

Documentacion del Proyecto Buscaminas en Android

Oswaldo Bayona
Rodrigo Castro
Jorge Vergara

11 de diciembre de 2013

1. Introducción

Nuestro proyecto de Lenguajes de Programacion del Primer Parcial,consiste en la implementacion del juego Buscaminas en Android. Para desarrollar esta aplicacion hemos usado el entorno de trabajo que nos provee Eclipse y hemos hecho las respectivas pruebas en los dispositivos que hemos tenido al alcance. La logica del juego es exactamente la misma que el juego de Buscaminas que todos conocemos pero nosotros hemos personalizado la interfaz grafica clasica con el objetivo de darle originalidad. Hemos hecho uso de muchos conceptos aprendidos sobre Programacion Orientada Objetos, tales como herencia, polimorfismo,interfaces,manejadores de eventos,etc algunos de estos conceptos seran tratados con mayor detalle mas adelante. En este documento trataremos de explicar brevemente la definicion de las clases mas relevantes del programa, plantearemos las razones por las cuales utilizamos lo que utilizamos. Ademas hemos incluido una seccion con el avance progresivo de la interfaz del proyecto en donde se muestra imagenes del trabajo que hemos venido realizando en estas semanas. Tambien definiremos el alcance de nuestra aplicacion en donde aclararemos los comportamientos que hemos implementado y las cosas que hemos considerado no incluir.

2. Justificación

En el desarrollo del proyecto hemos aplicado el paradigma de programación orientado a objetos. Hemos creado varias clases para dividir el problema en pequeñas partes, entre estas clases tenemos.

Interface Observer Esta interface la hemos definido para poder implementar el patrón de diseño Observer, consta de dos eventos Update que sirven para notificar al Observador de los cambios que ocurren en el Observado.

2.1. Clase Celda

Esta es una de las clase principales del proyecto simula cada una de las casillas del juego, esta clase extiende de Button e implementa Observer.

¿Por qué extiende de Button?

Esto se lo hizo principalmente para facilitar el manejo de eventos como click, Drag and Drop utilizando los listeners que están definidos para este componente.

¿Por qué implementa Observer?

Lo hemos implementado así para lograr en forma más eficiente el algoritmo para descubrir las celdas, cada celda tiene un ArrayList con sus celdas adyacentes las cuales observan un cambio de esta celda para descubrirse o no dependiendo de la situación, esto involucra que todas las celdas son Observers y a la vez son Observadas por sus adyacentes. Aparte de estas dos características todas las celdas poseen un estado que puede ser CUBIERTA, DESCUBIERTA, BANDERA que sirve para que la celda sepa cómo comportarse ante algún evento.

2.2. Clase Tablero

Esta clase al igual que celda es una de las más importantes, es básicamente un conjunto de celdas que se organizan en forma de matriz, pero hemos decidido utilizar un HashMap para contener las celdas, porque al no ser muchas celdas se pueden generar muy fáciles las claves y el acceso a cada elemento del Mapa es muy rápido, esta clase extiende de View e implementa Observer.

¿Por qué se extiende de View?

La razón es muy simple el tablero se lo tiene que instanciar dentro de una actividad por lo cual la mejor opción era que extendiera de View

¿Por qué se implementa Observer?

Básicamente esto es para poder determinar si el juego continúa o finaliza, para esto el Tablero debe observar cuando alguna de las celdas cambie y realizar las validaciones correspondientes.

