

Prueba técnica argumentación

Link del repositorio: <https://github.com/JoanMarcBardes/PruebaTecnicaUnityDeveloper>

Para hacer la prueba se ha seguido los principios de **SOLID** y se a tenido en cuenta la cambiabilidad y modularidad del código, por ello se ha utilizado el patrón de diseño **Adapter**.

De este modo se han creado diferentes interfaces: [IAnswers](#), [IScore](#), [IStatament](#), [IFade](#) y [IAnswerButton](#). Con esto podemos crear la clase [GameControllerBase](#) donde solo utiliza interfaces y así se pueden crear diferentes clases que hereden de ella donde se especifique que tipos de [IAnswers](#), [IScore](#), [IStatament](#)... se van a utilizar. Como se ha hecho con [GameController](#) donde se utilizan las clases [Score](#), [Statement](#) y [Answer](#).

Del mismo modo la clase [Answers](#) y [Statement](#) solo utilizan interfaces para poder especificar los tipos en sus clases derivas y así proporcionar mayor escalabilidad. Cave remarcar que la clase [Statement](#) se ha hecho abstracta para obligar a sus herederos a implementar la función **SetItem**, ya que la clase consta de un object que debe ser tratado expresamente. Con esto lo que logramos es que el enunciado pueda ser cambiado fácilmente, es decir, se ha creado la clase [StatementText](#) donde el enunciado es un texto, como se requiere para la prueba, pero perfectamente se podría crear una nueva clase donde el enunciado fuera una imagen y el código se adaptaría perfectamente sin tener que modificar nada.

Con [IFade](#) podemos hacer el fadeIn o fadeOut sin importar de que tipo de objeto son, ya que con [FadeGraphic](#) podemos hacer el fade de objetos que tengan un componente Graphic (textos, imágenes...) y con [FadeCanvasGrup](#) que tengan el componente CanvasGrup.

Se han utilizado namespaces para tener el código bien organizado y ademas asegurarse que no hay dependencias entere clases que no deseamos. De este modo clases bases solo deberán incluir namespaces que sean Shared.

También se ha creado la clase statica [Utils](#), que contiene funciones staticas muy genéricas y así desacoplar el código.