

Dragon Warrior Formula Guide

by Ryan8bit

Updated to v1.0 on Jan 19, 2011

```
*****
*
*           Dragon Warrior (NES) Formulas v1.0           *
*                   By Ryan8bit                         *
*                   generalkillt@yahoo.com              *
*                   01/07/11                            *
*
*****
```

Table of Contents

- I. Introduction
- II. Simplified Formulas
 - A. Journeying
 - B. Combat
 - C. Chests
 - D. Name choice and stats
- III. Precise Formulas and Code
 - A. Common algorithms
 - B. Journeying
 - C. Combat
 - D. Chests
 - E. Name choice and stats
- IV. Frequently Asked Questions

```
*****
* I. Introduction *
*****
```

I had often been curious about the formulas used to calculate damage and all sorts of other things in Dragon Warrior. Mostly I wanted to make games based on it, but I also wanted to use them to build a simulator of the game that could crank out data that could be analyzed. I have since learned the assembly language of the NES in order to learn all I could about the game's mechanics. With that knowledge, I have programmed my simulator which I will have a FAQ like this one for at a later date.

So for whatever purpose you may have, I hope you get usage out of these formulas.

I've broken down the sections into simple and complex, so that if you just want to get the general idea you want section II. If you're more into exact formulas and some actual code, you want section III.

```
*****
* II. Simplified Formulas *
*****
```

These formulas are simplified and as a result may not be exact when it comes to

rounding. For more advanced assembly code formulas, see section III.

A. Journeying

-- Items and Spells --

Herb: +23-30 HP
HEAL: +10-17 HP
HEALMORE: +85-100 HP

Torch: Unlimited
Radiant: 200 total steps
Radius of 3 squares: 80 steps
Radius of 2 squares: 60 steps
Radius of 1 square : 60 steps

REPEL: 128 total steps
Only works against enemies who have strength that is less than the hero's defense.
Fairy Water has the exact same effect.

The Silver Harp summons a Slime, Red Slime, Drakee, Ghost, Magician, or Scorpion, all with roughly equal chance.

-- Encounters --

The chances for an encounter on the overworld vary by terrain type, and by location. For every step you take, you have a random chance to enter combat.

In the zones where you fight only Slimes and Red Slimes around Tantegel (which I call Zone 0), the encounter rates are significantly lower.

Zone 0

Grasslands: 1/48
Bridge: 1/48
Forest: 1/32
Hills: 1/32
Desert: 1/16 (There is only one square of desert in this region)

Every other zone, including dungeons

Grasslands: 1/24
Swamp: 1/16
Forest: 1/16
Hills: 1/8
Desert: 1/8

Stairs: 1/24
Chest: 1/24
Barrier: 1/16
Brick: 1/16

-- Monster Sets --

The entire world map is broken down into an 8x8 grid. Each cell of the grid has a different zone, or set of enemies. This is a representation of the grids corresponding to zone number:

```

3 3 2 2 3 5 4 5
3 2 1 2 3 3 4 5
4 1 0 0 1 3 4 5
5 1 1 12 9 6 6 6
5 5 4 12 12 7 7 7
10 9 8 12 12 12 8 7
10 10 11 12 13 13 9 8
11 11 12 13 13 12 9 9

```

So for instance, the top left corner represents the 15x15 tile area surrounding Garinham. <http://www.gamefaqs.com/console/nes/file/563408/55534> is a good example of this visually.

Every zone has five possible enemies in them, and some are duplicates. The chances of encountering each one is roughly equal.

```

Zone 0: Slime, Red Slime, Slime, Red Slime, Slime
Zone 1: Red Slime, Slime, Red Slime, Drakee, Red Slime
Zone 2: Slime, Ghost, Drakee, Ghost, Red Slime
Zone 3: Red Slime, Red Slime, Drakee, Ghost, Magician
Zone 4: Ghost, Magician, Magidrakee, Magidrakee, Scorpion
Zone 5: Ghost, Magician, Magidrakee, Scorpion, Skeleton
Zone 6: Magidrakee, Scorpion, Skeleton, Warlock, Wolf
Zone 7: Skeleton, Warlock, Metal Scorpion, Wolf, Wolf
Zone 8: Metal Scorpion, Wraith, Wolflord, Wolflord, Goldman
Zone 9: Wraith, Wyvern, Wolflord, Wyvern, Goldman
Zone 10: Wyvern, Rogue Scorpion, Wraith Knight, Knight, Demon Knight
Zone 11: Wraith Knight, Knight, Magiwyvern, Demon Knight, Metal Slime
Zone 12: Knight, Magiwyvern, Demon Knight, Werewolf, Starwyvern
Zone 13: Werewolf, Green Dragon, Starwyvern, Starwyvern, Wizard
Zone 14: Poltergeist, Droll, Drakeema, Skeleton, Warlock
Zone 15: Specter, Wolflord, Druinlord, Drollmagi, Wraith Knight
Zone 16: Werewolf, Green Dragon, Starwyvern, Wizard, Axe Knight
Zone 17: Wizard, Axe Knight, Blue Dragon, Blue Dragon, Stoneman
Zone 18: Wizard, Stoneman, Armored Knight, Armored Knight, Red Dragon
Zone 19: Ghost, Magician, Scorpion, Druin, Druin

```

B. Combat

--Enemies--

These are all the stats of the enemies:

Strength is used in figuring out how much damage will be done to you.

Agility is used in figuring out how much damage will be reduced for the enemy, and it is used to determine how hard it is for you to run away.

HP is how many hit points the enemy has, and this typically varies.

SLEEP resist, STOPSPELL resist, and HURT resist are the chances that the spell will have no effect on the enemy. HURT and HURTMORE have the same chance.

Dodge is the chance that the enemy will dodge any physical attack.

GP is how much gold will be awarded, and this typically varies.

XP is how many experience points will be awarded to you.

Pattern is what the enemy decides to do when attacking. It starts with the first line and if it doesn't do that attack, it moves to the next line.

#00 : Slime

Strength: 5
Agility: 3
HP: 3
SLEEP resist: 0 / 16
STOPSPELL resist: 15 / 16
HURT resist: 0 / 16
Dodge: 1 / 64
XP: 1
GP: 1
Pattern: Attack only

#01 : Red Slime

Strength: 7
Agility: 3
HP: 4
SLEEP resist: 0 / 16
STOPSPELL resist: 15 / 16
HURT resist: 0 / 16
Dodge: 1 / 64
XP: 1
GP: 2
Pattern: Attack only

#02 : Drakee

Strength: 9
Agility: 6
HP: 5 - 6
SLEEP resist: 0 / 16
STOPSPELL resist: 15 / 16
HURT resist: 0 / 16
Dodge: 1 / 64
XP: 2
GP: 2
Pattern: Attack only

#03 : Ghost

Strength: 11

Agility: 8
HP: 6 - 7
SLEEP resist: 0 / 16
STOPSPELL resist: 15 / 16
HURT resist: 0 / 16
Dodge: 4 / 64
XP: 3
GP: 3 - 4
Pattern: Attack only

#04 : Magician

Strength: 11
Agility: 12
HP: 10 - 13
SLEEP resist: 0 / 16
STOPSPELL resist: 0 / 16
HURT resist: 0 / 16
Dodge: 1 / 64
XP: 4
GP: 9 - 11
Pattern: 50 % to cast HURT
Otherwise attack

#05 : Magidrakee

Strength: 14
Agility: 14
HP: 12 - 15
SLEEP resist: 0 / 16
STOPSPELL resist: 0 / 16
HURT resist: 0 / 16
Dodge: 1 / 64
XP: 5
GP: 9 - 11
Pattern: 50 % to cast HURT
Otherwise attack

#06 : Scorpion

Strength: 18
Agility: 16
HP: 16 - 20
SLEEP resist: 0 / 16
STOPSPELL resist: 15 / 16
HURT resist: 0 / 16
Dodge: 1 / 64
XP: 6
GP: 12 - 15
Pattern: Attack only

#07 : Druin

Strength: 20
Agility: 18
HP: 17 - 22
SLEEP resist: 0 / 16
STOPSPELL resist: 15 / 16
HURT resist: 0 / 16
Dodge: 2 / 64

XP: 7
GP: 12 - 15
Pattern: Attack only

#08 : Poltergeist

Strength: 18
Agility: 20
HP: 18 - 23
SLEEP resist: 0 / 16
STOPSPELL resist: 0 / 16
HURT resist: 0 / 16
Dodge: 6 / 64
XP: 8
GP: 13 - 17
Pattern: 75 % to cast HURT
Otherwise attack

#09 : Droll

Strength: 24
Agility: 24
HP: 19 - 25
SLEEP resist: 0 / 16
STOPSPELL resist: 14 / 16
HURT resist: 0 / 16
Dodge: 2 / 64
XP: 10
GP: 18 - 24
Pattern: Attack only

#10 : Drakeema

Strength: 22
Agility: 26
HP: 16 - 20
SLEEP resist: 2 / 16
STOPSPELL resist: 0 / 16
HURT resist: 0 / 16
Dodge: 6 / 64
XP: 11
GP: 15 - 19
Pattern: 25 % to cast HEAL, but only if its HP < 1/4 of its max HP
Otherwise 50 % to cast HURT
Otherwise attack

#11 : Skeleton

Strength: 28
Agility: 22
HP: 23 - 30
SLEEP resist: 0 / 16
STOPSPELL resist: 15 / 16
HURT resist: 0 / 16
Dodge: 4 / 64
XP: 11
GP: 22 - 29
Pattern: Attack only

#12 : Warlock

Strength: 28
Agility: 22
HP: 23 - 30
SLEEP resist: 3 / 16
STOPSPELL resist: 1 / 16
HURT resist: 0 / 16
Dodge: 2 / 64
XP: 13
GP: 26 - 34
Pattern: 25 % to cast SLEEP, but only if you're not asleep
Otherwise 50 % to cast HURT
Otherwise attack

#13 : Metal Scorpion

Strength: 36
Agility: 42
HP: 17 - 22
SLEEP resist: 0 / 16
STOPSPELL resist: 15 / 16
HURT resist: 0 / 16
Dodge: 2 / 64
XP: 14
GP: 30 - 39
Pattern: Attack only

#14 : Wolf

Strength: 40
Agility: 30
HP: 26 - 34
SLEEP resist: 1 / 16
STOPSPELL resist: 15 / 16
HURT resist: 0 / 16
Dodge: 2 / 64
XP: 16
GP: 37 - 49
Pattern: Attack only

#15 : Wraith

Strength: 44
Agility: 34
HP: 28 - 36
SLEEP resist: 7 / 16
STOPSPELL resist: 0 / 16
HURT resist: 0 / 16
Dodge: 4 / 64
XP: 17
GP: 45 - 59
Pattern: 25 % to cast HEAL, but only if its HP < 1/4 of its max HP
Otherwise attack

#16 : Metal Slime

Strength: 10
Agility: 255
HP: 4
SLEEP resist: 15 / 16

STOPSPELL resist: 15 / 16
HURT resist: 15 / 16
Dodge: 1 / 64
XP: 115
GP: 4 - 5
Pattern: 75 % to cast HURT
Otherwise attack

#17 : Specter

Strength: 40
Agility: 38
HP: 28 - 36
SLEEP resist: 3 / 16
STOPSPELL resist: 1 / 16
HURT resist: 0 / 16
Dodge: 4 / 64
XP: 18
GP: 52 - 69
Pattern: 25 % to cast SLEEP, but only if you're not asleep
Otherwise 75 % to cast HURT
Otherwise attack

#18 : Wolflord

Strength: 50
Agility: 36
HP: 29 - 38
SLEEP resist: 4 / 16
STOPSPELL resist: 7 / 16
HURT resist: 0 / 16
Dodge: 2 / 64
XP: 20
GP: 60 - 79
Pattern: 50 % to cast STOPSPELL, but only if your spell isn't blocked
Otherwise attack

#19 : Druinlord

Strength: 47
Agility: 40
HP: 27 - 35
SLEEP resist: 15 / 16
STOPSPELL resist: 0 / 16
HURT resist: 0 / 16
Dodge: 4 / 64
XP: 20
GP: 63 - 84
Pattern: 75 % to cast HEAL, but only if its HP < 1/4 of its max HP
Otherwise 25 % to cast HURT
Otherwise attack

#20 : Drollmagi

Strength: 52
Agility: 50
HP: 29 - 38
SLEEP resist: 2 / 16
STOPSPELL resist: 2 / 16
HURT resist: 0 / 16

Dodge: 1 / 64
XP: 22
GP: 67 - 89
Pattern: 50 % to cast STOPSPELL, but only if your spell isn't blocked
Otherwise attack

#21 : Wyvern

Strength: 56
Agility: 48
HP: 32 - 42
SLEEP resist: 4 / 16
STOPSPELL resist: 15 / 16
HURT resist: 0 / 16
Dodge: 2 / 64
XP: 24
GP: 75 - 99
Pattern: Attack only

