```
// NPC（学習プレイヤー）と対戦して遊ぶための，GUIのソースプログラム
//--------------------------------------------------------------------------------------------
/*
    ¥brief  sanmoku game interface
    ¥author Tsubasa Fukuyama
    ¥date   Mar.31 2008
*/
//--------------------------------------------------------------------------------------------
#define _CRT_SECURE_NO_WARNINGS
#define WIN32
#include "FL/Fl.H"
#include "FL/Fl_Window.H"
#include "FL/Fl_Box.H"
#include "FL/Fl_Button.H"
#include "FL/Fl_Menu_Bar.H"
#include "FL/Fl_Menu_Item.H"
#include "FL/Fl_ask.H"
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <Windows.h>

// ボタン配列
Fl_Button *b[9];

// 状態配列：0-無し，1-人間（O），2-学習プレイヤー（X）
int state[9] = {0, 0, 0, 0, 0, 0, 0, 0, 0};

//先攻，1-学習プレイヤー，0-人間
int player = 1;

// ゲームの状況：0-続行，1-終了
int gameflag = 0;

Fl_Window *window = new Fl_Window(170,190,"OX Game");

//--------------------------------------------------------------------------------------------
/*
    ¥brief 状態をファイルstate.txtに出力.
*/
void output(){
  int k;
  FILE *fp, *lock;

  // ロックファイルの生成
  lock = fopen("lock.m", "w");

  // 状態ベクトルを出力
  fp = fopen("state.txt","w");
  for(k=0;k<9;k++){
    fprintf(fp,"%d:",state[k]);
  }

  // ゲームの状況を出力
  fprintf(fp,"¥n%d",gameflag);

  fclose(fp);
  fclose(lock);

  // ロックファイルの削除
  remove("lock.m");
}
//--------------------------------------------------------------------------------------------

//--------------------------------------------------------------------------------------------
/*
    ¥brief 勝敗確認.
*/
int judge(int pos, int turn) {
  int i,j,k;
  char message[256] = "";

  i = (int)pos / 3;
  j = pos % 3;

  for(k=0; k<3; k++)
    if(state[i*3+k] != turn)
      break;

  if(k == 3) {
    for(k=0; k<3; k++)
      b[i*3+k]->color(turn==1?FL_RED:FL_BLUE);
      window->redraw();

    sprintf(message, "%s Win%s!!", turn==1?"You":"COM", turn==1?"":"s");
    fl_alert(message);
    return 1;
  }

  for(k=0; k<3; k++)
    if(state[j+k*3] != turn)
      break;

  if(k == 3) {
    for(k=0; k<3; k++)
      b[j+k*3]->color(turn==1?FL_RED:FL_BLUE);
    window->redraw();

    sprintf(message, "%s Win%s!!", turn==1?"You":"COM", turn==1?"":"s");
    fl_alert(message);
    return 1;
  }

  if(i==j) {
    for(k=0; k<3; k++)
```

```
106       if(state[4*k] != turn)
107         break;
108     if(k == 3) {
109       for(k=0; k<3; k++)
110         b[4*k]->color(turn==1?FL_RED:FL_BLUE);
111       window->redraw();
112 
113       sprintf(message, "%s Win%s!!", turn==1?"You":"COM", turn==1?"":"s");
114       fl_alert(message);
115       return 1;
116     }
117   }
118   if((2-i)==j) {
119     for(k=0; k<3; k++)
120       if(state[2*(k+1)] != turn)
121         break;
122     if(k == 3) {
123       for(k=0; k<3; k++)
124         b[2*(k+1)]->color(turn==1?FL_RED:FL_BLUE);
125       window->redraw();
126 
127       sprintf(message, "%s Win%s!!", turn==1?"You":"COM", turn==1?"":"s");
128       fl_alert(message);
129       return 1;
130     }
131   }
132   return 0;
133 }
134 //--------------------------------------------------------------------------------------------
135 
136 
137 //--------------------------------------------------------------------------------------------
138 /*
139     ¥brief 行動をファイルaction.txtから入力.
