

Prueba Técnica Madbricks Agosto 2019

El objetivo de esta prueba es crear 3 herramientas modulares que se puedan usar en varios juegos. Y finalmente integrarlas en un juego particular.

1. Cree una clase GameManager que implemente la interfaz:

```
public interface IGameManager
{
    //El parámetro Vector2 es la posición en píxeles en pantalla del Tap.
    public event Action<Vector2> OnTouchStarted;
    public event Action<Vector2> OnTouchEnded;
    public event Action<Vector2> OnTapped;

    public bool isTouching;
}
```

Esta clase debe funcionar tanto en Android como en PC y en el editor. En PC y en el editor el click izquierdo del mouse reemplazará la acción del dedo.

2.
 - a. Cree una interfaz ICharacterManager con los siguientes eventos y propiedades expuestas:
 - i. event Action OnCharacterDied;
 - ii. event Action<int> OnScoreChanged; //El parametro int es el cambio en el puntaje.
 - iii. int Lives {get;}
 - iv. int Score {get;}
 - b. Cree una clase CharacterUIManager con referencia a un ICharacterManager con los siguientes campos:
 - i. [SerializeField] private Text livesText;
 - ii. [SerializeField] private Text scoreText;Estos textos
3. Desarrolle un PoolManager y un abstract PoolObject con las siguientes características:
 - a. El PoolManager debe tener un campo de objectPrefab, en donde se asigna un prefab del objeto de tipo PoolObject que se desea instanciar.
 - b. Debe existir una clase PoolObjectParams con un atributo public Vector3 startingPosition.
 - c. El PoolObject debe tener un método público abstracto InitializePoolObject(PoolObjectParams parameters) que se va a usar para iniciar el PoolObject según la necesidad del juego.
 - d. El PoolManager debe tener un método público AddObject(PoolObjectParams parameters) que Instancie/inicialice un objeto en la posición descrita.

- e. El PoolObject debe tener un evento OnPoolObjectFinished. El PoolManager debe estar suscrito a ese evento, y debe responder a éste desactivando y guardando al objeto.
 - f. En este sistema se espera que no se instancien más objetos de los necesarios, pero, que se puedan seguir instanciando según la demanda.
4. Desarrollar un juego que use los sistemas 1, 2 y 3 con las siguientes reglas:
- a. Un cubo se mueve con velocidad constante hacia la derecha/izquierda.
 - b. Cuando el cubo se choca contra una pared cambia de dirección.
 - c. El nivel es pequeño, dos paredes altas marcan los límites.
 - d. Cada vez que el jugador hace tap el cubo salta.
 - e. De forma aleatoria van saliendo cubos verdes y rojos, si el jugador se choca contra uno verde gana un punto, si choca contra un rojo pierde una vida.
 - f. El controlador del jugador que desarrollen debe implementar la interfaz del punto 2.
 - g. El UI muestra el puntaje y las vidas restantes.
 - h. Al perder todas las vidas debe mostrar una pantalla de Game Over con un boton de restart.
 - i. Los cubos rojos y verdes idealmente usan el sistema de pool que se desarrollo, y estos desaparecen del gameplay después de 5 segundos o al chocar contra el jugador.
 - j. El input debe usar el sistema de input del punto 1.