Mastermind

Identificador del equipo: 43.3

Paula Barrachina Cáceres: paula.barrachina@estudiantat.upc.edu
Marc Castro Chávez: marc.castro.chavez@estudiantat.upc.edu
Natalia Dai: natalia.dai@estudiantat.upc.edu
Enric Esteban Galian: enric.esteban@estudiantat.upc.edu

Versión de la entrega: 3.0

3ª Entrega PROP

Índice

1. Juegos de prueba

Objetos de prueba: Crear perfil (registro) y cargar perfil (inicio de sesión).
Elementos integrados a la prueba: Controlador de presentación, controlador de dominio, controlador de perfil, perfil, controlador de persistencia y gestor de perfil.
Views: menuInicio, inicioSesion, registro, menuPrincipal.
Valores estudiados: Nombre de usuario “paula” y contraseña “123” que ya estaba en el credentials.txt de persistencia de los perfiles, y nombre de usuario “marc” y contraseña “123” que crearemos ahora.
Operativa: Hemos compilado y nos ha salido la *view* de menuInicio donde se presentan 3 botones, “Iniciar sesión”, “Crear nueva cuenta” y “Salir del juego”. Le damos al primer botón entrando con “paula” de nombre y “123” de contraseña y nos deja entrar ya que estaba creada la cuenta de antes, entonces se nos mostrará la pantalla de menuPrincipal, que tiene varias funcionalidades que explicaremos más adelante. En el caso de que no estaba la cuenta creada (probamos con “marc” y “123”) nos sale un *popup* diciendo que hemos introducido campos incorrectos. Por otro lado si queremos volver para hacer una nueva cuenta le damos a “Volver” y le damos ahora al botón de “Crear nueva cuenta” en la *view* de menuInicio que nos redirigirá a la *view* de registro.

Aquí primero probamos a crear otra cuenta con nombre de usuario “paula” y cualquier contraseña y nos sale un *popup* avisándonos de que ya existe un usuario con ese nombre, que probemos con otro. En este momento introducimos los parámetros nuevos mencionados anteriormente “marc” y “123” y vemos que nos redirige a la pantalla de menuInicio, un indicio de que se ha creado correctamente. También podemos ver en el fichero credentials.txt una nueva entrada con esos valores. A continuación, probamos en la *view* de inicioSesion con “marc” y “123” y vemos que nos deja entrar a la pantalla de menuPrincipal.

Objetos de prueba: Cerrar sesión.
Elementos integrados a la prueba: Controlador de presentación, controlador de dominio y controlador de perfil.
Views: menuPrincipal y menuInicio.
Valores estudiados: -
Operativa: Hemos compilado y una vez iniciada la sesión correctamente, hacemos cualquier cosa dentro de la pantalla de menuPrincipal (o no) y luego queremos cerrar sesión así que pulsamos el botón de la esquina superior izquierda “Cerrar sesión”. Al hacerlo nos lleva al menú de inicio de sesión, la *view* de menuInicio, cerrando nuestra sesión de antes, como debería de pasar.

Objetos de prueba: Salir.
Elementos integrados a la prueba: Controlador de presentación
Views: menuInicio.
Valores estudiados: -
Operativa: Hemos compilado y una vez en la pantalla de menuInicio si pulsamos el botón de “Salir del juego” para comprobar que sale de la aplicación, se sale correctamente. El programa termina como se esperaba.

Objetos de prueba: Crear partida, Configurar Partida y Elegir Inteligencia
Elementos integrados a la prueba: Partida, controladores de presentación, dominio, persistencia, gestorPartida y partida.
Views: CrearPartida
Valores estudiados: Usuario “enric”, nombre de partida “test” y partida con nombre “repe” guardada.
Operativa: Hemos compilado el programa y una vez iniciada la sesión correctamente con el usuario “enric”, seleccionamos en el menú principal “crear una partida”. Una vez dentro, podemos hay varias opciones para configurar la partida. Primero, tenemos que poner el nombre de la partida. Si lo dejamos en blanco, nos saldrá un aviso cuando pulsemos “Crear”, ya que no se puede crear una partida sin nombre. En el caso de poner un nombre de una partida ya creada, también nos saldrá un aviso, ya que no pueden haber 2 partidas con el mismo nombre. Siguiendo la prueba le pondremos de nombre de partida “test”.

