

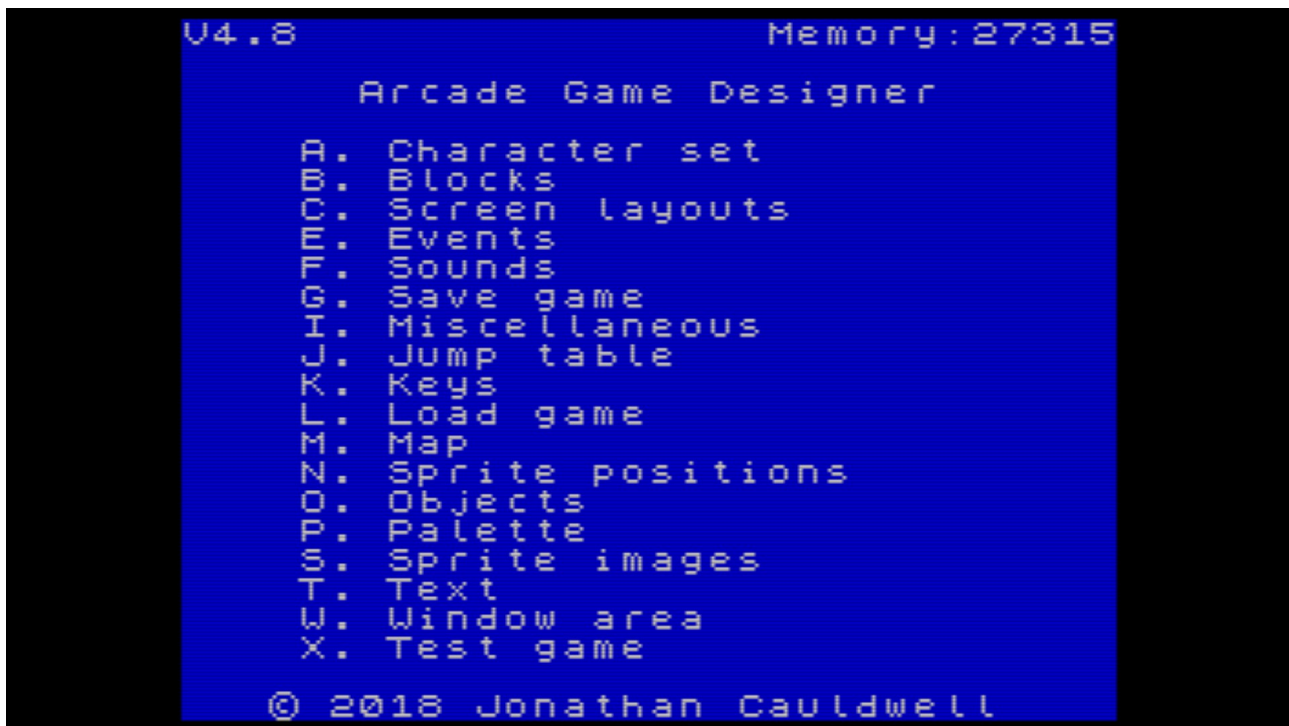
Arcade Game Designer

Copyright Jonathan Cauldwell, 2008-2019

(Deutsche Übersetzung : Björn-Lars Feddersen : Dez.2019)



Anleitung für Version 4.8



(Die Übersetzung der Anleitung ist nicht perfekt. Bei Fehlern und Textverbesserungen eine eMail an: [bjoern-lars.feddersen\(at\)web.de](mailto:bjoern-lars.feddersen(at)web.de) Betreff: AGD 4.8)

Inhalt

Vorwort	Seite 03
Ein Spiel erstellen	Seite 04
Hauptmenü, Window area	Seite 05
Palette, Keys	Seite 06
Block Design	Seite 07
Screen Layouts	Seite 08
Sprite Images	Seite 09
Sprite Positions	Seite 10
Objects	Seite 11
Map Layout	Seite 13
Jump Table	Seite 14
Sounds	Seite 15
Save Game, Load Game	Seite 16
Test Game, Miscellaneous	Seite 17
Text, Events	Seite 19
Befehle, Funktionen und Variablen	Seite 23
Character Set, Informationen	Seite 33

Wie der PGD (Platform Game Designer) und SEUD (Shoot-Em-Up Designer) so ist auch der Arcade Game Designer ein perfektes Werkzeug um seine eigenen Spiele zu erstellen.

Dieser ist jedoch etwas komplexer als die beiden anderen.

Scrollende Shoot-Em-Ups lassen sich damit nicht erstellen. Er bietet jedoch die Möglichkeit eine Vielfalt an Arcade Spielen ohne all zu großen Aufwand zu entwickeln. Vorausgesetzt dass du geduldig bist und dich auch von kleinen Rückschlägen nicht entmutigen lässt.

Spiele die mit dem AGD erstellt wurden, können frei vertrieben und verbreitet werden.

AGD wird von vielen Autoren benutzt und ist verantwortlich für eine ganze Reihe an Spielen für den ZX Spectrum. Einige Beispiele sind Apulija-13, Hedgehogs, Kyd Cadet, Land of Mire Mare und Trooper:Point 5.

Alle diese Spiele sind durchaus empfehlenswert und lassen sich in den Archiven von www.worldofspectrum.org finden.

Das ist jedoch nicht alles. AGD bietet jetzt auch : Explosionseffekte, „Magischen Staub“, „Kondensstreifen“, größere Sprites, „Lasereffekte“, „Laufende Textnachrichten“ und die Möglichkeit Spiele mit einem professionellerem Aussehen zu gestalten.

Der AGD hat seine eigene „built-in language“. Eine vereinfachte Sprache mit einfachen Funktionen und ein paar „built-in“ Variablen. Du kannst auch den Spiele Code mit den integrierten Vorlagen an dein Spielprojekt anpassen.

Die Anzahl der verwendbaren Befehle und Funktionen ist jedoch begrenzt.

Eine Syntaxprüfung ist im AGD nicht vorhanden.

Wenn das Programm einen Befehl nicht versteht, springt der Cursor in die entsprechende Textzeile. Somit weiß der Anwender wo der Fehler zu suchen ist.

Weil ich einen Spiele Designer für jedermann entwickeln wollte, ist das Programm kostenlos auf meiner Website als Download erhältlich. Das Urheberrecht verbleibt jedoch beim Autor.

Spenden sind nicht Notwendig. Wenn du aber deine Anerkennung für meine Arbeit zeigen möchtest, kaufe doch eines der Programme oder Spiele auf www.cronosoft.co.uk.

AGD 4.8 ist die aktuellste Version. Die Benutzung erfolgt jedoch ohne jegliche Garantie.

Bitte erwartet keinerlei Support via eMail. Ich Beantworte jedoch gerne Fragen oder nehme Verbesserungsvorschläge im AGD Forum an. Wenn es die Zeit zulässt erscheinen ab und an neue Updates. Bis dahin speichert euer Projekt so oft wie möglich ab. Bei Verwendung eines Emulators vorzugsweise mit der „Snapshot“ Funktion.

Das wichtigste von allem. Habt Spass beim Erstellen eurer Projekte. Ich bin dann mal auf dem Weg zu Ripley um mir einen Drink zu genehmigen.

Jonathan Cauldwell, 04. April 2008.

@zxspectrumdev

<http://www.spanglefish.com/egghead/>

(Ich bin nicht mehr auf Twitter. Das ZXSpectrumDev-Konto ist jemand anderes, der vorgibt, ich zu sein)

Ein Spiel erstellen

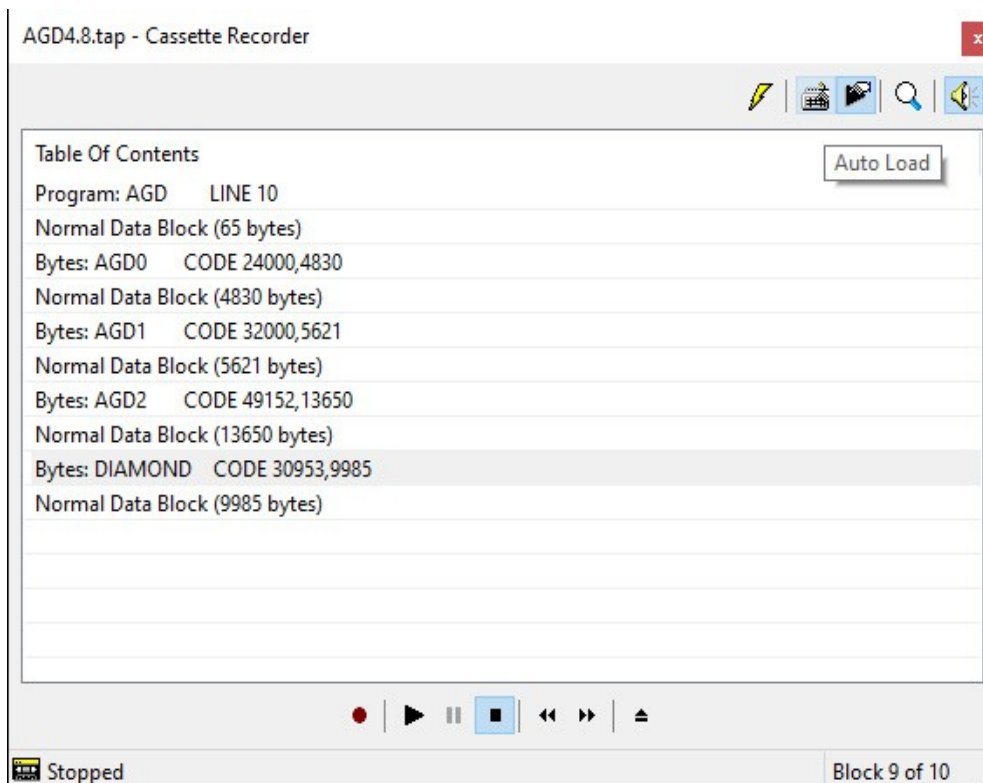
AGD erfordert ein häufiges wechseln der Kassetten für das Laden und Speichern. Benutzt ihr einen Emulator, empfiehlt es sich die Funktion zum automatischen Laden beim „Tapewechsel“ zu deaktivieren. Ein Kassettenwechsel im Emulator könnte ansonsten zu einem kompletten Verlust der bisherigen Arbeit an dem Spiel führen.

Also „Load Tapes Automatically“ Deaktivieren!

Diese Option ist bei jedem Emulator individuell zu finden. Folgendes bezieht sich auf den „SPECTACULATOR“ Emulator.

Im Emulator auf den Reiter „View“ und „Cassette Recorder“ auswählen.

Im Fenster des Tape Recorder befinden sich fünf Symbole. Mit dem zweiten Symbol von links lässt sich „Auto Load“ ein-, bzw. ausschalten. Jetzt kann die Entwicklung des Projekts mit dem AGD beginnen.

