

GYMKANA PUEBLOS

DESCRIPCIÓN

INTRODUCCIÓN

Gymkana Pueblos es una aplicación gamificada, que permite al usuario aprender sobre la localidad donde se lleva a cabo la ruta, mediante el uso de textos informativos sobre el lugar, preguntas que requieren la participación del usuario tanto con los textos dentro del juego como con el lugar donde trascurren los acertijos, y, por último, mapas que nos permiten localizar los distintos puntos de ruta.

FLUJO

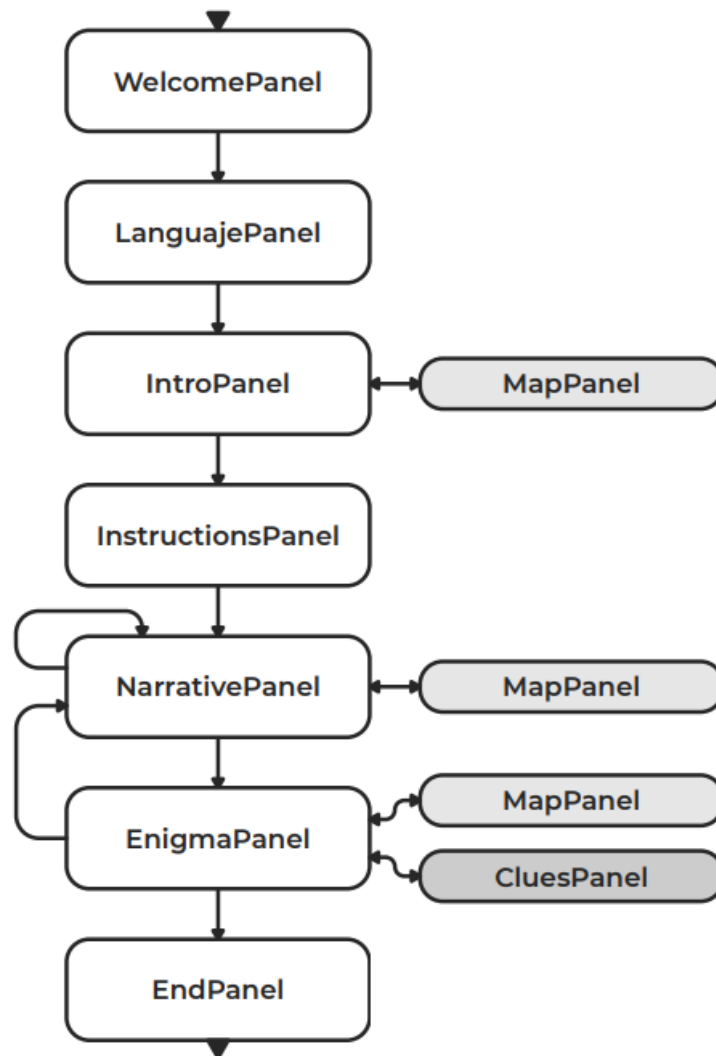


Ilustración 1. Flujo de pantalla de Álvaro

Cada escena representa una ruta distinta, pero todas siguen el mismo flujo (Ilustración 1) donde dependiendo de la extensión de la gymkana tendrá un número determinado de paneles narrativos (textos con información de la ruta) y paneles enigma (preguntas sobre la ruta).

TEXTOS

Cada escena tiene configurado un enlace de donde puede sacar los textos en formato JSON. Estos se bajan cada vez que la aplicación se inicia y se usan para modificar los assets que contienen la información de los textos. Esto permite que en caso de que no sea posible la descarga de los textos de los servidores del cliente se pueda continuar con la aplicación, haciendo uso de los textos originales de la escena.

MAPAS

La aplicación hace uso del plugin [Open Street Map Client](#) para la integración de los mapas.

Se han tenido que realizar algunas modificaciones respecto a la configuración por defecto del prefab map, debido a conflictos con la plataforma objetivo WebGL:

- Los enlaces a los servidores han tenido que modificarse sustituyendo http por https para evitar un conflicto de carga de recursos inseguros en un servidor seguro.
- La cache ha tenido que ser deshabilitada.
- El script que permite los gestos de móvil tenía limitado la sensibilidad mínima de desplazamiento, no permitiendo limitar los gestos a solo zoom y rotación, por lo que se ha modificado para que permita dejar a 0 la sensibilidad del desplazamiento de coordenadas.

