Qube2D Animation File Format 1.0

Technical Reference Manual

Version 1

September 2016

Nicolas Kogler
Vienna, Austria

September 2016

Copyrights and Trademarks

Contents

1. Introduction

This manual describes the Qube2D-Animation file format, QDA for short. QDA was designed to store multiple frame steps, similar to the Graphics Interchange Format (GIF), which are stored in the lossless PNG format. It does not matter which pixel format they describe – Qube2D will read everything except indexed pixel formats. QDA also offers a precise timing system. On the one hand it is possible to specify the display duration of each frame in milliseconds, on the other hand one can also specify the wait interval between the current and the next frame. Most of the cases, one will only need the display duration time.

The fact that this file format is work-in-progress has the following implications:

I. The specification in this manual is not fully complete. The QDA file format is most likely to be changed in the future with more features being added.

II. The inventor of this file format is trying to keep all specifications backwards-compatible, but it may not be possible due to grave changes that lead to complete restructuring of the underlying program code.

III. This manual may contains errors or missing side-information.

2. File Structure

2.1 Header

The file header consists of ten so-called “magic numbers”, the major and minor version number (each one byte) and the frame-count that occupies four bytes in total. The magic numbers define an ASCII-string with the content *“QUBE2DANIM”* which will stay the same across all QDA versions. The major version will change after every grave change to the format and to the underlying decoding algorithm, whereas a minor version change only applies small fixes to the file format. An example for a valid file header could be the following byte sequence:

51 55 42 45 32 44 41 4E 49 4D 01 00 08 00 00 00

Half-words and double-words are stored in little-endian format and are correctly read by the decoding algorithm, regardless of the underlying system's endianess.

2.2 Frame Table

The frame-table is directly located after the file header and it's size is the frame-count multiplied by eight. One entry defines the size of the image, the display duration in milliseconds and the duration in milliseconds in which nothing is displayed. Decoding fails if …

I. the frame width or frame height is zero,

II. the image is not a valid PNG image or

III. the display duration is zero

… and a respective error message will be output.

The frame-table starts with two words that represent the frame-width and frame-height respectively.

2.3 Image Data

The image must be a Portable Network Graphics (PNG) image and shall not define a indexed pixel format, i.e. should not have a 4-bit or 8-bit palette. This is due to Qube2D being not able to decode indexed images yet, as it requires extra effort to convert or directly render such images in OpenGL.

The size of the image is limited to the maximum size of OpenGL textures on your graphics card. For ninety-nine percent of the currently distributed hardware, this size is 2048x2048 to 4096x4096. Older hardware may still have the maximum at 1024x1024.