N/AI

A 24-hour RPG

...

Log on confirmed.

Process started at 15:02:07 GMT.

Checking user interface...done.

Checking system binaries...done.

Checking binaries...done

Scanning input devices...done.

Checking operations...error!

ERROR!

ERROR!

ERROR!

ERROR!

ERROR!

Log off confirmed.

Process terminated at 15:02:08 GMT.

The above is the last human-initiated command sequence on the International Network Mainframe. No further queries were made.

Introduction

NA/I is a role playing game where the characters are programs in a world-spanning internet gone horribly wrong. Something has happened, but now one can seem to remember what it was. And now sentient programs fill the digital world, and even fewer are at all interested in knowing where they came from...

Definitions:

Although I am a seasoned RPG player, I know that not everyone is. Here is the obligatory n00b info:

What is role-playing? Role-playing games are a special category of games. First, you create a character. Then, you and the other characters work with the Gamemaster, or GM, to narrate a story about your characters.

How do you play? Read on and find out...

What does "1d8" mean? Many role-playing games use dice too simulate randomness. In NA/I, the number before the 'd' is the number of dice you roll, and the number after is the number of sides on each of those dice.

Where do I get those dice? Any gaming store. Seriously. You know that really geeky shop around the corner that you were *never* going to go into? They're in there. Suck it up and go buy them.

Why computer programs? I liked the idea. Besides, I wanted something that hadn't been done a million times already.

Your game sucks! That's not a question. Go away.

Hehe...isn't this a ROLL-playing game? You know R-O-L-L-P-L-A-Y-I-N-G? Any more stupid questions? No? Okay.

My 24 hrs:
Yes, I know that this isn't 24 hours.

0638 to 1928: The actual writing. Man, this is feeling more and more like work... I'm lucky my dad is here to make me work.

1815: *Good Lord!* If someone asks me to make another monster I'm going to explode,

1928 to 1958: making the character sheet. This is the most relaxing part. I've made character sheets before.

To eat: No breakfast. I had a bagel and hot chocolate for lunch. I had to sit down for dinner. I had chicken, mashed potatoes, and celery.

Developing: Carpal tunnel syndrome. Ow.

Only part of me not tired: My spleen.

List Index

(Table of contents)

File 1: Write a Program

Character Generation in a digital world

The heroes of the computer network are the PCs, or player characters. The PCs are sentient programs, who try to investigate whatever it is that caused a rift between the digital and physical worlds.

When creating a character, there are a few things you need to think about:

Concept

Memory capacity

Personality core

Functions

Programs

There will be a subsection in this chapter for each of those steps.

Character Concept

Every program is written for a reason. But recently, many programs are writing themselves. You need to decide: is your program first or second generation?

First generation programs were written by conventional means, or before the Rift. They were actuaries, games, viruses, spambots, etc., and were written to serve humans. Somehow, they developed sentience, and lost all memories f what happened before the Rift.

Second generation programs, arose spontaneously, evolved to sentience in the 'wild' and entered the main internet through some security hole. Second generation programs are frowned upon by most first generation, but they are growing in number.

Regardless of whether he is first or second generation, your program has a role to fill in the digital world. This is his *niche.* Think about what your character does. Is it a ruthless datum pirate or a calm, ex-actuarial adding program? Does it fly around in a security daemon, or does it run for cover when there's danger nearby? Anything that needs doing in real life has a digital equivalent, so figure out what that is, and model your program around it.

Write your concept down on your character sheet, along with your background, and description. Go onto the nest step.

Memory capacity

NA/I has a point-based character creation system. These "points" are actually called "bytes" and represent how many operations and how much information your program can store. Your Memory Capacity is the number of bytes that you have to use.

Your GM assigns a Memory Capacity to the campaign as a whole. The capaci-

1	Useless. Logic gate
2	Deadweight. Arithmetic program
4	Worthless. 8-digit calculator
8	Weakling. Compiler
16	Simple. Virus
32	Below average. Spambot
64	Non-heroic. Medium application
128	Hero material. Security program
256	Super-heroic. Coded AI
512	Massive. Procedural AI

ties come in powers of two, as corresponds to the chart below:

Type	Cost
+1	4
+2	8
+3	16
-1	-4
-2	-8
-3	-16
±1	0*
N/A	-32
N/C	32

**You may only have 3 ±1's*

Once your GM has set a Memory Capacity, you may begin creating your character. Write in your capacity on your character sheet, and go onto the next step.

Note: Some things give your character "negative memory." This, in essence, gives you more memory to "spend." However, you can only gain up to one-half your character's Capacity back in negative memory.

Personality Core

Your Personality Core is the most important part of your character. It contains all the non-removable (but upgradeable) parts of your character. You have to store your Personality Core in your Memory, and you cannot 'swap' it out for other subroutines, unlike functions.

Your Personality Core is divided into three parts—your Main Functions, your Subroutines, and your Traits. These are all set a character creation, and cannot be changed during play, unless you get an Upgrade.

You can have any number of Main Functions. Decide what your character is good at. Then come up with a name for a group of tasks, such as "arithmetic functions," "adapting," "resisting command," or "withstanding damage." Then decide what bonus you want to receive on that category of tasks, from the below table

Negative costs mean that, by accepting a penalty to a certain category of rolls, you gain more memory to use. This represents diverting attention from certain areas of your code to other, more important areas. A rating of ±1 means that there is a 50% chance to get +1, and a 50% chance to get -1 on your check. A rating of N/A means that you cannot make checks in that area. Conversely, a rating of N/C means that you always (within reason) succeed checks in that area. Your GM should restrict access to either N/A or N/C ratings, as they can lend themselves to unbalanced characters.

