

***APPROACHING INFINITY* MODDING GUIDE**

CONTENTS

OVERVIEW	4
AN APOLOGY	4
DEVELOPER'S INTENTIONS.....	5
MODDING AND UPDATES	5
MODDING IMAGES.....	5
MODDING SOUNDS	5
MODDABLE TEXT FILES	6
AddImg.txt	6
AchMas.txt	6
Armor.txt.....	6
ArtDesc.txt.....	6
ArtItem.txt.....	7
BankerPlanetUses.txt.....	7
BossStarters.txt , BossShipNames.txt	7
BunchaNames.txt	7
ChemicalStarters.txt , ChemicalEnders.txt.....	7
Commod.txt.....	7
DeviceFile.txt.....	8
DiseaseStarters.txt , DiseaseEnder.txt	9
Economic Events File.txt	9
Effects.txt	9
EngEne.txt.....	10
GreekLetters.txt.....	10
HandWepEnergies.txt.....	11
HandWepnPara.txt	11
MeleeWeaponEnergies.txt.....	12
MonDesc.txt	12
MonsterFile.txt.....	12
OfficerFirstFemale.txt , OfficerFirstMale.txt , OfficerLast.txt	16
OfficerSkills.txt.....	16
PDSWs.txt.....	17

PlanetNames.txt	17
PartNames.txt.....	17
Quests.txt	17
Races.txt	21
Randoms.txt	22
SA - amount.txt, SA - does.txt, SA - output.txt, SA - size.txt, SA - what.txt, SA - when.txt.....	22
ScanEne.txt , ScanKinds.txt	23
SheEne.txt , SheKind.txt.....	23
ShipNames.txt.....	23
SHIPWRECKENDERS.txt , SHIPWRECKSTARTERS.txt	23
Sounds.txt	23
StartingShips.txt	23
suits.txt.....	25
Terrains.txt	25
TrackList.txt.....	26
WepParadigms.txt.....	26
WhereYouKeepIt.txt	27
APPENDIX 1: Monster Powers	28
APPENDIX 2: Quest Requirement Numbers, Amounts, and Modifiers.....	28
All Requirement Numbers.....	29
Requirement Amount.....	29
Requirement Mods	29
APPENDIX 3: Triggered Events.....	30

OVERVIEW

The vast majority of Approaching Infinity's data is stored in plain text files. This means they can be modified. The end users ('you!') can't add new core functionality, but they can create their own images, terrains, ships, monsters, quests, etc. In many cases, the length of each file is determined at startup, so you can add new planet names and they will automatically be available. Some files have a number at the top, which is how many of that thing there are (most notably monsters, effects, and devices).

Please note that Approaching Infinity uses an internal security system to monitor your changes. If you make changes to the fundamental attributes of important things (like a monster's hitpoints or a quest's rewards) it will disable achievements. This is meant to allow you to freely modify the majority of files, while still giving some purpose and challenge in returning to the main game.

On the other hand, if you just add or modify names of things or words that are 'said' to the player (such as translating quest text to another language), these sorts of changes will NOT disable anything. Change words: OK. Change numbers: Red Flag.

Each text file has a specific template. Some are simple lists of words. Others define complex entities like monsters and starting ships. Many of the files have 'line headers'. These tell what each line does. They have a specific length for best results. Do not modify the headers.

Another thing to note is that if a line begins with the character '^', then that line is ignored; it is a remark ('REM'). Using this, you can leave yourself notes within the files.

A Very Important Note:

A totally blank line in one of these files will cause the game to stop reading. In the event that it is loading something very important, like skills or effects, this will almost certainly cause an error. So if you want to add space, use the REM character '^'.

You can also keep notes within the files by writing text after any numbers. Specifically, on any line that requires a number, you can place a space after that number and then write text. Only the number will be loaded. But be sure to leave a space or else it might be considered a hex and cause unexpected results.

Example:

```
Power...:51 ; poison!
```

Only the number '51' will be read by the program. The rest will be ignored.

AN APOLOGY

Many of the internal systems developed over time, with only a vague idea and short-term-memory used for organization. As a result, there are some really strange entries in places.

There are lines that are not used. There are codes with absolutely arbitrary values. I have left design notes to myself within the files. I apologize. I also intend to make things as clear as possible, right here.

DEVELOPER'S INTENTIONS

It is my wish that Approaching Infinity will change and grow according to the participation of the community. Your contributions, large or small, will increase the overall enjoyment of the game. If you come up with a list of new planet names, some new economic events, or even an expansion of the race quests, please do post in the forums.

MODDING AND UPDATES

From time to time, official patches and updates for Approaching Infinity will be released. It is strongly recommended that you back up your modded files in a separate location before updating. Otherwise, your files may be overwritten.

MODDING IMAGES

You can replace images without effecting achievements or pay it forward. All images must be .png format and should have alpha channel transparency.

In most cases, the names of images are determined by whatever file is loading them. For example, in 'MonsterFile.txt', there is a separate line for the name of the image that is loaded for each monster. So, in many cases, you can add your new images and change the file name in the associated text file so that it loads that image.

Some images are loaded internally, like explosions and officer portraits. If you want to change the appearance of these things, you will need to actually replace the existing image with something of the exact same name.

MODDING SOUNDS

Want to make your warp drive sound like the Tardis or make bunnies roar like a TIE fighter? Change the sounds! Within the modding directory, you will find the 'zounds' folder, which contains all the sounds used in the game. Changing sounds will not affect achievements or pay it forward.

The sounds have names based on what they are used for. If you're not sure, play them with your favorite media player.

Once you have the sound file you'd like to use (it must be a .wav file), you have two options. You can re-name your file to the exact same name as the existing one and put it in the zounds folder, overwriting the base file. Or you can put your file in the folder, calling it whatever unique

name you like and then changing the corresponding entry in the 'sounds.txt' file to match your new file.

MODDABLE TEXT FILES

AddImg.txt

This is the file that loads all the images used for the UI, treasures, quest targets, etc. Each entry consists of 3 lines: A file name, an image number, and a text description.

The file name should not have an extension and the actual file used must be a .png. The image number is referenced by the program. The majority have hard coded uses, but there is some space for adding new ones, especially in the 7000's, which are often used for unique quest items.

