

Money Heist

Albert Oliveras

27 de gener de 2026



# 1 Regles del Joc

La Fàbrica Nacional de Moneda i Timbre es veu novament amenaçada. Les gestes del Professor i els seus acompanyants són replicades per diversos grups amb la mateixa estructura: un professor amb altes habilitats intel·lectuals i un grup de soldats disposats a seguir les seves enginyoses ordres. Per tal de no aixecar massa soroll, no s'utiliza cap arma convencional, sinó silenciosos verins que es poden llençar als grups rivals. L'objectiu està clar: recollir la major quantitat de diners possible, possiblement eliminant també el major nombre d'unitats rivals.

Es tracta d'un joc per a quatre jugadors (quatre bandes) identificats amb números de 0 a 3. Cada jugador té el control d'una sèrie de soldats i d'un professor.

El joc té una durada de 200 rondes, numerades de la 0 la 199. Cada unitat, ja sigui professor o soldat, pot efectuar com a màxim una acció per ronda.

Cada soldat comença la partida amb una certa quantitat d'unitats de verí i cada professor amb una certat quantitat d'unitats de salut. El taulell està ple de kits de càrrega de verí i kits de salut. Quan una unitat recull un kit, aquest va a parar al pot comú de la seva banda. Tot soldat d'una banda pot accedir al seu pot de verins, i tot professor al seu pot comú de salut. En el tauler també hi ha caixes fortes, que guarden quantitats indecents d'or, i diners no protegits per caixes fortes.

Els professors es poden moure, intentar obrir una caixa forta o bé curar les seves ferides utilitzant el pot comú de salut. Els soldats es poden moure, potser atacant a unitats rivals, o bé recarregar verí utilitzant el pot comú de verí. Durant aquestes rondes les bandes aniran acumulant punts i guanyarà la partida qui tingui més punts en acabar l'última ronda.

El tauler del joc té dimensions  $60 \times 60$ . Les unitats no es poden moure en cap cas fora d'ell. Una posició del tauler ve determinada per un parell d'enters  $(f, c)$  on  $0 \leq f < 60$  i  $0 \leq c < 60$ . La posició de més a dalt i a l'esquerra és la  $(0, 0)$ , mentre que la de més a baix i a la dreta és  $(59, 59)$ . Per tant, la primera coordenada ( $f$  de fila) és la que indica la posició en l'eix vertical i la segona ( $c$  de columna) en l'eix horitzontal. Cada cel·la del tauler o bé forma part d'un passadís o bé és una paret. Les unitats no poden en cap cas travessar les parets i s'han de moure necessàriament pels passadissos.

Moviments dels professors. Un professor es pot moure pel tauler de la manera següent:

- Pot accedir a qualsevol de les 4 cel·les adjacents en horitzontal i vertical.
- Si es mou cap a una cel·la on hi ha una paret, el moviment s'ignorarà.
- Si es mou cap a una cel·la ocupa per una certa quantitat de diners, els recollirà i s'incrementaran els punts de la seva banda. En acabar la ronda, una altre objecte de tipus diners reapareixerà en una altra posició.

- Si es mou cap a una cel·la ocupada per un kit de salut, el pot comú de salut de la seva banda s'incrementarà, i el professor ocuparà la cel·la tot fent desaparèixer el kit. En acabar la ronda, un altre kit de salut reapareixerà en una altra posició. Notem que no trobarem mai una cel·la ocupada per un kit i un professor o soldat.
- Si es mou cap a una cel·la ocupada per un kit de càrrega de verí, el pot comú de verins de la seva banda s'incrementarà, i el professor ocuparà la cel·la tot fent desaparèixer el kit. En acabar la ronda, un altre kit de càrrega de verí reapareixerà en una altra posició. Notem que no trobarem mai una cel·la ocupada per un kit de càrrega de verí i un professor o soldat.
- Si es mou cap a una cel·la ocupada per l'entrada a una caixa forta, el moviment s'ignorarà. Més endavant explicarem com un professor pot obrir una caixa forta.
- Si es mou cap a una cel·la ocupada per una unitat de qualsevol banda, ja sigui professor o soldat, el moviment s'ignorarà.
- Si es mou cap a una cel·la buida (sense cap kit, unitat, caixa forta o diners), el moviment s'efectuarà.

