Requis et pistes pour les futures boîtes à outils d'interaction graphiques

Alexandre Demeure, Sylvie Rouillard, François Bérard, Gaelle Calvary

Laboratoire CLIPS-IMAG, équipe IIHM
385, rue de la bibliothèque – BP 53
38041 Grenoble Cedex 9, France
{alexandre.demeure ; sylvie.rouillard ; françois.berard ; gaelle.calvary}@imag.fr

RESUME
Cet article vise à identifier les requis fonctionnels des futures boîtes à outils d'interaction graphiques (BAOIG). Il propose des pistes de réflexion pour l'implémentation et notamment une séparation fonctionnelle des préoccupations pour la distribution.

MOTS CLES : Interfaces Homme-Machine, graphique, graphe de scène, distribution, répartition, multi-écran, multi-entrées.

ABSTRACT
This paper aims to identify functional requirements that graphical toolkits should support. It proposes some ideas for implementation and a functional separation of concerns for distributed user-interfaces.

CATEGORIES AND SUBJECT DESCRIPTORS: H.5.2 [GUI].

GENERAL TERMS: Design

KEYWORDS : graphical user interfaces, scene graph, distribution, multi-screen, multi-input.

INTRODUCTION
Au début des années 1980, les plates-formes d'interaction se sont stabilisées sous la forme clavier-souris-écran. Cette stabilité, qui a perduré jusqu'au milieu des années 1990, a permis la mise au point de BAOIG toujours plus adaptées. Les choses ont cependant commencé à sérieusement changer vers le milieu des années 1990 avec l'arrivée de nouvelles plates-formes d'interaction, tels les PDA, les grandes surfaces tactiles et les vidéo projecteurs. Parallèlement, de nouvelles cartes graphiques puissantes, rendant possible le rendu 3D temps réel de qualité à haute résolution d'affichage, sont devenues accessibles à tous. Ces changements militent pour de nouvelles BAOIG.

Dans cet article, nous commencerons par expliciter les besoins par rapport aux nouveaux systèmes développés. Ces besoins sont ceux du concepteur et découlent de ceux des utilisateurs. Ils nous conduisent à esquisser des pistes de solution pour l'implémentation des futures BAOIG.

LES BESOINS
Plusieurs systèmes interactifs ont déjà été développés pour tirer parti de l'environnement informatique moderne. Ils ont posé de nouveaux besoins comme l'orientation arbitraire de tout ou partie de l'interface, l'utilisation de la translucidité ou encore la gestion de plusieurs flux d'entrées, que ce soient des pointeurs ou des dispositifs de saisies [11]. De tels systèmes n'auraient pas pu être développés il y a quelques années, simplement parce que le matériel alors disponible n'aurait pas pu les faire fonctionner à une vitesse acceptable pour l'utilisateur.

Nous décrivons ici les besoins courants pour ces nouveaux systèmes interactifs.

Combinaison d'images
La combinaison d'image est notamment utilisée dans les tool-glasses [2] par le biais de la transparence et dans les travaux de Baudisch [1] par le biais du multi-blending, qui consiste in fine en un traitement d'image en fonction de deux textures. Du point de vue de l'utilisateur, la combinaison d'images trouve sa justification théorique sur deux points. Premièrement, malgré la disponibilité d'écrans de plus en plus larges, la surface d'affichage n'est pas infinie et la combinaison d'images est une solution pour augmenter artificiellement cette surface. Deuxièmement, l'attention de l'utilisateur ne se porte que sur une petite partie de la surface d'affichage. En combinant les images, on permet à l'utilisateur d'avoir plusieurs informations sur la même zone, ce qui permet de réduire le temps passé à chercher une information à l'écran. Ce principe est notamment employé dans des jeux où la carte du terrain est superposée à la vue subjective de l'avatar.