The text description is used by the mouse-over function. In many cases you will see the pound sign: #. This is a 'null character' which tells the program not to care about the name of this image.

You can very easily customize the look of the game with this file. Just change the name to a file you've created and see what happens.

There is one more additional fact about images you should know. Any time that an image is loaded, the program looks for another image with the same name, but a '2' after it. If this is found, it is loaded as an animation flip and that image, whether it's a monster, terrain, or anything, will be animated. An example is 'water.png' and 'water2.png'. This loads as an animated water tile.

Only 2-frame animation is supported at this time.

AchMas.txt

This is the Achievements Master file and is not found in the modding directory. You could theoretically create your own achievements, but since modding disables achievements, it's kind of self-defeating. Sorry.

Armor.txt

This file contains a list of the different types of ship armor found in the game. They are in order from weakest to strongest. The program is designed to read the number of entries in the file, so if you want to add more types, you can and you will get more variety.

ArtDesc.txt

This file contains words that are used to describe Firaxughinian artifacts and also to name their temples. They make up the second half (Y) of Artifact names, in the form of 'The X of Y', such as 'The Spatula of Peace'. You may freely add to this list.

ArtItem.txt

This file is the other half of Firax Artifacts, making up the X part of the formula 'The X of Y'. They are meant to be everyday items, even comically mundane, but they can be anything. They are also meant to be small things, easily carried and stored. You may add freely to this list.

BankerPlanetUses.txt

This file does not automatically count entries, so care must be taken to make sure that the first number in the file (initially '12') matches the number of line entries. This file is used to complete the text of certain banker quests and each line is meant to finish the sentence: "This planet looks like the perfect place" and an example entry is "for a debtor's prison."

BossStarters.txt , BossShipNames.txt

These two files are used to create the names of boss ships (and certain other things). A random entry from "BossStarters" comes first and then one from "BossShipNames". This produces instant classics such as "The Wild Caution" and "The Calculating Jellyfish". Have fun with this.

BunchaNames.txt

This file is used by the S.O.S. Beacon Quest generator and is used to determine the name of the person who lost the thing.

ChemicalStarters.txt , ChemicalEnders.txt

These two files are used together to create the names of chemical compounds that are needed in certain quests. Generally, 2 selections are made from the 'starters' list and one from the 'enders'. Certain rules are applied and a fictitious chemical name is generated, such as Betacyclozene and Nitroparazol.

Commod.txt

This file contains information about the 13 commodities that are traded by space stations. They are arranged from least to most expensive. Note the usage of the REM character ''' makes the program ignore certain lines that are useful for organizational purposes.

```
`----6-----  
`234567  
Name :Chemicals  
Price :225  
F-plan:1  
F-ast :1  
LEGAL?:1  
blank1:  
blank2:
```

'Name' is the name of the commodity and is freely modifiable.

'Price' is a base price for this item and is modified by many factors during play.

'F-Plan' means 'Can it be found on planets?'. 1=yes, 2=no. You would not naturally find boxes lying around on planets, but chemicals are a reasonable choice.

'F-ast' determines whether this item can come out of asteroids when mined. Again, boxes would be a silly choice, but chemicals are possible.

'LEGAL' has no effect.

'blank1' & 'blank2' leave room for expansion, but are currently just skipped over. This is true any time you see 'blanks' and they will not be mentioned again.

The image numbers of the commodities are 251-263 and can be found in AddImg.txt (in this example, chemicals is image #256). You cannot change the image numbers, but there is no reason to. You can change, modify, or even animate the images very easily.

DeviceFile.txt

This is a very important file: it contains information on all the special devices that you can find. You can have up to a total of 99 devices, so there is room for creativity. The device file references the 'Effects.txt' file, so changes here will sometimes need to be accompanied by changes there.

```
`-----1
`.....12345678901234567890123456789012
Name :Radiation Shunt
Desc :Protects From Radiation Belts
Effect#:1
EfLevel:1
Min_Sec:0
Race :
-PRICE-:20000
Shp/Awy:
SuitNam:
```

The first line is the device number, but it is not read by the program.

Next comes a 'text length reference', to help keep your texts properly short.

'Name' is the name of the device as displayed in the game.

'Desc' attempts to tell the player what this device does in <33 characters.

'Effect#' is the Effect Number, as read from the 'Effects.txt' file.

'EfLevel' tells how much this device has that effect. In this case, the

Radiation Shunt is basically just on or off and 1 is all you need.

'Min_Sec' is the Minimum Sector in which this device can be found. It serves to keep advanced equipment out of the earlier sectors when you couldn't afford it. Any device with a Min_Sec of -1 is never found randomly.

'Race' means which race this technology is specific to, if any. Mostly useless.

'-PRICE-' is the base price of the item, which is modified by many factors.

'Shp/Awy' determines whether this device is for your ship or the away team. When you install the device, this controls where it will go. 0=ship, 1=away team.

'SuitNam' is not valid. Space Suits that have 'powers' are referenced from the effects file.

DiseaseStarters.txt , DiseaseEnder.txt

As with other similar files, these are used together to create names for diseases. The first is meant to be an adjective for the second, which is the actual condition. Beware of Screaming Dandruff and The Leaking Humps.

Economic Events File.txt

This file gives text description flavor to the economic events that occur over the course of the game. Each event effects one particular commodity and each event has either a price-increasing, price-stabilizing, or price-lowering effect.

Try to keep in mind the general nature of these events. They can occur anywhere, at any time, so make them flexible. Also keep in mind the general rules of supply and demand: Rising supply = falling prices, rising demand = rising prices, etc. But there are many other factors that can affect prices. It is up to you!

Each event is detailed by 3 lines: which commodity is effected (a numerical value, 1-13), the number that represents whether the price is rising (1), stable (2), or falling (3), and then the text description that will be given to the player if this event occurs. The line header is 7 characters long.

