Journal de travail du travail de Bachelor N° 228

Aussi disponible sur la forge du projet : https://forge.ing.he-arc.ch/projects/he-arc-inf-1819-dlm-tb-213/wiki/Wiki

19.02.2019

- Contact avec Mr. Gobron et Mr. Senn.
- Récupération de gants haptique
- Recherche sur le domaine de l'haptique
- Début de rédaction du cahier des charges
- Introduction au LaTeX

26.02.2019

- Recherche sur le domaine de l'haptique
- Finalisation rédaction cahier des charges
- Rédaction d'un document sur la recherche
- Discussion avec Mr. Gobron --> Je dois commander 3 dispositifs haptiques différents pour en avoir 5 au total.
- Début des recherches des différents dispositifs haptiques

05.03.2019

- Etude sur le domaine de l'haptique
- Rédaction objectifs principaux cahier des charges
- Recherche et résumé du matériel --> Envoi d'un mail à Gobron, il sera difficile de commander 3 **dispositifs différents** car la plupart possèdent les mêmes caractéristiques ou se présentent sous la même forme.
- Suite du cours introduction au LaTeX (openclassroom)

12.03.2019

- Finalisation du cahier des charges --> Application de la signature ainsi que celle de Mr. Julien Senn
- Première utilisation des gants Sense Glove avec une application de leur entreprise --> Fonctionnel mais je n'ai pas eu le temps de bien tester le retour de force ou le hand tracking.
- Recherche de matériel +++ Il faut commander cette semaine
- Conversion de ma liste de dispositifs haptiques en tableau Excel !
- Finalisation du tableau (de 11h30 à 17h30)

19.03.2019

- Etat de l'art centré sur l'haptique en informatique !
- "Réunion" avec l'équipe de recherche du groupe Imagerie pour commander les gants ! OK
- Analyse des différents types de dispositifs (Gants, exosquelettes, joystick)
- Grosse rédaction sur mon document d'état de l'art
- Update du planning

26.03.2019

- Fin de l'analyse des différents types de dispositifs
- J'ai reçu les gants **Hi5**
- Essais et tests des SenseGlove avec leur application
- Essaie et tests des Hi5 avec leur application --> Pas réussi à les utiliser
- J'ai besoin de VIVE Tracker et de la station de base, il faut donc que j'aille sur un autre PC...
- Prochaine fois : Début de la rédaction concernant les Hi5 ou SenseGlove (à choisir encore)
- Avantages, inconvénients, comment les utiliser, comment utiliser le SDK, etc.

02.04.2019

- Début de la création d'un projet Unity. Je pars sur un concept simple. Une table sur laquelle sont posés plusieurs objets, de forme, taille, matériau différent
- Utilisation des SenseGlove : Chaque gant est représenté par un Prefab. Chaque objet sur lequel qui doit être ressenti doit avoir des scripts en fonctions de ses propriétés (Dur, mou, cassable ou pas, si on peut l'attraper ou pas, etc.)
- Importation d'assets pour le parc, la vue première personne et les objets.
- Comme les SenseGlove ne possèdent pas de handtracking, j'utilise la roulette de la souris pour les faire monter/descendre !
- Mr. Senn va checker pour un bracelet sur lequel je pourrais attacher un VIVE Tracker afin d'améliorer l'immersion grâce à un tracking précis.

09.04.2019

- Continuation du projet Unity :
- Les gants Sense Glove sont partiellement intégrés. Je peux les voir dans l'application, prendre des objets avec et ressentir ces derniers.
- Petit bémol à corriger, si j'active la mise à jour du poignet, les mains restent dans la même rotation que celle sauvegardé pendant la calibration et ne suivent pas la caméra.
- Si je désactive cette option (Update wrist), les gants suivent la caméra mais ne peuvent pas "bouger"
- Je vais faire des recherches plus approfondies sur l'haptique et Unity grâce à Google Scholar pour enrichir mon état de l'art et mes bibliographies.

