



Títol: Interfície d'un joc de Scrabble

Volum: 1/1

Alumne: Xavier Vilagrasa Ventura

Director/Ponent: Marc Vigo Anglada

Departament: Llenguatges i Sistemes Informàtics

Data: 2 - 7 - 2009





DADES DEL PROJECTE

Títol del Projecte: Interfície d'un joc de Scrabble

Nom de l'estudiant: Xavier Vilagrasa Ventura

Titulació: Enginyeria Informàtica Superior

Crèdits: 37,5

Director/Ponent: Marc Vigo Anglada

Departament: Llenguatges i Sistemes Informàtics (LSI)

MEMBRES DEL TRIBUNAL

President: Ernest Teniente Lopez

Vocal: Natalia Sadovskaia Nurimanova

Secretari: Marc Vigo Anglada

QUALIFICACIÓ

Qualificació numèrica:

Qualificació descriptiva:

Data:





Índex

Índex	5
1 INTRODUCCIÓ	11
1.1 RESUM	11
1.2 HISTÒRIA	12
1.2.1 ORÍGENS DEL JOC	12
1.2.2 EL SCRABBLE A CATALUNYA	14
1.3 ORGANITZACIÓ DE LA MEMÒRIA	15
1.4 JUSTIFICACIÓ	16
1.5 OBJECTIUS	17
1.6 TECNOLOGIES UTILITZADES	18
1.6.1 JOGL	18
1.6.2 NETBEANS COM A EINA DE PROGRAMACIÓ	20
2 ANÀLISI DE REQUERIMENTS	21
2.1 INTRODUCCIÓ	21
2.1.1 ÀMBIT DEL DOCUMENT	21
2.1.2 NOUS REQUERIMENTS	22
2.2 DESCRIPCIÓ GENERAL	24
2.2.1 PERSPECTIVA DE L'APLICACIÓ	25
2.2.2 CARACTERÍSTIQUES DE L'USUARI	26
2.2.3 REQUISITS DEL SISTEMA	26
3 ESPECIFICACIÓ	27
3.1 INTRODUCCIÓ ALS CASOS D'ÚS	27
3.1.1 QUÈ ÉS UN CAS D'ÚS	27
3.1.2 A QUI AFECTA EL CAS D'ÚS	27
3.1.3 COM ES PODEN IDENTIFICAR	28
3.1.4 FINS A QUIN PUNT DESCOMPONEM ELS CASOS D'US	28
3.1.5 OBJECTIUS GENERALS	29
3.1.6 OBJECTIUS D'USUARI	29
3.1.7 SUBOBJECTIUS	29
3.1.8 ERRORS HABITUALS	29



3.2	LLISTATS CASOS D'ÚS	30
3.3	DIAGRAMA DE CASOS D'ÚS	32
3.3.1	GESTIÓ DE JUGADORS	33
3.3.2	GESTIÓ DE DICCIONARIS	34
3.3.3	GESTIÓ DE REGLES	34
3.3.4	GESTIÓ DEL RÀNKING	34
3.3.5	GESTIÓ D'IDIOMES	35
3.3.6	GESTIÓ DE PARTIDES	35
3.4	ESPECIFICACIÓ CASOS D'ÚS	38
3.4.1	ESPECIFICACIÓ: GESTIÓ DE DICCIONARIS	38
3.4.2	ESPECIFICACIÓ: GESTIÓ DE REGLES	43
3.4.3	ESPECIFICACIÓ: GESTIÓ DE PARTIDES	43
3.5	MODEL DE COMPORTAMENT	58
3.5.1	DIAGRAMES DE SEQÜÈNCIA	58
3.5.2	CONTRACTES DE LES OPERACIONS	66
3.6	MODEL DE DADES	75
3.6.1	DIAGRAMA D'ESTATS DE CLASSES	75
3.6.2	DIAGRAMES PRINCIPALS	76
4	DISSENY	77
4.1	CAPA DE PRESENTACIÓ	78
4.1.1	DIAGRAMA CAPA PRESENTACIÓ	78
4.1.2	DISSENY DE LES PANTALLES	81
4.2	CAPA DE DADES	90
4.2.1	PERSISTÈNCIES DE LES DADES	90
4.2.2	CANVIS EN ELS FITXERS DE L'APLICACIÓ	91
4.3	CAPA DE DOMINI	98
4.3.1	DIAGRAMA DE CLASSES	99
4.3.2	PATRONS DE DISSENY: CONTROLADOR	101
4.3.3	CANVI EN EL DISSENY DE CARREGA DE DICCIONARIS	102
4.3.4	DIAGRAMES DE SEQÜÈNCIA	103
4.3.5	DIAGRAMA DE SEQÜÈNCIA: afegirParaula	103
4.3.6	DIAGRAMA DE SEQÜÈNCIA: eliminarParaula	104
4.3.7	DIAGRAMA DE SEQÜÈNCIA: consultarParaula	105
4.3.8	DIAGRAMA DE SEQÜÈNCIA: definirCategoriesParaules	106
4.3.9	DIAGRAMA DE SEQÜÈNCIA: realitzarSegüentTorn	107
4.3.10	DIAGRAMA DE SEQÜÈNCIA: mostrarAjuda	108



4.3.11	DIAGRAMA DE SEQÜÈNCIA: pausarTemps, tornarPartida	108
4.3.12	DIAGRAMA DE SEQÜÈNCIA: veureSac	109
5	IMPLEMENTACIÓ	111
5.1	INTRODUCCIÓ	111
5.2	CAPA DE PRESENTACIÓ	111
5.2.1	IMPLEMENTACIÓ PANTALLES	112
5.3	CAPA DE DADES	123
5.3.1	DIRECTORIS DE FITXERS	124
5.4	CAPA DE DOMINI	124
5.4.1	CANVIS EN DICCIONARI	124
5.4.2	CANVIS EN LA INTEL·LIGÈNCIA ARTIFICIAL DELS ROBOTS	125
5.5	IMPLEMENTACIÓ DE CONTROL D'ERRORS NO CONTROLATS	125
5.6	PROVES DEL SISTEMA	126
6	CONCLUSIONS FINALS	127
6.1	APORTACIÓ PERSONAL	127
6.2	POSSIBLES MILLORES EN L'APLICACIÓ	128
7	BIBLIOGRAFIA	129
7.1	LLIBRES CONSULTATS	129
7.2	REFERÈNCIES CONSULTADES	129
8	ANNEX 1. ESTUDI TEMPORAL I ECONÒMIC	131
8.1	PLANIFICACIÓ INICIAL	131
8.1.1	PLANIFICACIÓ DE TASQUES	131
8.1.2	PLANIFICACIÓ TEMPORAL	132
8.1.3	DIAGRAMA GANTT	133
8.1.4	ESTUDI ECONÒMIC	134
9	ANNEX 2. MANUAL D'INSTAL·LACIÓ	137
9.1	INTRODUCCIÓ	137
9.2	JAVA DEVELOPMENT KIT	137
9.3	INSTAL·LACIÓ BIBLIOTEQUES OPENGL	138
9.3.1	WINDOWS	138



9.3.2	LINUX.....	139
9.4	EXECUCIÓ DEL PROGRAMA.....	139
10	ANNEX 2. MANUAL D'USUARI.....	141
10.1	INTRODUCCIÓ	141
10.2	INICI PROGRAMA	141
10.3	PANTALLA PRINCIPAL	142
10.4	GESTIÓ DE JUGADORS	144
10.4.1	CONSULTAR ESTADÍSTIQUES JUGADOR.....	146
10.4.2	ELIMINAR JUGADOR	146
10.4.3	CREAR UN JUGADOR NOU	147
10.5	GESTIÓ DICCIONARIS.....	148
10.5.1	CONSULTAR PARAULA.....	149
10.5.2	ESBORRAR PARAULA.....	149
10.5.3	AFEGIR PARAULA.....	149
10.5.4	GESTIONAR CATEGORIES DE PARAULES	150
10.6	GESTIÓ DE RÀNKING.....	151
10.7	MODIFICAR REGLES.....	152
10.7.1	CARACTERÍSTIQUES DE LA GESTIÓ DEL TEMPS	153
10.8	CARREGAR PARTIDA.....	154
10.9	CANVIAR D'IDIOMA.....	155
10.10	JUGAR AL SCRABBLE.....	156
10.10.1	PREPARANT LA PARTIDA.....	156
10.10.2	CALCULANT EL TORN DE LA PARTIDA.....	158
10.10.3	PANELL PARTIDA	159
10.11	AMPLIACIÓ: CREACIÓ DICCIONARIS.....	173
11	ANNEX 3. INSTRUCCIONS DEL SCRABBLE	175
11.1	NORMES BÀSIQUES DEL SCRABBLE.....	175
11.1.1	INICI DE LA PARTIDA.....	175
11.1.2	DESENVOLUPAMENT.....	175
11.1.3	AGAFAR I CANVIAR PECES.....	177
11.1.4	FINAL DE LA PARTIDA	178
11.1.5	DICCIONARI ARBITRAL.....	179



11.1.6	CONTROL DEL TEMPS.....	179
11.2	NORMES DE LES PARTIDES DUPLICADES	180
11.2.1	DEFINICIÓ.....	180
11.2.2	NORMES DE LA PARTIDA.....	180
11.2.3	FUNCIONAMENT	180
11.2.4	FULL D'AJUDA PER A PARTIDES DUPLICADES.....	182
12	ANNEX 4. GLOSSARI.....	183



1 INTRODUCCIÓ

1.1 RESUM

El Projecte Final de Carrera *Interfície d'un Joc de Scrabble* és una aplicació basada en el projecte cursat anteriorment “*Un joc de Scrabble*” realitzat per l'estudiant Rafael Olivera i es basa en un programa que permet jugar al famós joc de paraules creuades *Scrabble*.

El nostre projecte consisteix en programar de nou tota la part visual del joc i afegir noves funcionalitats. Des del punt de vista d'un usuari sense coneixements previs pot semblar que aquest PFC és un programa totalment diferent al joc anterior, però en realitat és una renovació completa de la interfície que l'usuari veurà, tot i que es continuarà utilitzant una part del projecte anterior.

El Scrabble és un joc molt popular en què poden participar fins a quatre jugadors. Cada un disposa d'unes fitxes amb lletres que ha de combinar per formar una paraula vàlida en el seu idioma. Les paraules es col·loquen en un taulell lligades a les ja existents, formant una estructura travada semblant als mots encreuats. Cada lletra té un valor determinat i algunes caselles del taulell multipliquen aquest valor. Guanya la persona que, en haver acabat les fitxes, suma més puntuació. Per validar una paraula jugada s'acostuma a fer servir un diccionari estàndard.

Aquesta aplicació permetrà jugar al joc Scrabble a través de l'ordinador amb diverses opcions extres respecta al joc de taula. Estarà enfocada tant a jugadors novells com a jugadors experts en el joc, ja que tindrà diversos nivells de dificultat per jugar contra robots simulats per la màquina. Per fer el joc més visual es mostrarà una representació del taulell en 3D que el jugador podrà veure durant tota la partida.



El joc tindrà diversos idiomes i les paraules es validaran a partir de llistes de paraules que l'usuari podrà personalitzar.

1.2 HISTÒRIA

A continuació comentarem breument l'evolució que ha tingut aquest joc a través de la història, i com s'ha popularitzat el joc a Catalunya.

1.2.1 ORÍGENS DEL JOC

A principis dels anys 30, després de graduar-se a la Universitat de Pennsylvania, un arquitecte sense feina anomenat Alfred Mosher Butts es decidí a dissenyar un joc de taula. Després d'estudiar els jocs existents, va concloure que tots els jocs existents s'ajustaven a tres categories diferents: jocs de nombres, com els daus o el bingo; jocs de moviment de peces, com els escacs o les dames; i jocs de paraules, com els anagrames.

Butts va decidir crear un joc que emprés d'igual manera sort i habilitat, tot combinant elements dels anagrames, dels trencaclosques i dels encreuats, uns passatemps populars ja als anys 20. Els jugadors prendrien set peces amb lletres impreses d'una bossa, i tot seguit intentarien formar paraules amb aquestes set lletres. La clau del joc seria l'anàlisi que Butts va fer de la llengua anglesa: Butts va analitzar les pàgines del New York Times per tal de calcular amb quina freqüència s'utilitza cada lletra de l'alfabet. Tot seguit, utilitzaria la freqüència de cada lletra per determinar quantes lletres de cada inclouria en el joc. Com a curiositat, només va incloure 4 "S", per tal que la possibilitat de fer el plural de les paraules no fes el joc massa fàcil.

Inicialment anomenat "Lexiko", més tard Butts va canviar el nom pel de "Criss Cross Words", i va començar a buscar un comprador. Els fabricants amb els quals va contactar al principi refusaren la idea, però Butts es va mantenir tenaç. Finalment va vendre els drets a l'emprenedor amant dels jocs James Brunot. Brunot hi va fer alguns ajustaments menors al disseny, i va canviar el nom del



joc pel de "Scrabble", un mot que significa més o menys "gratar frenèticament" (de l'holandès "Schrabben", gratar, fregar, esgarrapar).

El 1948 el joc ja era una marca registrada, i James Brunot i la seva dona van convertir una escola abandonada de Dodgington, Connecticut, en una fàbrica de Scrabble. El 1949 els Brunot van fabricar 2.400 jocs, però van perdre la suma de 450\$. El joc, tot i així, anava guanyant popularitat de mica en mica.

Segons diu la llegenda, el "boom" de el Scrabble arribà el 1952 quan Jack Strauss, president de les botigues Macy's, va jugar a aquest joc durant les seves vacances. Després de tornar de vacances, es va sorprendre en veure que la seva botiga no disposava d'aquest joc i va fer una gran comanda tot declarant que, en un any, "tothom n'havia de tenir un". El creixement va ser tan gran que, el mateix any 1952, veient que ja no podien donar abast a la demanda, els Brunot van traspasar els drets de fabricació i distribució a l'empresa Selchow and Righter, amb seu a Long Island, curiosament una de les que inicialment havia rebutjat la idea.

Selchow & Righter van comprar la marca registrada el 1972. J. W. Spear & Sons van començar a vendre el joc a Austràlia i al Regne Unit el 1955 i actualment són una filial de l'empresa Mattel. El 1986, Selchow and Righter van vendre el joc a Coleco, que aviat cauria en bancarrota i tots els seus drets serien comprats per Hasbro.



1.2.2 EL SCRABBLE A CATALUNYA

Des de 1990, a la trentena de versions del joc, entre les quals la russa, la txeca, l'hebrea o l'àrab, s'hi va afegir la catalana. Aquell any, a instàncies de l'expert en jocs Oriol Comas i amb el patrocini del Departament d'Ensenyament de la Generalitat de Catalunya, l'editor mataroní Borràs va editar la versió catalana basada en un rigorós estudi de les freqüències d'aparició de cada lletra en català.

El resultat fou un alfabet català de Scrabble en el qual les fitxes de més valor (10 punts) són els dígrafs L·L i NY i les lletres Ç i X les quals només apareixen un cop, en l'altre extrem hi ha 13 fitxes E de només un punt. També hi ha comodins batejats amb el nom d'"escarrassos" per indicació de l'escriptor Avel·li Artís-Gener, Tísner.

De fet, en Tísner va ser l'introduïdor de el Scrabble en l'àmbit dels Països Catalans. El pare dels mots encreuats en català, a primers dels 70, havia adaptat manualment un taulell anglès i hi jugava tot sovint amb la seva dona Lluïsa i la gent de la quinta d'en Jaume Fuster, M. Antònia Oliver i Montserrat Roig. També va ser el director de la primera partida duplicada, el 26/3/90, al Saló de l'Ensenyament entre cinquanta parelles d'estudiants, direcció que repetiria en el saló de l'any 1991 i en una altra partida duplicada a la localitat mallorquina d'Artà, en una trobada de professors de català a les Balears.

L'aparició de el Scrabble en català va provocar el naixement d'una certa scrabblemanía en cercles reduïts. El 1991 es va iniciar a Barcelona la primera lliga catalana de Scrabble, la Lliga CUSCA on jugadors que provenien de l'enigmística professional, la correcció lingüística, el periodisme, la literatura, les ciències l'administració, la docència, etc. s'aplegaven al voltant d'un taulell per jugar amb la llengua. Des de llavors, el nombre de jugadors i de clubs ha anat creixent arreu del territori.

De la bona rebuda que aquest joc de paraules va tenir a casa nostra n'és indicatiu el fet que els lectors de l'AVUI han estat gaudint durant molt de temps al joc gràcies a la secció creada per en Miquel Sesé a la pàgina de passatemps (actualment s'ha substituït per un joc que està en creixement, els "Sudokus").

1.3 ORGANITZACIÓ DE LA MEMÒRIA

Aquest document està estructurat de la forma següent:

Introducció: En aquest apartat comentarem breument en què consisteix aquest PFC, les motivacions per realitzar aquest projecte, i en quines tecnologies s'ha programat.

Anàlisi de requeriments: Mostrarem totes les funcions que el programa realitzarà i les ampliacions afegides respecte al PFC anterior.

Especificació: Un cop sapiguem totes les funcionalitats que el programa ha de realitzar podrem mostrar les operacions més importants que l'usuari haurà de dur a terme en l'aplicació.

Disseny: Gràcies a l'etapa d'especificació sabrem les operacions que s'han de dur a terme. En aquest apartat decidirem com haurem de dissenyar el programa per poder dur a terme totes les operacions definides.

Implementació: Comentarem els detalls més importants de la implementació que hàgim realitzat basant-nos en el disseny comentat en la fase anterior.

Estudi temporal i econòmic: Mostrarem el temps que hem trigat a realitzar el projecte i el cost econòmic invertit en el projecte.

Conclusions: Coneixements adquirits i conclusions sobre la realització de tot el projecte.

Bibliografia

Annexos: En aquest apartat inclourem el manual d'usuari del programa, el manual d'instal·lació, les instruccions del joc i finalment un glossari de termes relacionats amb el Scrabble.



1.4 JUSTIFICACIÓ

El Projecte Final de Carrera serà la consolidació d'una etapa molt important de la meua vida que he escollit i estic a punt de finalitzar, per tant, no considero que l'elecció del projecte sigui una acció trivial.

Fent un recorregut a la meua carrera i a tots els projectes que he anat programant, i, perquè no dir-ho, patint també, m'he adonat què amb els projectes que més orgullós m'he sentit era en aquells programes amb una interfície visual que permetia a usuaris sense coneixements informàtics avançats poder gaudir i opinar d'ells. Per exemple, en l'assignatura del pla antic PROP ens va tocar fer la implementació del joc "Aballone" i va ser de les pràctiques que més em va agradar, tant és així que, més endavant en un projecte posterior de l'assignatura PXC ens vam reunir els mateixos companys que havíem fet PROP i vam decidir fer una ampliació del joc Aballone.

Per tant, tenia l'idea inicial que el meu Projecte Final de Carrera estaria relacionat amb algun joc, o amb un treball que el pogués mostrar a un usuari no professional i em pogués comentar la seva opinió personal.

Probablement no m'hauria motivat programar un producte preparat només pel món professional que en mostrar-lo a algun company que no fos informàtic no l'entengués.

Per aquestes raons em va semblar interessant la proposta d'en Marc Vigo de realitzar aquesta ampliació d'una aplicació programada en Java.

L'altre punt important era el llenguatge de programació a utilitzar en el projecte. Personalment Java ha estat el llenguatge amb què més còmode m'he sentit durant la carrera i mentre escric aquestes línies em trobo treballant a jornada completa com a programador Java, per tant, la realització del projecte em servirà per tenir més experiència en aquest llenguatge.

Finalment, em cridava l'atenció representar el taulell en JOGL(Java OpenGL). Coneixia a través de la universitat el llenguatge OpenGL però no havia utilitzat mai les biblioteques de JOGL, per tant, vaig pensar que seria molt interessant



per a la meua formació conèixer l'ús que fa Java de les biblioteques OpenGL que tant bons resultats ha donat.

Segurament hi ha més raons, però aquestes potser han estat les principals que m'han portat a programar un joc en Java com a punt final de la meua carrera.

1.5 OBJECTIUS

L'objectiu d'aquest projecte és tornar a programar la interfície d'usuari del PFC original aplicant diverses funcionalitats noves.

En etapes posteriors comentarem amb més detall els canvis que s'han fet en les classes aprofitades.

L' *Interfície d'un joc de Scrabble* tindrà una interfície totalment renovada mantenint totes les funcionalitats ja existents i afegint-hi de noves. En primer lloc mostrarem , a grans trets, els objectius que s'hauran de realitzar. En l'apartat d'*Anàlisi de requeriments*, mostrarem més detalladament les funcionalitats existents i les noves funcionalitats que hi afegirem:

- El joc ha de permetre jugar partides entre jugadors simulats per la màquina i permetre diferents nivells de dificultats.
- Permetre jugar partides duplicades.
- Gestionar jugadors.
- Gestionar regles.
- Gestionar diccionaris.
- Gestionar partides.
- Gestionar idiomes.
- Gestionar estadístiques.

1.6 TECNOLOGIES UTILITZADES

Per realitzar de nou la capa de presentació s'havia de programar forçosament en el mateix llenguatge de l'aplicació original, és a dir, en Java.

L'altra decisió era decidir el mètode per programar la interfície del taulell.

En aquest punt teníem l'opció d'implementar el taulell usant biblioteques Java per interfícies 3D o en un taulell estàtic en 2D usant botons i mètodes proporcionats per Java.

L'usuari sempre prefereix veure un taulell en tres dimensions en lloc de tenir una imatge estàtica en dos dimensions, per tant, l'opció de programar el taulell en 2D es va descartar per qüestions estètiques.

Un cop decidit que s'implementarà en tres dimensions es decideix seguir utilitzant les biblioteques JOpenGL. La raó és la importància que té l'OpenGL en les interfícies 3D actuals, a més de permetre augmentar la meva experiència personal com a programador Java.

1.6.1 JOGL



Java OpenGL (JOGL) és una biblioteca que permet accedir a OPENGL mitjançant programació en Java. Actualment està sent desenvolupat pel Game Technology Group de Sun Microsystems, i és la implementació de referència per a JSR-231 (Java Bindings for OpenGL).



JOGL permet accedir a la majoria de característiques disponibles per als programadors de C , amb l'excepció de les crides a les biblioteques GLUT¹ (ja que Java conté els seus propis sistemes de finestres, AWT i Swing).

OpenGL esta basat en un llenguatge de processos (llenguatge C) i Java és un llenguatge orientat a objectes. Per facilitar l'adaptació a Java, la biblioteca JOGL simplement exposa les funcions de l'OpenGL, en lloc d'intentar realitzar un mapeig complet del codi OpenGL per a transformar-ho i adaptar-ho al paradigma d'orientació a objectes. De fet, la majoria del codi de JOGL s'ha generat automàticament a partir de les capçaleres de les biblioteques C d'OpenGL, mitjançant una eina automàtica *Gluegen*, que va ser programada específicament amb aquest propòsit.

Aquesta decisió en el disseny té els seus avantatges i els seus desavantatges. La naturalesa de processos i de màquina d'estats d' OpenGL és inconsistent amb la forma habitual de programar en Java (orientat a objectes), la qual cosa pot deixar perplexos a molts programadors. No obstant això, la conversió directa realitzada de les funcions OpenGL a mètodes Java permet la conversió del codi d'aplicacions C ja existents molt més simple. La fina capa d'abstracció proporcionada per JOGL fa que l'execució sigui molt eficient, encara que resulta molt més difícil de programar que altres biblioteques de molt més alt nivell com JAVA3D. Atès que la majoria del codi està generat automàticament, els canvis que es produeixin en OpenGL s'adapta fàcil i ràpidament a JOGL.

¹ Biblioteca d'OpenGL que proporciona diverses funcionalitats de declaració i gestió de finestres.

1.6.2 NETBEANS COM A EINA DE PROGRAMACIÓ



Un cop sabem els llenguatges que utilitzarem el següent pas és escollir l'entorn de desenvolupament per implementar el codi.

Entre les nombroses opcions que hi ha en el mercat actualment s'ha decidit desenvolupar l'aplicació utilitzant el programa *Netbeans*.

Les raons d'aquesta elecció són simples:

- És un programa gratuït.
- Té un potent editor de finestres.
- He estat programant des dels inicis de la carrera amb aquest entorn i estic familiaritzat amb la seva interfície.

La decisió de programar l'aplicació en *NetBeans* implica el inconvenient de migrar el programa original cap a aquesta entorn. Tenint en compte el grau de familiarització que tinc amb el programa, l'elecció de perdre un cert temps per migrar el codi a canvi de programar sota una interfície molt més familiar ha facilitat molt la meua elecció.

2 ANÀLISI DE REQUERIMENTS

L'anàlisi dels requeriments és el punt de partida de tot projecte i ens servirà com a guia per etapes posteriors com l'especificació, el disseny i la posterior implementació del sistema.

Aquesta etapa ens servirà per realitzar una explicació clara i concisa de les idees que es té del projecte però sense prendre cap decisió de programació.

2.1 INTRODUCCIÓ

2.1.1 ÀMBIT DEL DOCUMENT

El primer propòsit del programa *Interfície d'un joc de Scrabble* és complir amb els requisits originals de l'aplicació:

- Proporcionar la possibilitat de jugar partides d'un a quatre jugadors.
- Permetre als usuaris administrar la gestió dels perfils de jugador. Un perfil de jugador en l'aplicació representa un usuari i pot ser de tipus robot o humà. A través d'aquests perfils els jugadors participaran en les partides.
- Permetre jugar partides amb diferents idiomes possibilitant a l'usuari ampliar els diccionaris.
- Mostrar els textos de l'aplicació en diferents idiomes.
- Permetre als usuaris configurar les regles de cada partida del joc i desar les seves preferències.
- Guardar estadístiques per cada perfil d'usuari i mostrar a l'usuari rànquings ordenants per criteris que triï el jugador.



2.1.2 NOUS REQUERIMENTS

A continuació es llisten les noves millores a nivell d'usuari que la nova interfície hauria de permetre organitzades per funcionalitats:

Gestió de les regles:

- Es mantenen totes les regles que hi havia anteriorment afegint les ampliacions següents:
 - Permetre escollir la puntuació del comodí durant la partida (zero o el valor que representa).
 - Demanar el valor del comodí només al final de cada torn, en lloc de demanar-lo cada cop que es col·loca la fitxa en el taulell.
 - Si el jugador valida una jugada determinada amb el temps esgotat, es restarà de la puntuació total de la jugada un punt per cada cinc segons que sobrepassi del límit del temps configurat en el joc. Aquesta norma s'aplica tant si el temps està comptabilitzat per torn o per jugador.

Gestió de diccionaris²:

- En l'aplicació anterior cada idioma tenia un únic fitxer on es trobaven la totalitat de les paraules vàlides per jugar. L'ampliació que es realitzarà ha de permetre carregar el diccionari d'un determinat idioma a partir de diversos fitxers, d'aquesta manera l'usuari podrà organitzar llistats de paraules d'un mateix idioma per categories (temps verbals, noms propis, diminutius, etc...) i escollir les categories que desitgi per la partida següent.

² Per "diccionari" s'entén el conjunt de paraules vàlides en el què es juguen les partides de Scrabble.



- Es millora la inicialització de les paraules que acceptarà el joc, carregant el diccionari de la partida al inici de l'aplicació enlloc de fer-ho cada cop que s'inicia una partida. Tindrem el desavantatge que el cost temporal al iniciar l'aplicació serà superior, però en canvi es millorarà substancialment cada cop que comencem una partida sense canviar de diccionari, ja que el diccionari estarà en memòria i l'usuari no haurà d'esperar que es carregui .

Gestió d'estadístiques dels jugadors:

- Ampliar les estadístiques originals afegint nou atributs que s'aniran actualitzant adequadament conforme el jugador augmenti l'experiència amb el joc:
 - Número de rondes sense puntuació per partida.
 - Mitjana de punts per partida.
 - Mitjana de punts per jugada.

Gestió de partides:

- Proporcionar dos nous tipus de partida per a jugadors del tipus humà:
 - *Partides duplicades*: Entenem per partida duplicada aquella en la qual tots els jugadors participen amb el mateix faristol i el mateix taulell. En aquest mode tots els jugadors competeixen en absoluta igualtat de condicions donat que s'elimina el possible desavantatge de tenir pitjor combinació de lletres.
 - *Partides igualades*: Aquest mode enfronta un jugador humà amb un jugador simulat per la màquina. La intel·ligència d'aquest robot no realitzarà jugades superiors a la mitjana del rival. D'aquesta manera s'afegeix un punt de competitivitat i es forçarà al jugador a superar les seves mitjanes per a guanyar al rival.

Canvis durant el torn de la partida:

- *Millora de la visibilitat de les fitxes del taulell:* Si l'usuari té una fitxa seleccionada i el cursor està sobre una casella ocupada, la fitxa del jugador s'ha de situar automàticament sobre la fitxa ocupada.
- *Sistema de suggeriment:* Permetre consultar una ajuda interna en què l'aplicació mostri una possible jugada amb les fitxes que es tingui en el moment determinat en el faristol.
- *Fitxes faristol:* Permetre personalitzar l'ordre de les fitxes del faristol.
- *Visibilitat taulell:* Limitar els angles de visió del taulell de joc, ja que no te sentit poder veure el taulell per sota. Per comoditat, permetre també l'opció de situar la com estava al inici de la partida.
- *Aturar partida:* Habilitar l'opció d'aturar la partida durant una partida en joc. Al tornar a la partida es continuarà amb el temps que restava al aturar la partida.
- *Sac partida:* Permetre comprovar totes les fitxes restants en el sac durant el transcurs de la partida.
- *Log partida:* Si un usuari comprova el log durant la partida no es mostrarà el faristol dels jugadors no humans, d'aquesta manera s'evita facilitar informació del rival a un usuari humà.

2.2 DESCRIPCIÓ GENERAL

La realització del document s'ha realitzat seguint les recomanacions de la *IEE* segons l'estàndard *830-1993*.

Aquestes recomanacions descriuen tot el procés de creació de l'especificació de requeriments de *software*, i té la intenció de complir l'estàndard en el àmbit d'Enginyeria del Software. Primer comentarem els requisits generals del projecte i després ens centrarem en els requisits específics necessaris per a l'aplicació.



2.2.1 PERSPECTIVA DE L'APLICACIÓ

Interfície de l'usuari: La interfície que visualitzarà l'usuari per interactuar amb el sistema estarà basada en menús. En aquest tipus d'aplicació sempre hi haurà una finestra amb diverses opcions i l'usuari podrà escollir l'opció que vol realitzar a través d'una pantalla.

Interfície de software: Les tecnologies que s'utilitzaran en l'aplicació seran JAVA (compilació JDK1.6.0_10) i la llibreria JOGL (Java OPENGL), que s'utilitzarà per la visualització 3D del taulell del joc. El desenvolupament de l'aplicació es farà sota Windows Vista, i es provarà l'execució sota Ubuntu. Per tant, no hauria d'haver-hi problema per que funcioni tant en altres compilacions de Linux com en altres versions de Windows. Les biblioteques en JOpenGL es proveiran junt amb el projecte per a què es pugui provar sota diferents sistemes.

Interfície de hardware: El hardware recomanat per executar l'aplicació és:

- Equip personal amb processador Dual Core
- 1024 MB de memòria RAM
- L'espai necessari per l'aplicació serà aproximadament 15 MB contant els diccionaris instal·lats per defecte i sense tenir en compte les biblioteques de Java ni el codi font de l'aplicació.
- Targeta gràfica de resolució 1280 x 800 preferiblement.
- Lector CD-ROM

Interfície de comunicació: No és necessari cap tipus de protocol, ja que l'aplicació no es comunica amb cap aplicació externa.

Restriccions de memòria: Si s'assigna a la màquina virtual JAVA menys de 1024 MB de memòria es poden produir ralentitzacions, tant en el procés de càrrega diccionari com a l'hora de buscar solucions en diccionaris grans. És recomanable no rebaixar la memòria assignada a la màquina virtual Java per tal de garantir una bona execució del programa.



2.2.2 CARACTERÍSTIQUES DE L'USUARI

L'aplicació haurà de ser suficientment intuïtiva per a què un usuari amb un mínim de coneixements informàtics pugui participar al joc aprofitant totes les opcions del joc. Per aquest objectiu es realitzaran pantalles intuïtives amb icones per a què a l'usuari no li sigui complicat navegar per l'aplicació.

A més a més, tots els usuaris de l'aplicació original podran jugar en la nova ampliació. Encara que la interfície del joc ha estat renovada completament, no li serà difícil familiaritzar-se a la nova interfície.

2.2.3 REQUISITS DEL SISTEMA

L'aplicació estarà programada sota Windows Vista i provada també en Ubuntu. No obstant com que està programada sota tecnologia lliure, en principi hauria de funcionar en qualsevol sistema operatiu que tingui instal·lat Java amb la versió de JDK1.6.0_10.

3 ESPECIFICACIÓ

3.1 INTRODUCCIÓ ALS CASOS D'ÚS

3.1.1 QUÈ ÉS UN CAS D'ÚS

Un cas d'ús captura un contracte entre algú que fa servir un sistema i el propi sistema. Descriu el comportament del sistema indicant com ha de reaccionar a les accions de l'usuari. Per tant, els casos d'ús en un sistema determinat són un subconjunt dels requisits (no inclouen la interfície gràfica d'usuari, regles de negoci, formats de dades, requisits tecnològics, requisits legals, etc.).

Un cas d'ús contempla diversos escenaris possibles. Alguns acaben bé i d'altres no, però tots els escenaris formen part del cas d'ús. Hi ha un escenari principal: aquell en el qual l'entitat que ha engegat el cas d'ús aconsegueix satisfer l'objectiu pel qual l'havia engegat (hauria de ser el més "habitual"). Ex: En el cas d'ús "Crear jugador", l'escenari principal és aquell en què l'usuari aconsegueix crear un nou jugador per a poder jugar les partides.

3.1.2 A QUI AFECTA EL CAS D'ÚS

Un cas d'ús descriu un contracte entre un "stakeholder" i el sistema. Un stakeholder és qualsevol entitat amb algun interès sobre el sistema, independentment de si interactua amb ell o no. Ex: El director general d'una empresa pot estar interessat en el sistema de facturació encara que no hi interactuï directament.

Un actor, en canvi, és una entitat (una persona o un sistema informàtic) que interactua amb el sistema. L'actor principal d'un cas d'ús és l'entitat que inicia la



interacció amb el sistema, en el nostre cas l'usuari interessat en jugar una partida.

3.1.3 COM ES PODEN IDENTIFICAR

L'actor principal, quan decideix interactuar amb el sistema, ho fa amb un objectiu. Aprofitarem aquest fet per descriure els casos d'ús en termes d'actors, objectius i per descobrir els casos d'ús:

Primer identifiquem els actors principals

Després identifiquem els casos d'ús per cada actor

Com que no només els actors tenen interessos relacionats amb el sistema, podem generalitzar aquest enfocament a tots els stakeholders. Un cop identificat el cas d'ús, hem de tenir en compte que quan un actor engega un cas d'ús amb un objectiu, el sistema té la responsabilitat de satisfer-ho. Per a dur a terme aquesta responsabilitat haurà d'identificar subobjectius, recolzar-se en altres actors de suport, establir mecanismes per a recuperar-se en cas de no poder satisfer aquest objectiu, etc. Aquesta descomposició ens porta a plantejar-nos la següent qüestió:

3.1.4 FINS A QUIN PUNT DESCOMPONEM ELS CASOS D'US

Existeixen tres nivells de detall.

- Objectius generals (summary) ("sky-level")
- Objectius d'usuari (user) ("sea-level")
- Subobjectius (subfunctions) ("fish-level")

3.1.5 OBJECTIUS GENERALS

Són el nivell més alt i tenen com a propòsit mostrar el context en el què operen els objectius d'usuari. Mostren la seqüència relativa i el cicle de vida dels objectius relacionats. A més, serveixen com a "resum" del contingut dels casos d'ús que té per sota.

3.1.6 OBJECTIUS D'USUARI

Són els més importants, ja que són aquells a causa dels quals els usuaris (de manera més general, els actors principals) interactuen amb el nostre sistema.

3.1.7 SUBOBJECTIUS

Són aquelles tasques que es requereixen per a dur a terme els objectius dels usuaris, com ara cercar un producte, guardar un fitxer, etc. Normalment, només s'han de descriure en aquells casos en què la interacció sigui rellevant per algun motiu i que no hagi quedat clara en el cas d'ús d'usuari que l'inclou.

3.1.8 ERRORS HABITUALS

Finalment, un petit recull d'errors habituals en el procés d'identificació i descripció dels casos d'ús:

Excessiva dependència de la interfície d'usuari

Els casos d'ús formen part del recull de requisits, no del manual d'usuari. Per tant, el cas d'ús hauria de parlar d'intencions i objectius, no de pantalles.

No parlar sobre el sistema

Si només indiquem què fa l'actor però no indiquem què fa el sistema, no estem descrivint el comportament del sistema. Per tant, sempre hem d'intentar descriure qui fa què.

