

'cnc (Castles & Crusades character generator)

'for geoCom BASIC compiler

'definition section

NAME "cnc"
CLASS "cnc 1.0"
AUTHOR "Cenbe"

'declaration section

INTVAR AT \$850a; random 'from definitions_ext
ROW 63 BYTEVAR AT \$84c1; mousepic 'mouse cursor sprite
ROW 63 BYTEVAR save_cursor
INTVAR x, d4, d6, d8, d10, d12, d20, die
BYTEVAR i, j, k, y, first, dice, roll, lowest, source_die, class, race, race_mod
BYTEVAR starting, dragging, assigning, attribute, attrs_done
ROW 4 BYTEVAR rolls4d6
ROW 6 BYTEVAR rolls, attributes, original_attributes, primes
STRVAR 4; version_string
STRVAR 64; status
STRVAR 64; prompt
STRVAR 3; modifier 'attribute bonus/penalty
LABEL title_bar, do_info, do_create, do_open, show_status, clear_status, exit
LABEL roll_dice, show_rolls, show_attributes, show_primes, show_attribute_values
LABEL show_prompt, clear_prompt, attributes_done, race_check, race_done
LABEL class_check, draw_classes, class_done, prime_check, primes_done
LABEL roll_scores, drag_check, do_drag, get_modifier, dummy

OBJFILE "cnc_icons"

'declare icons before icon tables that use them

OBJECT ok_icon, attr_icons, class_icons, race_icons, prime_icons

OBJECT die_cursor, radio_clear, radio_set

OBJFILE "cnc_constants"

OBJECT race_names, class_names, race_modifiers, class_primes

OBJECT attribute_names, attr_abbrevs

OBJFILE "cnc_obj"

OBJECT drag_process, info_dialog

'declare subsidiary menus first

OBJECT geos_menu, file_menu, main_menu

'command section

version_string = "1.0"

starting = 1

CLS

'save mouse cursor

i = 0: REPEAT

(save_cursor<i>) = (mousepic<i>)

INC i

UNTIL (i == 63)

'initialize GEOS GetRandom divisors:

d4 = (65520 / 4): d6 = (65520 / 6): d8 = (65520 / 8)

d10 = (65520 / 10): d12 = (65520 / 12): d20 = (65520 / 20)

dragging = 0

prompt = (" Castles & Crusades " + (version_string + " ")): GOSUB title_bar

MENU main_menu, 0

FRAME 0, 187, 319, 199, 255 'status area, 255 is solid line

GOSUB clear_status

PROCESS drag_process, 1 'one process in table

MOUSE ON

MAINLOOP

@title_bar

```
PATTERN 9 'horizontal stripes
RECT 49, 0, 319, 14 '48 is right edge of menu, 14 is bottom
PATTERN 0 'clear
x = (((319 - 49) - (PLEN prompt)) / 2) + 49)
RECT x, 1, (x + (PLEN prompt)), 13
SETPOSx, 9
PRINT prompt;
RETURN
```

@do_info

```
FIRSTMENU
DIALOG info_dialog
RETURN
```

@exit

```
FIRSTMENU
END
```

@do_create

```
FIRSTMENU
IF (starting == 1) THEN
    starting = 0
ELSE
    prompt = (" Castles & Crusades " + (version_string + " ")): GOSUB title_bar
ENDIF
i = 0: REPEAT
    (attributes<i>) = 0: (primes<i>) = 0
    INC i
UNTIL (i == 6)
GOSUB roll_scores
GOSUB show_attributes: GOSUB show_rolls
prompt = "Drag dice rolls to assign,": GOSUB show_prompt
ICONS attr_icons 'just the OK icon
ON 1 GOTO drag_check 'otherPressVector
RETURN
```