Transformation arbitraire

Toutes les transformations géométriques sont envisageables, mais celles utilisées se limitent en général au cas de matrices 4x4 car ce sont les seules à être accélérées par les cartes graphiques actuelles. On les utilise pour modéliser une transformation dans un espace de coordonnées homogènes modélisant un espace 3D. Parmi les transformations ainsi modélisables, les plus courantes sont :

- La translation : Elle est supportée depuis toujours.
- La rotation : Lorsque la surface d'affichage est horizontale, les utilisateurs peuvent être disposés tout autour, les éléments de l'interface doivent donc pouvoir être orientés de façon quelconque [11].
- L'étirement : Les interfaces zoomables semblent adaptées dans le domaine de la navigation dans de grands espaces de données [7].
- La projection perspective : Elle est utile dans les interfaces 3D [10].

Des transformations plus complexes, non représentables par des matrices 4x4 seules, sont envisageable et d'ors et déjà à l'œuvre dans des interfaces comme celle d'OSX. D'un point de vue matérielle, de nouvelles unités de calcul, les vertex shaders, sont disponibles dans les cartes graphiques. Elles permettent une transformation arbitraire des sommets pour peu qu'on la programme.

Multi-entrées

Que ce soit pour réaliser des manipulations bi-manuelles ou interagir à plusieurs, les utilisateurs ont besoin de plusieurs pointeurs et autres entrées (saisie…) [6] [4].

Distribution

L'environnement informatique de l'utilisateur est de plus en plus riche, les surfaces d'affichage de plus en plus nombreuses. Une BAOIG permettant de créer des interfaces distribuables sur plusieurs surfaces permettrait à l'utilisateur de tirer parti de cette diversité [6] [12]. Plusieurs surfaces physiques pourraient alors être vu comme une unique surface logique.

Plasticité

La diversité des environnements informatiques disponibles, en termes de surface d'affichage ou de puissance de calcul, est telle qu'un interacteur adapté à une grande surface ne peut pas l'être à une petite. Ceci peut s'avérer gênant dans le cas où l'interface est distribuée entre des surfaces hétérogènes (ex PC et PDA). C'est l'objet de la plasticité des interfaces [13] qui vise l'adaptation des systèmes interactifs dans le respect de leur utilisabilité. Une liste de choix pourrait ainsi se remodeler en menu déroulant si l'espace qui lui était alloué venait à être trop petit [5] [3].

LES PISTES DE SOLUTION

Pour faire face aux besoins précédemment cités, les concepteurs des systèmes interactifs ont développé des solutions. Toutefois aucune ne couvre tous ces besoins. Nous nous proposons ici, au vue des expériences acquises, d'émettre des pistes de solution, pour certaines étayées par la pratique.

Combinaison d'images

De nets progrès ont été faits en combinaison rapide d'images. Le sous problème du calcul rapide de la translucidité est déjà résolu depuis quelques années avec l'incorporation au niveau de la carte graphique de mécanismes dédiés (alpha-blending). Les évolutions récentes des cartes laissent supposer que les pixels-shaders résoudront le problème en général [1]. Les pixels shaders ne sont rien d'autre que des unités programmables de la carte graphique dédiée au calcul sur les pixels. Mais en déportant ces calculs du CPU vers la carte, le gain de performance est très sensible. Reste que cette technologie est encore jeune et en perpétuelle évolution, l'apparition de standards comme OpenGL 2 devrait répandre leur utilisation.

Transformation arbitraire

Tant qu'on reste dans le domaine des matrices 4x4 modélisant des transformations homogènes, les choses sont simples, principalement parce que ces transformations sont codées au niveau de la carte, mais aussi parce qu'on peut les composer. La solution la plus commune est de s'orienter vers une bibliothèque graphique de bas niveau tirant parti des cartes graphiques modernes. Pour des questions de portabilité entre plates-formes, le choix porte en général sur OpenGL.

