

Death overhaul - guide de l'évent-making

Activer les features du mod :

Choix offert au joueur VS choix imposé par le Maker

Une fois le mod bien installé, il faudra activer les features principales in-game grâce à la manipulation des switches et variables que vous avez choisis en event-making. Comme une image vaut 1000 mots, voici un exemple d'événement de départ :

Liste des commandes d'événement :

```
<> Message : Bienvenue dans Pokémon XYZ. Es-tu un garçon ou une fille ?
<> Choix : Garçon, Fille
: Si [Garçon]
  <> Script : $pokemon_party.trainer.set_gender(false)
  <> Script : gp.set_appearance_set("Heros-G")
  <>
: Si [Fille]
  <> Script : $pokemon_party.trainer.set_gender(true)
  <> Script : gp.set_appearance_set("Heros-F")
  <>
: Fin Choix
<> Message : Quel est ton nom ?
<> Entrer un nom : Héros Silvynis, 13 caractères
<> Commentaire : Le maker laisse le choix au joueur pour le mode Nuzlocke
: : avec les clauses de zone et de mort jumelées.
<> Message : Activer le Nuzlocke ?
<> Choix : Oui, Non
: Si [Oui]
  <> Interrupteur [0164: Area Clause] = Activé
  <> Variable [0124: Mode Nuzlocke] = 0
  <> Variable [0125: Backup Nuzlocke] = 0
  <>
: Si [Non]
  <>
: Fin Choix
<> Message : Tes réglages sont terminés, ton aventure Pokémon commence !
<> Commentaire : Le maker impose dans son jeu le poison à l'ancienne.
: : (Pas de choix proposé au joueur.)
<> Interrupteur [0169: Old School Poison] = Activé
<> Téléporter l'équipe : [020: Bourg-Palette] (X: 019,Y: 016), Bas, Pas de fondu
<>
```

Dans cet exemple, après la demande du sexe et du nom :

- Le Maker laisse le choix au joueur pour l'activation du mode Nuzlocke (avec ses 2 règles majeures jumelées).
- Mais impose dans son jeu le Poison à l'ancienne. (L'interrupteur sera activé sans laisser le choix au joueur.

Jumelage des paramètres

Bien entendu, Death Overhaul propose bien plus d'options que dans cet exemple. En attendant qu'un vrai menu de paramétrage soit codé, vous pouvez :

- Soit demander au joueur ce qu'il souhaite pour chaque réglage proposé indépendamment.
- Soit jumeler certains paramètres. (Cela permet de simplifier les choix proposés.)

Deux méthodes permettent le jumelage :

– Soit en changeant en même temps dans l'événement tous les interrupteurs et variables liés. Obligatoire lorsqu'un interrupteur est couplé à une variable. (Cas dans l'exemple ci-dessus pour l'AreaClause et la Mortalité.)

– Soit en faisant pointer les variables ou interrupteurs jumelés sur le même numéro dans **90000 Death overhaul - Gestion cycle vie-mort.rb**. (Variables à gérer séparément des interrupteurs.)

La première méthode est conseillée si vous souhaitez conserver la souplesse du choix pour l'avenir. La seconde méthode est à privilégier si vous êtes sûr que plusieurs clauses resteront toujours liées en elles dans votre jeu.

Backup des paramètres choisis par le joueur

Si durant votre scénario, vous prévoyez un changement forcé et temporaire d'un paramètre que le joueur a eu l'occasion de choisir, il faudra dédier en plus une variable (si c'est une variable) ou un interrupteur (si c'est un interrupteur) de backup.

Liste des commandes d'événement :

```
<> Message : Salut, veux-tu activer ou désactiver le Nuzlocke ?
<> Choix : Activer, Désactiver
: Si [Activer]
  <> Interrupteur [0164: Area Clause] = Activé
  <> Variable [0124: Mode Nuzlocke] = 2
  <> Variable [0125: Backup Nuzlocke] = 2
  <> Message : Nuzlocke activé !
  <>
: Si [Désactiver]
  <> Interrupteur [0164: Area Clause] = Désactivé
  <> Variable [0124: Mode Nuzlocke] = 0
  <> Variable [0125: Backup Nuzlocke] = 0
  <> Message : Nuzlocke désactivé !
  <>
: Fin Choix
<>
```

