

Memòria del Projecte Final de Carrera

PuzleFotoApp

Autor: Francisco Javier Marín Gallego
Consultor: Jordi Ceballos Villach
Universitat Oberta de Catalunya

10 de juny de 2014

A Nekane,
que supo adaptarse al duro calendario
y me apoyó en todo momento.

Al pequeño Koldo,
que añadió la dosis justa de incertidumbre al proyecto.

Índex

Índex de figures	5
Índex de taules	6
1 Introducció	7
2 Objectiu del projecte	8
2.1 Justificació del projecte	8
3 Llistat de funcionalitats	10
4 Calendari del projecte	11
4.1 Lliuraments establerts	11
4.2 Calendari	11
4.3 Detall de l'entrega de la PAC-1	12
4.4 Detall de l'entrega de la PAC-2	12
4.5 Detall de l'entrega de la PAC-3	13
4.6 Detall de l'entrega del Lliurament Final	13
5 Recursos i infraestructura	15
5.1 Programari utilitzat	15
5.2 Maquinari utilitzat	16
6 Tecnologia	18
7 Gestió de riscos del projecte	20
8 Anàlisi funcional	21
8.1 Requeriments funcionals	21
8.2 Requeriments no funcionals	21
8.3 Funcionalitats del sistema	22
8.4 Casos d'ús	23
9 Disseny tècnic	27
9.1 Arquitectura de l'aplicació	27
9.2 Descripció de les tecnologies a utilitzar	28

9.3	Model relacional de la base de dades	29
9.3.1	Taula SavedGame	29
9.4	Disseny de classes	30
9.4.1	Classes del Model	30
9.4.2	Classes de la Vista	31
9.4.3	Classes del Controlador	34
9.4.4	Altres definicions	35
10	Prototip	37
10.1	Pantalla d'inici	38
10.2	Pantalla de selecció de puzle	38
10.2.1	Puzle educatiu	39
10.2.2	Fer fotografia	39
10.2.3	Galeria d'imatges	40
10.2.4	Acceptar imatge	40
10.2.5	Partides desades	41
10.3	Pantalla de configuració del puzle	41
10.4	Pantalla de joc	42
10.5	Pantalla per sortir del joc	42
11	Implementació	43
11.1	Model	45
11.2	Vista	47
11.2.1	Classes i paquets	47
11.2.2	Recursos i XML	52
11.3	Controlador	54
11.4	Altres classes de suport	55
12	Funcionament de la aplicació	57
12.1	Instal·lació	57
12.2	Pantalles	58
12.2.1	Pantalla d'inici (P1)	59
12.2.2	Pantalla de selecció de tipus de puzle (P2)	59
12.2.3	Pantalla de selecció de puzle educatiu (P3)	59
12.2.4	Pantalla de puzle a partir d'una fotografia (P4)	60
12.2.5	Pantalla de puzle a partir de la galeria (P5)	60
12.2.6	Pantalla de puzle de partides desades (P6)	60
12.2.7	Pantalla d'acceptació del puzle personalitzat (P7)	61
12.2.8	Pantalla de selecció de dificultat del puzle (P8)	61
12.2.9	Pantalla d'informació de l'aplicació (P9)	61
12.2.10	Pantalla de joc (P10)	62
12.2.11	Pantalla de partida desada (P11)	64
12.2.12	Pantalla per desar partida abans de sortir (P12)	64
12.3	Navegació entre pantalles	64

13	Conclusions	66
13.1	Assoliment d'objectius	66
13.2	Canvis respecte el disseny inicial	67
13.3	Valoració personal	68
13.4	Millores	69
14	Fonts d'informació	70
14.1	Bibliografia	70
14.2	Recursos web Android	70
14.3	Recursos web sobre puzles	71
14.4	Altres recursos	72

Índex de figures

1.1	Joc del 15	7
4.1	Resum del calendari del projecte	12
4.2	Detall del calendari de la PAC-1	12
4.3	Detall del calendari de la PAC-2	12
4.4	Detall del calendari de la PAC-3	13
4.5	Detall del Lliurament Final	13
6.1	Diagrama d'Android	18
8.1	Casos d'ús	23
9.1	Model-Vista-Controlador	27
9.2	Model relacional de la base de dades	29
9.3	Classes del Model	30
9.4	Classes de la Vista	31
9.5	Cicle de vida genèric de qualsevol Activity	32
9.6	Classes del Controlador	34
11.1	Arbre del projecte	44
11.2	Relació entre paquets	45
11.3	RectangleImageView	50
11.4	PaintCanvasPieces	51
11.5	Arbre de recursos i XML	52
12.1	Configuració origen desconegut	57
12.2	Transicions entre pantalles	65

Índex de taules

4.1	Lliuraments establerts	11
4.2	Detall de les tasques de la PAC-1	12
4.3	Detall de les tasques de la PAC-2	13
4.4	Detall de les tasques de la PAC-3	13
4.5	Detall de les tasques del Lliurament Final	14
7.1	Taula de riscos i mitigacions	20
8.1	Cas d'ús abstracte CU01	24
8.2	Cas d'ús CU01-01	24
8.3	Cas d'ús CU01-02	24
8.4	Cas d'ús CU01-03	24
8.5	Cas d'ús CU01-04	25
8.6	Cas d'ús CU01-05	25
8.7	Cas d'ús CU02	25
8.8	Cas d'ús CU03	25
8.9	Cas d'ús CU04	26
8.10	Cas d'ús CU05	26
8.11	Cas d'ús CU06	26
9.1	Taula SavedGame	29
9.2	Mètodes de PuzzleFotoSQLiteOpenHelper	30
9.3	Mètodes de PuzzleFotoSQLite	30
9.4	Descripció de classes de la Vista	33
9.5	Mètodes de PlayerSelectedValues	33
9.6	Mètodes de CanvasPiece	34
9.7	Mètodes de PaintCanvasPieces	34
9.8	Mètodes de Panel	35
9.9	Mètodes de Piece	35

Capítol 1

Introducció

Els puzles lliscants, també coneguts genèricament com *Jocs del 15*¹ (ja que són la versió més coneguda) van ser creats a l'any 1878 per Sam Loyd. Des de la seva aparició van tenir molt èxit, sobre tot als Estats Units.



Figura 1.1: Joc del 15

Com es pot veure a la Figura 1.1 el joc està format per un taulell de (4×4) – 1 peces (d'aquí el seu nom) i un espai buit que s'aprofita per canviar la posició de les peces del puzle. L'objectiu d'aquest joc és aconseguir ordenar totes les peces començant amb una distribució aleatòria, total o parcialment desordenada. La Figura 1.1 mostra la distribució que han de tenir les peces un cop s'ha resolt el joc.

L'inventor d'aquest joc va oferir 1000 dòlars de l'època a la persona que resolgués una configuració que ell havia predeterminat, però ningú ho va aconseguir. No va ser fins 12 anys després de la seva invenció quan dos científics van descobrir que, de totes les combinacions aleatòries inicials de les peces, només la meitat es podien resoldre. Efectivament, Loyd sabia des del principi que els seus 1000 dòlars estaven en lloc segur.

¹Juego del 15. [en línia] http://es.wikipedia.org/wiki/Juego_del_15 [data de consulta: 12/03/2014]

Capítol 2

Objectiu del projecte

A partir del funcionament dels puzles lliscants es vol crear una aplicació amb caràcter educatiu i apta per a persones de qualsevol edat. L'aplicació permetrà moure les peces del puzle de la mateixa manera que en els puzles lliscants, és a dir, aprofitant un espai buit que hi ha al tauler.

Aquesta aplicació basarà la seva funcionalitat en el Joc del 15, però canviant les peces numèriques per imatges. D'aquesta manera el joc serà més atractiu per a qualsevol usuari. Així doncs, la correcta ordenació de totes les peces permetrà visualitzar la imatge o fotografia utilitzada en el puzle.

Pel fet que no totes les distribucions es poden resoldre, l'aplicació haurà de comprovar que la distribució inicial de les peces mostrades a l'usuari és resoluble. Dit d'una altra manera, es descartaran aquelles distribucions de peces inicials sense possibilitat de resolució, amb l'objectiu de no frustrar al jugador.

Un altre objectiu del projecte és de caràcter personal i consisteix en conèixer les eines i la manera de treballar en aplicacions per a mòbils amb sistema operatiu Android¹.

2.1 Justificació del projecte

La major part de les actuals aplicacions d'entreteniment amb més èxit tenen un enfocament marcadament competitiu, és a dir, el jugador ha de superar cada repte proposat per poder continuar jugant. Per exemple, a algunes aplicacions com *Real Racing 3*² o d'altres com *Candy Crush Saga*³ l'usuari ha d'anar superant proves i el joc suggereix constantment la possibilitat de comprar 'milliores' (enfocament del negoci) que faran avançar més ràpidament pels diferents nivells del joc.

¹Android. [en línia] <http://www.android.com> [data de consulta: 12/03/2014]

²Real Racing 3. [en línia] <http://www.ea.com> [data de consulta: 12/03/2014]

³Candy Crash Saga. [en línia] <http://www.candycrushsaga.com> [data de consulta: 12/03/2014]

L'aplicació **PuzleFotoApp** no té l'objectiu 'addictiu' o de negoci que tenen les aplicacions esmentades anteriorment, únicament vol oferir la possibilitat de presentar un joc tradicional com els puzles lliscants (que tenen un caràcter més aviat educatiu) i sense voler o poder competir amb la sofisticació/tecnologia que envolta qualsevol joc d'èxit actual. Un dels objectius d'aquest projecte és aprofitar la popularitat actual dels dispositius mòbils per donar a conèixer un joc tradicional com el puzle lliscant.

Capítol 3

Llistat de funcionalitats

Els puzles tradicionals de peces lliscants acostumen a tenir una configuració típica de $(4 \times 4) - 1$ peces (que no es poden extreure del tauler) i disposen d'un espai buit que s'utilitza per poder canviar la posició de les peces. A part d'aquesta configuració també es poden trobar versions de 3×3 peces.

L'objectiu d'aquest projecte és crear una aplicació que haurà de permetre recrear la jugabilitat que tenen aquests tipus de puzles. Així doncs, tindrem les següents funcionalitats principals:

- Permetre a l'usuari resoldre puzles creats a partir d'imatges incorporades en l'aplicació.
- Permetre a l'usuari resoldre puzles creats a partir d'imatges o fotografies personalitzades
- Permetre a l'usuari l'elecció de la configuració del puzle (2×2 , 3×3 i 4×4).
- Permetre a l'usuari resoldre puzles sense restricció de temps o nombre de moviments (enfocament educatiu, no competitiu).
- Permetre a l'usuari desar la configuració d'una partida iniciada. Per tant, també s'haurà de permetre reprendre la resolució dels puzles en un altre moment.

Capítol 4

Calendari del projecte

El Projecte Final de Carrera segueix el model d'avaluació continuada, i de la mateixa manera que a la resta d'assignatures de la carrera, està basat en l'entrega de diverses PAC. Per aquesta raó, el calendari del projecte tindrà com a fites principals els lliuraments de cadascuna de les PAC. La data d'aquestes fites ha sigut fixada pel consultor de l'assignatura.

4.1 Lliuraments establerts

A continuació es mostra una taula amb les dates d'entrega que s'han establert per cada PAC:

Dates	Fita	Lliurament
12/03/2014	PAC-1	Pla de treball
16/04/2014	PAC-2	Anàlisi funcional, disseny tecnològic i prototip
21/05/2014	PAC-3	Implementació
11/06/2014	Lliurament Final	Memòria i vídeo amb la presentació del projecte

Taula 4.1: Lliuraments establerts

4.2 Calendari

La data d'inici del semestre en què es realitza l'assignatura del Projecte Final de Carrera és el dia 26/02/2014 però la data real en què he iniciat aquest projecte és el dia 03/03/2014. La data final del projecte s'estableix el dia planificat per realitzar el Lliurament Final, és a dir, el dia 11 de Juny de 2014.

Com es pot veure al diagrama de Gantt de la Figura 4.1, el projecte té una duració de 101 dies. D'aquest calendari no s'han eliminat els dies festius per tal d'aprofitar al màxim els dies disponibles.

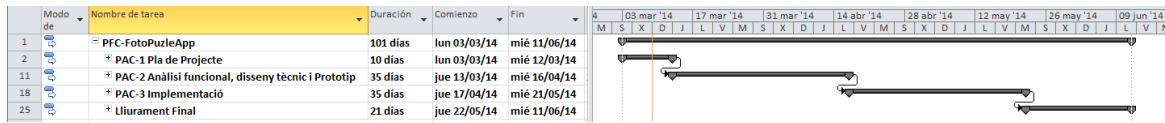


Figura 4.1: Resum del calendari del projecte

4.3 Detall de l'entrega de la PAC-1

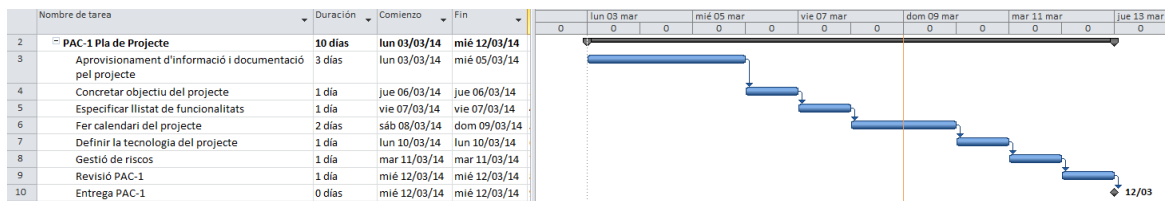


Figura 4.2: Detall del calendari de la PAC-1

Id. tasca	Nom de la tasca	Descripció de la tasca
3	Aprovisionament d'informació i documentació pel projecte	Aprovisionament de manuals de programació per Android, cerca d'eines que es faran servir durant el projecte, etc.
4	Concretar objectius del projecte	Definir què es vol desenvolupar
5	Especificar llistat de funcionalitats	Definir les funcionalitats principals que proporcionarà l'aplicació
6	Fer calendari del projecte	Detallar la data d'inici i de finalització de cada tasca, així com les dependències entre tasques
7	Definir la tecnologia del projecte	Definir la tecnologia que es farà servir per crear l'aplicació
8	Gestió de riscos	Identificar, analitzar i planificar mitigacions pels riscos del projecte
9	Revisió PAC-1	Revisió del document
10	Entrega PAC-1	Lliurament del document

Taula 4.2: Detall de les tasques de la PAC-1

4.4 Detall de l'entrega de la PAC-2

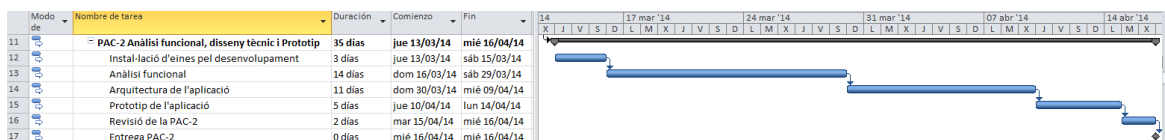


Figura 4.3: Detall del calendari de la PAC-2

Id. tasca	Nom de la tasca	Descripció de la tasca
12	Instal·lació d'eines pel desenvolupament	Descàrrega, instal·lació i proves d'integració de les eines que es faran servir
13	Anàlisi funcional	Estudi de requeriments, especificació de funcionalitats i especificació formal dels casos d'ús
14	Arquitectura de l'aplicació	Definició de l'arquitectura del sistema, disseny de classes, disseny de l'esquema de la base de dades i diagrama de classes
15	Prototip de l'aplicació	Crear un prototip per mostrar quin serà a grans trets l'aspecte de l'aplicació
16	Revisió de la PAC-2	Revisar el document
17	Entrega PAC-2	Lliurament del document i del prototip

Taula 4.3: Detall de les tasques de la PAC-2

4.5 Detall de l'entrega de la PAC-3

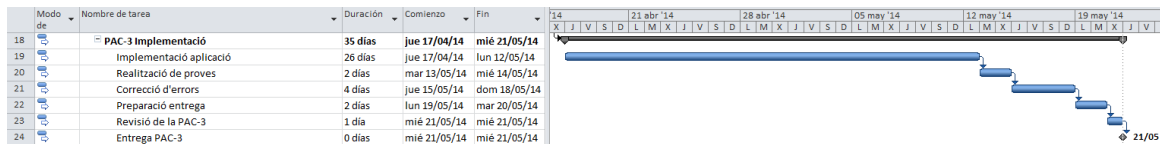


Figura 4.4: Detall del calendari de la PAC-3

Id. tasca	Nom de la tasca	Descripció de la tasca
19	Implementació de l'aplicació	Desenvolupament del programari de l'aplicació i preparació del joc de proves
20	Realització de proves	Realització de proves per acceptar el funcionament de l'aplicació
21	Correcció d'errors	Solucionar els errors de l'aplicació que s'han vist durant la realització de proves
22	Preparació entrega	Preparació del codi font i la documentació que s'ha d'entregar
23	Revisió de la PAC-3	Revisar el document
24	Entrega PAC-3	Lliurament del document i del codi font de l'aplicació

Taula 4.4: Detall de les tasques de la PAC-3

4.6 Detall de l'entrega del Lliurament Final

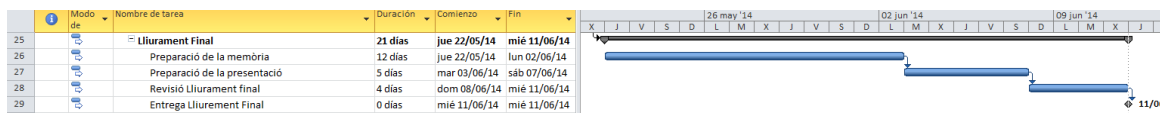


Figura 4.5: Detall del Lliurament Final

Id. tasca	Nom de la tasca	Descripció de la tasca
26	Preparació de la memòria	Recopilació i revisió de la documentació del projecte
27	Preparació de la presentació	Preparació de la presentació i el vídeo que inclou la presentació del projecte
28	Revisió del Lliurament Final	Revisió de la documentació final del projecte
29	Entrega del Lliurament Final	Lliurament de la documentació final del projecte

Taula 4.5: Detall de les tasques del Lliurament Final

Capítol 5

Recursos i infraestructura

En les etapes de disseny, desenvolupament i proves de l'aplicació ha sigut necessària la utilització de programari i maquinari diversos. A continuació es mostra un llistat descriptiu de tots els elements emprats.