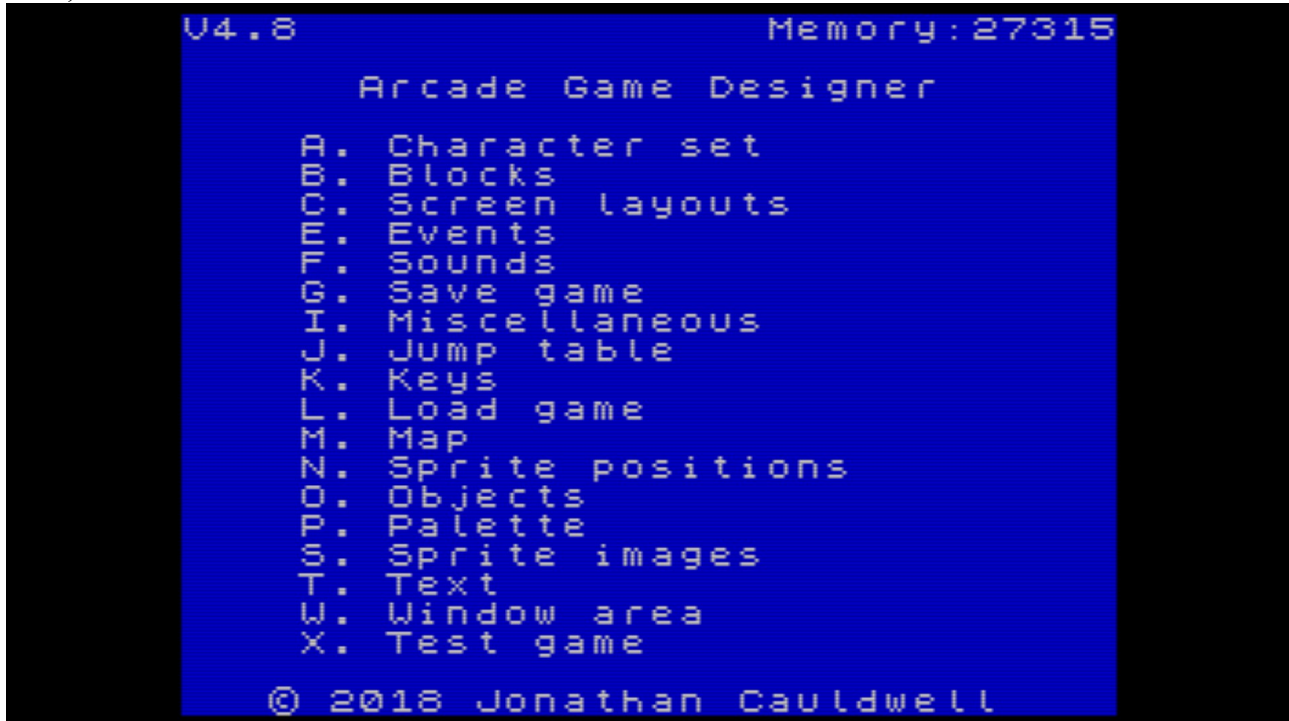


Grundsätzlich sind Spiele eine Kombination aus verschiedenen Elementen. Zeichenblöcke werden kombiniert um den Inhalt des Spielbildschirms zu füllen und zu gestalten. Sprites (einschließlich der eigenen Spielfigur) bekommen eine Startposition in jedem „Game Screen“ und eine Zuordnung was mit diesem Sprite geschehen soll. Für jeden Sprite Typ wird dann noch die Logik festgelegt. Also für den Spielcharakter die Gegner und die Objekte.

Wie z.B. das Einsammeln von Gegenständen, die Kollision mit anderen Objekten oder ob eine Mauer die Spielfigur aufhält oder sich durch diese hindurch bewegen kann.

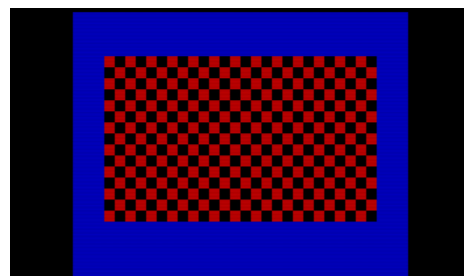
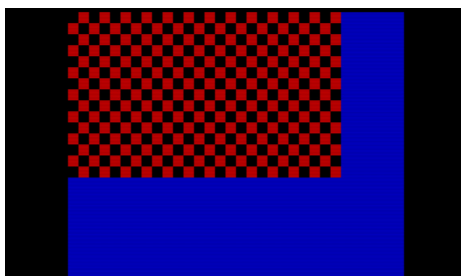
Hauptmenü

Im Hauptmenü kannst du Auswählen was du gerade bearbeiten möchtest. Einfach über die Tastatur den jeweiligen Buchstaben drücken. Wie viel Speicherplatz dir noch für dein Projekt zur Verfügung steht, ist oben rechts am Bildschirm ersichtlich.



Window area (Taste W)

In der „Window area“ kann die Größe und Position des Spielfeldes eingestellt werden. Das rot/schwarze Karo Muster stellt also die komplette Spielfläche dar. Mit den Cursor Tasten kann das ganze Spielfeld verschoben werden



(Spielfeld verschoben mit Hilfe der Cursor Tasten)

Wenn die Position und Größe der Spielfläche verändert wird, kann sich die Datenmenge ändern, die für jede Spalte oder Zeile von Blöcken in jedem Bildschirm erforderlich ist, und somit die aktuellen Ebenen ungültig machen. In diesem Fall fragt das Programm ob sie die Kartendaten löschen möchten, bevor Sie fortfahren können.

Am besten überlegen sie sich gleich zu Anfang wie groß das Spielfeld sein soll und bleiben dann bei der von ihnen gewählten Einstellung.

Tastenfunktion:

1 = Spielfeld verkleinern

2 = Spielfeld vergrößern

Q = Spielfeld kürzen

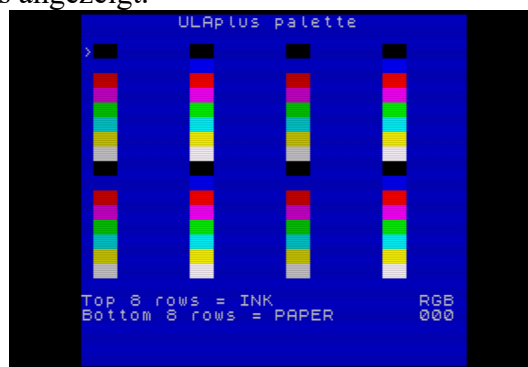
A = Spielfeld verlängern

Cursor Tasten (Position des Feldes verschieben)

Palette (Taste P)

Seit Version 3 unterstützt der AGD Standardmäßig ULApplus (64 Farben Modus) fähige Spectrum Computer und Emulatoren.

Mit dem ULA+ Palette Editor können sie sich in den 64 Farben der 4-Farben-Nachschlagetabellen (CLUTs) bewegen und diese ändern. Der RGB-Wert der ausgewählten Farbe wird in der unteren rechten Ecke des Bildschirms angezeigt.



Tastenfunktion:

R = ändert die rote Farbkomponente

G = ändert die grüne Farbkomponente

B = ändert die blaue Farbkomponente

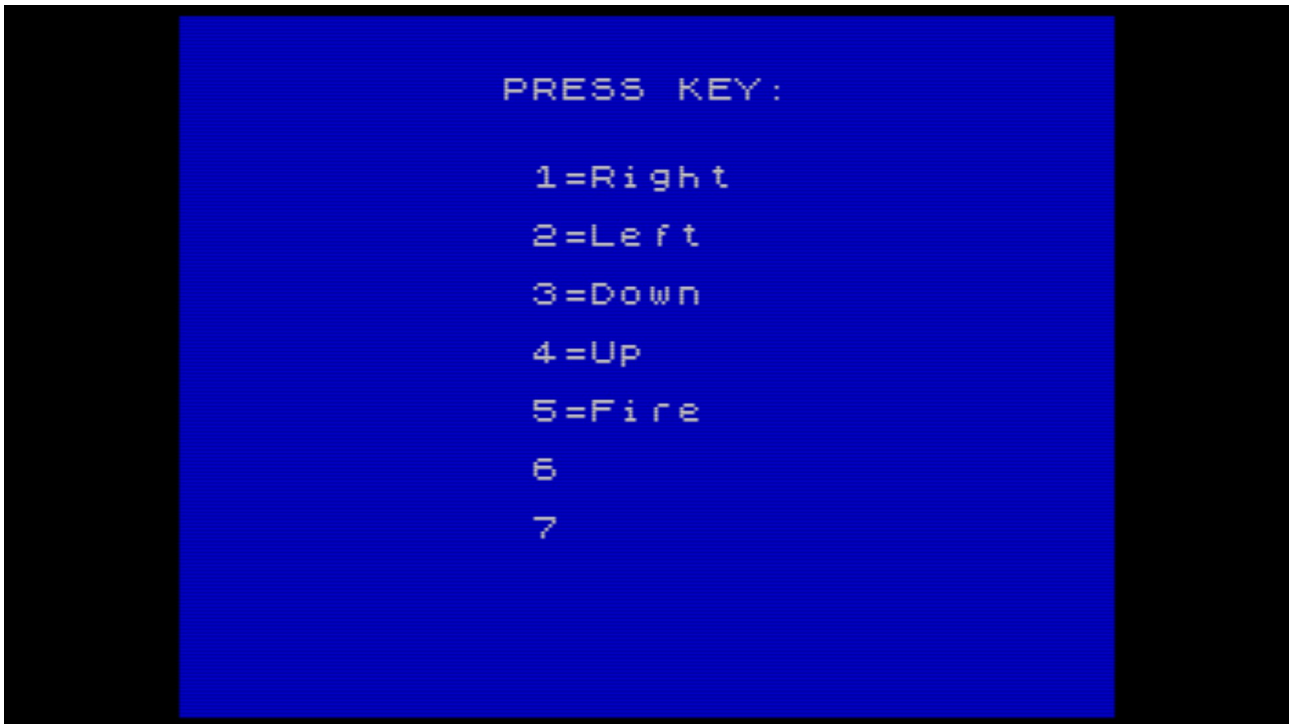
ENTER = Zurück ins Hauptmenü

Keys (Taste K)

Hier können sie festlegen welche Tasten des Keyboards für die Steuerung des Spiels verwendet werden. Sieben Tasten können belegt werden und der Kempston Joystick ist auf die ersten fünf Tasten „Vorprogrammiert“. Also die Richtungen (links,rechts,oben und unten) und die Feuertaste des Joysticks.

AGD ist standardmäßig auf Tastatur eingestellt. Wenn sie jedoch technisch versiert sind, können sie den Kempston-Joystick in Ihrem Spiel über POKE 32005,1 oder den Sinclair-Joystick mit POKE

32005,2 verwenden. Für die Tastatur POKE 32005 mit einem anderen Wert.



Block Design (Taste B)

Hier wird das Spielfeld zum Leben erweckt. Mit den Blöcken werden z.B. Mauern, Plattformen, Leitern oder Designelemente gestaltet. Jeder Block hat eine Größe von 8 x 8 Pixeln und wird einem bestimmten Typ zugeordnet.

Die verschiedenen Typen sind :

EMPTY SPACE
NORMAL PLATFORM

SOLID WALL
LADDER

FODDER

- Spielfigur kann sich frei durch diesen Raum bewegen.
- Spielfigur bewegt sich frei an Plattformen von links, rechts und unten vorbei.
- Ein Block der aus jeder Richtung unpassierbar ist.
- Spieler bewegt sich frei durch Leiter Blöcke aus jeder Richtung.
- Wie SOLID WALL, kann mit bestimmten Funktionen entfernt werden.

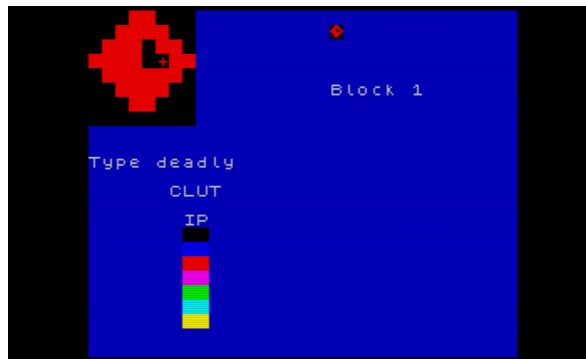
DEADLY

- Sprites bewegen sich hindurch (Eine Kollision hiermit kostet ein Spieler Leben).

CUSTOM

- Sprites bewegen sich hindurch (Benutzerdefinierte Funktion bei Kontakt).

Das Kreuz lässt dich mit den Cursortasten durch den Block bewegen. Die „Space“ oder „0“ Taste setzt oder löscht ein Pixel an dieser Stelle des Blockes. Die Farbe kann mit den Tasten (P,I,B) verändert werden. Auch ist es möglich die Helligkeit zu ändern oder den Block mit einer Farbe „blinken“ zu lassen.

