

TD6 : UML

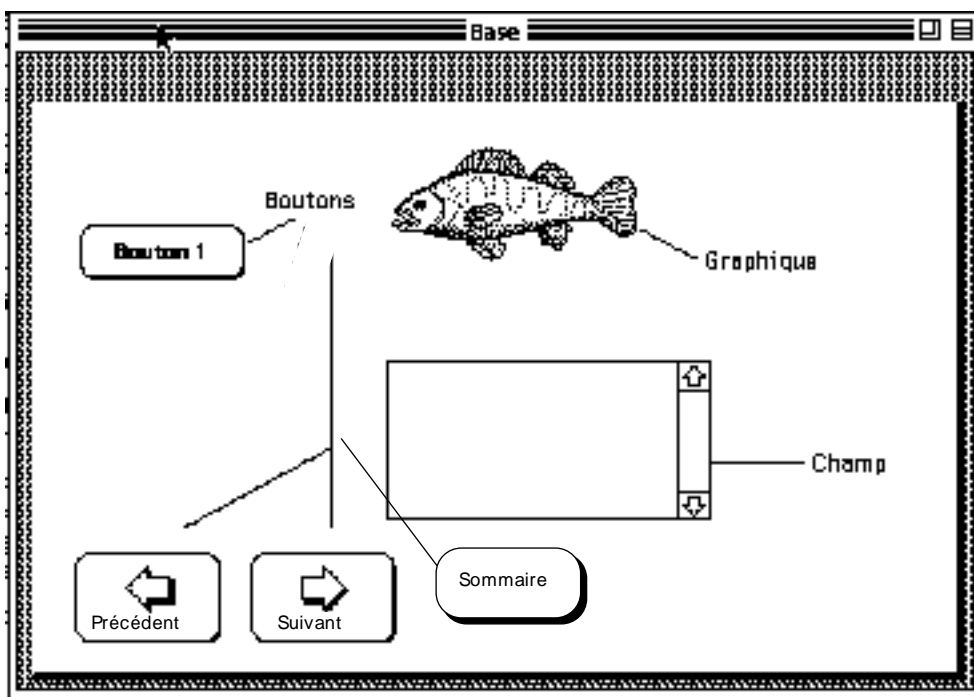
Diagrammes de classes / diagrammes de séquences : Navigation dans une pile Hypercard.

Apparue dans les années 1980 avec les premiers Macintosh, l'application Hypercard était très en avance sur son époque car elle contenait déjà les concepts objet ainsi que la navigation hypertexte. Les documents manipulés par l'application sont appelés « **piles** » et sont composés de « **cartes** » à l'image du fichier physique de fiches bristol.

Toute pile comporte une carte sommaire particulière qui reste toujours sur le dessus de la pile, l'ordre des autres pouvant être changé en particulier par des fonction de tri suivant différentes clés. La notion d'ordre des cartes est en effet conservée, par conséquent toute carte a une suivante (sauf évidemment la dernière), et une précédente (sauf évidemment la première qui est toujours la carte sommaire).

La carte sommaire peut être utilisée (et l'est en général) pour faire une « hyper table des matières » permettant d'accéder directement à certaines cartes « tête de chapitre ».

Les cartes contiennent des « widgets » (Windows objects) qui peuvent être des boutons, des champs texte (pouvant servir pour saisir une entrée ou au contraire afficher un résultat) et des graphiques bitmap ou vectoriel (voir copie d'écran d'un Mac SE l'époque). Dans la réalité un widget devant figurer dans toutes les cartes était placé dans un « fond de carte » équivalent de ce que l'on appelle « masque des diapositives » sous Powerpoint.



Les widgets possèdent un script écrit dans un langage particulier (Hypertalk, véritable petit langage objet). Les méthodes du script sont déroulées lorsque l'objet reçoit un message. L'un de ces messages « Onclick() » est envoyé à l'objet par l'utilisateur lorsque celui-ci clique sur l'objet. D'autres messages provenant de l'utilisateur sont par exemple envoyés à un champ texte lorsqu'une donnée y est saisie etc. Le script d'un objet peut effectuer des calculs, utiliser tous les outils de l'application (dont les outils de dessin pour créer des animations), afficher des résultats, envoyer des messages à d'autres objets etc.

On ne s'intéressera pas dans cet exercice aux fonctions des graphiques et des champs texte, mais seulement aux boutons. Grâce à leur script les boutons peuvent en réalité effectuer des fonctions

très variées (recherche, tri, sélection, voire calculs ou animations graphiques) mais on s'intéresse ici à celle qui permet une navigation dans la pile exactement comme le font les liens hypertexte des sites web. Les boutons dont la méthode ayant cette fonction seront dénommés dans la suite « boutons lien ». Parmi les boutons lien il en existe trois particuliers (qui sont d'ailleurs prédéfinis dans une bibliothèque utilisable par les développeurs de piles Hypercard), à savoir les boutons « précédent », « suivant » et « sommaire » permettant respectivement la navigation vers la carte précédente, suivante et la carte sommaire (on peut les voir sur la figure, un concept tout à fait semblable peut être trouvé sur bien des sites web). Dans notre cas on ajoutera à ces trois types de bouton lien un quatrième type que l'on appellera par exemple « bouton chapitre » : placé dans l'hyper table des matières de la carte sommaire, il permettent une navigation rapide vers les premières cartes des chapitres.

On fera les hypothèses suivantes (dont certaines sont plus compliquées que n'est la réalité, ce afin d'illustrer le plus possible des alternatives de modélisation offertes par l'UML) :

- L'affichage d'une carte est réalisée en lui envoyant un message afficher(). La carte s'adresse alors à tous ses widgets en leur envoyant un message afficher().
- Tout bouton possède un attribut nommé « intitulé » qui contient une chaîne de caractère (éventuellement vide). Les boutons « précédent », « suivant » et « sommaire » sont identifiés par cet attribut contenant ces chaînes de caractères.
- D'autres attributs peuvent être ajoutés : en particulier les boutons chapitres possèdent un attribut supplémentaire appelé « numéro » donnant le numéro du chapitre.
- Il faut tenir compte de la contrainte suivante (non conforme à la réalité, ajoutée pour complexifier un peu le modèle). La responsabilité de la navigation est située au niveau des cartes (ce sont elles qui connaissent leur précédente et suivante, ainsi pour la carte sommaire que les cartes début de chapitre : cela devra être pris en compte dans le diagramme de classes), la fonction des boutons étant très limitée. Les boutons sont de simples objets réactifs dont la seule fonction est d'informer la carte à laquelle ils appartiennent qu'ils viennent d'être cliqués. De plus la carte ne connaît alors que l'identité du bouton, et doit se débrouiller pour récupérer les attributs dont elle a besoin.

Question : On demande de construire un modèle cohérent composé d'un diagramme de classes et d'un ou plusieurs diagrammes de séquences illustrant un ou plusieurs scénarios significatifs de navigation (significatif voulant dire que l'on y voit au moins un exemple de chaque type de navigation).