5.1 Programari utilitzat

Programari que s'ha fet servir durant la realització del projecte:

- **Android:** Sistema operatiu on s'executa l'aplicació.
- **SDK:** Conjunt de llibreries que fa servir l'aplicació. Aquestes llibreries, entre d'altres, proporcionen l'accés a bases de dades **SQLite**.
- **SQLiteManager:** Extensió de Firefox per la gestió de qualsevol base de dades SQLite.
- **ADT:** Entorn d'edició, compilació i depuració de l'aplicació. Aquest entorn inclou l'editor Eclipse, el gestor de màquines virtuals d'Android (AVD Manager), el gestor de versions de SDK d'Android (Android SDK Manager) i està preconfigurat per desenvolupar aplicacions amb llenguatge JAVA.
- **DIA:** Aplicació per realitzar els diagrames UML que s'han utilitzat principalment en l'etapa de disseny.
- **Evolus Pencil:** Eina de prototipatge i maquetació per dissenyar les pantalles abans de desenvolupar l'aplicació.

A continuació es descriuen altres eines de suport que no s'utilitzen en el propi desenvolupament de l'aplicació però que són necessàries en el projecte:

- La utilització conjunta de **TexMaker** i **MikTex**, com a editor/processador de textos i gestor de paquets respectivament, per crear la memòria del projecte mitjançant el llenguatge \LaTeX .

- **Microsoft Project:** per crear la planificació temporal del projecte.
- **TortoiseSVN:** programari per realitzar la gestió de versions de la documentació, el codi font i la resta d'arxius utilitzats en el projecte.
- **FileZilla:** client FTP per realitzar transferència d'arxius entre l'ordinador de desenvolupament i el dispositiu Android en el que s'han realitzat les proves.
- **FastStone:** programari per visualitzar i editar imatges.
- **Paint:** programari per editar imatges.
- **GraphViz:** programari utilitzat per la creació de diagrames.
- **Camtasia Studio:** programari utilitzat per l'edició del vídeo de presentació del projecte.

5.2 Maquinari utilitzat

El maquinari utilitzat en les diferents etapes del projecte ha sigut un ordinador HP¹ i un telèfon mòbil Motorola². La fase de desenvolupament de l'aplicació i les primeres proves s'han realitzat sobre l'ordinador. En canvi, les proves exhaustives finals s'han realitzat sobre el dispositiu mòbil.

A continuació s'especifiquen les característiques del maquinari utilitzat:



HP Pavillion 500-129es:

- Intel® Core™ i7-4770 a 3,4 GHz
- RAM: 8Gb DDR3
- Disc Dur 1 TB sATA 7200 rpm
- AMD Radeon HD 8570 2 GB DDR3
- Microsoft Windows 8

¹HP. [en línia] <http://www.hp.com> [data de consulta: 26/05/2014]

²Motorola. [en línia] <http://www.motorola.es> [data de consulta: 26/05/2014]

**Motorola Moto G:**

- Qualcomm Snapdragon 400 a 1,2 Ghz
- Pantalla Gorilla Glass de 4,5"
- Resolució: 1280 x 720 (HD)
- Memòria emmagatzament: 8 Gb
- RAM: 1 Gb
- Android Kit Kat 4.4.2

Capítol 6

Tecnologia

El projecte consisteix en desenvolupar una aplicació que s'executa en un dispositiu mòbil, fent servir un PC per desenvolupar i depurar l'aplicació. L'execució de l'aplicació no requereix cap altre dispositiu ni tecnologia externa al propi dispositiu mòbil.

El sistema operatiu en què s'executa l'aplicació és Android (basat en Linux). Està estructurat en capes per tal de facilitar el desenvolupament d'aplicacions. A continuació es mostra quins són els elements principals que formen la seva arquitectura¹:



Figura 6.1: Diagrama d'Android

Els elements o components de l'Arquitectura estan estructurats en capes i cada capa

¹Diagrama Android. [en línia] http://commons.wikimedia.org/wiki/File:Diagrama_android.png [data de consulta: 12/03/2014]

només fa servir els components de la capa inferior. Per aquesta raó, aquesta arquitectura també s'anomena arquitectura de pila. A continuació es fa un breu resum de la funcionalitat² de cadascuna de les capes:

1. Capa d'aplicació: En aquesta capa es troben totes les aplicacions que fa servir directament l'usuari (widgets, aplicacions que porta el sistema, aplicacions que instal·la l'usuari, etc.)
2. Framework d'aplicació: Es tracta de totes les classes o serveis que fa servir qualsevol component de la capa d'aplicació (Biblioteques per reproduir vídeo, administrador de finestres, gestor de bases de dades, etc.)
3. Entorn d'execució: En aquesta capa es troba la màquina virtual Dalvik que s'encarrega d'executar les aplicacions no natives.
4. Llibreries: Es tracta del conjunt de llibreries natives (C i C++) que acostumen a ésser proporcionades pel fabricant del dispositiu.
5. Kernel de Linux: En aquesta capa trobem el nucli del sistema operatiu Android.

D'una banda, en relació amb les aplicacions per Android, podem trobar desenvolupadors que utilitzen el llenguatge Java (aplicacions administrades) per realitzar les aplicacions que s'executen sobre la màquina virtual Dalvik³ (especialment optimitzada per ser ràpida i eficient en dispositius mòbils). D'altra banda, tenim els desenvolupaments nadius en llenguatge C i C++. En el cas concret de l'aplicació **PuzleFotoApp** s'utilitza el llenguatge **JAVA** i es fa servir **SQLite**⁴ com a gestor de base de dades.

²Arquitectura de Android.[en línia]<http://columna80.wordpress.com/2011/02/17/arquitectura-de-android/>[data de consulta: 12/03/2014]

³Dalvik. [en línia] [http://en.wikipedia.org/wiki/Dalvik_\(software\)](http://en.wikipedia.org/wiki/Dalvik_(software)) [data de consulta: 12/03/2014]

⁴SQLite. [en línia] <http://en.wikipedia.org/wiki/SQLite> [data de consulta: 12/03/2014]

Capítol 7

Gestió de riscos del projecte

Per afrontar el projecte amb unes garanties mínimes d'èxit cal realitzar una gestió de riscos. Aquesta manera de treballar permetrà augmentar la probabilitat de rebre events que influeixen positivament en el projecte, així com disminuir la probabilitat de rebre events que poden influir negativament. A continuació s'indiquen les tasques que s'han de portar a terme per realitzar la gestió de riscos del projecte:

1. Identificar els riscos del projecte.
2. Realitzar l'anàlisi qualitatiu (impacte) i quantitatiu (probabilitat) dels riscos identificats.
3. Planificar mitigacions associades als riscos i realitzar un seguiment.

A la taula de riscos i mitigacions es pot observar la identificació de riscos i mitigacions d'aquest projecte:

RISC	DESCRIPCIÓ	PROBABILITAT	IMPACTE	MITIGACIÓ
Dificultats al desenvolupar amb Android	La utilització d'una tecnologia nova sense formació prèvia pot dificultar el desenvolupament del projecte	Mitja	Alt	Aprovisionament de documentació per fer front als dubtes que apareixen
Planificació incorrecte del projecte	Una planificació incorrecte d'una fase pot endarrerir el projecte	Mitja	Mitjà	Ser estricte amb el calendari i re-programar certes tasques si és necessari
Conciliació de la vida familiar i malalties	Tenir cura d'un nadó i malalties que impedeixen realitzar el projecte	Inevitable	Mitjà	Conscienciació familiar i seguir les recomanacions de les autoritats sanitàries
Afectació del projecte per temes laborals	L'horari laboral pot acabar interferint amb les hores dedicades al projecte.	Baixa	Mitjà	Informar a l'empresa amb temps. Agafar dies de vacances puntualment
Avaria en els dispositius utilitzats	Avaria dels equips de desenvolupament o del dispositiu mòbil	Baixa	Mitjà	Fer còpies de seguretat habitualment. Maquinari addicional per contingències

Taula 7.1: Taula de riscos i mitigacions

Capítol 8

Anàlisi funcional

En aquest apartat es presenten els requeriments de l'aplicació, les funcionalitats i els diagrames de casos d'ús per tenir una perspectiva més propera al funcionament de l'aplicació **PuzleFotoApp**.

8.1 Requeriments funcionals

L'aplicació **PuzleFotoApp** ha de:

- recrear la jugabilitat dels puzles lliscants (aprofitant la tecnologia i la facilitat d'ús dels dispositius mòbils actuals),
- permetre deixar un puzle a mig resoldre per continuar amb la partida en un altre moment,
- permetre a l'usuari escollir la imatge que s'utilitzarà en el puzle, fent ús de la càmera del dispositiu o de la galeria d'imatges del dispositiu.

8.2 Requeriments no funcionals

A continuació s'indiquen els requeriments no funcionals de l'aplicació:

- L'aplicació s'ha d'executar en un dispositiu compatible amb el sistema operatiu Android entre la versió 2.2 (Froyo¹) i la versió 4.4 (Kit Kat²). En termes de llibreries pel desenvolupament equival a les versions que hi ha entre la 8 i la 19 del Software Development Kit (SDK).
- El dispositiu ha de disposar de càmera digital.
- El dispositiu ha de disposar de pantalla tàctil.

¹Froyo. [en línia] <http://developer.android.com/about/versions/android-2.2.html> [data de consulta: 14/04/2014]

²Kit Kat. [en línia] <http://developer.android.com/about/versions/android-4.4.html> [data de consulta: 14/04/2014]

De tota la gama de dispositius amb el sistema operatiu Android que hi ha al mercat, l'aplicació només es prova en un dispositiu de marca *Motorola* i model *Moto G* amb la versió Kit Kat 4.4.2, de manera que no es garanteix el correcte funcionament en la resta de dispositius (encara que la implementació té en compte les restriccions per la versió Froyo 2.2).

8.3 Funcionalitats del sistema

Tenint en compte els requeriments indicats en les seccions anteriors l'aplicació haurà d'implementar les següents funcionalitats:

- Oferir puzles educatius. Aquests puzles mostren una breu explicació relacionada amb la imatge, una vegada que s'ha resolt el puzle.
- Permetre escollir altres imatges per crear el puzle. Aquestes imatges es poden obtenir de la càmera del dispositiu o de la galeria d'imatges del dispositiu.
- Permetre a l'usuari escollir la configuració del nombre de peces del puzle: 2×2 , 3×3 o 4×4 .
- Barrejar les peces d'un puzle perquè l'usuari pugui començar una partida. La posició inicial de cadascuna de les peces ha de permetre que el puzle sigui resoluble. És a dir, es descartaran les disposicions de peces que generin puzles no resolubles.
- Permetre la resolució del puzle (per part del jugador) mitjançant moviments verticals i horitzontals (mai en diagonal) de les peces que formen el puzle. Només es podran moure aquelles peces que tenen l'espai buit al seu costat (verticalment o horitzontalment parlant). La interacció de l'usuari amb la pantalla tàctil del dispositiu ha de fer possible el moviment de les peces.
- Permetre guardar (de manera persistent) l'estat que té un puzle (només en les partides no finalitzades).
- Permetre continuar jugant una partida que es va guardar de manera persistent.

8.4 Casos d'ús

El diagrama de casos d'ús associat a l'aplicació no és gaire complexe en quant a nombre d'actors o a nombre total de casos d'ús. A continuació es mostren els casos d'ús de **PuzleFotoApp**:

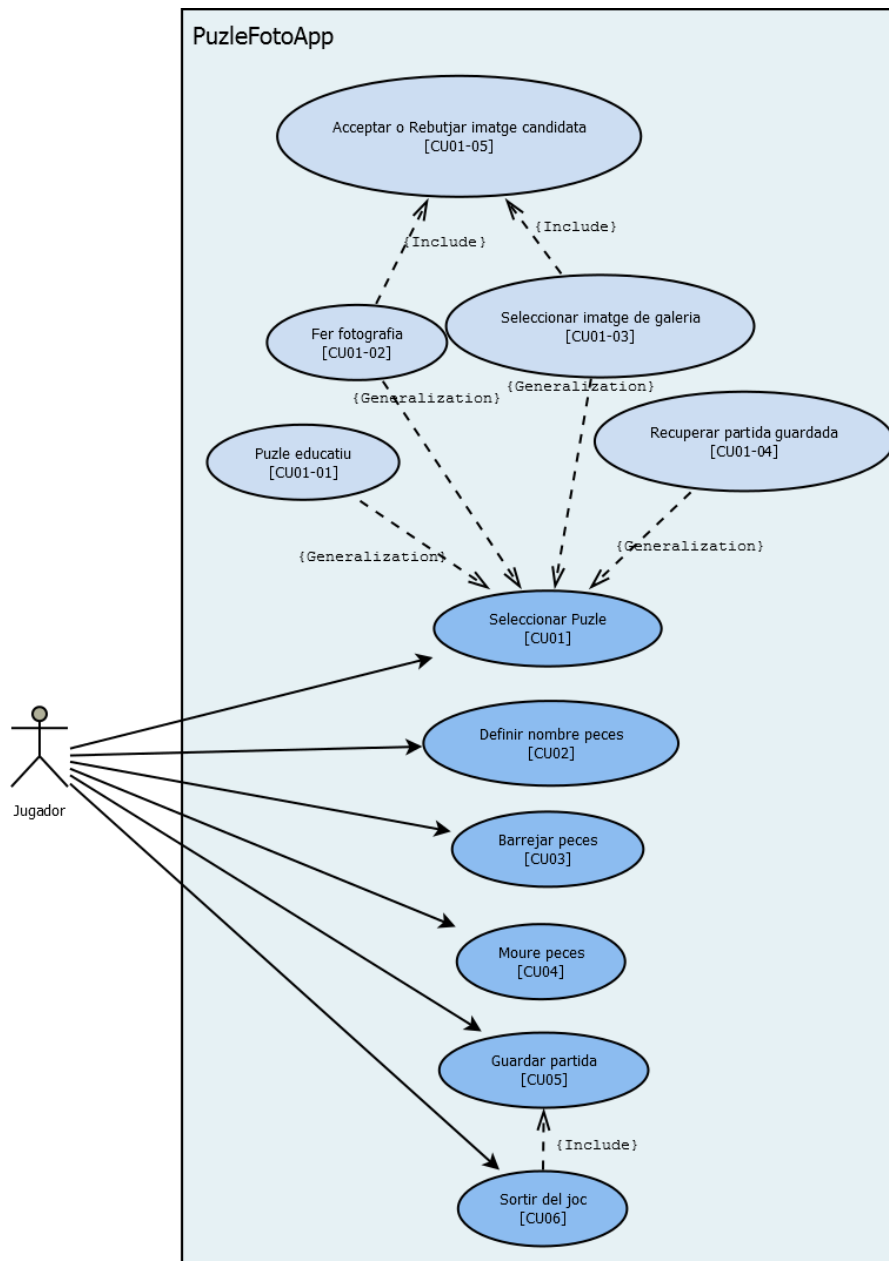


Figura 8.1: Casos d'ús

A continuació es detallan cadascun dels casos d'ús que es mostren a la Figura 8.1, on l'únic actor (per tots els casos) és l'usuari. El primer cas d'ús (CU01) té lloc quan s'ha iniciat l'aplicació.