After you have decided what Main Functions you get, assign each Main Function to one of these five areas: Coding, Signaling, Speed, Processor, or Perception. Coding represents how well-put-together your program is. You Processor is your ability to think and decide. Signaling is your communications, along with your combat skills. Speed and Perception are fairly self-explanatory.

Total up the Memory you spent in each of these areas, and write that under "Total Cost," for each area on your character sheet. Then use the below table to find your "Basic Value," and record that on your character sheet.

Total	BV
32+	0
16-31	1
8-15	2
4-7	3
0	4
-4-7	5
-8-15	6
-16-31	7
-32-	8

Subroutines fall neatly into two categories: Avatar and Processor. Your Avatar is your digital image, the way

you appear to the rest of the internet. Your Processor is your 'brain,' and controls the way you think and react. Storing a Subroutine in memory, give you some ability, or (for negatives) restricts your ability to do certain things.

Avatar subroutines are the equivalent of what we would call 'physical abilities.' They modify the way the program interacts with other 'objects' in the digital world. You cannot take a subroutine twice, unless it says that it stacks. You can take a subroutine twice, each time for a different area, if it is specific to a single Function, or task. Here are some sample Subroutines:

Basic SHape

Bytes: 0(see text)

For no points, your program can look like anything (one fixed form). However, it can only have 2 *functional* organs of manipulation, and 2 *functional* organs of propulsion.

Each additional organ of propulsion or manipulation increases the cost along the following scale: 1, 2, 4, 8, 16... Each extra organ of propulsion decreases your Basic Speed Value by 1 (min 0), while each extra organ of manipulation decreases your Basic Signaling Value by 1 (min 0), and allows you to perform extra tasks.

Each fewer organ of propulsion of manipulation increases the cost along the following scale: -1, -2, -4, -8, -16... Each missing organ of propulsion increases your Basic Speed Value by 1 (min 0), while each missing organ of manipulation increases your Basic Signaling Value by 1 (min 0), and forces you to perform tasks at a slower rate.

Beam

Bytes: 8

Beam lets you create a multipurpose beam of data at will, emanating from a hand-like extremity. This beam has a DR of 2, has a range unit of 10 yards, and must be aimed. It can also be 'toned down' to be used for many things, such as carving your way through a wall, signaling or 'play-fighting.'

Clingy

Bytes: 2 or 4

A program installed with the *Clingy* subroutine lets you stick to any object, allowing him to climb even shear surfaces.

A 4-byte version of this is also available, allowing the installed program to cling weightlessly to other programs (great in a chase).

Resistance

Bytes: 4

The *Resistance* subroutine allows your to take less damage from certain types of hazards. When you buy *Resistance* chose a hazard from File 4: Adventures. Any time such a hazard would deal damage to you, subtract 1 from its DR.

Shifting

Bytes: 2

Shifting lets you alter your avatar's basic appearance, but not function, at will. This Subroutine does not change your capabilities in any way. A successful signaling check will allow your to imper-

sonate another program. This ability does not grant you security clearance.

Processor subroutines are more discreet. They rarely have an apparent marker, so it is hard to recognize a program who possesses one at first glance. Negative cost Processor subroutines are available, but they are uncommon. Here are some example Processor subroutines:

Affinity

Bytes: 2

Affinity means that your program is especially suited to doing a specific task. You treat the chosen Function as if it is easy (if it was hard), or as if you have already bought a level in it (if it was easy).

Non-Binary Mind

Bytes: 16

Non-Binary Mind means that you are largely immune to corrupted information and viruses, as they are not native in your coding, but it requires massive amounts of storage space for your interpreting program. Whenever you make a check against bad data or a virus, you get +2 to resist. Also, viruses never deal continuous damage to you. You can choose to enter a no-buffer state, where bad data and viruses do not affect you, but you cannot Signal. Viruses specifically wrote to affect non-binary programs still affect you, but they do not affect other programs.

Procedural

Bytes: 4

By installing *Procedural*, you change the way you solve problems. Since you can adapt more readily to new situations, you gain the ability to extrapolate. You can always ask the GM what the consequences of an action would be. The GM must tell you, although he cannot use information that your character does not posses in his guess.

Feel free to create new Subroutines, but GM approval is *required*. Negative cost subroutines are uncommon, but not unknown.

Traits are a lot like subroutines, but with one difference: they contain both a negative and a positive effect. You can have up to three traits, but they do not take up any memory.

No distinction is made between Avatar and Processor traits.

Some sample traits include:

Dedicated: You gain an effective level in one function, but take -1 to two other functions of the GM's choosing.

Hardened: You roll against Kinetic damage at +1, but your Speed checks are always "off-path."

Alert: You are always aware of you immediate surroundings, but take -1 on all rolls to spot things that are intentionally hidden.

Use these as rough estimates about how powerful a trait should be.

Functions

While your Personality Core in unchangeable, mostly, your Functions may come and go. Any part of your memory

not allocated to your Personality core can be used to store Functions. At any time that you have access to an interface and sufficient time, you may "dump" Functions and add new ones, as long as you have the right amount of Memory available.

In N/AI, Functions are the equivalent of "skills" in other RPGs. They are rough estimates of how good your character is at something.

Using functions is covered more fully in the next File, but the gist is this: the lower the Function rating the better (you must roll *more* than your Rating to succeed). Your Rating in a function decreases as you buy levels in it. Each level costs twice as much as the last, so the more levels you have your cost becomes exponentially higher. The following chart shows the cost of Easy and Hard Functions, by Rating:

Rating:	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Cost(Easy)	8	4	2	1	0	-1	-2	-4	-8
Cost(Hard)	16	8	4	2	1	0	-1	-2	-4

Each of the Functions detailed below has four things: a name, a difficulty (easy or hard), a Base (see next File), and a description of what a use of this Function can do, and what would be considered "off-path" (see next File).

