

# Changelog

## Semana 1:

17/06/2022:

- Se dio inicio al proyecto el cual se realizará con SurfaceView.
- Para una primera toma de contacto este día fue de pura lectura de los apuntes de forma que pudiera tener una idea de como utilizar el SurfaceView. Únicamente se creó el proyecto.

18/06/2022:

- Empecé la realización del proyecto siguiendo los pasos de los apuntes, primero empecé con la clase que sería SurfaceView, creándola y añadiendo los métodos necesarios para su funcionamiento.
- También se hizo en el MainActivity que el juego se iniciara en horizontal y que este no se pueda girar.

19/06/2022

- Para poder tener una visión inicial del proyecto y de lo que estaba haciendo cree la clase Pantalla de la cual heredarán el resto de pantallas que vaya creando después. Se le dio un color de fondo inicial para comprobar su funcionamiento.
- Una vez comprobado que la clase Pantalla funciona correctamente se crearon clases para las que serían el resto de pantallas del juego las cuales heredarán de Pantalla, éstas son: PantallaMenuPrincipal, PantallaJuego, PantallaEstadisticas, PantallaAjustes, PantallaCreditos, PantallaAyuda. Seguidamente se le añadió a cada una de ellas los métodos principales que iban a usar, estos métodos eran: dibujar(), actualizarFisica(), onTouch()... Y luego se le dio un color de fondo distinto a cada una de ellas.
- Ya con las pantallas creadas solo faltaba una forma de cambiar entre ellas para comprobar que las cree correctamente así que se añadió en el onTouch de la pantalla inicial (PantallaMenuPrincipal) un cambio de escena que me envíe a alguna de las demás pantallas, luego manualmente fui cambiando ese número para ver que el resto de pantallas estuvieran correctamente.
- Para gestionar un cambio ordenado entre pantallas cree un método el cual guardará únicamente las constantes (método Constantes) que voy a usar, estas constantes son el número de escena propio de cada pantalla, luego en mi clase SurfaceView y mediante un método de cambio de escena gestiono por un switch el cambio entre las distintas pantallas creadas.

## Semana 2:

24/01/2022:

- Diseñe un botón para volver a la pantalla anterior (la cual siempre será la de el menú principal). Esto lo hice con un método el cual coloque en Pantalla el cual dibujaria un rect (simbolizando mi botón) y en el onTouch devolviendo al pulsar en sus coordenadas el número de escena (el cual tengo en mi clase Constantes) de la pantalla PantallaMenuPrincipal. Seguidamente hice la prueba del botón desde la pantalla Estadísticas para ver que funcionara correctamente.

25/01/2022:

- Añadir el botón de volver a todas las pantallas creadas menos PantallaMenuPrincipal.
- Empecé a editar lo que sería mi primera pantalla, la pantalla PantallaMenuPrincipal.
- Empecé a dividir la pantalla según las opciones (botones) que tendría y realice pruebas de coordenadas con los Rect.

26/01/2022:

- Ya con el cálculo para los Rect hecho y todos colocados a la misma distancia uno del otro y del mismo tamaño empecé a escribir y calcular las coordenadas del texto que tendrían dentro.
- Me dediqué a darle una apariencia aceptable al menú principal con todo centrado y una imagen de fondo la cual en principio será de prueba.

## Semana 3:

31/01/2022:

- Ya con la pantalla del menú principal un poco diseñada tocaba con las demás pantallas, empecé a darle el mismo fondo a todas (lo heredan de Pantalla) y comencé a calcular para cada una su título, de forma que quedara centrado en todas.
- Una vez todas tenían su título centrado empecé con la clase PantallaEstadísticas a darle formato y calcular las coordenadas de los textos que tendría.

01/02/2022:

- Continuo terminando de dar un diseño básico a cada pantalla, ahora con la de Ajustes, calculando coordenadas y el texto que tendría. Faltarán la pantalla de créditos y la de ayuda ya que los créditos los tendré una vez tenga todos los sprites, música y efectos a usar; y la pantalla de ayuda serán imágenes del juego con la especificación de cómo se juega y su dinámica.

