

GOING



Ever fancied making your own levels for Commando? Well, read on and I will show you how. Making levels in Commando is easy, real easy. Even the most novice of Spectrum users can have a go.

The good news is that we don't have to worry about the enemy soldiers. They will interact with the scenery automatically. For the majority of the time, all that is required is for you to look up the code for the graphic you require (see table below) and decide which column to locate it. There are some small exceptions to the rule however, such as having to slot the different components that make up bridges and rivers together with predetermined amounts of space.

Have a quick browse at the graphical table below and we'll set about writing our first level. Look up the value for a tree and a single grenade. The graphical code for a tree is \$83 and for a single grenade it is \$0F.

We will place the tree in column 10 (ten) and the grenade in column 1 (one). In hexadecimal notation the equivalent of 10 is \$0A and 1 is \$01.

Let's put them into place then:

\$83, \$0A ; *Here's the tree in column \$0A*
\$0F, \$01 ; *Here's the single grenade in column \$01*

Putting these items one after another means that they will appear on the same **ROW**. That's all very well until we want to put objects at different heights on the screen.

In order to achieve this we use Land Space and a quick glance at the table shows the code of Land Space to be \$00. Inserting Land Space tells the program to move up the screen before commencing to draw a new object. How far we move up the screen depends on the value we give to Land Space. The value can be a number from \$00-\$FF (0-255). The smaller the number, the less we move and likewise, the greater the number, the further we move and the more land we have to traverse before reaching the next object.

Examples

\$00, \$42 ; *Example 1*

\$00, \$CE ; *Example 2*

\$00, \$6A ; *Example 3*

Let's update our program then:

```
$83, $0A
$00, $33      ; ← Here's our "space"
$0F, $01
```

This means that there will be a nice amount of walking space between our tree and the next object, a single grenade. In this case it works out to be approximately half a screen worth of space.

Now that our program is starting to grow we better start to keep an eye on it so load up your copy of Commando and open your assembler.

I will be using Spin's assembler. EmuZWin has a built in assembler and works in a similar fashion. For other users such as those using Spectaculator I recommend that you first type out the program using notepad and then use PasmO. (Full instructions at end of this tutorial)

<http://www.arrakis.es/~ninsesabe/pasmo/>

It is also worth pointing out at this stage that during testing "graphical corruption" will occur but do not worry. At this stage we are testing the design only.

Now have a quick browse at the 'Levels Memory Layout' table below and you will see that the code for level 1 starts at \$9236.

With Spin, go to Tools>z80 assembler and the assembler will open showing a new page. Now type the following but press the TAB key before entering each line.

```
org $9236          ;this is the address of the start of level one

defb $83, $0A      ;a tree
defb $00, $33      ;space
defb $0F, $01      ;a single grenade
defb $00, $FF      ;this is just some "extra" space so as not
                   ;to clutter our level with the original game
                   ;data which is still in memory
```

Load up your original game of Commando then go back to the assembler window and press ALT+A to assemble the data. If all has gone well, the assembler will report NO errors and you can now start the game. Hey, look at that! There's our tree and grenade but what's this?!? When we get killed the game resets. This brings us to an important lesson about spawn points:

\$22 Spawn Point/Level End

This (\$22) will be followed by either \$00 or \$01.

\$22, \$00 signifies a spawn point i.e. after death you will regenerate at this point.

\$22, \$01 signifies that you would like the level to end and fight the last remaining barrage of soldiers.

An important point to remember here is that ALL new games must start with \$22, \$00. Upon death the program backtracks to the last instance of \$22 and then adds two so it knows where to restart from after you lose a life. At the end of a level, the last piece of code will always be \$22, \$01 and thus adding two here will always point to the start of the next level. However, at the beginning of level one we have no previous 'Level End' code and thus must enter a spawn point first. Failure to do so will result in a crash.

So we can see that we need to start our level with \$22, \$00 and our program now reads thus:

```
org $9236          ;start of level one

defb $22, $00      ;this will ALWAYS be your first line
defb $83, $0A
defb $00, $33
defb $0F, $01
defb $00, $FF
```

Another point to remember with spawn points is that if we have an instruction such as the alarm sounding or NME with rocket launchers in action (explained later) then we must initiate these again after a spawn point has been reached.

Next, let's add a sandbag position.

\$8A, \$8B, \$8C Sandbags

A simple case of deciding which column you want the sandbags to appear in. Just take into consideration that the end sections take up two columns and the middle one three columns.

Example 1

```
$8A, $00      ;
$8B, $02      ;a full width sandbag defensive position starting at column $00
$8C, $05      ;
```

Example 2

\$8B, \$1D ;draws a semi width (left and middle sections only) sandbag
\$8A, \$1B ;defensive position at the RHS of the screen

Example 3

\$8A, \$00 ;draws a very wide sandbag defensive
\$8B, \$02 ;position starting at column \$00
\$8B, \$05 ;The second \$8B won't contain an NME soldier though as they
\$8B, \$08 ;appear only every other instance of \$8B. This example contains
\$8C, \$0B ;two enemy soldiers behind the sandbags

That seems simple enough so we will draw the full width sandbag position to the right of the screen starting in column \$14.

Our program now becomes:

```
org $9236

defb $22, $00
defb $83, $0A
defb $00, $33
defb $0F, $01
defb $00, $26 ;I've added a bit more space thus
               ;moving up the screen again
defb $8A, $14 ;draw sandbags LHS in column $14
defb $8B, $16 ;draw sandbags middle with NME starting at column $16
defb $8C, $19 ;draw sandbags RHS in column $19
               ;i.e. 3 columns after the middle section
defb $00, $FF
```

Now you've got the hang of sandbags let's add a trench.

\$23, \$24, \$25 Trenches

Exactly the same principle as for sandbags except the middle section takes up four columns.

Example 1

\$23, \$05 ;
\$24, \$07 ;draws a full width trench starting at column \$05
\$25, \$0B ;

Example 2

\$23, \$09 ;draws a very wide trench starting at column \$09
\$24, \$0B ;this example will have the full quota of
\$24, \$0F ;four NME soldiers as for every middle section
\$24, \$13 ;drawn (\$24) there will be an NME positioned
\$24, \$17 ;there unlike the sandbags
\$25, \$1B

As the sandbags are to the right we will place the trench to the left starting in column three.

org \$9236

defb \$22, \$00
defb \$83, \$0A
defb \$00, \$33
defb \$0F, \$01
defb \$00, \$26
defb \$8A, \$14
defb \$8B, \$16
defb \$8C, \$19

defb \$00, \$40 ;Adding a bit more space to move up the screen again
defb \$23, \$03 ;draw a trench LHS in column \$03
defb \$24, \$05 ;draw a trench middle with NME in column \$05
defb \$25, \$09 ;draw a trench RHS in column \$09

defb \$00, \$FF

Another grenade bonus is required now I think. We will we also add a boulder with NME, a mortar and another tree too. These all follow the same principle whereby you just need to decide which column to place the object in.

org \$9236

defb \$22, \$00
defb \$83, \$0A
defb \$00, \$33
defb \$0F, \$01
defb \$00, \$26
defb \$8A, \$14
defb \$8B, \$16
defb \$8C, \$19
defb \$00, \$40
defb \$23, \$03
defb \$24, \$05
defb \$25, \$09

```

defb $00, $37 ;Added more space after the trench
defb $10, $0A ;draws a triple grenade in column $0A
defb $83, $12 ;draws a tree in column $12
defb $87, $05 ;draws a boulder with NME in column $05
defb $12, $00 ;draws a right mortar in column $00

```

```

defb $00, $FF

```

When you test out the above you will immediately notice that the new items are all on the same row. This is because we didn't add any space in between drawing the objects. We only added it after drawing the trench and the next four items appear one after another with no space between. When you want objects on the same row just don't add any space.

Next, we will add a cliff.

\$82 \$26 Cliff and Cliff Edges

In normal circumstances the screen comprises of columns numbered from \$00 (LHS) to \$1F (RHS). When dealing with Cliffs we can also add \$80 onto the column to "mirror" the image.

Note that the last column we can start drawing a cliff (due to the size of the graphic) so that it fits on the screen without overlap is \$1A. This also applies to other larger objects.

Example

```

$82, $10      ;draw a cliff in column $10
$82, $8A      ;draw a cliff in column $0A and mirror it by adding $80

               ;Result is a nice big cliff.

```

To complete the cliff we add cliff edges. With cliff edges we use the same system as for cliffs.

Example

```

$82, $10      ;draw a cliff in column $10
$82, $8A      ;draw a cliff in column $0A and mirror it by adding $80
$26, $09      ;draw a cliff edge in column $09
$26, $96      ;draw a cliff edge in column $16 and
               ;add $80 to mirror the image.

               ;Result is a nice big cliff with cliff edges.

```

However, when we are drawing cliffs at the edge of the screen it is normal practice that we wish for soldiers to jump off them and attack. To allow this we must make a slight change to the numbers we add onto the column value for the cliff edges. For single cliffs you add \$40 onto the column and to mirror the image add \$C0.

