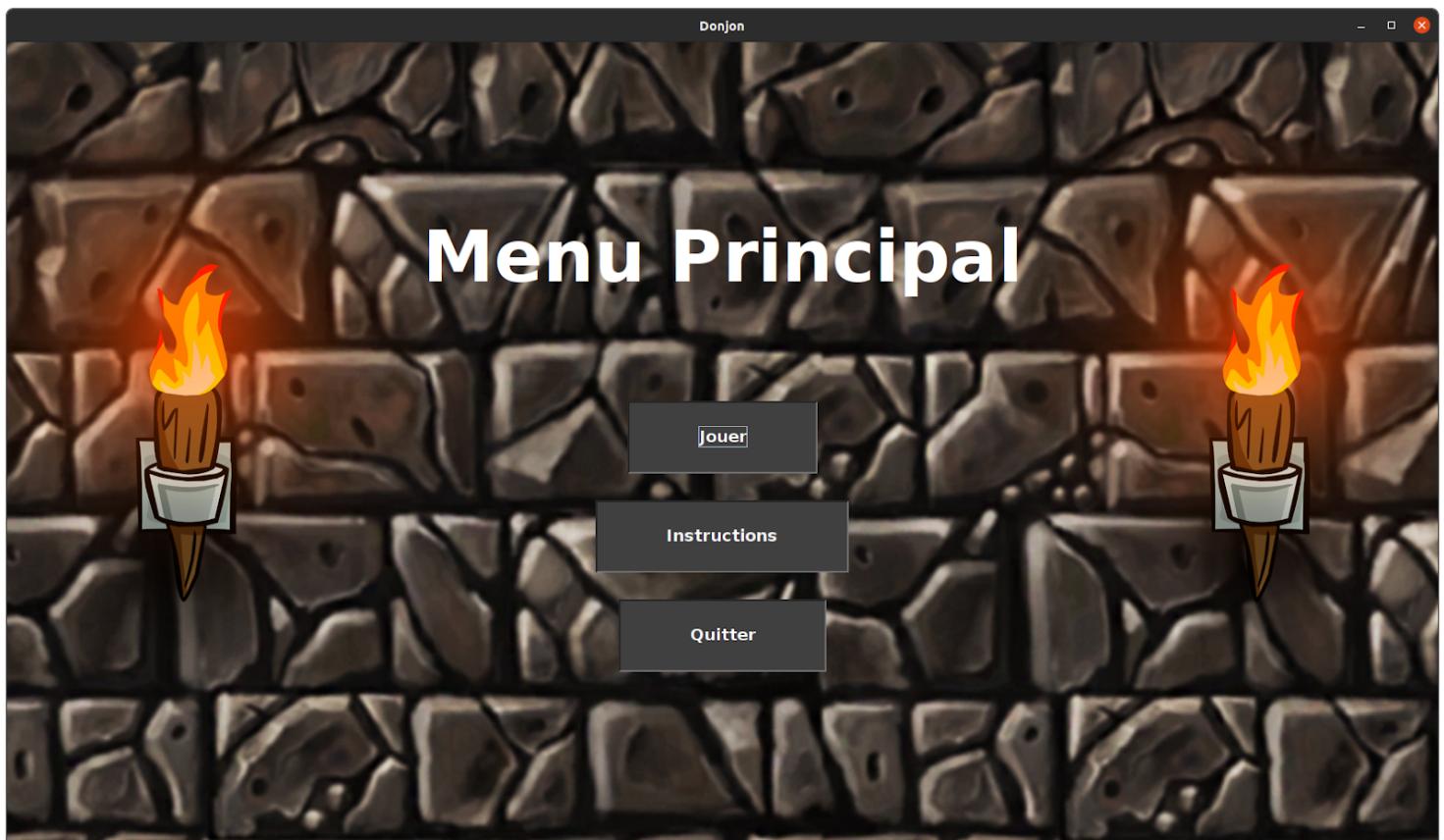


RAPPORT PROJET IHM :

