

# Projet Technique de Programmation : PolyTron

---

## I. Présentation :

Il s'agit de produire un programme (fichiers source et un exécutable testé et opérationnel), dont le sujet est présenté dans la suite du document. Le développement de ce programme se fera en langage C.

### A. Objectif du projet

- Mettre en œuvre les notions du cours de C
- Aborder des notions d'architecture logicielle
- Développer un regard critique dans l'implémentation
- Organiser son implémentation et avoir une vision globale du travail à réaliser
- Définir des interfaces
- Développer une interface graphique
- Intégrer une librairie externe
- Gestion d'évènements
- Aborder des notions de temps réel
- Aborder des notions de gestion de fichier

### B. Organisation

Le projet est individuel.

Le projet est pensé pour être réalisable durant les TP du cours de « Technique de Programmation ».

### C. Ce qu'il faut rendre

- Le programme source complet largement commenté
- Un exécutable testé et opérationnel
- Présentation oral du travail effectué de 10 minutes lors de la dernière séance :
  - Organisation du programme: découpage en fonctions, rôle de ces fonctions, explications du programme
  - Mode d'emploi du programme
  - Bilan qualitatif du travail, difficultés rencontrées, etc.

Le tout sera placé dans un fichier archive à m'envoyer par mail.

## **D. Evaluation du projet**

Elle repose sur les éléments suivants:

- Le programme source:
  - Respect de l'énoncé
  - Qualité technique du programme: découpage en fonctions, instructions, algorithmes, efficacité, gestion des erreurs...
  - Présentation du programme: indentation, commentaires et nommage
- Présentation oral du travail :
  - Organisation du programme et son mode d'emploi
  - Démonstration du programme
  - Interrogation individuelle sur le travail réalisé
  - Bilan

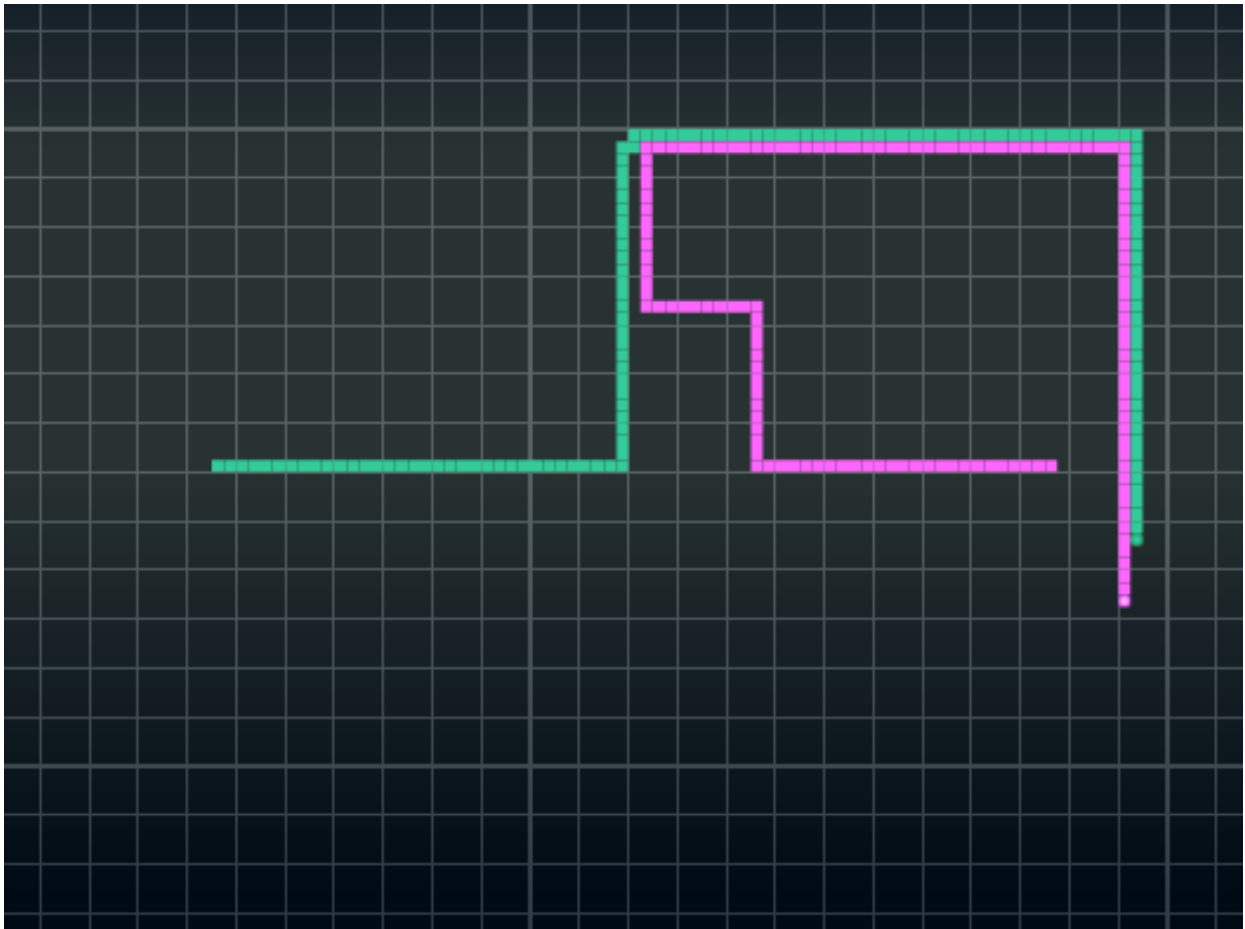
## II. Présentation du projet : PolyTron

