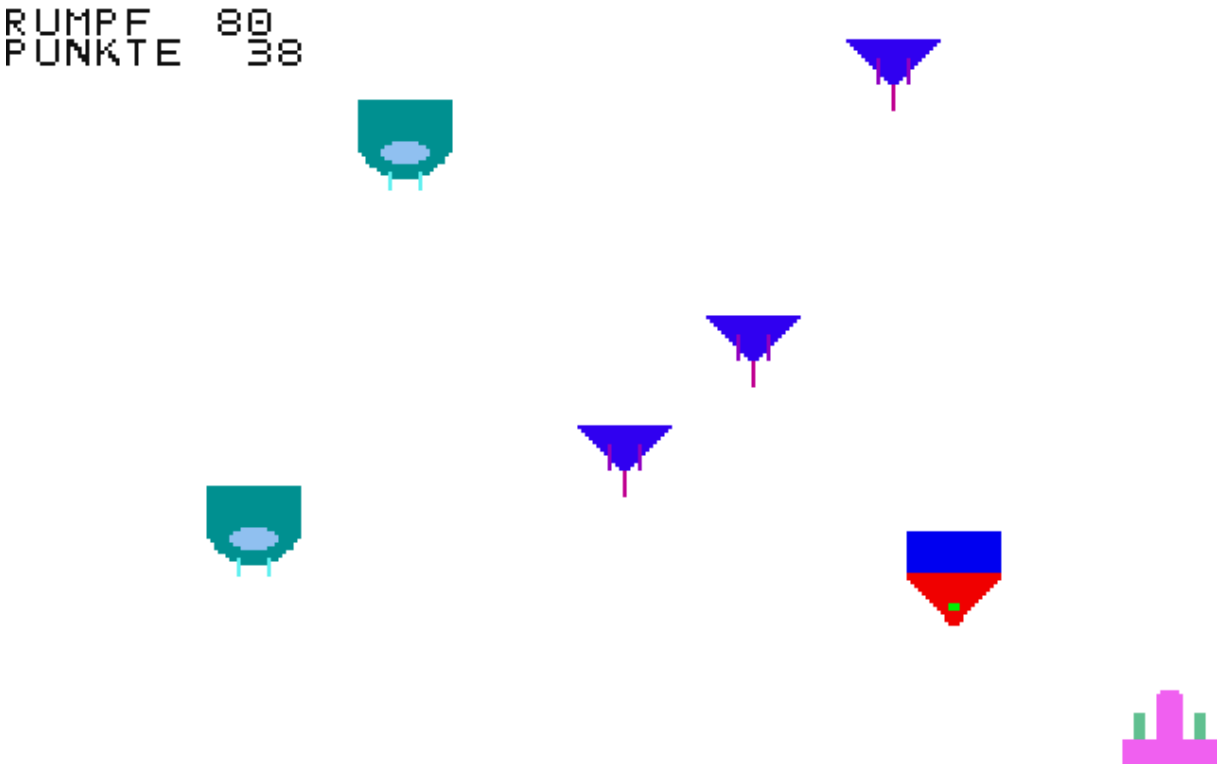
Space Fight Dokumentation



Space Fight
Installation und Codeerklärungen
Spiel Version 1.0
Handbuch Version 1.0

Inhalt

Einleitung

Das Spiel Space Fight ist ein Demonstrationsprogramm welches für folgende Ziele entwickelt wurde:

Höchstens 2 kb Code und Daten, RAM beliebig
Lauffähig ohne ein Betriebssystem unter einem 286-PC
VGA Grafik 320x200

In dem kleinen Spiel hat man ein Raumschiff, welches man mit der Nach-Rechts-Taste und der Nach-Links-Taste bewegen kann. Mit der Leertaste kann man kleine Raketen abfeuern, die zur Zerstörung der feindlichen Raumschiffe dienen. Bei jedem Abschuss gibt es Punkte, wobei ein Zusammenprall das eigene Raumschiff beschädigt. Bei einem Totalschaden ist das Spiel zuende.