PROJET : DONJON

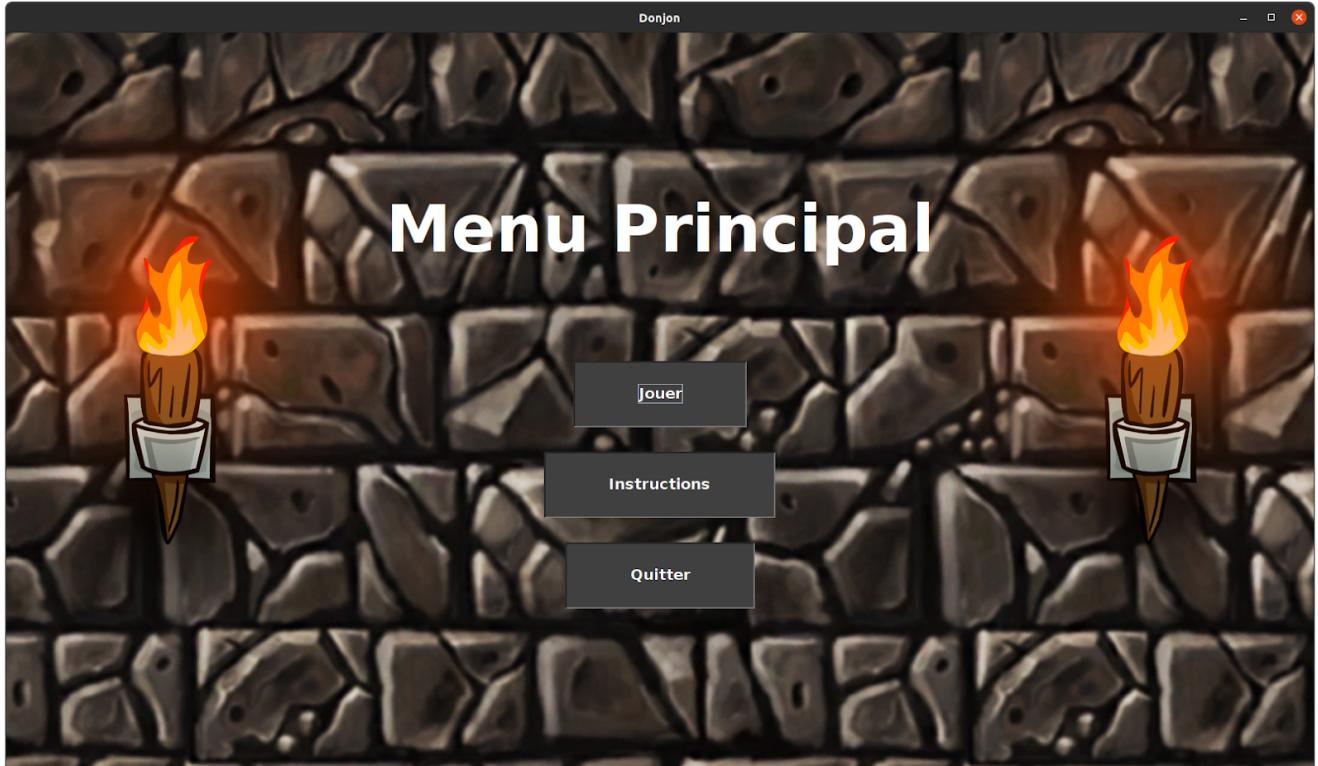


Par : Antoni BLANCHE
Marie PELLIER

SOMMAIRE

Introduction.....	2
Présentation.....	
1)Travail préparatoire.....	3
2)Fonctionnalités.....	
Fonctionnement.....	
Conclusion.....	

INTRODUCTION



Ce projet avait pour but de nous faire travailler à partir d'une API en java donnée par notre professeur.