Identificador	CU01
Nom	Seleccionar Puzle
Descripció	Aquest cas d'ús és una generalització dels casos d'ús CU01-01, CU01-02, CU01-03 i CU01-04. És a dir, el cas d'ús CU01 és abstracte i per això existeixen els casos d'ús especialitzats CU01-01, CU01-02, CU01-03 i CU01-04. <i>(Generalització.[en línia] http://www.uml-diagrams.org/use-case.html#generalization [data de consulta: 20/03/2014])</i>
Precondició	L'usuari encara no ha escollit un puzle per jugar
Postcondició	L'usuari ha seleccionat un puzle per jugar
Flux d'execució	Veure flux d'execució de CU01-01, CU01-02, CU01-03, CU01-04

Taula 8.1: Cas d'ús abstracte CU01

Identificador	CU01-01
Nom	Puzle educatiu
Descripció	L'usuari selecciona un dels puzles educatius que li ofereix l'aplicació
Precondició	L'usuari encara no ha escollit un puzle
Postcondició	L'usuari ha seleccionat un puzle educatiu
Flux d'execució	Següent: CU02

Taula 8.2: Cas d'ús CU01-01

Identificador	CU01-02
Nom	Fer fotografia
Descripció	L'usuari captura una imatge amb la càmera del dispositiu mòbil com a imatge candidata del puzle.
Precondició	L'usuari encara no ha escollit un puzle
Postcondició	L'usuari ha fet una fotografia amb la càmera del dispositiu. Aquesta imatge serà candidata a utilitzar-se com a imatge del puzle
Flux d'execució	Següent: CU01-05

Taula 8.3: Cas d'ús CU01-02

Identificador	CU01-03
Nom	Seleccionar imatge de galeria
Descripció	L'usuari selecciona una imatge de la galeria d'imatges del dispositiu.
Precondició	L'usuari encara no ha escollit un puzle
Postcondició	L'usuari ha seleccionat una imatge de la galeria d'imatges del dispositiu. Aquesta imatge serà candidata a utilitzar-se com a imatge del puzle
Flux d'execució	Següent: CU01-05

Taula 8.4: Cas d'ús CU01-03

Identificador	CU01-04
Nom	Recuperar partida guardada
Descripció	L'usuari ha decidit continuar jugant una partida que s'havia guardat prèviament
Precondició	L'usuari encara no ha escollit un puzle
Postcondició	L'usuari ha seleccionat un puzle amb la mateixa imatge i la disposició de les peces que hi havien l'última vegada que es va guardar la partida
Flux d'execució	Següent: CU04

Taula 8.5: Cas d'ús CU01-04

Identificador	CU01-05
Nom	Acceptar o rebutjar imatge candidata
Descripció	L'usuari ha de decidir si accepta o rebutja la imatge (capturada amb la càmera o seleccionada des de la galeria del dispositiu)
Precondició	L'usuari encara no ha escollit la imatge que farà servir en el puzle
Postcondició	L'usuari utilitza la imatge escollida en el puzle
Postcondició (alternativa)	L'usuari descarta la imatge i no la farà servir en el puzle
Flux d'execució	Següent: CU02 (en cas d'acceptar) o CU01 (en cas de rebutjar)

Taula 8.6: Cas d'ús CU01-05

Identificador	CU02
Nom	Definir nombre peces
Descripció	L'usuari selecciona les dimensions que tindrà el tauler. Per un tauler de $N \times N$ posicions es faran servir $N \times N - 1$ peces i un espai buit
Precondició	L'usuari encara no ha escollit el nombre de peces que tindrà el puzle
Postcondició	L'usuari ha seleccionat el nombre de peces que té el puzle
Flux d'execució	Següent: CU03

Taula 8.7: Cas d'ús CU02

Identificador	CU03
Nom	Barrejar peces
Descripció	L'usuari prem el botó de barrejar peces i les peces que formen el puzle queden barrejades per poder començar a jugar
Precondició	L'usuari encara no ha barrejat les peces per iniciar la partida
Postcondició	L'usuari ha barrejat les peces del puzle per poder començar a jugar
Flux d'execució	Següent: CU04

Taula 8.8: Cas d'ús CU03

Identificador	CU04
Nom	Moure peces
Descripció	L'usuari toca una de les peces del puzle per canviar la disposició de les peces
Precondició	L'usuari encara no ha resolt la partida
Postcondició	L'usuari ha mogut una de les peces. La partida finalitzarà quan totes estiguin ordenades
Flux d'execució	Següent: CU04 (Moure peces) o CU05 (Guardar partida) o CU06 (Sortir del joc)

Taula 8.9: Cas d'ús CU04

Identificador	CU05
Nom	Guardar partida
Descripció	L'usuari prem el botó per guardar la partida en el decurs del joc
Precondició	No hi ha guardada de manera persistent la imatge amb la disposició actual de les peces del puzle
Postcondició	L'usuari ha guardat de manera persistent la imatge i la disposició actual de les peces del puzle per poder continuar la partida més endavant
Flux d'execució	Següent: CU04

Taula 8.10: Cas d'ús CU05

Identificador	CU06
Nom	Sortir del joc
Descripció	L'usuari prem el botó per sortir del joc
Precondició	L'usuari ja ha iniciat una partida o continua jugant una partida anteriorment guardada
Postcondició	L'usuari ha sortit del joc de manera controlada, és a dir, guardant la situació actual de la partida o descartant la partida (no es podrà continuar en un altre moment)
Flux d'execució	Aquest cas d'ús és el darrer en executar-se
Inclusions	CU05 (Es dona la possibilitat de guardar la partida abans de sortir del joc)

Taula 8.11: Cas d'ús CU06

Capítol 9

Disseny tècnic

En aquest capítol es presenta el disseny tècnic de l'aplicació **PuzleFotoApp**. És a dir, es descriu la seva arquitectura, la tecnologia utilitzada i el seu disseny (a nivell de classes i de base de dades).

9.1 Arquitectura de l'aplicació

L'arquitectura de **PuzleFotoApp** fa servir el patró de disseny Model-Vista-Controlador¹. Aquest patró de disseny es caracteritza per la separació que hi ha entre cadascuna de les parts que el formen.

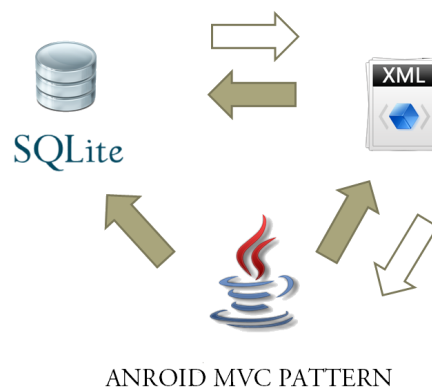


Figura 9.1: Model-Vista-Controlador

Per un costat, el model representa la base de dades de l'aplicació. D'altra banda, la vista s'implementa mitjançant l'ajuda d'arxius XML on es defineixen els elements que es mostren a cada pantalla. Finalment, el controlador està format per les classes

¹MVC Web. [en línia] http://es.wikipedia.org/wiki/Modelo_Vista_Controlador [data de consulta: 02/04/2014]

Java que contenen la lògica de l'aplicació, és a dir, on s'implementen les regles de funcionament del puzle.

El patró de disseny² mostrat a la Figura 9.1 no només es fa servir en aplicacions Android, també es fa servir en la majoria d'aplicacions Web que podem trobar actualment (amb alguns matisos), ja que la separació dels seus components permet facilitar les tasques de desenvolupament i de manteniment.

9.2 Descripció de les tecnologies a utilitzar

La tecnologia utilitzada per realitzar l'aplicació **PuzleFotoApp** està influenciada pel sistema operatiu amb què s'executa l'aplicació, és a dir, Android. Tal i com es va explicar al capítol 6, aquest sistema operatiu està estructurat en capes i cada capa fa servir els components de la capa immediatament inferior.

El llenguatge de programació utilitzat en el projecte és Java, però s'utilitzen les funcionalitats que aporten les llibreries pròpies d'Android³, enfocades especialment a dispositius mòbils. A continuació es destaquen algunes de les funcionalitats que es fan servir a l'aplicació:

- **Activity:** Són classes en què es defineixen les pantalles que veu l'usuari i on es gestiona la interacció entre l'aplicació i l'usuari.
- **Intents:** Són descripcions abstractes de les operacions que una aplicació és capaç de realitzar. Aquesta informació pot ser utilitzada per una altra aplicació que estigui interessada en reutilitzar una determinada funcionalitat.
- **Parcel:** Permet el traspàs d'informació (cadena, nombres, fins i tot objectes) entre 2 activitats de l'aplicació. Es tracta d'una versió adaptada als dispositius mòbils de la funcionalitat *Serializable* de Java.
- **SQLite:** Base de dades que es caracteritza per la seva lleugeresa i per la facilitat d'ús. Ofereix la possibilitat de construir les instruccions de consulta en 'cru', mitjançant el llenguatge SQL, o amb instruccions equivalents, sense necessitat de fer servir el llenguatge SQL.

A més, cal remarcar que la tecnologia utilitzada depèn de la versió de la llibreria utilitzada (veure apartat 8.2), ja que cada versió (del SDK) posa a disposició del desenvolupador un nombre diferent de funcionalitats.

²Model-Vista-Controlador. [en línia] <http://techworkslab.blogspot.com.es/2013/10/hi-friends-today-i-am-going-to-start.html> [data de consulta: 02/04/2014]

³Android API. [en línia] <http://developer.android.com/reference/packages.html> [data de consulta: 05/04/2014]

9.3 Model relacional de la base de dades

La base de dades de l'aplicació ha d'emmagatzemar la informació relacionada amb les partides desades pel jugador. A continuació es mostra el model relacional, que només conté una taula:

SavedGame	
*idSavedGame	INTEGER
*idPuzleEducational	INTEGER
°pathFile	TEXT
*dimensions	INTEGER
*puzledate	TIMESTAMP
*piecePosition	TEXT
*puzleType	INTEGER
°educationalTitle	TEXT
°educationalMessage	TEXT
°aRL	INTEGER
°aRR	INTEGER
°aRT	INTEGER
°aRB	INTEGER

Figura 9.2: Model relacional de la base de dades

A continuació es descriu breument cadascun dels atributs de la taula.

9.3.1 Taula SavedGame

Atribut	Descripció	Tipus	Clau
idSavedGame	Identificador de la partida guardada	Numèric	Si
idPuzleEducational	Identificador pels puzles educatius (identificador d'imatge)	Numèric	No
pathFile	Ubicació de l'arxiu amb la imatge (per puzles no educatius)	Text	No
dimensions	Nombre total de peces que formen el puzle	Numèric	No
puzledate	Data en què es va guardar la partida	Numèric	No
piecePosition	Cadena de text que descriu la posició de les peces	Text	No
puzleType	Tipus de puzle (educatiu, fotografia, etc...)	Numèric	No
educationalTitle	Títol associat a un puzle educatiu	Text	No
educationalMessage	Missatge mostrat al resoldre un puzle educatiu	Text	No
aRL	Coordenades de retall de la imatge (esquerra) per puzles no educatius	Numèric	No
aRR	Coordenades de retall de la imatge (dreta) per puzles no educatius	Numèric	No
aRT	Coordenades de retall de la imatge (superior) per puzles no educatius	Numèric	No
aRB	Coordenades de retall de la imatge (inferior) per puzles no educatius	Numèric	No

Taula 9.1: Taula SavedGame

9.4 Disseny de classes

En aquest apartat es descriuen les classes utilitzades en l'aplicació, segons la seva pertinença al model, a la vista o al controlador.

9.4.1 Classes del Model

Les classes que pertanyen al model estan relacionades amb la base de dades i amb els elements que s'emmagatzemen a la base de dades. Aquestes classes realitzen tasques com la creació de la base de dades, la consulta de dades o la inserció de nous elements. A continuació es mostra el diagrama de classes:

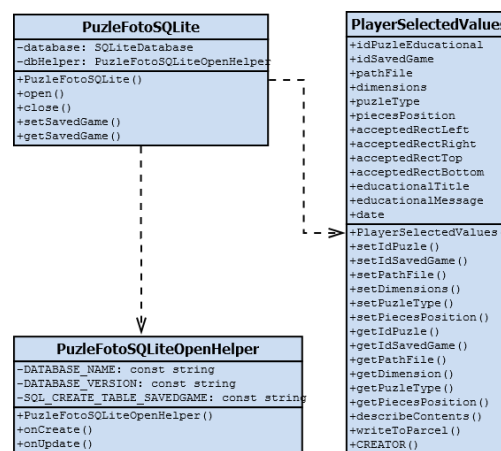


Figura 9.3: Classes del Model

En aquesta figura es mostra la classe PlayerSelectedValues, encara que també es fa servir com a classe de la vista (s'explica en la següent secció). A continuació s'expliquen els mètodes de la classe PuzzleFotoSQLiteOpenHelper i PuzzleFotoSQLite:

Mètode	Descripció
PuzzleFotoSQLiteOpenHelper	Constructor de la classe
onCreate	Crea la base de dades amb la taula SavedGame
onUpdate	Actualització de la base de dades si es canvia l'identificador de versió

Taula 9.2: Mètodes de PuzzleFotoSQLiteOpenHelper

Mètode	Descripció
PuzzleFotoSQLite	Constructor de la classe
open	Connecta amb la base de dades en mode lectura/escriptura
close	Desconnecta de la base de dades
setSavedGame	Guarda el contingut de PlayerSelectedValues a la base de dades
getSavedGame	Actualitza PlayerSelectedValues amb el contingut de la base de dades

Taula 9.3: Mètodes de PuzzleFotoSQLite

9.4.2 Classes de la Vista

Les classes que pertanyen a la vista contenen les interfícies que l'usuari de l'aplicació veu quan executa l'aplicació. Aquestes classes realitzen tasques de presentació d'informació en la pantalla del dispositiu i capturen la informació que l'usuari introdueix en el moment d'interactuar amb l'aplicació.

Per tal de poder visualitzar correctament (en aquest document) el diagrama de classes, s'han eliminat algunes relacions, s'han obviat la majoria dels seus mètodes i s'han deixat aquells que realment aporten una funcionalitat distintiva a cada classe.