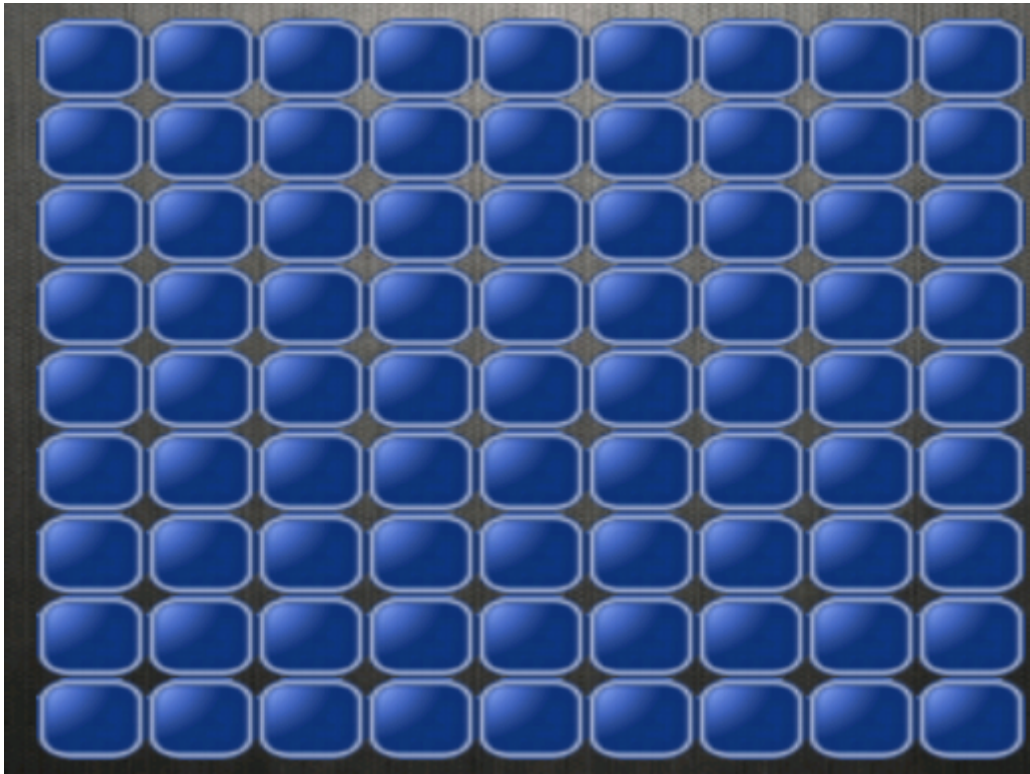


Figura 1

2.3. Clase Barra de Menu

Esta clase es muy sencilla consiste en un cronometro, un botón que sirve para reiniciar el juego, una bandera y un contador de banderas, y extiende de TableLayout.

¿Por qué extiende de tableLayout?

Porque la barra tiene varias componentes y una forma muy fácil de organizarlos es con el TableLayout.



Figura 2

2.4. Comportamiento Drap and Drop

Se decidió utilizar Drap and Drop porque en Android no existe el Click derecho para poder colocar banderas en una celda, básicamente consiste en poder arrastrar una bandera encima de una celda y luego poder sacar la bandera de la celda arrastrándola.

