

DURAND Remi
LEDRU Mathieu
LOUIS CLEMENT Justin

Visualisation

Acquisition des données

Afin de déterminer quel sera la façon dont nous récupérerons les données, nous avons fait un QOC pour nous aider dans nos décisions.

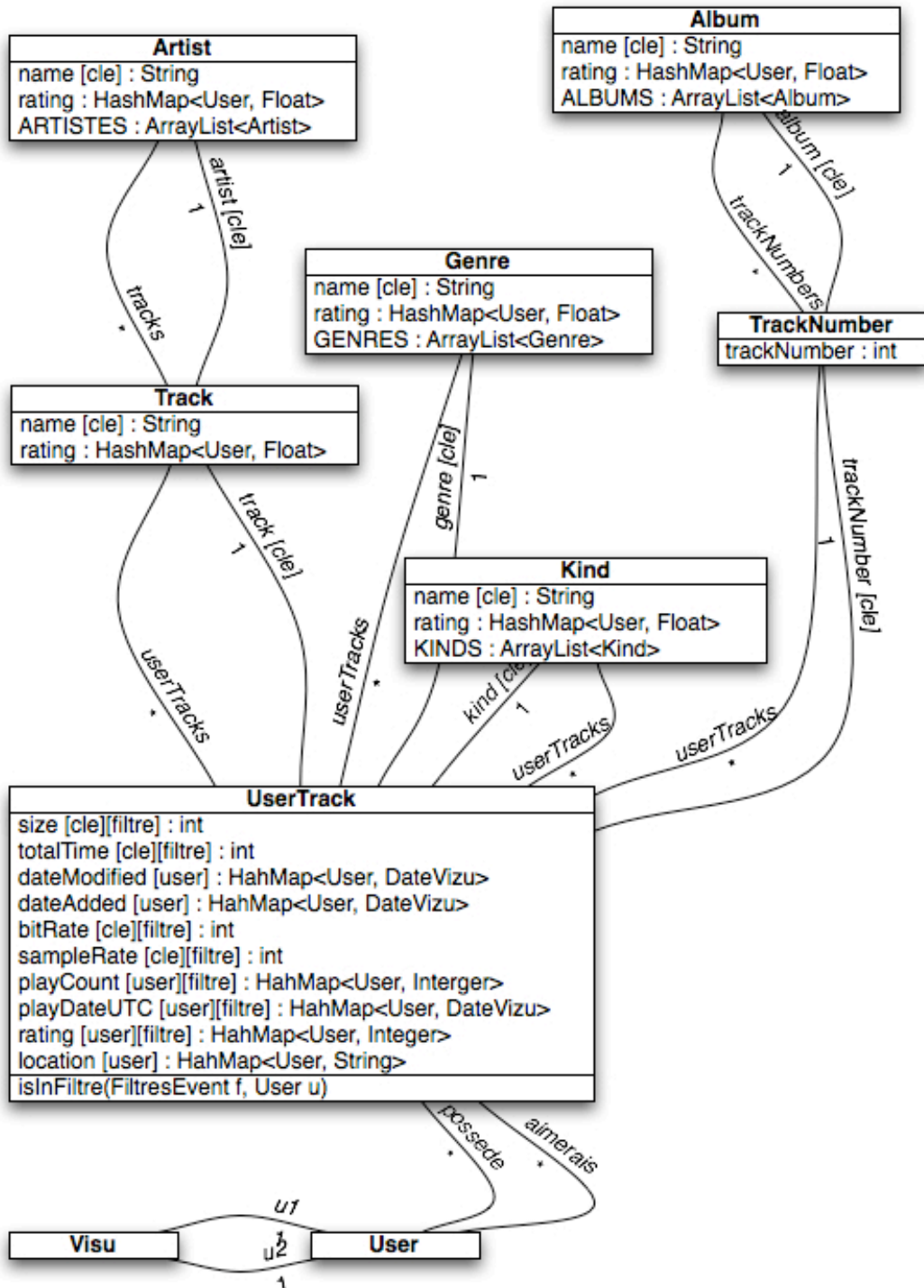
QOC Données								
Question	Options						Critères	poids des critères
	site internet	texte descriptif bitorrent	itunes	podcast	base de données	médiathèque		
Quelles données?	80	20	60	60	90	100	données fiables	10
	50	10	100	50	100	75	données triées / complètes	10
	10	75	100	50	100	10	données faciles à extraire	20
	5	10	100	100	50	0	données par utilisateur	30
	90	75	50	25	75	100	données exotiques/volumineuses	10
	95	60	50	100	5	25	données faciles à trouver numériquement	20
					Total des critères (%)			100
	Options pondérées							
	8	2	6	6	9	10		
	5	1	10	5	10	7,5		
	2	15	20	10	20	2		
	1,5	3	30	30	15	0		
	9	7,5	5	2,5	7,5	10		
	19	12	10	20	1	5		
Total (%)	44,5	40,5	81	73,5	62,5	34,5		

Nous avons alors choisit d'exporter la bibliothèque utilisateur du logiciel iTunes pour récupérer les données brutes.