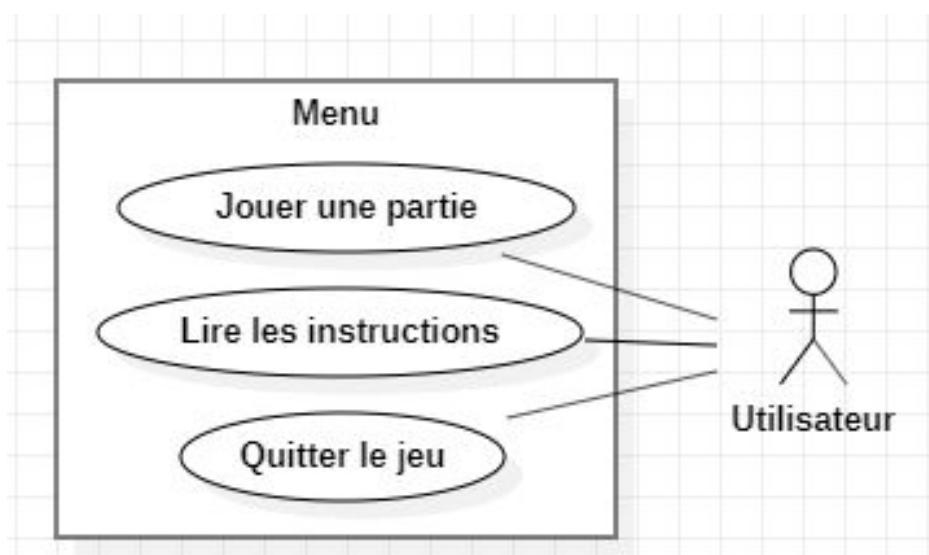
Il s'agissait de réaliser un jeu en fausse 3D, qui met en scène un donjon avec des salles. Le joueur peut tourner dans sa salle actuelle et se déplacer de salle en salle en passant par des portes (ouvertes/fermées/fermées à clef). Il peut aussi consulter une carte dont seules les salles déjà visitées sont visibles. Il possède un inventaire où sont rangés des objets mis à disposition dans les salles. Ces objets peuvent être (re)posés au sol, utilisés (sur une porte par exemple) ou être détruits.

Une version persistante du projet était aussi demandée afin de permettre au joueur de sauvegarder et charger sa partie plus tard.

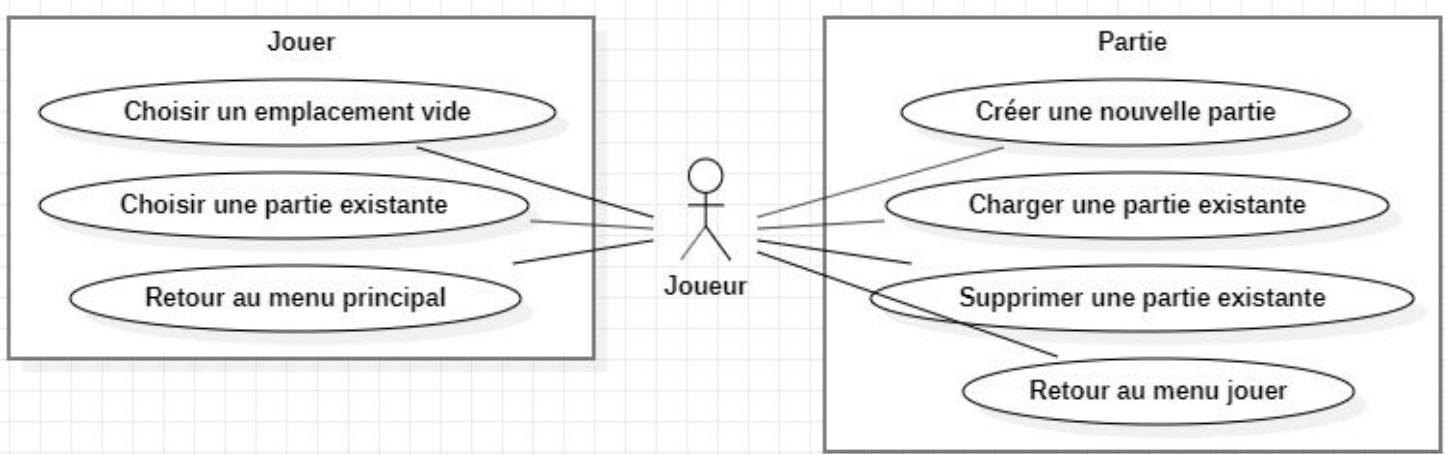
PRÉSENTATION

1) Travail préparatoire

Afin de mener à bien ce projet, la réalisation de diagrammes préparatoires était vivement recommandée. Ceux-ci permettant de visualiser la structure globale du programme. Cependant, nous avons réalisé des diagrammes uniquement pour les menus et la base de données, cette dernière n'étant même pas opérationnelle.