La siguiente opción es escoger la Inteligencia del juego. Está implementado el algoritmo *Five Guess* y el algoritmo Genético. Por defecto está escogido el primero y podemos ver que si seleccionamos el segundo se excluyen mutuamente.

Finalmente, para configurar los parámetros de la partida tenemos 2 opciones: Escoger una dificultad o introducir los parámetros manualmente. Por defecto nos dejará escoger una dificultad entre fácil, medio o difícil. Si pulsamos el botón de “Opciones avanzadas”, nos saldrá la opción de escoger los parámetros tú mismo. En esta prueba vamos a configurar la partida con parámetros personalizados de la siguiente manera:

Repeticiones NO, Espacios SI, Rol CB, Turnos 4, Intentos 4, Colores Extra: Magenta.

Le damos a “Crear” y vemos como nos salta a la siguiente vista para empezar el turno. Para ver si hemos creado la partida correctamente, podemos ver el texto plano donde se guardan las partidas y vemos la siguiente línea:

```
:enric,test,0,-1,4,CB,false,true,4,[MAGENTA],[],1;
```

Podemos apreciar que sí que se ha creado correctamente, ya que se han guardado los parámetros que le hemos introducido: (1) user: :enric; (2) nombre: test; (3) puntuación: 0 (4) turno actual : -1 (5) turnos: 4; (6) rol: CB; (7) Repeticiones: false; (8) Blanks: true; (9) Intentos: 4, (10) ColoresExtra: Magenta; (11) Vturnos (12) IA: 1 (Genetic).

Objetos de prueba: Borrar partida.
Elementos integrados a la prueba: Partida, controlador de dominio, controlador de partida, gestor de partida y controlador de persistencia.
Views: Gestionar Partidas.
Valores estudiados: Partidas guardadas en memoria: *game1, game2* del usuario “paula” y *game3* del usuario “nat”.
Operativa: Hemos compilado el programa y una vez iniciada la sesión correctamente con el usuario “paula”, seleccionamos en el menú principal *Gestionar partidas*. Una vez dentro, comprobamos que se nos muestran *game1* y game2, es decir, solo las partidas del usuario de “paula”, con los atributos indicados en la tabla. A continuación, seleccionamos el botón de borrar y comprobamos que salta un error al no haber seleccionado ninguna partida, entonces seleccionamos *game1* y presionamos borrar otra vez, y esta vez vemos que se elimina de la tabla correctamente. Si cerramos sesión, volvemos a iniciar sesión con el mismo usuario y volvemos al menú de gestionar partidas de este, vemos que efectivamente *game1* borrado se ha borrado adecuadamente. También comprobamos que en el archivo de memoria se haya borrado *game1*.

Objetos de prueba: Consultar Ranking, Consultar récords y gestor de ranking
Elementos integrados a la prueba: Perfil, controlador de dominio, controlador de perfil y controlador de persistencia.
Views: Consultar Récords, Consultar Ranking.
Valores estudiados: Dos récords guardados en memoria <nat,200> del usuario “nat”, <paula,100> del usuario “paula”, un récord creado en ejecución <paula,300> del usuario “paula”.
Operativa: Hemos compilado el programa y una vez iniciada la sesión correctamente con el usuario “paula”, seleccionamos en el menú principal *Consultar récords personales*. Una vez dentro, comprobamos que se nos muestra 100, siendo la máxima puntuación obtenida por usuario registrado. Volvemos al menú principal y seleccionamos *Consultar ranking global,* donde vemos que aparecen <nat,200> y <paula,100>. Volvemos al menú principal y jugamos una partida entera donde quedamos con 300 puntos. Vamos a comprobar en *Consultar récords personales* que nuestro récord se haya añadido y vemos que la tabla indica 300 y 100 mostrando nuestros récords obtenidos. Vamos a ver si el ranking ha sido actualizado y la tabla del ranking muestra <paula,300>,<nat,200> y <paula,100> concluyendo que el ranking se ha actualizado correctamente. Finalmente comprobamos que en el archivo de gestor de ranking están guardados los datos correctamente.

