

# Documentation de JDR-Bot

## Présentation de JDR-Bot :

JDR-Bot vous permet de jouer à différents jeux (ou scénarios), de type Jeux de rôle, Histoires dont vous êtes le héros (ou AVH), Escape Game, JDR textuel (façon Colossal Cave). Il permet aussi de jouer à des jeux très différents, comme des combats au tour à tour, des casinos, etc. Il permet de lancer des pièces ou des dés.

Pour tester ou plus d'information, rejoignez-nous sur discord : <https://discord.gg/Z63DtVV>

Vos retours (avis, bugs, suggestions) seront lus avec une grande attention.

Vous pouvez aussi aller voir le github du bot : <https://github.com/Cyril-Fiesta/jdr-bot>

Les scénarios de bases, les images et les sons utilisés sont hébergés sur :

<http://cyril-fiesta.fr/jdr-bot/scenarios/>

Cependant, il vous est possible de changer l'url (via la commande **!lien\_url**) si vous désirez jouer avec vos propres scénarios.

Pour ajouter le bot sur votre serveur : [https://discord.com/api/oauth2/authorize?](https://discord.com/api/oauth2/authorize?client_id=521137132857982976&permissions=70671424&scope=bot)

[client\\_id=521137132857982976&permissions=70671424&scope=bot](https://discord.com/api/oauth2/authorize?client_id=521137132857982976&permissions=70671424&scope=bot)

N'hésitez pas à me contacter sur discord en cas de questions, soucis ou suggestions !

Pour fonctionner, le bot aura besoin d'un canal textuel dont le nom contient « jdr-bot », et d'un canal vocal dont le nom exacte est « JDR-Bot ».

JDR-Bot est en constante évolution, il peut donc y avoir des bugs ou des choses à améliorer. En cas de mise à jour ou de maintenance, vous serez averti à l'avance par le bot, afin d'avoir le temps de terminer vos parties en cours.

Certains des scénarios actuels sont des scénarios de tests et de démonstration. Les scénarios évolueront aussi, d'autres feront leur apparition.

Vous pouvez écrire et proposer vos propres scénarios et, si vous le souhaitez, me les soumettre pour les inclure à la liste des scénarios de bases.

Merci de me signaler les éventuels problèmes, vos questions, et vos retour via le serveur discord Make&Play.

## Soutiens et Partenaires :

JDR-Bot continue de se développer grâce à différents soutient et partenaires tels que :

- [L'union des Rôlistes](#) (Leur discord est très accueillant ! Et des auteurs travaillent actuellement sur des scénarios pour JDR-Bot)
- [VirtualJDR](#) (discord très accueillant aussi)
- [Ludinn](#) (vidéo twitch, youtube, discord)
- [Moiki](#) : Site permettant d'écrire des fictions interactives, d'y jouer et de les exporter au format de JDR-Bot
- [GameFlow](#) : Éditeur des livres-jeux « Ma Première aventure » qui m'a autorisé à adapté « En Quête du Dragon » sur JDR-Bot
- Et d'autres partenaires divers (discord, forum, site, etc.)

Un grand merci à tous les testeurs, aux partenaires, aux auteurs et à ceux qui suivent le bot et son évolution.

## Licences :

L'ensemble du bot (le projet, les scénarios de bases, le fonctionnement, ...) sont sous licence Creative Commons (CC BY-SA 4.0) ( <https://creativecommons.org/licenses/by-sa/4.0/> )

Les images et sons utilisés dans les scénarios proviennent de différentes sources. Ils sont sous différentes licences autorisant leur utilisation dans des projets personnels ou commerciaux. Merci de respecter le travail de leurs auteurs, et de ne pas les utiliser en dehors des scénarios du bot.

Les scénarios sont la propriété de leurs auteurs/éditeurs respectifs. S'ils sont sous licence, elle sera précisée en commentaire dans le scénario.

## Les auteurs sur JDR-Bot :

De part le temps qu'ils ont investi pour mettre a disposition leurs fictions/jeux/univers/etc. à tout les utilisateurs de JDR-Bot, je tiens à remercier chaque auteur. Pour les œuvres libres de droit (ou avec autorisation de l'auteur) que j'ai adapté sur JDR-bot, le lien de la source est présent dans le scénario directement.

- **Vharung** (<http://leschroniquesdeliber.fr/>) : *Liber*
- **Kaelhem** (<https://moiki.fr>) : Exporter Moiki > JDR-Bot, *dino-park, journee-de-merde, le-pendu*
- **Obat** (Le Prisme des Créateurs : <https://discord.gg/CQMpjNF>) : *aventure\_spline* (qui deviendra par la suite une grosse aventure avec plusieurs histoires, battle royale, mini-jeux, etc.), Live de parties et créations de scénarios de JDR-Bot, et plusieurs exemples et systèmes (sauvegarde, compétence)
- **Doublure Stylo** (<https://www.doublurestylo.com>): *une-galerie-pour-sulimo*
- **Onirie** : *Le-Sept*

# Comment Jouer :

Les commandes du bot commencent par « **j!** ». Il faut au préalable créer un salon textuel et un salon vocal, nommés « **JDR-Bot** ».

