

Dancing Cube Django

Projet Django - HE-Arc - 29.04.2020

Ergün Kiliç, Julien Linder, François Bouthillier

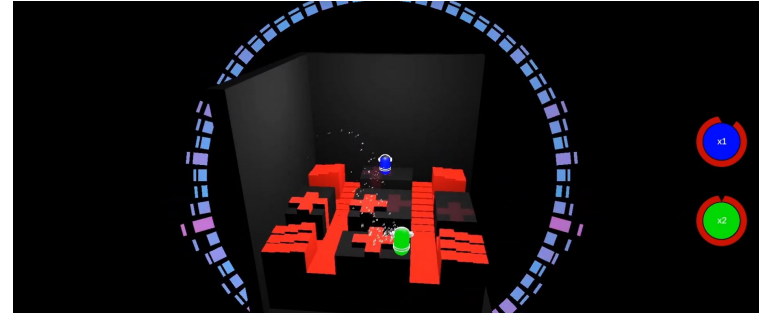
À l'attention de Monsieur David Grunenwald

Sommaire

1. Contexte
2. Cahier des charges
3. Démonstration
4. Relations de la base de données
5. Map
6. Upload / Download
7. Traduction
8. Système de recherche
9. Système de vote
10. Catégories (taggit)
11. Conclusion

1. Contexte

- Jeu musical de rythme et réflexe
- Etienne Hüsler et Alexander Wohlfahrt
- Partage de création
- BeatSaber , OSU et autres



2. Cahier des charges

- MVP
 - Structure principale du site
 - Système d'authentification: utilisateurs et administrateurs
 - Niveaux
 - Téléchargement (tous)
 - Publication et système de notes (utilisateurs)
 - Modification et suppression (auteurs et admins)
- Objectifs principaux
 - Recherche par catégories ou par texte
 - Prise en main d'un autre hébergeur
- Objectifs Secondaires
 - Signalement d'un niveau
 - Déploiement sur un autre hébergeur
 - API: téléchargement d'un niveau
 - Page d'accueil avec texte introductif
 - Site web en plusieurs langues (avec système de traduction aisé)
 - Catégorie *hot* listant les niveaux récents qui fonctionnent

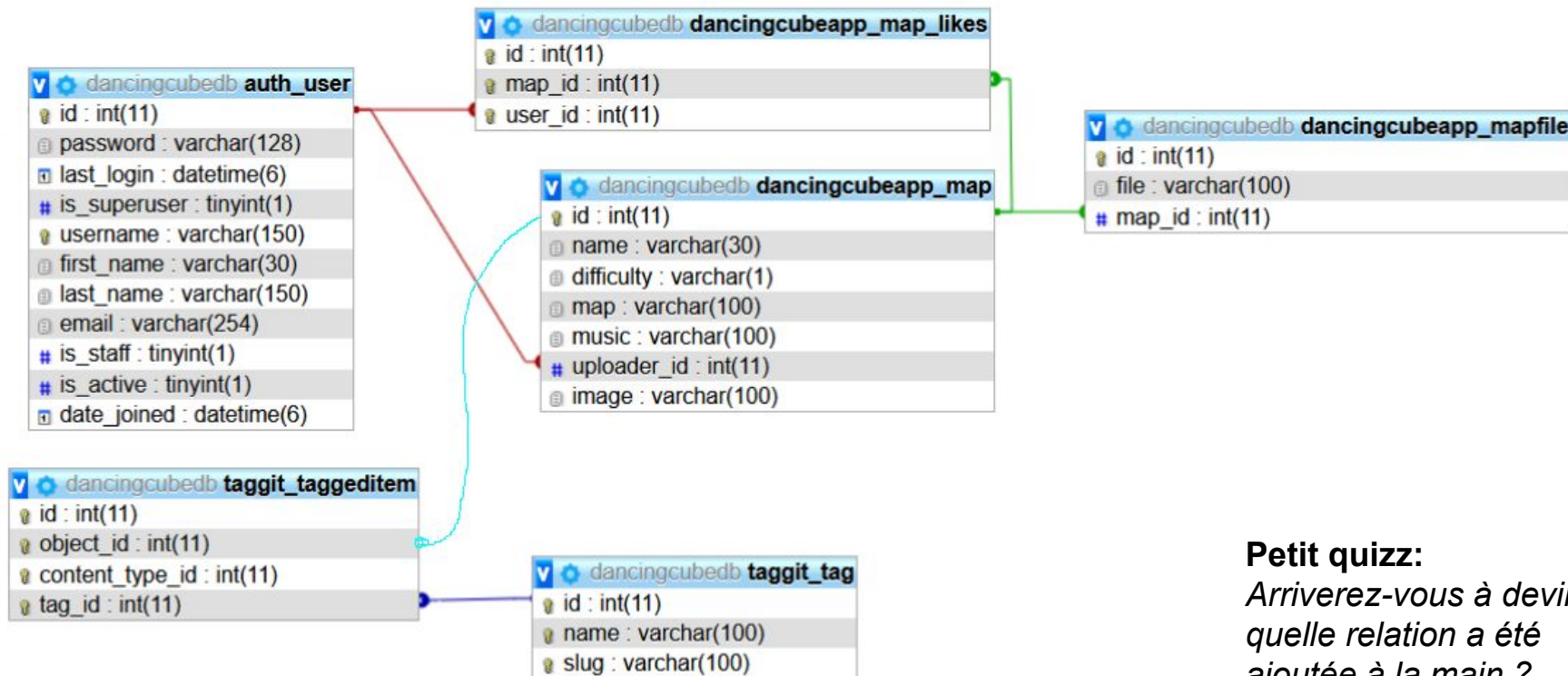
Notes des mandants:

- *Code propre et compréhensible*
- *Projet évolutif et adaptatif*
- *Code correctement commenté*
- *Site en anglais*

3. Démonstration

<https://dancing3.srvz-webapp.he-arc.ch/>

4. Relations de la base de données



Petit quizz:

*Arriverez-vous à deviner
quelle relation a été
ajoutée à la main ?*

Question sponsorisée par PAINT, merci à eux

5. Maps

- Niveau de jeu
 - Image
 - Musique
 - Difficulté
 - Niveau de map
 - Problème de taille de fichiers
 - Tag
- Modifiable
 - Selon l'utilisateur
- Téléchargeable

6. Upload / Download

- File/ImageField
 - Problème de gestion des gros fichiers
- Download zip on fly
 - Image
 - Music
 - Map(s)

7. Traduction

Django offre un système de traduction du site:

- Avec des balises, on indique les chaînes de caractères à traduire
- Django fournit des utilitaires pour extraire les chaînes à traduire dans un fichier de messages:

```
django-admin makemessage -l fr
```

Les traducteurs ne travaillent que dans ces fichiers là.

- Le fichier doit finalement être compilé

```
django-admin compilemessages
```

- Le serveur doit être relancé

8. Traduction - Exemple

```
from django.utils.translation import gettext as _  
  
def get_object(self, queryset=None):  
    obj = super(MapUpdateView, self).get_object(queryset)  
    if obj.uploader == self.request.user or self.request.u  
    return obj  
    else:  
        # Translators: user updating a map he doesn't own  
        raise Http404(_("You don't own this object"))
```

```
{% load i18n %}
```

```
{% comment %}Translators: Perhaps let 'Maps' in english, since it's the main model.{% endcomment %}  
{% trans "Maps" %}
```

DANCINGCUBEDJANGO	locale > fr > LC_MESSAGES > 🇫🇷 django.po
> .git	62
> .vscode	63 #. Translators: Perhaps let 'Maps' in english, since it's the main model.
> config	64 #: ../dancingcubeapp/templates/dancingcubeapp/nav.html:16
> dancingcube	65 msgid "Maps"
> dancingcubeapp	66 msgstr "Cartes"
> locale > fr > LC_MESSAGES	67 You, 19 days ago • close #15. Translation system is done, even if no
🇫🇷 django.mo	68 #: ../dancingcubeapp/templates/dancingcubeapp/nav.html:21
🇫🇷 django.po	69 msgid "Links"
	70 msgstr "Liens"
	71

Django choisit selon la préférence du navigateur, ou du choix de l'utilisateur.
Possibilité de faire aussi un système *monsiteweb.com/fr/...*

<https://github.com/HE-Arc/DancingCubeDjango/wiki/Translations>

9. Système de recherche

[/search?q=ma+query](#)

- Utilisation des filter

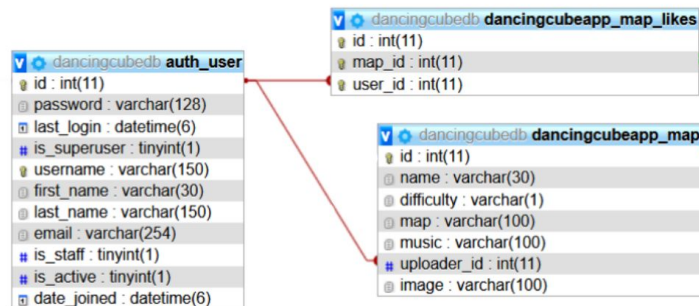
```
qs = Map.objects.all() # All maps

if query_term != '' and query_term is not None:
    qs = qs.filter(
        Q(name__icontains=query_term) |
        Q(music__icontains=query_term) |
        Q(uploader__username__icontains=query_term)
    ).distinct()
```

10. Système de vote

- Utilisation d'ajax (jQuery)

```
if request.user.is_authenticated:
    if map.likes.filter(id=request.user.id).exists():
        map.likes.remove(request.user) # dislike
        is_liked = False
    else:
        map.likes.add(request.user) # like
        is_liked = True
```



10. Catégories (taggit)

Volonté de catégoriser les maps

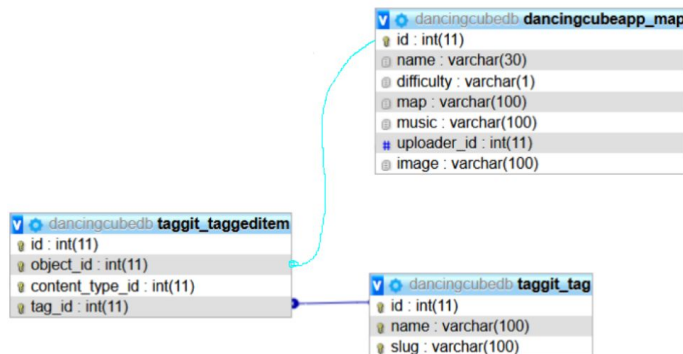
pip install **django-taggit**, *taggit* dans les INSTALLED_APPS, *migrate* puis:

```
class Map(models.Model):  
    ... tags = TaggableManager() # Init taggit, to tags our maps
```

Cela va générer automatiquement nos tables et un input dans les formulaires

Accès aux maps: </maps/tags/tag1/>

Issue: Ajouter un tag après coup



11. Problèmes

- Problèmes à résoudre
 - Ajouter tag
 - Update map
- Difficultés rencontrées
 - Static files
 - Download

12. Conclusion

- Futur de l'application
 - Map
 - Dossier par Niveau
 - Nombre de téléchargement
 - "Algorithme" pour filtrage hot
 - Signalement
 - Notification pour administrateur
- Avis des mandants
 - Améliorer le système de recherche
 - Plus de statistiques
 - Nouvelles features