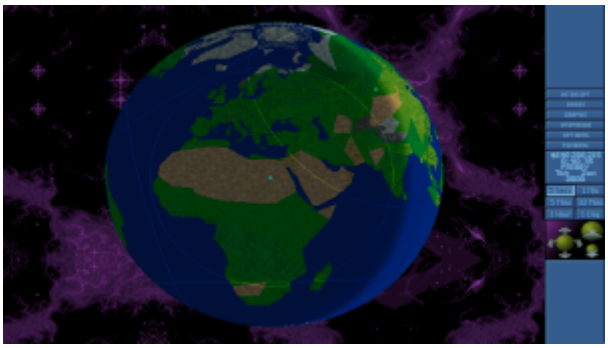
OpenXcom

Infos

OpenXCom ist im Prinzip nichts anderes, als X-COM UFO defense bzw. Terror from the Deep nochmal neu nach programmiert. Das Spiel hängt dabei jedoch von den Original Grafikdaten ab, man braucht UFO also noch im Original. Gibt's z. B. noch bei GOG und Steam. Da das Original allerdings auf den aktuellen Systemen praktisch nicht mehr läuft, kann man das gerade noch verschmerzen. Richtig cool ist allerdings der Umstand, dass es das Spiel auch für Linux gibt.

Auch wurden einige Änderungen vorgenommen, die das Gameplay verbessern. So kann man die Einstellung setzen, dass vor dem Ausführen eines Geh-Befehls angezeigt wird, welchen Weg der Soldat nehmen wird und was die Zeiteinheitenkosten sind. Türen können jetzt, wie bei TFTD, aufgemacht werden, ohne zwangsläufig auch gleich durch zu gehen. Man kann seitwärts gehen, man kann rennen, das Basis-Layout beim Starten eines neuen Spiels selbst bestimmen. Auch genial ist die Möglichkeit sich die Radar-Reichweite anzeigen zu lassen und den Kurs, den das Raumschiff abfliegen wird.



Man kann den Soldaten auch anweisen, ohne Rücksicht auf fehlende Visierlinie auf das Ziel zu ballern. Gerade mit dem Autoschuss zusammen kann man damit Hindernisse wegballern, mit der Chance, dass er mit den verbleibenden Schüssen doch noch das Ziel trifft. Auch kann man jetzt endlich die Reihenfolge der Soldaten in einem Raumschiff festlegen und deren Waffenlayout wird gespeichert (sodass man nicht vor jeder Schlacht alles neu sortieren muss). Man kann auch einstellen, dass man Raumschiffe abheben lassen kann, obwohl ihr Treibstoff und ihre Munition nicht 100% hat. Scrollrad an der Maus hat jetzt auch eine Bewandnis, nämlich Ebenen wechseln. Auch kann man mit der Tastatur scrollen, die Soldaten durch wechseln. Die Anzahl der Savegames ist nicht mehr limitiert, weil er nur noch nach Dateinamen geht. Lauter so Dinge, die ich persönlich sehr stark finde.

Auch wurden einige Bugs behoben. Z. B. dass der Avenger nie mehr als 50% vom Treibstoff benutzt hat ist jetzt nicht mehr so. Man kann endlich die Panzer wirklich frei bewegen, ohne Einschränkungen durch die Fehler in der Maus-Steuerung. Ebenfalls kann man jetzt, wenn man denn keine Lust hat, lang und breit eine (neue) Kampagne zu starten, einfach direkt eine Schlacht generieren lassen, mit Auswahl welches UFO, welche Rasse, welcher Technologiestand, mit was kreuzt man selbst auf. Gutes Anfänger-Training für alle, die UFO2000 spielen wollen. 😀



Installation

Es gibt ein relativ altes Installations-Paket mit OpenXcom 1.0 stable, also nicht Nightly. Auch wenn »stable« erst mal suggeriert, dass das die bessere Variante wäre, sind die Nightlies aus diversen Gründen zu empfehlen:

- mehr Bugs gefixt,
- mehr Funktionen,
- Mods basieren inzwischen eigentlich immer auf der Ordner-Struktur der Nightlies, gehen also nicht mit der stable
- es gibt auch TFTD als direkt mit installierten Mod.

