f2	22	25	27		
f3	19	20			
f4	6				

mà jugador:
1 8 21 29 34

Fig. 7. Exemple de tirar la carta amb menor diferència a una de les files

Seleccionarem la carta 21, ja que és la que té menor diferència:

$$21 - 20 = 1$$

$$8 - 6 = 2$$

$$29 - 2 = 2$$

$$34 - 31 = 3$$

Aquest algorisme té 3 variants que es poden utilitzar:

1. En cas de tenir dues o més cartes amb la mateixa diferència a la mà, es tira la carta més baixa.
2. La mateixa casuística, però en aquest cas es tira la carta més alta.
3. L'agent intel·ligent coneix totes les cartes que falten per tirar i, per tant, en el cas anterior si la carta 28 ja s'hagués tirat en el joc, la carta 29 i la carta 21 tindrien la mateixa diferència amb una de les cartes (1 unitat de diferència).

- Triar carta segons puntuació en el taulell:

aquest algorisme avalua l'estat del taulell i la llargada de les files per seleccionar quina de les cartes de la mà tirar. Se li assigna al taulell una valoració per les columnes de cada fila, tenint una valoració d'1 la columna 2 i una valoració de 4 la columna 5, sent la valoració més baixa la més bona i la més alta la més dolenta. Llavors, la puntuació de la carta és causada per a quina columna aniria a dins de cada fila. Per exemple, el taulell de la Fig. 8 estaria representat de la següent manera: La carta de la mà seleccionada serà la carta 8, ja que té una valoració de 2, mentre que les altres cartes de la mà tenen una valoració superior, tot i algunes ser més properes a la carta d'alguna de les files (carta 22).

f1	30	31	32/2	33/3	34/4
f2	22	25	27	28/3	29/4
f3	19	20	21	22/3	23/4
f4	6	7/1	8/2	9/3	10/4

mà jugador:
1 8 22 29 34

Fig. 8. Exemple de tirar la carta segons l'algorisme de puntuació en el taulell

Aquest algorisme també té 5 possible variants:

1. En cas d'haver-hi dues cartes amb la mateixa valoració (menor valoració) es tira la carta més baixa.
2. La mateixa casuística, però en aquest cas es tira la carta més alta.
3. L'agent té en compte les cartes que s'han tirat (i falten per tirar) a l'hora de valorar una casella del taulell. Per exemple, si ja s'ha jugat la carta 7 del joc, el taulell tindria la valoració que veiem a la Fig. 9.

f1	30	31	32/2	33/3	34/4
f2	22	25	27	28/3	29/4
f3	19	20	21	22/3	23/4
f4	6	8/1	9/2	10/3	11/4

mà jugador:
1 8 22 29 34

Fig. 9. Exemple de valoracions de les possibles cartes respecte la columna que ocupen

4. L'agent selecciona la carta amb major valoració en lloc d'en menor valoració. I d'aquesta tira la carta amb nombre més baix.
5. La mateixa casuística que l'anterior, però tirant la carta amb el nombre més alt.

- **Triar cartes a partir del rang de cartes possibles per fila sense quedar-se amb les cartes:** establir el rang de cartes que es poden tirar a cadascuna de les files. Per exemple, en el següent cas de l'estat del tauler de la Fig. 10, el rang de f1 és: [32,34]; f2 [28,29], f3 [22,23]; f4 [7,10].

f1	30	31	32	33	34
f2	22	25	27	28	29
f3	19	20	21	22	23
f4	6	7	8	9	10

Fig. 10. Exemple del tauler de l'algorisme de tirar cartes a partir del rang

A partir d'aquí l'agent mira si té alguna de les seves cartes dins d'aquests rangs i pot triar:

1. La carta més baixa.
 2. La carta més alta.
 3. Tirar la carta més baixa de la fila en la qual té més cartes a la mà dins del rang.
- **Tirar carta baixa:** l'agent intel·ligent avalua si amb les cartes de la seva mà pot llençar una carta baixa. Aquesta a priori li serveix per quedar-se amb les cartes d'una fila i introduir la seva com a primera de la fila (ja que no sap si algú tirarà una carta més baixa).
 - **Triar aleatòriament una carta:** es tracta de tirar una carta aleatòriament de la mà de l'agent. Es pot utilitzar especialment en aquells estats de la partida en què cap dels nostres algorismes ens pot donar una bona resposta.

