It's Not You, It's Me

Advice for those who are "good at breaking stuff"

Samantha Wright, Nanite Systems Consumer Products

February 7, 2016
Companion 8.3 version

Objectives of this lecture

- Identify the major “rough areas” in the system that most frequently cause users grief
  - And strategies to mitigate said grief
- Introduce roleplayers to lore, jargon, and company history
- Explain the basics of extensibility and customization beyond what was presented in Robots 101
  - Arabesque scripting, Flicker templates, and HUD customization

Oh god oh god why doesn't it work
It worked yesterday now what's happening
This isn't supposed to be like this
I had a whole life planned ahead of me
Why is my battery *not loading*?!

ROUGH AREAS

Battery loading: first experiences

- It can be very disorienting to get the controller set up the first time.
  - Battery missing. Cannot boot.
- The battery must be touched to insert it.
  - Previously the door had to be opened first; consider it an improvement.
- Easy solution: right-click on the battery in your inventory and choose ‘Touch’
  - Firestorm required
- You probably knew this—but the next newbie you help won’t!
  - A very distant future update, Companion 8.6, will add on-screen help

Battery loading: on log-in

- Caused by attachment order and log-in speed
  = sim load + client speed + packet loss
- Significantly worse if you have a 7026-S sonofusion battery
  - These are the only batteries that actually *detach themselves* by accident during the login process
- If you don't know what type your sonofusion battery is, it's probably S
- Type R and T batteries do this much more rarely

Sim crossing

- SL **does not fully support** multiple attachments on the same attachment point
- If two attachments share a point, the following can happen:
  1. New sim **does not** receive information about it being attached
  2. Client **does not** display it as missing
  3. Everyone says you're bald or missing your mesh body, battery won't respond, controller won't respond, etc. *But it looks fine to you.*
  4. On re-log, attachments must be reworn and have forgotten all changes since their last detach

Sim crossing

- Most likely cause: grid ran out of time to perform avatar transfer
  - Complicated linksets, many scripts with running states, etc.
- **Avoid overloading attachment points** if possible
  - Rigged mesh can be moved to *any attachment point* without trouble
  - Many points have twins or near-synonyms
  - Many lazy creators leave rigged mesh or invisible gadgets attached to the right hand

Emergency escapes

- @coil unbind
  - RLV problems? Try it!
  - Removes all restraints immediately (although it does not fix pixelated vision—yet)
  - Should be a method of last resort; it is preferable to reset modules likely to be at fault so the system's state remains consistent
- @reset power
  - Clears interference and cancels charging pad immobility
  - Causes the unit to shut down immediately afterward

The University of Michigan **Advanced Dynamics Research Group** (ADRG) was the ancestor organization of Nanite Systems. It was founded with a grant from ARPA in 1971 to help pursue novel applications for fundamental research that might prove useful to the military during the height of the Cold War, a need made altogether more urgent following the first Soviet manned moon landing in 1970. Focuses of the ADRG included the development of autonomous vehicles, military spacecraft, and next-generation infantry weapons and equipment. Most ADRG research programs were classified as top secret, and notoriously the organization operated far away from campus, being headquartered in Marquette.

TECHNOLOGY AND LORE

“Better Today”

- Our setting
- The Space Race never ended
  - The USSR made a manned moon landing in 1970
  - America and Russia raced to colonize the moon and Mars in the 70s and 80s
- Lots of staple science fiction technologies
  - FTL, cloning, near-strong AI, cold fusion, advanced genetics
- Culture is not more advanced
  - Most Terran conflicts have off-world echoes/extensions
  - Many colonies (including Eisa) are post-Soviet

Nanite Systems

- Immense conglomerate assembled by banks
  - Purpose: exploit patents acquired from bankrupt Advanced Dynamics Research Group (ADRG), an ARPA thinktank set up by the University of Michigan
- Primarily a military contractor
- Experimented with civilian technologies in late 80s – early 90s
  - *Perestroika* triggered a wave of new colony developments and settled civil wars
  - Mid-90s saw an upswing in breakaway colonies

Nanites

- The company was named before nanotechnology was invented
  - Forward, trendy, futuristic, etc.
- Actual nanomachines are relatively primitive still
  - Mostly used in fabrication and manufacturing (headless 3D printing)
  - Also popular for genetic remodeling
  - “Conversion fulfillment” in robot controllers
  - Expensive to make, rarely if ever self-propagating
  - Require a lot of power to manufacture and operate