Example

\$82, \$9A ;draw a mirrored cliff in column \$1A (adding \$80 to mirror)
\$26, \$59 ;still drawing a cliff edge in column \$19 but now adding \$40

Or

\$82, \$00 ;draw a cliff in column \$00 ie. LHS of screen
\$26, \$C6 ;draw the cliff edge in column \$06 but mirror it by adding \$C0

Now watch as the soldiers jump off the cliffs to attack.

To allow jumping from double cliffs, a normal and a mirrored cliff side by side, you add \$60 onto the column and when the image is mirrored you add \$20.

Example

\$82, \$80 ;draw a mirrored cliff in column \$00 (adding \$80 to mirror)
\$82, \$06 ;draw a cliff in column \$06
\$26, \$EC ;draw a mirrored cliff edge in column \$0C
 ;adding \$C0 to mirror then \$20 to allow cliff jumping

Or

\$82, \$1A ;draw a cliff in column \$1A
\$82, \$94 ;draw a mirrored cliff in column \$14 (adding \$80 to mirror)
\$26, \$73 ;draw a cliff edge in column \$13 (add \$60 to allow cliff jumping)

I've chosen to include the single cliff that soldiers can jump off and place it at the RHS of the screen. At this stage we will also include the POW bonus.

\$20 POW

Every time you want a POW bonus to appear just add \$20, \$00 into your code. They will always start running from the same position in the centre of the screen.

Placing them in our program:

```
org $9236

defb $22, $00
defb $83, $0A
defb $00, $33
defb $0F, $01
defb $00, $26
defb $8A, $14
defb $8B, $16
defb $8C, $19
defb $00, $40
```

```

defb $23, $03
defb $24, $05
defb $25, $09
defb $00, $37
defb $10, $0A
defb $83, $12
defb $87, $05
defb $12, $00

defb $00, $20 ;add more space
defb $82, $9A ;draw a cliff in column $1A ie. RHS of screen
defb $26, $59 ;drawing a cliff edge in column $19 but now adding $40
                ;onto the column to allow soldiers to jump off
defb $00, $31 ;add more space and then
defb $20, $00 ;include the POW bonus
                ←
defb $00, $FF

```

With the program starting to get rather large now only new code that we will be adding to our program will be shown from this point. All newly added code will be added between the last but one and the last line each time. This is shown by ← above. The finished demo, to which you can refer, is included with this document and is named demo.asm

Now we'll add some transport.

\$1A, \$1B, \$1C Truck, Motorbike and Jeep

When you want one of these vehicles to appear, use the following code

```

$1A, $00      ;Truck (The second value is ALWAYS $00)
$1C, $FF      ;Jeep (The second value is ALWAYS $FF)
$1B, $FF      ;Motorbike (The second value is USUALLY $FF)
                ;The Motorbike second value ($FF) can be
                ;substituted for a value $01-$71)

```

With motorbikes, a value of \$FF signifies that the motorbike will travel left→right across the screen before disappearing from view. You can change the \$FF for a value \$01-\$71 so that it still travels across the screen but stops and remains in view. \$01 is the LHS of the screen and \$71 is the rightmost. \$1B, \$3C will stop the motorbike approximately half way across the screen.

Let's add those into our code:

```
$00, $72      ;a nice amount of space to allow room for the POW bonus
$1B, $FF      ;initiate the motorbike
$00, $45      ;some more space
$1C, $FF      ;initiate the jeep
$00, $32      ;space
$1A, $00      ;initiate the truck
```

Next we'll add a barracks and a cave at opposite ends of the screen.

\$95 Barracks

These MUST appear in column \$00 on the left hand side of the screen so the code for the barracks will always be \$95, \$00.

\$1D, \$1E, \$1F Caves

Simple to draw and to use. We just have to remember to insert a set value of space in between drawing the different cave sections.

Single Cave

```
$1F, $1A      ;draw a cave bottom section
$00, $03      ;insert some space. This value will ALWAYS be the same
$1E, $1A      ;draw the main cave section
$00, $13      ;insert more space. This value will ALWAYS be the same
$1D, $1A      ;finally add the top cave section
```

Multiple Caves

For multiple caves you would repeat adding \$1E, \$1A and \$00, \$13 after the \$00, \$13 above before finally adding the \$1D, \$1A to end the caves.

Example

```
$1F, $1A      ;draw a cave bottom section
$00, $03      ;insert some space. This value will ALWAYS be the same
$1E, $1A      ;draw the main cave section
$00, $13      ;insert more space. This value will ALWAYS be the same
$1E, $1A      ;draw the main cave section
$00, $13      ;insert more space. This value will ALWAYS be the same
$1E, $1A      ;draw the main cave section
$00, $13      ;insert more space. This value will ALWAYS be the same
$1D, $1A      ;finally add the top cave section
```

Adding the single cave into our program:

```
$00, $32      ;space
$95, $00      ;draw a barracks in column $00
               ;no space between the barracks and cave
               ;signifies that they will be on the same row
$1F, $1A      ;draw a cave bottom section in column $1A
$00, $03      ;insert the required space
$1E, $1A      ;draw a cave in column $1A
$00, $13      ;insert the required space
$1D, $1A      ;draw a cave top section in column $1A
```

The player must be pretty tired by this point so we will boost their spirit with the fact that they have reached the halfway bridge and include that next.

Halfway bridge

You can put as many of these in your level as you wish but the tunnel must always be in the middle. The code will remain the same, the only exception being whether to include a motorbike riding across it or not.

Example (Without Motorbike)

```
$09, $0B      ;draw wall from right to left starting in column $0B
$09, $1F      ;draw wall from right to left starting in column $1F
$96, $0C      ;draw tunnel
$00, $17      ;space
$17, $1F      ;draw white road from right to left starting in column $1F
$00, $0C      ;space
$08, $1F      ;draw final bridge section
```

Example (With Motorbike)

```
$09, $0B      ;draw wall from right to left starting in column $0B
$09, $1F      ;draw wall from right to left starting in column $1F
$96, $0C      ;draw tunnel
$00, $17      ;space
$17, $1F      ;draw white road from right to left starting in column $1F
$00, $01      ;space
$1B, $3C      ;draw moving motorbike to position $3C
               ;(approximately half way across screen)
$00, $0B      ;space
$08, $1F      ;draw final bridge section
```

We will use the 'with motorbike' option so we'll include everything printed under the 'Example (With Motorbike)' heading above PLUS an initial line of \$00, \$41 for some extra space before commencing to draw the bridge. Another idea will be to include a spawn point at this point so tag the following code onto the end as well.

```

$00, $55      ;space
$22, $00      ;spawn point
$00, $54      ;space
               ;take particular note that this amount of space will mean that the
               ;next object (log crossing) is not drawn on the screen straight
               ;away (see below) and thus will appear correctly.

```

We will start the next section of our level with a log crossing.

River Crossings

There are two options available here. Simply copy and paste the code below for your own level. You can have the log crossing to the right or the log crossing to the left. The first one you encounter on a level must not be on the starting screen otherwise you won't see the full body of water.

Code for LEFT river crossing

```

$98, $08      ;draw the logs in column $08
$00, $02      ;land space
$0D, $10      ;draw the water
$00, $19      ;land space
$19, $10      ;draw water strip 1
$00, $05      ;land space
$0E, $10      ;draw water strip 2

```

Code for RIGHT river crossing

```

$98, $10      ;draw the logs in column $10
$00, $02      ;land space
$0D, $08      ;draw the water
$00, $19      ;land space
$19, $08      ;draw water strip 1
$00, $05      ;land space
$0E, $08      ;draw water strip 2

```

Decide which type you would like then copy the code into your program. After negotiating the log crossing we will up the tempo with an attack by rocket launcher wielding soldiers. Whilst the attack is taking place we can still add other items onto the map as desired. In this case we will add a bunker and a hut.

\$28 NME with rocket launcher

This takes the form \$28, \$XX where XX is a number \$01-\$FF. The easiest way to remember this is that the lower the number of XX, the greater the ferocity of the attack will be. When the player has reached a certain point and you wish to turn the attack off, simply add \$28, \$00 into your code.

Example 1

```
$28, $10      ;A nasty attack
$00, $60      ;over this area of land space
$28, $00      ;Attack finishes here
```

The above example is at its most simplest. You can in fact place any number of items or commands in between starting the attack and finishing it.

You can also change the frequency of the attack as you go.

Example

```
$28, $40      ;varying the attack
<perform some action>
$28, $1E
<perform some action>
$28, $00
```

\$81 Bunker

Pick a column for the bunker from \$00-\$1A. Now depending on the column you chose, the bunker will be drawn facing left or right automatically.

\$94 Big and small huts

Choose a column \$00-\$1A for the small hut. For the larger hut add \$80 onto the column. Max. column value for large huts is \$19. Adding \$80 gives \$99.

Example

```
$94, $09      ;draws a SMALL hut in column $09
$94, $89      ;draws a LARGE hut in column $09 ie. $09+$80
```

Now we will include this data into our program.

```
$28, $30      ;initiate the attack by the NME with rocket launchers
$00, $36      ;space
$94, $8B      ;draws a large hut in column $0B
$00, $47      ;space
```

```

$81, $05      ;draw a bunker in column $05
$00, $63      ;space
$28, $00      ;end of NME with rocket launchers

```

Next, we will insert a road.

