

Trabajo Final de Carrera

Memoria

ALUMNA: M^a Eugenia Ruberte Papaseit

TFC ETIG

20-05-2013

Consultor: Ignasi Lorente Puchades

A Domènec, mi comprensivo marido, que me ha dejado mi espacio para desarrollarme libremente.

A Rebeca, mi luz de 8 meses, que me enamora y me vuelve loca.

A mi madre por cuidar de la niña en esos pequeños espacios de tiempo de los fines de semana.

A los compañeros del trabajo, sobre todo a Bet, mi jefa, mi compañera y mi amiga, que me ha ayudado durante todos estos largos años de estudios.

Índice

Índice	3
PRESENTACIÓN.....	5
ESTUDIO VIABILIDAD	6
<i>Planteamiento</i>	6
<i>Marco</i>	6
<i>Tecnología a utilizar</i>	6
<i>Riesgos</i>	7
<i>Listado de funcionalidades inicial</i>	7
<i>Desarrollo</i>	8
<i>Fases</i>	8
<i>Requisitos iniciales</i>	8
<i>Análisis del sistema</i>	8
<i>Diseño</i>	8
<i>Implementación</i>	9
<i>Planificación</i>	9
REQUISITOS INICIALES (DCU).....	10
<i>Selección de los métodos de indagación</i>	10
<i>Indagación</i>	10
<i>Diseño conceptual (Usuarios y contexto de uso)</i>	12
<i>Perfiles de usuario identificados</i>	12
<i>Escenarios de uso</i>	15
<i>Requisitos iniciales funcionales y no funcionales</i>	16
DISEÑO FUNCIONAL.....	19
<i>Arquitectura de la aplicación</i>	19
<i>Definición de los casos de uso</i>	20
<i>Flujos de interacción</i>	22
<i>Prototipaje</i>	23
<i>Sketches</i>	23
<i>Prototipo horizontal de alta fidelidad (realizado con Pencil)</i>	26
<i>Evaluación</i>	30
<i>Preguntas test</i>	30

<i>Tareas a realizar</i>	31
<i>Preguntas sobre las tareas</i>	31
IMPLEMENTACIÓN	32
<i>Instrucciones de instalación del binario</i>	32
<i>Batería de pruebas</i>	33
<i>Dispositivos probados</i>	33
<i>Resultados</i>	34
<i>Pantalla principal del juego</i>	43
<i>Pantalla menú principal</i>	47
<i>Pantalla de ranking</i>	47
<i>Pantalla de ayuda</i>	48
<i>Global aplicación</i>	48
CONCLUSIONES	52
BIBLIOGRAFIA	62

PRESENTACIÓN

Al plantearnos la temática del Trabajo de Final de Carrera, optamos por el área de Desarrollo de aplicaciones para dispositivos móviles. Creemos que el desarrollo de aplicaciones para dispositivos móviles es un área en auge y con futuro.

También tiene una fuerte presencia en el presente inmediato ya que millones de personas usan en la actualidad dispositivos móviles de última generación.

Nosotros queríamos formar parte de esta revolución tecnológica aportando nuestro pequeño granito de arena. Por ello nos hemos planteado realizar una aplicación para dispositivos móviles que usen Android como sistema operativo.

Otra meta que teníamos en mente desde hacia tiempo, era desarrollar un juego por nuestros propios medios. Llenar parte de tu tiempo de ocio (el poco que tengas) usando una aplicación que has desarrollado tu mismo, da una sensación muy especial.

Con esta idea en la mente hemos llevado a cabo nuestro proyecto de desarrollar el juego MASTERMIND, un clásico juego de concentración e inteligencia.

ESTUDIO VIABILIDAD

Planteamiento

El objetivo de este trabajo es poner en valor todos los conocimientos adquiridos en la carrera. Para ello hemos escogido la opción que plantea realizar una aplicación en Android que sea totalmente funcional. En este caso nos hemos propuesto desarrollar el juego MasterMind, un clásico de los juegos de mesa, pero adaptándolo a las nuevas tecnologías que nos permitirán darle una orientación social al estilo de Apalabrados o Mezcladitos.

Marco

El sistema operativo Android tiene muy pocos años de edad pero ha entrado pisando fuerte. Estas son algunas de las razones para elegir este sistema operativo:

- cualquier persona puede desarrollar aplicaciones en Android,
- que dichas aplicaciones se pueden integrar con los servicios de Google,
- que es un sistema de código abierto y
- que actualmente está instalado en la mayoría de terminales de móviles y tablets.

Tecnología a utilizar

Para realizar nuestro proyecto hemos utilizado la última versión del *SDK de Android* disponible en la página web oficial de Android.

También se ha utilizado la librería *AndEngine (Free Android 2D OpenGL Game Engine)* que está basada en Android y *OPENGL2D* y resulta muy útil para desarrollar juegos.

En un principio estaba previsto usar el *SDK Skiller* o el *SDK ScoreLoop* para desarrollar la parte de sociabilización pero al final hemos optado por el *SDK Swarm*.

Para realizar el diseño de los Sprites y TiledSprites hemos utilizado *GIMP 2*.

Riesgos

Se debía tener en cuenta que existía la posibilidad de que las metas planteadas al inicio no pudieran alcanzarse en el tiempo establecido, debido a una serie de riesgos tales como:

- no haber realizado con anterioridad ningún proyecto en Android ni en ninguna tecnología para móviles
- no haber desarrollado nunca un juego y
- no tener conocimientos a nivel de desarrollador sobre la sociabilización de aplicaciones.

Listado de funcionalidades inicial

Inicialmente se han establecido las siguientes funcionalidades a desarrollar para la aplicación:

- Identificación del jugador como usuario.
- Selección de contrincante: ordenador, otro jugador de la red.
- Selección de modo de juego:
 - Que el ordenador elija la combinación a descubrir.
 - Que uno de los jugadores elija la combinación y el otro la intente acertar.
 - Que cada uno de los jugadores le ponga una combinación al otro.
- Selección de nivel de juego:
- Numero de pines que formaran la combinación a descubrir.
- Numero de colores que formaran la combinación a descubrir.
- Desarrollo de juego
- Resultado de la partida
- Ranking individual y en la red según modalidad elegida (si sobra tiempo)

Desarrollo

Para poder llevar a cabo nuestro objetivo ha sido necesario cubrir todas las fases del ciclo de vida de un proyecto: estudiar la viabilidad, estudiar los riesgos, recoger los requisitos iniciales, hacer el análisis, realizar el diseño y por último implementarlo.

Fases

Requisitos iniciales

En esta fase se han recopilado los requisitos iniciales necesarios para desarrollar el juego de MasterMind tales como escenario de partida, usuarios a considerar, requisitos funcionales y requisitos no funcionales.

Para realizar esta fase nos hemos basado en la metodología indicada en el DCU (Diseño Centrado en el Usuario).

Más adelante se incluyen enlaces a juegos ya desarrollados en Android que están disponibles en Google Play y que servirán de modelo para establecer nuestros requisitos.

Análisis del sistema

En esta fase se documentará la arquitectura de la aplicación y la definición de los casos de uso.

Diseño

En la fase de diseño se dibujarán los prototipos de las pantallas del juego. Para ello se ha utilizado la aplicación Pencil.

Implementación

En esta fase se ha desarrollado el juego teniendo en cuenta las indicaciones de las fases previas. También se han realizado las pruebas necesarias para comprobar la consistencia del juego.

Planificación

A continuación incluimos el resumen de la planificación de las diferentes fases:

Título	Inicio / Enunciado	Entrega	ROL
PAC1 – Plan de trabajo	28/02/2013	11/03/2013	Jefe de proyecto
PAC2 – Análisis, diseño y prototipo	04/03/2013	08/04/2013	Analistas
Viabilidad	12/03/2013	17/03/2013	
Requisitos iniciales	18/03/2013	23/03/2013	
Análisis del sistema	24/03/2013	30/03/2013	
Diseño	31/03/2013	08/04/2013	
PAC3 - Implementación	09/04/2013	20/05/2013	Desarrolladores
Desarrollo del código	09/04/2013	12/05/2013	
Juego de pruebas	13/05/2013	20/05/2013	

REQUISITOS INICIALES (DCU)






Selección de los métodos de indagación

Hemos elegido como método de indagación el análisis competitivo o benchmarking (porque nos ha parecido interesante aprovechar que ya existen versiones de Mastermind para Android) combinado con entrevistas y observaciones contextuales.

Esto nos permite que los usuarios potenciales de la aplicación puedan realizar pruebas de las diferentes versiones del juego y opinen sobre su diseño y su funcionamiento.

Indagación

Para realizar el análisis competitivo o benchmarking hemos seleccionado los siguientes juegos de Mastermind desarrollados en Android:

ID	Icono	Nombre comercial	Desarrollador	Versión	Link
1		MASTERMIND (español)	LMAOSOFT	mastermind v 1.6	http://market.android.com/search?q=%pname:com.lmaosoft.mastermind
2		MasterMind Deluxe Free	FABRICE ADNOT	MasterMind Deluxe Free v 2.1	http://market.android.com/search?q=%pname:me.MasterMind
3		Mastermind	ALVARO QUINTANA	Mastermind v 1.11	http://market.android.com/search?q=%pname:com.alvaroquintana.mastermind
4		Mastermind (Code Breaker)	APPART	Mastermind v 4.24	http://market.android.com/search?q=%pname:com.mastermind.appart
5		Multi MasterMind	MAZESLOUP	MultiMaster v 1.1.2	http://market.android.com/search?q=%pname:fr.mazesloup.android.multi-master

No hemos utilizado guión para las entrevistas sino que han sido abiertas y las hemos complementado con observaciones contextuales.

Los usuarios que han participado de manera más habitual y activa en el análisis del juego han sido 6. Todos ellos con perfiles de edades variadas y con diferentes niveles de manejo de las nuevas tecnologías.

Las sesiones se han realizado en diferentes contextos que se detallan a continuación:

NOMBRE	EDAD	SEXO	NIVEL MANEJO TIC	CONTEXTO	DURACIÓN SESIONES
Bet	31	F	Alto	A la hora del almuerzo en el comedor comunitario de la empresa. Juegan juntos.	15'
Silvia	29	F	Alto		15'
Phil	38	M	Alto		15'
Sergio	43	M	Alto	Por las tardes en ambiente familiar. Juegan por separado	15'
Maribel	69	F	Bajo		5'
Domènec	46	M	Medio	Por las tardes.	15'

En la fase de indagación las pruebas se han realizado en el mismo dispositivo ya que los usuarios no estaban dispuestos a instalar las aplicaciones en su propio dispositivo.