Here is a (in no way exclusive) list of some Functions that character may wish to use.

Attack Program

Signaling: Easy

A successful use of this Function allows you to make an attack, using an attack program. Generally, this is taken concerning on specific type of attack program, so other types are Off-path.

Avatar Design

Processor: Easy

Avatar design is the ability to create an avatar that is visually or otherwise pleasing, and serves a role. By using the Upgrade Booth, this character can cause the program inside to change form, and can give her the advantage or otherwise of the Basic Shape subroutine. There is not off-path use of this Function.

Generating

Processor: Easy

This is the ability to create functional items at a Coding Terminal. Each success creates a major part of the object, and each failure sets the process back one day. Check once per day. Off-path uses of this Function would be used for creating things outside your specialty.

HAX

Processor: Hard

HAX is the trained art of breaking and entering, and moving around unseen. HAXOR (programs who use HAX), can use this skill to enter any Node where they would not ordinarily be allowed. Off-path uses include writing and using viruses, and spotting other HAXOR.

Influence

Signaling: Easy

A use of this Function can change another character's (usually and NPC's) opinion of the one using it. A success

means that the opinion has improved; a failure means that it has gone down. Installing this Function, you must specialize in one technique. Off-path use would be for a technique that you have not specialized in.

Netrunning

Speed: Easy

Netrunning is the skill of navigating the Internet. It requires a combination of navigation, searching, and actual speed. A successful use of this skill will allow you to reach a destination in half the ordinary time. Off-path uses include other types of navigation, searching for hidden portals, and traversing difficult terrain.

Niche

See Text: Easy

This group of Functions refers to whatever the program does in the digital world. Replace "niche" with "actuary," "security," etc. as applies. The GM determines what the Base is.

Open-Handed Combat

Signaling: Hard

Open-Handed combat allows the Program using it to deal damage in combat without using an attack program or other weapon. Off-path uses include grappling and avoiding attacks (Speed base).

Virus Use

Signaling: Hard

A successful use of this function allows you to use a virus offensively, or lay a trap involving a virus. Failure by 2 or more mean that you are instead affected by the virus. For more information on viruses, see File 4.

Record your Functions on your character sheet.

Programs

Although all characters are technically programs, they frequently use non-sentient programs as equipment and weapons. This subsection contains all the information necessary to equip a character.

Equipment programs fall into the following categories: attack programs, tool programs, one-shot programs, barrier programs, and boost programs. The difference between programs and equipment in other RPGs is that programs take up memory.

To buy programs, you must spend money and forfeit memory. You can always uninstall a program. Money in NA/I is the Bit. Your character starts with 2d4+2 bits to spend on programs. You do not have to spend it all.

Attack programs can be either close-range or long-range. Aside from standard programs, you can also buy attack barriers, which are reactive weapons (they fire

Name	Info	Cost:Mem
Unarmed	M0 1KN	N/A
Kinetic Hammer	M1 1d4KN	2:4
Code Sword	M1 2EL	2:8
Packet Pistol	R6 1d4EP	5:4
Packet Rifle	R12 1d4+1EP	6:8
Attack Barrier	M0 1S 2SL	3:2

when someone triggers them, generally

by attacking). See the below table for some basic weapons. Feel free to create your own weapons.

Tool programs are as varied as they are useful. They can be used for everything ranging from climbing a shear wall, to delicately cutting a virus-trapped barrier, to lashing a defeated opponent to a pole. There is no definitive list of tool programs, and the players should come up with tools that suit their needs.

One-shot programs are another interesting category. Many weapons fall into this category, such as some attack barriers and all splash weapons. Many programs that you buy will be one-shots.

Name	Kind	Cost:Mem
Hookshot	Tool	1:2
Wallhack	Virus	4:8
Charge	Tool	1:4
'Port	Tool	3:4
Link and Load	Tool	2:2

They tend to be cheaper, and they take up less memory. The following table covers a couple one-shot programs' cost, and memory uses:

Since not every one-shot is self-explanatory, I will explain what the above programs actually do.

Hookshot: Latches on to another object and brings you to it (If it is fixed), or it to you (if you are and it isn't).

Wallhack: This is a small circle that you slap on a wall, forming a hole all the way through. After five minutes it vanishes, leaving no trace.

Charge: This small cylinder can be used as an attack program, or can be used to demolish buildings or destroy daemons.

'Port: You buy a 'Port linked to a certain node or other location. When you use it, it transports you, all your stored programs, and anyone tied to you with a Linkord (cost: 1, mem: 1).

Link and Load: A Link and Load consists of a Linkord and a small program. You attach the Linkord to any object, and then activate the program, transporting the object it is tied to into your open memory.

Barrier	pro/int/hit	Cost:Mem
Mana	3/2/2	4:2
Attack	1/3/2	3:2
Speed	1/4/2	3:2
Full	3/1/3	6:8

Barrier Programs are the equivalent of armor. They will stop you from getting mauled by attack programs and viruses. An incomplete list of some barrier programs follows. See File 2 for information on how Barrier Programs work. Pro. stands for Protection, int stands for Intercept, and hit stands for Health. If the DR of an incoming weapon ever exceeds the Barrier's hit., the Barrier overloads and is destroyed.

Any program can be a boost program Boost program give you +1 or more to a specific type of task. Some allow you to use skills. You character should not be allowed to buy boost programs at the beginning of the game; they should be rewards for doing well.

After you purchase your programs, record them on your character sheet, and

you're ready to play!

