

PokéModr

Ben Boeckel

©2005 Nerdy Productions

November 5, 2005

Contents

1	Introduction	3
2	Main Menu	5
2.1	Set Folder	5
2.2	Configuration	5
2.3	Version	5
3	Items	6
3.1	Item Effects	6
3.2	Ball Effects	8
3.3	Berry Effects	8
4	Moves	9
4.1	Move Effects	9
4.2	Stat Values	13
5	Tiles	15
6	Sprites	16
7	Pokémon	17
7.1	TM/HM Learning	18
7.2	Move Learning	18
7.3	Stats	18
7.4	Breeding	18
8	Stores	20
9	Maps	21
9.1	Map Appearance	21
9.2	Tile Properties	21
9.2.1	Tile Effects	22

9.2.2	People Values	23
9.3	Trainers	23
9.4	Warps	24
9.5	Wild Pokémon Encounters	25
10	Submitting	26
11	Thanks	27
12	Version History	29
12.1	v.5	29
12.2	v.5PR3	29
12.3	v.5PR2	29
12.4	v.5PR1	29
12.5	v.5b	30
12.6	v.4b2	30
12.7	v.4b	31
12.8	v.3b	31
12.9	v.2b	31
12.10	v.1b	32
13	Miscellaneous	33
A	Flags	34
B	Game Mechanics	35
B.1	Stats and Damage	35
B.2	Pokémon Growth Styles	36
B.3	Stat Multipliers	37
B.4	Type Charts	37
B.5	Hidden Power	39
B.6	Critical Hits	39
B.7	Status Effects	40
B.8	Catching	41
B.9	Breeding	41

1 Introduction

Thank you for downloading PokéModr. To install PokéModr, send pokemodr.**z, pokemodr.**y, and pokemodr-data.**y to your calculator. This program allows PokéMods to be made which will be able to be played on the 68k series of calculators. Almost everything can be edited, down to how many HP are cured by a Potion (if one is made, that is). But unfortunately, the inner workings such as the battle engine, movement, and other basics will be set in stone. While we are on the subject of basics, here are the controls:

- F1-F5 (TI89) or R-O (TI92/TI92+/V200)
 - The menu on the bottom shows what each key does
- 2ND
 - Increase Color/Tile in Sprite/Tile/Map Editing
- ◇
 - Decrease Color/Tile in Sprite/Tile/Map Editing
- Up
 - Moves the selector up
- Down
 - Moves the selector down
- Left
 - Edits the option
 - Moves the selector left
 - Enter Sub-modding
- Right
 - Edits the option
 - Moves the selector left
 - Enter Sub-modding
- Enter
 - Confirms

There are a lot of things to do in this program and every part is needed for the game to work right. I would suggest that you write down all of the data which may be important to the game and making the mods because you must keep track of *all* (or at least the most important things) the data that goes into your PokéMod to make the validation process quicker. This may seem unnecessary, but it will cut weeks off of the time it takes to get your PokéMod released and it will speed to the top of the list of PokéMods to be finished compared to one that a staff member of [Nerdy Productions](#) must go through and check it by hand. This also makes sure that you create a game that is quality because you are taking the time to create the tools that the player has to get through the game and making the path that he or she must take. A lot of time has and will be put into this project, so please make it worth our time to actually put your playable PokéMod out onto the Great Information

Highway. Please use this document to help you create your PokéMod and get it to [Nerdy Productions](#) for it to be validated and made into a playable game. I cannot wait to see what ideas everyone has with this program.

2 Main Menu

Most of it is straight forward, so I am only going to explain the items that may be a little unclear.

2.1 Set Folder

This triggers a dialog box in which the current folder being used is displayed as well as a dialog box asking for which folder the files that are going to be modded are going to be saved/read from. This does not check to verify that the currently set files are still existent in that folder, so be careful when choosing the folder.

2.2 Configuration

This displays a dialog box with the names of the folder, as well as the names of the current files set in the configuration. The number in the parentheses is how many elements of the given type are stored in that file. After that, if there are any, the list of the map files that are currently set are displayed. This is useful for checking how many of each data type you have created.

2.3 Version

The number in the lower-left corner of the screen. This should be used to check for updates on the [Nerdy Productions](#) website.

3 Items

Creating and modifying items is one of the simplest things to do. A move file is required to edit items because the TM/HM effect value calls the name of the move from that file for easy reference. Items are used throughout the game to enhance or heal Pokémon. They can also be used to access new areas and get items. Anything that seems unreasonable or unbalanced will be rejected, such as a cheap item that heals all health. Here is how an item is described:

- **Name**
-The name of the item as it will appear in the game. This is used by other Modding sections to refer to the item.
- **Sellability**
-Whether or not the item can be sold at a store. The item can still be bought at a store regardless of this flag.
- **Bitter**
-If this is set, the item decreases happiness when used.
- **Price**
-How much the item costs in a store. When sold, half this amount will be received.
- **Effect**
-The effect that the item has. See [this table](#) for what each effect means and how the values affect the item's usage.
- **Effect Values**
-Values that modify how the effect works.

3.1 Item Effects

Effect	Description	Value 1	Value 2
No Effect	Does nothing	N/A	N/A
TM/HM	Teaches a move	Number of the TM or HM	Move that is taught
Full Restore Cure	Heals everything	N/A	N/A
Full Heal Cure	Clears status	N/A	N/A
HP Cure	Cures HP	HP restored	N/A
Burn Cure	Cures Burn	N/A	N/A
Sleep Cure	Cures Sleep	N/A	N/A
<i>continued on next page</i>			