Curacions dels professors. La salut de tot professor té un valor màxim  $M$ , que de fet coincideix amb la salut amb la que comença la partida. En qualsevol moment, un professor pot transferir punts de salut del pot comú de la seva banda per recarregar la seva pròpia salut fins a  $M$ . Si el pot comú no disposa de prou punts, s'utilitzaran tots els punts del pot comú (que quedarà buit) per incrementar la salut del professor, però aquesta evidentment quedarà per sota d' $M$ .

Obertura de caixes fortes. Una caixa forta conté una única casella amb or i, de les 8 caselles adjacents, 7 d'elles són parets impossibles de destruir, i una d'elles és la porta, que es pot obrir amb una certa clau secreta. Un exemple és la següent figura, on  $O$  és la casella que conté l'or i  $P$  la casella que conté la porta. Notem que  $P$  només pot estar just a sobre, sota, dreta o esquerra de l'or, mai en diagonal.

	X	X	X	
	X	O	X	
	X	P	X	
		A		

La casella  $A$  indica la casella d'accés a la caixa forta. Des d'aquesta casella, i escollint la direcció correcta (en aquest cas concret cap amunt) un professor, mai un soldat, pot intentar obrir la caixa forta.

Cada caixa forta té associada una pista, que consisteix en  $3n$  enters entre  $-20$  i  $20$ . Els nombres són diferents per a cada caixa, però  $n$  és sempre la mateixa. Per tal d'obrir la caixa forta, cal saber agrupar els nombres de la pista en  $n$  tripletes tals que un element de cada tripleta és la suma dels altres dos. Per exemple, si les pistes de la caixa són

2	3	6	-3	3	1	-3	0	1	0	3	3
---	---	---	----	---	---	----	---	---	---	---	---

una clau (no necessàriament única) per obrir la caixa forta és

0	2	1	2	3	0	1	2	0	3	3	1
---	---	---	---	---	---	---	---	---	---	---	---

que indica que la primera tripleta és  $\{2, 1, 1\}$ , que la segona tripleta és  $\{6, -3, 3\}$ , que la tercera tripleta és  $\{3, -3, 0\}$  i que la quarta tripleta és  $\{3, 0, 3\}$ . Per tant, les tripletes s'identificaran amb enters consecutius començant per zero. Sabem que sempre existeix almenys una clau.

Si un professor es troba a la casella d'accés  $A$ , pot obrir la caixa forta si sap la clau i identifica la direcció que cal per accedir a  $P$  des d' $A$ . Si la clau o la direcció són incorrectes, s'ignora l'acció. En cas contrari, la banda recull l'or, guanya una quantitat important de punts, i la caixa forta desapareix. En acabar la ronda, es genera una altra caixa forta amb una nova clau en una altra posició.

Moviments dels soldats. Un soldat es pot moure pel tauler de la manera següent:

- Pot accedir a qualsevol de les 4 cel·les adjacents en horitzontal i vertical.
- Si es mou cap a una cel·la on hi ha una paret, diners, una caixa forta, algun kit o una cel·la buida, el comportament és el mateix que si fos un professor.
- Si es mou cap a una cel·la ocupada per una unitat de la seva banda, ja sigui professor o soldat, el moviment s'ignorarà.
- Si es mou cap a una cel·la ocupada per una unitat rival, però ell no té verí, el moviment també s'ignorarà.
- Si disposa de verí i es mou cap a un soldat rival, aquest últim morirà. El soldat que ha iniciat l'atac no es mourà de cel·la i la seva banda rebrà una certa quantitat prefixada de punts. En acabar la ronda, un nou soldat reapareixerà en una altra posició. Amb un 70% de possibilitats pertanyerà a la banda del soldat guanyador, i amb un 10% a cadascuna de les altres tres bandes.
- Si disposa de verí i es mou cap a un professor rival, aquest últim utilitzarà el seu coneixement per minimitzar l'efecte del verí. No obstant, la seva salut es decrementarà en un nombre prefixat d'unitats. Si passa a ser zero o negativa, el professor morirà i trigarà un cert nombre de rondes a ser reemplaçat per un altre professor que pertanyerà a la mateixa banda. El soldat que ha iniciat l'atac no es mourà en cap cas, i si l'atac ha acabat

amb la mort del professor, la seva banda del soldat rebrà un quantitat prefixada de punts.

Càrrega de verí. Cada soldat pot portar un màxim nombre  $V$  d'unitats de verí. En qualsevol moment, un soldat pot transferir punts de verí del pot comú de la seva banda per recarregar la seva quantitat de verí fins a  $V$ . Si el pot comú no disposa de prou punts, s'utilitzaran tots els punts del pot comú (que quedarà buit) per incrementar el verí del soldat, però aquest nombre evidentment quedarà per sota de  $V$ .

Com a resultat de les anteriors regles, el nombre de soldats és constant a l'inici de cada ronda al llarg de tot el joc, però aquests podran anar canviant de banda. Pel que fa als professors, cada banda tindrà com a molt un professor viu. És important remarcar que les unitats tenen un identificador que mai canvia, ni tan sols quan canvien de banda. Aquests identificadors són naturals consecutius començant en 0.

La quantitat de kits de salut, de verí, de caixes fortes i de diners també és constant al llarg de tot el joc. Mentre que cada kit dona la mateixa quantitat de salut o verí, i cada caixa forta conté la mateixa quantitat d'or, no totes les cel·les amb diners en contenen la mateixa quantitat.

Regeneració de d'objectes. Cada vegada que cal regenerar un kit, diners, una caixa forta, un professor o soldat, aquest reapareixerà sempre en una cel·la buida  $C$  i tal que no hi ha cap unitat ni cap objecte en les posicions que l'envolten (les marcades amb una  $x$  a la taula):

x	x	x	x	x
x	x	x	x	x
x	x	C	x	x
x	x	x	x	x
x	x	x	x	x

Si en el moment de reaparèixer no existeix cap cel·la segura en aquest sentit, l'objecte reapareixerà en una cel·la buida, que no tingui cap unitat ni cap objecte.

Càlcul de la puntuació. En finalitzar una ronda, la puntuació d'una banda es veu incrementada si s'ha produït alguna de les següents situacions: s'ha matat un soldat o un professor, o bé s'han recollit diners o or.

Les constants que indiquen el nombres de punts obtinguts per cada aspecte, com d'altres que especifiquen els paràmetres inicials del joc, estan definides a l'arxiu d'entrada `default.cnf`. Totes les partides es jugaran amb exactament els valors donats en aquest arxiu.

Execució d'ordres. A cada ronda es pot donar més d'una ordre a la mateixa unitat, tot i que només se seleccionarà la primera d'elles (si n'hi ha alguna). Tot programa que intenti donar més de 1000 comandes durant la mateixa ronda s'avortarà.

Cada ronda, les ordres seleccionades dels quatre jugadors s'executaran amb ordre aleatori, però respectant l'ordre relatiu entre les unitats d'una mateixa banda. Com a conseqüència de la norma anterior, considereu la possibilitat de donar les ordres a les vostres unitats a cada ronda de més urgent a menys urgent.

