

Práctica 1: Squash

Objetivo

Vas a crear un juego de Squash que consiste en una pelota que rebotará contra las paredes. El jugador, moviendo una “raqueta”, la rebotará de regreso. El jugador pierde cuando no alcanza a pegarle a la pelota y ésta sale de pantalla por la parte inferior.

Los objetivos de aprendizaje son:

- creación y animación de objetos UIView
- uso de ViewControllers
- uso de la documentación para implementar nuevas funciones.

Lee con atención los requerimientos de esta aplicación antes de iniciarla

Requerimientos mínimos

1. La aplicación será universal, esto es, que se ejecute nativamente en iPhone o en iPad. Por nitidamente se refiere a que la aplicación utilice el espacio de pantalla apropiado según el dispositivo.
2. La única orientación permitida para cualquier dispositivo es apaisada.
3. La aplicación tendrá cuatro pantallas implementadas cada una con su MVC correspondiente:
 - a. Pantalla inicial
 - b. Instrucciones
 - c. Juego
 - d. Marcadores
4. Pantalla inicial:
 - a. Al iniciar la aplicación el usuario verá la pantalla inicial con las opciones “Iniciar juego”, “Marcadores recientes” e “Instrucciones” las cuales llevarán a las pantallas correspondiente.
5. Instrucciones
 - a. El usuario verá las instrucciones del juego y un botón que regrese a la pantalla inicial
6. Juego:
 - a. El usuario moverá la raqueta horizontalmente arrastrándola con un dedo.
 - b. La pelota puede rebotar en la pared superior y en las laterales.
 - c. Cada vez que la pelota golpee la raqueta el marcador se incrementará un punto. El marcador debe ser visible al usuario en un lugar apropiado.
 - d. Obstáculos:
 - i. cada vez que el marcador llegue a múltiplos de 10 deberá aparecer un obstáculo en el cual la pelota pueda rebotar.
 - ii. la ubicación del obstáculo será aleatoria cumpliendo las siguientes condiciones:
 1. no debe aparecer obstáculo alguno en el tercio inferior de la pantalla.
 2. los obstáculos no se deben encimar, tocar ni salir de la pantalla en ninguna dirección.
 - e. tamaño:
 - i. todos los obstáculos deben ser de la misma altura
 - ii. debe haber tres tamaños de ancho

- iii. cada tamaño será de un color característico
- iv. se seleccionará al azar.
- f. deberán ser implementados mediante una subclase de `UIView` cuyo inicializador designado recibirá las coordenadas y tamaño correspondiente.
- g. Cuando la pelota salga del campo aparecerá la leyenda respectiva en una alerta que contenga un botón que lleve a la vista de marcadores máximos (TIP: revisa la clase `UIAlertController`).

7. Marcadores:

- a. Se mostrará:
 - i. el puntaje máximo histórico
 - ii. los de los últimos 5 juegos
 - iii. para cada entrada se debe mostrar el día, mes, año y hora (en formato hh:mm) en que terminó la partida correspondiente.
 - iv. un botón para borrar todos los marcadores
 - v. un botón para regresar a la pantalla inicial
- b. La información mostrada debe persistir entre ejecuciones de la aplicación (TIP: revisa la clase `UserDefaults` en la documentación)

Tips

1. Para el diseño gráfico de la aplicación te puedes inspirar en juegos similares de los años 80 como Pong.
2. Configura el tamaño de la raqueta y la velocidad de la pelota para que el juego no sea ni muy fácil ni muy difícil.

Puntos extra

1. Que la pelota cambie de color cada vez que choque con la raqueta. (+0.5)
2. Sonidos: cada vez que la pelota choque con la pared o con la raqueta que se emita un sonido. El sonido debe ser distinto para los choques con la pared o la raqueta. Así mismo que haya una melodía al iniciar y finalizar las partidas. (+1)
3. Pausa: añadir un botón de pausa que pause y reanude el juego (+1)

Evaluación

Uno de los objetivos del semestre es que escribas código que compile sin errores ni warnings. Algunos motivos por lo que la calificación se afectará:

- El proyecto no compila (se evaluará con cero)
- El proyecto compila con warning (se deducirá dos puntos)
- No se completaron los requerimientos mínimos
- Algún concepto fundamental no se respetó (por ejemplo división entre modelo, vista y controlador) (-1 punto)
- El código no está comentado. Los comentarios deben permitir la comprensión del código. Como mínimo se espera que se describa lo que cada método hace, las variables que recibe y lo que regresa (-1 punto)
- El código está descuidado y es de difícil lectura (indentación inconsistente, escritura incorrecta de variables, objetos, clases y métodos, etc) (-1 punto)
- Por cada error de ortografía se deducirá una décima.

Archivos a entregar

Las tareas y proyectos se deben entregar en un .zip que contenga el subdirectorio de tu proyecto. Asegúrate de enviar todos los archivos necesarios para la compilación. El archivo .zip se debe llamar como tu número de cuenta y lo deberás subir a Brightspace en el espacio correspondiente a la actividad. Puedes realizar varias entregas, pero sólo la última se evaluará. No se permitirá entregas después de la fecha límite.