**Introduction**

This set of plugins adds the ability to participate in Pokemon Contests (based on the Generation 3 style of contests) for Pokemon Essentials v20.1. Along with that, it provides ways to create Pokeblocks to improve your Pokemon’s condition for Cool, Beauty, Cute, Smart, and Tough!

The basis for both of these plugins was based on bo4p5687’s code for older versions of Pokemon Essentials, though I did a major rewrite of most of the code. I’ve included credit for their work, and links to their original releases in the meta file for both of these plugins.

As of August 2023, this is in **Beta**. This is only guaranteed to work with base Pokemon Essentials v20.1 and has not gone through testing with other plugins. I want to work on making it more compatible with other plugins, as well as general improvements and bug fixes that may be found, before releasing version 1.0 of these plugins. If you find an issue or an incompatibility you want me to look into, please write it up in the Discussion section of my RelicCastle post for these.

Thanks for downloading!

**Dependencies**

**The Pokeblocks Plugin now requires** [**Berry Core and Dex**](https://reliccastle.com/resources/1424/)**.**

**Basic Setup**

1. **Create a backup of your project.** I’m 96% sure nothing in my plugin will break your project, but it never hurts to back it up before installing a larger plugin, especially since I consider this to be in Beta for now.
2. Install the Pokeblocks and Pokemon Contests plugins by copying their respective folders included into your project’s Plugins folder.
3. Move the files in the Audio, Graphics, and PBS folders included into the same respective folders in your project.
   1. When using the BGM included, **give credit to ENLS’s Pre-Looped Music Library.**
      1. <https://reliccastle.com/resources/663/>
4. Add the items defined in the included Items\_Pokeblocks and Items\_Contest files to your items PBS file.
5. Create a Contest Heart Bubble animation in Database > Animations.
   1. If you downloaded the Pokemon Contests Sample Project, you can just copy the contest heart bubble animation directly from there. Otherwise:
   2. In Animations, duplicate one of the existing bubbles, like Exclaim bubble.
   3. Replace that animation’s Animation Graphic with the provided Overworld contest heart file
   4. Take note of the ID of this animation, to use in Plugins > Pokemon Contests > 001\_Contests\_Map\_Settings
6. Explore the settings in Plugins > Pokeblocks > 000\_Pokeblock\_Settings, and Plugins > Pokemon Contests > 000\_Contests\_General\_Settings, 001\_Contests\_Map\_Settings, 002\_Contests\_Trainer\_Settings.
7. **If you have custom Berries...**
   1. Open the berry\_data file in PBS.
   2. Define each of your custom berries. **Without setting this up, using your custom berries for Pokeblocks will cause errors.**
   3. If you have Plugins > Pokeblocks > 000\_Pokeblock\_Settings > SIMPLIFIED\_BERRY\_BLENDING set to true, consider adding your custom berries to SIMPLE\_POKEBLOCK\_PLUS\_PROBABILITY
8. **If you have custom Natures...**
   1. Open Plugins > Pokeblocks > 000\_Pokeblock\_Settings
   2. In NATURE\_FLAVOR\_PREFERENCES, add each of your custom natures to the Likes and Dislikes groups.
9. **If you have custom Moves...**
   1. Make sure you have compiled data after you’ve installed the plugins. If in doubt, in the Debug menu, run Other Options… > Compile Data.
   2. In the Debug menu, run Pokeblock/Contest Options… > Contest Move Files… > Create Movesx\_Contest PBS File. This will recreate the movesx\_contest file while including all moves in the moves file.
   3. Open the movesx\_contest file in PBS.
   4. Locate all your custom moves in the file, and define contest data for them (see Contest Documentation > Defining Contest Moves section below)
   5. Once done, recompile data (Debug > Other Options… > Compile Data)
   6. You’re all set!
   7. If you add more custom moves later, you can either add the information manually to movesx\_contest, or just run Debug > Pokeblock/Contest Options… > Contest Move Files… > Create Movesx\_Contest PBS File again to add an undefined version of them to movesx\_contest.