Objectius de molt baix nivell

La descomposició de casos d'ús generals en casos d'ús detallats és semblant a la descomposició funcional, però els casos d'ús són requisits, no disseny i, per tant, l'estructura dels casos d'ús no s'hauria de reflectir al codi. La descomposició funcional no va bé per als sistemes OO ja que dificulta la reutilització dels objectes.

3.2 LLISTATS CASOS D'ÚS

En aquest apartat mostrarem tots els casos d'ús de l'aplicació i, per evitar repetició innecessària d'informació descompondrem en la seva totalitat tan sols aquells no existien a l'aplicació anterior. Pels demés casos d'ús només en farem un breu resum.

Per a facilitar la comprensió, la descomposició dels casos d'ús es realitza a nivell d'usuari permetent veure la interacció que té l'usuari amb l'aplicació.

A continuació es mostren tots els casos d'ús organitzats per funcionalitats en les següents taules:

- Gestió de jugadors:
 - *Crear un jugador*
 - *Esborrar un jugador*
 - *Veure detalls del jugador*

- Gestió de diccionaris:
 - *Afegir una paraula*
 - *Esborrar una paraula*
 - *Consultar una paraula*
 - *Definir una llista de paraules*
 - *Definir una llista paraules per omisió*



- Gestió de regles:
 - *Modificar les regles*

- Gestió del rànking:
 - *Veure el rànking*

- Gestió d'idiomes:
 - *Seleccionar idioma de la partida*
 - *Seleccionar idioma de l'aplicació*
 - *Definir idioma de la partida per omissió*
 - *Definir idioma de l'aplicació per la partida*

- Gestió de partida:
 - *Jugar tipus de partida*
 - *Carregar una partida*
 - *Desar una partida*
 - *Seleccionar un jugador*
 - *Desseleccionar un jugador*
 - *Pausar la partida*
 - *Tornar a la partida*
 - *Remoure fitxes*
 - *Canviar fitxa del faristol*
 - *Canviar fitxes*
 - *Desfer una jugada*
 - *Passar el torn*
 - *Realitzar una jugada (col·locar fitxa al taulell)*
 - *Realitzar una jugada (treure fitxa del taulell)*
 - *Realitzar una jugada (moure fitxa del taulell)*
 - *Realitzar una jugada(moure fitxa del taulell)*
 - *Validar una jugada*
 - *Reciclar els comodins*



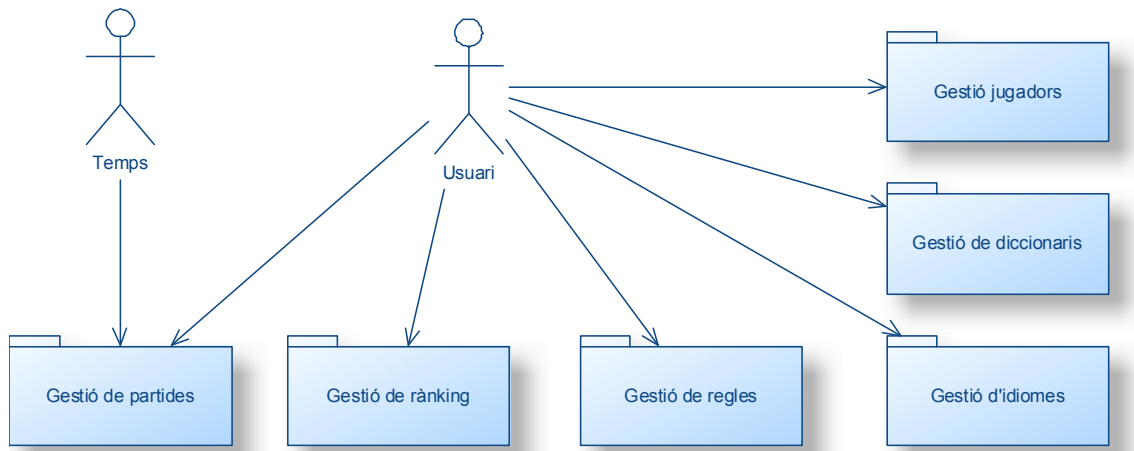
- *Fer senyors*
- *Següent torn*
- *Veure el log*
- *Desar el log*
- *Veure el sac*
- *Veure ajuda*
- *Finalitzar la partida*
- *Escolir les fitxes en una partida duplicada*
- *Mostrar l'ajuda*
- *Centrar la càmera*
- *Començar el torn duplicat*
- *Validar els comodins*

3.3 DIAGRAMA DE CASOS D'ÚS

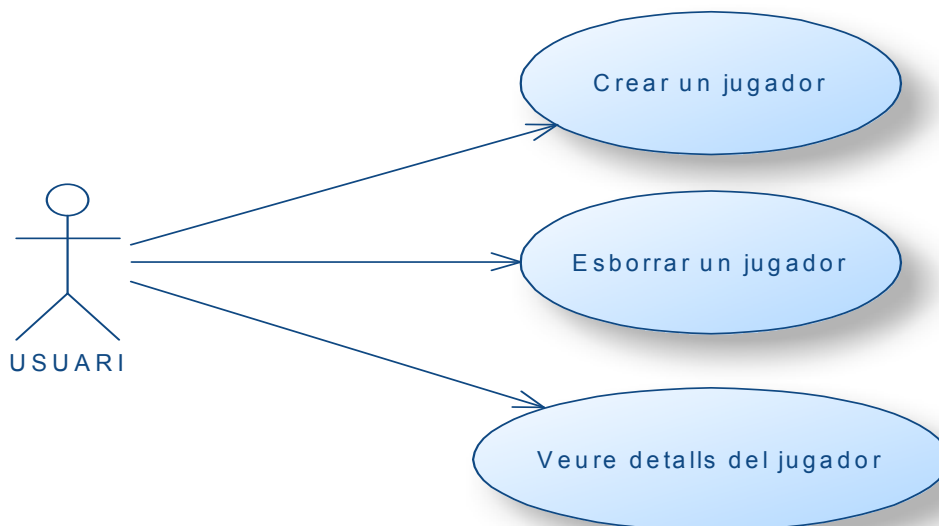
El diagrama de casos d'ús mostra els actors i les seves interaccions amb el sistema, mostra qui pot fer què i les relacions que existeixen entre les accions que es poden dur a terme. Per facilitar la comprensió del diagrama, hem empaquetat diversos casos d'ús per funcionalitats mostrant les interaccions més detalladament en el següent subapartat. En la majoria dels casos l'únic actor que hi participa serà l'usuari que està interactuant amb el sistema. En la gestió de partides aquest actor pot ser un jugador humà o un jugador simulat per la maquina, en els altres casos el actor sempre serà una persona humana.

En la gestió de partides també hi pot intervenir l'actor "Temps" que serà l'encarregat de controlar el temps que li manca al jugador actual, sempre que les regles del joc ho indiquin.

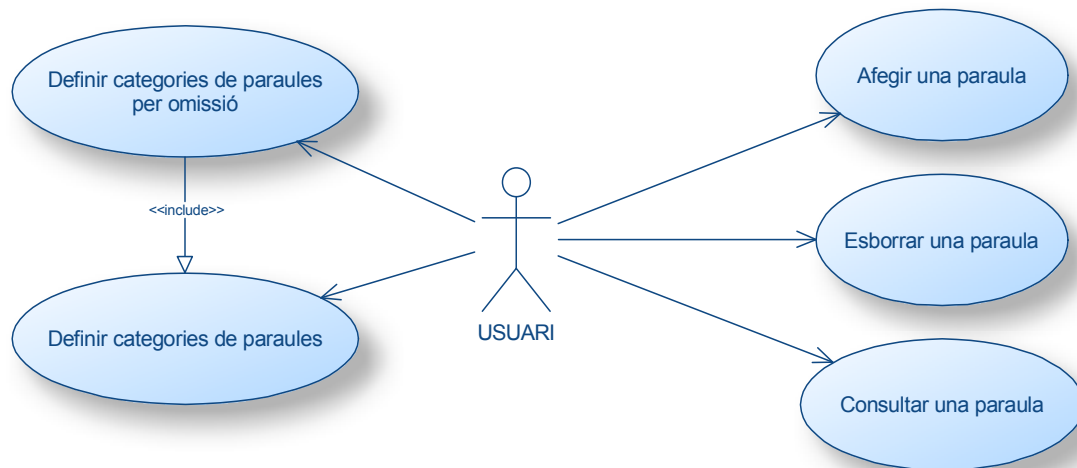
A continuació es mostra el diagrama general estructurat per funcionalitats:



3.3.1 GESTIÓ DE JUGADORS



3.3.2 GESTIÓ DE DICCIONARIS



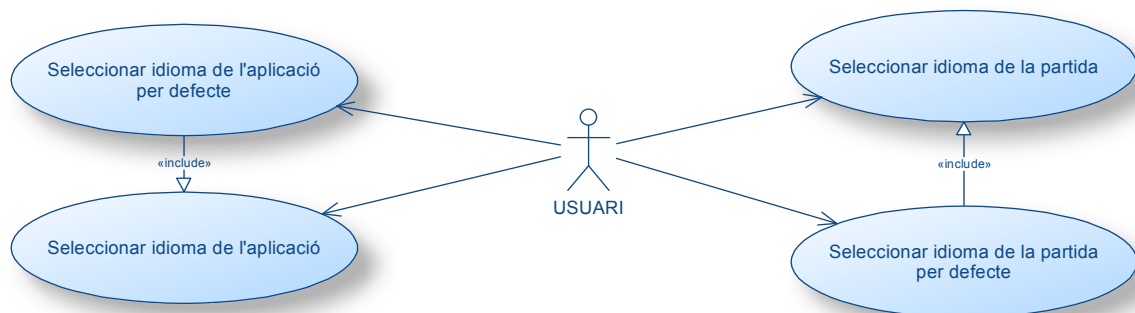
3.3.3 GESTIÓ DE REGLES



3.3.4 GESTIÓ DEL RÀNKING

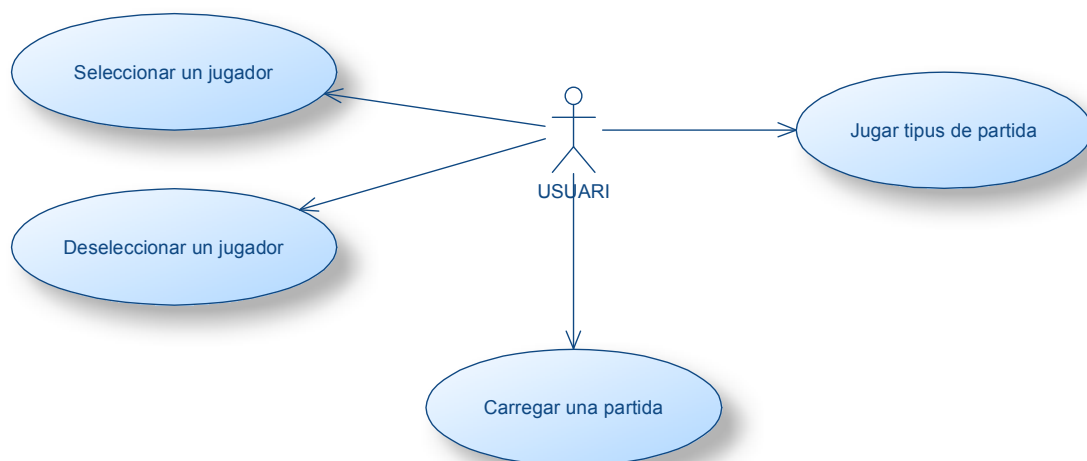


3.3.5 GESTIÓ D'IDIOMES

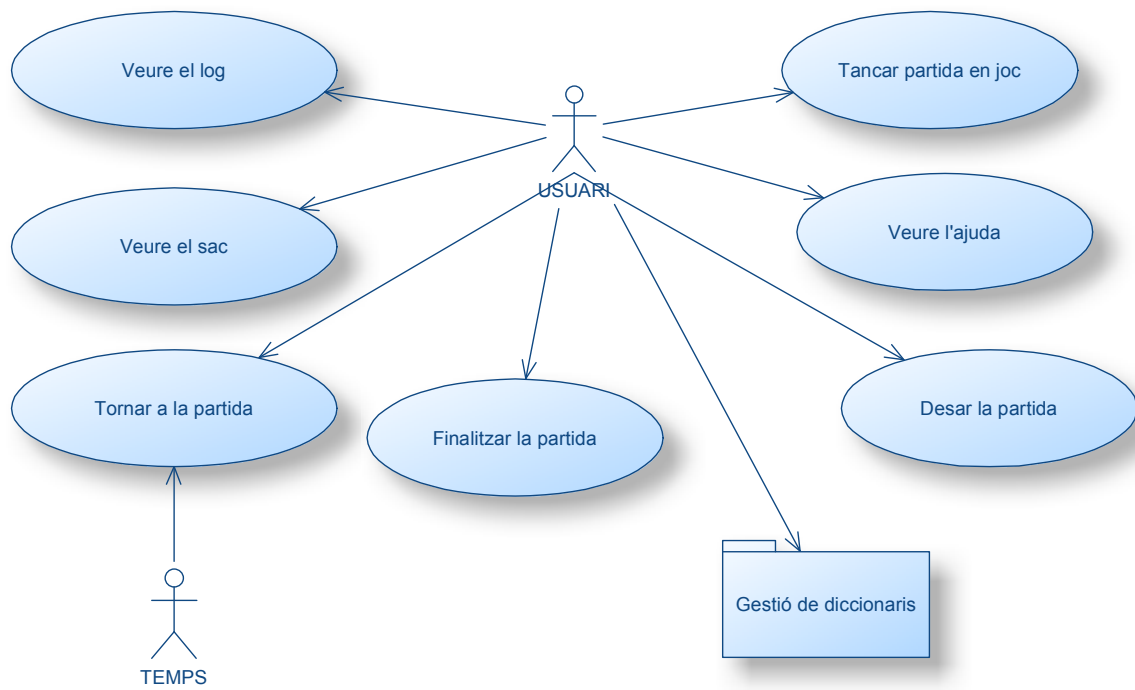


3.3.6 GESTIÓ DE PARTIDES

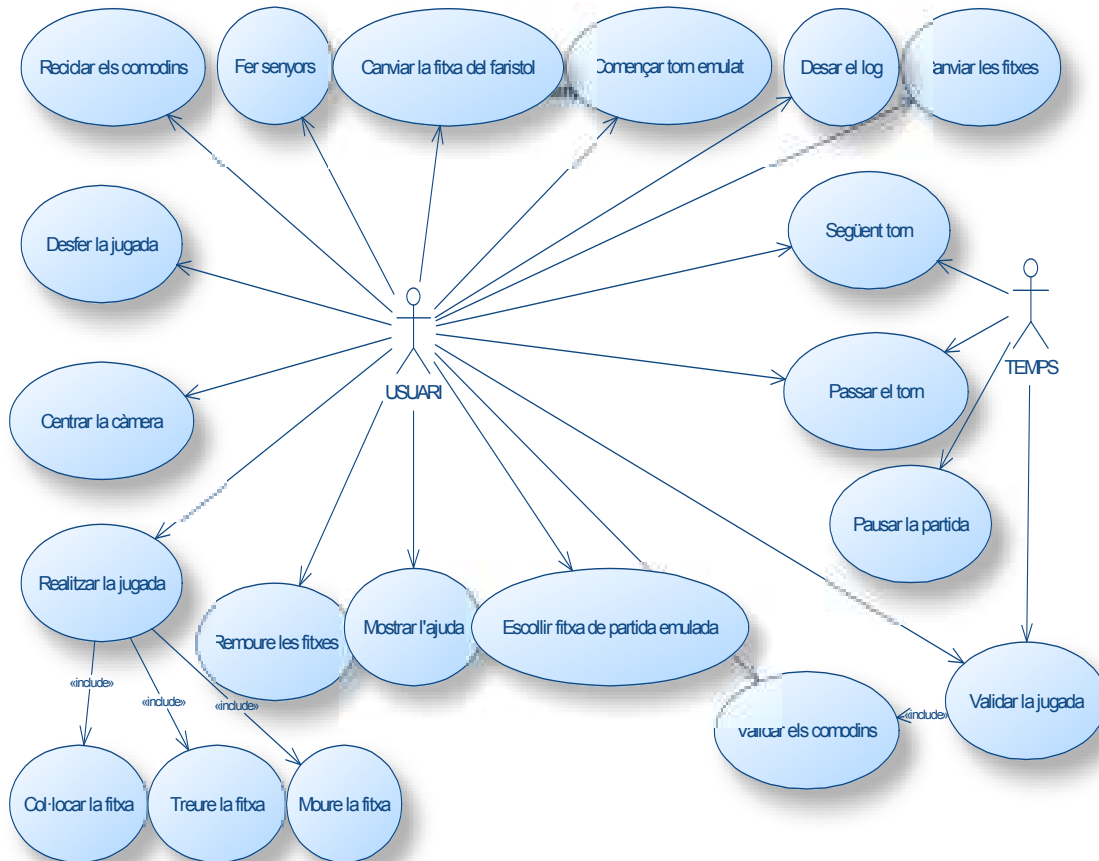
Possibles casos d'ús abans de començar la partida:



Possibles casos d'ús amb la partida pausada:



Possibles casos d'ús amb la partida en joc:



3.4 ESPECIFICACIÓ CASOS D'ÚS

Les especificacions de casos d'ús permeten observar el comportament del sistema i el intercanvi de missatges que hi ha entre l'usuari i el propi sistema. La metodologia que seguirem per a l'especificació dels casos d'ús serà d'alt nivell i expandida.

Per cada cas d'ús mostrarem els següents apartats:

- *Cas d'ús*: Nom del cas d'ús.
- *Actors*: Actor que inicia el cas d'ús actual.
- *Propòsit*: Objectiu del cas d'ús.
- *Resum*: Breu descripció de las accions que es duen a terme per a executar el cas d'ús descrit.
- *Tipus*: Indiquem si és:
 - o Primari, secundari o opcional
 - o Real o essencial
- *Curs típic del successos*: Descripció detallada del successos durant el cas d'ús, mostrant la interacció que hi ha entre el sistema i els actors participants.
- *Curs alternatiu*: Descripció dels cursos alternatius dins del curs típic del cas d'ús actual.

Per evitar una excessiva extensió del document, especificarem només els casos d'ús més destacables de la nostre aplicació.

3.4.1 ESPECIFICACIÓ: GESTIÓ DE DICCIONARIS

A continuació mostrem l'especificació dels casos d'ús relacionats amb la gestió dels diccionaris de l'aplicació.

En aquest paquet hi ha un nou cas d'ús respecte el projecte original "*definir llista de paraules*", i els anteriors casos d'ús s'han ampliat permetent a l'usuari escollir en quina categoria pertany la paraula en què vol fer l'operació (consultar, afegir o esborrar).



3.4.1.1 AFEGIR UNA PARAULA

- *Cas d'ús:* Afegir una paraula
- *Actors:* L'usuari serà l'iniciador.
- *Propòsit:* Afegir una paraula en el diccionari que hi ha actualment.
- *Resum:* La paraula introduïda per l'usuari s'afegeix en la categoria del diccionari escollit per l'usuari.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*

Accions dels actors

1. L'usuari escriu la paraula que vol afegir en el diccionari que hi ha actualment.
2. L'usuari escull en quina categoria del diccionari vol afegir la paraula.
3. L'usuari dóna l'ordre d'afegir la paraula que ha introduït.

Resposta del sistema

4. El sistema comprova que la paraula no existeix en el diccionari actual.
5. El sistema afegeix la paraula en el fitxer de paraules que l'usuari ha indicat.



- *Curs alternatiu:*
 - 4. Si l'usuari no ha introduït cap paraula o la paraula existeix en el diccionari, el sistema informarà a l'usuari i s'avorta el cas d'ús.

3.4.1.2 ESBORRAR UNA PARAULA

- *Cas d'ús:* Esborrar una paraula
- *Actors:* L'usuari serà l'iniciador.
- *Propòsit:* Esborrar una paraula vàlida del diccionari que hi ha actualment.
- *Resum:* La paraula introduïda per l'usuari s'esborrarà de la categoria de paraules que l'usuari hagi escollit.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*

Accions dels actors

Resposta del sistema

1. L'usuari escriu la paraula que vol eliminar del diccionari que hi ha actualment.

2. L'usuari escull en quina categoria del diccionari vol eliminar la paraula.

3. L'usuari dóna l'ordre d'esborrar la paraula que ha introduït.

4. El sistema comprova que la paraula existeix en el diccionari actual.



5. El sistema elimina la paraula del llistat de paraules que l'usuari ha indicat.

- *Curs alternatiu:*
 - o 4. Si l'usuari no ha introduït cap paraula o la paraula no existeix en el diccionari, el sistema informará a l'usuari.

3.4.1.3 CONSULTAR UNA PARAULA

- *Cas d'ús:* Consultar una paraula
- *Actors:* L'usuari serà l'iniciador.
- *Propòsit:* Consultar si la paraula introduïda es vàlida amb el diccionari actual
- *Resum:* El sistema informará al usuari si la paraula que s'està consultant es vàlida per a jugar amb la configuració actual, i indicarà en quin llistat es troba disponible.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*

Accions dels actors

Resposta del sistema

1. L'usuari escriu la paraula que vol consultar del diccionari que hi ha actualment.

2. L'usuari dona l'ordre de consultar la paraula que ha introduït.

3. El sistema comprova que la paraula existeix en el diccionari actual.

4. El sistema indica a l'usuari si existeix la paraula i en quin llistat es troba disponible.

- *Curs alternatiu:*
 - o 3. Si l'usuari no ha introduït cap paraula o la paraula no existeix en el diccionari, el sistema informarà a l'usuari.

3.4.1.4 DEFINIR CATEGORIES DE PARAULES

- *Cas d'ús:* Definir categories de paraules
- *Actors:* L'usuari serà l'iniciador.
- *Propòsit:* Definir les categories a carregar per la partida.
- *Resum:* L'usuari escollirà quins fitxers de paraules es volen afegir en el diccionari on es jugarà la partida, finalment podrà definir la selecció actual per omisió.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*

Accions dels actors

Resposta del sistema

2. L'usuari escollirà les categories que vol carregar.

3. L'usuari dona l'ordre de carregar la selecció,

1. El sistema carrega totes els categories possibles pel diccionari actual.



opcionalment pot establir la seva elecció per omissió.

4. El sistema carrega l'elecció de l'usuari.

- *Curs alternatiu:*
 - o No existeixen cursos alternatius.

3.4.2 ESPECIFICACIÓ: GESTIÓ DE REGLES

El cas d'ús inclòs en la gestió de regles (*Modificar Regles*) es comporta igual que en l'aplicació anterior. Comentar que l'usuari selecciona d'un llistat quines regles vol aplicar i com vol configurar-les, addicionalment, es permet marcar aquest conjunt de regles per omissió.

Un cop l'usuari tingui la seva configuració seleccionada dóna l'ordre al sistema per carregar la configuració .

3.4.3 ESPECIFICACIÓ: GESTIÓ DE PARTIDES

A continuació mostrem els nous casos d'ús de l'aplicació relacionats amb la gestió de partides.

De l'aplicació original, podem destacar el cas d'ús "*Desar una partida*" i "*Carregar una partida*".

En el primer cas, durant la partida l'usuari escull l'opció de desar la partida tot indicant al sistema la ruta on guardar la configuració. Un cop confirmada, el sistema desa totes les dades necessàries per permetre continuar la partida en altre moment.

El cas d'ús "*carregar una partida*" l'usuari escollirà de la pantalla inicial l'opció de "carregar partida" i indicarà al sistema la ruta on es troba el fitxer a carregar. Si el fitxer és vàlid, el sistema carregarà la partida i deixarà la partida en joc, si el fitxer no és vàlid el sistema informarà a l'usuari.

3.4.3.1 JUGAR TIPUS DE PARTIDA

- *Cas d'ús:* Jugar tipus de partida
- *Actors:* L'usuari serà l'iniciador.
- *Propòsit:* Començar una partida amb un determinat nombre de jugadors, i un determinat tipus de partida (normal, igualada o duplicada)
- *Resum:* L'usuari seleccionarà quin tipus de partida vol jugar un cop tingui els jugadors participants, i a continuació començarà la partida.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*

Accions dels actors

Resposta del sistema

1. L'usuari selecciona el tipus de partida que vol començar a jugar.

2. L'usuari dona l'ordre de començar la partida.

3. El sistema carregarà tota la configuració actual i la partida començarà amb el mode que ha indicat l'usuari.

- *Curs alternatiu:*
 - 3. Si l'usuari dona l'ordre de començar la partida sense cap jugador seleccionat el sistema informarà que s'ha de seleccionar un jugador com a mínim.



3.4.3.2 SELECCIONAR UN JUGADOR

- *Cas d'ús:* Seleccionar un jugador
- *Actors:* L'usuari serà l'iniciador.
- *Propòsit:* Seleccionar un jugador per la partida.
- *Resum:* El jugador indicat és candidat a participar en la partida.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*

Accions dels actors

Resposta del sistema

2. L'usuari selecciona un jugador i l'indica com a participant per la partida.

1. El sistema carrega tots els jugadors disponibles.

3. El sistema registra el jugador com a participant de la futura partida.

- *Curs alternatiu:*
 - No existeixen cursos alternatius.



3.4.3.3 DESSELECCIONAR UN JUGADOR

- *Cas d'ús:* Deseleccionar un jugador
- *Actors:* L'usuari serà l'iniciador.
- *Propòsit:* Deseleccionar un jugador per la partida.
- *Resum:* El jugador indicat no participarà en la següent partida.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*

Accions dels actors

Resposta del sistema

1. L'usuari selecciona un jugador participant per la següent partida i dona l'ordre.

2. El sistema elimina el jugador com a participant de la pròxima partida.

- *Curs alternatiu:*
 - No existeixen cursos alternatius.



3.4.3.4 COMENÇAR UN TORN DUPLICAT

- *Cas d'ús:* Començar un torn duplicat.
- *Actors:* L'usuari serà l'iniciador.
- *Propòsit:* Avançar al inici del torn en una partida duplicada.
- *Resum:* El jugador selecciona les fitxes que vol utilitzar pel següent torn i avança al inici del torn.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*

Accions dels actors

Resposta del sistema

1. L'usuari selecciona les fitxes que desitgi pel pròxim torn.

2. L'usuari dona l'ordre d'avançar.

3. El sistema avança a la partida amb les fitxes seleccionades per l'usuari.

- *Curs alternatiu:*
 - No existeixen cursos alternatius.



3.4.3.5 CANVIAR UNA FITXA DEL FARISTOL

- *Cas d'ús:* Canviar fitxa del faristol
- *Actors:* L'usuari serà l'iniciador.
- *Propòsit:* Intercanviar dos fitxes.
- *Resum:* El jugador canvia de lloc dos fitxes del seu faristol.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*

Accions dels actors

Resposta del sistema

1. L'usuari selecciona una fitxa del seu faristol.

2. L'usuari selecciona una altra fitxa del faristol donant l'ordre d'intercanviar.

3. El sistema intercanvia l'ordre de les fitxes.

- *Curs alternatiu:*
 - 2. Si l'usuari no vol intercanviar la fitxa, pot seleccionar una altra fitxa del faristol sense donar l'ordre d'intercanvi avortant el cas d'ús descrit.



3.4.3.6 MOSTRAR L'AJUDA

- *Cas d'ús:* Mostrar l'ajuda
- *Actors:* L'usuari serà l'iniciador.
- *Propòsit:* Mostrar una possible jugada a un jugador humà.
- *Resum:* El sistema li mostrarà una jugada al jugador amb les fitxes actuals del faristol.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*

Accions dels actors

Resposta del sistema

1. L'usuari dona l'ordre al sistema.

2. El sistema mostra una possible jugada al taulell.

- *Curs alternatiu:*
 - 2. Si el sistema informa que no ha trobat cap jugada.



3.4.3.7 SEGÜENT TORN

- *Cas d'ús:* Següent torn
- *Actors:* L'usuari serà l'iniciador.
- *Propòsit:* El sistema avança al pròxim torn.
- *Resum:* El sistema dóna per finalitzat el torn i avança al següent.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*

Accions dels actors

Resposta del sistema

1. L'usuari dóna l'ordre al sistema.

2. El sistema dóna per finalitzada la jugada, actualitza totes les dades i avança al proper torn.

- *Curs alternatiu:*
 - 2. En el cas que finalitzi la partida, veurà totes les puntuacions pertinents, i mostrarà el guanyador de la partida permetent desar el log de la partida completa.



3.4.3.8 VALIDAR UNA JUGADA

- *Cas d'ús:* Validar una jugada
- *Actors:* L'usuari serà l'iniciador.
- *Propòsit:* Validar la jugada realitzada.
- *Resum:* Es validarà la jugada mostrant al jugador les paraules realitzades amb les seves respectives puntuacions.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*

Accions dels actors

Resposta del sistema

1. L'usuari dona l'ordre al sistema.

2. El sistema comprova que no hi hagi cap comodí recent posat.
2. bis. *Include:* Validar comodins.

3. El sistema comprova que la col·locació de les fitxes sigui vàlida.

4. El sistema comprova la validesa de les paraules.

5. El sistema finalitza el torn i mostra les puntuacions.

- *Curs alternatiu:*
 - 2. Si hi ha un comodí recent jugat s'efectua el cas 2"bis" abans d'executar el punt 3.
 - 3. Si la col·locació de les fitxes no és correcte, el sistema informa que la jugada no és vàlida i avorta el cas d'ús.



- 4. Si les paraules no són vàlides en el diccionari actual, s'avorta el cas d'ús informant a l'usuari de les paraules invàlides.

3.4.3.9 VALIDAR ELS COMODINS

- *Cas d'ús:* Validar els comodins
- *Actors:* -
- *Propòsit:* Mostrar quina lletra representa el comodí.
- *Resum:* El jugador seleccionarà d'un llistat quina/es fitxa/es representa el/s comodí/ins.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*

Accions dels actors

2. L'usuari selecciona la lletra representada pel comodí.

Resposta del sistema

1. El sistema mostra un llistat de lletres per que l'usuari indiqui quina fitxa representa el comodí marcat.

- El sistema comprova que no hi hagi un altre comodí sense lletra.

- *Curs alternatiu:*
 - 3. Si queda un altre comodí per validar, es repeteix el pas 1.



3.4.3.10 CENTRAR LA CÀMERA

- *Cas d'ús:* Centrar la càmera
- *Actors:* -
- *Propòsit:* Deixar la càmera del taulell en la posició inicial.
- *Resum:* Els angles de visió de la càmera es reinicien per tenir una visió completa del taulell de joc.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*

Accions dels actors

Resposta del sistema

1. L'usuari dona l'ordre de reiniciar la càmera.

2. El sistema mostra el taulell complet perpendicular a la càmera.

- *Curs alternatiu:*
 - No existeixen cursos alternatius.



3.4.3.11 PAUSAR LA PARTIDA

- *Cas d'ús:* Pausar Partida
- *Actors:* L'usuari serà l'iniciador.
- *Propòsit:* Aturar la partida.
- *Resum:* La partida s'atura, el taulell deixa de ser visible i aturem el temps si s'està utilitzant en les regles del joc actuals.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*

Accions dels actors

Resposta del sistema

1. L'usuari dona l'ordre d'aturar el joc.

2. El sistema emmagatzema el temps restant que li queda al jugador, i mostra les opcions de joc en partida pausada.

- *Curs alternatiu:*
 - No existeixen cursos alternatius.

3.4.3.12 TORNAR A LA PARTIDA

- *Cas d'ús:* Tornar a la partida
- *Actors:* L'usuari serà l'iniciador.
- *Propòsit:* Retornar a la partida.
- *Resum:* La partida torna al moment en que s'havia pausat, carregant el temps que hi havia anteriorment (si les regles actuals ho permeten).
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*
-

Accions dels actors

1. L'usuari dona l'ordre de tornar a la partida.

Resposta del sistema

2. El sistema recupera la configuració que hi havia anteriorment, i la partida continua.

- *Curs alternatiu:*
 - No existeixen cursos alternatius.



3.4.3.13 VEURE EL SAC

- *Cas d'ús:* Veure el sac
- *Actors:* L'usuari serà l'iniciador.
- *Propòsit:* Veure les fitxes restants del sac.
- *Resum:* Es mostrarà el llistat de fitxes que resten en el sac.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*
-

Accions dels actors

Resposta del sistema

1. L'usuari dona l'ordre de veure les fitxes que queden en el sac.

2. El sistema mostra el llistat de fitxes que hi ha en el moment determinat al sac.

- *Curs alternatiu:*
 - No existeixen cursos alternatius.



3.4.3.14 FINALITZAR LA PARTIDA

- *Cas d'ús:* Finalitzar la partida
- *Actors:* L'usuari serà l'iniciador.
- *Propòsit:* Acabar la partida que hi ha en joc.
- *Resum:* El sistema acaba la partida iniciada i mostra la pantalla inicial.
- *Tipus:*
 - Primari
 - Essencial
- *Curs típic del successos:*

Accions dels actors

Resposta del sistema

1. L'usuari dona l'ordre de finalitzar la partida.

2. El sistema esborra tota la informació de la partida i mostra la pantalla inicial.

- *Curs alternatiu:*
 - 1. Si l'usuari no confirma l'ordre, el joc continua en mode pausat.

3.5 MODEL DE COMPORAMENT

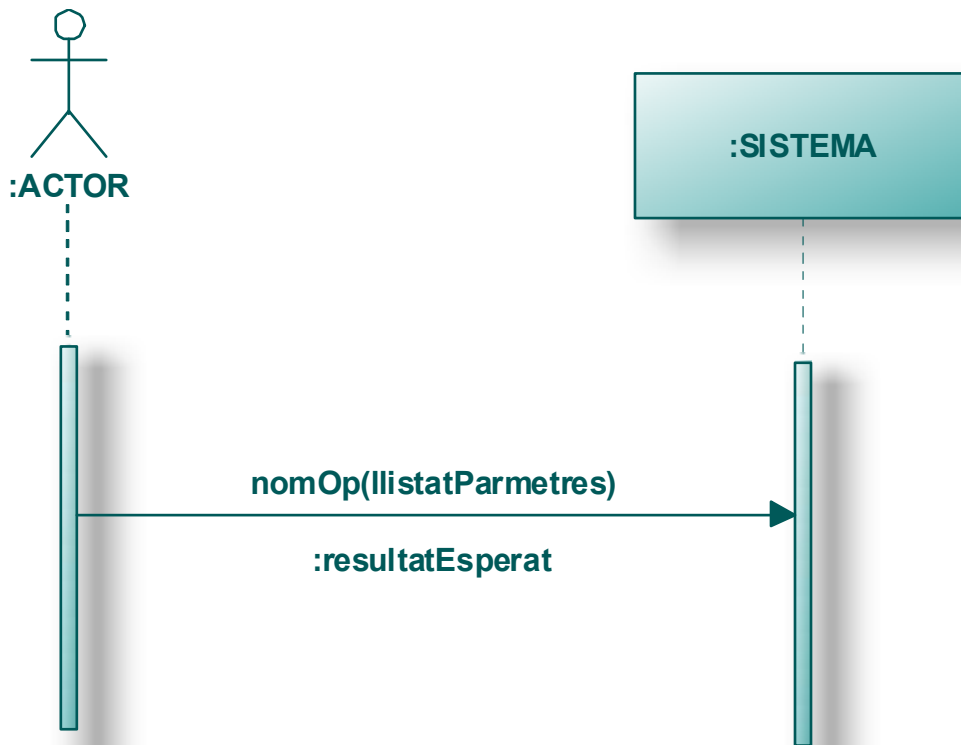
En aquest apartat es recullen els diagrames de seqüència del sistema. Un diagrama de seqüència mostra les interaccions entre objectes ordenats en seqüència temporal. En els diagrames es representen els objectes que es troben a l'escenari i la seqüència de missatges intercanviats entre els objectes per dur a terme la funcionalitat descrita per l'escenari. L'existència d'aquests diagrames permeten l'execució dels casos d'ús descrits anteriorment i reflecteixen el comportament del sistema quan l'usuari interactua amb cadascun dels casos d'ús.

Per evitar la repetició de casos d'ús només mostrarem els contractes que en el projecte anterior no existien.

3.5.1 DIAGRAMES DE SEQÜÈNCIA

Els diagrames de seqüència UML permeten modelar la interacció entre diversos participants que intercanvien missatges per a dur a terme una certa tasca. Un dels usos més habituals d'aquests diagrames és mostrar un escenari de funcionament d'un programa orientat a objectes; en aquest cas, es mostra com un conjunt d'instàncies de programari (objectes del llenguatge de programació triat) interactuant quan una d'elles rep una invocació d'una operació.

A continuació mostrem l'esquema amb el que mostrarem els diagrames:



Primer mostrarem l'actor que executarà les crides al sistema, i a la dreta el sistema, que executarà les operacions. Entre els dos mostrarem les relacions que es duen a terme per executar una determinada operació a través de funcions, cada funció podrà tenir una sèrie de paràmetres d'entrada, i al finalitzar l'operació poden retornar un resultat determinat.