Stockage des données

Les données fournies contiennent différentes informations sur chaque morceau de musique : le rating du morceau, l'artiste, l'album, le genre, le bit rate, la dernière date d'ajout, la dernière date d'écoute, le nombre de fois que le morceau a été écouté, la localisation du fichier sur le disque, etc, etc ...

Voici le diagramme UML des données :



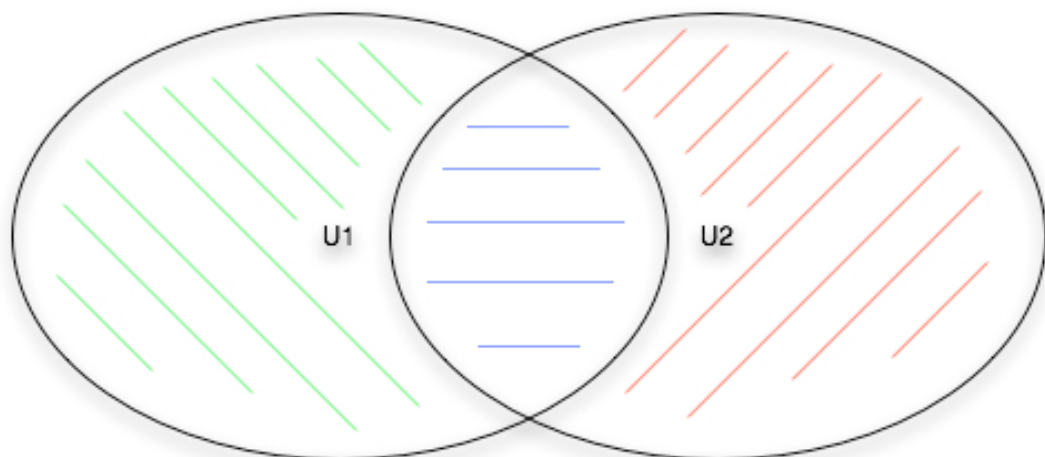
Le programme se charge alors de récupérer toutes les données en mémoire en parcourant le fichier xml fourni par iTunes. Puis, chaque morceau est classé dans une instance unique pour chaque nouvelle information parsée.

Par exemple, si un artiste est parsé plusieurs fois dans le fichier xml, alors il ne sera stocké qu'une seule fois en mémoire. Et ceci est la même chose pour le genre, l'artiste, et l'instance d'un morceau (la classe UserTrack) contenant toutes les autres informations relatives au morceau.

Chaque type d'information ajoutée en mémoire est stocké soit en temps que clef, soit en temps que filtre, soit en temps qu'utilisateur. Les informations de type clef permettent de supprimer les doublons pour les morceaux tels que les artistes, types, genres, ... Les informations de filtres seront utilisés pour filtrer les morceaux affichés à l'écran à l'aide des bi-slider qui permettront d'affiner ou d'élargir la sélection. Les informations utilisateurs sont utiles pour déterminer quel est l'utilisateur qui possède les informations d'une instance d'un morceau.

Afin d'améliorer les performances de recherche d'instances de morceau ou de filtrage, des listes ont été ajoutées aux données pour accéder directement aux informations. Par exemple, en sélectionnant un artiste, nous accédons directement à tous ces morceaux en parcourant la liste. De même que l'on connaît tous les morceaux d'un même genre ou d'un même type de format.

