

Atelier TESTS

Groupe 7

Dimitri PERREAUX
Thibault SCORIELLE
Nolan DELCROIX

Entreprise

Game Freak

But

Développer un NOUVEAU jeu Pokémon
Web App

Contraintes Matérielles

Doit fonctionner sur:

- PC
- Mac
- Android 10+

Spécifications Fonctionnelles

- 1 utilisateur = 1 compte par adresse mail
 - connection possible via compte Facebook ou Google
=> compte rattaché au mail lié au service externe
- 1 utilisateur = 1 dresseur
 - => possède :
 - des Pokémon
 - des objets
 - => peuvent :
 - se battre les uns contre les autres
 - voir :
 - pokémons qu'ils possèdent
 - objets qu'ils possèdent
- jeu contient un CLASSEMENT :
 - 1 victoire = + (10 points + nbr Pokémon restant invaincu)
 - 1 défaite = - (10 points + nbr Pokémon restant invaincu)
- 1 équipe de pokémons = 6 pokémons

- Rotation de l'équipe :
 - Dresseur obtient liste de 5 pokémons d'un coup :
=> Dresseur doit choisir 1 pokémon de la liste de rotation pour en remplacer 1 de son équipe (mais pas le plus récent)
- Dresseur obtient 1 objet par jour :
 - jusqu'à 3 MAX en gagnant des combats

Spécifications techniques

- Front-end développé en javascript via React
- Back-end développé en Python via FastAPI
- Base de donnée en PostgreSQL
- Déploiement dans un conteneur via Google Platform

Ressources impliquées

- Environnement de travail (software)
- Serveurs
- ??

KPIs:

- Couverture de code
- Nombre de crash
- Taux de réussite des tests
- Coût par défaut détecté

Budget

- Nombre de défaut détecté x coût
- Test de performance → Serveur de prod et serveur de test différent, utilisation de serveur robuste
- Tests de la sécurité / comptes utilisateurs (dû à la multiplicité des moyens de créer un compte)

Cas de tests

- tests positif

- 1) L'utilisateur se connecte et échange un pokémon de son équipe avec un de ceux proposé par la liste de rotation
- 2) L'utilisateur se connecte et fait 3 victoires (vérifier 3 objets)

- tests négatifs

- 3) L'utilisateur tente de se connecter avec un compte non valide.
- 4) L'utilisateur tente de se connecter sur le même compte simultanément à partir de 2 appareils différents.

Evaluation des risques

x	1	2	3	4	5
1					6)
2				4)	
3	3)				5)
4					
5					

5) Un pic de connexion fait crasher le serveur.

6) L'utilisateur fait une injection SQL lors de son inscription par exemple

Critères d'entrée et de sortie

Entrée

- Disposer d'un PC, d'un mac ou d'un téléphone sous android 10+
- Avoir un compte facebook et google
- Avoir un accès à Internet
- Avoir l'application
- Avoir/louer les serveurs (prod et test)
- Avoir des testeurs

- Cas de test disponible
- Disponibilité d'environnement de test
- Disponibilité des outils de test

Sortie

- Validation des tests avec risque élevé
- Taux de couverture définit atteint
- Les tests positifs sont validés et aucun défaut majeur
- Aucun problème de sécurité détecté

Planification

- cas de test 1 : L'utilisateur se connecte et fait un combat (victoire ? défaite ?)
- cas de test 2 : L'utilisateur se connecte et fait 3 victoires (vérifier 3 objets)
- cas de test 4 : L'utilisateur tente de se connecter sur le même compte simultanément à partir de 2 appareils différents.
- cas de test 5 : Un pic de connexion fait crasher le serveur.
- cas de test 6 : L'utilisateur fait une injection SQL lors de son inscription par exemple
- cas de test 3 : L'utilisateur tente de se connecter avec un compte non valide.

Priorisation

- Priorité 1 : cas de tests positif
- Priorité 2 : cas de tests négatifs impactants
- Priorité 3 : cas de tests négatifs restants

Dépendances

Faire stress test serveur après cas positif 1/ et 2/ (car on doit avoir des scénarios fonctionnels pour le stress test (avec Locust))

Facteurs d'effort de test

Location et paramétrage des serveurs de tests afin de vérifier que les tests de charge et de montée en charge fonctionnent. Va demander du temps et de l'argent, lié à un haut facteur de risque.