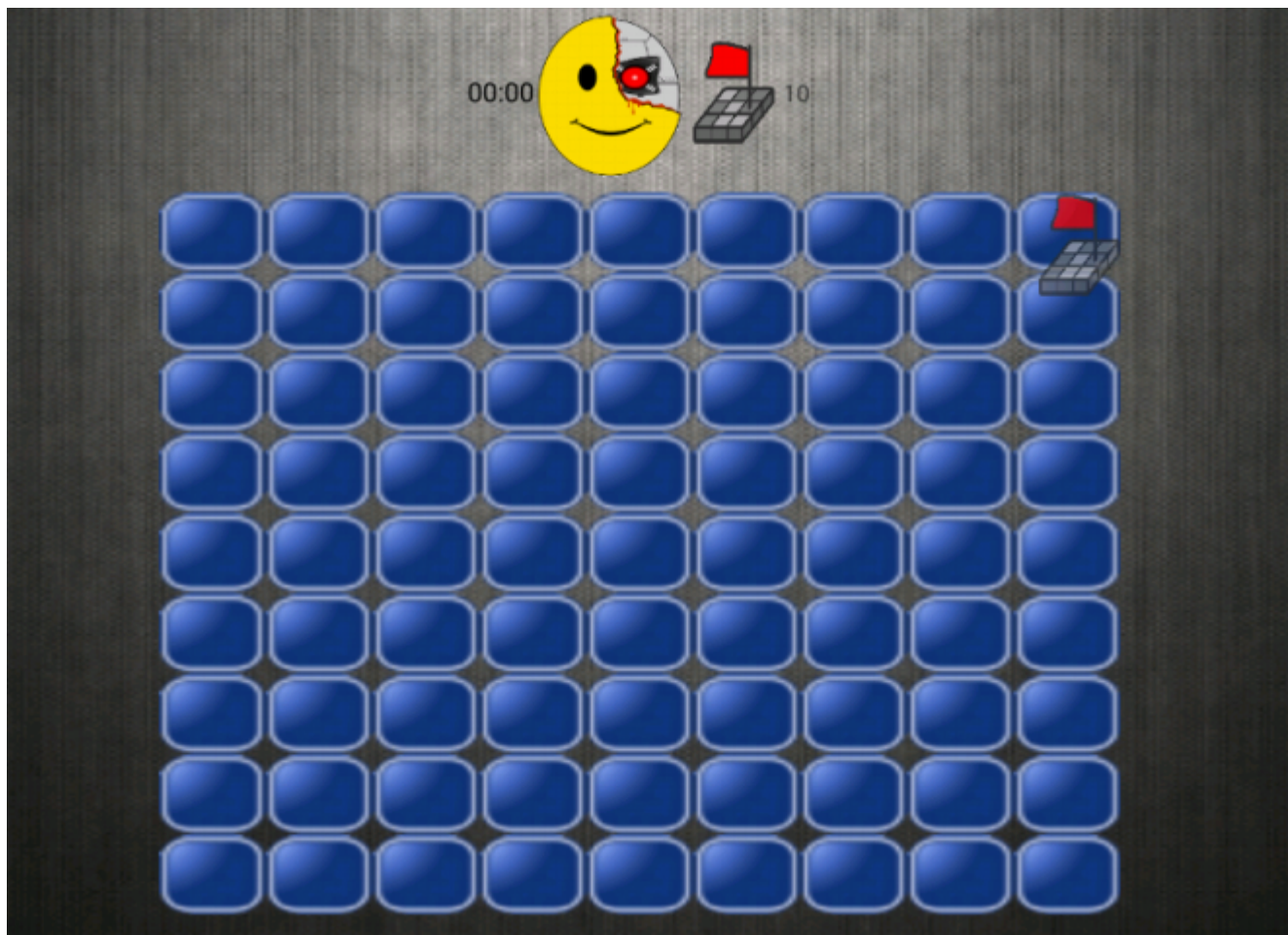


Figura 3

2.5. Scroll

Decidimos utilizar un ScrollView para colocar el tablero porque mientras mayor es el numero de celdas del tablero para que se vean todas las celdas en la pantalla se tendrían que colocar en un tamaño muy pequeño el cual no permitiría jugar al usuario como es debido, esto se soluciono con el scrollView.

3. Avance Interfaz Grafica

Esta seccion tiene por objetivo mostrar imagenes de los diferentes estados de la interfaz grafica del proyecto.

La figura 4 muestra el Tablero de nuestra aplicacion en los primeros dias de implementacion, como podemos ver los botones no poseen ningun caracteristica visual en particular e incluso el Tablero no muestra todas las celdas del juego como deberia

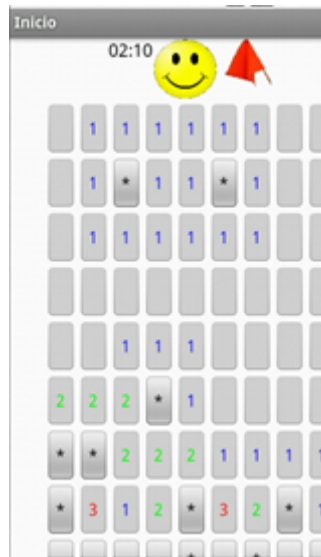


Figura 4

En la figura cinco podemos observar que la interfaz grafica se ha mejorado un poco, se integraron recursos graficos con mejor aspecto visual, podemos ver que el Tablero en modo facil se muestra sin ningun problema

