

## TD - Projet Zeldiabolo - Liste des fonctionnalités

---

Ce document liste les fonctionnalités possibles pour votre projet. Choisissez quelques fonctionnalités parmi celles-ci pour construire la prochaine itération.

### Guide pour le projet

La démarche à suivre est à la suivante

- **30min** Choix de 1/2 fonctionnalités
  - ☐ ces fonctionnalités seront à mettre en œuvre à la prochaine version ;
  - ☐ concentrez vous sur celles qui paraissent les plus importantes ;
  - ☐ Les critères de validation à respecter sont décrits avec chaque fonctionnalité.
- **1h** Conception
  - ☐ pour chaque fonctionnalité, écrire un diagramme de séquence ;
  - ☐ créer un nouveau diagramme de classe à partir du diagramme de la version précédente et de vos diagrammes de séquence pour inclure les nouvelles fonctionnalités ;
  - ☐ se mettre d'accord sur les diagrammes au sein du groupe ;
  - ☐ partager le travail entre les membres du groupe.
- **1h30** Implémentation
  - ☐ écrire les tests ;
  - ☐ écrire les commentaires puis le code et valider avec les tests.
- **1h** Intégration
  - ☐ valider les nouvelles fonctionnalités et vérifiez que les tests précédents ne sont pas cassés ;
  - ☐ écrire un texte bilan (fonctionnalités validées, problèmes rencontrés, modification de la conception initiale)
  - ☐ ajouter un tag sur la version et passer à la suivante.

Votre projet sera évalué en fonction de la conception de votre application et du respect de la démarche (justesse des documents de conception, tests unitaires, partage du travail, etc ...).

A chaque itération, vous **devez** adapter les fonctionnalités et les critères de validation à votre projet.

Vous ferez attention à choisir des fonctionnalités intéressantes pour mettre en valeur votre projet. Il NE s'agit PAS de faire 5 sous-classes de monstre presque identiques sans ajout majeur.

# Liste des fonctionnalités possibles

## Table des matières

<b>1</b>	<b>Gestion des Labyrinthes</b>	<b>3</b>
1.1	Création d'un labyrinthe par défaut . . . . .	3
1.2	Collision avec les Murs . . . . .	3
1.3	Gestion des niveaux . . . . .	3
1.4	Cases déclencheur . . . . .	3
1.5	Cases piégées . . . . .	4
1.6	Ouverture de Passages secrets . . . . .	4
1.7	Fermeture de Passage . . . . .	4
1.8	Gestion d'un labyrinthe multi-étages . . . . .	5
1.9	Génération automatique de labyrinthe . . . . .	5
<b>2</b>	<b>Comportement des monstres</b>	<b>6</b>
2.1	Génération de monstres immobiles . . . . .	6
2.2	Déplacement aléatoire des monstres . . . . .	6
2.3	Monstres attirés par le héros . . . . .	6
2.4	Monstres au comportement intelligent . . . . .	7
2.5	Monstres avec intelligence variable . . . . .	7
<b>3</b>	<b>Monstres et Bestioles en tout genre</b>	<b>9</b>
3.1	Collision avec un monstre . . . . .	9
3.2	Fantômes . . . . .	9
3.3	Troll . . . . .	9
3.4	Serpent . . . . .	9
<b>4</b>	<b>Mon ami gélatineux</b>	<b>11</b>
4.1	Blob gélatineux immobile . . . . .	11
4.2	Propagation de blob . . . . .	11
4.3	Blob l'éplonge . . . . .	11
4.4	Déplacement du cerveau du Blob . . . . .	12
<b>5</b>	<b>Gestion des attaques</b>	<b>13</b>
5.1	Attaque des monstres . . . . .	13
5.2	Attaque du joueur . . . . .	13
5.3	Affichage des attaques . . . . .	13
5.4	Attaque directionnelle . . . . .	13
5.5	Mort des monstres . . . . .	14
<b>6</b>	<b>Gestion de la fin du jeu</b>	<b>15</b>
6.1	Fin du jeu : mort du héros . . . . .	15
6.2	Mise en place de l'amulette . . . . .	15
6.3	Acquisition de l'amulette . . . . .	15
6.4	Fin du jeu : victoire du héros . . . . .	15

<b>7</b>	<b>Inventaire</b>	<b>17</b>
7.1	Inventaire . . . . .	17
7.2	Armes . . . . .	17
7.3	Boucliers . . . . .	17
7.4	Affichage de l'inventaire . . . . .	18
<b>8</b>	<b>Attaque à distance</b>	<b>19</b>
8.1	Attaque à distance du joueur . . . . .	19
8.2	Vitesse des flèches . . . . .	19
8.3	Jet de pierre . . . . .	19
<b>9</b>	<b>Jeu de type Bomberman</b>	<b>21</b>
9.1	Déclenchement de Bombes à blanc . . . . .	21
9.2	Dégâts d'une bombe . . . . .	21
9.3	Dépôt d'une bombe . . . . .	21
9.4	Murs friables . . . . .	22
9.5	Monstres incendiaires . . . . .	22
9.6	Monstres esquivant de bombe . . . . .	22
9.7	Poussage de bombe . . . . .	22
9.8	Déplacement de bombe . . . . .	23
<b>10</b>	<b>Jeu de type Sokoban</b>	<b>24</b>
10.1	Mur Déplaçable . . . . .	24
10.2	Probleme de Sokoban . . . . .	24
10.3	Sol verglacé . . . . .	24
10.4	Propulsion de mur déplaçable . . . . .	25
10.5	Ecrasage de monstre . . . . .	25
<b>11</b>	<b>Interface utilisateur</b>	<b>26</b>
11.1	Affichage avec des sprites . . . . .	26
11.2	Configuration du clavier . . . . .	26
11.3	Gestion d'un menu . . . . .	26