15.04.2019 - 18.04.2019

- Recherche scientifique sur l'haptique
- Références
- Avancement de l’état de l’art

23.04.2019

- Recherche scientifique sur l'haptique
- Références
- Avancement de l’état de l’art

30.04.2019

- Conclusion de la partie état de l'art
- Tableau récapitulatif des dispositifs haptiques

- Planning de Gantt de la partie intégration
- Rapide test des gants Senso.
- Début des spécifications

20.05.2019

- Mise en place de la structure de l'application
- Recherche et tests avec les SenseGlove
- J'arrive à interagir avec un bâton, une pierre et une boule. La boule, une fois cassée, libère une pépite d'or avec laquelle on peut interagir
- Problème de texture avec la stressball

21.05.2019

- Continuation de l'intégration des gants SenseGlove
- J'ai envoyé un mail à la firme SenseGlove pour avoir des détails et explications sur l'utilisation et la fonctionnalité de certaines classes (celles agissant sur une rotation sur un certain axe, voir le wiki)
- La stressball fonctionne. Le problème venait de la manière dont cette dernière était reliée à la main quand elle était attrapée (ne pas utiliser la méthode **follow**)
- Pour l'ours en peluche, j'ai préféré laisser la méthode d'attrapage par défaut. Elle permet de caresser et serrer l'ours avec un maximum de ressenti. Par contre, sa texture mesh n'est pas 100% identique à l'image, pas grave.
- Il ne reste plus que le bouton, j'ai appliqué un matériel dur pour son cadre et laissé la sensation de vibration quand on appuie dessus.
- TODO, à l'appui du bouton, déployer une explosion faisant vibrer au maximum chaque vibrateur des 2 mains, à voir si ça marche.
- Essai d'utilisation de la classe TAP_FEEDBACK (permettant une vibration quand un objet tenu dans la main entre en collision avec un autre objet) --> Ce script ne marche pas (et j'ai bien tout essayé)

22.05.2019

- Tous les objets sont mis en place et fonctionnent
- Le baton en bois est dur (80%) avec un ressenti de texture provoqué par des faibles vibrations
- La pierre est à la dureté maximale sans vibrations
- La boule "cassable" se casse avec minimum 3 doigts et fait tomber une pépite en or
- Le bouton "explosion" provoque une explosion faisant vibrer graduellement chaque doigt en partant de 100 jusqu'à 0 (pour augmenter la sensation de ressenti avec l'explosion)
- La stressball est prenable, le ressenti fonctionne et la déformation de texture aussi
- L'ours en peluche est prenable, le ressenti fonctionne (plus mou que la stressball) et la déformation de texture aussi
- Ajout d'un 2ème gros bouton (avec animations) dont l'activation se fait avec minimum 3 doigts. Tant que les colliders sont activés, des vibrations se font ressentir aux doigts
- TODO : Ajout d'un effet pluie au clic sur le 2ème gros bouton --> chaque goutte d'eau qui touche un doigt provoque une vibration au doigt touché

23.05.2019

- Avancement sur le document SENSEGLOVE - DETAILS.
- Description du SDK SenseGlove et de ses classes principales.
- Effet de pluie généré par des particules (ParticleSystem) --> ne collident pas avec le bout des doigts car le capsuleCollider des doigts a son isTrigger à true.

- Création d'un 2ème collider pour chaque doigt (au même endroit) puis ajout d'un rigidBody et d'un capsuleCollider dont le isTrigger est à false.

24.05.2019

- Finalisation de la scène SenseGlove_Park
- La collision entre les particules de pluie et les mains (doigts et paume) se fait correctement grâce au 2ème collider placé à chaque bout de doigt.
- La calibration par la méthode Interpolation_Flexion fonctionne. C'est celle appliquée par défaut. Les autres méthodes ne fonctionnent pas bien. La team travaille dessus.
- Test de connexion / déconnexion de gant en temps réel
- Avancement du rapport

27.05.2019

- SenseGlove --> Fini
- Rédaction partie calibration des SenseGlove (dans le document)
- Commencement de l'utilisation des gants Senso
- Installation du driver
- Documentation
- Premiers tests, première prise en main, première lecture du SDK
- Les gants Senso se positionnent automatique l'un sur l'autre dans l'environnement virtuel. J'ai donc modifié le SDK pour passer une valeur statique afin que les gants soient bien positionnés à droite et à gauche.
- Premiers tests de vibration, comme pour les Sense Glove, j'ai ajouté un Collider à chaque bout de doigt. Dès que celui-ci entre en collision avec la table (mon premier objet), le doigt collisionné vibre