<i>continued from previous page</i>			
Effect	Description	Value 1	Value 2
Poison Cure	Cures Poison	N/A	N/A
Freeze Cure	Cures Freeze	N/A	N/A
Paralyze Cure	Cures Paralysis	N/A	N/A
Accuracy (Battle)	Boosts accuracy	Steps boosted	N/A
Attack (Battle)	Boosts attack	Steps boosted	N/A
Defense (Battle)	Boosts defense	Steps boosted	N/A
Special (Battle)	Boosts special	Steps boosted	N/A
Speed (Battle)	Boosts speed	Steps boosted	N/A
Run (Battle)	Escapes battle	Chance (out of 255)	N/A
Evolution	Evolves Pokémon	Number associated	N/A
Ball	Catches wild Pokémon	Type of Ball (see this table)	value/2 multiplier
Map	Does nothing	N/A	N/A
Fishing Rod	Allows fishing	Encounter list	N/A
Repel	Repels wilds	Steps it works for	N/A
Attack Boost	Boosts attack EXP	EXP/use (max 25500)	N/A
Defense Boost	Boosts defense EXP	EXP/use (max 25500)	N/A
Special Boost	Boosts special EXP	EXP/use (max 25500)	N/A
Speed Boost	Boosts speed EXP	EXP/use (max 25500)	N/A
HP Boost	Boosts HP EXP	EXP/use (max 25500)	N/A
PP Boost	Boosts PP	1PP/use/value	N/A
Revive	Revives Pokémon	Restores HP/value	N/A
Escape	Go to Pokémon Center	N/A	N/A
PP Restore (1)	Restore PP (1 move)	PP restored	N/A
PP Restore	Restore PP	PP restored	N/A
Itemfinder	Detects hidden items	N/A	N/A
Special Guard	Extra special defense	Turns it lasts	N/A
EXP Share	Shares EXP with party	EXP/value shared	N/A
Biking	Enables faster travel	N/A	N/A
Level Boost	Increases level	Levels increased	N/A
Scope	Shows Pokémon	Scope number	N/A
Key	Opens barriers	Key number	N/A
Coin	Used for gambling	N/A	N/A
<i>continued on next page</i>			

continued from previous page

Effect	Description	Value 1	Value 2
Coin Holder	Holds coins	Max coins held	N/A
Voucher	Redeem for items	Number associated	One-time or infinite
Flute	Awakens Pokémon	N/A	N/A
Berry	Does things when held in battle (see this table)	What it does	N/A
Acorn	Can be converted into another item	Which item is made	N/A
Leftovers	Heals 6.25% of HP_{Max} every turn	N/A	
Stat Modifier	Modifies stats	Stat modified	

3.2 Ball Effects

This is how the Ball is supposed to be used as and what effects are needed for the multiplier to work is.

Type	Effect Needed
Standard	The multiplier is standard
Safari	Only used in Safari Zone-type area
Master	Never fails to catch
Level	Multiplier is ignored and special multipliers apply
Fast	Applies if the Pokémon can run
Love	Applies if enemy has the opposite gender as the player's current
Moon	Applies if the enemy evolves by a Moon Stone
Lure	Applies if enemy was caught using a Fishing Rod
Friend	Pokémon will start with 200 Happiness points

3.3 Berry Effects

This is how the Ball is supposed to be used as and what effects are needed for the multiplier to work is.

Type	Effect
10HP	Heals 10HP when HP is below 25%
30HP	Heals 30HP when HP is below 25%
80HP	Heals 80HP when HP is below 25%
100HP	Heals 100HP when HP is below 25%
Confusion	Cures Confusion
Sleep	Cures Sleep
Burn	Cures Burn
Freeze	Cures Freeze
Paralyze	Cures Paralyze
Poison	Cures Poison
All	Cures any ailment

4 Moves

Creating and modifying moves is also simple to do. There are no files required for modding moves. Moves are what make the game playable as a strategy game. Moves that are considered unfair or unbalanced, such as a 40PP, OHKO, Never Miss, Stat Boosting move, will be rejected. Here is how a move is described:

- **Name**
-The name of the move that will appear in the game. This is what other Modding section refer to this move as.
- **Type**
-The type of the move. This decides the effectiveness of the move against other types if the move deals damage. If it does not deal damage, it is only for classification purposes.
- **PP**
-How many Power Points are given to the item by default. To get the maximum amount of PP a move can have, multiply this number by 8/5 and drop any decimal places that may occur.
- **Power**
-The power that the move has. Some effects ignore this value.
- **Accuracy**
-The chance (out of 255) that this move will hit the opponent. If this misses, all effects (except Recoil (if miss)) do not have a chance of happening.
- **Effect #**
-These are the effects that modify how the attack works. See [this table](#) for what each means and how the value affects the move.
- **Effect Chance**
-The chance (out of 255) that the effect will happen.
- **Effect Value**
-The value that modifies the effect.

4.1 Move Effects

Effect	Description	Value
No Effect	Does nothing	N/A
HP Player	Takes users HP away	Takes HP/value
<i>continued on next page</i>		

<i>continued from previous page</i>		
Effect	Description	Value
HP Enemy	Does damage relative to amount left	HP left/ value left
Attack Player	Raises user attack	Stat modifier
Attack Enemy	Lowers enemy attack	Stat modifier
Defense Player	Raises user defense	Stat modifier
Defense Enemy	Lowers enemy defense	Stat modifier
Speed Player	Raises user speed	Stat modifier
Speed Enemy	Lowers enemy speed	Stat modifier
Special Player	Raises user special	Stat modifier
Special Enemy	Lowers enemy special	Stat modifier
Evasion	Raises user evasion	Stat modifier
Accuracy	Lowers enemy accuracy	Stat modifier
Flinch	Opponent loses a turn	N/A
Freeze	Freezes enemy	N/A
Burn	Burns enemy	N/A
Sleep	Puts enemy to sleep	N/A
Poison	Poisons enemy	N/A
Paralyzed	Paralyzes enemy	N/A
Confuse	Confuses enemy	N/A
Toxic Poison	Toxic poisons enemy	N/A
Recoil (if hit)	Damages user	Damage/ value done
Absorb	Absorbs HP from enemy	Damage/ value given
Lossless Swap	Switches and keeps stat modifiers	N/A
Trap	Prevent enemy from escaping	N/A
Repeat (1 Turn)	Repeat move value /turn	Max hits/turn
Repeat	Repeat move value turns	Max repeats
Bide	Absorbs damage	Damage* value back
First	Goes first	N/A
Counter (Phys)	Returns physical damage	Damage* value done
Counter (Spec)	Returns special damage	Damage* value done
Seed	Seeds the enemy	Damage/ value given
Substitute	Dummy blocks attacks	Has HP/ value to start
Fly Turn	Flys into air	Turns flying
<i>continued on next page</i>		

