

Algoritme de generació

Per la generació de Taulers Enunciat de kakuros hem decidit que es poden fixar 4 paràmetres que configuren la generació: Els elementals, que són les dimensions x i y, i dos més que en par afecten a la dificultat del kakuro i que ens serviran per implementar els diferents nivells de dificultat: el número de caselles blanques totals del tauler, i el número de caselles blanques amb un valor ja establert.

Combinacions

Com que hem vist que la quantitat de combinacions existents d'un kakuro està bastant limitat, hem creat una classe que es diu Combinacions on hi té hardcoded tot les possible combinacions, aquesta classe serà el nostre suport principal per els algoritmes de la classe Algoritme.

Solucionador de kakuro

Tal i com hem indicat a l'apartat anterior, el nostre solucionador utilitzarà les combinacions ja hardcoded per millorar en la eficiència del programa. Ficarem un TauleEnunciat(Kakuro buit), i retornarà un TaulerComencat (Kakuro amb valors introduïts) en aquest cas resolt, en cas de que no trobi solució, el mètode retornarà un null.

Modes de joc

Hem decidit que podem implementar 2 modes de joc: El Arcade, i el contrarellotge. L'arcade és el kakuro bàsic, original, i on només es cronometra a l'usuari. En la modalitat contrarellotge en canvi, l'usuari té un temps limitat per resoldre el kakuro.

Els Rànking

Hem pensat dos possibles sistemes de rànking:

1. Un rànking per cada TaulerEnunciat del repositori d'Enunciats.
2. Un rànking per Dificultat

Identificació de cada TaulerEnunciat:

Per identificar de forma única cada tauler i alhora poder detectar generacions o propostes duplicades, creiem convenient establir una identificació mitjançant el seu hash, i hem optat per utilitzar MD5, que és senzill d'utilitzar, i té prou bona dispersió i la probabilitat de col·lisió és baixa.

Usuaris:

Un usuari tindrà un nom i una contrasenya associats, això implica que no poden haver-hi dos usuaris amb el mateix nom. Es podrà canviar la contrasenya.

Configuració:

Hem pensat que, com que cadascú té unes preferències a l'hora de triar quin kakuro fer, s'hauria de poder redefinir la configuració **per defecte** a nivell d'usuari (cada usuari tindrà la seva configuració per defecte, que no és necessàriament la mateixa que la d'una partida).

Repositoris:

Hem pensat de fer dos possibles repositoris:

1. Un repositori general (amb tots els kakuros pujats)

2. Un repositori de l'usuari (amb els kakuros pujats de l'usuari)

Pausar/Continuar partida:

Creiem convenient l'opció de pausar partida, on s'amagarà la pantalla de joc per tal que l'usuari no vegi el tauler que està jugant, per evitar que es facin trampes, sobretot en el mode contrarellotge.

Kakuro:

Hem decidit que un kakuro tindrà només una solució. Per tant, tots els que generem tindran únicament una solució.

Pistes:

A més de la pista que se'ns demana a l'enunciat, hem decidit que, ja que els nostres kakuros tenen una solució, podem permetre a l'usuari (a canvi de perdre el joc) mostrar la solució final del kakuro.

Testing:

Hem decidit fer testing incremental: vam començar implementant les classes més elementals, com la classe Casella, per tal d'evitar haver de fer Stubs, i així poder implementar les classes "superiors" amb aquestes primeres ja fetes, la qual cosa creiem que ens ha facilitat bastant el desenvolupament.