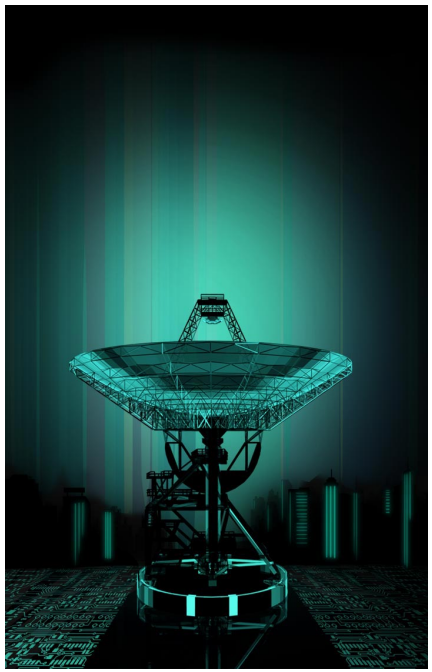
# Dwarf Fortress
# Reversing for !!fun!! and ruby



**RECON 2012**

Yoann "*jj*" Guillot
Sogeti / ESEC R&D
yoann.guillot(at)sogeti.com

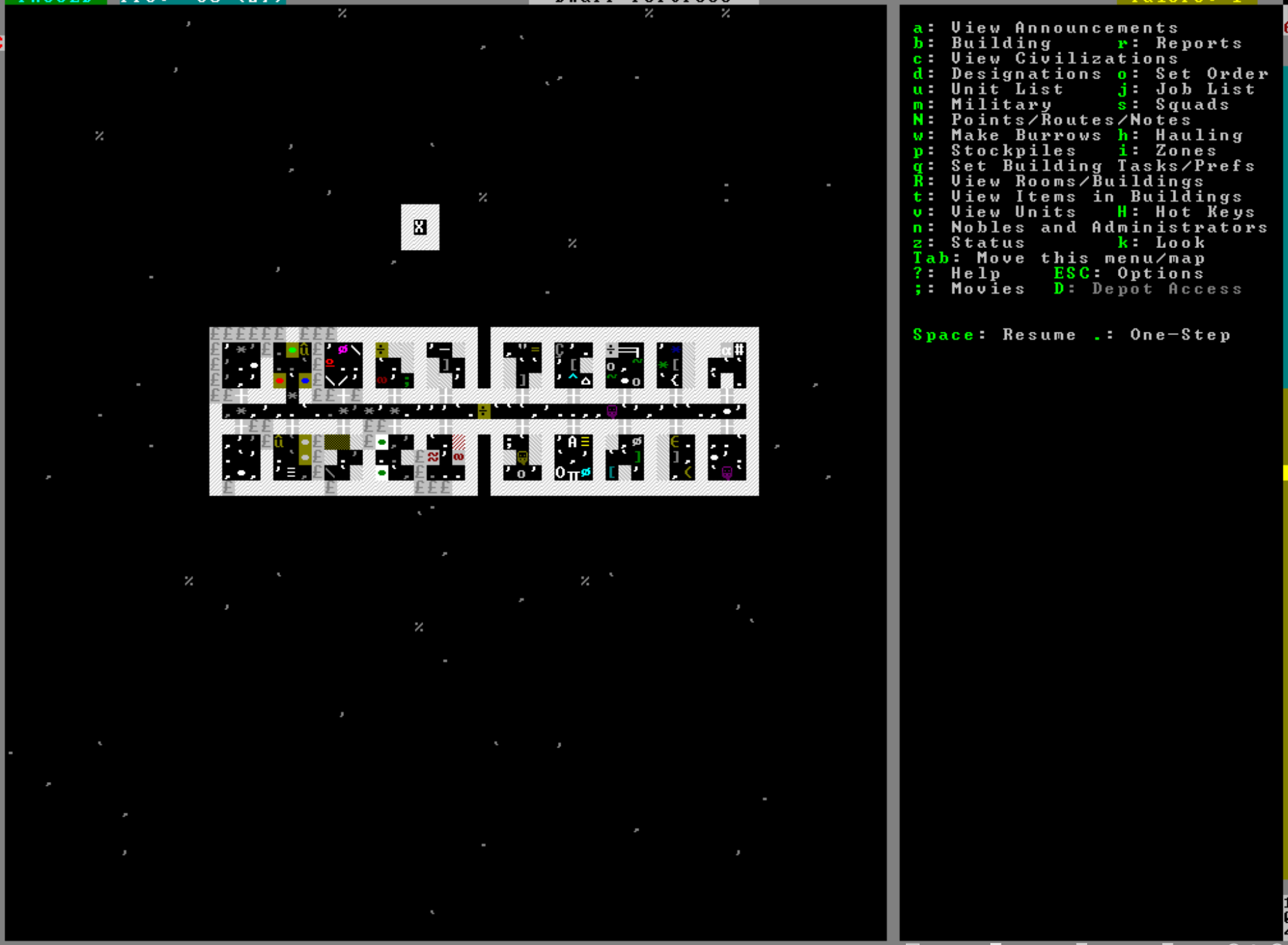
# Plan

# The game

- `http://www.bay12games.com/dwarves/`
- Indie game
- Free
- Massively Singleplayer Offline Role-Playing Game
- Fantasy universe simulation
- Very detailed world
- Steep learning curve

# The game

- Adventurer
  - Human, dwarf, elf
  - Quests based
- Dwarf Fortress
  - 7 dwarves
  - survival horror


*PAUSED* FPS: 30 (29)
Dwarf Fortress
Idlers: 1
a: View Announcements
b: Building r: Reports
c: View Civilizations
d: Designations o: Set Order
u: Unit List j: Job List
m: Military s: Squads
N: Points/Routes/Notes
w: Make Burrows h: Hauling
p: Stockpiles i: Zones
q: Set Building Tasks/Prefs
R: View Rooms/Buildings
t: View Items in Buildings
v: View Units H: Hot Keys
n: Nobles and Administrators
z: Status k: Look
Tab: Move this menu/map
?: Help ESC: Options
;: Movies D: Depot Access
Space: Resume .: One-Step

# The dev

- Single developper for 10y
  - Tarn "*Toady One*" Adams
  - Lives off donations
- Closed source
- C++ code
  - Toady is *not* a programmer
- Frequent new releases
- Windows/Linux/MacOS x86 binaries

# The hacks

- Active hacking community
- Part for cheats
- Part for game improvements
  - Esp. UI
- Backed by the developper

# Plan

# Before

- everybody has his own tool
- poking at the game memory
- unique or shared offsets

# Before

- everybody has his own tool
- poking at the game memory
- unique or shared offsets
- new game release $\rightarrow$ drama