<i>continued from previous page</i>		
Effect	Description	Value
Never Miss	Never misses	N/A
One Hit K.O.	KOs enemy	N/A
Dream Eater	Absorbs HP if sleeping	Damage/ value given
Endure	Have at least 1HP next turn	Chance/ value for consecutive uses
Explosion	KOs user	Extra damage factor
>=1HP Left	Never KO enemy	N/A
Flail	Do damage inverse to HP left	N/A
Freeze Break	Thaws user	N/A
Surf	HM surf	N/A
Fly	HM fly	N/A
Cut	HM cut	N/A
Flash	HM flash	N/A
Strength	HM strength	N/A
Clear	Clears stat modifier	See this table
Power Builder	Power grows each time it hits in a row	Factor
Delayed Hit	Damages later	Turn delay
Recoil (if miss)	Does damage to user if misses	Damage/ value to user if miss
Hidden Power	Type/power depends	N/A
Present	Random effect	N/A
Locks On	Next attack will not miss	N/A
Rest	Heals status and puts user to sleep	Number of turns taken to rest
Recovery Turn	User recovers from attack	Turns to recover
Heal	Heals user's HP	Heals HP/ value
Status Shield	Stops status changes	Turns effective
Combine/Split HP	Splits HP	N/A
Damage==Level	Does level in damage	N/A
Mimic	Learns last move for battle	N/A
Transform	User takes enemy's abilities	PP to start with
Metronome	Does random move	N/A
Money	Receive money for each hit	Get damage* value
Mirror	Do enemy's last move	N/A
Spikes	Damages when enemy switches	Damage done
<i>continued on next page</i>		

<i>continued from previous page</i>		
Effect	Description	Value
Fixed Damage	Does a certain amount of damage	Damage dealt
Power Up	Next move will be better at Critical Hits	Chance/ value for consecutive uses
Sketch	Learn last move	N/A
Sleep Talk	Attacks while sleeping	N/A
Escape	Escapes from battle	N/A
Protect	Protects against attack	Chance/ value for consecutive uses
Perish Song	KOs user and enemy	Turns until KO'd
Confuse Self	Confuses user after use	N/A
Break Free	Spikes, Seed cleared	N/A
Rain	Makes weather rainy	Turns it lasts
Sun	Makes weather sunny	Turns it lasts
Sandstorm	Makes weather sandy	Turns it lasts
Crit Hit Bonus	Extra Critical Hits	Domain
Encore	Enemy repeats move	N/A
Softboiled	Heals HP	HP/ value healed
Sp. Att Player	Raises user sp. attack	Stat modifier
Sp. Att Enemy	Lowers enemy sp. attack	Stat modifier
Sp. Def Player	Raises user sp. defense	Stat modifier
Sp. Def Enemy	Lowers enemy sp. defense	Stat modifier
Hit Fly	Extra damage if digging	Damage* value
Hit Dig	Extra damage if flying	Damage* value
Dig Turn	Digs underground	Turns underground
Headbutt	Headbutts trees	N/A
Waterfall	Climbs waterfalls	N/A
Rain Bonus	Bonus if rain	Damage* value done
Sunny Bonus	Bonus if sunny	Damage* value done
Rain Hinder	Hinders if rain	Damage/ value done
Sunny Hinder	Hinders if sunny	Damage/ value done
Minimize	Sets minimized flag	N/A
Minimize Bonus	Bonus if hitting minimized	Damage* value done
Pursuit	Bonus if enemy switches	Damage* value done
All Stat Party	Raises all stat modifiers	Stat modifier
Whirlpool	HM whirlpool	N/A
<i>continued on next page</i>		

<i>continued from previous page</i>		
Effect	Description	Value
Second	Goes second	N/A
Conversion	Changes your type to the same as one of your moves	N/A
Conversion2	Changes type to one that is resistant to the enemy's last move	N/A
Switch Enemy	Forces enemy to switch	N/A
Copy Stats	Copys enemy's stat modifiers to yours	N/A
Disable	Disables enemy's move	N/A
Special Shield	Prevents special damage	Damage*value done
Physical Shield	Prevents physical damage	Damage*value done
Range (Fixed)	Does a range of damage	1-value done
Range (Relative)	Does a range of damage	1-(HP*value/2) done
Evasion Enemy	Lowers enemy Evasion	Stat Modifier
Accuracy Player	Raises player Accuracy	Stat modifier
Rock Smash	Smashes rocks	N/A
Happiness	Does damage based on happiness	Happy or Mad status
Tri Attack	Either causes Burn, Freeze, or Paralysis	N/A
Attract	If enemy is opposite gender, will fail to attract 1/2 the time	N/A
Ignore	Ignores an enemy modifier when attacking	Which stat is ignored
Nightmare	Deals set amount of damage each turn enemy is asleep	$HP_{Max}/value$ done
Defense Curl	Sets Defense Curl flag	N/A
Def. Curl Bonus	Gets a power boost if a Defense Curl move was used	value times as powerful
Mist	Prevents changing stat modifiers	N/A
Thief	Steals the enemy's item	N/A

4.2 Stat Values

These are the effects for clearing.