La signification des instances possède et aimerai pour les deux utilisateurs est la suivante :



U1 possède = ——— + ———

U1 aimerai = ———

U2 possède = ——— + ———

U2 aimerai = ———

Création de l'espace et intégration

Interface Disque

L'interface affichera sous une forme hiérarchique les morceaux afin de pouvoir regrouper des morceaux communs et de pouvoir faire des tas. Les morceaux pris séparément seront alors mis en commun et l'œil préférera voir les tas de morceaux plutôt qu'un amas de morceau répartis sur l'interface. De plus le regroupement se fera sur plusieurs critères caractérisant les morceaux, que cela soit au niveau du genre, de la date d'ajout, du nombre de fois où le morceau à été joué, de l'artiste, etc... Les différents types de regroupement devront permettre à l'utilisateur de faire et de défaire des tas de morceaux, car il n'existe pas vraiment de hiérarchie caractérisant un morceau de musique. En effet, on peut prendre l'exemple d'un artiste qui a fait deux morceaux de musique qui sont d'un genre différent, ce qui veut dire que le regroupement par artiste n'implique pas le regroupement par genre. Réciproquement, deux morceaux de musique du même genre n'ont pas forcément le même artiste. C'est la raison pour laquelle l'interface devra pouvoir grouper ainsi que dégrouper un tas de morceau par rapport à un critère donné.

Nous nous sommes encore une fois aidé d'un COQ pour nous guider dans nos décisions.

QOC Données						
Question	Options				Critères	poids des critères
Quelle représentation	Arbre Simple	TreeMaps	Radial Space Filling	Triangle Tesslation		
	75	35	60	75	facile a mettre en œuvre	50
	0	0	100	0	forme d'un CD	5
	35	50	50	25	lisible	25
	20	80	60	70	rempli bien l'espace	20
				Total des critères (%)		100
	Options ponderées					
	37,5	17,5	30	37,5		
	0	0	5	0		
	8,75	12,5	12,5	6,25		
	4	16	12	14		
Total (%)	50,25	46	59,5	57,75		

Nous avons finalement opté pour le Radial Space Fiffing.

Interface Sphère

L'interface affichera les morceaux par ordre de préférence. Le but étant de pouvoir montrer à l'utilisateur final les morceaux de musique qu'il pourrait emprunter à son ami et qu'il aurait de bonnes chances d'apprécier. Il a fallu une représentation ordonnée par les notes avec la possibilité de répartir ces musiques sur l'écran en fonction d'un autre critère. Pour choisir le type de visualisation, nous nous sommes là aussi aidé d'un COQ:

Les différentes options étaient les suivantes:

- Le tableau ordonné: les notes dans un axe, l'autre critère dans l'autre axe, avec dans chaque case les musiques qui correspondent aux coordonnées des deux axes. Au niveau point de la vue sur l'espace, on utilise des barres de défilements.

- Le Cercle hyperbolique. Les notes sont symbolisés par la distance au centre, l'autre critère par l'angle. Au niveau du point de vue sur les données, on utilise le système hyperbolique qui permet à une zone de « focus » de se déplacer sur le cercle, zoomant la zone. On peut faire l'analogie avec une nappe que l'on déplace sur une sphère, en vue de dessus.

- Le tapis hyperbolique qui ressemble un peu au tableau à deux dimensions, mais en changeant le point de vue sur l'espace, avec un déplacement sans barre de défilements.

QOC données					
Question	Options			Critères	Poids
	Tableau ordonné	Cercle hyperbolique	Nappe Hyperbolique		
Quelle Représentation	90	50	75	Facile à créer	10
	80	60	75	Compréhensible	30
	100	70	80	Lisible	20
	30	90	50	Joli	30
	20	70	65	Utilise bien l'espace	10

Total des critères (%):

Options pondérées			
	9,00	5,00	7,50
	24,00	18,00	22,50
	20,00	14,00	16,00
	9,00	27,00	15,00
	2,00	7,00	6,50
Total (%)	64,00	71,00	67,50

Nous avons finalement opté pour le Cercle hyperbolique.

