Cranberry

File format for sprites

Version 1.0 on July 8, 2017

Table of contents

Concept

The sprite class in cranberry should be able to play animations, bound to a certain trigger key. The JSON-based file format for the sprite class gives the possibility to define a spritesheet to use and an arbitrary number of movements which can be triggered through a key press. The movements are furthermore highly customizable.

There are two modes supported: default and tile. The tile mode allows to move a sprite a certain amount of units and other key input is blocked during that movement. Idle frames are also supported, in case someone implements a "facing" feature, where a short key press does not move the sprite but just faces it in a specific direction. Last but not least, the frames attribute is designed in a way in which you can organize the spritesheet *however* you want, since the *rect* member lets you choose any values. Lastly, the *duration* member simply tells cranberry for how long this frame should be shown on the surface.

File format specification

Header

Version attribute

The required version attribute specifies the version of the file format. Cranberry is backwards-compatible, that means that old versions are not simply obsolete but can still be used as you desire. It can be any integer number.

Sheet attribute

The required sheet attribute specifies the image file to use. While this should be a *Qt resource path* (starting with ":/"), cranberry also supports relative paths to the executable (e.g. "/assets/sheets/foo.png").

Movements attribute

Name attribute

The required name attribute specifies the name of the movement, which should be unique. You will need it to bind the *keyboard key/mouse button* (which will trigger the movement) programmatically.

Mode attribute

The optional mode attribute changes the way the sprite moves/animates. There are two modes as of today: default and tile. Tile implies that the sprite should be moved a specific amount of units and any other keys are blocked during that period. The mode attribute requires the specification of the *advance* attribute. By default, this attribute is *default*.

Advance attribute

The required advanceX and advanceY attributes specifies the amount of units the sprite should move. By default, this attribute is *0*. Can be negative values, too.

Idle attribute

The optional idle attribute specifies a frame that contains no movement. If a programmer happens to implement a "facing" feature, this attribute should be used. This JSON object contains 4 members, *x*, *y*, *w* and *h* representing x coordinate, y coordinate, width and height respectively.

Frames attribute

Rect attribute

The required rect attribute specifies the source rectangle within the spritesheet that should be used as frame. This JSON object contains 4 members, *x*, *y*, *w* and *h* representing x

coordinate, y coordinate, width and height respectively.

Duration attribute

The required duration attribute specifies for how long the frame should be shown on the surface. The time should be specified in milliseconds. Note that you should not use values too little (at constant 60 frames per second, one frame lasts 16,6 milliseconds). It can be an integer number or a floating-point number.

Example

This example shows a tile-based movement/animation (32 units).

```json
{
  "version" : 1,
  "sheet" : ":/spritesheets/sprite.png",
  "movements" : [
    {
      "name" : "walk",
      "mode" : "tile",
      "advanceX" : 32,
      "advanceY" : 0,
      "idle" : {
        "x" : 0,
        "y" : 0,
        "w" : 256,
        "h" : 256
      },
      "frames" : [
        {
          "duration" : 20,
          "rect" : {
            "x" : 256,
            "y" : 0,
            "w" : 256,
            "h" : 256
          }
        }
      ]
    }
  ]
}
```