# 1 Gestion des Labyrinthes

## 1.1 Création d'un labyrinthe par défaut

**Difficulté :** ★☆☆☆☆

Au lancement du jeu, un labyrinthe par défaut est créé et le personnage est placé au centre de celui-ci.

### Critères de validation

- Le labyrinthe par défaut est constitué d'une arène de taille 10\*10 ;
- Le personnage est placé au centre de l'arène ;
- Les monstres (s'il y en a) sont placés sur des cases vides (à adapter).

## 1.2 Collision avec les Murs

**Difficulté :** ★★☆☆☆

Lorsque le joueur demande à déplacer le personnage en direction d'un mur, le personnage reste bloqué et ne se déplace pas.

### Critères de validation

- La collision fonctionne dans les 4 directions ;
- Si la case est vide, le personnage avance.

## 1.3 Gestion des niveaux

**Difficulté :** ★★☆☆☆

Le jeu est constitué de plusieurs niveaux.

### Critères de validation

- Un niveau décrit le labyrinthe et les monstres qui y sont contenus.
- Il est possible de choisir le niveau à lancer au démarrage de l'application (en modifiant la chaîne de caractère passée dans le main).
- Testez en ajoutant un second niveau à définir au préalable.

## 1.4 Cases déclencheur

**Difficulté :** ★★★☆☆

Lorsque une entité arrive sur certaines cases, ces cases déclenchent un effet.

### Critères de validation

- Les cases avec effet sont des cases traversables.
- Lorsqu'un personnage (héros ou monstre) marche sur cette case, elle déclenche immédiatement un effet
- cet effet se traduit pour le moment, par un affichage dans la console à préciser.

## 1.5 Cases piégées

Difficulté : ★★★☆☆

Lorsque une entité (héros ou monstre) arrive sur une case piégée, il subit des dégâts.

### Critères de validation

- Les dégâts ne se déclenchent que lorsque le joueur arrive sur la case, il peut ensuite rester sur cette case aussi longtemps qu'il le souhaite sans subir d'autres dégâts.
- Lorsque la case devient à nouveau vide, le piège se réamorç.
- (optionnel) Lorsqu'une case a été détectée comme un piège, elle s'affiche différemment dans le labyrinthe (en brun) ce qui permet au joueur d'en tenir compte ensuite.

## 1.6 Ouverture de Passages secrets

Difficulté : ★★★★★☆

Lorsque une entité (héros ou monstre) arrive sur une case d'ouverture, le labyrinthe se modifie et la porte liée à cette case (située à un autre endroit dans le labyrinthe) s'ouvre.

Les portes peuvent donc constituer des pièges cachés puisque une case peut ouvrir les portes qui protègent le héros d'une horde de monstres.

### Critères de validation

- Une porte est un type de case particulière. Une porte n'a pas d'orientation et occupe toute une case.
- Une porte peut être fermée ou ouverte. Une porte fermée bloque le trajet et une porte ouverte est traversable.
- La porte s'ouvre au passage du personnage sur la case qui déclenche son ouverture.
- Une fois la porte ouverte, elle reste ouverte.
- Les portes sont affichées d'une manière particulière : elles sont de couleur noire et sont représentées par des carrés qui occupent toute la case. Le carré est plein ou vide en fonction de l'état de la porte (ouverte/fermée).

## 1.7 Fermeture de Passage

Difficulté : ★★★★★☆

Lorsque une entité (héros ou monstre) arrive sur une case de type fermeture, le labyrinthe se modifie et la porte liée à cette case se referme. Cela permet de faire des pièges retors (une case ouvre la porte, la case derrière la porte la referme).

### Critères de validation

- Cette case a l'effet inverse de la case précédente.
- Une porte fermée reste fermée.

- Une même porte peut être manipulée par plusieurs cases (de type fermeture/ouverture).

## 1.8 Gestion d'un labyrinthe multi-étages

**Difficulté :** ★★★★★☆

Le labyrinthe est constitué de plusieurs étages. Chaque étage correspond à un niveau particulier. Des escaliers permettent de passer d'un étage à un autre.