02/02/2022:

- Día dedicado a la búsqueda de sprites y efectos varios para luego poder tener una lista de la cual escoger.

## **Semana 4**

07/02/2022

- Se crea una carpeta de personajes donde se guardarán todos los personajes del juego. En ella se crean las clases Héroe, Vida y dos carpetas más que son las armas y los enemigos donde se guardaran las clases para cada arma y los distintos enemigos respectivamente.

## **Semana 5**

## **Semana 6**

En las semanas anteriores se estuvo trabajando en un diseño de juego el cual terminó siendo descartado ya que se me dificultaba su realización en el tiempo solicitado. Aunque no se descarta esa idea para un proyecto futuro.

Se conservan de igual forma ciertas clases creadas en estos días como la de Héroe, Vida, Enemigo, CriaturaOrdinaria.

La clase Héroe simplemente dibujaba un Rect el cual será ahora dibujado en el centro de la pantalla (antes a un costado).

La clase Vida simplemente es un Rect y es la que indica la vida de cada personaje, todos los personajes tanto enemigos como el héroe disponen de un objeto Vida el cual indicará su vida actual mediante una barra debajo del personaje.

La clase Enemigo es una clase de la cual iban a heredar todos los tipos de enemigos, actualmente solo heredará de ella CriaturaOrdinaria y si el tiempo es suficiente un enemigo Jefe.

la clase Arma la cual dispone de sus propios métodos para dibujar, actualizar y mover la bala creada , este último moverá la bala a la derecha o izquierda

dependiendo de donde pulsemos (simplemente si se supera la mitad de la pantalla dispara a la derecha, sino a la izquierda).

La clase `CriaturaOrdinaria` es un tipo de enemigo el cual como el resto de enemigos dispone de su propia imagen, velocidad, daño de ataque y vida. (Se tiene previsto añadir de futuro o si el tiempo de entrega alcanza distinta velocidad de ataque para cada enemigo).

25/02/2022:

- Se toma la decisión de rehacer el juego con una dinámica más sencilla.

27/02/2022:

- Se da inicio a la nueva dinámica del juego la cual será muy parecida a la pensada anteriormente. Todo lo realizado a partir de ahora hasta nuevo aviso es en la `PantallaJuego`.
- Se crea un objeto Héroe el cual estará posicionado en el centro de la pantalla.
- Se crea un objeto tipo `CriaturaOrdinaria` para hacer pruebas de funcionamiento, el enemigo aparecerá por un costado de la pantalla y se moverá hasta las coordenadas del Héroe gracias a sus métodos de `moverEnemigo()`, `dibujar()`, y `actualizarRect()`, este último pertenece a `Enemigo`, el resto son propios de cada tipo de enemigo ya que todos realizarán distintas acciones.
- Una vez comprobado que el enemigo se acerca al Héroe y funciona correctamente pasamos a hacer pruebas para poder disparar con el Héroe, para ésto añadimos a nuestra clase de juego un objeto de tipo arma.
- Primero hacemos pruebas con una bala, al pulsar una vez (`onTouch` de la pantalla de juego) se crea una bala y ésta se mueve hasta superar el ancho de pantalla máximo o mínimo, dependiendo de hacia donde se haya disparado (lo comprobamos con el `onTouch()` si tocamos a más de la mitad del ancho de pantalla va a la derecha, sino a la izquierda).
- Una vez comprobado que funciona correctamente la bala hacemos pruebas con un array de balas el cual se ejecutará cada vez que se pulse la pantalla e irá borrando del array dichas balas una vez superen el límite de pantalla.

## Semana 7

28/02/2022

- Ya que vimos que el array de balas funciona correctamente añadimos la opción para que al intersectarse las balas con los enemigos también se eliminen del array, dando el efecto de impacto.