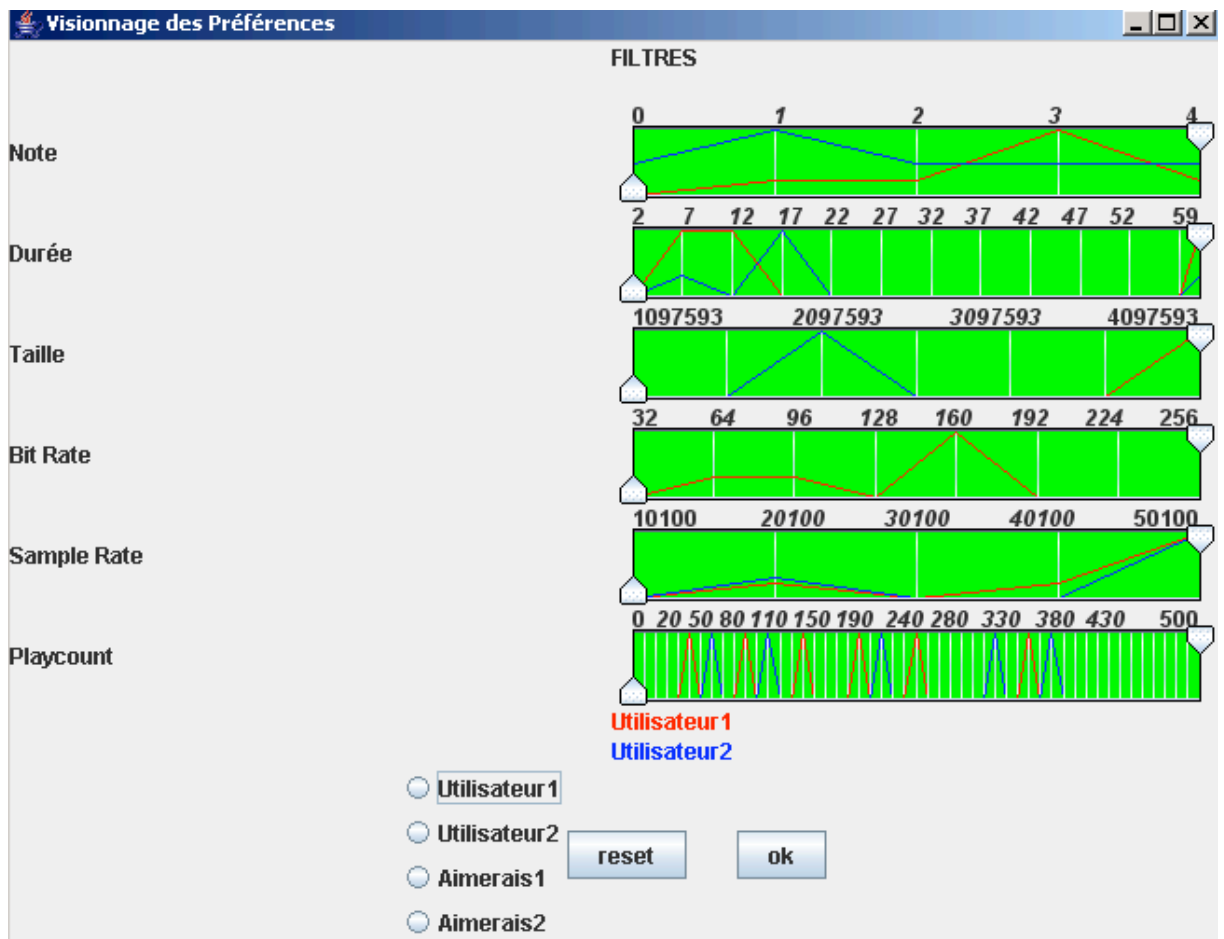
Affichage de l'espace et navigation

Bislider

Nous allons présenter maintenant l'interface permettant de réaliser des filtres grâce à l'utilisation de bisliders. Les bisliders sont des sliders particuliers, permettant de sélectionner non pas une valeur mais un intervalle de valeurs grâce à l'utilisation de deux curseurs pour un même slider : le curseur pour la valeur minimale représenté par une flèche orientée vers le haut, et un curseur pour la valeur maximale représenté par une flèche orientée vers le bas.