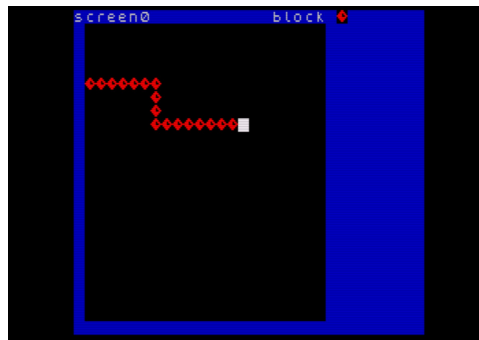


- Q = Nach links durch die Liste der Blockeigenschaften bewegen.
- W = Nach rechts durch die Liste der Blockeigenschaften bewegen.
- L = Letzter Block.
- N = Nächster Block.
- P = Paper colour (Hintergrund).
- I = Ink colour (Farbe des Objekts).
- B = Verändert die Helligkeit oder lässt den Block „blinken“.
- M = Kopiert den aktuellen Block in die Zwischenablage.
- K = Fügt den Block aus der Zwischenablage ein.
- C = Aktuellen Block löschen.
- X = Einen neuen Zeichenblock erstellen.
- D = Aktuellen Zeichenblock löschen.
- ENTER = Zurück zum Hauptmenü.

Screen Layouts (Taste C)

Mit dem Cursor bewegen sie sich durch das Spielfeld. Den Block den sie an die Cursorposition platzieren möchten wählen sie über die Tasten (1 und 2). Alternativ können sie auch den „Fast Draw“ Modus mit der Taste (F) aktivieren. Dabei wird der aktuelle Block automatisch mit Bewegung der Cursortasten an der Cursorposition eingefügt. Um den „Fast Draw“ Modus wieder zu verlassen erneut Taste (F) drücken.

Es ist auch möglich einen kompletten Bildschirm Inhalt zu kopieren (Taste M) und ihn in einen neuen Bildschirm einzufügen (Taste K) um diesen dann weiter bearbeiten und anpassen zu können.



Tastenfunktion:

1	= Nach links durch die verfügbaren Blocks schalten.
2	= Nach rechts durch die verfügbaren Blocks schalten.
F	= Fast Draw Modus aktivieren/deaktivieren.
M	= Den aktuellen Bildschirm markieren.
K	= Einfügen des markierten Bildschirms.
N	= Nächster Bildschirm.
P	= Vorheriger Bildschirm.
X	= Neuen Bildschirm erstellen.
D	= Aktuellen Bildschirm löschen.
ENTER	= Zurück zum Hauptmenü

Sprite Images (Taste S)

Sprites haben eine Größe von 16 x 16 Pixeln und können beliebig viele Frames (Einzelbilder) haben. Diese werden z.B. benötigt um für eine Spielfigur eine Bewegungsanimation zu erstellen. Mit den Cursortasten können sie sich frei durch den Sprite Block bewegen. Mit den Tasten „SPACE“ oder „0“ werden die einzelnen Bildpunkte gezeichnet oder entfernt.

Da die Kollisionserkennung von AGD koordinatenbasiert ist, wäre es ratsam den 16x16-Bereich so weit wie möglich auszufüllen. Besonders beim Spieler und Gegner Sprite ist dies Sinnvoll.

Spiele wie Monty Mole, Egghead oder Blob sind Beispiele für eine 16x16 große Pixelfläche.

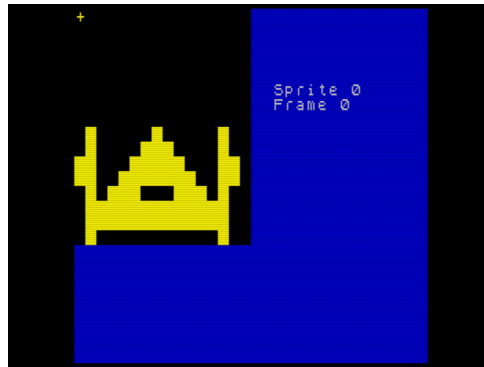
Bei Objekten die die Spielfigur einsammelt ist das vollständige ausfüllen der Fläche nicht ganz so wichtig. Der Spieler bemerkt es kaum wenn er ein Objekt aus einer Entfernung von 3 Pixeln aufnimmt.

Der Sprite-Editor verfügt auch über einen „Notizblock“ (Scratchpad).

Es ist möglich SCREEN\$ Dateien von einem Kassetten Image (Tape) zu Laden (Taste L).

Sie können ein 16x16 Pixel Bereich aus dem Scratchpad mit der Taste (G) in die Zwischenablage kopieren. Mit der Taste (K) wird der Bereich aus der Zwischenablage in den Sprite Block eingefügt. Vorsicht, einige Emulatoren laden eine TAP Datei automatisch sobald diese „eingelegt“ wird. Dies wird AGD aus dem Speicher löschen, einschließlich der Arbeit die bis dahin an dem Programm getätigt wurde.

Vorher unbedingt die „Auto Load“ Funktion im Emulator deaktivieren.



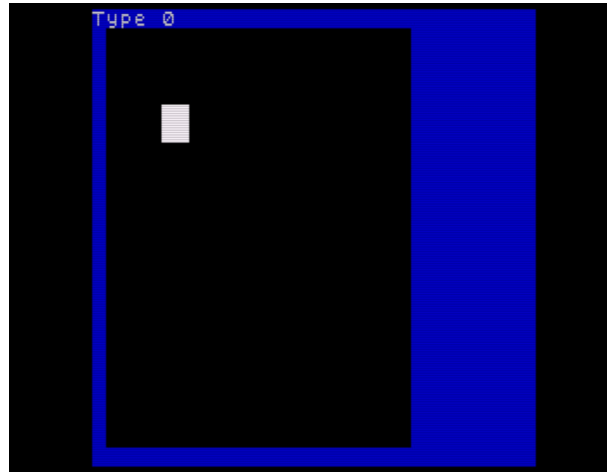
Tastenfunktion:

X	= Sprite einfügen.
D	= Sprite löschen.
C	= Aktuelles Sprite löschen.
G	= Kopiert ein Sprite aus dem „scratchpad“ in die Zwischenablage.
L	= Lädt ein „scratchpad“ Bild.
M	= Kopiert einen Sprite in die Zwischenablage.
K	= Fügt einen Sprite aus der Zwischenablage ein.
H	= Sprite horizontal spiegeln.
V	= Sprite vertikal spiegeln.
N	= Nächstes Sprite auswählen.
P	= Vorheriges Sprite auswählen.
I	= Neues Bild einfügen.
R	= Bild entfernen.
F	= Nächstes Bild.
ENTER	= Zurück zum Hauptmenü.

Sprite Positions (Taste N)

Auf diese Weise können Sie Sprites mit den Cursortasten an ihren Startpositionen für jede Ebene positionieren. Es stehen 8 Spritetypen zur Verfügung, und Sprites können wie folgt eingerichtet werden: Sie können Boni, feindliche Sprites, die Startposition des Spieler-Sprites oder jeden anderen von ihnen eingerichteten Spritetypp positionieren. Sprite-Bilder sind von Sprite-Typen getrennt. Bilder sind nur die Bilder, die sie im Sprite-Editor zeichnen, während das Verhalten eines

Sprites durch seinen Typ definiert wird. Typ 0 wird normalerweise für das Sprite (oder die Sprites) des Spielers verwendet, obwohl sie dies ändern können. Sie müssen jedem sich bewegenden Gegenstand, z. B. Außerirdischen, Kugeln oder Sammelbaren Objekten, einen Typ zuweisen und ihre Bewegungen mit Hilfe einer einfachen Skriptsprache bestimmen. Informationen zum bewegen von Sprites finden sie im Abschnitt „Events“.



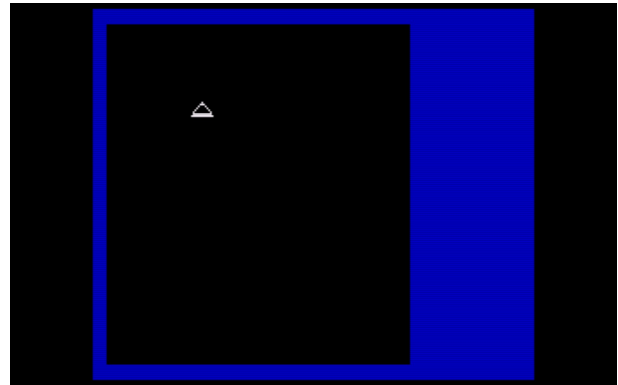
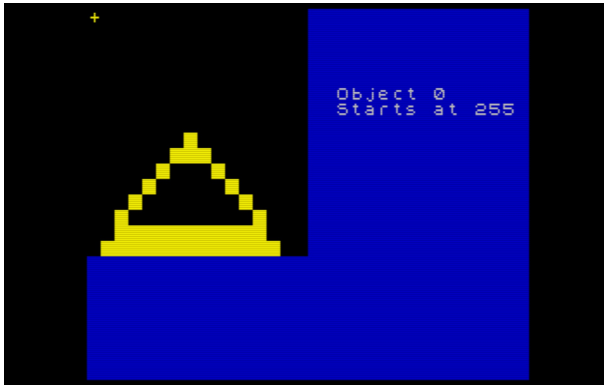
Tastenfunktion:

N	= Nächster Bildschirm.
P	= Vorheriger Bildschirm.
Q	= Bewege nächsten Sprite.
I	= Sprite Image ändern.
T	= Sprite Typ ändern.
D	= Aktuelles Sprite vom Bildschirm löschen.
X	= Ein neues Sprite dem Bildschirm hinzufügen.
ENTER	= Zurück zum Hauptmenü.

Objects (Taste O)

Im Unterschied zu Sprites sind Objekte statische Elemente, die ein Sprite möglicherweise aufnimmt oder ablegt. Wie Sprites bestehen sie aus Bildern mit 16 x 16 Pixeln, obwohl sie nur einen Frame haben. Diese können ähnlich wie Sprite-Bilder bearbeitet werden. Für jedes Objekt müssen sie zu Beginn des Spiels seine Position definieren. Im Gegensatz zu Sprites erscheinen Objekte nicht jedes

mal wieder, wenn der Spieler einen Raum betritt, sodass sie in einem Raum aufgenommen und in einem anderen abgelegt werden können. Dies macht sie ideal für Arcade-Abenteuerspiele. Der Startbildschirm wird mit den Tasten Q und W gewechselt. Ein Startbildschirm von 255 zeigt an, dass das Objekt das Spiel im Inventar des Spielers startet. Objekten, die zu Beginn des Spiels fehlen, d.h. auf keinem Bildschirm oder im Inventar des Spielers erscheinen, sollte ein Startbildschirm von 254 zugewiesen werden. Drücken Sie P, um ein Objekt an der festgelegten Position auf dem Startbildschirm zu platzieren.



Tastenfunktion:

X	= Objekt einfügen.
D	= Objekt löschen.
C	= Aktuelles Bild löschen.
M	= Kopiert Bild in die Zwischenablage.
K	= Fügt Bild aus der Zwischenablage ein.
N	= Nächstes Objekt.
L	= Letztes Objekt.
W	= Nächster Startbildschirm.
Q	= Vorheriger Startbildschirm.
P	= Positioniert das Objekt auf dem Startbildschirm.

Objekte können im Ereigniscode mit **GOT**, **GET**, **PUT** und **DETECTOBJ** behandelt werden. Darüber hinaus speichert die **OBJ**-Variable das Ergebnis des letzten **DETECTOBJ**-Befehls, obwohl sie es bei Bedarf auch für andere Zwecke verwenden können. **IF GOT** „n“ ist erfüllt, wenn der Spieler das Objekt „n“ in seinem Inventar hat. **GET** „n“ legt das Objekt „n“ im Inventar des Spielers ab (unabhängig davon, wo sich das Objekt gerade befindet). **PUT** „n“ legt ein Objekt an der aktuellen Sprite-Position ab und **DETECTOBJ** prüft, ob das aktuelle Sprite ein Objekt berührt, und platziert das Ergebnis in der **OBJ**-Variablen. Wird an der Sprite-Position kein Objekt erkannt, wird der Wert 255 zurückgegeben.

Mit dem Folgenden Code werden beispielsweise Objekte durch die sich ein Sprite bewegt

automatisch eingesammelt.

```
DETECTOBJ  
IF OBJ <> 255  
GET OBJ  
ENDIF
```

Da **GET** und **PUT** nicht überprüfen, ob sich das Objekt bereits im Inventar des Spielers befindet, können sie verwendet werden, um Objekte auf dem Bildschirm zu platzieren oder sie mehr oder weniger nach Belieben zu entfernen. Möglicherweise möchten sie, dass feindliche Sprites Boni ablegen und diese nach einigen Sekunden verschwinden. Nur weil sie ein Objekt vom Bildschirm entfernen, müssen sie nicht die Punkte oder den Bonus vergeben, die der Spieler durch das Sammeln selbst erhalten hätte.

