

Rapport du projet d'informatique

Florent Collin, Bac 1 info

Institut d'Informatique, Faculté des Sciences, UMONS

Mai 2018

Table des matières

1	Introduction	3
2	Répartition des tâches au sein du groupe	4
3	Modélisation des concepts du jeu	4
3.1	Le plateau et les mondes parallèles	4
3.2	Les cellules	4
3.3	Les Items	5
4	Architecture	6
4.1	Traitements	6
4.2	L'interface graphique	6
5	Implémentation et choix personnels	7
5.1	Le plateau	7
5.2	Les règles	8
5.3	L'héritage sur les items	9
5.4	Lecture des niveaux	10
5.5	Les paramètres de l'utilisateur	11
6	Points forts et points faibles	11
6.1	Points forts	11
6.2	Points faibles	12
7	Améliorations	12
7.1	Améliorations implémentées	12
7.2	Améliorations possibles	12
8	Conclusion	13
A	Annexe : Mini-guide utilisateur	14
B	Annexe : Erreurs connues	16

1 Introduction

Ce rapport présente les différents éléments de conception du jeu «Baba Is You» qui a été réalisé dans le cadre du projet d'informatique en Bac 1 Info. «Baba Is You» est un jeu de puzzle très particulier qui a été inventé lors du Nordic Game Jam 2017 par Arvi Teikari [5].

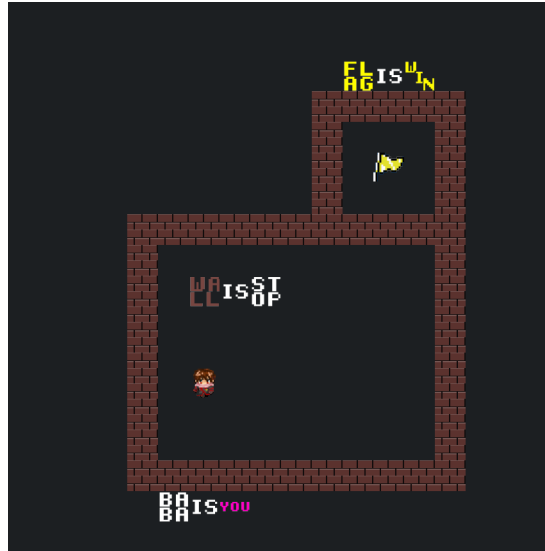


FIGURE 1 – Un des niveaux de Baba Is You

Ce qui différencie «Baba Is You» des autres jeux de puzzle est son principe de règles entièrement dynamiques. Toute la complexité du code provient de ce système. En effet, dans un jeu plus classique les règles qui définissent les actions que le joueur a le droit de faire sont «hard» codées¹ alors qu'ici le joueur peut modifier la majorité des règles en alignant des suites d'objets.

L'objectif principal de ce projet est de concevoir une application logicielle qui met en œuvre les concepts vus au cours de Programmation et Algorithmique 1 et 2. Cette application codée en Java permet via une interface graphique de jouer au moins aux quatre premiers niveaux du jeu original[5] ainsi que de pouvoir sauvegarder une partie en cours et charger un niveau de l'utilisateur. Dans la section 6, les améliorations de ces fonctionnalités de base seront décrites.

Dans la suite de ce rapport, les sections exposeront la répartition des tâches dans le groupe, la modélisation des concepts du jeu ainsi que l'architecture de l'application. Par la suite, des choix personnels effectués durant la conception seront argumentés. Pour

1. Hard code signifie construire un programme non modifiable par des paramètres d'entrée ou « coder en dur ». [4]

finir, les points forts ainsi que les points faibles seront énoncés et un mini-guide utilisateur sera décrit pour que l'utilisateur puisse utiliser correctement l'application.

2 Répartition des tâches au sein du groupe

Voici la répartition des tâches au sein du groupe :

- Réalisation de l'application : Florent COLLIN
- Réalisation du rapport : Florent COLLIN

Valentin ANTOINE ayant décidé de changer d'orientation, n'a pas tenu à s'impliquer dans la réalisation du projet.

3 Modélisation des concepts du jeu

L'objectif de cette section est d'expliquer globalement les concepts gérés par le jeu. Nous utiliserons pour cela une analyse top-down².

