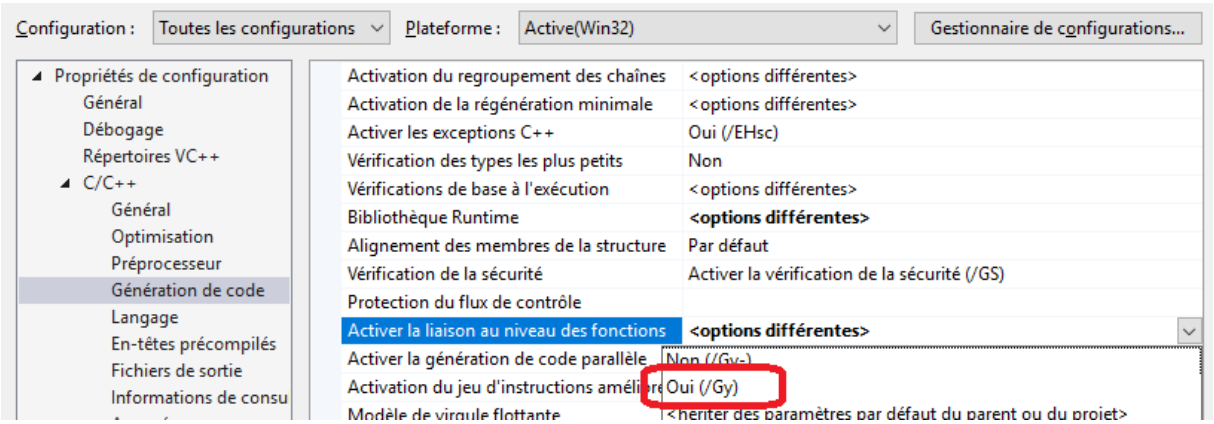
Cours 8IAR125: Intelligence Artificielle pour le Jeu Vidéo

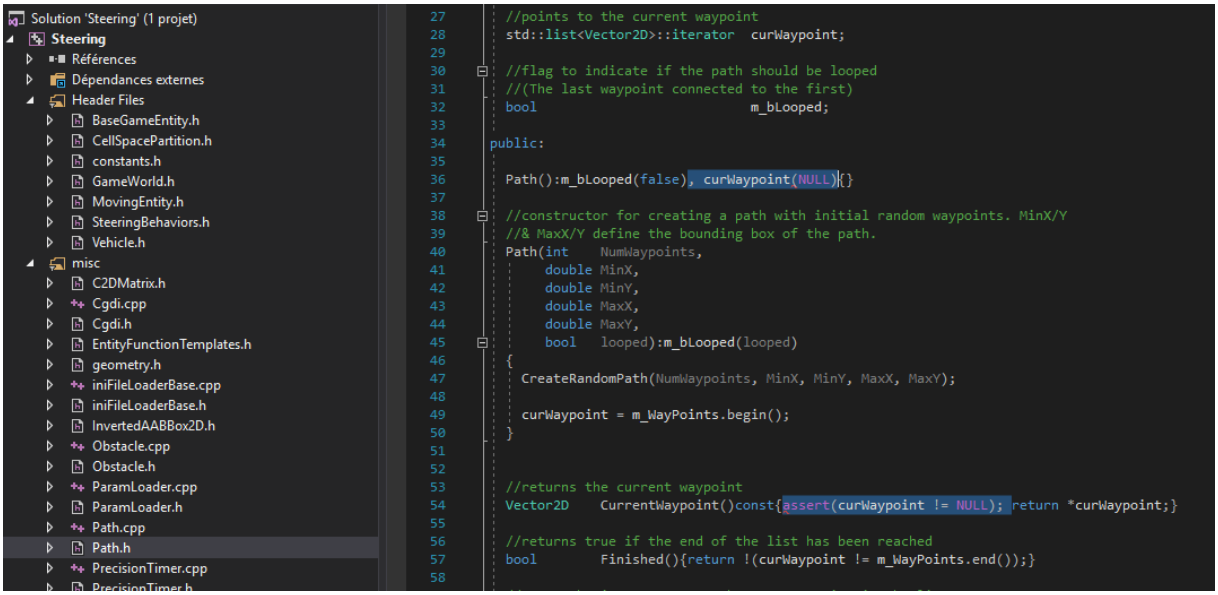
Instructions et conseils
Laboratoire #2 : Steering Behavior

