



Jon Ander Pérez Valdivielso
David Prado Muñoz
David Puerto Caldero

Objetivos y funcionalidad

El Obvial es un juego que mezcla los principios básicos de el juego de la Oca y el Trivial. El objetivo del juego es llegar el primero a la última casilla. Como en la Oca, existen casillas que harán el avance del jugador más rápido, y otras que, sin embargo, lo entorpecerán. Asimismo, el jugador deberá responder una pregunta por cada casilla en la que caiga, siempre que dicha casilla no tenga una función especial. Las preguntas que se le formulen al jugador podrán ser de los siguientes temas: Series de TV, Películas, Deportes, Videojuegos y Curiosidades. Además, por cada pregunta por cada respuesta correcta, el jugador obtendrá una tirada extra, lo que implica que el turno del jugador sólo finalizará en caso de que caiga en una casilla en la que deba quedarse retenido (CasillaRetención) o falle una pregunta. Cuando el jugador llegue a la última casilla, deberá responder correctamente una pregunta de cada tema, tarea para la cual dispondrá de tres turnos. Un turno pasará cuando el jugador falle una pregunta. El jugador sólo se proclamará vencedor en el caso de que responda las cinco preguntas en un mismo turno. En caso de que pasen los tres turnos y el jugador no haya conseguido responder las preguntas, el jugador volverá a la primera casilla, y deberá volver a recorrer todo el tablero.

Las casillas que harán que el jugador pierda el turno y permanezca retenido serán la casilla de posada, la casilla de laberinto, la casilla de carcel y la casilla de pozo. La casilla de la calavera, además de hacer que el jugador pierda el turno, también hará que éste vuelva a la primera casilla, obligándole a realizar otra vez el recorrido completo del tablero.

En este juego, como en la Oca, pueden participar desde 2 hasta 4 jugadores, y cada uno de dichos jugadores tendrá asignado un color (Amarillo, Azul, Rojo o Verde). Cada jugador elegirá su color antes de iniciar la partida, y también deberá anotar su nombre. Estas dos características se introducirán mediante una ventana emergente al pinchar en Opciones/Configurar Jugadores en la ventana principal.

En dicha ventana, además de la barra de menú con los campos Opciones y Juego, aparecerán el tablero, en el que estarán las fichas, y en la parte derecha de la ventana aparecerán las preguntas con cada una de las posibles respuestas. Encima de la pregunta aparecerá también el tiempo que le queda al jugador para responderla, y si el jugador no ha respondido correctamente a la pregunta pasado ese tiempo, perderá el turno.

En cada turno de cada jugador aparecerá una ventana con 4 opciones: tirar dado, pasar turno, turnos pendientes y acabar juego.

La opción tirar dado mostrará una ventana en la que aparecerá una animación de un dado "rodando" y un botón tirar dado. Cuando el jugador pulse dicho botón el dado se detendrá y mostrará la tirada del jugador.

La opción pasar turno sólo se activará cuando el jugador esté retenido en una casilla, y hará que el turno pase al siguiente jugador, decrementando el número de turnos que el jugador debe estar retenido en dicha casilla.

La opción turnos pendientes sólo se activará cuando el jugador esté retenido en una casilla, y sacará por pantalla una ventana con los turnos que el jugador debe permanecer en una casilla.

La opción acabar juego estará activa en cualquier turno. Al presionarlo, el juego se dará por terminado.

Una vez el jugador selecciona la opción tirar dado, su ficha avanza tantas casillas como las

indicadas por el dado, y en la parte derecha aparecerá una nueva pregunta. Si el jugador responde correctamente aparecerá una ventana congratulando al jugador por su acierto, para después mostrarle otra vez la ventana de turno. Sin embargo, si la respuesta es incorrecta, aparecerá una ventana mostrando al mensaje de error y después aparecerá la ventana de turno para el siguiente jugador.

Modelo del dominio:

Carta: Contiene la información de una carta, es decir, contiene la pregunta, las cuatro respuestas posibles y la respuesta correcta. También contiene un booleano que no dirá si la carta ha sido usada, (para que no se use otra vez) y un número de carta, que es el número por el que deberá buscarse la carta en su respectiva lista.

Casilla: Contiene la información relativa a la casilla (número de casilla y posición que ocupa dicha casilla en la interfaz gráfica, así como un número que actuará como identificador para saber de qué clase (normal, retención o traslado) es la casilla. La casilla también contiene la información de todos los jugadores que se encuentran en esa casilla.

CasillaRetención: Contiene toda la información que tendría una casilla normal (la hereda), y además de eso también contiene los turnos que tendrá que quedarse retenido el jugador en esa casilla en caso de caer en ella.

CasillaTraslado: Contiene toda la información que tendría una casilla normal (la hereda), y además de eso también contiene el número de la casilla a la que avanza el jugador en caso de caer en ella.