Per l'acció de ChooseLine els agents intel·ligents podran tenir les següents funcionalitats:

- **Triar la fila amb menys valor de bous:** selecciona la fila que la suma dels bous de les seves cartes és menor
- **Triar la fila sense cartes dins el rang:** l'agent pot triar la fila en la qual no hi hagi possibles cartes que hi podria tirar de la seva mà. Ex: com podem veure a la Fig. 11 el jugador seleccionarà la f4, ja que a la f1 té la carta 34 dins el rang, a la f2 la 29 i a la f3 la 22.

f1	30	31	32	33	34
f2	22	25	27	28	29
f3	19	20	21	22	23
f4	6	7	8	9	10

mà jugador:
1 22 29 34

Fig. 11. Exemple de taulell i mà de l'algorisme de triar la fila sense cartes dins del rang

8 RESULTATS

Per avaluar els diferents algorismes, a cadascuna de la definició dels algorismes se'ls hi ha atorgat un nombre per simplicitat com podem veure a la Fig. 12.

#Numero atorgat a l'algorisme	Nom de la funció de l'algorisme
1	tirarCartaBaixaNoCartaBaixa
21	tirarCartaLimitFila
22	tirarCartaLimitFilaConeix
31	tirarCartaMenysDiferencia (minim)
32	tirarCartaMenysDiferencia (maxim)
33	tirarCartaMenysDiferenciaConeix
41	tirarCartaSegonsPuntuacio (minim)
42	tirarCartaSegonsPuntuacio (minimAlt)
43	tirarCartaSegonsPuntuacio (maxim)
44	tirarCartaSegonsPuntuacio (maximAlt)
45	tirarCartaSegonsPuntuacioConeix
51	tirarCartaRangPossibleiNoMenjar (minim)
52	tirarCartaRangPossibleiNoMenjar (maxim)
53	tirarCartaRangPossibleiNoMenjar (rang)
6	tirarCartaBaixa
7	tirarCartaAleatoria

Fig. 12. Algorismes automàtics per seleccionar carta amb el seu identificador

8.1 Mètriques

Per tal d'avaluar els diferents algorismes s'han fet diverses proves amb diferents nombres de jugadors i avaluant diferents mètriques. S'ha dut a terme amb 2, 3, 4 i 5 jugadors, i s'han avaluat les següents mètriques:

- Percentatge de partides guanyades
- Percentatge de partides perdudes
- Mitjana de punts negatius quan es guanya la partida
- Mitjana de punts negatius de l'adversari guanyador quan es perd la partida
- Mitjana de punts negatius de l'adversari guanyador quan no es perd ni es guanya la partida (no avaluada amb 2 jugadors, ja que no es pot donar aquesta casuística)
- Mitjana de rondes de la partida.

Aquestes mètriques han sigut seleccionades perquè s'han considerat les més idònies per la rellevància que tenen en el joc, ja quedeterminen molt clarament l'estat final de la partida, i en conseqüència al llarg de tota la partida.

8.2 Resultats amb dos jugadors

Per jocs amb 2 jugadors s'han dut a terme una sèrie de partides enfrontant tots els algorismes entre ells (inclòs el duel entre un algorisme contra una versió d'ell mateix) amb un nombre de partides de 1000, 2500 i 10000. Aquests algorismes si en usar el seu algorisme no troben una carta seleccionada, usen l'algorisme de la carta baixa

(excepte el de la carta baixa que no pot tornar-se a usar perquè tornaria a no tenir una carta seleccionada), i si aquest algorisme tampoc selecciona cap carta, usa l'algorisme de triar una carta aleatòriament.

Així doncs, tenim 3 anàlisis amb 3 matrius quadrades per cadascuna de les mètriques. Els resultats de realitzar 1000, 2500 i 10000 partides per cadascun dels enfrontaments han sigut força similars.

