Description of changes made to ioquake to disable C ai

- **ioquake-latest/ioquake3/code/game/ai_main.c**

```
int BotAI(int client, float thinktime) {
  bot_state_t *bs;
  char buf[1024], *args;
  int j;

  trap_EA_ResetInput(client);

  bs = botstates[client];
  if (!bs || !bs->inuse) {
    BotAI_Print(PRT_FATAL, ``BotAI: client %d is not setup0, client);
    return qfalse;
  }

  //retrieve the current client state
  BotAI_GetClientState( client, &bs->cur_ps );
```

Description of changes made to ioquake to disable C ai

- **ioquake-latest/ioquake3/code/game/ai_main.c**

```
#if 1
    if (trap_use_c_ai (client))    // new code goes here
      BotAI_Print(PRT_MESSAGE, ''yes found use_c_ai is true !!!!0);
    else
      BotAI_Print(PRT_MESSAGE, ``yes found use_c_ai is false !!!!0);
    if (! trap_use_c_ai (client))
      return qtrue;

      // original code follows
#endif
```

Description of changes made to ioquake to disable C ai

- the above code has since been modified as it returns too early

Introducing a trap

- notice the new trap function `trap_use_c_ai (client)`
- how is this implemented?
  - it is an ioquake system call which will be used to make calls between the main program and the shared library

Introducing a trap

- **ioquake-latest/ioquake3/code/game/g_syscalls.asm:54**

```
equ trap_GetEnemy               -49
equ trap_SetEnemy               -50
equ trap_HaveReachedGoal        -51
equ trap_use_c_ai               -52

equ memset -101
equ memcpy -102
```

- notice that it is possible to add extra traps in descending order from `-53` to `-101`

Introducing a trap

- **ioquake-latest/ioquake3/code/game/g_syscalls.c:170**

```
// new code goes here!
int trap_use_c_ai( int ch) {
  return syscall( G_USE_C_AI, ch );
}
// end of new code
```

Introducing a trap

- **ioquake-latest/ioquake3/code/game/g_local.h**

```
void    trap_GetEnemy (int num, int ch, int *enemy);
void    trap_HaveReachedGoal (int num, int ch);
int    trap_use_c_ai (int ch);   // new code goes here
voidtrap_GetServerinfo( char *buffer, int bufferSize );
voidtrap_SetBrushModel( gentity_t *ent, const char *name );
```

Introducing a trap

- **ioquake-latest/ioquake3/code/game/g_public.h:215**

```
    G_GETENEMY, // (int clientNum, int *enemy) // gaius added this
    G_SETENEMY, // (int clientNum, int enemy) // gaius added this
    G_HAVEREACHEDGOAL, // (int clientNum, int enemy) // gaius added this
    G_USE_C_AI,  // this allows us to test whether the C AI is enabled

    G_GET_ENTITY_TOKEN,    // qboolean ( char *buffer, int bufferSize )
    // Retrieves the next string token from the entity spawn text, returning
```

Introducing a trap

- **ioquake-latest/ioquake3/code/server/sv_game.c**

```
    return SV_SetEnemy (args[1], args[2], args[3] ); // gaius
  case G_HAVEREACHEDGOAL:
    return SV_HaveReachedGoal (args[1], args[2] ); // gaius
  case G_USE_C_AI:  // new code here
    return SV_use_c_ai (args[1]); // new code here

  case G_SET_CONFIGSTRING:
    SV_SetConfigstring( args[1], VMA(2) );
```

Introducing a trap

- **ioquake-latest/ioquake3/code/server/sv_bot.c**

```
int SV_use_c_ai (int client)
{
  return use_c_ai (client);
}
```