.PTE Format: (terrain texture pages)

- DWORD: Number of texture pages
- DWORD: Signature (0x23458977)
- DWORD: Version code (0x00000001, unused)
- DWORD: Number of mipmaps per texture page (0x00000003 usually)

- array of texture pages:

  - 256x256 raw RGB bitmap, plus the mipmaps

Notes:

Each texture page contains 16 tile textures 64x64. Tile texture is originally drawn at 62x62, with the pixels replicated at the borders.

.PVE Format: (terrain ground geometry)

- DWORD: Signature (0x00003333)
- DWORD: NumTilesPerX
- DWORD: NumTilesPerY
- FLOAT: MaxHeight
- FLOAT: MinHeight

- array of tiles:

  - WORD: IdTexture[2]
  - WORD: wFlags
  - BYTE: ucMaxHeight
  - BYTE: ucExtra

- array of vertex data

  - Height map: BYTE * ((NumTilesPerX * 2 + 1) * (NumTilesPerY * 2 + 1))
  - Color map: DWORD * ((NumTilesPerX * 2 + 1) * (NumTilesPerY * 2 + 1))

Notes:

Texture identifier is actually (texture page *16 + texture tile inside page). If the tile doesn't have texture, use -1.

To convert a compressed (byte-sized) height into "real" height:

```
    float fVertexHeight = ((MaxHeight - MinHeight) / 255.0) * Heightmap[i] + MinHeight;
```

Each tile is tessellated into 2x2 vertices to get extra geometry detail. For example:

```
*---*---*
|   |   |
*---*---*
|   |   |
*---*---*
```

The alpha channel of the vertex color contains the factor used to mix between the two textures present in a given tile.

Tile wFlags field:

```
    QUAD0_DIAGONAL2 = 0x0001
    QUAD1_DIAGONAL2 = 0x0002
    QUAD2_DIAGONAL2 = 0x0004
    QUAD3_DIAGONAL2 = 0x0008
    TEXTURE_ROTATE90 = 0x0010
    TEXTURE_ROTATE180 = 0x0020
    TEXTURE_ROTATE270 = 0x0040
    TEXTURE_MIRRORX = 0x0080
    TEXTURE_MIRRORY = 0x0100
    DYNAMIC_MIX = 0x0200
    TILE_NOTESSELLATE = 0x0400
    TILE_FLAT = 0x0800
    TILE_VISIBLE = 0x8000
```

.MOB Format (terrain objects)

- DWORD: Signature "MOB" + 0x1 (0x4D, 0x4F, 0x42, 0x01)
- DWORD: Number of objects

- array of objects:

  - DWORD: Flags: DWORD
  - BYTE[64]: PBA name
  - Vector3: Position X Y Z as floats
  - FLOAT: Orientation
  - FLOAT: Wind factor - for trees that swing in the wind.

=========

Notes:

=========

Object flags:

```
    NO_SHADOW  = 0x0001
    LARGE  = 0x0002
```

.MLG Format (ground logic info)

DWORD (NumTilesPerX x NumTilesPerY from PVE)

- array of tile logic information:

=======
Notes:
=======

```
MASK_FLAGS          = 0x07ffffff,
MASK_TERRAINTYPE    = 0xf8000000,
```

In programming terms, flags are usually single-bit values that can be combined together. Each flag represents whether a different property is active or not for that tile. For example, you could have a certain tile with the properties on tiles that contain fortress gates:

FLAG_FLAT | FLAG_WALL_OPEN | FLAG_DOOR | FLAG_NO_SIEGE | FLAG_BOCKED_BUILD

```
Flags:
    FLAG_NOPASSABLE = 0x0001
    FLAG_FLAT = 0x000010 // [0º10º)
    FLAG_TILT1 = 0x000020 // [10º30º)
    FLAG_TILT2 = 0x000040 // [30º50º)
    FLAG_TILT3 = 0x000080 // >= 50º
    FLAG_STRUCTURE = 0x000100
    FLAG_BRIDGE = 0x000200
    FLAG_MODIFIED = 0x000400
    FLAG_DOOR = 0x000800
    FLAG_NO_SIEGE = 0x001000
    FLAG_BUILDABLE = 0x002000
    FLAG_WALL_OPEN = 0x004000
    FLAG_WALL_BLOCKED = 0x008000
    FLAG_FORTRESS = 0x010000
    FLAG_BOCKED_BUILD = 0x020000
    FLAG_CALCULO_IA = 0x040000 (ignore)
    FLAG_FLAT_STRUCTURE = 0x080000
    FLAG_SCENERY_END = 0x100000
    FLAG_WALL_GROUND = 0x200000
    FLAG_BLOCK_WAR_MACHINES = 0x400000

Terrain types:
      NORMAL = 0
      SAND = 1
      DEEP_WATER = 2
      SHORE = 3
      GRASSFIELD = 4
      LIGHT_TREES = 5
      FOREST = 6
      WALL = 7
      BLOCKED = 8
```

SNOW = 9
UNUSED = 10
SCENERY_END = 11

.H2O (water and shore effects)

- DWORD: Signature (0x00000002)

[textures chunk]
- WORD: chunk id
- DWORD: chunk length
- DWORD: textures count

- array of textures:

  - WORD: chunk id
  - DWORD: chunk length
  - 256x256 raw RGBA bitmap

[waters chunk]
- DWORD: waters count

- array of waters:

  - WORD: chunk id
  - DWORD: chunk length
  - Block 1: (00 F0 AD BA 0D F0 AD BA 0D F0 AD BA 0D F0 AD BA
    0D F0 AD BA 0D F0 AD BA 0D F0 AD BA 0D F0 AD BA)
  - Block 2: (00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00)
  - DWORD: vertex count
  - Vertex array: vertex count * size of Vertex
  - DWORD: vertex index count
  - Vertex index array: WORD * vertex index count

=========

Notes:

=========

```
structure Vertex
{
  float x,y,z;
  char color[4];
  float u,v;
}
```

Blocks 1 & 2 are just bullshit data.

.PRA (grass areas)

- DWORD: Signature ".PRA" (0x2E, 0x50, 0x52, 0x41)
- DWORD: Version code (0x00000003)
- DWORD: grass PBA name length
-     BYTE[grass PBA name length] : PBA name
- DWORD: grass area count

- array of grass areas:

  - DWORD: grass patch count
  - DWORD: area X boundary
  - DWORD: area Y boundary
  - DWORD: area Z boundary
  - DWORD: area W boundary
  - BYTE: grass area name length
  -     BYTE[grass area name length] : name
  - grass patch array: grass patch count * size of GrassPatch

=========

Notes:

=========

```
GrassPatch
{
   DWORD: position X
   DWORD: position Y
   DWORD: high flags
   DWORD: low flags
   color: RGB (3 bytes)
}
```

The area (Z, W) boundaries are offsets from (X, Y).

.MUR (fortress walls)

To do ...