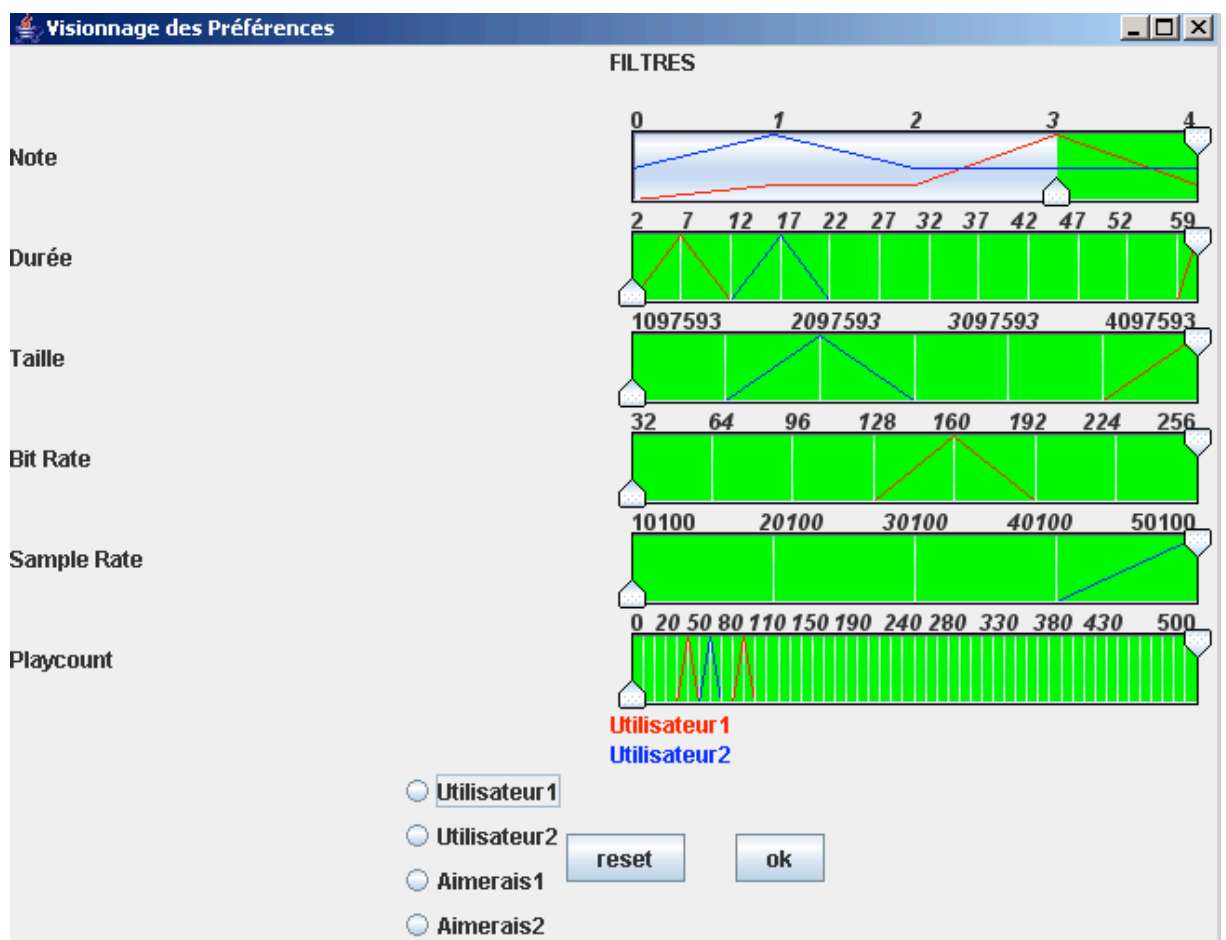
Ainsi, nous avons délimité l'intervalle qui nous permettra de choisir un ensemble de valeurs pour une catégorie donnée. Par exemple si nous souhaitons afficher les musiques dont la note est comprise entre 3 et 4, nous allons déplacer le curseur minimum vers la valeur 3 et le curseur maximum vers la valeur 4.

Voici ci-dessous une image représentant l'interface de filtres avec plusieurs sélections possibles.



