

BOUSRY Fazal  
FRICOTTEAU Vincent  
MOSMANT Thomas  
REMY Michael

M2 Informatique  
2013-2014

# Projet Android

---

*DeadHal*

---



UNIVERSITÉ  
PARIS-EST  
MARNE-LA-VALLÉE





## Table des matières

1. Introduction.....	4
2. Utilisation .....	5
2.1 Menu Edition .....	5
2.2 Menu Navigation .....	7
2.2.1 Navigation monotouch.....	7
2.2.2 Navigation au gyroscope .....	7
2.2.3 Itinéraire .....	7
2.3 Autres Menus .....	8
2.3.1 Ouverture .....	8
2.3.2 Sauvegarde .....	8
2.3.3 Nouveau labyrinthe .....	8
2.3.4 Help et About .....	8
3. Choix d'implémentation.....	9
4. Difficultés rencontrées .....	9
5. Améliorations possibles.....	11
6. Conclusion .....	11

## 1. Introduction

DeadHal est une application Android pour mobiles et tablettes sous Android qui est un système d'exploitation open source utilisant le noyau Linux.

Elle permet de construire et de visualiser des labyrinthes pouvant être utilisé pour une application ludique mais également pour cartographier l'intérieur des bâtiments et pouvoir s'y repérer.

Ce projet a été réalisé dans lors notre 2<sup>ème</sup> année du Master Informatique à l'université de Marne-la-Vallée dans le cadre du cours d'Android dispensé par Mr Chilowicz. L'application est en anglais.

On définit un labyrinthe comme un ensemble de salles reliées entre-elles par le point d'intersection du couloir avec les salles. Une salle est définie par son périmètre ayant une rectangulaire, ce périmètre comportant des accès à un ou plusieurs couloirs, qui sont des points d'intersection. Les couloirs qui relient 2 salles peuvent être unidirectionnels ou bidirectionnels. Ainsi pour passer d'une salle A vers une salle B, il faut emprunter un couloir menant vers la salle B (en respectant le sens autorisé par le couloir). On peut donc modéliser le labyrinthe par un graphe orienté, les sommets étant les salles, les arcs les couloirs reliant les pièces.

## 2. Utilisation

### 2.1 Menu Edition

Ce mode permet de créer son propre labyrinthe. Pour l'activer il suffit de cliquer sur le cadenas pour qu'il apparaisse déverrouillé (illustration 1.1).

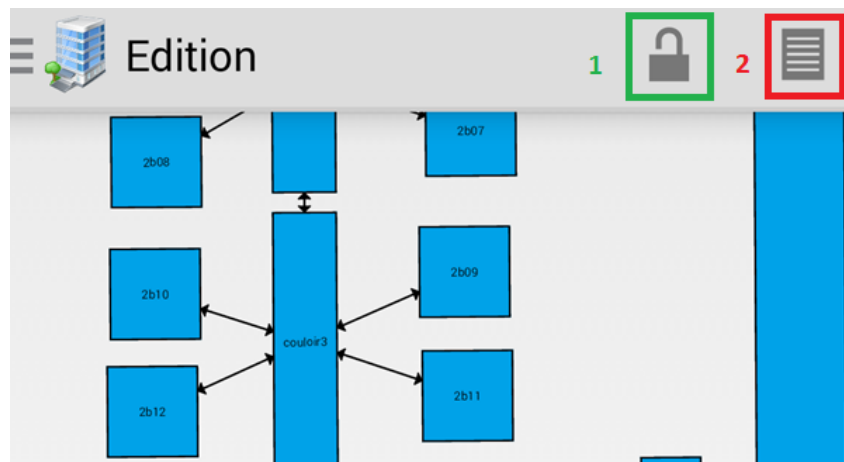


Illustration 1

Pour ajouter une pièce l'application dispose d'un drawer qui s'active en haut à droite (illustration 1.2) ou en le tirant du bord droit de l'écran. Une fois dessus, il suffit de cliquer sur « add room » pour ajouter une pièce (illustration 2.1).

