# V I C 2 0 M P A G D v 1 . 0

## 1.0 History:

### 1.1 AGD:

AGD or Arcade Game Designer is a program for the ZX Spectrum created by Jonathan Cauldwell to make creating games easy and understandable for everyone. AGD is event based which means that code is executed when a certain event occurs. The AGD program has 3 parts, editors, a compiler and an engine.

**Editors:**

Editors can be used for designing graphics and code like blocks, sprites, objects, screens, map and eventcode. The gamecode consists of AGD commands, a kind of pseudo game language which defines all graphics and describes the game algoritm.

**Compiler:**

The AGD commands can be converted to asm code by using a compiler which is also included in AGD. Most of the commands set some parameters and do a call to an engine routine.

**Engine:**

The engine consists general routines for playing a game.

### 1.2 MPAGD:

Because the original AGD only ran on a ZX Spectrum, Jonathan started to create a more versatile version to also create games for other platforms instead of the ZX Spectrum. This is how MPAGD or Multiple Platform Arcade Game Designer was born. The goal was to create general game code which can be compiled for different platforms.

This means that for every platform, you need a dedicated compiler and engine. Because graphics can differ a lot for other platforms, you need special editors to create graphics in MPAGD. Jonathan wrote WinAGD which contains graphic editors for several platforms and a text editor for code. Then, depending on the chosen platform, a MPAGD Suite is started to compile, assemble and emulate a created game.

### 1.3 WinAGD:

WinAGD is a Windows program with editors. You can define the game graphics and add the event code in WinAGD. When you build the game, an AGD text file DELETEME.AGD is created and passed to the Suite. The Suite compiles, assembles and emulates the game.

## 2.0 VIC20 port:

### 2.1 System requirements

To run MPAGD games you need a PAL/NTSC VIC20 with these specifications:

- RAM : Standard

Expansion Block 0: $0400-$0fff

Block 1: $2000-$3fff

Block 2: $4000-$5fff

Block 3: $6000-$7fff

Block 5: $a000-$bfff

- Disc : Turboload is selectable in MPAGD

MPAGD is designed to run a game on a PAL or NTSC machine without difference.

The VIC20 port does have a few restrictions when using MPAGD:

- Screen size max 22 x 11 big characters (8x16)

- Resolution high res mode max 178 x 178 pixels

- Resolution multicolour mode max 89 x 178 pixels

- Colour can be set for a 8 x 16 block

### 2.2 VIC20 MPAGD Suite

A MPAGD Suite is a workaround for creating and testing games for multiple platforms. How does the this work?

1. *Create an AGD text file:*

First you create a text file (the so called .AGD file) which contains commands to define all graphics and the game plot. This can be done manually or by using WinAGD. The commands are AGD commands and can be found in the manual of MPAGD.

1. *Compile the AGD text file to an asm source file:*

After creating the .AGD file, you need to compile this file.

The compiler compiles the AGD commands into asm instructions and merges an engine file. The engine file contains routines needed for playing the game.

1. *Assemble the asm source file to a bin file:*

After the asm file is created, an assembler is used to assemble a binary file.

This binary file can be used for an emulator. If the used platform used disk images, a diskimage is created and the binary file is added to this image.

1. *Start an emulator with the bin file:*

The final step is starting an emulator with the binary file or diskimage for testing.

**Folder structure:**

You need a special folder structure to automate the process.

The folder structure looks like this:

Suite VIC20 Bitmap

+-> <AGD> , the compiler and engine are stored here

+-> <CC65> , the assembler is stored here

+-> <Convert> , a program to extract an AGD text file from a Spectrum snapshot is stored here

+-> <GTK3VICE-3.8-win64> , the VICE emulator is stored here

+-> <Music> , screens to be displayed are stored here

+-> <Scripts> , AGD test files are stored here

+-> build.bat , script file to start the process

### 2.3 Create hires game

- Screen : bitmap hires mode 178 x 178

The screen is switched to 22 x 11 big character mode

- Character : 8 x 16 pixels

- Colours : 8

PAPER : 0 - 7, sets the background colour for the complete screen

INK : 0 - 7, sets the front colour per attribute colour block

BORDER : 0 - 15, sets border colour.

### 2.4 Create multicolour game

- Screen : bitmapmode multicolour mode 89 x 178

The screen is switched to 22 x 11 big character mode

- Characters : 4 x 16 pixels

- Colours : 16

PAPER : 8 - 15, switch to Multicolour mode (for colours 0-7)

INK : 8 - 15, switch to Multicolour mode (for colours 0-7)

BORDER : 0 - 15, sets border colour.

AUXCOLOUR : 0 - 15, sets the auxiliary colour for multicolour mode

### 2.5 Create game using WinnAGD (hack)

Use Atom port

## 3.0 Features:

### 3.1 Joystick

- Input : Keyboard

Joystick

### 3.2 Turbodisk

The Turboload file is stored in the folder ...\CC65\Turboload

You can compile Turboload with BUILD TURBOLOAD

This compiles the asm file into a binary file and copies TURBODISK into the ...\Suite VIC20 Bitmap\CC65 folder.

The BUILD.BAT for MPAGD adds this file standard to the diskimage.

You can use this file by adding code to the MPAGD file:

LOAD "TURBODISK"

CALL $a400

Add this code in EVENT INTROMENU and only load it once.

### 3.3 Music

The music player file is stored in the folder ...\CC65\Music

You can compile eg PLAYER1 with BUILD PLAYER1

This compiles the asm file into a binary file and copies it into the ...\Suite VIC20 Bitmap\Music folder.

The BUILD.BAT for MPAGD adds this file standard to the diskimage.

There is an interrupt routine which is called at 50Hz

- Loadaddress : $a900

- Filesize : max $c000-$a900 = 5888 bytes (5,75kB)

- Zeropage : $90-$ff

- API calls : $a900 = jmp music\_init

$a903 = jmp music\_on

$a906 = jmp music\_off

$a909 = jmp music\_play

Add this code in EVENT INTROMENU and only load it once.

LOAD "PLAYER1" ; load music

USER 0 ; music\_init

The API calls can be called in the code by adding:

USER 0 ; music\_init

USER 1 ; music\_on

USER 2 ; music\_off

USER 3 ; music\_play

### 3.4 Pictures

Multipaint

The picture file is stored in the folder ...\CC65\Picture

A picture needs 2 files, a data- and a colour file.

The name of the picture should be the gamename with an addition:

1c for colour file

1d for data file

If you have more pictures in the same game, call them ...1c, ... 2c, etc

You can compile eg PICTURE1C with BUILD PICTURE1C

This compiles the asm file into a binary file and copies it into the ...\Suite VIC20 Bitmap\Pictures folder.

The BUILD.BAT for MPAGD looks for pictures with the gamename an adds these files to the diskimage.

You can display a picture by adding code to the MPAGD file:

CLS

LOAD "GAME1C"

LOAD "GAME1C"

### 3.5 User routines

## 4.0 Credits

- Jonathan

- Kees

- Mauro

- Sebastian

-