



Universitat de Girona
Escola Politècnica Superior

Projecte/Treball Final de Carrera

Estudi: Eng. Tècn. Informàtica de Gestió. Pla 1993

Títol: Portabilitat d'aplicacions de PC a PDA mitjançant un cas d'estudi: ScummVM

Document: Resum

Alumne: Marc Coll Passolas

Director/Tutor: Gustavo Patow
Departament: Informàtica i Matemàtica Aplicada
Àrea: LSI

Convocatòria (mes/any): 09/2006

Introducció

Els assistents personals digitals, PDA i els telèfons mòbils, són cada dia més freqüents, arribant a igualar a ordinadors molt més potents en quant a velocitat i versatilitat, dissenyats originalment amb una finalitat completament diferent. Però tot i així, encara existeix una gran quantitat de *software* que no ha estat desenvolupat per aquests dispositius, però que són molt reconeguts en l'àmbit dels ordinadors personals o PC.

Aquests dispositius tenen una sèrie de limitacions en comparació als ordinadors personals, com per exemple:

- Pantalla petita i de baixa resolució.
- Absència de dispositius d'entrada i sortida habituals com són el teclat, el ratolí, disquetes i altres dispositius d'entrada i sortida. En la majoria dels casos només trobem uns botons i una pantalla tàctil, sobre la qual l'usuari pot escriure mitjançant un llapis, propi de cada dispositiu.
- etc.

Aquestes diferències amb els PC fan que el fet de portar una aplicació, que originalment ha estat dissenyada per un PC, sigui una feina molt pesada i que en la majoria dels casos s'opti per refer l'aplicació començant des de zero.

Dintre del món dels dispositius mòbils, existeix una gran quantitat de dispositius diferents (telèfons mòbils, PDA, etc...). Actualment la majoria funcionen amb *Win CE*, una versió específica per dispositius portàtils del conegut sistema operatiu de *Microsoft* o bé amb *Symbian*, molt utilitzat en els telèfons mòbils. La marca pionera en els dispositius mòbils anomenats PDA ha estat Palm, i és per això que nosaltres hem escollit un dispositiu amb el sistema operatiu *Palm OS* per a la realització de l'estudi. El nostre dispositiu presenta la majoria de les limitacions abans esmentades i algunes més, però per la realització del nostre estudi ens centrarem en les següents:

- Pantalla petita, de baixa resolució (160x160) i amb poca profunditat de color (escala de grisos, 16 colors, 256 colors).
- Absència de dispositius d'entrada i sortida com el teclat, el ratolí, disquetes, en comptes d'aquests, disposa de 6 botons i una pantalla tàctil, sobre la qual l'usuari pot escriure mitjançant un llapis.

Objectius

L'objectiu del treball és estudiar la portabilitat dels programes que han estat originalment dissenyats per un PC a un dispositiu mòbil, en concret a un PDA amb el sistema operatiu *Palm OS*. Per al nostre treball, hem escollit una aplicació com a cas d'estudi que ens permet donar solucions a les limitacions que presenta el dispositiu amb *Palm OS*. Aquesta aplicació s'anomena *ScummVM*, i es tracta d'una implementació de codi lliure del conegut motor de videojocs *SCUMM*, creat i utilitzat per *LucasArts* en les seves aventures gràfiques.

Donarem possibles solucions a la problemàtica que genera el fet que la pantalla del dispositiu mòbil utilitzat tingui una resolució de 160x160 a l'hora de mostrar imatges i textos *renderitzats* per una resolució superior. També solucionarem el problema que representa el fet de no tenir ratolí, ni altres dispositius d'entrada i sortida tradicionals. La intenció és explicar també el procés que s'hauria de seguir per poder portar aplicacions de PC a *Palm OS* i les eines que s'utilitzarien. Tot això es farà sempre de la manera menys invasiva possible, és a dir, els canvis al codi font de l'aplicació original seran els mínims per garantir el correcte funcionament del programa en la nova plataforma.

