零基础学习ejoy2d——PPM贴图

gaccob

2004 年 2 月 5 日

1. PPM贴图格式

- PBM, portable bitmap, 单色图(1 bit).
- PGM, portable gray map, 灰度图.
- PPM, portable pixel map, 真彩图.

PBM / PGM / PPM图像的文件格式分为两部分: 文件头和数据部分. 一个典型PPM头的sample:

```
p6 # ppm format
1024 1024 # height & width
255 # depth, could be greater than 255
```

PPM的格式有p3和p6, p3表示用ascii码(文本)来表示数据, p6表示以字节码(二进制)来表示, 每一个像素按(r, g, b)的格式来存储.

```
P3
4 4
15
0  0  0  0  0  0  0  0  0 15  0 15
0  0  0  0 15  7  0  0  0  0  0  0
0  0  0  0  0  0  0 15  7  0  0  0
15 0 15  0  0  0  0  0  0  0  0  0
```

PGM与PPM类似, 格式有p2和p5, p2表示文本, p5表示二进制.

```
P2
18 7
15
0 0 0 0 0 0 0 0 0 0 0 0 0  0  0  0  0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0
```

```
0 3 0 0 0 0 0 7 0 0 0 0 0 11  0  0  0 0
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11  0 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11  0  0  0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0
0 0 0 0 0 0 0 0 0 0 0 0 0  0  0  0  0 0
```

对于PBM来说, 格式为p1, 但是文件头中没有最大颜色, 因为用0和1来表示就可以了.

```
P1
24 7
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0
0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0
0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 1 0
0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

参考文章: http://www.cppblog.com/windcsn/archive/2005/11/11/ppm.html

2. ejoy2d中的PPM源码阅读

ejoy2d中的PPM贴图的处理都在lib/ppm.h和lib/ppm.c中. ejoy2d中用PGM贴图来描述alpha通道, 用PPM贴图来描述rgb通道.

```
struct ppm {
    // 指PPM贴图的格式, p1-p6分别对应1-6
    int type;
    // 图像的深度, 一般有255(8位)和15(4位)
    int depth;
    // 步长, alpha是1, rgb是3, rgba就是4
    int step;
    // 长 & 宽
    int width;
    int height;
    // 图像数据
    uint8_t *buffer;
};

// 载入PPM文件头
static int
ppm_header(FILE *f, struct ppm *ppm) {
    ......
}

```

```
21      // 载入PPM文件数据
22      // 这里会根据type(p1, p2, ...)的不同, 做对应的解析并载入
23      // skip是为了有一个初始offset(适用于rgb ppm载入alpha的情况)
24      static int
25      ppm_data(struct ppm *ppm, FILE *f, int id, int skip) {
26          ......
27      }
28
29      // 载入PPM文件, 调用ppm_header()和ppm_data()完成.
30      // 如果是rgba, 需要从两个贴图文件一起载入(会做一致性校验)
31      static int
32      loadppm_from_file(FILE *rgb, FILE *alpha, struct ppm *ppm) {
33          ......
34      }
35
36      // 载入PPM文件的lua接口
37      // lua输入参数:
38      //  string ppm_name
39      // 输出lua结果:
40      //  string format(这里约定的格式有: RGBA8, RGB8, ALPHA8, RGBA4, RGB4,
            ALPHA4)
41      //  int width
42      //  int height
43      //  table buffer(ppm数据部分)
44      static int
45      loadppm(lua_State *L) {
46          ......
47      }
48
49      // 载入PPM文件到texture(纹理)的lua接口
50      // lua输入参数:
51      //  string ppm_name
52      // lua输出参数:
53      //  TODO: 这个到后面讲texture时再看
54      static int
55      loadtexture(lua_State *L) {
56          ......
57      }
58
59      // 根据format(上面描述的RGBA8等格式), 设置ppm数据: type, depth, step
60      static void
61      ppm_type(lua_State *L, const char * format, struct ppm *ppm)
            {
62          ......
63      }
64
65      // 从lua中读取数据, 保存rgb的PPM贴图(写文件), P6二进制格式.
66      static void
67      save_rgb(lua_State *L, int step, int depth) {
```

```
68          ......
69      }
70
71      // 从lua中读取数据, 保存alpha的PGM贴图(写文件), P5二进制格式.
72      static void
73      save_alpha(lua_State *L, int step, int depth, int offset) {
74          ......
75      }
76
77      // 保存PPM文件的lua接口, 调用save_rgb()和save_alpha()实现.
78      // lua输入参数:
79      //  string save_filename(保存的文件名)
80      //  string format(同上)
81      //  int width
82      //  int height
83      //  table buffer(ppm数据部分)
84      // 输出lua结果:
85      //
86      static int
87      saveppm(lua_State *L) {
88          ......
89      }
90
91      // lua的导出接口
92      int
93      ejoy2d_ppm(lua_State *L) {
94          luaL_Reg l[] = {
95              { "texture", loadtexture },
96              { "load", loadppm },
97              { "save", saveppm },
98              { NULL, NULL },
99          };
100         luaL_newlib(L,l);
101         return 1;
102     }
```