#22 : Rogue Scorpion

Strength: 60
Agility: 90
HP: 27 - 35
SLEEP resist: 7 / 16
STOPSPELL resist: 15 / 16
HURT resist: 0 / 16
Dodge: 2 / 64
XP: 26
GP: 82 - 109
Pattern: Attack only

#23 : Wraith Knight

Strength: 68
Agility: 56
HP: 35 - 46
SLEEP resist: 5 / 16
STOPSPELL resist: 0 / 16
HURT resist: 3 / 16
Dodge: 4 / 64
XP: 28
GP: 90 - 119
Pattern: 75 % to cast HEAL, but only if its HP < 1/4 of its max HP
Otherwise attack

#24 : Golem

Strength: 120
Agility: 60
HP: 53 - 70
SLEEP resist: 15 / 16
STOPSPELL resist: 15 / 16
HURT resist: 15 / 16
Dodge: 0 / 64
XP: 5
GP: 7 - 9
Pattern: Attack only

#25 : Goldman

Strength: 48
Agility: 40
HP: 38 - 50
SLEEP resist: 13 / 16
STOPSPELL resist: 15 / 16
HURT resist: 0 / 16
Dodge: 1 / 64
XP: 6
GP: 150 - 199
Pattern: Attack only

#26 : Knight

Strength: 76
Agility: 78
HP: 42 - 55
SLEEP resist: 6 / 16
STOPSPELL resist: 7 / 16
HURT resist: 0 / 16
Dodge: 1 / 64
XP: 33
GP: 97 - 129
Pattern: 50 % to cast STOPSPELL, but only if your spell isn't blocked
Otherwise attack

#27 : Magiwyvern

Strength: 78
Agility: 68
HP: 44 - 58
SLEEP resist: 2 / 16
STOPSPELL resist: 0 / 16
HURT resist: 0 / 16
Dodge: 2 / 64
XP: 34
GP: 105 - 139
Pattern: 50 % to cast SLEEP, but only if you're not asleep
Otherwise attack

#28 : Demon Knight

Strength: 79
Agility: 64
HP: 38 - 50
SLEEP resist: 15 / 16
STOPSPELL resist: 15 / 16
HURT resist: 15 / 16
Dodge: 15 / 64
XP: 37
GP: 112 - 149
Pattern: Attack only

#29 : Werewolf

Strength: 86
Agility: 70
HP: 46 - 60
SLEEP resist: 7 / 16
STOPSPELL resist: 15 / 16

HURT resist: 0 / 16
Dodge: 7 / 64
XP: 40
GP: 116 - 154
Pattern: Attack only

#30 : Green Dragon

Strength: 88
Agility: 74
HP: 49 - 65
SLEEP resist: 7 / 16
STOPSPELL resist: 15 / 16
HURT resist: 2 / 16
Dodge: 2 / 64
XP: 45
GP: 120 - 159
Pattern: 25 % to breathe fire
Otherwise attack

#31 : Starwyvern

Strength: 86
Agility: 80
HP: 49 - 65
SLEEP resist: 8 / 16
STOPSPELL resist: 0 / 16
HURT resist: 1 / 16
Dodge: 2 / 64
XP: 43
GP: 120 - 159
Pattern: 75 % to cast HEALMORE, but only if its HP < 1/4 of its max HP
Otherwise 25 % to breathe fire
Otherwise attack

#32 : Wizard

Strength: 80
Agility: 70
HP: 49 - 65
SLEEP resist: 15 / 16
STOPSPELL resist: 7 / 16
HURT resist: 15 / 16
Dodge: 2 / 64
XP: 50
GP: 123 - 164
Pattern: 50 % to cast HURTMORE
Otherwise attack

#33 : Axe Knight

Strength: 94
Agility: 82
HP: 53 - 70
SLEEP resist: 15 / 16
STOPSPELL resist: 3 / 16
HURT resist: 1 / 16
Dodge: 1 / 64
XP: 54
GP: 123 - 164

Pattern: 25 % to cast SLEEP, but only if you're not asleep
Otherwise attack

#34 : Blue Dragon

Strength: 98
Agility: 84
HP: 53 - 70
SLEEP resist: 15 / 16
STOPSPELL resist: 15 / 16
HURT resist: 7 / 16
Dodge: 2 / 64
XP: 60
GP: 112 - 149
Pattern: 25 % to breathe fire
Otherwise attack

#35 : Stoneman

Strength: 100
Agility: 40
HP: 121 - 160
SLEEP resist: 2 / 16
STOPSPELL resist: 15 / 16
HURT resist: 7 / 16
Dodge: 1 / 64
XP: 65
GP: 105 - 139
Pattern: Attack only

#36 : Armored Knight

Strength: 105
Agility: 86
HP: 68 - 90
SLEEP resist: 15 / 16
STOPSPELL resist: 7 / 16
HURT resist: 1 / 16
Dodge: 2 / 64
XP: 70
GP: 105 - 139
Pattern: 75 % to cast HEALMORE, but only if its HP < 1/4 of its max HP
Otherwise 25 % to cast HURTMORE
Otherwise attack

#37 : Red Dragon

Strength: 120
Agility: 90
HP: 76 - 100
SLEEP resist: 15 / 16
STOPSPELL resist: 7 / 16
HURT resist: 15 / 16
Dodge: 2 / 64
XP: 100
GP: 105 - 139
Pattern: 25 % to cast SLEEP, but only if you're not asleep
Otherwise 25 % to breathe fire
Otherwise attack

#38 : Dragonlord first form

Strength: 90
Agility: 75
HP: 76 - 100
SLEEP resist: 15 / 16
STOPSPELL resist: 15 / 16
HURT resist: 15 / 16
Dodge: 0 / 64
XP: 0
GP: 0
Pattern: 25 % to cast STOPSPELL, but only if your spell isn't blocked
Otherwise 75 % to cast HURTMORE
Otherwise attack

#39 : Dragonlord second form

Strength: 140
Agility: 200
HP: 130
SLEEP resist: 15 / 16
STOPSPELL resist: 15 / 16
HURT resist: 15 / 16
Dodge: 0 / 64
XP: 0
GP: 0
Pattern: 50 % to breathe a very strong fire
Otherwise attack

--Enemy attack formulas--

~Attacks~

There are two formulas for attack damage for enemies. The standard range is from:

$(\text{EnemyStrength} - \text{HeroDefense} / 2) / 4,$

to:

$(\text{EnemyStrength} - \text{HeroDefense} / 2) / 2$

The hero's defense is equal to his agility / 2 rounded down, plus the modifiers for his equipment.

The other type of attack happens if your defense power is greater than or equal to the enemy's strength. In that case, the range is from:

0

to:

$(\text{enemyStrength} + 4) / 6$

~Spells~

NOTE: Enemy HURT and HEAL classes of spells have different ranges than the

hero's spells.

HURT does 3 - 10 damage
HURTMORE does 30 - 45 damage

Both Magic Armor and Erdrick's Armor will reduce HURT spells by 1/3. So:

HURT does 2 - 6 damage vs. Erdrick's and Magic Armor
HURTMORE does 20 - 30 damage vs. Erdrick's and Magic Armor

SLEEP always puts you to sleep, and there is no resisting it. Waking up from sleep is a 50/50 chance, but because of the limitations of the game programming it's not possible for it to last for more than 6 turns.

STOPSPELL has a 50/50 chance of working against you. If you have Erdrick's Armor it has no chance of working.

HEAL recovers 20 - 27 HP
HEALMORE recovers 85 - 100 HP

~Fire breath~

There are two types of fire breath. Only the Dragonlord's second form has the stronger breath. The rest of the enemies that have breath attacks only have the weaker type of breath.

Weak breath does 16 - 23 damage
Strong breath does 65 - 72 damage

The only thing that protects against fire breath is Erdrick's Armor, which reduces the damage by 1/3. So:

Weak breath vs. Erdrick's Armor does 10 - 14 damage
Strong breath vs. Erdrick's Armor does 42 - 48 damage

--Hero attack formulas--

~Attacks~

All enemies except the Golem and the Dragonlord have the capability of dodging any attack, including excellent moves, as is listed in the enemy chart above.

Just like the standard enemy attack, the range for the hero's standard attack damage is from:

$(\text{HeroAttack} - \text{EnemyAgility} / 2) / 4,$

to:

$(\text{HeroAttack} - \text{EnemyAgility} / 2) / 2$

If the damage done is less than 1, then there is 50/50 chance that you will do either no damage, or 1 damage.

Excellent moves have a 1/32 chance of happening per attack, and the formulas completely ignore the defense of the enemy. They range from:

Hero attack / 2

to:

Hero attack

Unfortunately, excellent moves are not possible against either form of the Dragonlord. They can also be dodged.

~Spells~

Hero spells have different ranges than enemy spells.

HURT does 5 - 12 damage

HURTMORE does 58 - 65 damage

SLEEP effectiveness depends on the enemy's SLEEP resist stat. You're always guaranteed one turn of attack, and after that turn it is a 1/3 chance that the enemy will wake up.

STOPSPELL effectiveness depends on the enemy's STOPSPELL resist stat. It never loses its effect.

--Running--

Enemy: If your strength is two times the enemy's strength or more, there is a 25% chance the enemy will run when it's given a turn. The enemy also makes this check right upon encounter, so he may get a chance to run right away regardless of initiative.

Hero: Certain groups of monsters have higher chances of blocking your running. The more difficult an enemy is, it may belong to a group that is better at blocking.

Group 1: #00 (Slime) - #19 (Druinlord)

Group 2: #20 (Drollmagi) - #29 (Werewolf)

Group 3: #30 (Green Dragon) - #34 (Blue Dragon)

Group 4: #35 (Stoneman) - #39 (Dragonlord second form)

So for instance, group 4 will block you much easier than group 1 will. On top of that is a test of agility. The test goes like this:

If $\text{HeroAgility} * \text{Random \#} < \text{EnemyAgility} * \text{Random \#} * \text{GroupFactor}$, then the enemy will block you.

Random # is a random number between 0 and 255.

GroupFactor depends on the group:

Group 1: 0.25

Group 2: 0.375

Group 3: 0.5

Group 4: 1.0

So you can see that enemies in group 1 are going to have a much lower product than enemies in group 4, meaning that it will be more likely that your product will be higher and that you will get away.

And yes, it is possible to run from the Dragonlord, although you will repeat the same dialog and have to fight the first form again. You can also run from the Axe Knight in Hauksness, the Golem outside Cantlin, and the Green Dragon in the Marsh Cave, but it will put you back a spot so that you can't just skip them and get their prize.

Also worth noting is that if the monster is asleep, you can run away every time.

--Initiative--

This is the same equation as running from enemies in Group 1, except instead of being blocked, the enemy will go first.

C. Chests

--Shrine west of Kol--

Staff of Rain

--Charlock--

B2:

Erdrick's Sword

B7:

Two Herbs, 500 - 755 gold, Magic Key, Wings, Cursed Belt

--Garinham--

Back room:

10 - 17 gold, Torch, Herb

--Grave of Garinham--

B1:

6-13 gold, 5-20 gold, Herb

B3:

Cursed Belt, Silver Harp

--Erdrick's Cave--

B2: Erdrick's Tablet

--Mountain Cave--

B1:

Herb

B2:

Fighter's Ring, Torch, 10 - 17 gold, 100 - 131 gold (this chest can also be the Death Necklace, which is a 1/32 chance, and can only be obtained once)

--Rimuldar--

Wings

--Tantegel--

Throne Room:

120 gold, Torch, Magic Key

Treasury:

All chests are 6 - 13 gold

Basement:

Stones of Sunlight

--The treasure chest bug--

The handling of treasure chests in Dragon Warrior was done in a very strange way. Whenever you take a chest, it doesn't simply check to see if you are standing on a chest, but rather it checks what your position is on the map and it checks if you've already taken a chest at that position. These positions are loaded into a collection of numbers. Whenever you leave a town or cave, this collection is wiped clean, which in part is why chests reappear when you re-enter. The only time this collection is not cleared is when you die because it was an oversight on the part of the programmers, which was later corrected in sequels to the game. Because the data is not cleared, it thinks you still picked up those specific chests at those specific coordinates.

Another thing is that whenever you enter the throne room in Tantegel, the original three chests that were in that room are placed into this collection to make certain you don't get them again. So if you die in one cave after having collected two chests, that makes the total chests it thinks you've collected as five. Because the maximum number of chest positions the game allotted is eight, that makes it so that when we pick up the fourth chest in the treasury, that it would be the ninth chest. Because the collection is full, it can't store that ninth chest's position, and therefore that position will never be on that list when you hit take. So if the game can't find that position on the list, the chest never disappears.

The bug allows you to pull as much gold as you want, which effectively gives you lots of gold really fast, and allows you to buy some of the best equipment very early in the game.

