1. Fonctions principales développées

Ce chapitre présente les fonctionnalités principales développées dans l’incrément 1 en utilisant une démarche par cas d’usage et par cas d’utilisation (CU).

1.a Rappel sur les cas d’usage

Les cas d’usage recensent les étapes essentielles du cycle de vie d’un produit depuis la fabrication du produit jusqu’à l’élimination ou recyclage de ce produit.  
À chaque étape du cycle de vie correspond un cas d’usage (si cette étape induit des fonctionnalités à définir pour le produit considéré). Pour chaque cas d’usage, plusieurs cas d’utilisations distincts peuvent être définis.  
Parmi les cas d’usage, on retrouve généralement ceux de fabrication du produit (comprenant ou non les activités de test du produit fabriqué), de conditionnement (paramétrage éventuel, expédition et transport), de commercialisation (paramétrage éventuel, mode de démonstration, installation sur site...), d’utilisation (par le ou les USERs), de maintenance (SAV ou diagnostique), de désinstallation et de recyclage (élimination ou valorisation).

1.b Rappel sur les cas d’utilisation

1.d CU Principal

|  |  |
| --- | --- |
| Titre | CU Stratégique : USER utilise ScoutBOT |
| Résumé | Un USER souhaite connecter JumpC et PocketRace pour piloter le robot, voir où le robot est dans la pièce, détecter les obstacles avec le lidar. L’USER peut passer du mode statique au mode dynamique. Il peut piloter le robot via PocketRace et le calibrer à tout moment. |
| Portée | Le SàE et ses périphériques |
| Niveau | Stratégique |
| Acteurs directs | USER, Balises |
| Acteurs indirects | Environnement |
| Préconditions | * Robot allumé et fonctionnel * JumpC et PocketRace déjà connecté en WIFI * Balises placées dans l’environnement |
| Garanties minimales | Message d’erreur si perte de connexion |
| Garanties en cas de succès | * Applications lancées * Connexion WIFI établie entre JumpC et PocketRace * Affichage de la position sur la carte de l’<ECRAN\_PRINCIPAL\_STATIQUE> * Pilotage du robot possible manuellement * Pilotage du robot vocalement possible * Extinction des applications et déconnexions si demande de l’USER |
| Scénario nominal | 1. User lance Pocket et Race 2. User lance JumpC. 3. JumpC affiche <ECRAN\_DEMARRAGE> pendant T\_CHARGE. 4. JumpC affiche <ECRAN\_CONNEXION>. 5. USER demande la connexion entre JumpC et Pocket. 6. JumpC affiche <ECRAN\_PRINCIPAL> 7. USER pilote le robot 8. USER demande le démarrage de la cartographie 9. USER quitte JumpC. |
| Variantes | 7-8 : [JumpC rafraîchit <ECRAN\_PRINCIPAL> tous les T\_RAFRAICHI]   1. JumpC rafraîchit l’<ECRAN\_PRINCIPAL>   8 : [USER ne souhaite pas lancer la cartographie]   1. Va en 7    1. : [USER demande à quitter JumpC] 2. Va en 9 |
| Extensions | 7-9 : [ERREUR\_DE\_CONNEXION] Incrément 1   1. JumpC affiche <ECRAN\_ERREUR\_CONNEXION> 2. PocketRace s’éteint 3. USER aquitte l’erreur de connexion 4. Va en 3.   7-9 : [ERREUR\_DE\_CONNEXION] Incrément 2   1. JumpC affiche <ECRAN\_ERREUR\_CONNEXION> 2. PocketRace se met en état de STAND\_BY. 3. USER acquitte l’erreur de connexion 4. JumpC affiche <ECRAN\_CONNEXION> 5. Va en 5.   7-9 : [ERREUR\_DE\_CONNEXION et USER demande à quitter JumpC]   1. JumpC ferme sa fenêtre. |
| Informations complémentaires |  |