28.05.2019

- Ajout des Senso dans le projet
- Ajout de 2 colliders pour chaque doigt (un avec isTrigger à true + script CollideManager et un avec isTrigger à false et un Kinematic Rigidbody)
- Le contact avec les objets marche
- Bouton 1 : Explosion --> Vibration graduellement jusqu'à 0
- Bouton 2 : Pluie --> Chaque doigt touché est vibré
- A faire : tester la température

29.05.2019

- La température ne marche pas.
- Les gants sont instables (déconnexion, latence) quand on effectue beaucoup d'action d'un coup
- Au touché d'un doigt avec la pierre, cette dernière se colle au doigt de façon à pouvoir la déplacer (c'est moche mais ça démontre le contact entre le doigt et l'objet)
- Refactorisation, commentaires XML

03.06.2019

- Les gants Senso sont intégrés. Nous notons leurs instabilités face aux "grosses" actions (beaucoup de vibrations, de mouvements)
- Début de l'utilisation des Hi5 Glove
- La calibration nécessite OBLIGATOIREMENT un Vive tracker sur chaque gant !
- Tous les exemples fournit par la firme Noitom nécessitent un casque de réalité virtuel (HTC Vive de préférence)

- Donc problématique pour mon projet, je devrai probablement intégrer les VIVE Tracker pour chaque gant
- Première intégration, les gants sont reconnus et le mouvement du poignet/doigts aussi. Je ne peux cependant pas les calibrer car je n'ai pas de VIVE tracker sur mon PC.

04.06.2019

- Suite des gants Hi5
- L'intégration se fait très mal. Obligation d'utiliser le HTC Vive et les Vive Tracker. Je modifie donc la structure de ma 3ème scène pour qu'elle supporte les accessoires de réalité virtuelle.
- Je peux connecter mes Hi5, les voir dans mon parc. Je peux les utiliser avec mes premiers objets à disposition : une pierre, une balle de foot et un cube en or
- La calibration est problématique ! Il y a trop d'interférences magnétiques dans la zone donc les gants se calibrent très, très mal !
- Je peux passer de ma scène de parc à ma scène de calibration en appuyant sur F2, je vais essayer de changer en mettant ça sur le clic d'un bouton.
- Documentation et SDK très incomplets ! La documentation suggère l'utilisation de classes donc certaines sont... VIDES ! (oui oui, pas de code dedans)
- Très mal commenté, je pense que je vais plutôt utiliser mes classes pour l'activation du clic sur le bouton.

05.06.019

- Suite des gants Hi5
- Pour l'intégration, j'ai suivi leur documentation d'intégration (SDK_Interaction) combiné à quelques modifications de ma part
- La calibration est très imprécise quand on la fait avec les 2 gants ensemble (malgré une démagnétisation)
- J'ai pu mettre en place ma scène avec les objets (Pierre, pépite d'or, balle de foot, 2 cubes de tailles différentes, 3 petites balles et 2 buts dans lesquels il faut lancer les balles)
- La prise en main d'un objet possédant un MESH COLLIDER est difficile et imprécise si cet objet est petit (pour différencier ce cas, j'ai mis une grosse pierre et une petite pépite, tous deux avec le même mesh collider)
- Les objets de type simple (cube, sphère, baton) se prennent en main sans difficulté
- TODO pour le jour suivant : Finaliser les buts (feux d'artifices, pluie, etc.), ajouter les informations de connexion (1/2 gants connectés, etc.). Ajout d'un bouton pour passer à la scène de calibration et changement du bouton RESET