Das Spiel benötigt *zwingend* die Original Daten von X-COM. Wenn eure original Datenträger inzwischen hinüber sind, könnt ihr die Files aber im Netz durchaus finden. OpenXcom braucht entweder Version 1.4 der DOS-Version oder aber die Collectors Edition. Wenn man als nicht-Besitzer korrekt vorgehen will, kauft man sich das Spiel bei Steam. Es gibt inzwischen auch einen von der Community erarbeiteten Patch, der u. A. Map-Bugs und ähnliches behebt. Zu finden ist er unter http://openxcom.org/download/extras/universal-patch.zip.

Windows

Das Spiel wird unter Windows einfach aus dem Archiv-File in einen entsprechenden Ordner entpackt. Die Original-Daten gehören dann ins Verzeichnis `UFO` bzw. `TFTD`.

Linux

über Paketquellen

Wie schon geschrieben, ist es eher nicht zu empfehlen, die stable zu nehmen, sondern lieber gleich die nightlies:

```
sudo add-apt-repository ppa:knapsu/openxcom
sudo apt-get update
sudo apt-get install openxcom
```

Dann noch die Original-Files an die richtige Stelle packen, direkt ins Verzeichnis `/usr/local/share/openxcom/UFO/`. Für TFTD ist es der Ordner

`/usr/local/share/openxcom/TFTD/`. Dort sucht OpenXcom nach den Original-Files. Ggf. auch an anderer Stelle, aber es macht Sinn, hier konsistent zu bleiben, um keine Probleme zu bekommen. Mods gehören übrigens nicht mehr ins Spiele-Verzeichnis sondern nach `$HOME/.local/share/openxcom/mods`, ein Verzeichnis tiefer, als auch die saved games liegen. Danach sollte man spielen können.

Kompilieren

Das Spiel gibt es zwar im Prinzip auch über die Paketverwaltung. Allerdings sind die Pakete nicht immer die aktuellsten. Man kann das ganze auch manuell erledigen:

Voraussetzungen

Es werden vom Original Spiel einige Verzeichnisse benötigt, um es überhaupt zum Laufen bringen zu können. Die X-COM-Version ist entweder 1.4 (=DOS), oder aber die Collectors-Edition (=Windows). Falls jemand das Spiel nicht hat, kann man es sich auch heute noch via Steam kaufen, oder aber man googlet mal danach, obs das nicht wo gibt (wenn man seine Installationsdatenträger verloren hat...) Gibt zumindest noch so Abandon-Ware-Seiten, die das anbieten. Da ist es ratsam, gleich nach der Collectors Edition zu suchen, weil man bei den DOS-Versionen nicht immer weiß, welche Version man jetzt wirklich erhalten hat. Benötigt werden die Verzeichnisse:

- `GEODATA`
- `GEOGRAPH`
- `MAPS`
- `ROUTES`
- `SOUND`
- `TERRAIN`
- `UFOGRAPH`
- `UFOINTRO`
- `UNITS`

Hinweis an dieser Stelle: Da Linux bei Dateinamen case-sensitive ist, *müssen* die Ordner in Großbuchstaben geschrieben sein. Ansonsten findet das Spiel unter Linux die Dateien einfach nicht.

Man sollte den Community-Patch installieren. Wenn man zukünftig öfter selbst kompilieren will, kann man diese Files auch einfach mit denen vom Original-Spiel zusammen packen. Erst die oben genannten Ordner in ein Verzeichnis, dann die Verzeichnisse aus dem Patch-Archiv auch dort hin entpacken und ggf. überschreiben lassen. Das Ergebnis kann man sich wieder in ein Archiv für spätere Verwendung bunkern.

