



TEMPS REELS



Projet de fin de semestre d'Algorithmique des Arbres

LY Steven | HAELLING Lucas

LY Steven

2020/2021

Licence 2 Semestre 4

UPEM - UGE

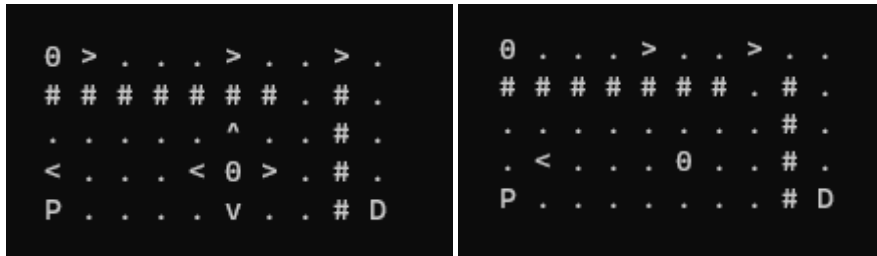
Sommaire

Introduction	3
Utilisation	3
Format de fichier d'un niveau personnalisé	4
Objets disponibles	4
Améliorations implémentés.....	4
Amélioration possibles.....	4
Répartition du travail	5
Problèmes rencontré	5
Problème lié a la situation	5
Problème lié au programme	5
Compétences Travaillés	5
Conclusion.....	6
Annexe	6
Horaires de travail.....	6
Sprites	7
Projectiles.....	7
Titre	7
Partie en mode graphique	8
Ecran de victoire	8
Ecran de défaite	8

Introduction

Le jeu qui va vous être présenté est un jeu de parcours ou le but du joueur sera de traverser une zone pour arriver à destination, mais arriver à destination ne sera pas une tâche facile, il vous faudra éviter des projectiles posés stratégiquement pour arriver à bonne destination. Vous devez aussi savoir qu'un lanceur ne lancera que des projectiles à intervalle régulier dans les quatre points cardinaux.

Dans ce jeu il vous sera possible d'importer un niveau déjà créé depuis un fichier texte ou bien jouer avec un niveau déjà prédéfini dans le jeu.



P représente le personnage

0 représente un lanceur de projectile

^ v < > représentent les
Projectiles

D représente la destination

représente un mur

. Représentent du vide

Utilisation

Pour pouvoir lancer le projet il vous faudra, vous placer dans le répertoire du makefile puis dans le terminal taper la commande make.

Par la suite nous vous invitons à entrer :

- `./tempsreels --test=<int>` Lance le niveau<int> en MLV si possible
 - `./tempsreels --test=<int> -a` Lance le niveau<int> en ascii sur le terminal
 - `./tempsreels --fichier=<fichier>` Lance le niveau <fichier> en MLV si possible
 - `./tempsreels --fichier=<fichier> -a` Lance le niveau <fichier> en ascii sur le terminal
 - `./tempsreels --usage` Explique comment lancer un niveau
-
- ❖ Pour lancer un niveau personnalisé il faudra que l'utilisateur place son niveau dans le dossier « niveau_joueur ».
 - ❖ Lorsque vous lancez `./tempsreels --fichier=<fichier>`, il suffira de mettre le nom du fichier et non le chemin.

Format de fichier d'un niveau personnalisé

Pour jouer à un niveau que vous aurez créé, il faudra respecter quelques conventions.

Objets disponibles

Objets en Ascii	Représentation
0 (zéro)	Lanceur
D	Destination
#	Mur
.	Vide
P	Personnage

- Chaque ligne doit avoir le même nombre de caractères.
 - Si une ligne a plus de caractères qu'une autre une erreur sera renvoyé (aka Segmentation Fault).
- Les caractères sont mis sans espaces.
- Un niveau sans destination est interminable.
- Il est important de mettre votre niveau dans le dossier niveau_joueur pour pouvoir jouer à votre niveau.
- Il est aussi important de Créer le personnage dans une zone éloigné des lanceurs.
- Il est possible de poser plusieurs destinations, plusieurs lanceurs, plusieurs murs.
- Il n'est néanmoins pas possible de créer plus d'un personnage.
- Pour des raisons de performances il faudra un minimum de un lanceur.

Si Vous voulez avoir la possibilité de jouer un niveau en affichage graphique, le niveau devra précisément ne pas avoir plus de 12 colonnes et 10 lignes. Si cela n'est pas le cas le jeu vous renverra en Ascii.

Améliorations implémentés

Nous avons implémenté comme améliorations, le déplacement du personnage, comme demandé dans le sujet. De plus nous avons aussi ajouté la possibilité de jouer avec un affichage graphique avec la librairie MLV.