@do_open

```
FIRSTMENU
RETURN
```

@drag_check

```
IF ((mousedata AND $80) <> 0): RETURN: ENDIF 'mouse release? then exit
INTERRUPT OFF
i = 0: REPEAT
  y = (36 + (i * 12))
  IF ((REGION 130, y, 158, (y+11)) AND ((rolls<i>) <> 0)) THEN
    'click on dice roll, roll hasn't already been assigned
    dragging = 1: assigning = 1
    source_die = i
    i = 6 'break
  ENDIF
  INC i
UNTIL (i >= 6)

IF (dragging == 0) THEN 'check if unassigning attribute
  i = 0: REPEAT
    y = (36 + (i * 12))
    IF ((REGION 88, y, 117, (y + 11)) AND ((attributes<i>) <> 0)) THEN
      'click on attribute that's been assigned (returning roll to pool)
      dragging = 1: assigning = 0
      source_die = i
      i = 6 'break
    ENDIF
    INC i
  UNTIL (i >= 6)
ENDIF
INTERRUPT ON
IF (dragging == 1) THEN
  'SETSPR 0, die_cursor 'doesnt work w/sprite 0?
  i = 0: REPEAT
    (mousepic<i>) = (BYTE AT((ADRdie_cursor) + (INTi)))
    INC i
  UNTIL (i == 63)
  RESTART 0 'processes count from 0
ENDIF
RETURN
```

```

@do_drag 'process handler
IF ((mousedata AND $80) == 0): RETURN: ENDIF 'mouse button still down?
IF (dragging == 0): RETURN: ENDIF 'not dragging? then ignore
dragging = 0: GOSUB clear_status
BLOCK 0 'turn off process
i = 0: REPEAT 'restore mouse cursor
    (mousepic<i>) = (save_cursor<i>)
    INC i
UNTIL (i == 63)
INTERRUPT OFF
i = 0: REPEAT
    y = (36 + (i * 12))
    IF (assigning == 1) THEN
        IF ((REGION 88, y, 117, (y + 11)) AND ((attributes<i>) == 0)) THEN
            (attributes<i>) = (rolls<source_die>): (rolls<source_die>) = 0
            GOSUB show_attribute_values: GOSUB show_rolls
            i = 6 'break
        ENDIF
    ELSE
        IF ((REGION 130, y, 158, (y + 11)) AND ((rolls<i>) == 0)) THEN
            (rolls<i>) = (attributes<source_die>): (attributes<source_die>) = 0
            GOSUB show_attribute_values: GOSUB show_rolls
            i = 6 'break
        ENDIF
    ENDIF
    INC i
UNTIL (i >= 6)
INTERRUPT ON
RETURN

```

@attributes_done

attrs_done = 1

i = 0: REPEAT

IF ((attributes<i>) == 0) THEN

attrs_done = 0

i = 6 'break

ENDIF

INC i

UNTIL (i >= 6)

IF (attrs_done == 0) THEN

STRNBOX "", "Not all attributes have been assigned.", ""

RETURN

ENDIF

i = 0 : REPEAT

(original_attributes<i>) = (attributes<i>) 'save attributes

INC i

UNTIL (i == 6)

ON 1 GOTO dummy

PATTERN 2 '50% stipple

RECT 8, 120, 159, 177 'erase attr done box

RECT 126, 22, 159, 111 'erase dice rolls

'draw race input box

FRAME 150, 22, 311, 111, 255

PATTERN 0 'clear

RECT 151, 23, 310, 110

SETPOS (INT154), 33: PRINT "/BSelect race:/P";

i = 0: REPEAT

y = (45 + (i * 12))

SETPOS (INT154), y: PRINT (race_names<i>);

j = 0: first = 1: REPEAT

'race_modifiers is an array of 18 BYTES, 3 per race

'high nybble is attribute (\$f0 if no mod), low nybble is modifier (twos-complement)

race_mod = (race_modifiers<((i * 3) + j)>)

IF ((race_mod AND \$f0) <> \$f0) 'attr (\$f0 if no mod)

IF (first == 1)

PRINT " (;: first = 0

ELSE

PRINT ", ";

ENDIF

k = (race_mod AND \$0f) 'lower byte is mod

IF ((k AND \$08) == 0)

PRINT "+";

ELSE

PRINT "-";

k = ((k - 1) EXOR \$0f) 'twos-complement

ENDIF

PRINT (STRk);: PRINT " ";: PRINT (attr_abbrevs<((race_mod AND \$f0)/16)>);

ENDIF

INC j

UNTIL (j == 3)

IF (first == 0)

PRINT ")";

ENDIF

INC i

UNTIL (i == 6)