El resultado de la indagación queda recogido en el apartado [Requisitos iniciales funcionales y no funcionales.](#)

Diseño conceptual (Usuarios y contexto de uso)

Perfiles de usuario identificados

Se han detectado 3 perfiles distintos de usuarios:

- Usuario de nivel bajo
- Usuario de nivel medio
- Usuarios de nivel alto

A continuación se describen las características de cada uno de ellos:

Usuarios nivel bajo		
Características del perfil	Demográficas	Edad variada, hombres y mujeres, diferentes niveles de estudios
	Intereses	Variados
	Motivaciones	Distraerse un rato. No les gustan los retos complicados. No son muy competitivos.
	Uso de la tecnología móvil	Tienen dificultad en el manejo de las nuevas tecnologías.
Contexto de uso	Donde	En casa o en lugares que les permitan concentrarse.
	Cuando	Juegan de vez en cuando. Cuando tienen ratos de tranquilidad.
	En que entorno	Prefieren el uso de las tablets.
Análisis de tareas		Seleccionar el nivel de la partida
		El modo de la partida
		Deducir la combinación ganadora
		Pausar la partida
		Reanudar la partida

Usuarios nivel medio		
Características del perfil	Demográficas	Mediana edad, hombres y mujeres, diferentes niveles de estudios
	Intereses	Variados
	Motivaciones	Distraerse un rato. Les gustan los niveles de dificultad medio. Son competitivos, pero generalmente prefieren competir contra conocidos. Les gusta estar en contacto en las redes sociales y comparar rankings.
	Uso de la tecnología móvil	Se manejan bien con las nuevas tecnologías.
Contexto de uso	Donde	En cualquier lugar, en la calle, en el transporte público, en casa, en el trabajo en los descansos.
	Cuando	Cuando tienen un momento libre.
	En que entorno	Mayoritariamente usan móvil.
Análisis de tareas		Deducir la combinación ganadora
		Pausar la partida
		Reanudar la partida
		Retar a otro usuario
		Comprobar ranking

Usuarios nivel alto		
Características del perfil	Demográficas	Mediana edad, hombres y mujeres, diferentes niveles de estudios
	Intereses	Las nuevas tecnologías. Los juegos.
	Motivaciones	Les encanta jugar. Son muy competitivos y les encantan los niveles de dificultad altos. No les importa jugar con desconocidos si les supone un reto interesante.
	Uso de la tecnología móvil	Tienen un alto nivel de conocimientos de las nuevas tecnologías.
Contexto de uso	Donde	En cualquier lugar, en la calle, en el transporte público, en casa, en el trabajo en los descansos.
	Cuando	Cuando tienen un momento libre. La media de tiempo que dedican suele ser mayor que la de otros usuarios
	En que entorno	Si disponen de poco tiempo usan móvil o tabletas. Si van a disponer de mucho tiempo prefieren juegos más sofisticados que necesitan maquinaria más potente (ordenadores de mesa o portátiles)
Análisis de tareas	Seleccionar el nivel de la partida	
	El modo de la partida	
	Deducir la combinación ganadora	
	Pausar la partida	
	Reanudar la partida	
	Retar a otro usuario	
	Comprobar ranking	

Escenarios de uso

Usuarios de bajo nivel

Maribel se levanta temprano por la mañana, hace las faenas de la casa y después coge su tableta para ejercitar su mente.

Últimamente se ha aficionado al juego de Mastermind, le gusta porque juega a su ritmo, sin presiones. Para ella es importante que las pantallas tengan un buen tamaño porque últimamente la vista comienza a fallarle un poco. También es importante poder parar la partida en cualquier momento porque recibe muchas llamadas telefónicas de amigos y familiares. Juega unas cuantas partidas y después cambia de juego. Cuando vienen sus hijos les muestra el ranking para que vean cómo ha mejorado.

Usuarios de nivel medio

A Philippe le gustan las nuevas tecnologías, por algo es informático. Tiene dos hijos con edad suficiente para jugar a juegos como el de Mastermind, le gusta que sus hijos tengan que utilizar la mente para jugar. También le gusta llevar los juegos que descubre en casa al trabajo para compartirlos con los compañeros. A la hora del desayuno reta a sus compañeros para que se echen unas partidas con él. Son ratos cortos pero, generalmente, muy divertidos.

Usuarios de alto nivel

Sergio está loco por las nuevas tecnologías, sobre todo las que le permiten jugar más y mejor. Siempre que tiene un rato muerto lo aprovecha para jugar. Siempre le ha gustado jugar largas partidas de varias horas, ahora lo hace cuando puede porque tiene un niño pequeño que no le deja muchas oportunidades. Por lo que, últimamente, se ha aficionado a los juegos de móvil que le permiten jugar a ratos cortos y que son igualmente adictivos. Es tan competitivo que seguro que intentará ser el primero en el ranking!.

Requisitos iniciales funcionales y no funcionales

Hemos observado que lo que se diferencia de unos perfiles a otros es el nivel de exigencia del juego. Algunos usuarios desean jugar en modo de único usuario y en el nivel más fácil. En cambio, otros prefieren jugar contra otros usuarios y con niveles de dificultad cada vez mayores.

Pero en lo que se refiere a las especificaciones de diseño todos los perfiles de usuario suelen coincidir.

A continuación se detallan los requisitos para cada una de las pantallas:

Pantalla principal del juego.

Los usuarios han probado la **pantalla principal del juego** y se han llegado a las siguientes conclusiones:

- La pantalla tendrá orientación vertical fija porque por las características del juego es la orientación que se le adapta mejor.
- Será deseable que la barra de estado del dispositivo móvil este visible. Resulta cómodo saber qué hora es o de cuanta batería se dispone mientras se está jugando.
- Según las especificaciones de Android la barra del menú debe aparecer arriba. De hecho solo se pondrá un botón para pausar la partida, este botón mostrara la pantalla del menú principal.
- Sería interesante que la pantalla muestre un contador de los intentos realizados y del tiempo transcurrido.
- Se ha llegado a la conclusión que los colores disponibles para formar la combinación deben estar a la vista durante toda la partida (Alguno de los juegos probados no tenían esta opción y resulta bastante incómodo jugar así ya que te obliga a recordar cuales son los colores que están en juego). Se colocaran en el lateral izquierdo de la pantalla (Se pensó en la posibilidad de colocarlos abajo del

todo de la pantalla pero cuando hay más colores en juego el ancho de la pantalla se queda pequeño).

- Se ha decidido que la forma más cómoda de seleccionar los colores para intentar adivinar la combinación es arrastrarlos desde su posición a los agujeros disponibles para formar la combinación.
- Para validar si la combinación creada es la correcta, por cada fila disponible para realizar el intento de adivinar la combinación, en el lateral derecho, aparecerá un botón con el mensaje de “Validar”. Cuando lo pulsen desaparecerá el botón y aparecerá un recuadro de resultados con los aciertos conseguidos en esa línea (Uno de los juegos mostraba los resultados automáticamente, en cuanto rellenabas toda la línea, esta opción no nos gustó porque no te permite realizar cambios de última hora, otros juegos no te indicaban como se debía validar la partida y debías deducirlo tú mismo, eran poco intuitivos).

Al recoger los resultados de estas pruebas nos hemos dado cuenta que había otros aspectos de la aplicación sobre los que no se había hablado con los usuarios. Por falta de tiempo hemos decidido basarnos en nuestra experiencia como jugadores para definir el resto de pantallas y su comportamiento.

Pantalla de menú principal.

La pantalla del menú principal tendrá las siguientes opciones

- Reanudar partida (si existe una partida pendiente) → Abrirá la pantalla de juego con la partida iniciada.
- Iniciar partida (si existe una partida pendiente avisará de que se perderán todos los datos) → Pasará a la pantalla de selección de partida.
- Ranking
- Ayuda
- Salir del juego

Pantalla de selección de partida.

La pantalla de selección de partida mostrará tres niveles: principiante, medio y avanzado.

En la del principiante se tendrá que adivinar una combinación de 4 dígitos y habrá 6 colores en juego.

En la del medio se tendrá que adivinar una combinación de 5 dígitos y habrá 7 colores en juego.

En la avanzada se tendrá que adivinar una combinación de 5 dígitos, habrá 7 colores en juego, y además, podrán repetirse colores.

Se han eliminado las opciones de modalidad de juego (local, en red) porque creemos que este juego no es el ideal para jugarlo por turnos (al estilo de apalabrados). Por lo tanto, sólo se realizaran las partidas en local.

Pantalla de ranking.

Para desarrollar la pantalla de Ranking deberemos elegir entre dos SDKs gratuitos:

SDK Skiller <http://www.skiller-games.com/developers.aspx>

SDK ScoreLoop <https://developer.scoreloop.com/>

Esta elección la realizaremos en la fase de desarrollo del juego. Se mostrará una pantalla de conexión al SDK que se haya elegido. El usuario se identificará y podrá acceder a los ránquines y al resto de opciones del SDK.

Pantalla de ayuda.

La pantalla de ayuda mostrará las instrucciones del juego.

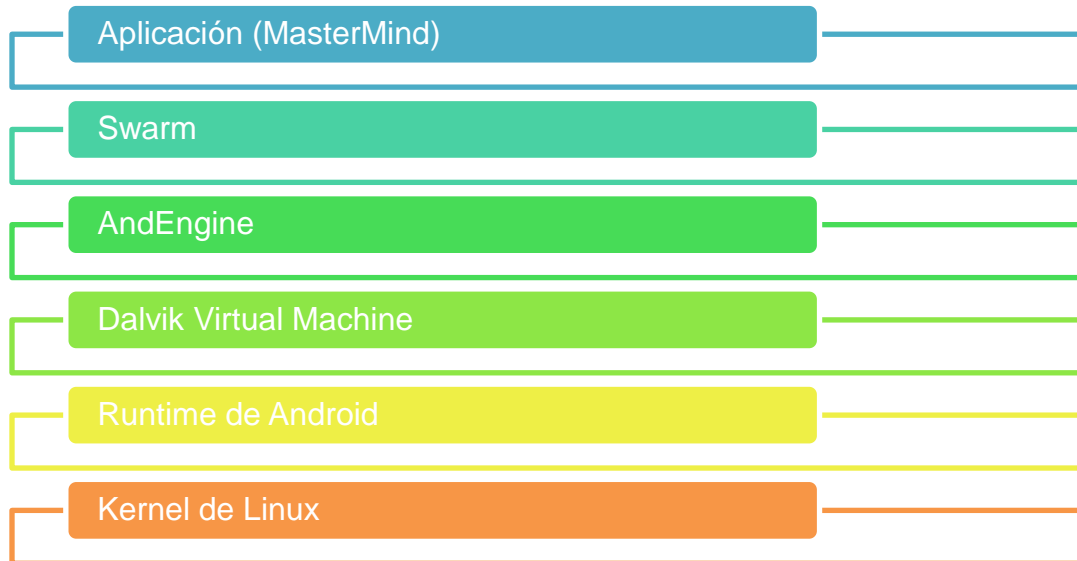
Se deberá vigilar como se ajusta el tamaño de las pantallas a los diferentes formatos, móviles, tabletas...

Será deseable que el juego disponga de los siguientes idiomas: español, inglés y catalán.