Zur Benutzung müssen evtl. noch Pakete nachinstalliert werden, insbesondere zum Compilieren:

- git (nur zum Kompilieren)
- SDL
- SDL
- SDL Mixer
- SDL Image
- SDL gfx Version 2.0.22 oder höher
- `yaml-cpp` Version 0.5 oder höher

- `boost` (Abhängigkeit für yaml-cpp)
- ggf. `cmake` und `cmake-curses`

Oder als ein Befehl zur Installation unter Ubuntu:

```
sudo apt-get install git libsdl1.2-dev libsdl-mixer1.2-dev libsdl-image1.2-dev libsdl-gfx1.2-dev libyaml-cpp-dev libboost-dev cmake cmake-curses-gui
```

Die `-dev`-Pakete sind nur zum Kompilieren notwendig, deren kompilierte Versionen sollten eigentlich mit installiert werden, wenn man OpenXcom aus einem PPA installiert.

Compiler-Lauf

Man kann, bzw. muss das Spiel evtl. auch mal selbst kompilieren, z. B. weil neuere Mods auch neuere OpenXcom-Versionen benötigen. Als Ubuntu-User muss man für Nightlies allerdings selbst zum Compiler greifen. Das ist aber auch nicht unbedingt sonderlich schwer. Man kann das im Prinzip auch weitestgehend automatisieren.

Quell-Dateien runterladen. Diese findet man immer unter https://github.com/SupSuper/OpenXcom, rechts unter dem Link »Download ZIP«. In ein beliebig benanntes Verzeichnis entpacken (wird hinterher nicht mehr unbedingt benötigt). Oder man macht die Geschichte gleich mit git:

```
git clone https://github.com/SupSuper/OpenXcom.git OpenXcom-master
```

Mit git hat den Vorteil, dass auch der Versions-String eindeutig identifiziert, welches Build da kompiliert wurde, z. B. wenn man Bug-Reports einreichen will. Das grenzt dann ziemlich genau ein, mit welcher Version das Problem auftritt.

1. Die oben bereits genannten Original-Files und die Patch-Files in den Ordner `bin/UFO/` der gerade entpackten Quell-Dateien kopieren/entpacken. Falls man noch TFTD rum liegen hat, das dann auch nach `bin/TFTD/` kopieren.
2. Ggf. den Versions-String anpassen: Normalerweise ist in diesen Quellfiles immer als Version die Nr. 1.0 angegeben, was so relativ wenig Aussage hat, weil man auf diese Weise z. B. gar nicht feststellen kann, was für eine Version man letztlich installiert hat, eine aus den PPAs, ein nightly, und wenn letzteres, von wann. Man kann daher in der Datei `src/version.h` den String `#define OPENXCOM_VERSION_SHORT "1.0"` entsprechend abändern, z. B. auf `#define OPENXCOM_VERSION_SHORT "1.0 Nightly 2015-06-05-12-13"`, für den 5.6.2015, 12:13 Uhr. Bei Git entfällt die Notwendigkeit dafür, es ist aber auch möglich, wenn man noch mehr Infos in der Version haben will. Bei Windows-vorkompilierten nightlys von der Website ist z. B. ein automatischer Eintrag von Git enthalten.
3. Ordner `build` im Quell-Verzeichnis anlegen und auf der Konsole dort hinein wechseln.
4. `cmake ..` oder alternativ `cmake -DCMAKE_BUILD_TYPE=Release ..` für den Fall, dass man ohnehin als Release in die Standard-Verzeichnisse installieren möchte. Das erspart den nächsten Schritt. Es lassen sich aber auch die Ordner angeben, wenn man das möchte, indem man mit `-DVariable=Variablenwert` die Pfade mit übergibt (siehe Punkte unter 6).
5. `ccmake ..`, dieser Schritt und der nachfolgende sind optional, wenn man `cmake` mit den direkt übergebenen Optionen ausgeführt hat.
6. Einstellungen setzen:
    1. `CMAKE_BUILD_TYPE` auf `Release` setzen, alternativ auf `Debug`.
    2. `CMAKE_INSTALL_PREFIX` auf den Pfad setzen, in den die OpenXcom-Binary später installiert werden soll. Standard ist `/usr/local/` (voreingestellt), womit das Spiel im