prompt = "Select race,": GOSUB show_prompt

ICONS race_icons

race = 255: ON 1 GOTO race_check

RETURN

@race_check

```
IF ((mousedata AND $80) <> 0): RETURN: ENDIF 'if mouse release, exit
INTERRUPT OFF
i = 0 : REPEAT
  y = (36 + (i * 12))
  IF (REGION 150, y, 311, (y + 11)) THEN
    IF ((race <> 255) AND (race <> i)) 'race already selected?
      y = (36 + (race * 12)) 'set Y position to previous race
      INVERT 152, (y + 1), 309, (y + 12) 'clear previous selection
      y = (36 + (i * 12)) 'restore Y position
      j = 0: REPEAT 'restore original attributes
        (attributes<j>) = (original_attributes<j>)
        INC j
      UNTIL (j == 6)
      race = 255 'previous race now deselected
    ENDIF
    INVERT 152, (y + 1), 309, (y + 12) 'show/clear selection
    IF (race == 255) THEN 'choosing race?
      race = i
      j = 0: REPEAT
        race_mod = (race_modifiers<((i * 3) + j)>)
        attribute = ((race_mod AND $f0) / 16)
        IF (attribute <> $f0) '$f0 if no mod
          k = (race_mod AND $0f) 'lower byte is mod
          IF ((k AND $08) <> 0) 'negative (two's complement)
            k = ((k - 1) EXOR $0f)
            (attributes<attribute>) = ((attributes<attribute>) - k)
          ELSE
            (attributes<attribute>) = ((attributes<attribute>) + k)
          ENDIF
        ENDIF
        INC j
      UNTIL (j == 3)
    ELSE 'no, clearing race
      race = 255
      j = 0: REPEAT
        (attributes<j>) = (original_attributes<j>)
        INC j
      UNTIL (j == 6)
    ENDIF
    i = 6 'break
  ENDIF
  INC i
UNTIL (i >= 6)
INTERRUPT ON
GOSUB show_attribute_values
RETURN
```

```

@race_done
IF (race == 255)
    STRNBOX "", "No race selected.", ""
    RETURN
ENDIF
ON 1 GOTO dummy 'turn off otherPressVector
PATTERN 2 '50% stipple
RECT 8, 120, 159, 177 'erase race done box
RECT 126, 22, 311, 111 'erase race input box
prompt = (" race: " + ( (race_names<race>) + " ")
GOSUB title_bar

'draw class input box
FRAME 198, 22, 311, 159, 255
PATTERN 0 'clear
RECT 199, 23, 310, 158
SETPOS 202, 33: PRINT "/BSelect class:/P";
i = 0: REPEAT
    y = (45 + (i * 12))
    SETPOS 202, y: PRINT (class_names<i>);: PRINT " (";
    PRINT (TEXTattribute_names, (class_primes<i>));: PRINT ")";
    INC i
UNTIL (i == 10)
prompt = "Select a character class,": GOSUB show_prompt
ICONS class_icons
class = 255: ON 1 GOTO class_check
RETURN

```

@class_check

```
IF ((mousedata AND $80) <> 0): RETURN: ENDIF 'if mouse release, exit
INTERRUPT OFF
i = 0: REPEAT
  y = (36 + (i * 12))
  IF (REGION 199, y, 310, (y+11)) THEN
    IF ((class <> 255) AND (class <> i)) 'class already selected?
      y = (36 + (class * 12)) 'set Y position to previous class
      INVERT 201, (y + 1), 308, (y + 12) 'clear previous selection
      y = (36 + (i * 12)) 'restore Y position
      j = 0: REPEAT 'clear primes
        (primes<j>) = 0
        INC j
      UNTIL (j == 6)
      class = 255 'previous class now deselected
    ENDIF
    INVERT 201, (y + 1), 308, (y + 12) 'show/clear selection
    IF (class == 255) THEN 'choosing class?
      class = i
      (primes<(class_primes<i>)>) = 1
    ELSE 'no, clearing class
      class = 255
      j = 0: REPEAT 'clear primes
        (primes<j>) = 0
        INC j
      UNTIL (j == 6)
    ENDIF
    i = 10 'break
  ENDIF
  INC i
UNTIL (i >= 10)
INTERRUPT ON
GOSUB show_primes
RETURN
```

@class_done

```
IF (class == 255)
  STRNBOX "", "No class selected.", ""
  RETURN
ENDIF
ON 1 GOTO dummy 'turn off otherPressVector
PATTERN 2 '50% stipple
RECT 8, 120, 159, 177 'erase class done box
RECT 198, 22, 311, 159 'erase class input box
prompt = ((" race: " + (race_names<race>)) + (" , class: " + ((class_names<class>) + " ")))
GOSUB title_bar
IF (race == 5) 'human
  prompt = "Select two additional primes,"
ELSE
  prompt = "Select one additional prime,"
ENDIF
GOSUB show_prompt
ICONS prime_icons
ON 1 GOTO prime_check
RETURN
```


@prime_check

```
IF ((mousedata AND $80) <> 0): RETURN: ENDIF 'if mouse release, exit
INTERRUPT OFF
i = 0: k = 0: REPEAT
    k = (k + (primes<i>)) 'count primes in k
    INC i
UNTIL (i == 6)
i = 0: REPEAT
    y = (38 + (i * 12))
    IF (REGION 16, y, 24, (y + 8)) THEN
        IF ((primes<i>) == 0) THEN 'attempting to set prime
            'humans get three primes, demi-humans get two
            IF (((race == 5) AND (k == 3)) OR ((race <> 5) AND (k == 2))) THEN
                STRNBOX "", "Maximum primes already selected.", ""
            ELSE
                (primes<i>) = 1
            ENDIF
        ELSE
            (primes<i>) = 0
        ENDIF
    ENDIF
    INC i
UNTIL (i == 6)
INTERRUPT ON
GOSUB show_primes
RETURN
```