- On a aussi utilisé des sprites pour pouvoir jouer dans l'interface graphique
- Comme autre Améliorations non dis dans le sujet, il a été décidé d'ajouter de la musique et des sons pour savoir si le joueur aura réussi ou non le niveau
- Aussi lorsque le joueur fini une partie un écran de victoire ou de défaite s'affiche selon si le joueur a gagné ou non avec une musique de victoire ou défaite. Il lui sera aussi communiqué le temps de complétion du niveau ou bien son temps de survie
- En graphique nous avons aussi décidé d'ajouter un timer pendant le jeu.
- Le personnage se déplace normalement dans le mode graphique, il n'est pas ralenti

Amélioration possibles

- Outil de création de niveau depuis l'interface du jeu
- Lors du jeu le personnage doit récupérer une clef ou un objet pour pouvoir aller à la destination
- Enigmes avec des endroits ou aller pour ouvrir la sortie
- D'autres types de projectile (a tête chercheuses)
- Des trappes cachées.
- Gestion tampon des touches.
- Enigme permettant de déplacer plusieurs personnages dans un niveau

Répartition du travail

Il n'y a pas vraiment eu de répartitions du travail, on a majoritairement travaillé ensemble. Nous avons utilisé un logiciel de partage de code pour travailler en même temps, chaque fonction ont été codé par l'un puis de nous puis complété ensemble.

Parmi ce qu'on a travaillé le plus séparément il y a eu :

- Lucas gestion de fichier (création d'un niveau à partir d'un fichier texte) et gestion des arguments
- Steven partie graphique

Problèmes rencontré

Problème lié a la situation

Il nous a été compliqué de réaliser le projet au vu de la crise sanitaire actuelle, la tenue des cours à été un peu lente ce qui nous a freiné sur le projet de plus le fait de programmer depuis chez soi avec quelqu'un sans pouvoir bien expliquer ce que l'on veut faire est compliqué.

Nos horaires aussi n'était pas si souvent correspondant du coup nous ne pouvions pas travailler a notre guise.

De plus, avec les examens finaux qui arrivaient à grand pas, nous préférons plus nous concentrer sur nos examens donc le projet est passé en second plan.

Problème lié au programme

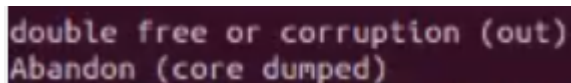
Le jeu a pour point d'origine en haut à gauche.

File de priorité de tas : Nous avons fortement bloqué sur les files de priorités dans le jeu, ne serait ce qu'au début il nous a été difficile de comprendre le sujet avec les tas données en exemple, par la suite nous avons pu nous débloquent.

Graphique pour le graphique nous avons un problème par rapport aux tailles de niveau, la version graphique ne supporte pas les niveaux de taille au dessus de (12,10) donc lorsque l'on n'est pas dans ces conditions le jeu se lancera en ascii.

Notre fonction `free_niveau()` ; crée une erreur que nous avons pas su régler.

Donc on a préféré ne pas libérer l'espace du niveau car nous préférons ne pas avoir d'erreurs (appel a la fonction mis en commentaires)



```
double free or corruption (out)
Abandon (core dumped)
```

Compétences Travaillés

Ce projet nous à permis :

- D'utiliser les Tas (Arbres Tournois Parfaits)
- Librairie MLV
- Gestion du temps
- Gestion de fichier
- Types Void*

Conclusion

Ce projet nous a été fortement bénéfique, notamment car ce projet était un projet en groupe. Il nous aura permis de réviser l'utilisation des tas et de la librairie MLV. La longueur et la complexité du projet étaient plutôt raisonnable (comparé à accoutumé). Le sujet était plutôt intéressant, il nous a donné des idées pour créer d'autres manières de jouer. Le projet nous aura aussi permis de comprendre comment travailler en équipe et à distance.

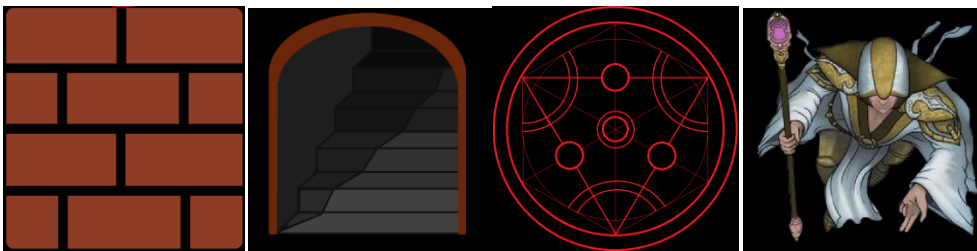
Annexe

Horaires de travail

Date	Travail
08-Avril	➤ malloc_Niveau
11-Avril	➤ free_niveau ➤ niveau0
12-Avril	➤ malloc_tas ➤ free_tas
14-Avril	➤ un_evenement_est_pret ➤ ote_minimum
18-Avril	➤ ote_minimum ➤ ajoute_evenement ➤ construit_tas
20-Avril	➤ ote_minimum ➤ ajoute_evenement ➤ affiche_tas_ev ➤ construit_Tas ➤ Affiche_Tas ➤ traite_evenement
22-Avril	➤ traite_lanceur ➤ traite_evenement ➤ gestion_donnee_suppl ➤ traite_projectile
23-Avril	➤ traite_lanceur ➤ traite_projectile ➤ makefile
26-Avril	➤ traite_evenement ➤ traite_lanceur ➤ gestion_ev ➤ traite_projectile
01-Mai	➤ debuggage
07-Mai	➤ affiche_lanceur ➤ main
09-Mai	➤ MLV ➤ modularisation
10-Mai	➤ modularisation ➤ makefile ➤ traite_personnage
13-Mai	➤ graphique ➤ traite_personnage ➤ vérification si on peut bouger le personnage