3.1 Le plateau et les mondes parallèles

« Baba Is You » est un jeu qui est composé de niveaux. Chaque niveau peut se voir sous la forme d'un plateau où l'on peut facilement discerner les nombreuses cellules qui le composent (Figure 3). Chaque plateau peut donc être décrit comme un tableau à deux dimensions constituées de cellules. À partir du niveau 5, le joueur découvre une nouvelle fonctionnalité du jeu : les mondes parallèles et les portails. Les portails bleus permettent de passer au monde parallèle suivant tandis que les portails rouges permettent de revenir au monde parallèle précédent. Chaque niveau peut donc contenir plusieurs mondes parallèles et chaque monde est représenté par un plateau différent (Figure 4).

3.2 Les cellules

Comme énoncé dans la sous-section précédente, les tableaux contiennent des cellules. Chaque cellule va contenir une liste d'item³ qui est ordonnée en fonction de la priorité d'affichage de chaque item, car dans le jeu plusieurs items peuvent se superposer dans une même cellule (Figure 2).



FIGURE 2 – Exemple d'une cellule qui contient plusieurs items

2. En informatique, une analyse top-down est une analyse qui consiste à partir d'un problème et de le subdiviser jusqu'à ce qu'il ne reste plus que de simples problèmes, faciles à implémenter.

3. Item = Un des objets visibles dans le jeu. Par exemple, les murs, les textes de règles, baba, sont des items.

3.3 Les Items

Les items sont les éléments visibles dans un niveau. Un item peut soit être un texte (ex. : le mot «WIN») soit un objet (ex. : un rocher). Chaque objet aura sa correspondance en texte.

Note : cette correspondance peut être changée dans le fichier «CorrespondingTextOrItem.json» qui se situe dans le dossier «settings».

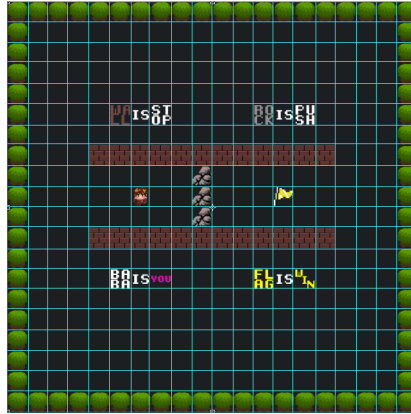


FIGURE 3 – Un plateau où les cellules sont visibles



FIGURE 4 – Niveau 5 - Monde parallèle 1 et 2

4 Architecture

L'architecture de l'application est dans le cadre de ce projet, très simple, mais mérite néanmoins de s'y intéresser (Figure 5). Afin de simplifier les interactions, l'application principale n'interagit qu'avec des fichiers stockés sur la machine de l'utilisateur. Dans le cadre de ce projet, une base de données aurait pu être utilisée, mais elle aurait complexifié la réalisation.

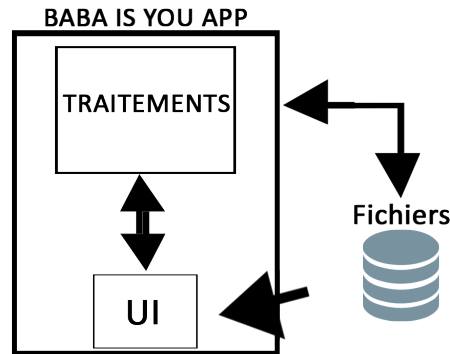


FIGURE 5 – Architecture de l'application

La partie interne de l'application est divisée en deux : les traitements et l'interface graphique (UI).

4.1 Traitements

La partie des traitements est la partie qui va gérer la majorité du programme. C'est elle qui va interagir avec les fichiers lors, par exemple, de la lecture d'un niveau. C'est aussi elle qui va gérer les différents plateaux et qui va permettre de déplacer des items à l'intérieur de ceux-ci. La partie des traitements s'échange aussi des informations avec l'interface graphique pour que celle-ci puisse afficher des informations à l'écran.

4.2 L'interface graphique

L'interface graphique va se charger d'interagir avec la partie des traitements en traduisant par exemple, les entrées clavier de l'utilisateur en commande pour le plateau. C'est elle qui va se charger de dire au plateau de déplacer les joueurs dans une direction. Comme on peut le voir sur la figure 5, les fichiers sont aussi utilisés par l'interface graphique. En effet lorsque celle-ci veut peindre le plateau elle accède aux images des items qui se situent dans le dossier «ressources».

Cette implémentation n'est pas la meilleure possible, la construction d'une DAL ⁴ aurait permis de séparer complètement l'UI de l'accès aux fichiers.

