

# PROJET SDD L2 2017

## *Univers et algorithmes labyrinthiques*

© 2013 – 2014 – Franck Lepoivre –  $v_{0.3}$

20 octobre 2013

*Troisième* version pour questions, commentaires et demandes de précisions. Cette version vous permet de commencer votre travail de documentation et de réflexion. Elle est susceptible de très nombreuses évolutions d'ici la fin du mois d'octobre, notamment en fonction de nos interactions.

### 1 Résumé

Le sujet porte sur la conception et la résolution de *labyrinthes*.

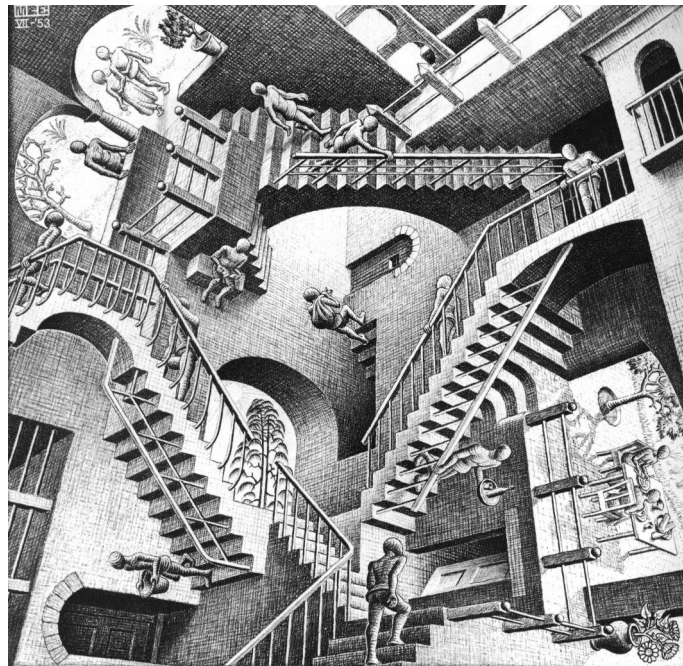


FIGURE 1 – Labyrinthe, Escher

Un individu ou un groupe d'individus est placé au milieu d'un espace clos et confiné. Il est possible de s'échapper de ce lieu en trouvant un itinéraire vers la sortie, pourvu que celle-ci existe. Mais il est plus probable d'y rester piégé, perdu dans le dédale de ses impasses et de ses boucles.



## 2 Sujet

### 2.1 But

Réaliser une application qui permet de concevoir, jouer à, et résoudre automatiquement des labyrinthes.

### 2.2 Objectifs pédagogiques

Le projet vous place dans une situation d'ingénierie qui vous oblige à :

**Réfléchir** aux fondamentaux d'une famille d'applications (les labyrinthes) pour en dégager un modèle général applicable à tous les cas singuliers.

**Traduire** ce modèle général en structures de données et algorithmes réutilisables.

**Appliquer** ce canevas à la réalisation de plusieurs applications (au moins deux).

### 2.3 Conseil

Ne pas programmer avant d'avoir analysé, modélisé, défini le périmètre de son projet et ses potentielles extensions.

### 2.4 Cahier des charges impératif

#### 2.4.1 Chargement et sauvegarde d'un labyrinthe

Votre application doit pouvoir charger et sauvegarder depuis et vers un fichier **de configuration** la définition d'un labyrinthe et d'un état de jeu **qui s'y juxtapose**. Cette fonctionnalité permettra d'assurer la persistance des parties et aussi de mettre les équipes en compétition.

Un format standard pour la définition des labyrinthes 2D à mailles carrées et 3D à mailles cubiques est spécifié en annexe<sup>3</sup>.

L'état de jeu est la donnée des joueurs, c'est-à-dire de l'ensemble de leurs attributs (avatar, position, points de vie, objets acquis, etc), des monstres (Minotaure, fantômes, etc), des ressources vitales (nourriture, objets qui confèrent un pouvoir), des pièges.

