

Rapport : Jeu en temps réel

Salma TOUAMI- Dylan DE JESUS MILITAR

0..... #####.##0.# P.....#D à t = 0ms (état initial)	0>..... #####.#^.#<0>.# P....v...#D à t = 1000ms	0.>..... #####.## ...<.0># P.....#D à t = 1300ms	0.>..... #####.## ..<.0.# P.....#D à t = 1600ms	0...>..... #####.## .<...0.# P.....#D à t = 1900ms
0>.>..... #####.#^.# .<...<0>.# P....v...#D à t = 2000ms	0>...>.... #####.#^.# <...<0>.# P....v...#D à t = 2200ms	0.>.>..... #####.## <...<.0># P.....#D à t = 2300ms	0.>...>.... #####.## ...<.0># P.....#D à t = 2500ms	0...>...>... #####.## .<...0.# P.....#D à t = 2600ms
0...>...>.. #####.## ..<.0.# P.....#D à t = 2800ms	0...>...>.. #####.## .<...0.# P.....#D à t = 2900ms	0>...>...>.. #####.#^.# .<...<0># P....v...#D à t = 3000ms etc.	

Introduction	2
Fonctionnement	2
Compilation	2
Ligne de commande	2
Le jeu	2
Implémentation	2
Améliorations	2
A - Déplacement	2
B - Option WallBreaker	2
C - Gestion de fichier	2
Difficultés rencontrés	3
Conclusion	3
Annexe	3

1. Introduction

L'objectif du projet était la conception d'un jeu en temps réel mettant en scène des lanceurs qui lancent des projectiles vers le personnage P, ce dernier a pour seul but d'arriver à la destination D mais pour compliquer un peu la tâche, on ajoute les murs # comme une barrière qu'il ne faut pas franchir .

De plus ce sujet comporte plusieurs parties qu'on peut le diviser en plusieurs étapes

Étape 1 :

- consiste à implémenter la gestion des niveaux

Étape 2 :

- consiste à implémenter la gestion et le traitement des événements

Étape 3 :

- consiste à permettre à l'utilisateur de déplacer son personnage dans le niveau .

Étape 4 :

- consiste à implémenter une ou plusieurs autres améliorations (exemple gérer le temps)

2. Fonctionnement

Le fonctionnement du programme se découpe en trois parties différentes, la compilation, la ligne de commande, et ce qui est en rapport même avec le fonctionnement du jeu.

1. Compilation

La compilation du projet se fait au moyen d'un **makefile** créé.

On retrouve dans ce projet les modules suivants (créé en plus des autres fournis):

- Niveau : Permettant de traiter les maps des niveaux, ainsi que leur création en général (créé les niveaux 1&2)
- Gestion_Fichier : Est un module qui sert l'affichage en sortie standard (fichier stdout) ainsi que l'amélioration
- Tas : Permettant la manipulation des différentes fonctions servant à la structure du tas
- Graphique : Le module était initialement prévu pour une partie graphique (amélioration (1)) cependant elle sert à gérer les événements lu au clavier et le déplacement (amélioration (0))

La commande **make clean** permet la suppression de fichier exécutables.

2. Ligne de commande

La ligne de commande est très simple:

- ./tempsreel --test=<int>
- ./tempsreel --test=<int> -WallBreaker-
- ./tempsreel --fichier=<filename>
- ./tempsreel --fichier=<filename> -WallBreaker-
- ./tempsreel --usage

Il existe donc 3 modes:

- > Niveaux créé par le programme
- > Lecture des niveaux dans les fichiers
- > Rappel de l'usage du jeu

Pour un des deux premiers modes on peut y intégrer un troisième argument (sur la ligne de commande) **-WallBreaker-**

Cette option permet le repoussage des murs.

3. Le jeu

Le programme se lance sur un message dans le terminal, si l'utilisateur appuie sur la touche entrée, le programme se lance en affichant la map du niveau correspondant

```

Pressez la touche Entrée pour commencer
4      0
3      ##### #
2              #
1              0 #
0      P        #D

      0123456789

```

L'utilisateur se déplace (il fait bouger le personnage P) jusqu'à atteindre la destination (D) des contraintes comme des murs (#) et des lanceurs (0) l'empêchent de se déplacer partout. (Les projectiles n'ont pas pu fonctionner)

Au moyen des touches:

- > 'z' = Haut
- > 'q' = Gauche
- > 's' = Bas

-> 'd' = Droite

Le personnage se déplace et fait face à des murs virtuels l'empêchant de sortir de la map créée.

Lorsque le personnage atteint son but, le niveau est réussi, et un message s'affiche.