Map Layout (Taste M)

AGD erlaubt es sequentielle Levels zu erstellen. Es bietet jedoch auch die Möglichkeit Spiele zu erstellen, in denen der Spieler eine Karte erkunden kann.

Die "Karte" Ihres Spiels ist in einem Raster von 10 x 8 Positionen angeordnet, die alle leer beginnen. Leere Stellen erscheinen als zwei Bindestriche "--". Räume als Bildschirmnummern z. B. "01" oder "12". Da jeder Raum im Screen Designer entworfen wurde, kann er an einer bestimmten Stelle in dieses Raster eingefügt werden.

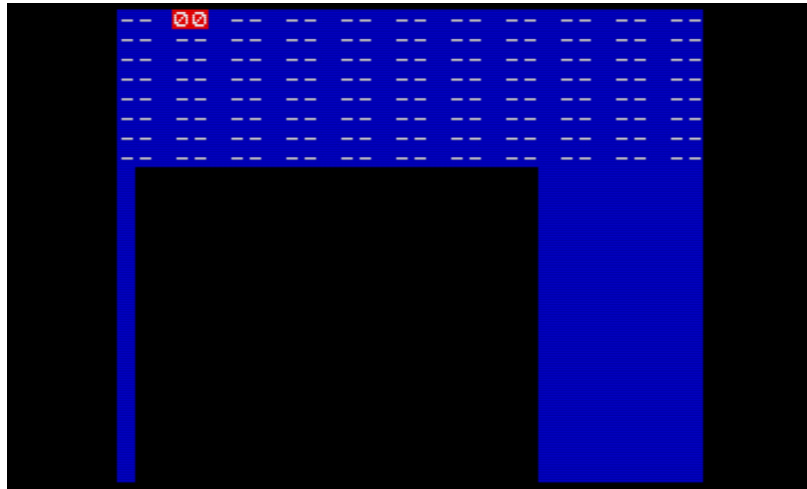
Um sich auf der Karte zu bewegen, verwenden Sie einfach die Befehle **SCREENUP**, **SCREENDOWN**, **SCREENLEFT**, **SCREENRIGHT**. Während des Spiels ist es dem Spieler nicht möglich, sich in ein leeres Gitterfeld zu bewegen.

Befehle wie **NEXTLEVEL** oder **LET SCREEN = 9** ändern die aktuelle Kartenposition nicht. Sie werden daher am besten in Spielen mit sequentiellen Ebenen verwendet.

Der rote Cursor kann mit den Cursortasten durch das Raster bewegt werden. Um den Raum an einer bestimmten Rasterposition zu ändern, bewegen Sie die Tasten '1' und '2'. Die Räume werden in den unteren zwei Dritteln des Bildschirms angezeigt. Ihre Karte folgt den Regeln der euklidischen Geometrie, aber sie können diese Regeln verändern, um ein verzerrtes Spielfeld zu erstellen. Wenn sie sich von einem Raum am linken Rand der Karte nach links bewegen, erscheint der Spieler wieder in dem Raum am rechten Rand der nächsten Reihe, sofern einer dort platziert ist. Wenn Sie sich von einem Raum am rechten Rand nach rechts bewegen, wird der Spieler in den Raum am äußersten linken Rand des Raums eine Reihe tiefer geführt. Wenn sie dies nicht möchten, sollten sie Wände auf den entsprechenden Bildschirmen errichten, um eine physische Barriere zu bilden.

Es ist auch möglich, einen Raum wiederzuverwenden, d.h. ihn mehrmals auf ihrer Karte erscheinen zu lassen. Wenn Sie von einem Bildschirm zum anderen wechseln, werden die Koordinaten des Player-Sprites nicht geändert. Sie sollten diese also gleichzeitig mit der Ausführung von **SCREENLEFT**, **SCREENRIGHT** usw. manuell einstellen. Alle für einen Bildschirm eingerichteten Spieler-Sprites werden nur für den allerersten Bildschirm verwendet oder um einen neuen Spieler-Sprite zu erzeugen, falls er auf diesem Bildschirm stirbt. Sollte der Spieler auf einem Bildschirm sterben, auf dem er keine Standardposition hat, wird er nicht wiederbelebt.

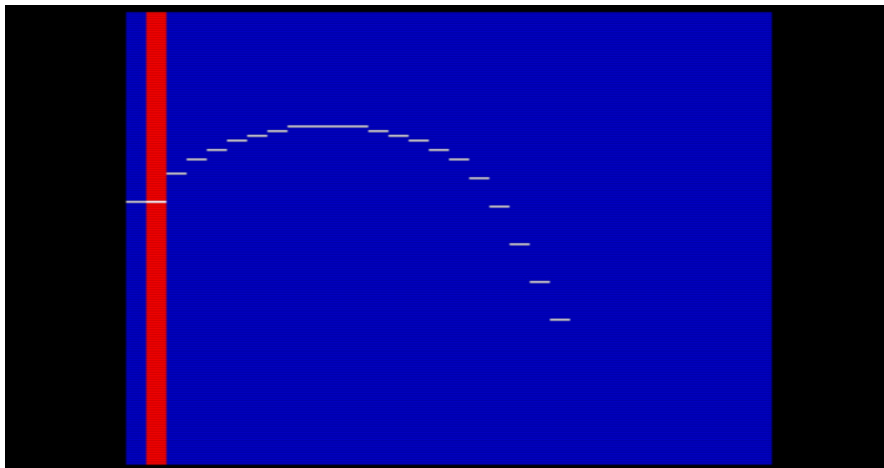
Drücken sie die Taste (X) um die Startposition des Spieler-Sprites festzulegen. Von dort beginnt der Spieler dann das Spiel.



Tastenfunktion:

- 1 = Wählt den vorherigen Raum aus der Liste.
- 2 = Wählt den nächsten Raum aus der Liste.
- X = Legt den Raum fest in dem der Spieler das Spiel beginnt.
- ENTER = Zurück zum Hauptmenü.

Jump Table (Taste J)

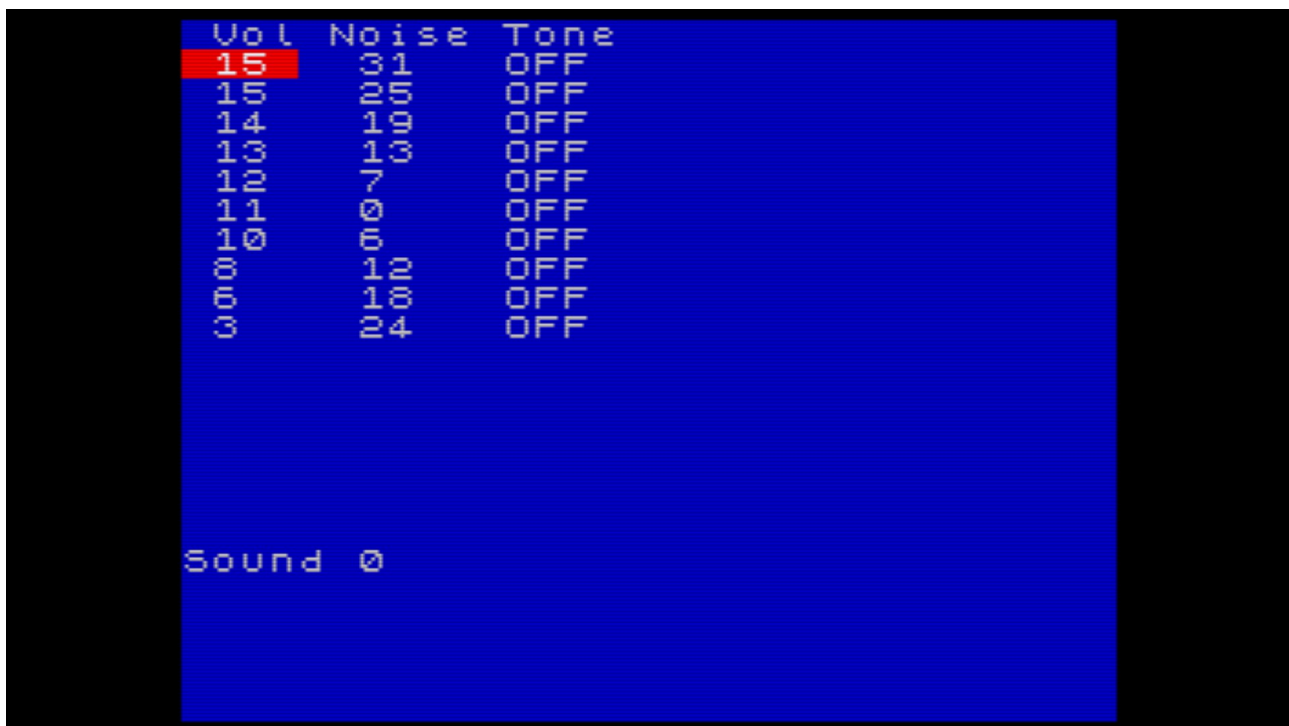


(Anpassung des Jump Tables)

Wenn Ihr Spielcharakter springen soll, können sie die Sprungtabelle bearbeiten. Auf diese Weise können sie die Winkel von Sprüngen und / oder Stürzen bearbeiten und bestimmen, wie hoch Sprites springen können oder wie schnell sie sich absenken, wenn sie durch Lücken im Boden fallen. Die Sprungtabelle ist in eine Reihe von Einzelschritten unterteilt, und sie können den Abstand zwischen diesen Schritten mit den Cursortasten ändern. Die rote Spalte steht für den

Schritt, den sie gerade bearbeiten. Wenn sie mit Ihrer Sprungtabelle zufrieden sind, drücken sie ENTER, um zum Hauptmenü zurückzukehren.

Sounds (Taste F)



Vol	Noise	Tone
15	31	OFF
15	25	OFF
14	19	OFF
13	13	OFF
12	7	OFF
11	0	OFF
10	6	OFF
8	12	OFF
6	18	OFF
3	24	OFF

Sound 0

Tastenfunktion:

H	= Aktuellen Sound anhören.
N	= Nächster Sound.
P	= Vorheriger Sound.
X	= Neuen Sound erstellen.
D	= Sound löschen.
ENTER	= Zurück zum Hauptmenü

Bewegen sie sich mit den Cursortasten durch die Werte und erhöhen und verringern sie diese mit den Tasten (1) und (2). Verwenden sie Tasten (3) und (4) um den Ton schrittweise zu erhöhen oder zu verringern.

Mit LEERTASTE oder (0) schalten Sie Rauschen (Noise) oder Ton (Tone) aus. Um einen Sound in Ihrem Spiel abzuspielen, verwenden Sie den Befehl **SOUND** im entsprechenden Ereignis. Sounds werden mit dem AY-Chip abgespielt.

Verwenden Sie den **BEEP**-Befehl für 48K-Pieptoneffekte.

Save Game (Taste G)

Fordert sie zur Eingabe eines Dateinamens für ihr Spiel auf und fragt sie, ob dem Codeblock ein BASIC-Ladeprogramm vorangestellt werden soll. Dieses lädt und führt das Programm automatisch aus. Auch ein Ladebildschirm kann mit abgespeichert werden, wenn dies in den Einstellungen Verschiedenes (Miscellaneous) festgelegt wurde.

Das Spiel wird dann mit oder ohne dem BASIC-Loader auf Band gespeichert.

Wenn sie das Spiel mit einem Loader speichern, kehrt es nach dem Spielende wieder zum Anfang zurück. Andernfalls kehrt es zu BASIC zurück, sodass sie ihre eigenen Routinen für das Spielende schreiben können. Die gespeicherte Codedatei kann später geladen und bearbeitet werden.

Vielleicht möchten sie sogar Ihren eigenen BASIC-Loader schreiben. Das einfachste Programm dafür wäre:

```
10 CLEAR 31231: LOAD "" CODE: RANDOMIZE USR 32000
```

Sie würden dieses BASIC-Programm mit so etwas wie:

```
SAVE "MyGame" LINE 10
```

auf Band speichern.