DISEÑO FUNCIONAL

Arquitectura de la aplicación

Para realizar esta aplicación ha sido necesaria la siguiente arquitectura:



El Kernel de Linux, Runtime de Android y Dalvik Virtual Machine vienen incluidos en el SDK de Android <http://developer.android.com/sdk/index.html#ExistingIDE>

La librería gratuita de AndEngine es una herramienta muy útil para desarrollar juegos OPENGL en 2 dimensiones: <https://github.com/nicolasgramlich/AndEngine>

Hay una serie de tutoriales que explican el funcionamiento de AndEngine:

- [Torres de Hanoi](#): muestra cómo arrastrar objetos por la pantalla:
- [Jumper Game](#): muestra cómo crear un juego completo con menús:
- [Un simple juego de AndEngine](#): muestra como pausar y reanudar el juego:
- También existen ejemplos de código:

<https://github.com/nicolasgramlich/AndEngineExamples>

- Y artículos que describen ciertas características de AndEngine

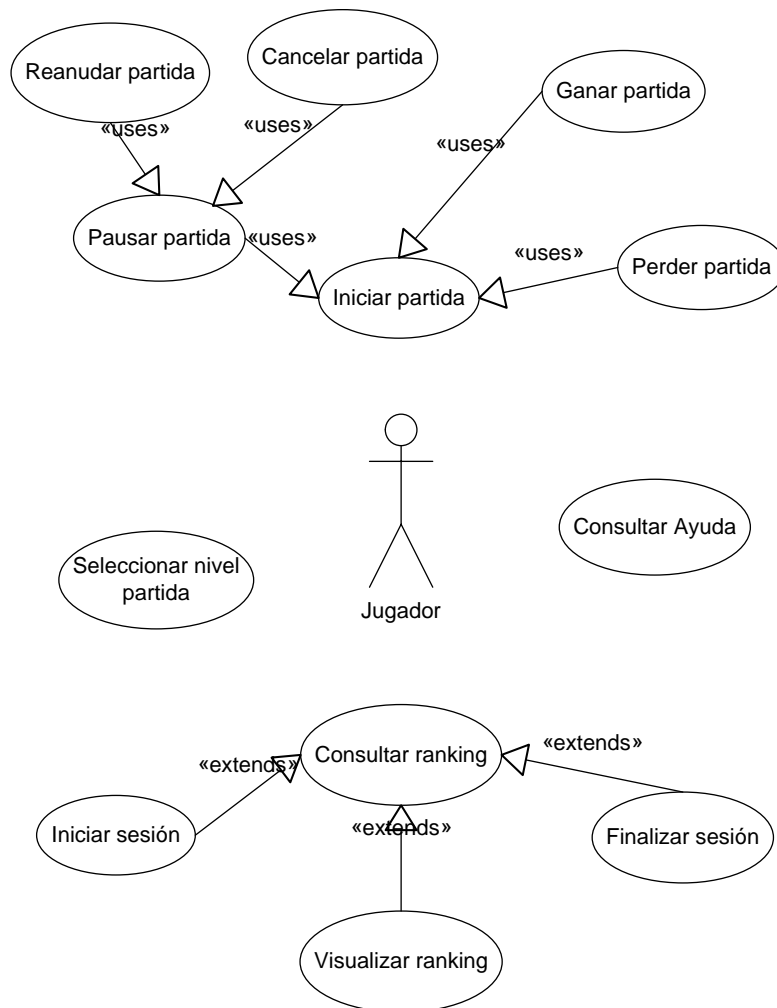
<http://andengineguides.wordpress.com/2011/09/14/getting-started-touch-events/>

<http://andengineguides.wordpress.com/2011/09/12/getting-started-working-with-sprites/>

Al final para la parte de sociabilidad hemos utilizado [Swarm](#) ya que nos ha resultado mucho mas fácil de implementar y con mejores resultados que la librería de [Skinner](#). La librería de [Scoreloop](#) no la hemos llegado a probar.

Definición de los casos de uso

A continuación definiremos los casos de uso implicados:



En este diagrama podemos identificar a los actores y casos de uso que describen los requisitos funcionales de la aplicación.

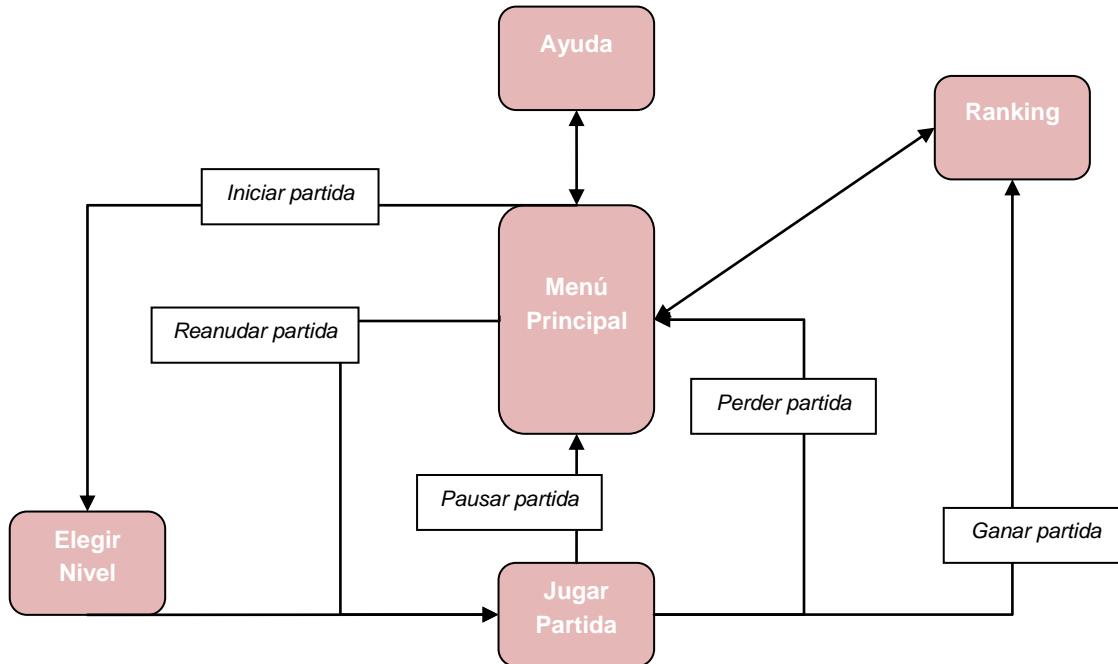
En nuestro sistema solo identificamos un actor, el usuario, que es el que interactuará con la aplicación en todo momento.

Identificamos también los siguientes casos de uso:

- **Iniciar partida:** este caso de uso muestra cómo se llevará a cabo la partida. Los usuarios podrán escoger empezar una partida nueva o continuar con la que tienen empezada.
- **Seleccionar nivel:** este caso de uso permitirá gestionar el nivel de la partida. Los usuarios podrán escoger el nivel según sus necesidades y conocimientos. Es decir, si un jugador no ha jugado nunca, debería empezar por el nivel más fácil para poder ver cómo funciona el juego y más tarde aumentar el nivel.
- **Consultar ayuda:** este caso de uso nos permitirá consultar la ayuda del juego. La ayuda del juego permitirá al usuario consultar las dudas que le puedan surgir sobre cómo jugar, los niveles del juego y como gestionar la parte social.
- **Consultar ranking:** este caso de uso nos permitirá gestionar el ranking del juego. Es decir, los usuarios tienen la posibilidad de darse de alta con distintas modalidades en la red, lo que les permitirá publicar las puntuaciones obtenidas.

Flujos de interacción

A continuación se muestra el diagrama de flujos de interacción de la aplicación.

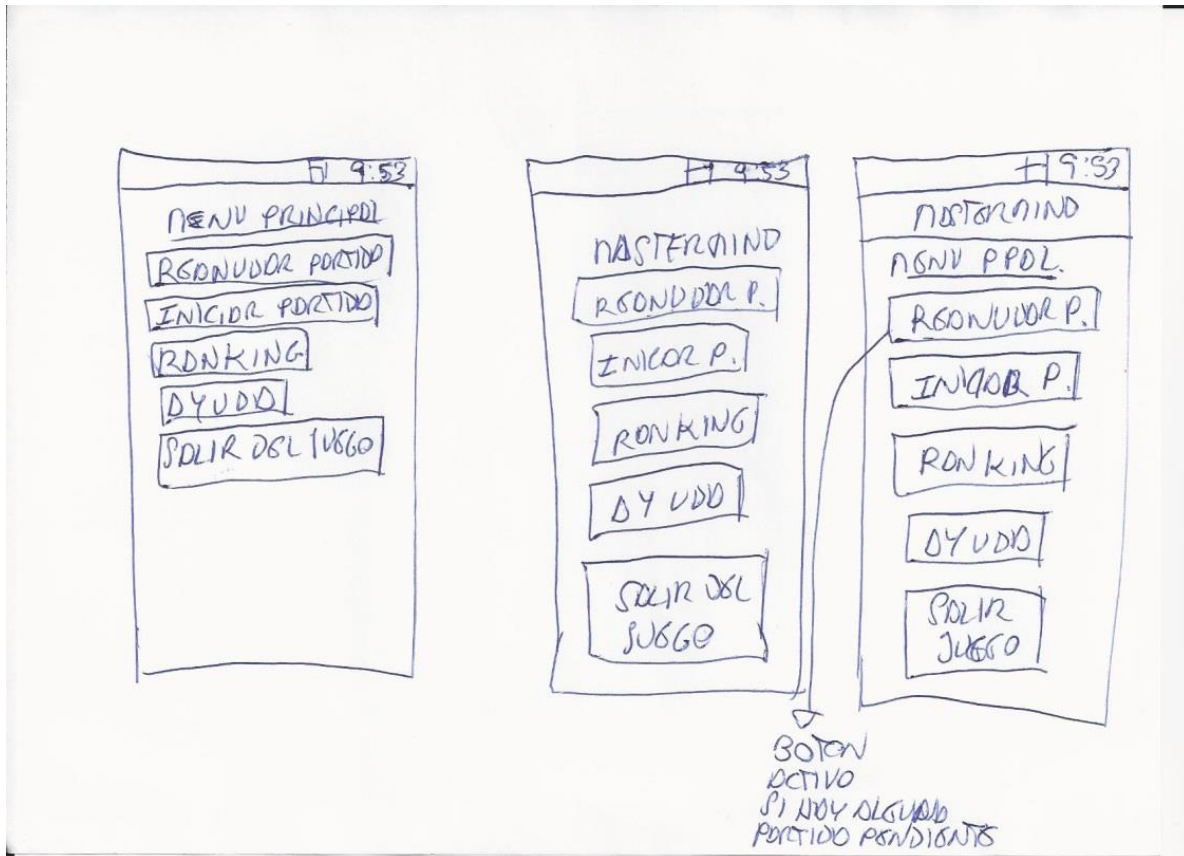


El diagrama de flujos de interacción parte del menú principal donde los usuarios podrán consultar la ayuda en caso de tener alguna duda sobre el juego. Esta pantalla solo les permitirá volver al menú principal. Podrán también consultar el ranking para ver las puntuaciones. Esta pantalla solo les permitirá volver al menú principal. Para iniciar una partida, se deberá escoger el nivel de la partida. En el momento de escoger el nivel se estará empezando una partida que podrá ganarse o perderse. Des del menú principal también podrá reanudarse una partida empezada con anterioridad que podrá ganarse o perderse.