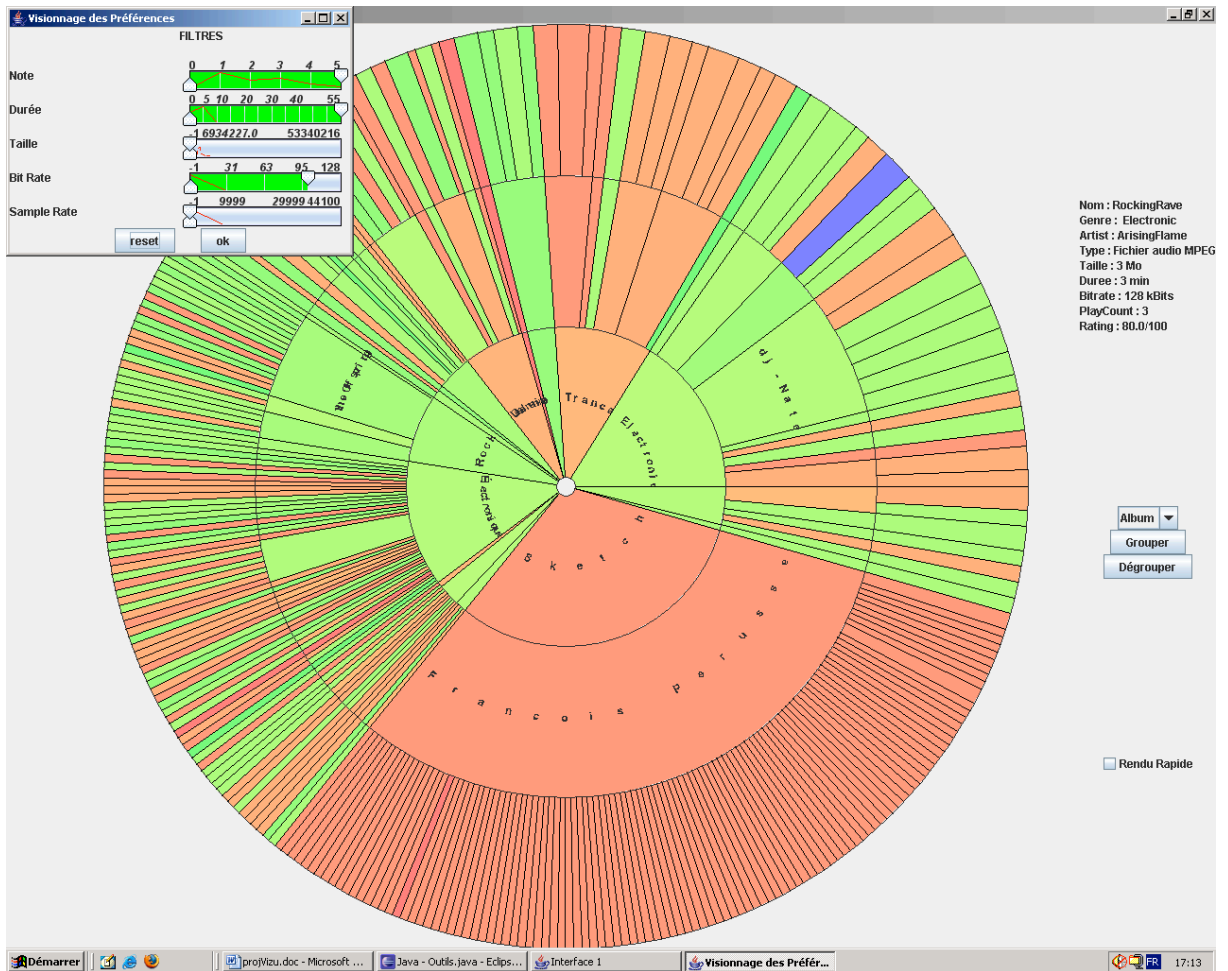
Nous pouvons sélectionner quel utilisateur nous souhaitons afficher. De plus, dans chaque bislider est affiché un histogramme montrant la répartition des utilisateurs pour chaque filtre souhaité ; ainsi, nous avons une idée de la répartition globale des titres suivant un critère donné, ce qui peut nous faciliter lorsqu'on ne souhaite pas sélectionner des intervalles dans lesquels il n'y a aucun fichier musical.

Il est également à noter que la modification d'un intervalle sur un bislider a des répercussions sur d'autres bisliders. En effet, si nous choisissons encore l'intervalle de notes comprises entre 3 et 4, alors les histogrammes dans chacun des bisliders d'en bas seront modifiés, car nous n'afficherons plus que les musiques dont la note est comprise entre 3 et 4, en plus du critère à proprement parler du bislider courant. Voyons le résultat dans cet exemple ci-dessous :



Nous devons remercier particulièrement notre professeur de visualisation : Mr Frédéric Vernier(enseignant-chercheur en informatique, maître de conférence à l'Université Paris-Sud (Paris-Orsay, Paris XI), frederic.vernier@limsi.fr) pour nous avoir permis d'utiliser son code source le plus à jour dans notre projet.

Interface Disque



Sur la première interface, nous avons deux fenêtres.

La première fenêtre, commune aux deux interfaces permet à l'utilisateur de filtrer des morceaux pour afficher plus ou moins globalement des informations qui l'intéressent.

L'interface principale au deuxième plan expose une visualisation de la bibliothèque musicale sous forme de disque (et ressemble aussi à un CD de musique).

Au démarrage du programme, tous les morceaux sont affichés tels quel et on ne peut a priori rien tirer de l'interface, mais juste se faire une idée générale de la bibliothèque grâce aux couleurs affichées. C'est la raison pour laquelle il est possible de regrouper les morceaux en les triant par l'intermédiaire des commandes « grouper » et « dégroupier » données à droite de l'interface. Cette fonctionnalité ajoute des sous-groupes visibles dans les cercles intérieurs du disque et les morceaux placés sur les feuilles en périphéries sont alors bien classés par rapport à leur parent respectif.

Il est possible de grouper et de dégroupier les morceaux selon différents attributs ce qui permet d'observer les données de plusieurs façons. On peut imaginer de regrouper les données pas seulement sur le genre, l'artiste et l'album, mais aussi sur le type de codage, le bitrate, la dernière écoute, le nombre de fois que le morceau a été écouté, etc...

