

## Ejercicio 1

Para arreglar la UI se cambió el UI Scale Mode del Canvas Scaler a "Scale With Screen Size" y se puso una resolución de referencia de 1920x1080 ya que tiene un aspecto ratio de 16:9. También se acomodaron los anchors de los elementos de UI para que se ajusten correctamente.

La transición de pantallas está compuesta por 2 clases, Screen y Screen Navigator. Screen tiene la lógica para abrir y cerrar las pantallas y ScreenNavigator se encarga de indicar a las Screen cuándo tienen que abrirse o cerrarse.

## Ejercicio 2

En los scripts de la carpeta Enemies se quitó "using Project.Exersice3.Repositories;" ya que no es necesaria. Se cambió el nombre del namespace "DefaultNamespace" por "NPCs" y se separaron las interfaces del archivo IEnemy en distintos archivos (INPC, IEnemy). Ya que un Enemy sería un NPC la interfaz IEnemy hereda de INPC. La interfaz IBoss no sería necesaria ya que, por ejemplo, un Boss sería un enemigo, y los métodos que tiene la interfaz IBoss también están en IEnemy, por lo que puede implementarse IEnemy directamente.

En el script de "gamemanager" se corrigió el nombre ya que no respeta la nomenclatura, se separó en distintos archivos el contenido según la funcionalidad a la que correspondía: AudioManager, SceneController y Player. Se corrigió el nombre del método PlayAudio en el AudioManager ya que estaba en español. Se agregó el GameManager, AudioManager y SceneController al namespace Managers.

Para los scripts de botones se creó la clase abstracta BaseButton la cual tiene los métodos IsActive y OnClick. El método OnClick es abstracto para que ExitButton, PlayButton y ReturnButton, que heredan de BaseButton puedan tener su propia funcionalidad.

## Ejercicio 3

Se creó la clase UIController para poder obtener la información de los LocalRepository solicitados y mostrarla en pantalla.

## Ejercicio 4

El problema que tiene es que si se quieren agregar más opciones de personalización hay que acceder al script para modificar el switch y agregar las nuevas posibilidades además de que el array puede tener un menor tamaño del que se usa en el switch y provocar errores o que lo que se solicita no corresponde con lo que se devuelve porque están los objetos del array en un orden que no corresponde con las posibilidades del switch. Una opción para mejorarlo sería

tener un array o lista de Cloth en vez de GameObjects y modificar el método GameObject GetClothPrefab(Cloth cloth) a GameObject GetClothPrefab (string clothId) y con un loop buscar entre todos los Cloth de la colección cuál corresponde al clothId y devolver el prefab. Otra opción similar es tener un diccionario Dictionary<string, GameObject> donde string es el ID de cada Cloth y GameObject sería el prefab correspondiente a ese ID y con el mismo método de la opción anterior se puede obtener el prefab en base al ID dado. Se puede utilizar también un ScriptableObject para almacenar los datos de ID y los prefab como si fuera una base de datos.