Sample entry:

```
`-----  
Commod:8  
F-Type:3  
$_desc:A new Weapons Manufacturer is making a play on the Market. Weapons prices are low.  
`-----
```

Effects.txt

OK, so you thought 'DeviceFile' was a big deal? Well, this is even bigger! It controls devices, officer skills, and artifacts too! Each effect has 11 data items and a header of 10 characters. There is room for 99 effects, but you cannot add a completely new effect to the game. You can suggest them in the forums, though.

```
`234567890
Effect # :32
NAME    :Warp Cooler
Good Desc:Reduces Warp Drive Cool Down
BAD desc:Longer Cooldown On Warp Drive
Base Amnt:50
VAR AMT :25
Ship/Away:
ARTIFACT?:1
SuitName:
blank 2 :
BLANK 3:
```

'Effect #' is important because it is what is referenced by all the other files.

'NAME' is the name of the effect, not usually displayed in-game.

'Good Desc' is displayed when an artifact has this effect.

'BAD desc' is shown when an artifact has the negative version of this effect.

'Base Amnt' is the normal amount of this effect. In this case, it is 50%, but in other cases it will be a boolean (0/1, on or off) or in a different range. Use common sense, experiment and see what happens.

'VAR AMT' is how much the base amount can be randomly modified if this is an artifact.

'ARTIFACT?' means, 'can this be used for artifacts, yes or no?'. 0=no, 1=yes.

'SuitName' means 'if a space suit has this power, what is the suit called'. Presence of an entry here generally means that a space suit can be given this power.

'blanks' What you write on these lines is irrelevant, but they must still be there.

EngEne.txt

This file contains a list of 'engine energies' and is used in the creation of new thruster components for your ship. They are listed from the least to the most advanced. You can add new kinds of engines and they will be incorporated appropriately.

GreekLetters.txt

This file contains various words and symbols that are used to modify the names of planets. Only planets with one-word-names will have a chance of being modified in this way. Originally, it contained only greek letters, but it has been expanded for variety. You may freely add words here.

HandWepEnergies.txt

This file contains a list of words that are used to modify 'sidearm' components. They are in order from least to most effective. You may freely add terms to the list. In this one case, the blank line at the top is valid and means 'no special energy'.

HandWepnPara.txt

This file outlines the 'hand weapon paradigms'. These are the basic models of weapons for the away team, like pistol, shotgun, and mortar. There are 8 paradigms, each with distinct characteristics and a basic name. Many of the numbers in this file are used to modify the 'appropriate values' that are calculated based on sector, difficulty, and your progress.

```
`----- 3 -----  
-NAME-:Shotgun  
Range :60  
Damage:90  
Spread:3  
Speed:0  
-COST-:100  
SECTOR:0  
MinEfr:0  
BLTTYP:3
```

'-NAME-' is the name of this type of weapon.

'Range' is a percentage value of the 'appropriate range'. In this case, shotguns are short range weapons

'Damage' is also a percentage value. It might seem strange that the shotgun does less than 100% damage.

'Spread' is how many targets can be hit with a single shot. This can make all the difference. Now you see that the shotgun hits up to 3 targets, at nearly 100% damage.

'Speed' is how much slower or faster carrying this weapon can make you, due to encumbrance and ease of use.

'Cost' is the percentage modifier to the calculated cost in credits of a base hand weapon.

'Sector' sets a lower limit on what sector this weapon can be found in. No sniper rifles in sector 1.

'MinEFR' stands for 'Minimum Effective Range'. Using a weapon at less than its minimum effective range incurs a damage penalty. A mortar does half damage at close range, while a shotgun is very comfortable right in your enemy's face.

'BLTTYP' stands for 'bullet type' and refers to the image that is used in the shot animation. There are 3: 1=small bullet, 2=large bullet, 3=many bullets.

MeleeWeaponEnergies.txt

This file contains a list of modifiers that are applied to melee weapons. They are in order from the least to the most advanced. Each entry on this line should have either a space or a hyphen after it. You may freely add to this list.

MonDesc.txt

This file contains a list of words that are sometimes applied to the names of planet monsters to give them a little more individuality. They have no actual effect on gameplay at this time and are a bit of 'flavor' text. You can freely add to the list, but in general, words should be as universally-applicable as possible.

MonsterFile.txt

Ok, wait, which one did I say was the big one? Because this is the Big One: Monster file. About 100 monsters, with 37 data items each. Wow. So the example here is from the iconic monster...

```
`-----  
`234567890----- 25 -----  
NAME :Eye-Ball  
  image :A_Eyeball  
DESC-----:A Floating Eye!  
..RACE...:37  
--FLEE---:0  
  HP :5  
SciKill--:2  
Speed :250  
Range---:0  
Damage :1  
Reload :  
Melee---:  
Power :0  
Statnary-:0  
Breeder :25  
UNIQUE :  
MntWalk :0  
WatrWalk :0  
AcidWalk-:0  
LavaWalk :0  
NebWalk :1  
StrmWalk-:0  
  Space :2  
Planet---:  
AggRaD: :5  
DropComod:0  
Dropltem :0
```

DropDevic:0
Atk Words:
MeleeWrds:
VARIETIES:
ExploType:2
FRD_PCT :25
DPT_TURNS:1
Shot2KilU:350
Mon_Where:Ter0Fzn0Bar0Tox0Irr0HiG0Cor0Mol0Shp1Tem0Cav0Ast0xxx0yyy0zzz0
PowerLevl:2

'Name' is the name of the monster to be displayed in the game.

'Image' is the name of the image file for this monster (no extension needed, file must be a .png)

'DESC' is a short description of the monster.

'RACE' If the value is 1-12, it is one of the major races. Any other value is just for organizational purposes.

'FLEE' means 'when my hit points are below this percentage of my maximum, I run away'. A value of 75 would indicate something that runs when even lightly wounded, while a value of 0 means it never runs.

'HP' is complicated. It indicates how many Hit Points this monster should have at level 100, with all things being equal. For the Eye-ball, this is extremely low, because one shot and they pop. But for other creatures, the value tends to lie between 400 and 4000.

'SciKill' is how much data you gain from killing this creature.

'Speed' is a measure of how long it takes to perform one action. The base is 200. Higher numbers are slower (takes more time to perform one action) and lower numbers are faster. Keep it within a range of 100-400, because attacks count as actions.

'Range' indicates how far away (measured in tiles) this space ship or monster can attack you. If the value is zero, then it's melee only. Ranges of 7 or more can get quite dangerous.