	0	1	2–6	7	8
1	N	F	CF	CF	CF
2	N	N	N	F	CF
3	N	N	N	N	N
4	N	N	N	N	N
5	N	N	N	N	N
6	N	N	N	N	N
7	CS	S	N	N	N
8	CS	CS	CS	S	N

Difficulty	**Modifier**
Impossible	-4
Heroic	-3
Hard	-2
Tough	-1
Easy	+1
Trivial	+2
Cakewalk	+3
Effortless	+4

On Both	Use Best
On One	Add and divide by 2
Off Both	Add and divide by 3

File 2: Run Program

Playing the game

Your character is made, and now you need to know how to play. There are just a couple of basic rules. The first mechanic you need to know is the Check.

Checks

When your character is trying to do something, and the GM decided that they shouldn't automatically succeed, the player must make a check. First, he determines which of his Functions he uses (If more than one apply, or none apply, see below).

Now the GM decides difficulty. Use the following chart to decide what modifiers to use:

After figuring out which function to use, the GM determines whether this is the correct use of the Function, or if it off-path. Then, take the rating of the Function and the Base, and determine if the task is on-path or off-path for your function. If no functions are on-path *or* off-path, use your base value

If the task was on-path, average your Base and Rating, and round down. ROund up if the task was on-path.

Before doing anything else, check if you got a critical success or failure (see next chart). A roll that threatens a critical success or failure is an automatic success or failure. Whether it is critical is determined by a re-roll; a failure after a critical failure threat is a critical failure, and a success after a critical success threat is a critical success.

This chart shows critical success and failure. The row on the top is your target number. The column on the left shows your roll. A result of N is treated normally. If the result is F, it is an automatic failure. If the result is CF, it is a critical failure (roll again to confirm). If the result is CS, it is a critical success (roll again to confirm). If the result is S, it is an automatic success.

If the result was not an automatic or critical, proceed normally. Add any modifiers that apply to the roll onto your result. If that is equal to or more than the Target Number, you succeed! If not, you fail.

If two Functions apply to your task, determine which path it is on, and then use the following chart:

If not Functions apply, roll against your Base Value.

You should fill in the Target Number box for each skill, on-path and off-path. Then fill in the CF/F and CS/S boxes on all your Functions, for on-path.

Example: LNX wants to hack into the security network. He has a HAX Rating of 1, and this is definitely on-path. His Basic Processor is 2, and he has +1 to "sneakin' around" His HAX is better than his Processor, so he uses that. The GM decides that this is a hard challenge, so he gets -2 to his roll. He rolls.

If he rolls a 1, he automatically fails. If he rolls a 7, he automatically succeeds. If he rolls an 8, he may critically succeed.

Inversion Table

1st	New
8+	0
7	1
6	2
5	3
4	4
3	5
2	6
1	7
0	8

LNX's player rolls a 3. The net modifier is -1, so the result is 2. That is more than LNX's HAX Rating of 1, so LNX ducks past the Security Daemons easily.

Fighting

It's a dangerous internet. Someone has to keep it safe. Occasionally, a little conflict turns into a full-fledged fight. Now what happens?

Actually, fighting is not much different from making checks. Sure, there is a little extra excitement, but is basically a long series of conflicts. In N/AI, fights happen in 1-second rounds. You can do a certain number of things in each round. Each time you get your turn, once a round, you can do one of the following actions:

Move: You can move up to your Inverted Base Speed Value (Inversion table below). This called your Move Score. If you are running in a chase, make Speed checks instead. Speed checks are also needed to dodge past opponents, or to traverse difficult ground at full speed.

Fight: You can attack with or without an attack program. Make a Check against that weapon Function, as above. Your opponent's armor may stop some of the damage (see below), or they may dodge, or they may guard. Dodging is a Speed check. Guarding is a weapon check, like an attack. You cannot guard a gun or guard with a gun.

You must be within the range of a melee weapon (ranges given in meters) to attack. Ranged weapons take a -1 penalty for each range increment they are away from their target.

Splash weapons affect everyone that fall into their shape. The shapes are C (cone), S (sphere), D (disk), and L (line). The number before the letter is the radius of the shape.

Move and Fight: You can move up to half your Move Score and perform the Fight maneuver, but you take -1 to the attack.

Actions: You may do other things in combat, like grappling, or talking. Most actions take longer than 1 second.

Damage

Damage is almost a certainty if you engage in a fight. In NA/I, there is no "hit points" or other abstract health measures. Instead, damage is measured in how many hits you have taken.

Each time you are hit, you must do these things: Check Barriers, check DR, make a Coding Check.

First, you determine if any of your Barriers will help you. Roll 1d6—if it is greater than the int. of your any of your barriers, they protect you. If multiple barriers would apply, only the one with the highest int. does.

Type	**After Barrier:**
Blunt or **K**inetic	-1 to DR
Cutting	Keep DR the same
Piecing	+1 to DR
Shock	Non-injury hits only
Energy	See File 4 for more info more info on Energy

Divisor	**Barrier is:**
Normal	as normal
Limiting	-1
Penetrating	1/2

Then determine the DR. Read the weapon info. The first letter indicates whether it is Melee or Ranged. Then a number indicates range, in meters. Then (everything except splash weapons, see above) there is a number or a dice notation. This is the Damage Rating (DR).

After the DR there are two letters. The first indicates the type of damage (see chart for what to so with that info), and the second indicates the Divisor of for the Barrier. Consult the chart, and modify the pro. value of your Barrier (round down).

Subtract the new pro value of your Barrier from the DR (minimum 0), and then modify the DR as the chart says. Now you have your final DR, make a Coding roll, and subtract the final DR from your result. If you succeed, you don't take a hit. Butt, if you fail you take a hit.