Tingueu en compte que s'aplica cada moviment sobre el tauler que resulta dels moviments anteriors. Per exemple, considereu el tauler

x	x	x	x
x	M	U	x
x	V	x	x
x	x	x	x

on la cel·la M conté diners i U i V dos soldats de diferents bandes. Imaginem que el jugador que controla U ha decidit que aquest vagi cap a l'esquerra, i el jugador que controla V ha decidit que vagi amunt. Si s'executa primer el moviment de V, aleshores U s'ha quedat sense diners, perquè V ja se'ls ha quedat i a més, la posterior execució del moviment de U és un atac cap a V. Si U disposa de verí, aleshores V mor. En la visualització de la partida veurem una transició de la matriu anterior cap a una situació on M i V han desaparegut. Òbviament, no hi podia haver un atac de U cap V en la configuració inicial, perquè els soldats no es poden moure en diagonal, però l'ordre d'execució de les comandes sí que ho ha fet possible. Tingueu això en compte quan no entengueu certes situacions durant la visualització de les partides.

Després de l'execució de tots els moviments de les unitats, es regeneren els professors que toca. A continuació, reapareixen els soldats que han mort, potser amb un canvi de banda. Fem notar que els professors i soldats que reapareixen, ho fan amb els paràmetres que un soldat o professor té a l'inici de la partida pel que fa a salut i verí. Després, i en aquest ordre, es regeneren els diners, els kits de salut i de verí, i les caixes fortes que han desaparegut durant la ronda. Finalment, s'actualitza la puntuació.

## 2 El Visor

A continuació descrivim el visor de partides:

- A la part superior hi ha botons que permeten reproduir o pausar la partida, anar al començament o al final de la partida, activar o desactivar el mode d'animació o obtenir una finestra d'ajuda amb més maneres de controlar com es reproduceix la partida. També hi trobareu la ronda actual i un

botó per tancar el visor. Una barra de desplaçament horitzontal mostra visualment en quin punt de la partida es troba la ronda actual.

- A la columna de l'esquerra, apareix cada jugador amb el nom i color corresponents. A sota es mostra la puntuació actual, la quantitat d'unitats del pot comú de salut, les de verí i el nombre de soldats del jugador. A les partides jugades a Jutge.org, també es mostra el percentatge de temps de CPU que s'ha consumit fins ara (si està esgotat, s'indica amb un 'out'). A la part superior dreta apareixen els colors dels jugadors ordenats per puntuació.
- Les cel·les que són paret tenen color gris fosc, gairebé negre.
- Els soldats es representen amb un quadrat del color corresponent.
- Els professors es representen amb un quadrat del color corresponent i contorn exterior negre.
- Els kits de salut es representen amb un cercle blau i contorn exterior negre.
- Els kits de verí es representen amb un cercle vermell i contorn exterior negre.
- Els diners es representen amb un quadrat negre que conté una creu groga.
- L'or es representa amb un quadrat groc.
- La porta d'una caixa forta es representa amb un quadrat de color gris clar.



### 3 Com programar un jugador

El primer que heu de fer és descarregar-vos el codi font. Aquest inclou un programa C++ que executa les partides i un visualitzador HTML per veure-les

en un format raonable i animat. A més, us proporcionem un jugador “Null” i un jugador “Demo” per facilitar el començament de la codificació del jugador.

### 3.1 Executar la primera partida

Aquí us explicarem com executar el joc sota Linux, però hauria de funcionar també sota Windows, Mac, FreeBSD, OpenSolaris, ... Només necessiteu una versió recent `g++`, el `make` instal·lat al sistema, a més d'un navegador modern com Firefox, Chrome o Safari.

1. Obriu una consola i feu `cd` al directori on us heu descarregat el codi font.
2. Si, per exemple, teniu una versió de Linux en 64 bits, executeu:

```
cp AIDummy.o.Linux64 AIDummy.o
```

Amb altres architectures, cal escollir els objectes adequats que trobareu al directori.