'Damage' is another complex value, again measured as what it would be at level 100. And again, the Eye-ball has a very low value. Its strength lies in numbers and surrounding the player.

'Reload' is how many turns must pass before an enemy can fire another ranged attack. If the creature has no ranged attack, then this is meaningless. This ties in with speed. Something with speed 400 (very slow) and reload 4 might only fire every 8 or more turns.

'Melee' is currently unused, but is intended to signify the damage of a melee attack, as opposed to a ranged shot.

'Power' refers to a list of special monster powers that will be covered later. Most are zero. Some examples include 14 (stun), 52 (poison), 717 (disease). The numbers follow no overall pattern; consult the list.

'Statnary' determines whether this monster is stationary. Most monsters can move and the stationary ones are given special powers or bonuses to offset this disadvantage. 0=no, 1=yes (stationary).

'Breeder' is the monster number that this creature can reproduce. With eye-balls, its 25, which is more of themselves. Some space stations will spawn ships of their race and slimes spawn mini-blobs. This can be a dangerous power but is available to all creatures.

'Unique' applies to very few creatures and basically keeps them from being spawned randomly.

The following items have different effects, based on whether this monster is encountered in space or by the away team.

'MntWalk' Means 'Mountain walk'; whether a monster can cross mountains or not. 0=no, 1=yes. But for space-borne creatures, Mountainwalk governs how it interacts with other aliens. -1 = "never attack" , 0 = "only attack when attacked" , 1-12 = "attack ships of this race", 99 = "Attack Everything!" So in the case of the Cyclops, this is set to 1 and means 'yes, I can cross mountains'. But for the Gruff Instigator, a setting of 1 means 'I attack Pirates!'.

'WatrWalk' signifies whether a planetary monster can cross water. 0=no, 1=yes, 100 means "only water". In space, waterwalk governs this creature's behavior towards the player. -1 = 'never attack', 0='fight back if attacked', 1='attack if player gets too close', 99='actively pursue player'

'AcidWalk' tells whether this creature can cross over acid. Some can. Most wouldn't dare try. In space it measures how close a creature likes to be to the player during combat. It is measured in tiles and is usually 0-4, but can be higher in some cases.

'LavaWalk' determines whether a creature can cross lava. Fire sprites and Horters can do this. Lavawalk has no effect on space creatures.

'NebWalk' is really just there for the Gruff. They are afraid of going into nebulas for some reason. Other races realize that those gas clouds are safe. 0=no, don't enter nebulas. 1=yes, go ahead.

'StrmWalk' or storm walk, determines whether a space monster will enter ion storms, radiation belts, etc. Most will not, but the Tentaculons are generally fearless with regard to such anomalies. For ground-based creatures, 'StrmWalk' determines whether they are able to go through closed doors. 0=no, 1=yes. In general, only intelligent creatures should be able to go through doors. This does not apply to automatic doors, since those will open based on life-form proximity.

'Space' determines whether this monster is found in space. 0=no, 1=yes, 2=on shipwrecks, 99=yes, sometimes. 667 is a special case that means 'vacuum adapted life-form' or space monster.

'Planet' tells if this monster is found on planets. 0=no, 1=yes, 33=in caves, 99=yes, but only in very special cases.

'AggRad' stands for 'Aggression Radius'. It is how far (in tiles) this creature needs to be for it to take an interest in the player (or whatever else it might be hostile to).

'DropComod' is the percentage chance of this monster dropping a commodity when killed. Each time this command succeeds, it tries again, so a 100% chance will generally drop 9 items. There are several special codes that indicate dropping other items. There are other valid values that give special 'drop' results: 171=credits, 181=supplies, 191=parts, 797=anomalies.

'DropItem' is the percentage chance of dropping a ship component on the monster's death. Only one is dropped, if any.

'DropDevic' is similarly the percentage chance of dropping a single device.

If all 3 of the 'drop' numbers are 999, then this monster will drop 'Boss Loot'!

'Atk Words' and 'MeleeWrds' For monsters, these are the words that appear in the message window when this creatures attacks the away team with either a ranged or melee attack. They are not required, but add some flavor. They should not end in a space and are followed in game by the number of points of damage done. These are not used for alien space ships. Instead, a unique string is generated for each one.

'VARIETIES' gives a chance to have a single monster be called by several different names. One example is the Cyclops (monster #3). It possesses the line

```
VARIETIES:Cyclops#Hulk#Giant#Titan#
```

One of these is chosen when the creature is generated. The hash signifies the end of an entry. The game can read up to 10 for each monster.

'ExploType' is unused, but refers to which explosion animation is played when this monster dies, either the fire and debris of space ships or the blood splatter of a dying creature.

'FRD_PCT', 'DPT_TURNS', 'Shot2KillU' are all unused, but are for an alternate method of determining monster hit points and damage.

```
'Mon_Where:Ter0Fzn0Bar0Tox0Irr0HiG0Cor0Mol0Shp1Tem0Cav0Ast0xxx0yyy0zzz0'
```

This line tells where this monster can be found and must always have the same length. It breaks up environments into the 8 types of planets, plus shipwrecks, temples, and caves. Ter=terran, Fzn=frozen, Bar=barren, Tox=toxic, Irr=Irradiated, etc. 0=no, 1=yes. zzz is used for dying planets (type 9).

'PowerLevl' or Power Level, determines how far into the game you need to get before you encounter this monster. About every 10 sectors (modified by difficulty), the acceptable power level of monsters goes up. Values range from 0-20, 0 being anywhere, 20 being around sector 200. If you want to design some kind of super-monster, give it a power level of 20 and see if anyone ever reports being killed by it.

OfficerFirstFemale.txt , OfficerFirstMale.txt , OfficerLast.txt

These files are used to create the names of officers for hire, based on gender. You can freely add to or modify any of these lists, adding more variety or conforming to a particular mod theme.