Pour jouer, il faut charger un scénario avec « **j!jouer** », puis utiliser les commandes de jeu :

- **j!lien\_url** Affiche le(s) ftp(s) où sont hébergés les scénarios (ou **j!url**, **j!lien** ou **j!link**)
  - **j!lien\_url ajouter votre\_url** Ajoute l'url dans la liste d'urls du bot
  - **j!lien\_url retirer votre\_url** Retire l'url de la liste d'urls du bot
  - **j!lien\_url base** Ne laisse que l'url de base et celles de la descriptions
- **j!liste\_scenarios** Affiche la liste de scénarios disponible (ou **j!scripts** ou **j!scenarios**)
- **j!jouer nom\_scénario** (ou **j!play** ou **j!load** ou **j!jo**).
- **j!avancer nom\_ou\_numéro\_de\_la\_salle** (ou **j!go** ou **j!move** ou **j!av**).
  - Il est possible que la salle où vous souhaitez aller ait un digicode par exemple, dans ce cas "j!avancer numéro\_salle code".
  - Dans d'autres cas, cela peut être une mise par exemple (casino) : "j!avancer numero\_salle mise", etc.
- **j!reculer** retourne dans la salle précédente (en vérifiant si c'est autorisé) (ou **j!back** ou **j!return**)
- **j!examiner meuble ou objet ou here/ici** (ce dernier répète le texte de la salle actuelle) (ou **j!ex** ou **j!look**)
- **j!prendre objet** (ramasse un objet) (ou **j!pr** ou **j!take**)
- **j!repondre** permet de répondre à une question ou une énigme dans un scénarios
- **j!inventaire** (affiche l'inventaire) (ou **j!liv** ou **j!item** ou **j!items**)
- **j!jeter objet** (enlève l'objet de votre inventaire. (ou **j!je** ou **j!throw**)
- **j!abandonner** (met fin à la partie) (ou **j!ab** ou **j!giveup**)
- **j!modifier variable** (modifie la valeur d'une variable dont le nom se termine par \_m) (ou **j!modif**, **j!edit**, **j!change** ou **j!changer**)

Le bot disposent aussi d'autres commandes :

- **j!ping** affiche le temps de réponse du bot et **j!info** ou **j!faq** les informations relatives au bot
- **j!setprefix** ou **j!prefix** permet de changer le préfixe du bot sur un serveur
- **j!jeter\_des XdY** (ou **j!roll**, **j!dice**, **j!dices**, **j!des** ou **j!lancer\_des**)
  - Il est possible d'additionner plusieurs lancers de dés et des valeurs
    - Par exemple « j!jeter\_des 3d6 + 2d20 + 5 »
  - Il est possible d'ajouter des paramètres, dans l'ordre suivant (chacun est optionnel, mais si vous les combiner, respecter cet ordre) :
    - « rX » => Relance les dés dont la valeur est X
    - « mX » => Ajoute un dé si la valeur du dé est X (cumulatif, et chaque dé supplémentaire peut lui même en donner un nouveau)
    - « !X » => Compte le nombre de dés égaux ou supérieurs à X
    - Par exemple : **j!jeter\_des 3d6r1m6!4** => Relance les « 1 », ajoute un dé pour chaque « 6 », et compte une réussite par « 4 ou + ».
- **j!lancer\_pieces nombre\_de\_pièces** (ou **j!coin**, **j!coins**, **j!flip**, **j!flipcoin** ou **j!pileface**)

# Schéma des scénarios

J'ai choisi de vous présenter la syntaxe des scénarios en 3 niveaux de difficultés. Le bot est capable de gérer des scénarios très complexes, et il est conseillé de se faire la main avec les fonctions de base.

**Tout d'abord, le scénario doit être enregistré en .txt, avec l'encodage « UTF-8 ».**

**Le niveau 1** correspond à des scénarios façon « livre dont vous êtes le héros » où le joueur choisit entre plusieurs numéros pour avancer, et peut ramasser et utiliser des objets (exemple : clés pour ouvrir des portes)

Il permet de gérer 2 types de fin de partie : 998 (par exemple « défaite ») et 999 (par exemple « victoire »). Chacun étant accompagné d'un texte.

**Le niveau 2** ajoute l'explication des objets invisible (ou état/conditions), les codes/cadenas, et les objets/états/conditions à ne PAS avoir pour pouvoir accéder à une salle.

Il permet davantage de possibilités, à la manière d'un escape game en livre par exemple

**Le niveau 3** inclut l'utilisation des variables, leurs déclarations, modifications, et utilisations dans le scénario (condition d'accès à une salle, game over, etc.), ainsi que l'utilisation de l'aléatoire dans le jeu

Il inclut aussi la possibilité de déclencher des événements (changer une variable, ajouter/enlever un objet, aller automatiquement sur une position, etc.), via la salle 997

Il détaille l'utilisation des réactions.

Tout cela permet une infinité de possibilités, vous réussirez sûrement à en trouver une à laquelle je n'ai absolument pas pensé, cependant il est assez complexe.

## **Tester et ajouter votre scénario au bot :**

Pour pouvoir lancer votre scénario, il doit être hébergé sur internet. Soit de façon temporaire (sur Sendyoo.com par exemple) soit sur un FTP.

Ensuite, il vous suffit de faire « j!jouer votre\_url/votre\_scenario.txt »

Si vous avez plusieurs scénarios sur le même ftp, vous pouvez utiliser la commande « j!lien\_url add votre\_url » pour ajouter l'url du dossier où sont stockés vos scénarios. Cependant vous devrez le faire à chaque redémarrage du bot. Pour le faire de façon constante, vous pouvez ajouter l'url dans la description de votre/vos salons #JDR-Bot

Lorsque votre scénario est terminé et fonctionnel, si vous souhaitez l'ajouter officiellement au bot, vous devez me l'envoyer afin que je l'ajoute au FTP officiel du bot.

