



INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

JUEGO PSP: MONOPOLY

Realizado por:
Jaime Bartolomé León Nadal
Israel Camacho Ruano

Dirigido por
Jose Ramón Portillo Fernández

Departamento
Matemática Aplicada I

Sevilla, JUNIO de 2011

Agradecimientos

Es difícil acordarnos y más nombrar a todas aquellas personas las cuales han puesto su granito de arena para animarnos y ayudarnos para poder acabar este trabajo, así que queremos dar las gracias sobre todo a:

Nuestras familias por ese apoyo incondicional en los momentos buenos y no tan buenos que hemos pasado frente al ordenador para poder finalizar este juego, a nuestro jefe de proyecto José Ramón Portillo Fernández, por su guía y camino mostrado para realizar un buen trabajo, nuestros amigos que nos aconsejaban en las posibles mejoras del juego y por último a toda persona anónima en la red que solucionan aquellos problemas que no consigues resolver a cambio de nada.

En definitiva, muchísimas gracias a todas las personas que han colaborado con nosotros para que podamos desarrollar este juego para una consola portátil.



ÍNDICE DE CONTENIDO

Resumen.....	1
Introducción.....	2
<i>Objetivos</i>	3
Análisis del Proyecto.....	4
<i>El juego</i>	4
Desarrollo del Proyecto.....	5
<i>Fases de desarrollo</i>	5
Diseño e Implementación.....	7
<i>Main.c</i>	8
<i>Atributos</i>	9
<i>Funciones</i>	10
<i>Métodos</i>	13
<i>Funciones (osLib)</i>	14
<i>Métodos (osLib)</i>	16
<i>Tablero.h</i>	
<i>struct</i>	23
<i>enum</i>	24
Herramientas.....	25
<i>Software de desarrollo</i>	26
<i>Software diseño gráfico</i>	27
Tiempo empleado.....	28
Conclusión.....	29



Juego Psp: Monopoly

III

Anexos.....	30
<i>Anexo 1: Liberar una PSP</i>	<i>31</i>
<i>Anexo 2: Instalación y ejecución del juego.....</i>	<i>43</i>
<i>Anexo 3: Instalación librería, Visual Studio y Compilador.....</i>	<i>45</i>
<i>Anexo 4: Manual de usuario, como jugador.....</i>	<i>47</i>
<i>Anexo 5: Reglas de Monopoly.....</i>	<i>58</i>



Juego Psp: Monopoly

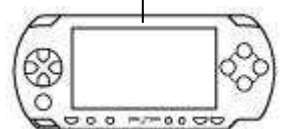
1

RESUMEN

El proyecto se basa en la creación del mundialmente conocido juego de mesa Monopoly adaptado a la consola portátil PSP de Sony.

Para ello hemos representado todas las operaciones que se pueden realizar, como son la subasta, la negociación, turno de tiradas, la construcción de viviendas y demás acciones que se ven en el juego. Es posible jugar una partida, contra uno o más jugadores controlados por la máquina, en la cual su inteligencia se basa en una pila de condiciones en las cuales se irán tomando decisiones a medida que avanza el juego, pudiendo entrar también en las negociaciones y subastas mencionadas anteriormente.

La finalidad del juego, que uno de los jugadores de la partida se haga con el mayor número de propiedades, con el mayor número de casas, para así llevar a todos los demás jugadores a la quiebra y tengan que abandonar la partida. El que lo consiga gana la partida. Otra opción de juego que no viene reflejado en este trabajo sería una partida con el mismo objetivo, excepto que ésta está limitada por un reloj configurable, cuando termina la cuenta atrás ganará el que más dinero y propiedades disponga.



INTRODUCCIÓN

Un videojuego es una aplicación realizada por un ordenador, a través de los programadores y diseñadores gráficos, con la cual pueden jugar un número limitados de personas, el primer objetivo es el entretenimiento. Un videojuego se muestra en un aparato electrónico (ordenador, videoconsola, smartPhones) donde los jugadores envían las funciones a realizar por la máquina, ésta le muestra al usuario una parte gráfica al instante.

Los primeros videojuegos modernos aparecieron en Estados Unidos en la década de los años 60 y desde entonces no han parado de avanzar y desarrollarse tecnológicamente. Hoy día existe muchísimos tipos de juegos, que van desde los juegos de mesas de toda la vida, aventuras gráficas, deportes, agilidad mental, plataformas y unos de los más cotizados hoy en día los shoot'em up (En el punto de mira). Todos estos tipos de juego se han ido adaptando a los avances hardware que tenía la máquina, mejorando tanto gráficamente como en jugabilidad.



Objetivos

- *Conocer y aprender como instalar aplicaciones caseras a PSP*
- *Aprender a desenvolverse con la librería OSLIB del lenguaje C.*
- *Aprender sobre el compilador que crea los ficheros necesarios para volcarlos a la PSP:*
 - *makefile*
- *Entender la funcionalidad de cada fichero creado:*
 - *Eboot.pbp*
 - *Botones.elf*
 - *Param.sfo*
- *Dejar lo mejor posible explicado el trabajo para su mejora a los próximos creadores de nuevos proyectos para la facultad.*



ANÁLISIS DEL PROYECTO

El juego

La aplicación es una adaptación al juego Monopoly, basado en el juego de mesa creado a principios del siglo XX, por Elysabeth Maggie, The landlord's game.



Imagen 1: Juego Monopoly en la PSP

Las instrucciones y las explicaciones del juego se encuentran reflejadas en anexos del documento, se puede mirar en el índice de éste.



DESARROLLO DEL PROYECTO

Fases de desarrollo

Los pasos que hemos seguido para la realización de este proyecto lo explicamos en las próximas páginas de este documento.

Iniciación

Lo primero que realizamos fue aprender a introducir aplicaciones caseras en la PSP, intentando comprender el compilador de la consola, preparar el medio de desarrollo , adentrarnos en el lenguaje de programación C y la librería soportada por dicha consola, OSLIB.

Compilador

Una de las cosas que nos han resultado más complicado ha sido la comprensión del compilador del ordenador, para que nos creara los ficheros necesarios para poder instalarlos en la PSP, aunque finalmente utilizásemos un compilador automático existe un modo de hacerlo todo manual, para ello necesitamos configurar el archivo makefile.

Prácticas con lenguaje y librería

Para ganar soltura a la hora de programar la máquina, empezamos a realizar cursos y ejemplos de dicho lenguaje encontrados en las diferentes páginas web en que navegábamos por internet, como pueden ser:

- <http://scenextreme.com/scenextreme/foro/index.php?PHPSESSID=lb1tk6nhd6vj3fpv7d7mq64pn6&board=19.0>



- <http://psp.scenebeta.com/tutorial/tutoriales-programacion-psp-y-avanzados-de-homebrew>

Imágenes del proyecto

Como veremos más adelante en la explicación de la librería y el funcionamiento de un juego en la PSP, procedimos a buscar y escanear todas las imágenes necesarias que formarán parte de nuestro proyecto.

Programando las condiciones

Una vez tuvimos todas las herramientas, imágenes y materiales necesario, procedimos a pasar a código todas las condiciones y restricciones que nos indicaba el juego, a cargar todas las imágenes en la memoria de la consola, y a empezar a darle forma al juego que finalmente está a vuestra disposición.

Programando jugadores computadora

Cuando tuvimos todo listo para que la partida las jugasen los jugadores reales, pasamos a que las tomas de decisiones que tienen los jugadores humanos las emulase la máquina dependiendo, del interés que tenga.

Probar el juego

Finalizada la aplicación e introducida en la consola probamos que su funcionamiento es correcto y el esperado, sin que se produjese ningún error y que el juego fuese fluido.



DISEÑO E IMPLEMENTACIÓN

En esta sección de la documentación abordaremos los diferentes métodos y funciones que utilizamos en el trabajo realizado tanto en la librería especial de OSLIB, como algunas de las funciones hechas por nosotros, más especiales que son un poco más complejas. Como ya mencionamos lo más utilizado será el lenguaje de programación C y una librería en concreto, aunque existes otros que pueden ser utilizados para la conseguir una aplicación como la que hemos creado:

Lenguajes: C++, PHYTON, LUA

Librerías: SDL, OSLIB, BULLET PHYSICAL (Todas ellas para C y C++).

En el siguiente enlace que le mostramos usted podrá ver con más detalle la composición de esta gran librería que nos facilita mucho la vida a la hora de programar para PSP:

<http://oslib.playeradvance.org/doc/>

A continuación se le explicará el contenido de los dos grandes archivos utilizados en el proyecto, con una imagen y nos introduciremos en ellas más profundamente para explicar tanto los instrumentos de OSLIB como las funciones y métodos creados por los autores.

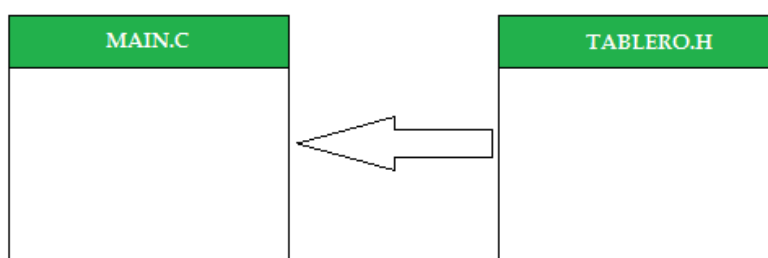
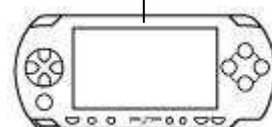


Imagen 2: Bloques que componen el proyecto

En la imagen 2, podemos ver los dos grandes documentos que forman partes del proyecto.



Dentro del archivo fuente, estarían todas las funciones y métodos utilizados en el monopoly, y todo el funcionamiento de este como puede ser: tirar los dados, subasta, compra de una propiedad, negociaciones.... Pero también están contenidos todos los componentes que tuvimos que coger de OSLIB, como: cargar imágenes en la memoria RAM, controlar las teclas pulsadas, escribir en la pantalla, cargar la música del juego; en definitiva todas aquellas acciones que configuran la consola, pero esas opciones serán mejores definidas en los anexos posteriores.

En el archivo de cabecera sin embargo contendrá todos los tipos, estructuras, enum y demás propiedades que vamos a utilizar en el proyecto: estructura Casilla, estructura Precio, enum tipo casilla...

Main.c

MAIN.C
FUNCIONES PROPIAS <ul style="list-style-type: none">- void Botones()- void dibujarJugadores(int)- int tirarDado(int)- void suerte(int)- void comunidad(int)- void subasta(int)- void construir(int)- void datosJugadores(void)- void pagar(int)- void accion (int)- void comprar(int)..
FUNCIONES OSLIB <ul style="list-style-type: none">- void oslQuit()- *OSL_IMAGE (char*,int,int)- oslCls()- oslStartDrawing()- oslSyncFrame()..

Imagen 3: Algunas funciones de Main.c



Antes de nada explicar que el código fuente está todo contenido en un bucle que se repetirá continuamente hasta que no termine la partida o se salga del juego. Como podemos observar el archivo Main.c es el alma de este proyecto, engloba desde las imágenes y música del juego hasta toda la lógica y condiciones que lo componen. Pasemos a ver alguno de los atributos, métodos y funciones más importantes creadas y todas aquellas que hemos necesitado de la ya conocida librería.