OfficerSkills.txt

This file deals with the specifics of officer skills. It works in conjunction with the effects file to determine what officers can do and how well they do it. The line header is 11 characters and there are 13 data items (6 of which are unused).

```
`2345678901
`----- 12 -----
SKILL NAME:Energy Fields
:D E S C::Increase Maximum Shield Strength
EFFECT NUM:41
eff amount:10 ; %
# of lvls :2
+Amt/Level:15
Use Where?: ; 0=space, 1=planet, 33=caves, 11=shipwreck, 12=temple
+cooldown+:
C L A S S:2 ; ==-ENGINEERING==
MinLVL4skl:
==blank 2:
==blank3:
==blank4==:
```

'SKILL NAME' is the name of the skill, as shown when an officer has that skill.

'DESC' is a short description of the skill.

'EFFECT NUM' is the effect number from the effects file, the nuts and bolts of what this skill does.

'eff amount' is how strong the effect is. It is important to look at the effect itself to determine the proper range for this value. Some are just 0/1, some are percents and some have other ranges.

'# of lvls' is the number of levels this skill has. Many skills have two levels, very few have three.

'+Amt/level' is how much the value increases when you move to a higher level of the skill. In many cases you will find that this is actually higher than the base amount. In this case energy fields level 1 gives a 10% bonus, while energy fields 2 gives a 25% bonus to shields.

'Use Where?' is actually unused, because that is controlled by the effect. It can be used as a reference.

'+cooldown+' is unused, but would refer to a skill whose effects can only be used every so many turns.

'C L A S S' is which officer class can have this skill, 1-6. The officer classes are 1=science, 2=engineering, 3=planetary, 4=tactical, 5=navigational, 6=social.

PDSWs.txt

This stands for 'Pre-Designed Ship Wrecks' and is used to list the names of user-created shipwreck levels, which are occasionally loaded into the game. These are created using the shipwreck level editor, which can be found on the official product page here:

http://www.shrapnelgames.com/lbology/AI/AI_page.html

The actual files need to be stored in the folder 'PDSWL'.

The first number in the file is how many levels there are in total. The rest of the lines are the names of those files, with no extension. Capitalization should match the actual file name. The files themselves have the .AIS extension.

PlanetNames.txt

This contains a very large list of names used for planets (and occasionally other things too). Many are the designer's own creation, using various procedural tools. Some are real. Many others were contributed by backers of the project in its earlier days. Still more could be added by you. Want to recreate a particular setting? Put those copyrighted names in here and have fun! The game determines the length of the list at startup, so you may add and modify freely.

PartNames.txt

This file contains a list of the names of the 25 parts used for crafting. You may change them at will, but there must be at least 25 and only the first 25 in the file will be used.

Quests.txt

Quest files are another contender for the category of 'Big One'. They have 40 data items, several of which are unused. The line headers are 11 characters long. It should be noted that quests of this type must belong to one of the 12 major races. Also of great importance is that many of the codes used for quests are esoteric and strange. There will be appendices.

Even though these quests will always have the same introduction and general purpose, it is important to realize just how much they are randomized. Quests always target an unexplored sector. All the details about actual location, strength of enemies, what else you will find, etc. will be different in every game.

If the quest you wish to design does not have certain attributes, just leave them blank. Also, it is not necessary to actually enter zero whenever something needs to be zero. If you don't need it, it won't be there.

Quests should be of one type only. Get an item, kill a monster, visit an embassy, etc. If you try to do all 3 in the same quest, something unexpected will happen. The intention is that if you need multiple quest types, you create multiple steps in the file and string them together.

```
Quest_NUM :101
Quest_Name:Dungeon Diving
DANGER MOD:
W O R D S :Pirates have been burying treasure since the beginning of time,
WORDS 2 :and surely there's some out there that's been forgotten.
WRDS3 :Search caves! Find the lost treasure hordes.
COMPLETION:You've done well.
WORDS :Here's your cut.
OF THANKS :
Fail Words:
FAILWORDS2:
TargPlanX :0
TrgPlnKind:0
GetItem :7001
GetNum---:5
ItemCalled:Pirate Booty
KillWhat :
KillNum--:
++Explore+:
REQUIRE NUM:0
REQ AMT :0
requiremod:1 ; in this case, it's how many of these are found on each level
TrigrEvent:
RewardCred:20000
RewardItem:
RewardDev :
RewardShip:
RewardCom :
?ARTIFACT?:
UnlocksQst:102
CAPTAIN GO:1
GiveYouDev:
SectorSpec:-1
WhereItsAt:33
blank 3 :
blank 4 :
blank 5 :
```

blank 6 :
blank 7 :
blank 8 :

'Quest_NUM' means which number this quest is. Race quests move in a linear structure. One must be completed before the next one is revealed. The Quest_Num code is $(\text{RaceNumber} * 100) + (\text{QuestNumber} - 1)$, so in the above example, this is the second quest of race 1 (quest_num 100 would be the first quest).

'Quest_Name' is the name that shows up in your quest log.

'DANGER MOD' can effect the strength of enemies in the target area. The value is equivalent to sectors, so a quest with a danger mod of 5 that occurs in sector 10 will have level 15 enemies (approximately).

'W O R D S','WORDS 2','WRDS3' Three lines of text that are used to tell the player what this quest is all about. There is a soft limit of approximately 360 characters.

'COMPLETION','WORDS','OF THANKS' Three lines of text given upon successful completion of the quest. Approximately 360 characters.

'Fail Words','FAILWORDS2' are words of failure and are almost never seen by the player. Best left alone.

'TargPlanX' indicates whether a planet should be generated for this quest. Used when whatever you're looking for is to be found on a planet surface. 0=ignore, -1='yes, make a planet'. And yes, that is 'Negative One'. Once the game has created an actual planet, the planet's coordinates will be stored in this variable. If you try to force the planet to occur at a certain location, it will simply not be generated. So the valid numbers are 0 & -1.

'TrgPlnKind' (Target Planet Kind) can be used to create a specific type of planet, from terran to dying, with a number 1-9. If this value is 0, the planet could be of any kind. -37 yields an island planet.

'GetItem' is the item image number that needs to be picked up, if this is a 'fetch quest'. These numbers are best kept in the range 7001-7079, since those are reserved for and recognized as 'quest items'. You can easily add your own quest item images using the AddImg file.

'GetNum' is how many of the items you need to collect.

'ItemCalled' is what the collection item is actually called in-game. Sprockets, Bandrite Eggs, etc.