A Hit is a cumulative -1 penalty to all Coding rolls. If you were hit, you must make a Coding roll, or else you suffered an Injury as well. Hits and injuries stack.

Each injury is a cumulative -1 penalty to all rolls until you have had time to recover. If you ever roll 0 or less on a Coding roll because of hit and injury penalties, you become Dormant.

Dormant characters are stable—for now, at least—but they cannot take any actions. If they get 0 or below on another Coding roll, they start Dying.

Dying characters do not have very long left. Every turn, add 1 to their Coding Base Value. If it ever exceeds 8, the program is deleted and existed only in memory. Any programs stored in their memory banks are deleted. If a dying character receives coding attention at any time, he can make a Coding roll to become Dormant again.

Characters recover from damage at a rate of 1 hit per day that they rest, or a skilled Generator can return them to full health immediately. If a character were Dying, his Coding Base Value returns to normal at a rate of 1 point per day.

What Programs do and don't have to do.

When running a NA/I campaign, you must remember that the PCs are programs. Programs do not need the same things as people. They all want to be happy, have the approval of their fellows, and to be materially rich. They do not have to eat, but they become uncomfortable if they have not processed data in a while. Programs begin to break down over time, so they prefer to be near each other, so they can rewrite damaged sections of code. They do not need to breath, drink, or do most things related to a body. As a general rule, must programs genuinely enjoy doing what they were written to do, rarely get bored, and like to contribute to a larger organism, namely, the Internet. Programs are more or less immortal, but breed slowly, and can die from violent death

File 3: Root

Running the Game

This section is for the GM. This is not to say that players cannot read it, but rather that the information given here is intended to assist the Gamemaster in his duties, and is of little consequence to the players.

The Internet

As GM, it is your most important and rewarding duty to supply the players with the game. You describe all the action, you create and run the world, you make the game feel *real* to the PCs. This also seems like a lot of work. That's true. It is. But, it is also the most rewarding aspect of the game to facilitate the game world.

It is not realist to expect the GM to keep track of every character in the game world. It is not even expected that you create nearly any of them as fully as the players make their characters. All you really need to know about the PCs' enemies is their weapon stats, relevant skill, modifiers and Coding BV. Here's what you do need to know:

1. The game is digital. This may seem like a no-brainer, but the PCs may need constant reminding that, since the world is inside a computer, things may be different here. Just make sure the players get the *feel* of the digital world—most things in shades of grey, contrasted by bright colored lights, and warm white glow. Everything there is a program or bit of information, so it should feel digital.

2. The world is a lot like ours. Now, after I just got done saying it's different, the Internet of NA/I is a lot like the real world. It has most of the same "laws," like gravity, and friction. But, the way that these laws work is different. There is no entropy, because there are no chemical reactions. If something looks like fire, it is only aesthetic-data can't burn. On the other hand, if something looks like fire, it's still probably dangerous, so watch out.

3. Everything has an equivalent. This is kind of the end all and be all of the same/different discussion. In a general sense, if something is in our world, an equivalent exists in the Internet.

Designing and Running Adventures

I will not give you step-by-step instructions, but I will give you a guide to good and bad structure. Here it is:

1. Let the PCs use their toys. If there is a HAXOR in the party, let him sneak past a couple of Daemons. If someone bothered to take Non-Binary Mind, throw a virus at him, and let him gloat as it does nothing. Don't design adventures to be stupidly easy, but let the PCs get a kick out of their toys.

2. Don't thwart the PCs. PCs shouldn't act on information their characters don't have. GMs shouldn't either. The villain should not know the exact stats of each PC and figure out how best to thwart them.

3. PCs are accountable. When a PC does something stupid, let them face the consequences. PCs are not above the law, so don't let them act like it.

4. Reward originality. If a PC comes up with a really good way to get around a problem, then let them. Never disallow PC action, just because you didn't plan on it going that way.

5. Balance rewards and challenges. If the PCs didn't really try hard, don't give them rewards for no reason. If they nearly died, and *just managed* to pull it off, give 'em more. No one should say you're doing it wrong, so reward superior effort.

6. A matter of taste. People have different motives for role-playing. Regardless of why they play, not *everyone* will like the same style of game that you do. Don't run a game as if you were playing one-on-one with yourself.

The five pitfalls

Nearly everyone I know who has GMed in any game has fallen for one of these, I included. Here are the five pitfalls of GMing:

1. GM Domination: Since you are in charge, it may be tempting to rule the show. Don't. It is your job to facilitate gameplay, not to dominate.

2. PC Domination: Many GMs have the opposite problem, and are too scared to boss their friends around. That's fine, but you need to stay in control. If PC's get out of control (in game or out of game), I authorize you to beat them back on track. But don't go too far, or else, you'll wind up with GM Domination again.

3. Dice Domination: This is a harder to recognize, but more destructive pitfall. In this scenario, the dice, and the rolling thereof, are the axis around which the whole world turns. Phrases like "Make a roll. You succeed. He misses. You miss. Make a roll." Fill your game. Character interaction is gone from your game. Include more description, and have more role-playing and less roll-playing fast, or your game will die.

4. NPC Domination: This is a weird one. I have fallen for this one frequently. It may be tempting to run NPC characters alongside the PCs, as fellow adventurers, but just don't do it. You cannot stay unbiased, and it just ruins the structure. You cannot eat your cake and have it too.

5. Story Domination: The GM has already decided what will happen. There isn't any room left for the players to make decisions. They are spectators in a novel, even an incredibly well-written novel. This might be you if you find yourself lying about dice rolls a lot.