Ce premier diagramme d'usage permet de voir ce que le menu va proposer : permettre de jouer une partie, lire les instructions ou quitter le jeu.

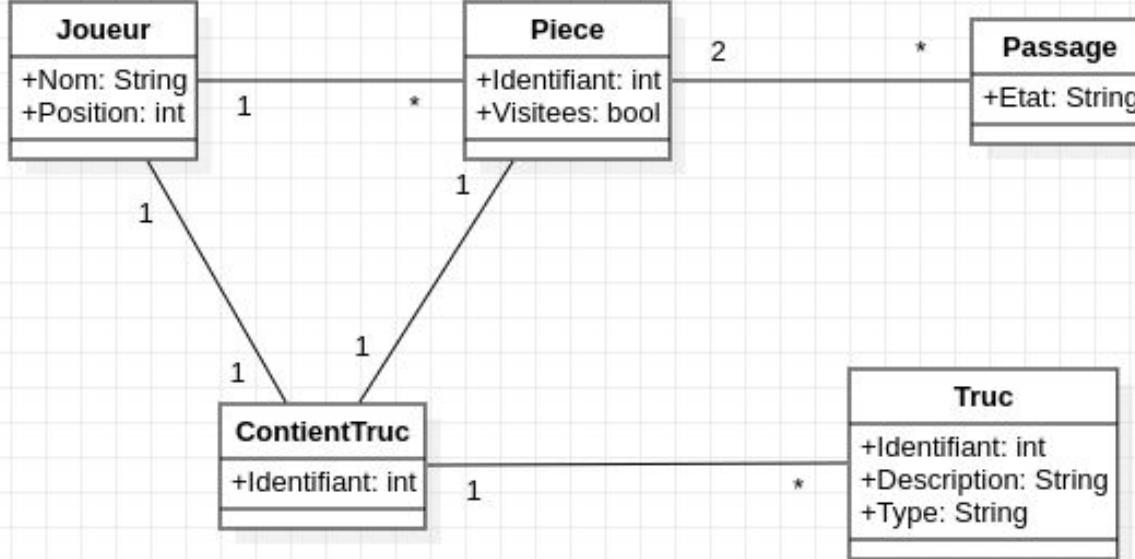


Si le joueur décide de lancer une partie, alors trois possibilités se proposent :

- soit il n'y a pas de partie existante alors il en crée une nouvelle en choisissant un emplacement vide.
- S'il y a une partie existante, alors il peut la charger en choisissant la partie.
- si il y a une partie existante, il peut la supprimer

Il peut aussi retourner au menu principal.

Ces fonctionnalités sont liées au modèle persistant.



Pour le modèle persistant, une base de données est mise en place afin de sauvegarder les données. Chaque partie générée possède plusieurs pièces qui peuvent contenir un Truc et au moins un passage vers une autre pièce. Le joueur attribue un nom à la partie, il se positionne dans une pièce et peut posséder des truc.

D'autres diagrammes sont disponibles dans le répertoire 'diagrammes/' du projet.