Atributos (Propios)

- **struct Jugador j[6]** → Se trata de un Array, que independientemente del número de jugadores (el máximo son 6) que hayan en la partida, contendrá para cada uno de ellos las posiciones de dicho Array, todo tipo de informaciones y datos sobre cada jugador, que irá actualizándose a medida que avanza la partida (En el juego la ficha es igual a la posición + 1)
- **struct casillas_calles tablero[40]** → En este caso, el Array va asociado como se puede ver al número de casillas en total que existen en el juego; aunque no todas sean del mismo tipo las consideramos todas en un mismo Array, que al igual que en el atributo anterior se irá actualizando y dando información a la maquina y jugadores a medida que avance el juego.
- **int casasBanca, hotelesBanca** → Comienzan con un valor definido, que irá disminuyendo su valor a medida que se construyan las viviendas en las distintas propiedades, en caso de ser 0, supondrá que la banca no dispone de más y por tanto, no se puede construir más hasta que un jugador los venda.
- **int t** → Variable que marcará quién es el siguiente en tirar los dados.



- **int dobles** ➔ Variable que tomara el valor 0 o 1, para saber si el jugador que está tirando los dados debe volver a tirar o le tocará al jugador siguiente en la partida (incrementando la variable **t** o dejándola tal como está).

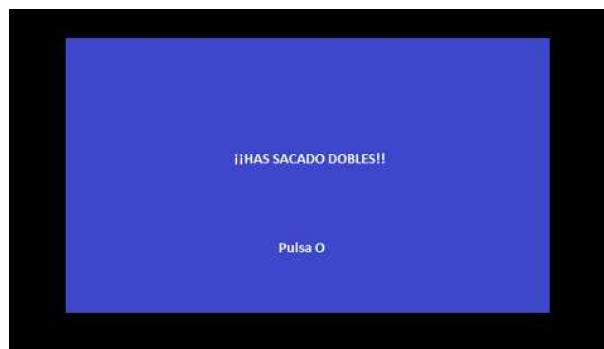


Imagen 4: Dibujo en la pantalla de PSP al sacar dobles

- **int pilaSuerte[16]** ➔ Guardará los valores aleatorios que nos entregue la función de **iniSuerte()**, que se ejecutará al comienzo de cada partida, y sus valores tendrán un orden distinto en cada una de ellas. Para comunidad existe una misma pila.

Variables propias existen muchas más, pero estas pueden ser según el punto de vista de los programadores las más utilizadas dentro del juego.

Métodos y Funciones (Propias)

Pasaremos a explicar con más detalle algunos de los métodos y funciones, pero no todos los que utilizamos para realizar este juego.

- **void Botones()** → método del trabajo que reúne todas las posibles teclas que podemos pulsar, con cada uno de los botones de la consola, tenemos definida una acción a realizar. Al estar en un bucle infinito, estará continuamente leyendo las teclas hasta que se pulse una, por ejemplo, al pulsar triangulo tiraremos los dados.
- **int tirarDado(int)** → como hemos dicho antes, al pulsar triangulo se tirará los dados, el valor que se le pasa es el jugador al cual le toca tirarlos, para conseguir unos valores aleatorios utilizamos, las funciones `srand(time(NULL))` y `rand%6 + 1` en dos ocasiones (consiguiendo un número comprendido entre 0 y 6).



Imagen 5: Muestra la tirada de dados

- **void iniJugador()** → configura todas las propiedades de cada struct jugador que están contenidos dentro del Array que vimos anteriormente.
- **struct casillas_calles rellena_Tablero(int)** → este método configura todas las casillas del tablero (carga precio de las casillas, hipoteca, propietario inicial) con la información necesaria para comenzar el juego.



- **void irCarcel(int)** ➔ aunque parezca un método simple, dentro de éste, se realizan varias acciones, como dibujar la ficha en la posición de la cárcel y configurar los atributos del jugador por caer en la cárcel.
- **void subasta(int)** ➔ el valor que se le pasa a este método es la casilla por la que se va a realizar la subasta. Una vez dentro del método este únicamente realiza un bucle, en el cual preguntará en orden a cada jugador si está interesado en pujar, si es así, pasaría a marcar la cantidad que estaría dispuesto a pagar. Continuará hasta que el número de jugadores de la subasta sea igual a 1, en ese caso el propietario será el que mayor dinero haya ofrecido.
- **void negociar(int)** ➔ método que entrará en acción cuando uno de los jugadores quiera comprar a otro una propiedad de la que él estuviera interesado. El valor pasado no es más que el jugador que hace el ofrecimiento.
- **void venta(int)** ➔ como parámetro se le envía el jugador que quiere realizar cualquier tipo de venta, este método mostrará todas las propiedades que tiene en su poder dicho jugador. La elegida se le quitará de las propiedades que le pertenecen y se le sumará a su capital, el dinero de la hipoteca de ésta.

Atributos (Librería)

- **OSL_IMAGE *mapa** ➔ existen infinidad de variables dentro de nuestro proyecto. Esta variable no es más que un atributo, que carga una imagen en la memoria de la consola.



- **OSL_CONTROLLER *osl_keys()** → es una variable que también se carga en la memoria donde se captura la última tecla pulsada por el usuario.(square, cross, triangle, circle, start, select, home, up, down, left, right ...)
- **OSL_SOUND *musica** → variable que contendrá la música de fondo del juego, una vez que la desactivemos no se podrá volver a cargar.

Estas son las variables más usadas de OSLIB, el proyecto contiene alguna más que explicaremos más adelante en el anexo de DOCUMENTACIÓN OSLIB.

Funciones y métodos (Librería)

Pasaremos a ver las funciones y método más utilizados en la librería que nos facilita la programación de cualquier aplicación para nuestra consola portátil.

Aunque en esta documentación no explicaremos todas las utilidades de esta librería, entraremos en detalle en muchas de ellas, que son las necesarias como mínimo para poder empezar a programar la PSP.



Funciones (Oslib)

- **OSL_IMAGE osLoadImageFile (char *direccionImagen, int localizacion, int formatoDePixel)** ➔ con esta función conseguimos cargar en una de las memorias de la consola (RAM o VRAM) las imágenes que vayamos a cargar en nuestro juego o aplicación. A esta función hay que pasarle otros atributos que pasaremos a explicar con detalle:
 - **direccionImagen** ➔ este parámetro hace referencia a la ubicación en la que se encuentra la imagen que queramos cargar, una vez cargada tendremos que comprobar su correcto funcionamiento. Por mantener un orden todas las imágenes serán con un mismo formato .PNG (Gráficos de red portátiles). Un ejemplo para entenderlo sería si tenemos una imagen llamada "prueba.png" en el disco duro C:. La ruta debería ser "C:/prueba.png".
 - **localizacion** ➔ Aquí tendremos que decir en cual de las dos memorias de la consola queremos guardar la imagen. Las dos posibles elecciones son:
 - **OSL_IN_VRAM** ➔ es la memoria reservada específicamente para gráficos, qué tan solo dispone de 2MB de capacidad aunque su velocidad es mayor que la memoria RAM.



- *OSL_IN_RAM → memoria de acceso aleatoria. Aunque esta sea de una velocidad menor, es más apropiada que la otra, ya que su capacidad es 10 veces mayor.*

Nosotros por capacidad (utilizamos una gran cantidad de imágenes), más que por velocidad, utilizaremos la opción OSL_IN_RAM.

- **formatoDePixel** → *con este parámetro el número de bits que acepta el aparato al que vayamos a introducir nuestro trabajo, en la PSP se dispone de una pantalla de 32 bits. Existen varias opciones:*
 - *OSL_PF_8888 → opción para el caso de una pantalla de 32 bits, éste será el valor del parámetro utilizado para trabajar con una PSP.*
 - *OSL_PF_5650 → este caso se pondría para el caso en que la pantalla fuese de 16 bits, en nuestro caso, no la utilizaremos para nada.*

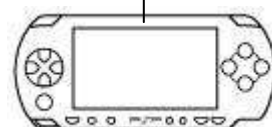


Métodos (Oslib)

- **PSP_MODULE_INFO (Char *nombre_Proyecto, 0, 1, 1) →** aunque no conocemos demasiado de este método, sabemos que es para obtener el archivo (EBOOT.PBP), este sería el juego compilado para que la consola lo reconozca. El parámetro nombre, sería el nombre del proyecto, y los números siguientes pasados como parámetros serían siempre igual, que vendrían a decir qué versiones de consola soporta nuestra aplicación, con esta configuración, el juego sería aceptado por toda máquina PSP.
- **oslInit (int banderas) →** inicializa la librería para empezar a trabajar. Para gente no experta, se le recomienda pasar como parámetro el valor 0.
- **oslInitGfx (int formatoDePixel, int dobleBufer) →** activa la parte gráfica de la librería. El primero de los parámetros, ya fue explicado con anterioridad. El segundo parámetro que se le pasa, prácticamente es un booleano, al cual se le pasará el valor 1 en caso de que queramos que se utilice un doble buffer, así conseguiremos que el parpadeo en la imagen sea despreciable. En caso de que le pasemos el valor 0, si no se hacen las cosas con gran maestría se notaría un parpadeo bastante notable.



- **oslSetTransparentColor (OSL_COLOR color)** ➔ método que sirve para crear un efecto de transparencia, desactivando el color que le pasemos como parámetro a la función, en las páginas de más arriba de la documentación está una pequeña explicación (pág. 15). Este color no se dibujará en ningún momento en el juego.
- **oslDebug (Char *texto)** ➔ método que muestra un mensaje en modo depuración, el formato es idéntico al de `oslPrintf ()`. El mensaje aparecerá en un recuadro donde habrán dos opciones, o bien continua la ejecución de la aplicación o bien sale de ella.
- **oslQuit ()** ➔ método que fuerza la terminación de la ejecución del programa que se esté en curso. Es una buena manera de tratar los bucles infinitos.
- **oslStartDrawing()** ➔ permite dibujar en la pantalla con la librería `osLib`, este método junto con `oslSyncFrame()` y `oslEndDrawing()` será obligatorio su llamada para poder mostrar una imagen.
- **oslSyncFrame()** ➔ se le llama para terminar con un frame, un frame es, lo que se conoce como un fotograma. La consola PSP muestra unos 60 por segundo. Con esto conseguimos que deje de refrescar la imagen.



- **oslEndDrawing ()** ➔ *finaliza el dibujado. Este método esperará a la GPU a que termine su renderización. Utilizarlo siempre junto con los dos métodos anteriores.*
- **oslEndGfx ()** ➔ *termina la parte grafica. Una vez ejecutado este método se libera la memoria RAM o VRAM, teniendo que cargar en cada vuelta de nuevo las imágenes que queramos mostrar.*
- **oslDrawImage (OSL_IMAGE imagen)** ➔ *momento en el que la consola mostrará la imagen por pantalla, el valor que habrá que pasarle como parámetro será aquel que se halla cargado en la memoria previamente. Esta función deberá llamarse en cada vuelta que se haga en el bucle principal.*
- **oslCls()** ➔ *borra toda imagen que pueda haber en la pantalla, dejándola limpia para la carga de nuevas imágenes o mensajes.*



- **oslMoveTo(int x, int y)** ➔ *mueve el cursor, que permite la escritura en la pantalla, a las posiciones indicadas por los parámetros. Los parámetros se toman como la medida de un carácter, (8x8 pixeles).*
- **oslPrintf(char str[1000])** ➔ *permite escribir en la pantalla cualquier mensaje que queramos, hay que tener en cuenta que únicamente reconoce código ANSI, por lo que algunos caracteres no los reconocerá la maquina (¿, ñ, Ñ...).*
- **oslReadKeys()** ➔ *método que comprueba si se ha pulsado alguna tecla de la consola y la guarda en la estructura osl_pad. El resultado se guarda en un puntero. Es la función que tendremos que llamar para tratar cualquier evento cuando pulsamos una tecla de la consola. Los botones de la consola serán:*





Imagen 6: Botones de la PSP.

Dentro de la última función vista tenemos distintas formas de tratar las acciones que se realizan sobre las teclas. Mencionáremos las más utilizadas.