Treball realitzat

Per aconseguir els nostres objectius, el primer que s'ha fet ha estat estudiar en detall el codi de la versió per PC de l'*ScummVM*, per poder localitzar les parts del codi que tracten els elements següents:

- les imatges dels jocs.
- el text dels jocs.
- els events dels dispositius d'entrada i sortida.

Un cop localitzats els mètodes i que tracten aquests tres elements, hem escollit quins seran els que utilitzarem per introduir els punts d'entrada de la nostra aplicació. D'aquesta manera, podrem accedir a les imatges, al text i als events, i modificar-los com ens calgui.

Control de les imatges:

Les imatges originals dels jocs que l'*ScummVM* interpreta, estan a una resolució de 320x200 i el PDA disposa d'una resolució de 160x160, és per això, que hem implementat la reducció a la meitat de les imatges originals. Per fer-ho hem utilitzat varis mètodes de reducció:

- **Reducció 2:1 sense filtres:** La reducció sense filtres, es basa en descartar un de cada dos píxels de la imatge original. Nosaltres hem decidit eliminar tots els píxels que ocupen posicions parells a dins la imatge. Aquest tipus de reducció ens presenta un grau molt elevat l'*aliasing*.
- **Reducció 2:1 amb filtres:** Per la reducció amb filtres hem escollit el mètode de filtrat mitjançant una matriu amb pesos. Aquesta reducció consisteix a aplicar una matriu de pesos de $N \times M$, que farà de màscara, on a cada casella hi trobarem un valor numèric (pes). Aquest pes indica la importància relativa del píxel que ocupa la mateixa posició que l'element de la matriu. Es tracta d'aplicar la matriu sobre la imatge original per calcular els píxels de la imatge reduïda a partir de la mitjana ponderada obtinguda d'aplicar la màscara a la imatge original. Els mètodes utilitzats són els següents:
 - Filtrat utilitzant una matriu de pesos de 2×2 on tots els elements de la matriu tenen el mateix pes.
 - Filtrat utilitzant una matriu de pesos de 3×3 on els elements centrals de la matriu tenen més pes (Aquest mètode es descarta en la versió final perquè la quantitat de càlculs a realitzar, fa que els jocs vagin molt lents i sigui impossible jugar-hi).

Per poder realitzar les operacions aritmètiques de la reducció amb pesos, necessitem saber el format en el que l'*ScummVM* emmagatzema el color d'un píxel. Ens trobem que utilitza els formats següents:

- En la versió per PC el color de cada píxel de la imatge està emmagatzemat en format RGB dins d'un enter de 16 bits. Per tant, si realitzem operacions aritmètiques directament amb el color en aquest format ens trobem que se'ns poden barrejar les components i corrompre el color original. A causa d'això, hem hagut d'implementar dos mètodes. Un mètode que a partir d'un color en aquest format, ens aïlli les seves components RGB per poder operar amb cada una d'elles per separat. I un altre mètode que a partir de tres components RGB ens generi un color amb el format de l'*ScummVM*.
- En la versió per *PalmOS* el color de cada píxel de la imatge és un índex a una paleta de colors on cada color està representat utilitzant el mateix format que a la versió de PC. A causa d'això, hem implementat dos

mètodes més que en la versió per PC. Un mètode que donat un índex de la paleta ens retorni el color que el representa i un altre que donat un color ens retorni l'índex de la paleta. Com que el color de la paleta està en el mateix format que en la versió per PC hem pogut utilitzar els mètodes implementats per realitzar els càlculs aritmètics necessaris per la reducció.

Control del text:

Analitzant el codi de l'*ScummVM* descobrim que tracta tota l'estona el text com a tal (text pur) fins al moment de mostrar-lo per pantalla que el *renderitza* convertint-lo en imatge.