### Critères de validation

- Les escaliers sont des cases traversables par le héros et les monstres.
- Lorsque le héros se trouve sur un escalier, il peut l'activer et il change alors de niveau.
- Les monstres du niveau où le héros se trouvait avant l'escalier n'apparaissent pas dans l'étage supérieur/inférieur. Ils restent à leur position et conservent leur points de vie.
- De nouveaux monstres peuvent être présents à l'étage supérieur en fonction du descriptif du niveau.
- Le Héros arrive aux mêmes coordonnées dans l'étage supérieur/inférieur après avoir emprunté les escaliers.
- Les escaliers peuvent être de deux types : un escalier qui monte ou un escalier qui descend.
- Les escaliers sont empruntables dans les deux sens : un escalier qui monte vers un étage possède un escalier qui descend à la même position dans l'étage supérieur (et inversement).

## 1.9 Génération automatique de labyrinthe

**Difficulté :** ★★★★★★

Les niveaux ne sont pas stockés en mémoire, mais le labyrinthe est généré automatiquement par le jeu (ce qui permet d'avoir des parties à chaque fois différentes)

### Critères de validation

- Les monstres, les objets et le héros sont bien placés sur des cases vides
- Vous pourrez vous inspirer de la page suivante  
<http://www.encyclopédie-incomplete.com/?Modelisation-et-Creation-d-un>
- les critères de validation sont à détailler en fonction de la méthode de génération suivie.

## 2 Comportement des monstres

### 2.1 Génération de monstres immobiles

Difficulté : ★☆☆☆☆

Au lancement du jeu, le jeu ajoute différents monstres dans l'environnement. Les monstres restent immobiles.

#### Critères de validation

- Il est possible de construire un labyrinthe par défaut contenant quelques monstres (le labyrinthe par défaut est à définir).
- Les monstres sont créés sur des cases vides.
- Il n'est pas possible de générer des monstres sur des cases non vides (position du joueur, d'autres monstres ou de murs)
- Les monstres doivent apparaître dans l'interface graphique sous la forme d'un cercle rouge.
- Les monstres ne se déplacent pas (mais attaquent éventuellement le héros s'il passe à proximité - cf taches *attaque* dans les parties suivantes).

### 2.2 Déplacement aléatoire des monstres

Difficulté : ★☆☆☆☆

Régulièrement, le jeu demande aux monstres de se déplacer dans une direction aléatoire.

#### Critères de validation

- A chaque évolution du jeu, que le joueur se déplace ou non, les monstres se déplacent.
- Les monstres sont bloqués par les obstacles. S'ils ont choisi de se diriger vers un obstacle (mur, joueur, ...), leur mouvement est alors annulé (sauf s'il s'agit de fantômes).
- Deux monstres ou un monstre et un joueur ne peuvent pas se trouver sur la même case (Si la tache *Collision entre monstre* est fonctionnelle).
- A chaque évolution, tous les monstres tentent de se déplacer.
- L'interface graphique se met à jour en fonction des déplacements des Monstres.

### 2.3 Monstres attirés par le héros

Difficulté : ★★☆☆☆

A chaque fois que les monstres se déplacent, les monstres tentent de réduire leur distance à vol d'oiseau par rapport au héros.

### Critères de validation

- Après son déplacement, un monstre doit être plus proche du héros qu'avant son déplacement (distance de Manhattan sans prise en compte des murs).
- Les monstres soumis aux obstacles peuvent se retrouver bloqués dans des cul-de-sac.

## 2.4 Monstres au comportement intelligent

Difficulté : ★★★★★☆

A chaque fois que les monstres se déplacent, ils se rapprochent dans la direction du héros en prenant en compte la présence des murs.

### Critères de validation

- Avant son déplacement, le monstre détermine le meilleur chemin pour atteindre le héros.
- Le monstre prend en compte les obstacles qui le concernent pour trouver le meilleur chemin.
- lorsqu'il se déplace, le monstre suit effectivement le meilleur chemin et parvient jusqu'au héros même s'il y a des obstacles.
- vous pourrez vous inspirer (a) des transparents *"IA dans le jeu vidéo"* (transparents 112-130) ou (b) (c) du site redblob (avec de nombreuses démonstrations autour du jeu vidéo)
  - (a) [https://members.loria.fr/VThomas/mediation/JV\\_IUT\\_2016/#ia](https://members.loria.fr/VThomas/mediation/JV_IUT_2016/#ia)
  - (b) <http://www.redblobgames.com/pathfinding/a-star/introduction.html>
  - (c) <http://www.redblobgames.com/pathfinding/tower-defense/>

## 2.5 Monstres avec intelligence variable

Difficulté : ★★★★★☆

En plus d'être de type différents (orc, troll, fantôme), les monstres peuvent avoir des degré d'intelligence différent. En fonction de ce degré d'intelligence, ils se déplacent différemment.

### Critères de validation

- Un monstre d'intelligence nulle reste immobile.
- Un monstre d'intelligence faible se déplace au hasard.
- Un monstre d'intelligence moyenne cherche à se rapprocher du héros sans prendre en compte les obstacles.
- Un monstre d'intelligence forte se rapproche du héros en évitant les obstacles.
- Plusieurs monstres avec des intelligences différentes peuvent être présents dans le même labyrinthe.
- Chaque type de monstre (Orc, Fantome, Troll) peut avoir n'importe quel type d'intelligence. On peut avoir dans le même labyrinthe des Orc d'intelligence nulle, faible, moyenne et forte. Idem pour les trolls et les fantômes.