Ordner `/usr/local/bin` landet.
3. DATADIR auf den Pfad des Ziel-Ordners für die Daten-Files (Originale) setzen. Die weiter oben in die Sourcen kopierten Files werden dann dort während des nachfolgenden Prozesses hin kopiert. Gibt man keinen Pfad an, landen die Daten in `/usr/local/share/openxcom/data/`.
4. c für configure drücken
5. e um aus dem configure-Bildschirm wieder raus zu kommen.
6. g zum Generieren und Beenden von `ccmake` drücken.
7. `make -j3`, man kann auch höhere Werte als 3 angeben, z. B. `10`, um den Kompilations-Prozess zu beschleunigen. Allerdings ist es nicht empfohlen, keine Werte anzugeben, damit es maximal schnell läuft. Es können unvorhergesehene Zustände auftreten, die das Kompilieren verhindern!
8. `make install`
9. Falls man beim Pfad `CMAKE_INSTALL_PREFIX` oder `DATADIR` einen Ordner angegeben hat, für den man die Schreibrechte nicht hat, muss man unter einem Benutzer kompilieren, der die Rechte dazu hat. Beim Standard-Pfad sind also Root-Rechte notwendig:
10. `sudo make install`

Damit ist die Kompilation abgeschlossen (sofern nicht irgendwelche Fehler dabei aufgetreten sind…). In diesem Fall dürfte man bei Standard-Pfaden einfach loslegen können. Hat man an den Pfade rum editiert, z. B. um das Spiel als User irgendwo zum Laufen zu bekommen, also ohne richtige Installation, kann man sich noch ein Start-Skript schreiben:

```
#!/bin/sh
"Pfadzu/openxcom" -data "Pfadzu/data"
```

mit den jeweiligen Pfaden zu OpenXcom bzw. dem Original-Verzeichnis. Das Skript dann unter `$HOME/bin/` ablegen, und z. B. `openxcom` nennen, ausführbar machen und schon kann man OpenXcom auch ohne Admin-Rechte für die Installation im eigenen Home-Bereich spielen, indem in der Konsole einfach `openxcom` eingibt. Das Skript kann natürlich auch irgendwo anders liegen, z. B. auf dem Desktop, wo man es dann auch anderweitig starten kann.

Desktopfile

Wenn man richtig viel Wert auf Optik legt, kann man für das eben erstellte Start-Skript auch ein Desktop-File anlegen, um das eben kompilierte OpenXcom z. B. auch in den Starter von Gnome integrieren zu können:

```
[Desktop Entry]
Name=OpenXcom
Exec=<Pfadname/zum/OpenXcom-Skript>
Type=Application
StartupNotify=true
Path=<Pfad/zum/OpenXcom-Skript>
Icon=<Pfad/zum/OpenXcom-Ordner>/common/openxcom.png
```

Dieses File dann z. B. unter dem Namen `OpenXcom.desktop` speichern und ausführbar machen. Dieses File kann im Prinzip irgendwo liegen, allerdings empfiehlt es sich für Ubuntu mit Gnome, es unter `~/.local/share/applications` abzulegen, da dort auch der Starter nachsucht. Für alle Benutzer ist der Ordner `/usr/share/applications` aber besser geeignet, weil so alle das Symbol

finden können.