A causa de la reducció que hem hagut d'aplicar, tot el text es torna il·legible, és per això que la nostra tasca ha estat recollir-lo just abans que l'*ScummVM* el converteixi a imatge. Un cop recollit, el podem tractar i mostrar per pantalla amb la tipografia de *PalmOS* utilitzant les llibreries de l'*SDK (Standard Development Kit)*. Abans d'emmagatzemar i tractar el text necessitem saber quins tipus de text tenim. Ens trobem que hi ha els tipus de text següents:

- **Text interactiu:** és el text sensible als clics de ratolí per part de l'usuari i que serveix per controlar els jocs. Per controlar aquest tipus de text, la nostra aplicació el guarda juntament amb una sèrie d'atributs. Com que és un text sensible als clics necessitem guardar les coordenades originals d'on l'*ScummVM* el mostra per pantalla, i les coordenades d'on el mostrem nosaltres per pantalla. Hem d'emmagatzemar les coordenades per dos motius: la reducció d'imatge que farà que el text no estigui a les mateixes coordenades (si no que estarà a les mateixes dividides per dos) i les diferències entre la tipografia de l'*ScummVM* i la de *PalmOS*. D'aquesta manera quan l'usuari faci clic al damunt del text que mostrem nosaltres, podem identificar quin text ha clicat i enviar un event de clic al sistema amb les coordenades reals del text. Hi ha dos tipus de text interactiu:
 - Text dels menús d'acció que hi ha per cada joc.
 - Text interactiu de les converses.
- **Text no interactiu:** és tot aquell text que es mostra per pantalla i amb el que l'usuari no hi pot interactuar, només llegir. La nostra aplicació no guarda aquest text, només el recull i el mostra per pantalla. Els tipus són:
 - Text de les converses entre personatges, o del narrador.
 - Text escollit del menú d'acció o de les converses interactives.

Control dels events d'entrada/sortida:

Com que els PDA no disposen de ratolí ni de teclat, hem hagut de definir dos dispositius d'entrada i sortida del PDA que ens puguin servir per substituir-los. Els dispositius escollits han estat els del llapis òptic (ratolí) i els dels botons del PDA (teclat). Com que l'*ScummVM* no reconeix els events generats per aquests dos dispositius, hem hagut de buscar el mètode que tracta els events a dins del codi de l'*ScummVM*. Un cop trobat el mètode podem capturar els events generats pel llapis òptic i pels botons del teclat per tractar-los i convertir-los en events de ratolí o de teclat perquè l'*ScummVM* els pugui interpretar. A causa de la reducció d'imatge hem tractat els events de la manera següent:

- Events de llapis òptic: identifiquem la zona de la pantalla on s'ha clicat. Si es tracta de la zona del text interactiu, la nostra aplicació busca el text on s'ha clicat i genera el clic de ratolí amb les coordenades reals d'aquest text. I si es tracta de la zona on no hi ha text interactiu recull les coordenades del clic, les multiplica per dos i genera un event de clic amb les noves coordenades.

- Events dels botons del PDA: només utilitzem els quatre botons que hi ha a la part central del PDA. Aquests botons ens serviran per fer la mateixa funció que les fletxes de direcció d'un teclat. En funció del botó que l'usuari hagi premut desplaçarem el cursor del ratolí, una posició en la direcció de la fletxa representada pel botó.

Resultats

Partint doncs, d'un PDA amb sistema operatiu *PalmOS* i de l'aplicació *ScummVM* com a cas d'estudi, hem obtingut una capa de portabilitat sobre l'*ScummVM* que resol els problemes següents:

- **Resolució de les imatges:** Imatges originals dels jocs al doble de resolució (320x200 píxels) de la que disposem al PDA (160x100 píxels), per resoldre-ho hem implementat 2 mètodes de reducció d'imatge.
 - Sense filtrat.
 - Amb filtrat utilitzant una matriu de pesos 2x2, on tots els elements de la matriu tenen el mateix pes.
- **Text dels jocs:** al reduir les imatges el text s'ha tornat il·legible, ja que aquest formava part de la imatge. Hem capturat tot el text abans de què es convertís a imatge i el mostrem com a text pur, utilitzant les llibreries de *PalmOS*. Els tipus de text són els següents:
 - Text interactiu.
 - Text del menú d'accions del joc.
 - Text de les converses interactives entre els personatges dels jocs.
 - Text no interactiu.
 - Text de les converses dels personatges o del narrador del joc.
 - Text de l'opció escollida del menú d'acció.