Clear Effect	What is cleared
Attack Player	Attack modifier of player
Speed Player	Speed modifier of player
<i>continued on next page</i>	

<i>continued from previous page</i>	
Clear Effect	What is cleared
Defense Player	Defense modifier of player
Special Player	Special modifier of player
Sp. Att Player	Special attack modifier of player
Sp. Def Player	Special defense modifier of player
Status Player	Status of player
Conf Player	Confusion of player
Evasion Player	Evasion modifier of player
Acc Player	Accuracy modifier of player
All Player	All modifiers of player
Attack Enemy	Attack modifier of enemy
Speed Enemy	Speed modifier of enemy
Defense Enemy	Defense modifier of enemy
Special Enemy	Special modifier of enemy
Sp. Att Enemy	Special attack modifier of enemy
Sp. Def Enemy	Special defense modifier of enemy
Status Enemy	Status of enemy
Conf Enemy	Confusion of enemy
Evasion Enemy	Evasion modifier of enemy
Acc Enemy	Accuracy modifier of enemy
All Enemy	All modifiers of enemy
Attack Both	Attack modifiers
Speed Both	Speed modifiers
Defense Both	Defense modifiers
Special Both	Special modifiers
Sp. Attack Both	Special attack modifiers
Sp. Defense Both	Special defense modifiers
Status Both	Status modifiers
Conf Both	Confusion
Evasion Both	Evasion modifiers
Acc Both	Accuracy modifiers
All Both	All modifiers
Status Party	All Status effects in the party

5 Tiles

Tiles are used in the map editing. They are interlaced 16x16 grayscale sprites. The matrix in the top left is an enlarged version of the tile. The black box is where the selector currently is. The tile is shown as it will in the map in a box towards the lower right. In the `pokemod.h` file, all tiles that allow the player to move through (such as ledges, grass, and ladders) must be noted by their number. Tiles that have a chance of a wild Pokémon appearing due to a certain effect (walking on, headbutting, or fishing), this also needs to be listed. Tiles that can be surfed on also need to be noted.

6 Sprites

Sprites are the pictures that are associated with each Pokémon. They go in order, i.e. the first sprite is associated with the first Pokémon. Although a single file can only hold 163 sprites, I will allow a second sprite file to be attached with the game and used from there. The editor will not show it correctly, but in the game, the calling routine will be modified to accomodate this. They are standard 40x40 grayscale sprites. The boxed-in area in the top left is an enlarged version of the sprite. The black box is where the selector currently is. The sprite is shown as it will in the editor or game in a box towards the lower right. If you would like to make sprites on the computer using another editor, you may, just be sure to make them 40x40 in dimentions. Just e-mail a member of [Nerdy Productions](#) the sprites themselves and one of us will take care of it.

7 Pokémon

This is where you can modify what each Pokémon's stats and data are set to. The first level of modding only requires a sprite file to be present. There is a limit on how powerful a Pokémon can be. Try to make it so that the whole game is balanced out across the playing field. Here is how it is broken down:

- **Name**
 - The name of the Pokémon that will appear in the game. This is how it will be referenced in the Modding program.
- **Evo Level**
 - The level at which the Pokémon evolves. Here is how the numbers are treated:
 - 1-100 is for natural evolution at a certain level
 - 101 means that it never evolves
 - 102 means that it will evolve only when traded
 - 103 means that it will evolve when happy
 - 104+ are evolution through stones¹
- **Evo Species**
 - The species that the Pokémon evolves into.
- **Picture**
 - The sprite which will appear next to it in the status screen.
- **Edit TM/HMs**
 - Goes to the sub-modding of which TMs and HMs the Pokémon can learn.
- **Types**
 - The types of the Pokémon.
- **Edit Moves**
 - Goes to the sub-modding of which moves are learned at which levels.
- **Style**
 - The formula which decides how many **Experience Points** are needed for a certain level.
- **Edit Stats**
 - Goes to the sub-modding of the stats.
- **Exp Factor**
 - This modifies how much experience is given by defeating this Pokémon in battle.

¹If you would like to have Eevee-like stone evolution, note it in the `pokemod.h` file.

- **Catch Factor**
-How easy it is to catch the Pokémon.
- **Can Run**
-Whether the Pokémon can run from battle or not.
- **Edit Breeding**
-Goes to the sub-modding of breeding and other miscellaneous values.

7.1 TM/HM Learning

This is just a listing of 128 TMs and HMs and whether or not they can be learned by the Pokémon. The break between TMs and HMs must be noted in the `pokemod.h` file.

7.2 Move Learning

Modding the moves learned by a Pokémon requires a move file because it calls the name of the move from there. The name of the move is followed by a level, which is where that Pokémon learns that level naturally.

7.3 Stats

Modding stats is easy. These stats are used to calculate the stats later. The higher the number, the better the stat will be. All three special stats are there to be compatible with both RBY and GSC stat rule sets. Here is a listing of the stats:

- HP
- Attack
- Defense
- Speed
- Special
- Special Attack
- Special Defense

7.4 Breeding

This is actually a bunch of things put together, but it has a focus on Breeding. [This section](#) has more information on Breeding. Here's how Breeding sub-modding is broken down:

- **Gender Factor**
-This decides how often the Pokémon is male versus female. If the DV_{Att} is less than this number, it is male. If it is 16, it is considered as one of the neuter group.

- **Hatch Steps**
 - How many steps it takes to hatch a Pokémon from its egg.
- **Species**
 - What Pokémon will come out of the egg coming from it. Usually this is the lowest stage in the evolutionary stage.
- **Breed Factors**
 - Used to determine compatability between Pokémon.
- **Special Moves**
 - Moves that can be learned through Move Tutors.
- **Egg Moves**
 - Moves that can be learned through breeding.

8 Stores

This tells the game which items are sold in the stores. An item file is needed to get the names of the items. There can be up to twelve items sold at a single store.

9 Maps

Maps are the most complex part of making a mod. There are warps, events, wild Pokémon encounters, trainers, and the map itself. Nothing is needed to actually make a map file, but in order to change anything except for the map name and warps, various files are needed. Here is how the map modding is arranged:

- **Edit Map**
-This goes to the sub-modding of the appearance of the map.
- **Name**
-This is simply the name of the map that will appear in a box in the bottom of the screen when the map is entered.
- **Edit Tile Properties**
-Goes to the sub-modding of the properties of tiles.
- **Edit Trainers**
-Goes to the sub-modding of the trainers.
- **Edit Warps**
-Goes to the sub-modding of the warps on the map.
- **Edit Wild Pokémon**
-Goes to the sub-modding of which Pokémon are found in the land and water (while surfing).
- **Fly to Warp**
-Tells the game which warp to go to when the map is flown to. A value of 64 tells the game that it cannot be flown to.