3. Executeu

```
make all
```

per compilar el joc i tots els jugadors. Tingueu en compte que el `Makefile` identifica com a jugador qualsevol fitxer que coincideixi amb `AI*.cc`

4. Es crea un fitxer executable anomenat `Game`. Aquest executable us permet executar una partida mitjançant una comanda com la següent:

```
./Game Demo Demo Demo Demo -s 30 < default.cnf > default.res
```

Aquesta comanda comença una partida, amb la llavor aleatòria 30, amb quatre instàncies del jugador Demo, al tauler definit a `default.cnf`. La sortida d'aquesta partida es redirigeix a `default.res`.

5. Per veure una partida, obriu el fitxer visualitzador `viewer.html` amb el navegador, per exemple executant des d'un terminal la comanda `firefox viewer.html`, i carregueu el fitxer `default.res`.

Utilitzeu

```
./Game --help
```

per veure la llista de paràmetres que es poden usar. Particularment útil és

```
./Game --list
```

per veure tots els noms de jugadors reconeguts.

En cas que sigui necessari, recordeu que podeu executar

```
make clean
```

per esborrar l'executable i els objectes i començar la compilació de nou.



### 3.2 Afegir el vostre jugador

Per crear un jugador nou amb, per exemple, nom `Tokyo`, copieu `AInull.cc` (un jugador buit que proporcionem com a plantilla) a un fitxer nou `AITokyo.cc`. A continuació, editeu el fitxer nou i canvieu la línia

```
#define PLAYER_NAME Null
```

a

```
#define PLAYER_NAME Tokyo
```

El nom que trieu pel vostre jugador ha de ser únic, no ofensiu i tenir com a màxim 12 caràcters. Aquest nom es mostrarà al lloc web i durant les partides.

A continuació, podeu començar a implementar el mètode `virtual play()`, heretat de la classe base `Player`. Aquest mètode, que serà cridat a cada ronda, ha de determinar les ordres que s'enviaran a les vostres unitats.

Podeu utilitzar definicions de tipus, variables i mètodes a la vostra classe de jugador, però el punt d'entrada del vostre codi serà sempre el mètode `play()`.

Des de la vostra classe jugador també podeu cridar funcions que trobareu especificades als arxius següents:

- `State.hh`: accedir a l'estat del joc.
- `Action.hh`: donar ordres a les vostres unitats.
- `Structs.hh`: estructures de dades útils.
- `Settings.hh`: accedir als paràmetres del joc.
- `Player.hh`: mètode `me()`.
- `Random.hh`: generar nombre aleatoris.

Trobareu un resum de tota aquesta informació a l'arxiu `api.pdf`. També podeu examinar el codi del jugador “Demo” a `AIDemo.cc` com a exemple de com usar aquestes funcions.

Tingueu en compte que no heu d'editar el mètode `factory()` de la classe del vostre jugador, ni l'última línia que afegeix el vostre jugador a la llista de jugadors disponibles.

### 3.3 Restriccions en enviar el vostre jugador

Quan creieu que el vostre jugador és prou fort per entrar a la competició, podeu enviar-lo al Jutge. Degut a que s'executarà en un entorn segur per prevenir trampes, algunes restriccions s'apliquen al vostre codi:

- Tot el vostre codi font ha d'estar en un sol fitxer (com `AITokyo.cc`).

- No podeu utilitzar variables globals (en el seu lloc, utilitzeu atributs a la vostra classe).
- Només teniu permès utilitzar biblioteques estàndard com `iostream`, `vector`, `map`, `set`, `queue`, `algoritme`, `cmath`, ... En molts casos, ni tan sols cal incloure la biblioteca corresponent.
- No podeu obrir fitxers ni fer cap altra crida a sistema (threads, forks, ...)
- El vostre temps de CPU i la memòria que utilitzeu seran limitats, mentre que no ho són al vostre entorn local quan executeu `./Game`.
- El vostre programa no ha d'escriure a cout ni llegir de cin. Podeu escriure informació de depuració a cerr, però els vostres enviaments al Jutge no han d'escriure res, ni tan sols a cerr. En cas de fer-ho, el comportament del vostre jugador pot no ser l'esperat.
- Qualsevol enviament al Jutge ha de ser un intent honest de jugar. Qualsevol intent de fer trampes de qualsevol manera serà durament penalitzat.
- Un cop hagueu enviat un jugador al Jutge que hagi derrotat al Dummy, podeu fer més enviaments però haureu de canviar el nom del jugador. És a dir, un cop un jugador ha vençut al Dummy, el seu nom queda bloquejat i no es pot reutilitzar.