Load Game (Taste L)

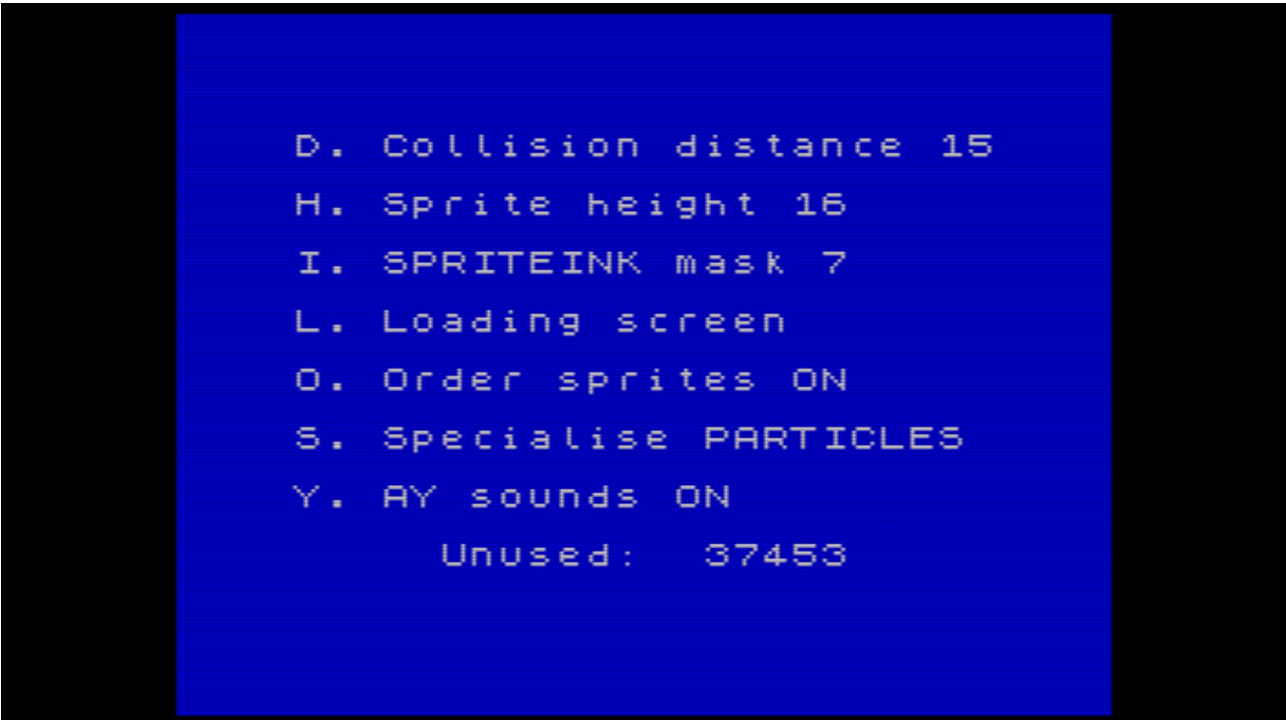
Lädt ein neues Spiel vom Band. Wenn Sie einen Emulator verwenden, müssen sie den Bandbrowser selbst bedienen. Emulatoren ohne Band-Browser und solche, die automatisch eingelegte Bänder laden, werden für AGD nicht empfohlen. AGD lädt nur Codedateien, die auch mit dem Arcade Game Designer erstellt wurden.

Test Game (Taste X)

Hier können sie ihr erstelltes Spiel testen. Mit ENTER gelangen sie wieder ins Hauptmenü.

Miscellaneous (Taste I)

Bietet verschiedene Optionen zum Anpassen der Spiele-Engine. Und zeigt die erste unbenutzte Speicheradresse an.



```
D. Collision distance 15
H. Sprite height 16
I. SPRITEINK mask 7
L. Loading screen
O. Order sprites ON
S. Specialise PARTICLES
Y. AY sounds ON
    Unused: 37453
```

Collision Distance

Die Kollisionsentfernung dient zum Einstellen der Sprite Kollisionserkennung. Standard-Sprites haben eine Größe von 16 x 16 Pixel. Normalerweise treffen sie aufeinander, wenn ihre Koordinaten weniger als 16 Pixel voneinander entfernt sind. Für ein klobiges Sprite wie Monty Mole oder Egghead ist dies ideal. Möglicherweise möchten Sie jedoch ein Spieler-Sprite zeichnen, das etwas schmaler ist und auf beiden Seiten Lücken aufweist. Sie möchten die Lücke nicht als Teil der Sprite-Kollisionserkennung zählen, daher können sie diesen Abstand auf eine niedrigere Pixelanzahl reduzieren. Beachten sie, dass diese Einstellung nur die horizontalen Abstände beeinflusst und die Kollisionserkennung zwischen Sprites definiert.

Sprite Height

Hier haben sie die Wahl die Größe der Sprites zu verändern. 16x16 oder 16x24 für Ihr Spiel. Beachten sie jedoch, dass 16x24-Sprites 50% mehr Arbeitsspeicher beanspruchen. Sie werden aufgefordert, zu bestätigen, dass sie die Sprite-Größe ändern möchten, und AGD konvertiert dann die Engine, sofern der Speicher dies zulässt. Alle vorhandenen Sprites werden in die neue Größe konvertiert. Sie sollten sie dann manuell im Sprite-Editor bearbeiten.

SPRITEINK Mask

Mit dem Befehl **SPRITEINK** können Sie normalerweise die Farbe eines Sprites ändern, ohne die Einstellung **BRIGHT** (Helligkeit) oder **FLASH** (Blinken) zu beeinflussen.

Wenn sie dies mit **SPRITEINK** tun möchten oder ihr Spiel unter Berücksichtigung von ULAplus entwerfen, müssen sie diese Einstellung ändern. Ein Wert von 7 ändert die FARBE, 71 ändert FARBE und HELBIGKEIT, 135 ändert FARBE und BLINKEN und 199 ändert FARBE, HELBIGKEIT und BLINKEN. Bei ULAplus können Sie mit einem Wert von 199 eine der vier Farbnachschlagetabellen auswählen. Um **BRIGHT** zu aktivieren, fügen sie dem **SPRITEINK**-Wert wie bei der BASIC ATTR-Funktion 64 hinzu. Fügen Sie 128 hinzu, um **FLASH** zu aktivieren.

Loading Screen

Hier ist es Möglich einen „Ladebildschirm“ welcher mit einem anderen Programm erstellt wurde zu importieren. AGD fordert sie auf zu bestätigen ob sie einen „Ladebildschirm“ laden wollen. Mit „y“ bestätigen sie dies und es wird von „Tape“ ein Bildschirm geladen und importiert.

Order Sprites

Wenn die Option Order Sprites (Sprites sortieren) aktiviert ist, wird die Sprite-Tabelle sortiert. Die Reihenfolge in der die Sprites angezeigt werden hängt von ihrer vertikalen Position ab, um Flimmern zu vermeiden. Wenn diese Option deaktiviert ist, variiert die Reihenfolge, in der die Sprites angezeigt werden, nicht von Frame zu Frame.

Specialise

Diese Option wählt aus, welche der drei Spezialanpassungen sie für Ihr Spiel verwenden möchten. Die verfügbaren Anpassungen sind **PARTICLES**, **EFFECTS** und **ADVENTURE**.

PARTICLES

Aktiviert die Befehle **TRAIL**, **EXPLODE** und **LASER**

Ideal für „Kondensstreifen“, „Magischen Staub“, Explosionen, Lasereffekte und um ihrem Shoot-Em-Up Spiel mehr „Glanz“ zu verleihen.

EFFECTS

Effects bietet die Befehle **FADE** und **TICKER**, um das Fenster auszublenden und einen Pixel-Bildlauf an einer bestimmten Stelle auf dem Bildschirm durchzuführen.

ADVENTURE

Aktiviert den **INV** Befehl, mit dem der Spieler Gegenstände in seinem Inventar auswählen und Wasserblöcke aktivieren kann. Es merkt sich auch alle Blöcke, die mit dem Befehl **PUTBLOCK** auf einem Bildschirm positioniert wurden, und zeichnet sie neu, wenn der Spieler den Bildschirm verlässt und wieder aufruft.

AY Sounds

Mit dieser Option können sie AY-Soundeffekte deaktivieren, falls sie stattdessen eine Interrupt-gesteuerte AY-Musikroutine einrichten möchten.

Text (Taste T)

Hier können sie festlegen, welche Textnachrichten ihr Spiel verwenden soll. Drücken sie ENTER, um zum Hauptmenü zurückzukehren.

N	= Nächste Textnachricht.
P	= Vorherige Textnachricht.
SPACE+0	= Bearbeitet aktuelle Textnachricht.
D	= Löscht Textnachricht.

Events (Taste E)

Hier kann der Spielcode erstellt und die Funktionsweise auf verschiedene Arten geändert werden. Der Editor und der Compiler können mit einer richtigen Sprache wie BASIC nicht konkurrieren. AGD bietet eine begrenzte Anzahl von Anweisungen, Funktionen und Variablen, die die Erstellung einer Vielzahl verschiedener Arcade-Spiele oder Arcade-Adventures ermöglichen sollen.

Es gibt verschiedene Ereignisse, für die sie den Code schreiben können. Abgesehen von den

Ereignissen, die zu bestimmten Zeiten im Spiel auftreten, gibt es welche, die 8 Sprite-Typen zugeordnet sind. Dies sind die Ereignisse, die die Bewegung und Logik jedes Spritetyps steuern. Sprite-Typ 0 ist normalerweise für das Sprite des Spielers reserviert, daher sollte mit diesem auch der Code beim Testen des Spiels überprüft werden. Also z.B ob das Sprite ein Leben verliert wenn es mit einem Gegnersprite kollidiert.

Der Rest liegt ganz bei ihnen. Sie können beispielsweise festlegen, dass die Sprite-Typen 1 und 2 unterschiedliche Aliens mit unterschiedlichen Bewegungsmustern sind, und Sprite-Typ 3 für Bonus-Sprites verwenden, die der Spieler einsammeln kann.



Player (type 0) control - Spieler Bewegung, Kollisionsabfrage

Sprite type 1 - Verhalten der Sprites mit dem Typ 1

Sprite type 2 - Verhalten der Sprites mit dem Typ 2

Sprite type 3 - Verhalten der Sprites mit dem Typ 3

Sprite type 4 - Verhalten der Sprites mit dem Typ 4

Sprite type 5 - Verhalten der Sprites mit dem Typ 5

Sprite type 6 - Verhalten der Sprites mit dem Typ 6

Sprite type 7 - Verhalten der Sprites mit dem Typ 7

Sprite type 8 - Verhalten der Sprites mit dem Typ 8

Initialise sprite – Ein Sprite initialisieren

Main loop 1 – Schleife 1 die in jedem Spiel durchläuft

Main loop 2 – Schleife 2 die in jedem Spiel durchläuft

Intro/menu – Bevor das Spiel beginnt

Game initialisation – Beim Spielstart, Variablen festlegen

Restart screen – Wenn der Spieler den Spielbildschirms wiederholt

Fell too far – Wenn das Sprite außerhalb des Jump Tables fällt

Kill player – Lebensverlust des Spielers durch ein KILL command

Lost game – Verlust des letzten Lebens / Spielende

Completed game – Das Spiel wurde erfolgreich beendet

Um die Einstellungen zu ändern, verwenden Sie den Cursor nach oben / unten, um das Ereignis auszuwählen, und drücken sie dann die Leertaste oder 0. Wenn für dieses Ereignis kein Code geschrieben wurde, wird ein weiteres Menü mit grundlegenden Codevorlagen angezeigt, mit dem sie beginnen können. Sie müssen keine Vorlage verwenden. Dies sind nur vorab geschriebene Codebeispiele, um verschiedene Funktionen auszuführen. Wenn sie eine Vorlage auswählen, kopiert der Editor einen Codeabschnitt in das Ereignis. Sie müssen jedoch nichts auswählen und können den Code auch manuell hinzufügen.