Objetos de prueba: Cargar partida.
Elementos integrados a la prueba: Partida, controlador de dominio, controlador de partida, gestor de partida y controlador de persistencia.
Views: Gestionar Partidas.
Valores estudiados: Partidas guardadas en memoria *game1*, *game2* del usuario “paula” y *game3* del usuario “nat”. Nueva partida creada en ejecución *game4*.
Operativa: Hemos compilado el programa y una vez iniciada la sesión correctamente con el usuario “paula”, seleccionamos en el menú principal *Gestionar tus partidas*. Una vez dentro, comprobamos que se nos muestran *game1* y *game2*, solo las partidas del usuario *paula*, con los atributos indicados en la tabla. A continuación, seleccionamos el botón de *Cargar* y comprobamos que salta un error al no haber seleccionado ninguna partida, entonces, seleccionamos *game1* y presionamos *Cargar* otra vez, esta vez viendo que sí nos lleva a jugar una partida. Si creamos una nueva partida *game4* y vamos al menú de carga vemos que efectivamente la partida aparece en las posibles partidas a cargar o borrar.

Objetos de prueba: Guardar Partida, Cerrar Partida.
Elementos integrados a la prueba: Partida, controlador de dominio, controlador de partida y gestor de partida.
Views: CM_tablero, CB_tablero y CM_CodeCreator.
Valores estudiados: Dos Partidas creadas *game1* y *game2*.
Operativa: Hemos ejecutado el programa y creado dos partidas distintas “game1” y “game2” con un usuario. Como al crear una partida esta se guarda automáticamente, al cargar las partidas nos muestra nuestras dos partidas creadas. Cargamos “game1”, jugamos dos rondas del primer turno, salimos y guardamos. Podemos ver que la *view* nos confirma que la partida se ha guardado correctamente y salimos al menú principal. Ahora volvemos al menú de carga para comprobar el estado de “game1”, que nos sale con el avance que hemos guardado. Probamos otra vez cargando el “game2” esta vez y jugamos una ronda del primer turno, pero esta vez salimos sin guardar pulsando directamente salir y comprobamos que esta vez al cargar no aparecen los datos, como se esperaba, ya que no hemos guardado nuestro progreso.

Para comprobar que en persistencia se guardan los valores adecuados al inicio de la ejecución vemos que no hay creados los game1 y game2 en el fichero. Una vez terminada la ejecución vemos que *game1* aparece con el progreso del turno donde nos hemos quedado, y que *game2* aparece como una partida recién creada y sin avances.
Vemos que los datos se han guardado correctamente comprobando que las funcionalidades van.

Objetos de prueba: Crear combinación, Respuesta Intento
Elementos integrados a la prueba: Partida, controlador de dominio, controlador de partida y controlador de presentación.
Views: CM_CodeCreator y CM_tablero.
Valores estudiados: Partida *game1*, entrada1 *[AZUL ROJO VERDE LILA]*, Partida *game2* entrada2 *[AZUL AZUL VERDE CYAN]*, Partida *game3* y entrada *[AZUL VERDE VACIO MAGENTA]*.
Operativa: Hemos ejecutado el programa y una vez iniciada la sesión correctamente, hemos creado "game1" con los parámetros por defecto e iniciado una partida como CodeMaker. Probamos a poner la entrada1 y acepta el código como válido. Guardamos y salimos al menú principal.
Creamos una nueva partida con nombre "game2" con colores repetidos y añadimos el color "Cyan". Empezamos la partida e introducimos como código la entrada2 y el sistema nos la acepta. Guardamos y repetimos el mismo proceso para la tercera partida *game3*, solo que esta vez añadimos el color "Magenta" y que se permitan vacíos. Jugamos e introducimos la entrada3 y el sistema la acepta. Con esto hemos comprobado que detecta bien la creación de combinaciones por parte del CodeMaker siendo una persona.

