|  |  |
| --- | --- |
| **Eléments de test de cas d’utilisation** | **Description spécifique au projet** |
| Cas de test | Test du lancement d’une nouvelle partie |
| Acteurs | Usager |
| Evènements déclencheurs | Appui sur le bouton “Nouvelle partie” depuis la page d’accueil |
| Parties prenantes et leurs intérêts | Usager - lancer une partie depuis le départ avec création d’un nouvel avatar |
| Niveau | Stratégique |
| Portée | Données usager pour la création d’une nouvelle partie :   * Nom de joueur - chaîne de caractères contenant uniquement des lettres majuscules, minuscules et chiffres (pas de caractères spéciaux) et d’une longueur maximale de 16 caractères. * Nom d’avatar - chaîne de caractères contenant uniquement des lettres majuscules, minuscules et chiffres (pas de caractères spéciaux) et d’une longueur maximale de 16 caractères. * Type d’avatar - carrousel proposant différents types d’avatars possibles à incarner (chien, chat, oiseau, poulpe, robot) |
| Pré-condition | Aucune |
| Post-condition | Un nouveau joueur est créé avec son avatar et une partie est lancée |
| Scénarii de test | 1. Toutes les données d’entrée vides (test limite/extrême) :    1. Nom\_Joueur, Nom\_Avatar = “ “ (chaîne vide)    2. L’usager soumet les données pour créer une nouvelle partie    3. Réponse attendue : un message d’erreur indiquant que les champs sont vides 2. Données d’entrée incomplète :    1. Le champ Nom\_Joueur ou Nom\_Avatar = “ “ (chaîne vide) et l’autre est correctement rempli    2. L’usager soumet les données pour créer une nouvelle partie    3. Réponse attendue : un message d’erreur indiquant que le champ est incorrect. 3. Données d’entrée incorrecte 1    1. Nom\_Joueur ou Nom\_Avatar contient des caractères non autorisés    2. L’usager soumet les données pour créer une nouvelle partie    3. Réponse attendue : un message d’erreur indiquant que le champs contient des caractères non autorisés (en précisant que seules des majuscules, minuscules et nombres sont autorisés) 4. Données d’entrée incorrecte 2    1. Nom\_Joueur ou Nom\_Avatar contient une chaîne de caractères trop longue    2. L’usager soumet les données pour créer une nouvelle partie    3. Réponse attendue : un message d’erreur indiquant que le champ est trop long (en précisant qu’il ne peut contenir que 16 caractères maximum) 5. Données d’entrée existant déjà dans la base de données :    1. Nom\_Joueur et Nom\_Avatar sont correctement renseignés mais existent déjà tous les deux dans la BDD    2. L’usager soumet les données pour créer une nouvelle partie    3. Réponse attendue : un message d’erreur indiquant que le personnage existe déjà et qu’il faut changer le Nom\_Joueur ou le Nom\_Avatar |
| Extensions |  |
| Contraintes |  |

|  |  |
| --- | --- |
| **Eléments de test de cas d’utilisation** | **Description spécifique au projet** |
| Cas de test | Test de chargement d’une partie déjà sauvegardée |
| Acteurs | Usager, base de donnée |
| Evènements déclencheurs | Appui sur la sauvegarde concernée dans le Menu “Charger une partie” |
| Parties prenantes et leurs intérêts | * Usager : reprendre une partie là où il l’a arrêtée * Base de donnée : donne accès au jeu à toutes les sauvegardes, pour les lister et les lancer |
| Niveau | Stratégique |
| Portée | * Données de sauvegarde   + Pseudonyme du joueur   + Nom de l’avatar   + Type de l’avatar   + Temps total de jeu   + Statistiques de l’avatar   + Date et heure de la fin de la session de jeu * Données du modèle   + Nom de l’avatar   + Type de l’avatar   + Temps total de jeu   + Statistiques de l’avatar   + Dates et heures du début de la nouvelle session de jeu |
| Pré-condition | Le joueur a sauvegardé au moins une partie |
| Post-condition | La partie reprend au même endroit qu’avant |
| Scénarii de test | 1. Cas nominal : une partie a été sauvegardée    1. Dans le menu “Charger une partie”, les informations suivantes sont affichées pour chaque sauvegarde : “Sauvegarde n°N”, noms du joueur et de son avatar    2. Aucune sauvegarde n’est sélectionnée par défaut → le bouton “Charger” est grisé    3. Le joueur sélectionne une sauvegarde en cliquant dessus   → mise en avant de la sauvegarde (eg. surligner)  → le bouton “Charger” devient accessible   1. Le joueur lance la partie → les informations sont récupérées : noms du joueur et de l’avatar, statistiques, temps total de jeu, date et heure, lieu 2. Aucune sauvegarde n’est disponible    1. Le bouton “Charger une partie” du menu d’accueil est grisé → on n’a pas accès au menu des sauvegardes 3. Sauvegarde corrompue    1. Lors de la sauvegarde, 2 copies sont faites, au cas où l’une des deux soit corrompue    2. Si seulement une est corrompue → le jeu charge l’autre    3. Si les 2 sont corrompues → le jeu affiche l’erreur “Erreur : sauvegarde <nom sauvegarde> corrompue” |
| Extensions |  |
| Contraintes |  |