4. Data access layer (en) (en français, couche d'accès aux données), en informatique, un composant logiciel réalisant l'interface et la séparation entre une base de données et les composants de haut niveau les exploitant[1].

5 Implémentation et choix personnels

Cette section traitera des différents choix d'implémentation qui ont été effectués durant la conception. Pour chaque choix, plusieurs solutions seront proposées avec leurs avantages et leurs inconvénients ce qui permettra d'en choisir une et de voir son implémentation en pseudo-code ou directement en Java.

5.1 Le plateau

Comme expliqué à la section 3, le plateau peut être vu comme un tableau à deux dimensions contenant des cellules, elles-mêmes contenant des items, mais l'implémentation pourrait aussi être une simple liste d'items, où chaque item retiendrait sa position et lorsque le programme a besoin de déplacer un item celui-ci va chercher dans cette liste quel item a cette position.

Solution 1 : Liste d'items

- Avantages :
 - Facilité de compréhension
 - Implémentation plus rapide
- Inconvénients :
 - Chaque item doit connaître sa position
 - Dès que le programme doit rechercher un item en particulier, la complexité de l'algorithme est en O_n
 - Quand un item se déplace, le programme doit modifier la position (x,y) de cet item

Solution 2 : Le tableau à deux dimensions

- Avantages :
 - Économie de mémoire, car aucun item ne connaît sa position⁵
 - Quand le programme déplace un item, il ne doit pas réattribuer une position (x,y) à l'item
 - Optimisation de la complexité algorithmique lorsqu'on sait où se trouve l'objet dans le tableau
- Inconvénients :
 - Difficulté de compréhension
 - Difficulté lors de l'implémentation
 - Complexité algorithmique en O_n^3 lors du parcours de l'entièreté du tableau

5. Après plusieurs tests, le fait que chaque item ne connaisse pas sa position a amélioré les performances de l'application.

Après avoir énuméré les solutions, la deuxième méthode semble se démarquer grâce à ses nombreux avantages et a donc été implémentée dans le projet. Cela implique que la mémoire des items est très limitée, ils ne connaissent pas leur position. La seule donnée qu'il possède est le nom de la classe de l'item qui est utilisé pour les dessiner lors de l'affichage graphique.

Les algorithmes de recherche d'un item en particulier seront donc en général de complexité On^3 . Ce qui peut paraître beaucoup, mais en général un plateau est un carré de 20 sur 20 cellules⁶, chaque cellule contenant habituellement moins de 2 items. Le programme devra donc effectuer une itération sur seulement 800 éléments. La vitesse d'exécution du programme ne sera donc pas compromise malgré la complexité des algorithmes de recherche.

5.2 Les règles

Les règles font partie intégrante du jeu, elles définissent comment le jeu doit réagir face à certaines situations. Les règles étant dynamiques, il faut pouvoir les stocker sous une certaine forme. Ici, encore plusieurs solutions sont possibles. Les règles pourraient être stockées sous la forme de chaînes de caractères ou sous la forme d'un item implémentant une interface(IRule) qui définirait si un item est un mot, une action, un verbe, etc,... Chaque item de règle⁷ implémenterait cette interface.



FIGURE 6 – Un exemple de règle

Solution 1 : Sous la forme de chaînes de caractères

- Avantages :
 - Permet d'afficher très facilement les règles actives.
- Inconvénients :
 - Les erreurs sont vite arrivées lors de la manipulation de chaînes de caractères.
 - Le nom d'un item de règle est hard codé ce qui complexifie les modifications de ce nom (si l'on veut modifier le nom d'un item on doit le faire dans tout le code).

6. Généralement, la complexité d'un algorithme se calcule dans le pire des cas possibles sauf qu'ici un plateau peut potentiellement avoir une taille infinie, on prend donc une moyenne.

7. Un item de règle est un texte «WALL» ou «PUSH» par exemple.

Solution 2 : Sous la forme d'objets IRule

- Avantages :
 - L'implémentation de nouveaux objets est dynamique (il ne faut pas indiquer quelque part comment s'appelle cet item).
 - L'héritage peut être utilisé si un item a exactement la même fonctionnalité qu'un autre.
 - Les classes des objets IRule peuvent se retrouver facilement dans un IDE ⁸.
- Inconvénients :
 - Moins explicite qu'une chaîne de caractères.
 - Plus coûteux en mémoire, car une règle sera représentée par une liste IRule tandis que dans la solution 1 une règle n'est qu'une simple séquence de caractères.