#### 2.4.2 Gameplay

Vous vous éveillez (seul ou en équipe), perdu(s) dans un labyrinthe. Vous ne disposez d'aucune carte et il se peut qu'il n'y ait aucune sortie<sup>4</sup>, mais tant que vous ne le savez pas, cela ne sert à rien de désespérer.

Vous ne connaissez que la disposition de l'endroit où vous vous trouvez (*vue subjective*<sup>5</sup>).

Vous pouvez vous déplacer le long des couloirs et escaliers qui s'offrent à vous, mais ne pouvez pas traverser ou enjambrer les murs (**2D**) **ni traverser les sols et plafonds (3D)**. Votre but est de trouver la sortie avant d'avoir épuisé votre temps de vie (qui s'égraine suivant un compte à rebours qui démarre avec le jeu) et avant de faire une mauvaise rencontre au milieu de cet endroit infernal.

Si vous trouvez la sortie, la partie se termine !

---

3. Pas dans la présente version du sujet.

4. On ne distingue pas entrée et sortie : il y a seulement de possibles ouvertures par lesquelles on pourrait s'échapper du labyrinthe.

5. ou *vue première personne*, c'est-à-dire du point de vue du joueur cf. sa position dans le labyrinthe.

## 2.5 Cahier des charges optionnel

### 2.5.1 Les bonus

Les bonus sont des *pouvoirs* conférés par des objets que vous pouvez croiser dans le dédale :

**Ravitaillement** Boisson et nourriture sont essentielles pour gagner du temps de vie.. surtout la boisson. Attention à la nourriture avariée et aux substances qui pourraient perturber l'acuité de vos sens et la perspicacité de votre raison.

**Marquage** Le marquage est la possibilité offerte à un joueur, par exemple en utilisant une craie à durée limitée (nombre de marques), de laisser des signes dans les endroits qu'il a visités. On peut envisager une certaine liberté dans les signes employés pour permettre aux joueurs d'élaborer des stratégies, notamment en contexte de partie multi-joueurs.

**Fil d'Ariane** Dérivé du marquage, le joueur découvre une pelote de fil qu'il va pouvoir dérouler à mesure qu'il avance. Attention, ce fil a une longueur limitée.

**Ailes d'Icare** Elles permettent au joueur qui les découvre et les emploie de prendre de la hauteur et d'obtenir une vue globale du labyrinthe. Le problème, c'est que les ailes du joueur fondent d'autant plus vite qu'il est haut et proche du soleil. Utiliser cette possibilité est donc très risqué. Les ailes font partie des objets qui peuvent être dissimulés dans le labyrinthe. Lorsque le joueur découvre un tel objet, il ne peut l'utiliser qu'une fois.

**Passe-muraille** Ce pouvoir permet d'effectuer quelques pas en traversant es murs, sols ou plafonds.

### 2.5.2 Multi-joueurs

On peut jouer à plusieurs en déployant toutes les stratégies sociales imaginables (coopération, compétition, etc). Cette possibilité ne devient intéressante qu'en intégrant la possibilité de jouer en réseau.

### 2.5.3 Monstres

Minautore, fantôme ou ce que vous voulez, .. un monstre bête n'est pas bien méchant et souvent de bonne compagnie. Mais attention au monstre malin et retord : fourbe, il déploiera ses stratagèmes et vous prendra par surprise. Pour opposer ses monstres malins aux équipes adverses, il faudra réfléchir à des algorithmes d'IA.

### 2.5.4 Snakes

Les joueurs devront prendre garde à ne pas croiser le chemin d'un SNAKE. Un Snake peut naître spontanément ou suite à l'action d'un joueur. Le snake se meut plus ou moins intelligemment au milieu du dédale et mange tout ce qu'il croise, joueurs, ressources, pièges, etc. Le snake meurt quand il se mange lui-même ou lorsqu'il est mangé par un autre snake.