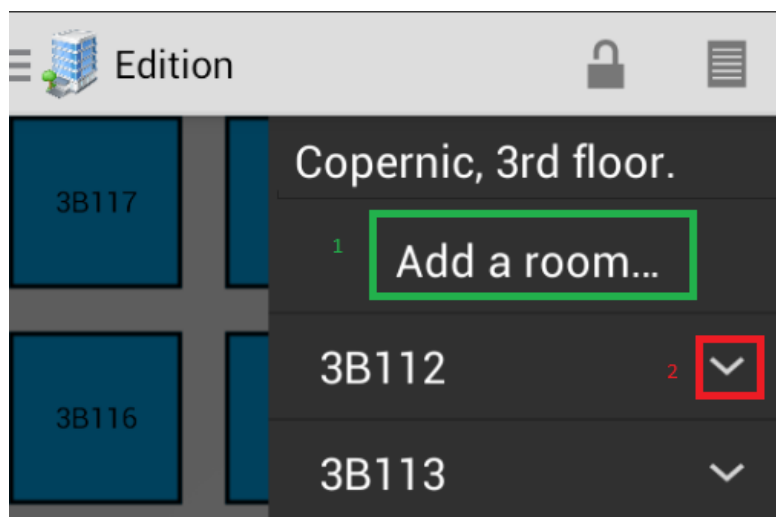


Illustration 2

La nouvelle pièce sera créer automatiquement sur la carte et pourra être redimensionnée ou déplacée.

A l'aide de la flèche situé à droite de chaque salle (illustration 2.2), une popup s'ouvre permettant de renommer, copier ou supprimer la salle sélectionnée.

Pour ajouter un corridor, il faut également se placer dans la barre de navigation de droite et cliquer sur le lien « add corridor » (illustration 3.1).

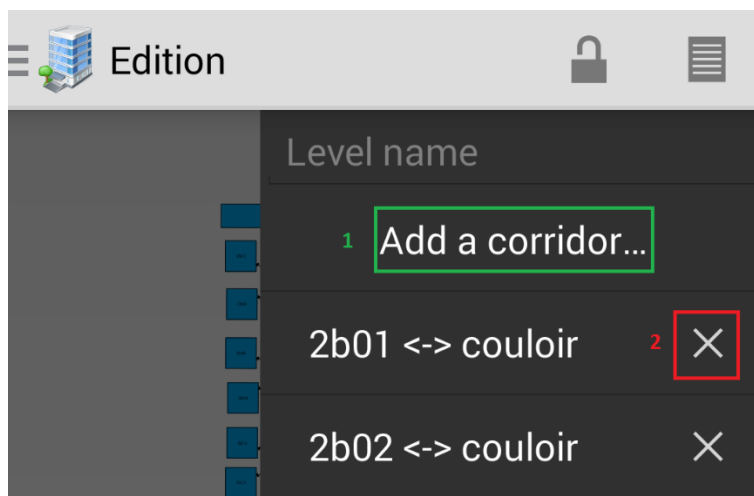


Illustration 3

La suppression d'un corridor se fait dans la barre gauche. Il suffit de cliquer sur la croix du corridor voulu (Illustration 3.2).

Pour ajouter un corridor bidirectionnel il faut cliquer sur la flèche en haut à gauche pour qu'elle soit à double sens (illustration 4.1).

Pour ajouter un corridor unidirectionnel il faut que la flèche soit à sens unique (illustration 4.2).

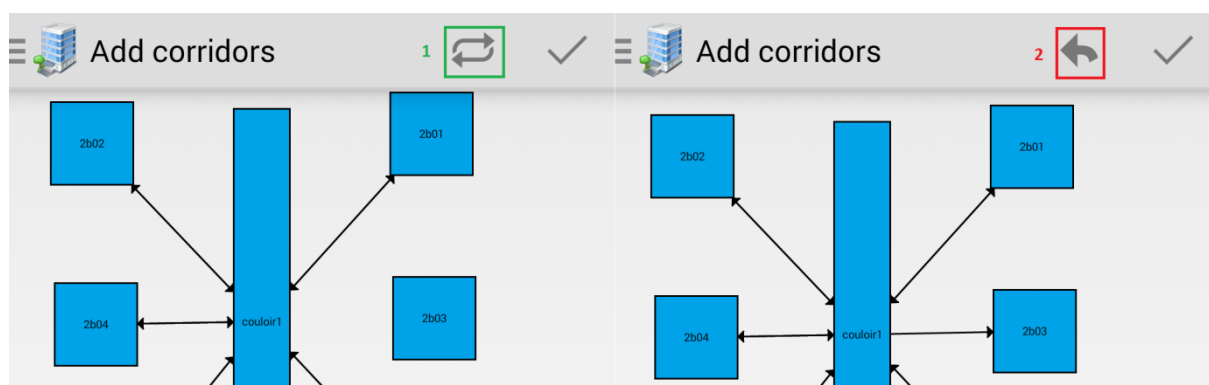


Illustration 4

Il est aussi possible de déplacer ou de redimensionner une salle en faisant une pression longue dessus, pour la déplacer il suffit de la déplacer avec son doigt et il est possible de la redimensionner à partir de 8 endroits, les 4 coins et les 4 bords. Enfin, la sélection d'un objet (salle ou couloir) dans la liste de gauche permet sa sélection. La sélection d'un couloir est uniquement utilisée pour mieux le repérer sur la carte.