Com podem observar a la Fig. 13, els algorismes que obtenen uns millors resultats en percentatge de partides guanyades són els algorismes 33 i 45 els quals actuen coneixent les cartes que falten per tirar amb un percentatge de partides guanyades entre un 70 i un 90%, seguits dels algorismes 31, 41, 42, 43, 44, 51, 52 i 53. Amb pitjor percentatge de victòries hi trobem els algorismes 1, 32, 6 i 7 entre el 10% i 50% els quals es podien preveure que obtindrien menor puntuació, ja que juguen tirant unes cartes molt concretes i, si no poden, juguen amb cartes aleatòries sense tenir en compte l'estat de la partida.

Aquests resultats són amb partides entre dos jugadors, motiu pel qual el percentatge de partides perdudes és inversament proporcional a les guanyades.

La mètrica de punts per guanyar, com podem apreciar a la taula 2, ens permet observar molt clarament que els algorismes que aconseguixen fer menys punts per guanyar són els mateixos algorismes que tenen millor ràtio de partides guanyades, el 33 i el 45, que estan més o menys entre els 35 i 45 punts de mitjana. Aquest fet ens mostra el perquè aquests algorismes solen guanyar més partides, ja que aconseguixen molts menys punts de mitjana si l'adversari cau derrotat.

Amb la darrera mètrica, la mitjana de rondes per guanyar, es pot observar clarament que el nombre de rondes per partida va entre 6 i 7 rondes per partida. Amb aquesta mètrica es poden tornar a observar uns bons resultats dels algorismes 33 i 45 respecte a la resta, ja que tarda menys rondes que la resta a guanyar i, per altra banda, també podem observar-hi que quan es juga tirant cartes aleatòriament la partida s'allarga més rondes, sigui per guanyar o per perdre.

8.3 Resultats amb múltiples jugadors

Per avaluar les partides amb 3, 4 i 5 jugadors s'han realitzat una sèrie de combinacions diferents com es pot

