

JERA Guía de desarrollo.

1. **JERA**
   1. **Sobre el lenguaje:**

JERA es el nombre que usaremos para el lenguaje que vamos a desarrollar, siendo sus siglas un acrónimo recursivo, JERA Es Realidad Aumentada. Este lenguaje basado en XML, será capaz de producir un archivo de salida en el formato del explorador web para realidad aumentada Wikitude.[1]

* 1. **Sobre el interpretador:**

Iniciaremos el proyecto desarrollando un interpretador usando la liberia de Python PLY, para la generación de un interpretador de nuestro lenguaje JERA.

Las estructuras XML a desarrollar, se realizaran siguiendo el patrón de POPCODE[2] para simplificar el uso del lenguaje.

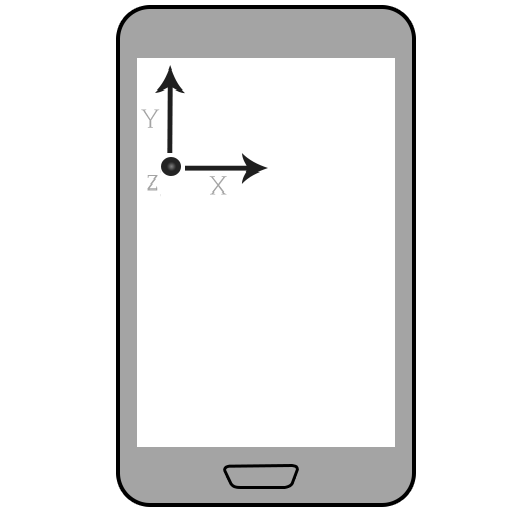
Para generar los archivos HTML, CSS, y JS respectivos a la descripción realizada, se usará la siguiente sintaxis:

./JERA <nombre\_archivo>.xml

*NOTA:* en el directorio actual de la llamada deben encontrarse las siguientes carpetas:

* Assets
  + Images
  + Stylesheets
  + JS
  + Models

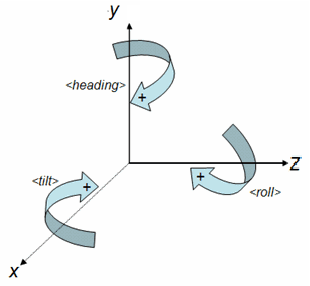
Encontrándose en Images, las imágenes deseadas para la aplicación de salida.

* + 1. **Definición de planos:**

Durante el desarrollo de esta guía, existen algunos elementos definidos con el atributo “position” el cual describirá la posición en la pantalla en la cual un elemento de JERA se encontrará.

Dicha posición podrá ser en tres dimensiones o en dos dimensiones, dependiendo del elemento al cual se haga referencia.

Para el caso de elementos que utilizan dos dimensiones se muestra mediante la representación “(X Y)” para la cual, el eje X crece hacia la derecha de la pantalla del dispositivo y la Y crece hacia la parte inferior de la misma. Siendo para el caso de tres dimensiones un caso similar, se muestra mediante la representación “(X Y Z)” siendo el eje Z creciente hacia el interior de la pantalla.



Para el caso de las Rotaciones, se presentan tres elementos tilt, roll y heading, los cuales respectivamente representan las rotaciones con respecto a los diferentes ejes, tilt el elemento gira con respecto a X, roll gira con respecto a Y y por último heading gira con respecto a Z.

Dependerá del objeto al cual se desee modificar dichos atributos las unidades del mismo, puesto que se manejan distintos casos, para elementos de Realidad Aumentada, se manejan unidades en SDUs, cuya referencia se encuentran en el siguiente enlace: <http://www.wikitude.com/external/doc/alr/module_AR.html>

Para elementos estáticos de la pantalla se manejarán unidades en pixeles, y para el caso de las rotaciones las unidades se manejarán en grados.

NOTA: el tamaño máximo de la posición en X,Y en pixeles dependerá del dispositivo en el cuál será ejecutada, por ejemplo, en un Samsung Galaxy S3 la posición en X,Y máxima visible es 720,1280.

