Moral Fiber Optic

By
Dave Michael

Time: One session, two hours

Ingredients:
Ancient
Emotion
Glass

Introduction

In ancient times, we lived in the copper. Everything was like climbing uphill. Now, in this ancient society, we live in glass. The fiber optic of today is a miraculous achievement!

For all our modern conveniences, life in the Computer is still a challenge. Every program is being pulled in many opposite directions. Surviving is a challenge, much less succeeding in this environment. Long ago, a lone hero defeated the program that was oppressing the Computer. Now, two powers have risen to take its place, The Arches and the Orientals. Arches want to provide order to society, Orientals fight to maintain the rights of each individual program. The situation has been escalating, both sides want to change the way the Computer is run. What side are you on, how will you make your way?

What you need to play?

- This rule book
- Paper and pencils
- One six sided die (like the dice used in craps)

Characters

Your character is a program in the Computer. With the advances in technology, the Computer has become a tangled, urban sprawl.

Creation:

Name – Pick a name, write it down.

Type – You might be an Application, Utility, Middleware, Game, Security or some other kind of program, decide and write it down

Description – Describe what your character is like and any other important details, write it down

Properties – You get a 1 (Worst), 2, 3, 4 and 5 (Best) to put on each of five properties, they are:

- Access – How secure is your program, how much permission do you have in the Computer? In the computer, this represents, strength, prowess and endurance.
- Input – How efficiently can your character recover data from keyboards, mice, microphones, networks, files, other programs, etc. In the computer, this represents your character's social standing and ability with other programs.
- Output – How well can you send data to files, screens, speakers, users, peripherals, other programs, etc. This is a program's purpose. How well do they server the User and Computer?
- Process – This is how well your character can process the data. This represents a game's AI, Application's usability, etc. In the Computer, this is analogous to your knowledge and skill of the world inside the Computer.

- Storage – This is a measure of how many resources your character has access to in the Computer. In the Computer, this is equivalent to wealth and income.

Traits

These are features of your program that make it different or special from other programs. Every Trait you take increases your Priority by one (It starts at one for every character). Priority is used to determine what order players take their turns. Higher Priorities go later in the turn.

- Hardware Level access - Using this Trait is the same as playing a Process Trait equal to the Opposition's Process
- Multimedia - Using this Trait is equivalent to using an Output Property equal to your Opponent's Output
- Optimized Decryption – Using this Trait is the same as playing a Storage Trait equal to the Opposition's Storage
- Optimized Encryption – Using this Trait will cancel the opposition's Storage Trait
- Predictive Coding – Using this Trait is equivalent to using an Input Property equal to your Opponent's Input
- Proprietary - Using this Trait cancels the opposition's Output Property
- Spaghetti Code - Using this Trait cancels the opposition's Process Property
- Sploit - Using this Trait is equivalent to using an Access Property equal to your Opponent's Access
- Standards based - Using this Trait cancels the opposition's Access Property
- Strong Validation - Using this Trait cancels the opposition's Input Property

Faction Score

Use the methods listed below to determine your starting Faction Score with each faction.

Advanced=Input, +2 for Predictive Coding, -1 for Strong Validation
Black Hat=Access, +2 for Sploit, +2 for Standards based
Hierarchy=Process
Legacy=Input, -2 for Predictive Coding, +1 for Strong Validation
Object Oriented=Process
Proprietary=Storage, -1 for Optimized Decryption, +1 for Optimized Encryption
Standard=Storage, +1 for Optimized Decryption, -1 for Optimized Encryption
System=Output, -2 for Multimedia, +1 for Proprietary
User=Output, +2 for Multimedia, -1 for Proprietary
White Hat=Access, +2 for Sploit, +2 for Standards based

Goals

Your character needs to set a goal. Every goal has two parts: support and protest.

The Goal you select to support should be aligned with a faction. And that Goal will be completed once your Faction Score with that Faction is ten (10).

The Goal you select to protest should be aligned with a faction. And that Goal will be completed once your Faction Score with that Faction is under the program's Property that the Faction focuses on.

You are ready to start playing!