Si vous êtes une communauté possédant son propre FTP avec plusieurs scénarios dessus, et que vous souhaitez que votre url soit officiellement ajoutée au bot, c'est aussi possible, contactez moi pour en discuter, je verrais au cas par cas ;)

# Niveau 1 :

Un exemple, regardez ensuite la partie dont la couleur correspond :

🏰 Château en T ... heu Hanté !|rafraichir\$🔄|inventaire\$📖

6

## 1 hall

*Vous êtes dans le hall d'un grand château.+n+&&*

*Vous pouvez aller dans la cuisine (2), la salle de bain (3) ou le salon (4).+n+&&*

*Vous pouvez aussi tenter de sortir par la fenêtre donnant sur le jardin (5).+n+(Rappel : [[PREFIX]]avancer [numéro] )*

|

2

3

*4|cle\_salon1 cle\_salon2|La porte est solide et fermée par 2 verrous.|Vous déverrouillez la porte grâce à vos clés.*

*5|marteau \$|Vous avez besoin d'un objet solide pour casser la fenêtre.|Vous brisez la fenêtre !*

*sortie->4|cle\_salon1 cle\_salon2|La porte est solide et fermée par 2 verrous.|Vous déverrouillez la porte grâce à vos clés.*

\*\*\*\*\*

## 2 cuisine

*La cuisine est sale !+n+La table est pleine de bazar !+n+&&*

*Vous pouvez retourner dans l'entrée (1) ou aller dans le cagibi (6).+n+&&*

*(Astuce : [[PREFIX]]examiner [meuble] ça peut aider, et [[PREFIX]]prendre [objet] aussi :))*

*cle\_salon1|table|Dans tout ce bazar, vous distinguez une belle clé (cle\_salon1).|Vous ramassez une clé poser sur la table.|La clé sert à ouvrir un verrou évidemment !*

1

6

\*\*\*\*\*

## 3 salle-de-bain

*Il n'y a rien d'intéressant dans la salle de bain...+n+&&*

*Ah quelque chose brille dans la baignoire !+n+&&*

*Vous pouvez retourner dans l'entrée (1).+n+&&*

*(Astuce : [[PREFIX]]examiner [meuble] ça peut aider, et [[PREFIX]]prendre [objet] aussi :))*

*cle\_salon2|baignoire|Il y a un fond d'eau croupi dans lequel est plongée une clé (cle\_salon2).|Vous ramassez une clé dans la baignoire.|C'est une clé, ouvre un dictionnaire si tu ne connais pas !*

1

\*\*\*\*\*

## 4 salon

*Vous entrez dans le salon et trouvez le trésor dans le salon !+n+&&*

*Vous le prenez et partez en courant. AU VOLEUR !!!!*

|

*999|Félicitation, vous avez gagné*

\*\*\*\*\*

## 5 fenetre

*Vous sauter à travers la fenêtre brisée.+n+&&*

*Un bout de verre vous entaille profondément le ventre.+n+&&*

*Après une longue agonie ... Vous êtes mort !!!*

|

*998|Vous avez perdu, c'est pas génial !*

\*\*\*\*\*

## 6 cagibi

*Il fait froid ici !+n+Il y a une caisse à outil (caisse).+n+&&*

*Vous pouvez retourner dans la cuisine(2)+n+&&*

*(|Astuce : j!examiner [objet] ça peut aider, et j!prendre [objet] aussi :)|)*

*marteau|caisse|Vous ouvrez la caisse à outil : elle contient un magnifique marteau rouge (marteau)|Vous trouvez un beau marteau bien solide !|Le marteau est rouge et bien solide.*

2

\*\*\*\*\*

Le scénario commence donc par un **TITRE**, qui peut être suivi par une (ou plusieurs) des options suivantes (séparé par « | » :

- rafraichir\$EMOJI => affiche sous chaque message une réaction pour re-afficher le texte de l'emplacement actuel
- inventaire\$EMOJI => affiche sous chaque message une réaction pour afficher l'inventaire
- v\_NomVariable\_v\$EMOJI => affiche sous chaque message une réaction pour afficher la valeur et la description de la variable « NomVariable »
- Salle\$EMOJI => affiche sous chaque message une réaction donnant accès à la salle « salle » (nom ou id de la salle). Il n'y a pas besoin d'ajouter manuellement la direction de cette salle dans chaque salle. Cette réaction n'apparaîtra pas s'il s'agit de la salle en cours.

« EMOJI » doit être remplacé par une emoji en unicode (par exemple 🔄 ou 🗺️)

Puis nous avons le **Nombre de SALLES** (une salle = un chapitre ou un passage du scénario)

A coté du nombre de salles, vous pouvez indiquer un markdown (« fix » par défaut)

Puis, chaque salle se décompose en plusieurs bloc (ou partie):