Il est possible en pressant la barre espace de quitter la partie, le programme s'arrête.

```
Pressez la touche Entrée pour commencer
4      0
3      ##### #
2              #
1              0 #
0      P      #D

      0123456789

-- PARTIE QUITTÉE --
```

3. Implémentation

1. Améliorations

A - Déplacement

Le déplacement du personnage (amélioration 0) s'effectue au moyen de la fonction :

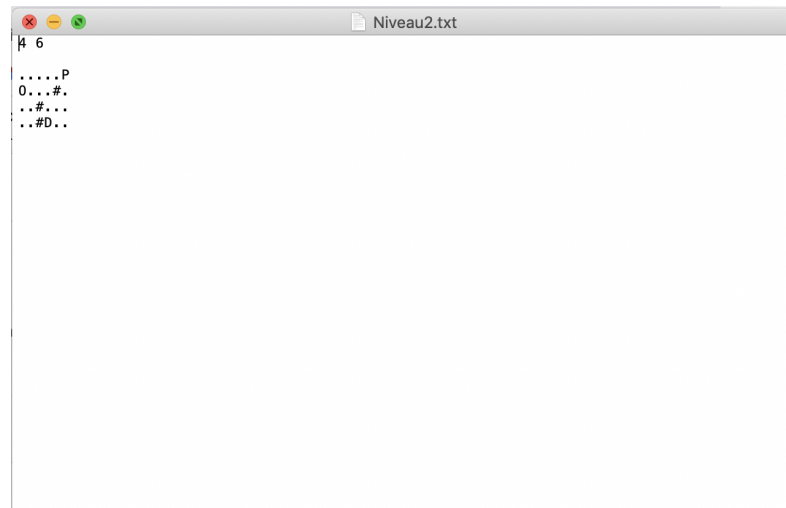
evenement();

La fonction prend en argument un caractère et permet de traiter la direction vers laquelle le personnage se dirige, si le personnage ne sort pas des limites du niveau.

B - Option WallBreaker

L'option -WallBreaker- est traitée dans la fonction vu dans le point précédent, en effet il suffit simplement de vérifier que le paramètre **Wbreaker_opt** vale 'o' et non 'n' ainsi si c'est le cas on s'autorise à déplacer le personnage sur le mur, et décaler le mur dans la direction du déplacement si, rien ne se trouve derrière. (amélioration (8))

C - Gestion de fichier



Fichier représentant le niveau 2

Cette amélioration est l'amélioration (12) celle ci représente la lecture d'un **niveau dans un fichier** en effet en lançant le programme avec l'option correspondante. Le programme lira dans le fichier les données du niveau créé dans ce fichier.

Pour ce faire, le fichier devra respecter un format bien précis:

- Les espaces sont représentés par des caractères '.'
- La première ligne du fichier correspond aux dimensions du niveau : **y x**
- Une ligne d'acrt entre dimensions et map
- Un personnage doit être présent dans la map représentant le niveau

Suite à cela le jeu se déroule normalement en utilisant les données du fichier.

4. Difficultés rencontrés:

La principale difficulté rencontrée lors de la conception de ce projet a été le contact du binôme. Il nous était impossible de pouvoir communiquer de manière satisfaisante sur la façon de travailler. Ainsi toutes les réflexions,... ont été abordées uniquement par écrit, provoquant par moments des difficultés de compréhension.

De plus, sur la fin de ce travail nous n'avions plus la possibilité d'échanger nos différents travaux, et avancées sur le projet en raison de problèmes techniques, ceci nous limitant dans notre progression.

En terme de programmation nous avons eu du mal sur:

- Les structures des projets étant un peu difficiles à comprendre puisque c'était pas par nos choix mais plutôt imposé, chose qui nous limite un peu .
- Le fait que le projet comporte plusieurs parties et fonctionnalités (gestion du temps , des niveaux,des personnages...) avec une durée de un mois .

5. Conclusion

Pour conclure,le projet très intéressant, original mais qui n'a pas été si facile , nous avons trouvé qu'il avait été très formateur pour nous apprendre à collaborer à distance tout en se répartissant les tâches , la maîtrise du temps.

Nous avons ainsi pu travailler sur un projet pouvant être utilisé par le grand public, chaque choix que pourrait faire l'utilisateur, afin de répondre à tous les cas possibles.

6. Annexe

Mis à part les constituants du programme, on retrouve un dossier annexe:

→ *Fichiers_Niveaux* composé de deux fichiers textes correspondant à deux niveaux enregistrés.

→ *Makefile* permettant la compilation du projet.