06.06.2019

- Suite des gants Hi5
- Les deux buts sont finis. Il faut lancer un objet (n'importe lequel) dans un des buts pour activer son effet (feu d'artifice / pluie).
- La collision d'une goutte de pluie sur la main déclenche une vibration. J'ai dû ajouter un collider (option isTrigger à false) à chaque paume de la main.
- 3 boutons sont disponibles : bouton explosion, bouton pour passer à la scène de calibration et bouton pour reset les objets de la table.
- Au fur et à mesure de l'utilisation des gants dans la journée, ces derniers deviennent instables (crash de l'application Unity)
- Début de la création d'un panneau servant à afficher du texte (information de connexion, batterie des gants, utilisation, etc.). Ce panneau sera posé à gauche de la table.
- TODO pour le jour suivant : Finaliser et placer le panneau, effectuer les derniers tests d'utilisation des gants. Tester les différentes méthodes de prise d'un objet (pinch, lift, clap)

07.06.2019

- Finalisation des gants Hi5
- Le panneau est placé avec les informations sur les gants (Batterie, magnétisation, connecté ou pas).
- Test finaux (avec Mr Senn).
- J'ai dû réduire la masse de tous les objets car plus la masse est grande, moins la collision est propre (le doigt "rentre" dans le gameObject quand on veut pousser le cube par exemple)
- J'ai bien mis les propriétés des RigidBody de mes objets (les propriétés Interpolate et Collision Detection) car les valeurs par défauts rendaient la collision indescriptible quand on lançait un objet et qu'on le rattrapait.
- Mr. Senn m'a proposé d'essayer de "bloquer" les doigts virtuels au contact de ces derniers avec un objet : Après quelques recherches, c'est impossible. En effet, la main virtuelle est directement reliée aux capteurs des gants. Il faudrait donc bloquer les doigts IRL pour avoir un effet visuel de blocage dans l'application.
- TODO : Empêcher l'utilisateur de traverser la table quand il rentre en collision avec.
- TODO : Les Hi5 étant finis, je vais repasser aux SenseGlove en intégrant le casque HTC Vive et les VIVE Tracker à la scène.

11.06.2019

- Hi5 finis
- Début de l'amélioration et intégration des Vive Tracker et HTC Vive avec les Sense Glove
- HTC Vive et Vive Tracker intégrés
- J'ai dû faire une rotation de 90° des gameobject gloves car le vive tracker est positionné de façon horizontale sur le gant haptique
- Mise en place du board (informations de connection et calibration)
- TODO : Bouton calibration à finir. Régler problème de mouvement de poignet (soit désactiver le poignet, soit refresh). Buts, bouton explosion, lancer les objets

12.06.2019

- Suite des gants SenseGlove
- J'ai finalisé l'ajout des Vive Tracker et du casque HTC Vive. Seul problème, les ViveTracker s'instancie aléatoirement et les gants doivent être placés en tant que *child* du Vive Tracker associé.
- Bouton Reset OK. Un bug apparaît aléatoirement ou les objets ne se réinitialisent pas correctement. A voir...
- Système de calibration (par le biais du panneau) marche. Reste à finir en testant si 1 ou 2 gants sont connectés et, dans le dernier cas, lancer la calibration des 2 gants en même temps.
- Pour la calibration, au clic du bouton concerné, le joueur a 2 secondes pour mettre sa main droite afin que le poignet soit calibré, la calibration des doigts commence ensuite.
- Test de lancer d'objets. C'est moche car le SDK des SenseGlove demande l'ouverture de la main pour lâcher l'objet, à voir si j'ai le temps de modifier ça...
- Pour tous les objets : releaseMethod : Must Open Hand !
- Pour améliorer le lancer, j'ai multiplié la vélocité d'un objet grabable par 1,8 (dans le script SenseGlove_Grabbable du SDK)
- Ajout des 2 buts. Comme le lancer est beaucoup plus difficile que pour les Hi5, je les ai rapprochés de la table.
- Le lancer fonctionne mais il faut bien comprendre le principe. L'utilisateur doit bien ouvrir complètement la main (et donc ne plus avoir de collider en contact avec l'objet) juste avant la fin du mouvement du lancer. C'est compliqué