Ici, un exemple où :

- Le Maker propose le choix au joueur d'activer ou de désactiver le mode Nuzlocke.
- Avec les règles d'Area Clause et de mortalité jumelées.
- Avec un backup du réglage de la mortalité. (Ainsi, si au premier combat, le joueur sera amené à affronter un Giratina niveau 70 avec son Starter niveau 5 alors qu'il a activé le Nuzlocke au moment de créer son personnage, la mortalité Nuzlocke pourra momentanément être désactivée pour ce combat perdu d'avance.)

- Si dans **00500 Death overhaul - master - Gestion cycle vie-mort**, le Maker a fait pointer les variables **DupesClause** et **ShinyClause** sur le même numéro que **AreaClause**, (soit le n°164 pour ce screen), alors ces trois clauses seront automatiquement jumelées.

Feature de blocage des objets en combat

J'ai utilisé une unique variable pour permettre un maximum de souplesse :

Fonctionnalité n°1 : Possibilité de proposer au joueur une option d'interdiction des objets au combat (uniquement contre les dresseurs ou sur tous les combats, avec ou sans autorisation de la poche des Balls).

Fonctionnalité n°2 : Possibilité pour le maker de temporairement bloquer l'accès au sac en combat. Plusieurs textes pour notifier cette interdiction sont inclus de série dans le mod (traduits en EN/FR/ES dans le CSV 100100).

La variable dédiée (**NoItemOnBattle** à déterminer dans le script maître « **00500 Death overhaul - master - Gestion cycle vie-mort** ») suffira si vous utilisez qu'une des deux fonctionnalités. Si vous combinez les deux, dédiez une seconde variable de votre choix qui servira de backup (à manipuler via événement-making).

Le mode correspond à la valeur de la variable « **NoItemOnBattle** »

Mode	Accès du sac en combat sauvage	Accès du sac en combat de dresseur
0	Sans restriction (Vanilla)	
1	Sans restriction	Bloqué avec message stipulant que le joueur l'a désactivé contre les dresseurs.
2	Autorisé, mais restreint à la poche des Pokéballs	Bloqué avec message stipulant que le joueur l'a désactivé contre les dresseurs.
3	Autorisé, mais restreint à la poche des Pokéballs	
4	Bloqué avec message stipulant que le joueur l'a désactivé sur tous les combats.	
5	Bloqué avec le message : L'utilisation d'objets n'est pas autorisée dans ce combat.	
6	Bloqué avec le message : L'utilisation d'objets n'est pas autorisée dans les combats en ligne.	
7	Bloqué avec le message : L'utilisation d'objets n'est pas autorisée dans la Tour de Combat.	
8	Bloqué avec le message : L'utilisation d'objets en combat est interdite en match officiel.	
9	Bloqué avec le message : L'utilisation d'objets en combat est interdite en match d'Arène.	
10	Bloqué avec le message : L'utilisation d'objets est interdite en combat de la Ligue Pokémon.	
11	Bloqué avec le message : L'utilisation d'objets en combat est interdite au Championnat de Keltios.	
12	Bloqué avec un message servant de placeholder (à personnaliser dans le CSV 100100)	

- Laisser sur une variable qui pointe à zéro pour ne pas utiliser cette feature dans votre fan-game.

- Les modes 1 à 4 sont destinés au joueur avec chacun leurs nuances. Je conseille au Maker de proposer 1 ou 2 des 4 modes (en plus du 0) s'il souhaite proposer un choix au joueur. 0 + 1 + 2 est la combinaison recommandée. (Ou 0 + 1 ou 0 + 2 si que deux choix.)

- Si l'accès à la poche des balls est possible, aucun message ne sera affiché à l'ouverture du sac.

- Le mode 3 peut être utile si votre fan-game souhaite interdire l'utilisation des objets en combat, mais permettre la capture des Pokémon des dresseurs. (Ex : un scénario où le héros est un voleur de Pokémon.)

- Il est déconseillé de proposer le mode 4 au joueur, car le blocage total du sac bloquera aussi l'accès aux Pokéballs.

