Fabien Brossier
Christophe Desclaux

Le TabooGeek

But du jeu Taboo initial:

Le jeu se joue à 2 équipes ou davantage. Les équipes sont obligatoirement composées de 2 personnes. L'équipe 1 commence. Le joueur 1 doit faire deviner un mot oralement au joueur 2. Cependant, le mot à deviner est accompagné d'une liste de 5 mots "interdits". Le joueur 1 ne doit pas prononcer ces 5 mots. Si le joueur 2 devine le mot, l'équipe gagne 1 point. S'il n'arrive pas à deviner le mot dans le temps imparti ou si le joueur 1 prononce un mot interdit, l'équipe ne gagne aucun point. C'est ensuite à l'équipe 2 de jouer, et ainsi de suite jusqu'à avoir fait X tours de jeu.
L'équipe qui a le plus de point à la fin du jeu a gagné.

Adaptation des règles au projet

Le jeu sera adapté sur informatique. Par conséquent, le jeu doit être en réseau pour au moins gérer 2 joueurs. Le mot à trouver est généré aléatoirement par l'ordinateur, choisi dans une liste. A partir de notre base de données il génère à partir des relations la liste des mots interdits. Cette liste et le mot à deviner sont affichés à l'écran du joueur 1. Celui-ci tape un premier indice et l'envoie au joueur 2. L'indice est analysé pour vérifier qu'il ne contient pas de mot interdit, auquel cas il est affiché sur l'écran du joueur 2. Le joueur 2 peut alors décider de proposer un mot. Si ce n'est pas le cas ou si le mot est incorrect, le joueur 1 est invité à proposer un nouvel indice, et ce jusqu'à ce que le mot ait été trouvé ou jusqu'à ce qu'un joueur abandonne la partie.

Use Case

1. Le joueur 1 doit faire deviner un mot au joueur 2 : le joueur choisi alors un mot dans la base de données (mot choisi parmi une liste de technologies informatiques).
2. Le pc indique alors une liste de mots à ne pas utiliser.

3. Le joueur doit alors faire trouver le mot le plus rapidement possible.

Exemple

1. Fabien choisi Hiver
2. Le pc lui dit de ne pas utiliser:
    a. froid
    b. flocon
    c. neige
    d. glacial
    e. ski
3. Fabien donne pour indice: **après l'automne**
4. Christophe répond: Pluie => faux
5. Fabien donne un indice: **avant le printemps**
6. Christophe répond: Hiver => vrai, Christophe gagne donc des points.

Partie serious game

Le but est ici de générer une base de données en parallèle de dbpedia qui va être enrichie avec les mots qui ont permis de trouver la réponse. Nous pensons qu'un critère de pondération sur le poids des mots permet de les différencier.

Mise en valeur sémantique du projet

Notre but sera au final de construire une base SKOS avec des entités informatiques qui soit le plus proche possible de la réalité. Ainsi une fois que le jeu aura été assez utilisé nous aurons une base de données la plus complète possible d'entités au format SKOS à laquelle nous ajoutons une pondération des éléments.

Les itérations de notre projet

1. Avoir une application qui quand on lui donne un mot, récupère la liste de 5 mots les plus liés dans dbpedia
    a. création d'une base dbpedia moins gros que la version original de 10To (fabien)
    b. connection 4store (christophe)
2. Pouvoir créer une base SKOS que l'on puisse incrémenter à l'aide de couples (entité, indice) (par exemple: ("php","web")
3. Avoir une base de données contenant tous les mots qu'on souhaite exploiter
4. Avoir une API qui permet au joueur de choisir un mot dans la base de données
5. Lier la partie jeu à la partie "serious": avec l'ajout de tous les indices données par l'utilisateur dans la base SKOS
6. Ajouter la pondération des indices dans la base SKOS
7. Faire une vérification des indices en fonction de l'orthographe
8. Ajouter la possibilité de demander au joueur grâce à quel indice il a trouvé la réponse (permet d'avoir une meilleur pondération)