- **Numéro de la salle** (dans l'ordre, sert de repère aux bots et à vous-même) et **Nom de la salle** (peut remplacer le numéro pour la commande « !avancer ». Par exemple «j!avancer
- **Description de la salle** (= bout de scénario) avec indication des différentes possibilités pour avancer, prendre, examiner (vous pouvez aussi ne pas indiquer certaines choses, si c'est utile pour votre scénario)
  - notez le « +n » dans le texte : Il indique un retour à la ligne dans le texte affiché
- **Objets dans la salle**
  - **Nom\_objet|Endroit\_où\_est\_l'objet|Description de l'endroit où est l'objet (un meuble ou tiroir par exemple)(1)|Texte lorsque l'on prend l'objet(2)|Description de l'objet(3)**
    - (1) = j!examiner meuble, (2) = j!prendre objet (si on ne l'a pas déjà dans l'inventaire), (3) = !examiner objet (que l'on soit dans la pièce de l'objet ou que l'on ait l'objet dans l'inventaire)
    - Il est possible de mettre plusieurs objets à la suite, sur la même case, en les séparant d'un « | » :
      - **Objet1|endroit1|description endroit 1|texte quand on prend l'objet 1|description de l'objet 1|Objet2|endroit2|description endroit 2|texte quand on prend l'objet 2|description de l'objet 2|.....**
    - Lorsqu'un objet est pris, il n'apparaît plus sur cette case, et le meuble n'est plus examinable (Vous verrez dans le niveau 2 comment le rendre examinable avec une nouvelle description)
  - **Salles (ou cases) accessibles avec !avancer [nombre], plusieurs possibilités :**
    - **Nombre (2 par exemple) => Chiffre seul, la salle est accessible sans conditions**
    - **Nombre|objet\_requis objet\_requis2 ...(1)|Description si on n'a pas un des objets requis(2)|Description si on a les objets requis(3)**
      - (1) : Il peut y avoir un ou plusieurs objets requis et/ou variable égale/inférieur/supérieur à X, il faudra alors tous les avoir. Chaque condition est séparée par un espace.
        - Si un « \$ » est présent dans les conditions, les conditions seront supprimées après la première réussite. Il n'y aura plus besoin de remplir cette condition pour passer. Dans le cas contraire, il faudra à chaque fois avoir le(s) objet(s) requis
      - (2) : On peut y indiquer les objets requis, ou les laisser secrets, selon le scénario, tout est possible.
      - (3) : Si un « \$ » est présent dans les conditions, cette descriptions disparaît après la première réussite. Même si on perd un des objets, la salle est considérée comme sans condition. Dans le cas contraire, le texte apparaîtra à chaque réussite.
    - **Alias->direction** : On peut attribuer un alias à une salle. Dans l'exemple, vous voyez « sortie → 4|... ». Cela permet de faire « j!avancer sortie », qui équivaut alors à « j!avancer 4 ».
      - Vous pouvez spécifier des conditions d'accès aux alias, comme dans l'exemple. Cela peut aussi servir à varier les conditions d'accès en fonction des alias (par exemple « sortie » => avoir les deux clés, « enfoncer\_porte->4 » => avoir un pied de biche par exemple)
    - **precedent** : Permet de rendre accessible la salle précédente (par exemple si on vient de la salle 52 ou « cuisine », on peut faire j!avancer 52 ou j!avancer cuisine).
      - On peut y mettre des conditions (objet\_requis ou variable)
    - **998 =>** Si ce chiffre est présent dans la liste des salles accessibles, cela signifie que la partie s'arrête avec une défaite (peu importe s'il y a d'autres salles accessibles)
    - **999 =>** Si ce chiffre est présent dans la liste des salles accessibles, cela signifie que la partie s'arrête avec une victoire (peu importe s'il y a d'autres salles accessibles)
    - **997 =>** Sera détaillé au niveau 3, permet d'exécuter une action (modifier une variable, changer de salle, etc.) automatiquement si une condition est respectée (avoir un certain objet par exemple)

- Enfin les salles se terminent par « \*\*\*\*\* » (peu importe le nombre de \*, minimum 5 à la suite.), afin de la séparer des autres.

La dernière ligne du scénario est optionnelle et sera expliqué à la fin du niveau 3. Elle permet d'attribuer une réaction à un alias. Vous pouvez l'ignorer pour l'instant, ou aller voir l'utilisation des réactions au niveau 3.

## Formater le texte :

Vous remarquerez certains caractères spéciaux dans le texte. Il ne faudra pas les utiliser en tant que texte mais pour leur fonction dans le bot.

- « << » et « >> » pour les liens de sons et musiques (expliqué après) ou des messages TTS
- « [[ » et « ]] » pour les liens d'images (expliqué après) ou pour du texte non formaté (par défaut le texte est en markdown « fix », cela vous permet de formater le texte comme vous le souhaitez).
- « +n+ » fait un retour à la ligne dans le texte affiché
- « {{ » et « }} » pour mettre un temps d'attente/de pause dans un texte ou une situation
- « && » mis en **fin de ligne** (sans caractère ni espace derrière!) permet de couper un bloc en plusieurs lignes. Le && indique au bot de fusionner la ligne avec la lignes suivantes, et de traiter le tout comme une seule ligne (celle du texte de la salle, celle des objets/variables/conditions, ...)
  - Cela n'ajoute pas de retour à la ligne dans le texte affiché (contrairement à « +n+ »), et il faut bien faire attention à respecter la syntaxe de la ligne avant coupure (pas d'espace supplémentaire, pas d'oublie de « | », etc.)
- « ## » Permet d'ajouter des commentaires à votre scénario. Le bot ignorera tout caractère d'une ligne se trouvant après « ## ».
- « | » Ce caractère est utilisé comme séparateur par le bot, ne l'utilisez pas dans vos textes.
- « @@ » Ce caractère sépare les différents effets d'un événement 997
- « \*\*\*\*\* » Sert à séparer les salles. Vous pouvez en utiliser entre 1 et 4 à la suite dans un texte. A partir de 5 à la suite, le bot considère que c'est une séparation de salles.

Il existe aussi les éléments suivants :

- « [[PREFIX]] » sera automatiquement remplacé par le prefix du bot selon le serveur.
- « [[INVENTAIRE]] » sera remplacé par la liste des objets présents dans l'inventaire.
- « [[SALLE:X]] » sera remplacé par le texte de la salle X

## Afficher des images :

Le bot est capable d'afficher des images. Pour cela vous pouvez mettre le lien de l'image entre « [[ » et « ]] ». Par exemple « Je suis un vampire[[[http://image\\_de\\_vampire.jpg](http://image_de_vampire.jpg)]]Je fais peur ! ». Le bot affichera le premier texte, l'image, puis le second texte.

L'url de l'image doit commencer par « http » et être entre « [[ » et « ]] » pour être reconnu par le bot.

## Jouer du son :

Le bot se connecte automatiquement au salon vocal « JDR-Bot » s'il existe, au début de chaque partie. Il se déconnecte en fin de partie ou en cas d'abandon.

Pour inclure un son dans votre scénario, il faut mettre le lien du son entre « << » et « >> ».

Par exemple « Le monstre vous envoie une bombe.<<<http://www.vos-sons/bombe.mp3>>>Vous subissez 50 points de dégâts ! »

Le bot ne peut émettre deux sons simultanément. Si un son est déjà en cours, le bot arrêtera l'ancien son pour mettre le nouveau.