3.5.1.1 DIAGRAMES: GESTIÓ DE JUGADORS, REGLES, RÀNKING, IDIOMES

Els diagrames de seqüència de les operacions de gestió de jugadors, de regles, de rànkings i d'idiomes són idèntics que en l'aplicació original.

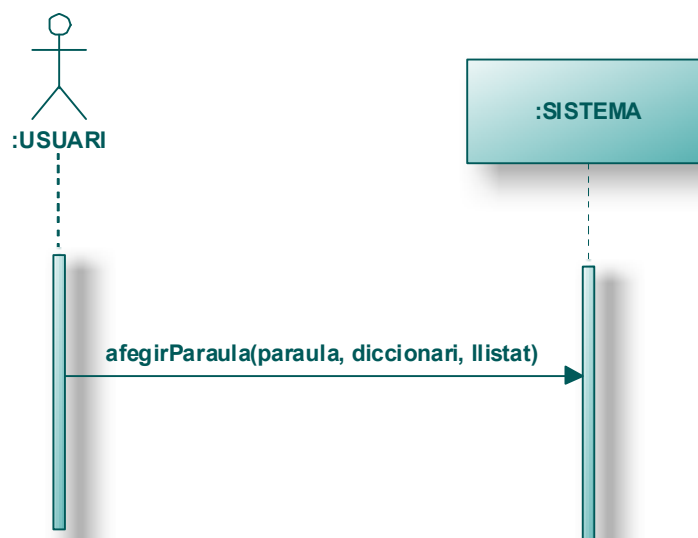
Els diagrames de seqüència de la gestió dels jugadors l'usuari sempre serà l'actor iniciador, enviant-hi com a paràmetre el nom del jugador que vol inserir, eliminar o consultar i el sistema li retornarà la informació proporcionada.

En el cas dels rànkings l'usuari demanarà a l'usuari la classificació passant com a paràmetre quin criteri d'ordenació vol, i el sistema li mostrarà la informació.

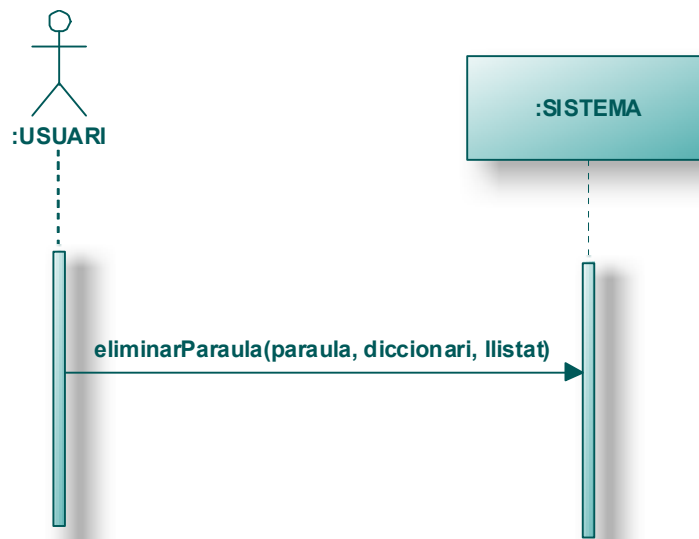
El cas dels idiomes i regles, el sistema li mostrarà el llistat de totes les opcions que pot modificar, i l'usuari un cop tingui la configuració al seu gust li enviarà l'ordre al sistema per a què les apliqui.

3.5.1.2 DIAGRAMES: GESTIÓ DE DICCIONARIS

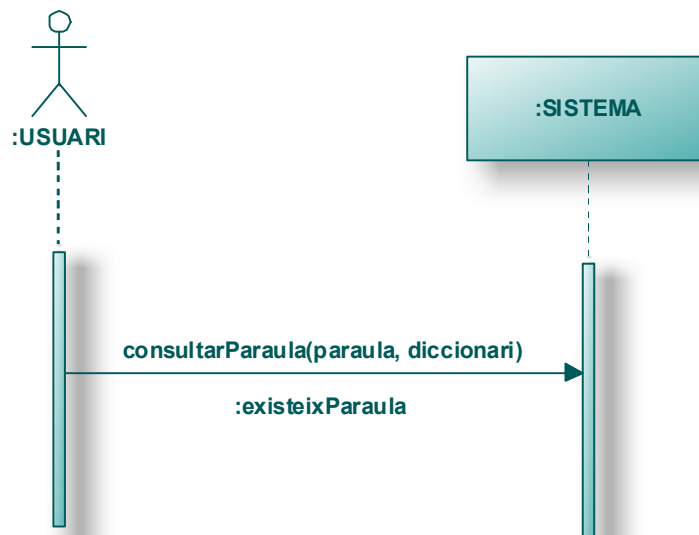
En aquesta apartat mostrarem els diagrames de seqüència corresponents a la gestió dels diccionaris.



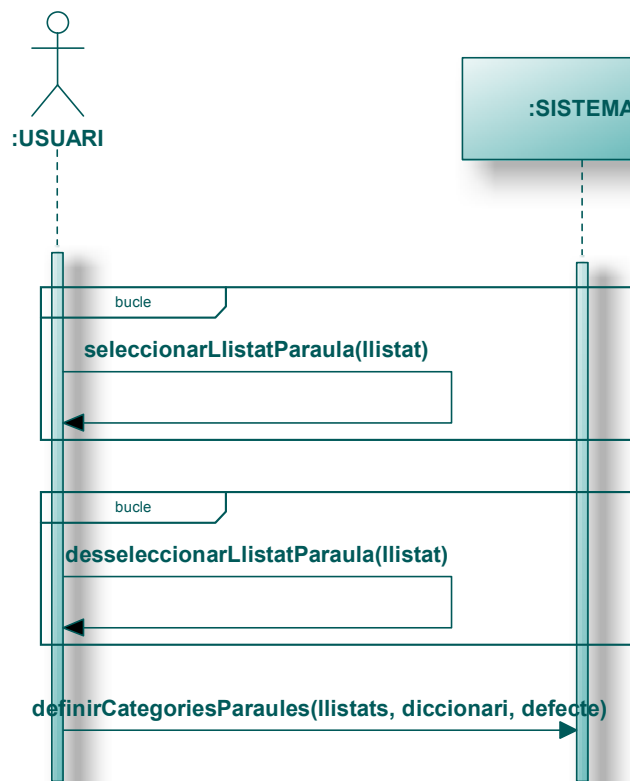
Afegir una paraula



Eliminar una paraula



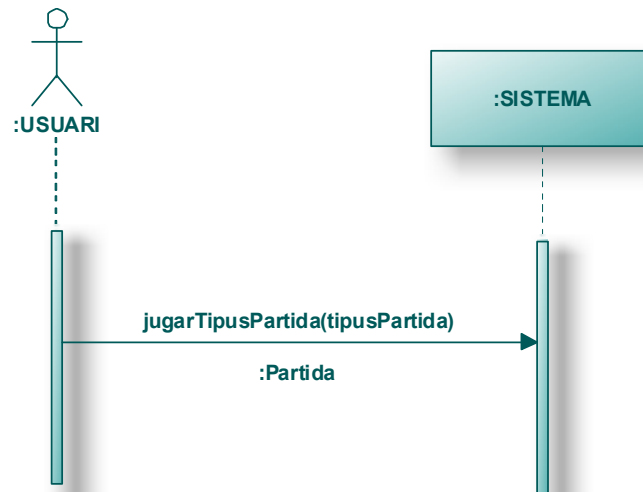
Consultar una paraula



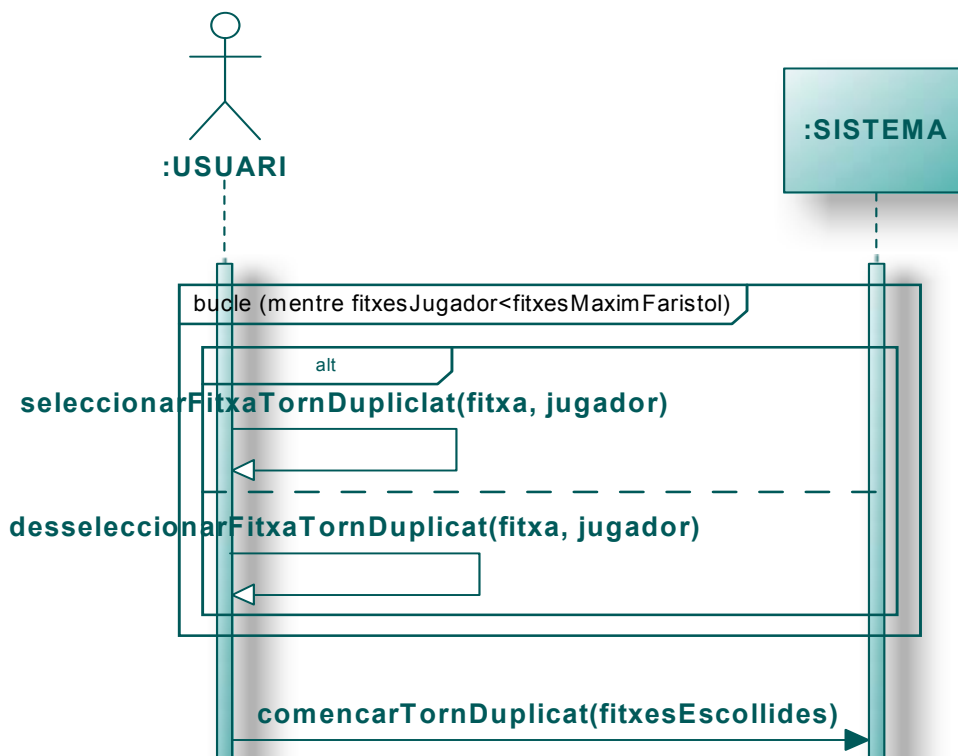
Definir categories de paraules

3.5.1.3 DIAGRAMES: GESTIÓ DE PARTIDES

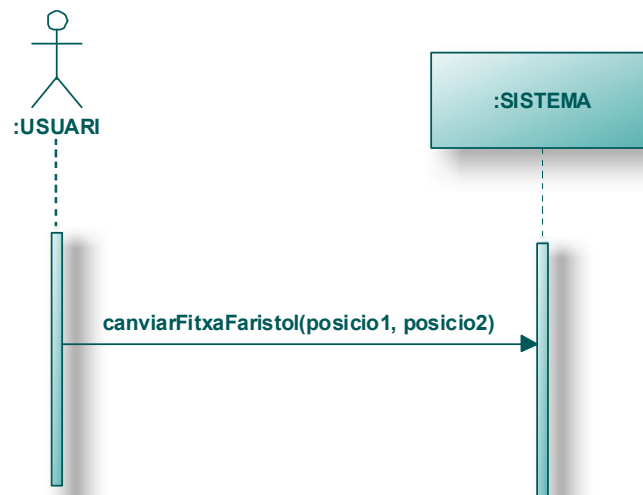
En aquest apartat mostrarem les operacions noves i modificades que es poden dur a terme durant la partida.



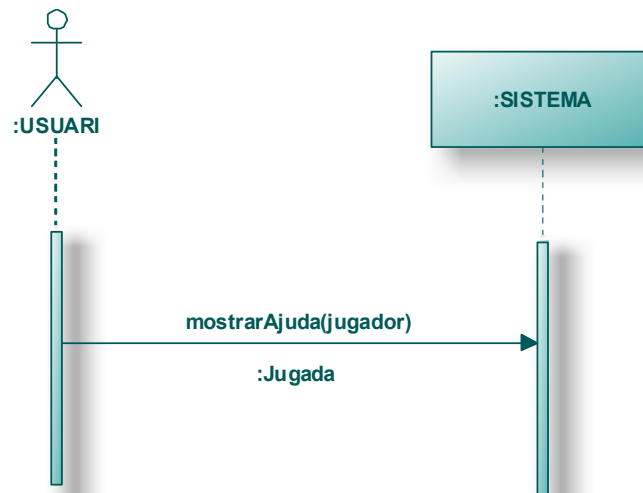
Jugar tipus de partida



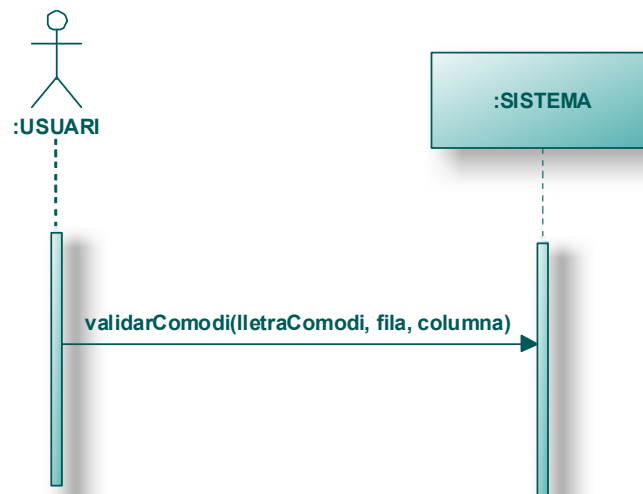
Escollir les fitxes en una partida duplicada



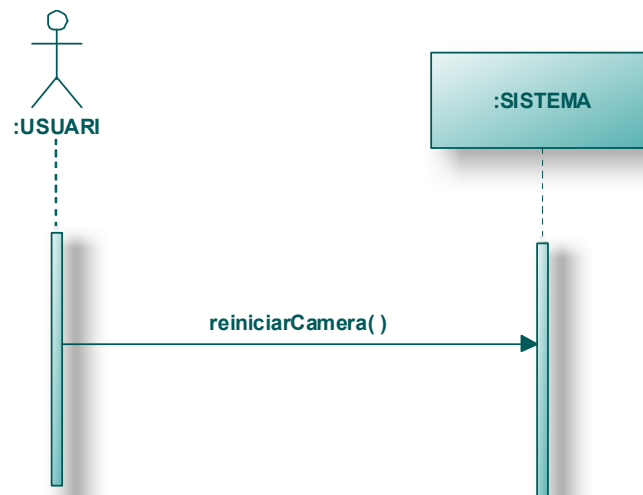
Canviar fitxa del faristol



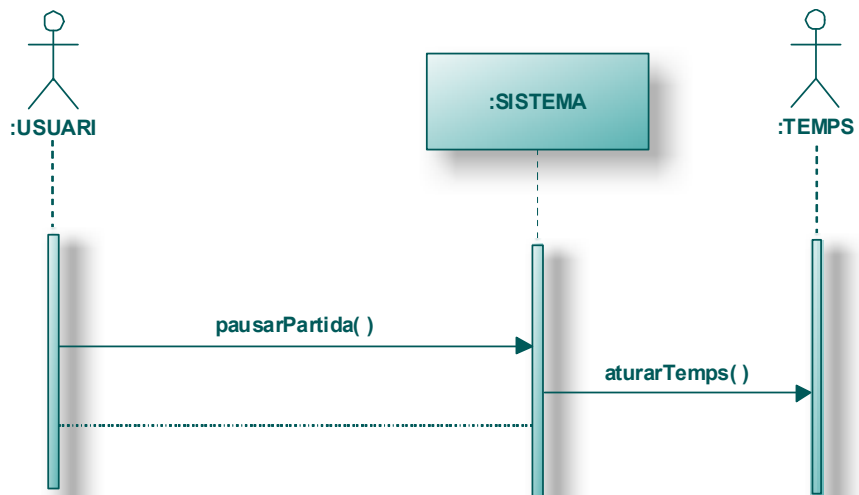
Mostrar l'ajuda



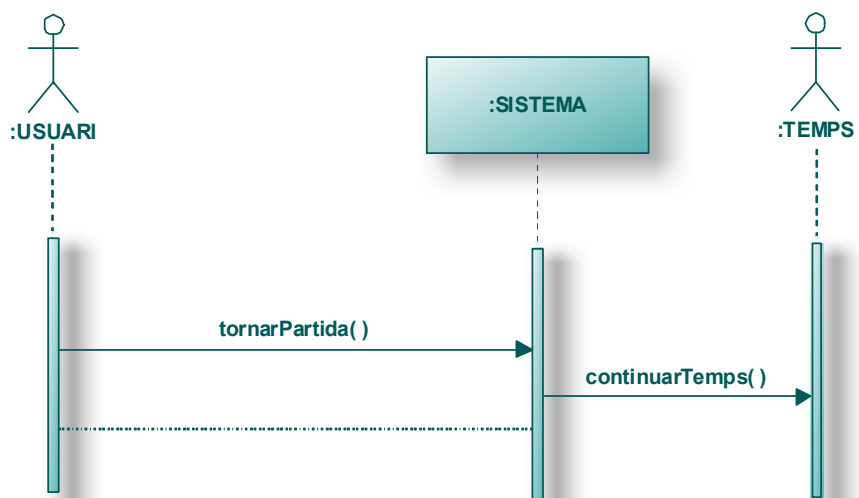
Validar el comodi



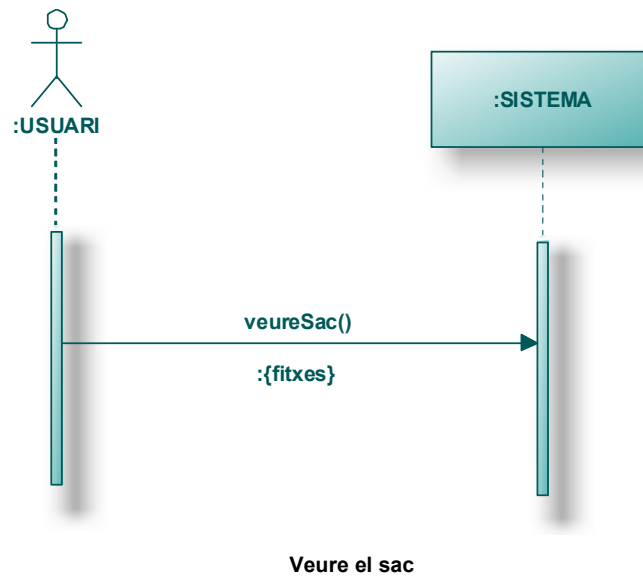
Reiniciar la càmera



Pausar la partida



Tornar a la partida



3.5.2 CONTRACTES DE LES OPERACIONS

En aquest apartat descriurem el comportament que té el sistema per a les noves operacions que hem definit en els diagrames de seqüència.

Per cada operació mostrem les condicions que s'han de complir abans d'executar l'operació, l'estat en que queda el sistema un cop executada l'operació, els paràmetres d'entrada per l'operació, i finalment el que retorna l'operació (en cas que retorni quelcom) .

Per tant, per a cada operació mostrarem:

Operació (*paràmetre1*, *paràmetre2*, ... , *paràmetreN*)

Nom: nomOperació (*paràmetre1*, *paràmetre2*, ... , *paràmetreN*)

Responsabilitats: Objectiu de l'operació.

Cas d'ús: Crear jugador

Excepcions: Excepcions que s'executaran en cas de que no es compleixi el curs normal de l'operació.

Precondicions: Condicions que s'han de complir abans d'executar l'operació.

Postcondicions: Condicions que es compleixen un cop executada l'operació.

Sortida: Paràmetres que retorna l'operació.



Els contractes que mostrarem a continuació seran els que han canviat o hem hagut de crear respecte l'aplicació original.

3.5.2.1 CONTRACTES: GESTIÓ DE JUGADORS

- `veureDetallsJugador(jugador)`

Nom: `veureDetallsJugador(jugador:String)`

Responsabilitats: Mostra a l'usuari totes les estadístiques del jugador amb nom "*jugador*".

Cas d'ús: Veure detalls del jugador.

Excepcions: -

Precondicions: El jugador existeix.

Postcondicions: -

Sortida: Retorna un panell amb la informació completa de l'usuari.

3.5.2.2 CONTRACTES: GESTIÓ DE DICCIONARIS

- `afegirParaula(paraula, diccionari, llistat)`

Nom: `afegirParaula(paraula:String, diccionari:String, llistat:String)`

Responsabilitats: Afegeix la paraula com una paraula vàlida en el diccionari actual.

Cas d'ús: Afegir una paraula

Excepcions: -

Precondicions: El diccionari existeix, i el paràmetre *paraula* no està en el diccionari actual.

Postcondicions: *paraula* és una paraula vàlida pel diccionari *diccionari* i s'ha inclòs al llistat indicat pel paràmetre *llistat*.

Sortida: -

- `eliminarParaula(paraula, diccionari, llistat)`

Nom: `eliminarParaula(paraula:String, diccionari:String, llistat:String)`

Responsabilitats: Elimina la paraula del diccionari actual.



Cas d'ús: Esborrar una paraula

Excepcions: -

Precondicions: El diccionari existeix, i el paràmetre *paraula* existeix en el diccionari actual, en el llistat indicat pel paràmetre *llista*.

Postcondicions: *paraula* ja no és una paraula vàlida pel diccionari *diccionari*.

Sortida: -

- `consultarParaula(Paraula, diccionari) : existeixParaula, llistat`

Nom: `consultarParaula (paraula:String, diccionari:String) : existeixParaula:bool, llistat:String`

Responsabilitats: Informa a l'usuari si la paraula *paraula* existeix.

Cas d'ús: Consultar una Paraula.

Excepcions: -

Precondicions: El diccionari entrat per paràmetre *diccionari*, existeix.

Postcondicions: -

Sortida: Retorna un booleà informant si la paraula existeix, i el llistat on pertany la paraula (si la paraula existeix).

- `seleccionarLlistatParaula(llistat)`

Nom: `seleccionarLlistatParaula (llistat:String)`

Responsabilitats: L'usuari selecciona un fitxer vàlid de paraules que formaran part del diccionari actual.

Cas d'ús: Definir categories de paraules

Excepcions: -

Precondicions: El fitxer indicat és vàlid i no està seleccionat.

Postcondicions: El llistat de les paraules queda seleccionat per a quan es defineixi en el diccionari actual.

Sortida: -

- `desseleccionarLlistatParaula(llistat)`

Nom: `desseleccionarLlistatParaula(llistat:String)`



Responsabilitats: El usuari desselecciona un fitxer vàlid de paraules que pot formar part del diccionari actual.

Cas d'ús: Definir categories de paraules.

Excepcions: -

Precondicions: El fitxer de paraules és vàlid i està seleccionat.

Postcondicions: El llistat de les paraules entrat com a paràmetre no es carregarà quan es defineixi en el diccionari actual.

Sortida: -

- `definirCategoriaParaules(llistats, diccionari, defecte)`

Nom: `definirCategoriaParaules` (*llistats*: String [], *diccionari*: String, *defecte*: bool)

Responsabilitats: El sistema carrega el diccionari amb totes les paraules que hi ha en els fitxers indicats en el vector de paraules *llistats*, i permet marcar aquest llistat per defecte cada cop que es carregui el diccionari actual si el paràmetre *defecte* així ho indica.

Cas d'ús: Definir categories de paraules

Excepcions: -

Precondicions: Els vector *llistats* conté noms de fitxers vàlids en la carpeta dels diccionaris.

Postcondicions: El conjunt de paraules que contenen els fitxers indicats en el paràmetre *llistats* seran els que formaran el diccionari actual. Si addicionalment s'ha indicat que es defineixi per omisió, cada cop que es carregui el idioma actual es carregarà el llistat de paraules actual.

Sortida: -



3.5.2.3 CONTRACTES: GESTIÓ DE PARTIDES

- `jugarTipusPartida(tipusPartida, jugadors)` : Partida

Nom: `jugarTipusPartida` (*tipusPartida*:String) : *Partida*: Panell

Responsabilitats: Començar la partida amb el mode que l'usuari esculli.

Cas d'ús: Jugar un tipus de partida.

Excepcions: -

Precondicions: Estan seleccionats els jugadors que participaran a la partida i el tipus de partida és vàlid.

Postcondicions: La partida està preparada per començar amb el tipus escollit per l'usuari.

Sortida: Retornarà la partida configurada per començar a jugar amb el mode escollit per l'usuari i els jugadors que prèviament s'ha elegit.

- `seleccionarFitxaTornDuplicat` (*fitxa*, jugador)

Nom: `seleccionarFitxaTornDuplicat` (*fitxa*:String,*jugador*:String)

Responsabilitats: El jugador *jugador* se li afegeix temporalment la fitxa *fitxa* en el faristol pel proper torn.

Cas d'ús: Començar un torn en mode de partida duplicada.

Excepcions: -

Precondicions: El jugador d'una partida duplicada que li toca el pròxim torn no té el faristol complet.

Postcondicions: Una de les fitxes del faristol del jugador pel pròxim torn és la definida pel paràmetre d'entrada *fitxa*.

Sortida: -

- `desseleccionarFitxaTornDuplicat` (*fitxa*, jugador)

Nom: `desseleccionarFitxaTornDuplicat` (*fitxa*:String, *jugador*:String)

Responsabilitats: Eliminar la fitxa entrada com a paràmetre al jugador que té el torn actualment.

Cas d'ús: Començar un torn duplicat.

Excepcions: -



Precondicions: El jugador entrat com a paràmetre té la fitxa *fitxa* seleccionada en el faristol pel torn següent.

Postcondicions: S'elimina la *fitxa* que tenia el *jugador* en el faristol.

Sortida: -

- `començarTornDuplicat (fitxesEscollides, jugador)`

Nom: `començarTornDuplicat (fitxesEscollides:String[], jugador:String)`

Responsabilitats: El jugador entrat com a paràmetre comença el torn amb les fitxes escollides.

Cas d'ús: Començar un torn duplicat.

Excepcions: -

Precondicions: L'usuari ha seleccionat totes les fitxes que vol pel pròxim torn.

Postcondicions: Comença el torn de la partida amb les fitxes seleccionades per l'usuari.

Sortida: -

- `canviarFitxaFaristol(posicio1, posicio2)`

Nom: `canviarFitxaFaristol (posicio1:int, posicio2:int)`

Responsabilitats: Intercanviar *posicio1* per *posicio2* de les fitxes del faristol del jugador que té el torn.

Cas d'ús: Canviar fitxa del faristol.

Excepcions: -

Precondicions: Les posicions dels paràmetres d'entrada són posicions del faristol vàlides.

Postcondicions: S'han canviat les posicions de les fitxes del faristol.

Sortida: -

- `mostrarAjuda(jugador):jugada`

Nom: `Mostrar Ajuda (jugador: String):jugada:Jugada`

Responsabilitats: Es mostra una jugada amb les fitxes que té el jugador entrat com a paràmetre.



Cas d'ús: Mostrar l'ajuda.

Excepcions: No s'ha trobat cap jugada.

Precondicions: El paràmetre d'entrada és el jugador que té el torn actualment.

Postcondicions: En el taulell hi haurà una jugada vàlida.

Sortida: El sistema retornarà una jugada vàlida i la mostrarà en el taulell.

- `validarComodi(lletraComodi, fila, columna)`

Nom: `validarComodi(lletraComodi:String, fila:int, columna:int)`

Responsabilitats: Indicar la lletra representada per *lletraComodi*, que està en la fila *fila* i en la columna *columna* del taulell.

Cas d'ús: Validar els comodins

Excepcions: -

Precondicions: Hi ha un comodí en *fila*, *columna* del taulell actual que s'ha introduït en la jugada actual i s'està actualment en el procés de validació de jugada.

Postcondicions: El comodí del taulell representa la lletra indicada pel usuari.

Sortida: -

- `reiniciarCamera()`

Nom: `reiniciarCamera()`

Responsabilitats: Es reinicia la càmera del taulell.

Cas d'ús: Reiniciar la càmera.

Excepcions: -

Precondicions: La partida està en joc.

Postcondicions: La càmera es centra amb els angles inicials.

Sortida: -

- `pausarPartida()`

Nom: `pausarPartida()`

Responsabilitats: El joc s'atura i el sistema mostra les funcions que es permeten amb el joc pausat, i executa "aturarTemps" si en la configuració actual del joc està el temps activat.



Cas d'ús: Pausar la partida.

Excepcions: -

Precondicions: La partida està en joc i no està pausada.

Postcondicions: El joc queda pausat.

Sortida: -

- aturarTemps()

Nom: aturarTemps()

Responsabilitats: El temps s'atura i enregistra el temps restant.

Cas d'ús: Pausar la partida

Excepcions: -

Precondicions: La partida està en joc pausat i el temps està activat en la configuració actual.

Postcondicions: El temps s'atura.

Sortida: -

- tornarPartida()

Nom: tornarPartida()

Responsabilitats: El sistema deixa d'estar pausat i executa l'operació "continuarTemps" si està el temps activat en la configuració actual.

Cas d'ús: Tornar a la Partida

Excepcions: -

Precondicions: La partida està en joc i pausada.

Postcondicions: El joc es reprèn amb l'estat que hi havia anteriorment a pausar la partida.

Sortida: -

- continuarTemps()

Nom: continuarTemps()

Responsabilitats: El temps s'atura i es reprèn amb el temps que hi havia anteriorment.

Cas d'ús: Pausar la partida



Excepcions: -

Precondicions: La partida està en joc i s'ha pausat.

Postcondicions: El temps es reprèn amb el valor que hi havia anteriorment (si està activat en les regles actuals).

Sortida: -

- veureSac()

Nom: veureSac()

Responsabilitats: Es mostra les fitxes que queden del sac.

Cas d'ús: Veure el sac

Excepcions: -

Precondicions: La partida està en joc i pausada.

Postcondicions: Es mostren les fitxes que queden en el sac.

Sortida: -

- finalitzarPartida()

Nom: finalitzarpartida()

Responsabilitats: Finalitza la partida sense guardar cap dada.

Cas d'ús: Finalitzar la partida.

Excepcions: -

Precondicions: La partida està en joc, pausada i amb el temps aturat (si les regles permeten que hi hagi temps).

Postcondicions: La partida ha finalitzat i es mostra la pantalla d'inici.

Sortida: -

3.6 MODEL DE DADES

El objectiu d'aquest model és crear diagrames d'estats per objectes i casos d'ús, a través de:

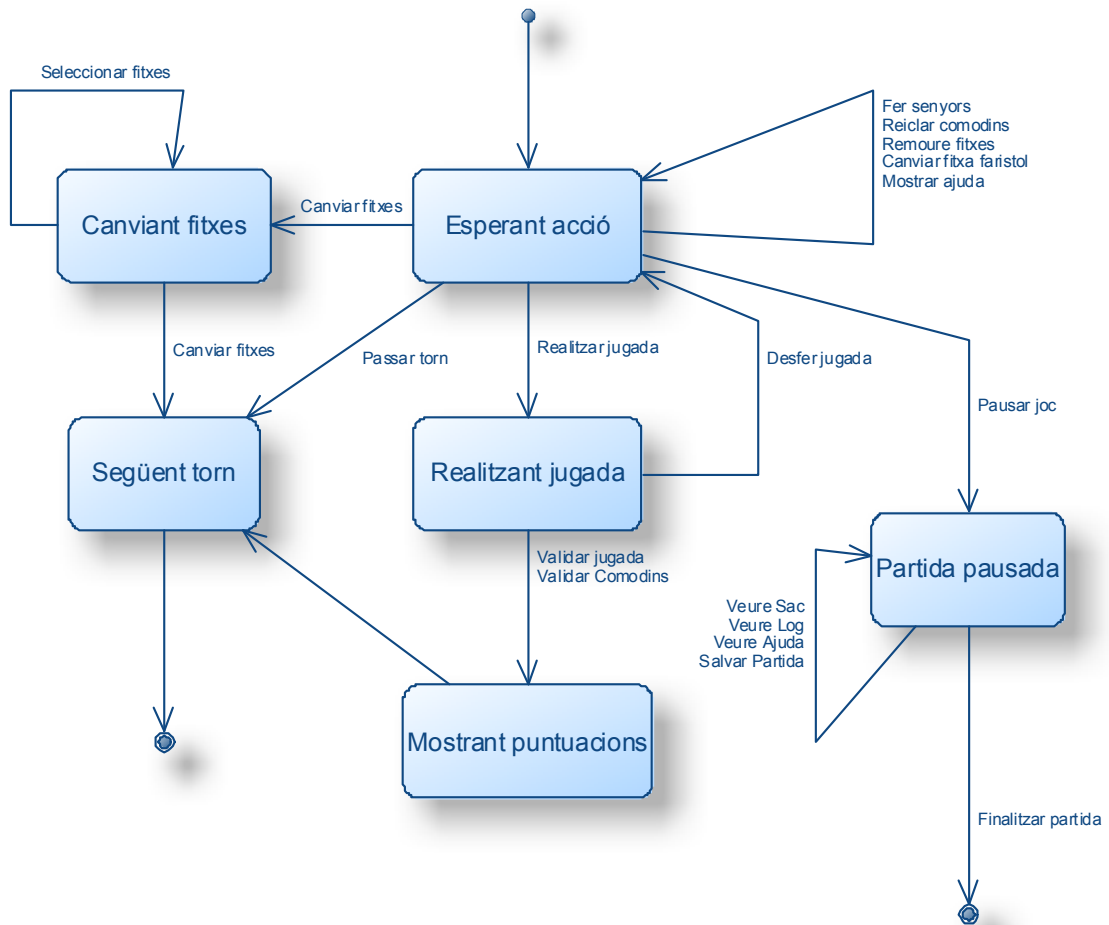
- Successos: Tot esdeveniment que requereix una resposta del sistema software.
- Estats: Condició d'un objecte o d'un cas d'ús en un moment del temps.
- Transicions: Canvi d'estat com a conseqüència d'un succés.

3.6.1 DIAGRAMA D'ESTATS DE CLASSES

Les ampliacions efectuades en el domini les explicarem més detalladament en les etapes de disseny i implementació ja que s'han afegit noves funcionalitats que afecten únicament en el disseny i la implementació, però els estats definits en la capa de domini seguirà sent idèntics l'aplicació original.

3.6.2 DIAGRAMES PRINCIPALS

En aquest apartat especificarem els estats que existeixen durant el transcurs del torn de la partida. Només comentarem els casos d'ús que tenen relació directe en el transcurs de la partida.

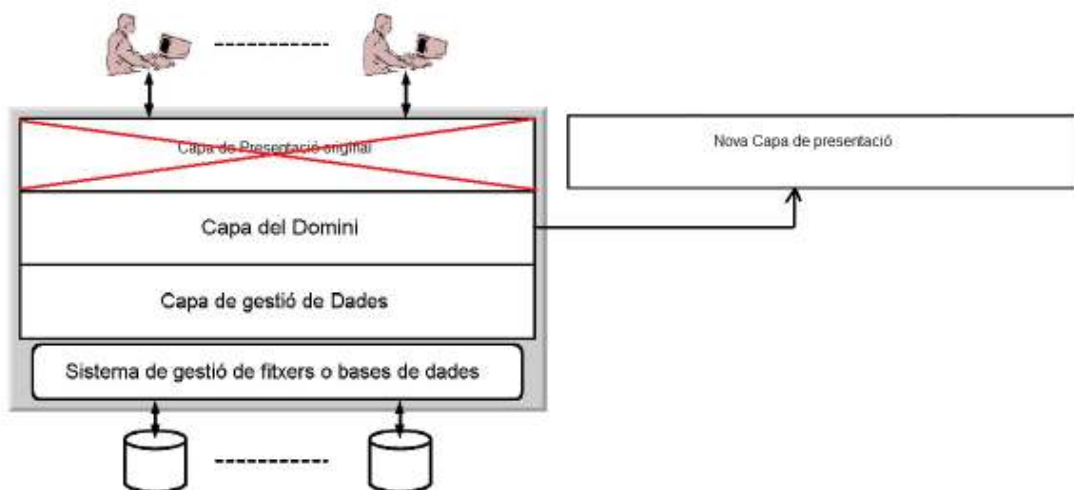


4 DISSENY

El disseny de l'aplicació inicial estava programat seguint el disseny de programació per capes (presentació, domini i dades).

El patró de disseny en capes té com a gran avantatge que facilita la canviabilitat (és a dir, extensibilitat, portabilitat, mantenibilitat i reestructurabilitat). Com a inconvenient, fa difícil d'obtenir l'eficiència òptima en afegir codi redundat necessari per permetre comunicar-se entre les diferents capes.

Gràcies a aquets avantatges es permetrà dissenyar novament tota la capa presentació i unir-la amb les capes de dades i de domini anteriors.



En les capes de domini i dades s'afegiran bàsicament noves funcions per permetre les noves funcionalitats però mantenint el disseny existent.



4.1 CAPA DE PRESENTACIÓ

La capa de presentació, o interfície, és l'encarregada de comunicar-se amb l'usuari. Tota la informació necessària del programa es mostra a través de finestres, està programada totalment de nou i es comunicarà amb les classes de domini que existien anteriorment.

Tant la informació mostrada del sistema com les ordres que l'usuari necessiti executar es farà a través d'aquesta capa.

4.1.1 DIAGRAMA CAPA PRESENTACIÓ

Per facilitar la comunicació entre domini i les diferents vistes de la capa de presentació, s'ha programat un controlador anomenat "Scrabble" amb el què les crides provinents de domini hauran d'invocar aquesta classe. Tanmateix, si l'usuari des de la pantalla requereix fer alguna acció (és a dir, que el flux de dades vagi de la capa de presentació cap a domini), el procediment anirà a l'inversa i les vistes hauran d'invocar aquest controlador per a comunicar-se amb el domini.