\$04, \$05, \$06 Roads and how to construct one

Roads run diagonally down from the left across half the screen.

```

$05, $10      ;from the middle of the screen draw a line diagonally down.
$00, $3F      ;Space ie. width of road. This value gives normal road width
$05, $00      ;before starting the bottom edging from the LHS of the screen

```

```

$06, $10      ;from the middle of the screen draw a line diagonally down.
$00, $3F      ;Space ie. width of road. This value gives normal road width
$06, $00      ;before starting the top edging from the LHS of the screen

```

Then we need to add road markings. Here is the finished code:

```

$05, $10      ;draw road bottom edge from column $10 diagonally down right
$00, $32      ;space
$04, $1B      ;draw road marking in column $1B
$00, $0C      ;space
$05, $00      ;draw road bottom edge from column $00 diagonally down right
$00, $0C      ;space
$04, $15      ;draw road marking in column $15
$00, $06      ;space
$06, $10      ;draw road top edge from column $10 diagonally down right
$00, $12      ;space
$04, $0F      ;draw road marking in column $0F
$00, $18      ;space
$04, $09      ;draw road marking in column $09
$00, $13      ;space
$06, $00      ;draw road top edge from column $00 diagonally down right
$00, $05      ;space
$04, $03      ;draw road marking in column $03

```

To make a road wider we can add space in between the \$05,\$00 and \$06,\$10 above i.e. \$00, \$50 gives a very wide road.

```

$05, $10      ;from the middle of the screen draw a line diagonally down.
$00, $3F      ;Space i.e. width of road. This value gives normal road width
$05, $00      ;before starting the bottom edging from the LHS of the screen

```

```

$00, $50      ;extra space to make a wider road

```

```

$06, $10      ;from the middle of the screen draw a line diagonally down.
$00, $3F      ;Space ie. width of road. This value gives normal road width

```

\$06, \$00 ;before starting the top edging from the LHS of the screen

Then it's just a case of altering the space to place the road markings correctly. The following code adds three road markings. You can play around with the code and design it just how you like.

```
$05, $10        ;draw road bottom edge from column $10 diagonally down right
$00, $3f        ;space
$05, $00        ;draw road bottom edge from column $00 diagonally down right
$00, $18        ;space
$04, $1C        ;draw road marking in column $1C
$00, $2D        ;space
$04, $10        ;draw road marking in column $10
$00, $0A        ;space
$06, $10        ;draw road top edge from column $10 diagonally down right
$00, $24        ;space
$04, $04        ;draw road marking in column $04
$00, $1A        ;space
$06, $00        ;draw road top edge from column $00 diagonally down right
```

Insert the code for the normal road into your code. After we have traversed the road we will insert another spawn point. At this point we will also turn on the audible alarm and have another attack of NME with rocket launchers.

\$27 Alarm

When you would like the alarm to sound during a game you switch it on with a '1' and turn it back off after a certain distance with a '0'.

Example

```
$27, $01        ;Turn the Alarm ON
.....        ;Perform some action
$27, $00        ;Turn the Alarm OFF
```

Adding this code into our program:

```
$22, $00        ;spawn point
$00, $5B        ;space***
$27, $01        ;turn the alarm ON
$28, $0A        ;initiate attack by NME with rocket launchers
$00, $88        ;space (attack over this distance)
$28, $00        ;end of NME with rocket launchers
$27, $00        ;turn the alarm OFF
```

*** take particular note how much space I have assigned before enabling the alarm. If you were to start from the spawn point the alarm will sound again. In the original game at the end of levels 4 and 8, not enough space is assigned and although the alarm is correctly turned on, you never hear it if starting from the last spawn point due to not enough space being assigned before initialization. You can of course add more space than that which is shown but the figure above is the absolute minimum for restarting the alarm after a spawn point.

Finally, we will add the fortress:

The Fortress

At the end of every level we need to build the fortress. This is achieved with the following code

```
$09, $0B      ;draw wall from right to left starting in column $0B
$09, $1F      ;draw wall from right to left starting in column $1F
$21, $14      ;draw the fortress gates in column $14
$00, $16      ;space
$22, $01      ;signal end of level and open the gates
```

This code will always remain the same except in cases where we want additional soldiers to be firing from the top of the wall. In this case we change the \$00, \$16 above to \$00, \$1E.

In the game code itself starting at \$E323 resides the routine for the additional soldiers on the wall.

```
$E323 LD A,($FDE5)      ;level
$E326 AND 03
$E328 JR NZ, no_soldiers_on_end_gate
$E32A soldiers_on_gate_routine
```

In the original game the AND 03 will make the gate top soldiers appear on levels 4 and 8. If we wanted the gate top soldiers to ALWAYS appear then we would never make the jump at \$E328. This is achieved with Poke \$E329,0 (Poke 58153,0). For the gate top soldiers NEVER to appear you would Poke \$E328,\$18 (Poke 58152,24).

You can also change the value next to the AND above for some more variation.

This can be achieved with Poke \$E327,X (Poke 58151,X) where X can be for example:

```
01      ;gate top soldiers will appear on levels 2,4,6,8
02      ;gate top soldiers will appear on levels 1,4,5,8
03      ;gate top soldiers will appear on levels 4,8
04      ;gate top soldiers will appear on levels 1,2,3,8
05      ;gate top soldiers will appear on levels 2,8
```

Just make sure you remember that if you code the level to have gate top soldiers then the space before the end of level signal must be \$00, \$1E and not \$00, \$16. If there are too many NME soldiers already on the screen when the fortress is reached, gate top soldiers will never appear anyway.

Making your own copy of Commando

After you have built your own levels it is time to finalise your 'personal' copy of Commando. You will need a TAPE image (TAP, TZX) of any copy of Commando. Snapshots of any kind (Z80, SNA) are **NOT** *usually* acceptable. However, where a snapshot has been taken before the game has had chance to run even just one byte, such as the 128K version with added music, then this is permitted.

For emulators with an assembler (Spin, EmuZWin)

1. Open the debugger and set a breakpoint at \$641E (25630).
2. Load your copy of Commando.
3. When the program breaks, switch to the assembler and assemble your new data.
4. Remove the breakpoint and exit the debugger to return to the game.
5. Save a snapshot.
6. Test thoroughly.

If your version of Spin is older than V0.6 then use the following method:

1. Open the debugger and set a breakpoint at \$641E (25630).
2. Load your copy of Commando.
3. When the program breaks, go to the EDIT menu and select 'Edit Mode'.
4. Enter JP \$641E in place of JP \$69D2.
5. Remove the breakpoint and exit the debugger.
6. You are now able to use the assembler.
7. After assembling return to the debugger, select Edit Mode again, this time returning the original instruction of JP \$69D2 (27090).
8. Exit the debugger.
9. Save and Test.

For emulators that can load a binary file (SpecEmu, Spectaculator)

This method is slightly more complicated.

You can use notepad or such like to build your levels. When you are happy with your work, save the result as **filename.ASM**

1. Download Pasmu or other assembler and create a binary file of the new levels.

Take note that you are creating a binary file of the levels only with this method. Thus your ORG will be \$9236 (start of level one).
No other ORG must be used. See step 6 below to add further pokes.

For Pasmu the command is **Pasmu filename.asm filename.bin**

2. Open Spectaculator and set a breakpoint at \$641E (25630).

3. When the program breaks you will see

\$641E C3 D2 69 JP \$69D2

Click on the C3 D2 69 and change to \$C3 \$1E \$64






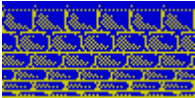











4. Remove the breakpoint and close the debugger.
5. Select 'Open File' and load the filename.BIN that you created with Pasmu.
6. Use the 'Poke Memory' option from the tools menu to add any extra pokes you require i.e.

















Poke 56567,X ;X=your character start column
Poke 56570,Y ;Y=how far down the screen your character starts.
Poke 58153,0 ;gate top soldiers on every level





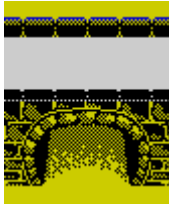

7. Open the debugger once more and navigate to \$641E
Click on the C3 1E 64 and change back to \$C3 \$D2 \$69

8. Exit the debugger.
9. Save a snapshot.
10. Test thoroughly.

Commando: Graphic Codes

Graphic	Hex Code	Decimal Code	Description
	\$00	00	Land Space
	\$04	04	Road Marking
	\$05	05	Road (Bottom Edging)
	\$06	06	Road (Top Edging)
	\$08	08	Bridge Section
	\$09	09	Wall
	\$0D	13	Water
	\$0E	14	Water Strip 2
	\$0F	15	Single Grenade
	\$10	16	Triple Grenade
	\$11	17	Quintuple Grenade
	\$12	18	Right Mortar
	\$13	19	Left Mortar
	\$17	23	Road on Bridge (White)
	\$19	25	Water Strip 1
	\$1A	26	Truck
	\$1B	27	Motorbike