Es gibt drei Raumschiffe, welche auch unterschiedlich viele Punkte geben:

4 Punkte

6 Punkte

8 Punkte

Dateiliste

- Ordner Roh
  - Dieser Ordner enthält alle Bilder im BMP Format
- Ordner Src
  - Dieser Ordner enthält den Quelltext spiel.asm
- Ordner Bin
  - Diskettenimage spiel.bin
  - Assembler nasm.exe
  - Diskettenimageprogramm rawrite.exe
  - Lauflängenbildkomprimierung bilder.exe
  - Schriftkomprimierung schrift.exe
  - 216-Farbauswähler farben.exe
- Ordner Dok
  - Diese Dokumentation handbuch.pdf
- Installationsdatei install.bat

Installation

Um das Spiel zu installieren, muss das 2 kb große Image auf die Diskette kopiert werden. Am einfachsten geht dies mit dem beiliegenden Installationsprogramm install.bat. Beachten Sie, dass alle Daten auf der Diskette gelöscht werden. Ist das Diskettenimage erzeugt, so muss der PC neugestartet werden und die Diskette muss sich im Laufwerk befinden. Sollte ein Virenscanner Probleme verursachen, muss zunächst die Diskette entfernt werden. Ist der PC am Neustarten, muss sie sofort wieder eingelegt werden. Zusätzlich muss noch beachtet werden, dass das Diskettenlaufwerk in der Bootsequenz vor der Festplatte liegt. Möglicherweise müssen hier Einstellungen im BIOS vorgenommen werden.

Das Spiel selbst benötigt außer der Diskette keinen weiteren Festplattenspeicher und greift auch nie auf einen anderen Datenträger zu.

Funktionsübersicht

Funktion inttostr
Parameter: ds:si: Pointer auf 6 Byte, ax: Zahl
Rückgabe: -
Diese Funktion wandelt die Zahl in einen nullterminierten String um.

Funktion keypress
Parameter: -
Rückgabe: -
Diese Funktion wird über einen Interrupt aufgerufen und stellt den Tastaturtreiber dar.

Funktion getkey
Parameter: -
Rückgabe: al: Tastaturcode, ah: Pufferstatus
Diese Funktion ließt einen Tastencode aus dem Tastenpuffer aus. Ist der Puffer leer, so wird ah auf 1 gesetzt, sonst auf 0. Der Tastencode wird in al gespeichert und ist nur gesetzt, wenn ah=0 ist.

Funktion print
Parameter: ds:si: Pointer auf nullterminierten String, di: Pixelposition, ch: Farbe
Rückgabe: -
Diese Funktion plottet einen String in den Bildpuffer, wobei in di die Pixelposition angegeben werden muss (x+y*320).

Funktion drawbuffer
Parameter: -
Rückgabe: -
Diese Funktion wartet auf einen Vertical Retrace und kopiert dann den Bildpuffer auf den Bildschirm.

Funktion clearscreen
Parameter: -
Rückgabe: -
Diese Funktion löscht den Bildpuffer.

Funktion expandfont
Parameter: -
Rückgabe: -
Diese Funktion entpackt die komprimierte Schrift an die Stelle 4000d

Funktion setchar
Parameter: al: Zeichennummer, di: Pixelposition ch: Farbe
Rückgabe: -
Diese Funktion plottet ein Zeichen in den Bildpuffer. Die Pixelposition wird in di angegeben (x+320*y).

Funktion genexplode
Parameter: ds:di: Pointer auf Zielbild
Rückgabe: -
Diese Funktion generiert ein sehr primitives Explosionsbild

Funktion expandpic
Parameter: ds:si: Quelldaten, ds:di: Zielbild
Rückgabe: -
Diese Funktion dekomprimiert eine Lauflängenbild.

Funktion showpic
Parameter: ds:si: Pointer auf Bild, ax: Y-Position, di: X-Position
Rückgabe: -
Diese Funktion zeichnet ein Bild in den Bildpuffer an den angegebenen Positionen. Dabei wird beachtet, dass es in der Y-Richtung abgeschnitten werden kann. Die X-Richtung muss sich innerhalb des Bildschirmes befinden.