## 2.2 Menu Navigation

### 2.2.1 Navigation monotouch

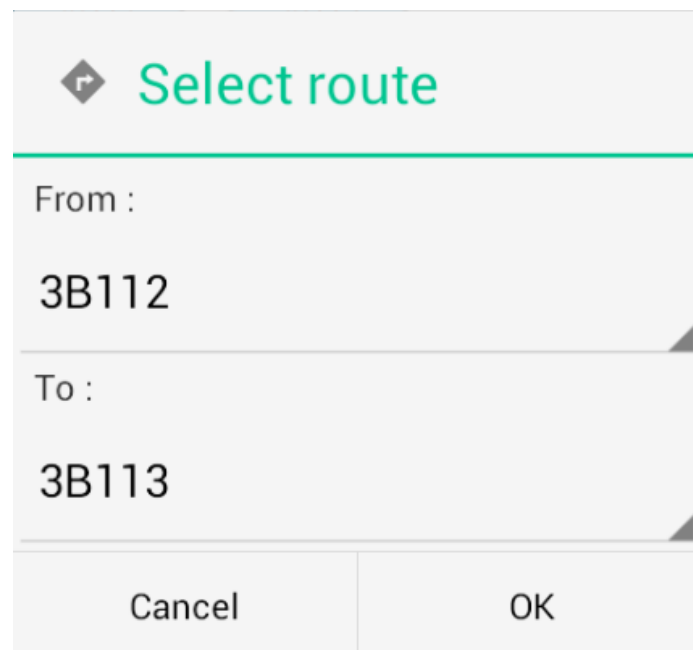
La navigation monotouch s'effectue lorsque le cadenas est verrouillé. Pour placer le pion, il suffit d'effectuer une pression longue sur une salle. Pour le déplacer il faut se placer sur lui-même et il suit les mouvements du doigt tant que celui-ci reste dans la pièce. Pour changer de salle l'utilisateur doit cliquer sur la salle voulue ou faire glisser son doigt sur celle-ci. Un message s'affichera lorsque le déplacement est impossible. Pour supprimer le pion il suffit de faire de nouveau une pression longue sur celui-ci.


### 2.2.2 Navigation au gyroscope

Pour utiliser la navigation à l'aide du gyroscope, l'utilisateur doit activer l'option en cliquant sur le bouton d'option de l'appareil et cocher l'option « Accelerometer ». Le robot se déplacera selon les mouvements du téléphone si le point arrive sur un couloir, le robot va dans la salle ou mène le couloir.

### 2.2.3 Itinéraire

Pour avoir un itinéraire entre deux pièces, l'utilisateur doit cliquer sur le bouton représentant une flèche dans un panneau. L'utilisateur devra ensuite choisir la pièce de départ ainsi que celle d'arrivée (illustration 5). Une fois son choix validé, le chemin le plus rapide sera indiqué sur le labyrinthe avec en vert la salle de départ, en rouge la pièce d'arrivée et les couloirs à emprunter en vert.



 **Select route**

From :  
3B112

To :  
3B113

Cancel OK

Illustration 5

## 2.3 Autres Menus

Ces menus sont accessibles via le menu gauche de l'application.

### 2.3.1 Ouverture

Pour effectuer une ouverture d'un labyrinthe, il faut se placer sur la barre de navigation de gauche et cliquer sur le lien « open » (illustration 6.1). L'utilisateur n'aura plus qu'à choisir le fichier à ouvrir. De plus il est possible à l'utilisateur de partager, renommer et supprimer un labyrinthe en effectuant une pression longue sur le nom du labyrinthe.

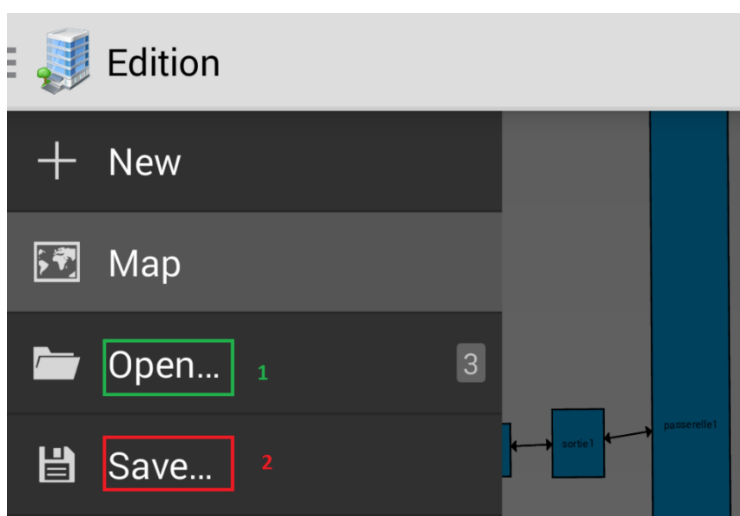


Illustration 6

### 2.3.2 Sauvegarde

Pour effectuer une sauvegarde, une fois le labyrinthe créé, il faut se placer sur la barre de navigation de gauche. Il suffit ensuite de cliquer sur le lien « save » (illustration 6.2) et d'indiquer un nom de fichier. Si le nom du fichier existe déjà une fenêtre s'ouvre pour demander si l'on veut le remplacer.

