Project SMD

10. Reading and writing pcx files

why do you ask
2013. 9. 27. 15:20

This document is a course-style log of developing a StarCraft viewer . Since I am also in the position of learning, what is the point of calling th is a 'course';
This is a project to make a map editor, realizing that making a map edito r involves understanding the overall structure of the game.
http://blog.naver.com/whyask37
---------- ----------

Let's get started.

Today, we will read game/punit.pcx and print the zergling properly.

To do that, you first need to know the pcx file format.

Source: http://bespin.org/~qz/pc-gpe/pcx.txt (Appears to be an official docume nt)

The PCX header is said to be:

Colored By **Color Scripter** ™

```
One struct PCXHeader {
  2   uint8 magic;                    // 0x0A
  3   uint8 version;                  // 5 from star
  4   uint8 encoding;                  // always 1
  5   uint8 bpp;                      // 8 in star
  6   uint16 xMin, yMin, xMax, yMax;     // image size
  7   uint16 hDPI, vDPI;                // ignore
  8   uint8 colormap[16][3];             // can be ignored
  9   uint8 reserved1;                // can be ignored
 10   uint8 nPlanes;                  // 1 in star
 11   uint16 bytePerLine;               // Always even.
 12   uint16 paletteInfo;                // 1:Color/BW, 2:Grayscale.
 13   uint16 hScreenSize, vScreenSize;
```

ure:

Colored By **Color Scripter** ™

```
One struct  PCXPalette {
  2    uint8 magic;        // always 0x0C (12). When a non-value is found at (end of file - 769)
  3                        // This pcx file does not use a 256 color palette. Star writes.
  4     struct  RGBbyte {
  5        uint8 r, g, b;
  6    } colortable[256];
  7 }
```

Yes. Easy, right? Similarly, make it smooth. (You say the lectures are going too fas t? It's okay. It's not a class.

Let's make the library that reads pcx similar to the library that reads grp. The libr ary to read pcx is much shorter.
There is no complex structure like grp. I think it's really good.

The pixel format of pcx files is just a series of horizontal lines. Each horizontal line is compressed with RLE, and when RLE is decompressed, it expands to the size of bytePerLine. (pcx is a format used in 16-bit systems, so bytePerLine is an even nu mber for 16-bit allocation)
If you read the number of bytes before imgW during bytePerLine, you read one li ne.

RLE decryption
1. Byte 11 in binary????? If:
   Repeat the next byte ??????
2. Or print as is

is. It seems to be much simpler than grp. This seems to be the reason why it was used a lot in the DOS days because of its high speed.

Then let's implement it. easy. The pcx file is data related to Starcraft data, so I pu t it under the scdata namespace.

whyask37's blog

```
Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

   1. The origin of this software must not be misrepresented; you must not
   claim that you wrote the original software. If you use this software
   in a product, an acknowledgment in the product documentation would be
   appreciated but is not required.

   2. Altered source versions must be plainly marked as such, and must not be
   misrepresented as being the original software.

   3. This notice may not be removed or altered from any source
   distribution.
*/


// The following code implements ZSoft PCX File Input/Output Library
// PCX File format spec:  http://bespin.org/~qz/pc-gpe/pcx.txt

#ifndef  PCX_HEADER

#include   "gui/gui.h"
#include   "typedef.h"
#include   "chunk.h"
#include  <vector>

namespace  scdata {
    // PCX File format declaration
#include  <packon.h>
    namespace  pcx {
        struct  PCXHeader {
          uint8 magic;                  // 0x0A
          uint8 version;
          uint8 encoding;                   // 1:RLE.
          uint8 bpp;                    // 8 for sc
          uint16 xMin, yMin, xMax, yMax;     // Image resolution
```

whyask37's blog

Colored By **Color Scripter** ™

```
One /*
  2 MapCanvas. Simple, yet powerful 3rd-party starcraft map editor.
  3 Copyright (c) 2013 Trgk (whyask37@naver.com)
  4
  5 This software is provided 'as-is', without any express or implied
  6 warranty. In no event will the authors be held liable for any damages
  7 arising from the use of this software.
  8
  9 Permission is granted to anyone to use this software for any purpose,
 10 including commercial applications, and to alter it and redistribute it
 11 freely, subject to the following restrictions:
 12
 13    1. The origin of this software must not be misrepresented; you must not
 14    claim that you wrote the original software. If you use this software
 15    in a product, an acknowledgment in the product documentation would be
 16    appreciated but is not required.
 17
 18    2. Altered source versions must be plainly marked as such, and must not be
 19    misrepresented as being the original software.
 20
 21    3. This notice may not be removed or altered from any source
 22    distribution.
 23 */
 24
 25 #include   "pcx.h"
 26
 27 namespace  scdata {
 28     int  LoadPCX(Image& img,  const  Chunk* chk,  bool  getPaletteIndex) {
 29         pcx::PCXHeader header;
```

whyask37's blog

```
57          colortable[i].b = *(palette_cur + 2);
58           if (getPaletteIndex) colortable[i].a = i;
59           else  colortable[i].a = 0;
60          palette_cur += 3;
61        }
62
63        //3. get image
64        image image;
65        image.LoadBlank(imgw, imgh);
66
67        const  uint8* data_cur = chk->data + 128;
68        const  uint8* data_limit = chk->len - 767;
69        // decompress RLE. very simple logic
70        for (y = 0 ; y < imgh ; y++) {
71          rep = 0;
72           for (x = 0 ; x < header. bytePerLine ; x++) {  //bytePerLine is decoded for single RLE line.
73              //Since SetPixel will be ignored if x > imgw, code can be simplified like this
74
75              if (rep > 0) {
76                 rep--;
77                 image.SetPixel(x, y, colortable[col]);
78              }
79
80              else  {
81                 if (data_cur == data_limit)  return  -1;
```

whyask37's blog

main.cpp

Colored By **Color Scripter** ™

```
One #include   "gui/gui.h"
  2 #include   "grp.h"
  3 #include   "pcx.h"
  4 #include   "chunk.h"
  5
  6 Chunk* getChunk( const   char * fname) {
  7    Chunk* chk;
  8    FILE *fp;
  9    fp = fopen(fname,  "rb" );
 10     if (fp == NULL)  return  NULL;
 11     else  {
 12        int  fsize;
 13       fseek(fp, 0, SEEK_END);
 14       fsize = ftell(fp);
 15       rewind(fp);
 16       chk = GetBlankChunk( " " , fsize);
```

whyask37's blog

Colored By **Color Scripter** ™

```
One       memset(&pal, 0,  sizeof (pal));
   2      FILE *fp = fopen( "twilight.wpe" ,  "rb" );
   3      fread(&pal, 4, 256, fp);
   4      fclose(fp);
   5       for (i = 0 ; i < 256 ; i++) {
   6          std::swap(pal.color[i].r, pal.color[i].b);  //RGBAbytes are in BGRA order. so we swap B,R rea
   7      }
```

In addition , I put getGrpWidth(), getGrpHeight(), etc. in grp.cpp and grp.h to know the width anc grp file.

whyask37's blog

```
40        for (i = 0 ; i < 256 ; i++) {
41           std::swap(pal.color[i].r, pal.color[i].b);  //RGBAbytes are in BGRA order. so we swap B,R rea
42        }
43
44        //2. apply punit.pcx
45        chk = getChunk( "tunit.pcx" );
46        if (chk == NULL) {
```

whyask37's blog

Well... the zergling is dead. it's okay.

whyask37's blog