* + 1. **Sobre el seguimiento de imágenes:**

Para generar los archivos de seguimiento de imágenes, es necesario de la herramienta web “vuforia”[5], la cual luego de cumplir con los requerimientos de la guía de “Wikitude” que se encuentra en este enlace: [GUÍA VUFORIA](http://forum.wikitude.com/documentation?p_p_id=5_WAR_knowledgebaseportlet&p_p_lifecycle=0&p_p_state=maximized&p_p_mode=view&_5_WAR_knowledgebaseportlet_mvcPath=%2Fsearch%2Fview_article.jsp&_5_WAR_knowledgebaseportlet_resourcePrimKey=35504)

Se obtendrá el archivo “.zip” utilizado para el reconocimiento de imágenes en JERA.

1. **Descripción del lenguaje:**

A continuación describimos los elementos del lenguaje que son implementados, dichos elementos son etiquetas de XML[3], cuyo nombre identifica el tipo de objeto al cual se quiere hacer referencia en la descripción de la aplicación a generar.

* 1. **<Scene>**

Los archivos XML de entrada de JERA deben ser de la siguiente forma:

<scene>

...

</scene>

Donde la etiqueta <scene> representará todo el programa que se desea realizar en realidad aumentaba basado en las acciones o parámetros que próximamente serán definidos.

* 1. **<manifest>**

Esta etiqueta se usa para representar la información relevante para la aplicación a generar, es decir, en dicha etiqueta se expresará el título de la aplicación, una breve descripción de la misma con la finalidad de representar datos descriptivos de lo que se desea generar y por último los datos de rastreo de imagen.

<scene>

<manifest

title="Escena de prueba"

description="Está es una representación de una escena"

tracker="Imágenes\_scena.zip"

/>

</scene>

**2.2.1 Atributos:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Tipo | Por defecto | Detalles |
| title | String |  | Título de la aplicación. |
| description | String |  | Descripción de la aplicación. |
| tracker | path | REQUERIDO | Ruta de datos de seguimiento. |

* 1. **<object>**

La etiqueta *object*, se utiliza para cargar o mostrar modelos en la escena.

<scene>

<manifest

...

/>

<object id=”Objeto1” source=”models/primitives /modelo\_objeto1.wt3” >

</object>

</scene>

* + 1. **Atributos:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Tipo | Por defecto | Detalles |
| id | String |  | Es el identificador para el elemento. |
| visible | boolean | true | false, el elemento no se muestra en pantalla. |
| position | x y z | 0 0 0 | Posición del elemento en la pantalla. |
| scale | x y z | 1 1 1 | Escala del elemento en pantalla. |
| rotation | tilt heading roll | 0 0 0 | Rotación con respecto a cada eje. |
| source | path | REQUERIDO | Fuente del archivo |
| target | path | REQUERIDO | Fuente de la imagen a visualizar. |

* 1. **<button>**

La etiqueta *button*, se utiliza para generar un elemento de interfaz “Botón” al cual  
se le podrán asignar acciones dentro del programa. Los botones de interfaz para JERA, son basados en bootstrap twitter[4] debido a su facilidad de integración en aplicaciones WEB y la estética que estos presentan.

<scene>

<manifest

...

/>

<button id=”Boton1” type=“warning” position=”0 0 0”>

…ACCIONES…

</button>

</scene>

* + 1. **Atributos:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Tipo | Por defecto | Detalles |
| id | String |  | Es el identificador para el elemento. |
| visible | boolean | true | false, el elemento no se muestra en pantalla. |
| position | x y z | 0 0 0 | Posición del elemento en la pantalla. |
| type | default, primary, info, success, danger, warning, inverse | default | Modelo del botón. |
| text | String | none | Nombre en pantalla del botón. |
| size | Integer | 1 | Tamaño extra del botón en píxeles. |

* + 1. **Tipo de Botones:**

|  |  |
| --- | --- |
| C:\Users\Alfred\Desktop\default.png | Botón de Default, color gris con gradiente. |
| C:\Users\Alfred\Desktop\primary.png | Botón de Primary, color azul. |
| C:\Users\Alfred\Desktop\Info.png | Botón de Infor, color aquamarin. |
| C:\Users\Alfred\Desktop\success.png | Botón de Succes, color verde. |
| C:\Users\Alfred\Desktop\warning.png | Botón de Warning, color naranja. |
| C:\Users\Alfred\Desktop\danger.png | Botón de Danger, color rojo. |
| C:\Users\Alfred\Desktop\inverse.png | Botón de Inverse, color negro. |