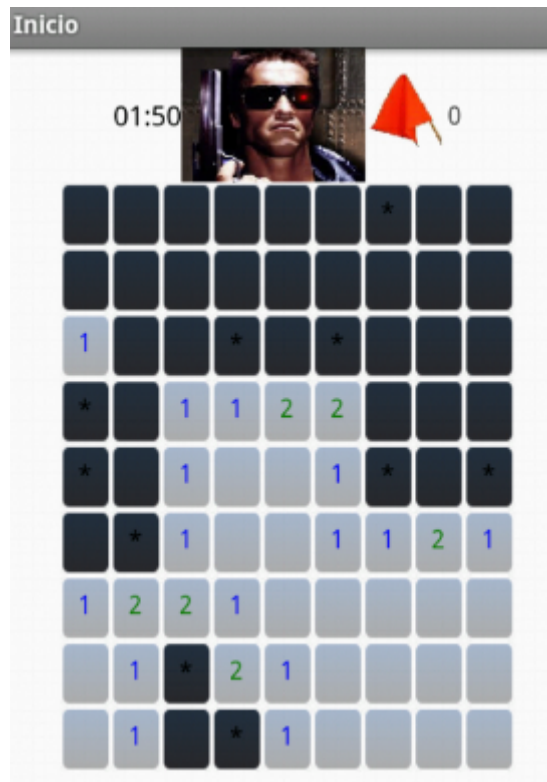


Figura 5

En la figura seis vemos el estado del menu principal del juego, en este punto del desarrollo de nuestra aplicacion pudimos darnos cuenta de lo importante que era las distintas densidades de los dispositivos en android, la figura 6 corresponde a la de un emulador de 800x600 con densidad xxhdpi (este tipo de caracteristicas son de telefonos celulares) en donde los recursos graficos que habiamos generado se veian exageradamente grandes, incluso el tablero de modo intermedio y experto no se mostraban por completo.



Figura 6

En base a lo planteado anteriormente decidimos hacer pruebas en distintas resoluciones y densidades para verificar en cual podiamos trabajar mas comodamente. El resultado fue que para densidades mdpi la aplicacion mostraba los componentes de forma mas eficiente , estas densidades se encuentran en distintas versiones de tablets y es por esta razon que decidimos pensar nuestro juego para estos dispositivos. La figura 7 muestra el estado del proyecto corriendo sobre un emulador de 1280x800 mdpi.



Figura 7

La figura 8 muestra el Tablero en Modo Experto en donde podemos ver que los botones se han ajustado de forma correcta y los recursos graficos se muestran con la resolucion apropiada. Cabe recalcar que hicimos uso de ScrollView para poder hacer mas facil la jugabilidad ya que si definiamos botones mas pequeños iba a ser imposible jugar sin equivocarse.