- Ya comprobado que las balas funcionan correctamente pasamos a hacer las pruebas con múltiples enemigos usando el mismo sistema que con balas, creando un array de enemigos los cuales mediante un número random se decidirá si aparecen de un lado o del otro. Esto lo hacemos con el método creado crearOrdaEnemigos() el cual se encargará de crear las hordas de enemigos que vendrán para cada ronda.
- Los enemigos aparecen correctamente pero vienen todos apilados de golpe por lo que para evitar esto implementamos en el método dibujar() un numero random (de momento entre 1 y 170) en el cual si sale un 1 aparece un enemigo del array, de esta forma hacemos que aparezcan aleatoriamente y poco a poco de un lado o de otro.

01/03/2022

- De momento las balas y los enemigos funcionan bien pero las balas impactan sobre los enemigos y nada sucede por lo que añadimos en el intersect de la bala el get de la vida del enemigo que intersecta la bala y le quitamos la vida correspondiente al daño que tenga la bala, de esta forma conseguimos el efecto de que matamos a los enemigos.
- Ya podemos matar a nuestros enemigos pero una vez muertos no aparecen más por lo que vamos a implementar el sistema de rondas para intentar hacer que cada vez que se acabe el array de enemigos se vuelva a llenar el array pero esta vez con más enemigos.
- Creamos el método comprobacionDeRondas() en el que, si el array de enemigos llega a tamaño 0 llamamos a este método, en el aumentamos el número de una variable creada llamada ronda y aumentamos el tamaño con el que se llenará nuevamente el array de enemigos.
- Visto que el método anterior funciona correctamente y que cada vez que acabamos con todos los enemigos vienen más vamos a tratar de crear un tiempo de espera entre cada ronda para que sea como un tiempo de reposo y preparación para la ronda que viene, esto lo hicimos con el System.currentTimeMillis() de forma que al acabar con todos los enemigos nos aparezca la ronda a la que pasamos en la esquina superior derecha de la pantalla durante un tiempo y pasado ese tiempo empiece la nueva ronda con más enemigos.

02/03/2022

- Para variar un poco más la dificultad del sistema de rondas en el método comprobacionDeRondas() introducimos que cada 3 rondas (de momento, luego se puede aumentar o disminuir) aumente la vida total de los enemigos en una cantidad que especifiquemos (de momento cada 3 rondas aumentará la vida de los enemigos en 30).
- Introdujimos una especie de sub menú al morir de forma que cuando el héroe se quede sin vida aparezca en la misma pantalla el mensaje de "Has muerto"

junto con las opciones de reintentar la cual nos reiniciará la pantalla de juego empezando en la ronda 1 y la opción de salir la cual nos enviará a la pantalla del menú principal.

- El resto del día fue dedicado a búsqueda de mas sprites y música de juego.

05/03/2022

- Introdujimos los sprites del héroe, esto lo hicimos en la propia clase de Héroe() de forma que al pulsar la pantalla (en el onTouch()) se dispare una bala y de el efecto de que nuestro personaje está disparando, aunque de momento apunta a una sola dirección.
- Introdujimos el método espejo() para invertir la imagen en caso de que nuestro héroe dispare al lado contrario, de momento no funciona correctamente.
- Creamos más rect alrededor del rect del propio héroe para dar un efecto de que está encerrado en una especie de búnker, de momento únicamente con un color de fondo.
- En la pantalla Ajustes se implementaron los botones para encender o apagar la música, los efectos de sonido, la vibración, cambiar el idioma o borrar el historial de estadísticas del juego y su cambio de texto con el onTouch() al pulsar en ellos.

## Semana 8

07/03/2022

- Se arregló el efecto de disparo de nuestro personaje con el método de espejo(), ahora ya dibuja correctamente los sprites dependiendo del lado al que disparemos.
- Introdujimos un control de frames para que se dibuje de forma más suave los sprites al disparar.