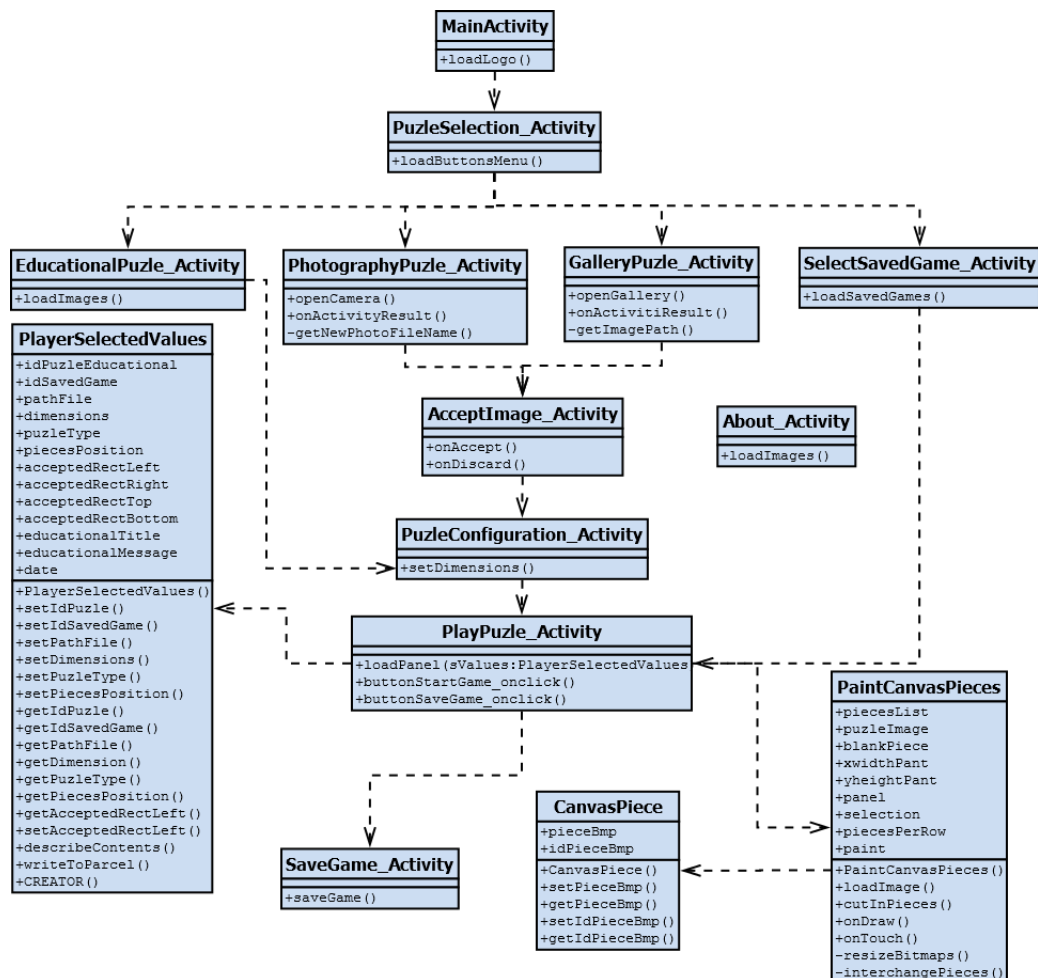


Figura 9.4: Classes de la Vista

Per simplificar el diagrama no s'han mostrat les relacions amb la classe `About_Activity`, però cal dir que estan relacionades totes les activitats. En l'explicació del contingut de cadascuna de les classes mostrades a la Figura 9.4 es diferencia entre aquelles que són de tipus *Activity* i aquelles que no ho són. Les classes de tipus *Activity* tenen les següents característiques:

- Tenen associada una de les pantalles que l'usuari de l'aplicació veu.
- Les activitats poden tenir les següents funcions: `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()`, `onDestroy()` i `onRestart()`.
- Totes les activitats fan servir Fragments per encapsular el que volen mostrar. Aquests Fragments poden tenir les funcions pròpies del cicle de vida d'un Fragment: `onAttach()`, `onCreateView()`, `onActivityCreated()`, `onDestroyView()`, etc.

A continuació es mostra amb més detall el cicle de vida⁴ d'una activitat des del seu 'llançament' fins el seu 'tancament':

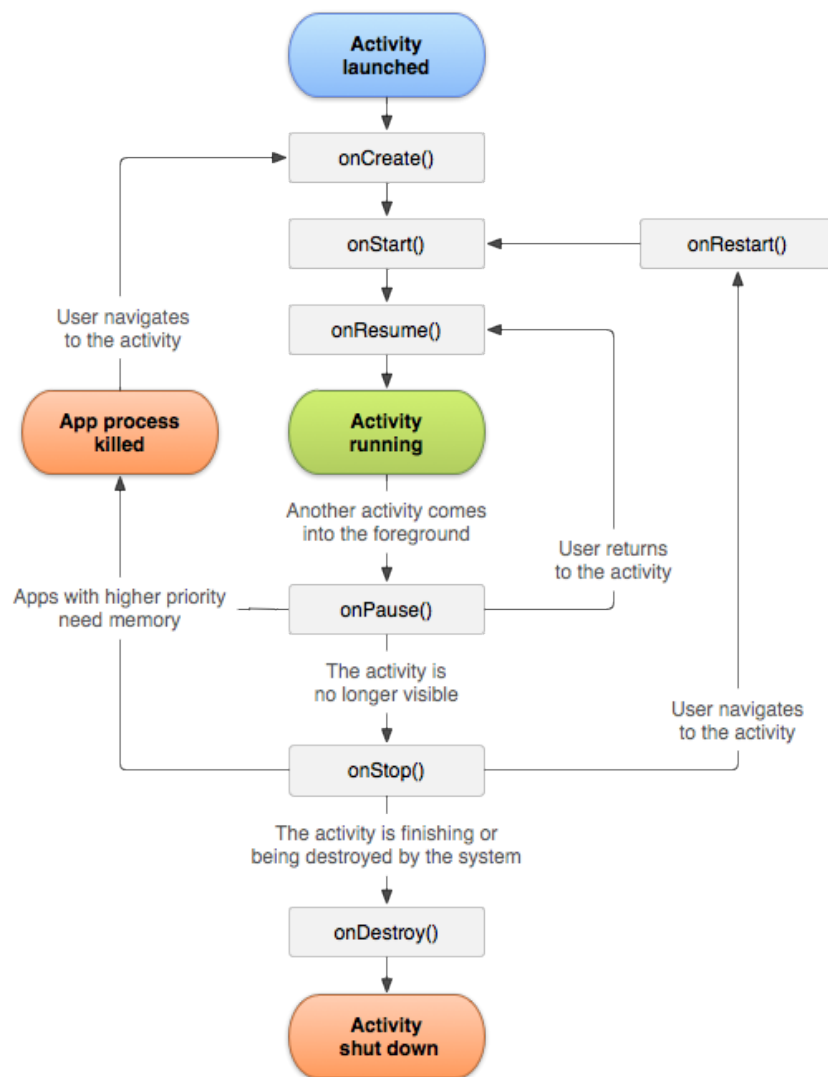


Figura 9.5: Cicle de vida genèric de qualsevol Activity

⁴CV. [en línia] http://developer.android.com/images/activity_lifecycle.png [data de consulta: 16/04/2014]

A continuació es fa una breu descripció de cada classe de tipus *Activity*:

Classe	Descripció
MainActivity	Mostra la pantalla d'entrada a l'aplicació
PuzleSelection_Activity	Mostra les diferents opcions de selecció d'un puzle
EducationalPuzle_Activity	Mostra una llista amb els puzles educatius disponibles
PhotographyPuzle_Activity	Mostra la càmera de fotos del dispositiu i permet fotografiar.
GalleryPuzle_Activity	Mostra la galeria d'imatges del dispositiu on es selecciona una imatge
SelectedSaveGame_Activity	Mostra les partides guardades que el jugador pot seleccionar
AcceptImage_Activity	Pantalla d'acceptació i rebuig de la imatge preseleccionada
PuzleConfiguration_Activity	Mostra al jugador la opció d'escollir el nombre de peces del puzle
PlayPuzle_Activity	Mostra la pantalla de joc
SavedGame_Activity	Pantalla en què el jugador guarda l'estat d'una partida

Taula 9.4: Descripció de classes de la Vista

A continuació s'explica amb més detall les classes que no són de tipus *Activity*: *PlayerSelectedValues*, *CanvasPiece* i *PainCanvasPieces*.

La classe *PlayerSelectedValues* conté els següents mètodes:

Mètode	Descripció
PlayerSelectedValues	Constructor de la classe
setIdPuzle	Actualitza l'identificador del puzle seleccionat
setIdSavedGame	Actualitza l'identificador de la partida guardada seleccionada
setPathFile	Actualitza el nom de l'arxiu de la imatge seleccionada
setDimensions	Actualitza les dimensions seleccionades del puzle
setPuzleType	Actualitza el tipus de puzle seleccionat
setPiecesPosition	Actualitza la posició de les peces amb el contingut de la partida guardada.
getIdPuzle	Llegeix l'identificador del puzle seleccionat
getIdSavedGame	Llegeix l'identificador de la partida guardada seleccionada
getPathFile	Llegeix el nom de l'arxiu de la imatge seleccionada
getDimensions	Llegeix les dimensions seleccionades del puzle
getPuzleType	Llegeix el tipus de puzle seleccionat
getPiecesPosition	Llegeix la posició de les peces amb el contingut de la partida guardada.
getAcceptedRectLeft	Llegeix la coordenada de l'esquerra del requadre de retall de la imatge
setAcceptedRectLeft	Actualitza la coordenada de l'esquerra del requadre de retall de la imatge
describeContent	Mètode necessari per implementar la interfície <i>Parcelable</i>
writeToParcel	Mètode necessari per implementar la interfície <i>Parcelable</i>
CREATOR	Mètode necessari per implementar la interfície <i>Parcelable</i>

Taula 9.5: Mètodes de *PlayerSelectedValues*

La classe *PlayerSelectedValues* s'utilitza per anar guardant allò que el jugador va seleccionant en cadascuna de les activitats que visita abans de començar a jugar. Com que aquesta classe s'ha d'anar intercanviant entre totes les activitats, necessita implementar la interfície *Parcelable*⁵ (és una de les maneres en què les activitats s'intercanvien informació).

⁵Parcelable. [en línia] <http://developer.android.com/reference/android/os/Parcelable.html> [data de consulta: 16/04/2014]

La classe CanvasPiece conté els següents mètodes:

Mètode	Descripció
CanvasPiece	Constructor de la classe
setPieceBmp	Actualitza el bitmap associat a una peça del puzle
getPieceBmp	Llegeix el bitmap associat a una peça del puzle
setIdPieceBmp	Actualitza l'identificador associat a una peça del puzle
getIdPieceBmp	Llegeix l'identificador associat a una peça del puzle

Taula 9.6: Mètodes de CanvasPiece

La classe PaintCanvasPieces conté els següents mètodes:

Mètode	Descripció
PaintCanvasPieces	Constructor de la classe
loadImage	Llegeix el contingut de l'arxiu que s'utilitza com a puzle
cutInPieces	Talla a trossos la imatge per crear les peces del puzle
setIdPieceBmp	Actualitza l'identificador associat a una peça del puzle
onDraw	Mostra en pantalla l'estat actual del puzle
onTouch	Associa les coordenades de pantalla a cada peça del puzle
resizeBitmaps	Escalat de les peces del puzle
interchangePieces	Intercanvia la posició de dos bitmaps del puzle

Taula 9.7: Mètodes de PaintCanvasPieces

9.4.3 Classes del Controlador

Les classes que pertanyen al Controlador són les encarregades d'implementar la lògica de funcionament del puzle.

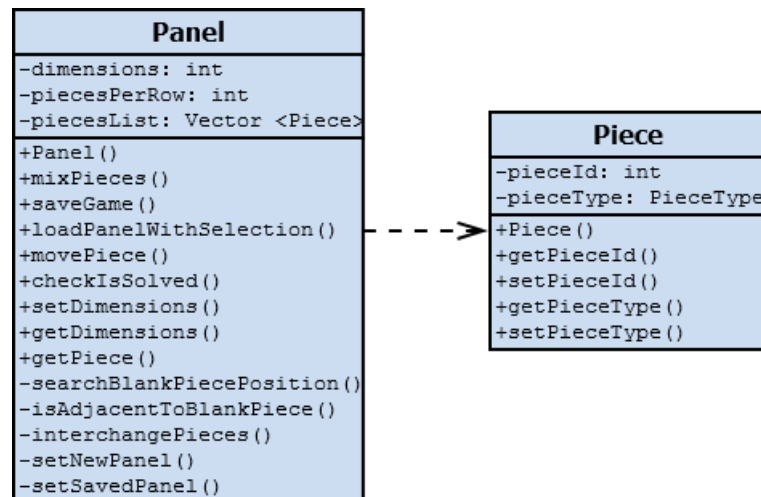


Figura 9.6: Classes del Controlador

A continuació s'expliquen els mètodes de la classe Panel:

Mètode	Descripció
Panel	Constructor de la classe
mixPieces	S'encarrega de desordenar les peces del panell
saveGame	Guarda l'estat actual de les peces del panell
loadPanelWithSelection	Carrega el panell tenint en compte la selecció del jugador
movePiece	Intenta moure una peça.
checkIsSolved	Comprova si la distribució actual de les peces correspon a la solució
setDimensions	Actualitza les dimensions del puzle
getDimensions	Llegeix les dimensions del puzle
getPiece	Llegeix la posició d'una peça del puzle
searchBlankPiecePosition	(Privat) Cerca la posició de l'espai buit al panell
isAdjacentToBlankPiece	(Privat) Mira si una peça es pot moure
interchangePieces	(Privat) Intercanvia la posició d'una peça i l'espai buit
setNewPanel	(Privat) Crea un puzle nou
setSavedPanel	(Privat) Crea un puzle amb el contingut d'una partida guardada.

Taula 9.8: Mètodes de Panel

A continuació s'expliquen els mètodes de la classe Piece:

Mètode	Descripció
Piece	Constructor de la classe
setPieceId	Actualitza l'identificador de la peça
getPieceId	Llegeix l'identificador de la peça
setPieceType	Actualitza el tipus de peça
getPieceType	Llegeix el tipus de peça

Taula 9.9: Mètodes de Piece

9.4.4 Altres definicions

A banda de les classes esmentades, s'han definit alguns enumerats que són utilitzats per algunes d'aquestes classes indicades.

PieceType: Enumerat.

- *NOT_PIECE*: Peça que no es pot utilitzar en el joc
- *BLANK_PIECE*: Peça especial que conté l'espai buit per poder moure les peces del puzle
- *NORMAL_PIECE*: Peça que conté una part de la imatge del puzle

PuzzleType: Classe que conté enumerats.

- *NULL_TYPE*: Puzle no vàlid
- *EDUCATIONAL_TYPE*: Puzle amb informació relacionada amb la imatge
- *PHOTOGRAPHY_TYPE*: Puzle creat a partir d'una fotografia
- *GALLERY_TYPE*: Puzle creat a partir d'una imatge de la galeria del dispositiu

- *SAVEDGAME_EDUCATIONAL*: Puzle creat a partir d'una partida guardada anteriorment de tipus EDUCATIONAL_TYPE
- *SAVEDGAME_PHOTOGRAPHY*: Puzle creat a partir d'una partida guardada anteriorment de tipus PHOTOGRAPHY_TYPE
- *SAVEDGAME_GALLERY*: Puzle creat a partir d'una partida guardada de tipus GALLERY_TYPE

Capítol 10

Prototip

En aquest capítol es mostra el prototip utilitzat durant el desenvolupament de l'aplicació. Aquest prototip ajuda a entendre el funcionament de l'aplicació abans de tenir els detalls d'implementació i de tenir la pròpia aplicació.

El prototip de l'aplicació **PuzleFotoApp** ha sigut realitzat amb l'aplicació *Evolus Pencil*¹. Es tracta d'una eina gratuïta, de codi font obert, disponible per la majoria de sistemes operatius d'ús general, raó per la qual s'ha escollit davant d'altres aplicacions com *WireFrameSketcher*².

Abans d'explicar en detall el prototip, cal dir que les pantalles utilitzades a l'aplicació real poden ser lleugerament diferents, però la funcionalitat final ha de ser la mateixa.

¹Evolus Pencil.[en línia] <http://pencil.evolus.vn/> [data de consulta: 23/03/2014]

²WireFrameSketcher.[en línia]<http://wireframesketcher.com> [data de consulta: 23/03/2014]

10.1 Pantalla d'inici



Aquesta pantalla actua com a pantalla d'entrada en l'aplicació.

La funció d'aquesta pantalla es redueix a presentar l'aplicació al jugador abans de començar a jugar.

10.2 Pantalla de selecció de puzle



Aquesta pantalla actua com un menú i mostra les diferents opcions de selecció de puzles:

- Puzle educatiu incorporat per l'aplicació
- Fer una fotografia amb la càmera del dispositiu per fer-la servir en el puzle
- Seleccionar una imatge emmagatzemada al dispositiu
- Seleccionar una de les partides desades

10.2.1 Puzle educatiu



Aquesta pantalla mostra els diferents puzles educatius que l'aplicació porta incorporats.

L'usuari ha de seleccionar una de les imatges que es mostren en pantalla. (Les imatges presentades no són les definitives, s'utilitzen només per mostrar com podria ser la disposició dels elements d'aquesta pantalla)

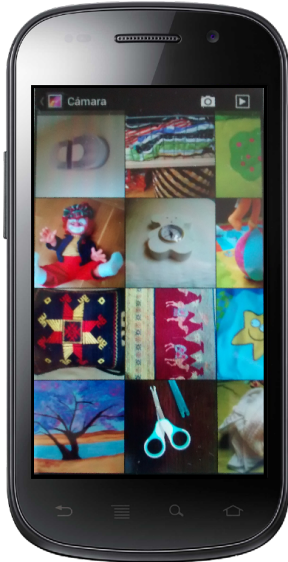
10.2.2 Fer fotografia



L'aplicació permet personalitzar els puzles fent servir imatges capturades amb la càmera del dispositiu.