  - regexps on the code
- orphaned tools unmaintenable

# DFHack

- by peterix
- framework for tools
- opensource
- “authoritative” source of offsets
- stable tool API across DF versions
- cross-OS

# History

- started as a .dll to include
- standalone binary per tool
- describe offsets in an xml
  - indexed by DF binary md5 / PE timestamp

# History

- started as a .dll to include
- standalone binary per tool
- describe offsets in an xml
  - indexed by DF binary md5 / PE timestamp
- moved in-process
  - speed
  - synchronization
  - malloc
- tools in a CLI
- RPC for GUI tools

# History

- offsets in xml → C++ definitions
  - easier handling of new field insertion in new DF
  - only latest DF supported
  - lack introspection

# Codegen

- by angavrilov
- game internal struct defs
- in XML
- perl backends
  - C++ headers
  - generate tool snippets

# Codegen

- by angavrilov
- game internal struct defs
- in XML
- perl backends
  - C++ headers
  - generate tool snippets
  - Ruby/whatever struct description

# Codegen

- by angavrilov
- game internal struct defs
- in XML
- perl backends
  - C++ headers
  - generate tool snippets
  - Ruby/whatever struct description
  - Offsets file for external tools

# Codegen

- by angavrilov
- game internal struct defs
- in XML
- perl backends
  - C++ headers
  - generate tool snippets
  - Ruby/whatever struct description
  - Offsets file for external tools
  - C headers

# IDA 5 C headers

**Warning**

The following applies to IDA free only

- Do not use reserved prefixes
  - Not for enum names
  - Not for struct fields
  - sub loc locret off byte algn unk ...

