

Projet de C

Square Escape

Brizion Pierre
Debiard Marc
Pons Pascal
Lasolle Nicolas



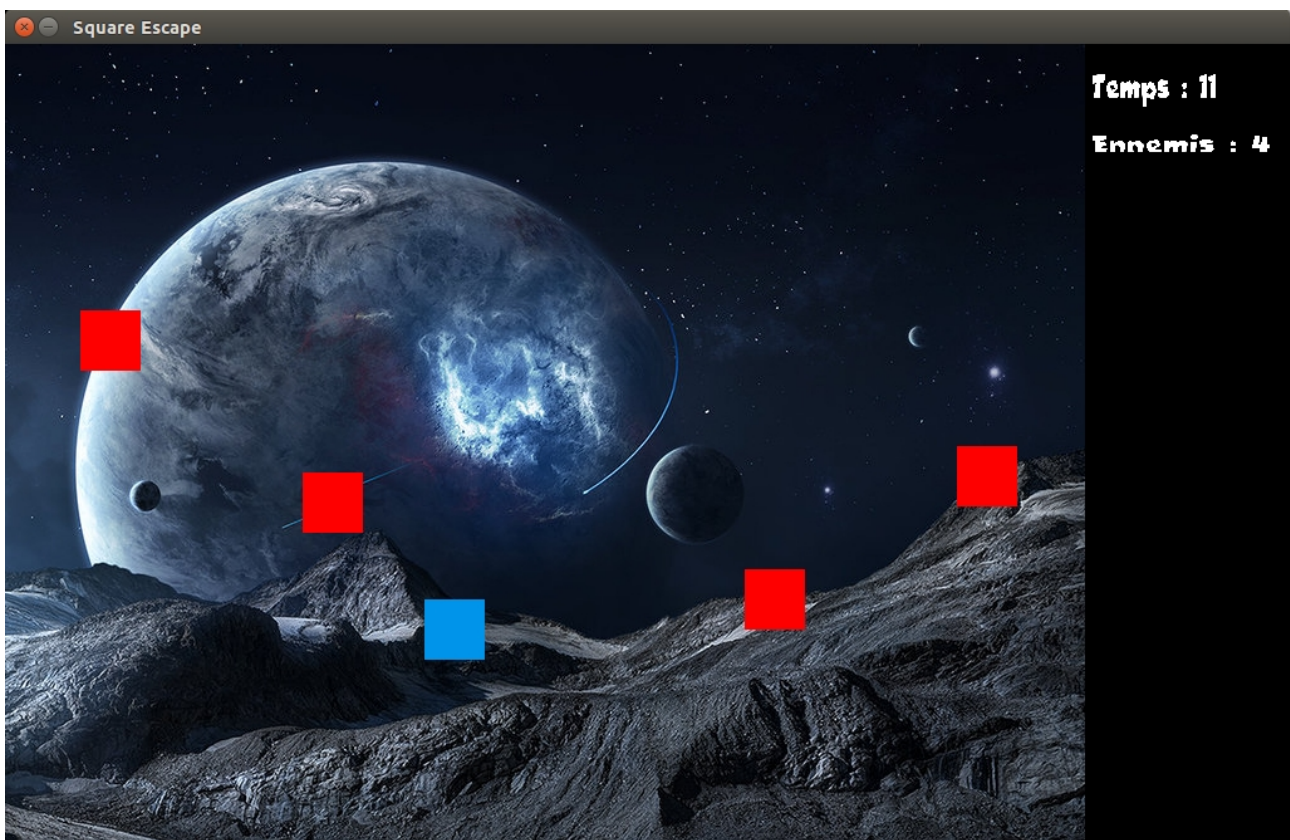
L'objectif principal de ce projet consiste en la réalisation d'un jeu en 2D, entièrement écrit en langage C. Pour la réalisation de l'interface graphique, notre choix se portera sur la SDL (Simple DirectMedia Layer), bibliothèque la plus couramment utilisée en C, et très pratique d'utilisation. Nous utiliserons la version la plus récente et plus stable à savoir la version 2.0.