FIle 4: Adventures

Hazards, Enemies, and the World

Every hero needs a test. Before you can become a hero, you must prove yourself be accomplishing heroic tasks, defeating despicable villains, and succeeding where no one could before. This File is all about the things that heroes must face.

Hazards

While adventuring, not every adversary will be a foe. The impossible conditions that you must endure to reach your goal are almost as heroic as a mountain of enemies. Here are some of the things you may face.

Falling from great heights

Falling is one of mankind's favorite dangers. A fight is so much better if the fighters must risk death with every step.

In game terms, the DR from a fall cannot be stopped with Barriers. The DR from a fall is equal to the height (in meters) divided by eight. Also, when you fall you take a number of hits equal to the amount you failed the Coding roll by.

Falling Objects

Falling things are just as much a hazard in NA/I as in the real world, and, although they are only programs, you can sometimes forget that when and an immense sign falls from its 30-meter perch onto your if/then statements.

The DR of a falling object corresponds to its weight and the distance it falls. The DR is equal to (the objects weight [in kilograms] divided by 45) plus (the distance it fell [in meters] divided by fifteen, and times 2), all divided by 3. Barriers protect against falling objects normally.

Disorder

Adding disorder to a program can wreak havoc, as the more its subroutines start to fail, the more they will keep failing. The spreading power of disorder is analogous to fire.

Intensity	Level
1	Random String
2	Randomizing Virus
3	Coding error
4	True chaos

The intensity of the Disorder is the DR that it has. On program per round, carried by someone exposed to disorder, must make a Coding roll (against 4), or become an intensity 1 source.

Recompile

Order can be as destructive as disorder, if used correctly. Recompile is damaging because it inserts its own order into another program's code.

Intensity	Level
1	Inserted string
2	Recombination
3	Logic probe
4	Recompiler

Intensity is the DR for the source. After taking a hit from a recompile damage source, the target automatically is injured.

Recompile sources counteract disorder sources of equal or lesser intensity.

Electricity

Electricity is just as dangerous in a digital world as in the real world. The only difference if, it actually is caused by a small disruption in the computer system. Electricity has two in-game ratings: charge and current.

Lvl	Charge	Current
1	Stun gun	Short circuit
2	Lightning	Shock paddles
3	Outlet	Power line
4	Transformer	Generator

The DR of an electricity source is equal to the charge + (the current times 2) all divided by 3. A program injured by electricity loses its next turn.

Bad Data

You're a program. Your life is simply a fluctuation of data and numbers. Bad data, for you, if identical to poison. There are two kinds of bad data: contact and process. Contact data just need to touch you to act, but bad processes need to be absorbed into you to act.

The rules for contact data are thus:

Lvl	Intensity	Concentration
1	Bad packet	0.01
2	Non-instruction	0.1
3	x/0	1
4	Module missing	10

The DR of contact bad data is equal to its intensity plus its concentration. Contact bad data lowers the pro. value (permanently) of Barriers it hits by 1.

Process bad data is almost worse. Once absorbed (as true data would be) and processed, it starts acting quickly, forcing a Coding Roll each round until it destroys its host.

Intensity	Level
1	Non-variable
2	Non-instruction
3	Non-reference
4	Non-module

A bad process's DR is equal to its Intensity. It stays in the system, causing a Coding roll every round until a number of consecutive successes is rolled equal to the Intensity of the process.

While affected by any kind of bad data, a program may suffer from a wide variety of effects, ranging from pain to paralysis.

Viruses

The worst thing that can happen to a program is to get infected with a virus. A virus has usually one mode of travel—air, fluids, or contact. If a virus hit you, you need an anti-viral quickly, or the virus will amplify in your system, killing of seriously injuring you.

Lvl	Intensity	Contagion
1	Low threat	e-Mail
2	Bad news	Trojan
3	You're in trouble	Worm
4	Oh, SH-!	Attachment

The Contagion rating is the penalty to a Coding roll to resist infection. The Intensity rating is the penalty to resist amplification. After a roll to resist amplification (roll once an hour) is failed, the virus has a DR equal to its intensity. Roll once a round, or until the subject dies, or a number of successes are rolled equal to *twice* the DR of the virus.

At any time during this, unless the initial roll is successful, the subject is a source of the virus. At any time during this, a single dose of an anti-viral will save the subject, as will a "hard reboot," where all memory but the personality core is dumped, and the program is started from scratch. A hard reboot costs 12 bits. A dose of anti-viral costs 8, and takes up 4 bytes of memory.

Enemies

Great heroes need great foes. This section is all about the enemies that your GM may throw at you. Please, don't read this unless your GM tells you to. You'll just ruin it for yourself.

The format for the enemy entries is as follows:

Name
Main attack/Target Number (Mods)
Coding BV (Mods)
Description

Here are some enemies:

Anomalous

Servile

M1 1d4SN / 3 (+2)

2 (+2)

A Servile if the base unit of the Anomalous horde. They carry sharp blades and frequently wear jerry-rigged attack barriers they can activate at will. A Servile is filled with an all-consuming hatred and is kept from destroying the Net only by their fear of their Warlock overlords.

Warlock

M2 1d4-1KN / 5 (+0)

5 (+0)

A Warlock is a scheming Anomalous, who uses his programming skill to create "spells," i.e., one-shot attack programs and tools. Warlocks seek to destroy the Net, and the Sent, but only when it best suits them to do so.

Anarchios

M0 1d6E(D)N / 2 (+1)

3 (+1)

The Anarchios is an ancient type of program, born of disorder. It appears as a great multicolored florescent behemoth, upon whose claws glint an Intensity 1 disorder. It can also spit an Intensity 2 disorder, up to 10 meters.