Quand on s'intéresse aux transformations géométriques, on en arrive rapidement à vouloir les combiner, les faire hériter. C'est pour répondre à ce problème que les graphes de scène furent développés. Initialement, ils furent utilisés pour modéliser des mondes 3D. Seules les BAOIG récentes utilisent cette structure [8] [14]. Outre les transformations géométriques, les graphes de scène offrent un moyen élégant de faire de l'héritage de propriétés (couleur par exemple). En poussant ce raisonnement, Ubit [14] met en œuvre la notion de contexte généralisé. Cette notion paraît une évolution logique et devrait être incluse dans les futures BAOIG basées sur les graphes de scène. Notons enfin que le graphe de scène permet très facilement de réaliser des vues multiples : cela revient à donner plusieurs pères à un nœud.

Multi-entrées

Nous avons vu qu'il devenait nécessaire de pouvoir gérer plusieurs pointeurs à la fois, voire plusieurs entrées (claviers…). Les futures BAOIG ne devront plus prendre pour hypothèse que les pointeurs sont de type souris. En effet, des systèmes déjà existants montrent qu'une coordonnée dans un espace 2D n'est plus suffisante (souris 3D, doigt avec l'orientation etc.). Nous pensons donc qu'il est important que la description des pointeurs inclue un meta-descripteur permettant de coder d'autres

informations (orientation d'un doigt…). A charge ensuite aux applications qui y seraient sensibles de décoder ce meta-descripteur pour en déduire les informations voulues.

La distribution

Plusieurs projets traitent de la distribution [6] [9] [12]. Il s'agit idéalement de la mettre en œuvre sans surcoût pour le programmeur et l'utilisateur. D'un point de vue fonctionnel, la répartition pose cinq problèmes :

- α : Découverte des ressources d'interaction, dispositifs de saisie, de pointage, surfaces ou espaces 3D. d'affichage, éventuellement position des utilisateurs.
- β : Mise en relation des espaces logiques et physiques. L'espace logique est un modèle de l'espace dans lequel les interacteurs prennent place, il s'agit classiquement d'un plan mais ce pourrait être un espace 3D. L'espace physique est un modèle de l'espace qui nous entoure, généralement 3D. L'utilisation typique va consister à plaquer des portions d'espaces logiques sur des portions d'espaces physiques (écrans, murs, etc.).
- χ : Expression des relations entre utilisateurs, dispositifs et surfaces/espaces. Il s'agit d'exprimer qu'un dispositif correspond à un utilisateur donné ou qu'une surface est disposée à gauche d'une autre.
- δ : Répartition des éléments présents dans l'espace logique sur l'espace physique. Il s'agira typiquement de distribuer un interacteur sur plusieurs écrans.
- ε : Outils graphiques, offrant primitives et interacteurs, ceux qui peupleront l'espace logique.

Les fonctionnalités précédemment décrites ont des dépendances que nous explicitons dans la figure 1. Une dépendance est symbolisée par un côté commun, une flèche indique le sens de la dépendance.

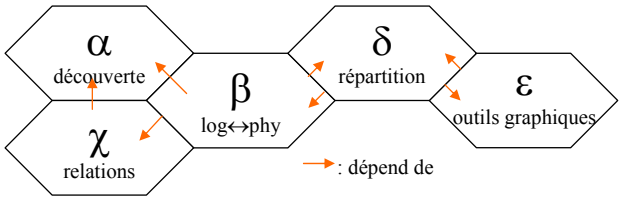

Figure 1: Décomposition fonctionnelle d'une boîte à outil assurant la distribution.

Observons maintenant les influences entre fonctionnalités.