Utilitzant aquest mètode descrit anteriorment, teníem dos possibilitats per a programar aquest controlador: estàtica o dinàmicament.

La primera idea fou programar-la estàticament per a la comoditat d'evitar instanciar la classe i fent ús de totes les crides de manera estàtica rebaixant així càrrega de memòria. Però donada la importància d'aquesta classe i per a evitar excessives consultes entre domini i vistes, s'ha trobat més eficient programar-la dinàmicament i passar l'objecte "Scrabble" com a referència als Controladors de domini i les vistes de la capa de Presentació. D'aquesta manera permetem guardar dinàmicament dades en el controlador per una possible consulta més ràpida en cas que sigui necessari, evitant un major número de crides a domini.

En cada canvi d'estat la finestra que està visible invoca el controlador principal "Scrabble" informant de l'acció que s'executa, i a continuació el controlador invocarà a tots els controladors de domini necessaris. Finalment, l'actual pantalla deixarà de ser visible i es mostrarà la pantalla necessària.

Cal destacar que existeix una pantalla especial (V_Partida_Scrabble) que s'encarregarà de gestionar diferents classes internament, és a dir, des del punt de vista del controlador "Scrabble", només existirà la classe principal "V_Partida_Scrabble" i aquesta s'encarregarà de configurar adequadament cada una.

A continuació es mostra el diagrama de la capa de presentació

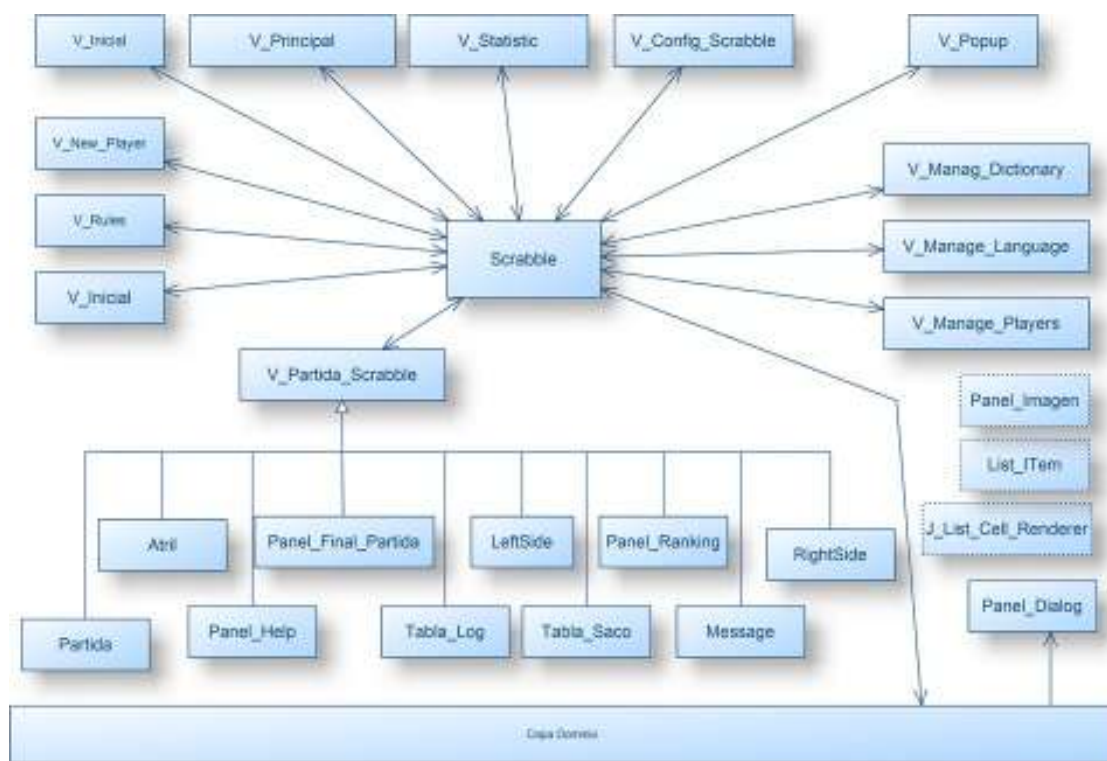
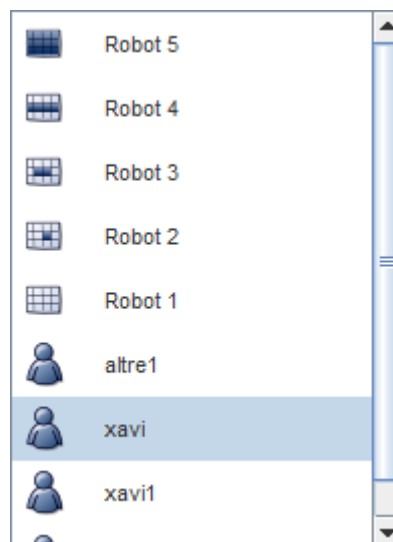


Diagrama capa presentació

Les tres classes genèriques `J_List_Cell_Renderer`, `List_Item` i `Panel_Imagen` són tres classes per a l'ús en qualsevol pantalla.

`Panel_Imagen` és una classe que permet afegir imatges de fons en una pantalla, en la fase d'implementació comentarem com s'ha d'invocar per utilitzar-la.

`List_Item` i `J_ListCell_Render` són les dos úniques classes de capa presentació que s'han aprofitat del projecte anterior. Aquestes dos classes permeten afegir un text i un icona en els `JList` de Java, afegint funcions per accedir al text seleccionat.



Disseny `J_List` nou

4.1.2 DISSENY DE LES PANTALLES

Totes les pantalles estaran programades en Java i utilitzaran la biblioteca gràfica per a Java "Java Swing".

A continuació mostrem el disseny de les diferents pantalles sense explicar les seves funcionalitats.

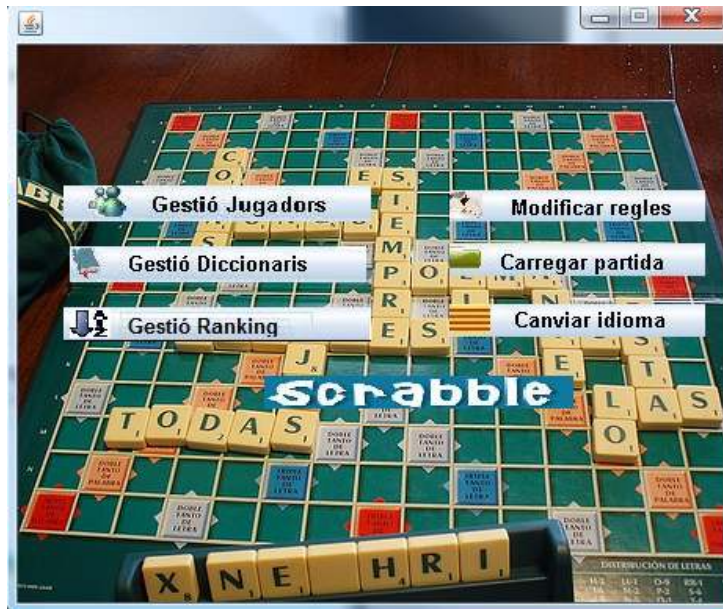
4.1.2.1 V_inicial

Pantalla inicial amb la que s'iniciarà l'aplicació.



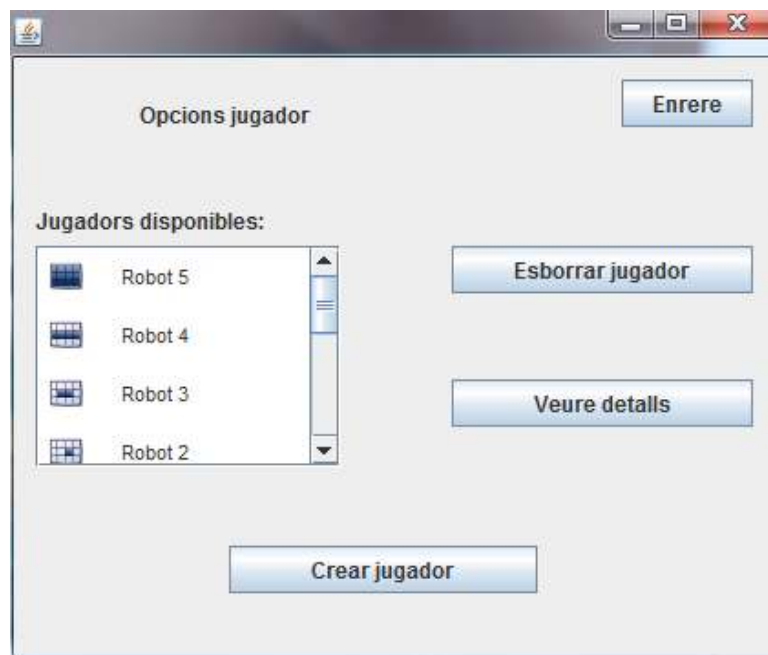
4.1.2.2 V_Principal

En aquesta pantalla l'usuari pot accedir a totes les opcions disponibles del joc, ja siguin per configurar la partida o per començar una nova partida.



4.1.2.3 V_Manage_Players

Pantalla utilitzada per la gestió dels jugadors.



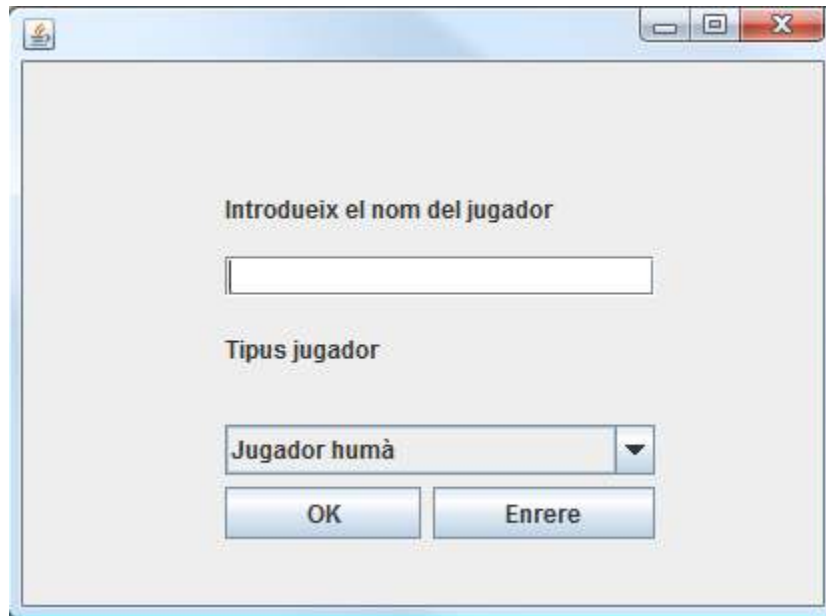
4.1.2.4 V_Statistic

Mostra les estadístiques d'un jugador determinat.

Jugador:	xavi1	
Scrabbles:	12	
Tipus jugador:	Humà	
Puntuació total:	12184	
Paraules totals:	472	
Millor partida:		
Puntuació:	542	
Paraules:	18	
MILLIARS		
Nº "0" per partida:	1	
Punts per partida:	197	
Punts per tirada:	37	
Millor jugada:		
Puntuació:	158	
Paraules:	2	
Millor paraula:		
Puntuació:	162	
Paraula:	PERLIADIS	
Totals:		
Partides guanyades:	5	
Partides perdudes:	3	
Partides empatades:	0	
<input type="button" value="Entreu"/>		

4.1.2.5 V_New_Player


Es permet donar d'alta un nou jugador.



The screenshot shows a dialog box titled "Introdueix el nom del jugador". It contains a text input field for the player's name. Below it is a label "Tipus jugador" followed by a dropdown menu currently set to "Jugador humà". At the bottom are two buttons: "OK" and "Enrere".

4.1.2.6 V_Manag_Dictionary

Pantalla per gestionar el diccionari actual.



The screenshot shows a window titled "GESTIÓ DICCIONARIS". It is divided into two sections. The top section, "GESTIÓ DICCIONARIS", shows "Diccionari actual per editar: Castellano" and a "Gestionar idiomes a carregar" button. To the right, there is a text input field "Escriu la paraula a consultar:" and a "Consulta" button. The bottom section, "GESTIÓ PARAULES", has two input fields: "Esborrar paraula" and "Afegir paraula", each with a corresponding button. A central "Enrere" button is also present.

4.1.2.7 V_Ranking

Pantalla per mostrar el rànking del jugador.



Ranking

Partides guanyades

	%	G	J	Jugador
1	71	5	7	Robot 1
2	52	5	8	scarl
3	33	1	3	abret
4	25	4	14	scarl
5	0	0	1	abre
6	-	0	0	RobotHumán
7	-	0	0	Robot 2
8	-	0	0	Robot 3
9	-	0	0	Robot 4
10	-	0	0	Robot 5

Partides perdudes

Puntuació actual

Scrabbles

Parades

Milior partida

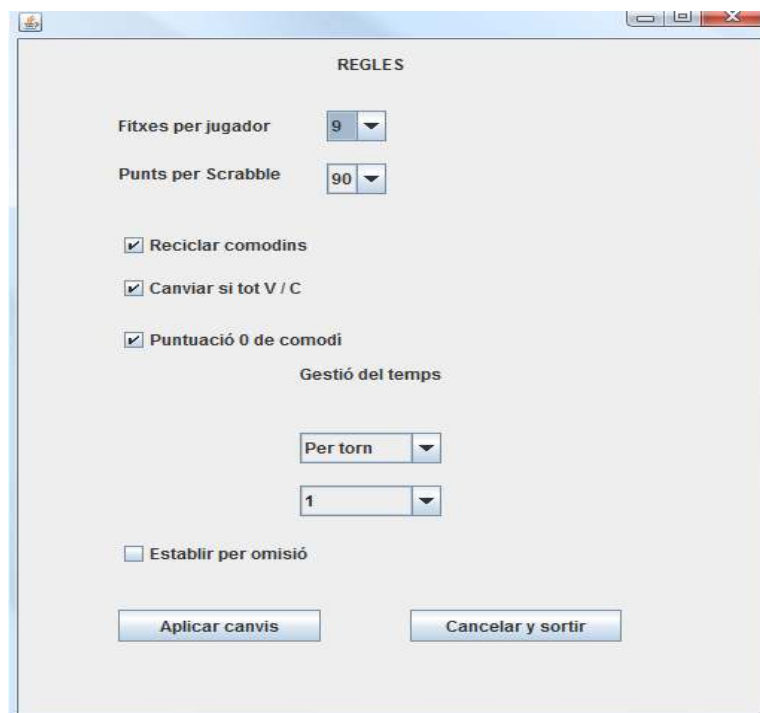
Milior jugada

Milior paraula

Enrere

4.1.2.8 V_Regles

Pantalla per configurar les regles de la partida.



REGLES

Fitxes per jugador: 9

Punts per Scrabble: 90

Reciclar comodins

Canviar si tot V / C

Puntuació 0 de comodi

Gestió del temps

Per torn

1

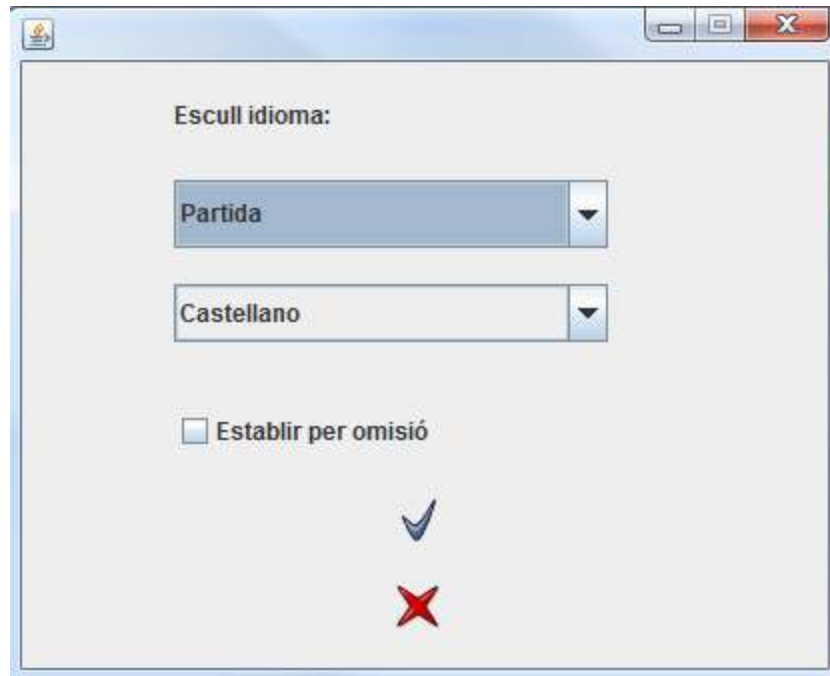
Establir per omisió

Aplicar canvis

Cancelar y sortir

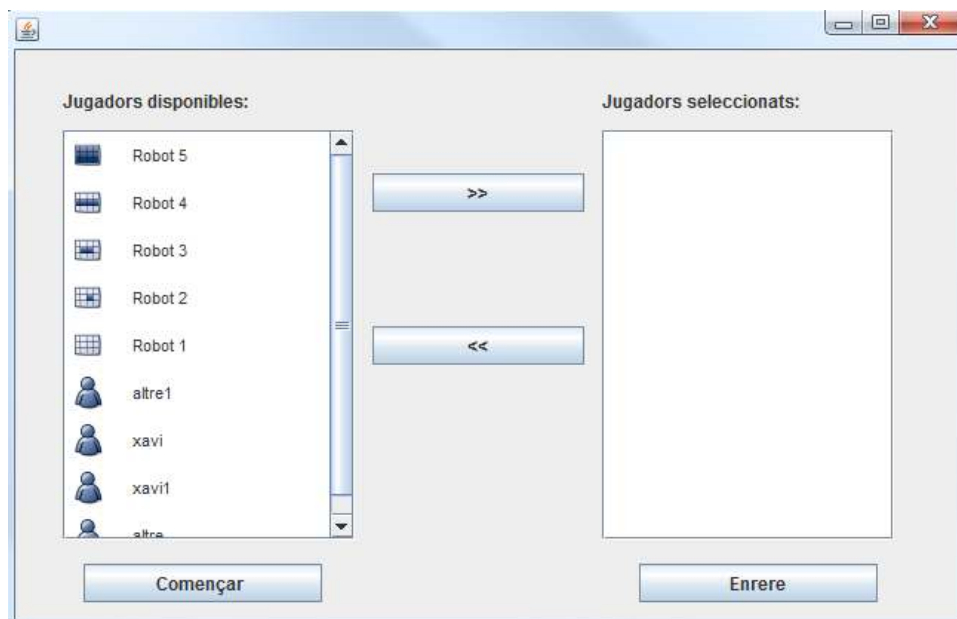
4.1.2.9 V_Manage_Language

Pantalla per configurar els idiomes de la partida.



4.1.2.10 V_Config_Scrabble

Pantalla per configurar els participants d'una partida.



4.1.2.11 V_Partida_Scrabble

Aquesta vista és possiblement la més important del joc ja que serà on es jugarà la partida. V_Partida_Scrabble és una vista-controlador que s'encarrega de situar en la finestra tots els components que gestiona internament detallats en el diagrama de la capa de presentació.

Els possibles dissenys de la pantalla es mostren a continuació.

Disseny de la pantalla durant el transcurs del torn:

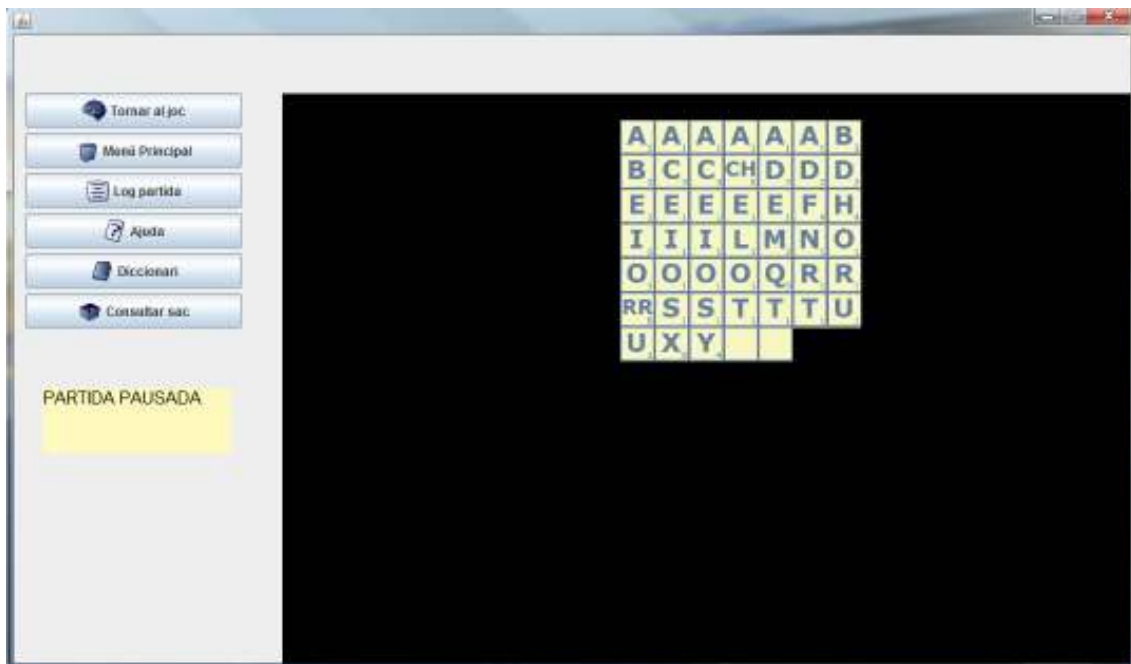


Disseny amb el torn de la partida validat:

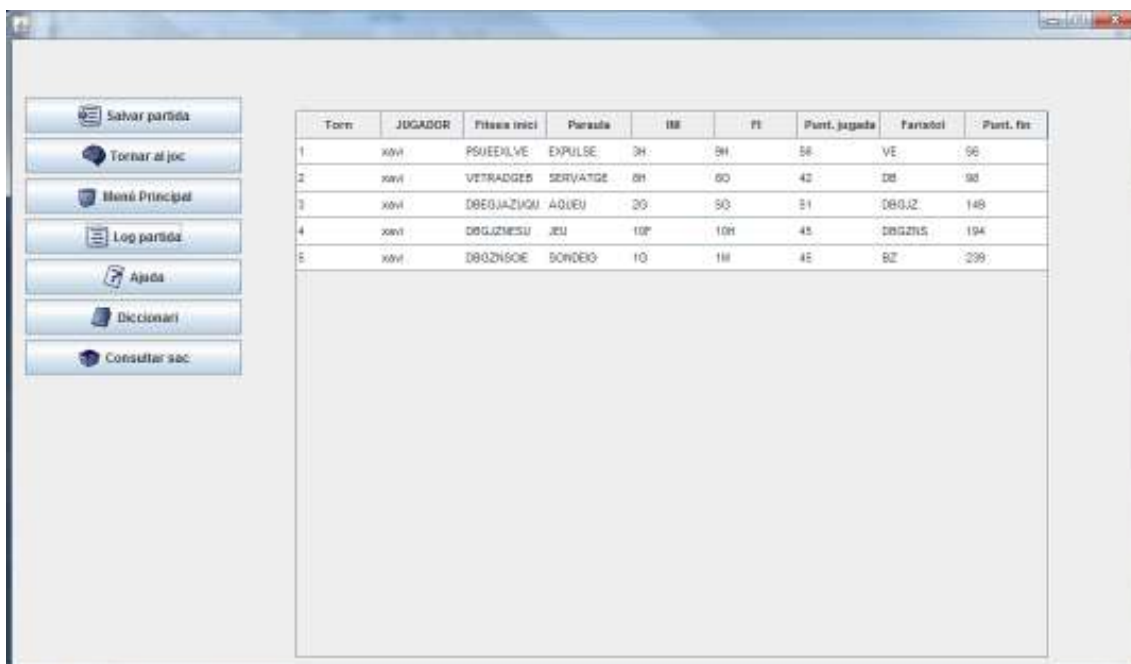


Amb la partida aturada es mostren diverses opcions en la part esquerra del panell. Depenen de l'opció que l'usuari demani, es mostrarà el sac, una ajuda del joc o el log de la partida en curs.

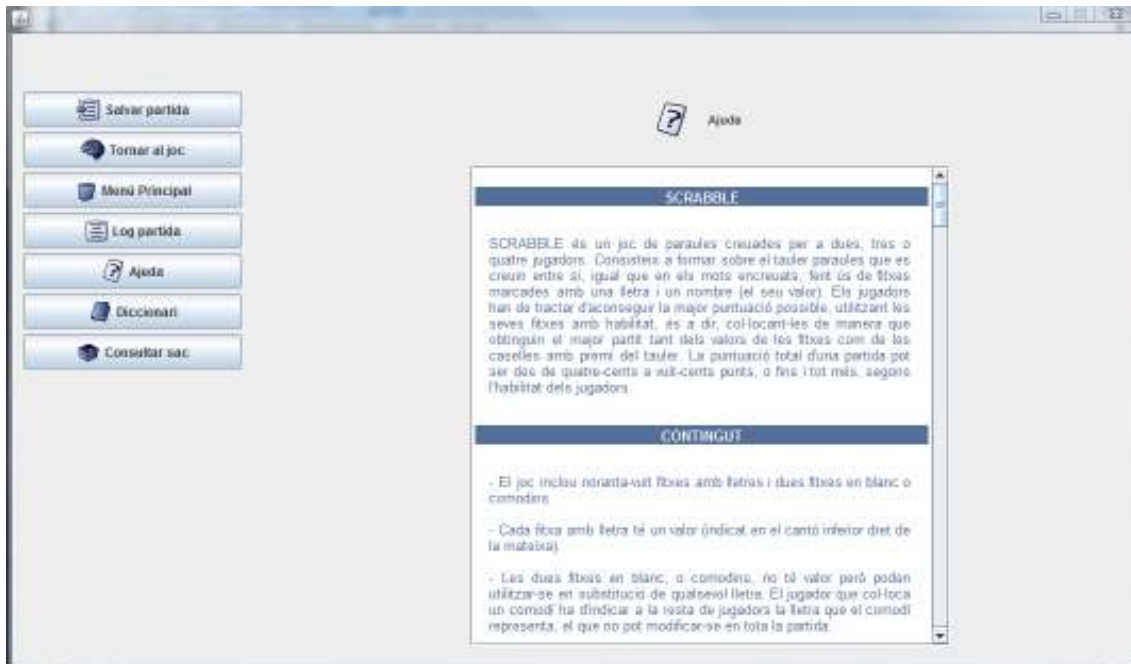
Partida aturada mostrant el sac de la partida:



Partida aturada mostrant el log de la partida:



Partida aturada mostrant l'ajuda:



4.2 CAPA DE DADES

4.2.1 PERSISTÈNCIES DE LES DADES

La persistència de les dades que utilitzarem serà al igual que en el projecte inicial, en fitxers. D'aquesta manera evitem fer canvis substancials en aquesta capa, i únicament afegirem les funcionalitats extres del projecte.

La forma en que desarem i obtindrem les dades seguirà el mateix mètode que en el projecte anterior, és a dir a través de mètodes propis del J2EE que permeten la interacció amb fitxers.

El tipus d'informació extra que desarem serà informació sobre jugadors, diccionaris vàlids, configuracions sobre el joc i dades pròpies sobre cada idioma de l'aplicació. En els pròxims diagrames mostrem amb més detall els nous atributs afegits en aquesta capa.

Per permetre la compatibilitat de l'aplicació per Linux i Windows, s'ha hagut de canviar el format de tots els fitxers que desàvem. En el manual de joc s'inclourà



facilitats per si es vol afegir paraules editant directament fitxers de l'aplicació sense trencar el format original.

El procediment que s'ha seguit és indicar en els mètodes de lectura/escriptura del J2EE el format en que s'haurà de llegir o escriure.

La codificació utilitzada per desar els fitxers serà en "UTF-8". Aquest procés es farà sempre per qualsevol operació de lectura/escriptura, ja sigui tant per llegir una paraula vàlida del diccionari, per posar un text en l'aplicació, per buscar una ruta de fitxer determinada, etc.

Si no s'indicava cap codificació per llegir/escriure en fitxers sorgien problemes en llegir caràcters especials en Linux.

4.2.2 CANVIS EN ELS FITXERS DE L'APLICACIÓ

En l'aplicació original hi havia perfils dels jugadors, partides salvades i les dades necessàries per a cada idioma disponible, com són el llistat de paraules vàlid pel joc (és a dir, el diccionari), el sac (el número de fitxes que hi ha per un determinat idioma i la seva respectiva puntuació per cada lletra), i els textos que es veuran en el transcurs de l'aplicació.

Òbviament, seguirem desant totes aquestes dades, però s'han fet alguns canvis de disseny per possibilitar funcionalitats extres indicades en l'anàlisi de requeriments:

-Al permetre gestionar diverses categories de paraules d'un mateix diccionari, hem eliminat el fitxer "*diccionario*", i s'ha afegit una carpeta anomenada "*Diccionarios*", en aquesta carpeta desarem tots els fitxers que permetrem carregar per la partida. D'aquesta manera durant l'execució de l'aplicació l'usuari podrà carregar les categories del diccionari desitjades.

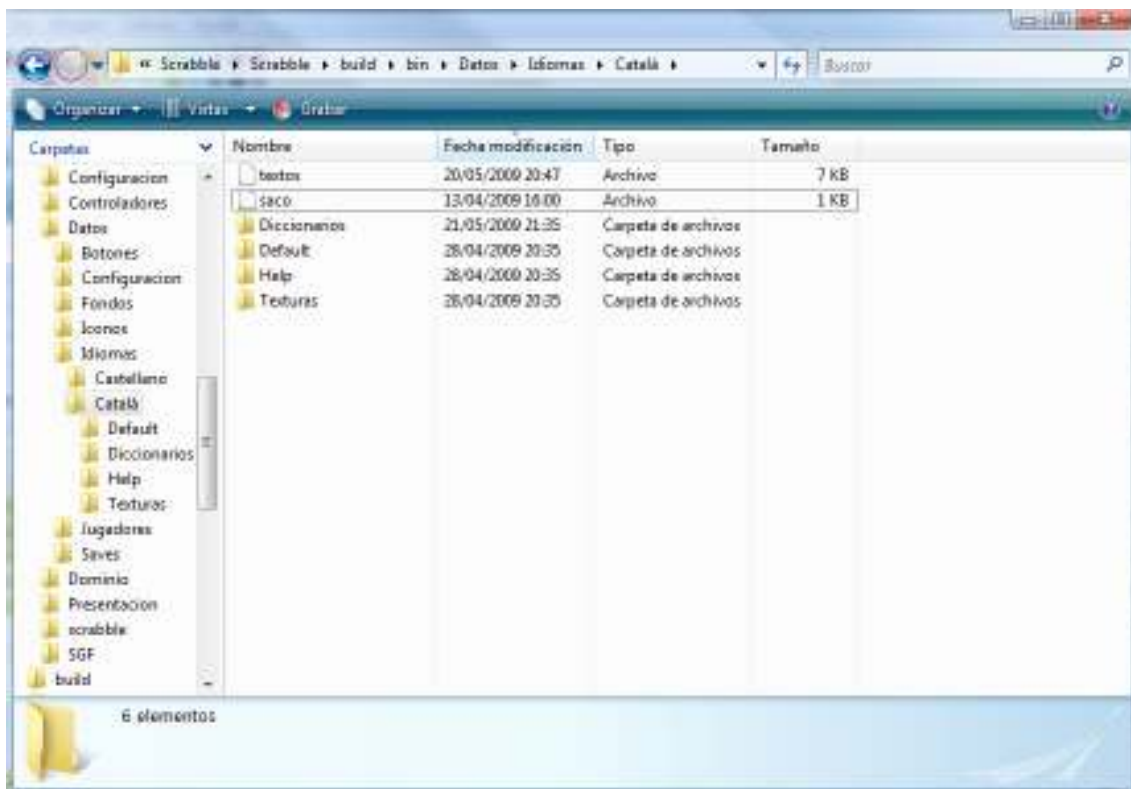
Aquests fitxers es desaran en el mateix format que tenia el fitxer "*diccionario*" de l'aplicació original, és a dir, un fitxer pla que contindrà un llistat de paraules seguides.

-Addicionalment es permet a l'usuari indicar quin llistat per defecte s'ha de carregar quan arrenqui l'aplicació o bé quan s'indiqui l'idioma per la partida. Per aquesta opció s'afegeix un nou fitxer anomenat "*list*" que es troba en la

carpeta “Default”. Aquest fitxer contindrà una llista amb els noms del fitxers que estan a la carpeta “Diccionarios” que es carregaran a l’escollir el idioma determinat.

-Els fitxers de partides salvades, logs, textos de l’aplicació , etc..., els desarem també en la nova codificació “UTF-8” anteriorment comentada.

Per tant, la ubicació dels fitxers en el disc serà la següent:



A continuació mostrem esquemàticament quina informació desarem en cada fitxer:

-Fitxers de jugadors:

Aquests fitxers els continuarem desant en format binari com en l’aplicació original. Afegirem les dades que necessitem per les noves funcionalitats, com per exemple les mitjanes de les partides.



Fitxers de jugadors
nom
tipusdeJugador{
HUMA
ROBOT1
ROBOT2
ROBOT3
ROBOT4
ROBOT5
ROBOTIGUALAT
}
puntuacioTotal
paraulesTotals
puntuacioMillorPartida
paraulesMillorPartida
puntuacioMillorJugada
paraulesMillorJugada
puntuacioMillorParaula
paraulaMillorParaula
partidesJugades
partidesGuanyades
partidesPerdudes
partidesEmpatades
mitjaPuntsPartida
mitjaPuntsTirada
mitja0Partida
scrabbles

-Fitxer de partides salvades:

En les partides guardades seguirem desant les dades en format binari i enregistrant tots els valors necessaris per poder continuar la partida en el punt que estava, inclosos els nous atributs necessaris per les noves ampliacions:

- Les categories del diccionari que s'està utilitzant en la partida.
- Les mitjanes actuals dels usuaris que hi ha en joc.
- Les noves regles que s'estan aplicant.



Fitxers partides salvades
<pre>int numeroJugadorsParticipants String[] jugadorsParticipants Per cada Jugador { String nomJugador int tipoJugador String millorParaulaPartidaActual int numParaulesPartidaActual int numParaulesMillorJugadaPartidaActual int puntuacioJugadorPartidaActual int puntuacioMillorJugadaPartidaActual int puntuacioMillorParaulaPartidaActual int numScrabblesPartidaActual int tempsRestantJugadorPartidaActual int numTornsSenseSolucioPartidaActual String lletresRestantsJugador } int numLletresSac String[] lletresSac String[][] taulellJoc int tornPartida int fitxesJugadorRegles int valorScrabbleRegles bool canviarScrabble bool ferSenyors int tipusTempsRegles int tempsJugadorRegles int tempsTornRegles bool esPartidaEmulada int numElementsIlistatDiccionari String[] categoriesDiccionari</pre>

-Fitxers llistats de paraules:

Els fitxers de diccionaris seran desats en fitxers de texts plans. En aquests fitxers es troben totes les paraules vàlides una darrere una altra; d'aquesta manera permetem que qualsevol usuari amb uns coneixements mínims pugui crear llistats de paraules personalitzats i es permetin afegir a l'aplicació simplement guardant el seu propi fitxer a la carpeta "Diccionario" del idioma que desitgi.

En el fitxer de “*categories per omisió*” desarem els noms dels fitxers de paraules vàlides que es carreguen per defecte en el idioma seleccionat.

Categoria diccionari
String paraulaValida1
String paraulaValida2
.
.
String paraulaValidaN

Categories per omisió
String llistatSubdiccionari1
String llistatSubdiccionari2
.
.
String llistatSubdiccionariN

-Fitxers d'ajuda:

Les ajudes de l'aplicació serà desada en format *HTML* com es trobava en l'aplicació original afegint-ne en el propi codi les funcionalitats noves.

Fitxers d'ajuda
<HTML>

-Fitxers de logs:

Un cop finalitzada la partida es permet desar el log de la partida.

Fitxers de logs
String[] jugadorsParticipants
Per cada jugada{
int numJugador
String nomJugador
String fitxesFaristolIniciTorn
String paraulaJugada
String iniciJugadaTaulell
String finalJugadaTaulell
String faristolFinalTorn
String puntuacioJugada
String puntuacioTotal
}

-Fitxers de configuració:

En els fitxers de configuració desarem les preferències del usuari sobre l'aplicació.

En el fitxer de les regles trobem les regles que es carreguen per omissió en l'aplicació, i en el fitxer de configuració desarem les preferències sobre el idioma de l'aplicació i de la partida.