Wichtig: Die eben installierten Files im Original-Ordner werden möglicherweise nicht verwendet, wenn in einem anderen der abgesuchten Pfade noch ein UFO- oder TFTD-Ordner gefunden wird. Daher empfiehlt es sich, alle anderen Original-Ordner entweder zu löschen oder wenigstens umzubenennen (z. B. `data alt` oder so ähnlich). Man kann evtl. weitere vorhandene Original-Ordner mit einem einfachen Befehl finden:

```
sudo find / -name BACKPALS.DAT
```

sudo ist notwendig, weil man sonst u. U. nicht alle Pfade findet, bzw. ständig Fehlermeldungen wegen fehlender Rechte die Ausgabe unterbrechen. Die Datei `BACKPALS.DAT` ist Teil des UFO-Verzeichnisses und sollte theoretisch nur ein mal auf der Platte vorkommen, sofern man sie nicht noch woanders gebunkert hat. Sollten sich mehrere Pfade finden lassen, die `data/GEODATA/BACKPALS.DAT` am Ende haben, sind das genau solche möglichen Problemstellen.

Automatisiert kompilieren

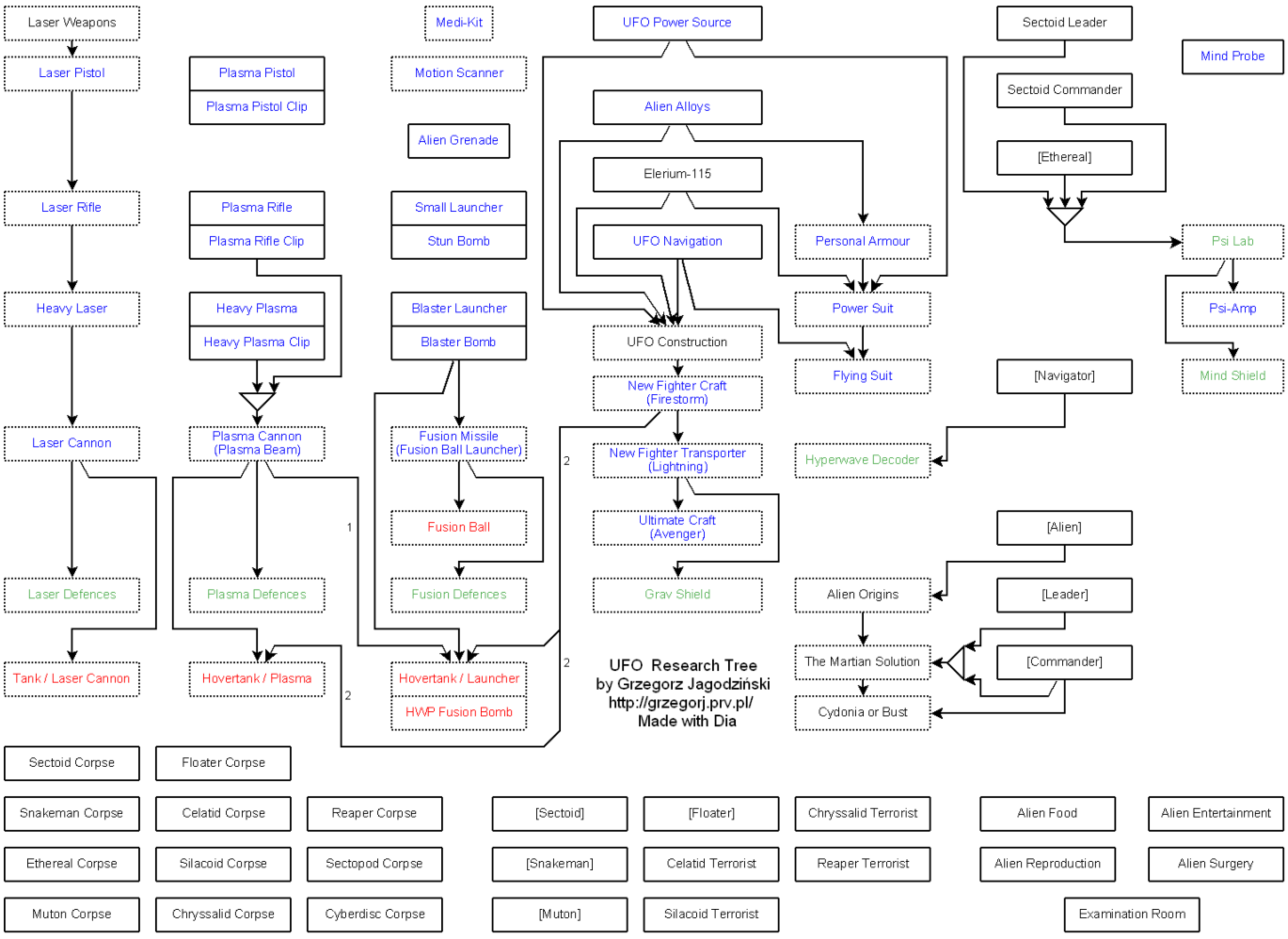
```
echo "Lade Quelldateien von github..."
wget https://github.com/SupSuper/OpenXcom/archive/master.zip
echo 'Entpacke "master.zip"...'
unzip master.zip
echo "Wechsle ins neu erstellte Quellen-Verzeichnis..."
cd OpenXcom-master
echo 'Entpacke Patch- und Vanilla-Files ins "bin/UFO/"-Verzeichnis...'
<Entpack- oder Kopierbefehl der Vanilla- und Patch-Files nach ./bin/UFO/>
d=`date +%Y-%m-%d-%H-%M`
echo Ändere Versions-String von \"1.0\" zu \"1.0 Nightly $d\"
perl -pi -e 's/SHORT "1.0"/SHORT "1.0 Nightly '$d'"/g' src/version.h
echo 'Lege Verzeichnis "build" an...'
mkdir build
echo 'Wechsle nach "build"...'
cd build
echo 'Konfiguriere make als "Release"...'
cmake -DCMAKE_BUILD_TYPE=Release ..
echo 'Compiliere OpenXcom...'
make -j10
echo 'Installiere OpenXcom...'
sudo make install
echo 'Entferne nicht mehr benötigte "master.zip"...'
rm ../../master.zip
```