- α→(β et χ) : β et χ dépendent de α qui les informe des ressources d'interaction détectées. β en a besoin pour connaître l'espace. χ en a besoin pour calculer les relations entre les entités physiques, par exemple le fait qu'une souris soit proche d'un clavier.
- χ→β : La position des surfaces entre elles dans l'espace physique peut permettre de calculer des mises en correspondance entre les espaces logique et physique. Par exemple, si un écran E2 se trouve à droite d'un écran E1 affichant l'espace logique entre 0 et 1024, on pourra en déduire que l'écran E2 affiche l'espace logique entre 1024 et 2048.
- β↔δ : β émet les vues sur l'espace logique et les surfaces physiques auxquels elles correspondent, permettant à δ d'effectuer les rendus. Le mécanisme de répartition renseigne β sur les zones occupées de l'espace logique. Par exemple qu'une zone contenant des informations n'est pas affichée.
- δ↔ε : ε émet, pour chaque plate-forme, les informations relatives aux interacteurs. Le mécanisme de répartition les répercute sur les autres plates-formes.

Les dépendances entre les modules fonctionnels n'ont pas toutes la même force. Nous voyons deux blocs se dessiner. D'un coté, nous trouvons le mécanisme de répartition qui est très dépendant des outils graphiques. Ils forment ensemble la couche basse de la distribution. De l'autre, nous trouvons la découverte des entités, le calcul des relations et la mise en correspondance entre les espace physique et logique. Ensemble, ils forment la partie haute, « intelligente » de la distribution. Il nous paraît indispensable de bien séparer au moins ces deux groupes. En effet, il est souhaitable de pouvoir factoriser les algorithmes de mise en correspondance. Il nous semble illusoire de penser qu'une seule BAOIG sera adaptée pour tout. Soit par les technologies mises en oeuvres, soit par leur logique intrinsèque (basée ou non sur des graphes de scène, ayant des interacteurs ou des primitives graphiques seulement, etc.), des BOAG très différentes verront le jour, ciblant chacun un problème précis.

En ce qui concerne la répartition (δ), en supposant qu'on adopte la structure de graphe de scène, nous proposons que le partage se fasse au niveau de sous graphes (figure 2). Pour l'ordinateur 1, les nœuds partagés par l'ordinateur 2 seront répliqués dans le sous graphe ayant pour racine le nœud pivot. L'ordinateur 2 fait de même de telle sorte que les sous graphes partagés par 1 et 2 soient les mêmes. Chacun accède à son sous graphe partagé par un nœud point de vue qui détermine la portion d'espace visualisée.

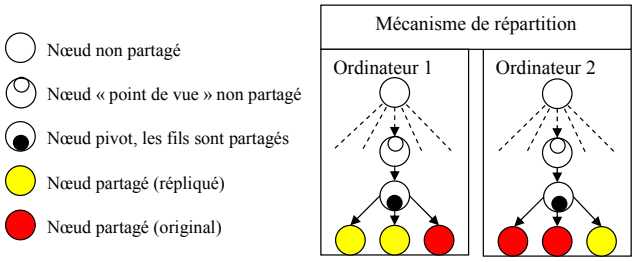

Figure 2: Partage de sous graphes entre ordinateurs.

La distribution de l'interface entre plusieurs surfaces pose le problème de son adéquation par rapport aux différentes plates-formes, c'est l'objet de la plasticité.

La plasticité

Bien que la plasticité ne se limite pas à des problèmes de remodelage, il nous paraît important que les futures BAOIG prennent en compte ce problème et facilitent la création d'interacteurs doués de polymorphisme [3] [5] ou permettent de basculer facilement entre plusieurs interacteurs (d'une liste à choix à un ensemble de radio-bouton par exemple). C'est le cas d'Ubit qui offre des mécanismes de base [14]. La plasticité est un requis nécessaire pour deux raisons :

- Les surfaces d'interaction seront toujours hétérogènes. Des PDA aux écrans de grande taille, tous trouveront leur place dans nos activités. Réaliser des interacteurs capables de s'adapter à ces surfaces semble indispensable.
- Même en considérant des surfaces homogènes, le changement de forme de l'interacteur peut être souhaitable. Par exemple, sur une surface où l'interaction se fait à plusieurs doigts, l'interacteur sera probablement présenté différemment que pour une interaction au stylet.