A les captures de la figura 1.0, mostrem un exemple del funcionament de l'*ScummVM* utilitzant la nostra aplicació que s'executa sobre *PalmOS*, comparada amb el joc original executant-se en un PC.

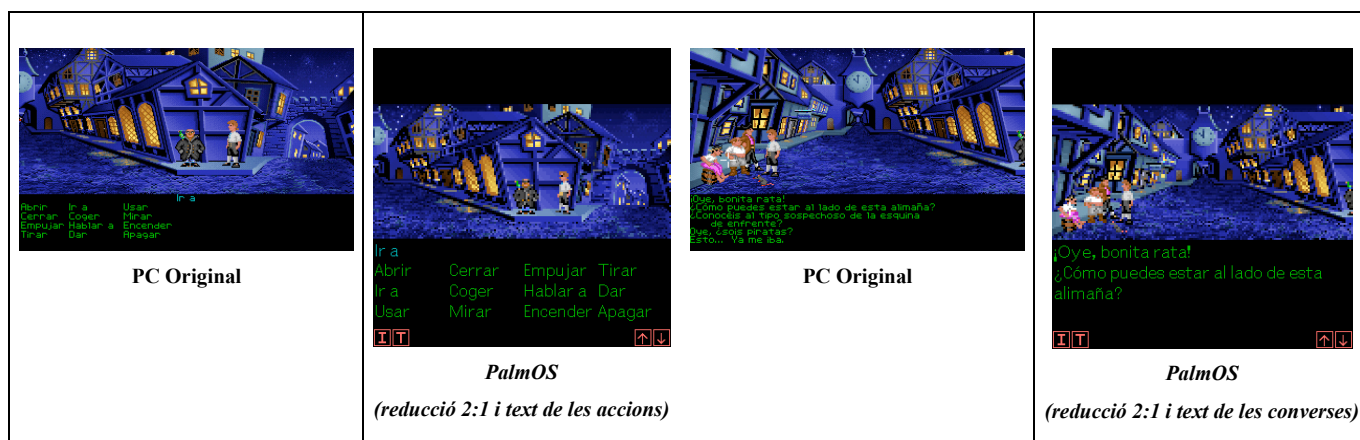


Fig 1.0 Captures dels resultats obtinguts

Un altre resultat ha estat l'obtenció d'una guia a seguir, en el cas de voler portar una aplicació de PC a un dispositiu mòbil. Aquesta guia s'ha extret a partir del procés seguit durant el desenvolupament de la nostra capa de portabilitat. Els passos generals de la guia serien els següents:

1. Documentar-se sobre la plataforma i el sistema operatiu al qual es vol portar l'aplicació.
2. Analitzar el codi de l'aplicació que es vol portar a la nova plataforma i assegurar-nos que no presenta limitacions de disseny o arquitectura que no permeti portar l'aplicació.
3. Identificar quins seran els problemes que haurem de resoldre per culpa de les limitacions de la nova plataforma respecte l'aplicació per PC (resolució i mida de la pantalla, velocitat del processador, mida de memòria, etc.).
4. Mirar d'entendre a grans trets el funcionament d'aquesta aplicació.
5. Localitzar i analitzar el funcionament dels mètodes i/o funcions encarregats de l'entrada/sortida de:
 - Imatges
 - Text
 - Events de teclat, ratolí i altres dispositius que interactuïn amb l'aplicació.
6. Localitzar les crides a aquests mètodes i/o funcions per poder establir els punts d'entrada al codi de l'aplicació per part de la capa de portabilitat.
7. Treballar sobre PC tant temps com sigui possible per:
 - Establir aquests punts d'entrada i recollir la informació que necessitem per la nostra capa de portabilitat.
 - Resoldre els problemes que afecten a les imatges, el text i control d'events de l'aplicació que volem portar respecte a la nova plataforma.
8. Definir la capa de portabilitat, el seu funcionament, les classes, mètodes i funcions que s'implementaran.
9. Finalment portar l'aplicació a la nova plataforma, canviant totes les crides al sistema per crides al sistema de la nova plataforma i adaptant el codi de la capa de portabilitat al nou sistema.