# IDA 5 C headers

**Warning**

The following applies to IDA free only

- Do not use reserved prefixes
  - Not for enum names
  - Not for struct fields
  - sub loc locret off byte algn unk ...
- Do not use complex pointers

# IDA 5 C headers

**Warning**

The following applies to IDA free only

- Do not use reserved prefixes
  - Not for enum names
  - Not for struct fields
  - sub loc locret off byte algn unk ...
- Do not use complex pointers
- No bitfields
  - Use enums instead
  - struct { moo:1; baa:1; xx:1 }
    → enum { moo=1, baa=2, xx=4 }

# Plan

# Scripting FTW

- Plugins are cool, scripting is best
  - Recompilation is boring and complex
  - Every plugin has his option parser
- Script lang + basic funcs + user programming = win
- can distribute full scripts

# Scripting FTW

- Plugins are cool, scripting is best
  - Recompilation is boring and complex
  - Every plugin has his option parser
- Script lang + basic funcs + user programming = win
- can distribute full scripts
- Give a man a script, he'll be fed for today
- Teach a man to code, he'll spawn fish in the sky

# Embedding ruby

- Ruby1.9 sucks
  - Hard to embed in a multithreaded app
  - 32bit compilation on x64 distro
  - Not sure about Windows

# Embedding ruby

- Ruby1.9 sucks
  - Hard to embed in a multithreaded app
  - 32bit compilation on x64 distro
  - Not sure about Windows
- Ruby1.8 works
- dlopen/LoadLibrary ftw

# Ruby to struct

- Can chose to def structs in C
  - codegen → ruby C ext
  - giant C file
  - static

# Ruby to struct

- Can chose to def structs in C
  - codegen → ruby C ext
  - giant C file
  - static
- Or def structs in ruby
  - codegen → ruby
  - basic C file
  - ???
  - profit

# Primitives

- Still need basic methods in C
  - Raw std::vector accessors
  - Raw std::string accessors
  - Raw memory (int/buffer)
  - malloc/free
  - Raw vmethod call
- Basic blocks for complex ruby code
- Wrap some dfhack APIs

# VMethods

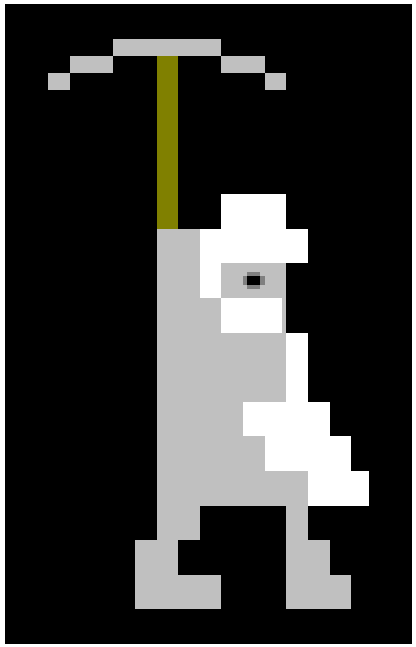
- Want to call any vfptr on any object
- On linux/gcc, __cdecl
  - virtual destructor takes 2 slots
- On windows/msvc, __thiscall
  - MSVC cannot cast fptr to __thiscall
  - use __fastcall, with dummy edx
  - need 1 variant per argument count (or asm magic)

# Plan

# Scanning for offsets

- find static analysis ways to autogenerate xmls
- code that with metasm
  - scan_nextid
  - scan_vtable
- autodetect struct layout modifications
  - from running process, using padding?
- autodetect vtable layout changes
- make ruby plugin compile everywhere

# Questions ?



# References


- http://www.bay12games.com/dwarves/
- http://github.com/peterix/dfhack/
- http://github.com/angavrilov/df-structures/
- http://github.com/jjyg/df_misc/
- http://metasm.cr0.org/