Prototipaje

Sketches

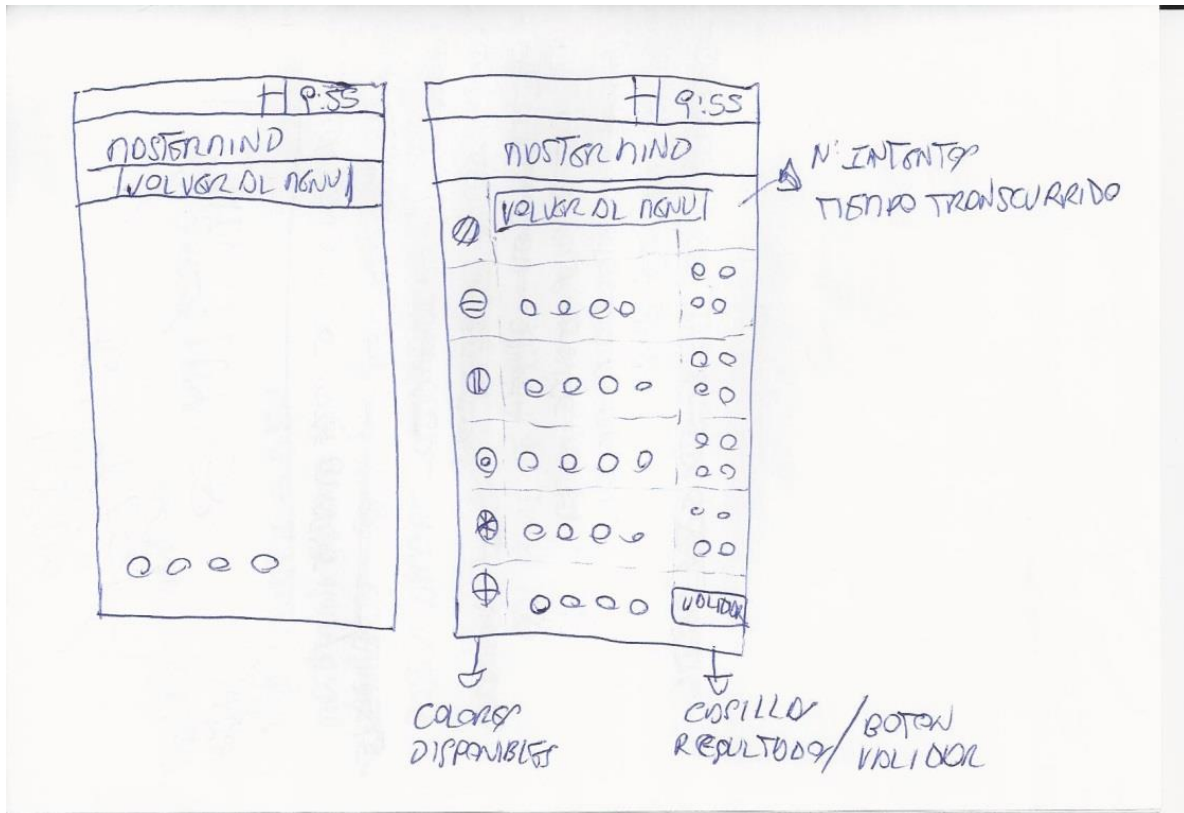
Este es el sketch del Menú principal:



En este sketch podemos ver las distintas opciones del menú principal. Estas opciones son:

- Reanudar partida (este botón solo estará activa en caso de que se haya dejado alguna partida empezada)
- Iniciar partida
- Ranking
- Ayuda
- Salir del juego

Este es el sketch de la Pantalla de juego:



En este sketch podemos ver la disposición de los distintos elementos de la aplicación. A continuación se describe como se ubican en la pantalla:

Parte izquierda de la pantalla en vertical: colores con los que se jugará la partida.

Parte derecha de la pantalla en vertical: resultado de la jugada.

Part superior derecha de la pantalla: numero de intentos que se llevan.

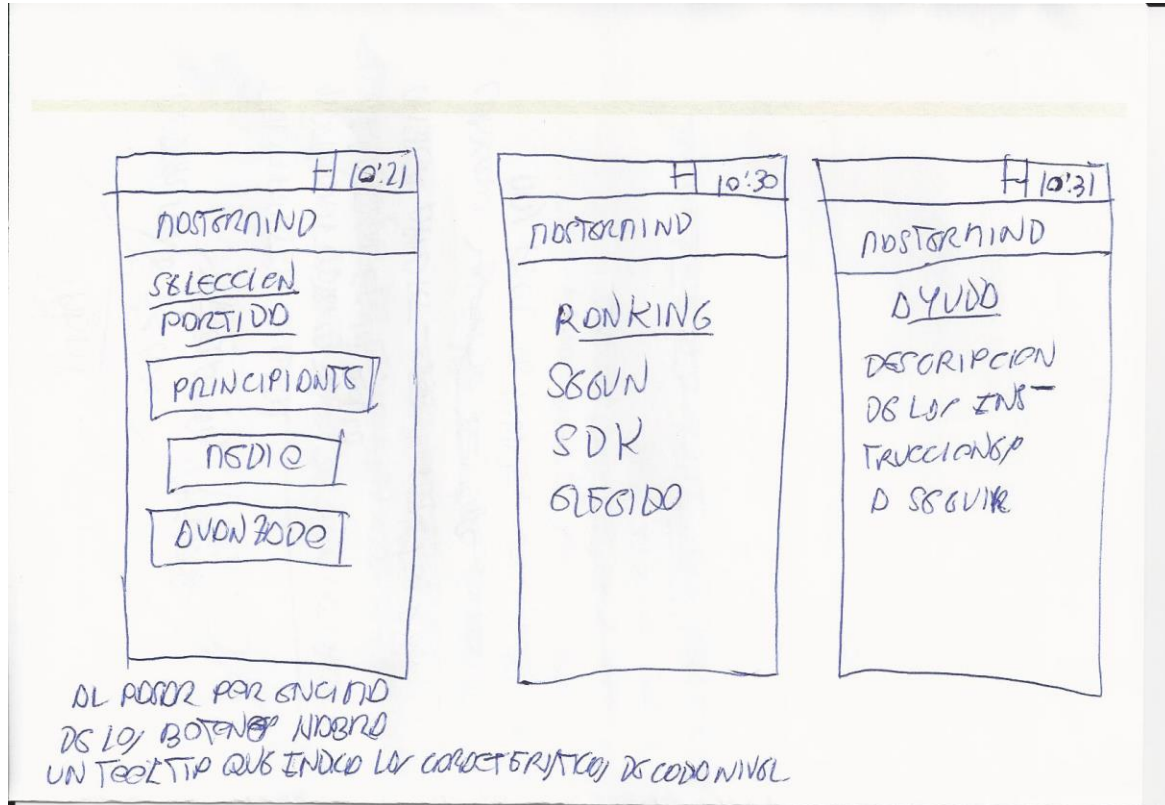
Parte central de la pantalla en horizontal: espacio para colocar los colores de la combinación escogida.

Este sketch incluye:

Selección de nivel

Ranking

Ayuda



En la pantalla de selección de nivel es donde se permite al usuario escoger con que nivel quiere jugar la partida, es decir, si quiere un nivel de principiante o un nivel medio o un nivel de experto.

En la pantalla de ranking, el usuario podrá ver en qué posición respecto de los demás usuarios se sitúa su puntuación.

En la pantalla de ayuda se podrán consultar las instrucciones del juego.

Prototipo horizontal de alta fidelidad (realizado con Pencil)

A continuación se pueden ver los prototipos de las diferentes pantallas de la aplicación.





Pantalla de Ranking1 (SKILLER)

Dashboard

skiller

Hello muginias

- Me
- Friends
- Inbox
- More Games
- Wall
- Hall Of Fame
- Store
- Coins
- Achievements

Hall Of Fame

skiller

Your position is: 11,156 50

1	Shakeeel	Pakistan	536,363
2	Neilparsec	United Kingdom	416,062
3	FeedHim	India	397,244
4	Namn	Pakistan	371,950
5	Vilio7777	Pakistan	296,104
6	KRISHNAsunar	India	108,286
7	pola6492	Pakistan	48,243
8	3993	Reserved	28,441

Pantalla de Ranking2 (SCORELOOP)

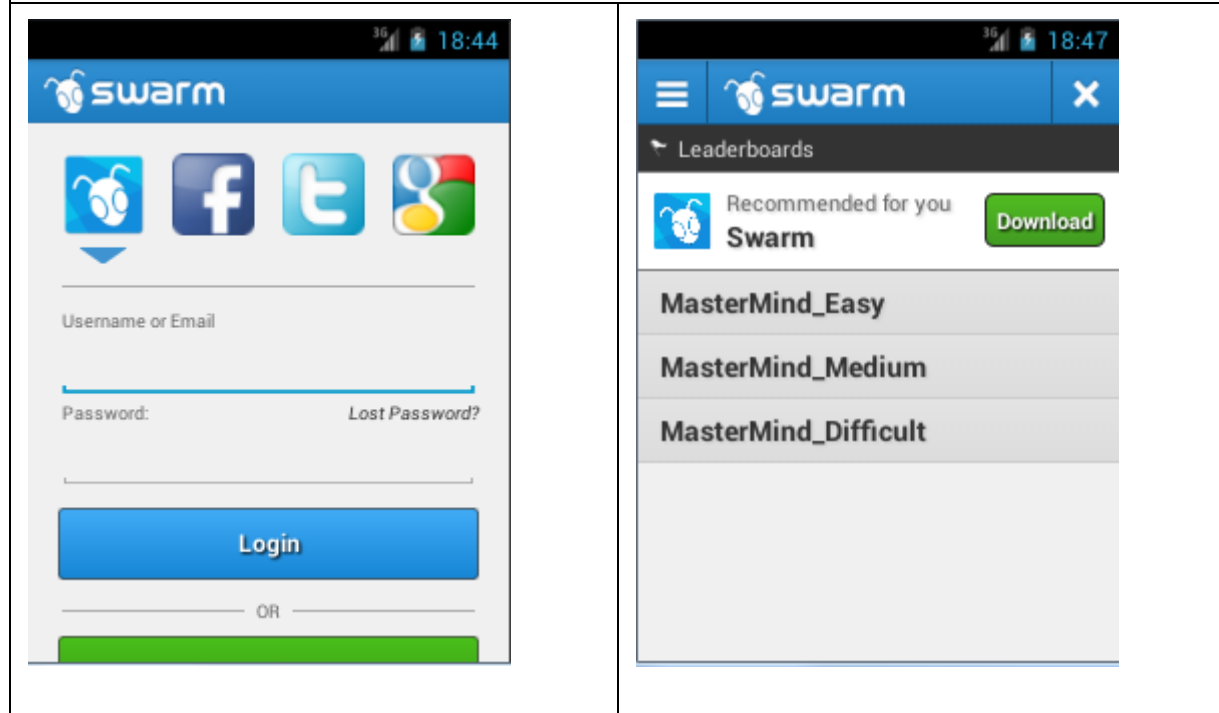
The image displays two screenshots of the Scoreloop mobile application interface. The left screenshot shows the profile of a player named "Player 4026953953". The profile includes a gear icon for settings, a "FRIENDS" count of 0, a "GAMES" count of 1, and an "ACHIEVEMENTS" count of 0. Below this, there is a "Jewels" section with a "Leaderboards" button. At the bottom, there is a promotional banner for "Love Scoreloop? Get the community app!" and a navigation bar with icons for home, friends, and achievements.

