

# Informe del Dilema del Presoner Iterat

## Dilema del Presoner

En teoria de jocs, el **dilema del presoner** és un tipus de joc de suma no nul·la en el qual dos jugadors poden «cooperar» o «traïr-se». En aquest joc, igual que en totes les situacions de teoria de jocs, l'única preocupació de cada jugador individual és maximitzar els seus beneficis, independentment del benefici de l'altre jugador.

Aquest dilema sol expressar-se com a dos presoners que són interrogats per separat i, segons les seves decisions, obtenen les següents condemnes:

	Presoner B calla	Presoner B confessa
Presoner A calla	A 1 any, B 1 any	A 3 anys, B 0 anys
Presoner A confessa	A 0 anys, B 3 anys	A 2 anys, B 2 anys

Com els presoners no poden preveure la decisió de l'altre, sempre trien “traïr-se”, ja que és la decisió que més beneficiosa de forma individual (0 anys si l'altre coopera i 2 anys enlloc de tres si l'altre ens traïx). Com tots dos arriben a la mateixa conclusió, el resultat previsible sempre és doble traïció. Una forma de demostra que les millors decisions individuals no generen la millor solució final (1 any per cada un si haguessin col·laborat).

## Dilema del Presoner Iterat

En el **dilema del presoner iterat**, la cooperació pot obtenir-se com un resultat d'equilibri. Aquí es juga repetidament, per la qual cosa, quan es repeteix el joc, s'ofereix a cada jugador l'oportunitat de castigar a l'altre jugador per la no cooperació en jocs anteriors. Així, l'incentiu per traïr pot ser superat per l'amenaça del càstig, la qual cosa condueix a un resultat cooperatiu. Si el joc es repeteix indefinidament s'assoleix un equilibri de Nash, tot i que la traïció segueix sent una possible situació d'equilibri.

Per entendre com funciona el joc cal entre la matriu d'opcions com:

	B coopera		B deserta	
A coopera	Presoner A	Presoner B	Presoner A	Presoner B
	R = 1	R = 1	S = 3	T = 0
A deserta	Presoner A	Presoner B	Presoner A	Presoner B
	T = 0	S = 3	P = 2	P = 2

- Temptació(T): La recompensa per desertar quan l'altre cooperi.
- Premi(R): La recompensa per la col·laboració mútua.
- Càstig(P): La recompensa per la deserció mútua.
- Pringat(S): La recompensa per 'fer el primo' i cooperar quan l'altre deserta.

Hi ha dos condicions que s'han de complir per a que la matriu d'utilitat correspongui a la del dilema del presoner.

1.  $T > R > P > S$
2.  $R > (T+S)/2$

La primera condició indica que la cooperació mútua (R) es millor que la deserció mútua (P), l'únic problema és que no podem saber si l'altre no caurà en la temptació (T) e intenti aprofitar-se de nosaltres (S). La segona indica que si en una ronda un jugador obté (T), malgrat que a l'altra obtingui (S), no podrà recuperar-se de no haver guanyat totes dues partides (R).

L'objectiu a llarg termini és guanyar el màxim de punts al final d'un nombre determinat de voltes. A cada volta, tots dos jugadors es decideixen entre cooperar (C) o desertar (D) de forma simultània i sense conèixer la decisió de l'altre (com a l'apartat anterior). Això sí, a l'analitzar cada ronda, s'atorguen els punts segons la matriu i s'informa a cada jugador de l'opció triada per l'altre jugador (de manera que, a la ronda  $n$  el jugador coneix, i pot utilitzar si vol, els resultats de les  $n-1$  rondes anteriors).

Anomenarem estratègia a l'algoritme seguit per cada jugador per tal de decidir-se entre cooperar o desertar (pot fer ús de la història de la partida jugada fins al moment). Exemples d'estratègies:

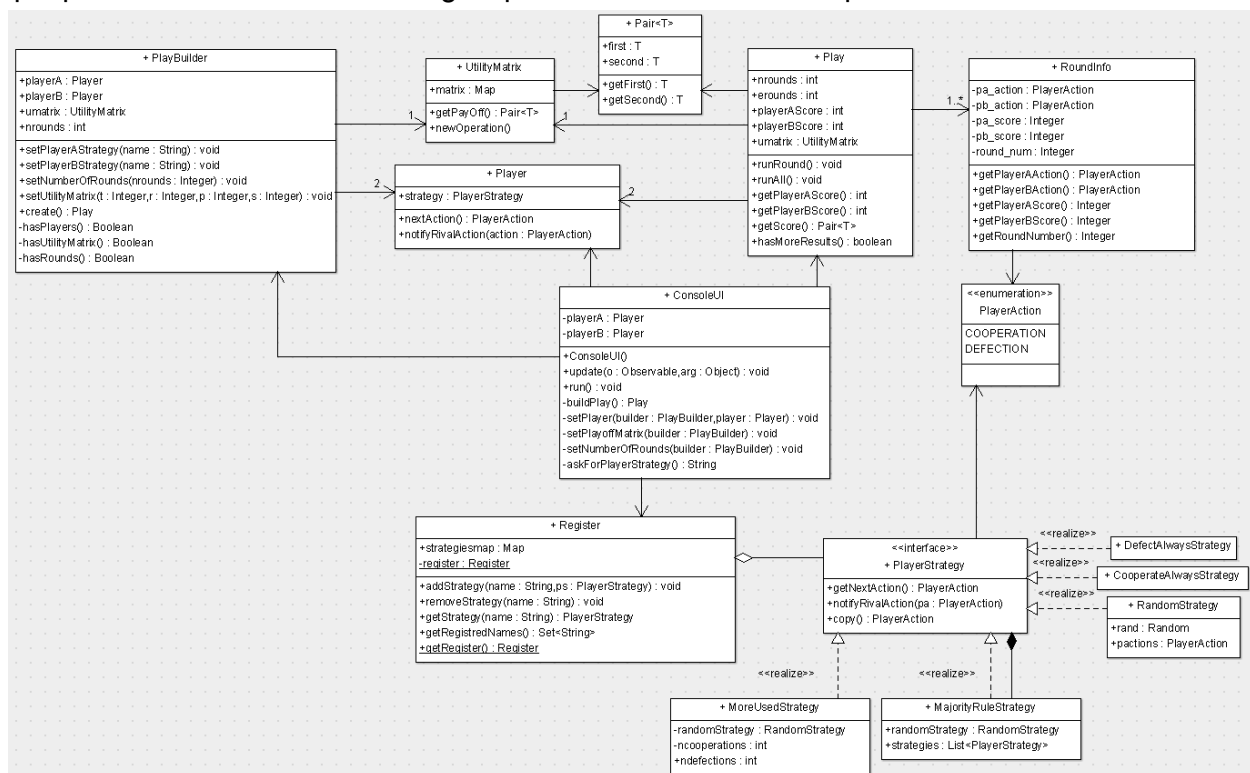
- **Per xula jo:** desertar sempre, independentment del que hagi fet l'altre.
- **Càndida:** cooperar sempre, independentment del que hagi fet l'altre. Com l'anterior, es tracta d'una estratègia cega.
- **Atzarosa:** escollir a l'atzar.
- **Intel·ligent:** tenir en compte les rondes anteriors per tan de prendre la decisió (p.e. cooperar a menys que l'altra hagi desertat més d'una vegada).
- **MoreUsedStrategy:** Aquesta estratègia respon amb l'acció que l'altre jugador ha utilitzat més vegades, si l'altre jugador ha realitzat el mateix nombre de vegades les possibles accions, respon amb una aleatorietat.
- **MajorityRuleStrategy:** Aquesta estratègia està composta per altres estratègies i per cada cop que ha de prendre una acció, respon amb l'estratègia més triada per les estratègies que la formen. En cas que totes les possibles accions s'hagin triat el mateix nombre de vegades, respon de forma aleatòria

No hi ha una estratègia òptima en el sentit de què és la millor davant qualsevol altra ja que, els resultats d'una estratègia depenen de quina sigui l'estratègia seguida per l'altre.

## El Programa

Es vol realitzar una aplicació per simular tornejos del dilema del presoner iterat. En concret, l'usuari podrà escollir l'estratègia emprada per cada jugador, el nombre de rondes que tindrà la partida i els valors de la matriu d'utilitat. De cara a fer això, el sistema tindrà implementat diverses estratègies predefinides (algunes d'elles cegues, algunes probabilistes i d'altres intel·ligents).

També permetrem la definició d'estratègies mixtes que es formin per combinació de dues o més estratègies definides en el sistema (així podrem ampliar el repertori d'estratègies). Per exemple: una forma de combinar seria escollir tres estratègies base i retornar el resultat de la majoria. Per a fer això, el sistema tindrà definit un mecanisme de combinació d'estratègies que permeti definir noves estratègies per a ser utilitzades a les partides.



## Model de Disseny

En aquest apartat es defineixen les classes principals que hi intervenen i les seves responsabilitats.

- Classe **UtilityMatrix**
  - Representa la matriu d'utilitat
- Classe **Play**
  - Representa una partida
  - Coneix els dos jugadors que participen
  - Coneix la matriu d'utilitat
  - Coneix la puntuació de la partida

- Executa una partida a un determinat número de rondes
- PlayBuilder
  - L'objectiu de tenir aquesta classe és aïllar el client d'algunes de les classes del domini com Player o UtilityMatrix
- Classe Player
  - Coneix la seva estratègia
  - Delega sobre l'estratègia les decisions
  - Informe a l'estratègia de les decisions de l'altre jugador
- Classe PlayerStrategy
  - defineix el mètode per prendre la decisió (col·laborar o desertar)
  - defineix el mètode per informar de la decisió a l'altre jugador (important per les estratègies intel·ligents)
  - defineix el mètode que permet fer una còpia profunda d'ell mateixa
- Classe DefectAlwaysStrategy
  - És la implementació de la estratègia "pa xula jo"
- Classe CooperateAlwaysStrategy
  - És la implementació de la estratègia candida
- Classe RandomStrategy
  - És la implementació de la estratègia atzarosa
- Classe MoreUsedStrategy
  - És la implementació de la estratègia MoreUsedStrategy
- Classe MajorityRuleStrategy
  - És una classe composta per altres estratègies, que implementa la estratègia MajorityRuleStrategy
- Classe RoundInfo
  - Representa el resultat de punts pel parell d'accions escollides pels dos jugadors
- Classe Register
  - Serveix per guardar, sota un nom, les estratègies definides al joc
  - Quan se li demana una estratègia, donat un nom, retorna una còpia de l'estratègia guardada sota aquest
- Pair<T>
  - S'ha implementat principalment per representar les parelles de valors de la payoff matrix, tot i que també s'utilitza en alguns mètodes, com en getScore() de la classe Play per tal de retornar la puntuació dels dos jugadors.