De plus, les sons ne se diffusent pas en continue, ils sont lancés une fois par le bot. Cependant si la personne retourne dans la salle où se trouve le son, le bot relancera le son.

Si vous souhaitez diffuser une musique d'ambiance, et mettre des sons à vos actions, vous pouvez donner, aux joueurs, le lien de votre musique d'ambiance en début de scénario, et utiliser le bot pour les sons d'action.

Si plusieurs parties sont en cours sur le même serveur (sur différents salons), les sons seront joués sur le même chan vocal (Un bot ne pouvant se connecter que sur un chan vocal par serveur).

## **Conditions ET/OU :**

Par défaut, les conditions (par objets requis, conditions ou variable) se cumulent. Si vous mettez 2 objets, il faudra avoir les 2 objets pour respecter la conditions (comme les clés dans l'exemple).

Si vous souhaitez qu'une salle soit accessible avec un objet OU un autre objet, vous pouvez mettre 2 fois la case de la salle, chaque ligne ayant une condition différente. Le bot tentera la première, puis la deuxième, etc.

Enfin si aucune des conditions n'est respecté, le bot affichera le texte correspondant à la dernière ligne testée.

## Niveau 2 :

### Les objets invisibles / Etats / etc. :

En plus de l'inventaire, vu précédemment, le joueur dispose d'un inventaire invisible où sont stockés tout ce que l'on ne veut pas voir apparaître dans l'inventaire visible du joueur. Par exemple :

- États (blessé, fatigué, motivé, etc.)
- Objets débloquent ou bloquent une salle
- Classe ou métier du personnage
- etc. Ce que vous en ferez dépend uniquement de votre scénario.

Retenez juste que ces « objets » n'apparaissent pas dans l'inventaire, et qu'ils sont « pris » ou « enlevé » automatiquement si le joueur va sur une case avec un objet « invisible » (sans commande « !prendre » donc)

Par exemple, dans une case de toilettes publiques, vous pouvez avoir :

- **dégouté|invisible|note invisible inutile|l'odeur vous dégoute au plus haut point (vous avez l'état "dégouté")|Vous avez envie de vomir à cause de l'odeur des toilettes !**
  - La mise en forme est la même que pour les objets de l'inventaire, sauf que l'endroit est remplacé par « invisible » (mot clé important pour le bot), et le 3eme paramètre n'est absolument pas utilisé par le bot (d'où la « note invisible inutile », mais libre à vous de mettre ce que vous voulez
  - Cela donne donc : **Nom\_Etat|invisible|note inutile|phrase affiché lors que l'on acquiert ou perd (-) cet état|Description de l'état.**

Un Objet/Etat/... de l'inventaire invisible peut être indiqué comme condition d'accès à une salle, au même titre que les objets. En mettant -état/objet/condition/... ( « -dégouté » par exemple) comme condition, cela signifie alors qu'il ne faut PAS AVOIR cet état/objet/condition.

Par exemple, si la salle (3) d'un scénario est une salle de danse, et qu'on y accède depuis le vestiaire, vous pourriez avoir :

Dancing House

2

1 entree

L'entrée donne accès a la salle de danse (2)

|

**2|-fatigue|Vous êtes trop fatigué pour danser, aller dormir!|C'est partie pour une nuit de folie !**

\*\*\*\*\*

2 danse

Bienvenue sur la piste de danse ! Quand vous serez fatigué, retourner dons à l'entrée (1).

**fatigue|invisible|note inutile|Vous êtes fatigué à force de danser|La fatigue n'est pas bonne pour le corps.**

1

\*\*\*\*\*

Donc, vous commencez par l'entrée. Jusqu'à là, tout va bien. Vous n'êtes pas fatigué, vous pouvez donc accéder à la salle de danse (2).

Dans la salle de danse, vous devenez automatiquement fatigué (mot clé « invisible » => votre inventaire invisible contient maintenant « fatigue »).

Lorsque vous retourner dans l'entrée, vous n'aurez plus accès à la salle de danse, dont l'accès est restreint par « -fatigue », ce qui signifie : Ne pas avoir « fatigue » dans son inventaire invisible (= ne pas être fatigué)

Contrairement aux objets (qui disparaissent de l'endroit lorsqu'ils sont ramassés), vous pouvez avoir de nouveau cet état/objet si vous le supprimez et que vous repassez par la salle où vous aviez eu cet état/objet précédemment. En revanche, vous ne pouvez pas avoir 2 fois simultanément (comme pour un objet)

## Utilisation avancée des objets :

Si dans une case, un objet invisible est précédé de « - » et que l'objet (ou état, ou autre ...) se trouve dans l'inventaire ou dans l'inventaire invisible, il en est automatiquement retiré. Le mot-clé « **invisible** » peut donc être aussi utilisé pour retirer un objet que le joueur a ramassé précédemment (avec « !prendre »)

### Exemple d'illustration :

Maisonnette

2

1 entree

Vous êtes maintenant devant la porte d'entrée, mais le bâtiment semble désert.+n+C'est la seule pièce disposant d'une lumière, mais vous avez une lampe sur la table.+n+Vous pouvez essayer d'aller dans le salon (2).

lampe|table|Une table est un meuble avec 4 pieds... Oh il y a deux lampes dessus !|Vous ramassez la lampe torche|La lampe vous permet de vous éclairer.|lampe|table|Une table est un meuble avec 4 pieds... Oh il y a une lampe dessus !|Vous ramassez la lampe torche|La lampe vous permet de vous éclairer.|penurie\_de\_lampe|table|Il n'y a plus rien dessus !|Comment avez-vous trouvez cet objet caché ? Pirate !|Je doute fort qu'un joueur lise ceci !

2|lampe|il fait trop sombre pour entrer sans lampe!|Vous entrez dans le salon en vous éclairant avec votre lampe

\*\*\*\*\*

2 salon

Vous pouvez retourner dans l'entrée (1).

