**Simulation de hockey de la LNH**

****

Lundi le 23 janvier 2017

Table des matières

[Description sommaire 2](#_Toc473482576)

[Diagramme de cas d’utilisation 3](#_Toc473482577)

[Prototypes d’interface 3](#_Toc473482578)

[Règles d’utilisation 4](#_Toc473482579)

[Description spécifique 5](#_Toc473482580)

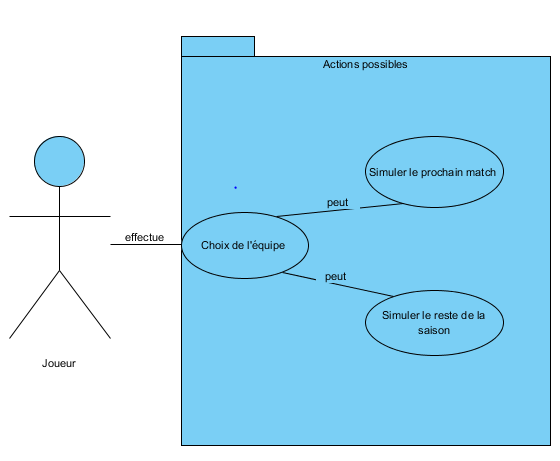
[Nouveauté 6](#_Toc473482581)

[Références utilisées 6](#_Toc473482582)

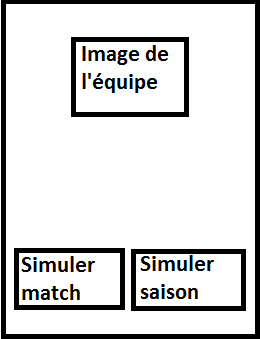
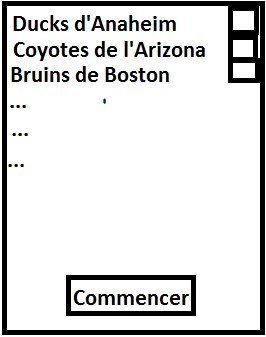
# Description sommaire

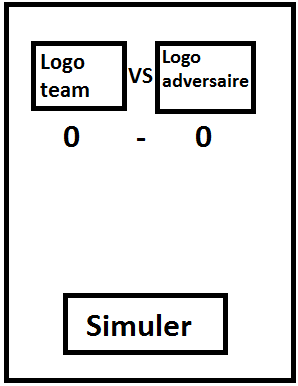
L’application que je vise à développer consiste de façon générale à sélectionner d’abord une équipe parmi les 30 qui forment la Ligue Nationale de Hockey (LNH) dans la première activité qui contiendra que le nom des équipes, le logo correspondant à chacune ainsi qu’un bouton qui permettra d’appeler la prochaine activité. Par la suite, une fois l’équipe sélectionnée, il sera possible de cliquer sur ce bouton qui changera l’interface selon l’équipe qui a été sélectionnée. Dans la deuxième activité, il sera possible de simuler le prochain match à l’horaire et aussi de simuler la saison entière d’un seul coup (il y a 82 matchs dans une saison). Si l’on clique sur le bouton pour simuler un match, une équipe aléatoire autre que celle qui a été sélectionnée va être déterminée au hasard pour représenter l’adversaire du match. L’aspect domicile/extérieur sera également déterminé au hasard. Une fois l’adversaire déterminé, un score sera attribué au match. L’équipe qui a été choisie au départ, au terme d’une saison de 82 matchs, aura une fiche du format VV-DD-DPDP où VV représente le nombre de victoires accumulé au cours de la saison, où DD représente le nombre de défaites et où DPDP représente le nombre de défaites en prolongation. L’équipe choisie, tout au long de la saison, possède un nombre de points qui est déterminé selon la fiche de l’équipe. Une victoire donne 2 points, une défaite en prolongation en donne un seul et une défaite ne donne aucun point. Donc, par exemple, une équipe ayant une fiche de 40-32-10 (somme équivalente à 82 matchs) au terme de la saison a 90 points.

# Diagramme de cas d’utilisation



# Prototypes d’interface





# Règles d’utilisation

* Le joueur devra choisir parmi les 30 équipes de la LNH laquelle il désire représenter au lancement de l’application.
* Le joueur devra appuyer sur le bouton « Commencer » afin de lancer le début de la saison. Si aucune équipe n’a été sélectionnée, l’action sera donc impossible.
* Une fois l’équipe correctement sélectionnée, le joueur aura le choix entre simuler le prochain match et simuler tous les matchs qu’il reste avant d’atteindre la limite d’une saison, c’est-à-dire 82 matchs.
* Si le joueur clique sur le bouton «Simuler le prochain match», un adversaire sera déterminé au hasard et le score initial de zéro à zéro sera affiché à l’écran.
* Le joueur devra appuyer sur le bouton «Simuler » afin que le score final remplace le score initial. **Le score final est déterminé au hasard et ne prend pas compte de la fiche de l’équipe représentée par le joueur.**
* Par contre, si le joueur clique sur le bouton «Simuler la saison», le reste de la saison (82 matchs auxquels on soustrait le nombre de matchs déjà joués) sera simulé et l’équipe se verra attribuée des victoires/défaites/défaites en prolongation de façon aléatoire (de sorte que la somme de tous ces résultats acquis soit égale au nombre de matchs qu’il reste à disputer au cours de la saison actuelle).
* Au terme d’une saison, le joueur aura la possibilité d’enregistrer l’équipe choisie ainsi que la fiche obtenue (VV-DD-DPDP).
* Qu’il enregistre ou non, l’application recommencera du début à partir de ce moment.