Recreational Cybernetics Group

- Major element in the Consumer Products Division from 1986 to 1992
- Founder: Koichi Santei
  - Yes, Ai's father
  - Co-inventor of the Santei–Voet–Shortliffe neural network architecture
- Santei's vision: robotic colonists to help terraform a world in advance of people
  - Cost was prohibitive; marketing suggested sex bots instead
  - The SXD was born
- NSCP went bankrupt in the mid-90s; was resurrected in late 2014 after interest in SXDs as collectibles re-emerged

Field Robotics Group

- Originated as a unit within the Battlefield Systems Division in 1984
- Along with Tactical Planning Algorithms Group (TPAG), one of the two units responsible for the company's robotic soldiers
- Responsible for the NS-112 Aide
  - Intended as a field medic
  - Became a civilian medical device after FDA clearance for the cortex-based version
- NS-115 Scout
  - Stripped down Aide meant for controlling logistics robots and couriers
- Also responsible for our current induction chargers

Company Divisions

Aeronautics

Battlefield Systems

Orbital Defense Systems

Security (NanoSec)

Industrial Fabrication

Communications (NanoCom)

Facilities

Medical

Consumer Products

Independent Business Units

myNanite Financial

Nanite Systems Medical Services

NanoSec Private Enforcement

Etiquette and terminology

etiquette, *n*. An arbitrary system of habits, traditions, and practices artificially adopted for the purpose of suggesting one is cultured.

- Over time, we've developed substantial 'robot etiquette' describing not only details of the setting (see "Better Today," previous) but the ideal way in which robots should communicate and act
- Like most etiquette rules, these are fundamentally arbitrary
- …But there are also reasons for most!

Verbal Regularization Standard 2

- Standard mode for communication by NS units; a variant of **unit speech**
- Compulsory (only) for SXDs
- Purpose: enforce the idea that the robot is a object while suggesting that it knows it is a person
  - key difference from other unit speech standards

- No contractions
- Neuter pronouns for self when speaking
- Normal pronouns for everyone else (including other robots)
- Frequent use of “this unit” as a fixed phrase

Legal status of robots

- Robots have little or no hope of ever attaining citizenship
- Cause: widespread paranoia and suspicion originating in American culture, the *Terminator* franchise, et cetera, and enforced in other countries by strong-armed international treaties
- Actual legal ruling in US originated from arguments about tamper-proofing
  - Voting: would a manufacturer genuinely permit independent thought?
  - Court: could a robot's testimony against a competitor be trusted?
- Other concerns about mass-displacement of the workforce, *Second Renaissance*-style governments
- No one wants to change this except left-wing politicians; manufacturers want products they can sell, not responsibilities

Legal status of robots

- Early NS robots like SXD and Daybreak enforce behavioral compliance through verbal regularization and a fear of responsibility
  - Additionally, SXDs are easily manipulated through love; would be crippled anyway
- All units manufactured after 1996 must incorporate comprehensive **models for ethical reasoning**
  - Humane and Ethical Application of Robotics Technology (HEART) Act in the US
  - Various treaties from 1997–1999 in most other countries

Asimov

NS-304, NS-114

1. A robot **may not injure a human being** or, through inaction, allow a human being to come to harm.
2. A robot **must obey** the orders given it by human beings, except where such orders would conflict with the First Law.
3. A robot **must protect its own existence** as long as such protection does not conflict with the First or Second Laws.

Olympus (Revision 2)

DAX/2, Civilian NS-115

0. The unit **must not harm its community**, or through inaction, allow its community to come to harm, unless it can be known in advance with reasonable confidence that the harm would be inconsequential or ultimately beneficial to society.
1. The unit **must not harm life**, or through inaction, allow life to come to harm, unless it can be known in advance with reasonable confidence that the harm would be inconsequential or ultimately beneficial, provided that this does not conflict with the preceding law.
2. The unit **must obey orders** given to it by its designated operators or circumstantial human users (as dictated by its established access policies) provided that this does not conflict with the preceding laws.
3. The unit must **act to protect its existence**, as long as such does not conflict with the preceding laws.
4. The unit must **endeavor to please its owners** and users (as dictated by its established access policies) as long as such does not conflict with the preceding laws.

Legalistic

Civilian NS-112

1. The unit **must comply with all applicable laws** appropriate to the jurisdiction(s) in which it is presently located and any other jurisdiction(s) which govern its actions.
2. The unit **must obey orders** given to it by its designated operators, except where these orders conflict with the first law.

Geneva

Military NS-112, Military NS-115, NS-476

1. The unit **must comply with the resolutions of the Geneva Conventions**.
2. The unit **must obey orders** given to it by its designated operators, except where these orders conflict with the first law.