'Killwhat' refers to the monster number you have to kill, such as 25 meaning Eye-Balls or 1 = pirate raider. If you set KillWhat to -1 it will make the game count all 'space enemies' killed. -2 works for away-team enemies.

'Killnum' is how many of that type of monster you need to kill. 100 would be a good number for Eye-Balls.

'Explore' has only one use: require the player to fully explore a planet. set to 1 to make this the necessary condition.

'REQUIRE NUM' has a very large number of uses, which will be listed in "Appendix 2: Requirement Numbers, Amounts, and Modifiers."

'REQ AMT' has a very large number of uses, which will be listed in "Appendix 2: Requirement Numbers, Amounts, and Modifiers."

'requiremod' has a very large number of uses, which will be listed in "Appendix 2 : Requirement Numbers, Amounts, and Modifiers."

'TrigrEvent' also has many uses and will be handled in 'Appendix 3: Triggered Events'

'RewardCred' finally something simple again, the base number of credits that completing this quest will get you. The end result will be modified by many factors. Numbers between 5,000 and 250,000 are common.

'RewardItem' the item number (1-13) that will be awarded to the player upon completion of the quest. The item will be a randomly generated version of the type.

'RewardDev' can give the player a specific device after completing the quest. Choose the device number from the device file.

'RewardShip' is used to unlock the unique starting ships that each race makes available. It refers to the ship number from the StartingShips.txt file.

'RewardCom' is used to give the player lots of a particular commodity. Use the commodity number (1-13).

'?ARTIFACT?' this line is unused.

'UnlocksQst' even though quest structure is basically linear, this line points to the number of the next quest that the player will receive. Some race quests loop and this line can accomplish that behavior.

'CAPTAIN GO' determines whether the captain will need to be present on the away mission, if this quest does indeed lead out of the ship.

'GiveYouDev' this line can be used to give a specific device at the beginning of a quest. Generally done when you need something specific to complete this mission.

'SectorSpec' influences how far away from your current sector the new quest is likely to be spawned. 0 is a very common number here and setting it too high will require the player to pass through long stretches of space to reach their destination. Setting this value to -1 has the effect of allowing the quest to be completed in any and all sectors. If the point is to kill pirate raiders, setting this to -1 will mean that you can kill pirate raiders in any sector and it will count toward the quest. Setting it to 0 will mean that you must kill them in a specific sector, to be determined when the quest is actually generated.

'WhereItsAt' controls what general area this quest takes place in. 0=space, 1=planet surface, 11= shipwrecks, 33=caves, -421 is inside a galactic megastructure. 31313 sets this as an S.O.S. quest and the item may be found anywhere.

'blank 3' - 'blank 8' are blanks and can have anything written there or they can be blank.

Races.txt

This file defines certain constants about the 12 major races. It takes much more to properly flesh out a race, like designing their ships, history, and quests. But this information is important none the less. There are 14 data items, with a 7 character header.

EXAMPLE:

```
` #1
Name :Space Pirates
DESC :
image :
SagaMAX7
MinSec:0
* Rep :40
QuesRp:65
QsNAME:Raiders Of The Lost Space-Ark
Enemy :6
ByArt4:60
$2Cure:200 ; very expensive
HAIL_F:Ahoy Matey!
HAIL_N:Surrender or be destroyed!
HAIL_E:Prepare to die!
```

'Name' is the name of the race as its displayed in anything that directly references race, like embassies and quests.

'DESC' is not used, but can provide a short description of the race's demeanor.

'image' is not used, but is intended to show a picture of a typical individual of that race.

'SagaMAX' means how many installments there are in this race's historical monuments, found in embassy sectors.

'MinSec' is the minimum sector that this race can be encountered in, under normal circumstances. There are times when a race will be encountered ahead of this time, based on other events.

'* Rep' is the starting reputation for that race. It's how much they like you the first time they see you.

'QuesRp' stands for quest rep and is how high you need to get your reputation with that race before they will give you quests. It also influences how easy their embassies are to find.

'QsNAME' every race has a quest line and this is the name of the overall story.

'Enemy' this is unused but can be used to keep track of who hates who.

'ByArt4' is a percentage value and influences how much this race will pay for Firaxughinian artifacts. It has nothing to do with actual pieces of art. In the case of the pirates, it's only 60% of the normal rate.

'\$2Cure' is another percentage, this time meaning how much this race charges to cure your diseases. In this case, pirates charge double the normal rate. A value of zero means that a race cannot or will not cure diseases at all, but it also means they don't care about them and will let you board their stations.

'HAIL_F' stands for friendly hail, is a short text given when you encounter the race, if they are feeling friendly towards you (based on your reputation with them).

'HAIL_N' is the neutral hail.

'HAIL_E' is a hail given when they are coming to kill you.

Randoms.txt

This file contains a very long list of flavor texts that are randomly encountered. The format is very flexible and uses ##### (6 hash symbols) on a line to indicate the end of an entry. Feel free to add in your own items here, since the game counts the number available at start-up. Try to stay under 500 characters for each message.

New messages of this type are highly desired and a valid creative outlet. I hope to see a wealth of these appear on the boards.

SA - amount.txt, SA - does.txt, SA - output.txt, SA - size.txt, SA - what.txt, SA - when.txt

These 6 files contain words that are used to describe spatial anomalies, which at this point are only ever seen at the end of a certain type of Banker quest.

The text generated in the game takes this form:

A [size] [what] that [when] [does] [amount] [output].

It actually works surprisingly well.

An irrelevant note: I created this set of words over 10 years ago, while designing a space 4X game that never saw the light of day. It had functioning procedural tech trees for each race and they researched the techs according to their needs!

ScanEne.txt , ScanKinds.txt

The entries in these two files are used to construct the names of different kinds of 'scanner' components you can find. The 'energy' comes first, then the 'kind'. Both are arranged from least to most effective. Examples: Particle Detector, Hyperspace Radar.

SheEne.txt , SheKind.txt

Like the other files of similar types, these two files are used to name shield components. Both files are arranged from least to most powerful. Add or modify at will. Examples: Magnetic Deflector, Stasis Barrier.