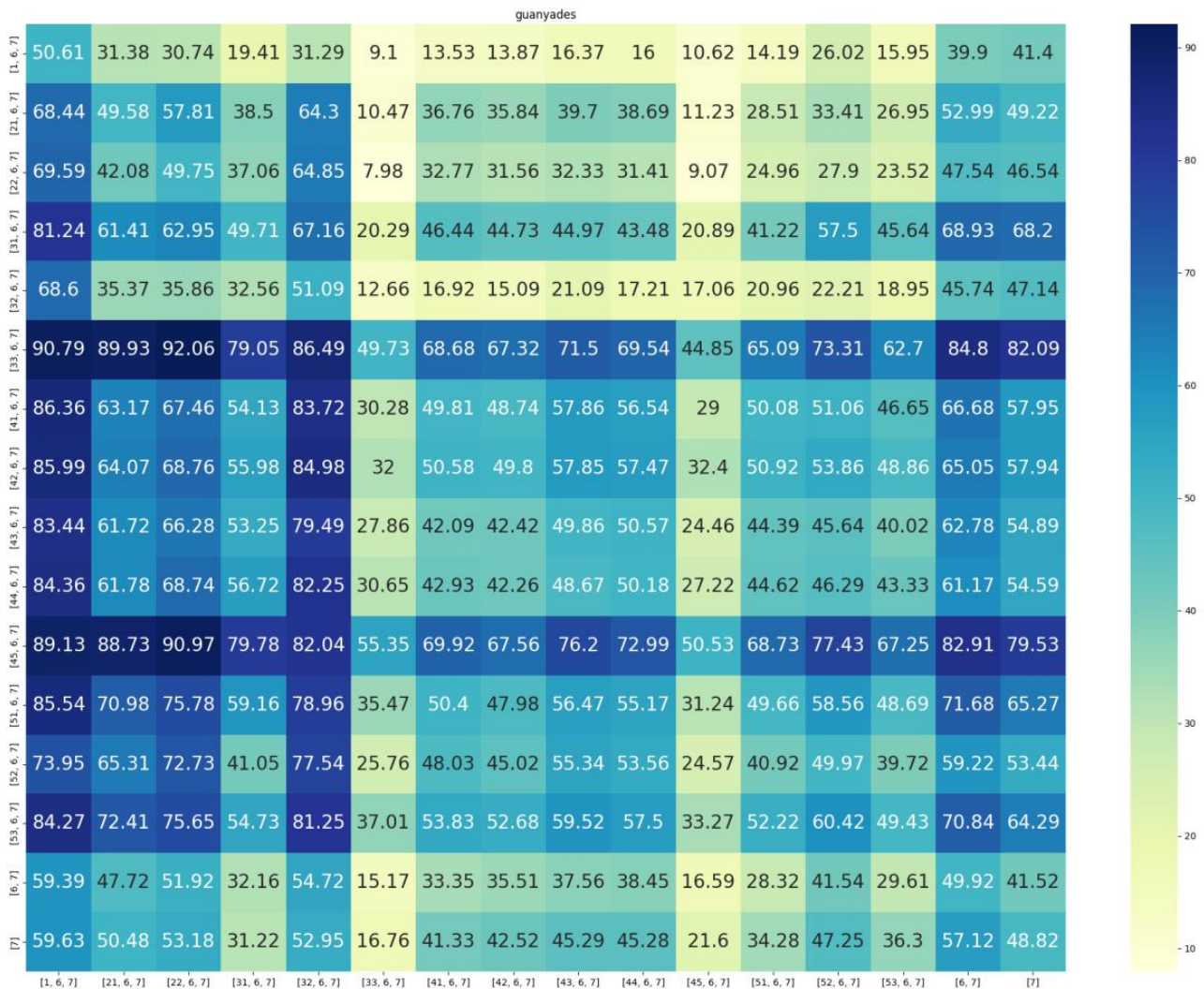


Fig. 13. Matriu de confusió de percentatge de partides guanyades de tots els algorismes amb 2 jugadors

veure a la taula 3, ja que era inviable realitzar el mateix procés que amb 2 jugadors. Per fer-ho, s'han seleccionat els dos algorismes (que són dos algorismes que tenen informació de les cartes ja tirades) que han donat millors resultats amb 2 jugadors, el 33 i el 45, els respectius algorismes, però sense conèixer les cartes ja tirades, el 31 i el 41 i finalment, l'algorisme 22 per veure com es comportava un algorisme diferent i que coneix les cartes ja tirades.

S'han realitzat les diferents combinacions que trobem a la taula 4 amb 10000 partides en cadascuna d'elles.

Abans de concloure els resultats, pel que fa a les proves realitzades, cal mencionar que el primer jugador de la combinació usa l'algorisme en qüestió (el 22, 31, 33, 41 o 45) per tirar la carta, i que si no selecciona cap carta, usa l'algorisme 6. Si aquest darrer tampoc la troba, acaba usant l'algorisme 7, que segur que selecciona una carta per la tirada. Per contra, els contrincants d'aquest jugador amb 3, 4 i 5 jugadors, és a dir els 2, 3 i 4 jugadors restants de la partida respectivament, no usen l'algorisme 6 en cas de no poder seleccionar cap carta, i usen directament l'algorisme 7.

Un cop dutes a terme les proves podem concloure el següent dels algorismes a partir de les mètriques:

Podem observar una diferència substancial pel que fa a utilitzar l'algorisme 6 abans de fer ús de l'algorisme 7 quan l'algorisme principal no pot seleccionar cap carta. Aquest fet, el trobem principalment quan els participants són 2 jugadors, ja que amb les proves realitzades s'ha obtingut que si sumàvem els percentatges de quan X s'enfronta a Y i Y s'enfronta a X, el percentatge de victòries sol ser superior al 100%, mentre que el de derrotes sol ser inferior al 100%.

Amb la mètrica de partides guanyades podem observar que els algorismes amb un percentatge més alt de victòries són clarament el 33 i el 45 els quals fan servir la mateixa lògica que el 31 i el 41, però coneixent les cartes que falten per jugar. Per contra, el que obté el percentatge més baix amb 3 jugadors és l'algorisme 22 i amb 4 i 5 jugadors són els algorismes 22 i 31.