- *held* → trata la tecla que ha sido pulsada la última. Un pequeño ejemplo de cómo tratar cuando se pulsa la X sería

`osl_keys->held.cross`

- *pressed* → únicamente se crea el evento cuando es pulsada la tecla, no se guarda en memoria como ocurriría con *held*. El mismo ejemplo que el anterior sería.

`osl_keys->pressed.cross`



- *released* ➔ *este modo consiste, en que será cierto cuando el jugador suelte la tecla, por ejemplo para saber si suelta la X sería.*

`osl_keys->released.cross`

Estos serán los tres eventos más utilizados en los trabajos más sencillos que se puedan realizar en nuestra consola. Existen diversas posibilidades, que podrán ser estudiadas con más detalle en la documentación de esta interesante librería.



Tablero.h

TABLERO.H
ENUM <ul style="list-style-type: none">- color_jugador- color_casillas- tipocasilla
STRUCT <ul style="list-style-type: none">- Jugador- precio- casillas_calles

Imagen 6: Propiedades usadas en main.c



Como hemos explicado anteriormente, esta cabecera, sirve para definir aquellos tipos que queremos utilizar en nuestro juego, en la imagen número 4 podemos ver aquellos que hemos definido y que aquí explicaremos un poco más en detalle como están compuestos estos tipos definidos.

struct

- **Jugador** ➔ aquí vendrán aquellas propiedades típicas en el monopoly que tiene un jugador, como el dinero, posición es la que se encuentra, número de casillas que tiene de cada color, si ha sacado dobles...
 - **int** colores[11]
 - **long int** capital
 - **int** posi
 - **int** cárcel
 - **int** encarcelado (actuará como un booleano)
 - **int** dobles (actuará como un booleano)
 - **int** tarjetasalida
 - **int** desinteres (actuará como un booleano)
 - **int** casas
 - **int** hoteles
- **casillas_calles** ➔ propiedades de cada casilla del tablero que dependerá de su configuración, dependiendo de un tipo enum, que veremos más adelante.
 - **int** numeroCasas
 - **int** num_edificios
 - **struct** precios precio
 - **int** propietario
 - **enum** tipocasilla tipocasillatab
 - **enum** color_casillas colorcasillatab
 - **char** *nombre



- **precios** ➔ *dinero que costará cada cosa de cada propiedad : casas, hotel, hipoteca, precio de compra de la propiedad*
 - **long int compra**
 - **long int casa0** (Lo que hay que pagar al jugador)
 - **long int casa1** (Lo que hay que pagar al jugador)
 - **long int casa2** (Lo que hay que pagar al jugador)
 - **long int casa3** (Lo que hay que pagar al jugador)
 - **long int casa4** (Lo que hay que pagar al jugador)
 - **long int hotel**
 - **long int casa** (Lo que le cuesta al jugador edificar)
 - **long int hipoteca**
 - **long int venta**

enum

- **color_casillas** ➔ *clasifica los tipos de casillas distintas que existen dentro del tablero, como pueden ser los colores, si son estaciones, si son compañías, etc....*
- **tipocasilla** ➔ *nos muestra si la casilla es de tipo calle (calle, estación o compañía) o es de tipo especial (suerte, impuestos, cárcel...), como hemos comentado anteriormente este enum marcará la configuración entre las distintas casillas.*
- **color_jugador** ➔ *marcará el color que tiene cada jugador aunque este enum no es muy utilizado en nuestro proyecto.*



HERRAMIENTAS

Software de desarrollo

Microsoft Visual Studio 2008

Se trata de un entorno de desarrollo integrado para los sistemas operativos de Windows, soporta multitud de lenguajes de programación tales como, C, Visual C++, Visual C#, Visual J#, ASP.NET, Visual Basic .NET, aunque actualmente se están desarrollando nuevas actualizaciones para soportar más lenguajes.

En nuestro caso hemos instalado la librería en dicho entorno de desarrollo, gracias a él podemos compilar y tratar los errores que vayan surgiendo a medida que estemos trabajando con él, en los anexos de este proyecto podrá ver cómo preparar este entorno de desarrollo para poder conseguir trabajar con él y Oslib

Este entorno fue creado en 1998, por la mundialmente famosa empresa de Microsoft creada por Bill Gates.

Notepad ++

Este es un editor de texto y de código fuente libre, con soporte para muchos lenguajes de programación, únicamente es soportado por los sistemas operativos Windows.

Aunque no te detecte los errores al instante cuando está programando, es un programa bastante útil.



Software de diseño gráfico

MS Paint

Microsoft Paint (cuyo nombre original es Paintbrush) fue desarrollado en el año 1982 por Microsoft. Es un programa de diseño gráfico muy simple que cuenta con las siguientes herramientas: selección de forma libre, selección, borrador/borrador de color, rellenar con color, tomar color, lupa, lápiz, pincel, aerógrafo, texto, línea, curva, rectángulo, polígono, elipse, y rectángulo redondeado. MS Paint no tiene la capacidad de crear automáticamente gradientes de color.



Hardware de desarrollo

PC

Componente	Detalle
CPU	Intel Core 2 Duo T6600, 2400 MHz
RAM	2 GB
Disco Duro	500 GB
Tarjeta Gráfica	Nvidia GeForce 8800 GTS

Tabla 1: Resumen del Hardware de desarrollo

PSP

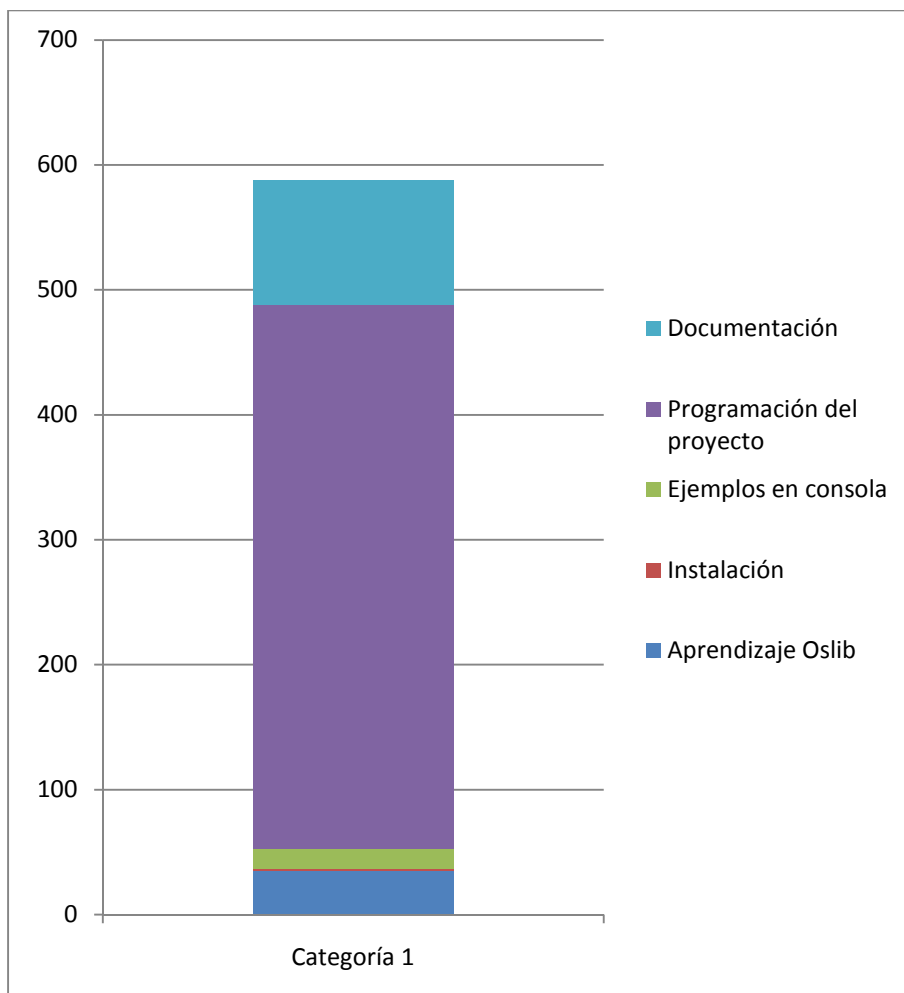
Componente	Detalle
CPU	Doble procesador MIPS R4000 333 MHz
RAM/VRAM	32/4 MB
Unidad de Almacenamiento	Memory Stick Pro Duo 4 GB
Tarjeta Gráfica	2 Mb 116 MHz

Tabla 2: Resumen del Hardware de desarrollo



TIEMPO EMPLEADO EN EL PROYECTO

Todas estas horas son tomadas como aproximaciones a los datos que hemos ido recolectando en el paso de los días, estando seguros que aumentamos en realidad esas cifras, debido a las múltiples complicaciones que nos surgían a medida que íbamos avanzando en el proyecto.



CONCLUSION Y DESARROLLO

Durante el desarrollo de nuestro trabajo, hemos intentado conseguir familiarizarnos tanto con la programación de una consola portátil como con la librería que nos permite crear cualquier aplicación en esta fantástica consola, creando un juego empezando desde cero.

Este proyecto ha sido una aplicación casera para una consola que se comercializa en todo el mundo, se ha abierto un camino para futuros proyectos en la Universidad de Sevilla, que quieran desarrollar y mejorar los aspectos y propiedades de un juego casero, dejándoles los ejemplos y los pasos a seguir que deben tomar para llegar hasta el objetivo que se marquen.

La librería libre Oslib permite realizar infinidad de cosas con nuestra consola la cual no hemos exprimido tanto lo que esta librería nos permite. Nuestra labor principal fue la de pelear con el compilador, crear los archivos necesarios a instalar en la PSP, preparar el entorno de desarrollo con la librería Oslib y buscar todo problema que le surge a cualquier programador a partir de su primera aplicación para cualquier plataforma (Hello World!), dejando el camino lo mas llano posible para que avancen a partir de aquí en un proyecto de mayor magnitud.



ANEXOS



ANEXO 1: LIBERAR UNA PSP

Modelos

Para poder instalarle aplicaciones caseras a una consola PSP, es necesario liberar la consola para que se puedan ejecutar sin que te muestre un mensaje de error cuando queramos arrancar dicha aplicación.

Existen múltiples versiones y modelos de esta magnífica consola, pero para no extendernos mucho en el tema explicaré, como preparar las versiones más nuevas y versiones un poco más antiguas como es el caso de mi consola.

Como hemos explicado antes existen 4 modelos de consolas, todas ellas con las mismas características en cuanto a hardware se refieren, pasaremos a mostrar cada una de ellas.

- **PSP 1000 (FAT)** → lanzada en el mercado japonés en Diciembre de 2004. Primera de las consolas en salir a la venta. En una de este tipo fue donde creamos este proyecto. Su diferencia con las dos siguientes, no es más que el grosor de la consola, el peso y que utiliza una batería más pesada que la 2000 o 3000 que veremos a continuación.



Imagen 7: Formato de PSP 1000



- **PSP 2000 (SLIM)** → sucesora de PSP 1000, su lanzamiento en Septiembre de 2007, esta consola trajo consigo unas mejoras bastante notables, como pueden ser el peso de la misma (casi un 40%), el tamaño ,menor tamaño de la batería , o una gran mejora como es una salida de video para poder conectar la consola a la televisión.



Imagen 8: Formato PSP 2000 o 3000

- **PSP 3000 (LITE)** → tercera entrega de la consola con pocas mejoras en cuanto a la anterior, lo único reseñable podríamos decir que, dispone de un micrófono, que las otra no las dispone, mejor resolución y la duración de la batería que se ve aumentada. En cuanto al diseño es exactamente el mismo al anterior, excepto en que la tecla home esta cambiada por el logo PS.



