

Appendix 2—Safe Alternate Source Code for Game

flag_hunter_safe.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <signal.h>
4  #include <unistd.h>
5  #include <time.h>
6
7
8  const int MAX_MAGE = 500;
9  const int GUARDIAN_MAX_HEALTH = 2000000000;
10 struct Warrior
11 {
12     int type;
13     int health;
14     int mage;
15     int deadly_slashed_used;
16     int burn_cost;
17     int shield_cost;
18     int damage;
19     int deadly_damage;
20     int deadly_cost;
21 };
22
23 struct Guardian
24 {
25     int type;
26     int health;
27     int damage;
28     int heal;
29 };
30
31 void winner_handler() {
32     printf("\033[32mCongraz! You complete the task!\033[0m\n");
33     exit(0);
34 }
35
36 int getInt(char prompt[], int lower, int upper) {
```

```

37     int a = -1;
38     while (!(a >= lower && a <= upper)) {
39         printf("%s", prompt);
40         scanf("%d", &a);
41         while(getchar() != 10);
42     }
43     return a;
44 }
45
46 void print_hero_info(int type) {
47     if (type == 0 || type == 1) {
48         printf("Hero: Mage\n Skills:\n");
49         printf("    [1] Burn: Causes 200000000 Damage [cost : 100 MP]\n");
50         printf("    [2] Drink potion: Refresh MP [cost : 0 MP]\n");
51         printf("    [3] Water Shield: Defense [cost : 25 MP]\n");
52     }
53     if (type == 0 || type == 2) {
54         printf("Hero: Slayer\n Skills:\n");
55         printf("    [1] Slash: Causes 200000000 Damage\n");
56         printf("    [2] Deadly Slash: Sacrifices 200000000 health and causes 500000000 damage!\n");
57     }
58 }
59
60 void print_status(struct Guardian* guardian, struct Warrior* warrior) {
61     printf("\n===== \n");
62     printf("\t----- \n");
63     printf("\tGuardian health: %d damage: %d\n", guardian->health, guardian->damage);
64     printf("\tYour health: %d mage: %d\n", warrior->health, warrior->mage);
65     printf("\t----- \n");
66     print_hero_info(warrior->type);
67 }
68
69 void heal_guardian(struct Guardian guardian) {
70     // Check Guardian Health Cap to avoid Overflow Bugs
71     if (guardian.health + guardian.heal <= GUARDIAN_MAX_HEALTH) {
72         guardian.health += guardian.heal;
73         printf("[!] Guardian heals %d health!\n", guardian.heal);
74     } else {
75         guardian.health = GUARDIAN_MAX_HEALTH;
76         printf("[!] Guardian heals %d health!\n", GUARDIAN_MAX_HEALTH - guardian.health);
77     }
78 }

```

```

78 }
79
80 void hunt() {
81     struct Warrior warrior;
82     int choice;
83     print_hero_info(0);
84     choice = getInt(">>Choose your hero:\n\t[1] Mage\n\t[2] Slayer\n>>", 1, 2);
85     if (choice == 1) {
86         warrior.type = 1;
87         warrior.health = 400000000;
88         warrior.mage = MAX_MAGE;
89         warrior.burn_cost = 100;
90         warrior.shield_cost = 25;
91         warrior.deadly_slashed_used = 0;
92         warrior.damage = 200000000;
93         warrior.deadly_damage = 0;
94         warrior.deadly_cost = 0;
95     } else if (choice == 2) {
96         warrior.type = 2;
97         warrior.health = 400000000;
98         warrior.mage = 0;
99         warrior.burn_cost = 0;
100        warrior.shield_cost = 0;
101        warrior.deadly_slashed_used = 0;
102        warrior.damage = 200000000;
103        warrior.deadly_damage = 500000000;
104        warrior.deadly_cost = 200000000;
105    }
106
107    struct Guardian guardian;
108    printf("\n!!!Guardian has appeared!!!\n");
109    guardian.type = 1;
110    guardian.health = GUARDIAN_MAX_HEALTH;
111    guardian.damage = 100000000;
112    guardian.heal = 5000000;
113
114    while (1) {
115        print_status(&guardian, &warrior);
116        // Warrior
117        if (warrior.type == 2) {
118            choice = getInt(">>Your choice of skill:\n>>", 1, 2);
119

```

