





Plan

I Présentation du jeu

II Coté technique

1) Technologies utilisées

RoR

4store

Html5

2) Conception de notre application

Stockage des données

Peuplage de la base

Requêtes utilisées

III Enrichissement de concepts

IV Démo



Présentation du jeu : Règles

- 2 joueurs
- Use Case :
 - Joueur1 doit faire deviner un mot au joueur 2 :
 - il choisit le mot dans la base de données
 - Le pc indique alors une liste de 5 mots Taboo ! (à ne pas prononcer).
 - Joueur1 tente de faire deviner à Joueur2 le mot à l'aide d'un seul mot et sans utiliser un mot Taboo !
- Jeu en plusieurs manches
 - celui qui marque le moins de points a gagné
- Le jeu devient de plus en plus difficile au fil des utilisations !



Présentation du jeu : exemple

Mot à faire deviner





Présentation du jeu : exemple

Les mots interdits

Touche (4)
Souris (3)
Cable (2)
Périphérique (2)
Interface (2)



Présentation du jeu : exemple

Les mots interdits

Touche (4)
Souris (3)
Cable (2)
Périphérique (2)
Interface (2)

Détails

Le mot à deviner : **Clavier**
Temps pour répondre : **1:23:8**



Présentation du jeu : exemple

Les mots interdits

Touche (4)
Souris (3)
Cable (2)
Périphérique (2)
Interface (2)

Détails

Le mot à deviner : **Clavier**
Temps pour répondre : **1:23:8**

Indique-nous l'indice qui a permis de trouver le mot. Cela améliore notre jeu et contribue à enrichir les concepts liés à l'informatique.





Présentation du jeu : exemple

Les mots interdits

Touche (4)
Souris (3)
Cable (2)
Périphérique (2)
Interface (2)

Détails

Le mot à deviner : **Clavier**

Temps pour répondre : **0:0:0**

Indique-nous l'indice qui a permis de trouver le mot. Cela améliore notre jeu et contribue à enrichir les concepts liés à l'informatique.





Plan

I Présentation du jeu

II Coté technique

1) Technologies utilisées

RoR

4store

Html5

2) Conception de notre application

Stockage des données

Peuplage de la base

Requêtes utilisées

III Enrichissement de concepts

IV Démo



Technologies utilisées : ROR

- Framework WEB
- Patron d'architecture MVC
 - Modèle = Serveur de triplets 4Store
 - Contrôleur = Requêtes SPARQL
 - Vue = templates HTML + JavaScript



Technologies utilisées : 4Store

- Serveur de triplets RDF
- Requêtes en mode CRUD
- EndPoint SPARQL en ligne !
 - <http://taboogeek.zouig.org:8081/test/>
- Des librairies pour TOUS les langages



Technologies utilisées : HTML5

- Utilisation de la balise dataList pour la génération de la complétion sur les mots de la base de données

Mot à faire deviner

Clavier

Tableur

Disque dur

Informatique

→

Mot à faire deviner

→



Plan

I Présentation du jeu

II Coté technique

1) Technologies utilisées

RoR

4store

Html5

2) Conception de notre application

Stockage des données

Peuplage de la base

Requêtes utilisées

III Enrichissement de concepts

IV Démo



Notre application : Stockage

- Stockage selon un dérivé de skos

```
<skos:Concept rdf:about="http://tabooGeek.zouig.org/#Informatique">
  <skos:definition>L'informatique est le domaine scientifique, tech... </skos:definition>

  <skos:prefLabel>Informatique</skos:prefLabel>
  <tabooGeek:element rdf:parseType='Resource'>
    <skos:altLabel>Ordinateur</skos:altLabel>
    <tabooGeek:weight>4</tabooGeek:weight>
  </tabooGeek:element>
  ...
  <tabooGeek:element rdf:parseType='Resource'>
    <skos:altLabel>Donnée</skos:altLabel>
    <tabooGeek:weight>3</tabooGeek:weight>
  </tabooGeek:element>
</skos:Concept>
```




Notre application : Peuplage

- Première génération avec jeu de tests
 - Exemple : Clavier, Périphérique, Souris...
- Récupération à l'aide de données provenant de DBpedia
 - Mise en place de requêtes de récupération sur un EndPoint DBPedia
 - Cependant mots trop complexes pour l'utilisateur lambda



Notre application : requêtes utilisées

```
SELECT ?prefLabel ?concept ?altLabel ?element ?weight  
WHERE {
```

```
  ?concept skos:prefLabel ?prefLabel .
```

```
  ?concept tabooGeek:element ?element .
```

```
  ?element skos:altLabel ?altLabel .
```

```
  ?element tabooGeek:weight ?weight .
```

```
  FILTER REGEX(?prefLabel, "" + @word + "")
```

```
}
```

```
ORDER BY DESC(?weight) LIMIT 5
```



Plan

I Présentation du jeu

II Coté technique

1) Technologies utilisées

RoR

4store

Html5

2) Conception de notre application

Stockage des données

Peuplage de la base

Requêtes utilisées

III Enrichissement de concepts

IV Démo



Enrichissement de concepts

- Mise en place du système de pondération
 - Mise en évidence des mots les plus proches du mot à trouver

```
<skos:Concept rdf:about="http://tabooGeek.zouig.org/#Informatique">  
  <skos:prefLabel>Informatique</skos:prefLabel>  
  <tabooGeek:element rdf:parseType='Resource'>  
    <skos:altLabel>Ordinateur</skos:altLabel><tabooGeek:weight>4</tabooGeek:weight>  
  </tabooGeek:element>  
  <tabooGeek:element rdf:parseType='Resource'>  
    <skos:altLabel>Donnée</skos:altLabel><tabooGeek:weight>3</tabooGeek:weight>  
  </tabooGeek:element>  
</skos:Concept>
```

- Utilisation de SPARQL 1.1 UPDATE



Place à la démo !



Pour le futur

- Augmentation de la base de mots
 - Adaptation et nettoyage du jeu de données
 - Ajout du broader et de la définition
- Gestion des points et du temps
- Jeu en réseau
 - Utilisation de l'API FaceBook
- Jeu pour plateforme Android



En bref

- Jeu qui nécessite un grand jeu de données nettoyé
- Découverte des possibilités de RubyOnRails
- Étude d'un cas concret lié au Web Sémantique
- Technologies puissantes mais trop jeunes

[illegible]