# Description spécifique

La première activité de l’application sera constituée de 30 TextViews qui seront les noms de chaque équipe ainsi que la ville ou état qu’elles représentent. À côté de chaque TextView se trouvera un ImageView représentant l’équipe mentionnée dans le TextView adjacent (par exemple le logo des Canadiens de Montréal sera obligatoirement situé à côté du TextView dans lequel sera écrit « Canadiens de Montréal »). Il sera obligatoire de sélectionner une équipe afin de progresser dans l’application. Pour en sélectionner une, le joueur devra cliquer sur l’ImageView de l’équipe voulue. Une fois cliquée, l’ImageView sera légèrement modifiée afin qu’il soit visible de voir qu’elle a été sélectionnée. Une fois cette étape franchie, il faudra cliquer sur le bouton « Commencer » afin de progresser dans l’application. Dès ce moment, la deuxième activité est appelée. Celle-ci contient un ImageView contenant le logo de l’équipe choisie dans l’activité précédente ainsi que deux boutons, l’un permettant de simuler le prochain match et l’autre permettant de simuler l’intégrité de ce qu’il reste en terme de matchs à la saison en cours. À partir de ce moment, pour progresser dans l’application, il faudra cliquer sur l’un des deux boutons présents. Si l’on clique sur le bouton « Simuler le prochain match », la troisième activité est appelée. Celle-là contient deux ImageViews dont l’une est celle qui contient le logo de l’équipe sélectionnée lors de la première activité et l’autre contient le logo de l’équipe qui sera l’équipe adverse lors de ce match que l’on s’apprête à simuler. Ce dernier ne pourra pas être le même que celui présent dans la première ImageView pour la simple raison que jouer un match de hockey contre nous-mêmes est impossible ou totalement illogique. Afin de gérer l’équipe qui sera générée de façon aléatoire pour représenter l’équipe adverse, un tableau d’objets ayant une capacité de 29 (étant donné que l’équipe du joueur ne doit pas s’y retrouver) sera rempli de toutes les équipes qu’il sera possible de jouer contre. Chacune de ces équipes se verra attribuée un indice allant de 1 à 29. De cette façon, il sera possible de générer à l’aide d’un Random un int qui déterminera laquelle des équipes présentes dans le tableau sera le prochain adversaire de l’équipe du joueur (par exemple : le joueur a sélectionné les Blackhawks de Chicago et le Random génère le chiffre 1. Logiquement, si les équipes sont placées dans le tableau en ordre alphabétique, l’adversaire choisi pour le prochain match sera les Ducks d’Anaheim étant donné qu’ils sont les premiers de la liste). Puisque les Blackhawks de Chicago seraient théoriquement à l’indice 7 et qu’ils seraient, dans le cas de l’exemple donné ci-dessus, l’équipe choisie par le joueur, l’indice 7 représenterait l’équipe venant à la suite des Blackhawks de Chicago dans la liste, c’est-à-dire celle qui en temps normal aurait dû avoir l’indice 8. C’est donc l’Avalanche du Colorado qui se verrait attribué l’indice 7 dans ce cas, si les équipes sont classées en ordre alphabétique dans le tableau. Il en irait ainsi de suite. Pour progresser dans l’application, le joueur devra cliquer sur le bouton « Simuler ». Dès que cette action est posée, le score est déterminé au hasard. Dans la troisième activité, il y a deux TextViews qui contiendront initialement le chiffre 0. Ce chiffre sera remplacé par d’autres chiffres une fois que le Random qui gère les scores aura déterminé la marque finale du match entre les deux équipes. Le Random ici prendra en charge deux choses. Tout d’abord, il déterminera si le match se terminera avec un score exceptionnel (score élevé dans lequel une équipe aura par exemple 7 buts ou plus) et le score en tant que tel par la suite. Bien entendu, le Random prendra également en charge la gestion des égalités, c’est-à-dire qu’un match ne pourra pas être nul (par exemple une marque finale de 3-3). Une fois le score déterminé, la fiche de l’équipe du joueur sera mise à jour (fiche qui sera initialement de 0-0-0). Si l’équipe du joueur a plus de buts que l’équipe adverse, la fiche sera désormais de 1-0-0. Si l’équipe perd, la fiche sera de 0-1-0 et si elle perd en prolongation, la fiche sera de 0-0-1 et ainsi de suite tout au long de la saison. Pour en revenir à la deuxième activité appelée lors de l’emploi de l’application, si le joueur clique sur le bouton « Simuler la saison », un toast sera affiché, informant le joueur de la fiche finale de l’équipe lors de la saison qui vient tout juste d’être simulée dans son intégrité. Au terme d’une saison simulée, le joueur se verra offrir l’opportunité de sauvegarder dans un fichier texte le nom de l’équipe qu’il représentait ainsi que la fiche finale de celle-ci à la fin de la saison. Le joueur peut toutefois refuser de sauvegarder. Voilà qui refuse le fonctionnement global de l’application que je vise à développer dans le cours. Certaines modifications seront certainement apportées tout au long de la session, bien sûr.

# Nouveauté

La nouvelle technologie (que nous n’avons pas vu au cours de Conception d’applications mobiles lors de la session précédente) que j’ai choisi d’appliquer dans mon application est l’utilisation de la classe OutputStreamWriter, classe héritant de Writer. Cette classe me permettra grâce à ses fonctionnalités de sauvegarder des données recueillies au cours de l’utilisation de mon application dans un fichier texte. Elle permet tout simplement d’écrire dans un fichier texte. Je pourrais aussi me servir de la classe OutputStreamReader afin de lire dans un fichier texte, mais ce n’est point nécessaire pour le bon fonctionnement de l’application.

# Références utilisées

* <https://developer.android.com/reference/java/io/OutputStreamWriter.html>
* <http://stackoverflow.com/questions/23320070/appending-a-string-to-an-existing-file-using-outputstreamwriter>

Charles Bouchard-Tremblay