|  |  |
| --- | --- |
| **Eléments de test de cas d’utilisation** | **Description spécifique au projet** |
| Cas de test | Test de changement de lieu avec son avatar |
| Acteurs | Usager |
| Evènements déclencheurs | Appuie sur la flèche de gauche ou sur la flèche de droite. |
| Parties prenantes et leurs intérêts | * Usager : Changer d’environnement de jeu. |
| Niveau | Stratégique |
| Portée | * environnement : instance d’une classe Environnement qui définit l'environnement dans lequel se trouve l’avatar. |
| Pré-condition | Le joueur a lancé une nouvelle partie, ou chargé une partie déjà existante. |
| Post-condition | L’avatar du joueur se situe dans un nouvel environnement. |
| Scénarii de test | 1. Cliquer sur la flèche de gauche.   Réponse attendue : Le Tamagotchi se situe dans un nouvel environnement.     1. Cliquer sur la flèche de droite.   Réponse attendue : Le Tamagotchi se situe dans un nouvel environnement.     1. Cliquer sur la flèche de gauche, puis sur la flèche de droite.   Réponse attendue : Le Tamagotchi a changé d’environnement, et est retourné dans son environnement de départ.     1. Cliquer sur la flèche de gauche jusqu’à ce qu’elle disparaisse.   Réponse attendue : Au bout d’un certain nombre d’environnements, le Tamagotchi est obligé de faire demi-tour. La flèche de Gauche doit donc disparaître, car il ne peut plus avancer.     1. Cliquer sur la flèche de droite jusqu’à ce qu’elle disparaisse.   Réponse attendue : Au bout d’un certain nombre d’environnements, le Tamagotchi est obligé de faire demi-tour. La flèche de gauche doit disparaître, car il ne peut plus avancer. |
| Extensions |  |
| Contraintes |  |

|  |  |
| --- | --- |
| **Eléments de test de cas d’utilisation** | **Description spécifique au projet** |
| Cas de test | Test de réalisation d’une action avec l’avatar. |
| Acteurs | Usager |
| Evènements déclencheurs | Appuie sur l’icône d’action pour interagir avec l’avatar |
| Parties prenantes et leurs intérêts | * Usager : Effectue une action pour influencer les statistiques de l’avatar |
| Niveau | Stratégique |
| Portée | * nom : une chaîne de caractères qui définit l'appellation de l’avatar. * type : un type d’avatar compris dans l’énumération de types : CHAT, CHIEN, OISEAU, POULPE, ROBOT. * santé : entier compris entre 1 et 10 * bonheur : entier compris entre 1 et 10 * nourriture : entier compris entre 1 et 10 * hygiène : entier compris entre 1 et 10 * toilette : entier compris entre 1 et 10 |
| Pré-condition | Une partie doit être en cours.  L’avatar doit être créé.  Les boutons d’interactions doivent être disponibles. |
| Post-condition | Les statistiques de l’avatar sont mises à jour. |
| Scénarii de test | 1. Si l’environnement sélectionné est la salle à manger : le bouton d’interaction situé en bas de l’écran doit afficher le texte correspondant à l’action “Manger”. L’utilisateur doit cliquer sur le bouton. L’animation correspondant à l’action manger doit se déclencher. Réponse attendue : les statistiques suivantes doivent être mises à jour : Nourriture,    1. Si Nourriture > 50% alors Bonheur augmente 2. Si l’environnement sélectionné est le salon : le bouton d’interaction situé en bas de l’écran doit afficher le texte correspondant à l’action “Jouer”. L’utilisateur doit cliquer sur le bouton. L’animation de “Jouer” se déclenche.  Réponse attendue : les statistiques suivantes doivent être mises à jour : Divertissement,    1. Si Divertissement > 50% alors Bonheur augmente 3. Si l’environnement sélectionné est la chambre: le bouton d’interaction situé en bas de l’écran doit afficher le texte correspondant à l'action “Dormir”. L’utilisateur doit cliquer sur le bouton pour que l’animation se déclenche. Réponse attendue : les statistiques suivantes doivent être mises à jour :  Energie,    1. Si Energie > 50% alors Santé augmente 4. Si l’environnement sélectionné est la salle de bain : le bouton d’interaction situé en bas de l’écran doit afficher le texte correspondant à l’action “se Laver” . L’utilisateur doit sélectionner le bouton pour que l’animation se déclenche.  Réponse attendue : les statistiques suivantes doivent être mises à jour : Hygiène,    1. Si Hygiène > 50% alors Santé augmente |
| Extensions |  |
| Contraintes |  |

