

Génie Logiciel

Cahier des tests d'intégration

ModIntelligence

Projet Autobots

23 décembre 2022

Table des matières

1	Description	3
1.1	Introduction	3
1.2	Objectifs	3
1.3	Environnement de tests	3
2	Plan de Test	4
2.1	Accueil	4
2.1.1	Description	4
2.1.2	Résultat(s) :	4
2.2	Inscription	4
2.2.1	Description	4
2.2.2	Résultat(s) :	4
2.3	Connexion	4
2.3.1	Description	4
2.3.2	Résultat(s) :	5
2.4	Paramètres	5
2.4.1	Description	5
2.4.2	Résultat(s) :	5
2.5	Véhicule	5
2.5.1	Description	5
2.5.2	Résultat(s) :	6
2.6	Environnement	6
2.6.1	Description	6
2.6.2	Résultat(s) :	6
2.7	Calcul des performances de véhicule	6
2.7.1	Description	6
2.7.2	Résultat(s) :	6
2.8	Portabilité	6
2.8.1	Description	6
2.8.2	Résultat(s) :	7

1 Description

1.1 Introduction

Pour déployer un logiciel, il est primordial lui faire subir des tests sous plusieurs aspects. Suivant notre méthode de travail basée sur le **Cycle en V**, il est nécessaire de suivre à bon escient les différentes étapes de la conception. Nous sommes actuellement dans la phase ascendante du processus de méthode de gestion. Après la réalisation du projet et suite à la validation des tests de recettes, les tests d'intégration de ce document devront être validés à leur tour afin de lancer la mise en production du logiciel **Autobot**.

1.2 Objectifs

Ces tests ont pour but de vérifier que le logiciel **Autobot** fonctionne conformément à sa conception générale et s'intègre avec le système d'information (**SI**).

Dans le cas où les tests ne sont pas validés, il sera nécessaire de reprendre ces derniers sur le document de conception général ou d'enrichir la production actuelle du logiciel.

1.3 Environnement de tests

Les tests seront réalisés avec des machines sous différents systèmes d'exploitation (Linux, Windows, MacOS). Il sera nécessaire de posséder une version à jour du moteur de jeu Unity3d.

Pour finir, pour tester le système d'impression il est utile de posséder l'imprimante industrielle qui sera connectée aux machines.

2 Plan de Test

2.1 Accueil

2.1.1 Description

☐ Lors du démarrage, le logiciel doit pouvoir récupérer rapidement la vidéo de présentation qui sera stockée sur une plate-forme vidéo afin de l'afficher.

2.1.2 Résultat(s) :

2.2 Inscription

2.2.1 Description

☐ Lors d'une inscription, le logiciel doit interagir efficacement avec la BDD pour enregistrer les informations du client.

☐ La page doit empêcher tous types d'attaques par injection SQL .

2.2.2 Résultat(s) :

2.3 Connexion

2.3.1 Description

☐ Lors d'une tentative de connexion, le logiciel doit vérifier la cohérences des valeurs entrée avec celles présentent dans la BDD de manière efficace et correct, aucune erreur ne doit être permise.

☐ La page doit empêcher tous types d'attaques par injection SQL.

2.3.2 Résultat(s) :

2.4 Paramètres

2.4.1 Description

- ☐ Lorsque des paramètres changent, le logiciel doit mettre à jour les nouvelles variables et écraser les anciennes, afin d'obtenir un résultat immédiat.
- ☐ Garder en mémoire les nouveaux paramètres si le logiciel est quitté.

2.4.2 Résultat(s) :

2.5 Véhicule

2.5.1 Description

- ☐ Si l'utilisateur charge un véhicule, la BDD doit pouvoir renvoyer efficacement et sans perte de données le résultat, la vitesse n'importe pas.
- ☐ Si l'utilisateur veut ajouter un élément au véhicule, la réaction de la base de donnée doit être rapide pour récupérer les informations de la pièce et afficher sur le rendu 3D le véhicule avec cette dernière ajoutée à son emplacement recommandé.
- ☐ Si l'utilisateur veut retirer un élément au véhicule, le rendu 3D doit être modifié en retirant la pièce du véhicule.
- ☐ Si l'utilisateur veut enregistrer un véhicule, la BDD doit permettre de stocker toutes les informations du véhicule.
- ☐ Si l'utilisateur veut enregistrer un véhicule, le logiciel doit pouvoir créer un fichier XML cohérent automatiquement.

2.5.2 Résultat(s) :

2.6 Environnement

2.6.1 Description

☐ Si l'utilisateur charge un environnement, la BDD doit pouvoir renvoyer efficacement et sans perte de données le résultat, la vitesse n'importe pas.

☐ Si l'utilisateur veut modifier l'environnement, le rendu 3D doit pouvoir afficher le nouvel environnement.

2.6.2 Résultat(s) :

2.7 Calcul des performances de véhicule

2.7.1 Description

☐ Le calcul du résultat doit être rapide et être stocké dans un historique (BDD).

☐ Si l'utilisateur veut enregistrer les résultats, la BDD doit stocker l'historique avec l'état du véhicule et de l'environnement.

2.7.2 Résultat(s) :

2.8 Portabilité

2.8.1 Description

☐ Le logiciel doit être capable de tourner de manière similaire sur les trois systèmes d'exploitations principaux (Linux, Windows, MacOS).

☐ Le logiciel doit pouvoir communiquer entre plusieurs systèmes d'exploitation différents.

2.8.2 Résultat(s) :