Amb la mètrica de les partides perdudes podem observar molt clarament que els algorismes 33 i 45 tenen uns resultats molt millors (és a dir, baixos) a la resta. Obtenen uns percentatges molt minsos.

Pel que fa a la mètrica de la mitjana dels punts guanyats com veiem a la Fig. 14 l'algorisme 33 és el que aconsegueix menors punts quan guanya contra l'algorisme 22, 31 i 41, apropant-se als 20 punts de mitjana, però quan juga contra l'algorisme 45, el 45 aconsegueix menors

punts negatius quan guanya. També, com podem veure a la Fig. 14 els punts que s'aconsegueixen en aquesta mètrica amb 5 jugadors són molt inferiors als resultats que teníem amb 2 jugadors.

Observant la mètrica dels punts de l'adversari quan aquest guanya es pot veure com disminueixen lleugerament, entre 4 i 5 punts de mitjana, a mesura que augmenta el nombre de participants.

Pel que fa als punts aconseguits quan no es guanya ni es perd, els algorismes que aconsegueixen menors punts són els que tenen un percentatge de victòries més altes i als que aconsegueixen menors punts quan guanyen les partides. També cal destacar que tot i que els punts aconseguits en aquesta mètrica disminueixen a mesura que augmenten els jugadors, tampoc disminueix gaire, ja que disminueixen 3-4 punts en augmentar un jugador.

Amb el nombre de mitjanes de rondes per partida podem observar molt clarament que a mesura que augmenten els participants de la partida, la partida dura menys rondes. Amb dos jugadors les partides duren entre 6 i 7 rondes, amb tres jugadors de 4,7 a 5,2, amb cinc jugadors de 4,15 a 4,40, i amb cinc de 3,9 a 4,1. Tanmateix, també hi ha diferències amb aquesta mètrica entre els algorismes, ja que obtenim que els algorismes amb millor percentatge de partides guanyades i que guanyen amb menors punts són els que aconsegueixen guanyar les seves partides amb una mitjana de rondes menors. En canvi, com pitjor són els valors de partides guanyades i de punts negatius dels algorismes, més rondes duren les partides quan aquests guanyen.

9 CONCLUSIONS

Aquest treball ha tractat el desenvolupament del joc de cartes Toma 6! i la seva informatització. A més, també s'hi han afegit algorismes automàtics en els punts de la partida que hi intervenen els jugadors: tirar carta i seleccionar fila.

Cal destacar que que en les instruccions del joc en una escala de l'1 al 5 de sort/estratègia, on 1 és sort i 5 estratègia, aquest joc el trobem valorat en un 3. Aquest fet constata que tot i que es puguin plantejar millors o pitjors estratègies, la sort juga un paper important en el desenvolupament del joc.

S'ha aconseguit dur a terme algorismes que tenen coneixement de l'estat del tauler i de les cartes de la mà. Algun d'ells també tenen més informació de l'estat de la partida, com és el coneixement de les cartes que ja s'han

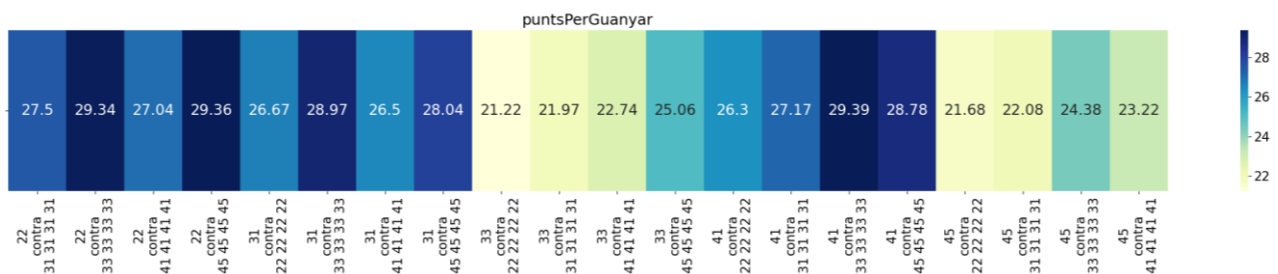


Fig. 14. Taula de resultats de la mètrica de punts per guanyar amb 5 jugadors

tirat, i en conseqüència de les cartes que falten per tirar.