- pour tester que cela fonctionne, il faut créer un seul labyrinthe contenant tous les couples (type de monstre/intelligence).

## 3 Monstres et Bestioles en tout genre

### 3.1 Collision avec un monstre

Difficulté : ★★☆☆☆

Lorsque le joueur demande au personnage de se déplacer vers une case contenant un monstre, le personnage ne se déplace pas.

#### Critères de validation

- La collision fonctionne dans les 4 directions.
- Si la case est vide, le personnage avance.
- Le monstre se déplace à chaque fois après le joueur.

### 3.2 Fantômes

Difficulté : ★★☆☆☆

Le labyrinthe dispose d'un nouveau type de monstre : les fantômes.

#### Critères de validation

- Les fantômes peuvent traverser des murs mais pas les autres personnages (les autres monstres ou le héros)
- Les autres monstres continuent à être bloqués par des murs.
- Les attaques des fantômes fonctionnent de la même manière que les attaques des monstres de base.

### 3.3 Troll

Difficulté : ★★☆☆☆

Les trolls sont un nouveau type de monstre. Un troll se régénère au fur et à mesure.

#### Critères de validation

- Un troll se déplace de la même manière qu'un monstre de base (les murs sont des obstacles).
- Si un Troll ne prend aucun dégât pendant un tour de jeu, il regagne un point de vie.
- les attaques des trolls fonctionnent de la même manière que les attaques des monstres de base.
- Un troll a par défaut 3 points de vie (à adapter).

### 3.4 Serpent

Difficulté : ★★★★★

Les serpents sont un nouveau type de monstre inspiré de sbake (cf [https://fr.wikipedia.org/wiki/Snake\\_\(jeu\\_vid%C3%A9o\)](https://fr.wikipedia.org/wiki/Snake_(jeu_vid%C3%A9o))). Un serpent occupe plusieurs cases, lorsqu'il se déplace, sa tête avance d'une case et son corps suit.

**Critères de validation**

- A chaque fois qu'un serpent se déplace, sa tête avance d'une case et sa queue se réduit d'une case.
- Toutes les cases occupées par le serpent sont non traversables et sont affichées à l'écran.
- Le corps d'un serpent peut avoir des virages (si le serpent avance en tournant).
- Un serpent ne peut pas avancer sur une case occupée (y compris sa queue). Si le serpent n'a pas de case libre adjacente, il ne se déplace pas. Un serpent peut donc se retrouver bloqué par sa propre queue.
- Toutes les cases du serpent sont attaquables par le héros. Dès que le serpent n'a plus de point de vie, il disparaît entièrement.

## 4 Mon ami gélatineux

Cette partie s'inspire d'un monstre de DROD (Deadly rooms of death) [http://drod.wikia.com/wiki/Tarstuff\\_mother\\_\(RPG\)](http://drod.wikia.com/wiki/Tarstuff_mother_(RPG)).

### 4.1 Blob gélatineux immobile

**Difficulté :** ★★★☆☆

Les blobs sont nouveau type de monstre. Ils peuvent occuper plusieurs cases et ne se déplacent pas. Chaque partie du blob peut être détruite.

#### Critères de validation

- Un blob peut occuper plusieurs cases dans le labyrinthe. Ces cases sont non traversables.
- Lorsqu'un blob doit se déplacer, il ne fait rien.
- Si un héros attaque une case occupée par le blob, cette partie du blob est détruite et la case devient à nouveau libre.
- Si le blob n'a plus de case occupée, celui-ci est entièrement détruit.

### 4.2 Propagation de blob

**Difficulté :** ★★★★★☆

A chaque pas de temps, plutôt que de se déplacer, les blobs grossissent, se répandent et envahissent progressivement le labyrinthe.

#### Critères de validation

- Contrairement aux blobs immobiles, quand un blob doit se déplacer, il se propage. Toutes les cases voisines libres des cases occupées par le blob deviennent des cases occupées par le blob.
- Si un héros attaque une case occupée par le blob, cette partie du blob est détruite et la case devient à nouveau libre.
- Lorsque le blob est entièrement détruit (toutes les cases sont détruites), il disparaît et ne grossit plus.
- Pour éviter que le blob ne se reproduise trop vite, il ne se propage que tous les n appels à la méthode se déplacer(par exemple tous les 4 fois).

### 4.3 Blob l'éplonge

**Difficulté :** ★★★★★

Blob l'éplonge est un blob doté d'un cerveau. Si la case du cerveau du blob est détruite, le blob est détruit. Dès qu'une partie du blob n'est plus connectée à son cerveau cette partie disparaît.

#### Critères de validation

- Les règles des blobs précédents s'appliquent.
- Le cerveau d'un blob est situé à case particulière occupée par le blob.
- Lorsque le cerveau d'un blob est détruit, le blob est détruit.
- Le blob ne peut être que d'un seul tenant : dès qu'une partie du blob n'est plus connectée au cerveau par du blob, cette partie est détruite.

