The graphics mode is automatically switched on by calling “clg” or printing CHR$ 12 or calling the vz_mode() function.

Quick start

```
zcc +vz -lm -o adventure.vz adv_a.c
```

-or-

```
zcc +vzansi -lm -o adventure.vz adv_a.c
```

Using The ”-startup=2” option a BASIC loader block is embedded in the CRT0 stub, so that the resulting binary file is ready for the “RUN” command. By default (startup=1) an auto-run binary block is created.

To get a WAV version of the program it is necessary to add the ”-cteate-app” and the ”-Cz–audio” options. To tweak the speed and gain little time ”-Cz–fast” is available.

appmake extras

Appmake is able to convert the compiled program from VZ to the newer CAS format; optionally a wav can be created for loading onto the original hardware, even in a slightly 'faster' mode.

To compile a program and generate a CAS and a WAV file:

```
zcc +vz -create-app -o rpn rpn.c -Cz--audio
```

To use appmake as a dumb converter and speed-up a little the audio stream:

```
appmake +vz -b airstrip.vz --audio --fast
```

Functions

void vz_bgrd(int n)

void vz_brick(void *addr, char byte)

void vz_char_draw(int x, int y, int c, char ch)

void vz_clrscr(void)

void vz_color(int n)

int vz_getch(void)

void vz_gotoxy(int x, int y)

void vz_inch(void)

void vz_line(int x1, int y1, int x2, int y2, int c)

char *vz_midstr(char *str, int pos)

void vz_mode(int n)

void vz_plot(int x, int y, int c)

void vz_score(void *addr, char byte)

void vz_setbase(void *start)

void vz_shape(int x, int y, int w, int h, int c, char *data)

void vz_sound(int freq, int cycles)

void vz_soundcopy(char *dst, char *src, int size, int sound1, int sound2)