15-Mai	<ul style="list-style-type: none"> ➤ graphique implémentation des sprites ➤ création une présentation des règles et objets dans l’affichage graphique ➤ Lecture de fichier pour permettre de jouer avec un niveau définit dans un fichier
16-Mai	<ul style="list-style-type: none"> ➤ Graphique ➤ Traite_personnage ➤ Verif_partie ➤ Affiche_victoire ➤ Son dans la partie graphique ➤ Gestion de fichier
17-Mai	<ul style="list-style-type: none"> ➤ Gestion des arguments ➤ Graphique
18-Mai	<ul style="list-style-type: none"> ➤ Graphique ➤ Gestion d’argument ➤ Gestion de fichier
19-Mai	<ul style="list-style-type: none"> ➤ Graphique ➤ Gestion de fichier ➤ Gestion d’arguments
21-Mai	<ul style="list-style-type: none"> ➤ Gestion des arguments (fin) ➤ Réglages des petits bugs rencontrés ➤ Optimisation du code

Sprites



Mur

Destination

Lanceur

Personnage

Projectiles



Projectile Haut

Projectile Droite

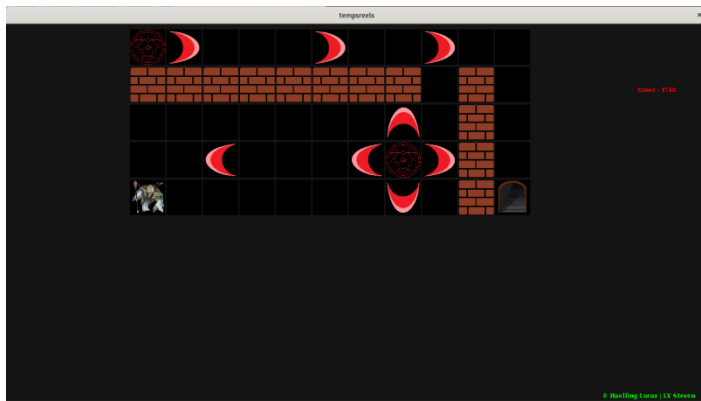
Projectile Bas

Projectile Gauche

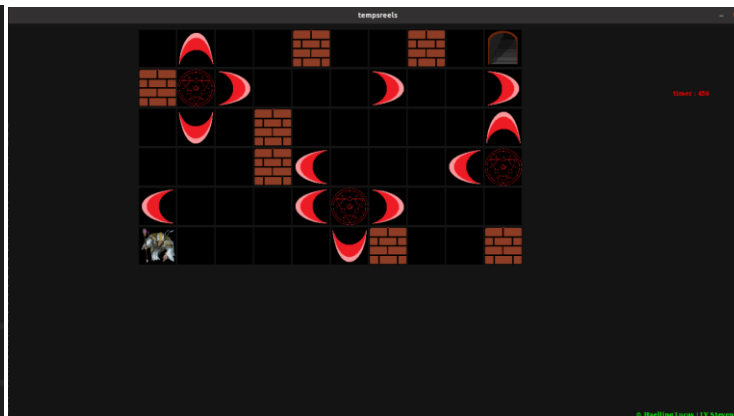
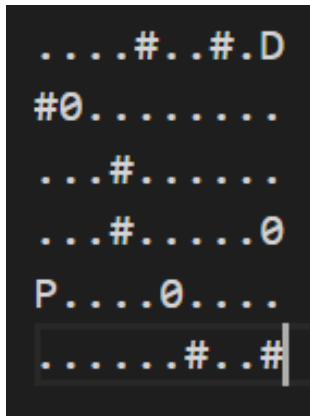
Titre



Partie en mode graphique

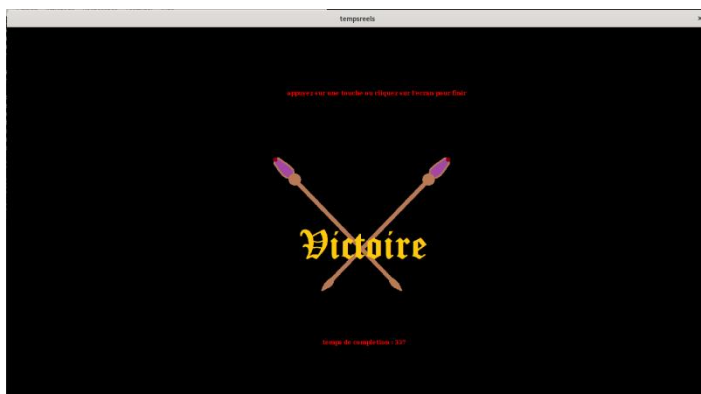


Niveau0



niveau personnalisé

Ecran de victoire



Ecran de défaite