Une option dans l'option : offrir la possibilité pour un joueur de se transformer pour un temps en Snake. Pourvu qu'il fasse attention à ne pas se manger lui-même, le joueur-snake est quasiment invincible. Contrairement à un snake classique, le joueur-snake fait du pas à pas et peut décider de rester immobile.

### 2.5.5 Indication directionnelle

Cette fonctionnalité importante s'appuie sur la résolution automatique du labyrinthe. Elle consiste, contre des points de vie supplémentaires, à indiquer au joueur une direction vers la

sortie la plus proche depuis l'endroit où il se situe. Cette fonctionnalité n'est pas toujours activée (elle ne peut notamment pas l'être si le labyrinthe n'a pas de sortie).

### 2.5.6 Espaces labyrinthiques complexes

Dépasser la simple structure d'univers labyrinthique 2D à maille carrée. En restant en 2D, on peut envisager des mailles triangulaires, hexagonales, ou un pavage irrégulier. On peut passer au 3D comme superposition de niveaux 2D reliés par des passages verticaux (escaliers). Le 4D peut venir avec des murs et des ouvertures mobiles, suivant des enchaînements logiques. En 3D ou 4D, la maille peut également être non cubique, non régulière.

## 2.6 Compétition entre équipes pour les bonus

Les meilleures équipes pourront être mises en compétition, en vue d'obtenir des points bonus, suivant les principes suivants<sup>6</sup> :

**Performances** Même labyrinthe (immense et difficile à résoudre), comparaison des performances spatiale et temporelle pour la résolution, et de la performance qualitative (optimalité de la solution).

**Arène** Un même labyrinthe, des équipes adverses qui y placent leurs dangers et ressources respectifs (mais dangers et ressources aussi pour l'adversaire qui n'a que le handicap de ne pas les connaître en nature).

**Créativité back-office** Généralisation du problème du labyrinthe : labyrinthes 3D / 4D à mailles irrégulières, etc, y compris ce qui n'est pas envisagé dans ces lignes.

**Créativité front-office** Élaboration du design graphique, plaisir à jouer. Les meilleurs réalisations pourront représenter l'école à l'occasion de la JPO<sup>7</sup>.

---

6. Pistes, mais dont la possibilité et les modalités restent à préciser et à valider.

7. Journée Portes Ouvertes.

## 3 Organisation

### 3.1 Encadrement et calendrier

Depuis 2012, le projet de SDD est encadré sur la durée de son exécution, et non pas seulement évalué en décembre ou janvier, au moment de la soutenance.

Vos tuteurs de projet sont :

**Groupe A** Kaïs Klai.

**Groupe B** Franck Lepoivre.

**Groupe C** Michel Landschoot.

**Groupe PL** Franck Lepoivre.

#### 3.1.1 Mise à disposition du sujet

Une première version du sujet sera soumise vers début octobre pour vous permettre de commencer à travailler et pour susciter vos questions et demandes de précisions. Cette première version de cadrage est susceptible de nombreuses évolutions au cours du mois d'octobre, notamment en fonction des réponses à vos questions.

#### 3.1.2 TP de lancement (TPP1)

Chaque groupe effectue un premier TP (de 4h) vers la mi-octobre. Ce sera l'occasion de constituer vos équipes, de prendre connaissance du sujet, effectuer vos premières recherches, initialiser votre travail d'élaboration.

**Groupe A** le samedi 12 octobre matin.

**Groupe B** le mardi 15 octobre après-midi.

**Groupe C** le mardi 15 octobre matin.

**Groupe PL** le mardi 15 octobre matin.

#### 3.1.3 Soutenance intermédiaire (S1)

Une soutenance intermédiaire aura lieu vers mi-novembre pour nous assurer que toutes les équipes ont commencé à travailler. Cette soutenance sera l'occasion d'un point individualisé (par équipe) sur la base d'un échange relativement informel. Il y aura cependant quelques points en jeu sur la note finale, qui seront notamment perdus par les équipes qui n'auraient pas commencé à travailler (sérieusement s'entend) :