9.1 Map Appearance

Modding the map is pretty straight forward. A tile file is needed to change the map's appearance. This is how it is going to look in the game (plus the trainers and tile effects and without the menu). The map has been reduced to a 64x64 area due to memory consumption with larger map sizes. This should be ample space for multiple maps on a single file (in different areas of course).

9.2 Tile Properties

Modding the properties that the tiles have is also pretty basic. An item file is needed so that the item and hidden item effects work correctly. Here is how a tile effect is broken down:

- **Property Number**
-This is the current properties that is being modified (there are a total of 64).
- **x**
-This is the x coordinate of the tile which it will affect.

- **y**
-This is the y coordinate of the tile which it will affect.
- **Existence Flag**
-This is the number of the flag that determines whether the property is applied or not. There are 512 flags that can be used. See [Appendix A](#) for more information on flags.
- **Flag Value**
-This tells the game under what condition the flag is in if the property will apply.
- **Sprite**
-This is the sprite that will accompany the property.
- **Effect**
-This is the effect of the property. See [this table](#) for a listing of each property.
- **Effect Values**
-This is the values which modifies how the effect is handled.
- **Can Move**
-This tells the game engine whether the effect can move around the map.

9.2.1 Tile Effects

These tables have information about the effects that a tile can have. All effects block movement into that tile except for hidden items.

Effect	Vaule 1	Value 2
Nothing	N/A	N/A
Item	Item	N/A
Person/Thing	Person effect	See below
Sign	Dialog number	N/A
Force Battle	Trainer number	N/A
Hidden Item	Item	N/A
Barrier	Key number	N/A
Strength Block	Block number	N/A
Item Giver	Item	Voucher number
Slots	N/A	N/A
Card Flip Game	N/A	N/A
Coin Redeeming	Redeem list	N/A

9.2.2 People Values

Effect	Description	Value 2
Person	A person	Dialog number
Link	Activates the link routines	Dialog number
Heal	Heals the party	Dialog number
Heal (Ctr)	Heals the party (Center Style)	Dialog number
PC	A regular PC	N/A
PC (Item)	A PC with access to items only	N/A
PC (Pkmn)	A PC with access to Pokémon only	N/A
Super PC	A special PC	N/A
Move Tutor	Teaches a specific move	Move taught
Daycare	Breeds and raises Pokémon	Which person it is
Other	Describe in <code>pokemod.h</code>	Case

9.3 Trainers

Trainers are the main source of income and experience in the game. There are many values that can be changed to modify them. Here is how they are broken down:

- **Trainer**
-This is the current trainer being modified (out of 64).
- **Sprite**
-This is the sprite that will represent the trainer.
- **x**
-This is the x coordinate of the trainer.
- **y**
-This is the y coordinate of the trainer.
- **Sight**
-This is how far the trainer can see.
- **Direction**
-This is the direction that the trainer is looking. The `Spinning/All/None` option means that the trainer will spin around in its place. If you wish to have a trainer have special properties, add it to the `pokemod.h` file.
- **Money**
-This is how much money you receive when you defeat the trainer.
- **Appear Flag**
-This is the flag that determines whether the trainer appears or not. See [Appendix A](#) for more information on flags.
- **Flag Value**
-This tells how to interpret that flag.

- **Beaten Flag**
-This is the flag tells the game whether the trainer was beaten or not.
- **Flag Value**
-This tells how to interpret the flag.
- **AI Level**
-This tells the battle engine how smart the trainer is.
- **Pokémon #**
-This is the species of the Pokémon that will be in this spot of the trainer's line-up.
- **Level**
-This is the level of that Pokémon.

9.4 Warps

Editing warps is a little tricky. In order to get these to point to the correct maps, that map already had to be created. This basically means that warps are usually left as the last thing to be modded. Here is how it is broken down:

- **Warp Number**
-This is the current warp (out of 64) being modded.
- **x**
-This is the x coordinate of the warp.
- **y**
-This is the y coordinate of the warp.
- **Dir In**
-When the character is warped from this point, this tells in which direction the warp is active in. In other words, if the option is **Down**, only entering from the bottom will activate this warp. The **Spinning/All/None** option means that the warp is activated no matter what direction it is entered from.
- **Dir Out**
-When the character is warped to this point, this tells in which direction the character will be moved automatically. The **Spinning/All/None** option means that the player will stay on that tile, facing down.
- **Type**
-This just tells how the game goes to the next map. If it is a **Door/Stair**, the screen fades to black, then back to the new map at that warp point. Warp Pads send the player to the top of the screen spinning. Holes send the player scolling down within its tile.
- **Flag Exist**
-This tells the game which flag to check as to whether the warp exists or not.

- **Flag Value**
-This tells the game what to look for when looking at the flag.
- **Flag Affected**
-This tells the game which flag to change (if needed) when the warp is activated.
- **Flag Value**
-This tells the game what to look for when looking at the flag.
- **To Map**
-This is the number of the map to warp to.
- **Warp**
-This is the number of the warp to teleport to in that map.

9.5 Wild Pokémon Encounters

Modding Pokémon Encounters are simple. Only a Pokémon file is needed. There are 8 encounter lists to mod. There are:

- Morning Grass
- Day Grass
- Night Grass
- Fishing Rod 0
- Fishing Rod 1
- Fishing Rod 2
- Headbutt
- Rock Smash

The Pokémon at the top of the encounter list are weighted heavier than the ones at the bottom, but how much exactly remains to be decided.

10 Submitting

There are a few things that are needed before the PokéMod is even considered to be ready for validation. Here is a (small and incomplete) checklist:

- All information must be put into a `pokemod.h` file for the mod's readme
- Any special events that should happen at certain times
- Any flags that should be treated as a group
- All dialog strings (up to 4096) (Trainers do not need strings, leaders do)
- Special Move Sets for any Gym Leaders, Elite Four, or Master trainers
- Anything else you would like to have different that is not able to be currently modded (storyline, Safari Zone-type area, etc.)
- Fill out [this](#) header file with pertinent information that will run behind the game and submit it with the rest of the PokéMod. Any PokéMod submitted without this will be ignored. It is updated separately from PokéModr, so be sure to download this file soon before uploading your Pokd.
- Upload in `.zip` format at [this form](#).

