

## i) Composants visuels

Le jeu de navigation aura :

1. Au minimum une interaction drag&drop : **version 1 : items dans un coffre qu'on peut drag and drop dans l'inventaire.** (Priorité : 1)
2. Utilisation des images pour les personnages : **verifié** ! (Priorité : 1), mais pas seulement (plateau, ressources, pièges, etc.)
3. Une manière de créer de nouveaux personnages (contrôlés par l'utilisateur ou automatisés) : **version 1 : page de personnalisation joueur en pixel art**
  - a. création aléatoire (Priorité : 1)
  - b. fenêtre pour la création graphique faite par l'utilisateur : **verifié** (Priorité : 2)
4. Un moyen de créer un plateau du jeu et ses propriétés
  - a. création aléatoire (Priorité : 1) : **version 1 : utilisation de Tiles pour créer un plateau**
  - b. fenêtre pour la création graphique faite par l'utilisateur (Priorité : 2)
5. Une fenêtre pour jouer au jeu de navigation (Priorité : 1)
6. Un moyen pour l'utilisateur d'interagir/contrôler son personnage, par exemple, le déplacer, ajuster sa direction, ... (Priorité : 1). Pensez à des interactions avancées (comme dessiner des chemins, faire glisser, traverser) et pas seulement à cliquer (Priorité : 2). **version 1 : navigation type RPG avec les touches du clavier**
7. Fournissez des retours clairs à l'utilisateur sur l'état de leur personnage (en mouvement, **version 1 : barre de vie s'affichant en haut à gauche qui évolue en fonction de la vie du perso** en combat, blessé, etc.), ainsi que sur l'état du plateau du jeu (Priorité : 1)

## ii) Composants applicatifs

Le jeu de navigation aura :

8. Une représentation interne du plateau du jeu (Priorité : 1). La représentation doit avoir des conditions de jeu qui ne consistent pas seulement à traverser le plateau du jeu pour gagner. Par exemple, collecter des éléments dans un ordre spécifique, remporter des combats, ajouter des événements aléatoires, brouillard de guerre ("fog of war")  
**version 1 : le but du jeu est de battre un certain Boss présent dans une certaine partie de la map**
9. Créer un plateau du jeu aléatoire qui peut être résolu (Priorité : 1). On note ici qu'un jeu type "labyrinthe" peut devenir très intéressant (vu complexe) au niveau algorithmique à générer aléatoirement, de façon qu'on peut le résoudre. Si vous visez un jeu qui a un

algorithme complexe, ce serait bien d'y discuter avec votre enseignant. On vous encourage à commencer avec une version simplifiée avec un remplaçant "fait main" au

début ou un algo très simplifié pour tester le reste des fonctionnalités.

10. Traiter les entrées de l'utilisateur pour déplacer le personnage (Priorité : 1)

11. Déterminer les conditions de victoire ou de défaite du jeu (Priorité : 1)

**version 1 : condition de victoire c'est battre le boss , condition de défaite : si la vie tombe à 0 ( declenche une killzone)**

12. Sauvegarder la disposition du plateau du jeu (Priorité : 2)

13. Sauvegarder le jeu (plateau, positions, ...) pour continuer le jeu plus tard (Priorité : 2)

14. Compter le temps et le nombre d'actions (Priorité : 3 sauf si important pour votre jeu)

15. Sauvegarder les personnages pour les réutiliser (Priorité : 3)