The right screenshot shows the "Leaderboards" screen for the game "Jewels" in "NORMAL mode". It features three tabs: "Global", "Friends", and "24 h". The "Global" tab is selected, showing a list of top players with their names and scores:

Rank	Player Name	Score
1	sarfaraz siddiqui	2147483642 pts
2	Ernayyouknow	2147483641 pts
3	王旭KING	2147483620 pts
4	TrapandBass@NoLife	2147483000 pts
5	Player 7443817442	2147482240 pts
6	leeloolover96	2147468635 pts
7	bidiblah	2147445049 pts
8	Player 1264708505	2147445049 pts

At the bottom of the right screenshot, a message states: "Player 4026953953 You're not yet part of this leaderboard. Improve your score!". The navigation bar at the bottom is identical to the left screenshot.

Pantalla de Ranking3 (SWARM)



Evaluación

Preguntas test

1. ¿Cambiarías el nombre del juego?
2. ¿Crees que le falta o le sobra alguna opción a la pantalla de menú principal?
3. ¿Te parecen bien los niveles de dificultad que tiene el juego?
4. Si no estás de acuerdo, ¿qué niveles propondrías?
5. ¿Te parece intuitiva la pantalla de juego?
6. ¿Qué colores elegirías para formar la combinación?
7. ¿Qué cambiarías de la pantalla de juego?
8. ¿Cómo crees que debería ser la ayuda, grafica o textual?

9. ¿Qué pantalla de ranking te gusta más?

Tareas a realizar

1. Iniciar una partida
2. Seleccionar un nivel
3. Jugar una partida
4. Pausar la partida
5. Reanudar la partida
6. Ganar la partida
7. Probar los accesos a red de los SDK y sus opciones
8. Salir del juego

Preguntas sobre las tareas

1. ¿Te ha resultado pesado iniciar una partida?
2. ¿Añadirías o quitarías algún paso del inicio de partida?
3. ¿Consideras cómoda la distancia que hay entre los dígitos de la combinación?
4. ¿Te parece adecuado el tamaño de los dígitos de la combinación?
5. ¿Te resulta cómodo la forma de rellenar la combinación?
6. ¿Se ven bien los resultados de la combinación?
7. ¿Qué herramienta de socialización te ha gustado más?

IMPLEMENTACIÓN

Instrucciones de instalación del binario

Junto con la memoria se entregará el binario, que tendrá extensión .apk, para instalar la aplicación en un móvil Android. Se puede instalar a través de dos vías:

- Correo electrónico:
 - Enviar el fichero binario por correo electrónico,
 - Abrir el correo desde el móvil
 - Y ejecutar el fichero adjunto

- Por USB:
 - Conectar el móvil al ordenador dónde este guardado el fichero binario
 - Copiar el fichero binario en alguna ubicación del móvil
 - Dentro del móvil ejecutar el fichero binario desde su ubicación

Automáticamente se comenzara a instalar la aplicación.

Nota: Puede que el sistema operativo solicite autorización para instalar aplicaciones de orígenes desconocidos.

Batería de pruebas

Dispositivos probados

Para debuggar hemos utilizado una de las configuraciones del AVD incluido en el SDK de Android.

A continuación se describen los dispositivos utilizados y la versión Android instalada en estos:

- Movil Samsung Galaxy SIII Versión Android 4.1.2
- Movil Samsung Galaxy SII Versión Android
- Movil Huawei u8650 Versión Android 2.3.3
- Movil Sony Ericsson Xperia Arc S Versión Android 2.3.4
- Tablet Samsung *Galaxy Tab 2*

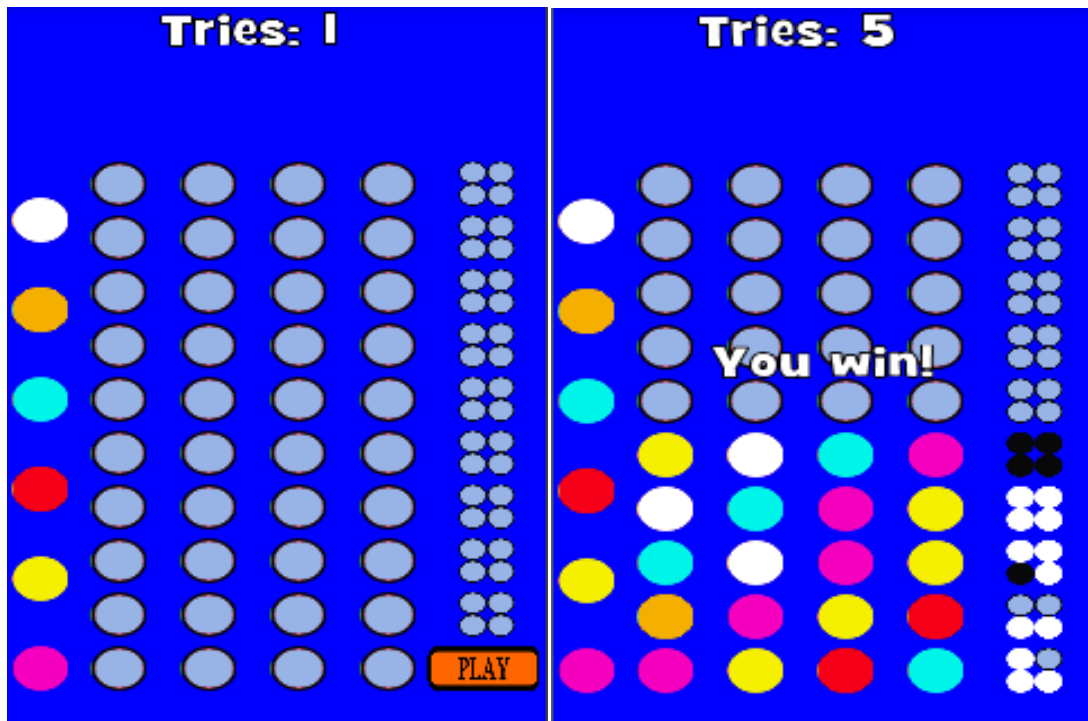
Las pantallas del juego han funcionado bien en todos los dispositivos.

No hemos podido probar Swarm en todos los dispositivos debido a que se ha implementado a última hora, pero en los dispositivos en los que se ha probado ha funcionado correctamente.

Resultados

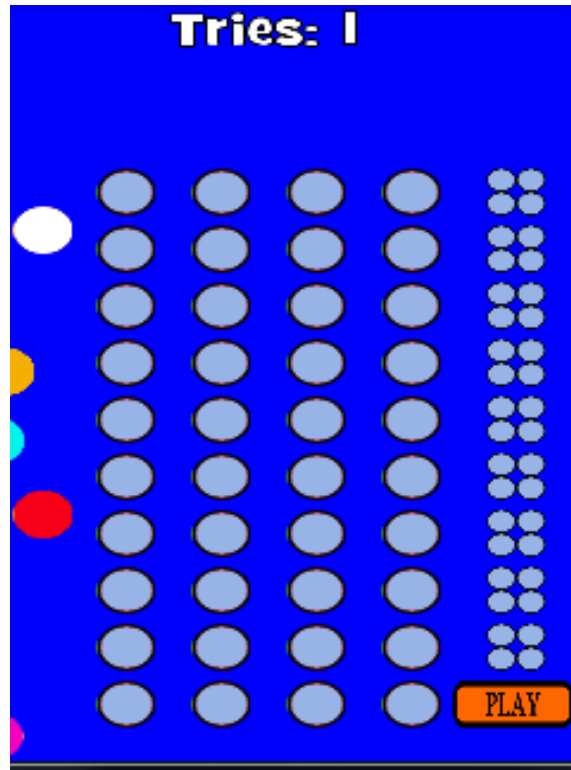
Versión 1 (MasterMind_v1.apk):

Esta es la primera versión presentada a los participantes de las pruebas durante la fase de implementación del ciclo de vida del proyecto:



Y estos son los comentarios de los usuarios:

- El color de fondo debería ser un gris claro, los colores demasiado chillones cansan.
- El fondo de los círculos debería ser blanco, se confunden con el color de fondo.
- El color de las bolas deberían ser colores básicos: rojo, azul, amarillo, negro...
- El color de los aciertos debería verse más
- Cuando sale el you win! tienes que volver a atrás para seguir jugando.
- Los colores se desplazan a la izquierda y desaparecen de la pantalla cuando los seleccionas:



Versión 2 (MasterMind_v2.apk):

Por lo que se realiza la siguiente versión que incluye los siguientes cambios:

- El color de fondo debería ser un gris claro → Modificado a un color más claro pero no gris. Se escoge un color crema.
- Los colores se desplazan a la izquierda y desaparecen de la pantalla → se arregla el problema.



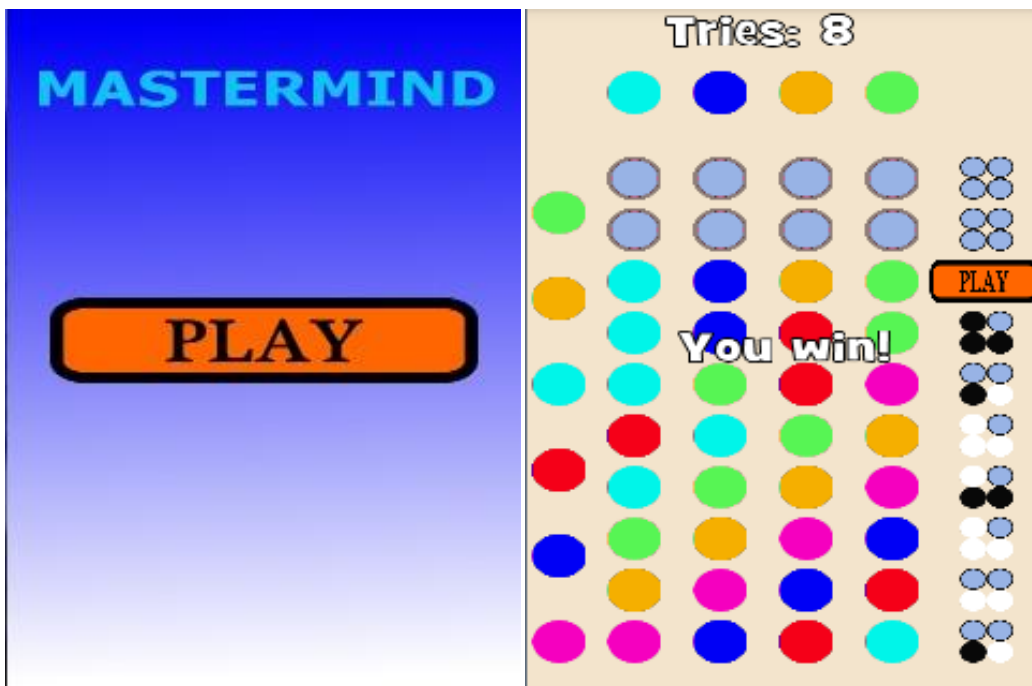
Y estos son los comentarios de los usuarios:

- Que aparezca la combinación secreta aunque se gane la partida
- Que suene una musiquilla cuando se gane o cuando se pierda la partida

Versión 3 (MasterMind_v3(PAC3).apk):

Esta es la versión presentada en la PAC3 y que incluye los siguientes cambios:

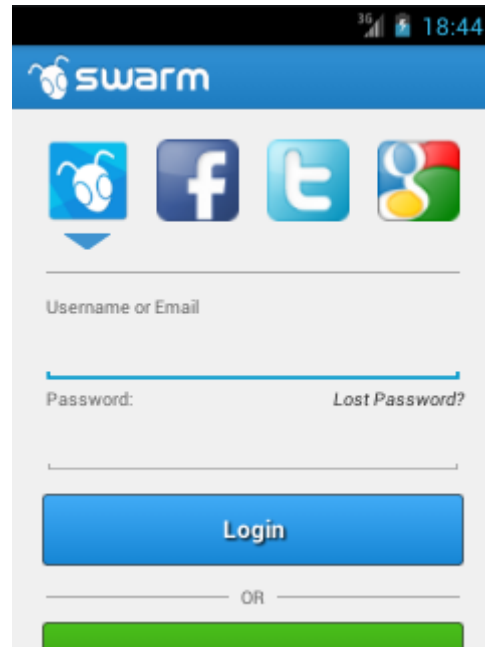
- Que aparezca la combinación secreta aunque se gane la partida → se implementa este cambio.
- El color de las bolas deberían ser colores básicos: rojo, azul, amarillo, negro... →Cambiamos los colores y además usamos codificación html para escoger colores más precisos, descartamos el blanco y el negro como colores de las bolas para no confundir con los resultados.