Dans le cas de l'utilisation d'un graphe de scène, la plasticité pourra porter sur les nœuds (interacteurs, primitives de dessins…) ou sur la structure du graphe lui-même. Ubit [14] en fournit un exemple en permettant de parcourir le graphe en fonction des informations se trouvant dans le contexte généralisé. Des branches n'étant par exemple pas accessibles pour des ordinateurs de type PDA ou, au contraire, leur étant réservées.

CONCLUSION

Nous observons un renouveau dans le domaine du développement des BAOIG, imposé par de nouveaux types d'applications. Nous avons posé un ensemble de requis de ce qu'il semble indispensable d'implémenter. Pour les problèmes de l'orientation arbitraire, de la composition d'images et de la gestion de plusieurs entrées, des solutions existent déjà. Il s'agit maintenant de les inclure dans les BAOIG. Pour les problèmes de la distribution et de la plasticité, aucune solution ne s'est encore imposée et de nombreux travaux restent à mener. Nous avons proposé une décomposition fonctionnelle dans le but d'aider à résoudre le problème de la distribution. Le problème de la plasticité fait pour sa part l'objet de nombreux travaux dont il conviendra de faciliter l'intégration dans les BAOIG.

BIBLIOGRAPHIE

1. Baudisch, P., Gutwin, C. *Multiblending: displaying overlapping windows simultaneously without the drawbacks of alpha blending.* In Proceeding of CHI 2004 (April 24-29, 2004, Vienna, Austria), ACM Press, New-York USA, 2004, pp. 367-374.
2. Bier, E.A., Stone, M., Pier, K., Buxton, W., DeRose, T. *Toolglass and magic lenses: the see-through interface.* In Proceeding of SIGGRAPH'93 (August 1993), ACM Press, New-York USA, 1993 pp.73-80.
3. Daassi O., Calvary, G., Coutaz J., Demeure A. *Comet : Une nouvelle génération de "widget" pour la Plasticité des Interfaces.* In IHM 2003 (November 2003), ACM Press., pp. 64-71.
4. Pierre Dragicevic, P. , Fekete J., *Input Device Selection and Interaction Configuration with ICON.* actes de la Conférence Internationale IHM-HCI 2001, , Lille, France, Springer Verlag, pp. 543-448.
5. Grolaux, D., Van Roy, P., Vanderdonckt, J. *QTk: An Integrated Model-Based Approach to Designing Executable User Interfaces*. 8th Int. Workshop DSVIS'2001 (Glasgow, June 13-15, 2001), Dept. of Comp. Sci., Univ. of Glasgow, Scotland, 2001, pp. 77-91.
6. Lachenal, C., Coutaz, J. *A Reference Framework for Multi-surface Interaction*. HCII'2003 (Human computer Interaction International) Crete.,
7. Photomesa, http://www.cs.umd.edu/hcil/photomesa/.
8. Piccolo, http://www.cs.umd.edu/hcil/piccolo/.
9. Project Gloss, http://iihm.imag.fr/projects/Gloss
10. Project Looking Glass, http://wwws.sun.com/software/looking_glass/index.html.
11. Shen, C., Vernier, F.D., Forlines, C., Ringel, M. *DiamondSpin: An Extensible Toolkit for Around-the-Table Interaction*. In Proceeding of CHI'04 (April 24-29, 2004, Vienna, Austria), ACM Press, New-York USA, 2004, pp. 167-174.
12. Sousa, J., Garlan, D. *Aura : an Architectural Framework for User Mobility in Ubiquitous Computing Environments*. IEEE-IFIP Conf. on Software Architecture, Montreal, 2002.
13. Thevenin, D., Coutaz, J. *Plasticity of User Interfaces: Framework and Research Agenda*. In Proceeding of Interact'99, Edinburgh, A. Sasse & C. Johnson Eds, IFIP IOS Press Publ., pp.110-117.
14. Ubit,http://www.infres.enst.fr/~elc/ubit/