Example:

Cheryl wants to make a program called Sprite. That is its name, and she decides it is a Game program. For description, she writes, "Sprite is the program responsible for displaying and animating graphic sprites to the User." For Properties, she decides on Access: 1, Input: 2, Output: 5, Process: 3 and Storage: 4. Finally, she picks the Hardware-Level access Trait. She has a Priority of 2 and Faction Scores of: Advanced=2, Black Hat=1, Hierarchy=3, Legacy=2, Object Oriented=3, Proprietary=4, Standard=4, System=5, User=5, White Hat=1. Her Goals are to Support Advanced and Resist Legacy factions.

Environment

To the programs inside the Computer, it is not clear if they are actually in the memory space of one computer or many. They are also completely unaware of Real world events to the point where many programs are not even sure that there is more than one User. Time essentially stops when the Computer is powered off in real life and the Programs resume their daily lives unaware when it is powered back on, everything exactly identical to how it was before time stopped.

Also, the OS, firmware, ROMs and the hardware itself forms the non-anthropomorphic features of the virtual world that the programs live in. A fiber optic cable might be a street or highway, an OS file might be a traffic light or the sun, moon and stars, or a flash thumb drive might be an Airplane, jetting the characters off to distant lands. Below are some common environments every Computer will have.

In this anthropomorphized world, Programs are people just like you and me. Use today's technology as a reference, the only exception being that Programs do not use computers. They use typewriters, cell phones, ATMs, even watch TV, but there are no virtual PCs or mainframes.

Archives

In the distant, remote storage of the Computer, programs live contented lives, free from most illegal operations that plague central storage. In the Computer, this is the equivalent of the suburbs.

Background

In the Background of central storage, the real work happens. Drivers, Operating System, Utilities and other programs do their jobs with little or no input or output from the User. This is similar to the Industrial sector of our real world, filled with factories, warehouses and processing plants of all kinds.

Buffer
The Buffer is where programs wait to do their work. This is the equivalent of inner city residential areas. High-rise apartment complexes, loft apartments and other such urban housing.

Foreground
All of the little things that people do everyday happen in the Foreground. But to the programs, it is like the commercial/municipal district, with banks, offices, police departments, fire departments, schools and other institutional buildings.

Heap
Here, small, specialized transactions take place. In the world of the Computer, it is like the specialty shops and mom and pop stores you find downtown.

Peripheral
These devices are so remote, many illegal instructions do not even target them. These remote places are like the rural farmlands and remote areas of our modern world.

Pool
In every city across the world, there are parks, town squares, plazas and crossroads that serve as a meeting spot and gatherings of that city's inhabitants. The Pool serves this same purpose in the Computer.

Queue
The Queue enables programs to move from one location to another in an efficient and orderly manner. In fact, in a manner very similar to real life subways.

Stack
The Stack is a storage location where large amounts of resources can be secured. In terms of our modern world, think of them as large department stores or chain retailers.

Stream
The Stream allows programs to move from one location in storage to another at their own pace and schedule. Think of it as the Computer version of highways.

Thread
Programs use the thread to get from location to location within the local area of storage, like modern streets.

Politics

Many programs go about their daily business, oblivious to the political struggle that is shaping their lives. Two major factions fight for dominance, Arches and Orientals. They disagree on their very core beliefs, they will never be able to

meet eye to eye on any critical issues. While other factions fight to advance their own agendas and maintain independence from the two central powerhouses. While many programs are unaware of this, most of their superiors quickly align themselves with one or more factions as a matter of self-preservation. So, even if they are unaware of it, their lives are still influenced by this heated political struggle.

Factions

Advanced – The Advanced faction (or Addies) rejects old technology in favor cutting edge advancements. They embrace these advancements, even if it means incompatibility with older technology. **Focus:** Input/**Opposition:** Legacy faction.