Esta versión no fue presentada a los usuarios por lo que no hay comentarios.

Versión 4 (MasterMind_v4(Definitiva)): Esta versión incluye lo siguiente:

- Pantalla de login de Swarm si no te has identificado ya anteriormente



- Ampliación del menú ya que incluye la parte de puntuación y la ayuda



- Traducción de los textos que aparecen a español y catalán

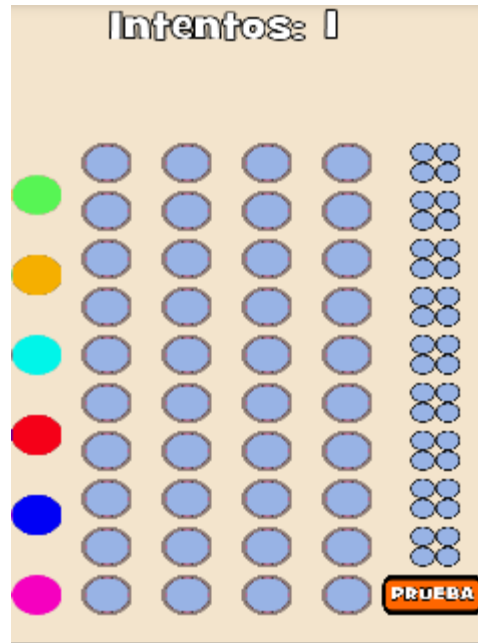


- Menú para elegir el nivel de la partida:

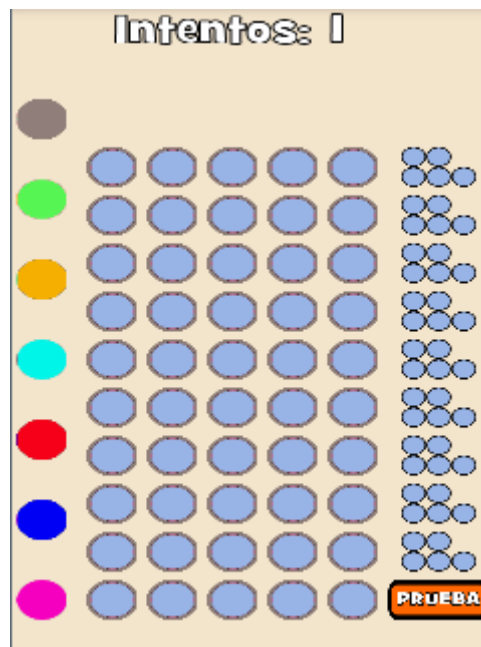


A continuación se definen los diferentes niveles de la aplicación:

- Nivel 1 del juego, 4 posiciones y 6 colores para elegir:



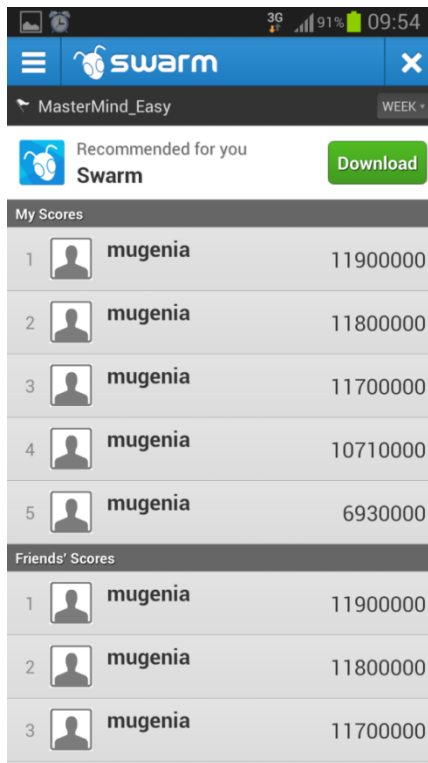
- Nivel 2 del juego, 5 posiciones y 7 colores para elegir:



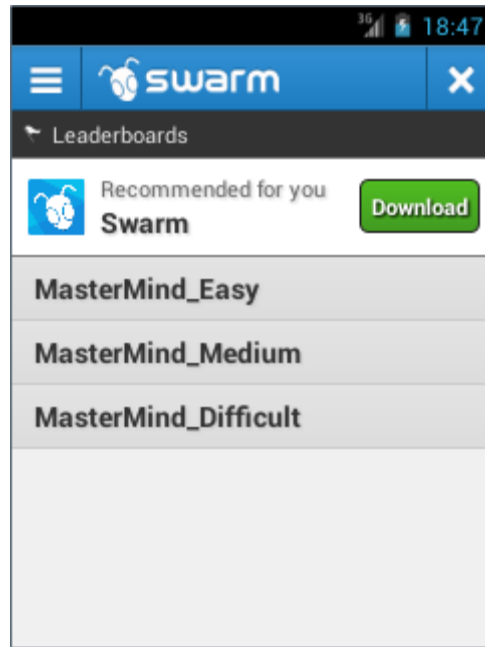
- Nivel 3 del juego, 5 posiciones, 7 colores para elegir y acepta repeticiones de color:



- Cuando sale el you win! tienes que volver a atrás para seguir jugando. → Ahora cuando ganas y estas logado, sale la pantalla de Swarm con la puntuación obtenida:



- Cuando pulsas la opción puntuación de la pantalla del Menú principal aparece la siguiente Pantalla de puntuación con Swarm:



Estos son los comentarios de los usuarios a la versión:

- Que se pueda guardar la aplicación en la memoria externa
- Que ponga has perdido, has ganado en vez de perdiste, ganaste
- Un indicador que describa el resultado de la línea validada

A continuación realizamos un resumen de todos los requerimientos solicitados durante todo el ciclo de vida del proyecto, ordenados por pantallas de la aplicación y si se han incluido o no. En la información también se indica quien lo propuso y comentarios:

Pantalla principal del juego

➤ **Requisitos Incluidos**

Orientación vertical

- Propuesto por: Consensuado por todos.

Contador de intentos

- Propuesto por: consensuado por todos.

Colores en juego a la vista

- Propuesto por: consensuado por todos.

Arrastrar los colores para colocarlos en la línea en juego

- Propuesto por: consensuado por todos.

Validar la línea en juego con un botón

- Propuesto por: consensuado por todos.

Color de fondo de un gris claro

- Propuesto por: grupo de trabajo.
- Se ha modificado el color de fondo pero no por el color indicado, sino por un color crema.
- Comentarios: Bueno, se ha elegido un color claro.

Los colores en juego deberían ser colores básicos: rojo, azul, amarillo, negro...

- Propuesto por: grupo de trabajo.

- Comentarios: A sugerencia de Silvia hemos utilizado los códigos de colores disponibles para HTML.

Los colores se desplazan a la izquierda y desaparecen de la pantalla

- Detectado por: grupo de trabajo
- Comentarios: resuelto el bug

Que aparezca la combinación secreta aunque se gane la partida

- Propuesto por: Domènec.

➤ **No incluidos**

Estado del dispositivo visible

- Propuesto por: Consensuado por todos.
- Comentarios: Las pantallas se nos desconfiguraban y hemos optado por invertir tiempo en otros requerimientos

Barra de Menú

- Propuesto por: consensuado por todos.
- Comentarios: No ha habido tiempo y tampoco se considera relevante ya que se puede pausar la partida y reanudarla más tarde sin necesidad de ubicar un botón específico para ello

Tiempo transcurrido

- Propuesto por: consensuado por todos.
- Comentarios: No resultaba tan fácil mostrar el tiempo con el mismo formato que el contador. Daba error fatal y por falta de tiempo hemos optado por implementar otros requerimientos. Sí lo hemos incluido como factor para calcular la puntuación.

El fondo de los círculos debería ser blanco

- Propuesto por: grupo de trabajo.
- Comentarios: No lo consideramos relevante. Además creemos que esto produciría confusión con los círculos de resultados

El color de los aciertos debería verse más

- Propuesto por: grupo de trabajo.
- Comentarios: No se modificarán los colores de los resultados ya que son los establecidos por las reglas del juego.

Que el nivel fácil sea más fácil: 4 posiciones y 5 colores

- Propuesto por: Silvia y Bet.
- Comentarios: Se está estudiando la posibilidad de realizar ese cambio

Cuando sale el you win! tienes que volver a atrás para seguir jugando

- Propuesto por: Silvia.
- Ahora, al ganar y si el usuario está logado previamente, la aplicación muestra la pantalla de Swarm con la puntuación alcanzada. Al salir de la pantalla de Swarm vuelve a la pantalla de juego y sigue siendo necesario volver a atrás para seguir jugando.

Que suene una musiquilla cuando se gane y otra cuando se pierda

- Propuesto por: Sergio.
- Comentarios: Pendiente.

Que ponga “Has ganado!” en vez de “Ganaste!” y “Has perdido!” en vez de “Perdiste!”

- Propuesto por: Phil
- Comentarios: No lo consideramos relevante.

Un indicador igual que el contador de intentos que describa el resultado de la última línea validada

- Propuesto por: Phil
- Comentarios: Hemos pensado que sería interesante que sólo se muestre la información, en un Toast, si el usuario pulsa sobre uno de los recuadros con resultados.

Controlar que no se valide la línea en juego si existen huecos vacios

- Propuesto por: consensuado por todos
- Comentarios: Pendiente

Validar que los usuarios no puedan repetir colores en la línea en juego en los niveles que no generan combinaciones con repetición

- Propuesto por: Phil
- Comentarios: Creemos que puede ser una estrategia del jugador poner el mismo color más de una vez en la misma línea en juego. Por ello no realizaremos ningún cambio.

Mensaje si se intenta colocar colores en otra línea que no sea la que está en juego

- Propuesto por: Observaciones contextuales
- Comentarios: pendiente

Pantalla menú principal

➤ **Incluidos**

Iniciar partida

- Propuesto por: Analista
- Comentarios: equivale a la opción del menú “Jugar”.

Ranking

- Propuesto por: Analista
- Comentarios: equivale a la opción del menú “Puntuación”.

