

Examen OS - Windows API

Polytech Paris Saclay - ESR 4

On souhaite concevoir une application Windows qui modélise une version simplifiée du classique jeu de **Tetris** (pas de rotation de pièces ni calcul de score) et qui peut s'exécuter sous maximum 2 instances simultanées. Utiliser le projet fourni et comparer votre solution à la démo attachée.

A. Utiliser l'objet de synchronisation approprié dans la fonction **limited** pour s'assurer que seulement **maxInst** instances (2) puissent exécuter en parallèle la fonction **test1**. Regarder attentivement l'aide sur les APIs de création et de modification d'état de l'objet choisi en passant les bons paramètres.

B. Créer un constructeur pour la classe **TetrisWind** compatible avec **test1** qui affiche une fenêtre non-déformable de titre "Tetris", avec un menu et de taille cliente horizontale de **DIMX*STEP** (12*32) pixels et verticale de **DIMY*STEP** (22*32) pixels. Passer un rectangle membre statique de taille approximative correcte au constructeur de la classe mère puis comparer dans le constructeur la différence entre la taille réelle et celle attendue de la zone cliente et finalement corriger la taille à l'aide d'une API qui change la taille de cette fenêtre.

C. Rajoutez un attribut tableau 2D de **DIMY x DIMX** qui garde les cases du jeu pouvant être occupées par les pièces, initialisées à chaque début de jeu à l'aide d'une méthode **Reset** à zéro (correspondant à l'état non-occupé alors qu'une case occupée par une pièce d'index **Idx** portera la valeur **Idx+1**). Rajouter le nécessaire pour afficher d'une manière persistante l'état du tableau sous forme de carrées de taille **STEPxSTEP** de contour noir et couleur intérieure désignée par le tableau statique constant **color** (fabriquer les pincesaux nécessaires à la construction puis gérer-les).

D. Modéliser l'état du jeu par un attribut **state** de type énumération **EState** qui aura une valeur parmi **ST_Stop** (jeu arrêté), **ST_NewPiece** (tirage d'une nouvelle pièce), **ST_Dropping** (pièce courante qui tombe), **ST_Dropped** (pièce qui est tombée et posée). La valeur initiale est bien évidemment **ST_Stop** mais elle change à **ST_NewPiece** si l'utilisateur clique sur l'élément de menu "Nouveau jeu". Dans cet état on active le *timer* avec une période de 600ms et on tire une nouvelle pièce en balayant d'une manière circulaire le tableau **piece** contenant les pièces de Tetris (l'index de la pièce courante est gardé dans l'attribut **currIdx** et sa position de type **POINT** dans l'attribut **currPos**). S'il n'y a pas de place pour la nouvelle pièce c'est la fin du jeu en passant à **ST_Stop** avec l'arrêt de *timer*. S'il y a de place alors elle est placée et on bascule à l'état **ST_Dropping** où elle tombe une ligne par coup de *timer*. Chaque fois on efface la pièce courante du tableau de cases et on teste sa nouvelle position (une ligne plus bas). Si la descente est impossible alors la pièce est posée définitivement sur l'ancienne position, on efface les éventuelles lignes complètement occupées et on passe à l'état **ST_NewPiece**. Si la descente est possible on place la pièce à la nouvelle position et on attend le prochain coup de *timer*.

Le code qui gère tous les changements d'états est placé dans les méthodes **WmCommand** et **WmTimer**. Chaque changement dans le tableau de cases invalide la fenêtre pour déclencher un affichage persistant (il n'y a jamais d'affichage non-persistant !).

Le code utilise plusieurs fois la méthode privée **Test** qui teste le placement de la pièce courante d'index **currIdx** et position **currPos** et la méthode **Replace** qui remplace les cases occupées par la pièce courante avec un autre index (**0** si l'on veut effacer ou **currIdx+1** si l'on veut la placer).

E. Utiliser les touches **VK_LEFT** et **VK_RIGHT** pour déplacer la pièce courante pendant l'état **ST_Dropping** (même démarche : effacer la pièce, tester la nouvelle position, placer et invalider si succès, revenir à l'ancienne position sinon). Utiliser la touche **VK_SPACE** pour activer ou non un état de pause pendant un jeu : pendant la pause on ignore le coup de *timer* et les touches de déplacement latéral. On change aussi le titre de la fenêtre (comme dans la démo).

F. La fermeture de la fenêtre par le menu ou la croix demande l'avis de l'utilisateur seulement pendant un jeu.