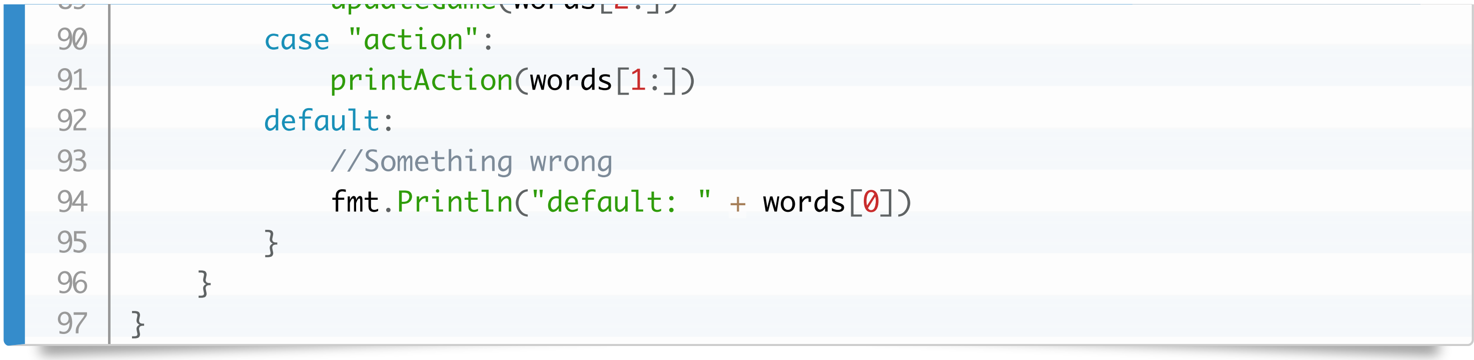
```go
package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)

var PLAYER int
var OPPONENT int
var gameState UltimateBoard
var validBoards []int
var timebank int
var time_per_move int
var time_left int

var ROUND int

func getInt(strVal string) int {
    val, err := strconv.Atoi(strVal)
    if err != nil {
        panic(err)
    }
    return val
}

func storeSettings(settings []string) {
    switch field := settings[0]; field {
    case "timebank":
        timebank = getInt(settings[1])
    case "time_per_move":
        time_per_move = getInt(settings[1])
    case "your_botid":
        PLAYER = getInt(settings[1])
        OPPONENT = 3 - PLAYER
    default:
        //dont care
    }
}

func updateGame(updates []string) {
    switch update := updates[0]; update {
    case "field":
        fields := projectFields(updates[1])

        for i := 0; i < 9; i++ {
            gameState[i] = fields[i]
        }
    case "macroboard":
        validBoards = make([]int, 0, 9)
        boardStrings := strings.Split(updates[1], ",")
        for i, boardString := range boardStrings {
            if isPlayable := getInt(boardString) == -1; isPlayable {
                validBoards = append(validBoards, i)
            }
        }
        // for _, y := range validBoards {
        //  fmt.Println(y)
        // }
    default:
        //dont care
    }
}

func printAction(params []string) {
    time_left = getInt(params[1])
    move := RunMonteCarlo(validBoards, &gameState)
    moveString := projectMove(move.board, move.square)
    fmt.Println(moveString)
}

func main() {
    ROUND = 1
    initFields()

    scanner := bufio.NewScanner(os.Stdin)
    for scanner.Scan() {
        line := scanner.Text()
        if len(line) == 0 {
            continue
        }

        switch words := strings.Fields(line); words[0] {
        case "settings":
            storeSettings(words[1:])
        case "update":
            updateGame(words[2:])
```



```
            updateGame(words[2:])
        case "action":
            printAction(words[1:])
        default:
            //Something wrong
            fmt.Println("default: " + words[0])
        }
    }
}
```