Random monster formation

This page lists the algorithms that affect random monster formations.

Random number generation

```
C0/C496: DA         PHX
C0/C497: EE A2 1F   INC $1FA2      (Increment battle counter.)
C0/C49A: D0 09      BNE $C4A5      (Determine whether to increment salt.)
C0/C49C: AD A3 1F   LDA $1FA3      (Load battle counter salt.)
C0/C49F: 18         CLC
C0/C4A0: 69 17      ADC #$17       (Increment salt +23.)
C0/C4A2: 8D A3 1F   STA $1FA3
C0/C4A5: AD A2 1F   LDA $1FA2      (Load battle counter as RNG seed.)
C0/C4A8: AA         TAX
C0/C4A9: BF 00 FD C0    LDA $C0FD00,X  (Load random number from RNG table.)
C0/C4AD: 18         CLC
C0/C4AE: 6D A3 1F   ADC $1FA3      (Add salt.)
C0/C4B1: FA         PLX
C0/C4B2: 60         RTS            (Return.)
```

Salt ensures that the same pattern doesn't repeat often.

Monster 2 and 4 packs

Generate a random number from 0-255.

2-packs

- formation 1 if 0-191 = 75 %
- formation 2 if 192-255 = 25 %

4-packs

- formation 1 if 0-79 = 31,25 %
- formation 2 if 80-159 = 31,25 %
- formation 3 if 160-239 = 31,25 %
- formation 4 if 240-255 = 6,25 %

2-packs

CF5000-CF53FF (256 items, 4 bytes each)

```
C0/A5A7: A5 EB      LDA $EB        (Load locations 2-pack.)
C0/A5A9: C2 20      REP #$20       (16 bit accum./memory)
C0/A5AB: 0A         ASL A
```

```
C0/A5AC: 0A         ASL A          (Determine offset.)
C0/A5AD: AA         TAX
C0/A5AE: 7B         TDC
C0/A5AF: E2 20      SEP #$20       (8 bit accum./memory)
C0/A5B1: 20 96 C4   JSR $C496      (Generate a random number 0-255.)
C0/A5B4: C9 C0      CMP #$C0       (Determine which formation from the 2-
pack to load.)
C0/A5B6: 90 02      BCC $A5BA
C0/A5B8: E8         INX
C0/A5B9: E8         INX
C0/A5BA: C2 20      REP #$20       (16 bit accum./memory)
C0/A5BC: BF 00 50 CF    LDA $CF5000,X  (Load monster formation.)
C0/A5C0: 8F E0 11 00    STA $0011E0    (Save formation number to memory.)
```

Overworld 4-packs

CF4800-CF4FFF (256 items, 8 bytes each)

```
C0/C236: A5 24      LDA $24        (Load zone's 4-pack)
C0/C238: C9 FF      CMP #$FF       (was it an invalid/Veldt pack?)
C0/C23A: D0 03      BNE $C23F      (continue if not)
C0/C23C: 4C DF C2   JMP $C2DF      (if it was, go pick a Veldt formation
instead.)

C0/C23F: C2 20      REP #$20       (16 bit accum./memory)
C0/C241: 0A         ASL A
C0/C242: 0A         ASL A          (Determine formation pack rom offset.)
C0/C243: 0A         ASL A
C0/C244: AA         TAX
C0/C245: 7B         TDC
C0/C246: E2 20      SEP #$20       (8 bit accum./memory)

C0/C248: 20 96 C4   JSR $C496      (Generate a random number.)
C0/C24B: C9 50      CMP #$50       (Determine whether to load formation 1.)
C0/C24D: 90 0E      BCC $C25D
C0/C24F: E8         INX
C0/C250: E8         INX            (Point to second formation in 4-pack.)
C0/C251: C9 A0      CMP #$A0
C0/C253: 90 08      BCC $C25D      (Determine whether the load formation 2.)
C0/C255: E8         INX
C0/C256: E8         INX            (Point to third formation in 4-pack.)
C0/C257: C9 F0      CMP #$F0
C0/C259: 90 02      BCC $C25D      (Determine whether to load formation 3.)
C0/C25B: E8         INX
C0/C25C: E8         INX            (Else, point to fourth formation in 4-
pack.)
C0/C25D: C2 20      REP #$20       (16 bit accum./memory)
C0/C25F: BF 00 48 CF    LDA $CF4800,X  (Load formation from 4-pack data.)
```

```
C0/C263: 8F E0 11 00    STA $0011E0    (Store formation number to memory.)
```