- On voit que ces gants n'ont pas été développés pour des applications ou l'utilisateur fait des mouvements vastes, contrairement aux Hi5.
- Par contre, les gants sont très solides. Aucune latence, pas de lag, le ressenti est bien réel.
- Gros problème, câblage encombrant
- TODO : Modifier les objets pour un rendu optimal (prise en main, lancé, etc.), Ajout de 2 autres balles (comme Hi5_Scene), Finaliser la calibration (quand 2 gants sont connectés et quand 1 gant est connecté)
- TODO : Refactorisation

13.06.2019

- Suite des gants SenseGlove
- J'ai modifié le rendu (prise en main et lancé) pour que ça soit plus simple mais la logique défini par le SDK de permet pas un lancer autant interactif qu'avec les Hi5
- Pour lancer un objet, il faut qu'il soit bien en main et effectuer un mouvement du bras plutôt que du poignet, à la fin du mouvement, il faut ouvrir complètement la main et ne plus avoir de contact avec l'objet pour que ce dernier soit lancé
- Finalisation de la calibration (Avec 2 gants connecté, la calibration se lance d'abord pour la main droite puis pour la main gauche)
- Bouton reset mis à jour. En effet, j'ai rencontré un gros problème qui venait quand un objet était pris puis relâché. Sa place en tant qu'enfant dans le container des objet n'était plus la même qu'au démarrage
- Après plusieurs heures à essayer de checker et la place et l'objet, j'ai trouvé une solution en utilisant les dictionnaires
- J'ai un dictionnaire<Transform, Position> et un dictionnaire<Transform, rotation>. Au clic du bouton, je parcours donc tous mes objets interactables et reset leurs position/rotation grâce aux dictionnaires
- Seul bémol, les objet *breakables* ne se reset pas à cause de leur méthode reset propre à eux (implémentée dans le SDK). J'ai donc préféré les laisser en kinematic pour éviter qu'ils ne partent n'importe où. En les laissant en kinematic, on aura toujours accès à ces objets.
- Ajout d'un dernier objet. Une sorte de cylindre jaune (posé à côté du cube noir) afin de ressentir la différence de forme
- Refactorisation complète des scripts senseGlove (les miens) avec les commentaires XML et suppressions de lignes inutiles
- TODO : Derniers tests puis redébut des SensoGlove

17.06.2019

- Début des gants Senso
- La prise d'un objet avance, ça reste moche mais je ne peux pas rendre l'action parfaite, je préfère me focaliser sur la logique d'interaction (vibration, ressenti, etc) des gants
- La température est là que pour les tests (confirmé par la team de développement)
- Le bouton Reset et le board ont été mis en place (board affiche la batterie des gants + informations)
- La scène n'a pas besoin de grands changements
- TODO : Finaliser la prise d'un objet, tester le lancer (sans Vive tracker ça va être impossible mais le faire avec une scène d'exemple et la souris comme point d'ancrage au lieu de la main)

19.06.2019

- Suite des gants Senso
- La prise en main d'un objet fonctionne en utilisant le pouce et l'index
- Test de la distance entre ces 2 doigts pour prendre/lâcher un objet
- L'algorithme de lancer fonctionne (vélocité, vitesse, vecteurs)

- La scène est finie.
- Seul problème, les gants dépendent d'une application externe, après plusieurs heures de recherches, impossible de diviser cette application pour l'intégrer directement à Unity
- TODO : Recherche sur l'application, serveur, protocole (TCP/UDP), etc.

20.06.2019

- Suite des gants Senso
- Finalisation de la scène, check si les application SENSO_BLE_SERVER et SENSO_UI sont démarrés sur le PC
- Si elles sont démarrées, lance le test de connexion des gants et affiches les informations sur le board
- Analyse des paquets : IMPOSSIBLE
- En 1 après-midi, je n'ai pas réussi à capturer une seule trame en utilisant Wireshark...
- Utilisation d'une application Python (https://github.com/agaryen/sensoglove) pour analyser les gants et les sockets
- Je ne peux pas créer ma propre application SENSO_UI, je suis obligé d'utiliser celle fournie par la société
- TODO : Suite des analyses, intégration du caste HTC VIVE