### 4.4 Déplacement du cerveau du Blob

Difficulté : ★★★☆☆

Lorsqu'il se propage, le blob a en plus la possibilité de déplacer la position de son cerveau d'une case (pour se protéger du joueur).

#### Critères de validation

- Les règles des blobs à cerveaux s'appliquent.
- A chaque déplacement, le blob peut déplacer son cerveau d'une case vers une case libre.
- Après son déplacement, le cerveau du blob doit se trouver sur une case occupée par le blob correspondant.
- Le blob est détruit dès que son cerveau est détruit.
- Toute partie non reliée au cerveau est automatiquement détruite.

## 5 Gestion des attaques

### 5.1 Attaque des monstres

**Difficulté :** ★★☆☆☆

Lorsqu'un monstre doit agir, si le monstre se trouve à côté du héros, alors il ne se déplace pas mais il attaque celui-ci.

#### Critères de validation

- Lorsque le monstre est à côté du joueur, il attaque et ne se déplace pas. Par défaut, le Héros perd 1 point de vie.
- Lorsque le monstre est loin du joueur, il n'attaque pas mais se déplace.

### 5.2 Attaque du joueur

**Difficulté :** ★★☆☆☆

Lorsque le joueur appuie sur une touche donnée (par défaut la touche espace), il effectue une attaque contre les monstres qui se trouvent à côté de lui.

#### Critères de validation

- L'attaque a effectivement lieu quand le joueur appuie sur la touche.
- TOUS les monstres à côté du joueur prennent un point de dégât.
- TOUS les monstres éloignés du joueur ne subissent pas de dégâts.
- S'il n'y a aucun monstre autour du joueur, l'attaque a quand même lieu et empêche le déplacement du joueur, mais les monstres ne prennent pas de dégâts.

### 5.3 Affichage des attaques

**Difficulté :** ★★☆☆☆

Quand un monstre ou le héros attaque, il change de couleur.

#### Critères de validation

- La couleur d'affichage du monstre change au moment où il attaque (passe de rouge à noir).
- La couleur d'affichage du héros change au moment où il attaque (passe de bleu à Bleu foncé).
- Après l'attaque, le joueur et le monstre reprennent une couleur normale.

### 5.4 Attaque directionnelle

**Difficulté :** ★★★☆☆

Quand le joueur demande à attaquer, le héros n'attaque que devant lui et ne blesse que le monstre qui se trouve dans cette direction.

#### Critères de validation

- L'attaque a lieu dans la dernière direction suivie par le héros.
- L'attaque ne peut toucher au maximum qu'un monstre et seulement si ce monstre se trouve juste devant le personnage.
- Les monstres situés à côté du personnage dans une autre direction ne subissent pas de dégâts.
- Si aucun monstre n'est présent devant le héros, aucun monstre ne subit de dégâts.

### 5.5 Mort des monstres

Difficulté : ★★☆☆☆

Lorsqu'un monstre n'a plus de points de vie, il meurt et disparaît du jeu.

#### Critères de validation

- Les monstres ont par défaut 2 points de vie et ne peuvent pas avoir de points de vie négatifs.
- Lorsqu'un monstre par défaut est touché deux fois, il disparaît.
- Le héros attaque avant le monstre, si le monstre meurt, le monstre n'a pas le temps de répliquer et disparaît avant.

## 6 Gestion de la fin du jeu

### 6.1 Fin du jeu : mort du héros

Difficulté : ★☆☆☆☆

Lorsque le héros n'a plus de points de vie, le jeu s'arrête et la partie est perdue.

#### Critères de validation

- Le héros par défaut dispose de 5 points de vie.
- Chaque monstre fait 1 point de dégât lorsque le héros se fait attaquer.
- Quand le héros meurt, un message de fin est affiché sur la console et le jeu s'arrête.

### 6.2 Mise en place de l'amulette

Difficulté : ★★☆☆☆

Au lancement du jeu, une amulette est placée sur une case vide du labyrinthe. Le placement de l'amulette est toujours le même et dépend du niveau.

#### Critères de validation

- L'amulette ne peut être placée que sur une case vide.
- Les monstres et le héros peuvent se situer sur la case de l'amulette.
- L'amulette est affichée dans le jeu sous la forme d'un cercle jaune sur la case vide où elle se trouve.

### 6.3 Acquisition de l'amulette

Difficulté : ★★☆☆☆

Dès que le joueur demande à déplacer le héros sur l'amulette, le héros se déplace et prend l'amulette.

#### Critères de validation

- L'amulette n'est plus sur le plateau (et n'est plus affichée).
- Le Héros possède l'amulette.
- Un monstre ne récupère pas l'amulette.

### 6.4 Fin du jeu : victoire du héros

Difficulté : ★★☆☆☆

Une fois que le héros a pris l'amulette, il peut retourner à l'entrée du labyrinthe et remporter le jeu.



**Critères de validation**

- Lorsque le héros possède l'amulette et qu'il retourne à l'entrée du labyrinthe, le jeu s'arrête et un message est affiché dans la console.
- Lorsque le héros ne possède pas l'amulette, le jeu ne s'arrête pas même si le héros retourne à l'entrée.