Die auswählbaren Optionen sind:

None

Manic Miner style controls

Player ladders and levels

Cybernoid style controls

Patrolling enemy (L/R)

Patrolling enemy (U/D)

Bouncing enemy

Ladders and levels pursuers

Static collectable

Pushable block

Moving platform

Bonus stopper minigame

Die erste Option (None) überspringt die Vorlagen und lässt sie den Code manuell eingeben.

Die nächsten Optionen sind Spieler-Steuerelemente.

Mit der Manic Miner Steuerung können sie sich nach links und rechts bewegen und springen.

Spieler-Leitern und -Levels verwenden die Bewegung nach links und rechts, sowie die Bewegung nach oben und unten wenn das Spielfeld Leitern beinhaltet.

Cybernoid-Steuerelemente sind links, rechts, feuer (nur Particle Engine) und hoch. Wenn sie nicht aufsteigen, bewegt sich das Spieler-Sprite wie in Cybernoid automatisch nach unten.

Es gibt auch feindliche Sprite-Vorlagen. Die Option Patrouillierender Feind (L / R) bewegt sich entlang einer Plattform nach links und rechts und kehrt um, wenn er gegen ein Hindernis stößt oder eine Lücke im Boden ist. Patrouillierender Feind (U / D) bewegt sich auf und ab und kehrt um, wenn er auf ein Hindernis trifft. Springende Gegner springen nur diagonal über den Bildschirm, bis sie auf etwas treffen, und ändern dann die Richtung. Leitern und Level-Verfolger gehen an Plattformen und Wänden entlang und steigen Leitern hoch und runter.

Statische Objekte sind Sprites, die auf dem Bildschirm verteilt sind (z.B. Münzen), die das Spieler-Sprite einsammeln kann. Bei Berührung mit dem Objekt verschwindet dieses dann und erhöht im besten Fall auch die Punktzahl.

Pushable Blocks reagieren auf Kollisionen mit dem Spieler-Sprite, indem sie aus dem Weg gehen. Sie prüfen auch auf Kollisionen mit anderen Sprites des Typs 1 und zerstören alle, auf die sie stoßen. Sie können das Verhalten aber im Code ändern.

Bewegliche Plattformen verhindern das Herunterfallen des Spieler-Sprites auf die selbe Weise wie es auch Standard Plattformen tun. Die Vorlage bewegt die Plattform von oben nach unten, aber mit einer kleinen Änderung können diese stattdessen auch von links nach rechts bewegt werden.

Sobald eine Vorlage ausgewählt ist oder sie ein Ereignis ausgewählt haben, das bereits einen Code enthält, können Sie den Code für dieses bestimmte Ereignis bearbeiten.

(Symbol/Shift + A) <=(EMULATOR TASTATUR) bringt sie wieder zurück zur „Event“ Auswahl.

Der Editor führt ihren Code aus und meldet sich, wenn er eine Codezeile nicht versteht.
In diesem Fall springt der Cursor in die fehlerhafte Zeile. Der Code im Editor wird direkt in Maschinensprache kompiliert.

Code Editor Tasten:

SYM-Y	Cut line
SYM-U	Paste line
SYM-Q	Home
SYM-E	End
CAPS-2	Toggle caps lock
CAPS-3	Delete forward
CAPS-4	Insert/overtyp
SYM-CAPS	Toggle extended mode

Befehle, Funktionen und Variablen

Funktionen können nur nach einem (IF) Befehl verwendet werden.

IF

Überprüft. Wenn die folgende Bedingung erfüllt ist, wird der Code bis zur nächsten **ENDIF**-Anweisung ausgeführt. **IF** kann mit einer Funktion oder zum Testen von Variablen oder Sprite-Parametern gegeneinander oder gegen bestimmte numerische Werte verwendet werden.

ENDIF

Markiert das Ende des bedingten Codes.

LET

Wie in BASIC können sie mit **LET** einer Variablen oder einem Sprite-Parameter einen Wert zuweisen. Der zugewiesene Wert kann eine Zahl oder eine andere Variable oder ein anderer Sprite-Parameter sein.

KEY

Funktion. Die Bedingung ist erfüllt, wenn die Taste gedrückt wird.

CANGOUP, CANGODOWN, CANGOLEFT, CANGORIGHT

Funktionen. Die Bedingung ist erfüllt, wenn sich das aktuelle Sprite nach oben / unten / links / rechts bewegen kann.

LADDERABOVE, LADDERBELOW

Funktionen. Die Bedingung ist erfüllt, wenn das aktuelle Sprite eine Leiter auf- / absteigen kann.

X, Y

Sprite Parameter. Dies sind die Koordinaten des aktuellen Sprites.

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T

Allgemeine Variablen. Diese enthalten 8-Bit-Werte, sodass Werte von 0 bis 255 möglich sind.

SCREEN, LIVES

Allgemeine Variablen. Diese enthalten die aktuelle Bildschirmnummer und die Anzahl der verbleibenden Leben.

TYPE

Sprite-Parameter. Dies ist der Typ des zu verarbeitenden Sprites. Die Einstellung dieses Parameters in Verbindung mit dem **IMAGE**-Parameter ändert das Verhalten des Sprites vollständig. Dies ist praktisch, um aus einem schädigenden Sprite einen Bonus zu machen oder um einen Sprite anzuhalten oder explodieren zu lassen bevor er getötet wird.

Sie können ein Sprite auch in TYP 0 ändern, ihn also unter die Kontrolle des Spielers bringen. Solange für den neuen Sprite-Typ im jeweiligen Ereignis der entsprechende Code eingerichtet ist, sind ihnen keine Grenzen gesetzt. Wenn Sie keinen Code für den Sprite-Typ eingerichtet haben, verbleibt ihr Sprite einfach an der Stelle an dem sie es platziert haben.

IMAGE, FRAME

Sprite-Parameter. Dies sind die im Sprite-Editor angezeigten Sprite- und Frame-Nummern. Sie können das Sprite entsprechend der Richtung ändern, in die der Spieler blickt, oder sie möchten dem Spieler eventuell eine Auswahl an Fahrzeugen zur Steuerung geben. Wenn Sie eine Frame-Nummer festlegen, die über das Limit des Sprites hinausgeht, wird ein anderes Sprite-Bild angezeigt. Verwenden Sie **FRAME** daher mit Vorsicht. Wenn Sie das Bild einstellen, empfiehlt es sich, den **FRAME** gleichzeitig auf 0 zu setzen. Es sei denn, sie haben einen sehr guten Grund dies nicht zu tun und es ist von ihnen so beabsichtigt.

DIRECTION, PARAMA, PARAMB

Sprite-Parameter. Sie können diese nach Belieben verwenden, um möglicherweise die Richtung anzugeben, in die sich ein bestimmtes Sprite bewegt, oder die bestimmte Phase, die es durchläuft. Von unschätzbarem Wert für jede Form feindlicher KI.

ANIMATE

Befehl. Animiert das aktuelle Sprite und durchläuft die Frames automatisch in aufsteigender Reihenfolge.

ANIMBACK

Befehl. Wie **ANIMATE**, wechselt jedoch in absteigender Reihenfolge durch die Frames.

NEXTLEVEL, RESTART

Befehl. Gehe zum nächsten Level und starte das aktuelle Level neu.

SPRITEUP, SPRITEDOWN, SPRITELEFT, SPRITERIGHT

Befehl. Bewegt das aktuelle Sprite entsprechend. Es wird keine Überprüfung auf Blockaden oder Bedingungen außerhalb des Bildschirms durchgeführt. Dieses müssen sie mit Funktionen wie **CANGOLEFT** oder **LADDERABOVE** selbst tun.

SPAWN

Befehl. Erwartet zwei Parameter für Sprite-Typ und Bild. Dadurch wird ein neues Sprite mit dem angegebenen Typ und Bild an der aktuellen Position des Sprites erstellt. Das neue Sprite wird mit **FRAME, DIRECTION, PARAMA** und **PARAMB** auf Null erstellt. Das aktuelle Sprite bleibt davon unberührt.

SPAWNED

Befehl. Sollte nur nach einer **SPAWN**-Anweisung verwendet werden. Dieser Befehl wählt das neu erzeugte Sprite aus. Jeder Code, der nach **SPAWNED** geschrieben wurde, bezieht sich auf das neue Sprite. Verwenden sie **ORIGINAL**, um zum ursprünglichen primären Sprite zurückzukehren. Alternativ können sie ihren Code im Sprite-Initialisierungsereignis platzieren. Dies entscheiden sie selbst.

REMOVE

Befehl. Entfernt das aktuelle Sprite aus der Tabelle. Nützlich zum Zerstören von Feinden oder zum Aufheben von Gegenständen.

PUTBLOCK

Befehl. Erfordert einen Parameter zur Angabe der Blocknummer. Dadurch wird der angegebene Block an der aktuellen Position von **LINE** und **COLUMN** auf dem Bildschirm platziert und kann zum Öffnen und Schließen von Türen hilfreich sein. Beachten Sie, dass beim Verlassen und erneuten Aufrufen des Bildschirms oder beim erneuten Zeichnen nach **MENU** oder **INV** die Standardposition des Bildschirms wiederhergestellt wird, es sei denn, sie haben die Adventure-Spezialisierung ausgewählt.

COLLISION

Funktion. Erfordert ein numerisches Argument, um den Sprite-Typ anzugeben. Bedingung ist erfüllt, wenn das aktuelle Sprite mit einem anderen Sprite des angegebenen Typs kollidiert.

OTHER

Befehl. Sollte nur nach einer erfolgreichen **COLLISION**-Prüfung verwendet werden. Dieser Befehl wählt das andere Sprite aus, d.h. das sekundäre, mit dem das ursprüngliche Sprite gerade kollidiert ist. Jeder nach **OTHER** geschriebene Code verweist auf das sekundäre Sprite. Verwenden sie **ORIGINAL**, um zum ursprünglichen primären Sprite zurückzukehren, wenn sie den Code für das sekundäre Sprite erstellt haben.

ORIGINAL

Befehl. Wird nach den Befehlen **OTHER** und **SPAWNED** verwendet und stellt das ursprüngliche Sprite wieder her.

ENDGAME

Befehl. Beendet erfolgreich das Spiel. Dies führt das Ereignis "Beendetes Spiel" aus. Das Beenden des letzten Bildschirms in einem Spiel mit aufeinanderfolgenden Levels bewirkt dasselbe. Die einzige andere Möglichkeit, ein Spiel zu beenden, besteht darin, dass der Spieler sein ganzes Leben verliert. Das Ereignis "Beendetes Spiel" wird dann jedoch nicht ausführt.

SHOWSCORE

Befehl. Zeigt die Punktezah an der aktuellen Cursorposition an. Sollte unmittelbar mit Anweisungen zum Einrichten der Zeilen- und Spaltenposition für den Cursor beginnen.

SCORE

Befehl. Erwartet die Eingabe einer Zahl um sie der Punktzahl hinzuzufügen. **SCORE** 100 erhöht die Gesamtsumme des Spielers um 100 Punkte. Die Werte 0 bis 255 sind gültig.

SOUND

Befehl. Startet einen Soundeffekt mit dem AY-Soundchip. Erwartet einen einzelnen Parameter für die wiederzugebende Sound Nummer.

BEEP

Befehl. Startet einen Signaltoneffekt. Erwartet einen einzelnen Parameter für die Klangdauer. Werte über 127 erzeugen weiße Rauscheffekte.

CLS

Befehl. Löscht den Bildschirminhalt.

BORDER

Befehl. Erwartet eine Eingabe von 0 bis einschließlich 7. Dies legt die Rahmenfarbe fest.

COLOUR

Befehl. Legt die permanenten Anzeigeattribute fest. Das Format:

128 * FLASH + 64 * BRIGHT + 8 * PAPER + INK.

Kann vor dem Anzeigen von Text oder Löschen des Bildschirms verwendet werden.

DELAY

Befehl. Pausiert für die angegebene Dauer.

MESSAGE

Befehl. Erwartet die Eingabe der Nummer der **MESSAGE** die Angezeigt werden soll.

KILL

Befehl. Initiiert das Kill-Player-Ereignis und verringert den Lebenszähler. Sie sollten den Lebenszähler im Initialisierungsereignis mit so etwas wie **LET LIVES = 3** einrichten.