- **PSP GO** → primera consola que cambia en el diseño en comparación con las tres vista con anterioridad. Salió al mercado en 2009. Y consigue reducir el peso de la consola en un 40%, convirtiéndola en una consola muy ligera. Única de la consolas que contiene una memoria interna, debido a que elimina los juegos cargados a través de los discos especiales UMD, también dispone de Bluetooth 2.0 que permite conectar auriculares inalámbricos.



Imagen 9: Formato PSP Go



Versiones importantes:

Como en todo equipo electrónico a medida que transcurre el tiempo se van actualizando dichos productos a versiones mejores y más recientes, con el fin de mejorar y arreglar posibles fallos que pudiera tener el producto (incluyendo para la lucha contra la piratería). Pasaremos a explicar las tres versiones más importantes que nos podemos encontrar en las consolas. La primera que se pudo modificar, la versión de la consola que disponemos y la última versión del mercado en el momento de hacer este trabajo.

Antes de ponernos a ver con más detalles las versiones, decir que, algunas consolas del mercado tienen una placa base (conocida como placa maldita) la cual no permite la adaptación de dicha consola, para qué ésta acepte aplicaciones hechas por usuarios.

- *1.50 ➔ padre de toda versión de la PSP. En esta versión se puede realizar cualquier cosa sin limitación alguna, para la adaptación de cualquier versión de consola lo primero que se realiza es llegar primero a esta versión y luego realizarle los cambios necesarios al firmware de la consola para conseguir el objeto deseado.*
- *5.00 M33-4 ➔ versión downgrade (preparada para meter aplicaciones caseras y demás utilidades) que dispone nuestra consola, en la cual hicimos el trabajo. Esta versión no tiene nada en especial, con respecto a las 1.50, excepto que trae una radio por internet, un chat de correo instantáneo y algunas aplicaciones que para nosotros no nos importa para nada.*



- 6.35 ➔ *ultima versión que en la actualidad viene instalada en la consola que está en venta. La única diferencia con respecto a las versiones explicadas anteriormente es que, una vez hecho el downgrade (en caso de apagar completamente la consola) tendría que realizar el downgrade. La única manera que existe de que se mantuviera, sería dejando la consola suspendida y no apagada (toque leve al botón de power).*

Pasos a realizar para el Downgrade

Antes de ponernos con el tutorial de cómo hacerle a la consola un downgrade informaros que es necesario disponer de unos accesorios indispensables para el proceso:

- **Memory Stick mágica** ➔ *esto no es más que, una tarjeta Memory Stick normal y corriente, que podemos comprar en el mercado, a la que se le realiza unas modificaciones que nos permita hacer un downgrade a la consola.*
- **Batería PSP pandorizada** ➔ *batería de PSP, modificada, con la cual lo único que se consigue es que al introducirla en la consola que queramos hacer el downgrade, se encienda automáticamente, sin necesidad de pulsar el botón Power.*

Estos dos accesorios pueden ser creados por nosotros, desde una consola ya preparada (realizado el Downgrade). Para ello utilizaremos los conocidos programas. UltraPandora Installer y Despertar de Cementerio V8.



Preparando la MS Mágica

Antes de nada y ponernos a preparar la tarjeta, necesitamos descargar el programa que convierte nuestra Memory Stick, esto lo podemos realizar desde la página web (es necesario registrarse):

- <http://psp.scenebeta.com/noticia/el-despertar-del-cementerio-universal-downgrade-unbrick-de-cualquier-firmware-psp-slim-fat>

Hay que saber que todo este proceso debe realizarse con una consola que ya disponga de su downgrade, en una con firmware oficial no sería posible. Conectamos la PSP al USB del PC, con la Memory Stick de al menos 256Mb dentro. Si se tienen datos importantes en la Stick, salvadlos al PC lo primero, pues la vamos a formatear. Formateamos la Stick desde el menú Ajustes/Formatear Memory Stick.

Descomprimimos y copiamos a la raíz Memory Stick el Despertar del Cementerio v8 que hemos descargado antes .Copiamos TODO el contenido del RAR a la raíz de la Stick. La raíz de la memoria no es más que la primera ventana que sale al abrir la tarjeta desde nuestro sistema operativo.

Ejecutamos en la PSP desde el menú Juegos la aplicación Despertar del cementerio v8. Cuando éste arranque únicamente tendremos que pulsar X y dejar que realice el proceso el programa, cuando éste termine, aparecerá el siguiente mensaje:

Please keep pressed for some seconds the key/s which you want to use to boot DC8...

Mantenemos pulsado el botón que queremos usar para arrancar en modo Despertar del Cementerio. Recomendamos que se mantenga pulsada la tecla L unos segundos. Así, si encendemos la PSP con L pulsado, entrareis en el menú Despertar del Cementerio. Si no pulsáis nada, la PSP arrancara normalmente aunque tenga insertada una Memory Stick Mágica y una Batería Pandora.



Preparando la MS Mágica

Para conseguir este programa podemos realizar la descarga desde el enlace:

- <http://psp.scenebeta.com/noticia/ultrapandora-installer>

En esta página es necesario registrarse para realizar cualquier tipo de descarga, aunque para todo aquel que esté interesado en meterse en este mundillo de la programación de PSP, esta página es de las mejores que puede encontrar en la red y en español.

Una vez descargado el programa, conectamos la consola a través del USB, e instalamos en el directorio "PSP/GAME" la carpeta obtenida en la descarga. Cuando terminamos con este proceso ejecutamos en la PSP desde el menú Juegos la aplicación *Ultra Pandora Installer*.

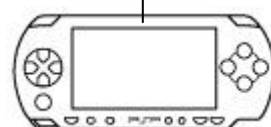


Imagen 10: Icono Ultrapandora mostrado en PSP

Cuando el programa se ejecute, vamos al siguiente icono y pulsamos la tecla X. Con estos simples pasos conseguimos tener ya la batería pandorizada.



Imagen 11: Menú dentro de UltraPandora



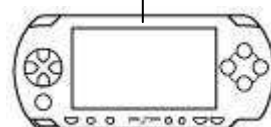
Una vez terminado el proceso de pandorización vamos a la derecha del todo del programa para salir de este.

Así pues ya tenemos los dos accesorios preparados para practicar nuestro primer downgrade.

Pasos (1000 o 2000)

Antes de comenzar es necesario tener la batería al 100%.

1. *Metemos la Memory Stick "mágica" en la PSP que va a ser hackeada (downgrade).*
2. *Con el cable de corriente desconectado, deja el botón L pulsado, mete la "batería pandorizada", la PSP debería encender sola.*
3. *Veremos que aparece un menú en la pantalla de la consola con varias opciones:*
 - a. **Install 5.00 M33** → *Pulsando X se nos instalará el Custom Firmware M33 en cualquier PSP (Slim o Fat).*
 - b. **Install 5.00 OFW** → *Pulsando X sobre esta opción se nos instalará el Firmware original 5.00.*



- c. **NAND Operations** → Esto nos dumpeará nuestra memoria Flash a la Memory Stick en forma de copia EXACTA. También sirve para restaurar la copia de la NAND, o para formatear la flash. IDStorage Tools son opciones avanzadas que solo deben ser utilizadas por expertos.
 - d. **Hardware Info** → Muestra información técnica de nuestra PSP.
 - e. **Test M33** → Nos permite probar el Custom Firmware M33 antes de instalarlo.
 - f. **Shutdown** → Apaga la PSP.
 - g. **Reboot Device** → Reinicia la PSP.
-
- 4. Simplemente, seleccionamos la opción que queramos (típicamente la primera) y pulsamos X. Podremos ver que los leds de Wifi y Memory Stick empiezan a parpadear.
 - 5. Para saber si se ha terminado, bastará con fijarse en la pantalla y esperar a que salga el mensaje de apagar (shut down) la PSP, presiona X para apagarla, y una vez apagada metemos la batería y la Stick "normales" y la reiniciamos.
 - 6. Ya tenemos finalizado el proceso, tenemos una consola que permite introducir aplicaciones caseras. Tendremos que tener un juego UMD en la consola para poder cargar los juegos.



Pasos (3000)

Everybody's Sukkiri (programa downgrade)

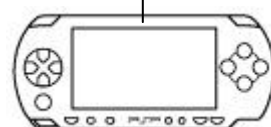
Esta guía sería para aquellas consolas que tienen la versión 6.35. El modo de trabajo es distinto aunque no muy complicado.

Lo primero que tenemos que hacer es verificar que la consola tiene la versión que os mencionamos. Para ello nos dirigimos a Ajustes – Ajustes del sistema – información del sistema, aquí podemos ver que se trata de esta versión.



Una vez hemos confirmado la versión, tendremos que descargar la aplicación Everybody's Sukkiri, e instalar todos los archivos en la raíz de la tarjeta de memoria. Cuando tengamos todo instalado nos vamos dentro de la consola ha, Juegos – Memory Stick – Everybody's Sukkiri

Una vez ejecutada la aplicación, tendremos que darle a la opción que aparece en la pantalla marcada y dejar que ésta haga su trabajo, cuando termine será reiniciada la consola automáticamente.

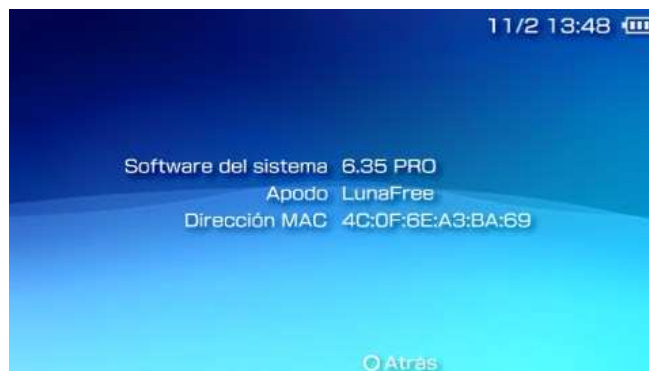


Juego Psp: Monopoly

42



Cuando la consola se reinicie sola, debemos ir a información del sistema nuevamente para confirmar que la consola ha sido modificada. Tendría que aparecerle ahora esta imagen. Con ello sabríamos que podemos cargar nuestras aplicaciones.



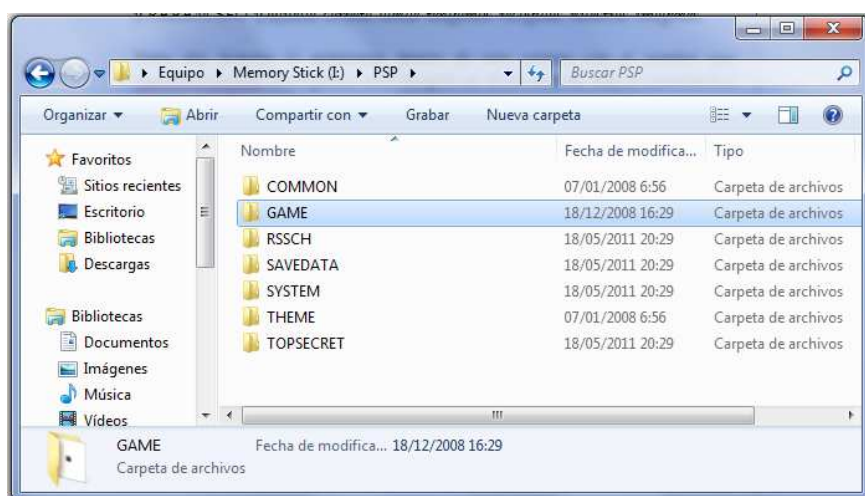
ANEXO 2: INSTALACIÓN Y EJECUCIÓN DEL JUEGO

Instalación

Para poder probar nuestro trabajo es necesario guardar nuestra aplicación, dentro de la Memory Stick de la consola. No es necesario tener en cuenta todos los ficheros que genera nuestro compilador. Únicamente nos interesa los ficheros EBOOT.PBP y PARAM.SFO y aquella carpeta donde tengamos recogidas nuestras imágenes.