D. Name choice and stats

The name you choose to give your hero will affect your stats and stat growth, so it becomes very important. Only the first four letters of the name are used in this calculation. Each letter has a corresponding numerical value, and they are added up to find the result.

This is a simplification of those corresponding numbers:

```

          a b c d e f
g h i j k l m n o p q r s t u v
w x y z A B C D E F G H I J K L
M N O P Q R S T U V W X Y Z
' . , - ? ! ) (

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

```

Take the values of the first four letters and add them up. This is our sum. If you divide that sum by 4, and take the remainder, that is your growth type. There are 4 different growth types, with each one focusing on two particular stats that will be reduced in the long term, but will be higher in the short term.

Type 0: Long term HP and MP, short term Strength and Agility

Type 1: Long term Strength and HP, short term Agility and MP

Type 2: Long term Agility and MP, short term Strength and HP

Type 3: Long term Strength and Agility, short term HP and MP

So below is a chart with all the stat numbers at each level. If you have long term growth in that category, then the number is unchanged. If you have short term growth in that category, then there is some math to get the adjusted number. Take the sum from above and divide by 4 again, this time rounding down. Take that number, divide by 4, and take the remainder. Take this number and add it to 9/10 of the number on the chart to get your final value.

Level	Strength	Agility	HP	MP
1	4	4	15	0
2	5	4	22	0
3	7	6	24	5
4	7	8	31	16
5	12	10	35	20
6	16	10	38	24
7	18	17	40	26
8	22	20	46	29
9	30	22	50	36
10	35	31	54	40
11	40	35	62	50
12	48	40	63	58
13	52	48	70	64
14	60	55	78	70
15	68	64	86	72
16	72	70	92	95
17	72	78	100	100
18	85	84	115	108
19	87	86	130	115
20	92	88	138	128
21	95	90	149	135
22	97	90	158	146
23	99	94	165	153
24	103	98	170	161
25	113	100	174	161
26	117	105	180	168
27	125	107	189	175
28	130	115	195	180
29	135	120	200	190
30	140	130	210	200

So as an example, I'll use the name Erdrick at level 20 and do the math.

So I'll look up the values for the first four letters:

E: 8
r: 11
d: 13
r: 11

My sum is then $8 + 11 + 13 + 11 = 43$

My growth type is the remainder of $43 / 4$, which is 3. Growth type 3 is long term Strength and Agility, so I can automatically pick those numbers off of the chart. My strength is 92 and my agility is 88.

So now I need to find my short term stats. Again I take $43 / 4$, but I round it down this time, giving me 10. I will take that result, divide by 4, and take the remainder, which is 2. So I will take that number and add it to 9/10 of what the base stat is. For HP it is 138, so $138 * 9/10 + 2$ will give me the result, which is 126. Doing the same process for MP I will get 117.

So my stats end up as:

Strength: 92
Agility: 88
HP: 126
MP: 117

* III. Precise Formulas and Code *

These are the advanced formulas that will reference assembly language straight from the game. Before reading onward, there are several algorithms you must be aware of.

A. Common algorithms

First and foremost is the random generator subroutine, which is located at \$C55B. It generates a random number between 0 and 255, which for future reference I will refer to as RAND. This number is stored at \$95. Here's how it works:

The code		What it does
-----		-----
\$C55B:A5 95	LDA \$0095	Load up the previous random number.
\$C55D:85 3D	STA \$003D	Put it into \$3D.
\$C55F:A5 94	LDA \$0094	Load up the random number's lower byte.
\$C561:85 3C	STA \$003C	Put it into \$3C.
\$C563:06 94	ASL \$0094	
\$C565:26 95	ROL \$0095	Multiply \$9495 by 2.

\$94 is the lower byte and \$95 is the higher byte. Together they are used as

one larger number. This is apparent by doing the ASL on \$94, and carrying the overflow into \$95. These byte pairs will be referred to as \$9495 and \$3C3D.

The code		What it does
\$C567:18	CLC	
\$C568:65 94	ADC \$0094	Add \$3C3D to the new \$9495.
\$C56A:85 94	STA \$0094	
\$C56C:A5 95	LDA \$0095	
\$C56E:65 3D	ADC \$003D	
\$C570:85 95	STA \$0095	
\$C572:A5 94	LDA \$0094	Add \$94 to \$95.
\$C574:18	CLC	
\$C575:65 95	ADC \$0095	
\$C577:85 95	STA \$0095	
\$C579:A5 94	LDA \$0094	Add 0x81 to \$9495.
\$C57B:18	CLC	
\$C57C:69 81	ADC #\$81	
\$C57E:85 94	STA \$0094	
\$C580:A5 95	LDA \$0095	
\$C582:69 00	ADC #\$00	
\$C584:85 95	STA \$0095	
\$C586:60	RTS	

So to simplify a little bit, we have:

```
$9495 = $9495 * 2 + $9495
$95 = $94 + $95
$9495 = $9495 + 0x81
```

We can simplify this a little bit more. The first line could be summed up as $\$9495 * 3$. For our purposes, the second line can be expressed as $\$9495 * 257$ since the overflow is inconsequential. So then we have:

```
$9495 *= 3
$9495 *= 257
$9495 += 0x81
```

This can then be combined into one line as:

```
$9495 = $9495 * 771 + 0x81
```

Or:

```
$9495 = $9495 * 0x303 + 0x81
```

This formula presents some limitations. The modulo of the low byte (\$94) by 4 can only ever equal 0 or 1 (bit 1 can never be 1). This means that \$94 only represents 128 values, where \$95 represents 256. This makes 32,768 possibilities.

The biggest limitation is when it is run consecutively and the consecutive values are used. While the distribution is even, running it consecutively will result in patterns cropping up. In the case of the hero waking up from SLEEP, which is a 50/50 chance, after a few turns it is impossible for him to stay asleep. A true random generator could go on for a long time, but this one is fairly limited. It does work quite well when waiting for input though. When the player is idle, the generator runs once a frame, which almost guarantees that it will be random.

The next algorithm to know is multiplication, which is located at \$C1C9.

The code		What it does
-----		-----
\$C1C9:A9 00	LDA #00	\$40 and \$41 form a double byte, and they represent
\$C1CB:85 40	STA \$0040	the answer to the multiplication. Here they are
\$C1CD:85 41	STA \$0041	zeroed.
\$C1CF:A5 3C	LDA \$003C	\$3C and \$3D also form a double byte, and they are
\$C1D1:05 3D	ORA \$003D	one of the factors. Here it checks if \$3C3D is 0
\$C1D3:F0 1A	BEQ \$C1EF	and it ends the subroutine if so.
\$C1D5:46 3D	LSR \$003D	
\$C1D7:66 3C	ROR \$003C	\$3C3D is divided by 2.
\$C1D9:90 0D	BCC \$C1E8	If the remainder is 0, it branches to \$C1E8.
\$C1DB:A5 3E	LDA \$003E	
\$C1DD:18	CLC	
\$C1DE:65 40	ADC \$0040	
\$C1E0:85 40	STA \$0040	\$3E and \$3F are the other double byte factor.
\$C1E2:A5 3F	LDA \$003F	Here they are added to \$4041 (the result).
\$C1E4:65 41	ADC \$0041	
\$C1E6:85 41	STA \$0041	
\$C1E8:06 3E	ASL \$003E	
\$C1EA:26 3F	ROL \$003F	\$3E3F is multiplied by 2.
\$C1EC:4C CF C1	JMP \$C1CF	The process is repeated until \$3C == \$3D
\$C1EF:60	RTS	

There are other FAQs that deal with the particulars of how this works, and it's rather lengthy. Suffice it to say that all that you need to know is that $\$3C3D * \$3E3F = \$4041$.

The last algorithm to know is division, which is located at \$C1F0.

The code		What it does
-----		-----
\$C1F0:A9 00	LDA #00	
\$C1F2:85 3D	STA \$003D	\$3D, part of a double byte \$3C3D, is set to 0.
\$C1F4:A0 10	LDY #10	Y = 0x10, which is setting up 16 loops.
\$C1F6:A9 00	LDA #00	
\$C1F8:06 3C	ASL \$003C	
\$C1FA:26 3D	ROL \$003D	
\$C1FC:85 40	STA \$0040	\$40 is set to \$3C3D * 2.
\$C1FE:65 40	ADC \$0040	The accumulator is set to \$40 (plus 1 if there is overflow).
\$C200:E6 3C	INC \$003C	\$3C++
\$C202:38	SEC	
\$C203:E5 3E	SBC \$003E	\$3E is subtracted from the accumulator.
\$C205:B0 05	BCS \$C20C	If there is no overflow, it branches to \$C20C.
\$C207:18	CLC	
\$C208:65 3E	ADC \$003E	Else \$3E is added to the accumulator.
\$C20A:C6 3C	DEC \$003C	\$3C--
\$C20C:88	DEY	Y--
\$C20D:D0 E9	BNE \$C1F8	If Y != 0, then repeat.
\$C20F:85 40	STA \$0040	Store the accumulator in \$40.
\$C211:60	RTS	

Like multiplication, the explanation for how this works is lengthy, so suffice

it to say that all you need to know is that \$3C3D / \$3E = \$3C remainder \$40.

Now that we know the basics of the algorithms, we can progress to the formulas.

B. Journeying

-- Items and Spells --

For all healing items and spells, the code is basically the same.

The code	What it does
-----	-----
\$DBB8:20 5B C5 JSR \$C55B	Do the random algorithm.
\$DBBB:A5 95 LDA \$0095	Load our random number and...
\$DBBD:29 07 AND #\$07	...limit it to a number between 0-7.
\$DBBF:18 CLC	
\$DBC0:69 0A ADC #\$0A	Add that number to 0x0A.
\$DBC2:18 CLC	
\$DBC3:65 C5 ADC \$00C5	Add that to your HP.
\$DBC5:B0 04 BCS \$DFCB	If it exceeds 255, branch to \$DFCB.
\$DBC7:C5 CA CMP \$00CA	Or if it exceeds your max HP, branch to \$DFCB.
\$DBC9:90 02 BCC \$DFCD	
\$DFCB:A5 CA LDA \$00CA	
\$DFCD:85 C5 STA \$00C5	Make the HP the max HP.

All the healing works this way. Here are the values for various spells/items:

Item	HP gained	Code location
Herb:	RAND & 0x0F + 0x17	\$DCFE
HEAL:	RAND & 0x07 + 0x0A	\$DBB8
HEALMORE:	RAND & 0x0F + 0x55	\$DBD7

Radiant spell

The code	What it does
-----	-----
\$DA6A:A9 50 LDA #\$50	
\$DA6C:85 DA STA \$00DA	Puts 0x50 into the radiant counter.

After initial casting, you have the three square radius for 0x50 steps. Every step you take does:

The code	What it does
-----	-----
\$CA5A:C6 DA DEC \$00DA	Decrease the radiant counter by 1.
\$CA5C:D0 0E BNE \$CA6C	If it's not 0, move on to other things.
\$CA5E:A5 D0 LDA \$00D0	If it is 0, check the radius (\$D0).
\$CA60:C9 01 CMP #\$01	If it's 1...
\$CA62:F0 08 BEQ \$CA6C	...Move on to other things.
\$CA64:A9 3C LDA #\$3C	Otherwise put 0x3C into \$DA.
\$CA66:85 DA STA \$00DA	
\$CA68:C6 D0 DEC \$00D0	

\$D0 is the number that signifies the radius. It is initially set at seven. Every time \$DA hits 0, \$D0 is subtracted by 2. The radius could be expressed as $(\$D0 - 1) / 2$. When it is 1, there is no more extra visible area. Torches simply set this value to 0x03 and they last until you leave the cave.

REPEL spell

The code	What it does
-----	-----
\$DA98:A9 FF LDA #\$FF	
\$DA9A:85 DB STA \$00DB	Put 0xFF into the repel counter.

After the initial casting you have the repel counter (\$DB) / 2 number of steps. The reason it is divided by 2 is because Fairy Water sets \$DB to 0xFE. They were separated in this way so that they could run the same formulas, but also to have different messages when the effect wears off. So every step subtracts 2 from the counter.