Malgré le désavantage au niveau du coût, la solution 2 permet de rendre plus dynamique l'ajout de nouveaux items. Elle permet aussi d'éviter beaucoup de fautes de frappe dues aux chaînes de caractères.

Maintenant que l'on sait comment sont stockées les règles, il faut s'intéresser à la façon dont les règles sont détectées. Avec l'implémentation du plateau, il faut implémenter un algorithme de recherche de règles se basant sur les données du plateau.

Idée générale :

1. Parcourir le tableau à deux dimensions contenant les cellules
2. Si un item de règle est trouvé.
 - (a) Regarder les deux cellules à droite, s'il existe deux mots permettant de créer une phrase, alors on ajoute cette règle.
 - (b) Idem que (a), pour les deux cellules inférieures.

5.3 L'héritage sur les items

Utiliser la programmation orientée objet était un des principaux objectifs de ce projet et l'implémentation des items permet de l'observer (Figure 7). Prenons deux exemples :

1. Un item Wall
 2. Un item TextRock
1. Wall hérite de RealItem, une classe abstraite qui sert à définir tous les objets réels ⁹. RealItem hérite lui de la classe Item qui sert à définir l'ensemble des items présents dans le jeu.
 2. TextRock hérite de RuleItem, une classe abstraite qui sert à définir les items de règles. Elle surcharge par exemple les méthodes isPush() ¹⁰ de la classe Item, car un item de règle est toujours déplaçable. La classe RuleItem implémente aussi l'interface IRule qui permet de trouver les règles (cfr. 5.2).

8. Un IDE(Integrated Development Environment) est un environnement de développement intégré.

9. Les objets réels signifient tous les items sauf les textes de règles.

10. isPush() est une méthode définie dans la classe Item qui renvoie «true» si l'item est déplaçable, «false» sinon

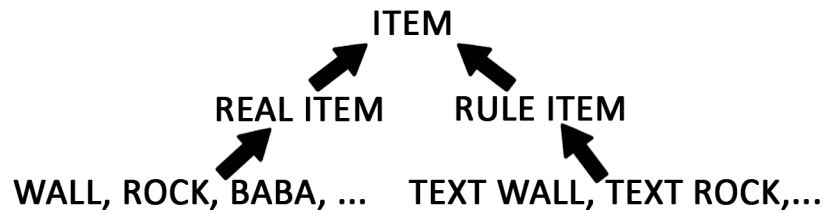


FIGURE 7 – L’héritage sur les items

5.4 Lecture des niveaux

L’une des consignes de l’énoncé obligeait de pouvoir lire un niveau à partir d’un fichier texte où chaque ligne contient la position d’un item. Par exemple, «Baba 1 10» signifie qu’un item Baba doit être placé à la 1re ligne et la 10e colonne. Le programme a donc besoin de savoir quelle instance d’item il doit insérer dans le tableau.

Deux solutions sont envisageables :

1. Implémenter un switch
2. Utiliser la réflexion pour instancier les items[2]

La première solution consiste à implémenter un switch qui va instancier un item en fonction de chaque chaîne de caractères. Tandis que la deuxième, va à partir d’une chaîne de caractères, trouver directement la classe correspondante, ce qui permettra par la suite d’instancier les items.

L’utilisation de la réflexion rendra plus dynamique la création de nouveaux items. En effet, en utilisant la réflexion, on évite de devoir ajouter des lignes dans le switch pour pouvoir utiliser ces nouveaux items. On réduit donc, par la même occasion le nombre de lignes de code.

Illustration du code des deux solutions :

```

// Implémentation du switch
switch(str) {
case "Rock": return new Rock(); break;
case "Baba": return new Baba(); break;
case "TextBaba": return new TextBaba(); break;
case "Wall": return new Wall(); break;
...
}
  
```

```
// Implémentation de la réflexion
// Code inspiré de
// https://stackoverflow.com/questions/22439436/loading-a-class-from-a-different-package
try {
    return Class.forName("game.element." + str);
} catch (ClassNotFoundException e) {
    System.out.println("ERROR: A name of an Item is wrong in the text
                        file: " + str);
}
```

5.5 Les paramètres de l'utilisateur

Dans l'application, l'utilisateur a la possibilité de sauvegarder ses propres raccourcis clavier. Il faut donc sauvegarder ces paramètres tout d'abord dans une collection, puis dans un fichier.