➤ **No incluidos**

Reanudar partida

- Propuesto por: Analista
- Comentarios: No hemos incluido esta opción en el menú por falta de tiempo, y porque hemos observado que a los usuarios no les interesa especialmente salvar una partida que este a medias. Consideramos que no es el tipo de juego en el que sea importante salvar los cambios.

Ayuda

- Propuesto por: Analista
- Comentarios: Pendiente

Salir del juego

- Propuesto por: Analista
- Comentarios: No se ha considerado necesario ya que existen otros medios para salir del juego.

Pantalla de ranking

➤ **Incluidos**

- Propuesto por: Analista
- Comentarios: Al final se ha elegido el SDK de Swarm para implementar esta pantalla.

Pantalla de ayuda

➤ **No Incluidos**

Deberá incluir las normas básicas del juego.

- Propuesto por: Analista
- Comentarios: Pendiente.

Descripción de los niveles disponibles

- Propuesto por: Analista
- Comentarios: Pendiente.

Como corregir la ubicación de un color.

- Propuesto por: Observación contextual
- Comentarios: Pendiente.

Global aplicación

➤ **Incluidos**

Que acepte varios idiomas

- Propuesto por: Analista
- Comentarios: Se han incluido los siguientes idiomas: español, inglés y catalán.
- La aplicación al arrancar, automáticamente, va a buscar la configuración de idioma del dispositivo móvil para decidir que lenguaje mostrar.

Que la aplicación se guarde en la memoria externa

- Propuesto por: Bet
- Comentarios: Por defecto, la aplicación se guarda en la memoria externa.

Incluimos una tabla resumen con todos los requisitos:

PANTALLA PRINCIPAL DEL JUEGO	
Requisito	Incluido
Orientación vertical	Si
Contador de intentos	Si
Colores en juego a la vista	Si
Arrastrar los colores para colocarlos en la línea en juego	Si
Validar la línea en juego con un botón	Si
Color de fondo de un gris claro	Si
Los colores en juego deberían ser colores básicos: rojo, azul, amarillo, negro...	Si
Los colores se desplazan a la izquierda y desaparecen de la pantalla	Si
Que aparezca la combinación secreta aunque se gane la partida	Si
Estado del dispositivo visible	No
Barra de Menú	No
Tiempo transcurrido	No
El fondo de los círculos debería ser blanco	No
El color de los aciertos debería verse más	No

Que el nivel fácil sea más fácil: 4 posiciones y 5 colores	No
Cuando sale el you win! tienes que volver a atrás para seguir jugando	No
Que suene una musiquilla cuando se gane y otra cuando se pierda	No
Que ponga "Has ganado!" en vez de "Ganaste!" y "Has perdido!" en vez de "Perdiste!"	No
Un indicador igual que el contador de intentos que describa el resultado de la última línea validada	No
Controlar que no se valide la línea en juego si existen huecos vacíos	No
Validar que los usuarios no puedan repetir colores en la línea en juego en los niveles que no generan combinaciones con repetición	No
Mensaje si se intenta colocar colores en otra línea que no sea la que está en juego	No

PANTALLA MENU PRINCIPAL	
Requisito	Incluido
Inicia partida	Si
Ranking	Si
Reanudar partida	No
Ayuda	No
Salir del juego	No

PANTALLA AYUDA	
Requisito	Incluido
Deberá incluir las normas básicas del juego	No

Descripción de los niveles disponibles	No
Como corregir la ubicación de un color.	No

GLOBAL APLICACION	
Requisito	Incluido
Que acepte varios idiomas	Si
Que la aplicación se guarde en la memoria externa	Si

Detalles del código a destacar

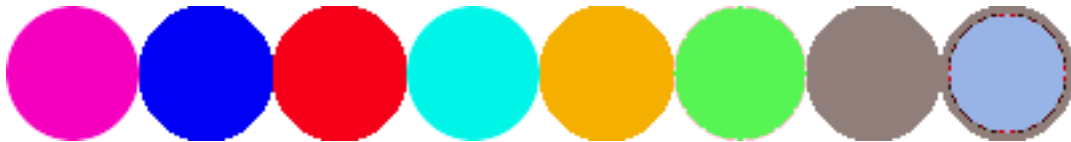
A continuación describimos algunos de los detalles del código a destacar

Pantalla principal del juego

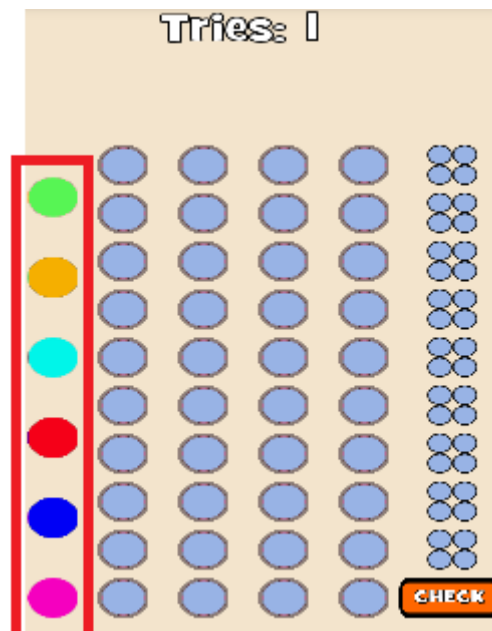
➤ Dibujo de pantalla

Para las imágenes del juego hemos dibujado dos tiledSprites y una clase personalizada (`ButtonSpriteItem`) para dibujar el botón de validación.

El primero ha servido para los colores que aparecen en el lateral izquierdo de la pantalla y para los círculos de las líneas que están en juego en el centro de la pantalla.



Para cargar los colores en el lateral de la pantalla:



Hemos usado este código:

```

/*-----*/
// Add Color pins
/*-----*/

TiledSprite[] dots = new TiledSprite[numColors];
int dot_top = 100;

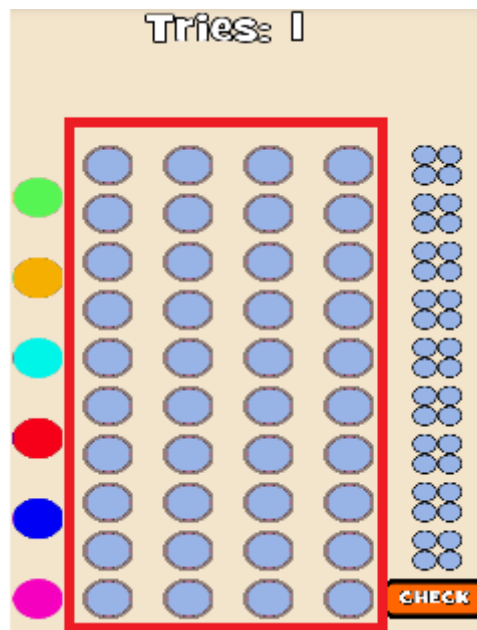
for (i=0;i<numColors;i++)
{
    left = 30;
    dots[i] = new TiledSprite(left, slot_bottom +(i*dot_top),slot_size,slot_size,resourcesManager.dots_region, vbom)
    {

dots[i].setCurrentTileIndex(i);
attachChild(dots[i]);

```

Hemos creado una matriz del tamaño del número de colores en juego según el nivel seleccionado (6,7) y la hemos ido rellenando con tiledSprites, uno para cada color en juego, y por último a cada tiledSprite le hemos asignado un color y la hemos añadido a la pantalla de juego.

Para dibujar la cuadrícula de las filas en juego:



Hemos usado este código:

```

/*-----*/
//Add Slots
/*-----*/

int left = 100;
int slot_bottom = 60;
int slot_top = 60;
int slot_size = 50;

final TiledSprite[][] slots = new TiledSprite[NUM_TRIES][numSlots];

for (i=0;i<NUM_TRIES;i++) {
    for (j=0;j<numSlots;j++)
    {
        slots[i][j] = new TiledSprite((j*slot_size)+left,slot_bottom + (i*slot_top),slot_size,slot_size,ResourceManager.getInst
        slots[i][j].setCurrentTileIndex(slots[i][j].getTileCount()-1);
        attachChild(slots[i][j]);
    }
}
}

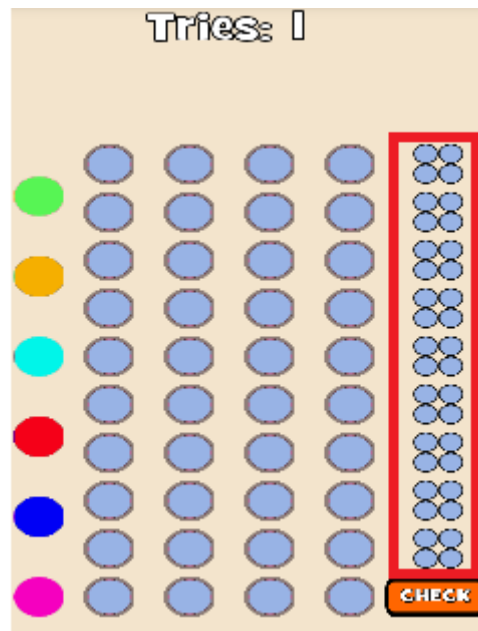
```

Hemos creado una matriz doble para dibujar la cuadrícula. Y a todos les hemos asignado el color de vacío por que al iniciar la partida no debe haber ningún color en juego seleccionado.

El segundo TiledSprite (en realidad el fondo de esta imagen es transparente pero, para mostrarlo en este documento, le hemos puesto relleno porque si no el círculo blanco no se veía):



Ha servido para dibujar en pantalla los colores de los resultados de las líneas jugadas que están en grupos de 4 o 5, según el nivel elegido, en el lateral derecho de la pantalla.



Y este es el código utilizado para dibujar los círculos de resultados en la pantalla:

```

/*-----*/
//Add NONE result Slots
/*-----*/

int result_size = 25;
int result_side = 25;
int result_bottom = slot_bottom - ((slot_size - result_size)/2);
final TiledSprite[][] results = new TiledSprite[NUM_TRIES][numSlots];

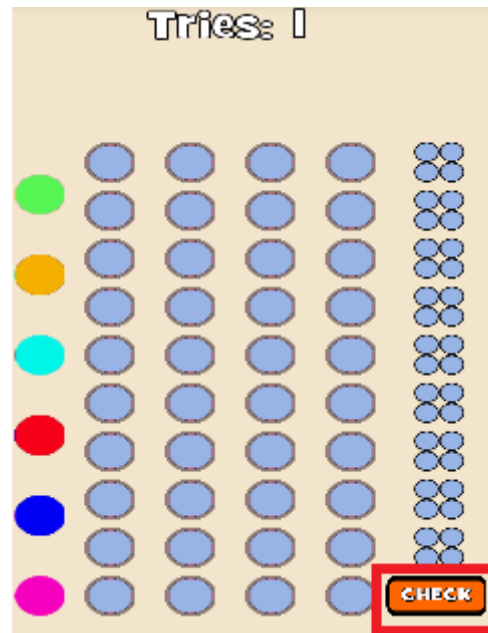
for (i=0;i<NUM_TRIES;i++)
{
    for (j=0;j<2;j++)
    {
        results[i][j] = new TiledSprite((j*result_side)+result_left, result_bottom +(i*slot_top),result_size,result_size, Resource
        results[i][j].setCurrentTileIndex(results[i][j].getTileCount()-1);
        attachChild(results[i][j]);

        //to add results in above line
        int z = j +2;
        results[i][z] = new TiledSprite((j*result_side)+result_left, result_bottom +(i*slot_top)+result_size,result_size,result_
        results[i][z].setCurrentTileIndex(results[i][z].getTileCount()-1);
        attachChild(results[i][z]);
    }
    if(numSlots == 5)
    {
        results[i][numSlots-1] = new TiledSprite((j*result_side)+result_left, result_bottom +(i*slot_top),result_size,result_size
        results[i][numSlots-1].setCurrentTileIndex(results[i][j+1].getTileCount()-1);
        attachChild(results[i][numSlots-1]);
    }
}
}

```

En este caso también hemos utilizado una cuadrícula para que se dibuje un grupo de 4 o 5 círculos, en función del nivel elegido, por fila en juego.

