

# Programming Proverbs

- 7. “Avoid unnecessary GOTO’s.”
- Henry F. Ledgard, “Programming Proverbs: Principles of Good Programming with Numerous Examples to Improve Programming Style and Proficiency”, (Hayden Computer Programming Series), Hayden Book Company, 1st edition, ISBN-13: 978-0810455221, December 1975.

## Current simple Remote procedure call API

- includes more than these, but these are some of the simplest functions
  
- `ourselves () : integer`
  - return the id for ourselves
  
- `noplayers () : integer`
  - return the number of bots and players in ioquake
  
- `player (id) : (id, 'string name')`

## Extending the Remote procedure call API

- introduce low hanging fruit first:
  
- `disableAI () : boolean`
  - disable the in game C AI
  - return True/False if successful
  - this has been implemented

## Extending the Remote procedure call API

- `enableAI () : boolean`
  - enable the in game C AI
  - return True/False if successful
  - this has been implemented

## Extending the Remote procedure call API

- `getview () : [x, y, z]`
  - get the viewing angle of bot
  
- `setview ([x, y, z]) : boolean`
  - set the viewing angle of bot

## Extending the Remote procedure call API

- `getposition () : [x, y, z]`
- get the position of bot
- `getvisible () : [[it0][it1][it2][it3][itn-1]]`
  - return a list of objects which are visible to bot
- where
- 

```
it0 = [integer, string, position] # object id, name, position  
position = [x, y, z]
```

## Extending the Remote procedure call API

- `turn (angle) : boolean`
  - return True/False if bot can turn angle degrees.
  
- `alignwith (oid)`
  - return True/False if bot can turn to align with object id, oid

## Extending the Remote procedure call API

- `stepLeft (distance)`
  - step left by distance units
  
- `stepRight (distance)`
  - step right by distance units
  
- `stepForward (distance)`
  - step forward by distance units
  
- `stepBack (distance)`
  - step backward by distance units



## Extending the Remote procedure call API

- improve the select remote procedure call to handle an event `BUMPEDe`
  
- `gettouching() : [id0, id1, id2]`
  - get list of touching objects
  
- `health (id) : integer`
  - return an integer of the health of bot, id

## Extending the Remote procedure call API

- `weapons(id) : [w0, w1, w2, w3, wn]`
  - return list of weapons
  - `w0 = [integer string, ammo]` # weapon id, name, ammo count
  
- `changeweapon(wid) : boolean`
  - return True/False is bot can change weapon to, wid