The way encounters work with the effect:

The code	What it does
-----	-----
\$CF20:A5 45 LDA \$0045	Load up the current area
\$CF22:C9 01 CMP #\$01	Check if you're on the overworld
\$CF24:D0 1E BNE \$CF44	
\$CF26:A5 DB LDA \$00DB	Load the repel counter
\$CF28:F0 1A BEQ \$CF44	If it's not 0...
\$CF2A:A5 CD LDA \$00CD	...load the hero's defense and...
\$CF2C:4A LSR	...divide that by 2...
\$CF2D:85 3E STA \$003E	...and put it into \$3E.
\$CF2F:A6 3C LDX \$003C	X = \$3C (at current, \$3C is the enemy ID)
\$CF31:BD FA F4 LDA \$F4FA,X	Load the current enemy's strength and...
\$CF34:38 SEC	
\$CF35:E5 3E SBC \$003E	...subtract \$3E (HeroDefense / 2) from it.
\$CF37:90 0A BCC \$CF43	Branch if there is overflow.
\$CF39:85 3E STA \$003E	Store (EnemyStrength - HeroDefense / 2) at \$3E.
\$CF3B:BD FA F4 LDA \$F4FA,X	Load the current enemy's strength and...
\$CF3E:4A LSR	...divide it by 2.
\$CF3F:C5 3E CMP \$003E	If $EnemyStrength / 2 > HeroDefense / 2$
\$CF41:90 01 BCC \$CF44	Branch
\$CF43:60 RTS	

There is some redundancy here. First it checks if the hero's defense is twice or more times the enemy's strength. If it's not, then it checks if the hero's defense is greater than the enemy's strength. The first check could essentially be cut, but probably wasn't for memory or speed reasons.

So the only formula worth concerning yourself with here is:

If $(HeroDefense \gg 1 > EnemyStrength \gg 1)$ then the enemy is evaded.

The bitwise right shifts make it worth noting that if your defense is say, 51, and the enemy's strength is say, 50, you won't evade them. So the formula can't be simplified to just $HeroDefense > EnemyStrength$ like I do in the

earlier section.

The Silver Harp

The code		What it does
\$DE18:A5 45	LDA \$0045	
\$DE1A:C9 01	CMP #\$01	Check to make sure you're on the overworld.
\$DE1C:D0 23	BNE \$DE41	
\$DE1E:20 5B C5	JSR \$C55B	
\$DE21:A5 95	LDA \$0095	Load up a random number.
\$DE23:29 07	AND #\$07	Limit that number to 0-7.
\$DE25:C9 05	CMP #\$05	If the number is 5...
\$DE27:F0 F5	BEQ \$DE1E	
\$DE29:C9 07	CMP #\$07	...or 7...
\$DE2B:F0 F1	BEQ \$DE1E	repeat the random process until it isn't.

This random number serves as the monster you will encounter. 5 is the Mag Drakee and 7 is the Druin. This shows that the summoning of enemies only works on the overworld.

Because of the limitations of the random generator, and the fact that it is repeated in succession if the result is 5 or 7, this makes the odds not an even 1/6 split for each remaining enemy. The actual odds are:

0 -	Slime:	5504/32768	~16.8%
1 -	Red Slime:	5472/32768	~16.7%
2 -	Drakee:	5504/32768	~16.8%
3 -	Ghost:	5440/32768	~16.6%
4 -	Magician:	5472/32768	~16.7%
6 -	Scorpion:	5376/32768	~16.4%

So there's not a whole lot of variance there. There are other areas, such as the chances of encountering an enemy from each zone set, that are a bit more varied.

-- Encounters --

For every step that you take, the random subroutine is called. First, it figures out what kind of tile you are on. Then it proceeds to check types of terrain. It starts with the swamp:

The code		What it does
\$CDC6:A5 E0	LDA \$00E0	Load the current tile.
\$CDC8:C9 06	CMP #\$06	Check if it's swamp.
\$CDCA:D0 36	BNE \$CE02	If it's not, move on.
\$CDCC:A5 BE	LDA \$00BE	If it is, load the equipment type.
\$CDCE:29 1C	AND #\$1C	
\$CDD0:C9 1C	CMP #\$1C	
\$CDD2:F0 27	BEQ \$CDFB	If you're wearing Erdrick's Armor skip the damage.
...		
\$CDDF:A5 C5	LDA \$00C5	
\$CDE1:38	SEC	
\$CDE2:E9 02	SBC #\$02	If not, subtract 2 hit points.
\$CDE4:B0 02	BCS \$CDE8	Check if HP is less than 0.

\$CDE6:A9 00	LDA #\$00	
\$CDE8:85 C5	STA \$00C5	
...		
\$CDF0:A5 C5	LDA \$00C5	
\$CDF2:D0 07	BNE \$CDFB	If you're still alive, skip ahead.
...		
\$CDFB:A9 0F	LDA #\$0F	
\$CDFD:25 95	AND \$0095	Limit the last random number to 0-15
\$CDFF:F0 7B	BEQ \$CE7C	If it's 0, move on to check the zone you're in.
\$CE01:60	RTS	

And then moves on to desert:

The code		What it does
-----		-----
\$CE02:C9 01	CMP #\$01	Check if it's desert.
\$CE04:F0 0D	BEQ \$CE13	
\$CE13:A9 07	LDA #\$07	
\$CE15:D0 E6	BNE \$CDFD	
\$CDFD:25 95	AND \$0095	Limit the random number to 0-7.
\$CDFF:F0 7B	BEQ \$CE7C	If it's 0, move on to check the zone you're in.
\$CE01:60	RTS	

And then to hills:

The code		What it does
-----		-----
\$CE06:C9 02	CMP #\$02	Check if it's hills.
\$CE08:D0 0D	BNE \$CE17	
...		
\$CE13:A9 07	LDA #\$07	
\$CE15:D0 E6	BNE \$CDFD	
\$CDFD:25 95	AND \$0095	Limit the random number to 0-7.
\$CDFF:F0 7B	BEQ \$CE7C	If it's 0, move on to check the zone you're in.
\$CE01:60	RTS	

And then forest:

The code		What it does
-----		-----
\$CE17:C9 0B	CMP #\$0B	Check if it's forest.
\$CE19:F0 44	BEQ \$CE5F	
\$CE5F:A9 0F	LDA #\$0F	
\$CE61:D0 9A	BNE \$CDFD	
\$CDFD:25 95	AND \$0095	Limit the random number to 0-15.
\$CDFF:F0 7B	BEQ \$CE7C	If it's 0, move on to check the zone you're in.
\$CE01:60	RTS	

And then brick floor:

The code		What it does
-----		-----
\$CE1B:C9 04	CMP #\$04	Check if it's brick floor.
\$CE1D:F0 40	BEQ \$CE5F	
\$CE5F:A9 0F	LDA #\$0F	
\$CE61:D0 9A	BNE \$CDFD	

```

$CDFD:25 95      AND $0095      Limit the random number to 0-15.
$CFFF:F0 7B      BEQ $CE7C      If it's 0, move on to check the zone you're in.
$CE01:60         RTS

```

And then if it's barrier:

The code	What it does	

\$CE1F:C9 0D	CMP #\$0D	Check if it's barrier.
\$CE21:D0 40	BNE \$CE63	
\$CE23:A5 BE	LDA \$00BE	If it is, load the equipment type.
\$CE25:29 1C	AND #\$1C	
\$CE27:C9 1C	CMP #\$1C	
\$CE29:F0 34	BEQ \$CE5F	If you're wearing Erdrick's Armor skip the damage.
\$CE5F:A9 0F	LDA #\$0F	
\$CE61:D0 9A	BNE \$CDFD	
\$CDFD:25 95	AND \$0095	Limit the random number to 0-15.
\$CFFF:F0 7B	BEQ \$CE7C	If it's 0, move on to check the zone you're in.
\$CE01:60	RTS	

And if it's none of the above:

The code	What it does	

\$CE63:A5 3A	LDA \$003A	Load the hero's x position.
\$CE65:4A	LSR	
\$CE66:B0 07	BCS \$CE6F	If x is odd, proceed to check if y is even.
\$CE68:A5 3B	LDA \$003B	If x is even, load the hero's y position.
\$CE6A:4A	LSR	
\$CE6B:90 07	BCC \$CE74	If y is even, the random number will be 0-31.
\$CE6D:B0 09	BCS \$CE78	If y is odd, the random number will be 0-15.
\$CE6F:A5 3B	LDA \$003B	If the x position was odd...
\$CE71:4A	LSR	
\$CE72:90 04	BCC \$CE78	..and if the y position is even, the random number
\$CE74:A9 1F	LDA #\$1F	will be 0-31. If y is odd, it will be 0-15.
\$CE76:D0 85	BNE \$CDFD	
\$CE78:A9 0F	LDA #\$0F	
\$CE7A:D0 FA	BNE \$CE76	

Basically, what this all does is set up a checker board style of encounters in places like grasslands. One type of checker is a 1 in 16 chance of encounter while the other type is a 1 in 32 chance. The point is to essentially make the encounter similar to a 1 in 24 chance.

Then, we start to check what zone we're in:

The code	What it does	

\$CE7C:A5 45	LDA \$0045	Check which area you're in.
\$CE7E:C9 01	CMP #\$01	If it's the overworld...
\$CE80:D0 56	BNE \$CED8	...don't branch.
\$CE82:A5 3B	LDA \$003B	Load the hero's y coordinate...
\$CE84:85 3C	STA \$003C	...and put it into \$3C.
\$CE86:A9 0F	LDA #\$0F	
\$CE88:85 3E	STA \$003E	Set \$3E to 0x0F.
\$CE8A:20 F0 C1	JSR \$C1F0	Run the division subroutine (\$3C / \$3E).
\$CE8D:A5 3C	LDA \$003C	Load the result of the division...

```

$CE8F:85 42    STA $0042    ...and put it into $42 for later.
$CE91:A5 3A    LDA $003A    Load the hero's X coordinate...
$CE93:85 3C    STA $003C    ...and put it into $3C.
$CE95:A9 0F    LDA #$0F
$CE97:85 3E    STA $003E    Set $3E to 0x0F.
$CE99:20 F0 C1 JSR $C1F0    Runs the division subroutine ($3C / $3E).
$CE9C:A5 42    LDA $0042    Load the hero's y grid...
$CE9E:0A      ASL
$CE9F:0A      ASL            ...and multiply it by 4.
$CEA0:85 3E    STA $003E    Put the result in $3E.
$CEA2:A5 3C    LDA $003C    Load the hero's x grid...
$CEA4:4A      LSR            ...and divide it by 2...
$CEA5:18      CLC
$CEA6:65 3E    ADC $003E    ...and add it to the previous y grid * 4.
$CEA8:AA      TAX            Transfer the result to X.
$CEA9:BD 22 F5 LDA $F522,X  Loads what zone grid you are in...
$CEAC:85 3E    STA $003E    ...and stores it in $3E.
$CEAE:A5 3C    LDA $003C    Load the hero's x grid again.
$CEB0:4A      LSR
$CEB1:B0 08    BCS $CEBB   If it's an odd number, get the small nibble of $3E
$CEB3:46 3E    LSR $003E   If it's even, get the large nibble of $3E.
$CEB5:46 3E    LSR $003E
$CEB7:46 3E    LSR $003E
$CEB9:46 3E    LSR $003E
$CEBB:A5 3E    LDA $003E
$CEBD:29 0F    AND #$0F
$CEBF:D0 43    BNE $CF04   If the zone isn't 0, go on to combat.

```

So what this does is divides your x and y coordinates by 15 to figure out what grid you are currently in. It then takes that grid and converts it into a sequential number to figure out what enemy zone this grid represents.

So for instance, let's say your coordinates are 0x2B, 0x2C (one square south of Tantegel). The x grid is 0x2B / 0x0F, which is 2 (the remainder is unimportant). The y grid is 0x2C / 0x0F, which is also 2. So now we follow the equation:

```
yGrid * 4 + (xGrid >> 1)
```

And we would have 2 * 4 + (2 >> 1), which is 9. This result is used as the offset from \$F522, which in this case would be \$F52B, which is 0x00. Normally we would find out which nibble is our zone (in this case it would be the large nibble), but it doesn't matter since they are both 0. Since the zone is 0, we go on to this next code:

The code	What it does
-----	-----
\$CEC1:20 5B C5 JSR \$C55B	If the zone is 0, run the random subroutine.
\$CEC4:A5 E0 LDA \$00E0	Load the current tile.
\$CEC6:C9 02 CMP #\$02	If it's hills...
\$CEC8:D0 07 BNE \$CED1	
\$CECA:A5 95 LDA \$0095	...load the random number and...
\$CECC:29 03 AND #\$03	...limit it to 0-3
\$CECE:F0 34 BEQ \$CF04	If it's 0, go on to picking the enemy.

Essentially what this does is that while you're in zone 0, hill encounters are reduced from 1/8 to 1/32. If it's not hills, the limitation is:

The code	What it does
----------	--------------

```

-----
$CED1:A5 95    LDA $0095    Load the random number and...
$CED3:29 01    AND #$01     ...limit it to 0 or 1.
$CED5:F0 2D    BEQ $CF04    If it's 0, go on to picking the enemy.

```

So every other type of terrain is limited by a factor of 2. That means you're much less likely to have encounters near Tantegel.

Now on to picking the enemy.