- Les modes 5 et plus sont destinés à des moments scénarisés dans le jeu et seront donc à activer manuellement via événement-making au moment voulu par le maker (ex : juste avant un combat d'Arène). Si vous avez offert au joueur la possibilité de choisir lui-même un mode de restriction, pensez à faire un backup de **NoItemOnBattle** via une autre variable avant de la modifier vous-même, puis restaurez-la à son ancienne valeur une fois le moment scénarisé terminé grâce au backup. (Si un tel choix n'est pas offert au joueur, restaurer **NoItemOnBattle** à 0 suffira, la variable backup n'est pas nécessaire.)

Conflits de mods possibles :

Pour qu'elle puisse fonctionner, cette feature a modifié la constante **POCKETS_PER_MODE** du script [01400 GamePlay\00300 Bag\00300 Bag.rb](#).

Si vous avez modifié les poches accessibles de votre sac PSDK, pensez à éditer la constante dans « [14330 Only Pokeballs accessible](#) » afin de la faire correspondre à vos besoins, puis rajoutez une ligne « ball: [2] » à l'array.

Modification temporaire d'un paramètre du mod par le Maker

Si vous des raisons scénaristiques, vous souhaitez attribuer temporairement une valeur précise à l'un des interrupteurs ou l'une des variables du mod, mais que le joueur a lui aussi un accès au paramètre qui y est lié, il vous faudra créer un interrupteur ou une variable « backup » pour chaque cas correspondant.

Avant la partie scénarisée (par exemple un combat avec objets interdits), faites ces deux commandes d'événement-making :

Votre backup variable x (ou interrupteur x) = variable x (ou interrupteur x)

Votre variable x (ou interrupteur x) = valeur déterminée par vous.

Une fois la partie scénarisée passée, restaurez la valeur choisie par le joueur en faisant la commande inverse.

Votre variable x (ou interrupteur x) = backup variable x (ou interrupteur x)

Utile par exemple pour :

- La tour de combat. Les clauses suivantes sont alors à activer :
 - Variable Difficulté = 0 (normale) (les adversaires doivent tous rester niveau 50)

– Variable « Objets interdits en combat » = 7 (Non autorisé avec message d’erreur : objets interdits dans la Tour de combat.)

– Variable « Mortalité » = 0 en l’absence de « clonage » de l’équipe du joueur (La tour de combat est comme un combat Link, les Pokémon ne meurent pas.)

- Lorsque le joueur a choisi le mode Nuzlocke alors qu’un combat que le joueur doit perdre est prévu dans le scénario (ou que le Nuzlocke ne doit pas commencer tout de suite).

- Un combat scénarisé qui doit illustrer la mortalité des Pokémon.

Comme une image vaut mille mots, voici une illustration avec la variable de mortalité fixée à 0 le temps d’un combat :

Liste des commandes d’événement :

```
<> Message : Un combat ? Attention, je suis niveau 50 !
<> Choix : Oui, Non
: Si [Oui]
  <> Commentaire : Sécurité au cas où le joueur a tous ses Pokés KO ou morts.
  <> Condition : Script : $pokemon_party.pokemon_alive ==0
    <> Message : Tu n'as pas de Pokémon apte à se battre.
    <> Étiquette : Aucun Pokémon
  <>
  : Fin Condition
  <> Message : C'est parti pour le combat ! Comme j'ai une équipe surpuissante,
  : je désactive le temps de notre match la mort de tes Pokémon.
  <> Variable [0125: Backup Nuzlocke] = Variable [0124: Mode Nuzlocke]
  <> Variable [0124: Mode Nuzlocke] = 0
  <> Variable [0008: ID Dresseur (combat)] = 4
  <> Combat : Combat de Dresseur
  : Si Victoire
    <> Message : Bravo pour ta victoire !
  <>
  : Si Fuite
    <> Message : Tu as voulu mettre fin au combat. Soit, je respecte ton choix.
  <>
  : Si Défaite
    <> Message : Va soigner tes Pokémon.
    <> Étiquette : Aucun Pokémon
  <>
  : Fin Combat
  <> Variable [0124: Mode Nuzlocke] = Variable [0125: Backup Nuzlocke]
  <>
: Si [Non]
  <>
: Fin Choix
<>
```

Si le joueur n’a de base pas accès au réglage de ces paramètres, la variable ou l’interrupteur de backup n’est pas nécessaire. Les commandes qui suivent suffiront :

Votre variable x (ou interrupteur x) = valeur déterminée par vous

(Phase scénarisée nécessitant ce réglage spécifique...)

