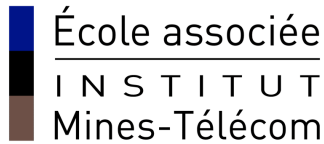


FLAP FLAP BUGGLE

KEJJI Lamy – LAPORTE-CHABASSE Quentin – SALVESTRINI Lolita



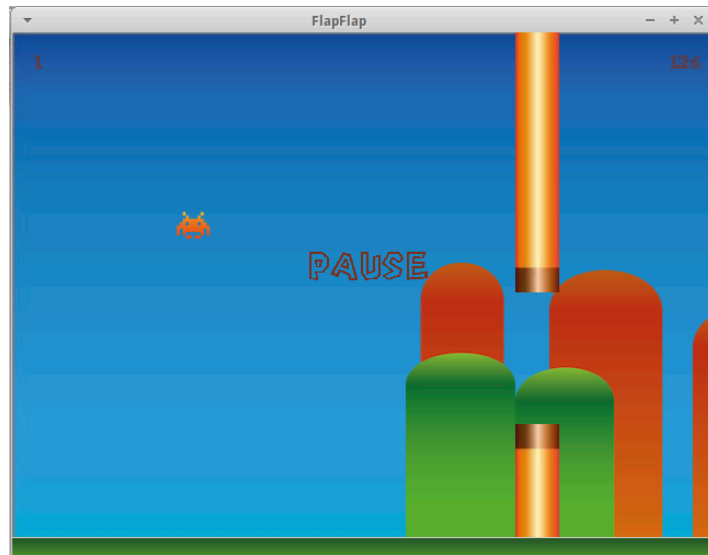
I. Présentation du jeu

Dans ce jeu le joueur doit faire avancer un Buggle en évitant des tuyaux présents en haut et en bas de l'écran. Si le Buggle touche le sol, le ciel ou les tuyaux, la partie est terminée ! Le jeu comporte également des bonus : augmentation du score lorsque le Buggle touche un bagel et disparition des obstacles pendant une période donnée lorsque le Buggle touche la boîte mystère.

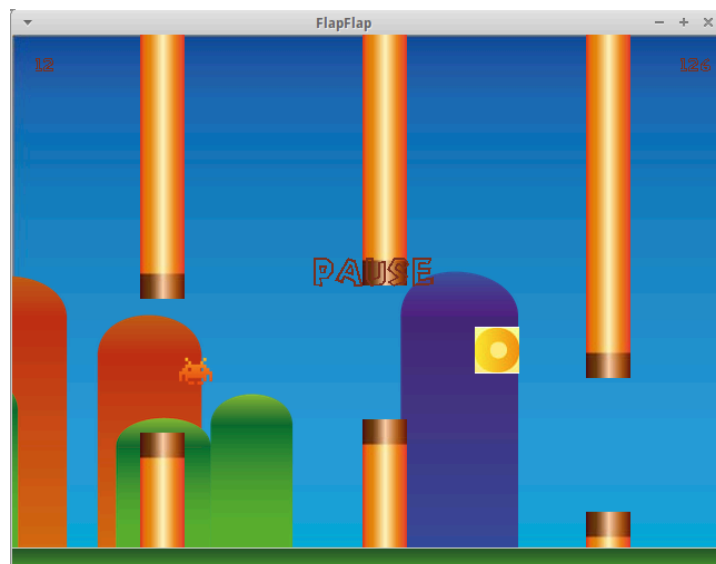
En haut à gauche de l'écran se trouve le score du joueur qui dépend du temps écoulé depuis le début de la partie. A droite se trouve le meilleur score enregistré dans un fichier texte. A chaque début de partie ce score est lu par le programme, dans le cas où le joueur bat son score, ce score sera enregistré à la fin de la partie dans ce même fichier texte.