21.06.2019

- Fin des gants Senso
- J'ai intégré le casque HTC Vive. Il ne manque plus que les Vive Trackers mais j'ai besoin d'un bracelet pour les attacher aux poignets.
- Comme Nicolas Sommer m'a dit que les Avatar VR arrivaient bientôt, j'ai préféré arrêter mes recherches sur la connexion client-serveur des gants Senso
- J'ai pu utiliser l'application python (lien section précédente) pour analyser les gants.
- Conclusion : Aucune modification des socket n'est possible. Aucune modification de l'application SENSO_BLE_SERVER ou SENSO_UI n'est possible.
- Pour pas perdre de temps, j'ai commencé l'intégration des Falcons aujourd'hui.
- Début des Novint Falcons
- Aucun SDK disponible, j'ai dû utiliser le package fournit par *kbogert* (https://github.com/kbogert/falconunity) me permettant de connecter les falcons au PC (via falconserver.exe)
- Importation du package dans ma nouvelle scène.
- Début des tests, les falcons sont reconnus et visibles dans Unity.
- Je mets tout de suite en place le board sur lequel je test la connexion des 2 appareils. Comme le serveur doit être obligatoirement lancé, je créé une méthode qui le lance si ce dernier n'est pas en cours d'exécution.
- L'affichage des batteries (batterie pleine si le falcon est connecté, batterie vide s'il est déconnecté) fonctionne.
- L'interaction entre le falcon et un gameObject semble se faire par le biais du script FalconRigidBody.cs
- Je préfère laisser l'image du falcon comme une sphère (dans Unity) et non comme une main.
- TODO : Continuer d'utiliser les falcons pour comprendre leurs fonctionnement, mettre en place la prise d'un objet (appuyer sur un bouton pour le prendre, relâcher le bouton pour lâcher l'objet)
- TODO : Améliorer la physique (certains objets flottent quand on les touchent...)

24.06.2019

- Suite des Falcons
- J'ai créé un script (Falcon_PickupManager) qui s'occupe de la prise d'un objet.

- L'algorithme fonctionne, j'ai voulu le mettre dans le script FalconRigidbody (script fournit avec le package gérant le ressenti entre l'objet et le falcon, poids, force, rugosité, etc.) mais cela ne marche pas bien (physique moche, je n'arrive pas à lancer l'objet, etc.)
- La prise et le lancer se passent comme ça :
    - Si le falcon touche l'objet et le bouton du milieu est pressé, l'objet se positionne à la place du falcon (dans le jeu hein...)
    - Le falcon récupère la masse de l'objet est l'applique dans que l'objet est pris (ressenti du poids de l'objet)
    - Au clic sur le même bouton (le SDK ne détecte pas le déclic d'un bouton), l'objet sera lancé. Je désactive le script FalconRigidbody car il interfère beaucoup trop avec la physique du jeu (l'objet ne tombe pas, c'est comme s'il était dans l'espace avec un frottement de 0...)
    - L'objet est donc lancé. Quand ce dernier atteint une vitesse de 0, je recharge le script FalconRigidbody
- Problème : la gravité n'existe pas. Le script FalconRigidbody agit comme si les objets étaient dans l'espace. Si je pousse un objet, ce dernier va avancer continuellement...
- TODO : Améliorer la physique, améliorer la prise d'un objet/lancer (essayer de détecter quand le bouton est relâché)
- TODO : Ajouter la gravité
- TODO : Si je n'y arrive pas, laisser les objets sans masse (cela les empêche de bouger mais permet le ressenti de rigidité et le retour de force)

25.06.2019

- Suite des Falcons
- Impossible de détecter quand un bouton est relâché... La prise d'un objet se fait donc comme cela : 1 clic = prise "activé", quand on se rapproche d'un objet *grabbable*, il sera automatiquement attrapé. Un nouveau clic sur le bouton permettra de désactiver la prise et relâchera donc l'objet (qui sera lancé en fonction du mouvement)
- En appliquant la gravité aux objets, ces derniers subissent une physique plutôt réaliste. J'ai donc laissé cette option (gravité du script FalconMain à -9.81 en Y). Les objets ne volent plus indéfiniment.
- Bouton explosion mis en place. Il applique une suite de force aléatoire sur les Falcons les faisant vibrer dans tous les sens.
- Bouton reset mis en place. Le reset ne fonctionne pas bien car les objets possèdent à la fois des données sur Unity (mesh, position, rotation, etc.) et dans le serveur (FalconServer). Il faut donc que je remédie à cela.
- TODO : Finir le bouton reset (essayer de supprimer la forme dans le serveur, replacer l'objet dans Unity et recréer la forme dans le serveur)
- TODO : Mettre en place les bons objets (inutile de mettre 3 boules comme pour les gants, 1 suffira.)
- TODO : Objets potentiels : Des sphères avec des textures pour bien ressentir les propriétés haptiques du falcon (ressenti de texture), cubes de différentes masses/tailles, Ours en peluche avec un effet "mou"