LINE, COLUMN

Variable. Diese bestimmen die Position, an der die Punktezahl oder Nachricht angezeigt wird. Sie sind temporär und ändern sich jedes mal, wenn ein Sprite angezeigt wird. Verwenden sie diese daher immer unmittelbar vor einem **MESSAGE**- oder **SHOWSCORE**-Befehl.

GETRANDOM

Funktion. Erzeugt eine Zufallszahl zwischen Null und der Eingabe und platziert sie in der **RND**-Variablen. **GETRANDOM** 100 generiert eine Zahl von 0 bis 99, **GETRANDOM** 2 generiert eine Null oder 1.

RND

Variable. Die letzte von **GETRANDOM** generierte Zufallszahl.

ADD, SUBTRACT

Befehl. Addiert oder subtrahiert zu oder von einem Sprite-Parameter oder einer Sprite-Variablen, z.B. **ADD 1 TO A** oder **SUBTRACT 5 FROM B**.

DISPLAY

Befehl. Zeigt den numerischen Wert in der folgenden Variablen oder dem folgenden Parameter an. Zum Beispiel **DISPLAY LIVES**.

SCREENUP, SCREENDOWN, SCREENLEFT, SCRENNRIGHT

Befehl. Bewegung des Bildschirm nach oben, unten, links oder rechts wenn dies möglich ist.

DEADLY

Funktion. Bedingung ist erfüllt, wenn das aktuelle Sprite einen **DEADLY** Block berührt.

CUSTOM

Funktion. Ähnlich wie **DEADLY**. Bedingung ist erfüllt wenn das aktuelle Sprite mit einen **CUSTOM**-Block berührt.

WAITKEY

Befehl. Erwartet eine Tastatureingabe.

JUMP

Befehl. Das aktuelle Sprite springt, sofern es sich nicht bereits in der Luft befindet und keine Wände im Weg sind. Es kann jedes Sprite zum Springen gebracht werden, nicht nur das direkt vom Spieler kontrolliert wird.

FALL

Befehl. Sofern das Sprite nicht bereits fällt oder springt, wird überprüft, ob das Sprite auf festem Boden steht. Wenn nicht, beginnt es mit der absteigenden Hälfte der Sprungtabelle zu fallen (Ihr Spiel findet diesen Punkt automatisch, auch wenn sie die Sprungtabelle geändert haben). **FALL** ist nützlich, wenn ein Sprite-Typ der Schwerkraft ausgesetzt ist, z. B. bei Plattform Spielen.

GOT

Funktion. Wird in **IF**-Anweisungen verwendet und erwartet einen einzelnen Beweis. Dies ist erfüllt, wenn das angegebene Objekt dem Spieler gehört. Der Beweis kann bei Bedarf eine Variable sein, daher ist **IF GOT 1** ebenso gültig wie **IF GOT A**. Sie können sogar einen Sprite-Parameter als Beweis verwenden, wenn sie dies wünschen.

GET

Befehl. Erwartet einen Parameter, der das abzurufende Objekt angibt. Platziert das angegebene Objekt im Inventar des Spielers, unabhängig davon, wo es sich befindet. Das Objekt befindet sich möglicherweise auf dem aktuellen Bildschirm, einem anderen Bildschirm oder fehlt. Sie müssen entscheiden, wann der Spieler ein bestimmtes Objekt bekommen kann. Sie können **DETECTOBJ** verwenden, um zu bestimmen, wann ein Sprite zuerst ein Objekt berührt.

PUT

Befehl. **PUT**-Befehl erfordert ein Argument, das die Objektnummer angibt. Das Objekt wird an der aktuellen Position des Sprites auf dem Bildschirm abgelegt, es sei denn, es befindet sich bereits auf dem aktuellen Bildschirm. Das Argument kann numerisch sein, eine Variable oder ein Sprite-Parameter.

DETECTOBJ

Befehl. Erkennt Objekte, die vom aktuellen Sprite berührt werden, und platziert das Ergebnis in der **OBJ**-Variablen. Wenn das Sprite mehr als ein Objekt berührt, wird nur das Objekt mit der niedrigsten Nummer erkannt. Wird kein Objekt erkannt, wird **OBJ** auf 255 gesetzt.

OBJ

Variable. Die letzte von **DETECTOBJ** erkannte Objektnummer.

CLOCK

Nur-Lese-Variable. Der aktuelle Bildzähler in 50stel Sekunden. Nützlich für Timings.

EXIT

Befehl. Die gesamte Verarbeitung des aktuellen Ereignisses wird abgebrochen.

REPEAT

Befehl. Benötigt einen Parameter, um die Anzahl der Wiederholungen zu bestimmen. Der Code bis zum nächsten **ENDREPEAT** wird so oft wiederholt. **REPEAT** kann nicht verschachtelt werden.

ENDREPEAT

Befehl. Dies markiert das Ende des zu wiederholenden Codeblocks. In einer **REPEAT**-, **ENDREPEAT**-Schleife platzierter Code wird vom Editor automatisch eingerückt, um die Lesbarkeit zu verbessern.

MENU

Befehl. Erfordert ein Argument, das die Nachrichtennummer angibt. Erzeugt ein Popup-Menü in der Mitte des Wiedergabebereichs, in dem die Optionen der angegebenen Nachrichtennummer entnommen werden, jeweils in einer eigenen Zeile. Der Spieler wählt dann eine Option unter Verwendung der Aufwärts-, Abwärts- und Feuerkontrolle. Normalerweise geschieht dies, wenn der Spieler eine bestimmte Taste drückt. Es könnte jedoch in einem anderen Sprite-Ereignis auftauchen, z. B. wenn ein bestimmtes Sprite mit dem Spieler kollidiert. Sie können eine Nachricht anzeigen - das Gespräch des Sprites mit dem Spieler - und dann mit **MENU** eine Antwort anfordern. Der Bildschirm wird nach **MENU** neu gezeichnet.

INV

Befehl. Erfordert ein Argument, das die Nachrichtennummer angibt. Ähnlich wie **MENU**, jedoch wird ein Menü erstellt, das auf Objekten basiert, die sich im Besitz des Spielers befinden. Richten Sie eine Liste der in Ihrem Spiel verwendeten Objektnamen in einer Nachricht mit jedem Element in einer separaten Zeile ein und verwenden sie **INV** „n“, wobei „n“ die Nachrichtennummer ist, die die Objektnamen enthält. **INV** zeigt dem Spieler ein Menü an, das nur die Elemente enthält, die sich derzeit im Inventar des Spielers befinden. Nützlich für die Auswahl eines zu verarbeitenden oder zu löschenden Elements. **INV** funktioniert nur mit der Adventure-Spezialisierung.

MULTIPLY

Befehl. Multiplizieren sie einen Sprite-Parameter oder eine Variable, z.B. **MULTIPLY C BY 7**.

DIVIDE

Befehl. Teilen Sie einen Sprite-Parameter oder eine Variable, z.B. **DIVIDE Y BY 8**.

SPRITEINK

Befehl. Stellt die Farbe für das aktuelle Sprite auf die angegebene Farbe ein, z. B. **SPRITEINK 4** oder **SPRITEINK PARAMB**. Wirkt sich nicht auf den Hintergrund, Blinken oder Helligkeit aus. Setzen sie **SPRITEINK** zu Beginn jedes Sprite-Ereignisses zurück, um zu vermeiden, dass eine Spur unterschiedlicher Farben zurückbleibt. Wenn ihre Hintergrundfarbe normalerweise 7 (weiß) ist, können Sie **SPRITEINK 7** zu Beginn des Sprite-Ereignisses verwenden, das Sprite wie gewohnt verschieben und dann auf die gewünschte Farbe einstellen. Statische sammelbare Sprites sollten den **SPRITEINK** nur dann auf den normalen Hintergrund zurücksetzen, wenn sie gesammelt werden. Ein Beispiel finden Sie im Demo-Spiel Diamond Geezer.

LASER

Befehl. Erwartet ein numerisches Argument. Feuert je nach Parameter einen Laser links oder rechts von der aktuellen Sprite-Position ab. Gerade Parameter feuern nach links, ungerade nach rechts. Sprite-Kollisionen mit Lasern können in den Sprite-Typ-Ereignissen mit **IF COLLISION 10** erkannt werden. **LASER** funktioniert nur, wenn die Partikelspezialisierung im Menü Verschiedenes ausgewählt ist.

TRAIL

Befehl. **TRAIL** ist nützlich für „Feenstaub- oder Kondensstreifen“ und erstellt ein Kondensstreifenpartikel in der Mitte des aktuellen Sprites. Wenn Sie zuerst **ADD 8 TO Y** oder **SUBTRACT 8 FROM Y** verwenden, können sie eine Spur links oder rechts vom Sprite erstellen. Denken sie daran, nach dem Befehl **TRAIL** erneut **ADD** oder **SUBTRACT 8** auszuführen, um die ursprünglichen Koordinaten des Sprites wiederherzustellen. Funktioniert nur mit der Partikelspezialisierung.

EXPLODE

Befehl. Erfordert ein numerisches Argument, um die Anzahl der Partikel anzugeben. Erzeugt eine Explosion an der aktuellen Sprite-Position. In Verbindung mit **REMOVE** entfällt mit **EXPLODE** die Notwendigkeit, eine Explosionsanimation für Ihre Sprites zu zeichnen. Wenn Sie vergessen, **EXPLODE** mit einem Parameter zu versehen, wird der nächste Befehl nicht richtig kompiliert. Wenn dieser Befehl **REMOVE** lautet, wird Ihr Sprite nicht entfernt und Sie werden sich fragen, warum es explodiert, aber im Spiel bleibt. Funktioniert nur mit der Partikelspezialisierung.

FADE

Befehl. Blendet das Fenster schwarz aus. Fade funktioniert nur mit aktivierter Effektspezialisierung.

TICKER

Befehl. **TICKER** scrollt eine Textnachricht an der aktuellen Position von **LINE** und **COLUMN**. Dieser erwartet 2 Argumente, das erste ist die Nachrichtennummer und das zweite die Breite der Nachricht in Zeichen. **TICKER** fährt fort und wiederholt die Nachricht, wenn sie das Ende erreicht. **TICKER 1 16** blättert Nachricht 1 in einem Streifen um die Hälfte der Bildschirmbreite. Geben Sie zum Ausschalten von **TICKER** eine nicht vorhandene Nachrichtennummer an, z. B. **TICKER 99 1**. Funktioniert nur, wenn die Effektspezialisierung aktiviert ist.

BIGMESSAGE

Befehl. Erfordert einen Parameter, der eine Textnachricht angibt. Zeigt die Nachricht in doppelter Höhe auf dem Bildschirm an. Nur zur Initialisierung der Effekte.

REDRAW

Befehl. Zeichnet den Bildschirm, Sprites, Objekte und Partikel neu. Wenn Sie ein Arcade-Adventure schreiben, möchten sie möglicherweise Nachrichten im Wiedergabefenster anzeigen. Mit **WAITKEY** eine Pause einlegen und dann **REDRAW** ausführen, bevor sie zur Aktion zurückkehren.

SILENCE

Befehl. Schaltet den AY-Soundchip aus.

CONTROLMENU

Befehl. **CONTROLMENU** ist nützlich für das Menü- / Intro-Ereignis und wartet darauf, dass der Player eine Steuerungsoption auswählt. Die Optionen sind 1 für die Tastatur, 2 für den Kempston-Joystick und 3 für den Sinclair-Joystick (Tasten 67890).

DIG

Befehl. Erwartet einen Parameter 0 bis 3, der die Richtung nach oben, unten links oder rechts angibt. Entfernt **FODDER**-Blöcke direkt neben dem Spieler-Sprite. Nur mit der Adventure-Spezialisierung verfügbar.

STAR

Befehl. **STAR** ist Teil der Partikel-Engine und erstellt bei wiederholtem Aufruf ein vertikales oder horizontales Sternfeld. Erwartet, dass ein Parameter die Richtung angibt, 0 für vertikal und 1 für horizontal.

BIGSCORE

Befehl. Wie **SHOWSCORE**, zeigt jedoch die Meldung in doppelter Höhe auf dem Bildschirm an. Nur zur Initialisierung der Effekte.