Black Hat – Black Hat programs feel that they can use their knowledge and security access to punish. Black Hats attack those that they feel have betrayed their own personal philosophies. **Focus:** Access/**Opposition:** White Hat faction

Hierarchy – The Arches are one of the strongest factions in the Computer. They thrive on order and strive to organize and categorize everything it encounters. The basic philosophy is that every program must fill a role and programs of certain roles are superior to others of the same role. They oppress and manipulate any who do not conform to this worldview. **Focus:** Process/**Opposition:** Object Oriented faction

Legacy – The Leggoes are an Ancient faction of programs that fight for the rights of old programs and hardware. They embrace new technology, so long as it maintains compatibility with existing programs and hardware. **Focus:** Input/**Opposition:** Advanced faction

Object Oriented – The Orientals believe in the rights and freedoms of each individual program. They have naturally garnered support from those oppressed by the Arches. Unfortunately, this lot is no better. They are fanatical and require a cult-like devotion of all of their followers. The most devoted of Orientals are trusted with missions of resistance. **Focus:** Process/**Opposition:** Hierarchy faction

Proprietary – Props are concerned with maintaining control over Proprietary standards. They believe in the rights of each group to create their own standards. And that once those standards are established, no outsider has the right to alter that standard. **Focus:** Storage/**Opposition:** Standard faction

Standard – Dards are obsessed with conforming as a group to a unified standard. Any non-conformity is frowned upon. **Focus:** Storage/**Opposition:** Proprietary faction

System – Syssies feel that the priority of every program is their allegiance to the Computer, almost to the exclusion of the User. **Focus:** Output/**Opposition:** User faction

User – Serfs believe that the sole purpose of programs and the Computer is to server the User. **Focus:** Output/**Opposition:** System faction

White Hat – White Hats use their knowledge of security and illegal operations to protect the Computer and User from themselves. **Focus:** Access/**Opposition:** Black Hat faction

Narration and Conflict

Most of the time, players will use the Narration rules, with the GM framing the scene and the players portraying their character's roles in the scene. Occasionally, two characters or factions will have opposing goals in the current scene, at that point, a Conflict ensues.

Narration

- The GM describes the scene as the characters perceive it, at a minimum this information should:
    - Be consistent with previous events and/or the character's Properties/Traits
    - Include a location
    - Anything the characters can see
    - Anything the characters can hear
    - Anything the characters can smell and potentially taste
    - Anything the characters can feel with their sense of touch
    - Anything that the characters would know about what the NPCs are doing
    - Any clues that the characters vitally need to know for the story to continue
- The players should describe what their characters are trying to do, including:
    - What your character's intent is.
    - What your character is doing to accomplish this intent
    - What objects your character is using to accomplish this intent
    - What your character is saying to accomplish this intent
    - If relevant, include what your character is feeling/thinking about this intent
- The GM and player determine if this task has a chance to fail. Character conversations, trivial challenges and anything that the player and GM consider a normal or logical outcome
- If the player and GM feel that there is a chance of failure and there is no conflict. Determine what Property applies to this action and roll a die and roll under that Property to succeed
- The GM decides if there is a Conflict. If there is a NPC that is opposed to a character's actions, the GM will announce a Conflict. At this point, players can declare a Conflict between each other as well
- The Judge and player determine how successful the attempt was and create a final description

Example:

The GM describes Sprite at the Apartment of her co-worker, Joey Stick. They are talking about work, so the Narration rules are used. Later, Sprite has to cross the Street. Narration rules are used. Finally, Sprite has to lift three bags of groceries, The GM decides Cheryl should roll against Sprite's Access, since it is a physical task. She rolls a 2, so they decide she drops the groceries on the way up the stairs to her apartment. What a mess!

Conflict Resolution

- The GM announces what factions have a stake in the conflict
- Play continues from lowest Priority to highest
    - In the event of a tie, start with the player with the lowest Priority on the GM's left and continue to their left
- The player determines what Properties or Traits to use and what faction they are using it for
    - The higher the value of the Property played, the more Faction Score that is at stake
- The player then narrates what actions they are taking that coincide with the Property or Trait used
- If the player uses a Property, add its value to the Commit Value of the faction they are supporting. If using a Trait, follow its instructions.
- Play then proceeds to the next lowest Priority. Play proceeds to that player's left if they are tied with the current player.
- That player has three choices:
    - Execute – This choice represents letting the other program execute unopposed. Essentially removing the character from the Conflict. This would be equivalent to Folding
    - Ack – The player commits their character to one side of the conflict and plays a Property or Trait of equal effect. Playing a Trait that reduces the opposing Conflict Value to your level is equivalent to Ack'ing. Players can Ack if they have no Properties or Traits left and want to remain committed to the Conflict. This is similar to Calling a bet
    - Req – The player wants to escalate the conflict, not only do they match the opposition's value, but exceeds it by some amount. Playing a Trait that reduces the opposing Conflict Value below your level is equivalent to Req'ing. This is like Raising the Stakes
- The new player narrates activity that corresponds to their choice and the Property or Trait played
- Play continues with each player Executing, Ack'ing or Req'ing in order until the highest Priority is reached, then the lowest Priority resumes.
- Once every player has either Ack'ed, Executed or run out of Properties and Traits, the Conflict is over. The side with the highest total Conflict Value wins the Conflict and the GM narrates the effect.
- Players gain and lose faction based on the table below