```

119     if (choice == 1) {
120         guardian.health -= warrior.damage;
121         warrior.health -= guardian.damage;
122         printf("[!] Slash deals %d damage to the Guardian!\n", warrior.damage);
123         printf("[!] Guardian cause %d damage to you!\n", guardian.damage);
124         heal_guardian(guardian);
125     } else if (choice == 2) {
126         if (warrior.deadly_slashed_used >= 1) printf("You can't use it more than once\n");
127         else if (warrior.health >= warrior.deadly_cost) {
128             warrior.health -= warrior.deadly_cost;
129             guardian.health -= warrior.deadly_damage;
130             printf("[!] Deadly Slash deals %d damage to the Guardian!\n", warrior.deadly_damage);
131             warrior.health -= guardian.damage;
132             printf("[!] Guardian cause %d damage to you!\n", guardian.damage);
133             heal_guardian(guardian);
134             warrior.deadly_slashed_used = 1;
135         } else {
136             printf("[!] Your health not enough!\n");
137         }
138     }
139 }
140
141 // Mage
142 else if (warrior.type == 1) {
143     choice = getInt(">>Your choice of skill:\n>>", 1, 3);
144     if (choice == 1) {
145         if (warrior.mage >= warrior.burn_cost) {
146             warrior.mage -= warrior.burn_cost;
147             guardian.health -= warrior.damage;
148             printf("[!] Magic Fire deals %d damage to the Guardian!\n", warrior.damage);
149         } else {
150             printf("[!] Your mage not enough!\n");
151         }
152     }
153
154     if (warrior.health >= guardian.damage) {
155         warrior.health -= guardian.damage;
156     } else {
157         warrior.health = 0;
158     }
159     printf("[!] Guardian cause %d damage to you!\n", guardian.damage);
160     heal_guardian(guardian);

```

```

160     } else if (choice == 2) {
161         warrior.mage = MAX_MAGE;
162         printf("[!] Your mage has been refreshed\n");
163         if (warrior.health >= guardian.damage) {
164             warrior.health -= guardian.damage;
165         } else {
166             warrior.health = 0;
167         }
168         printf("[!] Guardian cause %d damage to you!\n", guardian.damage);
169         heal_guardian(guardian);
170     } else if (choice == 3) {
171         if (warrior.mage >= warrior.shield_cost) {
172             warrior.mage -= warrior.shield_cost;
173             printf("[!] You use your water shield\n");
174             printf("[!] Guardian cause %d damage to you!\n", guardian.damage/10);
175             warrior.health -= guardian.damage/10;
176             heal_guardian(guardian);
177         } else {
178             printf("[!] Your mage not enough!\n");
179         }
180     }
181 }
182 }
183 if (warrior.health <= 0) {
184     printf("[FINAL STATUS]\n\tYour health: %d\n\tThe Guardian: %d\n", warrior.health, guardian.health);
185     printf("You lose\n");
186     break;
187 }
188 if (guardian.health <= 0) {
189     printf("[FINAL STATUS]\n\tYour health: %d\n\tThe Guardian: %d\n", warrior.health, guardian.health);
190     printf("You win!\n");
191     winner_handler();
192 }
193 }
194 }
195 }
196 void setup() {
197     setvbuf(stdin, NULL, _IONBF, 0);
198     setvbuf(stdout, NULL, _IONBF, 0);
199     setvbuf(stderr, NULL, _IONBF, 0);
200 }
201

```

```
202 int main() {
203     setup();
204     printf("\n=====\\n");
205     printf("          Welcome! Defeat the Guardian to gain your points.\\n");
206     printf("=====\\n\\n");
207     printf(">>press enter to start>>");
208     getchar();
209     hunt();
210     return 0;
211 }
212
```