Aquesta pantalla mostra l'aplicació fotogràfica que porta incorporat el dispositiu.

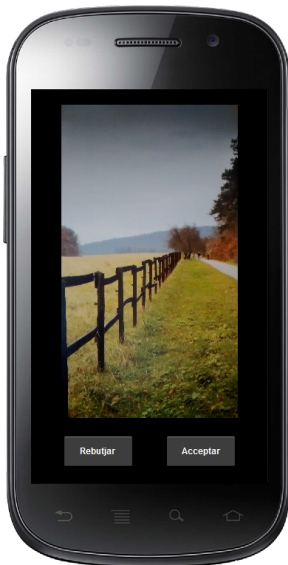
10.2.3 Galeria d'imatges



Aquesta pantalla mostra l'aplicació de galeria d'imatges que porta incorporat el dispositiu

L'usuari podrà escollir qualsevol imatge que estigui emmagatzemada en el dispositiu per fer-la servir en el puzle.

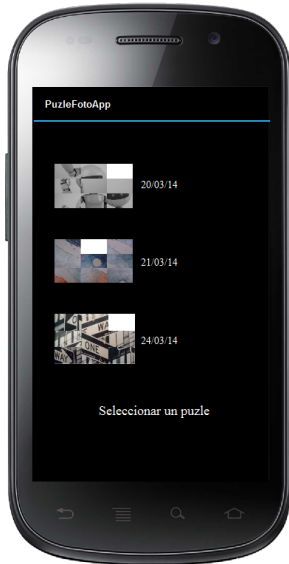
10.2.4 Acceptar imatge



Aquesta pantalla mostra la fotografia que l'usuari ha capturat amb l'objectiu d'utilitzar-la en el puzle personalitzat. Aquesta pantalla també pot mostrar la imatge que l'usuari ha seleccionat de la galeria d'imatges del dispositiu.

Si l'usuari està satisfet amb la fotografia o la imatge que es mostra en pantalla ha de pulsar el botó *Acceptar*. En cas contrari, haurà de pulsar el botó *Rebutjar*.

10.2.5 Partides desades



Aquesta pantalla mostra les partides que l'usuari ha desat prèviament.

L'usuari podrà escollir qualsevol partida de les que hi ha desades (amb la disposició de peces que hi havia en el moment en què es va desar la partida) per continuar jugant.

10.3 Pantalla de configuració del puzle



Aquesta pantalla mostra les diferents dimensions que podrà tenir un puzle.

Cada dimensió té associat un nivell de dificultat diferent. La configuració més senzilla és la de 2x2 i la més complexa és la de 4x4.

El jugador haurà d'escollir una configuració tenint en compte el grau de dificultat que vol assumir.

10.4 Pantalla de joc

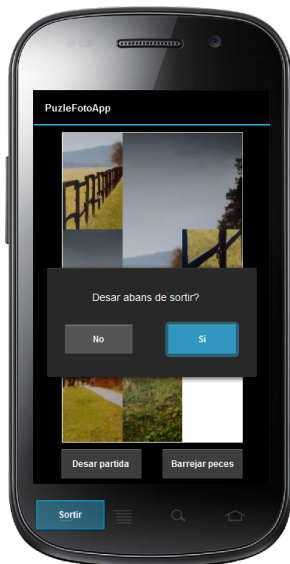


A la part inferior d'aquesta pantalla es mostren els botons *Desar partida*, el botó *Barrejar peces* i el botó *Sortir* (botó del dispositiu que aquí s'ha marcat com un botó gris per remarcar-ho).

El jugador haurà de polsar el botó *Barrejar peces* abans de començar a jugar. A continuació, haurà de moure les peces que envolten l'espai buit (verticalment i horitzontalment) per ordenar les peces i així resoldre el puzle.

Abans de resoldre el puzle l'usuari podrà guardar l'estat actual de la partida i podrà sortir del joc en qualsevol moment.

10.5 Pantalla per sortir del joc



Aquesta pantalla mostra una finestra emergent a sobre de la pantalla de joc.

Les opcions que s'ofereixen a l'usuari són:

- sortir del joc, guardant la disposició actual de les peces (per poder continuar més endavant)
- sortir del joc, sense guardar la disposició actual de les peces (ja no es podrà reprendre aquesta partida)

Capítol 11

Implementació

En aquest capítol es descriuen les principals decisions que s'han pres a l'hora d'implementar l'aplicació **PuzleFotoApp**. Degut a les característiques específiques d'un sistema operatiu com Android i a les limitacions dels dispositius mòbils, moltes decisions s'han escollit tenint en compte els seus punts forts i evitant els seus punts febles. En conseqüència, una aplicació similar per PC (per exemple) hauria tingut una implementació diferent a la que s'ha realitzat en aquest projecte (encara que el llenguatge podria ser Java en tots dos casos).

Abans d'entrar a detallar la implementació realitzada, cal dir que en tot moment s'ha tingut present que l'aplicació s'executarà en un dispositiu amb una interfície reduïda. És a dir, pantalles d'entre 3,5 i 5 polsades en el cas de mòbils o de fins a 10 polsades en cas de les tauletes. Per aquesta raó, un dels esforços principals ha sigut intentar crear unes interfícies que s'adaptin de manera adequada a tots aquests dispositius.

Un altre punt clau a l'hora d'implementar l'aplicació, ha sigut la memòria del dispositiu. Moltes decisions han anat enfocades a utilitzar les eines que dona Android per tal d'optimitzar la memòria que requereix l'aplicació. En aquest sentit, cal dir que Android ofereix solucions adequades a la majoria de situacions de manera més o menys senzilla. En el cas concret de la visualització d'imatges en el dispositiu, ha sigut necessari un tractament específic de les imatges tenint en compte la diversitat de mesures de les fotografies actuals (3, 5, 8, 10 i fins i tot 20 Mpx). Aquestes imatges necessiten un aprovisionament massiu de memòria que acaben provocant els temuts "OutOfMemory". Però fent un ús adequat de les llibreries es poden arribar a fer servir imatges molt més grans sense cap tipus de problema¹.

L'explicació dels detalls d'implementació del projecte es realitzaran tenint en compte l'arquitectura de l'aplicació, és a dir: el Model, la Vista i el Controlador. A més, cal

¹Per comprovar la robustesa de la solució emprada s'ha provat una imatge de 80Mpx. i s'ha pogut fer servir com a puzle sense errors d'exhauriment de memòria.

dir que la descripció de la implementació es centra en els detalls propis de l'aplicació **PuzleFotoApp**, deixant de banda determinats aspectes de les llibreries d'Android i altres detalls que envolten la implementació.

D'entrada cal presentar tots els elements utilitzats durant el desenvolupament. En la imatge següent es mostra l'arbre del projecte en Eclipse:

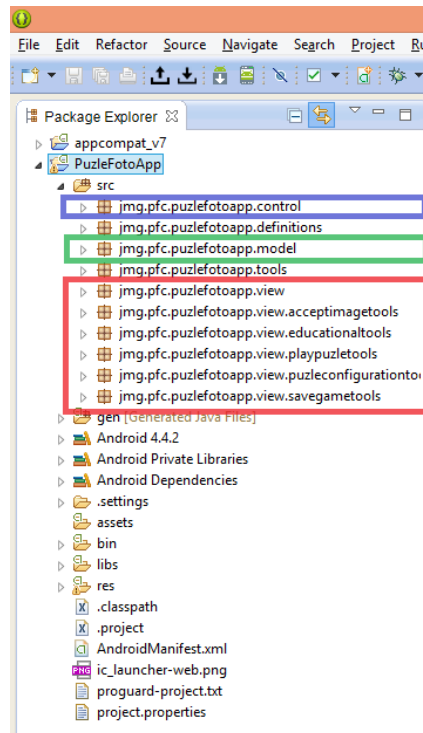


Figura 11.1: Arbre del projecte

A la Figura 11.1 es mostren els principals *packages* creats en l'aplicació. En color vermell es marquen els paquets associats a la vista; en color verd els paquets associats al model i en color blau els paquets associats al controlador. La resta de paquets són d'ús general i es fan servir dins dels paquets marcats en vermell, blau i verd. Aquesta distribució de paquets coincideix amb el que s'ha decidit en l'etapa de disseny, però classifica de manera més detallada les classes que es van indicar i s'afegeixen algunes de noves que s'han considerat d'utilitat durant la implementació.

Des del punt de vista de la utilització dels paquets, cal destacar el paquet view on es defineixen les classes amb les pantalles de l'aplicació. Des d'aquest paquet es desencadena la utilització de la resta de paquets de l'aplicació. En la figura següent s'han representat les seves interaccions:

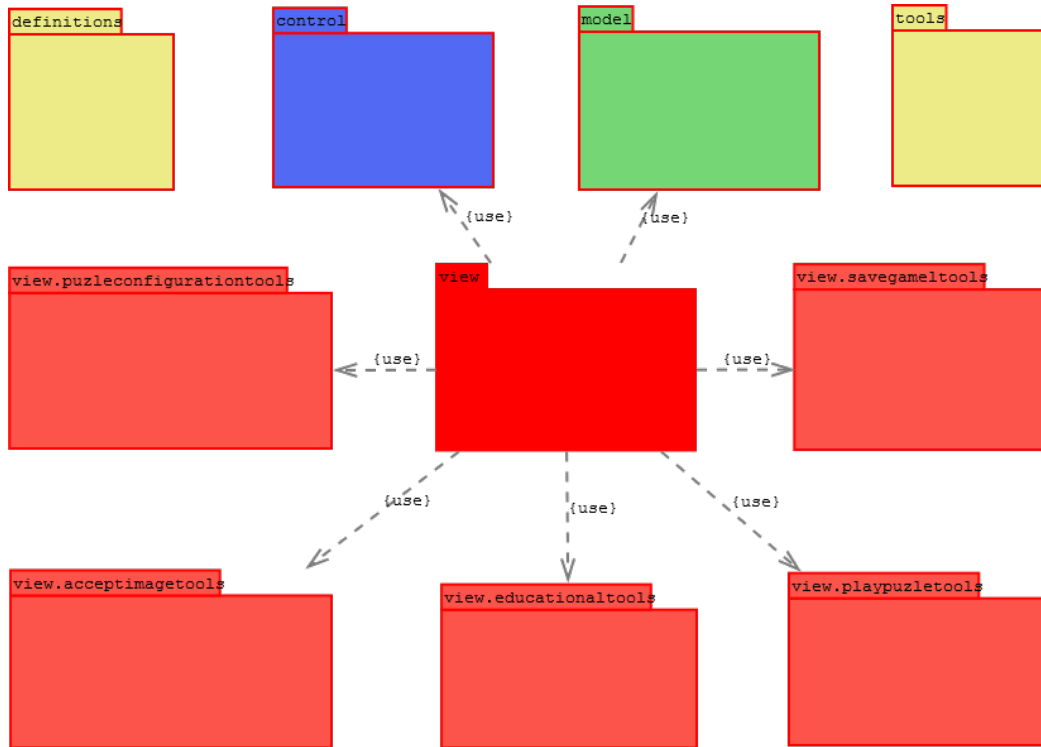


Figura 11.2: Relació entre paquets

Per simplificar la Figura 11.2 no s'han dibuixat les relacions entre els paquets *definitions* i *tools* amb la resta, ja que es considera que es poden fer servir des de tots ells.

Els apartats següents descriuen amb més detall tots els paquets de l'aplicació.

11.1 Model

El paquet **jmg.pfc.puzlefotoapp.model** conté les classes relacionades amb el model:

- **PuzleFotoSQLite**: Conté les instruccions per inserir, actualitzar i eliminar registres en la base de dades.
- **PuzleFotoSQLiteOpenHelper**: Conté el codi font relacionat amb la creació i connexió amb la base de dades.

La base de dades utilitzada en l'aplicació és SQLite, un gestor de base de dades que utilitza el llenguatge SQL. Aquesta base de dades ofereix la potencia suficient per la majoria de les aplicacions implementades per dispositius mòbils.

L'aplicació **PuzleFotoApp** fa servir una taula en la base de dades d'SQLite per emmagatzemar tota la informació necessària per poder continuar jugant una partida que s'ha desat prèviament.

Les llibreries d'Android que donen accés a SQLite permeten fer servir instruccions directes d'SQL (raw SQL) i instruccions mitjançant mètodes específics. En el projecte s'han fet servir principalment instruccions directes d'SQL, encara que també s'ha fet servir alguna de l'altre tipus per comprovar les dues tècniques. A continuació es mostra un exemple de cada tipus:

```
1 // Esborra la taula si existeix
2 db.execSQL("DROP TABLE IF EXISTS SavedGame");
3 // Esborra un registre de la taula SavedGame
4 db.delete("SavedGame", "idSavedGame" + "=" + idSavedGame, null);
```

La cadena de text SQL per la creació de la taula és mostra a continuació:

```
1 // String per crear la taula
2 private static final String SQL_CREATE_TABLE_SAVEDGAME =
3 "CREATE TABLE SavedGame (idSavedGame INTEGER PRIMARY "
4 + "KEY AUTOINCREMENT,"
5 + " idPuzleEducational INTEGER,"
6 + " dimensions INTEGER, piecePosition TEXT,"
7 + " pathFile TEXT, puzzleType INTEGER, "
8 + " educationalTitle TEXT, educationalMessage TEXT,"
9 + " puzledate TIMESTAMP NOT NULL DEFAULT current_timestamp,"
10 + " aRL INTEGER, aRR INTEGER, aRT INTEGER, aRB INTEGER)";
```

SQLite permet decidir com actualitzar la base de dades si es modifica el número de versió. En aquest cas s'ha decidit esborrar el seu contingut, ja que aquesta qüestió queda fora de l'abast dels requisits definits. A continuació es mostra el codi utilitzat:

```
1 if (lastVersion!=newVersion)
2 {
3     // Esborra la taula
4     db.execSQL("DROP TABLE IF EXISTS SavedGame");
5     // Torna a crear la taula
6     onCreate(db);
7 }
```

L'aplicació **PuzleFotoApp** permet la inserció de nous registres en la taula de partides desades, però l'esborrat no es va preveure en l'etapa de disseny. Per evitar

que la taula creixi *ad infinitum*, s'esborraran tots aquells registres associats a puzles resolts, mitjançant una crida a la següent funció:

```
1 public void delSavedGame(int idSavedGame)
2 {
3     open();
4     database.delete("SavedGame", "idSavedGame" + "=" +
5         idSavedGame, null);
6     close();
7 }
```

A més, cada vegada que es llegeix tot el contingut de la base de dades per mostrar-ho en el llistat de partides desades es fa la següent comprovació:

- Ha d'existir l'arxiu de la imatge del puzle que es va desar. En cas contrari es borra el registre de la base de dades.

D'aquesta manera, si l'usuari decideix esborrar del dispositiu una imatge que es feia servir en una partida desada, l'aplicació no intentarà carregar una imatge que ja no existeix.

En relació amb la inserció i l'actualització dels registres de la base de dades, cal comentar que s'ha realitzat sense l'ús de transaccions. Això és degut a la senzillesa de les instruccions SQL utilitzades (no hi ha la necessitat de realitzar accions sobre taules diferents de manera atòmica).

11.2 Vista

Els elements que formen la vista de l'aplicació es poden dividir en dos parts:

- Classes i paquets: En aquesta part es descriuen les característiques de les classes utilitzades i la seva distribució en paquets.
- Recursos i XML: En aquesta part es descriuen els arxius XML utilitzats per dissenyar l'aspecte de les diferents pantalles i per configurar alguns aspectes de l'aplicació. A més es descriuen les ubicacions dels recursos (imatges, sons, etc.) que s'han utilitzat en l'aplicació.