11 Thanks

- TI
- Makers of TIGCC
- Makers of EXTGraph
- Makers of AdvDialogs
- Makers of Pokémon:
 - GameFreak
 - Nintendo
- Everyone who helped figure out problems:
 - hypersonic
 - joeyadams
 - Lionel Debroux
 - MSCoder610
 - Kevin Kofler
- Support/Other:
 - Jordan Krage
 - Kévin Ouelett
 - Kerm Martian
 - TI-Freak8x
- Anyone else I forgot
- All of the resources I used:
 - PokéMap-
A program I used to get some information by [PR Translations](#)
 - [Bulbapedia](#)-
A large database of information on Pokémon
 - [PokéFAQ](#)-
Another Pokémon database
 - [Serebii.net](#)-
Yet another Pokémon database
 - [Pokémon Elite](#)-
Yet *another* Pokémon database

- R. Jones-
Author of the *Treatise on Pokémon Strategy*
- Scott Walker-
Author of a [list of Pokémon trainers](#) from RBY
- The walkthroughs I used
- Makers of the packages that are used in this readme:
 - amsmath
 - amssymb
 - array
 - hyperref
 - xtab
 - geometry
 - rotating
 - multirow

12 Version History

12.1 v.5

- v.5
- November 5, 2005
- New Features:
 - Some readme errors fixed
 - Sprite Modding bug fixed

12.2 v.5PR3

- v.5 Private Release 3
- October 30, 2005
- New Features:
 - Miscellaneous annoyance fixes
 - Sprite Modding blocked for now

12.3 v.5PR2

- v.5 Private Release 2
- October 30, 2005
- New Features:
 - Adding Maps no longer causes a crash
 - Item and Warp Modding fixed (some options didn't change correctly)
 - Infinite loop in Configuration fixed
 - Effects that went off the 89 screen shortened to fit

12.4 v.5PR1

- v.5 Private Release 1
- October 29, 2005
- New Features:
 - Added Breeding and Catching to modding
 - Added more **Item Effects** and **Move Effects**

- More Reference Information documented
- PokéMods may now be submitted
- Added/modified some things for trainers
- All structures have been bit-field optimized as much as possible
- Gave more warps, trainers, tile effects (64 of each may now exist)
- Reduced map area for memory reasons (45% conserved per map file)
- Better readme and walkthrough of making a PokéMod

12.5 v.5b

- v.5 β
- August 29, 2005
- New Features:
 - Annoyance in move modding fixed
 - Added validation system for configuration file (makes previous configuration files invalid)
 - Fixed a bug in the opening of a file
 - Some readme fixes/additions
 - Other fixes

12.6 v.4b2

- v.4 β 2
- August 28, 2005
- New Features:
 - Bug fixes
 - When some things were changed in the source, placement and other things weren't fully changed. This release fixes that.
 - 2000 bytes smaller (uncompressed)
 - Rearranged appendices

12.7 v.4b

- v.4 β
- August 27, 2005
- New Features:
 - MANY more move effects added
 - Map editing works now
 - This readme was created
 - Minor bug fixes
 - Stats now support RBY and GSC standards
 - To support the new features and changes, the following files that were created by ealier version are now invalid:
 - * Items
 - * Moves
 - * Map
 - * Pokémon

12.8 v.3b

- v.3 β
- August 23, 2005
- New Features:
 - Customized Advanced Dialogs GUI
 - Bug fixes

12.9 v.2b

- v.2 β
- August 20, 2005
- New Features:
 - Quick fix to a HUGE bug in v.1 β

12.10 v.1b

- v.1 β
- August 20, 2005
- New Features:
 - First Release

13 Miscellaneous

Crashes It is not my fault if this program causes your calculator to crash. If you have found a bug, please e-mail me at MathStuf@gmail.com and describe in detail the steps to replicate the crash.

Glitches All of the glitches from the original games are removed. Moves that did not work correctly are also fixed. Here is a listing of what has changed/will be gone:

- GameShark (RBY/GSC)-Command Post Plus is there...plus the Super PC
- Missingno. (RBY)-If you do not know it...
- OHKO Moves (RBY)-Could not hit if slower than enemy
- Substitute (RBY)-Did not block status changes
- Counter (RBY)-Only worked for Normal and Fighting attacks
- Haze (RBY)-Cleared enemy status, but not player's
- Present (GS)-Just did not work
- Pokémon Specific items (GSC)-Causes unfairness and will not be supported

Downloading This program may only be found on these sites (links to these are allowed, it just may not be *on* the site), if it is found anywhere else, please e-mail me at MathStuf@gmail.com:

- [Nerdy Productions](#)
- ti-news.net
- ticalc.org

Suggestions If you have any suggestions (i.e. new effects), please e-mail me or put a message up at my [forums](#).

A Flags

Flags determine a lot of what is shown on the map and whether certain events have happened. Although 512 are available, there are a few reserved ones that are used by the game itself. Here is a listing of what those reserved flags are (more may be reserved later):

Flag	Use
0	Always set
1	Always unset
2	Whether the character is on land or water
3 – 10	Used for Cut Trees
11 – 14	Daycare/Breeding

B Game Mechanics

There are a lot of formulae and such that work behind the scenes that define some of the most common things. I have decided to list them here. It is also to show how the game itself works so that it is not a mystery as to how the game actually arrives at its values.

B.1 Stats and Damage

This table has the formulae used for stats and damage.

Result	Formula
Stats	$\frac{(2b + 2d + s)\ell}{100} + u$
Damage	$\left(\min \left(\frac{(2 + \lfloor \frac{2}{5}\ell \rfloor) pStat_{Att}}{50Stat_{Def}}, 997 \right) + 2 \right) x$
Stat Points	$\min \left(\left\lfloor \frac{\sqrt{e - 1} + 1}{4} \right\rfloor, 63 \right)$

DV_{HP} is dependant on the other DV 's. Here is its formula:

$$8(DV_{Att} \bmod 2) + 4(DV_{Def} \bmod 2) + 2(DV_{Spd} \bmod 2) + DV_{Att} \bmod 2$$

Here is the listing of the variables used in the stat and damage formulae along with what each of them stands for.