-lampe|invisible|note inutile|La lampe vous échappe des mains et se casse ... Vous vous en débarrassez.|La lampe vous permet de vous éclairer.

1

\*\*\*\*\*

Nous pouvons donc ramasser une lampe dans la première salle. Cette lampe est indispensable pour accéder à la deuxième salle. Cependant, en allant dans la deuxième salle, elle se casse (= est retiré de l'inventaire) automatiquement (mot clé « **invisible** »).

Nous pouvons retourner dans la première salle, mais il n'y aura plus de lampes, et nous sommes coincés. Il est possible de mettre plusieurs lampes disponibles, en mettant plusieurs fois le même objet, dans le même meuble (ou endroit).

Cela permet de gérer un nombre de vies par exemple, ou de changer la description du meuble. Pour créer une description de ce meuble vide, rajouter un objet (dont les joueurs ignorent le nom et/ou la présence, pour éviter le « j!prendre »), avec le même meuble que les objets précédents, mais avec une autre description (et vous mettez n'importe quoi dans les autres champs).

## Niveau 3 :

Le niveau 3 inclut l'utilisation des variables, leurs déclarations, modifications, et utilisations dans le scénario (condition d'accès à une salle, game over, etc.), ainsi que l'utilisation de l'aléatoire dans le jeu

Il inclut aussi la possibilité de déclencher des événements (changer une variable, ajouter/enlever un objet, aller automatiquement sur une position, etc.), via la salle 997

Il permet une infinité de possibilités, vous réussirez sûrement à en trouver une à laquelle je n'ai absolument pas pensé, cependant il est assez complexe.

### Les variables :

Tout d'abord, une variable est un élément du jeu dont la valeur peut être modifiée. Prenons par exemple le nombre de vies dans un RPG : la variable pourrait avoir comme nom « points\_de\_vie » et sa valeur 5, ou points\_de\_vie = 5.

On peut compter des blessures, un nombre de tours, un nombre de clés, etc. Tout est possible.

Voilà maintenant comment inclure et utiliser une variable dans le bot :

```
[...]  
*****  
2 Premier_jet  
Vous lancez un Dé 6. Si vous obtenez 5 ou 6, vous serez amené dans la pièce suivante.  
des_six|variable|%=1:6|Vous obtenez v_resultat_v|Un dé de 6 faces.  
1|v_des_six_v.in.1-4|Votre résultat est trop bon pour recommencer voyons ! Aller donc recevoir votre médaille (3) !  
Vous retournez au début pour recommencer...  
3|v_des_six_v.in.5-6|Votre résultat est trop mauvais pour continuer, retourner au début (1) !|Félicitation, vous aller  
recevoir une médaille pour ça !  
*****  
[...]
```

Les variables se déclarent de la même façon que les objets, sur la même ligne dans une salle. Ignorez pour l'instant les lignes en noir.

Voyons chaque partie en détail :

- **Nom de la variable : sans espace, ni accent.**
- « variable » est le mot-clé indiquant au bot qu'il s'agit ici d'une variable (c'est semblable au fonctionnement de « invisible »)
- **Attribution ou modification de la valeur de la variable :**
  - « = X » : La variable a pour valeur X
  - « %=X:Y » : La variable a pour valeur aléatoire un nombre compris entre X et Y (dans notre exemple, des\_six aura une valeur comprise entre 1 et 6, comme un dé classique)
  - « %+X:Y » : On ajoute un nombre aléatoire entre X et Y à la variable.
  - « %-X:Y » : On soustrait un nombre aléatoire entre X et Y à la variable.
  - « +X » : On ajoute X à la valeur actuelle de la variable.
  - « -X » : On soustrait X à la valeur actuelle de la variable (elle peut être négative)
- **Phrase à afficher.** On peut ne rien afficher (on met alors « null »), et on peut afficher dans la phrase la valeur d'une variable en l'entourant de « v\_ ». Par exemple « v\_resultat\_v » affichera la valeur du random.
  - « v\_resultat\_v », « v\_valeur\_v », « v\_reponse\_v » sont des variables pré-définies par le bot.
    - « v\_resultat » : Est égale au résultat du dernier aléatoire utilisé (par exemple « %=1,6 »)
    - « v\_valeur » : Est égale au dernier « code » ou « mise » indiquant dans une commande « !avancer [case] [code] »
    - « v\_reponse » : Est utilisé avec la commande « j!repondre »
- **Description de la variable (on peut y inclure la valeur d'une variable avec « v\_nom\_variable\_v »)**

## Utilisation des variables comme conditions :

Tout comme les objets et les conditions (mot-clé « invisible »), les variables peuvent être utilisées comme condition. Pour indiquer au bot que nous utilisons une variable, nous l'entourons de « v\_ » et « \_v ».  
Dans notre exemple, nous avons 2 salles accessibles :

- la « 1 » est accessible si notre résultat est entre 1 et 4
- la « 3 » est accessible si notre résultat est entre 5 et 6

L'ordre des phrases d'accès ou de non-respect de la condition est le même qu'avec des objets ou états. Il est possible de combiner différents types de conditions (objets, états et variable)

L'utilisation de variable comme condition se présente sous la forme : [variable ou valeur].[opérateur].[variable ou valeur]. Le bot prend en compte les opérateurs suivant :

- = : **Égal à ...**
- > : **Supérieur à ...**
- >= : **Supérieur ou égal à ...**
- < : **Inférieur à ...**
- <= : **Inférieur ou égal à ...**
- != : **Différent de ...**
- **in.X-Y : compris entre X et Y**
- **out.X-Y : n'est pas compris entre X et Y**

Par exemple `v_score_v.>=.50` => on vérifie si le score est supérieur ou égale à 50. Ou encore `v_prix_v.out.3-10` => on vérifie si le prix n'est pas compris entre 3 et 10€.