-- Monster Sets --

The code		What it does
\$CF04:85 3E	STA \$003E	Store the zone number at \$3E.
\$CF06:0A	ASL	
\$CF07:0A	ASL	
\$CF08:18	CLC	
\$CF09:65 3E	ADC \$003E	Multiply the zone number by 5.
\$CF0B:85 3E	STA \$003E	Put the result in \$3E.
\$CF0D:20 5B C5	JSR \$C55B	Run the random subroutine.
\$CF10:A5 95	LDA \$0095	
\$CF12:29 07	AND #\$07	Limit the random number to 0-7.
\$CF14:C9 05	CMP #\$05	If it's 5-7...
\$CF16:B0 F5	BCS \$CF0D	...go back and repeat until it's not.
\$CF18:65 3E	ADC \$003E	Add the random number to \$3E.
\$CF1A:AA	TAX	
\$CF1B:BD 4F F5	LDA \$F54F,X	Use the result as the offset from \$F54F to get the enemy ID.
\$CF1E:85 3C	STA \$003C	Store the ID at \$3C.

So since there are five possibilities per zone, this figures out the zone * 5 + a random number between 0 and 4. From there, it checks the offset at \$F54F to see what enemy ID is there. The chart at \$F54F is as follows:

```

00 01 00 01 00
01 00 01 02 01
00 03 02 03 01
01 01 02 03 04
03 04 05 05 06
03 04 05 06 0B
05 06 0B 0C 0E
0B 0C 0D 0E 0E
0D 0F 12 12 19
0F 15 12 15 19
15 16 17 1A 1C
17 1A 1B 1C 10
1A 1B 1C 1D 1F
1D 1E 1F 1F 20
08 09 0A 0B 0C
11 12 13 14 17
1D 1E 1F 20 21
20 21 22 22 23
20 23 24 24 25
03 04 06 07 07

```

Each enemy ID corresponds to this chart:

Zone 0: Slime, Red Slime, Slime, Red Slime, Slime
 Zone 1: Red Slime, Slime, Red Slime, Drakee, Red Slime
 Zone 2: Slime, Ghost, Drakee, Ghost, Red Slime
 Zone 3: Red Slime, Red Slime, Drakee, Ghost, Magician
 Zone 4: Ghost, Magician, Magidrakee, Magidrakee, Scorpion
 Zone 5: Ghost, Magician, Magidrakee, Scorpion, Skeleton
 Zone 6: Magidrakee, Scorpion, Skeleton, Warlock, Wolf
 Zone 7: Skeleton, Warlock, Metal Scorpion, Wolf, Wolf
 Zone 8: Metal Scorpion, Wraith, Wolflord, Wolflord, Goldman
 Zone 9: Wraith, Wyvern, Wolflord, Wyvern, Goldman
 Zone 10: Wyvern, Rogue Scorpion, Wraith Knight, Knight, Demon Knight
 Zone 11: Wraith Knight, Knight, Magiwyvern, Demon Knight, Metal Slime
 Zone 12: Knight, Magiwyvern, Demon Knight, Werewolf, Starwyvern
 Zone 13: Werewolf, Green Dragon, Starwyvern, Starwyvern, Wizard
 Zone 14: Poltergeist, Droll, Drakeema, Skeleton, Warlock
 Zone 15: Specter, Wolflord, Druinlord, Drollmagi, Wraith Knight
 Zone 16: Werewolf, Green Dragon, Starwyvern, Wizard, Axe Knight
 Zone 17: Wizard, Axe Knight, Blue Dragon, Blue Dragon, Stoneman
 Zone 18: Wizard, Stoneman, Armored Knight, Armored Knight, Red Dragon
 Zone 19: Ghost, Magician, Scorpion, Druin, Druin

Each enemy has a roughly equal chance to be encountered, although because of the limitations of the random generator, they are slightly imbalanced. It's called once to determine if there is combat, once again if you're in zone 0, and then however many times it takes to get a number between 0 and 4 instead of 0 and 7 (there is a maximum of 7). It's mostly because of that last part that the odds are uneven.

Enemy #	Chance to encounter
0	416/2048 ~20.3%
1	448/2048 ~21.9%
2	448/2048 ~21.9%
3	336/2048 ~16.4%
4	400/2048 ~19.5%

Since there is one extra call in zone 0, the odds there are different:

Enemy #	Chance to encounter
0	368/2048 ~18.0%
1	384/2048 ~18.8%
2	448/2048 ~21.9%
3	416/2048 ~20.3%
4	432/2048 ~21.1%

 C. Combat

--Enemies--

These are all the stats of the enemies.

ID	Enemy Name	Str	Agi	HP	Pat	SR	DR	XP	GP
00	Slime	05	03	03	00	0F	01	01	02
01	Red Slime	07	03	04	00	0F	01	01	03

02	Drakee	09	06	06	00	0F	01	02	03
03	Ghost	0B	08	07	00	0F	04	03	05
04	Magician	0B	0C	0D	02	00	01	04	0C
05	Magidrakee	0E	0E	0F	02	00	01	05	0C
06	Scorpion	12	10	14	00	0F	01	06	10
07	Druin	14	12	16	00	0F	02	07	10
08	Poltergeist	12	14	17	03	00	06	08	12
09	Droll	18	18	19	00	0E	02	0A	19
0A	Drakeema	16	1A	14	92	20	06	0B	14
0B	Skeleton	1C	16	1E	00	0F	04	0B	1E
0C	Warlock	1C	16	1E	12	31	02	0D	23
0D	Metal Scorpion	24	2A	16	00	0F	02	0E	28
0E	Wolf	28	1E	22	00	1F	02	10	32
0F	Wraith	2C	22	24	90	70	04	11	3C
10	Metal Slime	0A	FF	04	03	FF	F1	73	06
11	Specter	28	26	24	13	31	04	12	46
12	Wolflord	32	24	26	60	47	02	14	50
13	Druinlord	2F	28	23	B1	F0	04	14	55
14	Drollmagi	34	32	26	60	22	01	16	5A
15	Wyvern	38	30	2A	00	4F	02	18	64
16	Rogue Scorpion	3C	5A	23	00	7F	02	1A	6E
17	Wraith Knight	44	38	2E	B0	50	34	1C	78
18	Golem	78	3C	46	00	FF	F0	05	0A
19	Goldman	30	28	32	00	DF	01	06	C8
1A	Knight	4C	4E	37	60	67	01	21	82
1B	Magiwyvern	4E	44	3A	20	20	02	22	8C
1C	Demon Knight	4F	40	32	00	FF	FF	25	96
1D	Werewolf	56	46	3C	00	7F	07	28	9B
1E	Green Dragon	58	4A	41	09	7F	22	2D	A0
1F	Starwyvern	56	50	41	F9	80	12	2B	A0
20	Wizard	50	46	41	06	F7	F2	32	A5
21	Axe Knight	5E	52	46	10	F3	11	36	A5
22	Blue Dragon	62	54	46	09	FF	72	3C	96
23	Stoneman	64	28	A0	00	2F	71	41	8C
24	Armored Knight	69	56	5A	F5	F7	12	46	8C
25	Red Dragon	78	5A	64	19	F7	F2	64	8C
26	Dragonlord 1	5A	4B	64	57	FF	F0	00	00
27	Dragonlord 2	8C	C8	82	0E	FF	F0	00	00

"Str" is the enemy's strength or attack power.

"Agi" is the enemy's agility.

"HP" is the enemy's hit points. This number is just the maximum that their HP can be, but it is variable. The code to calculate it is:

The code

What it does

```

-----
$E599:AD 02 01 LDA $0102      Load the enemy's max HP...
$E59C:85 3E   STA $003E      ...and put it in $3E.
$E59E:20 5B C5 JSR $C55B
$E5A1:A5 95   LDA $0095

```

```

$E5A3:85 3C      STA $003C      Put the random number into $3C.
$E5A5:A9 00      LDA #$00
$E5A7:85 3D      STA $003D
$E5A9:85 3F      STA $003F
$E5AB:20 C9 C1   JSR $C1C9      Multiply the random number by the max HP.
$E5AE:A5 41      LDA $0041      Load the result / 256.
$E5B0:4A         LSR
$E5B1:4A         LSR           Divide it by 4.
$E5B2:85 40      STA $0040      Store the result at $40.
$E5B4:AD 02 01   LDA $0102      Load the enemy's max HP.
$E5B7:38         SEC
$E5B8:E5 40      SBC $0040      Subtract $40 from the max HP.
$E5BA:85 E2      STA $00E2      Set the enemy's current HP to that result.

```

The result is:

```
MaxHP - ((RAND * MaxHP) >> 10)
```

So that will be a range from 75.1-100% of the value. The only exception to this is the Dragonlord's second form. He always has the full value for HP.

"Pat" is the enemy's attack pattern. See below for the more advanced info on how this number works.

"SR" is status resistance. The first nibble / 16 is the chance to resist SLEEP:

The code	What it does
-----	-----
\$E770:AD 04 01 LDA \$0104	Load the enemy's status resistance.
\$E773:4A LSR	
\$E774:4A LSR	
\$E775:4A LSR	
\$E776:4A LSR	Get the first nibble (SLEEP resistance).
\$E777:20 46 E9 JSR \$E946	
\$E946:85 3E STA \$003E	Store it at \$3E.
\$E948:20 5B C5 JSR \$C55B	
\$E94B:A5 95 LDA \$0095	Get the random number...
\$E94D:29 0F AND #\$0F	...and limit it to 0-15.
\$E94F:C5 3E CMP \$003E	If that number is greater or equal to the enemy's
\$E951:90 01 BCC \$E954	SLEEP resistance, then it is successful.
\$E953:60 RTS	

The second nibble / 16 is the chance to resist STOPSPELL:

The code	What it does
-----	-----
\$E78A:AD 04 01 LDA \$0104	Load the enemy's status resistance.
\$E78D:29 0F AND #\$0F	Get the second nibble (STOPSPELL resistance).
\$E78F:20 46 E9 JSR \$E946	
\$E946:85 3E STA \$003E	Store it at \$3E.
\$E948:20 5B C5 JSR \$C55B	
\$E94B:A5 95 LDA \$0095	Get the random number...
\$E94D:29 0F AND #\$0F	...and limit it to 0-15.
\$E94F:C5 3E CMP \$003E	If that number is greater or equal to the enemy's
\$E951:90 01 BCC \$E954	STOPSPELL resistance, then it is successful.

\$E953:60 RTS

So for instance, a Slime's SR is 0F. That means SLEEP will always work and that STOPSPELL has a 15/16 chance of failing. Enemies who don't use magic have F as their STOPSPELL resistance, but this still allows a small chance of it working even though it will have no real effect.

"DR" is damage resistance. The first nibble / 16 is the chance that HURT and HURTMORE will have no effect:

The code		What it does
-----		-----
\$E73A:AD 05 01	LDA \$0105	Load the enemy's damage resistance.
\$E73D:4A	LSR	
\$E73E:4A	LSR	
\$E73F:4A	LSR	
\$E740:4A	LSR	Get the first nibble (HURT resistance).
\$E741:20 46 E9	JSR \$E946	
\$E946:85 3E	STA \$003E	Store it at \$3E.
\$E948:20 5B C5	JSR \$C55B	
\$E94B:A5 95	LDA \$0095	Get the random number...
\$E94D:29 0F	AND #\$0F	...and limit it to 0-15.
\$E94F:C5 3E	CMP \$003E	If that number is greater or equal to the enemy's
\$E951:90 01	BCC \$E954	HURT resistance, then it is successful.
\$E953:60	RTS	

The second nibble / 64 is the chance of dodging:

The code		What it does
-----		-----
\$E66E:20 5B C5	JSR \$C55B	
\$E671:A5 95	LDA \$0095	Get a random number and...
\$E673:29 3F	AND #\$3F	...limit it to 0-63.
\$E675:85 95	STA \$0095	Store the number at \$95.
\$E677:AD 05 01	LDA \$0105	Load the enemy's damage resistance.
\$E67A:29 0F	AND #\$0F	Get the second nibble (dodging chance).
\$E67C:F0 1C	BEQ \$E69A	If it is 0, the enemy dodges.
\$E67E:38	SEC	
\$E67F:E9 01	SBC #\$01	Subtract 1 from the dodging chance.
\$E681:C5 95	CMP \$0095	If that number is greater or equal to the previous
\$E683:90 15	BCC \$E69A	random number (0-63), the enemy dodges.

So for instance, a Slime's DR is 01. That means that HURT spells will always connect and that it has a 1/64 chance to dodge an attack.

"XP" is the experience you will get.