26.06.2019

- Suite des Falcons
- Le bouton reset est fini (avec la solution proposée)
- Les différents objets sont mis en place !!
- Attention, si trop d'objets ont le script Falcon_Rigidbody, le serveur lag et les falcon ne suivent plus

- J'ai donc enlevé le script pour tous les objets style arbre, roche, fontaine, bancs, etc. Seul les sols et l'eau ont ce script (et les objets interactables bien sûr)
- J'ai mis 3 sphères avec : Une texture rugueuse, une mi-rugueuse et une lisse pour bien sentir la différence
- J'ai fini l'instanciation des objets interactables, comme c'est des Falcons, je n'ai pas mis autant d'objet que pour les autres gants.
- J'ai cependant mis 2 pierres (masse et poids différents), 2 cubes (idem), 2 boules (même masse, utilisé pour les lancer dans les buts), et les 3 grosses sphères mais elles ne bougent pas.
- TODO : Intégrer le casque HTC Vive

28.06.2019

- Fin des Falcons
- J'ai intégré le casque HTC Vive (comme pour les autres gants, rien de très compliqué)
- J'ai ajouté un dernier objet, une sorte de mur (en taille réduite) non attrapable. Le but de cet objet est de le faire tomber grâce la force appliqué aux falcons.
- Début des AvatarVR, installation des drivers et tests des gants avec l'application NDSuite.
- Les gants fonctionnent bien, les moteurs de vibration aussi
- Gros problème... le SDK coûte 2500€ (pour 1 an) et j'en ai besoin pour intégrer les gants à mon projet, je l'ai donc dit à Mr. Senn et il me tient au courant pour la suite.
- En attendant, j'ai commencé la rédaction du document sur les SenseGlove
- Ce document contiendra : la description des gants, leurs caractéristiques et propriétés, la description du SDK (décrire comment il marche et les classes principales). J'ajouterai ensuite comment intégrer les gants à une application Unity et finalement j'expliquerai les problèmes rencontrés.
- Lundi je serais absent car j'ai un entretien d'embauche à 14h (je préfère me préparer le matin)
- TODO : Continuer le document (si la license des AvatarVr n'a pas été achetée)

02.07.2019

- Suite du document sur les SenseGlove
- Refonte complète de la description des gants
- Description complète du SDK (description des scripts principaux, leurs utilisations et objectifs)
- Description de la calibration (Comment elle fonctionne et les différentes méthodes de calibration)
- Début de la rédaction mode d'emploi d'intégration.
- Ce chapitre sera présent pour chaque gant et décrira étape par étape comment intégrer les gants à n'importe quel projet Unity. Ce chapitre sera plus ou moins long en fonction des gants utilisés.
- Je décris aussi dans ce chapitre comment interagir avec les objets de la scène (si le système d'intégration est compris dans le SDK). Ce sera généralement une suite de script à ajouter et configurer aux objets avec lesquels nous souhaitons interagir.
- J'ai vu dans le SDK qu'il y avait un prefab SenseGloveHand - Left VR / Right VR. Ils contiennent le script Teleport, permettant à l'utilisateur de se téléporter à un endroit du monde virtuel si ce dernier est pointé du doigt.
- Cette fonction est inutile dans notre cas, la table étant statique, l'utilisateur n'aura pas à se déplacer dans la map.
- J'ai par la même occasion re-testé les gants, tout marche toujours niquel.
- TODO : Suite de la documentation (fin du chapitre mode d'emploi intégration + problèmes rencontrés)