Per fer les proves i obtenir els resultats d'aquests algorismes s'han comprovat els seus valors en partides de 2, 3, 4 i 5 jugadors.

Els algorismes de seleccionar carta que millors resultats han donat són els algorismes 33 i 45. L'algorisme 33 selecciona la carta de la mà amb menor diferència amb alguna de les cartes de la fila. L'algorisme 45 selecciona la carta de la mà que menor puntuació tindria al tauler, segons a la columna en que es situaria. Aquests dos algorismes tenen informació de les cartes que falten per tirar i són els que han obtingut els millors resultats en totes les mètriques. Se'n poden destacar el percentatge de partides guanyades, en els quals han arribat a assolir més del 90% en partides de dos jugadors, i la mitjana de punts per partides on l'algorisme guanya la partida, que ha sigut el menor de tots, on ha arribat a tenir una mitjana mínima propera als 20 punts amb 5 jugadors.

10 SEGÜENTS PASSOS

Els passos següents d'aquest treball serien provar noves combinacions d'algorismes entre els ja existents, com per exemple, usar l'algorisme que tira la carta de la mà més baixa que no és carta baixa (en lloc de o conjuntament amb el de carta baixa) quan l'algorisme principal no pot seleccionar cap carta.

També s'hi podrien afegir probabilitats o heurístiques, ja sigui desenvolupant un algorisme que usi aquestes dues tècniques per seleccionar una carta, o també un algorisme per valorar amb aquestes dues tècniques quin algorisme dels desenvolupats en aquest treball és millor utilitzar en cadascuna de les situacions.

A banda dels algorismes per seleccionar una carta, també es podrien fer més proves per analitzar si hi hauria algun algorisme que donés iguals o millors resultats que l'algorisme de seleccionar la fila amb menors bous.

Per completar la jugabilitat del joc de taula fora bo dur a terme una interfície gràfica per poder jugar amb facilitat al joc i així provar en casos reals la capacitat guanyadora o no dels algorismes. Aquesta interfície es podria dur a terme tant amb un o diversos jugadors reals contra un o diversos jugadors intel·ligents.

AGRAÏMENTS

Voldria expressar el meu sincer agraïment al meu tutor, el Dr. Jorge Bernal del Nozal, per la seva guia, els seus consells i el seu suport constant durant tot aquest procés. Així mateix, per la seva paciència i els seus comentaris que han estat molt importants per a la realització d'aquest treball.

BIBLIOGRAFIA

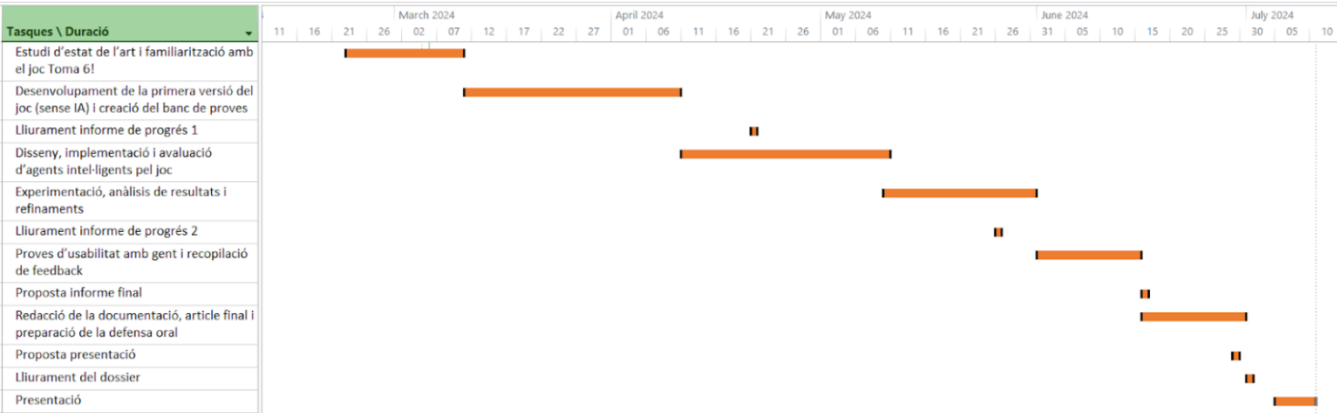
- [1] Toma 6!: https://de.wikipedia.org/wiki/6_nimmt!
- [2] Niklaus, J., Alberti, M., Pondenkandath, V., Ingold, R., & Liwicki, M. (2019, June). Survey of artificial intelligence for card games and its application to the Swiss game Jass. In 2019 6th Swiss Conference on Data Science (SDS) (pp. 25-30). IEEE.
- [3] Schaeffer, J., & Van den Herik, H. J. (2002). Games, computers,