* + 1. **Atributos:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Tipo | Por defecto | Detalles |
| id | String |  | Identificador del Texto. |
| size | Integer | 20 | Tamaño de la fuente. |
| font | String | Helvetica | Tipo de fuente para el texto a introducir, dicho tipo de fuente debe estar instalada en el servidor donde se ejecutará la aplicación. |
| position | x y | 0 0 | Posición del elemento en la pantalla. |

* 1. **<Audio>**

La etiqueta *audio* se utiliza para cargar un archivo de audio en la aplicación.

<audio filename=” nombreAUDIO.wav” playonload=”True” />

* + 1. **Atributos:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Tipo | Por defecto | Detalles |
| filename | String | REQUERIDO | Ruta del audio a cargar |
| playonload | Boolean | True | Se especifica si se desea reproducir el audio al cargar la aplicación. |

* 1. **<Event>**

La etiqueta *Event* agrupa acciones las cuales son ejecutadas unas vez que se ejecuta algún tipo de accionador para el conjunto de acciones.

La lista de accionadores es la siguiente:

* Onclick – Usado para eventos en botones

La lista de acciones disponibles para los eventos es la siguiente:

* Set
* Transicition
* Play

<event id=”EventoID” type=”onclick”>

...

<set … />

</event>

* + 1. **Atributos:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Tipo | Por defecto | Detalles |
| id | String | REQUERIDO | El id del evento. |
| type | onlcick | REQUERIDO | Tipo de accionador del evento. |
| times | integer | 0 | Tiempo de ejecución de las acciones. |

* + 1. **<Set>**

La etiqueta *Set* es usada para cambiar el valor de un atributo booleano que sirve como parámetro a otra etiqueta.

<scene>

<manifest

...

<set tag=”Objeto1” what=”visible” to=”false”/>

</scene>

* + - 1. **Atributos:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Tipo | Por defecto | Detalles |
| tag | String | REQUERIDO | El id de la etiqueta a la cual se cambiara el parámetro. |
| what | String | REQUERIDO | Nombre del atributo que se desea cambiar. |
| to | true o false | REQUERIDO | Valor del parámetro luego del cambio. |

* + 1. **<Transition>**

La etiqueta *transition* es utilizada para modificar las propiedades de un *objeto* en el tiempo.

<scene>

<manifest

...

/>

<object id=”Objeto1” source=”models/primitives /modelo\_objeto1.wt3” position=”1 1 1” >

</object>

<transition id=”Trans1” object=”Objeto1” what=”position” end=”2 1 3” length=”1000” />

</scene>

* + - 1. **Atributos:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Tipo | Por defecto | Detalles |
| id | String |  | Identificador del elemento |
| visible | boolean | true | Si es *false*, el elemento no se muestra en pantalla. |
| object | Objeto id |  | Identificador del objeto al cual se le realizaran modificaciones en sus propiedades. |
| what | *position, scale, rotation* |  | Propiedad a cambiar. |
| start | x y z |  | El valor inicial de la transición. No obligatorio |
| end | x y z |  | El valor final de la transición. |
| length | Integer | 1000 | Tiempo de la transición. En milisegundos |
| times | Integer | 1 | Número de repeticiones de la transición.  -1: repeticiones indefinidas |

* + 1. **<Play>**

La etiqueta *play* es utilizada para reproducir un elemento de audio.

<scene>

<manifest

...

<play filename=”nombreAUDIO.wav” />

</scene>

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Tipo | Por defecto | Detalles |
| filename | String |  | Identificador del elemento |
| times | Integer | 1 | Cantidad de veces a repetir el audio, si se desea repetir una cantidad infinita de veces colocar -1 |

**Referencias:**

[1] Wikitude. Sitio Web: <http://www.wikitude.com/>

[2] POPCODE. Sitio Web: <http://popcode.info/>

[3] Wikipedia XML. Sitio Web: <http://en.wikipedia.org/wiki/XML>

[4] Bootstrap Twitter. Sitio Web: <http://twitter.github.io/bootstrap/>

[5] Vuforia Developers. Sitio Web: <https://developer.vuforia.com/>