課題 1

1.1 課題内容の補足

クリア条件として、岩をよけて画面から消えると +1 点・岩に当たると-5 点・10 点でゲームクリアという仕 様にした.

1.2 プログラムリスト

課題で作成したプログラムをソースコード1に示す.

```
ソースコード 1 課題のソースコード
1
   * memes_2012.c: 2012年度MEMEs 総合演習
2
3
4
   */
5
  #include <stdlib.h>
6
  #include "iodefine.h"
  #include "typedefine.h"
  #define printf ((int (*)(const char *,...))0x00007c7c)
10
11
#define SW6 (PD.DR.BIT.B18)
  #define SW5 (PD.DR.BIT.B17)
13
  #define SW4 (PD.DR.BIT.B16)
14
16 #define LED6 (PE.DR.BIT.B11)
  #define LED_ON (0)
17
  #define LED_OFF (1)
18
19
  #define DIG1 (PE.DR.BIT.B3)
20
  #define DIG2 (PE.DR.BIT.B2)
  #define DIG3 (PE.DR.BIT.B1)
^{23}
  #define SPK (PE.DR.BIT.B0)
24
^{25}
26 #define LCD_RS (PA.DR.BIT.B22)
  #define LCD_E (PA.DR.BIT.B23)
27
  #define LCD_RW (PD.DR.BIT.B23)
  #define LCD_DATA (PD.DR.BYTE.HH)
29
  #define Boolean int
31
  #define True 1
  #define False 0
33
35 #define NMROF_ROCKS 6
```

```
36
   struct position {
37
38
     int x;
     int y;
39
     int active;
40
41
   int point;
42
43
   //当たり判定フラグ
44
   int flag = 0;
45
46
   void wait_us(_UINT);
47
   void LCD_inst(_SBYTE);
48
   \mathbf{void} \ \mathrm{LCD\_data}(\_\mathrm{SBYTE});
49
50
   void LCD_cursor(_UINT, _UINT);
   void LCD_putch(_SBYTE);
   void LCD_putstr(_SBYTE *);
52
   void LCD_cls(void);
53
   \mathbf{void} \ \mathrm{LCD\_init}(\mathbf{void});
   void init_peior(void);
55
56
   void init_paior(void);
57
58
59
   // -- 使用する関数群 --
60
   // -----
61
   void wait_us(_UINT us) {
62
63
           _UINT val;
64
           val = us * 10 / 16;
65
           if (val >= 0xffff)
66
                  val = 0xffff;
67
68
           CMT0.CMCOR = val;
69
           CMT0.CMCSR.BIT.CMF &= 0;
70
           CMT.CMSTR.BIT.STR0 = 1;
71
           while (!CMT0.CMCSR.BIT.CMF);
72
           CMT0.CMCSR.BIT.CMF = 0;
73
74
           CMT.CMSTR.BIT.STR0 = 0;
75
76
   void LCD_inst(_SBYTE inst) {
77
           LCD_E = 0;
78
79
           LCD_RS = 0;
           LCD_RW = 0;
80
           LCD_E = 1;
81
           LCD\_DATA = inst;
82
           wait_us(1);
83
           LCD_E = 0;
84
```

```
85
             wait_us(40);
86
    }
87
    void LCD_data(_SBYTE data) {
88
             LCD_E = 0;
89
             LCD_RS = 1;
90
             LCD\_RW = 0;
91
             LCD_E = 1;
92
             LCD\_DATA = data;
93
             wait_us(1);
94
             LCD_E = 0;
95
             wait_us(40);
96
97
98
99
    void LCD_cursor(_UINT x, _UINT y) {
             if (x > 15)
100
                      x = 15;
101
             if (y > 1)
102
103
                      y = 1;
             LCD_{inst}(0x80 \mid x \mid y << 6);
104
105
106
    void LCD_putch(_SBYTE ch) {
107
             LCD_{-}data(ch);
108
    }
109
110
    void LCD_putstr(_SBYTE *str) {
111
112
             _SBYTE ch;
113
             while (ch = *str++)
114
                      LCD_putch(ch);
115
116
117
    \mathbf{void} \ \mathrm{LCD\_cls}(\mathbf{void}) \ \{
118
             LCD\_inst(0x01);
119
             wait_us(1640);
120
121
122
123
    \mathbf{void} \ \mathrm{LCD\_init}(\mathbf{void}) \ \{
             wait_us(45000);
124
             LCD_{inst}(0x30);
125
             wait_us(4100);
126
             LCD_{inst}(0x30);
127
             wait_us(100);
128
             LCD_{inst}(0x30);
129
130
             LCD_{inst}(0x38);
131
             LCD_{inst}(0x08);
132
             LCD_{inst}(0x01);
133
```

```
wait_us(1640);
134
           LCD_{inst}(0x06);
135
           LCD_{inst}(0x0c);
136
137
138
139
    // -- ゲーム用の関数群 --
140
141
    // -- 自分を移動 --
142
   void move_me(struct position *me)
143
144
           struct position old_position;
145
146
           old_position.x = me -> x;
147
           old_position.y = me -> y;
148
149
           if (AD0.ADDR0 < 0x4000) {
150
                  // -- ジョイスティック上 --
151
                  me->y = 0;
152
           else if (AD0.ADDR0 > 0xc000) 
153
                  // -- ジョイスティック下 --
154
                  me -> y = 1;
155
           }
156
157
           if (AD0.ADDR1 < 0x4000) {
158
                   //-- ジョイスティック右 --
159
                  if(me->x>=15){
160
                          me->x = 15;
161
162
                  \} else \{
                         me->x += 1;
163
164
           } else if (AD0.ADDR1 > 0xc000) {
165
                   //-- ジョイスティック左 --
166
                  if(me->x <= 0){
167
                         me -> x = 0;
168
                  } else {}
169
                         me->x-=1;
170
