# Liste des glitch possibles

* Underflow/overflow sur nombre
* Stackoverflow
* OOB (out of bound)
* Manipulation de program counter (pokemon, coin case glitch)
* Warp glitch (mario) [avec la compétence vol par exemple)
* Screen warp (zelda)
* Pokemon merge glitch (pokemon)
* Swim on land glitch (ratchet and clank, mario)
* Back in time glitch (zelda)
* Super jump (tous les jeux havoc mal codé)

Manipulation du program counter

Le joueur entre dans l’auberge du jour 2, largement corrompu par l’intervention du virus dans le tutoriel, manque de chance, la porte d’entrée et son entourage a été transformée en amas de pixels et le warp renvoi directement à l’intérieur de l’auberge. Pour sortir il faut manipuler les sprites pour réparer la porte.

Le rôle de la coin case es joué par le registre qui sert à tester le résultat de l’énigme (montrer que le registre est glitch)

Le rôle du « code de fortune » est joué par les objets au sol dans la réserve de gauche à droite

L’utilisation du registre fait glitcher successivement les 4 sprites de la zone pour indiquer qu’ils servent d’instructions

Quand une chaise est rencontrée, l’état (bug/non bug) de tous les sprites de la zone bug sont inversés

Quand une bouteille est rencontrée, tous les glitch sont décalé d’un cran vers la gauche

Quand Carl est rencontré, tous les sprite de la ligne sur laquelle il se trouve glitchent

Quand une case vide est rencontrée, la sortie glitch

Etat initial de la zone de la porte :

|  |  |  |  |
| --- | --- | --- | --- |
|  | X | X |  |
| X |  | X | X |
| X | Sortie |  |  |

Instructions corrects :

|  |  |  |  |
| --- | --- | --- | --- |
| Bouteille | Carl | Vide | Chaise |

Utilisation futur : il est possible de trouver un objet plus tard dans le jeu à mettre à l’emplacement vide pour avancer dans une branche caché de l’histoire si besoin (en éditant le warp de la sortie, note : la case vide édite la sortie, c’est l’indice). Le coin case glitch est polyvalent, le nôtre peut l’être aussi

# Alternative : Utilisation du screen warp

# Utilisation du back in time

Le joueur rentre dans une zone fermé, l’écran glitch avec l’affichage de l’écran de menu du jeu, puis sélection automatique de « nouveau jeu ». Après cela, le joueur vois son personnage dans la zone fermé, mais un sprite sur deux (damier) correspond à la taverne de départ (sans pnj ni blocage). Le joueur se voit donc à deux endroits en même temps. La version « back in time » et la version normal. Son but est de sortir avec les deux en même temps sachant que les deux effectuent les mêmes actions en même temps. Il faudra utiliser intelligemment les murs pour bloquer un personnage et avancer avec l’autre. Ceci permet d’accéder à une version glitché « back in time » de crossbell. Ici il devient plus facile de naviguer puisque les deux personnages sont superposés. Le but est d’utiliser la version back in time pour accéder à un warp bloqué derrière un PNJ dans la version normale. Une fois ceci fait, le back in time prend fin et crossbell est définitivement bloqué dans sa version « back in time » sans avoir les deux versions du joueur. Ceci avance grandement le niveau de corruption du monde (murder run)

# Alternative : warp glitch

Il existe dans la ville de durin une porte bloqué et bug qui peut warp le joueur vers n’importe quel warp du jeu. En accédant à la base de données et en rentrant une valeur particulière il est possible que ce warp envoi à l’endroit du warp bloqué de crossbell, skipant l’énigme du back in time. Cette technique ne corrompt pas crossbell et n’augmente pas la corruption du monde. Pour traverser la porte il faut y accéder depuis le mur et donc effectuer un OOB juste avant. Pour ce faire, il faut donner un coup à un cochon (Référence à zelda) qui va devenir particulièrement vénère et knock back le joueur dès qu’il le trouvera. Le knock back peut envoyer directement dans la porte, même si elle n’est pas accessible. Pour donner un coup au cochon, la commande boire peut être utilisé pour que le tavernier accepte que c’est une bonne idée de frapper un cochon