## 7 Inventaire

### 7.1 Inventaire

**Difficulté :** ★★☆☆☆

Des objets (sans utilité pour le moment) sont disposés dans des cases vides du labyrinthe, lorsque le héros appuie sur la touche d'utilisation ("E" par défaut), il récupère les objets de la case où il se trouve et les ajoute dans son inventaire.

#### Critères de validation

- L'inventaire est de taille infinie.
- Lorsque le joueur cherche à prendre un objet alors qu'il n'y a rien sur la case où il se trouve, rien ne se passe.
- Lorsqu'un joueur prend un objet, l'objet disparaît du labyrinthe (et n'est plus affiché) mais apparaît dans son inventaire.
- A chaque évolution du jeu, le jeu affiche dans la console l'inventaire du héros.
- Les objets sont affichés dans le labyrinthe tant qu'ils sont présents (sous la forme d'un cercle noir)

### 7.2 Armes

**Difficulté :** ★★☆☆☆

Les armes sont des objets que le héros peut prendre. Une arme augmente d'un certain nombre les dégâts du héros.

#### Critères de validation

- Une arme est définie par ses dégâts.
- Lorsque le héros prend une arme, l'arme est ajoutée à son inventaire et disparaît du labyrinthe.
- Lorsqu'un héros possède une arme et attaque un monstre, son attaque est augmentée des dégâts de l'arme.
- Un héros ne peut utiliser qu'une seule arme à la fois. Lorsqu'un héros possède plusieurs armes dans son inventaire, il s'équipe automatiquement de l'arme la plus forte.

### 7.3 Boucliers

**Difficulté :** ★★☆☆☆

Les boucliers sont des objets que le héros peut prendre. Un bouclier fournit une protection temporaire et augmente les dégâts que le héros peut subir.

#### Critères de validation

- Un bouclier est défini par sa protection.
- Lorsque le héros prend un bouclier, le bouclier est ajouté à son inventaire et disparaît du labyrinthe.

- Lorsqu'un héros subit des dégâts, c'est d'abord la protection du bouclier qui diminue avant que le joueur ne perde des points de vie.
- Lorsqu'un bouclier n'a plus de protection, il est détruit.
- Un héros ne peut utiliser qu'un seul bouclier à la fois. Lorsqu'un joueur reçoit un coup, il s'équipe automatiquement du bouclier qui est le plus protecteur à ce moment.

## 7.4 Affichage de l'inventaire

**Difficulté :** ★★★☆☆

L'inventaire est affiché dans la fenêtre de jeu.

### Critères de validation

- L'inventaire est représenté par 6 cases carrées en bas de la fenêtre de jeu
- A chaque fois qu'un joueur prend un objet, en plus de disparaître dans le labyrinthe, celui-ci s'affiche dans l'inventaire
- Les objets sélectionnés (meilleure arme et meilleur bouclier) sont d'une couleur différente.
- Les objets affichés dans l'inventaire ont des formes spécifiques (rond pour un bouclier et un trait vertical pour une arme).
- Lorsqu'un bouclier est détruit, il disparaît de l'inventaire.

## 8 Attaque à distance

### 8.1 Attaque à distance du joueur

**Difficulté :** ★★★★★☆

Le joueur dispose d'une touche particulière pour tirer à l'arc. Lorsqu'il appuie sur cette touche ("f" par défaut), le héros lance une flèche qui traverse l'écran dans la direction suivie par le héros.

#### Critères de validation

- La flèche se déplace de manière rectiligne à partir du héros et selon la direction de son dernier déplacement.
- la flèche se déplace à la vitesse des monstres et du héros (cf tâche suivante)
- Lorsqu'elle rencontre un obstacle (monstre ou mur), la flèche disparaît et fait 1 point de dégât si c'est un monstre.
- Tant qu'elle n'est pas détruite, la flèche se déplace d'une case par évolution du jeu.
- La flèche est affichée dans la fenêtre de jeu sous la forme d'un petit point rouge centré dans la case où la flèche se trouve.

### 8.2 Vitesse des flèches

**Difficulté :** ★★☆☆☆☆

Les flèches lancées par le joueur vont deux fois plus vite que les déplacements des monstres et du joueur.

#### Critères de validation

- A chaque évolution du jeu, la flèche se déplace de deux cases.
- La détection d'obstacles se fait dans toutes les cases traversées (pas juste la première et la dernière)

### 8.3 Jet de pierre

**Difficulté :** ★★★★★☆

Les gobelins sont un nouveau type de monstre qui ne se battent qu'à distance en jetant des pierres.

#### Critères de validation

- Les jets de pierre fonctionnent de la même manière qu'un tir de flèche, mais ce sont les gobelins qui peuvent les faire.
- Un goblin ne jette une pierre dans la direction du héros que s'il y a une ligne directe entre les deux sans aucun obstacle.
- si la pierre touche le héros, il perd 1 point de vie.

- Tant qu'un gobelin a une vue direct sur le héros, il ne se déplace pas mais tire une pierre.
- Si le héros n'est plus visible, le gobelin reprend ses déplacements.

