

Clavier braille

Documentation développeur

Le projet est divisé en deux sous projets distincts :

- « InputSimulator » qui gère la création des évènements de touches
- « clavier braille » regroupe l'interface graphique et la récupération des évènements clavier

InputSimulator

Ce projet provient de la source suivante : <http://inputsimulator.codeplex.com/>

Toutes les classes et les structures de ce projet sont des redéfinitions de classes et de structures données par les API Windows à l'exception de la classe « InputSimulator ». Cette classe permet normalement la simulation d'évènements claviers et souris mais nous l'avons modifiée et allégée pour ne conserver que les parties qui nous sont réellement utiles. Ainsi, nous n'avons conservé qu'une fonction permettant de générer un évènement correspondant à un caractère.

Cette fonction, « SimulateKeyPress », prend en paramètre un caractère et va ensuite générer deux évènements : un keyDown et un keyUp pour simuler un appui sur ce caractère. Nous parlons bien ici de caractère car on ne simule pas l'appui sur une touche mais directement d'un caractère particulier. Il est ainsi possible de générer le caractère « ∞ » bien qu'aucune touche ne lui corresponde directement.

Clavier braille

C'est le sous projet principal. Il gère l'interface graphique, la récupération des touches appuyées et la traduction de ces touches en caractères. Il est décomposé en plusieurs classes :

hookClavier

C'est la classe gérant la récupération évènements et la traduction des touches. Elle se sert notamment d'un hook sur le clavier. Les hook sont des fonctions permettant « d'attraper » tous les évènements claviers qui sont générés pour pouvoir déclencher des actions suivant les évènements reçus.

Les attributs « keyboardHookProc », « keyboardHookStruct », « WH_KEYBOARD_LL », « WM_KEYDOWN » et « WM_KEYUP » sont des réécritures de variables ou fonctions définies dans les API Windows. Les attributs personnalisés sont :

- « hhook » : attribut qui contiendra le hook clavier
- « correspondance » : la liste qui contiendra l'ensemble des touches braille enfoncées à un instant donné
- « HookedKeys » : la liste de toutes les touches qui sont écoutées. On y retrouve l'ensemble des touches braille (F, D, S, J, K, L, Q, M) et N

La classe contient aussi diverses méthodes :

- « hook » : cette fonction génère un hook permettant d'écouter les événements claviers qui sont générés. Ces événements sont alors bloqués pour permettre l'utilisation du clavier en « mode braille »
- « unhook » permet de supprimer le hook généré précédemment, arrêtant ainsi l'écoute des événements et libérant le clavier du « mode braille »
- « hookProc » est la fonction principale, c'est elle qui est appelée à chaque événement. Elle va se charger de la récupération des événements et de la transmission ou non de ceux-ci. Elle se déroule comme suit :
 - o Tout d'abord, si l'évènement correspond au relâchement de la barre d'espace, elle va demander la génération d'un caractère via la méthode « getCorrespondance »
 - o Sinon, si l'évènement détecté est la touche « entrée » ou un caractère que l'application a généré, on transmet l'évènement, permettant ainsi d'afficher le caractère, de valider un champ ou de créer une nouvelle ligne
 - o Enfin, si on n'est dans aucun des cas précédent et que la touche fait partie de celles que l'on écoute (définies dans « HookedKeys »), on va soit simuler un retour arrière (pour un appui sur « N ») soit ajouter ou retirer la touche dans la liste « correspondance »
- « getCorrespondance » se charge de traduire les touches braille en caractère. Pour cela on a défini un code : chaque touche braille vaut 2^x où x est son numéro dans les points braille. Ainsi, on fait la somme des valeurs des points contenus dans « correspondance ». On recherche ensuite dans le dictionnaire de correspondance si cette valeur existe et si c'est le cas on génère le caractère correspondant.

BrailleForm

Classe définissant l'interface graphique. Affiche un champ de texte pour permettre de tester aisément le fonctionnement de l'application et une « checkbox » pour activer ou désactiver le mode braille à volonté.

braille

Classe statique constituée d'un dictionnaire contenant la liste des correspondances entre les caractères et leur valeur (telle que calculée dans la méthode « `getCorrespondance` » de la classe `hookClavier`).