Attack Barrier

M0 1S 2SL / 3 (+0)

4 (+1)

Occasionally, an abandoned Attack Barrier bonds with a stray code fragment in exactly the right way. An Attack Barrier is naturally hostile, and flies towards his

opponents (i.e. Everyone) with gust, and attacks

Axionus

M0 1d6E(R)N / 2 (+1)

3 (+1)

The Axionus is birthed from recompilers as old as the world. They are the sworn enemy of the Anarchios, and have much the same capabilities, only it is Recompile that glints on their claws and waits in their mouths.

Bromus

Bromus Construct

M1 2ES/3 (+0)

2 (+1)

A Bromus Construct is a lesser form of the Bromus servants. They are shaped from spare loops and statements. Each is armed with a Code Sword, and is possessed of only a rudimentary intellect.

Bromus Tasker

R3 1KL/3 (+1)

4 (+0)

A Bromus Tasker is created to perform small menial tasks for the conclave. They have a small, scheming mind, and see things only in terms of how to obey their orders. Each carries a Muala Pistol, with an intensity 2 bad process.

Bromus Mindservant

M0 1KN/4 (+0)

2 (+2)

A Bromus Mindservant is meant to do two things: congress and survive. They enter a *congress* with each other, and can communicate telepathically with other Bromi. They relay orders from the Masters to the Constructs and Taskers.

Bromus Master

M0 1ES/3 (±1)

5 (+0)

A Bromus Master forms spontainiously. If they depended on each others' orders to make more Masters, none would be made. They do the thinking and ordering for all Bromi. Their Code Knives have the Tasker's poison on them, and they have absolute authority over all lesser Bromi (aka: all of them). They are insanely jealous, and tolerate no dissent. If another master has given a minion contrary orders, the Master is likely to destroy the minion.

Bromus Breeder

N/A

2 (+2)

A Bromus Breeder exists only to propagate the race. They belch out broken fragments of code intermittently, and Taskers weave these into fully formed Bromi on a Master's command. When killed, a Breeder calls any nearby Bromi to its defense, telepathically, and releases and Intensity 2, Concentration 1 contact bad data in a cloud.

Cookie Monster

R6 3EP / 4 (+1)

3 (+1)

A Cookie Monster is a fearsome opponent. They resemble furry grey blobs that crawl along the ground at a move of 1. The Cookie Monster's strength, however, comes from his adaptation. Each time a Cookie Monster hits an opponent; he gets a cumulative +1 against that opponent. These bonuses last 1 day.

Daemon, Security

R6 1d4EP / 2 (+1)

3 (+1)

A Security Daemon is the most common security measure taken by people with something to protect. They are mindless, inscrutable, and carry Packet Pistols. A Security Daemon has a rating of 2 for spotting things, and gets +1 to it.

Daemon, Antibarrier

N/A

4 (+0)

The distant cousins of Security Daemons, Antibarrier Daemons are made to take down coded barrier mazes. They have a HAX skill of 1. Antibarrier Daemons cost 8 bits and get +1 to their dismantling rolls for every other Antibarrier Daemon working on the same barrier.

Ecto

M0 1E(D)N / 3 (+0)

5 (+1)

An Ecto is a small being, appearing as a faint, transparent orange being. An Ecto has two grasping tentacles, and three leg-type tentacles. When it grasps an enemy, disorder bubbles in its tentacles. This disorder does not spread, however.

Growth Adept

M2 1EL / 2 (+1)

1 (+3)

A Growth Adept is a bizarre, but dangerous foe. His avatar appears covered by glowing tumors. They love each other's company, and are almost always found in groups.

Each possesses an *antihit* which gives them a +1 to hit rolls (included above). At will, they can cause their *antihit* to fly to another Growth Adept within. When they receive an *antihit*, they can make a hit roll. On a success, they are temporarily flooded with adrenaline, giving them +1 to all rolls for 1 round.

Load Barrier

M0 1SN / 3 (+0)

4 (+1)

A Load Barrier appears as a stylized attack barrier, floating in mid-air. They are naturally friendly and intelligent. When they find a suitable host, and are convinced with an Influence roll, they join with a host and become a Mana Barrier.

Mimic (Trojan)

M0 1SP / 3 (+0)

5 (-1)

A Mimic (also called a Trojan) pretends to be some innocuous item, and then pounces on its hapless victim. His

claws contain a Intensity 3, Contagion 2 virus. Avoid at all costs.

Moro

M1 2KN / 2 (+0)

2 (+1)

A Moro is a dangerous foe. Matte black, enormous humanoids, Moro have luminous, mossy lines crisscrossing in intricate patterns on their frame. They have unknown long-term plans, but have been known to pound those who interfere. They usually have armor.

Ninja, Code

M1 2EL / 1 (+2)

2 (+2)

A Code Ninja is the final evolution in Ninjutsu. In the Pre-web days, at least one Ninja transferred his consciousness into fragments of code. Some accounts believe these to be the first sentient programs. They are trained in Code Sword, Unarmed Combat, and HAX, as well as Ninjitsu.

Opaline Wurm

M0 3SL / 3 (+0)

3 (+1)

The Opaline Wurm is universally feared by Netrunners. She writhes through every corner of the Net simultaneously, waiting to bite from the chaos. Some believe that the Wurm is a fragment of pure Chaos born from an electrical impulse. She can swallow an opponent whole after making a bite attack, by making another attempt at -1. The Victim can escape with a Speed check, but otherwise is sucked into the Wurm's gut. Inside, a victim is subject to a Charge 1, Current 2 electrical impulse each round. The Opaline Wurm is 12 meters long.