Funktion timer
Parameter: -
Rückgabe: -
Diese Funktion wird von einem Interrupt aufgerufen und sendet ein Timersignal an das Spiel. Zusätzlich wird der alte Timer aufgerufen.

Funktion ctrlgame
Parameter: -
Rückgabe: -
Diese Funktion kontrolliert das Menü, das Spiel und das Game Over.

Funktion objekt
Parameter: ax: X-Position, dl: Y-Position, dh: Typ, bl: Geschwindigkeit, bh: Richtung
Rückgabe: -
Diese Funktion erstellt ein neues Objekt mit den angegebenen Daten.

Funktion moveup
Parameter: -
Rückgabe: -
Diese Funktion bewegt ein Objekt mit der Geschwindigkeit nach oben.

Funktion delobj
Parameter: di: Objektnummer
Rückgabe: -
Diese Funktion löscht das angegebene Objekt.

Funktion movedown
Parameter: -
Rückgabe: -
Diese Funktion bewegt ein Objekt mit der Geschwindigkeit nach unten.

Funktion random
Parameter: cx: Anzahl der möglichen Werte
Rückgabe: dx: Zufallswert
Diese Funktion gibt einen Zufallswert zurück im Bereich [0,cx-1] zurück.

Funktion checkrakete
Parameter: di: Raketenobjekt
Rückgabe: -
Diese Funktion überprüft, ob die angegebene Rakete ein gegnerisches Raumschiff getroffen hat.

Funktion checkcrash
Parameter: di: Raumschiff
Rückgabe: -
Diese Funktion überprüft ob ein gegnerisches Raumschiff das eigene gerammt hat.

Datenabspeicherung

Hier soll noch einmal kurz erläutert werden, wie die Schrift und die Bilder abgespeichert werden.

Schrift

Die Schrift besteht aus 26 Großbuchstaben A-Z und 10 Zahlen 0-9. Jedes Zeichen ist 6 Pixel breit und 7 Pixel hoch. Zwischen den Zeichen besteht kein Freiraum. Alle Zeichen hintereinander ergeben eine Länge von 216 Pixel. Nun werden immer 8 Pixel in ein Byte zusammengefasst. Das macht pro Reihe 27 Byte und insgesamt 27 Byte x 7 = 189 Byte. Die Schrift wird dann durch die Funktion expandfont an die Stelle 4000 entpackt. Jeder Pixel benötigt nun 1 Byte und enthält entweder 0 oder 255. Dadurch wird das Zeichnen beschleunigt.

Bilder

Alle Bilder, bis auf das Explosionsbild, werden durch eine Lauflängencodierung komprimiert. Dabei werden alle Pixel von oben links nach unten rechts hintereinandergelegt. Nun überprüft das Komprimierungsprogramm, wie viele Pixel einer Farbe hintereinander liegen. Liegen jedoch mehr als 31 Pixel hintereinander, so wird eine neue Linie begonnen. Nun werden Linienlänge und Farbnummer zusammen in ein Byte gepackt. Die unteren 3 Bits sind für die Farbnummer, die oberen 5 Bits für die Länge. Da jedoch nie mehr als vier Farben benötigt

werden, ist die Farbtabelle von 8 Bytes auf 4 Bytes gekürzt worden. Das Spiel dekomprimiert die Bilder zunächst um sie später schneller zeichnen zu können.

Explosion

Die Explosion ist eine Reihe zufälliger Pixel hintereinander. Es gibt dabei eine Explosionsfarbtabelle von 6 Farben, wobei 3 jedoch schwarz sind, und die anderen rot, gelb und orange. Die Generierung des Bildes erfolgt über die Funktion genexplode.

Palette

Die Farbpalette nutzt von den 256 möglichen Farben nur 216. Um nicht 216*3=648 Bytes zu verschwenden wurde ein Palettengenerator integriert. Jede Farbe (Rot, Grün, Blau) gibt es jeweils in 6 Helligkeitsstufen, was 6x6x6=216 mögliche Farben macht. Zu beachten ist, dass die Farbeinstellungen nicht von 0-255, sondern von 0-63 geht.