	\$1C	28	Jeep
	\$1D	29	Cave (Top Edge)
	\$1E	30	Cave (Main)
	\$1F	31	Cave (Bottom Edge)
	\$20	32	POW
	\$21	33	Fortress Gates
Spawn Point/Level End	\$22	34	Spawn Point/Level End
	\$23	35	Trench LHS
	\$24	36	Trench Middle with NME
	\$25	37	Trench RHS
	\$26	38	Cliff Edge
Alarm	\$27	39	Alarm
	\$28	40	NME with Rocket Launcher
	\$81	129	Bunker
	\$82	130	Cliff
	\$83	131	Tree
	\$87	135	Boulder with NME
	\$8A	138	Sandbags LHS

	\$8B	139	Sandbags Middle with NME
	\$8C	140	Sandbags RHS
	\$94	148	Hut
	\$95	149	Barracks
	\$96	150	Tunnel
	\$98	152	Logs
n/a	\$FF	255	Levels End Marker

Commando: Levels Memory Layout

Level	Code Start Hex	Code End Hex	Code Start Decimal	Code End Decimal	Length (Bytes)
1	\$9236	\$933F	37430	37695	266
2	\$9340	\$9417	37696	37911	216
3	\$9418	\$94E7	37912	38119	208
4	\$94E8	\$9597	38120	38295	176
5	\$9598	\$9693	38296	38547	252
6	\$9694	\$978D	38548	38797	250
7	\$978E	\$9817	38798	38935	138
8	\$9818	\$98CD	38936	39117	183
End Marker	\$98CE	\$98CE	39118	39118	1
				Total:	1689

You can have as many or as few levels as you would like provided your levels total no more than 1689 bytes. You could have many smaller levels or one huge level. The choice is yours. Just remember to end your level(s) with the end marker \$FF.

Miscellaneous information

Amount of time to wait before first enemy appears

When you start a level you can specify the pause before enemy soldiers start to appear. The code resides at the following address:

```
$DD38      LD (IY+5D),X      ;X can be a number from $00-$FF
                                   ;$00 is the longest delay
                                   ;$01 is the shortest ($01=default)
```

To enter the delay, use Poke 55635,X

In an assembler use the following syntax:

```
org $DD3B
```

```
defb X
```

Spawn point (starting point) at the beginning of each level

Poke \$DCF7,X (56567)= column you start in \$00 = left, \$1D = right

Poke \$DCFA,Y (56570)= position down screen you start \$00=top, \$55=bottom

In an assembler use the following syntax:

```
org $DCF7
```

```
defb X
```

```
org $DCFA
```

```
defb Y
```

How many lives you want to start with

Poke \$7983,X (31107)

The number of bombs you start with is usually the amount of lives PLUS one.
You can alter the INC A at \$7987(31111) if you so desire for a different effect.

Remove this line to make the game really hard. Not recommended.

```
$E2ED      DEC (IY+$5D)
```

Poke 58093→5,0

Complete Levels Disassembly (original Commando)

It is a good idea to keep the map of Commando handy for a better understanding.
Here's the link to one at World of Spectrum.

<ftp://ftp.worldofspectrum.org/pub/sinclair/games-maps/c/Commando.png>

Looking at the memory table we see that level 1 occupies an area from \$9236-\$933F.

Level 1 \$9236-\$933F

org \$9236

```
defb $22, $00 ;start of levels marker and spawn point
defb $00, $04 ;space
defb $83, $0C ;draw a tree in column $0C
defb $00, $05 ;space
defb $83, $0A ;draw a tree in column $0A
defb $83, $0E ;draw a tree in column $0E still on same row
defb $00, $1E ;space
defb $83, $1C ;draw a tree in column $1C
defb $83, $02 ;draw a tree in column $02
defb $00, $15 ;space
defb $82, $9A ;draw a mirrored cliff in column $1A
defb $26, $59 ;draw a cliff edge that soldiers can jump off
                ;(added $40) in column $19
defb $00, $14 ;space
defb $87, $14 ;draw a boulder with NME in column $14
defb $00, $14 ;space
defb $83, $08 ;draw a tree in column $08
defb $00, $1E ;space
defb $83, $12 ;draw a tree in column $12
defb $83, $16 ;draw another tree in column $16
defb $00, $05 ;space
defb $83, $14 ;draw a tree in column $14
defb $00, $19 ;space
defb $83, $09 ;draw a tree in column $09
defb $00, $14 ;space
defb $82, $9A ;draw a mirrored cliff in column $1A
defb $26, $59 ;draw a cliff edge that soldiers can jump off
                ;(added $40) in column $19
defb $00, $28 ;space
defb $83, $01 ;draw a tree in column $01
defb $83, $17 ;draw another tree in column $17
defb $83, $05 ;draw yet another tree in column $05 all on the same row
defb $00, $06 ;space
defb $83, $03 ;draw a tree in column $03
defb $00, $10 ;space
defb $8A, $11 ;draw sandbags LHS in column $11
```

```

defb $8B, $13 ;draw sandbags middle with NME in column $13
defb $8B, $16 ;draw sandbags middle in column $16
defb $8C, $19 ;draw sandbags RHS in column $19
defb $00, $08 ;space
defb $83, $0D ;draw a tree in column $0D
defb $0F, $11 ;draw a single grenade in column $11
defb $00, $1C ;space
defb $83, $18 ;draw a tree in column $18
defb $82, $00 ;draw a cliff in column $00

defb $26, $C6 ;draw a mirrored cliff edge that soldiers can jump off
                ;(added $C0) in column $06
defb $00, $28 ;space
defb $83, $19 ;draw a tree in column $19
defb $00, $1E ;space
defb $82, $8A ;draw a mirrored cliff in column $0A
defb $82, $10 ;draw a cliff in column $10
defb $26, $09 ;draw a cliff edge in column $09
defb $26, $96 ;draw a mirrored cliff edge in column $16
defb $00, $36 ;space
defb $87, $14 ;draw boulder with NME in column $14
defb $00, $04 ;space
defb $20, $00 ;initiate prisoners of war
defb $00, $05 ;space
defb $22, $00 ;spawn point
defb $00, $23 ;space
defb $83, $02 ;draw a tree in column $02
defb $00, $14 ;space
defb $83, $13 ;draw a tree in column $13
defb $00, $1E ;space
defb $8B, $1D ;draw sandbags middle with NME in column $1D
defb $8B, $1A ;draw sandbags middle in column $1A
defb $8A, $18 ;draw sandbags LHS in column $18
defb $00, $08 ;space
defb $13, $19 ;draw a left mortar in column $19
defb $00, $06 ;space
defb $11, $16 ;draw a quintet grenade in column $16
defb $00, $0D ;space
defb $09, $0B ;draw a wall in column $0B. draws right>left
defb $09, $1F ;draw wall in column $1F. draws right>left
defb $96, $0C ;draw a tunnel in column $0C
defb $00, $17 ;space
defb $17, $1F ;draw white road on bridge right>left starting
                ;at column $1F (RHS)
defb $00, $01 ;space
defb $1B, $3C ;draw a motorbike manoeuvring to position $3C
                ;(approx. half way across screen)
defb $00, $0B ;space
defb $08, $1F ;draw bridge section
defb $00, $14 ;space

```

```

defb $87, $0F ;draw a boulder with NME in column $0F
defb $00, $32 ;space
defb $22, $00 ;spawn point
defb $00, $1E ;space
defb $87, $01 ;draw a boulder with NME in column $01
defb $00, $23 ;space
defb $87, $09 ;draw a boulder with NME in column $09
defb $00, $14 ;space
defb $87, $12 ;draw a boulder with NME in column $12
defb $00, $1E ;space
defb $87, $01 ;draw a boulder with NME in column $01
defb $87, $0E ;draw a boulder with NME in column $0E
defb $87, $1B ;draw a boulder with NME in column $1B
defb $00, $19 ;space
defb $8A, $02 ;draw sandbags LHS in column $02
defb $8B, $04 ;draw sandbags middle with NME in column $04
defb $8B, $07 ;draw sandbags middle in column $07
defb $8C, $0A ;draw sandbags RHS in column $0A
defb $00, $0A ;space
defb $11, $03 ;draw a quintuple grenade in column $03
defb $00, $0F ;space
defb $8A, $13 ;draw sandbags LHS in column $13
defb $8B, $15 ;draw sandbags middle with NME in column $15
defb $8B, $18 ;draw sandbags middle in column $18
defb $8C, $1B ;draw sandbags RHS in column $1B
defb $00, $12 ;space
defb $8A, $04 ;draw sandbags LHS in column $04
defb $8B, $06 ;draw sandbags middle with NME in column $06
defb $8C, $09 ;draw sandbags RHS in column $09
defb $00, $09 ;space
defb $10, $05 ;draw a triple grenade in column $05
defb $00, $05 ;space
defb $22, $00 ;spawn point
defb $00, $32 ;space
defb $8A, $0D ;draw sandbags LHS in column $0D
defb $8B, $0F ;draw sandbags middle with NME in column $0F
defb $8C, $12 ;draw sandbags RHS in column $12
defb $00, $19 ;space
defb $8A, $00 ;draw sandbags LHS in column $00
defb $8B, $02 ;draw sandbags middle with NME in column $02
defb $8C, $05 ;draw sandbags RHS in column $05
defb $00, $1C ;space
defb $8B, $1D ;draw sandbags middle with NME in column $1D
defb $8B, $1A ;draw sandbags middle in column $1A
defb $8A, $18 ;draw sandbags LHS in column $18
defb $00, $08 ;space
defb $0F, $18 ;draw a single grenade in column $18
defb $00, $32 ;space
defb $09, $0B ;draw a wall in column $0B draws right>left
defb $09, $1F ;draw a wall in column $1F draws right>left