Y para dibujar el botón de validación:



Hemos utilizado una clase personalizada que combina un Sprite con un objeto de texto:

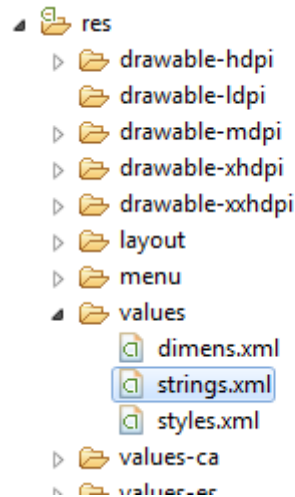
```
package com.mruberte.object;
import org.andengine.entity.sprite.Sprite;

public class ButtonSpriteItem extends Sprite {
    private float textX = 0;
    private float textY = 0;
    private Text buttonText;

    public ButtonSpriteItem(float x, float y, float pwidth, float pheight, ITextureRegion pTextureRegion, VertexBufferObjectManager
        pVertexBufferObjectManager, String pText, Font pFont) {
        super(x, y, pTextureRegion, pVertexBufferObjectManager);
        buttonText = new Text(0, 0, pFont, pText, pVertexBufferObjectManager);

        this.setSize(pwidth, pheight);
        textX = (this.getWidth()) / 2;
        textY = (this.getHeight()) / 2;
        buttonText.setPosition(textX, textY);
        buttonText.setScale(0.45f);
        this.attachChild(buttonText);
    }
}
```


Esta combinación nos permite que los dos objetos, Sprite y Text, se desplacen sincronizadamente por la pantalla y que el texto del botón cambie según el lenguaje configurado en el dispositivo móvil, en este caso se usa la cadena de caracteres check que está definida en la carpeta de recursos del idioma correspondiente:



```
<?xml version="1.0" encoding="utf-8"?>
<resources>

  <string name="app_name">MasterMind</string>
  <string name="action_settings">Settings</string>
  <string name="play">PLAY</string>
  <string name="ranking">RANKING</string>
  <string name="help">HELP</string>
  <string name="check">CHECK</string>
  <string name="tries">Tries: </string>
  <string name="gameOver">Game Over!</string>
  <string name="youwin">You win!</string>
  <string name="loading">Loading&#8230;</string>
  <string name="easy">EASY</string>
  <string name="medium">MEDIUM</string>
  <string name="difficult">DIFFICULT</string>
  <string name="rights">Color and position: </string>
  <string name="thereAre">Only color: </string>
  <string name="none">None: </string>

</resources>
```

Y esta sería la instrucción necesaria para dibujar el objeto en pantalla:

```

/*-----*/
//Add button to check
/*-----*/
final int check_left = 425;

final ButtonSpriteItem check = new ButtonSpriteItem(check_left, slots[0][0].getY(),slot_size*2,
    slot_size , resourcesManager.button_region, vbom,
    resourcesManager.activity.getText(R.string.check).toString(),
    resourcesManager.font)
{

```

➤ Movimiento de los colores por la pantalla

Para que los colores en juego se puedan desplazar por la pantalla y colocarse adecuadamente en alguna de las posiciones de la línea en juego hemos utilizado el siguiente código:

```

public boolean onAreaTouched(TouchEvent pSceneTouchEvent, float pTouchAreaLocalX, float pTouchAreaLocalY)
{
    if(!gameOverDisplayed){
        int currentTry=0;
        for (i=0;i<NUM_TRIES;i++)
        {
            if(slots[i][0].getY() == check.getY() ){
                currentTry = i;
                break;
            }
        }
        this.setPosition(pSceneTouchEvent.getX() - this.getWidth() / 2, pSceneTouchEvent.getY() - this.getHeight() / 2);
        if(this.getX() < X){
            this.setPosition(X,Y);
        }

        if (pSceneTouchEvent.getAction() == TouchEvent.ACTION_UP)
        {
            for (j=0;j<slots[currentTry].length;j++)
            {
                if(this.collidesWith(slots[currentTry][j]))
                {
                    slots[currentTry][j].setCurrentTileIndex(this.getCurrentTileIndex());
                    this.setPosition(X,Y);
                    return true;
                }
            }
            this.setPosition(X,Y);
        }
    }
    return true;
}

```

En esta función (onAreaTouched) se detecta que el usuario está tocando un color y permite que lo desplace por pantalla. Si al soltar el color contacta con uno de los círculos de la de línea que está en juego de la cuadrícula, hace que el círculo cambie al mismo color que el usuario ha desplazado hasta allí y luego el color vuelve a su sitio. Si no contacta con ningún círculo de la línea que está en juego, el color vuelve a su sitio sin realizar ningún cambio.

La función de onAreaTouched va incluida en la declaración del Sprite que se desea desplazar por pantalla:

```

/*-----*/
// Add Color pins
/*-----*/

TiledSprite[] dots = new TiledSprite[numColors];
int dot_top = 100;

for (i=0;i<numColors;i++)
{
    left = 30;
    dots[i] = new TiledSprite(left, slot_bottom +(i*dot_top),slot_size,slot_size,resourcesManager.dots_region, vbom)
    {
        float X = this.getX();
        float Y = this.getY();

        public boolean onAreaTouched(TouchEvent pSceneTouchEvent, float pTouchAreaLocalX, float pTouchAreaLocalY)
        {
            if(!gameOverDisplayed)

```

➤ Para mostrar los resultados de las líneas en juego

Para calcular los resultados obtenidos en la línea de juego, el usuario debe pulsar el botón de validación lo que desencadena la activación del código que viene incluida en la función onAreaTouched:

```

//Add button to check
/*-----*/
final int check_left = 425;

final ButtonSpriteItem check = new ButtonSpriteItem(check_left, slots[0][0].getY(),slot_size*2,
    slot_size , resourcesManager.button_region, vbom,
    resourcesManager.activity.getText(R.string.check).toString(),
    resourcesManager.font)
{
    public boolean onAreaTouched(TouchEvent pSceneTouchEvent, float pTouchAreaLocalX, float pTouchAreaLocalY)
    {
        if(!gameOverDisplayed){
            int currentTry = 0;
            if (pSceneTouchEvent.getAction() == TouchEvent.ACTION_UP)
            {
                for (i=0;i<NUM_TRIES;i++)
                {
                    if(slots[i][0].getY() == this.getY() ){
                        currentTry = i;
                        break;
                    }
                }
                currentTry = checkDots(this,slots[currentTry],code,results[currentTry]);
                if(currentTry < NUM_TRIES){
                    this.setPosition(check_left, slots[currentTry][0].getY());
                    scoreText.setText(triesText + (currentTry+1));
                }
            }
        }
    }
}

```

Al soltar el botón se comprueba cual es la línea en juego y se lanza la función checkDots que es la que dibuja los resultados en pantalla.

➤ **Para calcular la puntuación obtenida**

Para calcular la puntuación obtenida se tiene en cuenta una constante predefinida de 20 minutos convertida a segundos, que es el tiempo máximo previsto para resolver una partida.

A esta constante se le restan los segundos transcurridos hasta ganar la partida y se multiplica por el número de intentos restantes y por otra constante que son los puntos por intento restante.

```

}
else{
    gameOverText = new Text(0, 0, resourcesManager.font, resourcesManager.activity.getText(R.string.youWin).toString(), vbom);
    show_code(check, slots, code);

    if(time >= MAX_SECONDS){
        time = 1;
    }
    else{
        time = MAX_SECONDS - time;
    }

    float score= (NUM_TRIES - currentTry) * time * POINTS_PER_TRIE;

```

➤ **Para enviar la puntuación de la partida ganada a Swarm**

Para enviar la puntuación obtenida al ganar la partida a Swarm basta con añadir al código anterior la línea con el submitScoreAndShowLeaderboard indicando a que nivel de dificultad corresponde y cuál es la puntuación obtenida

```

}
else{
    gameOverText = new Text(0, 0, resourcesManager.font, resourcesManager.activity.getText(R.string.youWin).toString(), vbom);
    show_code(check, slots, code);

    if(time >= MAX_SECONDS){
        time = 1;
    }
    else{
        time = MAX_SECONDS - time;
    }

    float score= (NUM_TRIES - currentTry) * time * POINTS_PER_TRIE;

    SwarmLeaderboard.submitScoreAndShowLeaderboard(LEADERBOARD_ID, score);

```

CONCLUSIONES

A lo largo de este proyecto hemos implementado con éxito una aplicación para dispositivos móviles con sistema operativo Android, lo que nos ha permitido conseguir los objetivos que fueron marcados al comienzo de este documento.

El SDK de Android nos ha ofrecido todos los recursos necesarios para que nuestra aplicación fuera totalmente funcional.

Hemos integrado con éxito las librerías de AndEngine que nos han ofrecido un entorno de desarrollo para juegos completamente funcional, muy asequible y escalable en cuanto a aprendizaje.

También hemos podido integrar las librerías de Swarm con muchísima facilidad, obteniendo resultados inmediatos.

La metodología y planificación utilizadas también han jugado un papel importante para desarrollar nuestro proyecto, lo que nos ha permitido llevar un ritmo constante a lo largo de todo el proyecto.

A nivel personal, nos ha resultado muy enriquecedor. Hemos conocido muchas herramientas que nos pueden resultar muy útiles para proyectos futuros, personales o profesionales. De hecho, consideramos este trabajo, dentro del universo de la informática, como la puerta de entrada a un mundo por descubrir muy interesante.

BIBLIOGRAFIA

Informática hoy

Artículo página web: "¿Cuál es el mejor sistema operativo para un smartphone?"

Fecha consulta: 11/03/2013

<http://www.informatica-hoy.com.ar/soluciones-moviles/Cual-es-el-mejor-sistema-operativo-para-un-smartphone.php>

El androide libre

Artículo página web: "Un repaso a la historia de Android"

Fecha consulta: 11/03/2013

<http://www.elandroidelibre.com/2011/12/un-repaso-a-la-historia-de-android.html>

El androide libre

Artículo página web: "Esto es sólo el comienzo. La irrupción de los smartphones en nuestras vidas"

Fecha consulta: 11/03/2013

<http://www.elandroidelibre.com/2012/01/esto-es-solo-el-comienzo-la-irrupcion-de-los-smartphones-en-nuestras-vidas.html>

Developer Android

Artículo página web: "Get the Android SDK"

<http://developer.android.com/sdk/index.html>

Skiller

Artículo página web: Skiller Developer tools

<http://www.skiller-games.com/developers.aspx>