Estos dos ficheros lo meteremos dentro de una carpeta con el nombre que queramos. Cuando lo tengamos todo recogido en dicha carpeta, ésta la pasaremos a nuestra consola.

Conectamos la consola al USB del PC e instalamos la carpeta que creamos con anterioridad en la ruta. PSP://PSP/GAME



Con esto tendríamos la instalación del juego concluida. Ahora únicamente nos quedaría ejecutar el juego, que es lo más simple que existe, de todo el trabajo.



Ejecución

Con todo ya listo para arrancar el juego, únicamente tenemos que navegar a través de la consola hasta.

Juegos – Memory Stick –PFC MONOPOLY



ANEXO 3: INSTALACIÓN LIBRERÍA VISUAL STUDIO 2008 Y COMPILADOR

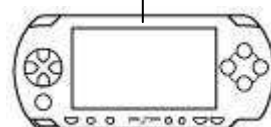
En esta sección os enseñaremos a cargar la librería Oslib dentro del entorno de desarrollo Visual Studio 2008, para que pueda detectar los errores en tiempo de programación y os compile y cree aquellos archivos necesarios para guardar en la consola portátil. Dejar constancia que éste no fue el método que utilizamos nosotros a la hora de crear el proyecto. Si no que utilizamos la segunda forma que se explica en este anexo más adelante, debido a la facilidad a la hora de echarlo a andar.

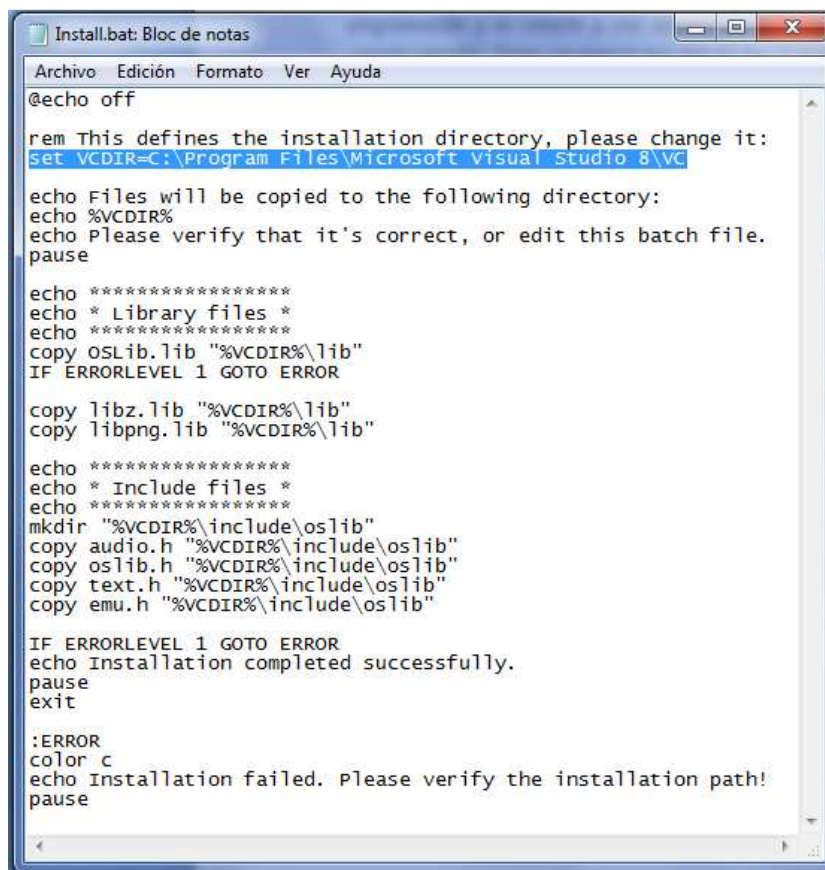
Visual Studio 2008

Lo primero que debemos hacer antes de nada es descargar la librería que queremos añadirle a este entorno de desarrollo. La versión más reciente que se encuentra en la página web de los creadores de la librería es la 6.90

- <http://www.megaupload.com/?d=ECKTIO2W>

Una vez descargada, descomprimos en cualquier lugar de nuestro ordenador y nos vamos en busca de la carpeta PC. En esta carpeta encontraremos un archivo llamado Install.bat, con el botón derecho de nuestro ratón pinchamos sobre dicho archivo y le damos a editar. Nos aparecerá un editor de notas con lo siguiente.





```
Install.bat: Bloc de notas
Archivo Edición Formato Ver Ayuda
@echo off
rem This defines the installation directory, please change it:
set VCDIR=C:\Program Files\Microsoft Visual Studio 8\VC
echo Files will be copied to the following directory:
echo %VCDIR%
echo Please verify that it's correct, or edit this batch file.
pause
echo *****
echo * Library files *
echo *****
copy oslib.lib "%VCDIR%\lib"
IF ERRORLEVEL 1 GOTO ERROR
copy libz.lib "%VCDIR%\lib"
copy libpng.lib "%VCDIR%\lib"
echo *****
echo * Include files *
echo *****
mkdir "%VCDIR%\include\oslib"
copy audio.h "%VCDIR%\include\oslib"
copy oslib.h "%VCDIR%\include\oslib"
copy text.h "%VCDIR%\include\oslib"
copy emu.h "%VCDIR%\include\oslib"
IF ERRORLEVEL 1 GOTO ERROR
echo Installation completed successfully.
pause
exit
:ERROR
color c
echo Installation failed. Please verify the installation path!
pause
```

Lo marcado en color azul tendremos que modificarlo y poner lo siguiente.

set VCDIR=C:\Archivos de programa\Microsoft Visual Studio 8\VC

Resulta ser lo mismo que aparece en la imagen pero en español. Una vez hecha la modificación guardamos los cambios y ejecutamos dicho archivo (install.bat). El último paso para terminar de cargar la librería en VS 2008 sería, irnos a la carpeta **oslib1/Install** y ejecutar en esa carpeta **install.bat**. Con estas pautas hemos conseguido cargar en visual studio 2008 esta magnífica librería. A partir de aquí podemos crear nuestros primeros ejemplos.



ANEXO 4: MANUAL DE USUARIO, COMO JUGAR.

En este apartado se introducirán los distintos aspectos que necesita conocer el usuario para poder disfrutar de una partida de Monopoly, desde como tirar los dados hasta como comprar y negociar por una propiedad.

Como hemos explicado en anexos anteriores necesitamos tener una consola preparada y la aplicación instalada en ella para poder comenzar una partida. Una vez comience el juego y pasen las imágenes de presentación pasaremos a la elección de jugadores. Si lo deseamos podemos jugar únicamente personas físicas o también jugar junto a la máquina en una partida mixta. Para ello aparecen dos imágenes en pantalla, una primera para elegir los jugadores reales (humanos) y a continuación se pasará a elegir los jugadores por computador.

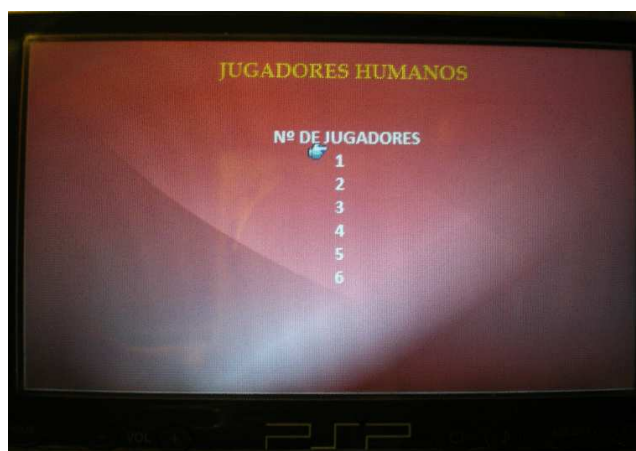


Imagen 17: Selección de jugadores en una partida

*Una vez se han elegido los jugador (con un máximo de hasta 6 jugadores), se tirará el dado para saber que jugador será el primero en comenzar la partida (la mayor puntuación gana). Terminada esta acción comienza lo interesante del juego, la partida. El jugador al cual sea su turno deberá pulsar la tecla **Δ** (triángulo) para tirar los dados.*



Juego Psp: Monopoly

48

Cuando caiga en la casilla que le marque los dados, a continuación deberá pulsar ☐ (cuadrado) para saber que opciones puede realizar sobre dicha casilla (en caso de que este libre saldrá el procedimiento de compra y si algún jugador fuera su propietario saldría los pasos que habría que realizar para acometer el pago de la propiedad).

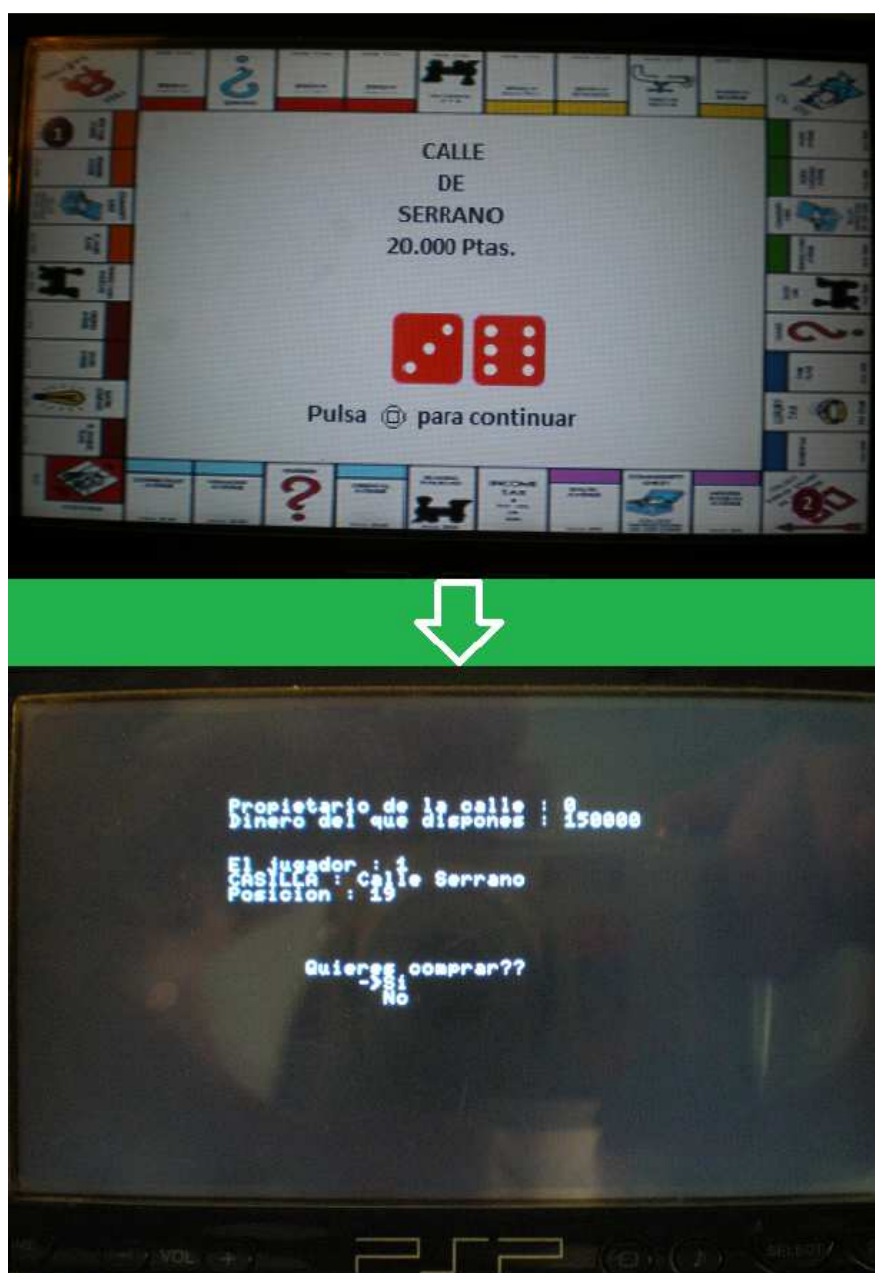


Imagen 18: Proceso para comprar una propiedad



Si el jugador que tiene la opción de realizar la compra de la propiedad, finalmente no lo hace, esta propiedad pasará inmediatamente a una puja, en la que todo jugador tendría la oportunidad de pujar por la propiedad. Si nadie le interesa esta propiedad (nadie puja por ella) seguirá estando libre hasta que nuevamente un jugador caiga en ella. La puja comenzará siempre desde 1, y el siguiente jugador hasta el último de la puja tendrá que ofrecer uno más que el anterior.

