

# Documentation captcha

Le captcha est disponible dans notre application dans la page /login.

## Front-end

Le captcha permet de jouer à un jeu de rapidité où il faut trier des objets qui vont dans l'océan (poisson / plongeur / pokemon eau / etc.)

Le but était de garder le thème du sujet nationale tout en callant le plus de référence à la culture jeux vidéo (serez-vous tous les retrouver ?).

Le challenge captcha en lui-même ne se base pas sur le score du joueur ou son efficacité mais sur les mouvements de sa souris pendant la partie.

Il y a trois issues à la partie :

- Le joueur atteint le score de -5, la partie échoue et il faut recommencer
- Le joueur atteint 10 et ses mouvements sont légitimes : Le captcha est validé
- Le joueur atteint 10 de score et ses mouvements sont suspects : il faut recommencer.

## Back end

### Analyse des Mouvements de Souris

Le backend utilise une stratégie sophistiquée d'analyse des mouvements de souris pour différencier les interactions humaines des interactions automatisées (bots).

#### Méthode de Classification

La classe EnhancedMouseMoveClassifier implémente un modèle de classification avancé qui extrait quatre caractéristiques clés des mouvements de souris :

Vitesse Horizontale Maximale :

Mesure la vitesse la plus rapide des mouvements horizontaux Les mouvements humains ont tendance à avoir des variations plus naturelles

Variance de la Vitesse Horizontale :

Calcule l'irrégularité des mouvements Les humains produisent des mouvements moins uniformes que les bots

Changements de Direction Horizontale :

Compte le nombre de changements de direction Les interactions humaines montrent plus de variations et d'hésitations

Distance Horizontale Totale :

Mesure l'amplitude totale des mouvements Différencie les parcours mécaniques des trajectoires plus fluides

#### Modèle d'Apprentissage

Le modèle utilise la Régression Logistique avec des données d'entraînement spécifiques :

Données de "bots" : Mouvements mécaniques, uniformes Données "humaines" : Mouvements variés, rapides, avec des changements de direction

#### Processus de Décision

Le classificateur calcule deux probabilités :

Probabilité d'être un humain Probabilité d'être un bot

La décision finale se base sur un seuil de 0.5 :

Si  $> 0.5$  : Considéré comme humain Si  $\leq 0.5$  : Considéré comme bot

### **Sécurisation et Stockage**

Pour renforcer la sécurité, chaque analyse est :

Générée avec un token unique Stockée temporairement dans Redis Associée à une durée de vie limitée

### **Flux de Sécurisation Complet**

Initialisation

Génération d'un identifiant client unique Création d'un token sécurisé

Génération de Token

Token chiffré avec JWT Stockage temporaire dans Redis

Analyse des Mouvements

Extraction et analyse des caractéristiques Décision humain/bot Stockage du résultat

Vérification Finale

Validation du token Vérification des mouvements Décision de procéder ou bloquer

### **Détails Techniques**

Bibliothèques Utilisées :

FastAPI pour l'API numpy pour les calculs scientifiques scikit-learn pour la classification Redis pour le stockage temporaire JWT pour la sécurisation des tokens

Sécurité

Clé secrète générée dynamiquement au démarrage Tokens avec durée de vie limitée Stockage temporaire des informations sensibles