Das Beispiel-Skript lädt die Quellen runter, entpackt sie, kopiert/entpackt die Original- bzw. Patch-Files ins UFO-Verzeichnis, ändert den Versions-String und kompiliert ein Release-build inkl. anschließender Installation. Die nicht mehr benötigte zip-Datei mit den Quellen wir auch noch entfernt.
Hinweis: Es ist nicht empfohlen, das Skript einfach zwei mal ablaufen zu lassen, wenn es möglich ist, dass sich einiges an der Ordner-Struktur geändert hat. Besser die alten OpenXcom-master-Ordner vor dem Ausführen löschen! (Lässt sich natürlich auch automatisieren, so wie vieles andere....) Man kann sich hier natürlich völlig austoben.

Tech-Tree

An dieser Stelle ein Hinweis: Für jemanden, der X-COM noch nicht nicht kennt, sollte der Tech-Tree aber erst mal egal sein. Wenigstens ein mal sollte man X-COM 1 oder 2 ohne Hinweise gespielt haben. Sonst geht der Funfaktor ggf. schneller verloren, als nötig. Für den Techtree gilt natürlich, dass er durch Mods teils ganz erhebliche Änderungen zum Original aufweisen kann, bis hin zu einer kompletten Konversion zu einem ganz anderen Spiel. Der originale Tech-Tree sieht so aus:

Vollständig



Spielstand-Infos

- Alien-Basen in OpenXcom-Spielständen: `alienBases` startet Liste der Alien-Basen. Steht bei einer dran: `discovered: true`, ist sie sichtbar.

Zurück zur Games-Datenbank und den LAN-Infos

From:
https://mwohlauer.d-n-s.name/wiki/ - **mwohlauer.d-n-s.name / www.mobile-infanterie.de**

Permanent link:
https://mwohlauer.d-n-s.name/wiki/doku.php?id=games:openxcom&rev=1588186197

Last update: **2020-04-29-20-49**