2) Fonctionnalités

A. Lire les instructions



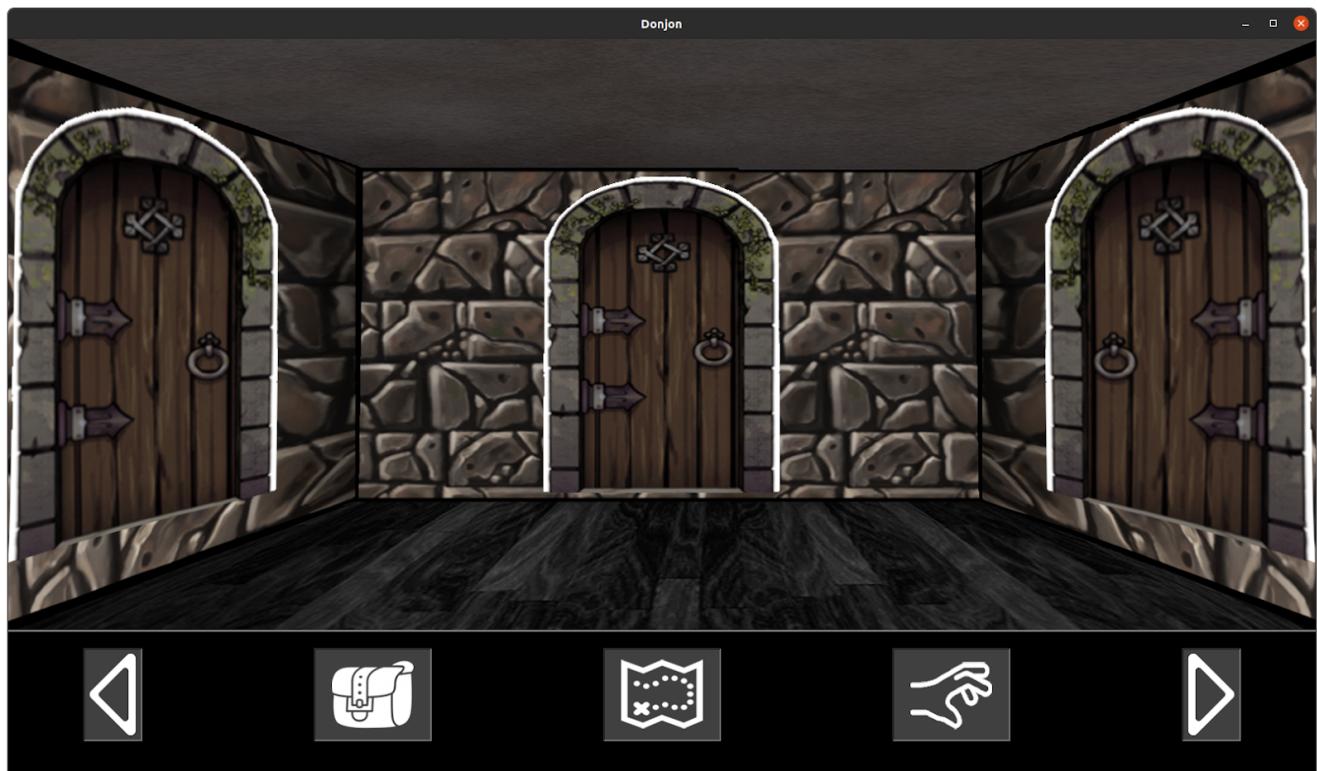
Depuis le menu principal, vous pourrez lire les instructions qui vous guiderons sur la manière de jouer au jeu. Chaque bouton vous permet de faire une action ou d'accéder à une autre partie du jeu. Il existe aussi un bouton avec une croix qui sert généralement à fermer l'onglet pour revenir au jeu. Enfin, pour se déplacer dans le donjon, il suffit de cliquer sur la porte qui se trouve droit devant.

B. Créer une partie



Depuis le menu, en sélectionnant un emplacement vide de partie (ce qui n'est pas difficile car les parties ne peuvent pas être sauvegardées), vous serez invité à entrer votre nom de joueur. Ce nom vous permettra de reprendre votre partie par la suite. Une fois que votre nom est correct, c'est-à-dire qu'il se compose d'au moins un caractère (et qu'il n'est pas déjà présent dans la base de données), votre partie commence.

C. Déplacements



Une fois la salle chargée, vous vous trouvez face au mur nord. Vous pouvez alors vous tourner face à un autre mur grâce aux flèches. Vous pouvez aussi passer dans une autre salle si la porte en face de vous est ouverte. Sinon, elle est soit fermée soit fermée à clef. Il vous faudra alors l'ouvrir soit en utilisant une clef soit en cliquant dessus. A chaque fois que vous quittez une salle, votre position est modifiée et enregistrée.