PROGRAMACIÓN

IDIOMA

El idioma lo representamos mediante un enumerado llamado **Language**. La información del idioma en uso se guarda en el manager principal y a esta se accede cada vez que se necesita representar un texto.

Los textos representados en cada idioma vienen de la mano de la clase **LanguageText**. La cual representa un *string* correspondiente al *texto* y un *enumerado* que representa el *idioma* en el que esta.

La representación de textos viene dividida en que tipo de texto es:

- Interfaz: Tenemos una componente **LanguageSelector** la cual nos permite guardar una lista de **LanguageText** y que al activarse su *gameObject* elige el texto correcto al consultar el idioma de la sesión.
- Gymkana: Las clases que las representan guardan los textos en listas de **LanguageText** y dentro de los assets que las contienen es donde se pueden obtener el texto en un idioma indicado.

CARTAS

La información de los textos del juego la guardamos en cartas. Tenemos una clase *Card* la cual posee la información básica que toda carta necesita: título y texto.

El resto de las cartas heredan de esta clase y añaden mas variables en caso de necesitar más información:

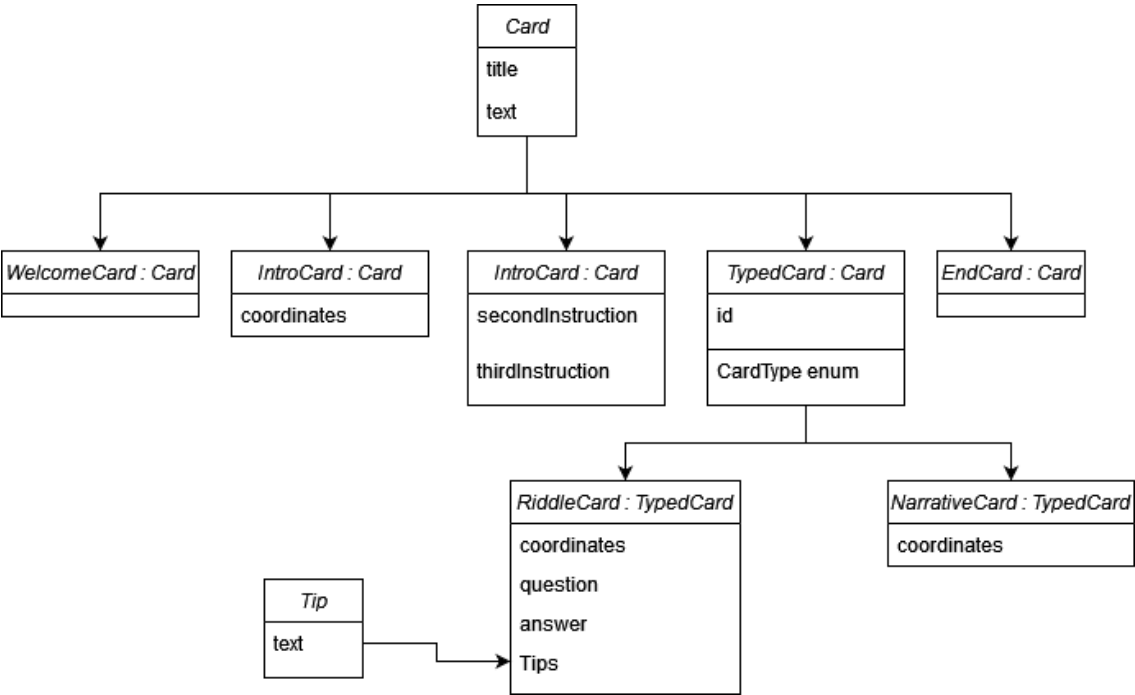


Ilustración 2. Diagrama de clases de cartas

SERIALIZACIÓN

La librería que se usa en la *des/serialización* es la de [Json.net](https://www.newtonsoft.com/json/), debido a que esta permite la *herencia* al contrario de por defecto de *Unity*.

