

TP 1 - Range Slider

Ivan Logre et Charles Bailly - TP originel de Renaud Blanch

Date butoir de remise du TP : 5 octobre 2018 à 8h

Le but de ce TP est d'implémenter, avec Java/SWING, une interface proposant des interactions avancées non-standard. Il faudra pour cela réaliser un widget qui n'est pas fourni par les boîtes à outils classiques de construction d'interface : le Range Slider, qui permet de sélectionner un ensemble de valeurs dans un intervalle. Ce widget sera ensuite utilisé pour reproduire un système historique : le HomeFinder présenté ci-dessous.

1 Contexte : le HomeFinder

Le [HomeFinder](#) est un prototype de recherche datant de 1993 qui permet de localiser et filtrer des annonces immobilières en fonctions de divers critères [1].

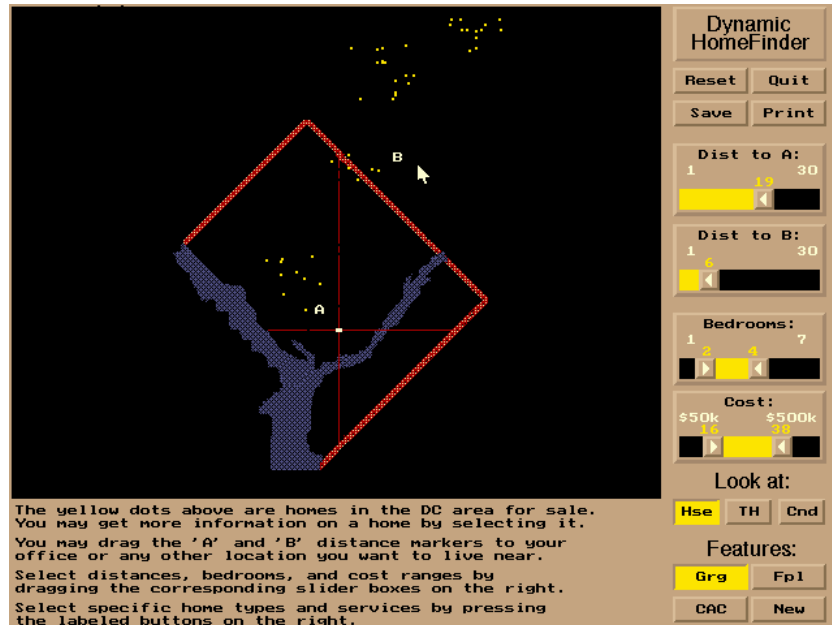


Figure 1: Capture d'écran de l'interface du HomeFinder

Une des propriétés intéressante du système est la possibilité de sélectionner un intervalle de valeur à l'aide d'un widget particulier : le Range Slider. Sur la capture d'écran de HomeFinder ci-dessus, deux de ces widgets sont utilisés pour définir un intervalle : le premier pour le nombre de chambres et le second pour le prix du bien immobilier.

2 Range Slider

Le RangeSlider ressemble à un Slider classique, mais il permet de sélectionner un intervalle plutôt qu'une valeur unique grâce à deux curseurs.



Figure 2: Exemple de Range Slider tiré de HomeFinder

1. En vous inspirant du [BoundedRangeModel](#) qui est le modèle utilisé par le JSlider de Java/SWING, spécifiez l'interface pouvant servir de modèle au RangeSlider.
2. Spécifiez l'interaction du RangeSlider, i.e., ce qu'il se passe lorsque l'on clique ou glisse les diverses parties du widget (curseurs, intervalle sélectionné, parties à l'extérieur de cet intervalle). L'utilisation de machines à états ou de tout autre formalisme à votre convenance pourra s'avérer judicieuse.
3. Réalisez le RangeSlider en Java/SWING. Le code du [JSlider](#) et de son [SliderUI](#) peut vous inspirer. Dans un premier temps, ne vous souciez pas du pluggable look-and-feel.
4. (Bonus) Améliorez votre réalisation en séparant maintenant son rendu au sein d'un composant RangeSliderUI. La description de [comment Java/SWING utilise le modèle MVC](#) pour ses composants pourra vous être utile.

3 Dynamic queries

1. Créez une classe Home qui modélise un bien immobilier (position géographique, nombre de pièces, valeur du bien). Créez un programme qui affiche sur une carte de biens immobiliers pris dans une liste initialisée avec des instances de Home tirées aléatoirement au lancement de votre programme.
2. Ajoutez à cette carte la possibilité de filtrer dynamiquement les biens affichés en ne retenant qu'un intervalle sur le nombre de pièces et/ou la valeur du bien. Cette sélection se fera à l'aide d'instance de votre RangeSlider.

3. (Bonus) Ajoutez la possibilité pour l'utilisateur de positionner à la souris des points A et B sur la carte, et de sélectionner des distances maximales à ces points pour filtrer les biens affichés

References

- [1] Christopher Williamson and Ben Shneiderman. The dynamic homefinder: Evaluating dynamic queries in a real-estate information exploration system. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '92, pages 338–346, New York, NY, USA, 1992. ACM.