**Groupe A** les mardi 19 novembre après-midi et samedi 23 novembre matin.

**Groupe B** les mardi 19 novembre après-midi et samedi 23 novembre matin.

**Groupe C** les mardi 19 novembre matin et samedi 23 novembre matin.

**Groupe PL** les mardi 19 novembre matin et samedi 23 novembre matin.

Les attentes pour la soutenance intermédiaire sont : travail de recherche d'information bien avancé, modélisation / conception des structures de données et des algorithmes bien avancées, un prototype qui fonctionne sur un périmètre fonctionnel restreint (par exemple pouvoir charger et jouer à un labyrinthe 2D à maille carrée, sans monstres ni bonus), une équipe où les rôles ont été partagés relativement à un plan de réalisation défini.



### 3.1.4 TP d'appui intermédiaire (TPP2)

Un TP d'appui intermédiaire aura lieu dans la semaine, au pire dans la quinzaine qui suit la soutenance intermédiaire.

**Groupe A** le samedi 30 novembre matin.

**Groupe B** le samedi 30 novembre matin.

**Groupe C** le samedi 30 novembre matin.

**Groupe PL** le samedi 7 décembre matin.

### 3.1.5 Soutenance finale (S2)

Soutenance finale, avant les vacances de Noël.

**Groupe A** le samedi 14 décembre matin et le mardi 17 décembre après-midi.

**Groupe B** le samedi 14 décembre matin et le mardi 17 décembre après-midi.

**Groupe C** le samedi 14 décembre matin et le mardi 17 décembre matin.

**Groupe PL** le mardi 17 décembre matin (et début d'après-midi).

Rappelons qu'une soutenance est d'abord un exercice de communication. Il faut savoir montrer et mettre en valeur en quelques minutes des heures de travail de conception et de réalisation. Un tel exercice se prépare. La présentation vivante d'une fonctionnalité restera toujours plus intéressante qu'une ligne de code inanimée. Pour relater les phases d'élaboration, de réflexion et de conception, c'est-à-dire vous en phase d'ingénierie face au problème posé, un diagramme conceptuel (mind diagram) peut être un choix judicieux et un support de communication plus efficace qu'un jeu de transparents.

## 3.2 Constitution des équipes

Seuls les binômes (deux étudiants d'un même groupe de TD) sont autorisés. Il peut y avoir, à titre dérogatoire, un unique trinôme dans les groupes de TD dont le nombre d'étudiants est impair. La constitution de l'unique trinôme éventuel est laissée à la discrétion du délégué du groupe de TD.

Les délégués de groupe sont chargés de produire une liste des équipes et des ordres de passage sur les plages horaires réservées pour les soutenances.

Il vous faudra avoir indiqué votre équipe à votre délégué avant le **vendredi 18 octobre au soir, au plus tard**. Au delà, votre équipe sera constituée d'office.

En cas de dysfonctionnement problématique au sein d'une équipe, n'attendez pas le dernier moment pour l'évoquer : votre tuteur peut vous aider à trouver une solution.

## 3.3 IHM et technologies tierces parties

Vous pouvez utiliser les technologies de votre choix.

Les technologies tierces parties employées devront être référencées dans votre livrable, et notamment rappelées dans les annexes de votre rapport.

Il est interdit d'utiliser des bibliothèques existantes pour vos SDD et algorithmes clés (en d'autres termes, pas de librairie, sauf librairie graphique).

Notez enfin que l'appréciation de l'IHM n'est qu'un constituant accessoire de la note. Évitez d'y investir trop de temps, au détriment de la recherche de la meilleure performance algorithmique. Chaque année, malgré cette mise en garde, de nombreux élèves sont frustrés par une piètre rétribution après un investissement significatif sur cette partie.

Cependant, les meilleures réalisations seront mises en compétition en vue de leur présentation aux JPO (Journées Portes Ouvertes) 2014 : cette compétition permettra aux meilleures équipes d'acquérir jusqu'à 2 pts bonus sur leur note de projet.