                  }
171
172
           }
173
           if (old_position.y != me->y || old_position.x != me->x) {
174
                  // -- 移動したとき .. 古い表示を消す --
175
                  flag = False;
176
                  LCD_cursor(old_position.x, old_position.y);
177
                  LCD_putch('□');
178
           }
179
           LCD\_cursor(me->x, me->y);
180
           LCD_putch('>');
181
182 }
```

```
183
184
     // -- 岩を移動 --
185
    void move_rock(struct position rock[], struct position me)
186
187
             int i;
188
189
             for (i = 0; i < NMROF\_ROCKS; i++) {
190
                      if (rock[i].active) {
191
           //当たり判定
192
                               if(rock[i].x == me.x \&\& rock[i].y == me.y \&\& flag == False){
193
                                       flag = True;
194
195
                                       point -= 5;
                                       printf("\rdot r); //debug
196
                               }
197
                               // 画面上に岩が存在する
198
                               LCD_cursor(rock[i].x, rock[i].y);
199
                               LCD_putch('□');
200
                               \mathbf{if}\;(\mathrm{rock}[i].x==0)\;\{
201
                                       point += 1;
202
                                       printf("¬\rPOINTS:¬%d", point);
203
                                       // 消去
204
                                       rock[i].active = 0;
205
                               } else {
206
                                       rock[i].x--;
207
208
                                       LCD_cursor(rock[i].x, rock[i].y);
                                       LCD_putch('*');
209
210
                               }
211
                      }
             }
212
213
214
215
     // -- 新しい岩を作る --
216
    \mathbf{void} \ \operatorname{new\_rock}(\mathbf{struct} \ \operatorname{position} \ \operatorname{rock}[])
217
218
             int i;
219
220
             for (i = 0; i < NMROF\_ROCKS; i++) {
221
                      \mathbf{if}\;(\mathrm{rock}[\mathrm{i}].\mathrm{active} == 0)\;\{
222
                               // -- 新しい岩 --
223
                               rock[i].active = 1;
224
                               rock[i].x = 15;
225
                               rock[i].y = rand() \% 2;
^{226}
                               LCD_cursor(rock[i].x, rock[i].y);
227
228
                               LCD_putch('*');
                               break;
229
                      }
230
             }
231
```

```
232
233
   void init_peior(void) {
234
     PFC.PEIORL.BIT.B3 = 1;
235
236
   void init_paior(void) {
237
     PFC.PAIORH.BYTE.L |= 0x0F;
238
239
240
241
242
   // -- メイン関数 --
243
   void main(){
244
          \mathbf{while}(1){
245
                struct position me; // 自分の車の座標
246
                 struct position rock[NMROF_ROCKS]; // 岩の座標
247
                int move_timing, new_timing;
248
                 int ad, i;
249
250
                 point = 0;
                flag = False;
251
252
                STB.CR4.BIT.\triangleAD0 = 0;
253
                 STB.CR4.BIT.\_CMT = 0;
254
                STB.CR4.BIT._{\text{M}}TU2 = 0;
255
256
                 CMT0.CMCSR.BIT.CKS = 1;
257
258
                 // MTU2 ch0
259
                MTU20.TCR.BIT.TPSC = 3; // 1/64選択
260
                MTU20.TCR.BIT.CCLR = 1; // TGRA のコンペアマッチでクリア
261
                 MTU20.TGRA = 31250 - 1; // 100ms
262
                 MTU20.TIER.BIT.TTGE = 1; //A/D変換開始要求を許可
263
264
265
                 //AD0
                 266
                 AD0.ADCSR.BIT.CH = 1; //AN0
267
                 AD0.ADCSR.BIT.TRGE = 1; // MTU2 からのトリガ有効
268
                 AD0.ADTSR.BIT.TRG0S = 1; // TGRA コンペアマッチでトリガ
269
270
                 // MTU2 ch1
271
                MTU21.TCR.BIT.TPSC = 3; // 1/64選択
272
                 MTU21.TCR.BIT.CCLR = 1; // TGRA のコンペアマッチでクリア
273
                MTU21.TGRA = 31250 - 1; // 100ms
274
275
276
                LCD_init();
277
                MTU2.TSTR.BIT.CST0 = 1; // MTU2 CH0 スタート
278
                 MTU2.TSTR.BIT.CST1 = 1; // MTU2 CH1 スタート
279
280
```

```
init_peior();
281
                   init_paior();
282
283
                   me.x = me.y = 0;
284
                   for (i = 0; i < NMROF\_ROCKS; i++)
285
                           rock[i].active = 0;
286
287
                    move\_timing = new\_timing = 0;
288
289
        //タイトル表示
290