1. Instruction d'installation de l'application

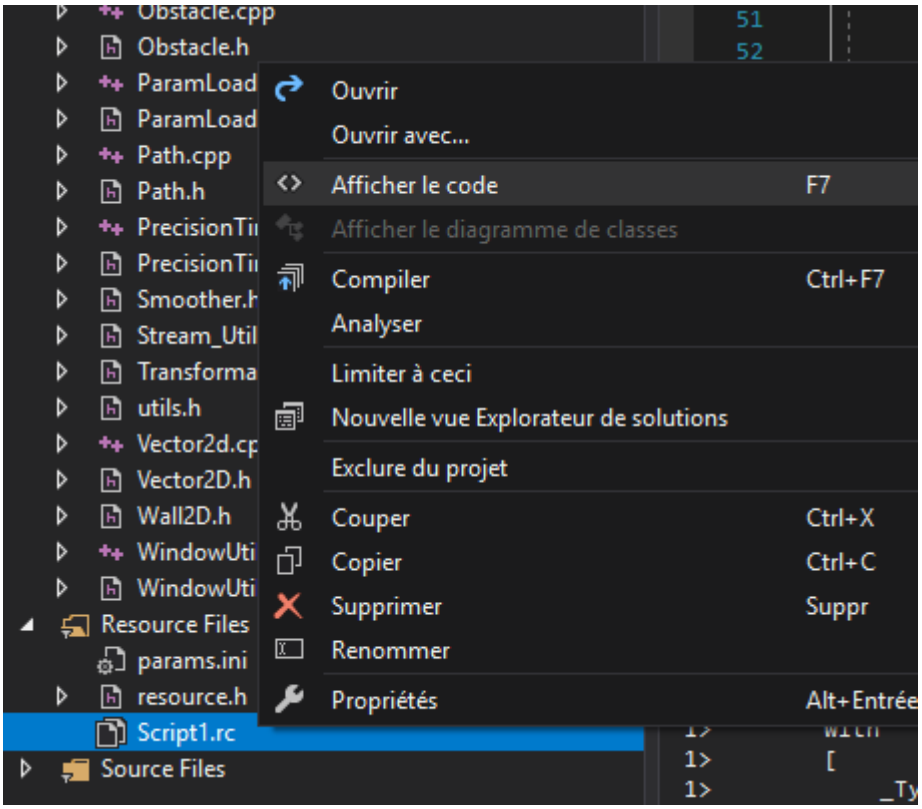
Suivez les mêmes étapes que pour le premier laboratoire jusqu'à la modification de l'option **Activer la liaison au niveau des fonctions** avec la valeur **Oui (/Gy)**. (*C/C++ > Génération de code*).



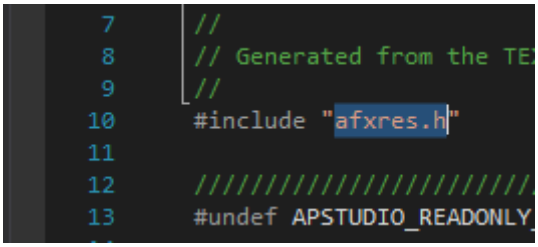
Le code a besoin d'être modifié pour pouvoir compiler. Ouvrez le fichier **misc/Path.h**. Supprimez , **curWaypoint(NULL)** à la ligne 36 et **assert(curWaypoint != NULL);** à la ligne 54 comme sur l'image suivante.

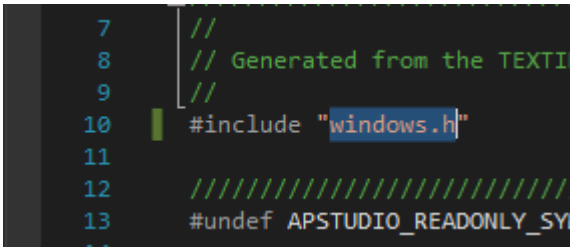


Ouvrez le fichier **Resources Files/Script1.rc** avec clic droit > Afficher le code.



A la ligne 10, remplacez **#include "afxres.h"** par **#include "windows.h"**.





Avant de générer, modifiez le nombre d'agents dans le fichier **Resources Files/params.ini** à la ligne 2 avec un nombre plus raisonnable comme 30.

Lors de la génération, vous aurez plusieurs « warning ». N'en tenez pas compte.

2. Conseils pour réaliser le laboratoire

1) Observer le constructeur de la classe **GameWorld**. Voyez comment il crée des instances de Vehicle et définit leur comportement à l'aide de la méthode **pVehicle->Steering()->FlockingOn()**, à la ligne 70 de GameWorld.cpp.

2) Pour les **classes d'agent poursuiveur et d'agent leader**, faite les **hériter de Vehicle**. Il vous suffira ensuite de correctement définir les **comportements attendus** à l'aide des mêmes méthodes que dans GameWorld mais cette fois dans le **constructeur** de vos nouvelles classes.

3) Pour les touches de clavier afin de contrôler le leader, référez-vous à la méthode **HandleKeyPresses** () qui se trouve dans la classe GameWorld à l'intérieur du fichier GameWorld.cpp.

Bonne continuation !