Variable	Description
ℓ	Level
b	Base stat
d	DV
s	Stat points
DV_{Att}	DV of Attack
DV_{Def}	DV of Defense
DV_{Spd}	DV of Speed
DV_{Spc}	DV of Special
e	Stat experience
u	$\begin{cases} \ell + 10, & \text{If stat is HP} \\ 5, & \text{else} \end{cases}$
$Stat_{Att}$	Stat of user's attack
$Stat_{Def}$	Stat of target's defense
p	Attack's power
x	Other multipliers
HP_{Max}	The maximum allowed HP of the Pokémon
HP_{Cur}	The current HP of the Pokémon
$Value_{Catch}$	The catch value of the Pokémon

These are what define the x in the damage formula. They are controlled by the effects of the move that is being used. What the enemy.

Multiplier	Values
Random Number	$\frac{217 + \text{rand}(39)}{255}$
STAB	$\begin{cases} 1.5, & \text{If move is same type as user} \\ 1, & \text{else} \end{cases}$
Type 1	Multiplier from type chart
Type 2	Multiplier from type chart
Critical Hit	2 if $\text{rand}(256)$ is in the domain
Minimize	$\begin{cases} \text{value}, & \text{If enemy ever minimized} \\ 1, & \text{else} \end{cases}$
Fly	$\begin{cases} \text{value}, & \text{If attack can hit flyers} \\ 0, & \text{else} \end{cases}$
Dig	$\begin{cases} \text{value}, & \text{If attack can hit diggers} \\ 0, & \text{else} \end{cases}$
Pursuit	$\begin{cases} \text{value}, & \text{If enemy switches} \\ 1, & \text{else} \end{cases}$
Rain	$\begin{cases} \text{value}, & \text{If move benefits} \\ 1.5, & \text{If move is Water} \\ \frac{1}{\text{value}}, & \text{If move is hindered} \\ .5, & \text{If move is Fire} \\ 1, & \text{else} \end{cases}$
Sunny	$\begin{cases} \text{value}, & \text{If move benefits} \\ 1.5, & \text{If move is Fire} \\ \frac{1}{\text{value}}, & \text{If move is hindered} \\ .5, & \text{If move is Water} \\ 1, & \text{else} \end{cases}$
Item	$\begin{cases} 1.1, & \text{If move type is boosted by held item} \\ 1, & \text{else} \end{cases}$

B.2 Pokémon Growth Styles

These are the functions used to determine how much experience is needed for a given level. Erratic and Fluctuating only appeared after GSC.

Style	Formula
Erratic	$\ell^3 \left(\frac{100 - \ell}{50} \right), \quad 0 < \ell \leq 50$
	$\ell^3 \left(\frac{150 - \ell}{100} \right), \quad 51 \leq \ell \leq 68$
	$\ell^3 \left(1.274 - \frac{1}{50} \left\lfloor \frac{\ell}{3} \right\rfloor - \begin{cases} 0.000, & \ell \bmod 3 = 0 \\ 0.008, & \ell \bmod 3 = 1 \\ 0.014, & \ell \bmod 3 = 2 \end{cases} \right), \quad 69 \leq \ell \leq 98$
	$\ell^3 \left(\frac{160 - \ell}{100} \right), \quad 99 \leq \ell \leq 100$
Fast	$.8\ell^3$
Normal	ℓ^3
Fading	$1.2\ell^3 - 15\ell^2 + 100\ell - 140$
Slow	$1.25\ell^3$
Fluctuating	$\ell^3 \left(\frac{24 + \lfloor \frac{\ell+1}{2} \rfloor}{50} \right), \quad 0 < \ell \leq 15$
	$\ell^3 \left(\frac{14 + \ell}{50} \right), \quad 16 \leq \ell \leq 35$
	$\ell^3 \left(\frac{32 + \lfloor \frac{\ell}{2} \rfloor}{50} \right), \quad 36 \leq \ell \leq 100$

B.3 Stat Multipliers

Here is the multipliers when your stat is modified in battle. These are multiplied to your stat whenever it is needed.

Stat Modifier	Multiplier
-6	1/4
-5	2/7
-4	1/3
-3	2/5
-2	1/2
-1	2/3
0	1
1	3/2
2	2
3	5/2
4	3
5	7/2
6	4

B.4 Type Charts

This is the type chart used in RBY.

		Enemy's Type														
		Normal	Fire	Water	Electric	Grass	Ice	Fighting	Poison	Ground	Flying	Psychic	Bug	Rock	Ghost	Dragon
Attack's Type	Normal												D	X		
	Fire		D	D		A	A					A	D		D	
	Water		A	D		D				A			A		D	
	Electric		A	D		D				X	A				D	
	Grass		D	A		D			D	A	D		D	A	D	
	Ice			D		A	D			A	A				A	
	Fighting	A					A		D		D	D	D	A	X	
	Poison					A			D	D			A	D	D	
	Ground		A		A	D			A		X		D	A		
	Flying				D	A		A					A	D		
	Psychic							A	A			D				
	Bug		D			A		D	A		D	A			D	
	Rock		A				A	D		D	A		A			
	Ghost														A	
	Dragon															A

This is the typechart used in GSC.