Dado: Contiene un método que cada vez que se le llame devolverá un número entre 1 y 6, correspondiente a la tirada del dado.

Inicializar: No contiene ningún tipo de información. Se encarga de cargar los datos de los archivos que contienen todas las preguntas y meter dichas preguntas en sus respectivas listas. Además de esto, también se encarga de "fabricar" la lista de casillas.

Jugador: Contiene la información referida a un jugador, es decir, su nombre, la posición en la que se encuentra, su identificador, el color de su ficha, un booleano que indica si el jugador ha finalizado la partida y un entero que muestra los turnos de retención que tiene que cumplir el jugador, en caso de que se halle retenido. En un principio cada jugador iba a tener asociada una ficha de su color, pero al final decidimos que el propio jugador fuese la ficha.

ListaCartas: Consiste en un ArrayList de Carta, es decir, en esta clase se almacenarán todas las cartas que hay para un tema, con sus respectivas preguntas y respuestas.

ListaCasillas: Consiste en tres ArrayList distintos, uno para las casillas normales, otro para las casillas de traslado y otro para las casillas de retención. Cada uno de estos ArrayLists contendrá las operaciones básicas de una lista.

ListaJugadores: Un ArrayList que contendrá a todos los jugadores que están tomando parte en la partida.

Listas: Es un array de 5 posiciones (una por cada tema). En cada una de esas posiciones se almacenará una ListaCartas.

Tablero: Es la clase encargada del desarrollo de toda la partida cuando la ejecutamos por consola.

Visualización: Esta clase era la encargada de mostrar toda la información de una partida por consola. La habíamos hecho por si acaso no nos daba tiempo a hacer la interfaz gráfica.

Problemas que nos han surgido

Tenemos la clase Casilla, de la que heredan CasillaTraslado y CasillaRetencion. A la hora de crear la clase ListaCasillas uno de sus atributos principales era un ArrayList <Casilla>. Al implementar el método buscarCasilla, dicho método nos devolvía un objeto de la clase Casilla. El problema surge al realizar la búsqueda de la clase CasillaRetencion, por ejemplo. Lo que sucede al buscar este tipo de casilla es que nos es imposible acceder a sus atributos puesto que el objeto que nos ha devuelto es de tipo Casilla, y esta clase no contiene los atributos que contiene CasillaRetencion.

Para solucionar este problema pensamos en hacer un cast, pero desechamos la idea puesto que pensabamos que los atributos a los que accedíamos no eran los que habíamos definido en la constructora.

La solución final que tomamos fue crear tres ArrayList distintos, uno para las casillas normales, otro para la clase CasillaRetencion y otro para la clase CasillaTraslado. De esta manera se soluciona el problema, pero duplicamos mucho código.

Después de implementarlo nos dimos cuenta de que haciendo un cast sí que podíamos acceder a los atributos, pero al vernos tan justos de tiempo decidimos dejarlo como estaba.

Otro problema que hemos tenido ha sido que al principio pensábamos en el proyecto como algo secuencial, pero ha medida que íbamos implementando la interfaz gráfica nos íbamos dando cuenta de que el usuario es el que lleva el mando, por lo que había que introducir el tratamiento de distintos eventos.

Al principio pensábamos en que la interfaz constase de un JFrame en el que estarían el tablero y la pregunta en un JPanel cada uno, cuando ésta apareciese. Al aparecer el menú del turno del jugador en un JDialog modal, el JPanel no dejaba al primer jugador responder la pregunta, debido a que mostraba el menú del turno del siguiente jugador, por lo que no se podía acceder a lo que aparecía en el JPanel.

Una de las soluciones que se nos ocurrió para este problema fue poner un Timer o algo parecido, para que esperase un tiempo hasta mostrar el menú del siguiente jugador, pero aún así seguía mostrándolo.

Finalmente hemos optado por mostrar la pregunta en otro JDialog, que ocupará el hueco en el que estaba el JPanel de la pregunta. Este JDialog lo hemos hecho modal, de modo que hasta que el jugador no responda a la pregunta o no se le pase el tiempo que tiene para responderla no aparecerá el menú del siguiente jugador.

Al solucionar este problema nos surgió otro. Y es que después de responder la pregunta todo sigue el camino que debería llevar, pero una vez termina el turno del jugador, bien porque ha fallado una pregunta o bien por otro motivo, se muestran otra vez todas las preguntas que el jugador ha tenido que responder anteriormente, como si el juego hubiese desarrollado un hilo paralelo al principal y hubiese ido almacenando ahí todas las preguntas que le han aparecido al jugador para después volver a formularlas.