11.2.1 Classes i paquets

El paquet `jmg.pfc.puzlefotoapp.view` conté les classes relacionades amb la vista en el patró de disseny MVC. Aquest paquet conté altres paquets que s'explicaran més endavant. Les classes del paquet view són:

- `About_Activity`: Pantalla d'informació sobre l'aplicació

- `AcceptImage_Activity`: Pantalla per acceptar o rebutjar la imatge que s'utilitzarà en el puzle.
- `EducationalPuzle_Activity`: Pantalla per escollir un puzle de tipus educatiu.
- `GalleryPuzle_Activity`: Classe per seleccionar una imatge de la galeria que es farà servir com a puzle.
- `Main_Activity`: Pantalla d'inici que es mostra al entrar en l'aplicació.
- `PhotographyPuzle_Activity`: Classe per realitzar una fotografia que es farà servir com a puzle.
- `PlayerSelectedValues`: Aquesta classe emmagatzema les diferents opcions seleccionades per l'usuari en totes les activitats de l'aplicació. El contingut d'aquesta classe s'utilitza per omplir un registre de la taula de partides desades en la base de dades. Aquesta classe s'ha ubicat en aquest paquet però també s'hauria pogut ubicar dins del paquet del model.
- `PlayPuzle_Activity`: Pantalla de joc en la que l'usuari pot resoldre el puzle.
- `PuzleConfiguration_Activity`: Pantalla per seleccionar les dimensions del puzle.
- `PuzleSelection_Activity`: Pantalla per seleccionar el tipus de puzle.
- `SaveGame_Activity`: Pantalla informativa que es mostra un cop s'ha desat la partida.
- `SelectSavedGame_Activity`: Pantalla per seleccionar una partida desada prèviament.

Totes aquestes classes (excepte `PlayerSelectedValues`) implementen les interfícies o pantalles que l'usuari veu al fer servir l'aplicació. Tenen en comú els següents punts:

Activity: Estenen la classe `Activity`.

Fragment: Internament contenen una classe de tipus `Fragment`. Aquesta classe actua com a 'caixa' on es posa tot el contingut de l'activitat. Encara que en aquest projecte no es treu profit específicament dels fragments, en les darreres versions d'Android és la solució recomanada per defecte (en comptes de treballar directament sobre l'activitat).

onActivityCreated Implementen aquesta funció que és l'estat del cicle de vida del `Fragment` en el que s'ha implementat les accions específiques de cada activitat.

finishActivity Implementen aquesta funció que és específica de cada activitat. S'encarrega d'alliberar els recursos utilitzats. D'aquesta manera s'aconsegueix que el *Garbage Collector* actuï el més aviat possible.

onOptionsItemSelected Aquesta funció configura el menú disponible per cada activitat. S'implementa directament en l'activitat (no en el fragment)

onOptionsItemSelected Aquesta funció dona resposta a la interacció de l'usuari amb el menú de cada activitat. Android ofereix la possibilitat de que la resposta a la interacció amb el menú sigui donada per l'activitat o el fragment que conté la activitat. En aquest cas, s'ha escollit donar resposta des del fragment, fent servir el codi següent en la creació del fragment:

```
1 public void onCreate(Bundle savedInstanceState) {  
2     super.onCreate(savedInstanceState);  
3     // El fragment també accedeix al menú  
4     setHasOptionsMenu(true);  
5 }
```

onBackPressed Aquesta funció implementa el funcionament específic que té cada activitat quan l'usuari ha polsat el botó de sortida (o de retrocés) del dispositiu.

startActivity, startActivityForResult Cada activitat fa ús d'aquestes funcions. La primera d'elles es fa servir quan es vol canviar l'activitat en curs. La segona es fa servir quan es crida a una activitat de la que s'espera obtenir una resposta i un resultat.

Altres paquets continguts dins de la vista i que donen suport a les classes anteriors són: **acceptimagetools**, **playpuzletools**, **puzleconfigurationtools** i **savegame-tools**. A continuació es descriu el seu contingut:

El paquet **jmg.pfc.puzlefotoapp.view.acceptimagetools** conté la classe utilitzada des de la classe `AcceptImage_Activity`:

RectangleImageView Classe per seleccionar una porció d'una imatge. Aquesta classe s'ha implementat com un component. És a dir, s'ha implementat de manera que es pot fer servir en qualsevol aplicació Android com seria el cas dels `RadioButton`, `ProgressBar`, etc. (encara que no s'ha implementat amb aquest objectiu i pot ser requereix certes adaptacions). La funcionalitat principal d'aquest component consisteix en mostrar una imatge i un requadre que l'usuari mou per sobre de la imatge. Cada vegada que es mou, es genera un event que informa de les coordenades del requadre a sobre de la imatge. Un exemple del seu aspecte és:

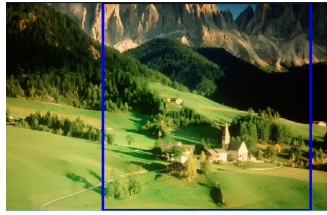


Figura 11.3: RectangleImageView

Les coordenades retornades per aquesta classe es faran servir per retallar la imatge i poder utilitzar-la com a puzle.

El paquet **jmg.pfc.puzlefotoapp.view.educationaltools** conté les classes utilitzades des de la classe `EducationalPuzle_Activity`:

- `AdapterItemEducationalPuzle`: Classe de suport que carrega els elements d'una llista lliscant
- `ItemEducationalPuzle`: Classe que conté el contingut d'un element en una llista lliscant

Aquestes dues classes s'utilitzen de manera conjunta amb el component *ListView*². Aquest component permet implementar una llista lliscant d'elements de qualsevol tipus. En aquest cas, el tipus és *ItemEducationalPuzle*. Com que la memòria del dispositiu és limitada només es carreguen en memòria els elements de la llista que es visualitzen per pantalla en cada moment. Per poder implementar aquest comportament és necessari utilitzar un adaptador que informa a *ListView* com ha de carregar cada element de la llista quan sigui necessari. Amb aquest propòsit es fa servir la classe *AdapterItemEducationalPuzle*.

El paquet **jmg.pfc.puzlefotoapp.view.playpuzletools** conté les classes utilitzades des de la classe `PlayPuzle_Activity`:

- `CanvasPiece`: Conté la informació d'una peça del puzle dibuixada en pantalla
- `EducationalDialog`: Classe que mostra el diàleg relacionat amb la informació educativa d'un puzle. Quan un puzle s'ha resolt es mostra el text associat al puzle educatiu.
- `PaintCanvasPieces`: Classe que gestiona el repintat de les peces del puzle i els esdeveniments generats sobre el puzle.
- `PuzleSounds`: Classe que reproduïx els sons de la pantalla de joc.

²ListView. [en línia] <http://developer.android.com/reference/android/widget/ListView.html> [data de consulta: 24/05/2014]

D'aquest paquet cal destacar la classe `PaintCanvasPieces`. Aquesta classe s'ha implementat com un component. És a dir, s'ha implementat de manera que es pot fer servir en qualsevol aplicació Android com seria el cas dels `RadioButton`, `ProgressBar`, etc. (encara que no s'ha implementat amb aquest objectiu i pot ser requereix certes adaptacions).

La funcionalitat principal de la classe `PaintCanvasPieces` consisteix en dibuixar les peces que formen el puzle i en donar resposta a les pulsacions de l'usuari. Aquesta tasca la realitza tenint en compte que ha d'adaptar les dimensions de la imatge original amb les dimensions del tauler del puzle (sense distorsionar la imatge). A més, és l'encarregat de trossejar la imatge per generar les diferents peces del puzle. Quan el puzle s'ha resolt es genera un event indicant aquesta situació. L'aspecte d'aquest component és:

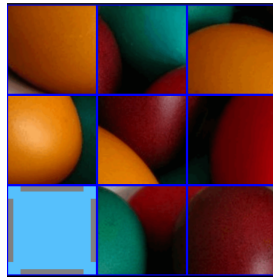


Figura 11.4: `PaintCanvasPieces`

El paquet `jmg.pfc.puzlefotoapp.view.puzleconfigurationtools` conté les classes utilitzades des de la classe `PuzleConfiguration_Activity`:

- `AdapterItemPuzleConfiguration`: Classe de suport que carrega els elements d'una llista lliscant
- `ItemPuzleConfiguration`: Classe que conté el contingut d'un element en una llista lliscant

L'objectiu d'aquestes classes és el mateix que l'explicat en el paquet *educational-tools*, és a dir, donar suport a la `ListView` que es fa servir per mostrar les diferents dimensions de puzle que l'usuari pot escollir.

El paquet `jmg.pfc.puzlefotoapp.view.savegametools` conté les classes utilitzades des de la classes `SaveGame_Activity`, `PlayPuzle_Activity` i `SelectSavedGame_Activity`. Totes aquestes classes estan relacionades amb la gestió de les partides guardades:

- `AdapterItemSavedGame`: Classe de suport que carrega els elements d'una llista lliscant

- `AlertDialog`: Classe que mostra un diàleg oferint la possibilitat de desar si es vol sortir de la pantalla de joc.
- `SavedGameDialog`: Classe que mostra un diàleg indicant que la partida ha sigut desada.

11.2.2 Recursos i XML

A més de les classes i els paquets cal mencionar els recursos i els arxius XML utilitzats. En la figura 11.5 es pot veure la ubicació dels elements que s'expliquen a continuació:

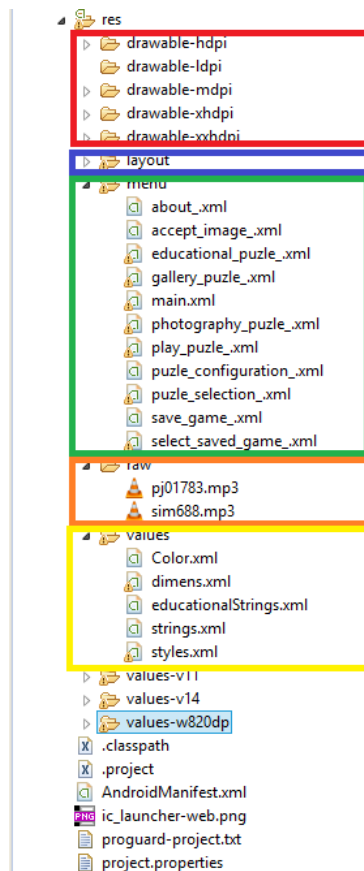


Figura 11.5: Arbre de recursos i XML

layout En el requadre de color blau de la figura es mostra el directori on es troben els arxius XML de configuració de les diferents pantalles de l'aplicació. Dins d'aquest directori també es defineix com es mostren els elements de les llistes lliscants, amb arxius XML independents.

drawable En el requadre de color vermell de la figura es mostren els directoris on s'ubiquen les imatges utilitzades en els puzles educatius. A més, es defineixen els *selectors* per canviar una imatge quan l'usuari ha polsat un botó amb imatge. Es

defineixen selectors per tots els botons amb imatge. Com a exemple, es mostra el selector associat al botó de la càmera:

```
1 <selector xmlns:android="http://schemas.android.com/apk/
  res/android">
2 <item android:state_pressed="false"
3       android:drawable="@drawable/camera"/>
4 <item android:state_pressed="true"
5       android:drawable="@drawable/camera_alt" />
6 </selector>
```

menu En el requadre de color verd es mostren els arxius XML utilitzats per definir els menús que es mostren en cada activitat. És a dir, cada activitat defineix un XML propi on s'indiquen les opcions de menú que gestiona.

raw En el requadre de color taronja es mostren els arxius 'mp3' que es fan servir en l'aplicació.

values En el requadre de color groc es mostren diferents arxius per definir valors que s'utilitzen en les activitats de l'aplicació: colors, dimensions, cadenes de text per cada puzle educatiu, cadenes de text que es mostren a les activitats i estils. El fet de definir les cadenes de text en aquests arxius permet una possible configuració (en el futur) de diferents idiomes dins de l'aplicació (només caldria fer arxius nous amb les traduccions per cada idioma) sense haver de fer grans canvis en el codi.

A banda d'aquests directoris es pot veure que apareixen altre arxius propis per la gestió de projecte en Eclipse i l'arxiu **AndroidManifest.xml**. En el manifest del projecte es defineixen:

- Les activitats del projecte i la orientació que es permet a cadascuna d'elles (Orientació vertical). Com a exemple es mostra la definició de l'activitat inicial:

```
1 <activity
2   android:name=" jmg.pfc.puzlefotoapp.view.MainActivity
3   "
4   android:label="@string/app_name"
5   android:screenOrientation="portrait" >
6   <intent-filter>
7     <action android:name="android.intent.action.MAIN"
8       />
9     <category android:name="android.intent.category.
10    LAUNCHER" />
11   </intent-filter>
12 </activity>
```

- Els usos i permisos de l'aplicació (versió del SDK i permisos per accedir al dispositiu):

```
1 <uses-sdk android:minSdkVersion="8"
2         android:targetSdkVersion="19" />
3
4 <uses-permission android:name="android.permission.
5     READ_EXTERNAL_STORAGE" />
6 <uses-permission android:name="android.permission.CAMERA"
7     />
8 <uses-permission android:name="android.permission.
9     WRITE_EXTERNAL_STORAGE" />
10 <uses-permission android:name="android.permission.
11     READ_LOGS" />
12 <uses-feature android:name="android.hardware.camera" />
```

11.3 Controlador

El paquet `jmg.pfc.puzlefotoapp.control` conté les classes relacionades amb el controlador en el patró de disseny MVC.

Panel Classe que conté la estructura de dades del puzle, concretament, una llista d'elements que representen les peces del puzle. En aquesta classe es comproven els moviments que s'accepten depenent de la posició de la peça en blanc.

En la secció d'introducció d'aquest document es va plantejar que la meitat dels puzles possibles no tenen solució. A més, es va plantejar la necessitat de que l'aplicació només oferís puzles que siguin resolubles. Per solucionar aquesta qüestió en la classe Panel, s'ha triat una implementació molt senzilla. Consisteix en començar amb el puzle ordenat i en moure aleatòriament les peces com si el jugador estigués realitzant moviments, per barrejar-les. D'aquesta manera ens estalviem la implementació de l'algorisme que verifica si una configuració aleatòria de peces es pot resoldre. A continuació es mostra una part del codi:

```
1 // El nombre de moviments es aleatori pero depen
2 // de les dimensions del puzle
3 anterior = -5;
4 nMovs=1;
5 if (piecesPerRow==2) {
6     nMovs=r.nextInt (10 - 2) + 2;
7 } else if (piecesPerRow==3) {
8     nMovs=r.nextInt (50 - 30) + 30;
9 } else if (piecesPerRow==4) {
10    nMovs=r.nextInt (90 - 70) + 70;
11 }
```

```
12
13 // Un cop que s'ha decidit el nombre de moviments
14 // s'apliquen sobre el puzle resolt.
15 blank=searchBlankPiecePosition();
16 for (i=0;i<nMovs;i++)
17 {
18     do {
19         aleat1 = r.nextInt (max - min) + min;
20     } while ((aleat1==anterior) || (!isAdjacentToBlankPiece(
21         aleat1,blank)));
22     anterior = blank;
23     interchangePieces(aleat1, blank);
24     blank = aleat1;
25 }
```

Piece Classe que conté la estructura de dades d'una peça del puzle. Cada objecte d'aquest tipus conté la informació del tipus de peça i l'identificador de la peça. Aquesta classe no té cap altre funcionalitat que la de contenir la informació esmentada.

11.4 Altres classes de suport

Com a classes d'ús general es fan servir:

LogTool Classe per copiar el contingut del log de l'aplicació en un arxiu en el dispositiu. D'aquesta manera es pot consultar el log de l'aplicació quan s'executa sobre el dispositiu (no el simulador). A partir de la versió 4 d'Android no es poden capturar els logs de l'aplicació (a no ser que l'usuari tingui permisos de root sobre el dispositiu) així que ha de ser la pròpia aplicació la que publiqui el seu log. Aquesta classe només s'ha fet servir durant el desenvolupament.

LoadImageTool Classe per carregar de manera eficient les imatges que es fan servir. Tenint en compte que els dispositius cada vegada fan fotografies amb més resolució, cal analitzar cada imatge per fer servir una resolució més reduïda que la resolució original. D'aquesta manera s'optimitza la utilització de la memòria del dispositiu. A continuació es mostra una part del codi que realitza la tasca esmentada:

```
1 // S'analitzen les dimensions de la imatge sense arribar
2 // a carregar-la en memòria
3 BitmapFactory.Options opt = new BitmapFactory.Options();
4 opt.inJustDecodeBounds = true;
5 opt.inPurgeable = true;
6 opt.inInputShareable=true;
```

```
7  BitmapFactory.decodeResource(res, resId, opt);
8  // Es calcula la resolucio mes petita que s'adapta a les
9  // dimensions del contenidor de la imatge.
10 opt.inSampleSize = calculateInSampleSize(opt, reqWidth,
    reqHeight, limit);
11 // Un cop calculades les dimensions desitjades es carrega
12 // la imatge.
13 opt.inJustDecodeBounds = false;
14 Bitmap b = BitmapFactory.decodeResource(res, resId, opt);
```

Capítol 12

Funcionament de la aplicació

En aquest capítol s'explica el funcionament de l'aplicació **PuzleFotoApp**. Encara que les seves funcionalitats no són gaire complicades, s'acompanya cadascuna d'elles amb una imatge per aconseguir una millor comprensió del que s'explica.