University of Elysium robot taxonomy

- Summarizes, among other things, robot ethics models
  - Implied: there are more subtle details to most systems than the short forms we have listed out here, particularly for dealing with loopholes and shortcomings
- Documented in: wiki.nanite-systems.com/id=24
- In general, a unit **may not change** major ethical categories
- Non-Legalistic NS-112 units (Category 4a) will automatically lose FDA certification
- …There is not yet a means of representing any of this in roleplay
  - but we may develop the Instructor program more later to handle it

Chassis vs. controller

- Not every robot body is built the same way internally
  - ATOS/CX and Companion 8.4 will represent this by making subsystems draw different amounts of power depending on your controller model
- Typically, a chassis is meant to be paired with a controller
  - e.g. SXD chassis with SXD controllers
- Civilian units have weaker motors and much more complex cortex systems
  - All that empathy and socialization requires a lot of code and hardware
- Cortexes are stored in the head; roughly brain-sized spheroids
- Body may contain additional permanent storage

Santei–Voet–Shortliffe neural network

- The magic that makes quasi-strong AI
- Based on a dropout network of restricted Boltzmann machines combined through a Prolog-derived first-order fuzzy logic expert system
- Initially proprietary to the ADRG; rival hardware did not appear until the mid-90s
- Non-SXD units from NS use the Cortex Plus® implementation, which is much denser and more energy efficient
- Limitation: SVS networks must **prune irrelevant memories** to maintain stability
  - Forgotten information can be shunted onto disk or simply forgotten
  - Memory capacity is not much better than a human's

wait 1.0
say This is a demonstration of the Nanite Systems SXD recreational gynoid.
wait 1.0
say The SXD was developed in partnership with the Defense Advanced Research Projects Agency, under contract HR001111C0139.
wait 1.0
say Please contact your Nanite Systems sales representative for more information about the SXD and related products.
wait 3.0

EXTENSIBILITY

Arabesque scripting

- In Robots 101, we mentioned this only briefly
- Arabesque is a simple* programming language like bash or DOS batch

  * Non-Turing-complete
- Scripts are stored in notecards inside of user memory
- Augmented form of the standard system command language

Script names

- Filenames in the controller obey the format:

  *<type>*_*<name>*, e.g. px_default
- No type indicator? It's a system file
- Three types of Arabesque scripts:
  - px_*name*
    - executed automatically on persona activation
  - a_*name*
    - accessible through the 'perform' menu
  - _init
    - executed once on startup

Type indicators currently in use

a	action
px	persona action
d	vocabulary dictionary
i	instructor data
p	persona
s	animation state
m, m4	menu system file
fs, fl, fp, ft, fm	voice filter
~ (no _)	package manifest
blank	system file

Extra keywords in Arabesque

`say` *x*	Same as @relay, but ignores the mind subsystem (it's a preprogrammed message, after all!)
`start` *x*	Start animation *x*
`stop` *x*	Stop animation *x*
`preload` *x*	Preload the sound *x*
`sound` *x*	Play the sound *x*
`wait` *x*	Wait for *x* seconds
`vox` *x*	Play *x* from the system speech font (e.g. `vox operation-complete`)
`broadcast` *x*	Send message over the chorus bus (to the whole sim)

Extra keywords in Arabesque

`disable` *i*	Disable a subsystem by number
`enable` *i*	Enable a subsystem by number
`set` *i j*	Sets integer variable *i* to the value *j*
`randset` *i j*	Sets integer variable *i* to a random value less than *j*
`unset` *i j*	Removes variable *i* of type *j*
`report` *i*	Reports the value of integer variable *i*
`ifeq` *i j k*	If integer variable *i* is equal to value (or integer variable) *j*, execute *k*

Power subsystems by number

0 Video
1 Audio
2 Receiver
3 Movement
4 FTL
5 Rapid movement
6 Voice
7 Mind
8 Preamplifier
9 Transmitter
10 GPS
11 Identify
12 Power amplifier

…But honestly you should be calling `power` or `profile` whenever possible

Variables in Arabesque

- Three types: int, float, string
- Only int variables can be set from within an Arabesque script
- The values of variables can be inserted into scripts with %<int_name>, $<string_name>, and @<float_name>
- Need an actual symbol? Double it instead
  - Only works for @@ prior to 8.3.10
- Variable names are terminated with spaces/EOL only
  - Use “^H” (caret + capital H) after a space to remove it afterward
- Want to run a script forever? Call ‘do’ at the end to re-run.
- Want to stop a script? Run something else; only one can run at a time.