## Les variables online :

Les variables online sont un type de variables dont la valeur est sauvegardé, sur un fichier sur le serveur hébergeant le bot, à la fin de la partie (998, 999 ou abandon). La valeur sauvegardé est chargé à chaque lancement du scénario. Elles se déclarent à la fin du scénario, après la dernière case (avant ou avant les réactions d'alias) sous la forme :

`Variable_o|Valeur_par_défaut|Description`

- **Le nom de la variable**
  - Doit se terminer par `_o` pour que le bot la considère bien comme une variable à sauvegarder/charger (par exemple `nb_victoires_o`).
  - Doit être sans espace ni accent
- **La valeur par défaut (Celle qui sera sauvegardé lors du premier lancement du scénario)**
- **Description de la variable qui sera afficher si un joueur examine la variable avec la commande « !examiner »**

Il faut les déclarer ligne par ligne (une variable = une ligne dans le scénario)

Ensuite, leur utilisation est identique aux variables classiques (`v_variable_o_v`, utilisation avec « |variable| », etc.) et leur valeur sera sauvegardée par le bot à la fin de la partie.

Par défaut, il existe, pour tout scénario, la variable « `nb_parties_o` » dont la valeur est augmentée de 1 à chaque valeur d'un scénario. Vous pouvez ainsi, dès la première ligne de votre scénario, affiche le nombre de fois que votre scénario à été joué (par exemple « Château (Ce scénario a été joué `v_nb_parties_o_v`) ». Bien que cette valeur puisse être modifié dans votre scénario (comme toute autre variable), cela n'a pas vraiment d'intérêt si vous voulez savoir le nombre réel de lancement de votre scénario (le bot l'augmentant de 1 automatiquement à chaque lancement)

Les variables textuelles peuvent aussi être online et fonctionnent de la même façon.

*Le scénario `train.txt` est un bon exemple (utilisation de `nb_parties_o` + compteur de victoires/défaites).*

## Les variables textuelles :

Les variables textuelles ont, pour valeur, un mot ou une phrase. Elles se déclarent/se modifient sur la même ligne que les variables classiques, les objets, états, etc., sous la forme :

*nom\_variable|variable\_t|Valeur|Phrase affichée lors de modification par le joueur|Description*

- **Nom de la variable : sans espace, ni accent.**
- « variable\_t » est le mot-clé indiquant au bot qu'il s'agit ici d'une variable textuelle (c'est semblable au fonctionnement de «variable»)
- Attribution ou modification de la valeur de la variable (un mot ou un texte)
- Cette phrase sera affichée uniquement lorsque le joueur modifiera la valeur via « j!repondre valeur »
- Description affichée si le joueur examine la variable

Pour afficher cette variable dans un texte, il faut l'entourer de « t\_ » et « \_t ». Par exemple « *Enchanté de te connaître t\_nom\_heros\_t* »

Elle peut être modifier par le joueur au moyen de la commande « **j!répondre Valeur** » (la Valeur pouvant être un mot, ou une phrase si elle est entourée de guillemet). Cette commande n'est utilisable que si la variable est déclarée dans cette salle.

Cela peut vous permettre de demander un nom de personnage par exemple.

Pour rendre une variable textuelle non-modifiable, il vous suffit de la déclarer, puis de passer automatiquement à une salle suivante avec un événement 997 par exemple.

*Le scénario train.txt est un bon exemple (demande et utilisation du nom du personnage).*

## **Variable Textuelle comme condition :**

Il est possible d'utiliser les variables textuelles comme condition d'accès. Elles sont compatibles avec 4 opérateurs :

- *t\_variable1\_t="Sangoku"*
  - La condition est vraie si la valeur de « variable1 » est « Sangoku »
- *t\_variable1\_t!="Sangoku"*
  - La condition est vraie si la valeur de « variable1 » n'est pas « Sangoku »
- *t\_variable1\_t.in."Sangoku"*
  - La condition est vraie si la valeur de « variable1 » est présente dans « Sangoku » (par exemple « goku » ou « San »)
- *t\_variable1\_t.out."Sangoku"*
  - La condition est vraie si la valeur de « variable1 » n'est présente dans « Sangoku » (par exemple « z » ou « pikapika »)

Il est possible de comparer à une autre variable textuelle (toujours avec « t\_ » et « \_t ») ou de changer la position selon le besoin. Par exemple mettre « "San".in.t\_variable1\_t » pour vérifier si « variable1 » contient « San »

## **La salle spéciale 997 :**

Au même titre que les salles « 998 » et « 999 », la salle « 997 » se met dans la ligne des cases accessibles.

Elle peut servir à aller automatiquement dans une autre salle, à modifier la valeur d'une variable, à ajouter ou retirer un objet ou un état, ou encore à afficher un texte.

Reprenons l'exemple précédent et faisons en sorte qu'il aille automatiquement à la salle 1 ou 3 en fonction du résultat :

```
[...]  
*****  
2 Premier_jet  
Vous lancez un Dé 6. Si vous obtenez 5 ou 6, vous serez amené dans la pièce suivante.  
des_six|variable|=1:6|Vous obtenez v_resultat_v|Un dé de 6 faces.|nb_essais_six|variable|+1|Vous avez fait  
v_nb_essais_six_v lancer.|Nombre de lancés de D6.  
997|v_des_six_v.in.1-4|1|Vous n'êtes pas en forme, recommencez !  
997|v_des_six_v.in.5-6|3|Félicitation, vous avez réussi au bout de v_nb_essais_six_v essais !  
*****  
[...]
```

Remarquez que j'ai rajouté une seconde variable (« v\_nb\_essais\_six\_v ») qui compte le nombre d'essais avant de réussir.