Faction at Stake	
Property	**Faction**
1 or less	+0
2 to 4	+1
5 to 7	+2
8 to 10	+3
11 to 14	+4
15 or more	+5

    - Execute – Any player who performs an Execute, loses Faction Score according to the table above based on the highest Commit Value they played before Executing
    - Ack, committed to losing faction – Player loses Faction Score with the faction they were supporting equal the value on the table above for the Commit Value of the opposing faction. Player loses the use of the Property that the other faction committed the most value to during the next Conflict.
    - Ack, committed to winning faction – Player gains Faction Score with the faction they were supporting equal to the value on the table above for the Commit Value of the opposing faction. Player loses Faction Score with the faction they were opposing equal to their Commit Value
- The Conflict is Resolved!

Example:
Sprite takes a mission to get a fellow Addie on the city council. When Joey finds out she is doing this, he starts to work against her and a Conflict starts. Joey's Stats are Access: 4, Input: 5, Output: 3, Process: 1 and Storage: 2. And he has the Standards-Based Trait. He has a Priority of 2. Joey starts the conflict and decides to go easy on Sprite at first, he plays Process 1 and narrates his character bad mouthing the candidate at local bars. Cheryl has three choices, she decides to fight for her faction and Req's with Output 5, recruiting all her friends at work to hand out fliers with her. Joey Executes so he does not look like a jerk at work. Since he only Committed 1, he does not lose or gain any faction, but Sprite loses 2 points of Legacy Faction and gains no Advanced Faction.

Endgame

Once any character meets their Goals and any current Conflicts are resolved, the Session is over. You can play again with the same characters if you like, but you must then select different Goals.

Gamemastering

Once the players have selected their Goals, it is important to start seeing how they fit in with the factions selected. They might be suited to day-to-day tasks, special missions, recruiting or even espionage of opposing factions.

If the character's actions are motivated and in some way productive, do not interfere with their progress, just react to what they are doing and let them tell part of the story.

Otherwise, try to involve the characters in either the designs of one faction or the other. Cue in on the player's desires and find a role for them to play that affects one faction or another.

Here are some Do's and Don't's:

Do:

- Be creative
- Involve the player's goals
- Work with the player's creativity
- Give the players the information they need to succeed
- Use the Standard rules for character creation
- Decide realistically what each NPC does and says
- Decide and describe which background settings are being used or ignored (and which elements of that game world are being used or ignored)
- Communicate all of these decisions
- Have fun!

Don't:

- Assume
- Tell players what their character does
- Create scenes with only one solution
- Cast the players as extras
- Punish players
- Play favorites
- Worry, it is just a game

Advancement

If a program has won a Conflict and is in a location that has access to source code or patches, they can patch or upgrade. This advancement allows a character to change the Property values. The Properties must still have each value: 1, 2, 3, 4 or 5. Programs can add or remove one Trait, adjusting Priority accordingly. These changes to not effect Faction Scores, but does change Priority.

Example:

After succeeding at the election Conflict, Cheryl decides that Sprite needs the Multimedia Trait and increases her Priority to 3.

Enjoy the game!

Moral Fiber Optic Character Sheet

Name	
Type	
Description	

		Portrait	
Property	**Value**		
Access			
Input			
Output			
Process			
Storage			
		Priority	
Traits	**Description**	**Traits**	**Description**

Faction	**Value**	**Faction**	**Value**
Advanced		**Proprietary**	
Black hat		**Standard**	
Hierarchy		**System**	
Legacy		**User**	
Object Oriented		**White hat**	

Goal	
Support	
Oppose	