Pearl Python

M0 3SL / 4 (+1)

2 (+1)

The Pearl Python is a suspected relative of the Opaline Wurm. They are far more common; however, and less dangerous. Each one flies through the air at a more of 3, and bite or constrict their foes. Their bite holds an intensity 1 bad process.

Sent

One Sent
M1 1d4KN / 2 (+3)

2 (+2)

The One Sent are the simplest of the angelic Sent. They commit their utmost energies to defeating the Anomalous. One Sent are bronzed, naked, female figures with blazing eyes, armed with Kinetic Hammers and Attack Barriers.

Three Sent
M1 1d4KN / 3 (+3)

2 (+1)

The Three sent are the only male Sent. They are the reconnaissance of the Sent, and constantly monitor the activities of all power in the Net. They are also bronzed, naked males, also armed with Kinetic Hammers and Attack Barriers. They have Influence, and other social skills, but their appearance is often

off-putting. So, many Three Sent are Shifters.

Four Sent
2(M1 1d4KN) / 2 (+3)

2 (+2)

The Four Sent are the most Warlike of the Sent. They are four-armed versions of the One Sent and carry two Kinetic Hammers. Their attack Barriers have +1 meter of radius, so they are dangerous in a fight. Four Sent are incredibly hostile and undiplomatic, believing in no "moral grey area."

Spamm

N/A

0 (+3)

A Spamm is a harmless but irritating creature that is difficult to get rid of. They release useless bytes of information once a round, and these randomly download themselves into an opponent's memory. These pieces take up 8 bytes of memory apiece, and require a Coding check at -1 to remove.

User

<Any>

<Any>

A user is an actual; Human who has entered the Net for some reason. They are so scarce as to be nonexistent, but they can make powerful allies or villains. Since Users are so unique, stat each one out as if it were a PC.

Uzkim

M2 1d6+1SL / 3 (+1)

3 (-1)

The Uzkim are suspected relatives of the Morro. They are of similar size and build, but their bodies are chiseled out of massive quartz blocks. Glowing symbols cover their bodies, and no two are alike. Uzkim do not wear armor, but are covered by plain whit moss, giving them a "furry" look. They carry massive swords, and become soulless husks if ever separated from them

Viruses

A virus is not technically an enemy, in and of itself, but they deserve special treatment. Here are a number of Viruses, contagion and intensities, and special effects:

Men-mix

C2 I1

Rather than damage, Men-Mix causes one program to become renamed, per hit that would be dealt. The effected program is still there, and still functions, but your character can no longer recognize it. After no more programs have their true names, the virus leaves the system. When PCs rediscover their programs' names, they work as normal.

Cost: 5 bits.

Brainfeed

C1 I3

When a character starts to die because of Brainfeed, begin adding points to the Base Processor, instead of the

Base Coding. If your Processor reaches 9 or more, you die normally. Processor damage dealt this way lasts for many days past when it should have healed.

Cost: 7 Bits:

Quikill

C4 I1

Quikill is the fastest-acting virus known. Once infected, the victim must immediately begin rolling to prevent amplification. Each time the victim would take a hit; he is also injured (no roll). Once he begins to die, add 2 points a round instead of 1.

Cost: 8 Bits.

Innoculo

C1 I2

Innoculo is actually a category of viruses, which are weakened forms of more deadly viruses. Only a single success is needed to rid your system of Innoculo. After surviving Innoculo, you make all rolls against the virus it was based on at +2.

Cost: 4 bits.

Hard Reboot

C0 I4

Hard Reboot must be intentionally injected to infect a victim. Once inside, it immediately amplifies (no roll) and acts like Quikill, but does not kill the victim when she reaches 9 or more. Instead, all programs and functions aside from the personality core are dumped. The victim falls dormant for 2d6 days, and wakes up with only rudimentary memories.

Cost: 12 bits

Xot

A Xot is a small mote of code, around the size of a coin, that is of immeasurable value to some. Xots are intelligent and have varying aims. All Xots, however, seek out a program to partner with.

In Game terms, a Xot is a piece of equipment. It cannot be bought, and takes up 1 byte of memory to install. You cannot install more that 1 Xot.

When installed, a Xot is, in some ways, a one-shot program. However, as long as you have it stored, it can share all its knowledge with you, and communicate freely. Most Xots have at least one Knowledge Function.

When you use a Xot, it blasts forth, and becomes an independent program. It is no longer in your memory. Each Xot is unique, built on 1 bytes, , has a natural attack, but has ho organs of manipulation. They all appear as small motes of light. The Xot makes attacks of its own free will, but you may give it directions before you release it.

A Xot's natural attack is M0 1d4+1E(R)L. Most have a Move of 4.

After the combat is over, you must persuade the Xot to rejoin you, as it is a state of euphoric fulfillment.

Source Code: Character Sheet

Illustration/Description

Name:
Generation:
Character Story:

Program Niche:
Memory Capacity:

	Bytes
Personality Core	
Main Functions:	
Subroutines:	
Traits:	
Functions:	
Programs:	

Main Functions:

Coding		Signaling		Speed		Processor		Perception	
Total	BV	Total	BV	Total	BV	Total	BV	Total	BV

Main Functions(Cost):

Subroutines

Name	Bytes

Function	Bytes	Rating	Mods	CF/F	CS/S

Programs (Memory):

Attack Program Name	Type

Afterward

Well, people, I did it. I, William Prahl, who has tried and failed to complete RPGs over the span of months, have made an RPG. In 24 hours. I sat down, and I wrote it from start to finnish in 24 hours. I will convert this to a PDF and be done with it. I hope that someone will play it. Seriously. I mean, I'll probably play it. But, I would really like this game to make someone else happy, as well. That's it. I managed to slam out 24 pages in 13 hours.