Voyons maintenant cela en détail :

- 997, c'est la salle indiquant une action automatique, vous l'aurez comprise
- Condition(s) pour que l'action se réalise. On a vu précédemment comment la présenter, par exemple « v\_vie\_v.>.0 ». Elle peut aussi prendre la valeur « null » (= aucune condition, l'action aura forcément lieu), ou utiliser des objets ou des états comme condition (ou les combinés.) Cette partie fonctionne de la même manière que les autres conditions.
- Action(s) réalisée(s): On peut en mettre plusieurs dans un même 997, en les séparant par « @@ ».
  - Modification d'une variable : (avec +, =, -, %=X:Y,%+X:Y, ou%-X:Y)
  - Changement de salle (on met le **numéro de la salle** ou v\_variable\_v (le numéro de salle sera la valeur de cette variable))
  - Ajout ou retrait d'un objet/état de l'inventaire : [nom de l'objet]&&[inventaire ou invisible]&&[description de l'objet ou de l'état]. Comme pour les autres objets ou états, on peut le précéder de « - » pour le supprimer
    - Par exemple : « tournevis&&inventaire&&Le tournevis sert à visser ou dévisser. »
    - Autre exemple : « -fatigue&&invisible&&Vous n'êtes plus fatigué ! »
  - Déclaration ou modification d'une variable numérique ou textuelle (online ou non): [nom de la variable]&&[variable ou variable\_t]&&[valeur de la variable]&&[description de la variable]
- Phrase à afficher. On peut ne rien afficher (on met alors « null »), et on peut afficher dans la phrase la valeur d'une variable en l'entourant de « v\_ » et « \_v » (si variable numérique) ou « t\_ » et « \_t » (si variable textuelle)

## Utilisation des actions personnalisées:

Vous pouvez définir des actions personnalisées, donnant accès à des commandes au nom de l'action.

Ces action se définissent soit dans une salle (action locale, possible uniquement dans cette salle) au même endroit que les directions et les événements 997 (donc après le texte de la salle), soit à la fin du scénario (au même endroit que les variables online, les alias, ...)

La syntaxe : « action:cible|condition(s) \${O-O}|effet|phrase si condition ok|phrase si condition pas ok »

Le joueur devra alors utiliser la commande « j!action cible ». Par exemple « casser:porte », le bot répondra à la commande « j!casser porte ».

Si l'action/commande écrite par le joueur n'existe pas, le bot l'indiquera. De même si la cible est incorrecte.

Il est possible de mettre « all » comme cible. Dans ce cas, la commande réalisera l'action, quelle que soit la cible tapée par le joueur.

La variable **action\_cible** (ou **t\_action\_cible\_t** selon l'endroit du scénario) contiendra la dernière cible tapée par le joueur (valide ou non), et la variable **action\_cible\_ok** contiendra la dernière cible valide tapée par le joueur.

Il est alors possible, avec « all » et ses 2 variables, de créer une multitude d'action tout en laissant une grande liberté au joueur.

Dernière chose : il est possible de spécifier plusieurs fois la même action:cible. Dans ce cas, la première dont les conditions sont respectés sera exécutés. Si aucune ne respecte les conditions, ce sera le texte de refus de la dernière qui sera affiché. De même, les actions locales sont testées avant les actions globales.

Le scénario « exemple\_actions.txt » vous montre des exemples d'utilisations concrets !

## Utilisation des réactions :

Vous pouvez ajouter des réactions dans votre scénario. Pour cela, il faut faire « \ » suivi d'une emoji sur discord, qui vous affichera la version unicode (une emoji plus petite). C'est cette emoji qu'il faut c/c dans le code, précédé de « § ». Suite a des souci d'affichage avec libreoffice, j'ai remplacer les emojis par « [\*-\*] », « (°\_°) » et « {O-O} », mais pour le bot, il faut bien utiliser le copier/coller du caractère unicode de l'emoji !

Vous pouvez utiliser les réactions des manières suivantes :

- Relier une salle (Numero Nom\_salle\${[\*-\*])
  - Le bot affichera cette réaction dans chaque salle où la salle correspondante est accessible. Cela fera la commande « j!avancer X » et vérifiera donc les conditions d'accès.
- Afficher la description d'un meuble ou d'un objet, prendre un objet. Par exemple :
  - marteau|caisse|Vous ouvrez la caisse à outil : elle contient un magnifique marteau rouge (marteau)\${[\*-\*]}

- Vous trouvez un beau marteau bien solide !§(°\_°)|Le marteau est rouge et bien solide !§{O-O}
- [\*-\*] affichera la description de la boîte à outil, (°\_°) la description du marteau, et {O-O} ajoutera le marteau à l'inventaire.
  - A un événement 997 : « 997|2§[\*-\*]|null|Ah ça ne sert à rien !|... C'est cassé maintenant ! »
    - La réaction se met donc dans la case « condition », et peut être combiné à des conditions (variables, objets ou conditions).
    - On spécifie un nombre d'utilisations « X§[\*-\*] », X étant le nombre d'utilisations
    - le 3ème champ (« null » dans l'exemple) est ce que fait l'événement (voir le chapitre sur la case 997)
    - La première phrase s'affiche quand l'événement est déclenché
    - La deuxième phrase s'affiche si les conditions ne sont pas respectées, ou si le nombre d'utilisations est atteint.
  - A une alias : « alias11§[\*-\*]|alias2§(°\_°)|alias\_inutile§{O-O} »
    - à la fin du scénario, après la dernière case, vous pouvez spécifier les réactions correspondant aux alias. Elles apparaîtront lorsque l'alias est utilisé. Elles sont optionnelles comme les autres.
    - Vous en avez une dans l'exemple du niveau 1.
  - A une action personnalisée : « action:cible|condition(s) §{O-O}|effet|phrase si condition ok|phrase si condition pas ok
    - Cela fonction pour les actions globales et locales. Dans le cas d'une action locale, la réaction n'apparaîtra que dans la salle où l'action est possible.