|  |  |
| --- | --- |
| Titre | Sous-CU 1 : Connexion entre JumpC et PocketRace |
| Résumé | L’USER connecte JumpC et PocketRace |
| Portée | SàE |
| Niveau | USER |
| Acteurs directs | USER |
| Acteurs indirects | Environnement |
| Préconditions | * Application JumpC et PocketRace lancées * Le wifi est déjà activé sur PocketRace et JumpC * JumpC connaît l’addresse ip de PocketRace et le port réseau * Port du réseau de PocketRace et communication fonctionnelle |
| Garanties minimales |  |
| Garanties en cas de succès | * Message de réussite de connexion côté JumpC |
| Scénario nominal | 1. USER demande à se connecter à Pocket 2. JumpC se connecte au réseau de Pocket 3. Pocket acquitte que la connexion a bien été établie. 4. JumpC affiche ECRAN\_CONNEXION\_REUSSITE |
| Variantes | 2 : [JumpC n’est pas appairé à Pocket]   1. JumpC affiche ECRAN\_CONNEXION\_FAILED 2. USER acquitte que la connexion n’a pas été établie 3. Va en 1. |
| Extensions |  |
| Informations complémentaires |  |

|  |  |
| --- | --- |
| Titre | Sous-CU 3 : USER pilote le robot |
| Résumé | L’USER va pouvoir piloter le robot manuellement avec JumpC via un des boutons directionnels. |
| Portée | SàE |
| Niveau | USER |
| Acteurs directs | USER |
| Acteurs indirects | Environnement |
| Préconditions | N.A. |
| Garanties minimales |  |
| Garanties en cas de succès | * Le robot se déplacera en fonction des commandes de l’USER |
| Scénario nominal | 1. USER saisit une direction sur JumpC 2. JumpC demande à Pocket d’exécuter la commande de direction 3. Pocket transmet la demande de direction à Race 4. Le robot se déplace dans la direction demandée |
| Variantes | 1 : [USER demande à effectuer une commande vocale]   1. JumpC démarre un enregistrement vocal de T\_ENREGISTRE 2. USER donne une commande vocale dans le micro 3. JumpC enregistre la commande vocale 4. JumpC décrypte la commande vocale en une direction 5. Va en 2. |
| Extensions | 1 : [DIRECTION\_INCONNUE]   1. JumpC n’envoie aucun message de direction à Pocket 2. Fin du CU |
| Informations complémentaires |  |

|  |  |
| --- | --- |
| Titre | Sous-CU 4 : USER quitte JumpC |
| Résumé |  |
| Portée |  |
| Niveau |  |
| Acteurs directs |  |
| Acteurs indirects |  |
| Préconditions |  |
| Garanties minimales |  |
| Garanties en cas de succès |  |
| Scénario nominal | 1. USER demande à quitter JumpC 2. JumpC envoie un signal d’extinction à Pocket 3. JumpC se deconnecte de Pocket 4. Pocket envoie un signal d’extinction à Race 5. Race et Pocket terminent leur programme 6. JumpC ferme sa fenêtre. |
| Variantes |  |
| Extensions |  |
| Informations complémentaires |  |

|  |  |
| --- | --- |
| Titre | Sous-CU 5 : USER démarre la cartographie sur JumpC |
| Résumé | USER choisit de démarrer la cartographie. Cela consiste |
| Portée |  |
| Niveau |  |
| Acteurs directs |  |
| Acteurs indirects |  |
| Préconditions |  |
| Garanties minimales |  |
| Garanties en cas de succès |  |
| Scénario nominal | 1. JumpC initialise l’affichage cartographe par défaut (STATIQUE) 2. JumpC envoie à Pocket une demande de démarrage de la cartographie 3. Pocket effectue le scanning initial, et traite les données 4. Pocket envoie les données du scanning initial à JumpC 5. JumpC reçoit le scanning initial et enregistre le mapping 6. Pocket lance la cartographie continue 7. JumpC affiche <ECRAN\_PRINCIPAL> avec la cartographie mise à jour 8. USER choisit l’affiche DYNAMIQUE 9. Fin du CU |
| Variantes | 8 : [USER choisit de rester en affichage STATIQUE]   * 1. Va en 9 |
| Extensions |  |
| Informations complémentaires |  |