



Projet Infographie I

wolf3d

42 staff staff@42.fr

Résumé: Ce projet consiste à implémenter un jeu utilisant la technique du raycasting.

Table des matières

I	Préambule	2
II	Sujet - Partie obligatoire	3
III	Sujet - Partie bonus	4
IV	Consignes	6
V	Notation	8

Chapitre I

Préambule

Voici ce qu'Internet a à dire au sujet du raycasting

- [http ://fr.wikipedia.org/wiki/Raycasting](http://fr.wikipedia.org/wiki/Raycasting)
- [http ://projet-moteur-3d.e-monsite.com/pages/raycasting/raycasting.html](http://projet-moteur-3d.e-monsite.com/pages/raycasting/raycasting.html)
- [http ://zupi.free.fr/PTuto/index.php?ch=ptuto&p=ray](http://zupi.free.fr/PTuto/index.php?ch=ptuto&p=ray)
- [https ://developer.mozilla.org/samples/raycaster/RayCaster.html](https://developer.mozilla.org/samples/raycaster/RayCaster.html)

Chapitre II

Sujet - Partie obligatoire

Ce miniproj consiste à créer graphiquement la représentation réaliste que l'on pourrait avoir à l'intérieur d'un labyrinthe. La représentation est réalisée en utilisant le principe de Ray-Casting.

- Vous avez le choix de la taille et de la forme du labyrinthe.
- Vous devrez gérer l'expose correctement.
- Il faudra appuyer sur la touche ESC pour quitter le programme.
- Les flèches du clavier permettent de se déplacer en temps réel dans le labyrinthe,
- La couleur des murs varie suivant l'orientation (nord, sud, est, ouest).



Chapitre III

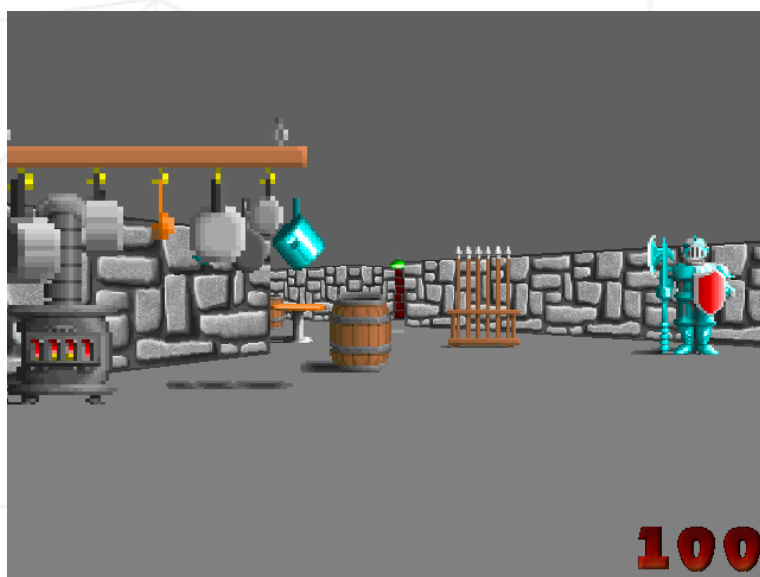
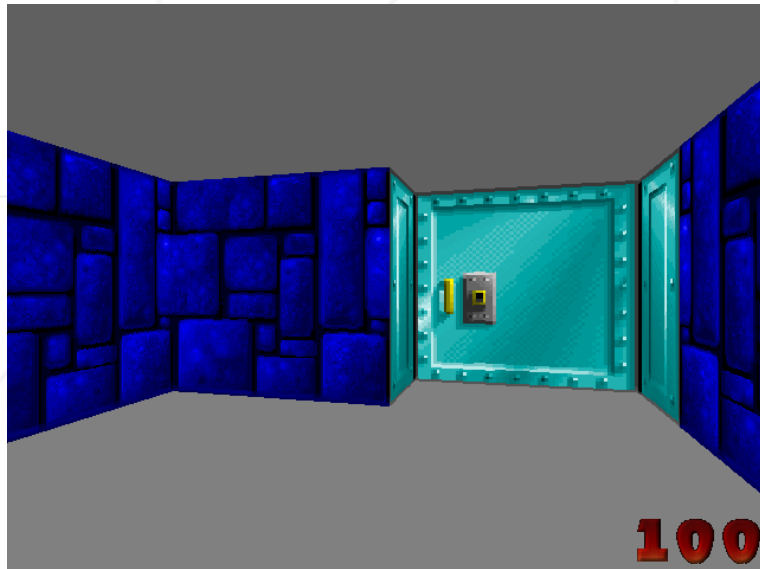
Sujet - Partie bonus