### 2.3.3 Nouveau labyrinthe

Il est possible de créer un nouveau labyrinthe. Pour cela il faut cliquer sur l'option « New » qui crée un labyrinthe vierge.

### 2.3.4 Help et About

Le menu Help permet d'avoir un guide d'utilisation au sein de l'application. Quant au menu About on y retrouve les informations sur l'application.



### 3. Choix d'implémentation

Pour répondre aux besoins du projet, nous avons d'abord défini très clairement les 3 composants principaux qui nous seraient utiles : Le Level (niveau), les Room (salles), les Corridor (couloirs). Nous n'avons pas vu pour cette première version l'utilité des portes et nous n'avons pas pour l'instant implémenté les points d'intérêts.

Le Level contient les Room et les Corridor.

Les Room sont caractérisées par un ID, un Nom et un Rectangle.

Les Corridor sont caractérisés par un booléen (définissant l'orientation ou non du couloir) et trois ID :

- L'ID du dit Corridor
- L'ID de la Room de départ
- L'ID de la Room d'arrivée

L'application comporte une Activity et se découpe en plusieurs Fragment. Les Fragments de gestion de niveau (édition de salles, édition de couloirs et navigation) sont caractérisés par plusieurs objets définissant un comportement propre :

- Un Level Handler, ou gestionnaire de niveau.
- Un Drawable, qui permet de définir des spécificités particulières au dessin. Il permet par exemple de dessiner le pion dans le cas du Navigation Level Handler.
- Et une View qui permet de récupérer les Motion Event et de les passer au Level Handler afin de les traiter convenablement, chaque Motion Event n'étant pas forcément traité de la même manière dépendamment de l'endroit où l'utilisateur se trouve dans l'application.

Pour le dessin de niveau, nous avons choisi de dessiner sur un canvas et d'utiliser une matrice pour permettre différents niveaux de scalabilité et de zoom du Level.

### 4. Difficultés rencontrées

Un des points abordé dans le sujet est celui de la position relative des salles. Nous n'avons pas implémenté ce point car nous pensons que notre application n'est pas encore assez importante pour nécessiter cette fonctionnalité. La motivation principale d'avoir des salles en position relative était celle d'éviter d'avoir à écrire les calculs à effectuer lors de la suppression d'un parent.

Cependant, la plus grosse difficulté rencontrée a été celle de la gestion de l'accélération matérielle. Le comportement semble totalement inattendu d'un périphérique à l'autre. Sur nos téléphones, l'accélération matérielle ne pose pas de soucis. Pourtant, nous avons remarqué qu'elle pose problème sur tablette. En effet, sur une Transformer Prime 4.1.1, si l'accélération matérielle est activée, l'application est fluide pendant une à deux minutes, puis freeze aussitôt. Une fois l'accélération matérielle désactivée, l'application est stable mais très lente. Le comportement d'une Nexus 7 ne sera pas le même d'une version à une autre. Ainsi en 4.1.0, l'accélération matérielle pose problème (application très lente), mais en 4.3 cela semble résolu. C'est un des gros problèmes de ce projet car

l'application peut être très lente lors du chargement de gros labyrinthe, et nous n'avons pas réussi à le régler pour le moment.

## 5. Améliorations possibles

De nombreuses améliorations sont possibles, et nous les aurions volontiers implémentées avec un peu plus de temps. Nous avons pensé aux améliorations suivantes :

- En premier lieu, régler le problème d'accélération matérielle.
- Optimisation du code : le code actuel effectue beaucoup de calcul qui peuvent être considérablement allégés.
- Fonction de sauvegarde automatique : sauvegarder à intervalles réguliers le labyrinthe en cours d'édition.
- Optimisation de la gestion du cycle de vie d'une activité : la nôtre semble plutôt stable mais peut être améliorée.
- Utilisation du pattern command pour la fonctionnalité d'« Undo ».
- Et tant d'autres...

## 6. Conclusion

Ce projet a été pour nous l'occasion de mieux découvrir l'éco système Android. Nous avons fait de nombreuses découvertes dans ce domaine et avons pu nous perfectionner. Enfin, nous aurions aimé pouvoir implémenter plus de fonctionnalités avant le rendu.