                   LCD\_cursor(5, 0);
291
                   LCD_putstr("INVADER");
292
293
                   LCD\_cursor(6, 1);
                   LCD_putstr("GAME");
294
295
                   //SW6 でゲーム開始
296
                   if (SW6){
297
                           \operatorname{printf}("START\n");
298
                           \mathbf{while}\ (1)\ \{
299
                                   PA.DR.BYTE.HL &= 0xF0;
300
                                   PA.DR.BYTE.HL |= point;
301
                                   DIG1 = 1;
302
                                   //SW5 が押されて...
303
                                   if(SW5){
304
                                           \mathbf{while}(1){
305
                                                   //SW5 が離されたら
306
                                                  if(!SW5){}
307
308
                                                          //一時停止中
309
                                                          printf("\rule");
                                                             //debug
                                                          \mathbf{while}(1){
310
                                                                  //もう一度押されたら再開
311
                                                                  if(SW6){
312
313
                                                                          break;
                                                                  }
314
                                                          }
315
316
                                                  if(SW6){
317
318
                                                          break;
319
                                           }
320
321
                                   //SW4 でリセット
322
                                   if(SW4){}
323
                                           LCD_init();
324
325
                                           LCD\_cursor(5, 0);
                                           LCD_putstr("RESET");
326
                                           LCD\_cursor(6, 1);
327
                                           LCD_putstr("SEE_YA!");
328
```

```
329
                                            wait_us(5000);
                                            \operatorname{printf}("\nSEE_{\sqcup}YA!\n");
330
                                            break;
331
332
                                    if(point > 9){
333
                                            //claer
334
                                            printf("\nCLEAR!\n"); //debug
335
336
                                            break;
                                    \} else if (point < 0)
337
                                            printf("\nGAME_OVER!\n"); //debug
338
                                            break;
339
340
                                    if (MTU21.TSR.BIT.TGFA) {
341
                                            // MTU2 ch1 コンペアマッチ発生 (100ms 毎)
342
                                            MTU21.TSR.BIT.TGFA = 0; // フラグクリア
343
344
                                            move_me(&me); // 自分移動
345
                                            if (move_timing++ \geq 2) {
346
347
                                                    move\_timing = 0;
                                                    move_rock(rock, me); // 岩を移動
348
349
                                                    if (\text{new\_timing} - - < = 0) {
                                                            new_timing = rand() * 5 / (RAND_MAX + 1);
350
                                                            new_rock(rock); // 新しい岩が出現
351
352
                                                    }
                                            }
353
                                    }
354
                            }
355
                    }
356
357
            }
358
```

1.3 使用した機能の説明

今回使用した機能について、技術的な説明を記す. MEMES サポートページ http://memes.sakura.ne.jp/memes/?page_id=295 も合わせて参照されたし.

1.3.1 スイッチ

ゲームの開始・一時停止・リセットを実現するために、スイッチを SW4 \sim SW6 まで使用した。それぞれのスイッチが押されることにより PD.DR.BIT.B16 PD.DR.BIT.B18 の値が自動的に変化することを用いて制作を行った。

1.3.2 ジョイスティック

上下左右にカーソルを移動するために使用した. A/D 変換を行ってアナログジョイスティックの値をデジタル値として取得している. AD0 を 2 チャンネルスキャンモードにし, AN0 を上下の動きに, AN1 を左右の動きに対応させた.

1.3.3 7セグメント LED

仕様上 1 桁使うことができれば良いので, シンプルに PA.DR.BYTE.HL |= point といった形で表示させた.

2 オリジナル機能

実装したオリジナル機能について示す.

- 起動時にメッセージを表示
- リセットしても、またゲームを一から開始できるようにした