		Enemy's Type																
		Normal	Fire	Water	Electric	Grass	Ice	Fighting	Poison	Ground	Flying	Psychic	Bug	Rock	Ghost	Dragon	Dark	Steel
Attack's Type	Normal												D	X			D	
	Fire		D	D		A	A					A	D		D		A	
	Water		A	D		D				A			A		D			
	Electric			D		D				X	A				D			
	Grass		D	A		D			D	A	D		D	A		D	D	
	Ice		D	D		A	D			A	A				A		D	
	Fighting	A					A		D		D	D	D	A	X		A	A
	Poison					A			D	D			A	D	D			X
	Ground		A		A	D			A		X		D	A				A
	Flying				D	A		A					A	D				D
	Psychic							A	A			D					X	D
	Bug		D			A		D	D		D	A			D		A	D
	Rock		A				A	D		D	A		A					D
	Ghost											A			A		D	D
	Dragon															A		D
	Dark							D				A			A		D	D
Steel		D	D	D		A							A				D	

The blank boxes means that the move does not have a multiplier. The other symbols are modified according to this chart:

Symbol	Multiplier
A	2
D	1/2
X	0

B.5 Hidden Power

Hidden Power is calculated at the time of the battle. The power and type of it depends on your DVs. To get the power, all of your DVs need to be known. Here's the basic formula for it:

$$\frac{321}{2} + \begin{cases} 20, & DV_{Att} \geq 8 \\ 0, & \text{else} \end{cases} + \begin{cases} 10, & DV_{Def} \geq 8 \\ 0, & \text{else} \end{cases} + \begin{cases} 5, & DV_{Spd} \geq 8 \\ 0, & \text{else} \end{cases} \\ + \begin{cases} \frac{21}{2}, & DV_{Spc} \geq 8 \\ 0, & \text{else} \end{cases} - \begin{cases} \frac{1}{2}, & DV_{Spc} \bmod 2 = 0 \\ 0, & \text{else} \end{cases} - \begin{cases} 1, & \left\lfloor \frac{DV_{Spc}}{2} \right\rfloor \bmod 2 = 0 \\ 0, & \text{else} \end{cases}$$

The type is determined by:

$$4(DV_{Att} \bmod 4) + DV_{Def} \bmod 4$$

Then use this chart to determine the type:

Result	Type
0	Fighting
1	Flying
2	Poison
3	Ground
4	Rock
5	Bug
6	Ghost
7	Steel
8	Fire
9	Water
10	Grass
11	Electric
12	Psychic
13	Ice
14	Dragon
15	Dark

B.6 Critical Hits

Critical Hits are determined by a random number compared to either the user's base speed or a range. This depends on whether it is under the RBY or GSC style of computing whether the move is a critical hit or not. I will explain the RBY way first. All skills that do

not have boosted **Critical Hits** has a number thrown out from 0 – 511. **Crit Hit Bonus** moves are chosen from 0 – 64 (although the max is $\frac{255}{256}$). **Power Up** moves boost the next move to be from 0 – 127, again with the max of $\frac{255}{256}$. This number is compared to the base speed of the Pokémon using it, if the number chosen is less, the move is a **Critical Hit**, ℓ in the damage calculation is doubled and all stat modifiers are ignored. In GSC there are 5 **Critical Hit Domains**.

Domain	Range
1	0 – 15
2	16 – 31
3	32 – 63
4	64 – 84
5	85 – 127

Some moves will start out in a boosted domain, but other than that, all of them start out in the first domain. A random number from 0-255 is chosen. If that number falls within the domain, it is a **Critical Hit** and it is applied to the x multiplier in the damage formula.

B.7 Status Effects

Some **Status Effects** have added side effects. Here is a table with each status and what its side effect (if any) is:

Status	Abbreviation	Effect
Sleep	SLP	The Pokémon stay cannot attack for up to 7 turns
Poison	PSN	1/8 of max HP is taken away each turn and 1HP every fourth step in the field
Toxic Poison	PSN	1/16 of max HP is taken away each turn as well as an additional 1/16 for every turn it affects the Pokémon until switched as well as 1HP taken away for every fourth step in the field
Confusion	None	50% of dealing damage to self rather than the opponent for up to 3 turns
Burn	BRN	1/8 of max HP is taken each turn and the attack is reduced by 50%
Freeze	FRZ	Cannot attack at all
Paralyze	PRZ	25% chance of not attacking and speed is reduced by 75%

B.8 Catching

The formula for whether a Pokémon is caught or not is the following:

$$\frac{Value_{Catch}(4HP_{Max} - 2HP_{Cur})}{HP_{Max}} + 1 + \begin{cases} 10, & Status = SLP, FRZ \\ 5, & Status = PRZ, PSN, BRN \\ 0, & \text{else} \end{cases}$$

This value is multiplied by the multiplier of the `Ball` and compared against a random integer from 0–255. If the random number is less than the calculated value, the Pokémon is caught.

B.9 Breeding

Breeding is one of the most complicated and in-depth things of the game for what is accomplished. First of all, compatibility is checked by the following checks:

- The genders are opposite each other (or one is a Ditto-style)
- The Pokémon share at least one `Breeding Factor` numbers and it is not a 0, 1, or an F, or one is Ditto-like, in which case the other Pokémon cannot have a 0 or an F
- Both DV_{Def} and the last 3 bits of DV_{SpC} cannot be the same (see below for why)

Now that the two are checked for compatibility, the species is chosen by the `Hatch Species` of the female (or male in the case of Ditto-like). If the female is a Nidoran-like, there is a 50/50 chance that the species will change to the Nidoran male counterpart. Then a random number from 0-15 is chosen and this becomes DV_{Att} . If this number is less than the `Gender Factor`, it is male, otherwise it is female (or neuter if the `Gender Factor` is 16). Then DV_{Def} and DV_{SpC} are carried down from the opposite gender parent (or the Ditto-like Pokémon if possible). There is a 50/50 chance that DV_{SpC} is offset by 8. These steps make the DV_{Def} and DV_{SpC} check mandatory. Then another random number from 0-15 is set as DV_{Spe} . Moves known at level 5 is a complicated process. First, the Pokémon starts with the moves it would know at level 5. Then, any moves that are on that Pokémon's level-up list and are known by *both* parents, is added to that list, pushing out any previously known moves that do not fit. Next, the TM/HM learn list is scanned and if the father knows the move and if the baby Pokémon is *only* able to learn it through the TM/HM, it is learned. The egg moves list is then scanned and if the father knows one of them, it is added to the move list. Any Pokémon that acts like Ditto or the Nidoran family must be noted in the `pokemod.h` file.