and artificial intelligence. *Artificial intelligence*, 134(1-2), 1-7.

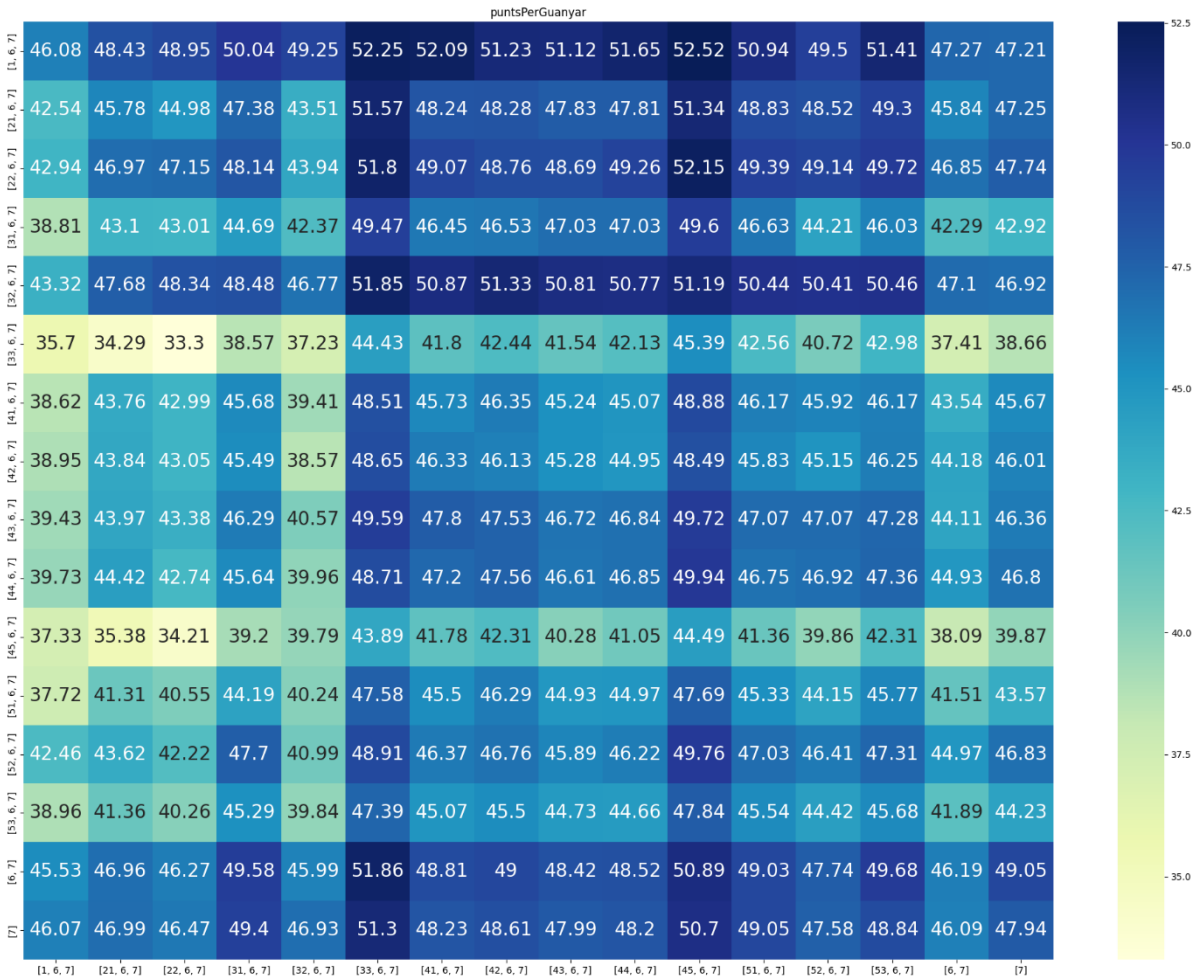
- [4] Baier, H., Sattaur, A., Powley, E. J., Devlin, S., Rollason, J., & Cowling, P. I. (2018). Emulating human play in a leading mobile card game. *IEEE Transactions on Games*, 11(4), 386-395.
- [5] Visual Studio Code: <https://code.visualstudio.com/>
- [6] Pycharm: <https://www.jetbrains.com/pycharm/>
- [7] Spyder: <https://www.spyder-ide.org/>
- [8] Python: <https://www.python.org/>
- [9] Kanban: <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban>
- [10] Trello: <https://trello.com/>