Fitxers de configuració
String idiomaDiccionariOmissio String[] idiomesAplicació String idiomaAplicacioOmissio String idiomaPartidaOmissio

Fitxers de regles
String numFitxesOmissio String puntuacioScrabbleOmissio bool canviarComodinsOmissio bool ferSenyorsOmissio int tipusReglesTempsOmissio int tempsJugadorOmissio int tempsTornOmissio bool puntuacioComodi

-Fitxers de localització:

Tots els textos que hi hagin en l'aplicació es desen en aquest fitxer cadascun en la carpeta corresponent al seu idioma.

Els textos consisteixen en una clau i el text en el seu corresponent idioma, tots els textos nous s'han afegit en aquest fitxer.

Fitxers de localització
per cada text de l'aplicació { String clau String text }

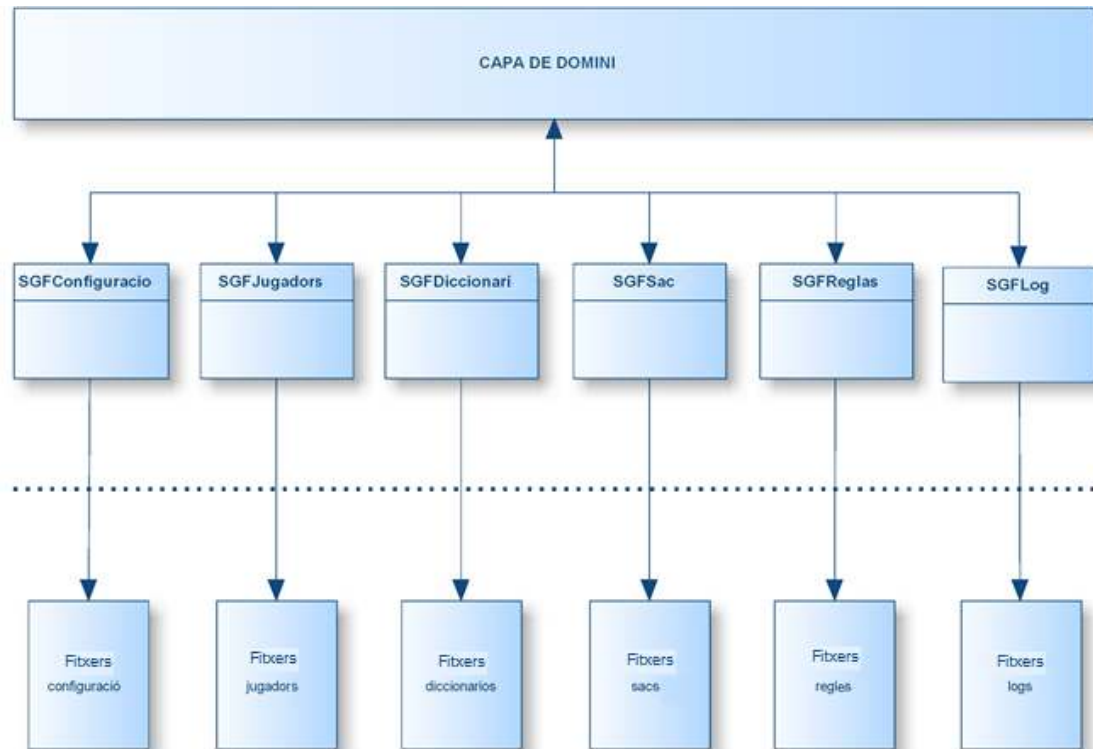


-Fitxers de sac:

En el sac desarem totes les lletres disponibles del idioma on està el fitxer, per cada lletra s'inclourà el valor de la lletra i les fitxes que hi ha amb aquest valor. Pels idiomes que tinguin lletres dobles, també hi haurà el valor de la lletra doble codificada, ja que per la implementació cada lletra ocupa un valor alfanumèric de una unitat de longitud.

Fitxers de sac
per cada lletra del idioma { String lletra String valor String numLletresSac }

Finalment mostrem el diagrama corresponent de la capa de dades, que s'ha mantingut idèntic al original.



4.3 CAPA DE DOMINI

Al tenir el joc programat seguint el model de programació per capes la renovació de la capa de presentació no hauria d'afectar en la capa del domini.

En el nostre projecte hem modificat la classe domini únicament per permetre executar les funcionalitats noves que afectin a capes inferiors.

Els canvis introduïts han estat únicament agregacions de noves funcions en les classes d'aquesta capa, però no ha estat necessari crear cap nova classe ni cap canvi de disseny transcendent.

Cal destacar la importància de la classe estàtica "Configuració", aquesta classe s'encarrega de proporcionar als diversos controladors de l'aplicació valors de la capa de dades tal com textos, o rutes de arxius. Al ser estàtica, no caldrà



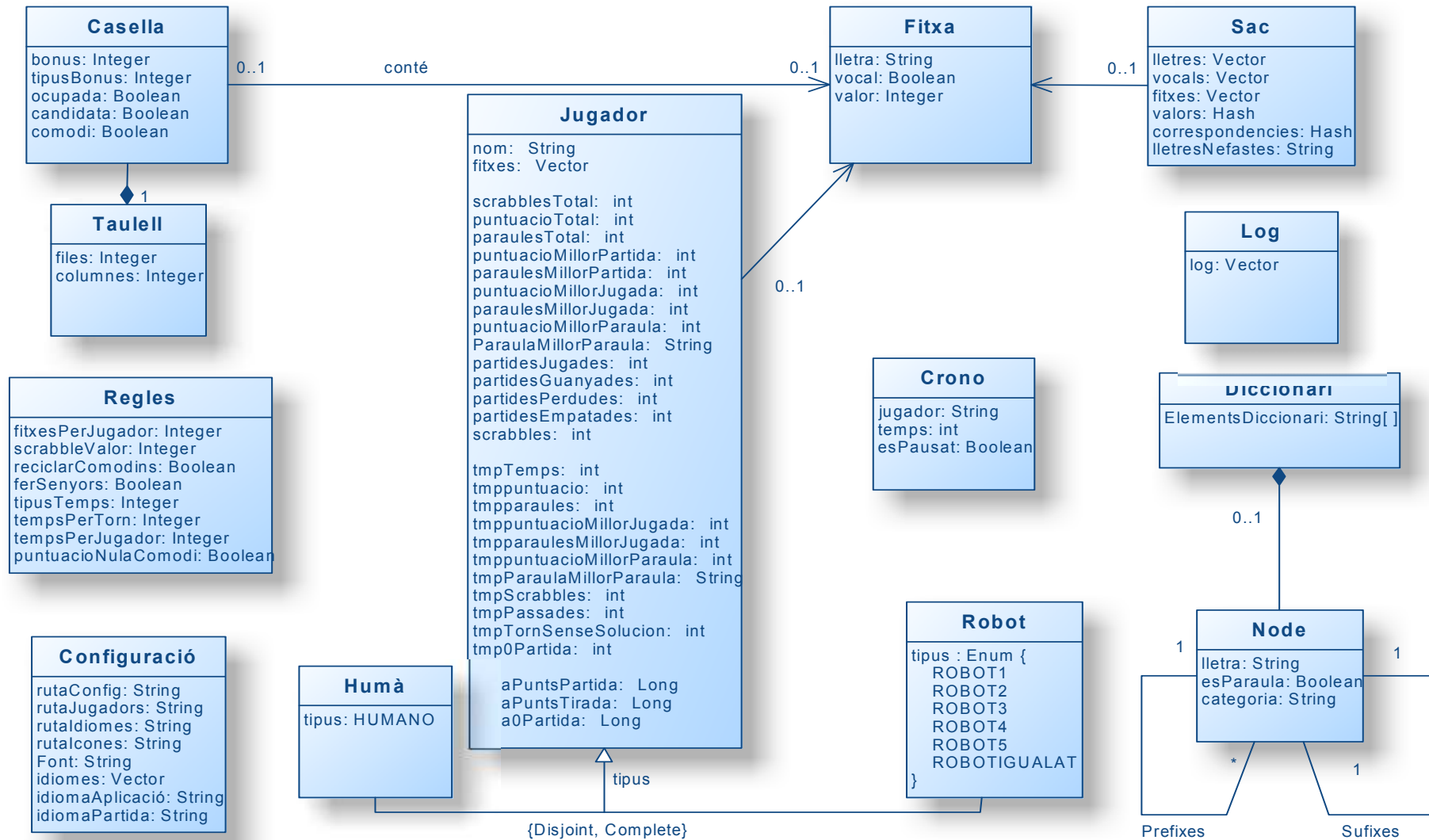
instanciar-la i hem permès que des de qualsevol punt de l'aplicació es pugui invocar les funcions d'aquesta classe.

En resum, els canvis d'aquesta capa únicament han estat ampliacions de funcions i atributs de classes ja existents, per tant, el diagrama de classes normalitzat serà idèntic a l'aplicació original amb nous camps.

Evitem per qüestió de comprensió, incloure les funcions de cada classe, ja que quedaria un diagrama excessivament gran

4.3.1 DIAGRAMA DE CLASSES

A la pagina següent es mostra el diagrama de domini:





Els canvis introduïts en el disseny són els següents:

- “Regles”“: S’ha afegit l’atribut “puntuacioNulaComodi” per permetre controlar la nova regla de la puntuació del comodí.
- “Jugador”“: S’han afegit nous atributs necessaris per l’ampliació d’estadístiques que hem realitzat, i per què en les partides igualades el “Robot igualat” pugui acotar les puntuacions consultant les mitjanes del jugador al que s’està enfrontant. Finalment hem afegit un nou tipus de jugador anomenat “ROBOTIGUALAT” que s’encarrega de enfrontar-se a un jugador en partides igualades.
- “Diccionari”“: Al ampliar el funcionament de les paraules vàlides de l’aplicació permetent carregar categories del diccionari, s’ha necessitat emmagatzemar en l’atribut “ElementsDiccionari” els noms dels fitxers del que està format el diccionari actual.
- “Node”“: S’ha afegit en cada node del que forma part el diccionari el nom del fitxer on s’ha llegit la paraula emmagatzemada (categoria de la paraula).
- La classe “Crono” s’ha permès reiniciar i pausar el temps per possibilitar la millora definida en etapes anteriors de pausar i arrancar el temps de les partides.

4.3.2 PATRONS DE DISSENY: CONTROLADOR

En el domini seguirem utilitzant el mateix patró de disseny que s’utilitzava anteriorment: el patró *Controlador*. Utilitzant aquest patró es té l’avantatge que cada esdeveniment extern que es rep provinent de l’usuari li arribarà al controlador general de domini i es reenviarà la informació al controlador especialitzat el qual s’encarregarà d’efectuar les operacions necessàries. La classe de presentació “Scrable” s’encarregarà sempre de comunicar-se amb el controlador principal de domini (“Controlador de partida”) i aquest es comunicarà amb els controlador i/o classes que necessiti per executar les funcionalitats demanades.

Per tant, totes les ampliacions efectuades en la capa de presentació que tinguin repercussions en domini i/o en dades es veuran sempre reflectides en el



controlador de la partida, i aquest controlador s'encarregarà d'invocar els controladors especialitzats, per aquesta raó el controlador de la partida ha estat el controlador de domini amb més modificacions.

Al existir ara nous tipus de partida, s'han hagut de dissenyar diferents inicis de torn tenint en compte quina partida s'està jugant, en els diagrames de seqüència del següent apartat es pot comprovar els canvis de disseny més importants que s'han realitzat per que el controlador segueixi comunicant-se entre el domini i la presentació de manera correcte.

4.3.3 CANVI EN EL DISSENY DE CARREGA DE DICCIONARIS

El procés de càrrega de diccionaris en memòria s'ha decidit carregar les paraules en el inici de l'aplicació, al canviar el diccionari a carregar o al canviar el idioma de la partida, en comptes de carregar-lo al inici de cada partida.

Amb aquest canvi s'alenteix el inici de l'aplicació i la modificació del diccionari de la partida degut a que en aquest moment es desarà en memòria les paraules que es permetran posar en el taulell. Però en canvi, es millora en eficiència a l' hora de començar una partida ja que el diccionari ja estarà carregat en memòria i l'usuari no haurà d'esperar.

Per exemple, suposant que un usuari arrenqui l'aplicació i jugui quatre partides seguides sense canviar d'idioma, el temps que l'usuari haurà d'esperar per la càrrega en memòria del diccionari serà únicament al iniciar l'aplicació, en canvi si la càrrega del diccionari s'hagués realitzat com en l'aplicació original, cada cop que s'iniciés una partida hauria d'esperar a que es carregui en memòria el diccionari.

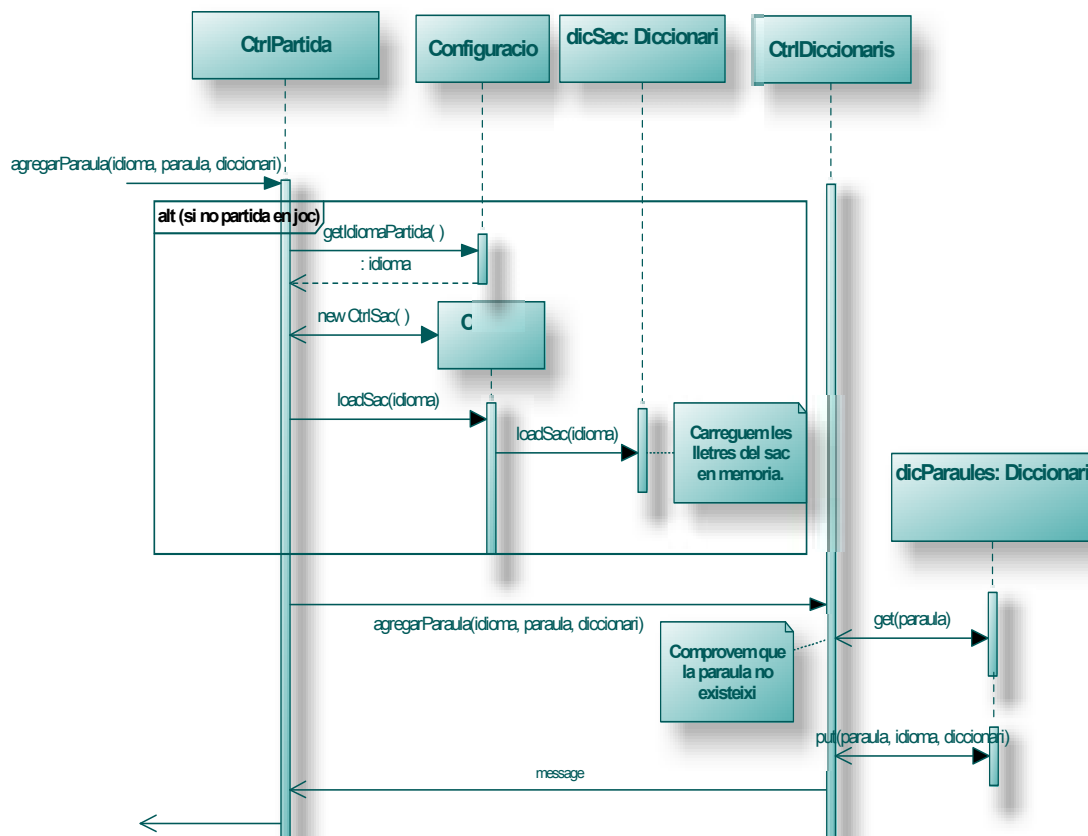
4.3.4 DIAGRAMES DE SEQÜÈNCIA

A continuació desglossarem els diagrames de seqüència de les operacions més importants.

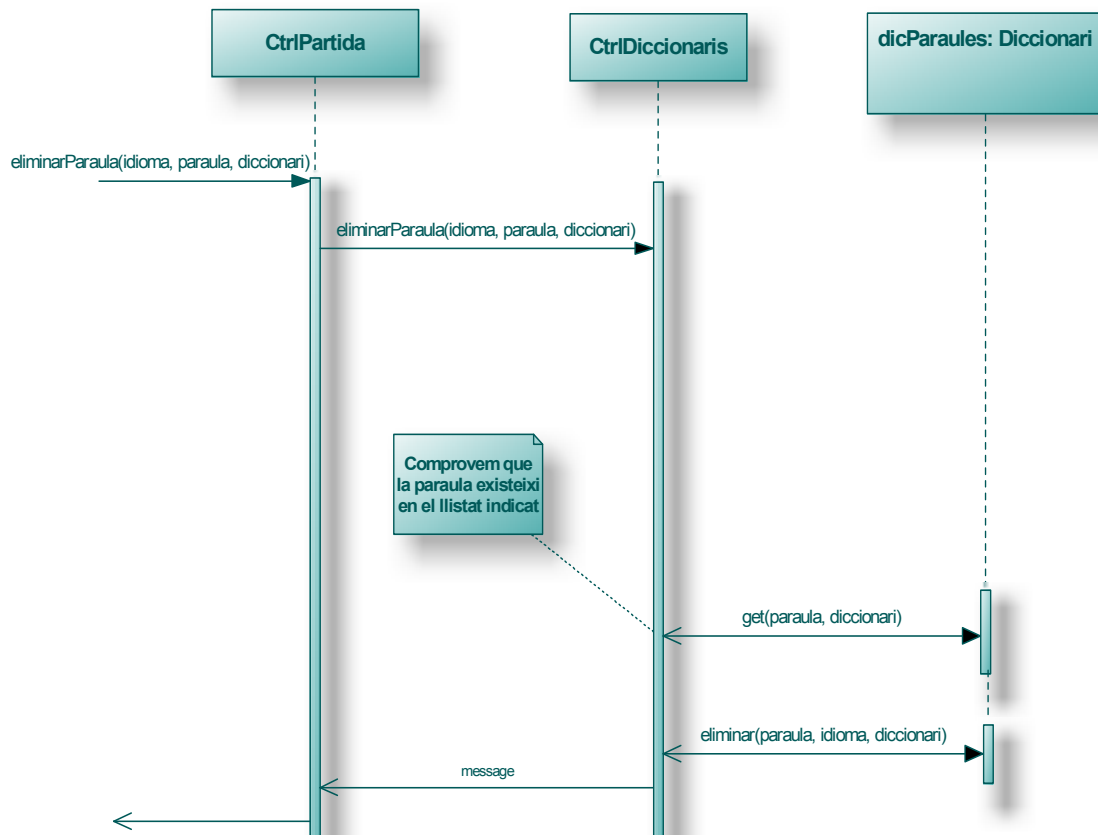
Hem evitat desglossar totes les operacions per evitar repetir diagrames efectuats en l'aplicació original i únicament mostrarem les operacions que més canvis han rebut.

Totes les crides comencen en el controlador principal de domini (CtrlPartida), aquesta crida serà invocada des de la classe "Scrabble" que com hem explicat anteriorment, s'encarrega de captar tots els esdeveniments de l'usuari i connectar amb domini.

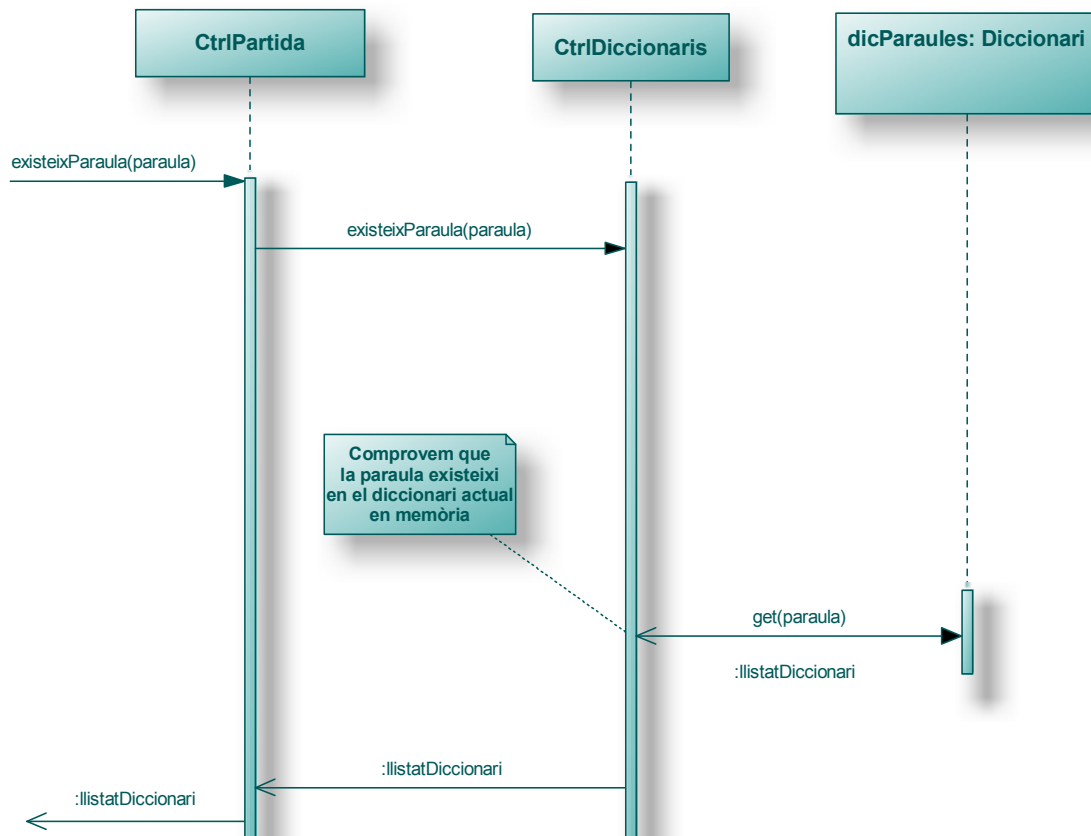
4.3.5 DIAGRAMA DE SEQÜÈNCIA: afegirParaula



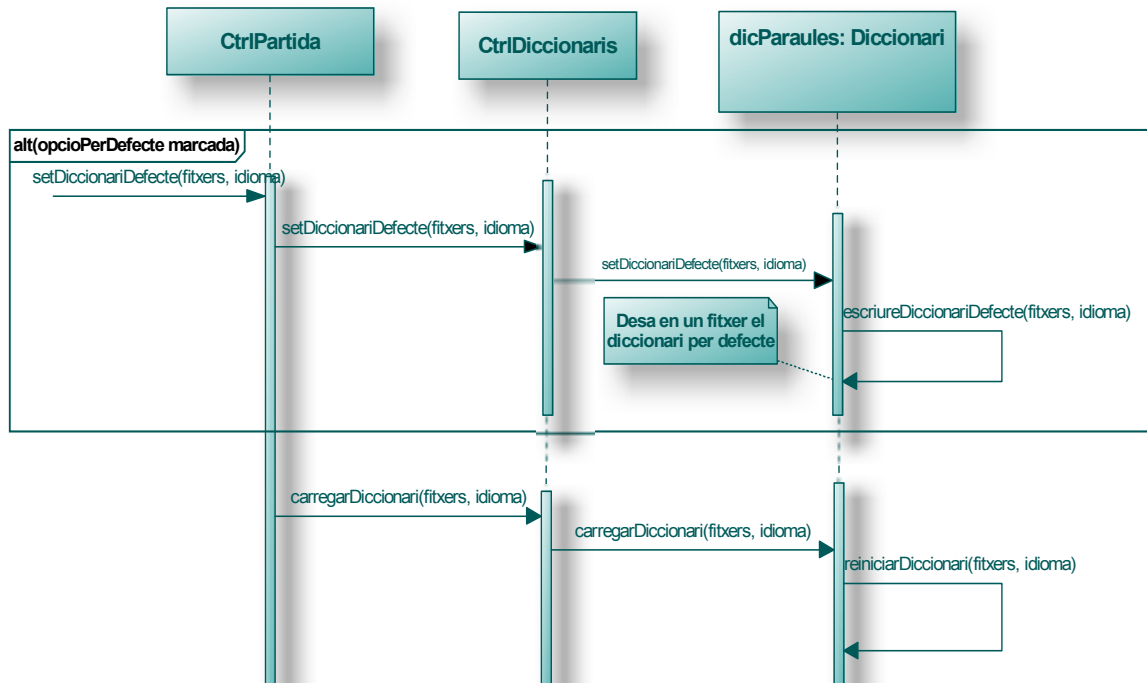
4.3.6 DIAGRAMA DE SEQÜÈNCIA: eliminarParaula



4.3.7 DIAGRAMA DE SEQÜÈNCIA: consultarParaula

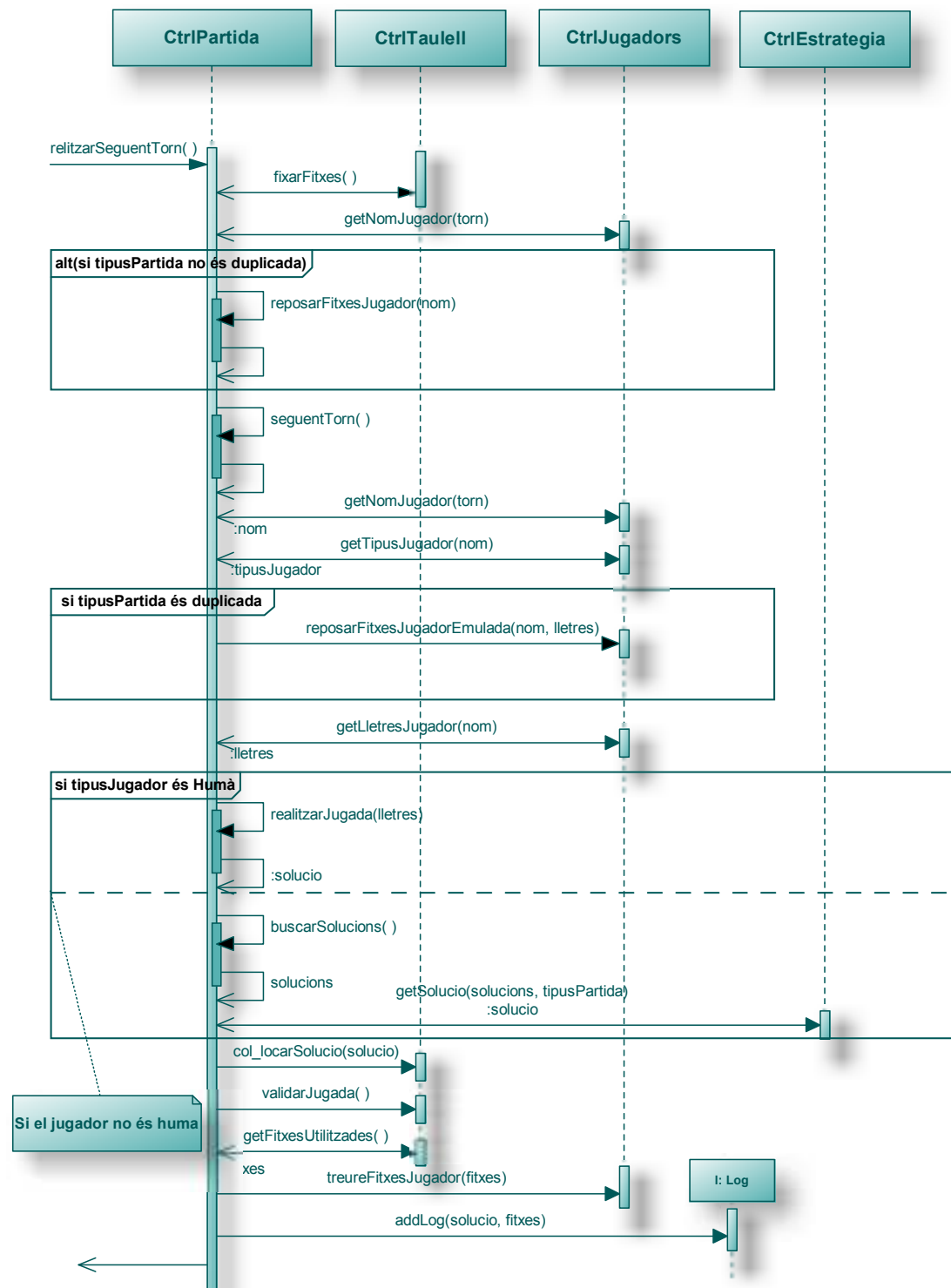


4.3.8 DIAGRAMA DE SEQÜÈNCIA: definirCategoriesParaules

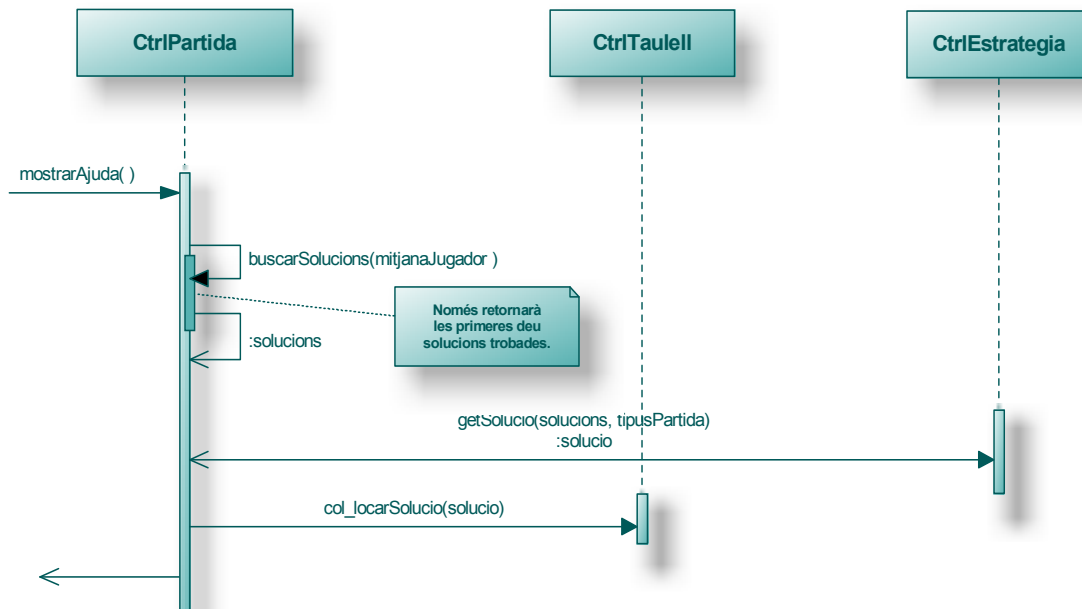


4.3.9 DIAGRAMA DE SEQÜÈNCIA: realitzarSegüentTorn

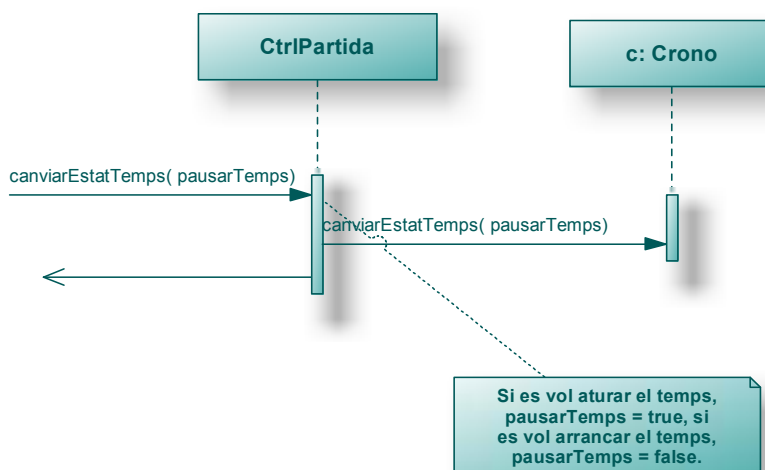
Aquesta funció ha mantingut l'esquema original, però afegint condicionals en l'execució en funció de quin tipus de partida s'està jugant.



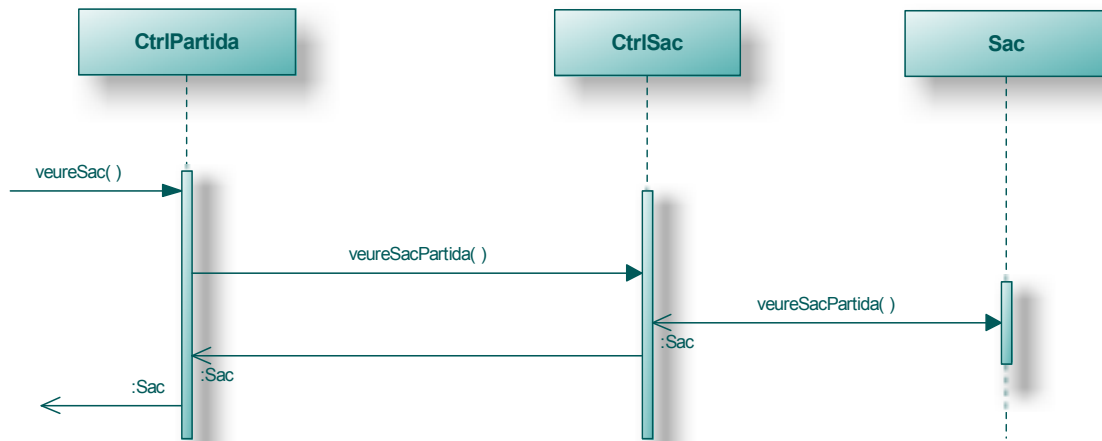
4.3.10 DIAGRAMA DE SEQÜÈNCIA: mostrarAjuda



4.3.11 DIAGRAMA DE SEQÜÈNCIA: pausarTemps, tornarPartida



4.3.12 DIAGRAMA DE SEQÜÈNCIA: veureSac





5 IMPLEMENTACIÓ

5.1 INTRODUCCIÓ

L'objectiu d'aquest capítol és detallar d'una manera general com s'implementa el disseny definit en etapes anteriors, a la vegada que eliminem tota la implementació de la capa de presentació del projecte anterior per substituir-la per les nostres.

En les fases anteriors s'ha mostrat el disseny que s'ha de portar a terme per poder realitzar l'idea, però arribat en aquest moment ens trobem en que tot el disseny s'haurà de plasmar sobre una tecnologia determinada (En el nostre cas en el llenguatge de programació Java).

Durant aquesta secció anirem mostrant com hem implementat la capa de presentació, i els canvis que s'han fet sobre la capa de dades i domini.

El projecte original es migrarà per fer la implementació en la plataforma *NetBeans*. Aquest programa conté un potent editor de pantalles i facilita la gestió dels esdeveniments.

5.2 CAPA DE PRESENTACIÓ

Aquesta capa serà la que substituïrem, per tant el primer pas serà identificar totes les crides a les classes de la capa de presentació original per anar substituint-les per les classes pròpies que s'aniran implementant.

Totes les pantalles que anirem creant utilitzaran les biblioteques de *JavaSwing* i seran implementades amb la plataforma *Netbeans* seguint el model de disseny creat en la fase anterior.

Com s'ha comentat en l'etapa de disseny aquesta capa està programada a través de diverses pantalles, en els següent apartats comentarem l'estructura de com s'han programat totes les pantalles excepte la pantalla principal del



taulell que té una complexitat superior a la resta de pantalles i s'explicarà en un apartat especial.

5.2.1 IMPLEMENTACIÓ PANTALLES

Al començar a programar l'aplicació ens trobem amb el problema inicial de quina classe utilitzar de les biblioteques que ens dona Java. Les dos típiques "finestres" per mostrar objectes en pantalla que ens prové Java són el "JFrame" i el "JDialog". Per tant el primer problema era quina utilitzar, JFrame o JDialog?

Pot semblar en termes generals que sigui independent utilitzar una o l'altra, però existeixen diverses diferències entre les dos:

Si instanciem un JFrame, sota la barra de programes del sistema operatiu utilitzat podem observar una nova pestanya indicant que s'ha obert una nova aplicació, en el cas del JDialog aquest fet no passa.

El JFrame té més mètodes de personalització que el JDialog.

El JDialog permet tenir una altra finestra per "sobre" seu, en canvi el JFrame no permet finestres "pare".

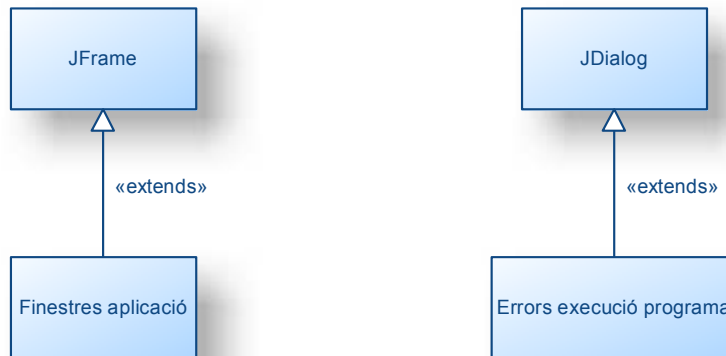
Estudiant aquestes característiques és evident que el JFrame ha de ser la finestra principal del programa, mentre que el JDialog l'utilitzarem per a pantalles secundàries com per exemple els errors de configuració que hi hagin en l'aplicació.