|  |  |
| --- | --- |
| **Eléments de test de cas d’utilisation** | **Description spécifique au projet** |
| Cas de test | Test de sauvegarde manuelle d’une partie. |
| Acteurs | Usager |
| Evènements déclencheurs | L’usager ouvre le menu et appuie sur le bouton de sauvegarde. |
| Parties prenantes et leurs intérêts | * Usager : Effectue une action pour sauvegarder sa partie |
| Niveau | Stratégique |
| Portée | * nom : une chaîne de caractères qui définit l'appellation de l’avatar. * type : un type d’avatar compris dans l’énumération de types : CHAT, CHIEN, OISEAU, POULPE, ROBOT. * santé : entier compris entre 1 et 10 * bonheur : entier compris entre 1 et 10 * nourriture : entier compris entre 1 et 10 * hygiène : entier compris entre 1 et 10 * toilette : entier compris entre 1 et 10 * lieu : valeur comprise dans l’énumération : JARDIN, SALLE DE BAIN, SALON, CHAMBRE. * date : la date système (sysdate) à laquelle la partie à été sauvegardée. |
| Pré-condition | Une partie doit être en cours.  L’avatar doit être créé.  Les boutons d’interactions doivent être disponibles.  La fonction de sauvegarde doit être fonctionnelle.  Le système de persistance des données doit être implémenté |
| Post-condition | La partie est sauvegardée dans un fichier situé dans l’arborescence de l’application. |
| Scénarii de test | On considère que le fonctionnement de la sauvegarde sera identique pour n’importe quel environnement du jeu.     1. La partie est en cours et l’utilisateur décide de sauvegarder sa partie.   Si il est en jeu et que son avatar est en action. Il peut ouvrir le menu et clique sur le bouton de sauvegarde. Il entre le nom de la sauvegarde. Il confirme la sauvegarde. Réponse attendue : il y à un temps d’attente et la partie est sauvegardée dans un fichier situé dans un dossier de sauvegarde dans l’arborescence du jeu. |
| Extensions |  |
| Contraintes |  |

|  |  |
| --- | --- |
| **Eléments de test de cas d’utilisation** | **Description spécifique au projet** |
| Cas de test | Test de sauvegarde automatique d’une partie. |
| Acteurs | Usager et Système |
| Evènements déclencheurs | 5 minutes se sont écoulées depuis le chargement de la partie ou depuis la dernière sauvegarde automatique. |
| Parties prenantes et leurs intérêts | * Usager : jouer au jeu sans interruptions et sans se soucier de sauvegarder la partie. * Système : sauvegarder la partie |
| Niveau | Stratégique |
| Portée | * nom : une chaîne de caractères qui définit l'appellation de l’avatar. * type : un type d’avatar compris dans l’énumération de types : CHAT, CHIEN, OISEAU, POULPE, ROBOT. * santé : entier compris entre 1 et 10 * bonheur : entier compris entre 1 et 10 * nourriture : entier compris entre 1 et 10 * hygiène : entier compris entre 1 et 10 * toilette : entier compris entre 1 et 10 * lieu : valeur comprise dans l’énumération : JARDIN, SALLE DE BAIN, SALON, CHAMBRE. * date : la date système (sysdate) à laquelle la partie à été sauvegardée. |
| Pré-condition | Une partie doit être en cours.  L’avatar doit être créé.  Les boutons d’interactions doivent être disponibles.  La fonction de sauvegarde doit être fonctionnelle.  Le système de persistance des données doit être implémenté |
| Post-condition | La partie est sauvegardée dans un fichier situé dans l’arborescence de l’application. |
| Scénarii de test | On considère que le fonctionnement de la sauvegarde sera identique pour n’importe quel environnement du jeu.  La partie est en cours et 5 minutes se sont écoulées depuis le chargement de la partie ou depuis la dernière sauvegarde automatique : la fonction de sauvegarde automatique se déclenche sans action de l’utilisateur.  Réponse attendue : un icône de sauvegarde automatique apparaît dans un coin de l’écran et la partie est sauvegardée dans un fichier situé dans un dossier de sauvegarde dans l’arborescence du jeu. |
| Extensions |  |
| Contraintes |  |