XML JDE specification

This document is a guide to understand the XML file format to create JDE games.
Main project is located in Github - JDE. This guide is for the version 0.02.1. XML format may change in future versions, when adding new elements.

Index

This document has been written by HarZe, for learning purposes, last updated at July 17, 2014.
License: Creative Commons BY-NC-SA 4.0

General format

Every XML file from this engine must be a tag *<jde-content>* with all the data inside of it.
Other XML files can be imported to split the used content in several XML files and have a better organization, to import XML files user the tag *<import>*. This could be a simple example of a game where only exists an enemy:

```
<jde-content>

   <import file="res/test_level/sprites.xml" />
   <import file="res/test_level/movements.xml" />
   <import file="res/test_level/bullets.xml" />
   <import file="res/test_level/hordes.xml" />
   <import file="res/test_level/enemies.xml" />

   <game name=”test_game”>
      <spawn
         time="1000"
         x="200"
         y="200" >

         <enemy-ref ref="test_enemy" />
      </spawn>
   </game>

</jde-content>
```

Please note that there are several imported XML files where the sprites, enemies, bullets… are defined, this one file alone is NOT a game.

About the *ref* attribute, this is the way you can reference previously declared elements, you can find an example in the next section.

Vertex

This tag defines a simple pair of values: coordinates of X-axis and Y-axis. For example:

<vertex x="10" y="10" />

Using this simple tag, you will learn how to reference previously declared elements, such as vertices. The previous example is not declared for later use, it can only be used inside of another element that has a vertex (such as *Movement*). If you want to declare an element to reutilize it later, you must name it just adding the attribute *name*:

<vertex name=”sample_vertex” x="10" y="10" />

This can be really useful to avoid repeating a lot of elements and organize the XML better. To use this vertex later, you must reference it using the same tag name adding *-ref*, just like this:

<vertex-ref ref=”sample” />

Once the vertex with name is declared, typing this reference will be equivalent to type a new *Vertex* with the same attributes. All attributes are mandatory.

IMPORTANT: this reference system can be used ONLY in certain elements, read this document to check which tags are referenceable and which are not.

Spritesheet

Loads and image (spritesheet) from image source. Must be referenced later by the sprites which image is a part of the spritesheet. This tag is referenceable, but not in the same way a *Vertex* is. So, you can not use <*spritesheet-ref*>, read *Sprite* to know how to reference it.

<spritesheet name=”spritesheet_name” file=”res/spritesheet.png” />

Both attributes are mandatory. There can not be a *Spritesheet* with no name or file.

Sprite

Sprites are a part of a *Spritesheet* that must be attached or referenced to any element that is drawable. Sprites can have a name to be referenced (using *<sprite-ref ref="name" />*) or not, in the second case, is supposed to be directly attached to an element, and therefore, used once. Example:

```
<sprite name="sample_sprite" sheet="sample_sheet"
   x="96" y="128" w="32" h="32" scaling="0.5" />
```

This *Sprite* is taking a part of the *sample_sheet*, which can be for example 256x256. The part taken is defined by *x,y,w* and *h*; being (*x, y*) the upper left corner coordinates, *w* the width and *h* the height; which is a rectangular region of the sheet. The *scaling* attribute allows to modify the size of the *Sprite*, imagine that you want to use the same *sheet* for a *bullet* and you want to make different size versions of the bullet, in this case the *scaling="0.5"* is converting 32x32 into 16x16, making a smaller bullet without using a different *spritesheet*.

All attributes are mandatory, except *scaling*, which default value is 1 if is not defined, and *name*.

Movement

It defines the movement of an entity: position and speed, direction. Basically, is vector with movement properties. It is referenciable.

```
<movement name="sample_move" speed="10" angle="30"
        <vertex x="10" y="10" />
</movement>
```

The *angle* defines the direction, please note that angle is measured as degrees, 0º will move down, 90º will move to the right, 180º up and 270º left. *Speed* is measured as pixels per second, when VGA, that means that if resolution is 1280x960, the speed value 1 will actually be 2 pixels per second. This way the speed becomes independent of the resolution of the game. The *vertex* declared inside is the initial position, which can be declared (previous example) or referenced:

```
<movement name="sample_move" speed="10" angle="30"
        <vertex-ref ref="sample_vertex" />
</movement>
```

There are not mandatory attributes, default values are 0, and default *vertex* is (0,0)

Bullet

A bullet must have: *sprite*, *hitbox* (not implemented yet) and *movement*. It is referenciable.

```
<bullet name="sample_bullet" >
    <hitbox-ref ref="hitbox_bullet" />
    <sprite-ref ref="sprite_bullet" />
    <movement-ref ref="move_bullet" />
</bullet>
```

Horde

Generates a set of bullets. It must contain a template bullet, which will be used to generate the bullets of the horde. All the attributes are optional, default values are 0, except for bullets (default value: 1) and repeat (default value: no). It is referenciable.

```
<horde name="horde_bullet_clockwise"
    angleend="360" anglestart="0"
    bullets="24"
    repeat="yes" interval="2000"
    timestart="100" timeend="1100" >
    <bullet-ref ref="bullet_bullet" />
</horde>
```

Enemy

It defines a simple enemy. Must have: sprite, hitbox (not implemented yet), movement and horde. Mandatory attributes: *health*. It is referenciable.

```
<enemy name="enemy_sample" health="100" >
    <hitbox-ref ref="enemy_hitbox" />
    <sprite-ref ref="enemy_sprite" />
    <movement-ref ref="enemy_move" />
    <horde-ref ref="enemy_horde" />
</enemy>
```

Spawn

This tag is not referenceable, can be only used inside *<game>* once. It spawns an entity (for now, it only spawns enemies) in the position given by (*x, y*) at the *time* in milliseconds from the start.

```
<spawn time="2000" x="200" y="200" >
    <enemy-ref ref="enemy_sample" />
</spawn>
```

Game

This tag should be used only once, is not referenceable and should be the last one. It contains only spawns. There is an example in the section *General format*.