Para comprobar que las respuestas de intentos estén bien programadas hemos cargado *game1* y continuado la partida hasta acabar el turno como CodeMaker introduciendo los *keypeg* negros y blancos según las respuestas del algritmo (cambian en cada partida).

Objetos de prueba: Pedir Pista, Probar Combinación
Elementos integrados a la prueba: Partida, controlador de dominio, controlador de partida y controlador de presentación.
Views: CM_tablero.
Valores estudiados: Pistas dadas por el algoritmo en tres diferentes partida, entrada1 *[AZUL ROJO VERDE LILA]*, entrada2 *[AZUL AZUL VERDE CYAN]* y entrada3 *[AZUL VERDE VACIO MAGENTA]*.
Operativa: Hemos ejecutado el programa y una vez iniciada la sesión correctamente, hemos creado "game1" con los parámetros por defecto e iniciado una partida como CodeBreaker. Una vez el algoritmo haya generado el código, probamos a poner la entrada1 y acepta el código como válido, así que guardamos y salimos al menú principal.
Creamos una nueva partida game2 con colores repetidos y añadimos el color "Cyan". Empezamos, jugamos la partida e introducimos como intento la entrada2 y el sistema nos la acepta. Guardamos y repetimos el mismo proceso para la tercera partida *game3*, solo que esta vez añadimos el color "Magenta" y que se permitan colores vacíos. Jugamos e introducimos el intento entrada3 y el sistema la acepta. Con esto hemos validado que comprueba bien si las combinaciones introducidas por el jugador son correctas según las características de la partida.

Para comprobar que las pistas para el CodeBreaker persona en cada intento están bien programadas, hemos cargado el *game1* y continuado la partida hasta acabar el turno como CodeBreaker, y en cada intento hemos pedido una pista y la hemos usado como respuesta hasta finalizar el turno (cambian las pistas en cada partida según las genere el algoritmo).

2. Relación de les clases implementadas por miembro del equipo

A continuación mostramos una tabla con las clases que ha implementado cada miembro del grupo. De cada clase hecha por un miembro, este mismo se ha encargado de hacer el testeo y su documentación. En el caso de los controladores y drivers, hemos intentado repartir equitativamente el trabajo entre los 4, ya que no todas las clases requerían de un controlador o driver.
Para repartir las views hemos decidido que cada uno implemente las views correspondientes a sus clases. En cuanto a los gestores de documentos han sido implementados por los que hicieron el controlador de partida y perfil ya que pensamos que sería más adecuado. Finalmente para la implementación del último algoritmo, hemos decidido hacer pair programming dada la complejidad.

Paula Barrachina	**Marc Castro**	**Natalia Dai**	**Enric Esteban**
Codebreaker	Turno	Codemaker	Partida
Ranking	Algoritmo	Perfil	CtrlPartida
Record	CtrlDominio	CtrlPerfil	DriverPartida
DriverDominio	DriverAlgoritmo	DriverPerfil	GestorPartida
Algoritmo Genético	Algoritmo Genético	menuInicio	CtrlPersistencia
GestionarPartidas	CB_Tablero	inicioSesion	normas
ConsultarRecord	CM_Tablero	registro	CrearPartida
ConsultarRanking	CM_CodeCreator	menuPrincipal	FinPartida
CtrlPresentación	EmpezarTurno	GestorPerfil	
		GestorRanking	
		CtrlPersistencia	

Verde: Controladores
Azul: Drivers
Naranja: Vistas
Rojo: Persistencia