En couleur, nous pouvons voir les morceaux que l'utilisateur aime en vert foncé, et n'aime pas en rouge foncé en passant dans l'ordre par le vert clair, jaunâtre et orange. La case bleue foncée indique que la souris est actuellement sur cette zone. Les Feuilles ont la couleur donnée par le rating du morceau et les parents des feuilles indiquent les moyennes des ratings de leurs enfants.

En espace, nous pouvons aussi constater que le nombre d'un album est d'autant plus grand que l'album possède des morceaux ce qui permet à l'œil de voir quels sont les types de morceaux les plus représentatifs.

Le texte est affiché qu'à partir d'un certain angle, sinon il ne serait pas lisible.

Ainsi, on peut sur ce screenshot dire que l'utilisateur possède une majeure partie de sketches des deux minutes du peuple et que celui-ci aime bien la musique électronique.

Lorsque l'on passe la souris sur le disque, les cases s'agrandissent pour faire un effet de zoom sur la zone que l'utilisateur est en train de pointer afin de mieux distinguer les morceaux.

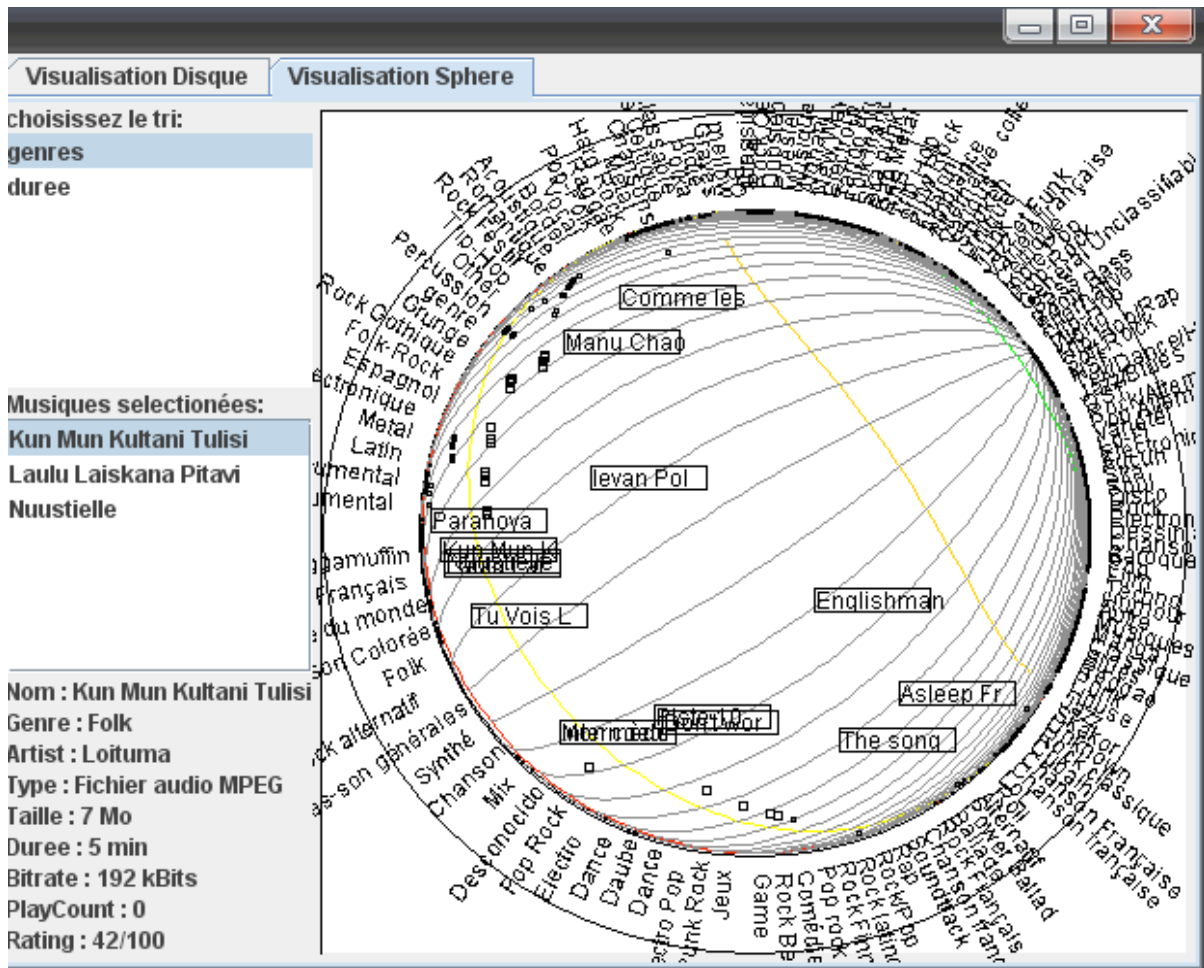
Il est possible de valider la case « rendu rapide » afin de simplifier l'affichage et augmenter la fluidité des animations si le programme venait à ralentir.

