

Curved-Screen Star-Wars:

Huemer Elke 0055927
Köckerbauer Thomas 0055217
Kühr Martin 9956557
Schumacher Paul 0055460

Soundserver:

Das Spiel verwendet zur Soundausgabe einen veränderten Bergen sound server (Unterstützung für MP3-Dateien wurde hinzugefügt), der veränderte Sourcecode liegt im "bergen" Unterverzeichnis des cave Projekts.

Zum Compilieren wird libmad (ein GPL MP3 Player library) benötigt (<http://www.underbit.com/products/mad/>).

Starten des Soundservers:

Schritt 0 (nur nötig wenn der Sound von einem vorherigen programm "falsch initialisiert" wurde):

```
sfplay $HOME/constant_sorrow.wav       (irgnedeine wav-Datei mit sfplay abspielen)
Cntrl-C   nach 4 Sekunden ton
```

Schritt 1 (library path zum libmad muss beim Starten von snerd angegeben werden wenn libmad nicht in den system library paths installiert ist):

```
cd /home/vrpk/vrpk28/cave/bergen/src/server
env LD_LIBRARY_PATH=/home/vrpk/vrpk28/libmad-0.15.0b/.libs ./snerd -srate 48000
>& /dev/null &
```

Compilieren:

Im Projektverzeichnis existieren 2 Makefiles (Makefile und GNUmakefile). Makefile ist für die IRIX version, GNUmakefile ist für die Linux version. In beiden Fällen wird jedoch gnumake zum Compilieren verwendet wodurch auf IRIX mit "gmake -f Makefile" compiliert werden muss da gnumake wenn vorhanden die Datei mit dem namen GNUmakefile bevorzugt.

Commandline-Parameter:

```
-fs                   - Programm wird in Fullscreen anstatt im Fenster gestartet
                      (Nur bei Desktop versionen)
-joy                  - Eingabe über Joystick-Input anstatt über Keyboard-Input
                      (bei der Linux version) bzw. anstatt über Cave-Input
                      (bei der Cave version) (In der Windows version ist kein
                      joystick input möglich (=> nur keyboard input))
```

Verwenden der Joystick-Input Eingabe:

Die Joystick-Input Eingabe besteht aus 2 Teilen:

Dem Joystick-Input Server der im Spiel selbst ist (in der Windows version nicht vorhanden) und dem Joystick-Input Client (udpjoyclient), einen linux-programm das die joystick eingabe aus dem linux joystick api ausliest und per udp an den server (= das spiel) weiterschickt.

Den "udpjoyclient" im udpjoyclient Verzeichnis starten mit:

```
./udpjoyclient <server>
```

Als server muss der Recher angegeben werden auf dem das Spiel dann gestartet wird. Damit der udpjoyclient funktioniert muss das /dev/js0 device existieren und funktionieren (überprüfbar mit "jstest /dev/js0").

Danach das Spiel mit der commandline option "-joy" starten.

Belegung der Tasten bei Keyboard-Input:

Cursortasten	- Raumschiff drehen
a	- Beschleunigen
y	- Bremsen
q	- gegen den Uhrzeigersinn drehen
w	- im Uhrzeigersinn drehen
1,2,3,... bis 0	- Geschwindigkeit auf eine vordefinierte Geschwindigkeit ändern (1 ist am langsamsten, 0 am schnellsten)
ESC	- Programm beenden
Space	- Laser abfeuern
x	- Missile abfeuern
+, -	- nächstes/vorheriges Objekt für missilelock auswählen

Belegung der Buttons bei Joystick-Input (entsprechend dem Linux joystick api):

Achse 0	- Raumschiff nach links/rechts drehen
Achse 1	- Raumschiff nach oben/unten drehen
Achse 2	- Raumschiff um eigene Achse rotieren
Achse 3	- Geschwindigkeit einstellen
Button 0	- Laser abfeuern
Button 2	- Missile abfeuern
Button 3,4	- nächstes/vorheriges Objekt für missilelock auswählen
Button 5	- Programm beenden

Belegung der Buttons bei Cave-Input (wand):

Joystick	- Raumschiff drehen
Button 1	- Laser abfeuern
Button 2	- Missile abfeuern
Button 3 (Joystick in Ruhestellung)	- mittlere Geschwindigkeit
Button 3 (Joystick nach vorne gedrückt)	- volle Geschwindigkeit
Button 3 (Joystick nach hinten gedrückt)	- Stillstand