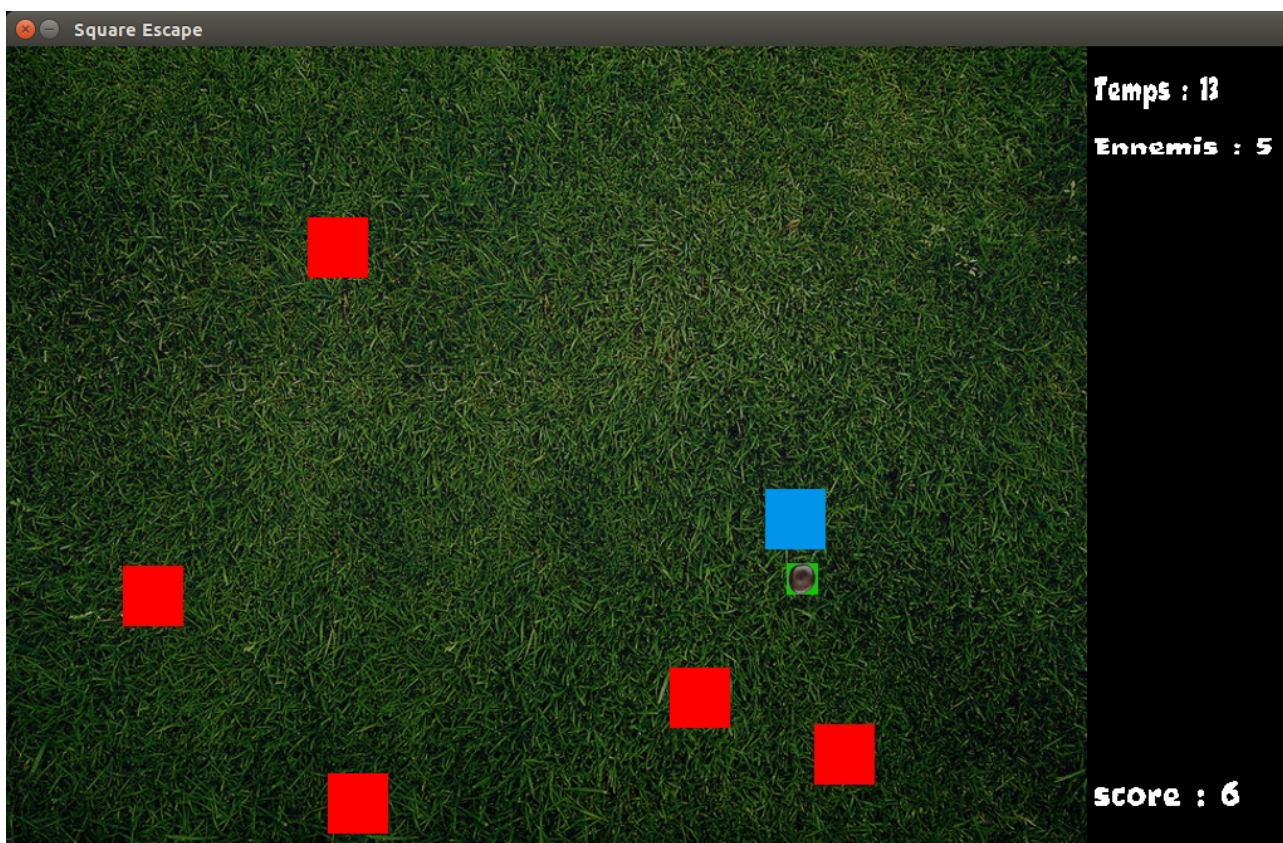
Objectif principal:

L'idée est de réaliser un jeu d'adresse dont voici les grandes lignes:

- Le joueur contrôle un cube qu'il peut déplacer librement dans une fenêtre donnée et fixe.
- Des cubes apparaissent à l'écran à intervalle de temps réguliers. Ils rebondissent contre les bords de la fenêtre de jeu.
- Le but du jeu est d'éviter les cubes à l'aide du clavier sachant que le joueur perd dès qu'il a touché un cube.
- Un système de score correspondant au nombre de cubes présents (et/ou score en secondes) au moment de la fin de partie devra être mis en place.