"GP" is the maximum gold you can get plus one. The code for gold received is:

The code		What it does
-----		-----
\$EA2A:AD 07 01	LDA \$0107	Load the enemy's gold value.
\$EA2D:85 3E	STA \$003E	Store it at \$3E.
\$EA2F:20 5B C5	JSR \$C55B	


```

$EA32:A5 95    LDA $0095    Get a random number and...
$EA34:29 3F    AND #$3F      ...limit it to 0-63.
$EA36:18      CLC
$EA37:69 C0    ADC #$C0      Add that number to 0xC0.
$EA39:85 3C    STA $003C     Store the result at $3C.
$EA3B:A9 00    LDA #$00
$EA3D:85 3D    STA $003D
$EA3F:85 3F    STA $003F
$EA41:20 C9 C1 JSR $C1C9     Multiply the enemy's gold value by $3C.
$EA44:A5 41    LDA $0041     Load the result / 256.
$EA46:85 00    STA $0000     And store it at $00

```

That result is later added to your gold total. So the formula is:

```
(GP * ((RAND & 0x3F) + 0xC0)) >> 8
```

So that will be a range from 75-99.9% of the value. You will never receive that full number.

--Enemy patterns--

The above mentioned enemy pattern is a little bit more elaborate than the other stats. An enemy can have up to two special moves. One focuses on status spells and healing, and the other focuses on damage spells and fire breath. Certain checks are made to see if they use these attacks. If they don't, they simply attack.

Special move 1:

The code	What it does
-----	-----
\$EB4B:20 5B C5 JSR \$C55B	Run the random subroutine for later.
\$EB4E:AD 03 01 LDA \$0103	Load the enemy's pattern.
\$EB51:29 30 AND #\$30	Get bits 4 and 5, which is the 1/4 chance of carrying out special move 1.
\$EB53:85 3C STA \$003C	Store it at \$3C.
\$EB55:A5 95 LDA \$0095	Load the previous random number.
\$EB57:29 30 AND #\$30	Get bits 4 and 5.
\$EB59:C5 3C CMP \$003C	If the random number is less than the enemy's pattern, they will carry out special move 1.
\$EB5B:B0 37 BCS \$EB94	Otherwise move on to special move 2.
\$EB5D:AD 03 01 LDA \$0103	Load the enemy's pattern.
\$EB60:29 C0 AND #\$C0	Get bits 6 and 7, which is special move 1.
\$EB62:D0 07 BNE \$EB6B	
\$EB64:A5 DF LDA \$00DF	If special move 1 is 0, it is to cast SLEEP.
\$EB66:30 2C BMI \$EB94	If the hero is already asleep, then move on to special move 2. Otherwise cast SLEEP, which is always successful.
\$EB6B:C9 40 CMP #\$40	
\$EB6D:D0 09 BNE \$EB78	
\$EB6F:A5 DF LDA \$00DF	If special move 1 is 1, it is to cast STOPSPELL.
\$EB71:29 10 AND #\$10	If the hero's spell is already stopped, then move on to special move 2. Otherwise cast STOPSPELL, which has a 50/50 chance of success unless equipped with Erdrick's Armor, in which case there is no chance.
\$EB73:D0 1F BNE \$EB94	
\$EB78:C9 80 CMP #\$80	

\$EB7A:D0 0C	BNE \$EB88	If special move 1 is 2, it is to cast HEAL.
\$EB7C:AD 02 01	LDA \$0102	Load the enemy's max HP.
\$EB7F:4A	LSR	
\$EB80:4A	LSR	Divide that by 4.
\$EB81:C5 E2	CMP \$00E2	If the enemy's current HP is less than a quarter
\$EB83:90 0F	BCC \$EB94	of its max HP, then it will carry out the HEAL
		spell. Otherwise move on to special move 2.
		If special move 1 is 3, it is to cast HEALMORE.
\$EB88:AD 02 01	LDA \$0102	Load the enemy's max HP.
\$EB8B:4A	LSR	
\$EB8C:4A	LSR	Divide that by 4.
\$EB8D:C5 E2	CMP \$00E2	If the enemy's current HP is less than a quarter
\$EB8F:90 03	BCC \$EB94	of its max HP, then it will carry out the HEALMORE
		spell. Otherwise move on to special move 2.

Special move 2:

The code	What it does
-----	-----
\$EB94:20 5B C5 JSR \$C55B	Run the random subroutine for later.
\$EB97:AD 03 01 LDA \$0103	Load the enemy's pattern.
\$EB9A:29 03 AND #\$03	Get bits 0 and 1, which is the 1/4 chance of
	carrying out special move 2.
\$EB9C:85 3C STA \$003C	Store it at \$3C.
\$EB9E:A5 95 LDA \$0095	Load the previous random number.
\$EBA0:29 03 AND #\$03	Get bits 0 and 1.
\$EBA2:C5 3C CMP \$003C	If the random number is less than the enemy's
\$EBA4:B0 1B BCS \$EBC1	pattern, they will carry out special move 2.
	Otherwise move on to attack.
\$EBA6:AD 03 01 LDA \$0103	Load the enemy's pattern.
\$EBA9:29 0C AND #\$0C	Get bits 2 and 3, which is special move 2.
\$EBAB:D0 03 BNE \$EBB0	If special move 2 is 0, it is to cast HURT.
\$EBB0:C9 04 CMP #\$04	
\$EBB2:D0 03 BNE \$EBB7	If special move 2 is 1, it is to cast HURTMORE.
\$EBB7:C9 08 CMP #\$08	
\$EBB9:D0 03 BNE \$EBBE	If special move 2 is 2, it is to breathe fire.
\$EBBE:4C E1 EC JMP \$ECE1	If special move 2 is 3, it is to breathe strong
	fire.

As an example of how it works, we will use a Starwyvern.

The Starwyvern's pattern is F9, which is 11111001 in binary.

The first bytes it checks are 4 & 5:

```
11111001
  ^^
```

This number / 4 is the chance that they will do special move 1. In this case, it is a 3 / 4 chance that the Starwyvern will do special move 1.

Bytes 6 & 7 correspond to what special move 1 is.

```
11111001
  ^^
```

00: Cast SLEEP

01: Cast STOPSPELL

10: Cast HEAL
11: Cast HEALMORE

The Starwyvern's ability then is HEALMORE.

There are always exceptions to if these will be performed. If the hero is already asleep, the enemy will not cast SLEEP. If the hero's spell is already stopped, the enemy will not cast STOPSPELL. If the enemy's HP \geq their max HP / 4, then it will not cast healing spells.

So if the enemy doesn't end up doing special move 1, it moves onto special move 2.

Bytes 0 & 1 are checked:

```
11111001
  ^ ^
```

Again, just like our first check, this number / 4 is the chance that they will do special move 2. In the case of the Starwyvern, it's chance is 1 / 4.

Bytes 2 & 3 correspond to what special move 2 is.

```
11111001
  ^ ^
```

00: Cast HURT
01: Cast HURTMORE
10: Breath attack 1
11: Breath attack 2

The Starwyvern's ability then is breath attack 1.

If this special attack isn't used, then the enemy will just do a regular attack.

So the starwyvern's attack chances can be broken down as such:

If starwyvern HP < 16

3/4 chance to HEAL, 1/16 chance to breathe fire, 3/16 chance to attack

If starwyvern HP \geq 16

1/4 chance to breathe fire, 3/4 chance to attack

--Enemy attack formulas--

~Attacks~

There are two formulas for attack damage for enemies. This is the starting code:

The code	What it does
-----	-----
\$EBCD:AD 00 01 LDA \$0100	Load the enemy's strength.
\$EBD0:85 42 STA \$0042	Store it at \$42.
\$EBD2:A5 CD LDA \$00CD	Load the hero's defense.
\$EBD4:85 43 STA \$0043	Store it at \$43.
\$EBD6:20 F4 EF JSR \$EFF4	
\$EFF4:46 43 LSR \$0043	Divide the hero's defense by 2.

\$EFF6:A5 42	LDA \$0042	
\$EFF8:4A	LSR	Divide the enemy's strength by 2.
\$EFF9:85 3E	STA \$003E	Store the result at \$3E
\$EFFB:E6 3E	INC \$003E	Add 1 to \$3E.
\$EFFD:A5 42	LDA \$0042	
\$EFFF:38	SEC	
\$F000:E5 43	SBC \$0043	Subtract the halved hero's defense from the enemy's strength.
\$F002:90 04	BCC \$F008	Branch if it's less than 0.
\$F004:C5 3E	CMP \$003E	Otherwise compare it to \$3E, which is the enemy's strength / 2 + 1.
\$F006:B0 28	BCS \$F030	If it's greater or equal, branch. If not, proceed to do lesser damage.

So \$F008 is where lesser damage is determined. \$F030 is where regular damage is determined. Lesser damage is done if $(\text{EnemyStrength} \gg 2)$ is less than $(\text{HeroDefense} \gg 2 + 1)$.

Lesser damage:

The code	What it does
-----	-----
\$F008:20 5B C5 JSR \$C55B	
\$F00B:A5 95 LDA \$0095	
\$F00D:85 3C STA \$003C	Put a random number at \$3C.
\$F00F:A9 00 LDA #\$00	
\$F011:85 3D STA \$003D	
\$F013:85 3F STA \$003F	
\$F015:20 C9 C1 JSR \$C1C9	Multiply the random number by \$3E.
\$F018:A5 41 LDA \$0041	Load the result / 256.
\$F01A:18 CLC	
\$F01B:69 02 ADC #\$02	Add 2 to that result.
\$F01D:85 3C STA \$003C	Store the result in \$3C.
\$F01F:A9 03 LDA #\$03	
\$F021:85 3E STA \$003E	
\$F023:4C F0 C1 JMP \$C1F0	Divide \$3C by 3.

So the end result is $\text{floor}(\frac{((\text{EnemyStrength} \gg 1 + 1) * \text{RAND}) \gg 8 + 2}{3})$. That's fairly simplified, although it could also be expressed as:

$\text{floor}(\frac{((\text{EnemyStrength} + 2) * \text{RAND} + 1024) \gg 9}{3})$

Regular damage:

The code	What it does
-----	-----
\$F030:85 42 STA \$0042	Store the enemy's strength minus the halved hero's defense at \$42.
\$F032:85 3E STA \$003E	Also store it at \$3E.
\$F034:E6 3E INC \$003E	Add 1 to \$3E.
\$F036:20 5B C5 JSR \$C55B	
\$F039:A5 95 LDA \$0095	
\$F03B:85 3C STA \$003C	Put a random number at \$3C.
\$F03D:A9 00 LDA #\$00	
\$F03F:85 3D STA \$003D	
\$F041:85 3F STA \$003F	
\$F043:20 C9 C1 JSR \$C1C9	Multiply the random number by \$3E
\$F046:A5 41 LDA \$0041	Load the result / 256 and...

```

$F048:18      CLC
$F049:65 42   ADC $0042    ...add it to $42.
$F04B:6A      ROR
$F04C:4A      LSR          Divide that result by 4.
$F04D:85 3C   STA $003C    Store the result at $3C.
$F04F:60      RTS

```

So the end result is $((\text{EnemyStrength} - (\text{HeroDefense} \gg 1) + 1) * \text{RAND}) \gg 8 + \text{EnemyStrength} - (\text{HeroDefense} \gg 1)) \gg 2$. This can be simplified as:

$$((256 + \text{RAND}) * (\text{EnemyStrength} - (\text{HeroDefense} \gg 1) + 1) - 256) \gg 10$$

~Spells~

Spell	Damage done	Code location
HURT	: RAND & 0x07 + 0x03	\$EC2A
HURTMORE	: RAND & 0x0F + 0x1E	\$EC5C

Both Magic Armor and Erdrick's Armor will reduce HURT spells:

The code	What it does
-----	-----
\$EC36:A5 BE LDA \$00BE	Load the hero's equipment byte.
\$EC38:29 1C AND #\$1C	Get bits 2-4, which is the armor.
\$EC3A:C9 1C CMP #\$1C	Check if it's Erdrick's Armor.
\$EC3C:F0 04 BEQ \$EC42	If it is, branch.
\$EC3E:C9 18 CMP #\$18	Otherwise, check if it's Magic Armor.
\$EC40:D0 10 BNE \$EC52	If it is, continue.
\$EC42:A5 00 LDA \$0000	Load the HURT damage.
\$EC44:85 3C STA \$003C	Store it at \$3C.
\$EC46:A9 03 LDA #\$03	
\$EC48:85 3E STA \$003E	
\$EC4A:20 F0 C1 JSR \$C1F0	Divide the HURT damage by 3.
\$EC4D:A5 3C LDA \$003C	
\$EC4F:0A ASL	Multiply the result by 2.
\$EC50:85 00 STA \$0000	

It's divided by 3, the remainder is discarded, and that result is multiplied by 2.