Dungeon 4-packs

CF5600-CF57FF (512 items, 1 byte each)

```
C0/C3D1: AE 82 00   LDX $0082      (Load map index.)
C0/C3D4: BF 00 56 CF    LDA $CF5600,X  (Load 4-pack number used on map)
C0/C3D8: C2 20      REP #$20       (16 bit accum./memory)
C0/C3DA: 0A         ASL A
C0/C3DB: 0A         ASL A          (Determine the pack's rom offset
(multiply by 8).)
C0/C3DC: 0A         ASL A
C0/C3DD: AA         TAX
C0/C3DE: 7B         TDC
C0/C3DF: E2 20      SEP #$20       (8 bit accum./memory)
C0/C3E1: 20 96 C4   JSR $C496      (Generate a random number 0-255.)
C0/C3E4: C9 50      CMP #$50       (Determine the pack's formation to load
(same as the overworld.))
C0/C3E6: 90 0E      BCC $C3F6
C0/C3E8: E8         INX
C0/C3E9: E8         INX
C0/C3EA: C9 A0      CMP #$A0
C0/C3EC: 90 08      BCC $C3F6
C0/C3EE: E8         INX
C0/C3EF: E8         INX
C0/C3F0: C9 F0      CMP #$F0
C0/C3F2: 90 02      BCC $C3F6
C0/C3F4: E8         INX
C0/C3F5: E8         INX
C0/C3F6: C2 20      REP #$20       (16 bit accum./memory)
C0/C3F8: BF 00 48 CF    LDA $CF4800,X  (Load formation from monster 4-pack
data.)
C0/C3FC: 8F E0 11 00    STA $0011E0    (Store formation number to memory.)
```

Floating continent adds 0-3 to the stored formation number in equal probabilities.

```
C2/310F: A9 00 80     LDA #$8000     (This bit informs that the formation
will be randomized with the next 3. It is defined in the pack data CF4800,X)
C2/3112: 1C E0 11     TRB $11E0      (Test highest bit of battle formation.)
C2/3115: F0 10        BEQ $3127      (Branch if formation is not random with
the next 3.)
C2/3117: E2 30        SEP #$30
C2/3119: 7B           TDC
C2/311A: 20 5A 4B     JSR $4B5A      (Generate random number from 0-255)

C2/4B5A: DA           PHX
C2/4B5B: E6 BE        INC $BE        (Increment RNG seed.)
C2/4B5D: A6 BE        LDX $BE
```

```
C2/4B5F: BF 00 FD C0  LDA $C0FD00,X  (Load random value from RNG Table.)
C2/4B63: FA           PLX
C2/4B64: 60           RTS

C2/311D: 29 03        AND #$03       (0-3 in equal distributions.)
C2/311F: C2 31        REP #$31       (Set 16-bit A, X and Y. clear Carry)
C2/3121: 6D E0 11     ADC $11E0      (Add the generated 0-3 to the battle
formation number.)
C2/3124: 8D E0 11     STA $11E0      (Store the new formation number.)
```

Example

Floating continent pack number 0x70, 000f4b80 b1 80 b4 80 b7 80 b9 80

- 0: Behemoth or Ninja x2 or Brainpan x2, Misfit, Apocrypha or Dragon
- 1: Apocrypha, Misfit x2 or Platinum Dragon x3 or Brainpan x3 or Behemoth, Misfit x2
- 2: Apocrypha or Apocrypha x3 or Dragon or Behemoth x2
- 3: Ninja x2 or Brainpan x2, Misfit, Apocrypha or Behemoth, Misfit x2 or Ninja x2, Platinum Dragon

Probabilities:

- Ninja x2 (80/4/256*2 –> 10/64)
- Brainpan x2, Misfit, Apocrypha (2*5/64=10/64)
- Behemoth, Misfit x2 (6/64)
- Dragon (1/64+5/64=6/64)
- Platinum Dragon x3 (5/64)
- Behemoth (80/4/256 –> 5/64)
- Brainpan x3 (5/64=5/64)
- Apocrypha (5/64)
- Apocrypha x3 (5/64)
- Apocrypha, Misfit x2 (5/64)
- Ninja x2, Platinum Dragon (1/64)
- Behemoth x2 (1/64)