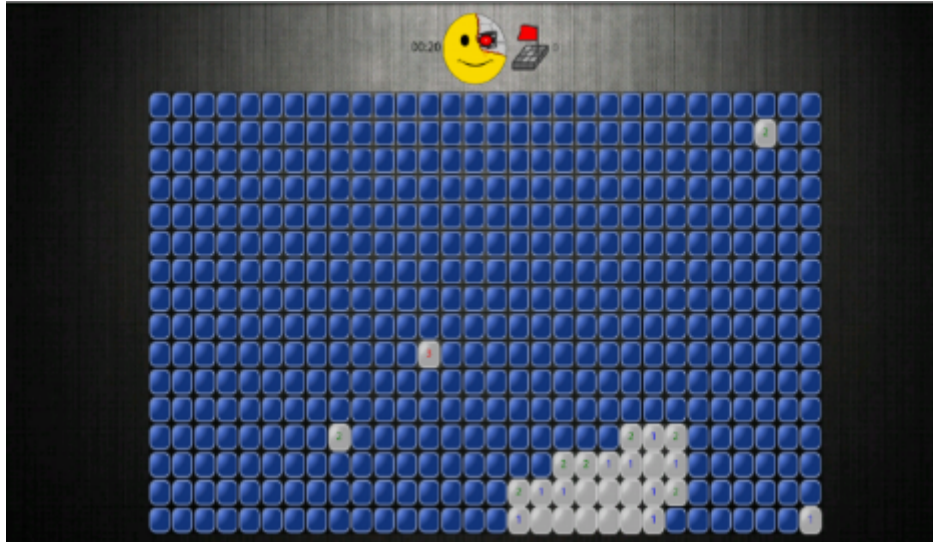


Figura 8

4. Alcance

El proyecto fue escrito en la versión de Android 4.3, con lo cual sólo funcionaría correctamente en dispositivos avanzados. Nuestro proyecto al inicio muestra un menú de opciones para elegir entre: Partida Nueva, Acerca De y Top Jugadores, como se muestra en la fig1. La opción Partida Nueva muestra otra actividad para elegir entre tres niveles, Fácil, Intermedio y Experto, la opción de personalizar el número de filas, columnas y bombas del tablero no ha sido implementada.



Figura 9

El tablero del juego muestra básicamente dos Views uno en la parte superior con opciones donde se muestran un cronómetro, un botón para reiniciar el juego, una bandera y un contador; el otro View muestra las celdas. Los algoritmos recursivos de la expansión de las celdas funcionan muy bien. Todas las opciones funcionan correctamente, cabe destacar que en el proyecto si fue posible implementar un drag and drop para arrastrar la bandera y ubicarla en la celda, además se puede desde cualquier celda arrastrar la bandera para ser colocada en otra celda o ser eliminada soltándose en un sitio libre de celdas.

Tenemos implementado varias formas de ganar el juego: una de ellas consiste en descubrir todas las celdas sin minas, pudiendo usar o no usar las banderas; otra manera es que en el momento de haber colocado banderas en todas las celdas con bombas, automáticamente se descubren las celdas que no han sido descubiertas y se gana el juego. Esta última forma se implementó para que el acto de ganar sea más rápido porque el juego también consiste en ganar en el menor tiempo posible y romper records. En el juego existe una cantidad de banderas infinitas, es posible llenar todo el tablero con ellas y seguir arrastrando aunque no haya espacio para ponerlas, en esta situación el contador no es afectado.



Figura 10: //Tablero del juego: arrastrando una bandera hasta una celda

Al momento de ganar el juego, la aplicación obtiene el tiempo del cronómetro, busca en su base de datos si el tiempo o las condiciones son apropiados para entrar al top. Las condiciones del top son las siguientes, si existen diez o menos jugadores en la lista, se permite el ingreso; si existen más de diez la aplicación revisa si el tiempo es menor a alguno de los existentes y permite el ingreso. Así como lo implementamos se permite tener infinitos jugadores en la lista. La base de datos consiste en tres ficheros .txt (uno para cada nivel) que son guardados en la memoria interna del dispositivo, los datos son presentados al usuario, ordenados de forma descendiente y usando un TableHost para visualizar mejor los resultados. Entre las cosas que no se pudieron implementar está conseguir mayor portabilidad de la aplicación, no se visualiza bien la resolución en teléfonos o Tablet muy pequeñas, o en dispositivos con densidades de xxhdpi en donde los componentes de la aplicación se ven exageradamente grandes, para dispositivos grandes con una resolución de 1280 X 800 con densidad de mdpi no se tiene ningún problema, tampoco se consiguió que al girar la pantalla del dispositivo la aplicación gire y se adapte a la nueva forma vertical, la aplicación bloquea el acelerómetro y mantiene la orientación en horizontal.