SLEEP:

The code	What it does
-----	-----
\$EC99:A5 DF LDA \$00DF	Load the byte that has the hero's sleep status.
\$EC9B:09 80 ORA #\$80	Set bit 7 to 1, which puts the hero to sleep.
\$EC9D:85 DF STA \$00DF	

SLEEP always puts you to sleep, and there is no resisting it. Waking up from sleep is a 50/50 chance:

The code	What it does
-----	-----
\$E5D6:20 5B C5 JSR \$C55B	

```

$E5D9:A5 95    LDA $0095    Get a random number.
$E5DB:4A      LSR
$E5DC:B0 07    BCS $E5E5    If it's an odd number, proceed to wake up.
$E5E5:A5 DF    LDA $00DF    Load the byte that has the hero's sleep status.
$E5E7:29 7F    AND #$7F     Set bit 7 to 0, which wakes the hero up.
$E5E9:85 DF    STA $00DF

```

Because of the limitations of the random generator, it's not possible for sleep to last on the hero for more than a set number of turns. Per turn, the random generator is run 4 times:

1. Check to see if the enemy does special move 1. Since special move 1 is SLEEP, it will not do this if you are sleeping. So it automatically goes to check 2.
2. Check to see if the enemy does special move 2.
3. If it does or doesn't do that move, it will run the random generator just once to calculate damage.
4. The check to see if the hero wakes up.

The results of this are:

Turns asleep	Chance of being asleep that long
1	100.0%
2	50.0%
3	25.0%
4	12.5%
5	~9.0%
6	~3.1%
7	~0.4%
8	0.0%

So the longest you could stay asleep would be 6 turns, although it is highly unlikely.

There can actually be an extra random check added in there, and that is if the enemy feels intimidated and wants to run. This really shouldn't be of concern since if you're a lot stronger than them, it shouldn't matter if you get sleeplocked. Regardless, these are the odds:

Turns asleep	Chance of being asleep that long
1	100.0%
2	62.5%
3	~22.9%
4	~9.0%
5	~3.3%
6	~3.1%
7	~1.0%
8	~0.6%
9	~0.4%
10	~0.2%
11	~0.2%
12	0.0%

So the odds are much better in the initial rounds, but it has the capacity to last 4 turns longer. Again, it probably wouldn't matter much if you are stronger than the enemy.

STOPSPELL has a 50/50 chance of working against you unless you are equipped with Erdrick's Armor, in which case it won't work at all:

The code		What it does
\$EC70:A5 BE	LDA \$00BE	Load the hero's equipment byte.
\$EC72:29 1C	AND #\$1C	Get bits 2-4, which is the armor.
\$EC74:C9 1C	CMP #\$1C	Check if it's Erdrick's Armor.
\$EC76:F0 16	BEQ \$EC8E	If it is, STOPSPELL will not work.
\$EC78:20 5B C5	JSR \$C55B	
\$EC7B:A5 95	LDA \$0095	Otherwise get a random number.
\$EC7D:4A	LSR	
\$EC7E:90 0E	BCC \$EC8E	If it's an odd number, your spell is stopped.
\$EC80:A5 DF	LDA \$00DF	Load the byte with the hero's STOPSPELL status.
\$EC82:09 10	ORA #\$10	Set bit 4 to 1, which stops the hero's spell.
\$EC84:85 DF	STA \$00DF	

Spell	HP recovered	Code location
HEAL	: RAND & 0x07 + 0x14	\$ECAD
HEALMORE	: RAND & 0x0F + 0x55	\$ECD5

~Fire breath~

There are two types of fire breath. Only the Dragonlord's second form has the stronger breath. The rest of the enemies that have breath attacks only have the weaker type of breath.

Breath type	Damage done	Code location
Weak breath	: RAND & 0x07 + 0x10	\$ECED
Strong breath	: RAND & 0x07 + 0x41	\$ECE1

The only thing that protects against fire breath is Erdrick's Armor, which reduces the damage:

The code		What it does
\$ECFC:A5 BE	LDA \$00BE	Load the hero's equipment byte.
\$ECFE:29 1C	AND #\$1C	Get bits 2-4, which is the armor.
\$ED00:C9 1C	CMP #\$1C	Check if it's Erdrick's Armor.
\$ED02:D0 10	BNE \$ED14	If it's not, do regular damage.
\$ED04:A5 00	LDA \$0000	
\$ED06:85 3C	STA \$003C	
\$ED08:A9 03	LDA #\$03	
\$ED0A:85 3E	STA \$003E	
\$ED0C:20 F0 C1	JSR \$C1F0	Otherwise divide the fire damage by 3.
\$ED0F:A5 3C	LDA \$003C	
\$ED11:0A	ASL	Multiply the result by 2.
\$ED12:85 00	STA \$0000	

It's divided by 3, the remainder is discarded, and that result is multiplied by 2.

--Hero attack formulas--

~Attacks~

First, the initial values are loaded:

The code		What it does
\$E60C:A5	CC LDA \$00CC	Get the hero's attack power.
\$E60E:85	42 STA \$0042	Store it at \$42 for later.
\$E610:AD	01 01 LDA \$0101	Get the enemy's agility.
\$E613:85	43 STA \$0043	Store it at \$43 for later.
\$E615:A5	E0 LDA \$00E0	Get the enemy type.
\$E617:C9	26 CMP #\$26	Check if it's the Dragonlord...
\$E619:F0	36 BEQ \$E651	
\$E61B:C9	27 CMP #\$27	...in either form.
\$E61D:F0	32 BEQ \$E651	

It branches to \$E651 in either case. This is where it determines if you can get an excellent move, and they aren't possible to achieve against the Dragonlord. So if the enemy is either form of the Dragonlord, it skips the check.

Excellent move check:

The code		What it does
\$E61F:20	5B C5 JSR \$C55B	
\$E622:A5	95 LDA \$0095	Load a random number.
\$E624:29	1F AND #\$1F	Limit that number to 0-31.
\$E626:D0	29 BNE \$E651	

Just like above, it will branch to \$E651 if the random result is not 0. If it is, it will calculate the damage for a critical hit.

Excellent move damage:

The code		What it does
\$E634:20	5B C5 JSR \$C55B	
\$E637:A5	95 LDA \$0095	Load a random number.
\$E639:85	3C STA \$003C	Store it at \$3C for later.
\$E63B:A5	CC LDA \$00CC	Load the hero's attack power.
\$E63D:4A	LSR	Divide it by 2.
\$E63E:85	3E STA \$003E	Put the result into \$3E.
\$E640:A9	00 LDA #\$00	
\$E642:85	3D STA \$003D	
\$E644:85	3F STA \$003F	
\$E646:20	C9 C1 JSR \$C1C9	Multiply the random number by the hero's attack >> 1.
\$E649:A5	CC LDA \$00CC	Load the hero's attack power.
\$E64B:38	SEC	
\$E64C:E5	41 SBC \$0041	Subtract our previous product / 256 from the hero's attack power.

So the end result for damage is $\text{HeroAttack} - ((\text{RAND} * (\text{Hero attack} \gg 1)) \gg 8)$

Regular damage:

The code	What it does
-----	-----
\$E651:20 E5 EF JSR \$EFE5	
\$EFE5:46 43 LSR \$0043	Divide the enemy agility by 2 for later.
\$EFE7:A5 42 LDA \$0042	Get the hero's attack power.
\$EFE9:38 SEC	
\$EFEA:E5 43 SBC \$0043	Subtract the halved enemy agility from it.
\$EFEC:90 38 BCC \$F026	If it's less than 0, branch to calculate minimal damage.
\$EFEE:C9 02 CMP #\$02	Otherwise check if it's 0 or 1.
\$EFF0:B0 3E BCS \$F030	If not, it branches to calculate the rest of regular damage.
\$EFF2:90 32 BCC \$F026	If so, it branches to calculate minimal damage.
\$F030:85 42 STA \$0042	Store HeroAttack - (EnemyAgility >> 1) at \$42.
\$F032:85 3E STA \$003E	Store the same number at \$3E.
\$F034:E6 3E INC \$003E	Add 1 to \$3E.
\$F036:20 5B C5 JSR \$C55B	
\$F039:A5 95 LDA \$0095	Loads a random number.
\$F03B:85 3C STA \$003C	Store it at \$3C.
\$F03D:A9 00 LDA #\$00	
\$F03F:85 3D STA \$003D	
\$F041:85 3F STA \$003F	
\$F043:20 C9 C1 JSR \$C1C9	Multiply the random number by \$3E (Heroattack - (EnemyAgility >> 1)).
\$F046:A5 41 LDA \$0041	Load the result / 256.
\$F048:18 CLC	
\$F049:65 42 ADC \$0042	Add that to HeroAttack - (EnemyAgility >> 1).
\$F04B:6A ROR	
\$F04C:4A LSR	Then divide that result by 4.
\$F04D:85 3C STA \$003C	Store the result at \$3C.
\$F04F:60 RTS	

So the end result for damage is:

$((\text{HeroAttack} - (\text{EnemyAgility} \gg 1)) * \text{RAND}) \gg 8 + \text{HeroAttack} - (\text{EnemyAgility} \gg 1) \gg 2$

This can be simplified as:

$((\text{RAND} + 256) * (\text{HeroAttack} - (\text{EnemyAgility} \gg 1))) \gg 10$

For the minimal damage, it may seem like the checks are redundant, but one is for numbers less than 0, and the other is for simply 0 and 1. This is just a limitation of the programming. Minimal damage is:

Minimal damage:

The code	What it does
-----	-----
\$F026:20 5B C5 JSR \$C55B	
\$F029:A5 95 LDA \$0095	Load a random number.
\$F02B:29 01 AND #\$01	Limits that number to 0 or 1.

\$F02D:85 3C STA \$003C Stores the result in \$3C for later.
\$F02F:60 RTS

The result is 0 or 1 damage.

~Spells~

Spell	Damage done	Code location
HURT	: RAND & 0x07 + 0x05	\$E744
HURTMORE	: RAND & 0x07 + 0x3A	\$E75F

--Running--

Every turn the enemy checks to see if it will run, even before the hero gets off a turn. Here's how it decides to run:

The code	What it does
-----	-----
\$EFB7:A5 C8 LDA \$00C8	Load the hero's strength.
\$EFB9:4A LSR	Divide it by 2.
\$EFBA:CD 00 01 CMP \$0100	Compare to the enemy's strength.
\$EFBD:90 25 BCC \$EFE4	If it's less, then continue to fight.
\$EFBF:20 5B C5 JSR \$C55B	
\$EFC2:A5 95 LDA \$0095	Otherwise get a random number.
\$EFC4:29 03 AND #\$03	Limit that number to 0-3.
\$EFC6:D0 1C BNE \$EFE4	If it's 0, then the enemy will run away.

Chances of the hero running:

The code	What it does
-----	-----
\$EE91:20 5B C5 JSR \$C55B	
\$EE94:A5 E0 LDA \$00E0	Load the current enemy.
\$EE96:C9 23 CMP #\$23	Compare its ID to 0x23.
\$EE98:90 05 BCC \$EE9F	
\$EE9A:A5 95 LDA \$0095	If it's greater or equal to 0x23, use a plain
\$EE9C:4C C7 EE JMP \$EEC7	random number 0-255.
\$EE9F:C9 1E CMP #\$1E	Otherwise if it's greater or equal to 0x1E...
\$EEA1:90 07 BCC \$EEAA	
\$EEA3:A5 95 LDA \$0095	
\$EEA5:29 7F AND #\$7F	...use a random number limited to 0-127.
\$EEA7:4C C7 EE JMP \$EEC7	
\$EEAA:C9 14 CMP #\$14	Otherwise if it's greater or equal to 0x14...
\$EEAC:90 12 BCC \$EEC0	
\$EEAE:A5 95 LDA \$0095	
\$EEB0:29 3F AND #\$3F	
\$EEB2:85 3E STA \$003E	...put a random number limited to 0-63 at \$3E.
\$EEB4:20 5B C5 JSR \$C55B	
\$EEB7:A5 95 LDA \$0095	
\$EEB9:29 1F AND #\$1F	
\$EEBB:65 3E ADC \$003E	Add another random number limited to 0-31 to \$3E.
\$EEBD:4C C7 EE JMP \$EEC7	This effectively makes the random number 0-94.
\$EEC0:20 5B C5 JSR \$C55B	
\$EEC3:A5 95 LDA \$0095	If the enemyID is less than 0x14, load a random

```

$EEC5:29 3F    AND #$3F    number and limit it to 0-63.
$EEC7:85 3C    STA $003C    Store whatever the limited random number is at $3C
$EEC9:AD 01 01 LDA $0101    Load the enemy's strength.
$EECC:85 3E    STA $003E    Store it at $3E.
$EECE:A9 00    LDA #$00
$EED0:85 3D    STA $003D
$EED2:85 3F    STA $003F
$EED4:20 C9 C1 JSR $C1C9    Multiply the random number by the enemy's strength
$EED7:A5 40    LDA $0040
$EED9:85 42    STA $0042
$EEDB:A5 41    LDA $0041
$EEDD:85 43    STA $0043    Store the double byte $4041 at $4243.
$EEDF:20 5B C5 JSR $C55B
$EEE2:A5 95    LDA $0095
$EEE4:85 3C    STA $003C    Put a random number at $3C.
$EEE6:A5 C9    LDA $00C9    Load the hero's agility.
$EEE8:85 3E    STA $003E
$EEEA:A9 00    LDA #$00
$EEEC:85 3D    STA $003D
$EEEE:85 3F    STA $003F
$EEF0:20 C9 C1 JSR $C1C9    Multiply the random number by the hero's agility.
$EEF3:A5 40    LDA $0040
$EEF5:38      SEC
$EEF6:E5 42    SBC $0042
$EEF8:A5 41    LDA $0041
$EEFA:E5 43    SBC $0043    Subtract the product $4041 from $4243.
$EEFC:60      RTS
$E89B:B0 07    BCS $E8A4    If the result is less than 0, the attempt to flee
                is successful.