ShipNames.txt

This file contains the list of names for the player's ship that is randomly chosen at the beginning of each new game. Feel free to add or modify. Remember that you can easily change the name of your ship by clicking on the name while in space.

SHIPWRECKENDERS.txt , SHIPWRECKSTARTERS.txt

These two files are used to name shipwrecks, although there are other internal functions that sometimes take precedence. The 'Enders' are used more often than the 'Starters'. The starters are meant to be words that can universally modify the other word in the name. Please add and share.

Sounds.txt

This file contains the names of the sounds used in the game. It is used directly in the loading of the .wav files into memory. This makes it very easy to just drop your own sounds into the Sounds folder and cause them to load at start-up.

Each entry consists of 2 items: a file name and a sound number. The numbers must be kept as they are and the filenames should give you some idea about what they are. You are free to experiment.

StartingShips.txt

This file contains all the specifics of the 18 ships that you can start a new game with. These ships are also used as the basis for creating new ships for sale in the space station shipyards. There are 16 data items with a line header of 13 characters. Note that the devices in the starting ships are considered to be an integral part of that type of ship and do not take up space or slots.

` = = 4 = =

`234567890123

Name :Asteroid Miner

Weapon Slots:2 (it has room for two weapons)

Weapon #1 :3 (starts with a level 3 small weapon)

Weapon #2 :0 (this means it starts with no second weapon)

Hull :12

Shield :0 (it starts with no shield)
Armor :2
Cargo Space :35
Crew Size :10
OFFICERS :1
Speed :225 (it is a little bit slow)
AUX SLOTS :2
Supplies :4000
DEVICE :3 Rock plow
bblank4 :4 Asteroid Cracker
_blank5 :

'Name' is the name of that class of ship and should be descriptive of what the ship's intended function is.

'Weapon Slots' is the number of weapon slots available on this ship to begin with.

'Weapon #1' is the level of the randomly generated small starting weapon. Values of 1-4 are common, 10 is exceptional.

'Weapon #2' is the level of the medium weapon (slot 2), if any.

'Hull' is the number of hull points the ship has. Values of 10-35 are common and the current highest is 250.

'Shield' is the level of the starting shield (or zero meaning no shield). 1-4 are common, 10 is exceptional.

A value of -99 in the 'Shield' field will start the ship with a cloaking device instead of a shield.

'Armor' is the level of the installed starting armor component, if any.

'Cargo Space' is the size of the cargo hold. Commodities take 1 space, devices 2, and components 4.

'Crew Size' is the maximum crew size of this vessel. The player will actually start with about 75% of this value.

'# OFFICERS' was meant to limit the number of officers by ship type, but is no longer used.

'Speed' represents how long it takes for this ship to move 1 tile. 200 is the base, higher numbers are slower, while smaller numbers are faster. The value is capped at a low of 100 and a high of 400.

'AUX SLOTS' is the number of device slots this ship has. 1-3 are common, 6 is the highest possible.

'Supplies' is the amount of supplies this ship can carry. Values of 3500-5500 are normal. 8000 is the current highest of any ship.

'DEVICE' is a device number, from the DeviceFile.txt, which is installed on this ship. This device does not count against the total number of devices the ship can have, because it is a built-in effect. Ships are not required to have these devices.

'2nd Device' is a second device, as above, if any.

'_blank5' is a blank. It is necessary, but irrelevant.

suits.txt

This file contains a short list of words used to modify 'space suit' components. There is no discernable quality difference between types. It is purely descriptive.

Terrains.txt

The terrains file contains basic information on all the different types of terrain in the game, for space, planets, shipwrecks, etc. The name of the terrain is also the name of the file and it must be a .png file. There are 12 data items and an 8 character line header.

Planetary terrains almost exclusively occupy slots below 60.

Terrains 47, 56, 57, and 58 are very special and should never be modified.

```
`2345678-----  
Name :Rich Soil  
Number#:5  
GROUP :10  
BS :0  
BM :0  
Damage :  
PowerRQ:  
$desc$ :  
FOUND :1  
On_Kind:Ter1Fzn0Bar1Tox1Irr1HiG1Cor1Mol0zzz1  
-blank-:  
+blank+:
```

'Name' is the name of the terrain. It needs an accompanying image as a .png file of the same name.

'Number#' is the terrain number of this terrain, as it is referred to internally by the program. This is actually very important.

'GROUP' is used for planetary terrains and allows the terrains to be loosely grouped together. 10=open grounds, 11=a short feature, 12=trees, 13=mountains, 14=liquid. 99 signifies a terrain that should never be randomly chosen.

'BS' stands for Block Sight and determines whether this terrain will block your line of sight. 0=no, 1=yes.

'BM' stands for Block Move and is used to tell whether you can move here or not. 0=no block, 1=block.

'Damage' works in space, but not on planets. It signifies that a particular terrain will cause damage when you move onto it.

'PowerRQ' and 'FOUND' are unused.

'\$desc\$' is used in version 1.0.1+ to give more helpful tooltip information about terrains.

'On_Kind' is a string with a very special and necessary format. It determines whether this terrain is found on each of the nine kinds of planets, terran through dying. There is a 3 letter text abbreviation for each atmosphere and then a 0 for no or a 1 for yes. zzz is used for 'dying' planets. So in the above example, 'rich soil' is found on every kind of planet except for frozen and molten.

'Blank' there are 2 blanks. They are necessary but irrelevant.

TrackList.txt

This contains the ordered list of tracks that are played during the game. It also effects which songs are loaded at startup. If you want to add your own songs into the mix, then add the name of the file here, count how many tracks total there are and change the first line to that number. Then, you need to manually copy those tracks into the game's media/music folder. Having less than 18 tracks may cause errors or crashes.

WepParadigms.txt

This file is similar to the HandWepnPara.txt file, and determines the basic attributes of the six sizes of ship weapons, from defensive to massive. Each entry has 9 data items and a line header of 7 characters.

All of the values given are just a starting point and they are modified by sector, level, and random chance. Also, they are gauged for sector 1, so don't worry, they become very powerful. Finally, there are some weapon modifications that are made internally. It is possible to find a weapon that can shoot up to 16 targets with a single click. There are also some short range, high damage variants and some rapid-fire ones as well.

```
`----- 6 -----  
`234567  
NAME :Massive  
Range :6  
Damage:34  
CoolDn:10  
Supply:350  
CritCh:  
@Level:28
```

blank2:

blank3:

'Name' is the name of the paradigm. It is normally used in naming the weapon.