APÈNDIX

TAULA 1. DIAGRAMA DE GANTT



TAULA 2. MATRIU DE CONFUSIÓ DE PUNTS NEGATIU ACONSEGUITS EN LES PARTIDES GUANYADES AMB 2 JUGADORS



TAULA 3. COMBINACIONS DE PARTIDES AMB 3, 4 I 5 JUGADORS (EL PRIMER JUGADOR ÉS L'ANALITZAT)

3 JUGADORS	4 JUGADORS	5 JUGADORS
22 CONTRA 31 I 31	22 CONTRA 31, 31 I 31	22 CONTRA 31, 31, 31 I 31
22 CONTRA 33 I 33	22 CONTRA 33, 33 I 33	22 CONTRA 33, 33, 33 I 33
22 CONTRA 41 I 41	22 CONTRA 41, 41 I 41	22 CONTRA 41, 41, 41, I 41
22 CONTRA 45 I 45	22 CONTRA 45, 45 I 45	22 CONTRA 45, 45, 45 I 45
31 CONTRA 22 I 22	31 CONTRA 22, 22 I 22	31 CONTRA 22, 22, 22 I 22
31 CONTRA 33 I 33	31 CONTRA 33, 33 I 33	31 CONTRA 33, 33, 33 I 33
31 CONTRA 41 I 41	31 CONTRA 41, 41 I 41	31 CONTRA 41, 41, 41 I 41
31 CONTRA 45 I 45	31 CONTRA 45, 45 I 45	31 CONTRA 45, 45, 45 I 45
33 CONTRA 22 I 22	33 CONTRA 22, 22 I 22	33 CONTRA 22, 22, 22 I 22
33 CONTRA 31 I 31	33 CONTRA 31, 31 I 31	33 CONTRA 31, 31, 31 I 31
33 CONTRA 41 I 41	33 CONTRA 41, 41 I 41	33 CONTRA 41, 41, 41 I 41
33 CONTRA 45 I 45	33 CONTRA 45, 45 I 45	33 CONTRA 45, 45, 45 I 45
41 CONTRA 22 I 22	41 CONTRA 22, 22 I 22	41 CONTRA 22, 22, 22 I 22
41 CONTRA 31 I 31	41 CONTRA 31, 31 I 31	41 CONTRA 31, 31, 31 I 31
41 CONTRA 33 I 33	41 CONTRA 33, 33 I 33	41 CONTRA 33, 33, 33 I 33
41 CONTRA 45 I 45	41 CONTRA 45, 45 I 45	41 CONTRA 45, 45, 45 I 45
45 CONTRA 22 I 22	45 CONTRA 22, 22 I 22	45 CONTRA 22, 22, 22 I 22
45 CONTRA 31 I 31	45 CONTRA 31, 31 I 31	45 CONTRA 31, 31, 31 I 31
45 CONTRA 33 I 33	45 CONTRA 33, 33 I 33	45 CONTRA 33, 33, 33 I 33
45 CONTRA 41 I 41	45 CONTRA 41, 41 I 41	45 CONTRA 41, 41, 41 I 41