**Contest Map Setup**

**Tileset**

1. Create a Contest Hall tileset in Database > Tilesets.
   1. If you downloaded the Pokemon Contests Sample Project, you can just copy the Contest Hall tileset directly from there. Otherwise just make a new one using the Contest Hall tileset graphic

**Maps**

1. In your Project, create 5 new empty maps. Take note of each of their IDs.
2. Save and close your Project.
3. In your Project’s Data folder, find the Map### files for each of the 5 maps you created, and delete them.
4. Rename each of the Map files provided in this plugin folder to match the name of the five Map files you deleted.
5. Copy the renamed Map files in this plugin folder into your Project’s Data folder.
6. Reopen your Project and go back to the 5 Maps. They should now be updated to use the newly copied maps.
7. In each of the Maps’ Properties, make sure to set the tileset to be the Contest Hall tileset you created.
8. Plugins > Pokemon Contests > 001\_Contests\_Map\_Settings, make sure to update specific settings to be the correct map ID
   1. DEFAULT\_LOBBY\_MAP\_ID should be the Contest Lobby Map
   2. ROOM\_MAP\_COORDINATES should include the appropriate Normal, Super, Hyper, and Master Contest Room Maps
9. With the maps created, now you just need to create a way to get into the Contest Lobby from your game! An Outside\_Contest tileset graphic is provided with this plugin that includes the Generation 3 contest hall graphic. You can use that to create a building in your own maps!
10. If you wish to create individual maps for each contest category + rank, now you can just duplicate the 4 non-Lobby maps and edit them. Make sure to also update the ROOM\_MAP\_COORDINATES settings with the new maps!

**Pokeblocks Documentation**

**Functions** **– Berry Blenders**

pbBerryBlender(NPCCount,NPCNames,ForceFailure)

* Description: Starts the Berry Blender minigame to create Pokeblocks
* Arguments:
  + NPCCount
    - Optional. Set to an integer 0-4
    - 0: Player plays by themself. If this isn't set, 0 is the default.
    - 1, 2, or 3: How many NPCs you wish to also play.
    - 4: The NPC is a Berry Master, that uses rarer berries.
  + NPCNames
    - Optional. Set to an array of names corresponding to the NPCs that will play
    - If this isn't set, it will use random names as defined in 000\_Pokeblock\_Settings
    - Example: set ["Ash","May","Dawn"] for the 3 NPCs to have those names.
  + ForceFailure
    - Optional. Set to true or false
    - If this isn't set, it defaults to false.
    - If true, the Pokeblock created will always be a Black Pokeblock
* Examples:
  + pbBerryBlender => Runs the minigame with just the player
  + pbBerryBlender(2) => Runs the minigame with 2 NPCs
  + pbBerryBlender(1,["Max"],true) => Runs the minigame with 1 NPC named Max, and will always give a Black Pokeblock
  + pbBerryBlender(0,nil,true) => Runs the minigame with just the player, and will always give a Black Pokeblock

pbBerryBlenderSimple

* Description: Starts the simplified Berry Blender scene to create Pokeblocks. There are no arguments for this function.

**Functions – Pokeblocks**

These functions are available if you wish use them outside of the items or minigame. For instance, if you want an NPC to give you a free Pokeblock, use pbGainPokeblock. Or, if you want an NPC to feed your Pokemon a special Pokeblock, you can call pbFeedPokeblock.

pbGainPokeblock(Pokeblock)

* Description: Give the player a Pokeblock
* Arguments:
  + Pokeblock
    - Required. A Pokeblock object (see pbCreatePokeblock below)

pbFeedPokeblock(Pokeblock)

* Description: Feed a Pokeblock to a party Pokemon (opens a scene to choose which one)
* Arguments:
  + Pokeblock
    - Required. A Pokeblock object (see pbCreatePokeblock below)

pbRemovePokeblock(Pokeblock)

* Description: Takes a Pokeblock from the player
* Arguments:
  + Pokeblock
    - Required. A Pokeblock object (see pbCreatePokeblock below)

pbCreatePokeblock(Color,Flavor,Plus,Feel)

* Description: Creates a Pokeblock object, which is used in the 3 above functions.
* Arguments:
  + Color
    - Required. The color of the Pokeblock, as a symbol
    - Examples: :Red or :Rainbow
  + Flavor
    - Required. An array for the flavor profile of the Pokeblock in the form [0,0,0,0,0]
    - Each value must be a positive integer or 0. The numbers represent each of the contest stats in this order: [Cool,Beauty,Cute,Smart,Tough]
  + Plus
    - Optional. Set to true or false
    - If this isn't set, it defaults to false.
    - If true, the Pokeblock will be a + variety.
  + Feel
    - Optional. A positive integer for the feel of the Pokeblock
    - This is unused for the simplified Generation 6 mode, so if using that this will never need to be set.
* Examples:
  + pbCreatePokeblock(:Red,[5,0,0,0,0]) => creates a basic Red Pokeblock
  + pbCreatePokeblock(:Blue,[0,15,0,0,3],true) => creates a Blue Pokeblock+ that also increases Toughness a bit
  + pbCreatePokeblock(:Rainbow,[10,10,12,10,9],false,25) => creates a Rainbow Pokeblock with a feel of 25.