STOPFALL

Befehl. Stoppt ein springendes oder fallendes Sprite in der Luft und ruft gegebenenfalls das Ereignis "Zu tief gefallen" auf.

BEISPIEL 1

Angenommen, wir haben ein Spielfenster mit 22 Zeichen und einer einzelnen Zeichenlücke oben und unten auf dem Bildschirm. Und möchten nun ein Kästchen um das Fenster zeichnen. Wir können dies tun, indem wir Meldungen im Spielinitialisierungsereignis anzeigen. Wir könnten Nachricht 0 als oberste Zeile einrichten, die über dem Fenster angezeigt wird, und Nachricht 1 als unterste Zeile. Um die Linien entlang der Seiten zu zeichnen, können sie diese in Nachricht 2 einrichten und dann mit einer REPEAT-Schleife zeichnen. So könnte es in etwa aussehen:

```
LET LINE = 0
MESSAGE 0
REPEAT 22
  MESSAGE 2
ENDREPEAT
MESSAGE 1
```

BEISPIEL 2

Wenn wir Sprite-Typ 3 auswählen und als Vorlage "Statisch sammelbar" wählen, werden die folgenden Codezeilen angezeigt:

```
IF COLLISION 0
  REMOVE
  SCORE 100
ENDIF
```

Wir können einen kurzen Beeper-Soundeffekt hinzufügen, indem Sie den Befehl BEEP 30 eingeben:

```
IF COLLISION 0
  BEEP 30
  REMOVE
  SCORE 100
ENDIF
```

Wenn wir weiter gehen, können wir am Anfang die folgenden Zeilen einfügen, um das Verhalten des Sprites zu ändern:

```
IF CANGOUP
  SPRITEUP
ELSE
  REMOVE
ENDIF
```

So sieht der bisherige Code bis jetzt aus:

```
IF CANGOUP
  SPRITEUP
ELSE
  REMOVE
ENDIF
IF COLLISION 0
  BEEP 30
  REMOVE
  SCORE 100
ENDIF
```

Wir haben jetzt ein Sprite welches aufgenommen kann, das sich nach oben bewegt, bis es auf etwas trifft und dann wie eine Blase verschwindet. Jetzt brauchen wir etwas Code, um unsere neuen Sprites am unteren Bildschirmrand zu erzeugen. Wir können dies in der Hauptschleife tun, also wählen Sie das Ereignis Hauptschleife 1 und fügen Sie diese Zeilen ein:

```
GETRANDOM 127
IF RND <= 6
  SPAWN 3 0
  SPAWNED
  IF TYPE = 3
    LET X = 168
    GETRANDOM 160
    LET Y = RND
    ADD 8 TO Y
  ENDIF
  ORIGINAL
ENDIF
```

Dieser Code generiert zuerst eine Zufallszahl von 0 bis 127. Wenn diese Zahl kleiner oder gleich 6 ist, wird ein Sprite vom Typ 3 (das zuvor bearbeitete Ereignis) und ein Sprite-Bild von 0 generiert. Wenn das Sprite erzeugt wurde, dann setzt es die x-Position auf den unteren Rand des Wiedergabebereichs (vorausgesetzt, der Wiedergabebereich endet 8 Pixel vom unteren Rand des Bildschirms entfernt) und wählt eine zufällige y-Position zwischen 8 und 168.

BEISPIEL 3

Angenommen, wir schreiben ein Top-Down-Labyrinthspiel im Atic-Atac-Stil und möchten patrouillierende Feinde einbeziehen. Wir können Sprite-Typ 4 auswählen und die Vorlage Patrouillierender Feind (L / R) auswählen. In dieser Vorlage wird jedoch davon ausgegangen, dass wir ein Plattformspiel schreiben. Daher müssen wir die Zeilen entfernen, die auf Lücken prüfen, da sich das Sprite nicht auf einer Plattform befindet. In der Vorlage wird diese Prüfung durchgeführt, indem 16 (die Sprite-Breite) zur horizontalen (Y) Koordinate des Sprites addiert oder subtrahiert wird und geprüft wird, ob das Sprite nach unten gehen kann. Wenn dies möglich ist, ändert das Sprite die Richtung, da eine Lücke vorhanden sein muss. In unserem Fall wollen wir einfach nach links oder rechts gehen, bis wir auf ein Hindernis stoßen, und uns dann umdrehen. Die unnötigen Zeilen müssen dann gelöscht werden. Unsere Vorlage sollte folgendermaßen aussehen:

```
IF PARAMA = 0
  IF CANGOLEFT
    SPRITELEFT
  ELSE
    LET PARAMA = 1
  ENDIF
ELSE
  IF CANGORIGHT
    SPRITERIGHT
  ELSE
    LET PARAMA = 0
  ENDIF
ENDIF
```

Character Set (Taste A)



AGD wird mit einem eigenen Schrifteeditor geliefert. Zeichen werden mit den Cursortasten mit 0 oder LEERTASTE bearbeitet, um ein Pixel zu setzen / zu entfernen. Der gesamte druckbare Bereich von 96 Zeichen wird am unteren Bildschirmrand angezeigt, und Sie können diese mit den Tasten N und P durchlaufen. Wenn Sie möchten, können Sie eine mit einem anderen Dienstprogramm erstellte Spectrum-Standardschrift in AGD laden. Stellen Sie zunächst sicher, dass bei Verwendung eines Emulators keine "Tapes" automatisch geladen werden, da sie sonst das Spiel verlieren, an dem Sie arbeiten. Drücken Sie L, um eine Schriftart zu laden.

P	= Vorheriger Buchstabe.
N	= Nächster Buchstabe.
L	= Lädt Schriftart von Kassette.
ENTER	= Zurück zum Hauptmenü.

Technische Informationen

Mit dem AGD erstellte Spiele sind eigenständig und sollten unabhängig vom Editor selbst funktionieren. Das Programm ändert den Interrupt-Modus nicht, sodass sie ihre eigenen Interrupts einrichten können, um beispielsweise Musik abzuspielen. Sie sollten die Interrupts jedoch nicht deaktivieren. Wenn Sie Ihre eigenen Interrupts mit IM2 einrichten möchten, stellen sie sicher, dass Ihre Serviceroutine das Byte an Speicherplatz 23672 bei jedem Frame erhöht, da Ihr Spiel sonst hängen bleibt. AGD ändert den Wert des IY-Registers nicht. IX wird hauptsächlich als Sprite-Zeiger verwendet wird.

Spiele belegen Speicheradressen ab ca. 31200, dies ist der Bereich, der vom Dienstprogramm gespeichert wird. Die genaue Startadresse variiert, wenn nicht-kritische Ereignisse von 31232 an abnehmen, um Platz im oberen RAM für das Spiel zu schaffen. Darunter befindet sich der Code für den Editor, der bei Bedarf von anderen RAM-Bänken eingelesen wird. Ganz oben im RAM werden die letzten 768 Bytes von 64768 bis 65535 als Dummy-Kollisionskartenbereich verwendet, um zwischen verschiedenen Arten von Blöcken zu unterscheiden - Wände, Leitern, Leerraum usw. Am Ende des Spiels steht eine Fläche von 300 Bytes für die Partikelmaschine zur Verfügung, falls Sie sich für den Einsatz von Lasern, Kondensstreifen oder Explosionen entscheiden. Dieser Puffer ist in den Adventure- oder Effekt-Engines nicht vorhanden. Für die Dekomprimierung der Bildschirmdaten ist kein Puffer erforderlich, da diese von der Routine, die den Bildschirm zeichnet, im laufenden Betrieb erweitert werden.

Wenn sie ihr fertiges Spiel speichern und sich dafür entscheiden, einen BASIC-Loader hinzuzufügen, wird eine automatisch ladende Version Ihres Spiels erstellt, die einen Ladebildschirm enthält falls sie einen im Menü "Miscellaneous" eingerichtet haben. Automatisch ausgeführte Spiele kehren nicht zu BASIC zurück und werden fortlaufend wiederholt. Sie müssen daher Ihren eigenen Intro-Bildschirm in AGD programmieren. Wenn sie keinen BASIC-Loader erstellen, kehrt Ihr Spiel am Ende zu BASIC zurück.

Spiele können über die Tastatur oder den Kempston-Joystick gesteuert werden. Dies erfolgt über das Byte unter der Adresse 32005. Der Wert 1 bedeutet, dass die Kempston-Steuerelemente für die ersten fünf Tasten verwendet werden. 2 verwendet die Sinclair-Joystick-Tasten 6, 7, 8, 9 und Null. Bei allen anderen Werten werden standardmäßig Tastatursteuerelemente verwendet.

AGD setzt die Systemvariable für die Tastaturwiederholrate auf die Adresse 23562. Das Sinclair-ROM setzt diese beim Start auf 5 und AGD senkt sie auf 3, um die Bearbeitung zu beschleunigen. Wenn Sie jedoch einen POKE 23562 mit einem anderen Wert als 5 AGD verwenden, wird dieser Wert nicht geändert. Sie können also mit POKE 23562,1 die Wiederholrate der Tastatur auf die maximale Geschwindigkeit einstellen, falls Sie dies wünschen.

Hinzufügen von Musik oder Unterprogrammen von Drittanbietern (nur Partikel / Effekte)

Wenn sie dem Spiel Ihre eigenen Routinen hinzufügen möchten, sollten sie diese drei Zeilen vorübergehend zu dem Ereignis hinzufügen, bei dem Sie die externe Routine aufrufen möchten:

```
ASM 205  
ASM 85  
ASM 5
```

Machen sie sich keine allzu großen Sorgen darüber, wie dies funktioniert. Es handelt sich lediglich um einen Dummy-Aufruf an einem Punkt im Sinclair-ROM, der nichts bewirkt. Wir werden das später ändern.

Wählen sie als nächstes "Miscellaneous" aus dem Menü und notieren die unten auf dem Bildschirm angezeigte Adresse. Dies ist Ihre Kompilierungsadresse, der Speicherort, an dem sie ihre externe Routine kompilieren sollten. Dann müssen sie mit einem Taschenrechner zwei Werte berechnen, um sie in die Ereignisse einzufügen, in denen sie zuvor die ASM-Werte hinzugefügt haben. Sie müssen

Ihre Kompilierungsadresse geteilt durch 256 und auch den Rest berechnen.

Kehren sie zu "Events" zurück, in denen Sie die ASM-Werte eingegeben haben, und ersetzen Sie 5 durch die Kompilierungsadresse geteilt durch 256 und 85 durch den Rest. **SPEICHERN SIE IHR SPIEL SOFORT OHNE GRUNDLADER-OPTION**. Testen Sie es zu diesem Zeitpunkt nicht, da die AGD abstürzt, weil ihre Routine nicht im Speicher ist. Sie müssen sicherstellen, dass ihr BASIC-Loader zusätzlich zu Ihrem Spiel ihre eigene Routine lädt.

Kempston Mouse

Nur wenige Spectrum-Spiele verwenden die Maussteuerung und AGD wurde nicht speziell für die Verwendung mit einem solchen Gerät entwickelt. AGD hat jedoch eine versteckte Routine zum erkennen der Kempston-Maus. Wählen Sie dazu die Abenteuerspezialisierung aus dem Menü "Miscellaneous" aus und fügen sie diese drei Befehle in eines ihrer Hauptschleifenereignisse ein:

ASM 205

ASM 192

ASM 129

Dadurch werden die Mauskoordinaten und der Status der Feuertasten in die Variablen A, B und C eingefügt.

Weitere Informationen

Paul Jenkinson hat eine Reihe exzellenter AGD-Video-Tutorials erstellt, die auf seiner Website zu finden sind:

<http://www.randomkak.blogspot.co.uk/p/agd-video-tutorials.html>

Im offiziellen AGD-Forum finden Sie Neuigkeiten zu den neuesten Versionen, Tipps zur optimalen Nutzung des Programmes sowie Details zu Spielen, die bereits mit AGD geschrieben wurden oder sich derzeit in der Entwicklung befinden. Das Forum finden Sie hier:

<http://arcadegamedesigner.proboards.com/>

Andere Quellen:

<http://www.funspot.it/category/agd/>

<http://www.vintageisthenewold.com/agd-arcade-game-designer-initial-setp/>



(DE Version 1.0)