En la *deserialización*, hacemos uso de esta librería en la clase **DataDeserializer**, donde tenemos un método para *deserializar* archivos y otro para hacerlo con *strings*.

La serialización se usa en la *Tool* **SerializerTool**, donde al pasarle distintos *assets* que contienen los datos de una *gymkana*, esta te genera un archivo **JSON**, el cual una vez lo tengamos alocado en un servidor, podremos descargarlo cada vez que ejecutemos la *gymkana*.

Por tanto, el flujo de los textos queda así:

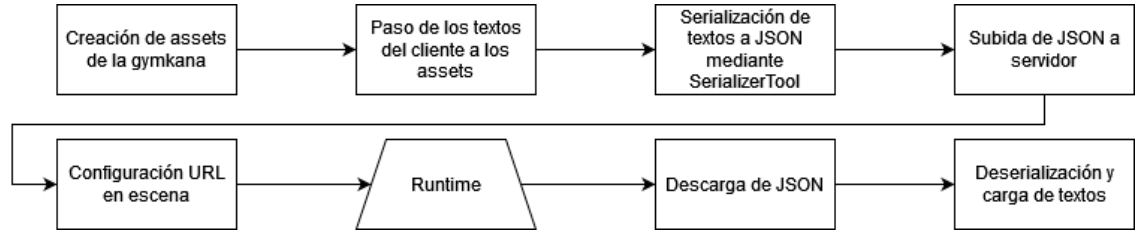


Ilustración 3. Flujo de textos

ASSETS DE GYMкана

Los *assets* que representan a la gymkana los creamos mediante *Scriptable Objects*. La estructura de clases que seguimos es la misma que en *cartas*, ya que estos *scriptables* contienen su carta homónima y métodos para acceder a los textos de dichas cartas según el idioma de la sesión. Además, contienen recursos correspondientes a la zona de la gymkana a la que representan, tales como *Sprites*.

MANAGERS

- **Session:** Se encarga de tener un idioma y coordenadas actuales, para que los *scriptables* y la UI puedan acceder a ellos y mostrar sus datos de forma correcta.
- **GymkanaManager:** Es el encargado de iterar entre los distintos paneles de la interfaz dependiendo del tipo de carta correspondiente en ese momento y aportar la información a la interfaz.
- **DataLoader:** Objeto en el que agrupamos las componentes encargadas de la carga de los textos desde servidor. **DataDownloader** se encarga de hacer una petición sobre el archivo que contiene los textos deseados (aquí es donde debemos cambiar al URL dependiendo del archivo a cargar) y los guarda, **DataDeserializer** (antes mencionado) y **TextLoader**, el cual recorre los distintos *assets* encargados de guardar la información de la gymkana y los sobrescribe con la información descargada y deserializada.

MENÚS

Lista de scripts que maneja sus respectivos paneles en la UI:

1. **WelcomePanel:** carga textos e imagen y en caso de no tener *Sprite* deshabilita la imagen.
2. **LanguagePanel:** se encarga de guardar el idioma de la sesión.
3. **IntroPanel:** Carga la introducción de la gymkana y permite acceder al mapa.
4. **MapPanel:** activa el objeto mapa, pasandole la información necesaria de la localización de esa parte de la ruta además de controlar la UI para un visionado correcto del mapa.
5. **InstructionPanel:** muestra las instrucciones de esa gymkana, cargando los tres textos de su *asset*.
6. **NarrativePanel:** Muestra textos y permite acceder al mapa de nuevo.
7. **RiddlePanel:** Carga los textos correspondientes, permite acceso al mapa y se comunica con los paneles de pistas y teclado en pantalla.
8. **CluesPanel:** Muestra las pistas correspondientes (3 pistas posibles).
9. **KeyboardPanel:** dentro de Riddle Panel en la UI. Se encarga de manejar las respuestas que introducimos por pantalla y se comunica con el panel de enigmas para saber si se ha acertado una respuesta y activa un panel u otro dependiendo de eso.
10. **EndPanel:** muestra la información correspondiente al final de la gymkana.