**Functions – Item Scenes**

These functions are available if you wish use them outside of the items. For instance, if for some reason you want the player to only access the Pokeblock Case through an event, you can call pbPokeblockCase within that event.

pbPokeblockCase

* Description: Opens the Pokeblock Case. Used by the POKEBLOCKCASE and POKEBLOCKKIT items. There are no arguments for this function.

pbPokeblockCondition

* Description: Opens the Pokemon Condition screen to review the your party's contest stats. Used by the POKEBLOCKCONDITION and POKEBLOCKKIT items. There are no arguments for this function.

pbPokeblockKit

* Description: Opens the Pokeblock Kit, which allows access to the Pokeblock Case, Berry Blender, and Pokemon Condition screen. There are no arguments for this function.

**Contest Documentation**

**Viewing Contest Info in Pokemon Summary**

If the Switch as defined in 000\_Contests\_General\_Settings > CONTEST\_INFO\_IN\_SUMMARY\_SWITCH is set to true, the Player can view a Pokemon’s contest info in Pokemon Summary. This includes Conditions and Move Information. This is toggled by using the SPECIAL key (default is the D Key) while viewing the SKILLS or MOVES tab in a Pokemon’s Summary, or while teaching a Pokemon a move.

**Defining Contest Moves**

Contest Moves are defined in the movesx\_contest PBS file. Keeping this as a separate file makes it easier to duplicate contest-specific information across moves. The structure is similar to [regular moves](https://essentialsdocs.fandom.com/wiki/Defining_a_move).

|  |  |  |
| --- | --- | --- |
| **Data** | **Description** | **Example** |
| [ID] | ID of the move. Must match a move as described in the moves PBS file. | [MEGAHORN] |
| ContestType | A Contest Type ID as defined in 004\_Contests\_Trainer\_Types\_Definitions | COOL |
| ContestHearts | Integer from 0-8. How much appeal is gained from using the move. | 4 |
| ContestJam | Integer from 0-8. How much the move will startle/jam the other Pokemon. | 3 |
| ContestFunctionCode | A Function ID as defined in 010\_Move\_Functions | StartleOthersMissNextTurn |
| ContestFlags | A comma-separated list of labels applied to the move which can be used to make it behave differently. The only existing flag is:   * CannotBeUsed – The move cannot be used in Contests. | CannotBeUsed |
| ContestDescription | The move’s contest description. This will be overwritten by the ContestFunctionCode description if 000\_Contests\_General\_Settings > GET\_MOVE\_DESCRIPTIONS\_FROM\_FUNCTION is true | Startles all other Pokémon. User cannot act in the next turn. |

**Defining Contest Trainers**

You can define specific trainers to appear in contests. Refer to the 004\_Contests\_Trainer\_Types\_Definitions file for examples of trainers already created from existing games, as well as further explanation of variables for you to create your own!

**Defining Contest Types**

You can make changes to the default contest types. These are set in the 004\_Contests\_Trainer\_Types\_Definitions file. You must have 5 defined contest types. I’d recommend only changing the :name or :long\_name values for these types, as those are the only ones that will be shown to Players. For instance, instead of “Smart” and “Smartness”, as it was in Generation 3, you can change the :SMART ContestType’s :name to be “Clever” and :long\_name to be “Cleverness” to match Generation 6.

**Functions – Contests**

pbPokemonContest(rank: <val>, category: <val>, pokemon: <val>)

* Description: Starts the workings of a Pokemon Contest. It begins by prepping the setup by asking the user to select the Rank, Category, and Pokemon they will use. If any of those are passed through as arguments, then the player does not select that value as it’d be set by that argument.
* Arguments (order of these do not matter):
  + rank: <val>
    - Optional. Replace <val> with the rank of the Pokemon Contest.
    - You can use a string for the name (“Normal”, “Super”, “Hyper”, or “Master”) or an integer 0-3 to represent it (0 = Normal, 1 = Super, 2 = Hyper, 3 = Master).
  + category: <val>
    - Optional. Replace <val> with the category of the Pokemon Contest.
    - You can use a string for the name as defined in 004\_Contests\_Trainer\_Types\_Definition or an integer 0-4 to represent it by its icon\_idex (By default, 0 = Cool, 1 = Beauty, 2 = Cute, 3 = Smart, 4 = Tough).
  + pokemon: <val>
    - Optional. Replace <val> with a valid Pokemon the player will use.
    - You can have the player manually select a Pokemon yourself this way. Otherwise, you can just have them select one though the function itself (by omitting this argument).
* Examples:
  + pbPokemonContest=> Starts a contest, and the player will choose the Rank and Category of the contest and which Pokemon to use for it.
  + pbPokemonContest(rank: 0, category: 0) => Starts a Normal Cool contest, and the player will choose the Pokemon to use for it.
  + pbPokemonContest(category: “Master”) => Starts a Master contest, and the player will choose the Category of the contest and which Pokemon to use for it.