```


defb \$21, \$14 ;draw the fortress gates
defb \$00, \$16 ;space\$16=no gate top soldiers
defb \$22, \$01 ;end of level

Level 2 \$9340-\$9417

org \$9340

defb \$00, \$5E ;space
defb \$1C, \$00 ;jeep
defb \$1A, \$00 ;truck
defb \$00, \$3C ;space
defb \$1B, \$FF ;motorbike
defb \$00, \$28 ;space
defb \$0F, \$07 ;draw a single grenade in column \$07
defb \$00, \$08 ;space
defb \$24, \$00 ;draw a trench middle with NME in column \$00
defb \$24, \$04 ;draw a trench middle with NME in column \$04
defb \$25, \$08 ;draw a trench right in \$08
defb \$00, \$19 ;space
defb \$23, \$14 ;draw a trench left in column \$14
defb \$24, \$16 ;draw a trench middle in column \$16
defb \$25, \$1A ;draw a trench right in column \$1A
defb \$00, \$1E ;space
defb \$23, \$02 ;draw a trench left in column \$02
defb \$24, \$04 ;draw a trench middle with NME in column \$04
defb \$25, \$08 ;draw a trench right in column \$08
defb \$00, \$08 ;space
defb \$23, \$16 ;draw a trench left in column \$16
defb \$24, \$18 ;draw a trench middle with NME in column \$18
defb \$25, \$1C ;draw a trench right in column \$1C
defb \$00, \$1E ;space
defb \$23, \$0A ;draw a trench left in column \$0A
defb \$24, \$0C ;draw a trench middle with NME in column \$0C
defb \$24, \$10 ;draw a trench middle with NME in column \$10
defb \$25, \$14 ;draw a trench right in column \$14
defb \$00, \$14 ;space
defb \$10, \$1B ;draw a triple grenade in column \$1B
defb \$00, \$0A ;space
defb \$22, \$00 ;spawn point
defb \$00, \$32 ;space
defb \$23, \$12 ;draw a trench left in column \$12
defb \$24, \$14 ;draw a trench middle with NME in column \$14
defb \$24, \$18 ;draw a trench middle with NME in column \$18
defb \$24, \$1C ;draw a trench middle with NME in column \$1C
defb \$00, \$19 ;space
defb \$23, \$01 ;draw a trench left in column \$01
defb \$24, \$03 ;draw a trench middle with NME in column \$03
defb \$25, \$07 ;draw a trench right in column \$07

```

defb $00, $1E ;space
defb $23, $0C ;draw a trench left in column $0C
defb $24, $0E ;draw a trench middle with NME in column $0E
defb $24, $12 ;draw a trench middle with NME in column $12
defb $24, $16 ;draw a trench middle with NME in column $16
defb $25, $1A ;draw a trench right in column $1A
defb $00, $1E ;space
defb $23, $02 ;draw a trench left in column $02
defb $24, $04 ;draw a trench middle with NME in column $04
defb $25, $08 ;draw a trench right in column $08
defb $00, $3C ;space
defb $10, $14 ;draw a triple grenade in column $14
defb $00, $0C ;space
defb $09, $0B ;draw a wall in column $0B. draws right>left
defb $09, $1F ;draw a wall in column $1F. draws right>left
defb $96, $0C ;draw a tunnel in column $0C
defb $00, $17 ;space
defb $17, $1F ;draw white road on bridge right>left
                ;starting at column $1F (RHS)
defb $00, $0C ;space
defb $08, $1F ;draw bridge section
defb $00, $50 ;space
defb $22, $00 ;spawn point
defb $00, $14 ;space
defb $95, $00 ;draw barracks in column $00
defb $00, $23 ;space
defb $10, $03 ;draw a triple grenade in column $03
defb $00, $1E ;space
defb $28, $19 ;initiate NME with rocket launchers. Ferocity =$19
defb $00, $0F ;space
defb $98, $10 ;draw a log crossing in column $10
defb $00, $02 ;space
defb $0D, $08 ;draw water
defb $00, $19 ;space
defb $19, $08 ;draw water strip1
defb $00, $05 ;space
defb $0E, $08 ;draw water strip2
defb $00, $0A ;space
defb $81, $00 ;draw a bunker in column $00
defb $00, $14 ;space
defb $28, $00 ;end of NME with rocket launchers
defb $00, $32 ;space
defb $81, $1A ;draw a bunker in column $1A
defb $00, $05 ;space
defb $22, $00 ;spawn point
defb $00, $3C ;space
defb $81, $00 ;draw a bunker in column $00
defb $00, $1E ;space
defb $1F, $1A ;draw a cave (bottom edge) in column $1A
defb $00, $03 ;space

```

```

defb $1E, $1A ;draw a cave in column $1A
defb $00, $0F ;space
defb $1F, $00 ;draw a cave (bottom edge) in column $00
defb $00, $03 ;space
defb $1E, $00 ;draw a cave in column $00
defb $1E, $1A ;draw a cave in column $1A
defb $00, $13 ;space
defb $1E, $00 ;draw a cave in column $00
defb $1E, $1A ;draw a cave in column $1A
defb $00, $13 ;space
defb $1D, $00 ;draw a cave (top edge) in column $00
defb $1D, $1A ;draw a cave (top edge) in column $1A
defb $00, $04 ;space
defb $09, $0B ;draw a wall in column $0B draws right>left
defb $09, $1F ;draw a wall in column $1F draws right>left
defb $21, $14 ;draw the fortress gates
defb $00, $16 ;space$16=no gate top soldiers
defb $22, $01 ;end of level

```

Level 3 \$9418-\$94E7

org \$9418

```

defb $00, $14 ;space
defb $94, $86 ;draw a large hut in column $06 ($80+$06)
defb $00, $19 ;space
defb $94, $94 ;draw a large hut in column $14
defb $00, $28 ;space
defb $94, $81 ;draw a large hut in column $01
defb $00, $14 ;space
defb $10, $16 ;draw a triple grenade in column $16
defb $00, $0A ;space
defb $94, $8B ;draw a large hut in column $0B
defb $00, $1E ;space
defb $94, $94 ;draw a large hut in column $14
defb $00, $0A ;space
defb $94, $84 ;draw a large hut in column $04
defb $00, $28 ;space
defb $94, $8D ;draw a large hut in column $0D
defb $00, $28 ;space
defb $94, $86 ;draw a large hut in column $06
defb $00, $19 ;space
defb $94, $99 ;draw a large hut in column $19
defb $00, $23 ;space
defb $23, $12 ;draw a trench LHS in column $12
defb $24, $14 ;draw a trench middle with NME in column $14
defb $25, $18 ;draw a trench RHS in column $18
defb $00, $0D ;space
defb $10, $14 ;draw a triple grenade in column $14

```

```

defb $94, $81 ;draw a large hut in column $01
defb $00, $16 ;space
defb $94, $8C ;draw a large hut in column $0C
defb $00, $32 ;space
defb $94, $86 ;draw a large hut in column $06
defb $00, $0A ;space
defb $22, $00 ;spawn point
defb $00, $05 ;space
defb $94, $93 ;draw a large hut in column $13
defb $00, $19 ;space
defb $24, $00 ;draw a trench middle with NME in column $00
defb $24, $04 ;draw a trench middle with NME in column $04
defb $25, $08 ;draw a trench RHS in column $08
defb $00, $1E ;space
defb $94, $88 ;draw a large hut in column $08
defb $00, $1E ;space
defb $23, $14 ;draw a trench LHS in column $14
defb $24, $16 ;draw a trench middle with NME in column $16
defb $25, $1A ;draw a trench RHS in column $1A
defb $00, $28 ;space
defb $94, $92 ;draw a large hut in column $12
defb $00, $1E ;space
defb $23, $00 ;draw a trench LHS in column $00
defb $24, $02 ;draw a trench middle with NME in column $02
defb $24, $06 ;draw a trench middle with NME in column $06
defb $25, $0A ;draw a trench RHS in column $0A
defb $00, $1E ;space
defb $11, $04 ;draw a quintuple grenade in column $04
defb $00, $28 ;space
defb $09, $0B ;draw a wall in column $0B. draws right>left
defb $09, $1F ;draw wall in column $1F. draws right>left
defb $96, $0C ;draw a tunnel in column $0C
defb $00, $17 ;space
defb $17, $1F ;draw white road on bridge right>left
                ;starting at column $1F (RHS)
defb $00, $0C ;space
defb $08, $1F ;draw bridge section
defb $00, $5A ;space
defb $1A, $00 ;truck
defb $00, $1E ;space
defb $1B, $FF ;motorbike
defb $00, $05 ;space
defb $22, $00 ;spawn point
defb $00, $28 ;space
defb $1A, $00 ;truck
defb $00, $23 ;space
defb $1B, $FF ;motorbike
defb $00, $1E ;space
defb $1C, $00 ;jeep
defb $00, $28 ;space