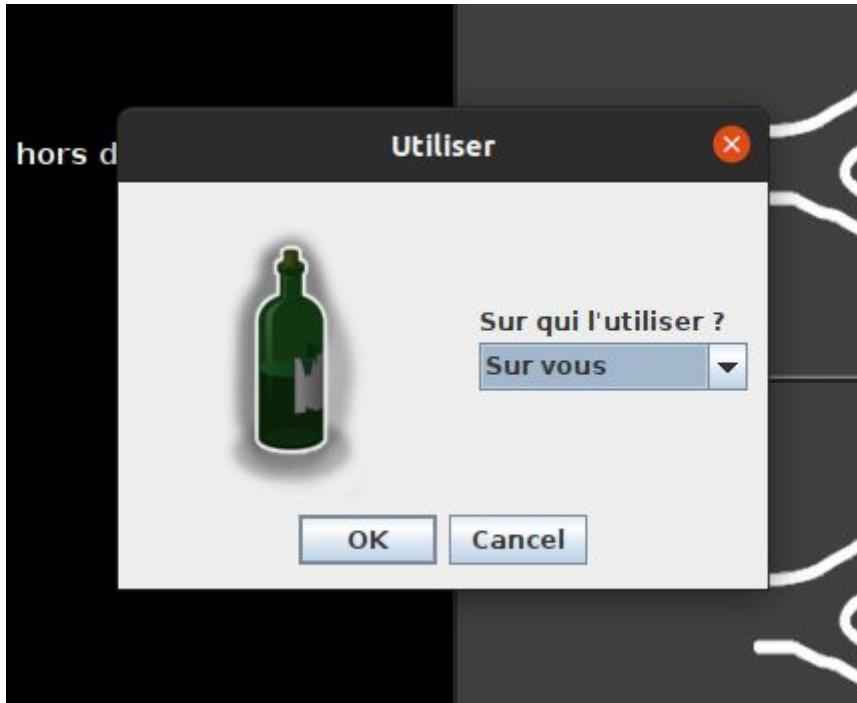
D. Gestion de l'inventaire



Le bouton le plus à droite sert à ramasser tous les trucs se trouvant dans la pièce. Ils sont automatiquement ajoutés à votre inventaire.

