

Université de Bordeaux

PROJET TECHNOLOGIQUE

L3

Rapport de Projet

Quentin Fergelot

Yohan Lematre



24 avril 2020

0/ Avant-propos

Suite aux conditions exceptionnelles de cette fin d'année et à l'obligation de devoir travailler à distance, avec tout ce que cela implique, d'un point de vue organisationnel et technique entre autres, il est évident que le travail rendu et présenté ici n'atteint pas les objectifs que nous aurions voulu et pu atteindre dans des conditions normales de travail.

En effet, nous avons rencontré des difficultés à constituer des environnements de travail à peu près corrects et enclin à l'avancée du projet, que ce soit pour cette UE ou pour d'autres. Nous avons atteint cette stabilité seulement une semaine avant ce rendu et avons pu reprendre dès lors le développement du projet.

Ainsi, si nous avons déjà entamé légèrement le travail sur la création, sauvegarde et chargement de cartes de jeu avant le début du confinement, la grande majorité de celui-ci n'a finalement pu être réalisé qu'au cours de la dernière semaine passée.

Nous espérons que ces conditions exceptionnelles pourront être prise en considération dans l'évaluation de ce rendu, et potentiellement excuser les quelques lacunes encore présentes ou fonctionnalités manquantes dans notre projet.

1/ Travail effectué

Nous avons réussi à avancer notre projet jusqu'à la fin de la cinquième feuille de TD, à l'exception de la feuille traitant des collisions plus avancées. Ainsi l'ensemble de ce qui a été demandé jusque-là a été traité et implémenté dans notre projet, qui fonctionne plutôt correctement dans sa globalité.

Ainsi le mode d'édition de carte est fonctionnel, de même que la sauvegarde et le chargement de carte personnalisée.

En jeu, le basculement en mode édition se fait en appuyant sur la touche « *e* » et permet dès lors de déplacer un curseur spécifique – un objet dynamique créé au lancement du jeu en même temps que Mario – permettant de positionner n'importe où sur la carte un bloc choisi par la touche « *espace* ».

Le bloc voulu est sélectionné avec la touche « tabulation » qui permet de parcourir l'ensemble des blocs disponibles. Un bloc vide est également présent et permet quand il est sélectionné d'enlever un bloc déjà présent pour le remplacer par un bloc d'air. Certains blocs ne sont toutefois pas encore présents dans le jeu, tels que les blocs de marbre, d'eau ou de TNT, notamment car ces blocs nécessitent des fonctionnalités supplémentaires que nous comptons réaliser plus tard dans le projet.

Le retour au mode jeu se fait en appuyant sur la touche « *p* » et permet de reprendre le contrôle de Mario là où on l'avait laissé précédemment.

Au lancement du jeu une carte par défaut est chargée, dont les informations sont contenues dans le fichier « *default_map.txt* ». Cependant une carte spécifique peut être chargée au lancement du jeu grâce à la commande :

`./game -l <map_to_load>`

Où **<map_to_load>** est le nom de la carte à charger, au format « *.txt* ».

En jeu, l'appui sur la touche « s » permet à tout instant de sauvegarder la carte courante dans un fichier nommé « *save_map_default.txt* ». Cette sauvegarde s'apparente à une sauvegarde rapide en jeu, puisqu'elle s'effectue toujours dans ce fichier. Pour réellement sauvegarder cette carte, dans le sens de la mettre de côté pour pouvoir la réutiliser ultérieurement tout en sauvegardant d'autres cartes, il faudra renommer le fichier « *save_map_default.txt* » avec un nom choisi.

Quelques légers problèmes persistent dans notre code, mais ils sont mineurs et n'empêchent pas de jouer au jeu correctement. Nous en parlons dans une partie ultérieure de ce rapport.

2/ Difficultés rencontrées

Les principales difficultés que nous avons rencontrées concernent la gestion de la carte et des différents blocs qu'elle contient, ainsi que la mise en place initiale du mode édition.

Nous avons réfléchi quelques temps à la meilleure façon d'aborder la gestion de la carte, et avons finalement opté pour la création d'objets statiques, différents des objets dynamiques précédemment créés.