Votre variable x (ou interrupteur x) = ancienne valeur (le plus souvent “0” pour une variable ou “désactivé” pour un interrupteur)

Ressusciter un Pokémon sélectionné dans l'équipe avec test s'il est bien mort

Variante plus propre de l'utilisation de ces trois lignes :

```
$pokemon_party.actors[0].status = 0  
$pokemon_party.actors[0].hp = $pokemon_party.actors[0].max_hp  
$pokemon_party.actors[0].death_c = nil
```

...en utilisant la commande script « **appel_menu_equipe** » de PSDK pour proposer une résurrection à un Pokémon. (“[0]” est remplacé par “[gv[26]]” – attention, à 4 endroits en tout !)

Liste des commandes d'événement :

```
<> Message : Veux-tu ressusciter un de tes Pokémon ?  
<> Choix : Oui, Non  
: Si [Oui]  
  <> Attendre : 1 frames  
  <> Script : appel_menu_equipe  
  <> Condition : Variable [0026: Var temporaire 1] >= 0  
    <> Condition : Script : $pokemon_party.actors[gv[26]].mort?  
      <> Script : $pokemon_party.actors[gv[26]].status = 0  
        :      : $pokemon_party.actors[gv[26]].hp = $pokemon_party.actors[gv[26]].max_hp  
        :      : $pokemon_party.actors[gv[26]].death_c = nil  
      <> Commentaire : Insérer ici bruitages et effets de votre choix.  
      <> Message : Ton Pokémon est revenu à la vie !  
      <>  
    : Sinon  
      <> Condition : Script : $pokemon_party.actors[gv[26]].mort_d?  
        <> Message : Trop tard, son âme est partie.  
        <>  
      : Sinon  
        <> Message : Ton Pokémon n'est pas mort.  
        <>  
      : Fin Condition  
    <>  
  : Fin Condition  
  <>  
: Fin Condition  
<>  
: Si [Non]  
  <>  
: Fin Choix  
<>
```

Si vous souhaitez que la résurrection soit aussi possible sur les Pokémon « morts définitivement », mettez comme condition :

```
$pokemon_party.actors[gv[26]].mort? or $pokemon_party.actors[gv[26]].mort_d?
```

Au lieu de seulement :

```
$pokemon_party.actors[gv[26]].mort?
```

(Et dans la partie « **Sinon** », un simple message “Ton Pokémon n’est pas mort” suffira.)

Si l'acte de résurrection a un coût pour le joueur, insérer la condition testant si le joueur a bien le prix demandé avant les commandes de résurrection (qui ne seront exécutées que si la condition est bonne), puis déduisez le prix demandé de l'inventaire du joueur.

Événements de la renaissance sous forme d'œuf des Légendaires :

En fin du script « 00500 Death overhaul - master - Gestion cycle vie-mort », il y a une « `def egg_g(pokemon)` ». Elle permet d'activer à distance l'interrupteur local A d'un événement existant : l'un de ceux (choisi aléatoirement) paramétré dans le double array « `EGG_POS` ». (Il est bien entendu possible de ne paramétrer qu'un seul spot pour un légendaire donné : ça sera forcément celui-ci.)

Pour paramétrer votre événement de spot d'apparition de l'œuf de Légendaire, il faudra créer un événement qu'il comporte 2 onglets (et relever son ID d'événement pour le renseigner en EGG-POS).

Onglet 1 (aucune condition)

Apparence : (aucune)

Commandes :

Event vide, ou si vous le souhaitez un texte du genre : « Il semblerait que Suicune soit passé / soit né ici. »

Onglet 2 (sur Interrupteur local A)

Apparence : l'œuf du Pokémon légendaire / activation sur touche action

Commandes :

(Insérer ici le message et le bruitage de votre choix.)

Script : `add_egg(245)`

Interrupteur local A : désactiver