

```
1 package main
2
3 import (
4     "bufio"
5     "fmt"
6     "os"
7     "strconv"
8     "strings"
9 )
10
11 var PLAYER int
12 var OPPONENT int
13 var gameState UltimateBoard
14 var validBoards []int
15 var timebank int
16 var time_per_move int
17
18 func getInt(strVal string) int {
19     val, err := strconv.Atoi(strVal)
20     if err != nil {
21         panic(err)
22     }
23     return val
24 }
25
26 func storeSettings(settings []string) {
27     switch field := settings[0]; field {
28     case "timebank":
29         timebank = getInt(settings[1])
30     case "time_per_move":
31         time_per_move = getInt(settings[1])
32     case "your_botid":
33         PLAYER = getInt(settings[1])
34         OPPONENT = 3 - PLAYER
35     default:
36         //dont care
37     }
38 }
39
40 func updateGame(updates []string) {
41     switch update := updates[0]; update {
42     case "field":
43         fields := projectFields(updates[1])
```

```

44     for i := 0; i < 9; i++ {
45         gameState[i] = fields[i]
46     }
47     case "macroboard":
48         validBoards = make([]int, 0, 9)
49         boardStrings := strings.Split(updates[1], ",")
50         for i, boardString := range boardStrings {
51             if isPlayable := getInt(boardString) == -1; isPlayable {
52                 validBoards = append(validBoards, i)
53             }
54         }
55         // for _, y := range validBoards {
56         //     fmt.Println(y)
57         // }
58     default:
59         //dont care
60 }
61 }
62
63 func printAction(params []string) {
64     move := RunMonteCarlo(validBoards, &gameState)
65     moveString := projectMove(move.board, move.square)
66     fmt.Println(moveString)
67 }
68
69 func main() {
70
71     scanner := bufio.NewScanner(os.Stdin)
72     for scanner.Scan() {
73         line := scanner.Text()
74         if len(line) == 0 {
75             continue
76         }
77
78         switch words := strings.Fields(line); words[0] {
79         case "settings":
80             storeSettings(words[1:])
81         case "update":
82             updateGame(words[2:])
83         case "action":
84             printAction(words[1:])
85         default:
86             //Something wrong
87             fmt.Println("default: " + words[0])
88         }
89     }

```