### A. But du jeu

PolyTron est un jeu de moto seul ou à 2 joueurs dont le but est de détruire l'adversaire en lui barrant la route.

Les 2 motos sont sur une grille fermée, et chaque moto laisse une trainée qui constitue un mur infranchissable pour les 2 joueurs. Ce mur est persistant tout au long d'une partie.

Les motos se déplacent automatiquement, le joueur ne peut que changer la direction.



## **B. Modes de jeu**

PolyTron propose 3 modes de jeu :

- Survie
- Player Vs Player
- Player Vs IA

### **1. Survie**

Dans ce mode de jeu, le joueur est seul sur la grille. Il doit essayer de survivre le plus longtemps possible.

Le joueur engrange des points à chaque pixel où il avance.

Un score est donc à afficher dans ce mode de jeu et sauvegardé pour les High Scores.

Une fois la partie terminée, on pourra par exemple afficher en grand le score.

### **2. Player Vs Player**

Dans ce mode de jeu, 2 joueurs s'affrontent.

Il faudra nommer chacun des joueurs et choisir 2 couleurs différentes afin de bien les distinguer.

Si les 2 joueurs se percutent simultanément, il y a match nul.

Une fois la partie terminée, on pourra par exemple afficher le nom du gagnant.

Libre à vous de faire une partie en plusieurs manches.

### **3. Player Vs IA**

Dans ce mode de jeu, le joueur joue contre une intelligence artificielle, c'est-à-dire qu'il faudra développer un algorithme pour générer des déplacements et des choix pour le joueur non humain.

Il faudra nommer chacun des joueurs et choisir 2 couleurs différentes afin de bien les distinguer.

Si les 2 joueurs se percutent simultanément, il y a match nul.

Une fois la partie terminée, on pourra par exemple afficher le nom du gagnant.

Libre à vous de faire une partie en plusieurs manches.

## C. Menu

Au démarrage du jeu, un menu s'affiche et propose les points suivants :

- Jouer :
  - Survie
  - Player Vs Player
  - Player Vs IA
- High Scores
- Paramètre :
  - Contrôles
  - Vitesse des motos

## D. High Scores

Le mode survie s'accompagne d'une gestion des High Scores :

A la fin d'une partie, le score est sauvegardé.

L'ensemble des scores des joueurs peut être affiché dans la catégorie High Scores.

Les High Scores doivent être affichés dans l'ordre du plus grand au plus petit score.

Pour conserver les scores même lorsqu'on a fermé le jeu, ceux-ci doivent être sauvegardés dans un fichier. Ce fichier contient par exemple les informations suivantes pour chaque ligne :

- Nom du joueur
- Score réalisé
- Classement

Windows propose des fonctions dans la library stdio.h afin d'interagir avec des fichiers :

- fopen : crée ou ouvre un fichier
- fread : lit des caractères
- fwrite : écrit des caractères
- fseek : déplacer le curseur dans le fichier

Le fichier de sauvegarde est placé dans le dossier courant (au niveau de l'exé).