Si no ho implementéssim d'aquesta manera podríem trobar problemes tals com:

- Si en el sistema operatiu s'obren diverses finestres d'altres programes al estar l'aplicació en curs, l'usuari pot trobar dificultats per fer visible la finestra del nostre joc al no estar en la barra dels programes del sistema operatiu.

-JDialog al permetre finestres "pare" és ideal per mostrar missatges d'error ja que la finestra filla estarà sempre visible per sobre de la finestra principal, i s'eviten problemes de control en la visibilitat.

-Per tant, s'ha arribat a la conclusió de programar totes les finestres principals utilitzant la classe de Java *javax.swing.JFrame*, i els errors que s'hagin de mostrar en l'aplicació seran *javax.swing.JDialog*.



Herència classes de presentació

5.2.1.1 PANTALLES PRINCIPALS

De l'etapa anterior decidim que les finestres principals seran extensions del JFrame i l'estructura de totes les finestres principals serà de la següent forma:

```
public class Nom_classe extends
javax.swing.JFrame {
    /*
    Atributs del Frame
    */
    public Nom_classe(paràmetres constructor) {

        /*
        Assignacions paràmetres
        */
        initComponents();
        /*
        Codi propi
        */
    }
}
```

En el codi observem que les classes s'estendran de JFrame, i en els atributs de la classe definim tots els paràmetres definits en el disseny.

En la constructora inicialitzem els atributs i el mètode *initComponents()* s'encarrega de configurar totes les propietats visuals de la pantalla, com pot ser el tipus de lletra, la grandària de la lletra, situació dels botons, etc.

Els components de la finestra seran també objectes de la llibreria *javax.swing*, aquests components podran ser, botons, etiquetes de text, llistats, ...

Amb aquesta llibreria assignem els icones necessaris als objectes creats cridant al mètode *.setIcon(new ImageIcon("ruta de la imatge")) ;*

Un cop definit el tipus de finestra que desitgem crear, i definida també l'estructura visual de la finestra manca programar com realitzarem les accions que demani l'usuari.

En l'etapa de disseny hem vist que el controlador Scrabble s'encarrega de gestionar totes les accions que l'usuari demana, per tant un dels paràmetres d'entrada dels constructors del JFrame serà la classe Scrabble, i per saber el moment en que haurem d'invocar aquest controlador serà a través de *Listeners* assignats als elements de la pantalla.

Un *Listener* bàsicament té la funció d'estar pendent de qualsevol acció del objecte que se li ha assignat, com per exemple un *click*, per llavors executar la funció de codi que li hem definit.

A continuació mostrem com s'implementa el procés d'assignació d'un *Listener* a un JButton :

```
jButton.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent  
    evt) {  
        funcioAlClickarElBoto(evt);  
    }  
});
```

Un cop definit el *Listener*, s'implementa la funció (en aquest cas *funcioAlClickarElBoto*) on invocarem al controlador principal Scrabble informant l'acció que l'usuari ha realitzat.

5.2.1.2 PANELL DE IMATGE

S'ha implementat una classe extra que pot utilitzar qualsevol JFrame per permetre afegir una imatge com a fons de la finestra.

La constructora d'aquesta funció retorna una extensió del JPanel amb la imatge indicada per a poder-la situar en el fons del JFrame.

Per invocar aquesta classe s'implementarà de la següent manera en la finestra que volem incloure-li la imatge:

```
JPanel panel=new PanelImagen(ruta_imatge);  
this.setContentPane(panel);  
this.getContentPane().setLayout(null);
```

Aquesta invocació és necessària fer-la just abans d'invocar la funció *initComponents()*. D'aquesta forma s'insereix el panell indicat amb la imatge que es passa per paràmetre.

5.2.1.3 PANELL PRINCIPAL PARTIDA

En aquesta secció ens centrarem en la classe *V_Partida_Scrabble* que també és una extensió del JFrame. Aquesta classe s'encarrega de tota la gestió de la pantalla durant una partida.

Per evitar complexitat en la implementació del codi d'aquesta finestra, s'ha dividit la pantalla en diferents "panells", d'aquesta manera aquest controlador configura la visibilitat i situació dels panells que siguin necessaris en cada moment de la partida, i invocarà les funcions de cada panell necessari per que es configurin independentment cada un. Les propietats de cada panell seran



configurades independentment per cada component, cada un d'ells és una extensió de la classe `JPanel` i estarà programada a part.

Un `JPanel` és un contenidor de propòsit general per components de pes lleuger. Com tots els contenidors *JavaSwing* es permeten posicionar i dimensionar dins un `JFrame` i permet afegir i configurar propietats i objectes dins el propi `JPanel`.

Durant la major part del transcurs de la vida d'aquesta finestra hi hauran cinc components principals visibles:

`Message.java`

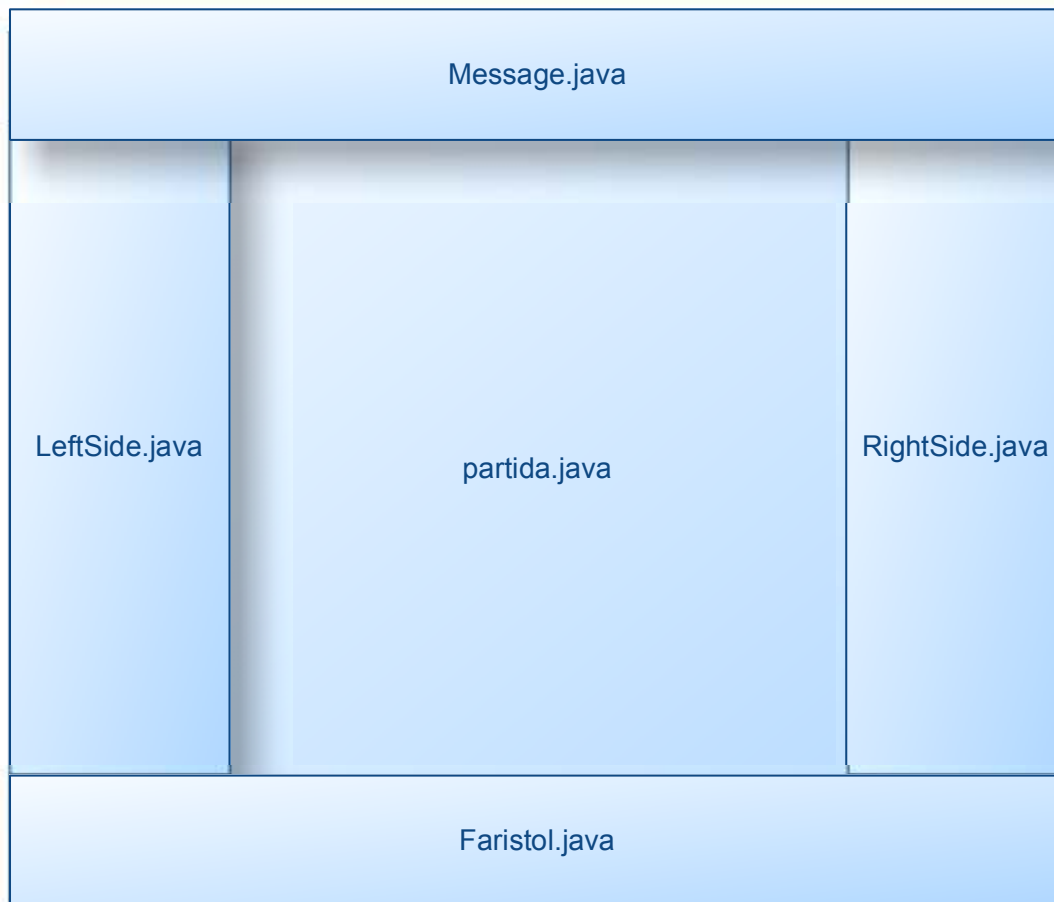
`LeftSide.java`

`RightSide.java`

`Partida.java`

`Faristol.java`

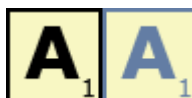
A part d'aquests components hi hauran altres que s'aniran mostrant en funció de les necessitats de l'usuari, com el log de la partida, el sac complet de la partida o una pantalla amb l'ajuda de l'aplicació.



Situació dels components principals de V_Partida_Scrabble.java

Cada component està implementat seguint la mateixa estructura dels JFrames anteriors , usant el mètode *initComponents()* per configurar les propietats visuals de cada component i els *listeners* configurats per informar de les accions que l'usuari pot realitzar.

El component Faristol.java és l'encarregat de mostrar les lletres de l'usuari amb el torn actual. Per mostrar la diferència d'una fitxa seleccionada a una desseleccionada s'ha decidit canviar la textura a carregar del botó, en el cas de fitxes seleccionades la textura que s'utilitza és més fosca:





El mètode que utilitzarem per a carregar la textura en un botó serà la següent:

```
jButton1.setIcon(new  
ImageIcon(getClass().getResource("Ruta_Textura")));
```

5.2.1.3.1 IMPLEMENTACIÓ TAULELL

El panell *Partida.java* s'encarrega de la representació del propi taulell de la partida i les fitxes del joc. S'ha decidit implementar-ho utilitzant biblioteques que proporciona Java de JOGL destinades a la visualització 3D per fer més visual l'aplicació. Aquesta biblioteca permet accedir al OPENGL mitjançant Java.

A continuació mostrarem les parts més destacables de la implementació en *JOGL* del taulell:

Inicialització

El primer cop que instanciem la classe, inicialitzarem les propietats 3D (GL_DEPTH_TEST, Color, Mida del buffer de profunditat...). Un cop inicialitzades les propietats també introduïrem en un Hash totes les possibles textures que es poden carregar en el taulell (caselles, lletres i colors de les fitxes). Aquesta implementació millora l'eficiència al moment de carregar les textures, però la inicialització del taulell pot ser una mica més lenta perquè haurà de carregar totes les possibles imatges que es veuran en el taulell.

En aquesta classe hi haurà el mètode *display* que serà l'encarregat de representar l'escena 3D que l'usuari veu en el panell.

Aquest mètode s'encarregarà de "pintar" cada cop el taulell, eliminant tota l'escena anterior i representant-la de nou.

S'han creat quatre possibles estats per la representació del taulell:

Mode calcular torn

Mode mostrar sac



Mode partida

Mode partida normal

Es representarà el taulell en funció de l'estat que tingui la partida consultant "flags" al controladors Scrabble. Un cop l'estat sigui conegut la funció *display()* s'encarrega de cridar la funció indicada de representació del taulell.

La representació del taulell en tots els casos utilitza la mateixa estructura, primer es mirarà les interaccions de l'usuari amb el panell OpenGL (Posició del ratolí, botons clicats, condicions de la partida..), per després representar el taulell. En el cas que el panell mostri el sac s'han inhabilitat les opcions del usuari ja que la pantalla serà merament informativa.

5.2.1.3.1.1 FASE 1: INTERACCIONS

Existeixen dos possibles interaccions que pot tenir l'usuari: moure la càmera interaccions amb una casella/fitxa del taulell.

En el cas del moviment de càmera, es controla a través d'esdeveniments, cada cop que l'usuari roti la càmera a través del ratolí, es captarà l'esdeveniment i es modificaran els valors dels angles de visió (en el codi anomenats *gAngleCubX*, *gAngleCuby*), que el mètode *display* utilitzarà posteriorment.

L'altre cas seria la interacció de l'usuari amb una casella. Per obtenir la posició en la que es troba el ratolí, en cada una de les caselles del taulell se li assigna un identificador, que coincidirà amb el número de casella referent al taulell, d'aquesta manera es sap en tot moment on està el ratolí . Per tant les possibles interaccions les controlariem de les següents maneres:

En el cas que l'usuari tingui una fitxa seleccionada, cada cop que el ratolí canviï de casella la fitxa es dibuixaria en l'altra casella.

En el cas que l'usuari no tingui fitxa seleccionada es mostrarà un cursor.

Quan l'usuari cliqui una fitxa es controlarà amb esdeveniments de ratolí, bàsicament es comprova que, en la casella on s'ha clicat hi ha una fitxa, i si s'està en el mode partida i la fitxa es pot retirar, es retirarà del taulell i serà la fitxa seleccionada.

L'estructura del mètode *display* per controlar les interaccions és el següent (per qüestió de comprensió no incloïem tot el codi, només les parts més importants):

```
//Inicialitzem la fitxa seleccionada
    gNumFitxaSeleccionada = 0;
//Obtenim la fitxa que l'usuari tingui del seu faristol
if (pEstaJugant) gLletraSelect = gScrabble.getFitxaSeleccionadaAtril();
    //Inicialitzem COLOR_BUFFER i DEPTH_BUFFER
    gl.glClear(GL.GL_COLOR_BUFFER_BIT | GL.GL_DEPTH_BUFFER_BIT);
    //Entrem en el mode render de SELECCIÓ i fixem càmera
    gl.glRenderMode(GL.GL_SELECT);
    fixar(gl, this.getWidth(), this.getHeight(), true);
    gl.glMatrixMode(GL.GL_MODELVIEW);
    gl.glLoadIdentity();
    gl.glTranslatef(0.0f, 0.0f, -10.5f);
    gl.glRotatef(gAngleCubX, 1.0f, 0.0f, 0.0f);
    gl.glRotatef(gAngleCubY, 0.0f, 1.0f, 0.0f);
    gl.glTranslatef(-6.0f, 6.0f, 0.0f);
//Definim les interaccions depenen del estat que ens trobem
    if(gModeSac||gModeDuplicat) dibuixarSacSencer(drawable, true);
    else if (pJugat) drawTab(drawable, true);
    else if(!gFinalitzada) dibuixarSacTorn(drawable, true);
    //Obtenim el número de hits i passem al mode RENDER
gNumFitxaSeleccionada = gl.glRenderMode(GL.GL_RENDER);
//obtenim el identificador de la interacció (es a dir la casella / fitxa on
//es troba el ratolí)
    gHits = gBufferSeleccio.get(3);
```

Extret del mètode *display()* de *Partida.java*

5.2.1.3.1.2 FASE 2: REPRESENTACIÓ DEL TAULELL

Un cop es sap com s'ha de representar el taulell, procedim a dibuixar el panell, entrant en el mode RENDER_MODE, en el cas anterior on estudiàvem les interaccions del usuari, estàvem en el mode SELECT_MODE.

En aquest mode per dibuixar la pantalla utilitzarem les mateixes funcions que en el apartat anterior, un cop hi hagi tota l'escena dibuixada, dibuixarem el cursor, ja sigui en forma de fitxa o de "mousse".

```
//Col·loquem la càmera en la posició actual
    gl.glMatrixMode(GL.GL_MODELVIEW);
    gl.glLoadIdentity();
    gl.glTranslatef(0.0f, 0.0f, -10.5f);
    gl.glRotatef(gAngleCubX, 1.0f, 0.0f, 0.0f);
    gl.glRotatef(gAngleCubY, 0.0f, 1.0f, 0.0f);
    gl.glTranslatef(-6.0f, 6.0f, 0.0f);
//Definim la funció del dibuix depenen del estat que ens trobem
    //En aquest mode dibuixem el sac sencer amb les lletres visibles
    if(gModeSac)dibuixarSacSencer(drawable, false);
//Aquest mode es el "normal" on mostrem el taulell i la partida
    else if (pJugant) drawTab(drawable, false);
//Dibuixarem les lletres del sac cap per avall per a calcular el torn
    else if(!gFinalitzada)dibuixarSac(drawable, false);
```

Extret del mètode display() de Partida.java

```

//Si han existit interaccions
    if ((gNumFitxaSeleccionada > 0) && (gLletraSelect !=
null)) {
        Image vCursorImage =
Toolkit.getDefaultToolkit().getImage("");
        Cursor vBlankCursor =
Toolkit.getDefaultToolkit().createCustomCursor(vCursorImage,
new Point(0, 0), "");
        setCursor(vBlankCursor);
        //Obtindrem on està situat el ratolí
        int c = gHits / gColumnes;
            int r = gHits % gFiles;
        if (pJugat) {
//Si en la casella on està el ratolí hi ha una lletra, es
//dibuixarà la lletra seleccionada a sobre
            if(gScrabble.getLletraCasella(c, r)!=null)
                dibuixarCasellaSeleccionada(gl, c, r, gLletraSelect,
false);
            else
                dibuixarCasellaSeleccionada(gl, c, r, gLletraSelect, true);
        }
    } else {
        setCursor(new Cursor(Cursor.HAND_CURSOR));
    }

```

Extret del mètode display() de Partida.java

La funció per representar el taulell anirà casella per casella i dibuixarà les textures necessàries (en funció que sigui taulell o fitxa) en les mateixes posicions on hem situat els identificadors en l'etapa de selecció. Existiran 3 nivells de profunditat, un pel taulell, que sempre estarà en el fons de l'escena, i dos per les fitxes. Per saber quines textures s'han de carregar en cada moment anirem consultant a domini mitjançant el controlador *Scrabble* que informará de l'estat del taulell i/o les fitxes que hi ha en cada posició.



Totes les textures les agafarem del Hash que haurem inicialitzat en la funció *init()* i que contindrà les textures de les caselles del taulell, les possibles lletres de les fitxes, etc..

5.3 CAPA DE DADES

En aquesta capa bàsicament s'han afegit totes les ampliacions que s'han comentat en el disseny.

Destacar el canvi de codificació que s'ha implementat per facilitar l'escriptura i lectura dels fitxers sota Linux utilitzant les biblioteques de gestió de fitxers que proporciona Java.

Per escriure un fitxer l'obrirem de la següent manera:

```
BufferedWriter bWriter = new BufferedWriter(new  
OutputStreamWriter(new FileOutputStream(fileSave), "UTF-  
8"));
```

De la mateixa manera per llegir el fitxer serà:

```
BufferedReader bReader = new BufferedReader(new  
InputStreamReader(new FileInputStream(fileSave), "UTF-  
8"));
```

D'aquesta manera tindrem els fitxers d'una manera uniforme per funcionar sota Linux i sota Windows.



5.3.1 DIRECTORIS DE FITXERS

L'estructura de fitxers com s'ha comentat en l'etapa de disseny anterior serà idèntica a l'aplicació original, amb l'única diferència que en el directori dels fitxers afegirem la carpeta de "Diccionaris" on es trobaran les diferents categories del diccionari, i eliminarem el fitxer "diccionario" ja que les paraules es carregaran dels fitxers que es trobin en la carpeta "Diccionario".

Per tant l'estructura serà:

- Configuració
- Icones
- Idiomes
- Jugadors
- Saves
- Diccionaris

5.4 CAPA DE DOMINI

En la capa de domini s'ha mantingut tota la mateixa estructura del projecte original, únicament s'han afegit i/o modificat els atributs i funcions necessàries per permetre executar les ampliacions comentades en l'etapa de disseny.

Els petits canvis d'implementació que s'han modificats venen relacionats amb la càrrega de diccionaris, un nou mode de robot simulat per la màquina, i petits canvis en el tractament d'errors.

5.4.1 CANVIS EN DICCIONARI

La implementació del diccionari continuarà idèntica però hem afegit en cada Node del diccionari la categoria a la que pertany la paraula (és a dir el fitxer on es troba la paraula carregada).



5.4.2 CANVIS EN LA INTEL·LIGÈNCIA ARTIFICIAL DELS ROBOTS

La forma de trobar solucions dels robots simulats per la màquina s'han mantingut idèntics al projecte inicial, però hem afegit un robot amb una nova intel·ligència per les partides igualades, aquest robot descartarà totes les solucions que tinguin una puntuació superior a un valor donat (en aquest cas la mitjana de punts per tirada del jugador rival).

5.5 IMPLEMENTACIÓ DE CONTROL D'ERRORS NO CONTROLATS

En tota aplicació més o menys gran, poden sorgir sempre errors no controlats. Java té el tractament d'excepcions per a poder controlar errors no esperats. En l'aplicació es farà que cada cop que trobem una excepció, es mostra en un JDialog el missatge de l'error i per la sortida estàndard mostrarem tota la pila del error.

L'estructura en les excepcions per tant serà la següent:

```
catch (Exception e)
{
//Mostrem el missatge de la excepció al usuari
new Panel_Dialog(TITOL_PANELL,e.getMessage(), Panel_Dialog.ERROR);
//Mostrem en la sortida estàndard la pila del Error
e.printStackTrace();
}
```



5.6 PROVES DEL SISTEMA

Arribats en aquest punt ja tenim totes les funcionalitats implementades i s'ha cregut oportú realitzar uns determinats tests a l'aplicació.

Per poder realitzar les proves el més exhaustives possibles s'ha anat distribuint algunes versions del programa a diversos familiars i companys per que puguin jugar al joc, i a la vegada testejar l'aplicació possibilitant trobar possibles errors que el programador no havia contemplat.

6 CONCLUSIONS FINALS

6.1 APORTACIÓ PERSONAL

La realització d'un projecte d'aquesta magnitud m'ha servit per adonar-me de la importància que té fer una bona planificació i tenir ben identificades clarament totes les etapes d'un projecte.

Durant la carrera mai havia tingut ocasió de realitzar un programa en què hagués de definir personalment tota la planificació del projecte, i he observat que durant el transcurs del projecte poden sortir molts contratemps que si no es té una bona planificació poden fer variar els objectius indicats en el inici.

Per realitzar el projecte m'ha estat útil els coneixements adquirits en Java al llarg de la carrera i molt especialment m'ha anat bé l'experiència obtinguda amb les assignatures com PROP que obligaven a fer programes d'una certa complexitat.

Per les parts d'especificació i disseny he aplicat tots els coneixements adquirits en les assignatures d'especificació en llenguatge UML que m'ha permès identificar les primeres etapes per saber com plantejar el projecte.

En resum, opino que la tasca d'encarregar-se personalment d'un projecte complet d'aquesta importància és molt útil pel futur que ens depara, i m'ha permès aplicar la majora dels coneixements adquirits durant tota la carrera en un mateix projecte.



6.2 POSSIBLES MILLORES EN L'APLICACIÓ

A continuació mostrarem possibles millores que es poden aplicar en un futur a aquest projecte:

- Permetre partides on-line entre jugadors afegint un xat que permeti intercanviar informació entre els jugadors participants.
- Implementar tot el programa en un servidor i que els clients accedeixin a través d'una interfície web. D'aquesta manera s'evita haver d'entregar el programa físicament a tots els usuaris per jugar a l'aplicació.
- Canviar la implementació de càrrega del diccionari per reduir el temps que l'usuari ha d'esperar. En l'actualitat cada cop es carrega el diccionari llegint una a una totes les paraules d'un fitxer de text i es construeix internament el diccionari en forma d'arbre, això implica una gran càrrega de memòria i temporal. La millora que es proposa és permetre desar en un fitxer en format binari la pròpia classe diccionari que es construeix a partir de la lectura de totes les paraules, i implementar nous mètodes de lectura i escriptura.

7 BIBLIOGRAFIA

A continuació mostrem la bibliografia bàsica que s'ha consultat per la realització d'aquest projecte.

7.1 LLIBRES CONSULTATS

- Dolors Costal, M. Ribera Sancho, Ernest Teniente
Enginyeria del software especificació: especificació de sistemes orientats a objectes amb la notació UML.
Edicions UPC, 2000 ISBN: 84-8301-379-7
- Rafael Olivera
Un joc de Scrabble.
Edicions UPC, 2000 ISBN: 84-8301-379-7

7.2 REFERÈNCIES CONSULTADES

- Mattel Scrabble
<http://www.mattelscrabble.com>
- Vikipèdia
<http://ca.wikipedia.org/wiki/Portada>
- Terminologia
<http://www.pancarta.net/scrabble/terminologia.htm>
- Diccionari Alternatiu No Oficial de el Scrabble en català
<http://montane.cat/danoscl/>



- Club Scrabble Manresa

<http://www.clubscrabblemanresa.cat/>

- Pàgina personal d'en Joan Moratinos

<http://www.jmoratinos.com/cgi-bin/index.php>

- IEEE *Std.* 830-1993

http://standards.ieee.org/reading/ieee/std_public/description/se/830-1993_desc.html

- JOGL

<https://jogl.dev.java.net/>

- JOGL: A Beginner's Guide and Tutorial

<http://www.cs.umd.edu/~meesh/kmconroy/JOGLTutorial/>

- JAVA

<http://java.sun.com/javase/6/docs/api/>

- iSpell

<http://lasr.cs.ucla.edu/geoff/ispell-dictionaries.html>

- NetBeans

<http://www.netbeans.org/>

8 ANNEX 1. ESTUDI TEMPORAL I ECONÒMIC

8.1 PLANIFICACIÓ INICIAL

A continuació exposarem la planificació que s'ha anat seguint per a la realització del programa.

8.1.1 PLANIFICACIÓ DE TASQUES

Primer de tot definim totes les etapes a realitzar:

Definició del projecte: Entendre sense entrar en detall en què consisteix el programa que s'haurà de realitzar.

Comprensió projecte original: Lectura i estudi de l'aplicació original. En aquesta etapa identificarem totes les crides que es realitzen de domini cap a la capa de presentació original.

Anàlisi de requeriments: Definir totes les funcionalitats noves que s'hauran d'afegir en el projecte.

Especificació: Especificar totes les operacions que es duran a terme en l'aplicació un cop coneixem les funcionalitats noves del projecte.

Eliminació de la capa de presentació: En aquesta fase eliminarem la capa de presentació original i prepararem la substitució per les noves classes que implementarem.

Canvis del disseny original: Un cop tenim l'especificació de les operacions dissenyarem tots els canvis que haurem de realitzar en domini i dades per a que l'aplicació faci les operacions noves.

Disseny: Dissenyar la capa de presentació a partir de l'especificació mostrada i els canvis que s'han fet en capes inferiors.



Implementació: Realitzar tota la creació del codi seguint els passos que hem indicat en etapes anteriors.

Proves: Execució de tests per comprovar que l'aplicació funciona com està previst.

Memòria i presentació: Creació de tota la documentació del projecte.

8.1.2 PLANIFICACIÓ TEMPORAL

En tota la realització d'un projecte és necessari fer una planificació temporal on es marcarà la duració aproximada que invertirem en cada tasca. Per aquest projecte es va realitzar una planificació inicial determinada però van sorgir varis problemes que es comenten a continuació i van fer variar la planificació.

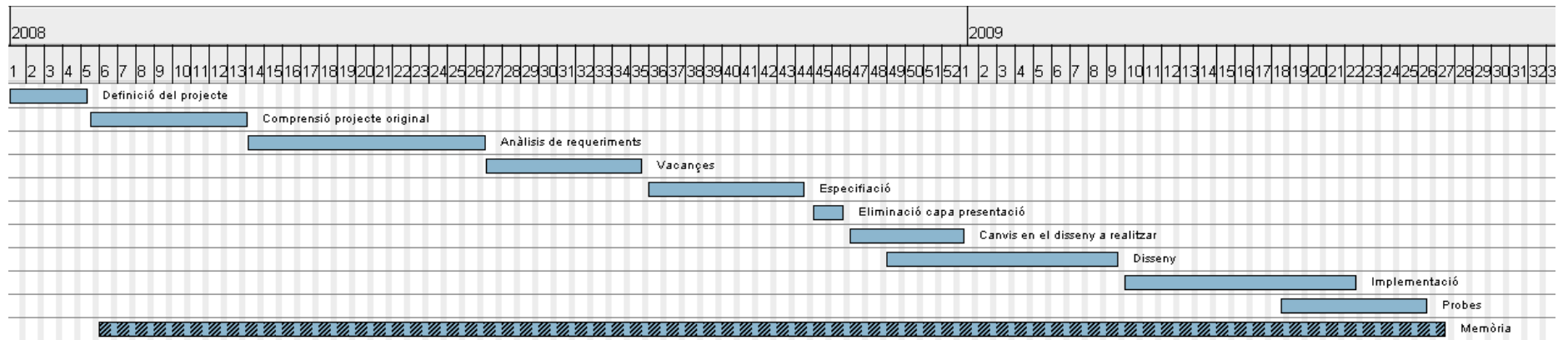
L'etapa de definició del projecte es va iniciar en gener de 2008 i l'idea era presentar-lo en 2 quadrimestres. La planificació inicial era intercalar vida laboral de mitja jornada amb la realització del projecte durant els primers sis mesos. La intenció era a partir del segon semestre del 2008 dedicar-me únicament al projecte, però per raons alienes a la universitat el juliol del 2008 vaig seguir amb la vida laboral de mitja jornada i el novembre la meva jornada laboral va passar a ser de jornada completa. Per aquesta raó a mitjans de l'any 2008 em vaig veure obligat a refer la planificació allargant el projecte un quadrimestre més, ja que per responsabilitats laborals m'era impossible la plena dedicació temporal al projecte.

La nova planificació es basava en tenir finalitzada tota l'especificació a finals del 2008 i alhora anar preparant el disseny que tindria l'aplicació. Durant el 2009 es començaria a realitzar la implementació del projecte. Aquestes dos etapes han estat possiblement les més costoses.

Per una altra banda durant tot el projecte s'ha anat documentant la memòria, documentant tots els avenços que s'anaven realitzant en el projecte.



8.1.3 DIAGRAMA GANTT



A partir del diagrama de Gantt extraurem les hores aproximades que s'ha estat en cada etapa del projecte per sumar el total d'hores utilitzat. La dedicació diària no ha estat sempre idèntica ja que com s'ha comentat anteriorment, per qüestions laborals no es podia assegurar la dedicació diària d'unes hores fixes.

Tasca	Hores invertides	Data inici	Data fi
Definició del projecte	40	1/01/08	31/01/08
Comprensió projecte inicial	40	1/02/08	1/04/08
Anàlisi requeriments	40	1/04/08	1/07/08
Vacances		1/07/08	31/08/08
Especificació	40	1/09/08	31/10/08
Eliminació capa presentació	10	3/11/08	15/11/08
Canvis en el disseny realitzat	20	17/11/08	31/12/08
Disseny	120	1/12/08	28/02/09
Implementació	180	2/03/09	30/05/09
Proves sistema	40	1/05/09	26/06/09
Memòria	140	4/02/08	3/07/09
TOTAL	670		

8.1.4 ESTUDI ECONÒMIC

Realitzar aquest estudi no és trivial ja que el procediment habitual d'un projecte d'una mínima magnitud requereix diversos treballadors, i en el nostre cas ha estat una única persona la que ha coordinat i realitzat totes les funcions. Per facilitar un càlcul aproximat del cost econòmic del projecte desglossarem tota la feina realitzada en diversos perfils de possibles treballadors d'una empresa.

Suposarem que han treballat els següents perfils per a dur a terme aquest projecte:

Cap de projecte: S'encarregaria de comprovar que es compleixen tots els requisits funcionals que s'han exigit, i controlaria periòdicament que l'aplicació avanci de manera correcta .

Analista: Seria l'encarregat de fer l'etapa d'especificació i disseny del sistema.

Programador gràfic: S'encarregaria de la implementació de tota la capa de presentació que l'analista haurà dissenyat.

Programador junior: Un programador junior podria encarregar-se d'actualitzar i ampliar les capes de domini i dades que s'han decidit canviar.

El *testing* de les proves aniria a càrrec del cap de projecte i els programadors. La documentació del projecte principalment la realitzaria l'analista, ja que ha estat l'encarregat de plasmar el disseny de l'aplicació tot i que els programadors participarien en algunes parts.

PERFIL	HORES	PREU (€/hora)	TOTAL
Cap de projecte	120	60 €	7200 €
Analista	300	45 €	13500 €
Programador gràfic	170	40 €	6800 €
Programador junior	80	30 €	2400 €
TOTAL	670		29900 €

El preu total de mà d'obra del projecte seria aproximadament uns 29900 €.

En aquest càlcul no comptabilitzarem el cost del hardware. Suposarem que no s'ha comprat un ordinador només per aquest projecte i que el cost del hardware està ja amortitzat.

El software utilitzat és gratuït a excepció del sistema operatiu utilitzat, el Windows Vista, i el Microsoft Office, utilitzat per documentar aquesta memòria. La resta de software emprat com NetBeans, Java i el Sistema Operatiu Ubuntu són totalment gratuïts.



8.1.4.1 PREU TOTAL

PRODUCTE	PREU
Windows Vista Business	479 €
Microsoft Office Vista	100 €
Mà d'obra	29900 €
TOTAL	30479 €

9 ANNEX 2. MANUAL D'INSTAL·LACIÓ

9.1 INTRODUCCIÓ

En aquest manual s'explicarà com executar el programa en Windows Vista o Linux.

Primer de tot l'usuari haurà de copiar la carpeta “*Scrabble*” del directori arrel del CD proporcionat en el directori que es desitgi.

9.2 JAVA DEVELOPMENT KIT

La versió JDK utilitzada en el projecte serà la versió JDK 1.6.0_10. En el directori “*Scrabble*” es troba les versions JDK de Windows i Linux de 32 bits. En aquestes versions ja van instal·lades les biblioteques de Java OpenGL que utilitza el programa.

Si l'usuari decideix utilitzar una altre versió de JDK haurà de mirar que sigui compatible amb la versió JDK 1.6.0_10 i instal·lar les biblioteques OPENGL que expliquem en el següent apartat.

Nota important: En cas que l'usuari decideixi utilitzar el JDK proporcionat en el CD no haurà de realitzar cap de les accions comentades en el apartat 9.3 !



9.3 INSTAL·LACIÓ BIBLIOTEQUES OPENGL

Si l'usuari decideix utilitzar un JDK propi haurà d'instal·lar les biblioteques OpenGL proporcionades en el CD. En funció del sistema operatiu utilitzat les accions a realitzar varien.

9.3.1 WINDOWS

En la instal·lació de la Java Virtual Machine s'haurà d'incorporar les biblioteques encarregades d'utilitzar Java OPENGL localitzades en el directori del CD "Unitat CD:\Software\JOGL\Windows". Tots els fitxers amb extensió *.dll* de la carpeta hauran de ser copiats en el directori *bin* del JRE utilitzat per Windows.

Suposant que el JDK es trobi instal·lat a "C:\Archivos de programa\Java\jdk1.6.0_10\", les biblioteques es copiaran a la carpeta "C:\Archivos de Programa\Java\jdk1.6.0_10\jre\bin".

El pas següent serà copiar el fitxer **JOGL.jar** localitzat en el directori del CD "Unitat CD:\Software\JOGL\" al directori *lib/ext* de la JVM utilitzada.

Suposant que el JDK està instal·lat en el mateix directori que en l'exemple anterior, el fitxer s'haurà de copiar a "C:\Archivos de Programa\Java\jdk1.6.0_10\jre\lib\ext".



9.3.2 LINUX

Per executar l'aplicació en Linux s'hauran d'instal·lar les biblioteques localitzades en la carpeta del CD "Unitat CD:\Software\JOGL\Linux" en el directori on es trobin les biblioteques amb extensió .so de la Java Virtual Machine utilitzada.

En el cas que s'utilitzi un JDK sota una arquitectura de 32 bits la carpeta on copiar les biblioteques seria a "[Carpeta_instal·lacio_JDK]\jre\lib\i386".

El pas següent serà copiar el fitxer **JOGL.jar** localitzat en el directori del CD "Unitat CD:\Software\JOGL\" al directori *lib/ext* de la JDK utilitzada:
[Carpeta_instal·lacio_JDK]\jre\lib\ext.

9.4 EXECUCIÓ DEL PROGRAMA

Si l'usuari està utilitzant el JDK proporcionat en el CD, únicament haurà d'anar a la carpeta "Scrable" que prèviament ha copiat en el seu disc dur, i executar el fitxer "*ScrableLinux.bat*" o "*ScrableWindows.bat*" en funció del sistema operatiu utilitzat.

Si en canvi l'usuari ha realitzat manualment la instal·lació de les biblioteques JOGL i vol utilitzar un JDK propi, per iniciar l'aplicació haurà d'executar de la carpeta "Scrable" el fitxer "*ScrableLinuxNotJDK.bat*" o "*ScrableWindowsNotJDK.bat*" en funció del sistema operatiu instal·lat.



10 ANNEX 2. MANUAL D'USUARI

10.1 INTRODUCCIÓ

El propòsit d'aquest manual és guiar als usuaris de l'aplicació realitzada en el projecte final de carrera "Interfície d'un joc de Scrabble".

A continuació es detallen totes les pantalles i operacions que s'hi poden dur a terme. Per facilitar a l'usuari la cerca a través del manual dels dubtes que li puguin aparèixer durant l'execució del joc, el manual s'ha estructurat de tal manera que cada secció sigui una opció de l'aplicació.