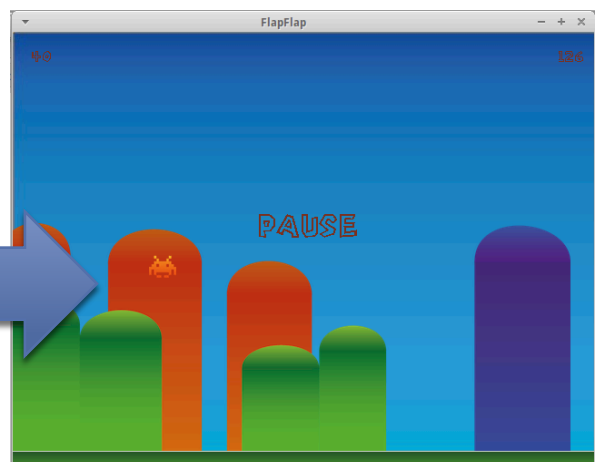
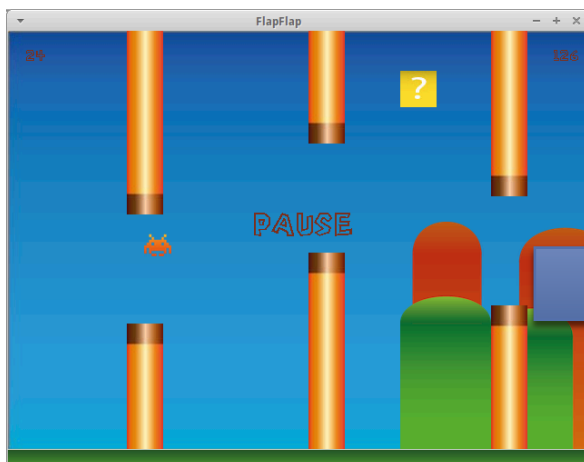
Ecran d'accueil du jeu



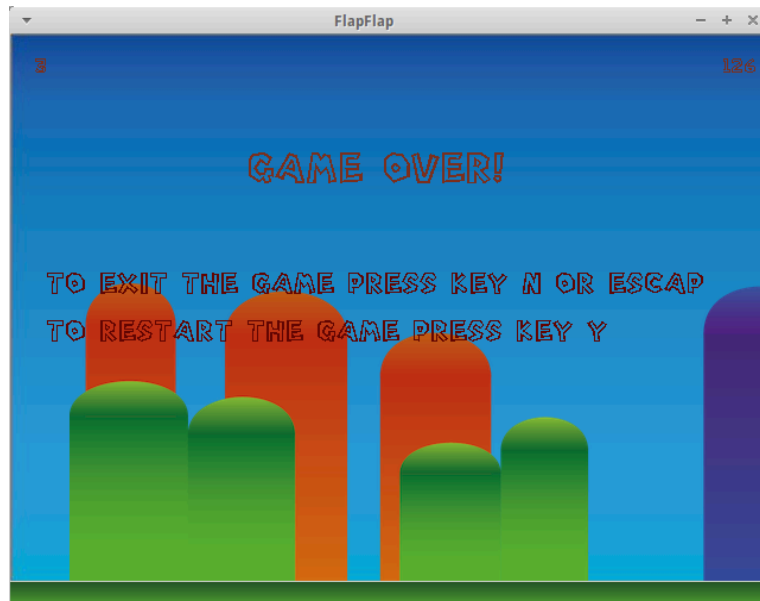
Possibilité de mettre le jeu en pause



Génération aléatoire des obstacles, génération d'un bonus bagel. Si le Buggle touche le bagel le joueur gagne un bonus point



Génération d'un second bonus mystère qui fait disparaître les obstacles pendant un temps limité



Quand le joueur rentre en collision avec un obstacle, le ciel ou la terre, l'écran « game over » s'affiche, le joueur a alors la possibilité de quitter le jeu ou de rejouer

II. Problèmes et solutions

A. La conception du jeu

Ce projet nous a amené à surmonter de nombreuses difficultés liées au codage mais aussi au design du jeu :

Tout d'abord, l'installation d'Allegro. Les tutoriels trouvés n'étaient pas suffisamment explicite pour des débutants sous linux. Il a donc fallu demander de l'aide à un camarade (Xavier Wirth) qui maîtrisait l'environnement linux et l'IDE Code Blocks.

Le raisonnement sur l'enchaînement des obstacles fut l'un de nos plus gros problèmes. En effet, la génération aléatoire des obstacles n'était pas intuitive, tout comme l'enchaînement « à l'infini ». On s'est rendu compte qu'il y avait un décalage entre ce que le joueur voit et ce qui est écrit dans le code. En effet, au début à un obstacle correspondait un bitmap mais on a finalement compris que chaque obstacle avait le même bitmap mais qu'il était décalé.

Pour la génération aléatoire des obstacles, on a 6 obstacles différents positionnés aléatoirement d'une façon « intelligente » c'est-à-dire que l'obstacle suivant prend en compte la position de l'obstacle précédent et rend ainsi le jeu « jouable » (évite les aberrations – les trop grands écarts entre les pipes).

Pour ce qui est de l'enchaînement des obstacles, il a fallu créer deux trames successives. A la disparition de la première (à gauche de l'écran), elle se régénère automatiquement et se place à la suite de la seconde trame.