Images du jeu :





Planning prévisionnel:

<u>Semaine</u>	<u>Objectifs</u>
30/03	Prise en main basique C+ SDL
06/04	Prise en main un peu plus poussée → Visualiser la façon dont va marcher notre jeu avec SDL
13/04	Déplacements du personnage, gestion des événements basique(quitter, rejouer,pause...)
20/04	Gestion des collisions avec objets fixes et bords de l'écran
27/04	Déplacement automatique des cubes
04/05	Aspect graphique du jeu+ gestion musique+gestion du score
11/05	Une première version complète jouable avec l'objectif principal atteint
18/05	Préparation soutenance+fonctionnalités annexes
25/05	Soutenance...

Ensemble des sources et aides:

- Open Classrooms → <http://openclassrooms.com/courses/developpez-vos-applications-3d-avec-opengl-3-3/creer-une-fenetre-avec-la-sdl-2-0>
 - <http://openclassrooms.com/courses/simplifier-les-events-avec-sdl>
 - <http://openclassrooms.com/courses/jouer-du-son-avec-sdl-mixer>
- Wiki officiel → <https://wiki.libsdl.org>
- Sur developpez.com → <http://alexandre-laurent.developpez.com/tutoriels/sdl-2>
- Un exemple complet de jeu SDL pour source inspiration → <http://www.meruvia.fr/index.php/big-tuto-sdl-2-nouveau>

Organisation du projet:

Pour mener à bien ce projet, nous avons tout d'abord dû apprendre à maîtriser les bases de la librairie SDL, chacun de notre côté. Une fois chose faite nous avons essayé de comprendre les algorithmes principaux et l'organisation globale du projet pour gérer l'ensemble du jeu. Le programme doit être découpé en différents modules pour plus de clarté. Nous avons donc un ensemble de fichiers d'en-tête. L'un qui inclut les librairies C nécessaires à la compilation, l'un pour les constantes globales, l'un pour les différentes structures du jeu, et les autres qui contiennent les prototypes des fonctions qui seront appelées dans le main. Notre fichier contenant la fonction principale «main» doit être le plus simple et court possible.

Nous avons aussi essayé de respecter certaines conventions de nommages. Les variables auront toute la première lettre de leur nom en minuscule. Les pointeurs devront avoir le nom commençant par un p (Object * pObject).

Pour travailler en équipe, nous avons choisi d'utiliser le dépôt GitHub et l'utilitaire Git pour remonter tout changement sur le dépôt distant. Cela permet de gérer en temps réel les différentes versions du projet et d'ainsi pouvoir observer les différentes modifications effectuées par les membres du groupe. Cependant nous avons du mal à gérer correctement cet outil dans le cas où plusieurs personnes travaillent simultanément sur une même portion du programme.

Difficultés rencontrées:

Voici les principales difficultés rencontrées au cours de ce projet :

- Peu d'informations sur la SDL2.0, beaucoup de liens vers la 1.2
- Des difficultés à démarrer le projet (avoir une base correcte avec un affichage de sprites sur un fond)
- Difficultés liées au travail de groupe: (partager les idées, se mettre d'accord sur un point donné, se répartir les tâches, gérer les versions sur GitHub quand il y a un travail en parallèle sur des mêmes fichiers)
- Problème de mémoire : bien gérer les pointeurs (un problème de suppression de texture a entraîné une fuite mémoire énorme (plusieurs Go...)) et trouver la source des erreurs de segmentation
- Traiter les choses les unes après les autres, ne pas partir dans tous les sens
- Créer des structures logiques et utiles qui pourront facilement être intégrables au programme
- Difficultés d'appréhension de certains principes C qui crée un déséquilibre au niveau des membres sur certains points
- Des changements mineurs qui impliquent de revoir certaines portions du code au fur et à mesure
- Pas le temps de remplir tous les objectifs annexes