pbShowLastContestWinner(rank: <val>, category: <val>)

* Description: Shows off the last Pokemon, and their Trainer’s name, that won a contest overall or of the specified Rank and/or Category. It shows the Pokemon’s sprite, Contest Rank and Category, Trainer name, and Pokemon name. If there is no winner to show, it does nothing.
  + Returns **true** if there was a winner to show off, otherwise returns **false**. This allows you to say something like “It’s an empty poster.” If there wasn’t a Pokemon to show off, by putting pbShowLastContestWinner as a script in a Conditional Branch.
* Arguments (order of these do not matter):
  + rank: <val>
    - Optional. Replace <val> with the rank of the Pokemon Contest. If you include a rank, it will show the last contest winner of that rank (either overall, or for a specific category if the category argument is included)
    - You can use a string for the name (“Normal”, “Super”, “Hyper”, or “Master”) or an integer 0-3 to represent it (0 = Normal, 1 = Super, 2 = Hyper, 3 = Master).
  + category: <val>
    - Optional. If you include a category, it will show the last contest winner of that category. If you do not include a category, it will show the last overall contest winner.
    - You can use a string for the name as defined in 004\_Contests\_Trainer\_Types\_Definition or an integer 0-4 to represent it by its icon\_idex (By default, 0 = Cool, 1 = Beauty, 2 = Cute, 3 = Smart, 4 = Tough).
* Examples:
  + pbShowLastContestWinner => Shows the icon and name (including the trainer) of the Pokemon who won the last contest that occurred, no matter the rank or category.
  + pbShowLastContestWinner(category: 2) => Shows the icon and name (including the trainer) of the Pokemon who won the last overall Cute contest that occurred
  + pbShowLastContestWinner(rank: “Hyper”) => Shows the icon and name (including the trainer) of the Pokemon who won the last overall Hyper Rank contest that occurred
  + pbShowLastContestWinner(rank: 3, category: “Tough”) => Shows the icon and name (including the trainer) of the Pokemon who won the last Master Rank Tough contest that occurred

pbGetLastContestWinner(rank: <val>, category: <val>, namesOnly: <val>)

* Description: Similar to pbShowLastContestWinner, except it doesn’t show the winner off. It instead returns an array consisting of the following:
  + [Trainer that won (either Player or ContestTrainer class), Pokemon that won (Pokemon class), Rank (integer), Category (integer)]
  + If there is no winner, returns nil.
* Arguments (order of these do not matter):
  + rank: <val>
    - Optional. Same as pbShowLastContestWinner
  + category: <val>
    - Optional. Same as pbShowLastContestWinner
  + namesOnly: <val>
    - Optional. Set to true or false
    - If false or not defined, nothing changes.
    - If true, will return the same style of array for the winner, but will only include the names of the Trainer, Pokemon, Rank and Category. For example:
      * [“Ash”, “Pikachu”, “Super”, “Cool”]

**Variables – Contests**

$PokemonGlobal.nextContestTrainerOne, $PokemonGlobal.nextContestTrainerTwo, or $PokemonGlobal.nextContestTrainerThree

* Description: If you set one of these variables before running a contest, then it will be that trainer in the next contest (if they meet the criteria of the contest). This is useful for Event Trainers, such as a region Champion. After a contest occurs, these values always get reset, so it will need to be reset.
* Possible values:
  + Symbol of a ContestTrainer id (as defined in 004\_Contests\_Trainer\_Types\_Definition)
  + Array of Symbols of ContestTrainer ids; the array needs to be a length of 5, as it will represent the different Category values available for contests. If you don’t want to set a trainer for any of the 5 Categories, simply set nil for that Category.
* Examples:
  + $PokemonGlobal.nextContestTrainerOne = :NormalChance => If you set this, then Trainer One in the next Normal Contest will be the trainer :NormalChance (it will only be true in a Normal Contest, because that’s the rank that trainer is defined for).
  + $PokemonGlobal.nextContestTrainerThree = [:Sp\_MasterCoolLisia,:Sp\_MasterBeautyLisia,:Sp\_MasterCuteLisia,:Sp\_MasterSmartLisia,:Sp\_MasterToughLisia] => If you set this, then Trainer Three in the next Master Contest will be a Lisia trainer, depending on which Category the contest will be (it will only be true in a Master Contest, because that’s the rank that these trainers are defined for).