Le type de fichier qui a été choisi est le fichier texte avec un format Json, car il est très répandu et rapide par rapport aux autres formats de fichiers textes. De plus, certaines bibliothèques Java permettent de sérialiser et de désérialiser un Json en une seule instruction. Le code sera donc lisible et facile de compréhension.

6 Points forts et points faibles

Dans cette section les points forts et les points faibles seront évoqués, mais ils restent des points de vue subjectifs et dépendront donc de chacun. La liste n'est évidemment pas exhaustive.

6.1 Points forts

- Le principal point fort de cette application logicielle est sa facilité à ajouter de nouveaux éléments. Il est très facile d'ajouter de nouveaux items, de donner les mêmes caractéristiques à un item, d'ajouter des paramètres, ou des succès¹¹.
- L'application dépasse les objectifs donnés dans l'énoncé du projet. Le jeu implémente 8 niveaux, dont 4 inédits, un menu, un éditeur graphique de niveaux, des succès, des paramètres modifiables, de la musique, etc, ... (cfr section 7.1)
- Le jeu est fluide et les chargements sont quasiment instantanés.

11. Dans le jeu, l'utilisateur en fonction de ses actions, va débloquent des succès.

6.2 Points faibles

- La musique peut parfois avoir certains soucis de latences. Cela vient du fait que la musique est un fichier .wav et non .mp3¹². La musique est donc plus lourde et longue à charger.
- À chaque déplacement, le jeu recherche les différents joueurs, ce qui ralentit un peu le jeu (même si cela n'est pas perceptible). Une des améliorations possibles serait de garder en mémoire la position des joueurs et de changer ces positions lorsqu'on déplace les joueurs dans une direction.
- Dans certaines parties du code, il y a de la redondance qui pourrait être supprimée en créant des méthodes supplémentaires.

7 Améliorations

À l'intérieur de cette section, les améliorations qui ont été ajoutées par rapport au jeu original et à l'énoncé sont évoquées ainsi que les améliorations qu'il est possible d'implémenter.

7.1 Améliorations implémentées

Voici la liste des améliorations implémentées dans le jeu par rapport à l'énoncé et au jeu original :

- 9 niveaux dont 5 inédits sont disponibles ;
- Des mondes parallèles ont été ajoutés (section 3.1).
- Plusieurs nouveaux items dont des portails ont été implémentés.
- L'utilisateur peut créer et charger ses propres niveaux personnalisés via l'éditeur graphique de niveau.
- Les raccourcis clavier sont modifiables et s'enregistrent automatiquement.
- L'utilisateur au fur et à mesure de sa progression, peut débloquent des succès.
- De la musique a été ajoutée au menu principal.
- Des sons sont joués lors de l'interaction avec certains éléments.

7.2 Améliorations possibles

Dans cette sous-section, les différentes améliorations possibles seront évoquées. En général, ces améliorations n'ont pas été implémentées par manque de temps.

12. La musique aurait pu être transformée via un logiciel en .mp3 sauf que les machines sur lesquelles le logiciel doit fonctionner ne reconnaissent pas les fichiers .mp3 soit parce que la version de Java n'est pas à jour soit parce que le système d'exploitation «Ubuntu 16.04» ne permet pas à JavaFx de lire des fichiers .mp3

Comme dit dans les points faibles, la méthode qui recherche les joueurs pourrait être améliorée, car celle-ci rafraîchit la liste des joueurs chaque fois qu'on se déplace sur le plateau. Il faudrait retenir la position des joueurs pour éviter de devoir rafraîchir cette liste.

La deuxième amélioration possible serait de permettre à l'utilisateur de partager ses propres niveaux personnalisés avec les autres joueurs.

Pour cela, on pourrait par exemple, créer une base de données en ligne ou utiliser un service de cloud tel que Google Drive. Google drive permet de stocker des fichiers et de les récupérer, tout cela peut être codé en Java via leur API[3].

La troisième amélioration serait de modifier certaines parties du code qui sont redondantes. Par exemple la méthode pour sauvegarder un niveau existe plusieurs fois en fonction du niveau que l'utilisateur veut enregistrer.

8 Conclusion

En conclusion, ce projet a permis de réaliser une application logicielle complète. Cette application a permis d'approfondir des points vus au cours de Programmation et Algorithmique 1 et 2, tels que la complexité algorithmique, l'héritage, le polymorphisme, etc... Mais cette application nous a aussi incité à faire une analyse top-down d'un jeu et à chercher différentes approches et solutions à un problème donné, pour finir, par créer une application logicielle entière en partant d'un modèle.