Bootvorgang

Damit die Diskette ohne Betriebssystem überhaupt starten kann, muss sie einen Bootsektor enthalten. Dies ist immer Sektor 0. Damit das Bios ihn als Bootsektor erkennt, müssen die letzen 2 Bytes 170 und 85 enthalten (aa55h). Zusätzlich sind noch einige Dinge zu beachten:

Zuerst sollte der Stack gesetzt werden, welcher hier in das Segment 3000h gesetzt wird. Als nächstes wird der erste Sektor (Sektor 0) an die Stelle 1000h:0000h kopiert, da das Bios diesen Sektor immer an die Stelle 0000h:7c00h lädt. Da nicht ganz klar ist, welcher Speicher noch benötigt wird, ist diese Kopieraktion ganz gut.

Als nächstes müssen die restlichen 3 Sektoren nachgeladen werden, um die vollen 2048 Bytes zu erhalten. Nun folgen noch Aufgaben, wie das Setzen eines 20 Hz Timers, das Starten des Tastaturtreibers und die Umschaltung in den Grafikmodus mit der Palettengenerierung. Zum Schluss werden noch die Daten dekomprimiert und das Spiel gestartet.

Quelltextanpassungen

Möchte man das Spiel leicht abändern, so kann man hier schauen, was leicht zu ändern ist.

Bilder

Die Bilder sind durchnummeriert von 1-5:

1. Eigenes Raumschiff
2. Gegner 1
3. Gegner 2
4. Gegner 3
5. Rakete

Jedes Bild ist 25 Pixel hoch, und bis auf die 5 Pixel breite Rakete, sind alle auch 25 Pixel breit. Dies sollte am besten so belassen werden. Mit dem mitgelieferten Farb- und Komprimierungsprogrammen können andere Raumschiffe erstellt werden. Zu beachten ist jedoch der knappe Speicherplatz und die Beschränkung auf 4 Farben.

```
dat_bild1:
db 65 ;Anzahl der Linien
db 25 ;Breite, am besten so lassen
db 0,197,134,0 ;4-Farbtabelle
db 248, 248, 248, 248, 88, 41, 152.... ;Bilddaten
```

Die Anzahl der Linien wird beim Komprimieren immer mit angegeben, die Farben müssen jedoch selbst mit dem Farbwahlprogramm ausgewählt werden.

Schrift

Die Schrift lässt sich leicht anpassen. Einfach das Schriftbild abändern und erneut komprimieren. Die Breite und Höhe sollte nicht geändert werden. Die Komprimierung übernimmt das entsprechende Programm.

Explosion

Hier kann man die Farben in der Farbtabelle ändern bei dat_farbexplo. Möchte man die Anzahl auch ändern, so muss man Farben hinzufügen und in der Funktion genexplode die Zeile mov cx,6 vor der Zeile call fkt_random ändern.

Raumschiffanzahl

In der Funktion ctrlgame muss die Zeile cmp [3472],byte 5 so abgeändert werden, dass die Zahl +1 die Anzahl der Raumschiffe angibt. Es sollte jedoch beachtet werden, dass mit Schüssen zusammen maximal 64 Objekte gleichzeitig auf dem Bildschirm sein können. Rechnet man mit maximal 8 Schlüssen, so kann man also 56 höchsten einstellen.

Mehr Speicherplatz

Hat man große Raumschiffbilder, so kann der Speicherplatz eng werden. Sollte er nicht ausreichen, so ist es am einfachsten, die 2048 Bytes zu überschreiten. Es sind dann jedoch 2 Anpassungen nötig: Die Zeile times 2048-(ende-boot_anfang) db 0 (ganz am Ende) muss entfernt werden und ganz am Anfang nach mov bx, 512 muss die Zeile mov ax,0203h geändert werden. Die letzten zwei Ziffern geben die Anzahl der nachzuladenden Sektoren an, es dürfen jedoch nicht mehr als 17 Sektoren nachgeladen werden. Gibt man mov ax,0207h an, so kann die Imagedatei bis zu 4 kb groß sein, was eigentlich ausreichen sollte.