

# Othello Jokoaren Memoria

Garai eta Urdax

2025eko Maiatzaren 15a

## 1. Sarrera

Dokumentu honek gure taldearen Othello jokoaren garapena zehazten du, CLIPS programazio-lengoaian garatua. Proiektuaren helburua mahai jokia simulatzea da, erabiltzailearen aurka jokatzeko gaitasuna duen algoritmoz osatutako adimen artifizial bat sortuz, non ordenagailuak estruktura heuristikoa desberdinak erabiltzen dituen mugimendu optimoak hautatzeko. Memoria honetan erabilitako datu-egiturak, estrategiak, erregelak eta lortutako ondorioak deskribatzen ditugu.

## 2. Erabilitako Datuen Deskribapena

Jokoaren funtzionamendua kudeatzeko, hainbat datu-egitura definitu dira CLIPS-en plantilla gisa. Hauek dira nagusiak:

- **Tablero:** Jokoaren taularen egoera orokorra adierazten du. 8x8 dimentsioko taula bat da, eta uneko txanda (*negro* edo *blanco*) gordetzen du.
- **Casilla:** Taulako gelaxka bakoitza adierazten du, bere koordenatuekin (*fila* eta *columna*) eta egoerarekin (*vacía*, *negra*, *blanca*).
- **Fichas-disponibles:** Jokalari bakoitzak (beltza edo zuria) dituen fitxa kopurua kontrolatzen du, hasieran 32 fitxa jokalari bakoitzeko.
- **Mensaje:** Erabiltzaileari mezuak erakusteko erabiltzen da, bereziki adimen artifizialaren ekintzak adierazteko.
- **Visualizar:** Taularen bistaratzea kontrolatzen du, noiz erakutsi behar den adieraziz.
- **Game-state:** Jokoaren egoera kudeatzen du, bereziki mugimendu legalik ez dagoenean txandak aldatzeko eta/edo jokia amaitzeko.

## 3. Erabilitako Estrategiak

Othello jokoaren inplementazioan, hainbat arau heuristikoa erabili dira ordenagailuaren estrategiak eta taktikak kontsistentzia batekin emateko, batez ere *auto-white* moduan (ordenagailuaren mugimendu automatikoak zuriertzat). Horretaz gain, hainbat kondizio eta estruktura esleitu zaizkio jokuari jokalariek eta adimen artifizialak arauetatik ez ateratzeko

### 3.1. Teknikak

- **Taulako mugimendu legalen identifikazioa:** `mugimendu_egokia` funtzioak gelaxka bakoitza aztertzen du, baldintza hauek egiaztatuz:
  - Gelaxka taulan dagoela (1etik 8ra arteko koordinatuak).
  - Gelaxka hutsik dagoela (`okupatuta_dago` funtzioaren bidez).
  - Mugimenduak gutxienez fitxa bat flankeatzen duela, kontrako koloreko fitxak inguratuz.
- **Fitxen flankea:** `AP` funtzioak mugimendu baten ondoren flankeatutako fitxak identifikatzen eta aldatzen ditu, zortzi norabideetan (horizontala, bertikala eta diagonalak) bilatuz.
- **Txanda automatikoen kudeaketa:** `mostrar-movimientos-disponibles` funtzioak mugimendu posibleak erakusten ditu eta, mugimendurik ez badago, txanda automatikoki aldatzen du. Bi txanda jarraian mugimendurik ez badago (bi jokalaria ataskatuta), jokoa amaitzen da.

### 3.2. Heuristikoak

Ordenagailuaren mugimenduak (*auto-white*) optimizatzeko, `calcular-volteos` funtzioan oinarritutako puntuazio-sistema bat erabili da. Mugimendu bakoitzaren balioa kalkulatu da, honako irizpideetan oinarrituta:

- **Flankeatutako fitxa kopurua:** Mugimendu bakoitzak zenbat fitxa kontrario flankeatzen dituen kontatzen da.
- **Kokapen estrategikoen lehentasuna:**
  - **Izkinak:** +5 puntu ematen zaizkie, egonkortasun handia dutelako (ezin dira flankeatu).
  - **Ertzak:** +2 puntu, izkinak baino egonkorragoak direlako baina ez hainbeste.
  - **Ertzetako fitxa kontrarioak:** Flankeatutako fitxak taularen ertzetan badaude, 1.5 puntu balio dute, estrategikoki garrantzitsuak direlako.
- **Posizio estrategikoen defentsa:** Mugimendu bat egitean, kontrarioak posizio estrategiko bat (4 fitxa edo gehiago flankeatzen dituen) jokalaria ezeztatu ahal zaion egiaztatzen da. Horrela bada, 2 puntu gehigarri ematen zaizkio mugimenduari, defentsa indartzeko.

Mugimendu posible guztien artean, puntuazio altuena dutenak hautatzen dira, eta berdinketarik badago, ausaz aukeratzen da bat. Ausazkotasunak jokoa dinamikoagoa mantentzen du eta bermatzen da algoritmo ez determinista (eta ondorioz, zailagoa aurreikusten arerioarentzat) bat dugula.

## 4. Erregelen Egitekoa

Proiektuan erabilitako erregela nagusien deskribapena hauxe da:

- **inicializar-tablero:** Taula hasieratzen du, 8x8 dimentsioekin eta beltzen txanda-rekin. *Game-state* eta *auto-white* egoerak ere aktibatzen ditu.
- **inicializar-fichas:** Jokalari bakoitzari 32 fitxa esleitzen dizkio hasieran.
- **crear-casillas-vacias:** Taulako gelaxka guztiak hutsik sortzen ditu.
- **colocar-fichas-iniciales:** Taularen erdian hasierako fitxak jartzen ditu (bi beltz eta bi zuri) eta fitxa kopurua eguneratzen du.
- **mostrar-tablero:** Taularen egoera bistaratzen du, gelaxken egoera (*u* hutsentzat, *N* beltzentzat, *B* zurientzat) eta fitxa kopurua erakutsiz.
- **mostrar-mensaje:** Erabiltzaileari unetik unerako jokoaren detaileak espezifikatzen dizkio.
- **auto-white-move:** Zuriaren txandan, mugimendu posibleak aztertzen ditu, puntuazio altuena duena hautatzen du eta (AP) funtzioaren bidez exekutatzen du.

## 5. Lortutako Ondorioak

Proiektua garatu ondoren, honako ondorioak atera dira:

- **Funtzionalitatea:** Sistemak Othello jokoaren arauak behar bezala inplementatzen ditu, mugimendu legalak egiaztatuz, fitxak flankeatuz eta txandak kudeatuz. *Auto-white* moduak erabiltzailearen aurka jokatzeko gaitasuna ematen du.
- **Estrategien eraginkortasuna:** Erabilitako heuristikoei (izkinak, ertzak eta posizio estrategikoen defentsak) ordenagailuaren joko nabarmen hobetzen dute, mugimendu adimentsuagoak eginez. Honela, beste algoritmo komplexuegiek baino oinarritzko errendimendu hobeagoa du, kalkulu kantitate murriztuago batekin.
- **Programazio forma berria:** Arauen bidezko programazio estrategia bat erabiltzeak eta ulertzeak etorkizunean gure aplikazioak garatzeko modu desberdin bat ikasteko aukera eman digu.