08/03/2022

- Se continuó con la búsqueda de música para el juego ya que no convencían las que se habían encontrado.
- Se implementó la música al juego, tanto para el menú como para el juego en sí aunque de momento con algunos fallos, ya que al arrancar la app la música del menú suena doble.
- También incorporamos un sonido de pistola para cuando se dispare el arma.
- Cálculos en la clase Vida() para tratar de dar efecto de que baja la barra al recibir daño aunque de momento no funciona correctamente.
- Implementación de los sprites de los enemigos, en este caso creamos los arrays en la clase Enemigo() pero le introducimos las imágenes desde la

clase `CriaturaOrdinaria()` ya que son las imagenes de ese enemigo en específico.

- Incorporamos igual como con el héroe un control para los frames de los enemigos de forma que se dibuje su efecto de caminar de una forma más suave y controlada, de momento están sólo los sprites de efecto al caminar y no se ajusta bien al rect del enemigo por lo que tendremos que recortar las imágenes o buscar distintos sprites.

09/03/2022

- Cree la clase `Datos` la cual va a contener los métodos necesarios para el guardado y devolución de datos tanto como para los strings, int, o boolean que se necesiten.
- Se intentó aplicar el guardado de datos con la pantalla ajustes aunque por el momento no funciona correctamente.

## Semana 9

14/03/2022

- Se cambió la imagen del fondo del menú para que no sea igual que la del juego.
- Se corrigió un error por el cual la música sonaba doble al iniciar la app.
- Se añade sonido a los enemigos.
- Se implementa el guardado de datos de las estadísticas mediante la clase `Datos`.

15/03/2022

- Se hicieron cambios en el diseño de la pantalla de estadísticas.
- Se hicieron cambios en el diseño de la pantalla de ajustes.
- Se implementó el guardado de todas las configuraciones en la pantalla ajustes gracias a la implementación de la clase `Datos`.
- Optimización de recursos y organización de código en la pantalla ajustes.
- Se creó la clase `PantallaBorrarHistorial` para la confirmación al pulsar en la opción de borrar historial de la pantalla de ajustes ya que resulta más ordenado que dibujarlo en la misma pantalla de ajustes. Por lo tanto se añadió su constante de `numEscena` en la clase `Constantes` y se añadió.
- Se hicieron cambios en el texto mostrado cuando se pierde la partida.
- Se realizó parte del comentario en javadoc de la app.
- Se optimizó y reorganizó el código en la clase `Pantalla`.

16/03/2022

- Implementación de las personas que trabajaron en la pantalla de créditos.
- Se empezó a implantar el sensor de movimiento.

- Se avanzó en el comentado de código con javadoc.
- Corregido un error por el cual seguía sonando la música del menú cuando se entraba a jugar.
- Implementación del segundo hardware con acelerómetro en el MainActivity, el cual se utilizara en la pantalla de juego como bonus para que al agitar el móvil se acabe con todos los enemigos que hay ACTUALMENTE en la ronda, es decir, no aplica a los que aún no han aparecido.
- Se implementaron los métodos necesarios para el funcionamiento correcto del acelerómetro. Como onSensorChanged() , onAccuracyChanged() (este es necesario para la implementación de los sensores aunque no lo utilizaremos), onPause() y onResume().
- Se implementó el acelerómetro en el juego de forma que cada 5 rondas (en principio) se dispondrá del bonus. Dicho bonus no será acumulable.
- Se implementaron los bitmap de las criaturas ordinarias al momento de atacar y cuando sean golpeados por una bala.
- Se añadió que las criaturas ordinarias al ser golpeadas por una bala estas frenan brevemente su avance.
- Se modificó el efecto de vibración para que vaya acorde con el ataque de los enemigos.

17/03/2022

- Se implementó un nuevo estilo de fuente para todo el juego.
- Se implementó una imagen para el powerUp del KBoom la cual aparecerá cuando éste esté disponible, también se implementó la posibilidad de activarlo tocando en dicha imagen.
- Se implementaron efectos de sonido para cuando obtenemos el kboom y para cuando lo utilizamos.
- Se implementaron efectos de sonido para cuando seamos golpeados por los enemigos.
- Se implementaron más nombres en la pantalla de créditos.
- Se hicieron mejoras en las distintas clases y se borró código innecesario.
- Se terminó la pantalla de ayuda.
- Se añadió icono personalizado al juego.