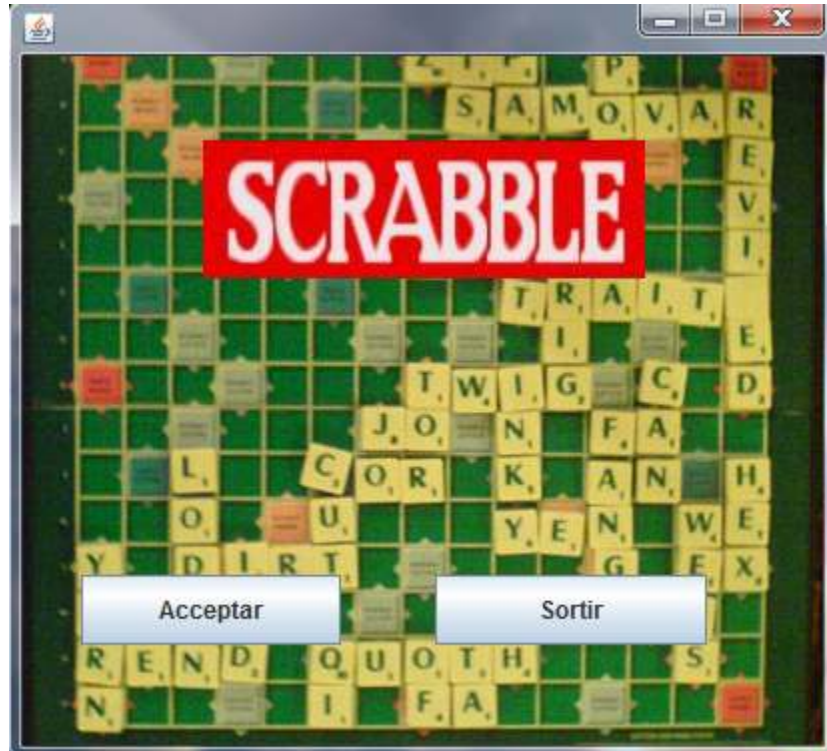
10.2 INICI PROGRAMA

Per iniciar l'aplicació haurem d'executar un dels quatre executables que es proporciona en el CD del joc en funció del Sistema Operatiu que s'utilitzi (Windows o Linux) i de la versió JDK que s'utilitza.

En el directori on es trobi el joc instal·lat l'usuari podrà consultar el fitxer "LLEGEIX-ME.txt" que explica els detalls de configuració del joc, i en la secció de la memòria "*Manual d'instal·lació*" l'usuari podrà informar-se com instal·lar l'aplicació.

10.3 PANTALLA PRINCIPAL

El primer cop que s'executi l'aplicació es mostra la pantalla inicial.



Pantalla inicial

Si l'usuari clica el botó "Acceptar" es mostra la pantalla principal, en cas contrari clicant "Sortir", l'aplicació finalitza.



Pantalla principal

Aquesta pantalla està dividida en diverses seccions, des d'aquest punt l'usuari pot navegar a totes les opcions de l'aplicació. En funció de les seves necessitats les accions a realitzar poden ser:

- Gestionar jugadors
- Gestionar diccionaris
- Gestionar rànkig
- Modificar regles
- Carregar partida
- Canviar idioma
- Jugar al Scrabble

10.4 GESTIÓ DE JUGADORS

En la pantalla de gestió de jugadors es permet crear, modificar o eliminar els perfils de l'aplicació.

A la part esquerra hi ha un llistat amb tots els jugadors disponibles que hi ha en la memòria del joc.

En l'aplicació existeixen dos tipus bàsics de jugador:

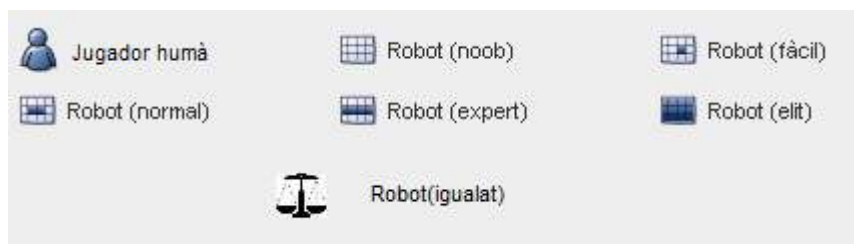
- Robot
- Humà

Un jugador humà és la representació d'un jugador real en el joc.

Els robots són usuaris simulats per l'aplicació per jugar partides de Scrabble, existeixen 6 tipus de robots:

- Robot igualat: És el robot que participa en les partides igualades³, les puntuacions en cada tirada d'aquest robot estan limitades a la mitjana de punts per jugada del jugador rival.
- 5 Robots de nivell: Existeixen 5 nivells de dificultat per jugar (noob, fàcil, normal, expert, elit).

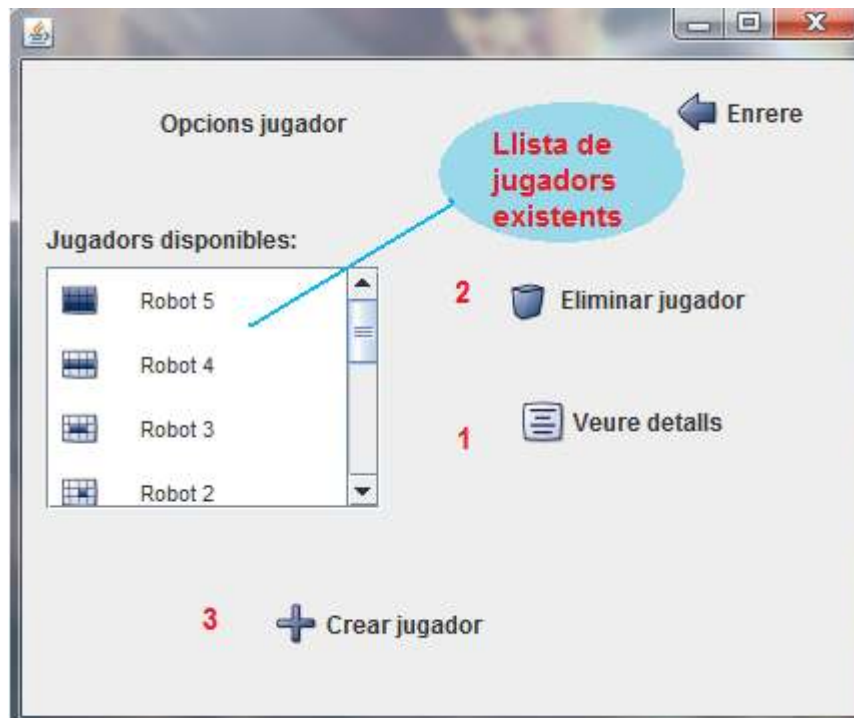
Les icones que identifiquen cada jugador són les següents:



Per cada jugador de la llista es permet:

- 1. Consultar les estadístiques
- 2. Eliminar el jugador definitivament de l'aplicació
- 3. Crear un jugador nou

³ Per a més informació sobre les partides igualades, consultar la secció de "Preparant la partida".




Pantalla de gestió dels jugadors

Si es té la intenció d'anar a les estadístiques del jugador o eliminar un jugador és necessari tenir el jugador desitjat seleccionat en la llista. L'usuari pot tornar a la pantalla principal en qualsevol moment del cicle de vida d'aquesta pantalla.

10.4.1 CONSULTAR ESTADÍSTIQUES JUGADOR

En la secció d'estadístiques es mostren totes les dades del jugador seleccionat. Clicant el botó "enrere" l'usuari torna a la pantalla de gestió de jugadors.

ESTADÍSTIQUES		
JUGADOR	xavi	
Scrabbles:	28	
Tipus jugador:	Huma	
Puntuació total:	20147	
Paraules totals:	852	
MILLOR PARTIDA	MITJANES	MILLOR JUGADA
Puntuació: 530	Nº "0" per partida: 1	Puntuació: 162
Paraules: 23	Punts per partida: 43	Paraules: 2
	Punts per tirada: 20	
MILLOR PARAULA	TOTALS	
Puntuació: 162	Partides guanyades: 5	
Paraula: EMOL-LIENT	Partides perdudes: 11	
	Partides empatades: 0	
 Enrere		

Pantalla de les estadístiques del jugador

10.4.2 ELIMINAR JUGADOR

Si l'usuari confirma la selecció, s'eliminen tots les dades relacionades amb el jugador seleccionat en el llistat d'usuaris.

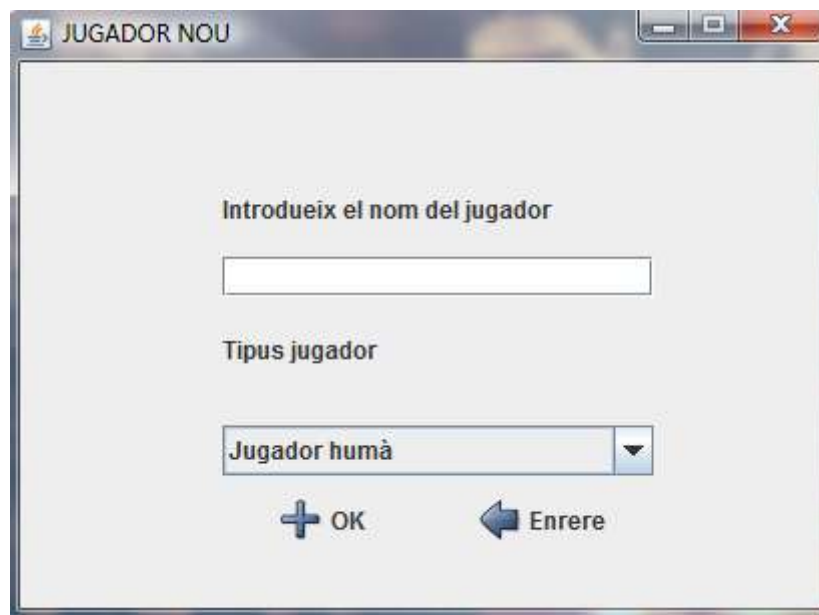
10.4.3 CREAR UN JUGADOR NOU

Aquesta pantalla demana les dades necessàries per a crear un jugador nou: el nom del jugador, i el tipus de jugador que es vol crear.

Els tipus de jugador possibles són:

- Jugador humà
- Robot de nivell (5 nivells de dificultat)
- Robot igualat

Només pot existir un jugador del tipus "Robot igualat" en el joc, per tant aquesta pantalla no permet crear un robot d'aquest tipus si ja existeix un, a no ser que s'elimini prèviament.



The screenshot shows a window titled "JUGADOR NOU". Inside, there is a label "Introdueix el nom del jugador" above a text input field. Below that is a label "Tipus jugador" above a dropdown menu showing "Jugador humà". At the bottom, there are two buttons: "+ OK" and "← Enrere".

Pantalla per crear un jugador



10.5 GESTIÓ DICCIONARIS

Aquesta pantalla permet configurar els diccionaris de l'aplicació. Abans de comentar les funcionalitats d'aquesta pantalla definirem els conceptes següents:

- Diccionari
- Categoria de la paraula

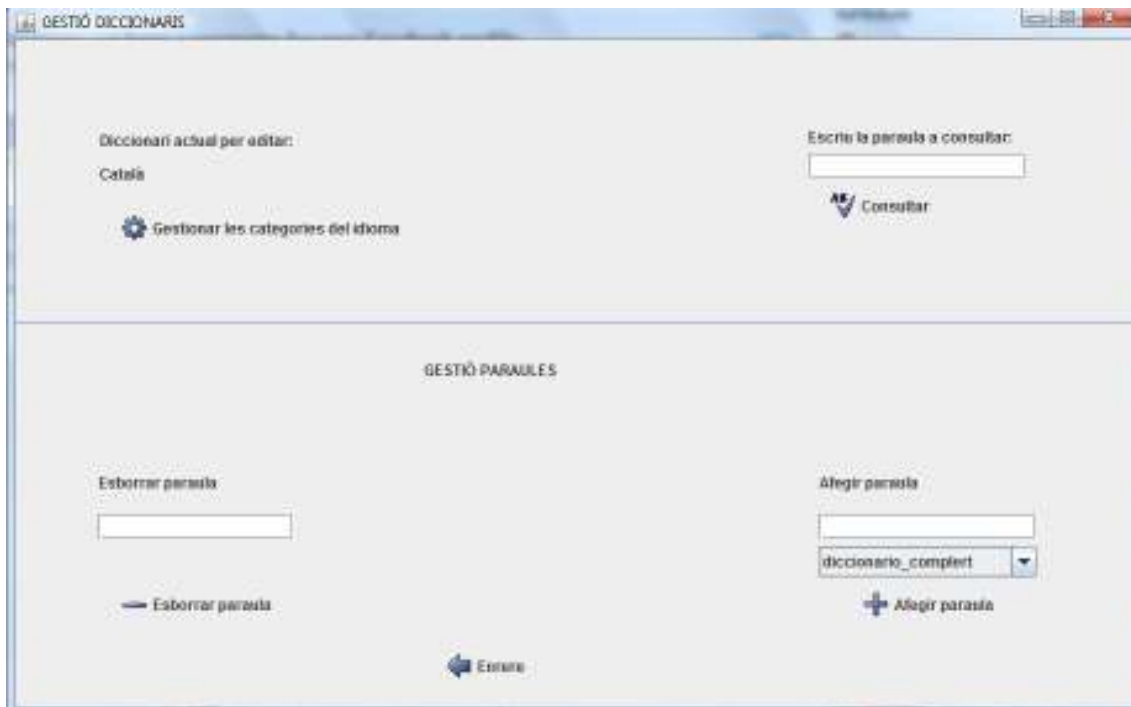
Un diccionari és el conjunt de paraules vàlides que es permeten introduir durant el transcurs d'una partida. Tot diccionari s'ha carregat prèviament d'una sèrie de fitxers localitzats en una determinada carpeta de l'aplicació⁴.

Es permet a l'usuari gestionar els fitxers del diccionari al seu gust i tenir diversos fitxers organitzats per categories (infinitiu, noms propis, diminutius, ...), per tant la categoria de la paraula és el nom del fitxer en el que es troba la paraula.

En aquesta pantalla l'usuari pot dur a terme les accions següents:

- 1. Consultar paraula
- 2. Esborrar paraula
- 3. Afegir paraula
- 4. Gestionar les categories de paraules

⁴ En el final del manual es mostren les carpetes que l'usuari pot modificar.



Pantalla de la Gestió dels diccionaris

10.5.1 CONSULTAR PARAULA

Introduint la paraula que es vol consultar, l'aplicació informa si existeix la paraula i mostra la categoria a la què pertany.

10.5.2 ESBORRAR PARAULA

Per eliminar una paraula del diccionari actual, l'usuari haurà d'introduir la paraula i tot seguit s'informa a l'usuari si la paraula ha estat correctament eliminada.

10.5.3 AFEGIR PARAULA

Com en el cas anterior, s'ha d'indicar la paraula que es vol afegir, i la categoria a la que es desarà. Si la paraula ja existeix l'aplicació informarà de l'error.

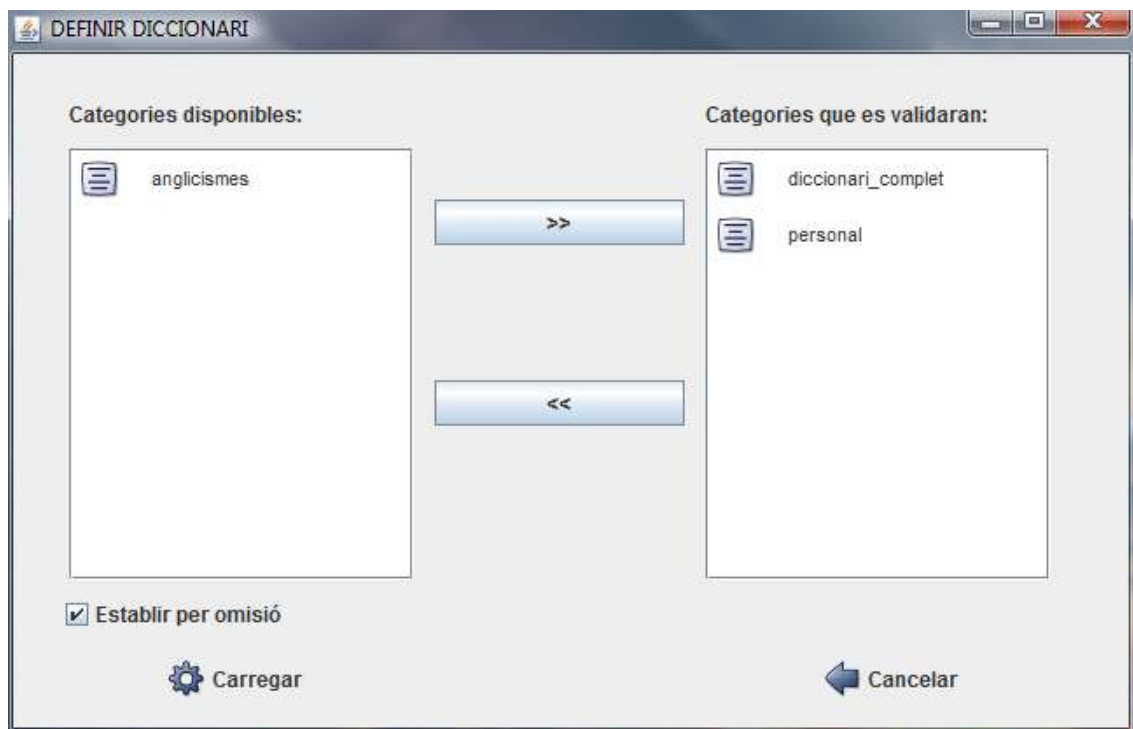
10.5.4 GESTIONAR CATEGORIES DE PARAULES

En aquesta secció es permet seleccionar quins llistats de paraules es volen afegir en el idioma que hi ha actualment configurat, més endavant es comenta com afegir llistats personalitzats de paraules.

La pantalla consisteix en dos llistats, el primer conté tots els fitxers de paraules vàlids existents que es permeten carregar, i el segon és el que l'aplicació carrega un cop es confirmi la selecció. Usant els botons centrals i seleccionant els diccionaris que es desitgin, l'usuari ha d'escollir la configuració del diccionari per la partida.

Opcionalment, es permet marcar la selecció per omisió, d'aquesta manera cada cop que es seleccioni el idioma actual es carregaran les paraules que hem seleccionat.


Si l'usuari clica en el botó "Cancel·lar" no es carregarà la selecció i es tornarà a la pantalla de gestió de diccionaris.



Pantalla per carregar llistats de paraules

10.6 GESTIÓ DE RÀNKING

En aquesta pantalla mostrem una classificació dels usuaris existents de l'aplicació segons el criteri d'ordenació que l'usuari hagi escollit.



Rànkinq

Partides guanyades

	%	G	J	JUGADOR
1	71	5	7	Robot 1
2	62	5	8	xavi1
3	33	1	3	altre1
4	31	5	16	xavi
5	0	0	1	Robot 2
6	0	0	1	altre
7	-	0	0	RobotHuman
8	-	0	0	Robot 3
9	-	0	0	Robot 4
10	-	0	0	Robot 5

Criteris d'ordenació

← Enrere

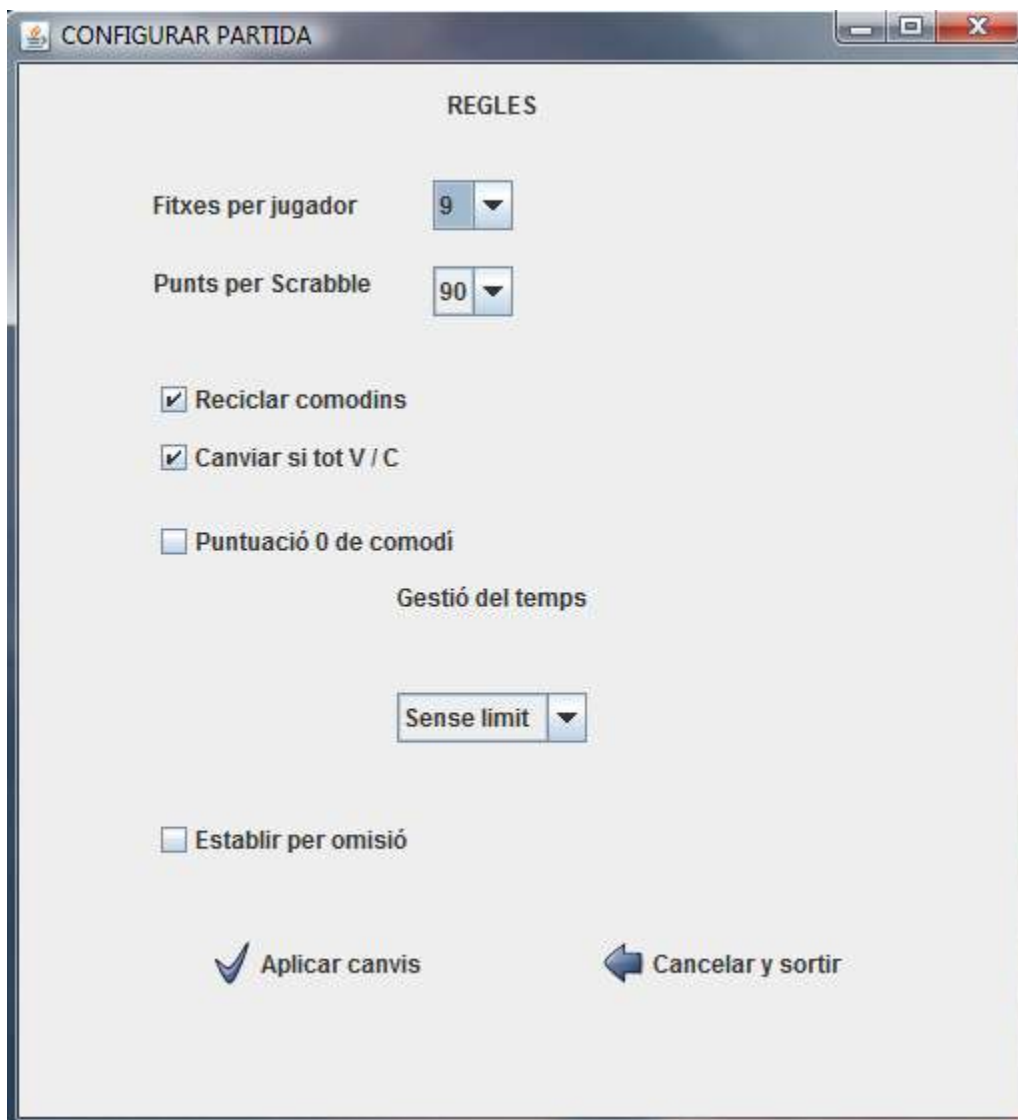
Pantalla rànkinq

10.7 MODIFICAR REGLES

Pantalla per configurar les regles que s'apliquen durant la partida, opcionalment també es permet escollir la configuració seleccionada per omissió, d'aquesta manera cada cop que s'iniciï l'aplicació es carregaran les regles escollides.

Les regles que es permeten configurar són les següents:

- *Fitxes per jugador:* Les fitxes que cada jugador té en el faristol per poder fer la jugada, es permet escollir de cinc a nou fitxes.
- *Punts per Scrabble:* Si en una jugada determinada el jugador ha posat totes les seves fitxes en el taulell, s'aconsegueix un 'Scrabble'; aquesta opció permet configurar la puntuació extra en el torn si s'efectua aquesta jugada.
- *Reciclar comodins:* Es permet intercanviar el comodí per la fitxa que representa si el jugador que té el torn té la fitxa en el seu faristol.
- *Canviar si tot V/C:* L'usuari del torn actual se li permet canviar totes les fitxes si aquesta opció està seleccionada i totes les seves fitxes són vocals o consonants.
- *Puntuació 0 comodí:* Si l'opció no està marcada el comodí puntuarà per la puntuació de la lletra que representa.
- *Gestió temps:* Es permet configurar el temps de la partida en funció de 3 variables
 - o *Sense límit:* El torn del jugador no està limitat.
 - o *Per torn:* El temps es comptabilitza en cada torn, i s'inicialitza al començar cada torn de la partida.
 - o *Per jugador:* El temps es comptabilitza durant tota la partida. Al començar el torn continuarà del temps que li restava al finalitzar la jugada anterior.



Pantalla gestió de regles

10.7.1 CARACTERÍSTIQUES DE LA GESTIÓ DEL TEMPS

En funció de la configuració que hi hagi activada la comptabilització de la jugada varia:

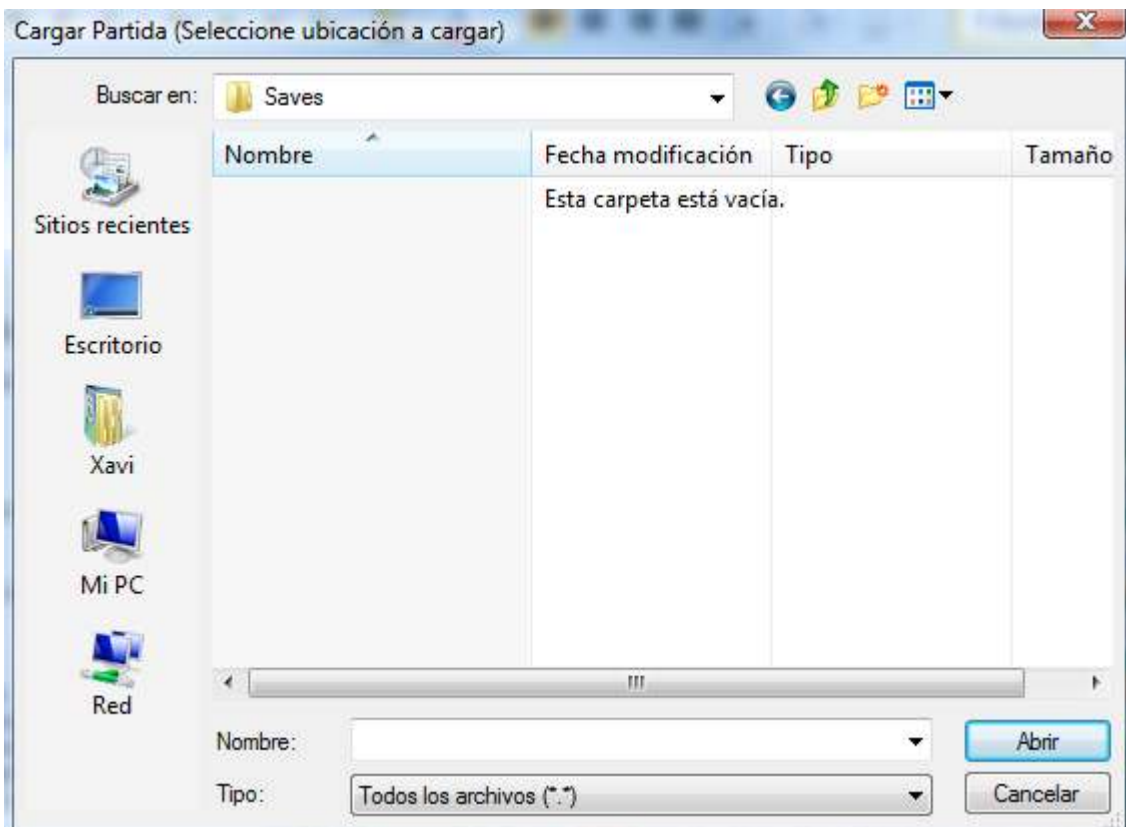
- Si es juga en el mode “temps per torn”, l’usuari té n minuts per validar una jugada, si es valida fora de temps es resta de la puntuació final un punt per cada cinc segons que passin.
- Si es juga en “temps per jugador”, l’usuari ha d’acabar la partida amb el temps indicat, si esgota el temps, com en el cas anterior, es resta

de la puntuació de cada jugada un punt per cada cinc segons passats.

10.8 CARREGAR PARTIDA

En aquesta opció es mostra un navegador de fitxers per que l'usuari esculli un fitxer d'una partida desitjada.

L'aplicació carregarà la partida un cop hagi llegit les dades del fitxer indicat.



Pantalla de selecció de partida

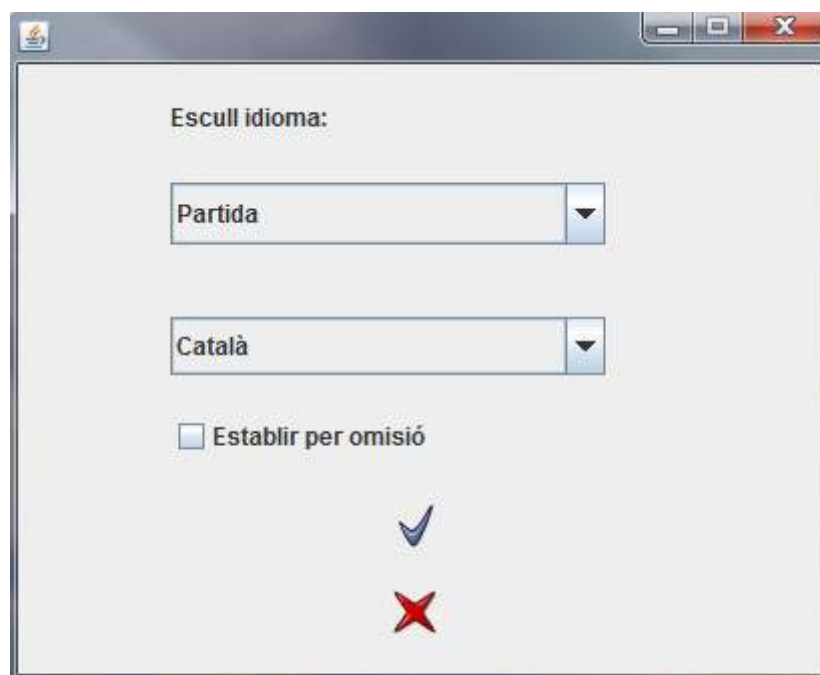
10.9 CANVIAR D'IDIOMA

La següent pantalla permet configurar els idiomes de l'aplicació en funció de dos valors:

- De l'aplicació
- De la partida.

El idioma de l'aplicació es refereix als textos que l'usuari llegeix en les pantalles. La segona opció carrega el diccionari per la partida en el idioma seleccionat.

Com s'ha vist anteriorment en altres pantalles, es permet establir la selecció per omissió.



Pantalla de selecció d'idioma

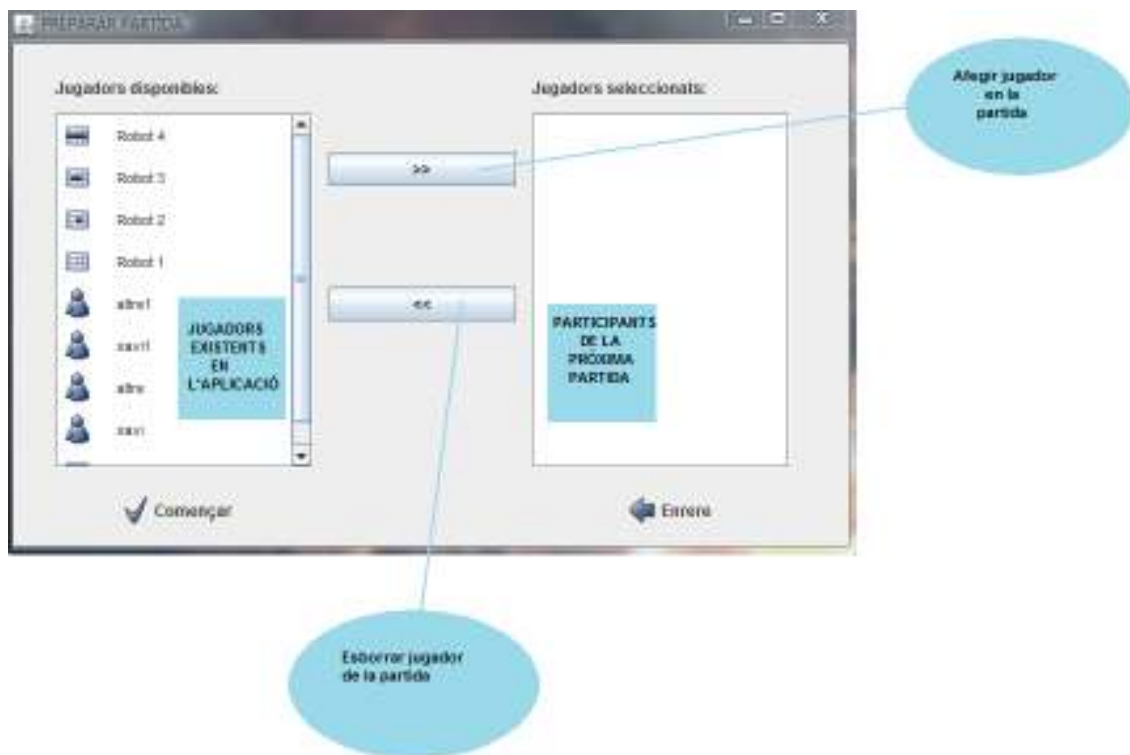
Nota: Si l'usuari ha escollit canviar el idioma del diccionari, és possible que l'aplicació es bloquegi uns segons per carregar en memòria el diccionari escollit.

10.10 JUGAR AL SCRABBLE

Un cop l'usuari cliqui sobre el botó "Scrabble" per començar una partida, el primer pas és introduir els jugadors que participaran.

10.10.1 PREPARANT LA PARTIDA

En aquest punt l'usuari ha d'introduir els jugadors que participaran i el tipus de partida que vol jugar.



Pantalla configuració jugadors

Inicialment es mostren dos llistats, en el primer llistat es troben tots els jugadors disponibles de l'aplicació. L'usuari utilitzant els botons centrals ha d'introduir els participants que jugaran en la partida (sempre un màxim de 4 jugadors).

Clicant en el botó "Començar" la partida carregarà el taulell, en cas contrari, si es clica "enrere" es torna a la vista principal.



És possible jugar 3 tipus de partida:

- Partida normal
- Partida duplicada
- Partida igualada

S'habilitaran dos opcions de joc extremes al seleccionar un jugador humà:

Jugar partida emulada
 Jugar partida igualada

10.10.1.1 JUGAR UNA PARTIDA IGUALADA

Per jugar una partida igualada l'usuari ha de seleccionar el jugador que hi participarà i posteriorment marcar el checkBox "jugar partida igualada". En una partida igualada només pot participar un jugador de tipus humà.

Aquest mode enfronta un jugador humà amb un jugador simulat per la màquina. La intel·ligència d'aquest robot no realitzarà jugades superiors a la mitjana del rival. D'aquesta manera s'afegeix un punt de competitivitat i es forçarà al jugador a superar les seves mitjanes per a guanyar al rival. Quantes més partides normals jugui un jugador determinat més reals seran les partides igualades.

10.10.1.2 JUGAR UNA PARTIDA DUPLICADA

Per jugar una partida duplicada, de la mateixa manera que en una igualada, s'ha de seleccionar un jugador humà i marcar el checkBox "jugar partida duplicada". Entenem per partida duplicada aquella en la qual tots els jugadors participen amb el mateix faristol i el mateix taulell. En aquest mode tots els jugadors competeixen en absoluta igualtat de condicions donat que s'elimina el possible desavantatge de tenir pitjor combinació de lletres, en les

instruccions del joc es detalla més àmpliament el funcionament de partides duplicades.

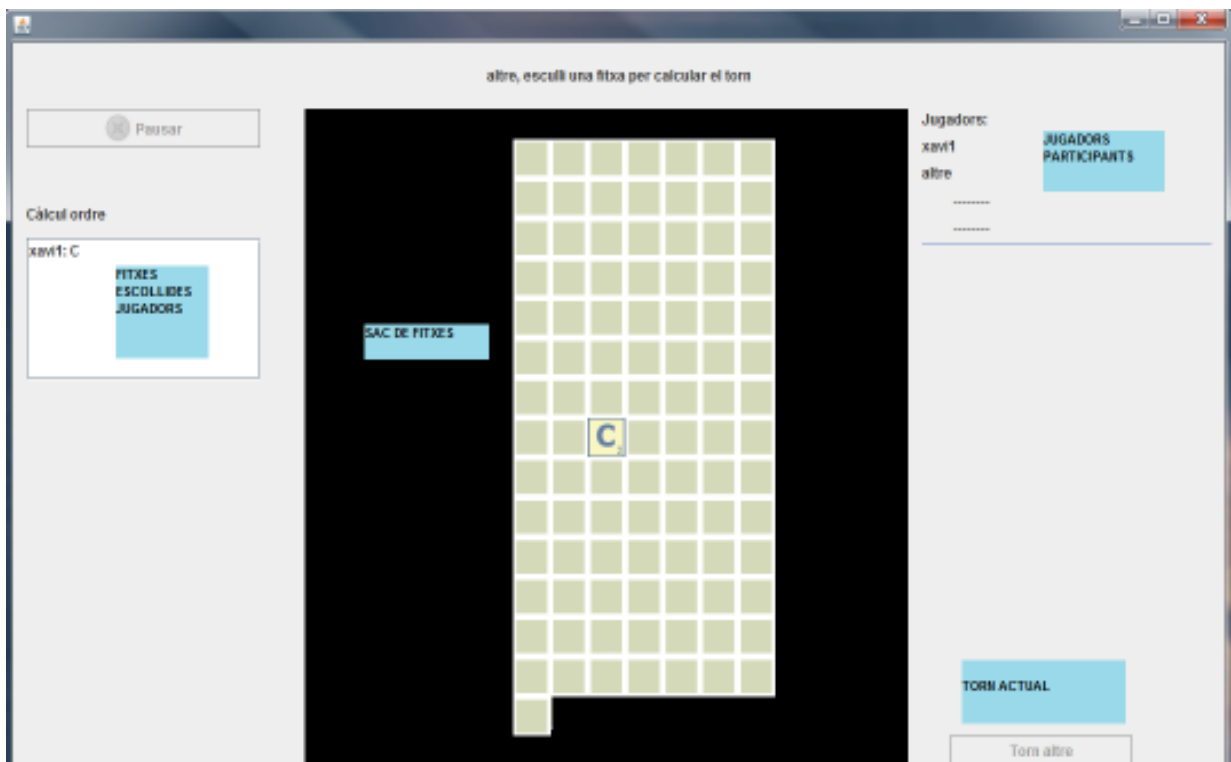
10.10.1.3 JUGAR UNA PARTIDA NORMAL

Si no es selecciona cap checkBox, la partida començarà amb les instruccions habituals d'una partida de Scrabble i els jugadors del llistat de la dreta com a participants.