Références

- [1] *DAL*. URL : <https://fr.wikipedia.org/wiki/DAL> (visité le 14/05/2018).
- [2] *Get Class from string*. URL : <https://stackoverflow.com/questions/2408789/getting-class-type-from-string> (visité le 15/05/2018).
- [3] *Google Drive Api*. URL : <https://developers.google.com/api-client-library/java/apis/drive/v3> (visité le 15/05/2018).
- [4] *Hard code*. URL : https://fr.wiktionary.org/wiki/hard_code (visité le 08/05/2018).
- [5] Avri TEIKARI. *Baba Is You*. URL : <https://hempuli.itch.io/baba-is-you> (visité le 07/05/2018).

A Annexe : Mini-guide utilisateur

Pour pouvoir utiliser l'application, le logiciel ANT¹³ doit être installé sur votre machine. La commande `ant run` permet de lancer le jeu. Si vous voulez réinitialiser les paramètres et les succès, veuillez utiliser la commande `ant clean_all`. Une fois que vous arrivez sur le menu principal (Figure 8) plusieurs actions vous sont proposées.



FIGURE 8 – Menu principal du jeu

Dans le menu des paramètres (Figure 9), pour changer un raccourci, vous devez :

1. Cliquez sur le raccourci que vous souhaitez modifier.
2. Appuyez sur le nouveau raccourci.

Une fois dans l'éditeur, vous avez la possibilité de créer votre propre niveau personnalisé.

Plusieurs astuces sont importantes à connaître :

- Vous pouvez ajouter et supprimer des mondes parallèles via les boutons «+» et «—» en haut à gauche.
- L'item que vous avez sélectionné se trouve en haut à droite.
- Si vous appuyez sur le clic droit de votre souris, vous allez directement sélectionner la gomme.
- Pour quitter l'éditeur, il vous suffit d'appuyer sur la touche «ESCAPE».

L'éditeur de niveau ne permet pas de créer des niveaux avec des dimensions différentes de 20*20 cellules. Donc si vous souhaitez créer un niveau personnalisé avec des dimensions différentes vous devez suivre les instructions suivantes :

13. ANT est téléchargeable gratuitement sur : <https://ant.apache.org/>

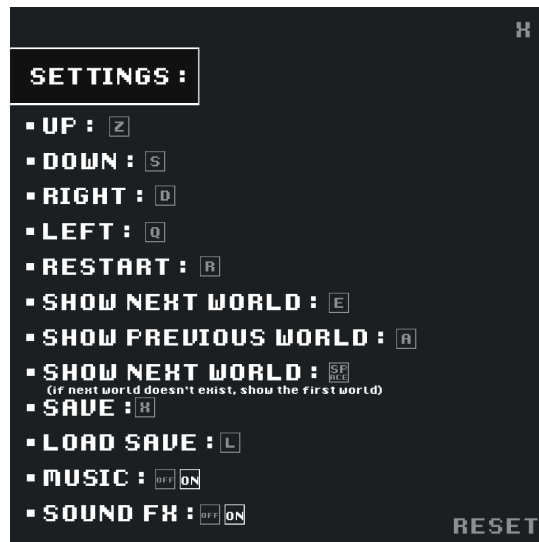


FIGURE 9 – Menu des paramètres



FIGURE 10 – Éditeur de niveau

1. Ouvrir un nouveau fichier texte.
2. «NomDuNiveau -1» doit être la première ligne du fichier. Le «-1» indique que le niveau est un niveau personnalisé.
3. Les prochaines lignes du fichier doivent être de la forme «NomItem x y» où x est le numéro de la ligne dans le tableau et y le numéro de la colonne.
4. Si vous voulez inclure plusieurs mondes parallèles, vous devez entre chaque monde réinscrire la première ligne (ie : NomDuNiveau -1)

B Annexe : Erreurs connues

La seule erreur connue survient lorsqu'on joue par exemple au niveau 1 et que l'on place deux items déplaçables l'un au-dessus de l'autre. En effet, lorsqu'on va avec un joueur pousser sur ces deux items vers la droite, l'un va se déplacer tout à droite et les deux items ne seront plus superposés. Ce bug provient de la partie récursive de la méthode de déplacement des items et aurait demandé quelques heures à être corrigé, mais par manque de temps cela n'a pas été fait. Ce qu'il est intéressant de remarquer, c'est que ce bug survient aussi dans le jeu original.