Le début de l'implémentation s'est montré lui plus délicat, puisque nous ne savions pas vraiment à cet instant comment représenter le curseur permettant de placer les différents blocs.

Celui-ci est finalement devenu un objet dynamique comme nous en parlions précédemment dans ce rapport.

3/ Éléments intéressants de notre projet

Avec la solution des objets statiques énoncée dans la partie précédente nous avons pu optimiser notre code, ce dont nous sommes particulièrement satisfaits.

En effet cela nous permet d'avoir uniquement une structure par bloc différent, chacune contenant le sprite de celui-ci, puis ces structures sont stockées dans un tableau « *static_objet* » de taille le nombre de bloc différents présent dans le jeu et défini dans le fichier « *object.h* ».

Ainsi chaque case du tableau représentant la carte n'est remplie qu'avec une constante, correspondant à l'indice du bloc équivalent dans le tableau « *static_objet* », puis au moment du rendu de la carte à l'écran les blocs de chaque case sont retrouvés grâce à ces constantes puis affichés.

Cela nous permet de réaliser un gain considérable sur le nombre d'objet alloués dynamiquement dans notre code, ce qui n'est pas négligeable.

Un autre point satisfaisant de notre projet est l'ajout d'options permettant de démarrer le jeu directement en mode édition depuis une carte vierge afin de constituer celle-ci comme bon nous semble.

Pour cela, il suffit de lancer le jeu avec la commande :

`./game -w <width> -ht <height>`

Où **<width>** et **<height>** sont les dimensions en nombre de blocs de la carte nouvellement générée. Celle-ci sera par défaut encadrée par un sol et un plafond, ainsi que des murs de chaque côté.

Les modalités de sauvegarde de la carte nouvellement créée restent les mêmes que vu précédemment.

4/ Problèmes connus et améliorations possibles

Comme nous l'avons évoqué quelques problèmes non résolus persistent dans le code de notre projet, plus ou moins importants.

Premièrement, pour rester sur la fonctionnalité permettant de démarrer le jeu directement en mode édition, nous avons remarqué qu'au lancement du jeu l'arrière-plan se met à défiler légèrement, jusqu'à ce que l'on presse la touche « p » pour faire apparaître Mario et commencer le jeu.

Deuxièmement, au chargement d'une carte, nous n'avons intégré aucune méthode de vérification du fichier lu.

Ainsi, un fichier de carte non-conforme sera lu sans aucune interruption par notre code, et peut alors conduire à de nombreux soucis que l'on peut aisément imaginer.

Ce problème est le plus important que nous ayons constaté jusque-là, et devra assurément être corrigé dans les prochaines avancées du projet.

Un autre problème dont nous avons connaissance concerne toujours la lecture de carte.

En effet celle-ci se fait actuellement caractère par caractère, et un traitement est donc effectué pour chaque caractère individuellement.

C'est assurément un souci d'optimisation qui pourra également être corrigé dans la suite du projet.

Enfin, notre système actuel de sauvegarde n'enregistre que les objets statiques pour le moment, soit uniquement les blocs constituant la carte. Les objets dynamiques tels que Mario ne sont pour l'instant pas sauvegardés en même temps que la carte.

Ainsi si l'on avance dans une carte, que nous sauvegardons, quittons et relançons la carte la position de Mario sera réinitialisée, et il faudra donc parcourir le niveau à nouveau jusqu'à retrouver l'avancement atteint précédemment.

Quant aux améliorations possibles de notre projet, nous pouvons évoquer l'ajout d'autres objets dynamiques tels que les chars, que nous n'avons pas encore traités, ce qui rend le jeu encore peu intéressant puisqu'aucun ennemi n'est présent.

On pourra également s'intéresser comme évoqué juste au-dessus à l'ajout des objets dynamiques dans la sauvegarde de carte, afin d'enregistrer l'avancée de Mario et des autres personnages dans un niveau.

Enfin, cette sauvegarde pourrait également bénéficier d'un changement pour enregistrer les données dans un fichier binaire, non lisible par un utilisateur mais se montrant plus pratique pour plusieurs points.