## 9 Jeu de type Bomberman

### 9.1 Déclenchement de Bombes à blanc

Difficulté : ★★★☆☆

Des bombes sont gérées par le jeu. Elles sont positionnées dans le labyrinthe, elles ont un timer, lorsque le temps restant arrive à 0, la bombe explose et disparaît.

#### Critères de validation

- La bombe est située sur une case vide du jeu.
- La bombe bloque le passage des joueurs et des monstres.
- La bombe dispose d'un décompte de temps qui diminue à chaque évolution du jeu.
- Lorsque le décompte arrive à 0, la bombe disparaît (cf fonctionnalités dégâts de bombe).
- Le temps restant est affiché sur la bombe.
- Après que la bombe ait explosé, la case est à nouveau traversable.

### 9.2 Dégâts d'une bombe

Difficulté : ★★★★★

Lorsqu'une bombe explose, des langues de feu jaillissent horizontalement et verticalement et font des dégâts sur les cases atteintes.

#### Critères de validation

- La bombe n'est déclenchée que quand son timer arrive à 0.
- Les langues de feu ne se propagent que horizontalement et verticalement (cf jeu bomberman).
- Les langues de feu ont une taille donnée (par exemple elles ne se propagent que de 3 cases dans chaque direction).
- Les murs et les obstacles bloquent la propagation des langues de feu.
- Tous les personnages traversés par les langues de feu subissent des dégâts.

### 9.3 Dépôt d'une bombe

Difficulté : ★★☆☆☆

Un Héros attaque désormais en posant des bombes sur sa case. Les bombes déposées sont en nombre limitées, leur timer est initialisé lorsqu'elles sont posées.

#### Critères de validation

- Lorsqu'un héros décide d'attaquer, il pose une bombe sur sa case.
- Le héros peut sortir de sa case, mais la case de la bombe n'est plus traversable tant que la bombe n'a pas explosé.

- Lorsque le timer de la bombe arrive à 0, la bombe explose et effectue des dégâts (cf autres fonctionnalités)
- Un héros dispose d'un nombre maximal de bombes posées simultanément : il ne peut pas poser plus de 3 bombes simultanément sur le labyrinthe.
- Une fois qu'une bombe a explosé, le héros peut à nouveau poser une bombe en remplacement sur le labyrinthe.

#### 9.4 Murs friables

**Difficulté :** ★★☆☆☆

Certaines cases du labyrinthe sont des murs friables qui peuvent être détruits par des bombes.

##### Critères de validation

- Un mur friable est un type de case du labyrinthe.
- Un mur friable n'est pas traversable (tant qu'il n'est pas détruit).
- Lorsqu'une explosion de bombe rencontre un mur friable, le mur friable empêche la propagation de l'explosion dans cette direction. Il est considéré comme détruit et peut être traversé.

#### 9.5 Monstres incendiaires

**Difficulté :** ★★★☆☆

Certains types de monstres peuvent poser des bombes sur les cases.

##### Critères de validation

- La pose des bombes est régie par les mêmes règles que celles de la pose d'une bombe pour le héros (cases bloquée, nombre maximal de bombes simultanément posées par héros, ...)

#### 9.6 Monstres esquivent la bombe

**Difficulté :** ★★★★★

Les monstres savent ce qu'est une bombe et vont essayer de se cacher de l'explosion.

##### Critères de validation

- Lorsqu'un monstre voit une bombe, il va avoir tendance à trouver une case où se protéger en attendant que la bombe explose.
- Le monstre peut utiliser des recherches de chemin (cf fonctionnalité IA avancée).
- Si le monstre n'a aucune possibilité de se cacher, il décide de ne rien faire.

#### 9.7 Poussage de bombe

**Difficulté :** ★★★☆☆

Un héros peut pousser une bombe qui se trouve devant lui.

#### Critères de validation

- le joueur dispose d'une action particulière (gérée par une touche spéciale) qui permet au héros de pousser une bombe devant lui.
- Une bombe ne peut être poussée que d'une case.
- Pour que la bombe puisse être poussée, il est nécessaire que la case derrière la bombe (par rapport au héros) soit disponible.
- Après que la bombe ait été poussée, son ancienne case redevient traversable et la nouvelle case devient non traversable.

### 9.8 Déplacement de bombe

Difficulté : ★★☆☆☆

Un héros peut prendre ou poser une bombe devant lui.

#### Critères de validation

- Le joueur dispose d'une action particulière (gérée par une touche spéciale) qui permet au héros de prendre une bombe devant lui dans ses mains et de se déplacer avec.
- Le joueur dispose d'une autre touche pour permettre au héros de poser la bombe devant lui.
- Lorsque le héros porte la bombe, la bombe ne bloque plus le passage.
- Le timer de la bombe continue à avancer et la bombe peut exploser quand le héros la porte.



## 10 Jeu de type Sokoban

### 10.1 Mur Déplaçable

Difficulté : ★★★☆☆

Certains murs sont des obstacles déplaçables. Lorsque le personnage se dirige sur une case contenant ce mur, le mur est poussé d'une case par le personnage.