```

```

defb $1A, $00 ;truck
defb $00, $3C ;space
defb $28, $1E ;initiate NME with rocket launchers. Ferocity =$1E
defb $00, $28 ;space
defb $1C, $00 ;jeep
defb $28, $00 ;end of NME with rocket launchers
defb $00, $05 ;space
defb $22, $00 ;spawn point
defb $00, $3C ;space
defb $0F, $04 ;draw a single grenade in column $04
defb $00, $3C ;space
defb $8A, $02 ;draw sandbags LHS in column $02
defb $8B, $04 ;draw sandbags middle with NME in column $04
defb $8C, $07 ;draw a sandbags RHS in column $07
defb $8A, $17 ;draw sandbags LHS in column $17
defb $8B, $19 ;draw sandbags middle with NME in column $19
defb $8C, $1C ;draw sandbags RHS in column $1C
defb $00, $05 ;space
defb $10, $0A ;draw a triple grenade in column $0A
defb $10, $13 ;draw a triple grenade in column $13
defb $00, $04 ;space
defb $13, $1B ;draw a left mortar in column $1B
defb $12, $03 ;draw a right mortar in column $03
defb $00, $14 ;space
defb $09, $0B ;draw a wall in column $0B draws right>left
defb $09, $1F ;draw a wall in column $1F draws right>left
defb $21, $14 ;draw the fortress gates
defb $00, $16 ;space$16=no gate top soldiers
defb $22, $01 ;end of level

```

Level 4 \$94E8-\$9597

org \$94E8

```

defb $00, $46 ;space
defb $28, $46 ;initiate NME with rocket launchers. Ferocity =$46
defb $00, $14 ;space

defb $98, $10 ;draw a log crossing in column $10
defb $00, $02 ;space
defb $0D, $08 ;draw water
defb $00, $19 ;space
defb $19, $08 ;draw water strip1
defb $00, $05 ;space
defb $0E, $08 ;draw water strip2
defb $00, $19 ;space
defb $10, $19 ;draw a triple grenade in level $19
defb $00, $19 ;space
defb $98, $08 ;draw a log crossing in column $08

```

```

defb $00, $02 ;space
defb $0D, $10 ;draw water
defb $00, $19 ;space
defb $19, $10 ;draw water strip1
defb $00, $05 ;space
defb $0E, $10 ;draw water strip2
defb $00, $3C ;space
defb $98, $10 ;draw a log crossing in column $10
defb $00, $02 ;space
defb $0D, $08 ;draw water
defb $00, $19 ;space
defb $19, $08 ;draw water strip1
defb $00, $05 ;space
defb $0E, $08 ;draw water strip2
defb $00, $1E ;space
defb $22, $00 ;spawn point
defb $00, $14 ;space
defb $28, $46 ;initiate NME with rocket launchers. Ferocity =$46
                        ;repeated instruction required due to spawn point
defb $83, $02 ;draw a tree in column $02
defb $00, $23 ;space
defb $83, $14 ;draw a tree in column $14
defb $00, $23 ;space
defb $05, $10 ;draw road bottom edging diagonal down>right from column $10
defb $00, $13 ;space
defb $27, $01 ;turn the alarm ON
defb $8A, $01 ;draw sandbags LHS in column $01
defb $8B, $03 ;draw sandbags middle with NME in column $03
defb $8B, $06 ;draw sandbags middle with NME in column $06
defb $8C, $09 ;draw sandbags RHS in column $09
defb $00, $1E ;space
defb $04, $1B ;draw a road marking in column $1B
defb $28, $14 ;initiate NME with rocket launchers. Ferocity =$14
defb $00, $0C ;space
defb $05, $00 ;draw road bottom edging diagonal down>right from column $00
defb $00, $0C ;space
defb $04, $15 ;draw a road marking in column $15
defb $11, $04 ;draw a quintuple grenade in column $04
defb $00, $06 ;space
defb $06, $10 ;draw road top edging diagonal down>right from column $10
defb $00, $12 ;space
defb $04, $0F ;draw a road marking in column $0F
defb $00, $18 ;space
defb $04, $09 ;draw a road marking in column $09
defb $00, $13 ;space
defb $06, $00 ;draw road top edging diagonal down>right from column $00
defb $00, $05 ;space
defb $04, $03 ;draw a road marking in column $03
defb $83, $14 ;draw a tree in column $14
defb $00, $14 ;space

```

```

defb $83, $1B ;draw a tree in column $1B
defb $00, $32 ;space
defb $81, $1A ;draw a bunker in column $1A
defb $00, $1E ;space
defb $28, $00 ;end of NME with rocket launchers
defb $00, $28 ;space
defb $87, $04 ;draw a boulder with NME in column $04
defb $00, $05 ;space
defb $22, $00 ;spawn point
defb $00, $28 ;space
defb $27, $01 ;turn the alarm ON
                ;repeated instruction required due to spawn point
                ;This won't actually turn the alarm back on.
                ; Not enough space assigned first
defb $11, $19 ;draw a quintuple grenade in column $19
defb $00, $14 ;space
defb $8A, $14 ;draw sandbags LHS in column $14
defb $8B, $16 ;draw sandbags middle with NME in column $16
defb $8C, $19 ;draw sandbags RHS in column $19
defb $00, $32 ;space
defb $27, $00 ;turn the alarm OFF
defb $10, $04 ;draw a triple grenade in column $04
defb $00, $32 ;space
defb $09, $0B ;draw a wall in column $0B draws right>left
defb $09, $1F ;draw a wall in column $1F draws right>left
defb $21, $14 ;draw the fortress gates
defb $00, $1E ;space$1E=gate top soldiers
defb $22, $01 ;end of level

```

Level 5 \$9598-\$9693

org \$9598

```

defb $00, $05 ;space
defb $83, $05 ;draw a tree in column $05
defb $00, $05 ;space
defb $83, $0B ;draw a tree in column $0B
defb $00, $1E ;space
defb $82, $80 ;draw a mirrored cliff in column $00
defb $26, $EC ;draw a mirrored cliff edge in column $0C and allow cliff jumping
defb $82, $06 ;draw a cliff in column $06
defb $00, $1E ;space
defb $82, $9A ;draw a mirrored cliff in column $1A
defb $26, $59 ;draw a cliff edge in column $19 and allow cliff jumping
defb $00, $1E ;space
defb $83, $03 ;draw a tree in column $03
defb $00, $05 ;space
defb $83, $07 ;draw a tree in column $07
defb $00, $14 ;space

```

```

defb $26, $1F ;draw a cliff edge in column $1F
defb $82, $80 ;draw a mirrored cliff in column $00
defb $82, $06 ;draw a cliff in column $06
defb $26, $EC ;draw a mirrored cliff edge in column $0C and allow cliff jumping
defb $00, $3C ;space
defb $0F, $09 ;draw a single grenade in column $09
defb $83, $0C ;draw a tree in column $0C
defb $00, $08 ;space
defb $83, $0A ;draw a tree in column $0A
defb $00, $28 ;space
defb $82, $8A ;draw a mirrored cliff in column $0A
defb $82, $10 ;draw a cliff in column $10
defb $26, $09 ;draw a cliff edge in column $09
defb $26, $96 ;draw a mirrored cliff edge in column $16
defb $00, $3C ;space
defb $82, $1A ;draw a cliff in column $1A
defb $82, $94 ;draw a mirrored cliff in column $14
defb $26, $73 ;draw a cliff edge in column $13 and allow cliff jumping
defb $00, $06 ;space
defb $82, $00 ;draw a cliff in column $00
defb $26, $C6 ;draw a mirrored cliff edge in column $06 and allow cliff jumping
defb $00, $32 ;space
defb $22, $00 ;spawn point
defb $00, $05 ;space
defb $83, $0A ;draw a tree in column $0A
defb $83, $0E ;draw a tree in column $0E
defb $00, $08 ;space
defb $83, $0C ;draw a tree in column $0C
defb $00, $28 ;space
defb $83, $14 ;draw a tree in column $14
defb $00, $03 ;space
defb $83, $18 ;draw a tree in column $18
defb $00, $07 ;space
defb $83, $1B ;draw a tree in column $1B
defb $00, $0A ;space
defb $82, $80 ;draw a mirrored cliff in column $00
defb $82, $06 ;draw a cliff in column $06
defb $26, $EC ;draw a mirrored cliff edge in column $0C and allow cliff jumping
defb $00, $0F ;space
defb $82, $1A ;draw a cliff in column $1A
defb $82, $94 ;draw a mirrored cliff in column $14
defb $26, $73 ;draw a cliff edge in column $13 and allow cliff jumping
defb $00, $3C ;space
defb $11, $09 ;draw a quintuple grenade in column $09
defb $00, $0E ;space
defb $09, $0B ;draw a wall in column $0B. draws right>left
defb $09, $1F ;draw wall in column $1F. draws right>left
defb $96, $0C ;draw a tunnel in column $0C
defb $00, $17 ;space