@primes_done

```
i = 0: k = 0: REPEAT
    k = (k + (primes<i>)) 'count primes in k
    INC i
UNTIL (i == 6)
IF (((race == 5) AND (k == 3)) OR ((race <> 5) AND (k == 2))) THEN
    GOSUB clear_prompt
    STRNBOX "", "Your character is ready to play!", ""
    ON 1 GOTO dummy
ELSE
    STRNBOX "", "Humans must have three primes, ", "demi-humans must have two."
ENDIF
RETURN
```

@roll_scores '4d6, drop the lowest

i = 0: REPEAT

 j = 0: REPEAT

 dice = 1: die = d6: GOSUB roll_dice

 (rolls4d6<j>) = roll

 INC j

 UNTIL (j == 4)

 j = 0: lowest = 0: REPEAT

 IF ((rolls4d6<j>) < (rolls4d6<lowest>))

 lowest = j

 ENDIF

 INC j

 UNTIL (j == 4)

 (rolls4d6<lowest>) = 0 'discard lowest roll

 j = 0: roll = 0: REPEAT

 roll = (roll + (rolls4d6<j>))

 INC j

 UNTIL (j == 4)

 (rolls<i>) = roll

 INC i

UNTIL (i == 6)

RETURN

@show_prompt

FRAME 8, 120, 159, 177, 255

PATTERN 0 'clear

RECT 9, 121, 158, 176

SETPOS 16, 132: PRINT prompt;

SETPOS 16, 143: PRINT "click OK when done.";

RETURN

@clear_prompt

PATTERN 2 '50% stipple

RECT 8, 120, 159, 177

RETURN

@get_modifier

```
IF (roll == 0) 'uninitialized
    modifier = "+0"
    RETURN
ENDIF
IF (roll < 2) 'half-orc, rolled 3 CHA with -2 penalty
    modifier = "-4"
ELSE
    IF (roll < 4)
        modifier = "-3"
    ELSE
        IF (roll < 6)
            modifier = "-2"
        ELSE
            IF (roll < 9)
                modifier = "-1"
            ELSE
                IF (roll < 13)
                    modifier = "+0"
                ELSE
                    IF (roll < 16)
                        modifier = "+1"
                    ELSE
                        IF (roll < 18)
                            modifier = "+2"
                        ELSE
                            modifier = "+3"
                        ENDIF
                    ENDIF
                ENDIF
            ENDIF
        ENDIF
    ENDIF
ENDIF
RETURN
```

@roll_dice

```
'pass dice, no. dice to roll; die, GetRandom divisor
'return roll, sum of rolled dice
'destroyed k
k = (dice - 1): roll = 0: REPEAT
    CALL $c187 'GetRandom
    roll = (roll + (LOW(((random - 1) / die) + 1)))
    INC k
UNTIL (k == dice)
RETURN
```

@show_rolls

```
FRAME 126, 22, 159, 111, 255
PATTERN 0 'clear
RECT 127, 23, 158, 110
SETPOS 130, 33: PRINT "/Brolls:/P"
i = 0: REPEAT
    y = (45 + (i * 12))
    SETPOS (INT130), y: PRINT (STR(rolls<i>));
    roll = (rolls<i>): GOSUB get_modifier
    SETPOS (INT144), y: PRINT modifier;
    INC i
UNTIL (i == 6)
RETURN
```

@show_attributes

```
FRAME 8, 22, 118, 111, 255 'clear entire attribute area
PATTERN 0 'clear
RECT 9, 23, 117, 110
SETPOS (INT16), 33: PRINT "/Battributes:/P"
i = 0: REPEAT
  y = (45 + (i * 12))
  SETPOS (INT30), y: PRINT (TEXTattribute_names, i);
  INC i
UNTIL (i == 6)
GOSUB show_primes
GOSUB show_attribute_values
RETURN
```

@show_primes

```
i = 0: REPEAT
  y = (38 + (i * 12))
  IF ((primes<i>) == 0)
    BITMAP 2, y, radio_clear
  ELSE
    BITMAP 2, y, radio_set
  ENDIF
  INC i
UNTIL (i == 6)
RETURN
```

@show_attribute_values

```
PATTERN 0
RECT 88, 35, 117, 110 'clear scores and modifiers
i = 0: REPEAT
  y = (45 + (i * 12))
  SETPOS (INT88), y: PRINT (STR(attributes<i>));
  roll = (attributes<i>): GOSUB get_modifier
  SETPOS (INT102), y: PRINT modifier;
  INC i
UNTIL (i == 6)
RETURN
```

@show_status

```
SETPOS 4, 196
PRINT status
RETURN
```

@clear_status

```
PATTERN 0 'clear
RECT 1, 188, 318, 198
RETURN
```

@dummy

```
RETURN
```