1 <sup>ST.</sup>	1520019	LJM
2 <sup>ND.</sup>	1363609	CJM
3 <sup>RD.</sup>	107400	APE
4 <sup>TH.</sup>	98410	BAT
5 <sup>TH.</sup>	89330	CAT
6 <sup>TH.</sup>	76300	DOG
7 <sup>TH.</sup>	60740	EEL
8 <sup>TH.</sup>	59630	FOX
9 <sup>TH.</sup>	49890	GNU
10 <sup>TH.</sup>	33960	HEN

## **E. Contrôles**

Les contrôles des motos se font dans les 4 directions : HAUT, BAS, GAUCHE et DROITE. On utilise le clavier pour diriger les motos.

Voici une proposition de jeu de touche :

- Joueur 1 :
  - Flèches numériques
- Joueur 2 :
  - Haut : Z
  - Bas : S
  - Gauche : Q
  - Droite : D

Il est toutefois possible de rendre les contrôles configurables. Un bouton pause ou exit peuvent être également rajoutés.

## **F. Vitesse de déplacement des motos**

Les motos se déplacent automatiquement, le joueur ne peut que changer la direction.

Le programme doit donc mettre à jour les emplacements des motos et des murs à chaque intervalle  $\Delta t$  de temps.

Windows propose des fonctions dans sa library Windows.h afin de gérer des évènements timés (ou timer) :

- SetTimer : Crée et démarre un timer. Prends une callback (fonction) en entrée
- KillTimer : Arrête un timer

Il est possible de rendre la vitesse des motos configurables afin de créer plusieurs niveaux de difficulté.

## **G. Interface graphique**

Le jeu propose une interface graphique en 2 dimensions montrant les motos vues du dessus évoluant dans une grille fermée.

Il est recommandé d'utiliser la library SDL pour l'interface graphique :

## **H. Musique**

Une musique et des bruitages accompagnent le jeu.

Il est recommandé d'utiliser la library SDL pour lire des sons

## **I. Evènements**

Afin de diriger la moto, il sera nécessaire de gérer des évènements sur le clavier.

Il est recommandé d'utiliser la library SDL pour gérer les évènements.

### III. Une library Multimedia : SDL

Quelques infos sur la library SDL :

- Open source
- Facile et complète
- Multimedia
- Compatible avec le C
- Télécharger la lib : <https://www.libsdl.org/>
- Documentation en ligne : <https://wiki.libsdl.org/>

Exemple de fonctions pouvant être utiles :

- General
  - SDL\_Init : charge la SDL en mémoire
  - SDL\_Quit : libère la SDL de la mémoire
  - SDL\_GetError : Retourne une erreur venant de la library
- Affichage
  - SDL\_SetVideoMode : Ouvre une fenêtre graphique
  - SDL\_WM\_SetCaption : Change le titre de la fenêtre
  - SDL\_FillRect / SDL\_MapRGB : Colorier une surface unie
  - SDL\_Flip : Rafraîchir la fenêtre
  - SDL\_CreateRGBSurface / SDL\_FreeSurface : Allouer / libérer nouvelle surface
  - SDL\_BlitSurface : Ajouter une surface
  - SDL\_LoadBMP : Charge une image
  - SDL\_WM\_SetIcon : Associer un icône à la fenêtre
- Évènement
  - SDL\_WaitEvent : Attends un évènement
  - SDL\_PollEvent : Récupère un évènement
  - SDL\_KEYDOWN / SDL\_KEYUP : Évènement clavier
  - SDL\_MOUSEBUTTONDOWN / SDL\_MOUSEBUTTONUP / SDL\_MOUSEMOTION : Évènement souris
- Audio
  - SDL\_OpenAudio : Initialisation d'une nouvelle couche audio
  - SDL\_CloseAudio : Fermeture d'une couche audio
  - SDL\_LoadWAV : Charge un fichier WAV
  - SDL\_FreeWAV : Libérer le buffer de lecture
  - SDL\_MixAudio : Lire le fichier audio
  - SDL\_PauseAudio : Suspendre une lecture

#### IV. Gestion des déplacements et des évènements