```



```

defb $17, $1F ;draw white road on bridge right>left
                ;starting at column $1F (RHS)
defb $00, $01 ;space
defb $1B, $3C ;draw a motorbike manoeuvring to position $3C
                ;(approx. half way across screen)
defb $00, $0B ;space
defb $08, $1F ;draw bridge section
defb $00, $28 ;space
defb $87, $0E ;draw a boulder with NME in column $0E
defb $00, $1E ;space
defb $87, $04 ;draw a boulder with NME in column $04
defb $00, $28 ;space
defb $22, $00 ;spawn point
defb $00, $05 ;space
defb $8A, $18 ;draw sandbags LHS in column $18
defb $8B, $1A ;draw sandbags middle with NME in column $1A
defb $8B, $1D ;draw sandbags middle with NME in column $1D
defb $00, $05 ;space
defb $87, $06 ;draw a boulder with NME in column $06
defb $00, $14 ;space
defb $87, $19 ;draw a boulder with NME in column $19
defb $00, $0F ;space
defb $87, $10 ;draw a boulder with NME in column $10
defb $00, $1E ;space
defb $87, $16 ;draw a boulder with NME in column $16
defb $8B, $00 ;draw sandbags middle with NME in column $00
defb $8C, $03 ;draw sandbags RHS in column $03
defb $00, $22 ;space
defb $87, $02 ;draw a boulder with NME in column $02
defb $00, $18 ;space
defb $87, $0B ;draw a boulder with NME in column $0B
defb $00, $14 ;space
defb $87, $12 ;draw a boulder with NME in column $12
defb $00, $08 ;space
defb $82, $80 ;draw a mirrored cliff in column $00
defb $82, $06 ;draw a cliff in column $06
defb $26, $EC ;draw a mirrored cliff edge in column $0C and allow cliff jumping
defb $00, $14 ;space
defb $87, $19 ;draw a boulder with NME in column $19
defb $00, $1E ;space
defb $8A, $0F ;draw sandbags LHS in column $0F
defb $8B, $11 ;draw sandbags middle with NME in column $11
defb $8C, $14 ;draw sandbags RHS in column $14
defb $00, $3C ;space
defb $28, $1E ;initiate NME with rocket launchers. Ferocity = $1E
defb $23, $0A ;draw a trench LHS in column $0A
defb $24, $0C ;draw a trench middle with NME in column $0C
defb $24, $10 ;draw a trench middle with NME in column $10
defb $25, $14 ;draw a trench RHS in column $14
defb $00, $14 ;space

```

```

defb $25, $0A ;draw a trench RHS in column $0A
defb $24, $06 ;draw a trench middle with NME in column $06
defb $23, $04 ;draw a trench LHS in column $04
defb $23, $14 ;draw a trench LHS in column $14
defb $24, $16 ;draw a trench middle with NME in column $16
defb $25, $1A ;draw a trench RHS in column $1A
defb $00, $0F ;space
defb $28, $00 ;end of NME with rocket launchers
defb $09, $0B ;draw a wall in column $0B draws right>left
defb $09, $1F ;draw a wall in column $1F draws right>left
defb $21, $14 ;draw the fortress gates
defb $00, $16 ;space$16=no gate top soldiers
defb $22, $01 ;end of level

```

Level 6 \$9694-\$978D

org \$9694

```

defb $00, $28 ;space
defb $81, $00 ;draw a bunker in column $00
defb $00, $32 ;space
defb $81, $1A ;draw a bunker in column $1A
defb $00, $32 ;space
defb $81, $00 ;draw a bunker in column $00
defb $00, $28 ;space
defb $81, $1A ;draw a bunker in column $1A
defb $00, $32 ;space
defb $1F, $1A ;draw a cave (bottom edge) in column $1A
defb $1F, $00 ;draw a cave (bottom edge) in column $00
defb $00, $03 ;space
defb $1E, $1A ;draw a cave in column $1A
defb $1E, $00 ;draw a cave in column $00
defb $00, $13 ;space
defb $11, $07 ;draw a quintuple grenade in column $07
defb $1E, $1A ;draw a cave in column $1A
defb $1E, $00 ;draw a cave in column $00
defb $00, $13 ;space
defb $1E, $1A ;draw a cave in column $1A
defb $1E, $00 ;draw a cave in column $00
defb $00, $13 ;space
defb $10, $16 ;draw a triple grenade in column $16
defb $1E, $1A ;draw a cave in column $1A
defb $1E, $00 ;draw a cave in column $00
defb $00, $13 ;space
defb $1E, $1A ;draw a cave in column $1A
defb $1E, $00 ;draw a cave in column $00
defb $00, $13 ;space
defb $1E, $1A ;draw a cave in column $1A
defb $1E, $00 ;draw a cave in column $00

```

```

defb $00, $13 ;space
defb $1D, $00 ;draw a cave (top edge) in column $00
defb $1D, $1A ;draw a cave (top edge) in column $1A
defb $00, $23 ;space
defb $09, $0B ;draw a wall in column $0B. draws right>left
defb $09, $1F ;draw wall in column $1F. draws right>left
defb $96, $0C ;draw a tunnel in column $0C
defb $00, $17 ;space
defb $17, $1F ;draw white road on bridge right>left
;starting at column $1F (RHS)

defb $00, $0C ;space
defb $08, $1F ;draw bridge section
defb $00, $3C ;space
defb $25, $00 ;draw a trench RHS in column $00
defb $00, $1E ;space
defb $22, $00 ;spawn point
defb $00, $05 ;space
defb $24, $1C ;draw a trench middle with NME in column $1C
defb $23, $1A ;draw a trench LHS in column $1A
defb $00, $21 ;space
defb $24, $00 ;draw a trench middle with NME in column $00
defb $24, $04 ;draw a trench middle with NME in column $04
defb $25, $08 ;draw a trench RHS in column $08
defb $00, $19 ;space
defb $23, $10 ;draw a trench LHS in column $10
defb $24, $12 ;draw a trench middle with NME in column $12
defb $24, $16 ;draw a trench middle with NME in column $16
defb $28, $14 ;initiate NME with rocket launchers. Ferocity = $14
defb $24, $1A ;draw a trench middle with NME in column $1A
defb $25, $1E ;draw a trench RHS in column $1E
defb $00, $0A ;space
defb $11, $04 ;draw a quintuple grenade in column $04
defb $00, $0E ;space
defb $98, $10 ;draw a log crossing in column $10
defb $00, $02 ;space
defb $0D, $08 ;draw water
defb $00, $19 ;space
defb $19, $08 ;draw water strip1
defb $00, $05 ;space
defb $0E, $08 ;draw water strip2
defb $00, $14 ;space
defb $28, $00 ;end of NME with rocket launchers
defb $23, $0A ;draw a trench LHS in column $0A
defb $24, $0C ;draw a trench middle with NME in column $0C
defb $24, $10 ;draw a trench middle with NME in column $10
defb $24, $14 ;draw a trench middle with NME in column $14
defb $25, $18 ;draw a trench RHS in column $18
defb $00, $28 ;space
defb $24, $00 ;draw a trench middle with NME in column $00
defb $24, $04 ;draw a trench middle with NME in column $04

```

```

defb $24, $08 ;draw a trench middle with NME in column $08
defb $25, $0C ;draw a trench RHS in column $0C
defb $00, $1E ;space
defb $23, $12 ;draw a trench LHS in column $12
defb $24, $14 ;draw a trench middle with NME in column $14
defb $25, $18 ;draw a trench RHS in column $18
defb $00, $05 ;space
defb $22, $00 ;spawn point
defb $00, $32 ;space
defb $23, $14 ;draw a trench LHS in column $14
defb $24, $16 ;draw a trench middle with NME in column $16
defb $25, $1A ;draw a trench RHS in column $1A
defb $00, $0E ;space
defb $10, $16 ;draw a triple grenade in column $16
defb $28, $19 ;initiate NME with rocket launchers. Ferocity =$19
defb $00, $1E ;space
defb $98, $08 ;draw a log crossing in column $08
defb $00, $02 ;space
defb $0D, $10 ;draw water
defb $00, $19 ;space
defb $19, $10 ;draw water strip1
defb $00, $05 ;space
defb $0E, $10 ;draw water strip2
defb $00, $19 ;space
defb $23, $06 ;draw a trench LHS in column $06
defb $24, $08 ;draw a trench middle with NME in column $08
defb $24, $0C ;draw a trench middle with NME in column $0C
defb $25, $10 ;draw a trench RHS in column $10
defb $00, $14 ;space
defb $23, $10 ;draw a trench LHS in column $10
defb $24, $12 ;draw a trench middle with NME in column $12
defb $28, $00 ;end of NME with rocket launchers
defb $24, $16 ;draw a trench middle with NME in column $16
defb $25, $1A ;draw a trench RHS in column $1A
defb $00, $0F ;space
defb $24, $00 ;draw a trench middle with NME in column $00
defb $24, $04 ;draw a trench middle with NME in column $04
defb $24, $08 ;draw a trench middle with NME in column $08
defb $25, $0C ;draw a trench RHS in column $0C
defb $00, $0F ;space
defb $09, $0B ;draw a wall in column $0B draws right>left
defb $09, $1F ;draw a wall in column $1F draws right>left
defb $21, $14 ;draw the fortress gates
defb $00, $16 ;space$16=no gate top soldiers
defb $22, $01 ;end of level