**Functions – Contest Stats**

New stats in the GameStats ($stats) class have been added to keep track of the number of contests won or participated in. I’ve provided functions to easily check these values to be used in events.

pbNumberContestsParticipated(rank: <val>, category: <val>)

* Description: Returns the number of contests you’ve participated in overall or of the specified rank and/or category. Shortened version that can be used:
  + pbNumContestsPart(r: <val>, c: <val>)
* Arguments (order of these do not matter):
  + rank: <val>
    - Optional. Replace <val> with the rank of the Pokemon Contest.
    - You can use a string for the name (“Normal”, “Super”, “Hyper”, or “Master”) or an integer 0-3 to represent it (0 = Normal, 1 = Super, 2 = Hyper, 3 = Master).
  + category: <val>
    - Optional. Replace <val> with the category of the Pokemon Contest.
    - You can use a string for the name as defined in 004\_Contests\_Trainer\_Types\_Definition or an integer 0-4 to represent it by its icon\_idex (By default, 0 = Cool, 1 = Beauty, 2 = Cute, 3 = Smart, 4 = Tough).
* Examples:
  + pbNumberContestsParticipated => Returns the total number of contests you’ve participated in, no matter the rank or category.
  + pbNumberContestsParticipated(category: 2) => Returns the total number of Cute contests you’ve participated in, no matter the rank.
  + pbNumContestsPart(r: “Hyper”) => Returns the total number of Hyper Rank contests you’ve participated in, no matter the category.
  + pbNumContestsPart(r: 3, c: “Tough”) => Returns the total number of Master Rank Tough contests you’ve participated in.

pbNumberContestsWon(rank: <val>, category: <val>)

* Description: Returns the number of contests you’ve won overall or of the specified rank and/or category. Shortened version that can be used:
  + pbNumContestsWon(r: <val>, c: <val>)
* Arguments (order of these do not matter):
  + rank: <val>
    - Optional. Replace <val> with the rank of the Pokemon Contest.
    - You can use a string for the name (“Normal”, “Super”, “Hyper”, or “Master”) or an integer 0-3 to represent it (0 = Normal, 1 = Super, 2 = Hyper, 3 = Master).
  + category: <val>
    - Optional. Replace <val> with the category of the Pokemon Contest.
    - You can use a string for the name as defined in 004\_Contests\_Trainer\_Types\_Definition or an integer 0-4 to represent it by its icon\_idex (By default, 0 = Cool, 1 = Beauty, 2 = Cute, 3 = Smart, 4 = Tough).
* Examples:
  + pbNumberContestsWon => Returns the total number of contests you’ve won, no matter the rank or category.
  + pbNumberContestsWon(category: 2) => Returns the total number of Cute contests you’ve won, no matter the rank.
  + pbNumContestsWon(r: “Hyper”) => Returns the total number of Hyper Rank contests you’ve won, no matter the category.
  + pbNumContestsWon(r: 3, c: “Tough”) => Returns the total number of Master Rank Tough contests you’ve won.

pbGiveFanItem(Rank)

* Description: Gives the player an item as defined in 015\_Fans > FAN\_ITEMS based on the rank you include as the argument. This is meant to use in conjunction with the new Contest Stats functions in events. For example, you can set up an event that checks that if you have won at least 5 Super Rank contests, the event gives you an Item from a selection of options set in FAN\_ITEMS.
* Arguments:
  + Rank
    - Optional. Set to a Rank value. You can use a string for the name (“Normal”, “Super”, “Hyper”, or “Master”) or an integer 0-3 to represent it (0 = Normal, 1 = Super, 2 = Hyper, 3 = Master).
    - If this isn't set, 0 (Normal) is the default.

**Compatibility Info**

* This plugin overwrites PokemonSummary\_Scene > pbScene, pbMoveSelection, pbChooseMoveToForget in order to allow toggling showing contest information in Pokemon Summary. This will likely result in incompatibilities. This will be worked on to be compatible in the future.
* Move animations for contests had to be created by duplicating existing animations code from base Essentials. If changes to move animations are done for the regular/battle version, those changes may have to be implemented again for the contest version. Note, some move animations may not be optimized for contests.
* Known Incompatibilities:
  + As of now, there are known incompatibilities with v21 of Essentials. These will be worked on in the future.