140 */
141 void action(void*) {
142   FILE *fp, *lock;
143   errno_t error;
144   errno_t error2;
145   char cdir[255];
146   GetCurrentDirectory(255, cdir);
147   // action.txtがあってもロックファイルがある間は書き込み中なので待つ
148   if((error=fopen_s(&fp,"action.txt","r")) != NULL || (error2=fopen_s(&lock, "lock.m", "r")) != NULL){
149     Fl::repeat_timeout(2.0,action);
150   }else{
151 
152     // 行動のマスに×を置き，状態を2とする
153     char str[10];
154     fgets(str,10,fp);
155     int num = atoi(str);
156     fclose(lock);
157 
158     b[num]->label("X");
159     state[num] = 2;
160     player = 0;
161     fclose(fp);
162     remove("action.txt");
163 
164     if(gameflag = judge((int)num, 2)){
165       output();
166       return;
167     }
168 
169   }
170 }
171 //--------------------------------------------------------------------------------------------
172 
173 //--------------------------------------------------------------------------------------------
174 /*
175     ¥brief マス上のボタンがクリックされた時のイベント.
176 */
177 void button_change(Fl_Widget* o, void *i) {
178   Fl_Button *button = (Fl_Button *)o;
179   int num;
180 
181   if(gameflag)
182     return;
183 
184   // 学習プレイヤーの順番の時は，何もしないで戻す
185   if(player != 0)
186     return;
187 
188   // 押されたボタンをマーク（○）に変更
189   if(state[(int)i] != 0)
190     return;
191   button->label("O");
192   state[(int)i] = 1;
193 
194   // ゲーム終了の場合は，状態を出力して終わり
195   if(gameflag = judge((int)i, 1)){
196     output();
197     return;
198   }
199 
200   // ゲーム続行の場合は，次は学習プレイヤーの番
201   for(num=0; num<9; num++)
202     if(state[num] == 0)
203       break;
204   if(num == 9)
205     return;
206 
207   // 状態を出力
208   output();
209   player = 1;
210 
```

```
211   // 行動ファイルを削除
212   remove("action.txt");
213
214   // タイマー
215   Fl::add_timeout(1.0,action);
216 }
217 //--------------------------------------------------------------------------------
218
219 //--------------------------------------------------------------------------------
220 /*
221     ¥brief resetボタンがクリックされた時のイベント.
222 */
223 void reset(Fl_Widget* o, void *i) {
224   int k;
225
226   // ボタンの初期化
227   for(k=0; k<9; k++) {
228     state[k] = 0;
229     b[k]->label("");
230     b[k]->color(FL_WHITE);
231   }
232
233   // フラグの初期化
234   gameflag = 0;
235
236   // 状態出力
237   output();
238
239   // タイマー開始
240   Fl::add_timeout(1.0,action);
241 }
242 //--------------------------------------------------------------------------------
243
244 //--------------------------------------------------------------------------------
245 /*
246     ¥brief 三目並べゲームのメイン.
247 */
248 int main(int argc, char **argv) {
249   remove("action.txt");
250   remove("state.txt");
251
252   // メニューバー
253   Fl_Menu_Bar *menubar = new Fl_Menu_Bar(0,0,300,20,0);
254   int k;
255
256   // ボタンの配置
257   for(k=0; k<9; k++) {
258     b[k] = new Fl_Button(5+55*(k%3),25+55*(int)(k/3),50,50,"");
259     b[k]->labelsize(36);
260     b[k]->labeltype(FL_SHADOW_LABEL);
261     b[k]->callback(button_change, (void *)k);
262     b[k]->color(FL_WHITE);
263   }
264
265   // メニューの追加
266   menubar->add("File/Restart", "", reset, 0);
267   window->end();
268   window->show(argc, argv);
269
270   // 学習プレイヤーが先行の場合
271   if(player==1){
272     output();
273     Fl::add_timeout(1.0,action);
274   }
275
276   return Fl::run();
277 }
```