## 4 Consells

- NO DONEU O DEMANEU EL VOSTRE CODI A NINGÚ. Ni tan sols una versió antiga. Ni fins i tot al vostre millor amic. Ni tans sols d'estudiants d'anys anteriors. Utilitzem detectors de plagi per comparar els vostres programes, també contra enviaments de jocs d'anys anteriors. No obstant, podeu compartir arxius objecte.  
  
Qualsevol plagi implicarà una nota de 0 en l'assignatura (no només del Joc) de tots els estudiants involucrats. Es podran també prendre mesures disciplinàries addicionals. Si els estudiants A i B es veuen implicats en un plagi, les mesures s'aplicaran als dos, independentment de qui va crear el codi original. No es farà cap excepció sota cap circumstància.
- Abans de competir amb els companys, concentreu-vos en derrotar al Dummy.
- Llegiu les capçaleres de les classes que aneu a utilitzar. No cal que mireu les parts privades o la implementació.
- Comenceu amb estratègies simples, fàcils d'implementar i depurar, ja que és exactament el que necessitareu al principi.
- Definiu mètodes auxiliars senzills (però útils) i assegureu-vos que funcionin correctament.

- Intenteu mantenir el vostre codi net. Això farà més fàcil canviar-lo i afegir noves estratègies.
- Com sempre, compileu i proveu el vostre codi sovint. És molt més fàcil rastrejar un error quan només heu canviat poques línies de codi.
- Utilitzeu `cerr` per produir informació de depuració i afegiu **asserts** per assegurar-vos que el vostre codi fa el que hauria de fer. Recordeu eliminar (o comentar) els **cerr** abans d'enviar el vostre codi al Jutge.org.
- Quan depureu un jugador, elimineu els cerrs que tingueu en el codi d'altres jugadors, per tal de veure només els missatges que desitgeu.
- Podeu utilitzar comandes com el **grep** de Linux per tal de filtrar la sortida produïda per **Game**.
- Activeu l'opció **DEBUG** al Makefile, que us permetrà obtenir traces útils quan el vostre programa avorta. També hi ha una opció **PROFILE** que podeu utilitzar per optimitzar codi.
- Si l'ús de `cerr` no és suficient per depurar el vostre codi, apreneu com utilitzar **valgrind**, **gdb** o qualsevol altra eina de depuració.
- Podeu analitzar els arxius produïts per **Game**, que descriuen com evoluciona el tauler a cada ronda.
- Conserveu una còpia de les versions antigues del vostre jugador. Feu-lo lluitar contra les seves versions anteriors per quantificar les millores.
- Assegureu-vos que el vostre programa sigui prou ràpid. El temps de CPU que es permet utilitzar és bastant curt.
- Intenteu esbrinar les estratègies dels altres jugadors observant diverses partides. D'aquesta manera, podeu intentar reaccionar als seus moviments, o fins i tot imiteu-los o milloreu-los amb el vostre propi codi.
- No espereu fins al darrer minut per enviar el jugador. Quan hi ha molts enviaments al mateix temps, el servidor triga més en executar les partides i podria ser ja massa tard!
- Podeu enviar noves versions del vostre programa en qualsevol moment.
- Recordeu: mantingueu el codi senzill, compileu-lo sovint i proveu-lo sovint, o us en penedireu.