UPMC SORBONNE UNIVERSITÉS

POLYTECH SORBONNE

Rapport du projet C++ :

Paradise Paper

Encadrants : C. Braunstein, J.-B. Brejon

Auteurs : Isma Bentoumi, Sophia Hakam, Suzanne Sleiman

Module : C++

Fillière : MAIN 4

1er Semestre Année 2017/2018

Table des matières

1 Description de l'application développée

Notre projet est un jeu de plateforme dans lequel le joueur est un personnage devant cumuler un certain nombre de pièces donnant un score afin de passer au niveau suivant. Chaque niveau représente un paradis fiscal. Pour atteindre un nouveau paradis fiscal, le joueur est limité en temps : il n'a que 60 secondes. A un niveau correspond un score : 20 pour le niveau 1 ; 40 pour le niveau 2 ; 60 pour le niveau 3. La difficulté augmente donc de niveau en niveau.
En parallèle, un autre personnage, le policier, représentant l'ennemi, se déplace sur sur le fond. Si le joueur touche le policier, ce dernier perd et doit recommencer le jeu à zéro, c'est-à-dire au niveau 1.
Des murs sont également affichés sur l'écran sur lesquels le joueur peut sauter pour attraper les pièces et échapper au policier. Pour résumer, nous avons donc :

- Le personnage
- L'ennemi représenté comme un policier
- Les murs nous aidant à attraper les pièces les plus difficiles
- Les pièces nous aidant à augmenter notre score final pour passer aux autres niveaux. Celle ci s'affiche une par une sur l'écran.
- Le chronomètre qui comptabilise le temps. Au bout de 60 sec, le chronomètre est remis à 0. Le joueur doit recommencer la partie.

Pour passer au niveau suivant le joueur doit :

- Attraper le nombre de pièces nécessaires pour acquérir le score demandé.
- Pour cela, il ne doit pas dépasser le temps imposé fixé à 60 sec.

Le joueur a deux façons de perdre :

- car il a touché le policier.
- car il n'a pas atteint le score imposé pour chaque niveau à temps.

Le joueur doit déplacer le personnage à l'aide des touches de direction du

clavier.

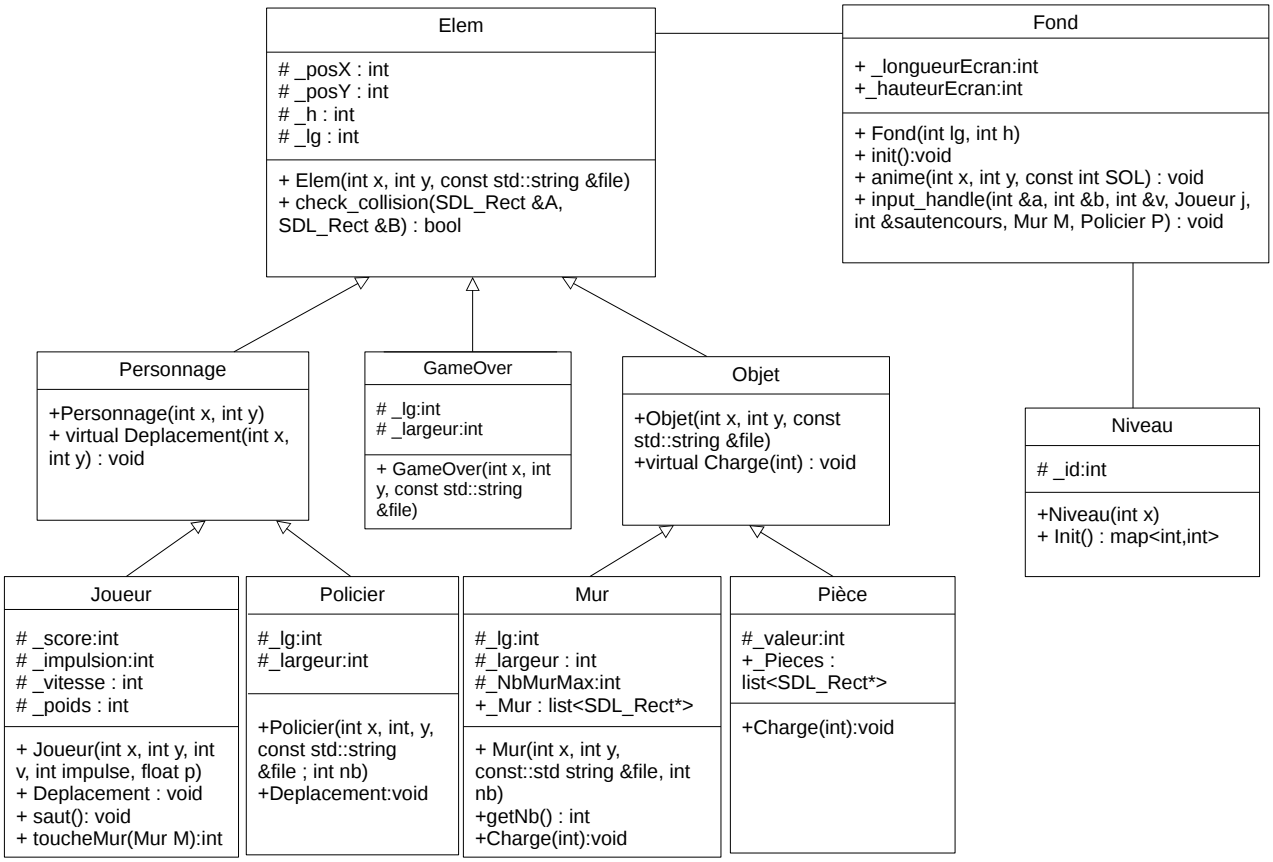
— La touche **UP** pour sauter verticalement.