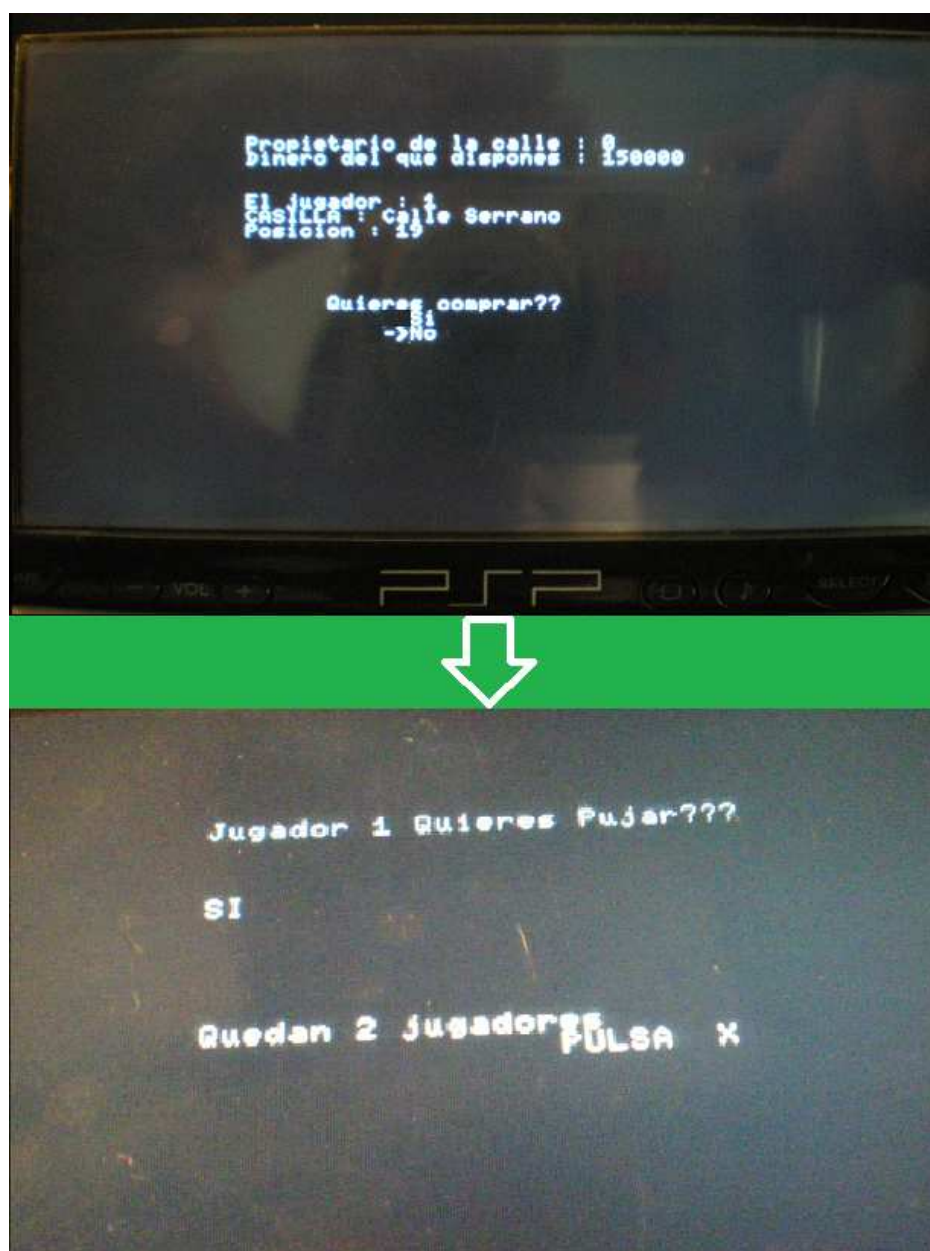
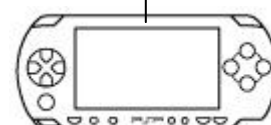


Imagen 19: Introducción de una subasta.

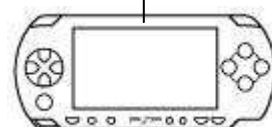


Ésta suele ser la actividad principal de nuestro juego, tirar los dados y realizar una acción sobre la casilla en la que se cae siguiendo todas las instrucciones que te vaya mostrando la pantalla. La acción cuando se cae en la casilla Suerte o Comunidad es idéntica a lo explicado con anterioridad, realizando la acción que las cartas mande la propia máquina, así el jugador no se tiene que preocupar de nada. A no ser que tenga el jugador que tomar alguna decisión.

Aunque parezca que el juego solo consiste en eso, existen muchas opciones que los jugadores pueden realizar durante su turno. Si pulsamos la tecla **select** se desplegará automáticamente el menú principal del juego donde podremos seleccionar cualquier opción que se nos antoje en ese momento, como la posibilidad de hipotecar una propiedad, intentar negociar con otro jugador, ver todas las características de cada jugador (dinero que tiene, propiedades, etc...), así como incluso dejar dicho jugador la partida o salir del juego. Todo ello seleccionable desde el menú que hemos mencionado con anterioridad. En caso de no querer hacer uso de ninguna de estas propiedades, si volvemos a pulsar la tecla **select** volveremos a la partida que hemos detenido. Una anotación que queremos hacer, es que la acción vender propiedad, viene a referirse a que el jugador está hipotecando su casa, en ningún momento pierde dicha propiedad.



Imagen 19: Menú principal del juego



Veremos ahora con imágenes y algunas explicaciones no muy extensas, cada una de las posibles opciones que existen en el menú principal.

Venta de Propiedad

*Cuando seleccionamos esta opción, la máquina muestra al jugador todas las casillas que le pertenece, pudiendo elegir entre cualquiera de ellas, pulsando la tecla **X** el jugador podrá hipotecar la casilla seleccionada, con esta acción el jugador no podrá cobrar a otros jugadores por caer en esta casilla, hasta que no quite la hipoteca a dicha propiedad, podrá ver usted mejor las reglas en el siguiente anexo.*

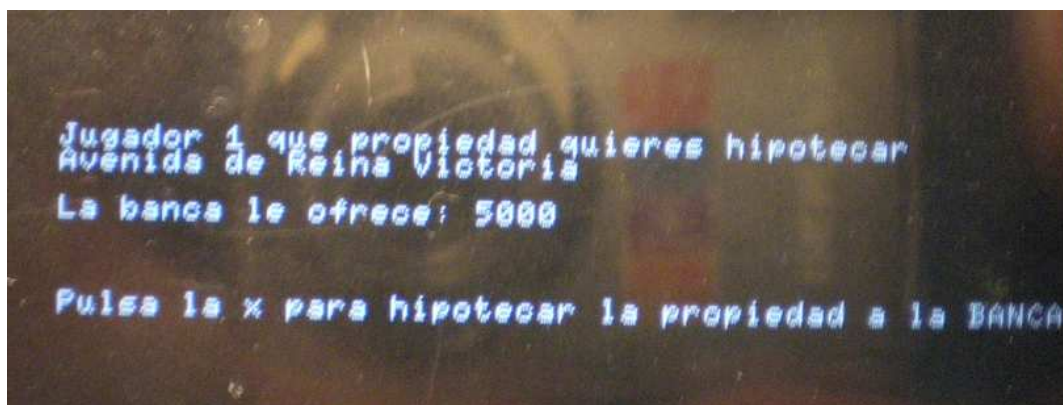


Imagen 20: Pantalla del juego cuando se quiere hipotecar.

Negociar

Segunda de las opciones que permite realizar el menú principal. En este caso el jugador vera en pantalla toda propiedad que haya sido comprada por cualquier jugador de la partida exceptuando a él. En casa una de las casillas que muestre la pantalla vendrá reflejado también a quien pertenece esta casilla.



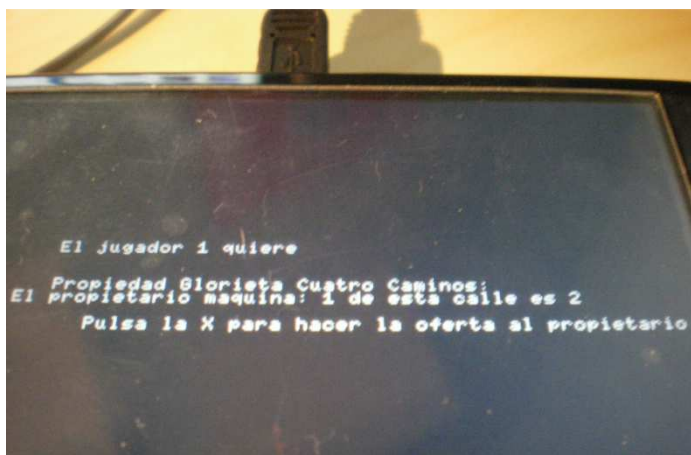


Imagen 21: Pantalla de negociación.

Si nos fijamos en un detalle de la imagen podemos ver una frase seguida de un número que dice, El propietario máquina: 1, esto no viene a referirse a que el jugador es la ficha 1, si no que esta dando a entender que esta negociando con la máquina (por aparecer el valor 1), en caso de que este número fuese el 0; el jugador sería una persona física. En caso de no querer hacer finalmente la negociación, pulsamos la tecla X para entrar dentro de la pantalla de ofrecer dinero y a continuación pulsamos la tecla O.

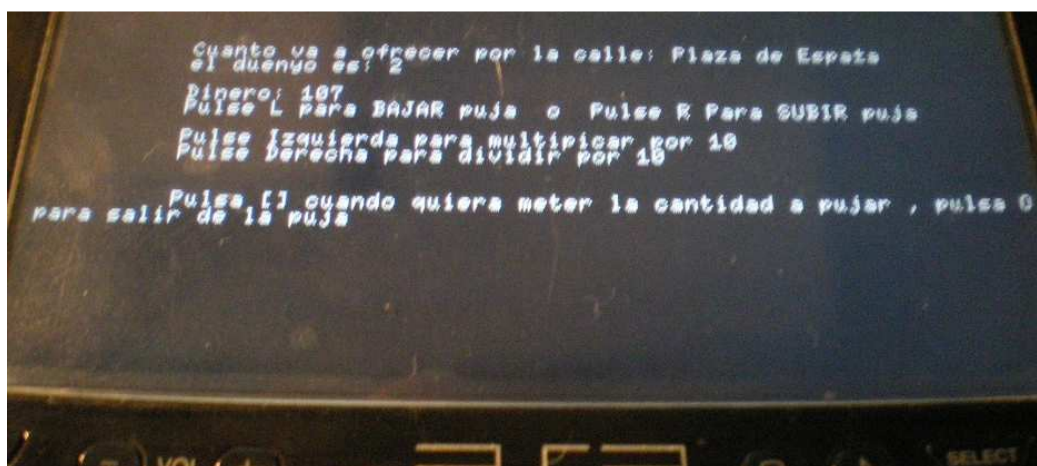


Imagen 22: Elección de dinero a ofrecer en negociación.