De la même manière la création des bonus fut difficile à réaliser, il a fallu tout d'abord structurer l'entité bonus et borner ses effets (elle ne devait pas perturber le fonctionnement du jeu). De plus, comme pour les obstacles le joueur doit percevoir l'apparition des bonus comme un événement aléatoire. La difficulté étant aussi de créer un phénomène aléatoire « intelligent ». Il faut garder une certaine cohérence dans l'apparition des bonus, sans quoi le jeu deviendrait rapidement injouable.

Finalement, le langage C nous offrant une grande liberté, ce qui peut être pratique mais parfois dangereux, il a fallu structurer le code pour regrouper les différentes entités du jeu. Grâce à ce projet, nous avons pris conscience de la nécessité d'une pensée et d'un code structurés.

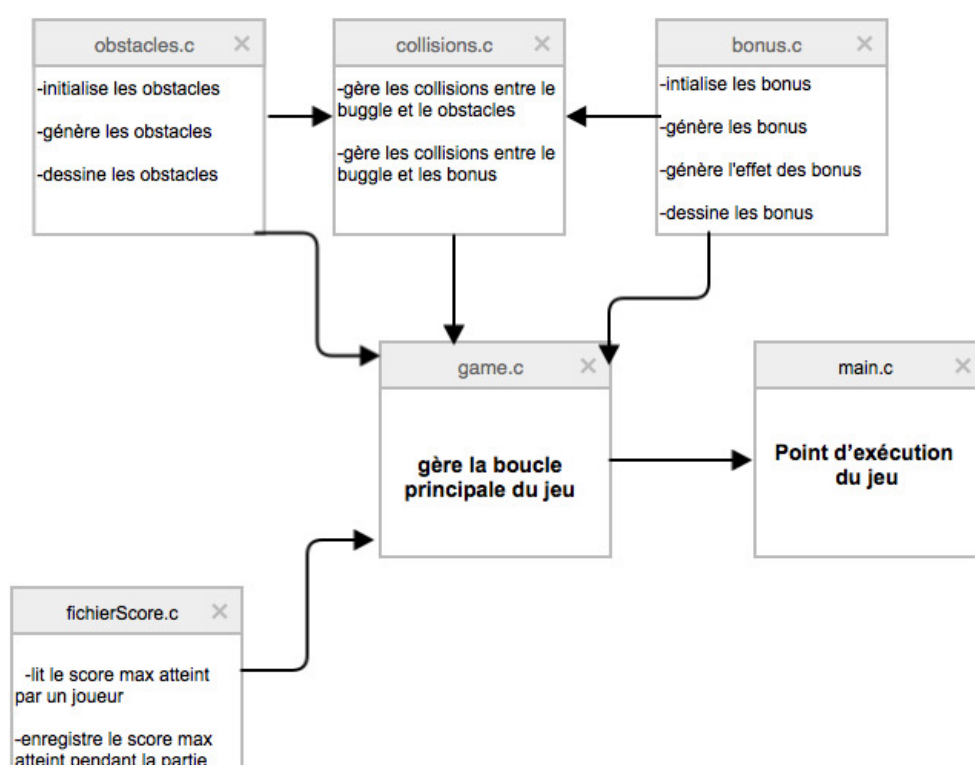
B. Travailler à trois autour d'un même projet

Travailler dans un groupe de trois personnes sur un projet se révèle beaucoup plus difficile que de travailler en binôme.

Le travail doit être partagé de manière équitable, ce qui n'est pas évident lorsque le niveau en programmation n'est pas homogène. En outre il faut un minimum de communication, ce qui nous a légèrement manqué (en particulier à cause des vacances scolaires et de l'éloignement géographique).

De plus, le projet s'est finalisé pendant les vacances, la mise en commun du code source s'est donc révélé compliquée, les services de partage de fichier tels que dropbox ou google drive n'étant pas adaptés pour cette utilisation.

III. L'organisation du projet



Nous avons scindé notre projet en différents fichiers, nous avons fait le choix de regrouper les méthodes et les structures par fonctionnalité.

IV. De l'ébauche au résultat final

Au moment du codage, le débogage a occupé une partie non négligeable de notre temps de travail.

Conception : Quentin 20h – Lolita 20h – Lamy 20h

Codage : Quentin 30h – Lolita 20h – Lamy 20h

Rédaction du rapport et des commentaires : Lolita 4h – Quentin 4h

V. Sources

Afin de mener à bien ce projet nous nous sommes appuyés sur quelques sources web :

- <http://www.anothergames.com>
- <http://www.openclassrooms.com>
- <https://www.allegro.cc/manual/5/>

VI. Licence

- L'ensemble des images constituant le jeu sont distribuées sous la licence creative commons CC-ZERO. (elles ont été créées à l'occasion de ce projet)
- Le code source est quant à lui distribué sous la licence GNU GPL V2.

Le jeu peut donc être distribué librement.