— La touche **RIGHT** pour se déplacer à droite.

— La touche **LEFT** pour se déplacer à gauche.

Si le joueur veut effectuer un saut tout en se déplaçant d'un coté de l'écran (à gauche ou à droite) en effectuant une parabole, il doit appuyer successivement sur les touches **UP/RIGHT** ou **UP/LEFT**.

2 Diagramme UML



3 Procédures d'installation et d'exécution du code

Pour l'affichage graphique, nous avons décidé d'utiliser la bibliothèque SDL.
Il s'agit d'une bibliothéque écrite en C et trés utilisée notamment pour les jeux en 2D (Comme dans notre cas).
Nous avons choisi cette bibliothèque pour plusieurs raisons :

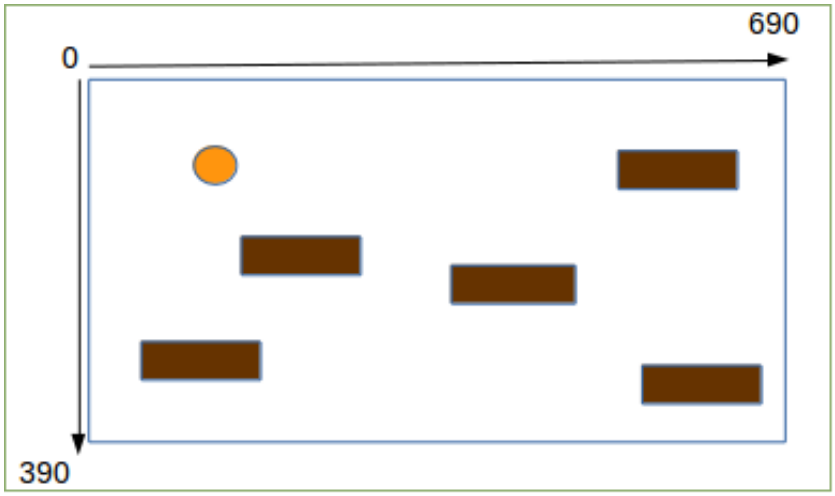
- Il s'agit d'une bibliothèque libre et gratuite que l'on a pu installée sur nos ordinateurs personnels.
- Cette bibliothéque est également disponible sur les ordinateurs de l'école.
- Aprés avoir effectué des recherches sur les différentes fonctions que posséde SDL, nous avons estimé qu'elle était adaptée à notre jeu.

Pour utiliser la librairie SDL, nous devons utiliser un include de SDL : **include <SDL/SDL.h>** et compiler le programme avec l'option **-lSDL**.

Sur le site de la SDL *www.libsdl.org* est disponible le téléchargement de la bibliothèque ainsi que sa documentation.

4 Description d'une partie de l'implémentation

Tout d'abord, pour éviter les problèmes de collision entre les murs et les pièces et pour assurer l'accessibilité de toutes les pièces, nous avons choisi de définir nous mêmes les différents niveaux
C'est à dire que pour chaque niveau, nous avons créé un fichier texte contenant l'ensemble des positions des murs et un autre fichier texte contenant l'ensemble des positions des pièces. La fonction **Charge** virtuelle définie dans la classe **Objet** parcourt les fichiers textes afin de créer une liste de type SDLRect pour les murs ainsi que pour les piéces.



La partie dont nous sommes le plus fière concerne le saut du personnage.

Pour gérer le saut du personnage, nous avons utilisé la fonction **Deplacement** de la classe **Joueur**.

Celle-ci gére le déplacement en x et le déplacement en y, c'est à dire, le saut.

Pour le saut, l'idée était d'attribuer au joueur une vitesse, une impulsion et un poids.

En appuyant sur la touche **UP**, le personnage doit être capable d'effectuer un saut, c'est à dire d'augmenter sa position y puis de la diminuer instantanément jusqu'à atteindre le sol.

Lorsque le joueur presse la touche **UP**, il donne à la vitesse du personnage la valeur de l'impulsion. Et c'est la force du poids qui va attirer le joueur vers le sol.

Puis, à chaque déplacement, selon l'axe vertical (à droite ou à gauche), la vitesse du personnage diminuera de la valeur de son poids jusqu'à ce qu'il revienne au sol.

Sa position en y étant égale à sa vitesse, le joueur va dessiner un demi-cercle en sautant (en formant une ellipse), jusqu'à son retour au sol ou sur un mur. Si aucune touche n'est pressée par l'utilisateur, (vers la droite ou vers la gauche), nous nous sommes servies de la fonction PollEvent de SDL pour que le programme considère un déplacement du personnage de 0 en x. Ainsi, il sautera sur place en modifiant uniquement sa position en y et sa vitesse.

5 Conclusion

Notre projet nous a permis de découvrir une application réelle du langage C++. Nous avons pu constater à quel point la programmation orientée objet pouvait être utile dans certains cas. Cela nous a appris à chercher par nous-même et à manipuler SDL. Contrairement aux TPs, nous avons produit plus ou moins notre propre projet même si ce projet présentait des consignes sous forme de contraintes, de limites et de thème. Pour changer de notre quotidien, nous avons pu faire preuve d'imagination. Nous n'étions pas vraiment guidée et nous avons appris à être à la fois autonome et autodidacte.