Lorsque l'on clique sur une case, alors la description de celle-ci est affichée sur la droite ce qui permet à l'utilisateur de s'informer en détail sur certaines parties de l'interface. Le clic sur une feuille du disque décrit le morceau dans son intégralité.

Interface Sphère

L'interface Sphère nous montre un cercle qui, quand on se déplace dessus, donne l'impression de bouger une sphère. Les meilleurs morceaux se trouvent au milieu de celle-ci.

En premier lieu, la colonne de gauche permet de Visualiser et interagir avec les propriétés propre à cette visualisation. En haut, se situe les différents critères pour calculer l'angle (et donc la position) du morceau. Au milieu et en bas, se situe la partie visualisation des morceaux. Lorsque l'on clique sur une étiquette (ou un amas d'étiquettes), elle(s) se placent avec son nom complet dans la fenêtre du milieu. On peut alors sélectionner avec toute la lisibilité voulu sur le morceau pour afficher ses propriétés en bas.



L'espace de visualisation est une surface équivalente à un disque vinyle. Le trou du centre étant là pour éviter que les morceaux avec une grosse note soient tous superposés. Pour passer De cette espace au point de vue vers l'espace, on effectue un « zoom » en un point. Toutes les données restent affichées sur l'écran mais lorsque l'on s'éloigne du point de zoom, on devient plus petit et on se rapproche de la bordure.

En faisant du drag&drop, on peut déplacer le point de zoom. En faisant rouler la molette de la souris, on change le niveau de zoom. Pour savoir où chaque morceau se situe, on part de sa position sur l'espace (fixe) en x et y. On calcule son angle et sa distance avec le point de zoom (lui aussi avec ses coordonnées sur l'espace de visualisation). Avec cela, on peut en déduire ses coordonnées dans le point de vue sur l'espace en x et y. La même méthode est employée pour construire les traits et les légendes.

La lisibilité est bien moindre lorsque l'on affiche la bibliothèque d'un utilisateur, car les morceaux sont alors tous les mêmes lignes de notes, car iTunes ne peut attribuer que les notes de 0, 20, 40, 60, 80 et 100. De même, lorsque l'on trie par genres et que ceux-ci sont extrêmement nombreux, il est difficile de bien distinguer quelle frange appartient à quel groupe dès que l'on zoome un peu et que l'on est assez éloigné du bord. Pour ce qui est des tris discrets comme celui en fonction de la durée du morceau, il y a souvent quelques morceaux très long qui faussent totalement la plage couverte. Il convient alors de bien sélectionner la plage couverte avec l'outil de sélection des préférences.

Conclusions

Ce projet a été pour nous une opportunité de s'attarder sur cet aspect essentiel des applications: l'interface homme-machine. Nous avons pu voir quels étaient les enjeux et les aboutissants de cette partie.

En ce qui concerne notre projet, Il est a l'heure actuelle limité par les performances lorsque les fichiers xml en entrée atteignent quelques méga-octets. Or le bibliothèques des utilisateurs « normaux » sont en général entre 1 et 10 méga-octets, selon le nombre de musiques qu'il possèdent. Il en résulte un problème de vitesse avec les bases les plus grosses. Un fichier de 1 méga est instantané à analyser, à afficher, un fichier de 5 méga met 1 minute à être importé, et l'affichage est saccadé. Des optimisations sont néanmoins encore possible.

Pour l'Analyse des résultats de notre fonction de calcul des morceaux conseillés, il ne propose sur nos tests jamais de morceaux avec une note supérieure à 4 (80). Cela est due au fait que lorsque une personne aime bien un style de musique, il possède beaucoup de musique dans ce style là, mais ne les note pas toutes avec de fortes notes. En effet, il sera moins regardant sur la qualité et écoutera des morceaux moins bon qu'il notera mal. Au final, utiliser la moyenne des notes pour analyser la préférence d'un utilisateur pour un gêner (et même les auteurs?) n'est pas pertinent. Il faudrait aussi y rajouter une composante qui tient compte du nombre de morceaux possédé pour ce genre. Nous avons donc là aussi des modifications à faire pour que le logiciel est un intérêt plus assuré.