Pour qui souhaite s'y essayer, quelques suggestions de librairies graphiques qui ne valent pas prescription :

**C** GTK

**C++** GTK+, Qt

**Java** classes JFC Swing (pas AWT)

**Web** regarder du côté des technologies WebGL

## 3.4 Évaluation

### 3.4.1 Comment gagner des points

Critères par ordre décroissant d'importance :

- Le programme fait ce qu'on attend de lui (partiellement, complètement, avec des plus).
- L'interaction utilisateur est ergonomique (facilité d'utilisation) et esthétique.
- Le code source est efficace (SDD, algorithmes efficaces).
- Le code source est propre (indentations, identifiants intelligibles, commentaires bien dosés).

### 3.4.2 Comment perdre des points

Critères par ordre décroissant de gravité :

- Même code, même partiel, livré par plusieurs équipes : 0 pour les équipes impliquées.
- Retard de livraison : 2 points par jour au *prorata temporis*.
- Non respect des consignes de packaging<sup>8</sup> du livrable : jusqu'à -3 pts suivant les cumuls.

### 3.4.3 Travail minimum pour espérer avoir 10

Le périmètre fonctionnel minimal consiste à permettre de jouer sans *intelligence artificielle* (résolution automatique). Il faut donc permettre le chargement du jeu dans sa configuration initiale, permettre au joueur de se déplacer en respectant les contraintes de déplacement, de détecter la fin de partie.

Ce périmètre fonctionnel minimum permet d'être noté sur 12 points : obtenir 12 serait alors la marque d'une réalisation parfaite sur ce périmètre (fonctionnement, ergonomie, etc). En d'autres termes, en l'absence de résolution automatique, il sera bien difficile d'obtenir la moyenne.

---

8. voir section suivante.



## 4 Packaging du livrable

### 4.1 Contenu du livrable

- Vos fichiers sources.
- SURTOUT PAS D'EXÉCUTABLE!
- Un script de compilation, déploiement et paramétrage pour Win32 accompagné d'un manuel d'installation minimal mais suffisant pour ne pas y passer plus de trois minutes.
- Vos fichiers de paramétrage et de configurations qui décrivent les différents jeux supportés par votre plate-forme (votre canevas).
- Votre documentation (Rapport et autres documents) au format PDF.
- Un fichier `README.txt` qui comprend impérativement deux sections :
  - Un état d'avancement honnête et fiable de vos travaux en 10 lignes maximum.
  - Les références aux sources qui ont pu inspirer votre conception, la reprise de code étant formellement interdite.
- Et en option, une mise en avant de fonctionnalités supplémentaires que vous aurez choisi de réaliser dans le but d'obtenir une meilleure note.

### 4.2 Contenu du rapport

Un **sommaire** détaillé avec pages numérotées.

**L'analyse fonctionnelle générale** D'environ cinq pages, elle explique quelles sont les principales données traitées et quelles sont les principales fonctionnalités du programme, comment on peut les regrouper, comment elles sont organisées dans le programme. Cette analyse aboutit à la présentation des modules fonctionnels qui sont décrits en détail dans l'analyse détaillée. L'analyse générale doit préciser le ou les choix importants que vous avez faits pour la conception de votre projet.

**L'analyse fonctionnelle détaillée** D'environ dix pages, elle précise, pour chaque module, son utilité, ses entrées et ses sorties, les données qu'il modifie. Il faut décrire les données utilisées et les algorithmes mis au point (logique de manipulation) avec rigueur, concision et clarté, le tout accompagné de schémas. **Cette description est abstraite<sup>9</sup> et s'appuie sur le langage algorithmique utilisé en cours et en TD.** C'est un travail de synthèse, de pédagogie et de communication ; les extraits de code sont proscrits de vos rapports.

**Annexes** Elles présentent au moins les technologies tierces employées. Et comme option recommandée, bibliographie et webographie listent les références qui ont permis d'étayer votre analyse et votre conception.