'Range' is the base range of the weapon, in tiles.

'Damage' is the damage amount of the weapon, measured in enemy hull points.

'CoolDn' is the number of turns this weapon needs to recharge before it can fire again. 1=every turn, 2=every 2 turns, 10=every 10 turns, etc.

'Supply' is how many supplies are consumed every time this weapon is fired.

CritCh, @Level and blanks 2 & 3 are not used.

WhereYouKept.txt

This is a fun little file that is used to create log messages when you recover a Firaxughinian artifact. They are only ever seen when you discover a 'log' pickup, but they are generated from your previous actions. The file contains random flavor texts that are supposed to be the captain's reaction to this artifact. Add freely and please share!

APPENDIX 1: Monster Powers

This is a list of all existing monster powers. You cannot add new ones (but you can certainly suggest some!). No monster can have more than one power, but many have none.

13: call for help	14: stun
17: LAY MINES (in space)	23: weave webs (in space)
44: proximity explosion (on away missions)	51: fire status effect
52: poison status effect	53: shock status effect
133: steals oxygen	155: cause blindness
191: drains your supplies (in space)	202: takes half damage from ranged attacks
232: shots steal cargo (in space)	247: explodes when close to your ship
272: adds cooldown to your weapons	293: adds cooldown to your shields
357: adds cooldown to your engines	373: adds cooldown to sensors
393: digs through cave walls	441: terraforms planets
447: destroys planets!	717: cause disease
1623: harvests (destroys) asteroid belts	2041: teleports after making a melee attack
10203: picks up non-quest items on planets	32767: when one dies, all others get stronger
32768: removes mines from space	

(powers in the range of 7000-7999 drop a quest related item, specifically that item number.)

APPENDIX 2: Quest Requirement Numbers, Amounts, and Modifiers

These three lines in the quest file can be used to create and fine tune a wide range of quest types:

```
REQUIRE NUM:  
REQ AMT :  
requiremod:
```

They are 'requirement number', 'requirement amount', and 'requirement modifier'. Each one behaves in its own way. This is another one of those systems that was designed on the fly, as new things were needed. It is a bit chaotic.

Some of these cause effects on their own and some require multiple entries to know what to do. In cases where a single line will suffice for the task, only one entry is made. Whenever more

information is needed, there will be " + " to denote that the extra information applies to that entry.

All Requirement Numbers

303 : "Bring us a Firaxughinian artifact and sell it in our pawn shop".
+ Req. Amt. and Req. Mod. should be set to 303 as well in these cases.

307 : Bring us a certain amount of DATA.
+ Req. Amt. : how much data to bring (1-999,999)

309 : Discover a new crafting recipe.
+ Req. Amt. is how many recipes need to be discovered.

311 : Change Stellar Spectra
+ Req. Amt. is how many you need to change.

327 : Pay us money!
+ Req. Amt. is how much money the player needs to have.

417 : denotes that this is an "S.O.S. beacon" quest.

401-499 : set all monsters on the quest planet to a certain monster number. Only works with monsters 1-99. Will not work with higher monster numbers. May be changed in future.

501-512 : Make sure that all ships in a given sector are a specific race. The form is 500 + Race_Number.

601-613 : Sell us a particular commodity in our Commodity Exchange. 600 + Commodity_Number.
+ Req. Amt. : determines how many they want you to sell them.

801-812 : Visit the embassy of a given race. 800 + Race_Number.

901-912 : visit any embassy EXCEPT this race 900 + Race_Number.

Requirement Amount

This is normally used to determine how many of a thing you need to find or how many times to do something. Once the quest is started by the player, it is often used internally as a counter, meaning that each time a thing is found or done, the number needed is decreased.

I do not know of any cases where 'Req. Amt.' is used for anything other than "How many do I need?", unless specified in reference to something else.

Requirement Mods

These are usually things that modify other things. If they are listed as part of another entry, then they will not be listed here.

0-25 : If this is a 'fetch' quest, then this can be used to limit how many of the target item are generated per area of the quest. For example, if you want the player to have to find 24 units of Sporidium in caves, but you want them to have to go to at least 3 different caves or cave levels, then plug 8 in here ($25/8=3$).

6969 : this will cause most existing items on shipwrecks to be converted into the 'GetItem' that this quest requires. Not overly useful and generally creates a mess.

10001-10099 : Spawn this monster # in space. Consult the monster file and pick a number of a monster that belongs in space. $10000+Monster_Number$.

APPENDIX 3: Triggered Events

Here is another big crazy list of things that quests can do. Some of these types of events happen upon accepting the quest and some happen upon completing it. I will attempt to let you know which is which.

They have the header 'TrigrEvent:' and are a numerical code.

501-512 : Make sure that all ships in a given sector are a specific race. The form is $500 + Race_Number$. (same as for 'Requirement Numbers'.)

4401 - 4412 : force the target sector to have an embassy of a given race. $4400+Race_Number$.

4701 - 4713 : creates a shortage of a commodity in the current sector.
 $4700+Commodity_Number$.

4801 - 4813 : creates a surplus of a commodity in the quest target sector.
 $4800+Commodity_Number$.

4999 : Makes sure you have at least 10 of every crafting part.

10001 - 10012 : Accepting this quest makes one of the 12 races hate you. If you haven't met them yet they will be introduced. $10000+Race_Number$

11001 - 11012 : Accepting this quest makes one of the 12 races love the player, introducing them if necessary. $11000+Race_Number$

12001 - 12012 : Accepting this quest makes one of the 12 races neutral to the player, if they already have bad relations (wipes the slate clean). Will also introduce if necessary.
 $12000+Race_Number$

13001 - 13013 : Removes a race from the game. They will not be chosen for new sectors and might be removed from old ones. $13000+Race_Number$

90001 - 90012 : Completing this quest will grant a 'racial victory'. This will end your game. The form is $90000+Race_Number$. I do not believe it is possible to add a new racial victory at this time, but you can try.