

# PROJET JAVA

## Réaliser un Captcha

---

# Table des matières

<b>Présentation du captcha et fonctionnement</b>	<b>3</b>
<b>Méthodes de travail</b>	<b>4</b>
<b>Fonctionnalités réalisées</b>	<b>5</b>
<b>Fonctionnalités apportées post soutenance</b>	<b>5</b>
<b>Procédure pour lancer le jeu</b>	<b>5</b>
<b>Architecture globale</b>	<b>6</b>
<b>Difficultés rencontrées et choix</b>	<b>7</b>
<b>Poursuite possible du projet</b>	<b>7</b>

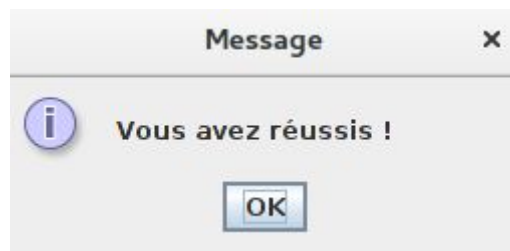
## Présentation du captcha et fonctionnement

Pour le projet Java de fin d'année IMAC2, il nous a été demandé de réaliser un captcha. Un captcha est un test requis pour accéder à un service internet. Il sert à sécuriser l'accès afin de différencier l'humain d'un robot malveillant.

Nous avons choisi pour notre thème captcha des personnages classés en 4 thèmes. Nous avons pris la catégorie *Anime*, contenant les sous-catégories suivantes : *Titeuf* et *Princess*, la catégorie *Game* avec : *Clash* (Royale) et *Fortnite* et pour finir la catégorie *Hero* avec les sous-catégories héros *Marvel* et héros *Dc*.

En arrivant sur le captcha, l'application nous demande de sélectionner toutes les images visibles d'une catégorie, par exemple "*Cliquez sur les images représentant : Hero*". L'utilisateur doit alors sélectionner les héros *Marvel* et *DC* et valider.

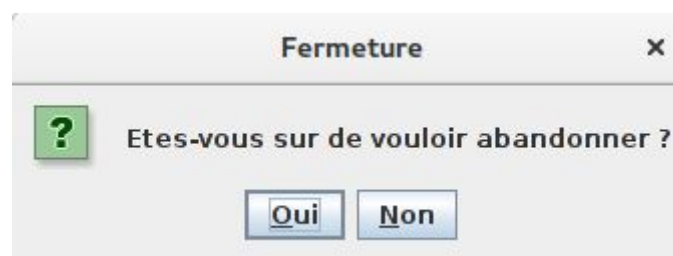
- Si l'utilisateur réussit, une pop-up s'ouvre pour lui indiquer qu'il a sélectionné les bonnes images et le captcha se ferme.



- Si l'utilisateur se trompe : une pop-up s'ouvre lui informant qu'il n'a pas réussi. On passe alors à la difficulté supérieure. L'utilisateur devra être plus précis dans sa sélection. L'application passe alors dans les sous-catégories et lui demandera de choisir toutes les images représentant spécifiquement un héros *Marvel* par exemple. Tant que l'utilisateur n'a pas réussi, le programme reste dans cette sous-catégorie.



- Si l'utilisateur ferme la fenêtre, le programme exécute cette option et considère comme un échec.



- Si l'utilisateur ne connaît pas spécialement la catégorie (par exemple il ne joue pas à Fortnite) alors il peut cliquer sur *réinitialiser*. Cela recharge le captcha aléatoirement lui proposant une nouvelle catégorie.

## Méthodes de travail

Nous avons utilisé *Git* et *Eclipse* pour réaliser notre application. Nous avons travaillé en quinconce afin que chacun puisse implémenter à son tour les fonctionnalités requises pour le projet.

Pour l'initialisation du projet, nous avons travaillé en pair programming afin de se mettre d'accord sur l'initialisation, la syntaxe de programmation à employer ou encore l'architecture globale.

## Fonctionnalités réalisées

A la première soutenance, nous avions un programme fonctionnel. Nous avons choisi de récupérer aléatoirement les catégories et les sous catégories pour afficher les images dans le désordre à chaque fois.

Nous avons implémenté deux niveaux de difficultés mais ceux ci étaient encore en statique. De ce fait, il nous a été conseillé de réaliser un parcours de notre arbre afin de récupérer le niveau de nos catégories et de savoir si nous sommes bien au premier niveau ou au deuxième ; respectivement le niveau "facile" ou le niveau "difficile".

En cliquant sur "réinitialiser", le captcha se recharge automatiquement en proposant une autre catégorie à sélectionner.

## Fonctionnalités apportées post soutenance

Suite à la pré-soutenance, nous avons eu quelques remarques quant à l'amélioration de notre projet concernant l'expérience utilisateur ou encore l'ergonomie dans les choix de développement.