Les bonus ne seront évalués que si votre partie obligatoire est PARFAITE. Par PARFAITE, on entend bien évidemment qu'elle est entièrement réalisée, et qu'il n'est pas possible de mettre son comportement en défaut, même en cas d'erreur aussi vicieuse soit-elle, de mauvaise utilisation, etc ... Concrètement, cela signifie que si votre partie obligatoire n'obtient pas TOUS les points à la notation, vos bonus seront intégralement IGNORÉS.

En option sur plein de points :

- On ne rentre pas dans les murs
- Des textures sur les murs
- Une texture de ciel
- Une texture de sol et/ou de plafond
- Des objets dans le labyrinthe
- On ne rentre pas dans certains objets
- Des objets que l'on ramasse pour faire des points
- Des portes qui s'ouvrent et se ferment
- Des mechants à combattre
- Des passages secrets
- Plein de niveaux



Chapitre IV

Consignes

- Ce projet ne sera corrigé que par des humains. Vous êtes donc libres d'organiser et nommer vos fichiers comme vous le désirez, en respectant néanmoins les contraintes listées ici.
- L'exécutable doit s'appeler `wolf3d`.
- Vous devez rendre un Makefile.
- Votre Makefile devra compiler le projet, et doit contenir les règles habituelles. Il ne doit recompiler le programme qu'en cas de nécessité.
- Si vous êtes malin et que vous utilisez votre bibliothèque `libft` pour votre `wolf3d`, vous devez en copier les sources et le `Makefile` associé dans un dossier nommé `libft` qui devra être à la racine de votre dépôt de rendu. Votre `Makefile` devra compiler la librairie, en appelant son `Makefile`, puis compiler votre projet.
- Vous ne devez pas avoir de variables globales.
- Votre projet doit être à la Norme.
- Vous devez gérer les erreurs de façon raisonnée. En aucun cas votre programme ne doit quitter de façon inattendue (Segmentation fault, etc...).
- Vous devez rendre, à la racine de votre dépôt de rendu, un fichier `auteur` contenant votre login suivi d'un '\n' :

```
$>cat -e auteur
xlogin$
$>
```

- Vous pouvez utiliser la minilibx pour l’affichage, mais si vous le souhaitez vous pouvez utiliser d’autres librairies d’affichage (X11, SDL, etc...). Si la librairie que vous utilisez n’est pas installée sur les machines de cluster, vous devrez fournir la librairie dans votre rendu, et elle devra se compiler automatiquement, sans autre action que de compiler votre rendu. Quelque soit la librairie d’affichage, vous ne devrez en utiliser que les fonctions de dessin basiques similaires à la minilibx.
- Dans le cadre de votre partie obligatoire, vous avez le droit d’utiliser les fonctions suivantes :
 - open
 - read
 - write
 - close
 - malloc
 - free
 - perror
 - strerror
 - exit
 - toutes les fonctions de la lib math (-lm et man 3 math)
 - ainsi que toutes les fonctions de la minilibx ou leur équivalent dans une autre librairie d’affichage :
 - mlx_new_window
 - mlx_pixel_put
 - mlx_new_image
 - mlx_loop
- Vous avez l’autorisation d’utiliser d’autres fonctions dans le cadre de vos bonus, à condition que leur utilisation soit dûment justifiée lors de votre correction. Soyez malins.
- Vous pouvez poser vos questions sur le forum, sur jabber, IRC, ...

Chapitre V

Notation

- La notation de `wolf3d` s'effectue en deux temps :
 - En premier lieu, votre partie obligatoire sera testée. Elle sera notée sur 20 points.
 - Ensuite, vos bonus seront évalués. Ils seront notés sur 10 points.
 - Ils ne seront évalués que si votre partie obligatoire est PARFAITE (Tout doit fonctionner comme attendu, et la gestion d'erreur doit être sans faille).
 - Vous obtiendrez entre 1 et plusieurs points par fonctionnalité bonus distincte et correctement réalisée (À la discrétion de vos correcteurs)
 - Également, l'optimisation de la qualité de certains éléments de votre code seront évalués et pourront donner lieu à des points supplémentaires dans cette partie bonus.
- Bon courage à tous !