12.1 Instal·lació

La instal·lació de l'aplicació només requereix executar l'arxiu següent:

- /PuzleFotoApp/bin/PuzleFotoApp.apk

Habitualment, els dispositius amb Android tenen una protecció per impedir la instal·lació d'aplicacions que procedeixen d'origens desconeguts. Per tant, si es vol instal·lar qualsevol aplicació a partir d'un arxiu *.apk* s'haurà de modificar la configuració de seguretat del dispositiu, marcant la opció d'origens desconeguts, com es mostra a la imatge següent:

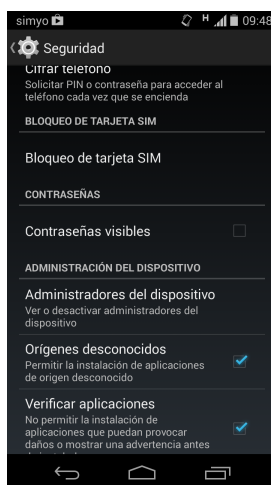


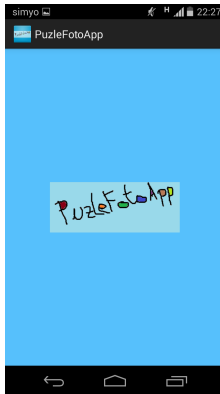
Figura 12.1: Configuració origen desconegut

12.2 Pantalles

Les funcionalitats que ofereix l'aplicació es poden ordenar segons la pantalla en la que es mostren a l'usuari. Abans d'explicar el funcionament de l'aplicació s'enumeren les pantalles que veu l'usuari de l'aplicació:

1. Pantalla d'inici (P1)
2. Pantalla de selecció de tipus de puzle (P2)
3. Pantalla de selecció de puzle educatiu (P3)
4. Pantalla de selecció de puzle a partir d'una fotografia (P4)
5. Pantalla de selecció de puzle a partir d'una imatge de la galeria (P5)
6. Pantalla de selecció de puzle a partir d'una partida desada anteriorment (P6)
7. Pantalla d'acceptació del puzle personalitzat (P7)
8. Pantalla de selecció de dificultat del puzle (P8)
9. Pantalla d'informació sobre l'aplicació (P9)
10. Pantalla de joc (P10)
11. Pantalla de partida desada (P11)
12. Pantalla per desar la partida abans d'abandonar la pantalla de joc (P12)

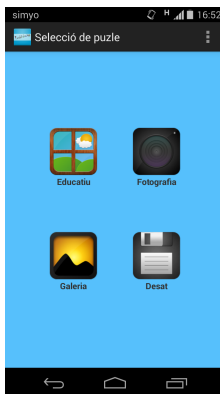
12.2.1 Pantalla d'inici (P1)



Aquesta pantalla és la primera que es mostra quan s'inicia l'execució de l'aplicació.

Es tracta d'una pantalla de presentació on l'usuari no ha de fer res. Al cap de 2 segons, canvia automàticament a la pantalla de selecció de puzle.

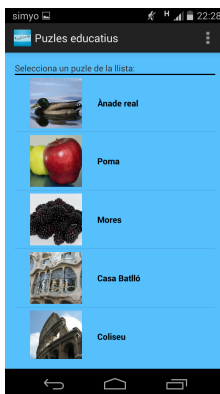
12.2.2 Pantalla de selecció de tipus de puzle (P2)



En aquesta pantalla es mostren les quatre opcions diferents que es poden escollir a l'hora de seleccionar un tipus de puzle: puzle educatiu, fotografia, galeria d'imatges i partides desafes.

Al polsar qualsevol imatge, aquesta canvia lleugerament de color com a retroalimentació cap a l'usuari.

12.2.3 Pantalla de selecció de puzle educatiu (P3)

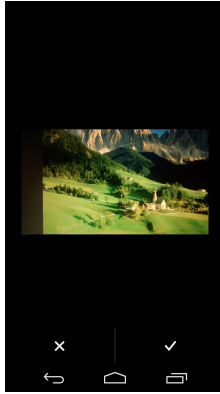


Aquesta pantalla mostra una llista lliscant amb els puzles de tipus educatiu.

Cada element de la llista conté la imatge del puzle i un nom identificatiu del puzle.

L'usuari ha de seleccionar un puzle qualsevol de la llista.

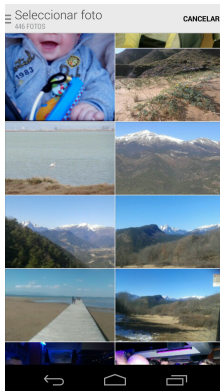
12.2.4 Pantalla de puzle a partir d'una fotografia (P4)



Aquesta pantalla es mostra en l'aplicació un cop que l'usuari ha realitzat la fotografia i haurà d'acceptar-la(V) o rebutjar-la (X)

Cal comentar que aquesta pantalla forma part de l'aplicació de la càmera que porta instal·lat el dispositiu. No ha sigut implementada en aquest projecte, però s'ha integrat dins de l'aplicació **PuzleFotoApp**.

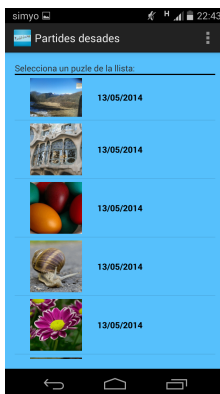
12.2.5 Pantalla de puzle a partir de la galeria (P5)



Aquesta pantalla es mostra en l'aplicació perquè l'usuari seleccioni una imatge.

Cal comentar que aquesta pantalla forma part de l'aplicació de galeria d'imatges que porta instal·lat el dispositiu. No ha sigut implementada en aquest projecte, però s'ha integrat dins de l'aplicació **PuzleFotoApp**.

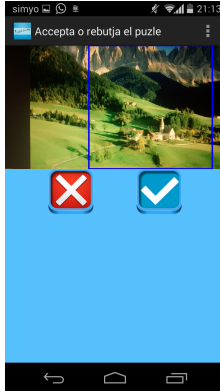
12.2.6 Pantalla de puzle de partides desades (P6)



Aquesta pantalla es mostra en l'aplicació amb l'objectiu de que l'usuari seleccioni una partida desada prèviament.

Es tracta d'una llista lliscant on es mostren les partides que l'usuari a desat prèviament. Cada element de la llista conté la imatge original del puzle i la data en la que es va desar la partida.

12.2.7 Pantalla d'acceptació del puzle personalitzat (P7)



Aquesta pantalla es mostra després de realitzar una fotografia o de seleccionar una imatge de la galeria.

L'usuari pot rebutjar la imatge mostrada si prem el botó vermell(X). En cas contrari haurà de moure el requadre lliscant blau per seleccionar un tros de la imatge fent-lo servir com a puzle. A continuació haurà de polsar el botó blau(V).

12.2.8 Pantalla de selecció de dificultat del puzle (P8)



Aquesta pantalla mostra una llista amb diferents opcions perquè el jugador pugui escollir la dificultat del puzle.

Cada opció de la llista conté una imatge on es veu la disposició de les peces i un text indicant la dificultat del puzle.

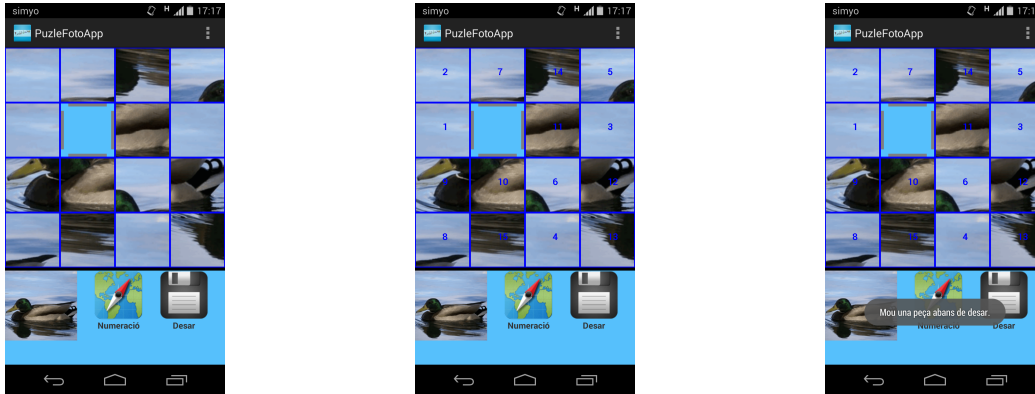
12.2.9 Pantalla d'informació de l'aplicació (P9)



Aquesta pantalla mostra el logotip i informació associada a l'aplicació.

La informació mostrada és: el nom de l'autor i la procedència d'alguns recursos utilitzats per realitzar l'aplicació (icones, sons i imatges dels puzles)

12.2.10 Pantalla de joc (P10)

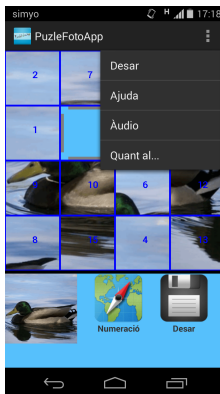


Aquestes imatges mostren algunes de les situacions que l'usuari es troba al interactuar en la pantalla de joc:

- La imatge de l'esquerra mostra el puzle en la part superior amb les peces desordenades. En la part inferior es mostra una imatge reduïda del puzle resolt, una brúixola sobre un mapa (Numeració) i un disc (Desar). La imatge reduïda serveix d'ajuda al jugador per saber on ha d'ubicar cada peça del puzle.
- La segona imatge mostra les peces numerades si es prem a sobre del mapa amb la brúixola (Numeració). Aquesta numeració indica la posició que ha d'ocupar cada peça per resoldre el puzle. S'ha de tenir en compte que el puzle es resoldrà quan les peces estiguin ordenades en ordre creixent d'esquerra a dreta i de dalt a baix. Si es torna a prémer a sobre del mapa amb la brúixola, desapareix la numeració de cada peça.
- La tercera imatge mostra un missatge al jugador: "Moure una peça abans de desar". Aquest missatge es mostra si no s'ha fet cap moviment des de la darrera vegada que es va desar la partida i es torna a prémer el botó Desar. D'aquesta manera s'estalvia desar la partida innecessàriament.

Una altra situació que pot veure el jugador en la pantalla de joc està relacionada amb el menú¹. En general, qualsevol menú dóna accés a determinades accions que es poden realitzar en un determinat context. A continuació s'explica breument cadascuna de les opcions del menú de la pantalla de joc:

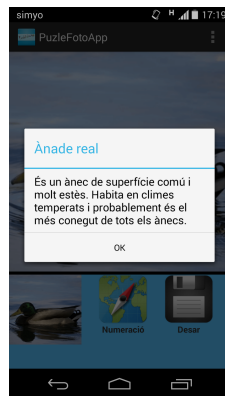
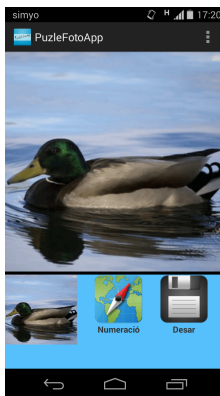
¹El menú es pot identificar a la pantalla com 3 punts verticals a la part superior dreta de la imatge. La representació pot ser diferent depenent de la versió d'Android del dispositiu



El jugador prem el menú que es mostra a la part superior dreta (o la tecla de menú del dispositiu) i es mostren les opcions:

- Desar: per desar la partida.
- Ajuda: Activa/desactiva la numeració
- Àudio: Activa/desactiva els sons del joc.
- Quant al...: Mostra la pantalla d'informació de l'aplicació.

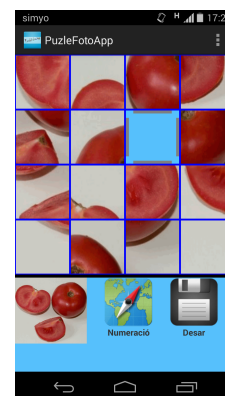
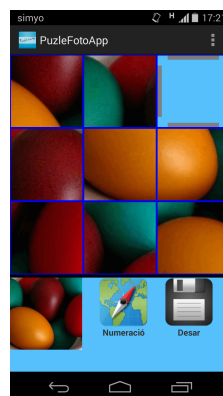
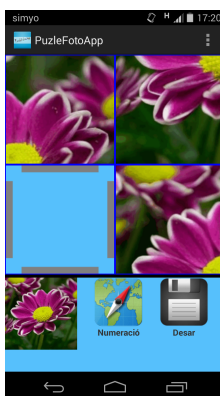
Si s'aconsegueix resoldre el puzle es poden mostrar les pantalles següents:



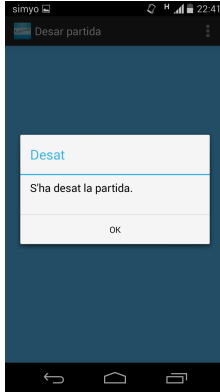
La imatge de l'esquerra mostra el puzle resolt.

La imatge de la dreta mostra el missatge visualitzat al resoldre un puzle educatiu. En la resta de puzles no es mostraria cap missatge.

A continuació es mostren exemples de pantalles de joc per configuracions de puzles de 2×2 , 3×3 i 4×4 :



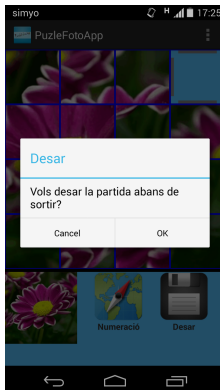
12.2.11 Pantalla de partida desada (P11)



Aquesta pantalla es mostra després de que l'usuari hagi guardat la partida

L'usuari haurà de polsar el botó OK per continuar en la pantalla de joc.

12.2.12 Pantalla per desar partida abans de sortir (P12)



Aquesta pantalla es mostra si el jugador ha polsat el botó per sortir i la configuració actual de la partida no ha estat desada (abans de resoldre).

L'usuari té la opció de desar abans de sortir per no perdre els moviments realitzats en el puzle. En cas contrari no es podrà reprendre la partida en curs.