10.10.2 CALCULANT EL TORN DE LA PARTIDA

Si en la partida participen dos o més jugadors, es carrega primer de tot la pantalla de càlcul del torn per definir l'ordre de tirada.

En aquesta pantalla hi ha totes les fitxes cap per avall i cada usuari en el ordre que indiqui l'aplicació ha d'escollir una fitxa clicant sobre ella a través de la següent pantalla:



Pantalla de càlcul de torn



Si un jugador li surt una fitxa ja escollida prèviament haurà de tornar a agafar fitxa. Un cop tots els jugadors escullin les fitxes l'aplicació carregarà la pantalla principal de joc. El torn es calcula en funció del ordre alfabètic de les fitxes escollides per cada jugador.

10.10.3 PANELL PARTIDA

Un cop el torn estigui definit es carreguen totes les opcions per començar la partida.

En el panell es mostren diverses opcions en funció de l'estat en què es troba la partida :

- Partida no pausada durant un torn
- Partida amb jugada validada
- Partida duplicada esperant el torn
- Partida pausada
- Partida finalitzada

10.10.3.1 PARTIDA NO PAUSADA DURANT UN TORN

Aquest mode es mostra quan un jugador no robot té el torn de jugada. En aquest panell es duen a terme totes les accions bàsiques del torn com moure les fitxes del faristol, aturar les partides, etc.



Pantalla de partida

El panell de la partida es pot dividir en 4 parts:

- Part central (taulell)
- Part superior (missatges d'aplicació)
- Part inferior (faristol)
- Parts laterals (opcions i estadístiques)

En les pròximes seccions comentem les accions possibles que es poden realitzar en cada secció del panell.



10.10.3.1.1 PART SUPERIOR

En aquesta part del panell es mostren els missatges que l'aplicació necessita mostrar a l'usuari.

10.10.3.1.2 PART LATERAL

En els laterals del panell es mostren estadístiques i accions que l'usuari pot executar.

A la dreta del panell es mostra la classificació temporal de la partida, i les següents estadístiques del jugador que té el torn:

- Puntuació de la partida.
- Scrabbles acumulats.
- Paraules formades.
- Dades de la millor jugada i de la millor paraula.

Sota les dades del jugador es poden mostrar en funció de les condicions de la partida les opcions següents:

- *Ajuda torn:* Es mostra una jugada sense validar amb les fitxes que es té actualment en el faristol.
- *Passar torn:* L'usuari que té el torn actual no efectua cap jugada i puntua zero punts.
- *Remoure fitxes:* Es remouen aleatòriament les fitxes del faristol.
- *Canviar comodí:* Si en el taulell hi ha un comodí jugat i l'usuari té en el faristol la fitxa que representa el comodí aquest s'intercanvia per la fitxa del faristol. Aquesta opció només s'habilita si les regles actuals així ho permeten.



A la part esquerra del panell es veurà inicialment el botó d'aturar la partida, les paraules invalides jugades (en el cas que l'usuari validi una jugada amb paraules invalides), i les opcions següents:

- *Desfer*: Elimina qualsevol fitxa jugada en el torn actual del taulell.
- *Canviar fixes*: Es passa al mode "canviar fitxes" explicat més endavant.
- *Canviar V/C*: Si les regles del joc així ho permeten, aquest botó permet canviar totes les fitxes del faristol del jugador actual, però només en el cas que el faristol del jugador siguin tot vocals o consonants (la jugada és coneguda com "fer senyors").

Cal comentar, que les jugades de canviar el comodí i de "fer senyors" només s'habiliten quan així ho permeti les condicions de la partida i les regles del joc.

10.10.3.1.3 FARISTOL

La part central d'aquesta secció esta reservada per mostrar el faristol del jugador humà actual (en cas d'un jugador 'robot', no es mostren les fitxes).

A la part esquerra es mostra el torn del jugador i el temps restant (si s'està jugant amb el temps activat). A la dreta es mostra un botó per centrar la càmera del taulell, i un altre per validar la jugada actual.

Per escollir una fitxa del faristol només s'ha de clicar amb el botó esquerre sobre la fitxa escollida, si es clica sobre una altra fitxa es canvia la fitxa seleccionada automàticament. Per col·locar una fitxa en el taulell, s'ha d'haver seleccionat prèviament .

Si es vol intercanviar dos fitxes de posició del faristol, l'usuari ha de seleccionar la primera fitxa amb el mètode comentat anteriorment per seguidament clicar amb el botó dret l'altre fitxa que vol intercanviar.

El botó validar l'usarem per validar la jugada creada en el torn actual.

Es passa a l'estat de "jugada validada" si la jugada és correcta, en canvi si s'ha introduït alguna jugada incorrecte el torn seguirà en el jugador actual, i l'aplicació mostrarà les paraules invalides en la part lateral esquerre del panell.



Pantalla de paraula invàlida

10.10.3.1.4 TAULELL

Aquí es mostra el taulell de la partida actual. Si es té una fitxa seleccionada (a través del faristol o del propi taulell) el ratolí es mostra en forma de la fitxa seleccionada, en cas contrari el cursor té la imatge de cursor.

En el taulell es poden dur a terme les següents accions:

- *Moure càmera:* Movent el ratolí amb el botó dret clicat es rotarà el taulell en la direcció escollida.
- *Seleccionar fitxa taulell:* Per agafar una fitxa del taulell situada en el torn actual l'usuari únicament ha de clicar en la fitxa desitjada sense tenir cap fitxa seleccionada.



- *Desseleccionar fitxa taulell:* Clicant el botó central del ratolí amb una fitxa seleccionada i el cursor sobre el taulell, aquesta es retorna al faristol.
- *Col·locar fitxa seleccionada:* Al clicar amb el botó esquerre sobre una casella lliure del taulell tenint una fitxa seleccionada aquesta es situa en la casella indicada.

10.10.3.1.5 ACCIONS POSSIBLES

Per avançar al següent torn l'usuari pot efectuar les següents accions :

- Generar una jugada
- Passar el torn
- Canviar fitxes

Un cop finalitzada qualsevol de les tres accions la partida avançarà a les puntuacions del torn actual.

10.10.3.1.5.1 GENERAR JUGADA

Un cop l'usuari hagi introduït la paraula que desitja haurà de validar la jugada clicant el botó validar.



Botó validar

Si en la jugada s'ha introduït algun comodí l'aplicació demana el valor del comodí marcat en color groc.



Validació comodí

Si la jugada no és vàlida, es mostren les paraules invalides.

10.10.3.1.5.2 PASSAR TORN

Clicant aquest botó l'usuari passa el torn sense puntuar.

10.10.3.1.5.3 CANVIAR FITXES

L'usuari ha clicat prèviament el botó “Canviar fitxes” per arribar en aquest estat. En aquest mode, l'usuari ha de clicar el botó de “validar” un cop tingui seleccionades les fitxes que desitgi intercanviar del faristol.

Per seleccionar les fitxes que l'usuari vol enviar al sac només les haurà de marcar amb el botó esquerre del faristol, i per anul·lar la selecció d'alguna fitxa tornar a clicar sobre la fitxa seleccionada.

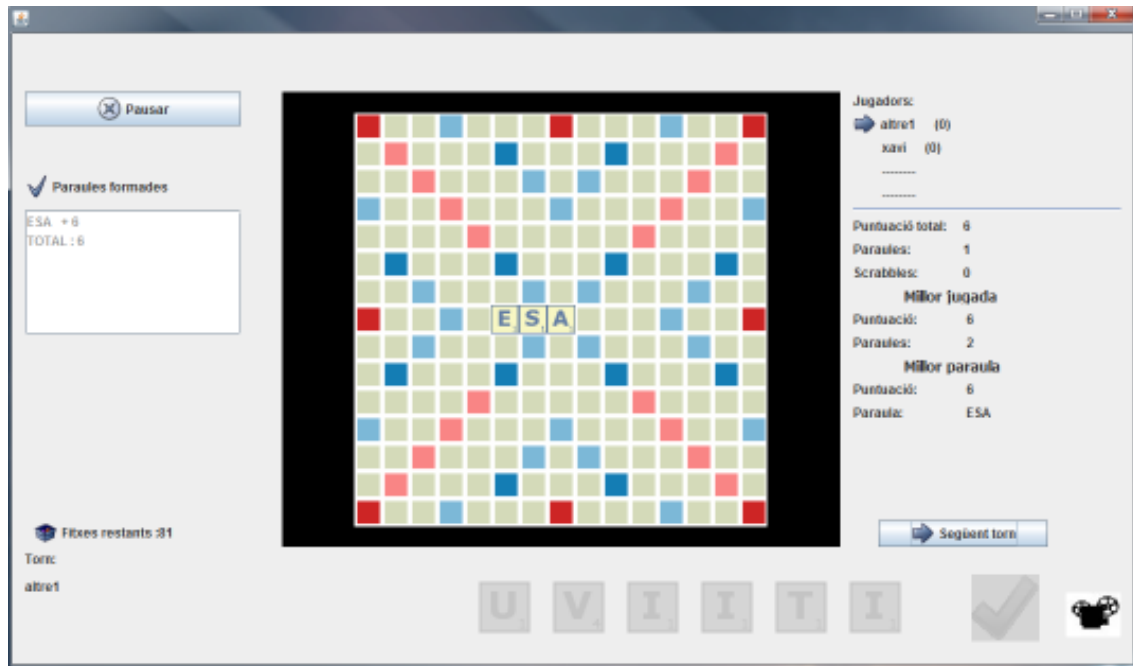


Pantalla en mode de canvi de fitxes

En qualsevol moment l'usuari pot anul·lar l'acció de canviar fitxes clicant en el botó “enrere”. No serà possible canviar les fitxes en les partides duplicades.

10.10.3.2 PARTIDA AMB JUGADA VALIDADA

Un cop el jugador hagi finalitzat el seu torn, el panell canviarà d'estat mostrant les puntuacions de la jugada.



Pantalla de puntuacions de la jugada

En aquest moment, l'aplicació continua permeten aturar la partida, però totes les altres funcions queden inhabilitades. Només es mostrarà la puntuació de la jugada actual i s'habilita el botó de "Següent torn" per continuar la partida.

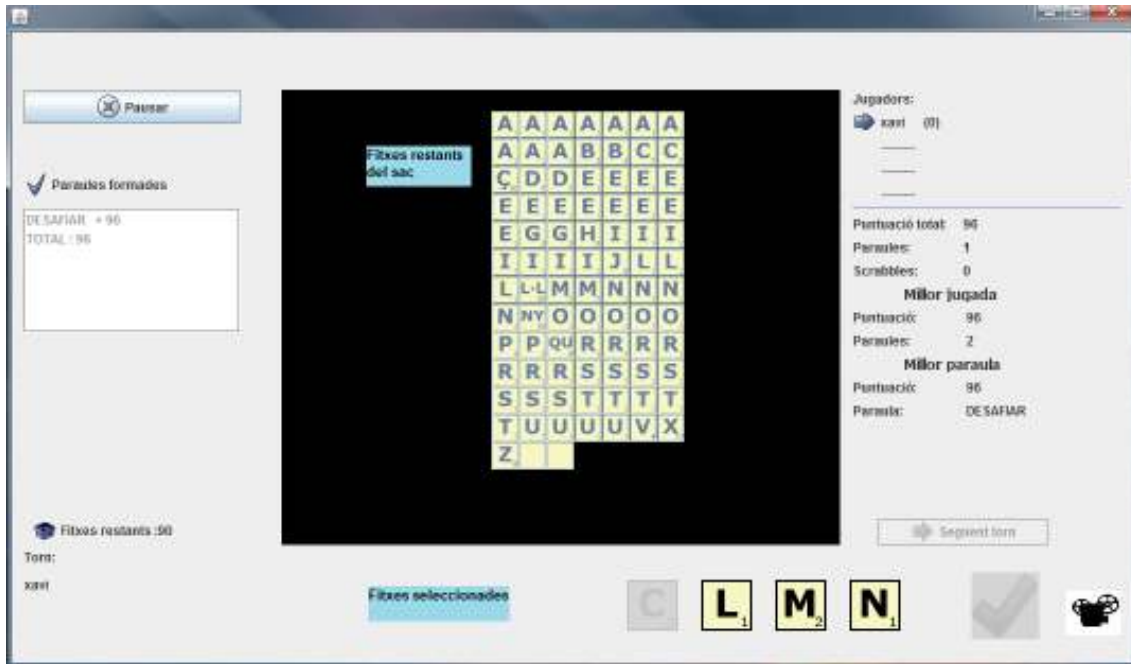
Al passar al següent torn, la partida pot:

- Finalitzar
- Passar al següent torn d'una partida duplicada
- Passar al següent torn en una partida no duplicada

Si finalitza es mostra la pantalla de partida finalitzada, informant de les puntuacions finals, si continua i s'està jugant una partida duplicada es passa al estat "Partida duplicada esperant torn" i en un altre cas, l'aplicació passa a l'estat de "Partida no pausada durant un torn".

10.10.3.3 PARTIDA DUPLICADA ESPERANT TORN

Si ens trobem en una partida duplicada al inici de cada torn s'ha d'escollir les fitxes que resten del faristol del sac de la partida.



Pantalla selecció torn partida duplicada

En aquest estat es permet aturar la partida, i per passar al inici del torn l'usuari ha d'escollir de les lletres restants en el sac, el número de fitxes que restin del faristol. Per retornar una de les fitxes seleccionades al sac l'usuari únicament haurà de clicar a la fitxa del faristol que no desitgi.

Un cop es tingui el faristol complet el botó "Següent torn" permet avançar al inici del torn.

10.10.3.4 PARTIDA PAUSADA

Si l'usuari decideix aturar la partida totes les opcions anteriors desapareixen i es mostren les següents accions:



Opcions partida pausada

10.10.3.4.1 DESAR PARTIDA

Es desarà la partida actual un cop l'usuari indiqui el nom i el directori, aquesta opció només està disponible durant el torn d'una partida.

10.10.3.4.2 TORNAR AL JOC

Torna a l'estat anterior.

10.10.3.4.3 MENÚ PRINCIPAL

Es carrega la pantalla inicial prèvia confirmació de que l'usuari vol abandonar la partida.

10.10.3.4.4 LOG PARTIDA

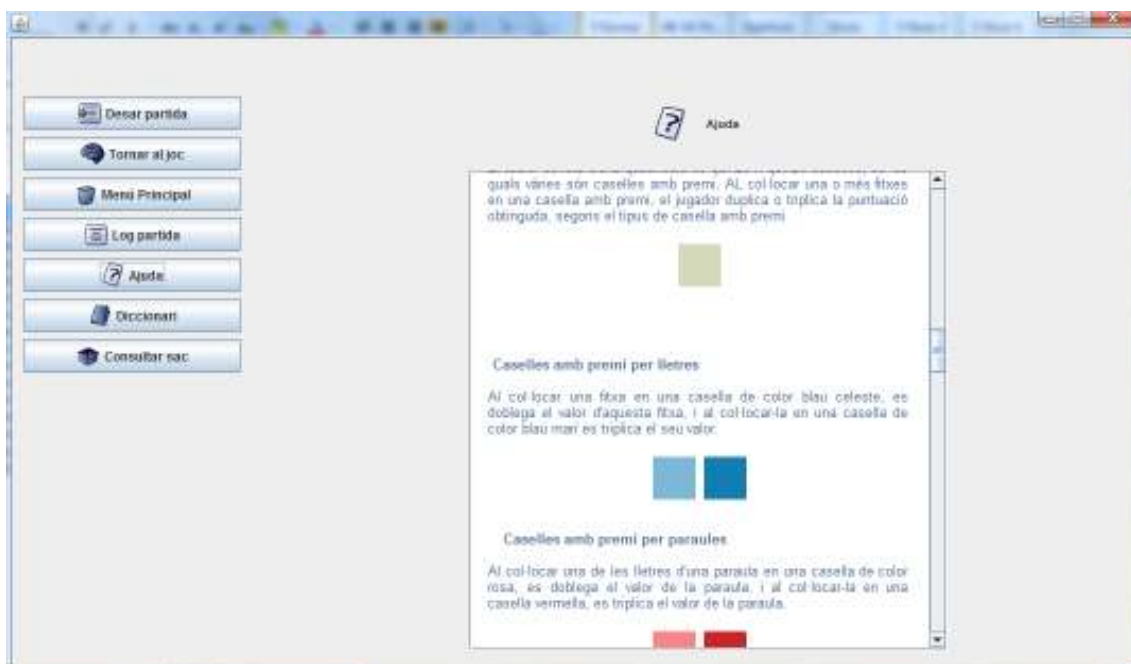
Mostra a l'usuari el log de la partida actual eliminant el faristol dels jugadors robots.

Torn	JUGADOR	Fitxes inici	Paraula	INI	FI	Punt. jugada	Faristol	Punt. fin
1	Robot 3	-----	BAQUETEGEU	1H	9H	126	-----	216
2	xavi	OULIAMDDA	ADULADA	2B	2H	22	OIM	22
3	Robot 3	-----	CAL-LUS	9E	9I	26	-----	242
4	xavi	OIMSANRSO	ADORMIN	1C	7C	20	SSO	42
5	Robot 3	-----	AJACE	9F	13F	29	-----	271
6	xavi	SSOBAEIAM	ABISMOSA	14B	14I	49	E	91
7	Robot 3	-----	AGUSE	11H	15H	30	-----	301

Log de la partida

10.10.3.4.5 AJUDA

Mostra les instruccions del joc 'Scrabble'.



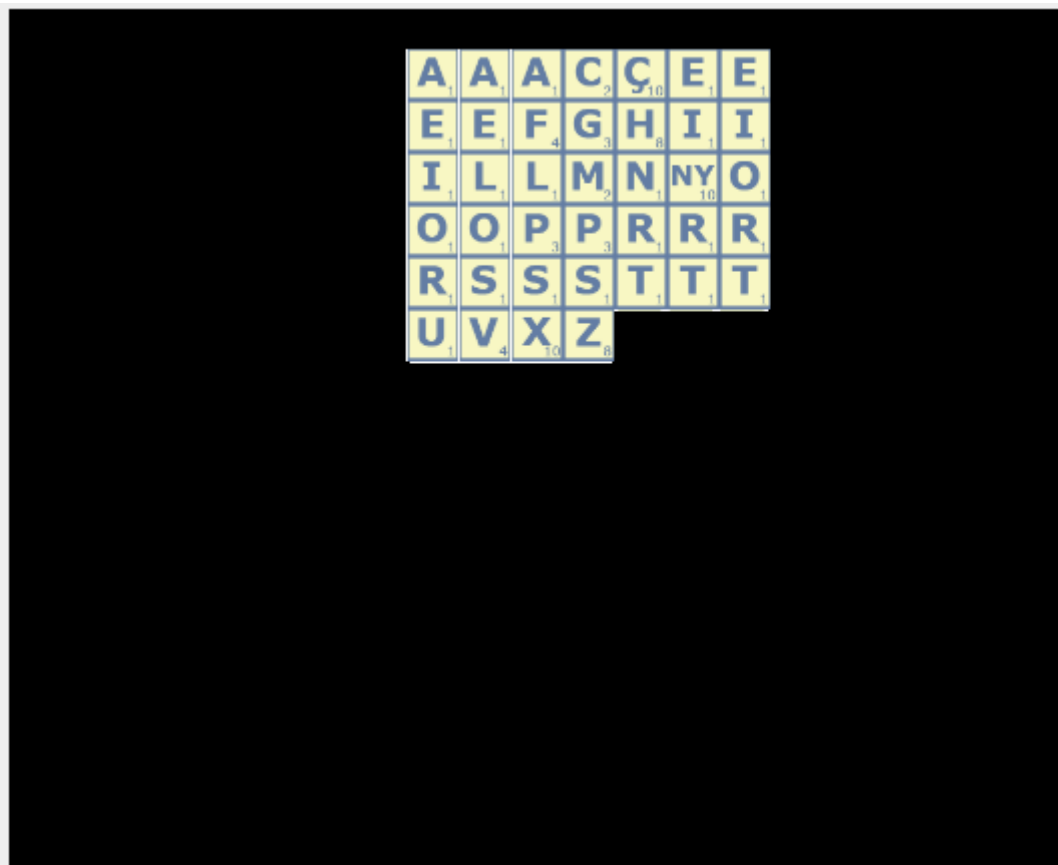
Ajuda de la partida

10.10.3.4.6 **DICCIONARI**

Durant la partida també es permet consultar la pantalla de “gestió de diccionaris” comentada en seccions anteriors, però no es permet gestionar les categories de les paraules que formen el diccionari actual.

10.10.3.4.7 **CONSULTAR SAC**

Mostra a l'usuari les fitxes restants del sac de la partida actual.



Sac de la partida

10.10.3.5 PARTIDA FINALITZADA

Al finalitzar la partida es mostra en el centre del panell les puntuacions finals i el guanyador de la partida, adicionalment es permet guardar el log de la partida.



	Robot 2	xavi
Puntuació:	225	349
Fitxes:	ERSIRE	NR
	-6	-2
Scrabbles:	0	0
	0	0
Puntuació:	219	347

Guanyador !

Final de partida



10.11 AMPLIACIÓ: CREACIÓ DICCIONARIS

Si es desitja afegir un diccionari personalitzat l'usuari pot crear tots els fitxers de paraules que desitgi.

El fitxer haurà de tenir les característiques següents:

- Cap paraula pot portar accents.
- Després de cada paraula ha d'haver-hi un salt de línia.
- El fitxer ha de ser guardat en "UTF-8 sense Bit Order Mask"⁵
- Un cop es té el llistat creat s'ha de desar en una direcció vàlida per a què l'aplicació l'accepti:

```
[Directori_Scrabble]\bin\Datos\[Idioma]\Diccionarios
```

En l'opció de "*gestió de categories de paraules*" de la pantalla "*gestió de jugadors*" es podrà seleccionar el llistat creat un cop desat el fitxer a la carpeta seleccionada.

⁵ Per a facilitar la codificació del fitxer, amb el programa gratuït Notepad++ es permet desar el fitxer en aquest format fàcilment en la columna "Format". Es permet descarregar el programa des de la següent URL: <http://notepad-plus.sourceforge.net/es/download.php>



11 ANNEX 3. INSTRUCCIONS DEL SCRABBLE

En aquesta secció, mostrarem les instruccions bàsiques del joc de taula “Scrabble”, si es vol conèixer el funcionament de l'aplicació s'ha de consultar la secció *“Manual d'usuari”*.

11.1 NORMES BÀSIQUES DEL SCRABBLE

El Scrabble és un joc de taula que es fonamenta en la combinació de peces –cada una amb una lletra- per tal de formar paraules. Un cop col·locades sobre el taulell, els jugadors obtenen una determinada puntuació segons el valor de cada lletra i la seva disposició en el taulell.

11.1.1 INICI DE LA PARTIDA

Per iniciar la partida, en cas que no es jugui sol, cada jugador agafarà una fitxa del total disponible. El jugador que hagi tret la peça més propera a la A en l'ordre alfabètic començarà la partida i en realitzarà l'anotació, si qualsevol jugador obté un comodí o bé te un fitxa repetida, tornarà a agafar una fitxa.

11.1.2 DESENVOLUPAMENT

El Scrabble basa el seu sistema de puntuació en l'aplicació dels factors multiplicadors del taulell a la suma del valor de les lletres usades en una jugada. Un jugador ha de tenir sempre, mentre n'hi hagi suficients, set peces al faristol. Si col·loca les set de cop, la seva jugada pot rebre un plus si així esta especificat en les regles del joc.

Les fitxes en blanc (comodins) poden substituir qualsevol peça del saquet, i el seu valor pot ser o bé 0 punts o el valor de la lletra que representa (en funció de les regles aplicades). Un cop disposat un comodí al taulell, si el jugador que està jugant té la fitxa que representa el comodí en el faristol podrà reemplaçar-la sempre que les regles aplicades en el moment de la partida li permeti.

Tret de la primera jugada, que ha de cobrir la casella central (H8), totes les jugades han d'utilitzar alguna lletra de les que ja són al taulell, de manera que les noves paraules estaran en contacte amb les ja existents, tot encreuant-les, allargant-les o bé formant nous mots de dues o més lletres. El sentit de lectura de les fitxes col·locades només pot ser d'esquerra a dreta o de dalt a baix. No és permès de col·locar fitxes en sentit vertical i horitzontal alhora en una mateixa jugada, tot i que les fitxes col·locades poden formar paraules en tots dos sentits de lectura.

Per comptar el valor d'una jugada cal sumar els valors de totes les paraules formades. Les fitxes que ja eren al taulell només atorguen el seu valor facial, mentre que les noves permeten aprofitar els factors multiplicadors de lletra o de paraula que hi ha en certes caselles del taulell.

A continuació mostrem els factor acumuladors que es pot trobar en un taulell estàndard de Scrabble:

Casella normal (Factor multiplicador 1):



Casella puntuació extra de lletra (Factor multiplicador 2)



Casella puntuació extra de lletra (Factor multiplicador 3)



Caselles puntuació extra de paraula (Factor multiplicador 2)



Casella puntuació extra de paraula (Factor multiplicador 3)



Els factors multiplicadors són acumulables: primer es compten els que afecten al doble o triple valor de lletra; posteriorment, els que afecten al doble o triple valor de paraula. Per tant, si en una mateixa jugada n'hi ha de tots dos tipus, primer es calcularà el valor de cadascuna de les lletres que formen la paraula, es sumaran, i després s'aplicarà el factor multiplicador de paraula.

Quan un jugador forma (per allargament) una paraula que cobreix dues caselles vermelles (que tripliquen paraula), el valor d'aquella paraula es multiplicarà per nou (nòuple). Si les dues caselles trepitjades alhora són duplicades, el valor final de la paraula es multiplicarà per quatre (quàdruple).

Quan un jugador cobreix una casella multiplicadora i aquesta casella passa a formar part tant d'una paraula vàlida en horitzontal com d'una paraula vàlida en vertical, el factor multiplicador actua en tots dos sentits de lectura.

Els dígrafs L·L i NY només poden ser utilitzats utilitzant la peça corresponent o un escarràs. Per tant, no s'admetrà l'ús de dues eles per formar una ela geminada, ni l'ús d'una ena i un escarràs (no existeix la peça Y) per formar la NY.

11.1.3 AGAFAR I CANVIAR PECES

Mentre en quedin suficients al saquet, cada jugador ha de tenir sempre set peces al faristol.

Sempre que les regles ho permetin, quan un jugador té set fitxes al faristol que són tot vocals o tot consonants (sense cap comodí), està autoritzat (no obligat) a canviar totes les lletres abans que tiri el jugador que el segueix, mostrant el



faristol al seu oponent, retornant totes les peces a la bossa i agafant-ne un altre cop set. En aquest cas, el jugador que canvia les lletres no perdrà el torn.

Si després de canviar-les segueix tenint tot vocals o tot consonants, podrà repetir la jugada per canviar les set fitxes del faristol.

Un jugador pot utilitzar el seu torn per canviar tantes lletres com vulgui. En aquest cas, però, no podrà col·locar cap fitxa al taulell i el torn passarà al jugador que el segueix.

El jugador que vulgui fer un canvi voluntari de fitxes, en primer lloc, seleccionarà del faristol quantes fitxes desitja canviar; un cop decidit procedirà a validar la jugada i passar el torn.

També es permet deixar passar el torn en qualsevol moment del joc, sense jugar ni canviar peces.

En qualsevol moment es podrà consultar el nombre de fitxes que queden en el sac i consultar quines queden.

11.1.4 FINAL DE LA PARTIDA

La partida s'acaba quan:

- a) Algun participant esgota totes les lletres del faristol i ja no en queda cap a la bossa. Aleshores es resta del total de punts de cada jugador els punts que li hagin quedat al faristol, i de la mateixa manera es sumen al total del jugador que ha tancat el joc.
- b) Tots els jugadors perden dos torns consecutius. En aquest cas cada jugador se li restarà de la puntuació final el valor de les peces que li quedin al faristol.



11.1.5 DICCIONARI ARBITRAL

El diccionari arbitral utilitzat en català és el Diccionari Oficial de el Scrabble en Català “Diccionari Oficial de el Scrabble en Català (DOSC)”
En qualsevol moment es podrà afegir, eliminar o consultar paraules del diccionari utilitzat, i si s'escau al inici de la partida es permetrà limitar la longitud de les paraules utilitzades.

11.1.6 CONTROL DEL TEMPS

Una jugada no es considerarà completa fins que el jugador no l'hagi col·locada al taulell, l'hagi validat, i hagi comprovat la puntuació obtinguda, en el precís moment que passi al següent torn és quan comença a comptar el temps per a la jugada de l'adversari.

Només s'aturarà el rellotge:

- a) al aturar la partida, per a consultar quelcom.
- b) un cop el jugador validi la jugada.

La gestió del temps en la partida pot ser de tres tipus:

a) Limitació per a jugada:

Cada jugador tindrà un temps determinat per les regles del joc per a una jugada, re iniciant-se el temps cada inici de la jugada.

b) Limitació per a jugador:

En aquest cas no hi haurà un temps determinat per jugada, sinó un temps total per a tota la partida que serà el escollit en les regles al inici de la partida, i que cada oponent podrà administrar com desitgi.

c) Sense limitació del temps:

En aquest cas el jugador no tindrà cap limitació temporal per a fer la seva jugada.



En els dos primers casos, si un jugador excedeix el temps, serà penalitzat a raó de 1 punt per cada 5 segons consumit de més, que es restaran de la puntuació total al final de cada jugada.

11.2 NORMES DE LES PARTIDES DUPLICADES

11.2.1 DEFINICIÓ

Entenem per partida duplicada aquella en la qual tots els jugadors juguen amb el mateix faristol i el mateix taulell. Per tant, el nombre de jugadors és il·limitat, i tots competeixen en absoluta igualtat de condicions donat que s'elimina el possible desavantatge de tenir pitjor combinació de lletres.

11.2.2 NORMES DE LA PARTIDA

El funcionament general de la partida pel que fa al joc, a les puntuacions i a les paraules acceptades es regeix per les normes mencionades anteriorment.

11.2.3 FUNCIONAMENT

- Tots els jugadors participaran junts en una mateixa sala. Es disposarà la infraestructura necessària perquè en tot moment siguin visibles per als participants tant el taulell com el faristol. No s'admetran incorporacions a la partida un cop començada.

- Hi haurà un director/a de partida, que serà l'encarregat/da de l'execució de les ordres entre torn i col·locació de fitxes en el taulell. També dirà en veu alta la jugada que es posa al taulell i les lletres que es trien del saquet, per tal d'evitar



confusions. El director també pot prendre part en la partida com un jugador més.

- En el moment en que les peces quedin disposades al faristol, els jugadors disposaran del temps que s'hagin definit en les regles actuals

- A part dels fulls de proves, es pot proporcionar a tots els jugadors un "full de partida". En aquest full hi haurà un taulell dibuixat on podran anar actualitzant les jugades, i una taula per anar anotant les jugades i les puntuacions. Cada jugador serà el responsable d'anotar els seus punts i de fer-ne la suma total.

- Un cop acabat el temps de cada jugada, el director de joc triarà la millor jugada entre tots els participants . Tot seguit es comprovarà la correcció de la jugada, es procedirà a col·locar-la sobre el taulell, i es completarà el faristol per a la següent.

- La partida acabarà en el moment que la millor jugada de l'últim faristol esgoti totes les lletres. Si en queden menys de set, es seguirà jugant fins que s'esgotin.

- Al final de la partida, si un jugador esgota les peces però els altres no, es restaran del total els punts de les fitxes que quedin al faristol de cada jugador.

- Guanyarà la duplicada el jugador que aconsegueixi una major puntuació total.

12 ANNEX 4. GLOSSARI

En aquest punt comentarem alfabèticament la terminologia que s'utilitza en el joc Scrabble, d'aquesta manera es pretén que el lector es situï en el àmbit del joc sense que tingui que buscar en una altre font fora del document.

- **Acabar.** Finalitzar la partida utilitzant totes les fitxes del faristol i sense que quedin fitxes restants en el sac.
- **Agafar.** Agafar fitxes del sac.
- **Anotació.** Procediment per aixecar acte en el desenvolupament d'una partida.
- **Anotació abreviada.** Anotació en la que únicament consten els punts obtinguts i acumulats per cada jugador.
- **Anotació estàndard.** Anotació en la que es pren constància de cada jugada, de les coordenades de cada jugada i el total acumulat.
- **Anotació integral.** Anotació que, a part de agafar les dades de l'anotació anterior, també es pren constància del faristol de cada jugador.
- **Benjamí.** Jugada que esta tocant una casella de doble o triple paraula.
- **Bicentenària.** Jugada en la que es cobren més de 199 punts.
- **Bonificació.** El fet de cobrar més punts en la jugada pel fet d'estar en una casella especial o fer "Scrabble".
- **Casella.** Cada una de las 225 cel·les del taulell. Hi hauran algunes Caselles especials:
 - Blau fluix. Duplica el valor de la fitxa (L2).
 - Blau fosc. Triplica el valor de la fitxa (L3).
 - Rosa. Duplica el valor de la paraula (P2).
 - Vermell. Triplica el valor de la paraula (P3).
- **Canviar fitxes.** Passar el torn per tenir el dret de substituir algunes de les fitxes del faristol per unes altres aleatòries del sac.
- **Centenària.** Jugada en la que es puntua més de 99 punts.
- **Cobrar.** Obtenir punts.



- **Coordenades.** Lletres i números (si la paraula es situa en vertical) o número i lletra (si es situa en horitzontal) que indiquen la casella corresponent a la primera fitxa d'una jugada.
- **Comodí.** Fitxa sense lletra que pot valer per qualsevol de les altres. Depenen de les regles pot valer zero, o la puntuació de la lletra que representi.
- **Quatricentenaria.** Jugada en la que es cobren més de 399 punts.
- **Quàdruple.** Jugada en la que es toquen dos caselles de tipus P2.
- **Diccionari arbitral.** Aquell que serveix per determinar la validesa d'una jugada.
- **Faristol.** Peça on es recolzen les fitxes del jugador, en la nostra aplicació seria el conjunt de fitxes que té el jugador en el torn per jugar.
- **Fitxa (lletra) cara.** Qualsevol de les que valen 8 o 10 punts.
- **Impugnar.** Sol·licitar comprovació de la validesa de la jugada.
- **Nònuple.** Jugada en la que es toquen dos caselles de tipus P3.
- **Obrir.** Col·locar en el taulell una jugada que incrementa les possibilitats dels que juguin a continuació de poder col·locar les seves jugades.
- **Paret.** Jugada en que la paraula nova es situa en paral·lel de una altra paraula provocant varies paraules perpendiculars a la paraula situada.
- **Partida.** Sèrie de torns que formen un joc complet.
- **Partida de 2, 3, o 4.** Partida en la que intervenen el número de jugadors corresponents
- **Passar.** Renunciar a fer jugada en el propi torn puntuant zero.
- **Quedada.** Jugada en la que s'utilitzen únicament una part de les fitxes del faristol tenint com objectiu no tant el que es puntua sinó, principalment, conservar fitxes susceptibles de provocar un faristol més adequat en el següent torn.
- **Sac.** Recipient que conté les fitxes que encara no han estat repartides.
- **Scrabble.** Jugada en la que es gasten les set fitxes del faristol i que té una bonificació indicada segons les regles del joc.
- **Tancar.** Col·locar en el taulell una jugada que redueix les possibilitats dels que juguen a continuació de poder col·locar la seva.



- **Taulell.** Superfície del joc, subdividida en 225 unitats repartides en 15 files i 15 columnes; les files s'identifiquen amb un número i les columnes amb una lletra.
- **Tètrada.** Aconseguir que un mateix jugador realitzi quatre Scrabbles en una partida.
- **Triada.** Aconseguir que un mateix jugador realitzi tres Scrabbles en una partida.
- **Tricentenaria.** Jugada en la que es puntua més de 299 punts.
- **Torn.** Ocasió de jugar, tant si es fa una jugada como si es passa el torn.
- **Zero.** Jugada nul·la en la que no es cobren punts, per exemple una jugada de zero seria al canviar fitxes.