#### Critères de validation

- Lorsque le personnage doit arriver sur un mur déplaçable et que la case derrière ce mur est vide, le mur est déplacé dans la case vide et le personnage se déplace ensuite.
- Le mur est déplacé dans la direction suivie par le personnage.
- Si la case derrière un mur est occupée (par un monstre, un mur ou un autre mur déplaçable), le mur ne peut pas être poussé et le personnage ne bouge pas.
- Les murs déplaçables sont représentés différemment des murs normaux dans l'interface graphique.

### 10.2 Probleme de Sokoban

Difficulté : ★★☆☆☆

Certains murs déplaçables sont des containers et certaines cases sont des cases d'arrivée. Le joueur remporte la partie si à un instant, tous les containers se situent sur des cases d'arrivée.

#### Critères de validation

- Les containers sont des murs déplaçables.
- La partie s'arrête uniquement quand tous les containers sont sur des cases d'arrivée.

### 10.3 Sol verglacé

Difficulté : ★★★☆☆

Certaines cases de sol sont recouvertes de glace. Lorsque le personnage se déplace dessus ou lorsqu'il y pousse un mur déplaçable, le mur continue à avancer dans la même direction jusqu'à un obstacle ou que le sol ne soit plus verglacé.

#### Critères de validation

- Les containers se comportent comme des murs déplaçables.
- Quand il glisse, un mur déplaçable s'arrête dès qu'il arrive sur une case normale.
- Quand il glisse, un mur déplaçable s'arrête dès qu'il rencontre un obstacle.

## 10.4 Propulsion de mur déplaçable

Difficulté : ★★★★★☆

Le personnage joueur dispose d'une action supplémentaire consistant à lancer un mur déplaçable en appuyant sur une touche dédiée à l'action (espace) et une touche directionnelle. Le mur se déplace dans la direction proposée par le personnage jusqu'à rencontrer un obstacle.

### Critères de validation

- La mur avance dans la direction proposée.
- Lorsqu'il s'arrête, la case derrière lui contient un obstacle (mur, mur déplaçable, monstre, ...)
- Lorsqu'on appuie sur la touche dédiée à la propulsion et qu'on appuie sur une touche de direction, le personnage ne se déplace pas (pour ne pas confondre avec le fait de propulser un mur avec le fait de se déplacer en poussant un mur).
- Si on propulse un mur alors qu'il a déjà un obstacle derrière lui, le mur ne bouge pas.
- Seuls les murs déplaçables peuvent être propulsés.

## 10.5 Ecrasage de monstre

Difficulté : ★★★★★☆☆

Lorsqu'un mur déplaçable est poussé (ou propulsé) et que la case d'arrivée contient un monstre, le mur est déplacé et ce monstre est détruit.

### Critères de validation

- Quand la case d'arrivée contient un monstre, le monstre est détruit, le mur se déplace sur sa case et s'arrête (s'il a été propulsé).
- Si la case d'arrivée est vide, le mur déplaçable est bien déplacé.

## 11 Interface utilisateur

### 11.1 Affichage avec des sprites

Difficulté : ★★★☆☆

La fenêtre n'affiche plus des figures de base (rond, ...) mais des sprites représentant les personnages et les objets ([http://fr.wikipedia.org/wiki/Sprite\\_%28jeu\\_vid%C3%A9o%29](http://fr.wikipedia.org/wiki/Sprite_%28jeu_vid%C3%A9o%29)).

#### Critères de validation

- Les sprites concernent les murs, les monstres et le héros.
- Ils sont chargés une fois pour toute au lancement de l'application et ils peuvent être affichés dans le `JPanel` grâce à la méthode `drawImage`.
- Vous pourrez utiliser l'image <http://www.qcfdesign.com/?p=354> qui est un ensemble de sprites pour le jeu `Desktop Dungeons`. Ces sprites sont disposés régulièrement dans l'image et sont donc facile à extraire à partir de l'`Image`.
- Il est possible de charger une image (de type `BufferedImage` en mémoire grâce à la classe `ImageIO` et la méthode statique `read`).

### 11.2 Configuration du clavier

Difficulté : ★★★☆☆

Il est possible de reconfigurer dans un fichier texte les touches pour contrôler le personnage.

#### Critères de validation

- Les touches sont décrites dans un fichier texte nommé "configTouches.txt"
- Si le fichier n'existe pas, ce sont les touches par défaut qui sont utilisées.
- le fichier de configuration est chargé une seule fois au lancement de l'application.

### 11.3 Gestion d'un menu

Difficulté : ★★★☆☆

Plutôt que de lancer directement le jeu, le joueur lance un menu qui lui permet de lancer le jeu ou de quitter l'application. Ce menu ré-apparaît lorsque le joueur gagne ou perd.

#### Critères de validation

- Le menu possède une image spécifique.
- Le menu propose deux options *jouer* ou *quitter*.
- Le menu s'affiche dans la même `JFrame` que le rendu du jeu.
- Pendant le menu, le jeu n'est pas créé et ne tourne pas.