12.3 Navegació entre pantalles

La descripció de la seqüència de pantalles que pot anar visitant l'usuari pot ser una mica complexa, així que s'ha fet un diagrama en el que es mostren les transicions permeses en l'aplicació. Cal remarcar que:

- La pantalla inicial és P1
- La pantalla del menú principal és P2
- La finalització de l'aplicació s'ha marcat com "Sortida"
- La pantalla de joc és P10

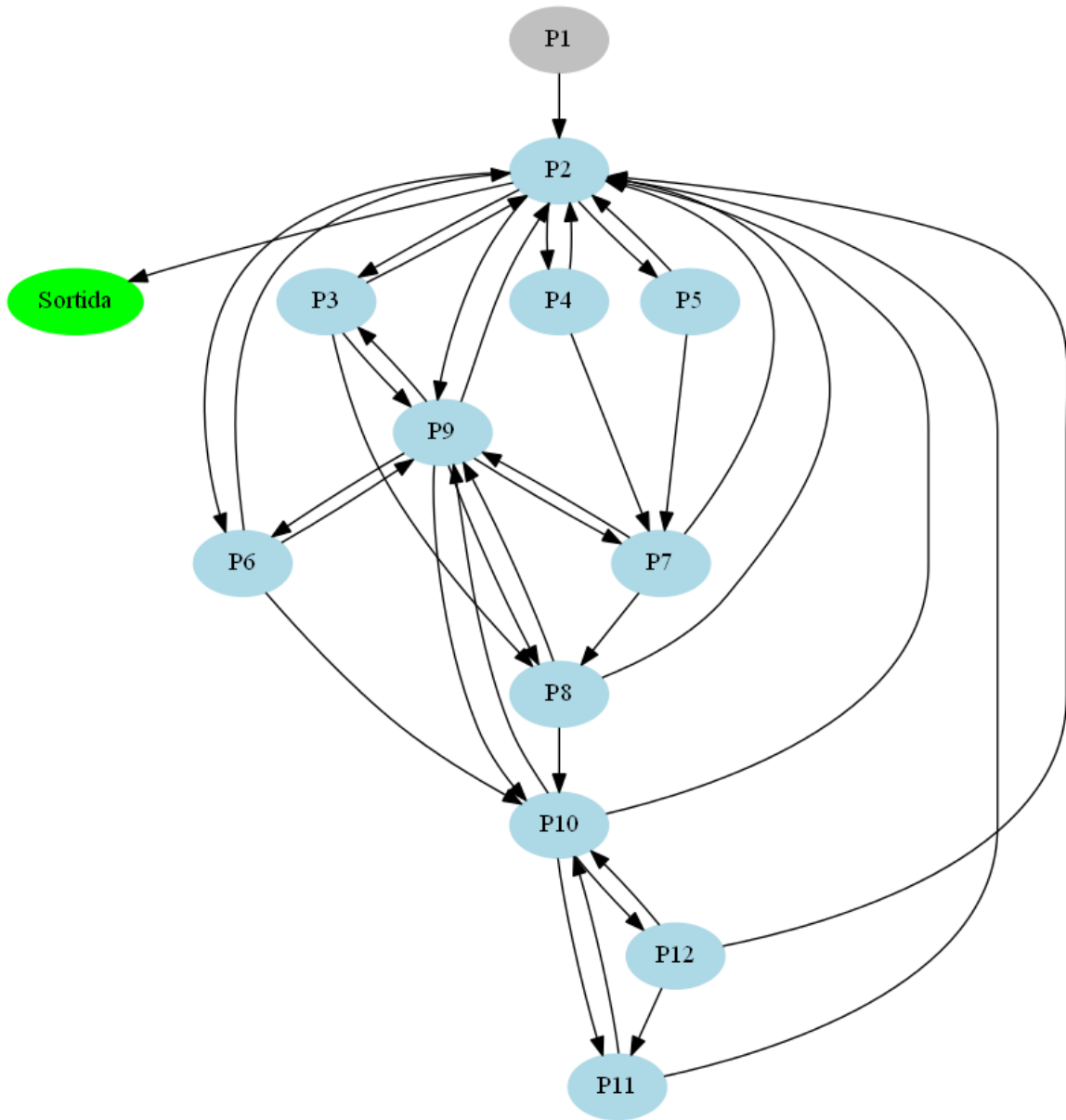


Figura 12.2: Transicions entre pantalles

Capítol 13

Conclusions

La realització del Projecte Final de Carrera ha suposat una gran càrrega de treball, ja que és necessària la realització de molta feina indirecta que acaba repercutint en el projecte. Per exemple:

- **Formació:** Moltes de les eines utilitzades en el projecte no les havia fet servir abans i la fase d'aprenentatge s'ha allargat més del que es va preveure (Llibreries d'Android, L^AT_EX, Camtasia, etc.)
- **Entorn de treball** Encara que l'aplicació realitzada no té un alt grau de complexitat, l'entorn de treball del projecte requereix molt programari que cal provar i configurar per fer-lo servir correctament.
- **Recerca de recursos:** L'aplicació fa servir diferents recursos com imatges, icones, sons... Tots aquests recursos s'han hagut de crear o cercar a Internet. El fet de tenir una formació nul·la en qüestions relacionades amb disseny gràfic ha dificultat molt aquesta tasca.

13.1 Assoliment d'objectius

Per avaluar si s'han assolit o no els objectius del projecte cal veure quins requisits o funcionalitats es van definir en l'etapa de planificació. A continuació es justifica cadascun d'ells:

- Recrear la jugabilitat d'un puzzle lliscant tradicional: s'ha creat una aplicació que permet moure les peces d'un puzzle de manera intuïtiva i sense limitacions de temps o de moviments.
- Oferir puzzles amb caràcter educatiu: s'ofereix la possibilitat de jugar a 20 puzzles educatius diferents que mostren informació relacionada amb la imatge.
- Permetre personalitzar els puzzles a partir d'imatges emmagatzemades al dispositiu: s'ofereix la possibilitat de seleccionar les imatges del dispositiu fent servir l'aplicació 'Fotos' o 'Galeria' del dispositiu. (L'aplicació utilitzada depèn de la versió d'Android i de les preferències d'utilització configurades per cada usuari).

- Permetre personalitzar els puzles a partir de fotografies fetes amb la càmera del dispositiu: s'ofereix la possibilitat de fer servir fotografies capturades durant l'execució de **PuzleFotoApp**
- Permetre desar les partides per jugar en un altre moment: s'ofereix la possibilitat de reprendre qualsevol de les partides desades anteriorment.
- Adquirir les nocions generals de com realitzar aplicacions en Android: encara que dominar totes les possibilitats de programació en Android és una tasca que requereix molt de temps, s'han assolit les nocions bàsiques per poder realitzar projectes de més envergadura.

Tot i que hi diversos punts que es poden millorar, en general s'han assolit tots els objectius que es van definir en l'etapa de planificació del projecte.

13.2 Canvis respecte el disseny inicial

En l'etapa d'anàlisi i disseny de qualsevol aplicació es defineix tot allò que es vol obtenir i com, però depenent de l'experiència personal i de la grandària del projecte, poden sorgir imprevistos o poden sorgir noves idees a l'hora de realitzar determinades tasques planificades.

En una primera etapa de disseny es va preveure l'emmagatzemament dels puzles educatius a la base de dades, però durant l'etapa d'implementació es va veure que és molt més eficient definir-los com a recursos de l'aplicació (en arxius XML, no en la base de dades). Per tant, la base de dades només s'ha fet servir per emmagatzemar les partides desades.

La pantalla d'acceptació de les fotografies realitzades i de les imatges de la galeria, estava pensada inicialment només per acceptar o rebutjar la imatge. Però com que les dimensions de les imatges poden ser molt diferents (depenent del seu origen), es va veure la necessitat de seleccionar només una part de la imatge. Per tant, en aquesta pantalla s'ha afegit la funcionalitat de retallar la imatge mitjançant l'espai ocupat per un requadre que es mostra a sobre de la imatge. D'aquesta manera s'aconsegueix mantenir l'*aspect ratio*¹ de la imatge per evitar les deformacions, ja que el tauler del puzle és quadrat i la imatge original acostuma a ésser rectangular.

Resoldre un puzle, depenent de les seves dimensions i de l'experiència del jugador, pot resultar una tasca frustrant. Per evitar que l'usuari no experimentat estigui temptat a deixar de banda l'aplicació, es va veure la necessitat de proporcionar una mica d'ajuda. En primer lloc, es va decidir afegir en la pantalla de joc una imatge reduïda del puzle resolt, perquè el jugador vegi on ha de col·locar cadascuna de les

¹Aspect Ratio. [en línia] [http://en.wikipedia.org/wiki/Aspect_ratio_\(image\)](http://en.wikipedia.org/wiki/Aspect_ratio_(image)) [data de consulta: 29/05/2014]

peces del puzle. Aquesta decisió, ha provocat l'eliminació del botó de barrejar peces (també s'elimina el cas d'ús CU03 associat). La funcionalitat d'aquest botó estava enfocada a que el jugador es prengués el seu temps per veure el puzle resolt abans de començar a jugar (ara ja no té sentit). En segon lloc, es va decidir oferir la possibilitat d'activar la numeració de les peces del puzle per identificar ràpidament el grau de desordre de cada puzle.

El comportament de l'activació de sons en l'aplicació depèn de la versió d'Android que hi ha instal·lada al dispositiu. Per exemple, en la versió 4.4.2 es permet augmentar o disminuir el volum de l'aplicació quan es fan servir els botons de volum del dispositiu. En canvi, en certes versions (com la 4.1.1) no es contempla aquest comportament. Per evitar que el volum de l'aplicació pugui ser una molèstia, des del menú de la pantalla de joc, s'ofereix la possibilitat d'activar o desactivar els sons que s'escolten quan s'està resolent un puzle.

En l'etapa de disseny, es va decidir que des de la pantalla de joc es podria finalitzar la partida i sortir del joc (tot alhora). Però durant l'etapa de realització de proves de l'aplicació, s'ha vist que és millor finalitzar la partida i anar a la pantalla del menú principal (en comptes de sortir de l'aplicació). D'aquesta manera, l'usuari té l'opció de començar una nova partida ràpidament. La sortida del joc només es pot realitzar des del menú principal de l'aplicació (Pantalla P2).

13.3 Valoració personal

La valoració personal en relació al projecte i a l'aplicació obtinguda és molt positiva, ja que s'han aconseguit tots els objectius que es van planificar. A més, la majoria de les variacions que s'han realitzat (respecte al disseny inicial) han sigut per afegir millores en el producte final.

La corba d'aprenentatge i les característiques pròpies d'Android, m'han fet avançar molt lentament al principi del desenvolupament, però a mida que es van assolint certs coneixements, l'aprenentatge és cada vegada més ràpid, permetent obtenir resultats eficients en molt poc temps. Gràcies a tota la informació disponible en la comunitat Android (documentació oficial, blogs personals, etc...) l'aprenentatge ha sigut un tasca entretinguda i motivant.

Deixant de banda la rigidesa del calendari del projecte (i tot el que això comporta), la realització del projecte ha sigut una experiència satisfactòria.

13.4 Millores

Encara que l'aplicació obtinguda ha cobert tots els requeriments que es van establir durant l'etapa de planificació, sempre hi ha punts a millorar. A continuació es descriuen algunes de les millores que es podrien afegir per completar l'aplicació:

- Rotació de la imatge en la pantalla d'acceptació: segons com s'ha realitzat una fotografia, pot aparèixer de cap per avall en el puzle. Així doncs, es podria afegir la possibilitat de girar la imatge des de la pantalla d'acceptació (just abans d'aplicar el retall amb el requadre blau).
- Afegir la possibilitat de configurar puzles de 5x5 i 6x6: Tenint en compte la mida de les pantalles de les tauletes i la mida d'alguns mòbils, no es desgavellat pensar en augmentar la dificultat dels puzles pels jugadors més agosarats. El disseny realitzat permet fer aquest canvi sense gaires modificacions.
- Afegir la possibilitat de configurar diferents idiomes: la implementació realitzada permet fer aquest canvi sense gaires modificacions (evidentment, sempre que disposem de les traduccions).
- Descarregar imatges d'Internet per fer-les servir com a imatges del puzle: la indicació de la URL de la imatge hauria de permetre descarregar-la des de la pròpia aplicació i utilitzar-la en el joc.
- Millorar la compatibilitat de l'aplicació amb diferents versions d'Android i diferents dispositius mòbils: la limitació temporal del projecte no ha permès provar millor l'aplicació en altres entorns.
- Afegir una pantalla (tipus tutorial) per ensenyar a resoldre puzles.
- Afegir més imatges de puzles educatius i classificar-les segons la temàtica a l'hora de seleccionar-les.
- Millorar el disseny gràfic de l'aplicació per que sigui més intuïtiva i més atractiva visualment.
- ...

Capítol 14

Fonts d'informació

La realització de l'aplicació no hauria sigut possible sense una formació (durant tot el projecte) relacionada amb la tecnologia Android. També ha sigut necessari aprendre algunes nocions relacionades amb els puzles lliscants que han ajudat a desenvolupar certs aspectes de l'aplicació. A més a més, ha calgut cercar informació relacionada amb totes les eines utilitzades durant el projecte.

A continuació es mostren les fonts d'informació utilitzades. Encara que en la llista següent no s'hagi anotat alguna de les fonts consultades, es mostren les que més repercussió han tingut sobre el projecte.

14.1 Bibliografia

- Lee, Wei-Meng. (2012). Android 4. Desarrollo de aplicaciones. Espanya: Anaya Multimedia/Wrox

14.2 Recursos web Android

android | Android developers [en línia]. <http://developer.android.com> . El principal recurs web per aprendre a desenvolupar aplicacions per Android.

stackoverflow [en línia]. <http://stackoverflow.com> . Stack Overflow es una web de preguntes i respostes per a programadors.

CoderzHeaven [en línia]. <http://www.coderzheaven.com> . Recull d'exemples de programació en diversos llenguatges.

Linux.com [en línia]. <http://www.linux.com> . Web relacionada amb tot el que envolta el sistema operatiu Linux.

AndroidHive [en línia]. <http://www.androidhive.info/> Web que conté diversos tutorials de programació en Android

Android hub 4 you [en línia]. <http://www.androidhub4you.com> . Recull d'exemples d'Android.

danielme [en línia] <http://danielme.com/tips-android/> Consells i articles relacionats amb el desenvolupament d'aplicacions Android.

sgoliver.net blog [en línia] <http://www.sgoliver.net> . Curs on-line de programació en Android

El baúl del programador [en línia] <http://elbauldelprogramador.com/> Blog sobre Android que recull articles propis i articles d'altres blogs relacionats amb Android.

Mundo geek [en línia] <http://mundogeek.net/android/> Traducció al castellà d'alguns articles de la web oficial developer.android.com

My experiments with technology [en línia] <http://technicalmumbojumbo.wordpress.com/category/android/> Blog amb tutorials sobre Android.

Android Point.. [en línia] <http://sunil-android.blogspot.com.es/2013/08/out-of-memory-exception-during-decode.html> . Article que parla sobre l'excepció de 'OutOfMemory' al carregar imatges.

SekthDroid [en línia] <http://sekthdroid.wordpress.com/> Blog en el que es tracten diverses qüestions relacionades amb el desenvolupament d'aplicacions Android

Curso Android [en línia] <http://www.androidcurso.com/> Curs on-line d'Android de la Universitat Politècnica de València.

La columna 80 [en línia] <http://columna80.wordpress.com/2011/02/17/arquitectura-de-android/> Article que conté informació relacionada amb l'arquitectura d'Android.

TechWorkSlab [en línia] <http://techworkslab.blogspot.com.es> . Blog d'Android que conté un article relacionat amb el patró de disseny Model-Vista-Controlador.

14.3 Recursos web sobre puzles

LCBSoft [en línia] http://www.lcbsoft.com/infesp/puzz_i02.htm Informació general sobre el joc del puzle lliscant

Blog de Jsbsan [en línia] <http://jsbsan.blogspot.com.es/2012/09/resolver-puzzles-deslizantes-de-8-y-15.html> . Blog on s'explica com resoldre puzles de 3x3 i 4x4 peces.

14.4 Altres recursos

Wikipedia [en línia] <http://es.wikipedia.org> . Enciclopèdia web de contingut obert basat en un model d'edició obert. (Castellà)

Wikipedia [en línia] <http://en.wikipedia.org> . Enciclopèdia web de contingut obert basat en un model d'edició obert. (Anglès)

Mathematics Stack Exchange [en línia] <http://math.stackexchange.com/> Web de preguntes i respostes per a professionals i aficionats a les matemàtiques.

Banco de imágenes y sonidos [en línia] <http://recursostic.educacion.es/bancoimagenes/web/> . Web del Ministeri d'Educació que ofereix recursos audiovisuals per estimular el desenvolupament de continguts educatius

Google [en línia] <http://www.google.es> . Buscador d'informació en internet.

Evolus Pencil [en línia] <http://pencil.evolus.vn>. Web del programari utilitzat per realitzar el prototip de l'aplicació del projecte.

WireFrameSketcher [en línia] <http://wireframesketcher>. Web que conté programari per realitzar prototips d'aplicacions Android.

GraphViz [en línia] <http://www.graphviz.org/> . Web que conté el programari utilitzat en la realització d'alguns diagrames del projecte.

Dia [en línia] <http://live.gnome.org/Dia> . Web que conté el programari utilitzat en la realització d'alguns diagrames del projecte.

SQLite [en línia] <http://www.sqlite.org/> . Documentació sobre les bases de dades SQLite.

L^AT_EX [en línia] <http://latex-project.org/> Web del llenguatge L^AT_EX.

TexMaker [en línia] <http://www.xmlmath.net/texmaker/> . Web de l'editor de L^AT_EX utilitzat per realitzar la memòria del projecte.

MikTeX [en línia] <http://miktex.org/> . Programari per la gestió de paquets utilitzats amb el TexMaker.

TeX Stack Exchange [en línia] <http://tex.stackexchange.com/> . Web de preguntes i respostes sobre el llenguatge L^AT_EX

Miniejercicios con L^AT_EX [en línia] <http://minisconlatex.blogspot.com.es/> . Blog que conté exemples d'utilització del llenguatge L^AT_EX

FastStone [en línia] <http://www.faststone.org/> Programari per visualitzar i editar imatges.

Camtasia Studio [en línia] <http://www.techsmith.com/> Programari de captura i edició de vídeo.

WinFF [en línia] <http://winff.org> . Programari per convertir arxius de vídeo a diferents formats.

UML Diagrams [en línia] <http://www.uml-diagrams.org> . Web amb informació sobre la creació de diagrames UML.