### 4.3 Livraison

**Une seule remise des travaux par équipe!!** (la double remise peut être sanctionnée dans la composante rigueur / discipline de la note). Remise au plus tard le **vendredi 13 décembre 2013**. Dès le lendemain, application d'une pénalité de 2 points par jour (au *prorata temporis*). Le livrable se présente sous la forme d'une archive RAR de nom : **PROJET SDD L2 GP - NOM 1**

---

<sup>9</sup>. algorithmes indépendants de la représentation interne des données. Surtout pas un pseudo C pour se donner bonne conscience.

- *NOM 2.RAR* envoyé par email à l'adresse de votre tuteur de projet, ou par tout autre moyen de substitution qu'il vous indiquerait.

**Objet** PROJET SDD L2 GP - *NOM 1* - *NOM 2*

Kaïs Klai (groupe A) : [kais.klai@lipn.univ-paris13.fr](mailto:kais.klai@lipn.univ-paris13.fr)

Franck Lepoivre (groupes B et PL) : [franck.lepoivre@gmail.com](mailto:franck.lepoivre@gmail.com),

Michel Landschoot (groupe C) : [mlandsnet@yahoo.fr](mailto:mlandsnet@yahoo.fr),

## 5 Annexes 1. Références

### 5.1 Bibliographie

#### 5.1.1 Théorie combinatoire des jeux



FIGURE 3 – Le labyrinthe du parc du château de Thoiry

### 5.2 Webographie

#### 5.2.1 Premières lectures

<http://www.astrolog.org/labyrnth/algrithm.htm>

#### 5.2.2 Étymologie / définition de *labyrinthe*

– <http://www.cnrtl.fr/etymologie/labyrinthe>

#### 5.2.3 Pages Wikipedia liées au sujet

- 
- <http://fr.wikipedia.org/wiki/Labyrinthe>
- <http://en.wikipedia.org/wiki/Labyrinth>
- <http://fr.wikipedia.org/wiki/Dédale>
- <http://en.wikipedia.org/wiki/Daedalus>
- [http://fr.wikipedia.org/wiki/Livre\\_de\\_Kells](http://fr.wikipedia.org/wiki/Livre_de_Kells)
- <http://fr.wikipedia.org/wiki/Minotaure>
- <http://en.wikipedia.org/wiki/Maze>
- <http://www.vopus.org/fr/gnose/anthropologie-gnostique/le-mystere-des-labyrinthes.html>

#### 5.2.4 Modèles et algorithmes labyrinthiques

- [http://fr.wikipedia.org/wiki/Modélisation\\_mathématique\\_d'un\\_labyrinthe](http://fr.wikipedia.org/wiki/Modélisation_mathématique_d'un_labyrinthe)
- [http://en.wikipedia.org/wiki/Maze\\_generation\\_algorithm](http://en.wikipedia.org/wiki/Maze_generation_algorithm)
- <http://ilay.org/yann/articles/maze/> : algorithme pratique et optimisation de code : la génération de labyrinthes
- <http://www.astrolog.org/labyrnth/algrithm.htm>

### 5.2.5 Répertoires de labyrinthes

Annuaire labyrinthes végétaux vus du ciel : [http://www.loisi-nature.com/Labyrinthes-vegetaux\\_a46.html](http://www.loisi-nature.com/Labyrinthes-vegetaux_a46.html)

## **6 Annexe 2. Histoire des labyrinthes**

## **7 Annexe 3. Répertoire de labyrinthes**



## **8 Annexe 4. Format pour la représentation des labyrinthes et des chemins**

Ce format de représentation est une convention simple qui permet de nous entendre sur le format des fichiers de configuration de vos jeux. Il permettra notamment de vous mettre en compétition, de tester vos réalisations sur la base des mêmes fichiers.

### **8.1 Exemple simple avec ...**

### **8.2 Exemple plus complexe avec ...**