C'est pourquoi, pour le rendu final, nous avons choisi de pouvoir fermer le programme par abandon ou par réussite comme si on se trouvait sur un vrai site. Le programme envoie alors "0" si l'utilisateur a réussi, ou "1" s'il a abandonné et donc il ne donnera pas accès à la suite du site internet.

Nous avons aussi automatisé les niveaux de difficulté avec la profondeur de notre arbre. Cela permettra à l'avenir de rajouter des sous-catégories dans l'arborescence de fichier tout en incrémentant le niveau.

## Procédure pour lancer le jeu

Télécharger le projet [ici](#) ou depuis l'ent, puis à partir de la racine du projet en ligne de commande linux :

```
cd ./CAPTCHA  
find . -name "*.java" -print | xargs javac -d classes  
rsync -avz --exclude '*.java' ./sources/ ./classes/  
java -cp classes fr.upem.captcha.ui.MainUi
```

Ou lancer :

```
./launch.sh
```

Le projet, fonctionnel sous windows et MacOS, est à lancer depuis Eclipse.

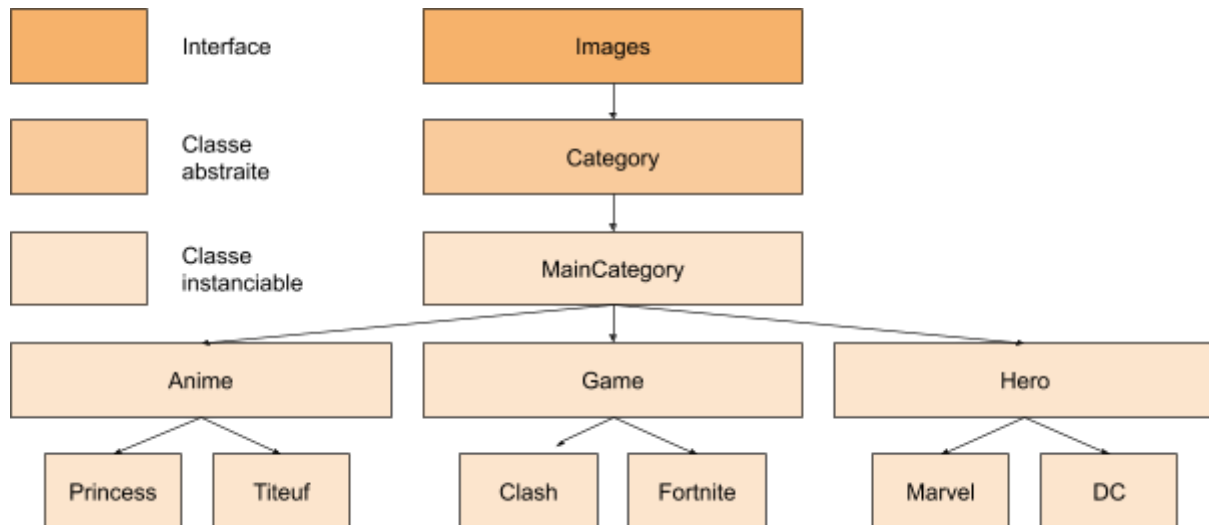
## Architecture globale

Nous avons une classe *MainUi* qui permet la gestion de l'interface graphique mais aussi le lancement et les actions principales de l'application.

Ensuite, nous avons une classe *MainController* qui permet la gestion du captcha, et notamment du choix de la catégorie sur laquelle on doit cliquer ou encore la comparaison entre les images correctes et les images sélectionnées sur l'interface graphique.

Au niveau des catégories, nous avons choisi une implémentation en arbre qui nous semblait être pertinente au vu de la structure de notre projet et notamment de l'arborescence de fichiers choisie. Chaque catégorie aura donc des (enfants) sous-catégories ainsi que des images (même si dans notre cas seules les feuilles de l'arbre ont des images).

Nous avons l'architecture suivante :



## Difficultés rencontrées

Premièrement, **nous avons fait le choix de ne pas créer un exécutable .jar** car cela remettait en cause notre traitement de fichiers pour la création de notre arborescence et donc la totalité de notre mode de fonctionnement du projet. Ce problème a été mentionné sur le forum du projet (depuis l'ent) et, n'ayant pas eu de réponse, nous avons décidé de vous créer un exécutable `launch.sh`, à la place.

## Conclusion et poursuites possibles

Ce projet de groupe nous a permis d'améliorer diverses compétences. En effet, l'utilisation d'une bibliothèque graphique et le développement de cette application en Java était nouveau pour chacun d'entre nous.

Nous pourrions ajouter des images et des sous-catégories pour augmenter le niveau de difficulté mais également, avec plus de temps de travail, nous pourrions rendre l'interface plus appréciable visuellement en tant qu'ingénieurs créatifs de la seule école publique d'ingénieurs alliant art et sciences.