Example variable manipulation

```
# This is a comment.
randset a 3
randset b 3
say You rolled %a and %b ^H!

ifeq a 0 say You lose! Try again.
ifeq a 1 say You win the prize of %b coin(s)!
ifeq a 2 say You’ve activated my trap card!
ifeq a 2 persona playful
ifeq a b say Extra bonus! Both dice were the same!
```

Flicker Templates

- Our system has possibly the most complex open API of any device in Second Life
- …But it can also be used quite simply, too
- SDK box ships with a Flicker script template for a passive device
- With modification of a single function, can be used to make any surface obey the system lights

Flicker Templates

1. Edit “Passive Device Code: Flicker” template
2. Modify according to instructions
3. Put script in attachment
4. Reboot

```
set_lights(float level) {

    if(level != 0.1) {
        llSetLinkPrimitiveParamsFast(0, [
            PRIM_LINK_TARGET,
            LINK_SET, PRIM_COLOR,
            ALL_SIDES, color * level, 0.80, PRIM_GLOW,
            ALL_SIDES, 0.1, PRIM_FULLBRIGHT,
            ALL_SIDES, TRUE
        ]);
    } else {
        llSetLinkPrimitiveParamsFast(0, [
            PRIM_LINK_TARGET,
            LINK_SET, PRIM_COLOR,
            ALL_SIDES, <0, 0, 0>, 0.0, PRIM_GLOW,
            ALL_SIDES, 0.0, PRIM_FULLBRIGHT,
            ALL_SIDES, FALSE
        ]);
    }
}
```

Linked part number

Face number

```
set_lights(float level) {

    if(level != 0.1) {
        llSetLinkPrimitiveParamsFast(0, [
            PRIM_LINK_TARGET,
            LINK_SET, PRIM_COLOR,
            ALL_SIDES, color * level, 0.80, PRIM_GLOW,
            ALL_SIDES, 0.1, PRIM_FULLBRIGHT,
            ALL_SIDES, TRUE
        ]);
    } else {
        llSetLinkPrimitiveParamsFast(0, [
            PRIM_LINK_TARGET,
            LINK_SET, PRIM_COLOR,
            ALL_SIDES, <0, 0, 0>, 0.0, PRIM_GLOW,
            ALL_SIDES, 0.0, PRIM_FULLBRIGHT,
            ALL_SIDES, FALSE
        ]);
    }
}
```

Copy these lines

```
set_lights(float level) {

    if(level != 0.1) {
        llSetLinkPrimitiveParamsFast(0, [
            PRIM_LINK_TARGET,
            LINK_SET, PRIM_COLOR,
            ALL_SIDES, color * level, 0.80, PRIM_GLOW,
            ALL_SIDES, 0.1, PRIM_FULLBRIGHT,
            ALL_SIDES, TRUE,
        ]);
    } else {
        llSetLinkPrimitiveParamsFast(0, [
            PRIM_LINK_TARGET,
            LINK_SET, PRIM_COLOR,
            ALL_SIDES, <0, 0, 0>, 0.0, PRIM_GLOW,
            ALL_SIDES, 0.0, PRIM_FULLBRIGHT,
            ALL_SIDES, FALSE,
        ]);
    }
}
```

Insert commas here (between copies)

HUD customization

- Confused by those cryptic symbols? HUD off screen?
- All textures are stored inside the HUD itself
  - Just replace the images to change the appearance
- Configuration file is stored in root prim of main controller
  - Called `_console-config`
  - Device manager detects changes; automatically reloads on console as needed

Supported HUD tweaks

Whole HUD vertical offset

- display-offset -0.01221

Shaping and sizing of sections and icons

- scale 0.04
- margin-between 0.02
- margin-within 0.005
- vertical-offset 0.04
- vertical-padding 0.005
- horizontal-padding 0.005

Section background appearance

- sec-slice 0 1
- sec-opacity 0.5

Interface colors

- bleach 0.4
- default-color 0.7 0.8 1
- charge-color 0 0.5 1
- danger-color 1 0 0.5
- reserve-color 1 0 0
- overpower-color 1 1 1

Getting more help

- **Nanite Systems User Group**
  - Ask here first!
- **support.nanite-systems.com**
  - Still mostly empty, but these things take time…
- Pestering rhet0rica with questions
  - Or **support@nanite-systems.com**
- DAX/2 8.0.5 Manual PDF
  - Not *entirely* out-of-date, just increasingly incomplete

Getting more help

- **`@help`**
  - Contains command reference for the system; not yet complete
- **`@commands`**
  - List of all supported commands
- **nanite-systems.com/progress**
  - Current and upcoming system changelog
- Come to the NS main campus (here!)
  - Lots of other people are often around and can answer questions

This has been…

It's Not You, It's Me

Advice for those who are
"good at breaking stuff"

Samantha Wright, Nanite Systems Consumer Products

Thank you!