```

Essentially the harder enemies get, the better the chances that they'll have of blocking you.

Here are the conditionals for being blocked:

```

Group 1 (0x00-0x13): (EnemyAgility * (RAND >> 2)) > (HeroAgility * RAND)
Group 2 (0x04-0x1D): (EnemyAgility * ((RAND >> 2) + (RAND >> 3))) >
                    (HeroAgility * RAND)
Group 3 (0x1E-0x22): (EnemyAgility * RAND) >> 1 > (HeroAgility * RAND)
Group 4 (0x23-0x27): (EnemyAgility * RAND) > (HeroAgility * RAND)

```

The only condition where running always works is if the enemy is asleep:

The code	What it does	
-----	-----	
\$E894:24 DF	BIT \$00DF	Get bit 6 of \$DF, which is if the enemy is asleep.
\$E896:70 0C	BVS \$E8A4	If he's asleep, running is successful.

--Initiative--

This is the same equation as running from enemies in Group 1 (\$EEC0), except instead of the hero being blocked, the enemy will go first.

```

*****
C. Chests
*****

```

Each chest type has an ID that corresponds to what treasure is inside.

Chest ID	Chest location	Contents	Code location
-----	-----	-----	-----
02	Charlock B7 (x2) Garinham back room Grave of Garinham B1 Mountain Cave B1	Herb	\$E248
03	Charlock B7 Tantegel throne room	Magic Key	\$E226
04	Garinham back room Mountain cave B2 Tantegel throne room	Torch	\$E26C
05			
06	Charlock B7 Rimuldar	Wings	\$E242
09	Mountain Cave B2	Fighter's Ring	\$E26C
0C	Charlock B7 Grave of Garinham B3	Cursed Belt	\$E26C
0D	Grave of Garinham B3	Silver Harp	\$E2B9
0E	Mountain Cave B2	0x05 + (RAND & 0x0F) Gold or 1/32 chance for the Death Necklace (but only one can be acquired).	\$E250
0F	Tantegel basement	Stones of Sunlight	\$E2DB
10	Shrine west of Kol	Staff of Rain	\$E26C
11	Charlock B2	Erdrick's Sword	\$E297
12	Garinham B1	0x05 + (RAND & 0x0F) Gold	\$E2F1
13	Garinham B1 Tantegel treasury (x4)	0x06 + (RAND & 0x07) Gold	\$E303
14	Garinham back room Mountain Cave B2	0x0A + (RAND & 0x07) Gold	\$E315
15	Charlock B7	0x1F4 + RAND Gold	\$E327
16	Tantegel throne room	120 Gold	\$E339
17	Erdrick's Cave B2	Erdrick's Tablet	\$E3C5

Notice that several of the locations refer to \$E26C. This code takes the chest ID and subtracts 3 from it to determine what item you should get.

Also, it would seem that various IDs are missing, but were never intended to be found in chests. Some of the same algorithms work with buying and receiving items.

--The treasure chest bug--

The code for opening a treasure chest:

The code	What it does
-----	-----
\$E209:A5 3A LDA \$003A	Load the hero's x position.
\$E20B:D9 21 03 CMP \$0321,Y	Check if that x position contains treasure.
\$E20E:D0 07 BNE \$E217	
\$E210:A5 3B LDA \$003B	If it does, load the hero's y position.
\$E212:D9 22 03 CMP \$0322,Y	Check if that y position contains treasure.
\$E215:F0 0A BEQ \$E221	
\$E221:B9 23 03 LDA \$0323,Y	If it does, load the treasure ID.

\$E224:85 DE STA \$00DE Store it at \$DE.

So instead of checking if the actual sprite behind the hero is a treasure chest, it looks to see if that is a valid position for treasure. We will see why this is prone to be buggy in a bit. After the above code it starts checking which treasure to give you, which we'll skip here. Instead we'll move to right after that where that position is added to an array of coordinates:

The code	What it does
-----	-----
\$E39A:A2 00 LDX #\$00	Set the pointer to the start of the array.
\$E39C:BD 1C 60 LDA \$601C,X	Check if the first slot is empty.
\$E39F:1D 1D 60 ORA \$601D,X	Also check if the second slot is empty.
\$E3A2:F0 07 BEQ \$E3AB	If they are full, then branch.
\$E3A4:E8 INX	
\$E3A5:E8 INX	Otherwise add 2 to the pointer.
\$E3A6:E0 10 CPX #\$10	If the pointer is greater than or equal to the
\$E3A8:D0 F2 BNE \$E39C	size of the array, then the checking is complete.
\$E3AA:60 RTS	Otherwise repeat the search for an empty slot.
\$E3AB:A5 3A LDA \$003A	If an empty slot was found, load the x coordinate
\$E3AD:9D 1C 60 STA \$601C,X	into it.
\$E3B0:A5 3B LDA \$003B	And load the y coordinate into the next slot.
\$E3B2:9D 1D 60 STA \$601D,X	

The problem here comes if the array is full. If the array is full, the coordinates of the current chest won't be stored there and consequently you'd be able to keep opening that chest. The programmers thought they got around this with various methods. They made sure that no dungeon would contain more chests than could fit into the array, which would be 8. And whenever you entered or exited a town or dungeon, the array was always wiped. What they didn't foresee was to clear the array when you die. So you could go into a dungeon, collect treasure chests, die, and wake up in the castle with that array still full. On top of that, every time you start off in the throne room of Tantegel, it automatically fills the last 3 positions of the array so that you aren't able to pick up the initial 3 treasures anymore. It doesn't matter that you can't see the chests because as previously mentioned, seeing them doesn't matter. It's the array of coordinates that matter. So if you get enough chests in a cave (2 minimum), die, and raid the treasury in Tantegel, you will come upon a chest that never disappears. All because that coordinate can never be stored in that full array.

D. Name choice and stats

The algorithm for figuring out stats in relation to your name. This loads the base values for the four main stats plus which spells you will have at that level:

The code	What it does
-----	-----
\$99C0:A4 3C LDY \$003C	
\$99C2:B1 22 LDA (\$22),Y	Get the strength value.
\$99C4:85 C8 STA \$00C8	Store it.
\$99C6:C8 INY	Increment the pointer.
\$99C7:B1 22 LDA (\$22),Y	Get the agility value.

\$99C9:85 C9	STA \$00C9	Store it.
\$99CB:C8	INY	Increment the pointer.
\$99CC:B1 22	LDA (\$22),Y	Get the max HP value.
\$99CE:85 CA	STA \$00CA	Store it.
\$99D0:C8	INY	Increment the pointer.
\$99D1:B1 22	LDA (\$22),Y	Get the max MP value.
\$99D3:85 CB	STA \$00CB	Store it.
\$99D5:C8	INY	Increment the pointer.
\$99D6:B1 22	LDA (\$22),Y	Get the higher level spells.
\$99D8:05 CF	ORA \$00CF	
\$99DA:85 CF	STA \$00CF	Store it in bits 0 and 1 of \$CF.
\$99DC:C8	INY	Increment the pointer.
\$99DD:B1 22	LDA (\$22),Y	Get the lower level spells.
\$99DF:85 CE	STA \$00CE	Store it at \$CE.

Later, the value of each of the first four letters of your name is added up:

The code		What it does
-----		-----
\$F085:A2 04	LDX #\$04	Set the pointer to 4.
\$F087:A9 00	LDA #\$00	Set the sum to 0.
\$F089:18	CLC	
\$F08A:7D B4 00	ADC \$00B4,X	Add the value of \$B4 offset by the pointer to the sum.
\$F08D:CA	DEX	Subtract 1 from the pointer.
\$F08E:D0 F9	BNE \$F089	If it's 0, continue.

Decide from that sum which attributes will be limited:

The code		What it does
-----		-----
\$F090:85 42	STA \$0042	Store the sum of the letters at \$42.
\$F092:29 03	AND #\$03	Get bits 0 and 1 of the sum.
\$F094:85 43	STA \$0043	Store that at \$43.
\$F096:A5 42	LDA \$0042	Load the sum.
\$F098:4A	LSR	
\$F099:4A	LSR	Divide it by 4.
\$F09A:29 03	AND #\$03	Get bits 0 and 1 of the result.
\$F09C:85 42	STA \$0042	Store that at \$42.
\$F09E:A5 43	LDA \$0043	Load \$43 (bits 0 and 1 of the sum).
\$FOA0:4A	LSR	
\$FOA1:B0 0A	BCS \$FOAD	
\$FOA3:A5 C8	LDA \$00C8	If it's an even number, limit the strength.
\$FOA5:20 0C F1	JSR \$F10C	
\$FOA8:85 C8	STA \$00C8	
\$FOAA:4C B6 F0	JMP \$F0B6	
\$FOAD:A5 CB	LDA \$00CB	If it's an odd number, limit the MP.
\$FOAF:F0 05	BEQ \$F0B6	
\$F0B1:20 0C F1	JSR \$F10C	
\$F0B4:85 CB	STA \$00CB	
\$F0B6:A5 43	LDA \$0043	Load \$43 (bits 0 and 1 of the sum).
\$F0B8:29 02	AND #\$02	Get just bit 1.
\$FOBA:D0 0A	BNE \$FOC6	If bit 1 of the sum is 0, limit the agility.
\$FOBC:A5 C9	LDA \$00C9	
\$FOBE:20 0C F1	JSR \$F10C	
\$FOC1:85 C9	STA \$00C9	
\$FOC3:4C CD F0	JMP \$FOCD	
\$FOC6:A5 CA	LDA \$00CA	If bit 1 of the sum is 1, limit the HP.

\$FOC8:20 0C F1 JSR \$F10C
\$FOCB:85 CA STA \$00CA

The limiting of the attributes:

The code		What it does
\$F10C:85 3C	STA \$003C	Store the current attribute at \$3C.
\$F10E:A9 09	LDA #\$09	
\$F110:85 3E	STA \$003E	
\$F112:A9 00	LDA #\$00	
\$F114:85 3D	STA \$003D	
\$F116:85 3F	STA \$003F	
\$F118:20 C9 C1	JSR \$C1C9	Multiply the attribute by 9.
\$F11B:A5 40	LDA \$0040	
\$F11D:85 3C	STA \$003C	
\$F11F:A5 41	LDA \$0041	
\$F121:85 3D	STA \$003D	Store the result in the double byte \$3C3D.
\$F123:A9 0A	LDA #\$0A	
\$F125:85 3E	STA \$003E	
\$F127:A9 00	LDA #\$00	
\$F129:85 3F	STA \$003F	
\$F12B:20 F4 C1	JSR \$C1F4	Divide the result by 10.
\$F12E:A5 3C	LDA \$003C	
\$F130:18	CLC	Take that result and add it to \$42, which is now
\$F131:65 42	ADC \$0042	bits 2 and 3 of the original sum.

So in general, the best sums are those where bits 2 and 3 are both 1. Bit 0 of the sum determines if it will be long term strength or long term MP. Bit 1 of the sum determines if it will be long term agility or long term HP.

* IV. Frequently Asked Questions *

Q: What is the best way to contact you?

A: I read the gamefaqs.com forum for Dragon Warrior fairly often. I seldom check my e-mail.

Q: Will you ever figure out formulas for other games like the rest of the Dragon Warrior series or other RPGs?

A: I never say never, but I have no plans in the immediate future.

Q: What does the Fighter's Ring do?

A: It has no meaningful purpose. It only changes what a single townspeople says. Don't pay any attention to websites, FAQs, or even original manuals that say otherwise. They all just assumed that it had a function and made a sloppy guess that has carried on for some time.

Q: Does the Silver Harp have any other effects depending on your location?

A: Nope. It always summons the same lame group of enemies.

Q: Is there a limit to how long enemies can be asleep?

A: No. The reason the hero's nap time is limited is because the limited random generator produces distinct patterns when run consecutively. If you sit by idly while waiting to make a command, the random generator is run 60 times per second. This effectively makes it a time-based random, which can result

in any outcome. Tool-assisted speedruns have taken advantage of this.

Q: Is it possible to get in an encounter the first step after battle?

A: Yes. There is no counter, so theoretically you could have infinite encounters in a row. This is actually used in various tool-assisted speedruns to level up quickly.

This document is copyright Ryan8bit and hosted by VGM with permission.