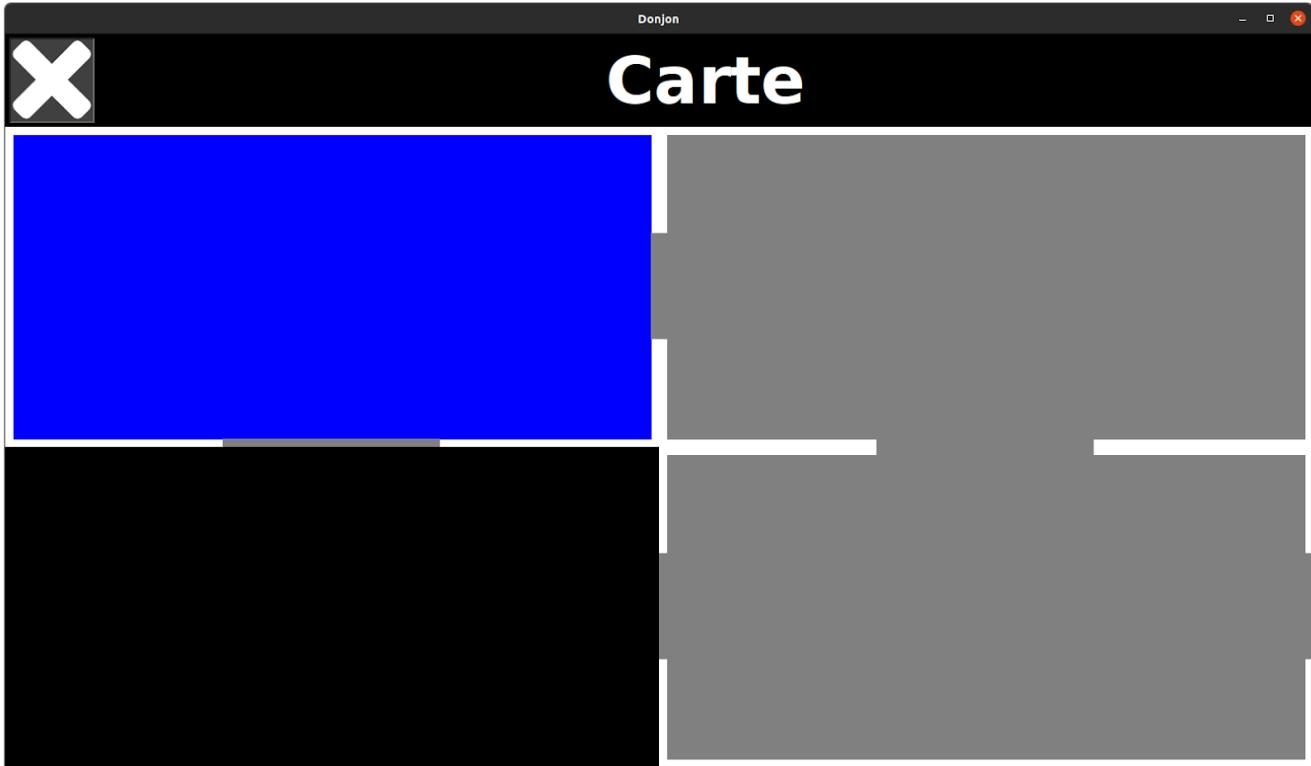


C'est depuis l'inventaire que vous pouvez utiliser ou déposer un truc. Si vous choisissez de déposer un truc, il se retrouve dans la salle. Vous pourrez alors le reprendre plus tard. Sinon, vous pouvez aussi décider de l'utiliser.



Dans ce cas, deux choix s'offrent à vous. Soit vous l'utilisez sur vous, à vos risques et périls ; soit sur le passage. Si le truc n'a aucun effet, alors une fenêtre vous en avertira (j'espère que vous apprécierez la référence).

E.La carte



Afin de vous aider (mais pas trop) dans votre ballade, la carte ne montre que les pièces ayant déjà été visitées. Elle s'actualise à chaque fois que vous quittez une pièce. La pièce dans laquelle vous vous trouvez s'affiche en bleu. Ainsi, vous ne risquez pas de vous perdre ! Un problème survient avec la carte lors de l'utilisation de modèles persistants : votre position ne s'affiche pas en bleu. De plus, nous avons été obligés d'adapter une partie du code pour que la carte continue de s'afficher.

F.Problèmes connus

Certaines fonctionnalités de notre projet ne sont pas encore opérationnelles à 100%. Premièrement, l'intégration des modèles persistants a considérablement augmenté le temps de chargement lors des transitions. Pour les cas les plus marqués de latence, nous avons décidé d'ouvrir une fenêtre qui indique que le chargement est en cours. Cependant, cette fenêtre n'affiche pas de texte, et nous ne savons pas pourquoi. Les usines ont aussi été retravaillées pour être plus rapide lors de la création des objets.

CONCLUSIONS

Marie : Ce projet était plutôt intéressant pour moi. Je me suis occupée principalement des vues du programme et le fait de se baser sur une API est quelque chose qui facilite la création du programme pour moi. Mais sa compréhension est autre chose, surtout quand on n'a pas l'habitude d'en aborder. On aurait dû mieux gérer notre temps, en faisant par exemple un emploi du temps détaillé afin de répartir les tâches convenablement.

Antoni : Le fait de devoir intégrer une API conçue par une personne extérieur à été une grande première pour moi. Bien que j'étais réticent au départ, j'ai pu constater le temps gagné grâce à celle-ci. Cependant, elle a commencé à vraiment poser problème lorsqu'il a fallu l'adapter à un modèle persistant. Son adaptation a été compliquée car certaines parties qui fonctionnaient très bien avec les modèles non persistants ont dû être revues. Je suis quand même satisfait du résultat, et d'avoir pu mettre en pratique la conception avec les diagrammes vus en cours d'ACDA au moins pour les menus.