```

Level 7 \$978E-\$9817

org \$978E

```
defb $00, $1E ;space
defb $94, $00 ;draw a small hut in column $00
defb $94, $14 ;draw a small hut in column $14
defb $94, $1A ;draw a small hut in column $1A
defb $00, $32 ;space
defb $94, $00 ;draw a small hut in column $00
defb $94, $06 ;draw a small hut in column $06
defb $94, $0C ;draw a small hut in column $0C
defb $00, $1E ;space
defb $94, $00 ;draw a small hut in column $00
defb $94, $14 ;draw a small hut in column $14
defb $94, $1A ;draw a small hut in column $1A
defb $00, $28 ;space
defb $94, $00 ;draw a small hut in column $00
defb $94, $06 ;draw a small hut in column $06
defb $94, $0C ;draw a small hut in column $0C
defb $00, $3C ;space
defb $94, $00 ;draw a small hut in column $00
defb $94, $06 ;draw a small hut in column $06
defb $94, $12 ;draw a small hut in column $12
defb $00, $28 ;space
defb $94, $0A ;draw a small hut in column $0A
defb $94, $10 ;draw a small hut in column $10
defb $00, $05 ;space
defb $22, $00 ;spawn point
defb $00, $32 ;space
defb $94, $00 ;draw a small hut in column $00
defb $94, $1A ;draw a small hut in column $1A
defb $00, $1E ;space
defb $94, $0D ;draw a small hut in column $0D
defb $00, $3C ;space
defb $12, $03 ;draw a right mortar in column $03
defb $00, $02 ;space
defb $11, $1A ;draw a quintuple grenade in column $1A
defb $00, $0F ;space
defb $09, $0B ;draw a wall in column $0B. draws right>left
defb $09, $1F ;draw wall in column $1F. draws right>left
defb $96, $0C ;draw a tunnel in column $0C
defb $00, $17 ;space
defb $17, $1F ;draw white road on bridge right>left
                    ;starting at column $1F (RHS)
defb $00, $0C ;space
defb $08, $1F ;draw bridge section
defb $00, $3C ;space
defb $1A, $00 ;truck
defb $00, $28 ;space
```

```

defb $22, $00 ;spawn point
defb $00, $3C ;space
defb $28, $28 ;initiate NME with rocket launchers. Ferocity =$28
defb $11, $04 ;draw a quintuple grenade in column $04
defb $00, $1E ;space
defb $1C, $00 ;jeep
defb $00, $50 ;space
defb $28, $0C ;initiate NME with rocket launchers. Ferocity =$0C
defb $00, $32 ;space
defb $11, $03 ;draw a quintuple grenade in column $03
defb $00, $64 ;space
defb $0F, $19 ;draw a single grenade in column $19
defb $00, $05 ;space
defb $22, $00 ;spawn point
defb $00, $05 ;space
defb $28, $0C ;initiate NME with rocket launchers. Ferocity =$0C
defb $00, $78 ;space
defb $28, $00 ;end of NME with rocket launchers
defb $00, $1E ;space
defb $09, $0B ;draw a wall in column $0B draws right>left
defb $09, $1F ;draw a wall in column $1F draws right>left
defb $21, $14 ;draw the fortress gates
defb $00, $16 ;space$16=no gate top soldiers
defb $22, $01 ;end of level

```

Level 8 \$9818-\$98CE

org \$9818

```

defb $00, $3C ;space
defb $8A, $07 ;draw sandbags LHS in column $07
defb $8B, $09 ;draw sandbags middle with NME in column $09
defb $8B, $0C ;draw sandbags middle with NME in column $0C
defb $8C, $0F ;draw sandbags RHS in column $0F
defb $00, $28 ;space
defb $98, $08 ;draw a log crossing in column $08
defb $00, $02 ;space
defb $0D, $10 ;draw water
defb $00, $19 ;space
defb $19, $10 ;draw water strip1
defb $00, $05 ;space
defb $0E, $10 ;draw water strip2
defb $00, $0F ;space
defb $98, $10 ;draw a log crossing in column $10
defb $00, $02 ;space
defb $0D, $08 ;draw water
defb $00, $19 ;space
defb $19, $08 ;draw water strip1
defb $00, $05 ;space

```

defb \$0E, \$08 ;draw water strip2
defb \$00, \$1E ;space
defb \$8B, \$00 ;draw sandbags middle with NME in column \$00
defb \$8B, \$03 ;draw sandbags middle with NME in column \$03
defb \$8C, \$06 ;draw sandbags RHS in column \$06
defb \$8B, \$1D ;draw sandbags middle with NME in column \$1D
defb \$8A, \$1B ;draw sandbags LHS in column \$1B
defb \$00, \$28 ;space
defb \$95, \$00 ;draw a barracks in column \$00
defb \$00, \$28 ;space
defb \$1B, \$FF ;motorbike
defb \$00, \$0A ;space
defb \$10, \$14 ;draw a triple grenade in column \$14
defb \$00, \$14 ;space
defb \$1B, \$FF ;motorbike
defb \$00, \$05 ;space
defb \$22, \$00 ;spawn point
defb \$00, \$1E ;space
defb \$11, \$15 ;draw a quintuple grenade in column \$15
defb \$00, \$05 ;space
defb \$83, \$02 ;draw a tree in column \$02
defb \$00, \$23 ;space
defb \$83, \$14 ;draw a tree in column \$14
defb \$00, \$23 ;space
defb \$05, \$10 ;draw road bottom edging diagonal down>right from column \$10
defb \$00, \$13 ;space
defb \$27, \$01 ;turn the alarm ON
defb \$8B, \$00 ;draw sandbags middle with NME in column \$00
defb \$8B, \$03 ;draw sandbags middle with NME in column \$03
defb \$8C, \$06 ;draw sandbags RHS in column \$06
defb \$00, \$0A ;space
defb \$12, \$02 ;draw a right mortar in column \$02
defb \$00, \$13 ;space
defb \$04, \$1B ;draw a road marking in column \$1B
defb \$28, \$14 ;initiate NME with rocket launchers. Ferocity =\$14
defb \$00, \$0C ;space
defb \$05, \$00 ;draw road bottom edging diagonal down>right from column \$00
defb \$00, \$0C ;space
defb \$04, \$15 ;draw a road marking in column \$15
defb \$00, \$06 ;space
defb \$06, \$10 ;draw road top edging diagonal down>right from column \$10
defb \$00, \$12 ;space
defb \$04, \$0F ;draw a road marking in column \$0F
defb \$00, \$18 ;space
defb \$04, \$09 ;draw a road marking in column \$09
defb \$00, \$13 ;space
defb \$06, \$00 ;draw road top edging diagonal down>right from column \$00
defb \$00, \$05 ;space
defb \$04, \$03 ;draw a road marking in column \$03
defb \$11, \$19 ;draw a quintuple grenade in column \$19

```

defb $00, $28 ;space
defb $10, $06 ;draw a triple grenade in column $06
defb $00, $14 ;space
defb $95, $00 ;draw a barracks in column $00
defb $00, $3C ;space
defb $0F, $0A ;draw a single grenade in column $0A
defb $00, $05 ;space
defb $22, $00 ;spawn point
defb $00, $05 ;space
defb $27, $01 ;turn the alarm ON
                ;repeated instruction required due to spawn point
                ;This won't actually turn the alarm back on.
                ; Not enough space assigned first
defb $28, $14 ;initiate NME with rocket launchers. Ferocity =$14
                ;repeated instruction required due to spawn point
defb $00, $82 ;space
defb $10, $04 ;draw a triple grenade in column $04
defb $28, $00 ;end of NME with rocket launchers
defb $27, $00 ;turn the alarm OFF
defb $00, $32 ;space
defb $09, $0B ;draw a wall in column $0B draws right>left
defb $09, $1F ;draw a wall in column $1F draws right>left
defb $21, $14 ;draw the fortress gates
defb $00, $1E ;space$1E=gate top soldiers
defb $22, $01 ;end of level

defb $FF      ;end of levels marker. Replay from Level 1.

```

Credits

- Commando originally published on ZX Spectrum in 1985 by Elite Systems Ltd on licence from Capcom. ZX Spectrum version programmed by Keith Burkhill and Nigel Alderton.
- Paul Johns
Wouldn't have happened without his idea/wish list.
- Team Spin (Paul Dunn, Mark Woodmass, Mark Boyd, Damien Guard)
Developed entirely using this emulator.

<http://www.zxspin.co.uk/>

- Martijn Van Der Heide
For hosting the document and providing World of Spectrum.

<http://www.worldofspectrum.org/>

- Allen Doe (mulder)
Many thanks for your continued help and feedback.

<http://www.j-one.co.uk/mulder/speccy/>

- Pasmó (C) 2004-2005 Julián Albo.

<http://www.arrakis.es/~ninsesabe/pasmo/>

- Original Commando map (via link) is by Pavero.
- EmuZWin emulator is by Vladimir Kladov.
SpecEmu emulator is by Mark Woodmass.
Spectaculator emulator is by Jonathan Needle.

<http://www.worldofspectrum.org/emulators.html>

Additional

This document has been put together to convey information only and is not intended to be a work of art. If you feel you could improve it, then please do so. If you would like to add to the document, then please do so.

If there is something that you do not understand relating to this document then send a PM to Mr. Anonymous at World of Spectrum.

January 2006