Una de las soluciones que se nos ocurrió fue cambiar a IGPrincipal por IGTurnoJugador para que fuese Observer de Listas. Pensabamos que al hacer el update en esta otra clase se nos solucionaría el problema, pero continuaba. Después de horas dándole vueltas, nos dimos cuenta de que no necesitabamos ningún update: al tratarse de un JDialog independiente (y no de un panel como lo teníamos anteriormente), podríamos abrirlo y cerrarlo directamente.

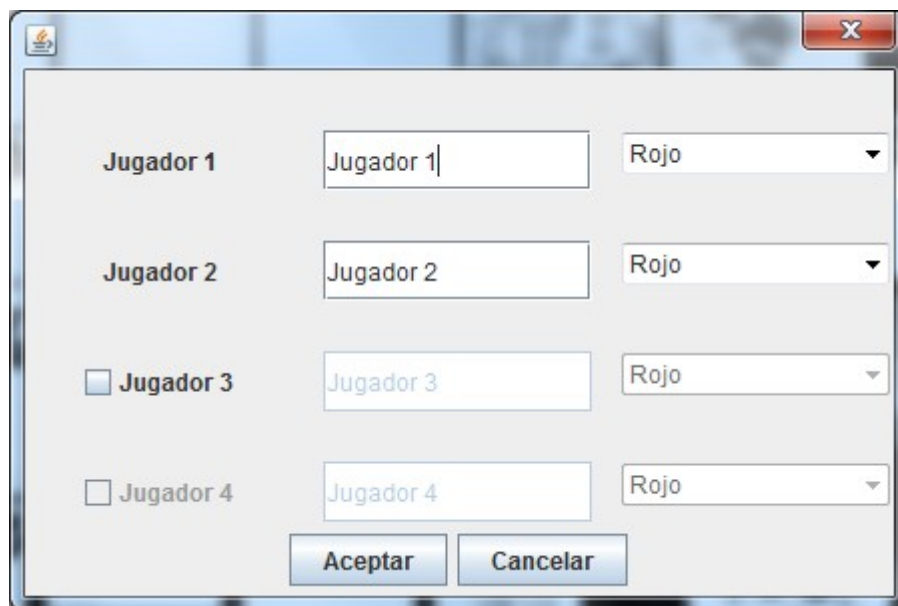
Interfaz Gráfica

Debido a la extensión del código, aquí solo daremos una idea general de lo que hace cada componente. En el código se explica con más detalle el funcionamiento.

IGPrincipal

- Juego.
 - Nuevo Juego: Inicia un juego nuevo. Desactivado si no se han elegido los jugadores.
 - Salir: Cierra el juego.
- Opciones.
 - Configurar Jugadores: Muestra la ventana en la que configuramos los jugadores.

IGopcionesJugadores



<input checked="" type="checkbox"/> Jugador 1	Jugador 1	Rojo
<input checked="" type="checkbox"/> Jugador 2	Jugador 2	Rojo
<input type="checkbox"/> Jugador 3	Jugador 3	Rojo
<input type="checkbox"/> Jugador 4	Jugador 4	Rojo

Aceptar Cancelar

- Columna 1.
 - CheckBox: Habilita la fila en la que se encuentra para poder dar valores al nuevo jugador.
- Columna 2.
 - TextField: Para darle nombre al jugador elegido.
- Columna 3.
 - Choice: En ella elegiremos el color de la ficha de nuestro jugador.
- Botón Aceptar: Recoge todos los valores para poder construir los jugadores.
- Botón Cancelar: Quita la ventana sin hacer nada más.

IGTurnoJugador



- Tirar dado: Crea una instancia de la clase dado para obtener un número y poder avanzar.
- Pasar turno: Activa cuando el jugador está retenido. Cada vez que es pulsado, actualiza los turnos de retención del jugador.
- Turnos pendientes: También activa cuando el jugador está retenido. Muestra cuantos turnos de retención le quedan al jugador.
- Acabar juego: Finaliza la partida actual

IGDado



- Botón Tirar Dado: Ejecuta el random de la clase Dado para obtener una número.

IGPregunta

The screenshot shows a mobile application interface for a question. At the top, a grey header bar contains the number '3'. Below this, the question text '¿Cuántas hermanas tiene Joey?' is displayed. The answer options are listed in a light grey area: '5', '6', '7', and '9', each preceded by an unselected radio button. At the bottom, there is a blue button labeled 'Aceptar'.

- Componente 1.
 - Tiempo: Muestra el tiempo que le queda al jugador para contestar la pregunta (30 segundos).
- Componente 2.
 - Pregunta: Muestra la pregunta de la carta.
- Componente 3.
 - RadioButtons: Sirven para elegir la opción correcta.
- Componente 4.
 - Botón Aceptar: Validamos nuestra respuesta. Si no hay ninguna respuesta marcada, muestra un aviso.