# BKSB File Format Specification

Reversed by Connor Virostek

## Contents

# 1. Introduction

BKSB is a binary, little-endian 3D model file format used exclusively by the iOS version of Temple Run, developed by Imangi Studios. Sections 2 – 4 describe the file structure.

# 2. Header Format

The header contains information about the model's version number, number of frames, and the amount of vertex/face data for each frame.

## 2.1 Version Number

The first byte of the file is always 0x76, or the character 'v'. Next is a 4-byte integer, which may be either 3 or 4. The main difference between these two versions is that v4 has UV lightmap coordinates for each vertex, while v3 does not. In Temple Run, nearly all models are v4 except for the coins, player models, and enemy models.

## 2.2 Frame Count

The number of animation frames is stored in a 4-byte integer. Note that BKSB uses vertex animation, so each frame of animation is stored as a separate mesh.

## 2.3 Frame Info

For each frame, there are 4 integers containing basic data about the its vertices and faces, as described in the following sections.

### 2.3.1 Vertex Count

4-byte integer containing the number of vertices.

### 2.3.2 Face Count

4-byte integer containing the number of vertex indices that make up the face data. Since each face is made up of 3 vertices, this is the face count multiplied by 3.

### 2.3.3 Vertex Data Start Index

4-byte integer containing the starting index for the frame's vertex data. For example, an index of 100 means that this frame's vertex data starts at the $100^{th}$ vertex in the total pool of vertices (as described in section 3).

### 2.3.4 Face Data Start Index

4-byte integer containing the starting index for the frame's face data. Like the face count, this integer is in terms of vertex indices rather than faces, so an index of 300 means that this frame's face data starts at the $300^{th}$ vertex index (or $100^{th}$ face) in the total pool of face data (as described in section 4).

# 3. Vertex Data

The vertex data is all stored in one big pool. It contains XYZ coordinates, UV coordinates, and lightmap UV coordinates if applicable.

## 3.1 Vertex Header

The vertex header is 47 bytes long for v3 models, and 66 bytes long for v4 models. I don't know what all of this data means, but it's nearly identical for any model of the same version so it isn't too important to understand it. The only data that changes is the second 4-byte integer, which contains the total number of vertices.

## 3.2 Vertices

Each vertex is made up of five 4-byte floats for v3 models and seven 4-byte floats for v4 models. The first 3 floats are the X, Y, and Z coordinates. The next 2 floats are the UV coordinates, and the last 2 floats (if applicable) are the lightmap UV coordinates. Note that UV coordinates of (0, 0) point to the upper left corner of the texture.

# 4. Face Data

Like the vertex data, the face data is all stored in one big pool.

## 4.1 Face Header

A 4-byte integer holds the number of vertex indices (the number of faces x 3). Three 4-byte integers follow, holding the numbers 0, 1, and 2. I am not sure what the purpose of these last three integers is, but they are the same for every BKSB.

## 4.2 Faces

A face is made up of 3 indices each point to a vertex. These indices are stored as 2-byte integers. Note that the indices start at 0.

(This concludes the file format specification.)

# 5. Textures

Temple Run uses the PVR format for its textures. PVRTexTool is a free program that converts to and from said format. First, encode the texture as PVRTC 2bpp RGBA or PVRTC 4bpp RGBA (depending on which the original texture uses), then save it as legacy (File > Save As Legacy...).