La teclas L o R en esta acción sirve para añadirle un 0 a la cantidad que vaya a ofrecerle al jugador o en caso contrario quitarle un 0.



Ver opciones jugador

Aquí podremos ver las características de cada jugador sin importar el turno de quien sea. Mostrará desde el primero al último, pudiendo ver en cada caso el dinero que tiene, el número de propiedades de cada color que tiene, si se trata de la máquina o es persona humana, si dispone de una tarjeta que le permita salir de la cárcel e incluso en la posición del tablero en la que se encuentra en la actualidad, en definitiva toda aquella información que un jugador le gustaría saber sobre sus oponentes o incluso sobre el mismo.

Para salir de esta opción, cuando estemos dentro de ella, pulsaríamos en nuestra consola portátil la tecla de la cruceta hacia abajo.

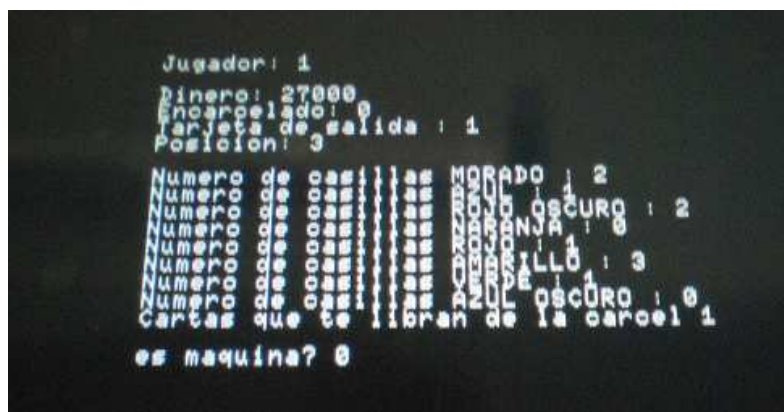
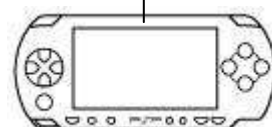


Imagen 23: Propiedades de un jugador.

Deja jugador partida o salir del juego

Estas dos posibles selecciones, como sus propios títulos indican, en el caso del primero, el jugador que sea su turno, abandonará la partida que se esté disputando, devolviéndole a la banca toda casa, hotel o propiedad que estuviera en su poder, su ficha desaparecería de la pantalla.

En el segundo caso, el jugador que lo seleccione saldrá de la aplicación que hemos creado con la correspondiente pérdida de la partida y datos que hasta el momento hubiera en ella.



Cárcel

Otros momento qué nos podemos encontrar en la partida es cuando uno de los jugadores, o bien, porque le toque una carta qué le mande a la cárcel o bien porque cae en la casilla qué te manda directamente a la cárcel, el jugador irá directamente a la cárcel perdiendo en ese momento el turno.

La únicas tres maneras de salir serian o bien, sacar dobles (no tendría derecho a volver a tirar los dados), utilizar una de las cartas que libra de la cárcel, o bien pagando 5000 Ptas.

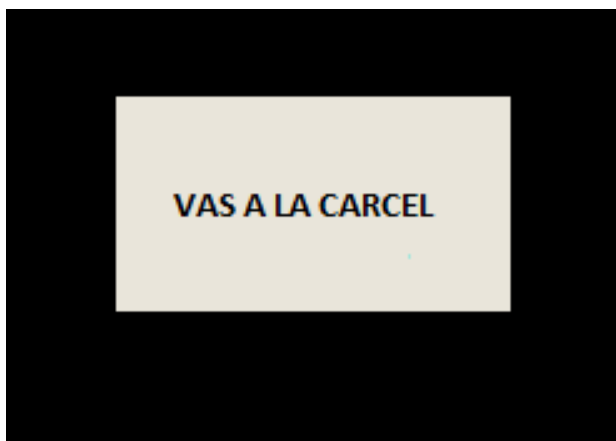


Imagen 24: Pantalla que se muestra cuando se cae en la cárcel.

Construir una casa u hotel.

Cuando uno de los jugadores tiene todas las calles de un mismo color en su propiedad, este jugador podrá construir una casa u hotel, pulsando cuando se le mande la consola. Pudiendo elegir entre las calles de ese mismo color en la cual queremos construir la casa.



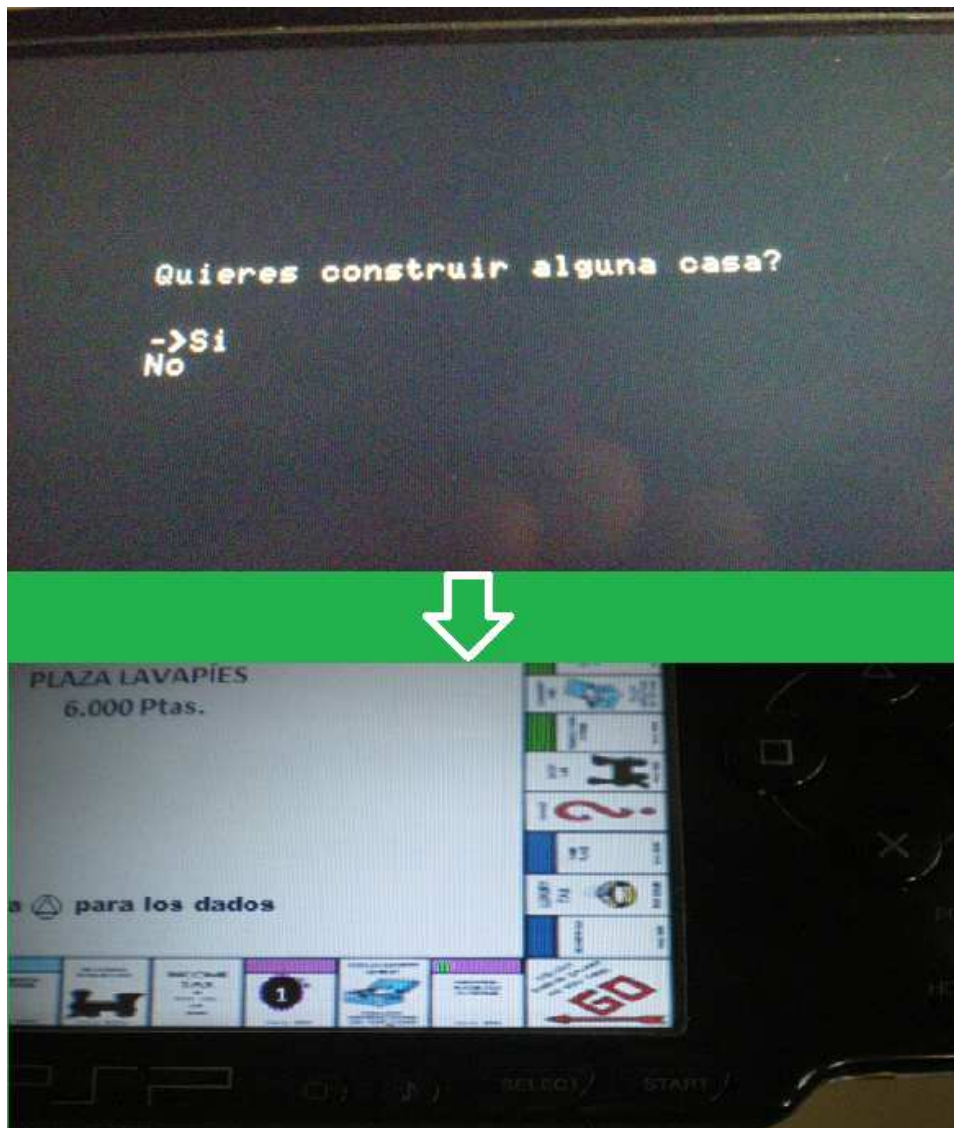


Imagen 25: Proceso de construir una propiedad

Tanto la imagen que pregunta si queremos construir una casa o un hotel resulta ser la misma, ya que tomamos en el juego que el hotel es como la 5ª casa que se construye en la propiedad, aunque solo se muestre el hotel.



Fin de la partida

Aquel jugador que gane la partida (único jugador que quede en el tablero). Será el ganador de la partida. Cuando esto suceda se saldrá el juego automáticamente de la aplicación viendo visto antes la siguiente imagen con la ficha de aquel jugador que haya conseguido tal logro.



Imagen 26: Fondo que sale en consola para el ganador.



ANEXO 5: REGLAS DEL MONOPOLY

Objetivos

El objetivo del juego es convertirse en el jugador más rico a través de la compra y venta de propiedades.

El equipo

El equipo se compone de un tablero, dos dados, fichas, 32 casas y 12 hoteles. Hay una baraja de cartas para los espacios de SUERTE y COMUNIDAD. Existe una tarjeta que asocia cada propiedad, y vales que sirven como dinero.

Preparación

Se pone el tablero sobre una mesa colocando las cartas de COMUNIDAD Y SUERTE con la cara hacia abajo, en los espacios destinados para las mismas en el tablero. A cada jugador se le entrega una ficha que representa a dicho jugador en sus viajes alrededor del tablero. A cada jugador se dan 150000 Ptas. El resto del dinero lo recibe la banca.

El banquero

En nuestro juego el banquero no será de mucha utilidad debido a que esta tarea la realiza la misma consola.



El banco

El banco, guarda además del dinero del banco, las cartas de cada propiedad, casa y hoteles antes de que las adquieran y utilicen los jugadores. El Banco paga los sueldos y gratificaciones, vende y embarga las propiedades a los jugadores y les entrega las cartas de cada propiedad; les vende casas y hoteles a los jugadores y les presta, cuando es necesario, dinero sobre las propiedades que cualquier jugador hipoteque. Al banco se le pagará todas las contribuciones, multas, préstamos e intereses y el precio de todas las propiedades que éste venda o embargue. El banco nunca se arruina.

El juego

Empezando por el banquero, cada jugador echa los dados por turno. El jugador que logra el total mayor inicia el juego. Después de colocar su ficha en el ángulo marcado por la palabra GO (Adelante), el jugador tirará los dados y adelantará su ficha en la dirección de la flecha el número de espacios que indiquen los dados. Cuando ha terminado la jugada le pasa el turno al jugador a su izquierda. Las fichas quedan en los espacios ocupados y siguen adelantando a partir de dicho punto cuando le toca el turno de nuevo a cada jugador. Pueden quedar a la vez dos o más fichas en el mismo espacio.

Si un jugador saca dobles, adelantará su ficha normalmente, es decir, tantos espacios como puntos haya sacado y quedará sujeto a los derechos o castigos que le correspondan al espacio al que haya llegado. Después de esto volverá a tirar los dados y adelantará su ficha del mismo modo. Si un jugador saca tres dobles seguidos en vez de adelantar la ficha, irá directamente al espacio marcado EN LA CARCEL.



Go (Adelante)

Cada vez que la ficha de un jugador va a parar o pasa sobre el espacio GO (Salida), sin tener en cuenta si ha llegado a dicho espacio o pasado por el mismo como resultado de tirar los dados o de robar una carta, el banquero le paga un sueldo de 20000 Ptas.

