



Parcial 2

PORTABILIDAD Y OPTIMIZACION

Cecilia Illueca | 1er Año | 26/Jun/2024

Introducción

Este documento tiene el propósito de explicar cómo cumplí con cada una de las 10 consignas del parcial

Clicker

- Hacer un juego simple que cuente cuantos clicks (o taps) se hicieron en un botón en un lapso de 10 segundos.
- El juego debe tener versión web y versión para dispositivos móviles. La interfaz visual se tiene que adaptar a la plataforma (y resolución) en la que se esté jugando.
- Puedo ver el mejor puntaje obtenido hasta el momento en la pantalla de juego.
- Puedo ver los créditos entrando a la pantalla de créditos.
- Ads: En Android, implementar:
 - un banner que esté visible todo el tiempo,
 - un interstitial que aparezca cada vez que se termina un juego y no se batió un nuevo récord.
 - un video reward que, al verlo, te dé 2 segundos mas en la próxima partida.
- Local Notis: En Android, implementar
 - una push notification que llegue a los 10 mins de haber jugado por última vez (en el texto de la noti debe decir el nombre del alumno/a que hizo el juego).

Consignas del parcial.

¿Cómo el juego cuenta cuántos clics o toques se hacen en un botón dentro de un período de 10 segundos?

La clase ClickerBehaviour se encarga de contar los clics. Cada clic se registra a través del método RegisterClick, que incrementa la propiedad Clicks. El setter de la propiedad Clicks invoca el método OnClicksChanged, el cual activa cualquier método suscrito, como la actualización de la interfaz de usuario (UI). El tiempo es gestionado por la clase timer, a través de una corrutina, que cuenta hasta el tiempo máximo establecido por el game manager. Cuando alcanza ese tiempo un booleano llamado timesUp se setea como cierto a través de un setter que invoca métodos como ReevaluateHighscore de la clase HighscoreBehaviour y EndGame del GameManager.

¿Cómo el juego tiene una versión web y una versión móvil?

El código utiliza directivas de compilación dependientes de la plataforma para diferenciar entre las versiones web y móvil. Por ejemplo, en las clases InterstitialManager y BannerManager, las directivas `#if UNITY_ANDROID || UNITY_IOS` y `#endif` aseguran que ciertos bloques de código solo se compilen en plataformas móviles. Este enfoque permite al juego tener implementaciones diferentes para las versiones web y móvil. La clase Initializer, encargado de inicializar las diferentes clases del programa con las referencias correctas utiliza esas directivas para inicializar las clases con la referencia adecuada para la plataforma del usuario.

¿Cómo se adapta la interfaz visual a la plataforma y resolución en la que se está jugando?

El código configura diferentes elementos de la interfaz de usuario (UI) y lienzos (canvases) para activarse o desactivarse según los estados del juego e interacciones del usuario. El componente Canvas Scaler de Unity se encargaría del escalado de la resolución, asegurando que la UI se adapte a diferentes tamaños de pantalla y resoluciones. Además, hay diferentes canvases para diferentes plataformas. La clase Initializer se encarga de activar el canvas correcto, de acuerdo a la plataforma del usuario, al inicio del juego.

¿Cómo pueden los usuarios ver la puntuación más alta alcanzada hasta ahora en la pantalla del juego?

El código utiliza componentes TextMeshProUGUI para mostrar información del juego. La puntuación más alta se almacena usando PlayerPrefs y se muestra en la pantalla del juego actualizando un elemento TextMeshProUGUI de manera similar a cómo se muestran los clics en ClickerBehaviour.

¿Cómo pueden los usuarios ver los créditos yendo a la pantalla de créditos?

La clase CreditsButton incluye un método OnClickCreditsButton que activa _creditsCanvas si _endScreen no está activo. Esto permite a los usuarios ver la pantalla de créditos al hacer clic en un botón que activa este método. Además, el método CloseCredits se usa para cerrar la pantalla de créditos cuando finaliza el tiempo, para mostrar la pantalla de fin de juego adecuadamente.

¿Cómo implementé un anuncio de banner que siempre está visible en Android?

La clase BannerManager es responsable de cargar y mostrar anuncios de banner. En el método Start, determina el ID de unidad de anuncios adecuado según la plataforma. El método Show configura y muestra el anuncio de banner utilizando la API de Advertisement de Unity, asegurando que el banner esté siempre visible en las plataformas admitidas. La clase AdsManager muestra el banner, y las clases InterstitialBehaviour y RewardedAdBehaviour se aseguran de ocultar al banner mientras están mostrando un ad, y volverlo a activar cuando esta se cierra.

¿Cómo implementé un anuncio intersticial que aparece cada vez que termina un juego sin establecer un nuevo récord en Android?

La clase InterstitialManager se encarga de cargar y mostrar anuncios intersticiales. El método ShowInterstitial muestra el anuncio si se ha cargado correctamente. Este método es llamado desde la lógica del juego desde la clase HighscoreBehaviour cada vez que un juego termina y no se establece un nuevo récord, asegurando que el anuncio se muestre en el momento adecuado.

¿Cómo implementé un anuncio de video recompensado que otorga a los jugadores 2 segundos extra en el próximo juego después de ser visto en Android?

La clase RewardedAdsManager está diseñada para manejar esto. El método ShowRewardedAd mostraría el anuncio, y al completarse, le da a los jugadores 2 segundos extra en el siguiente juego.

¿Cómo implementé una notificación push que llega 10 minutos después de que se jugó el último juego en Android?

La clase LocalNotisManager configura y programa notificaciones push. En el método ScheduleNotis, se configura una notificación para que se active 10 minutos después del tiempo actual. Este método se llama después de verificar si se han creado los canales de notificación, asegurando que la notificación push se envíe en el momento correcto.

¿Cómo aseguré que el texto de la notificación incluya el nombre del estudiante que hizo el juego?

El método ScheduleNotis en la clase LocalNotisManager establece explícitamente el texto de la notificación para incluir el nombre del estudiante: "Este es el push notification de Cecilia Illueca, PYO_Parcial 2!". Esto asegura que cada notificación programada incluya el nombre del estudiante, cumpliendo con el requisito del proyecto.