Sin embargo, las 20000 Ptas. se pagan una sola vez por cada vuelta al tablero en la que se llegue al espacio GO o se pase por el mismo. Si un jugador que pasa sobre GO la primer vez que echa los dados va a parar a COMUNIDAD dos espacio mas allá o a Suerte siete espacio más allá, si logra coger la carta "Avance hasta GO", el jugador recibirá 20000 Ptas. de la primera vez que paso y otro 20000 por ir al GO la segunda vez siguiendo las instrucciones de la carta.

Compra de una propiedad

Cuando la ficha de un jugador va a parar a una propiedad sin dueño, este jugador tiene la opción de comprar dicha propiedad al Banco por el precio que va impreso en la misma. El jugador recibe una carta de la propiedad que lo acredita como dueño y que debe colocar boca arriba delante de sí mismo.

Si el jugador opta por no comprar, la propiedad es puesta inmediatamente a la venta en subasta por el banquero y es vendida al mejor postor. El comprador pagará al Banco la cantidad estipulada en la licitación y recibirá la carta correspondiente de escritura de la propiedad. Todos los jugadores pueden pujar, incluso el jugador que no acepto la opción de compra al precio impreso en la propiedad. Cualquier precio puede servir de base para comenzar la subasta.



Llegada a una propiedad con dueño

Si la ficha de un jugador llega a una propiedad que tiene dueño, el dueño de dicha finca le cobra el alquiler estipulado en la lista impresa en la tarjeta de la propiedad correspondiente a la misma.

Si el solar está hipotecado no puede cobrarse alquiler. La hipoteca de la propiedad se indica volviendo boca abajo frente al dueño de la tarjeta, en nuestro caso se hará automáticamente.

Nota: *Si en el solar hay una o varias casas el alquiler es mayor que el que le corresponde a un solar sin construcción.*

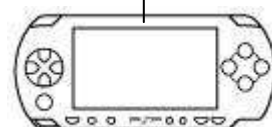
Es sumamente ventajoso poseer las propiedades de todas las fincas de un mismo color, porque esto autoriza al dueño a cobrar doble cuando un jugador cae en una de las casillas del mismo color. Esta regla se aplica a los solares no hipotecados, aún en el caso de que algún otro solar del mismo grupo esté hipotecado.

Es mucho más ventajoso ser dueño de casas y hoteles que de solares sin construir porque los alquileres de los primeros son mucho más altos que los de los solares sin construir.

Llegada a los espacios de Comunidad o Suerte

Cuando esto ocurre, el jugador roba la carta de encima de la baraja que se indique en la casilla en la que haya caído el jugador, y luego después de cumplir las instrucciones impresas en la misma, la vuelve a poner cara abajo debajo de dicha baraja.

La carta "Salir libre de la cárcel" se conserva hasta que se utilice y una vez utilizada se la vuelve a poner debajo de la baraja. Si el jugador que la posee no desea utilizarla, la puede vender en cualquier momento a otro jugador al precio que le convenga.



Impuesto

Cuando el jugador llega al espacio de contribuciones tiene dos opciones: estimar su impuesto en la cantidad de 20000 Ptas. y pagarlo al banco o pagar el 10% de su patrimonio total al Banco. El patrimonio total se calcula sumando el dinero en efectivo, el precio impuesto a las propiedad (hipotecadas o no) y el precio de costo de todos los edificios que posea.

El jugador debe decidir qué hacer antes de empezar a sumar el valor de sus bienes.

La cárcel

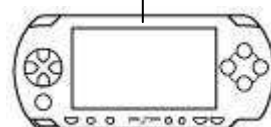
El jugador va a la cárcel cuando:

- *Su pieza va a parar al espacio marcado "Váyase a la cárcel"*
- *Roba una carta marcada "Váyase a la cárcel"*
- *Al tirar los dados saca dobles tres veces seguidas.*

Cuando se envía a la cárcel a un jugador no puede cobrar el sueldo de 20000 Ptas. en dicho movimiento porque sin importar el lugar en donde se halle situada su pieza en el recorrido del tablero, tiene que trasladarla directamente a la cárcel. Siempre que se le envía a la cárcel termina el turno del jugador.

Si en el curso de la una jugada normal la ficha de un jugador llega al espacio "Visita", el jugador no sufre ningún castigo y sigue delante de la manera normal cuando le toca de nuevo el turno.

El jugador puede salir de la cárcel de alguna de las siguientes maneras:



- *Sacando dobles en alguno de los tres turnos siguientes al de entrada a la cárcel. Si consigue hacerlo, avanza inmediatamente la ficha por el número de espacios indicados al sacar los dobles. En este caso, a pesar de haber sacado dobles no usara el otro turno.*
- *Usando la carta "Salir libre de la cárcel gratis" si esta se posee.*
- *Comprando la carta anteriormente mencionada a otro jugador y usándola.*
- *Pagando una multa de 5000 Ptas. antes de echar los dados en cualquiera de sus dos turnos siguientes.*

Si al llegar el tercer turno el jugador no ha sacado dobles, deberá pagar la multa de 5000 Ptas. Entonces sale de la cárcel y deberá mover su ficha la cantidad de espacios que indique su tirada en los dados.

Aunque esté en la cárcel el jugador puede comprar y vender propiedades, casas, hoteles y cobrar alquileres.

Parada Libre

El jugador cuya ficha va a parar a este espacio no recibe ningún dinero, propiedad ni premios de ninguna clase. Se trata simplemente de un lugar de descanso gratis.



Las casas

Cuando un jugador posee todas las propiedades de un grupo del mismo color puede comprar casas al Banco y levantarlas en dicha propiedad.

Si compra una sola casa, la situará en cualquiera de las propiedades del grupo. La segunda casa que compre y levante tendrá que situarse en uno de los solares no ocupados de este grupo completo de un solo color o de algún otro de dichos grupos de su propiedad.

En la escritura de propiedad del solar se indica el precio que debe pagarle al banco por cada casa.

Aún sobre los solares sin construir pertenecientes a un grupo completo de un solo color de su propiedad puede cobrar doble alquiler a algún adversario cuya ficha llegue a los mismo.

El jugador podrá comprar y construir, de conformidad con las reglas que anteceden y en cualquier momento, todas las casas que juzgue conveniente y le permita su situación económica. Pero debe construir en partes igual (es decir que no podrá construir más de una casa en un solar de un grupo de un determinado color hasta haber construido una casa en cada solar de dicho grupo. Cuando lo haya hecho podrá entonces empezar a construir la segunda fila de casas y así sucesivamente hasta llegar al límite de cuatro casas por cada solar. Por ejemplo, no podrá construir tres casas en un solo solar mientras exista tan sólo una casa en otro solar del mismo grupo).

De la misma forma que hay que construir en partes iguales, también hay que deshacerse en partes iguales de las casas al volverlas a vender al Banco (ver Venta de propiedades).



Los hoteles

Antes de poder comprar un edificio de hotel, el jugador ha de tener cuatro casas en cada solar de un grupo completo de un solo color. Cuando lo logre podrá comprar del Banco un hotel para levantarlo en cualquier solar de dicho grupo de un solo color, entregándole al Banco a cambio del mismo las cuatro casas allí existentes y el precio del hotel que indique la Escritura de propiedad. En cada solar no puede construirse más de un hotel.

La escasez de edificios

Cuando el Banco no tenga casas para vender, los jugadores que deseen construir tendrán que esperar para hacerlo a que algún otro jugador devuelva o venda al Banco sus casas. Si se dispone de una pequeña cantidad de casas y hoteles y dos o más jugadores desean comprar un número superior al que tiene el Banco, las casas y hoteles tienen que venderse en una subasta al mejor postor.

Venta de propiedades

Todo jugador que sea dueño de solares sin edificar, ferrocarriles y servicios públicos (pero no de edificios) podrá venderlos a cualquier otro jugador en una operación privada y por la cantidad que convenga a ambos. No obstante, no podrá venderse ningún solar que pertenezca a un grupo de color en el que haya algún otro solar que contenga edificios. Antes de que el dueño pueda vender un solar de dicho grupo de color, tendrá que vender al Banco dichos edificios.



Los jugadores pueden volver a vender las casas y hoteles al banco en cualquier momento y a la mitad de precio que pagaron por las mismas.

Todas las casas de un grupo de color deben venderse una a una, parejamente y a la inversa de la forma en que se construyeron.

Todos los hoteles de un grupo de color pueden venderse al mismo tiempo y entero o de una casa por vez (un hotel equivale a cinco casas). Deben venderse parejamente y a la inversa en que se construyeron.

Hipotecas

Las propiedades sin construcción pueden verse afectadas en cualquier momento y solo a través del Banco. Antes de poder hipotecar un solar con construcción, todos los edificios construidos en dicho solar y en todos los demás solares pertenecientes al mismo grupo de color deben ser vendidos al Banco a mitad de precio. El valor hipotecario de cada finca va impreso en la carta de dicha propiedad.

Sobre los solares o servicios públicos hipotecados no pueden cobrarse alquiler, pero sin embargo puede cobrarse sobre las propiedades no hipotecadas pertenecientes al mismo grupo.

Para levantar la hipoteca, el propietario tiene que pagarle al Banco la cantidad de la hipoteca más un 10% de interés. Cuando todas las propiedades de un grupo de color dejen de estar hipotecadas, el propietario podrá volver a comprar la casa al Banco por el precio total.

El jugador que hipoteca su propiedad retiene la posesión de la misma, de modo que ningún jugador podrá obtenerla levantando para ello la hipoteca que tiene el Banco. No obstante, dicho propietario puede vender su propiedad hipotecada a otro jugador a cualquier precio que convengan. El nuevo propietario podrá, si lo desea, levantar inmediatamente la hipoteca, pero para esto tendrá que pagarle al Banco el capital de la hipoteca más el 10% de intereses.



Si no desea levantar inmediatamente la hipoteca, tendrá que pagarle al Banco el 10% de interés en el momento que compre la propiedad del otro jugador y , si mas tarde decide levantar la hipoteca tendrá que pagarle al Banco la cantidad correspondiente a la hipoteca mas otro 10 % de interés.

Quiebra

Un jugador entra en quiebra cuando debe más a otro jugador o al Banco que lo que puede pagar. Si es deudor de otro jugador tiene que entregarle a dicho jugador todos sus bienes y retirarse del juego. Al efectuar dicha liquidación si posee casa o bancos, estos debe devolverlos al Banco a cambio de dinero, por cantidades igual a la mitad del precio que pagó por ellos, y entregarle el dinero al acreedor. Si posee propiedades hipotecadas también le entregara dichas propiedades a su acreedor, pero en este caso el nuevo propietario tiene que pagarle al Banco inmediatamente los interés del prestamos que son el 10% del valor de la propiedad. Luego de hacerlo. El nuevo propietario podrá, a su elección, pagar la cantidad correspondiente y terminar la hipoteca o retener la propiedad y tomar la decisión de levantar la hipoteca mas adelante. Si retiene de esta forma la hipoteca hasta un turno siguiente, cuando levante la hipoteca deberá pagar de nuevo los intereses.

Si el jugador es deudor del banco, en vez de serlo de otro jugador, (debido a que no puede pagar contribuciones o multas aun después de vender las construcciones y propiedades hipotecada) le entregará todos los bienes al Banco. En dicho caso el Banco venderá inmediatamente en una subasta todos los bienes adquiridos con excepción de los edificios. El jugador quebrado tiene que retirarse inmediatamente del juego. El último jugador que queda gana el juego.



Miscelánea

Los jugadores sólo pueden tomar prestado dinero del Banco mediante la hipoteca de sus propiedades. Ningún jugador puede pedir prestado o prestarle dinero a otro jugador.