03.07.2019

- Suite du document sur les SenseGlove
- J'ai fini le mode d'emploi d'intégration (de manière très basique). Pour plus d'infos, j'ai ajouté le lien vers le wiki du dépôt GitHub du SDK.
- En faisant des tests, j'ai cassé un SenseGlove (le droite)... Les câbles et connexions marchent encore. J'ai pu le "réparer" en mettant beaucoup de scotch. Il ne faut pas forcer l'appuie sur le pouce si ce dernier est bloqué par le retour de force.
- La scène fonctionne toujours bien, j'ai séparé la table en deux. A gauche les objets qui ne sont pas fait pour être lancé, même si on peut quand même les lancer... (boutons, objet pouvant être cassé, ours en peluche), et à droite, les objets pouvant être lancé.
- TODO : Reprendre les Senso et commencer le document les décrivant

04.07.2019

- Suite du document sur les Senso
- J'ai repris les mêmes points que le document sur les SenseGlove.
- Description
- Propriétés / caractéristiques
- Première connexion au gant (décrit les étapes d'installation du driver + lancement et configuration du SENSO_BLE_SERVER et SENSO_UI
- Calibration
- Description SDK
- Architecture (diagramme UML simplifié avec les classes principales + description)
- Mode d'emploi d'intégration (attention, juste intégration. Les gants Senso ne sont pas fait pour des interactions complexes avec les objets. D'ailleurs, aucun script permettant une interaction n'est inclus dans le SDK)
- Re test de la scène Senso_Park
- Tout marche encore bien, j'ai refactorisé 2-3 bouts de codes, supprimé quelques fonctions inutiles (sendVibro 2x dans le HandsController)
- Il manque plus que l'intégration des Vive Tracker, j'attends toujours le bracelet...
- TODO : Commencer le document sur les Hi5 !

08.07.2019

- J'ai fait le menu d'accueil
- Ajout d'un bouton sur chaque scène permettant de revenir au menu
- Tests avec les SenseGlove --> Bug : les SenseGloves ne sont pas reconnus au rechargement de leur scène respective. Après de longues recherches, j'ai envoyé un mail à la team car quelque chose cloche vraiment.
- Tests avec les Hi5 --> OK. La scène se charge correctement et je peux switcher entre les scènes sans erreur.
- Hi5 : Ajout du code contrôlant la connexion des gants (pour l'affichage sur le board) dans Hi5_InteractionManager.
- Refactorisation complète (button_1 en button_calibration/explosion, etc.)
- SenseGlove : le chargement des gants ne se fait pas correctement si les Vive Tracker sont chargé différement.
- TODO : Charger les Vive Tracker correctement
- TODO : Charger les SenseGlove correctement au changement de scène (essayer le DoNotDestroy sur pratiquement tous les objets...)
- TODO : Test des Senso et Falcon

09.07.2019

- Pour charger les Vive Tracker correctement, j'ai ajouté un menu permettant de sélectionner les ID (de 1 à 4, car il y a 2 bases Stations et 2 VIVE Tracker)
- Ce menu est dispo que pour les SenseGlove et les gants Senso (mais je n'ai pas encore pu tester les Vive Tracker dessus...)
- Impossible de charger correctement une 2ème fois les Senso Glove, à mettre dans le rapport.
- Test des Senso et Falcon OK
- Nettoyage complet du projet (Suppression de fichiers inutiles, refactorisation, etc.)
- Pour les Falcon, ajout d'un menu ou l'utilisateur sélectionne le chemin du FalconServer
- Idem pour les Senso mais ça ne va pas jouer, en effet, je voulais faire choisir le script et le lancer depuis Unity mais les chemins ne correspondent pas. Le script est lancé depuis le répertoire Unity et il ne trouve donc pas les app SERVER et UI.
- TODO : Document Hi5 et Falcon
- TODO : Test des Avatar VR !!!

10.07.2019

- Test des Avatar VR, ils fonctionnent (avec la démo)
- J'ai passé la matinée à finaliser les gants (notamment les Senso avec les Vive Tracker)
- Finalisation de la scène Falcon aussi
- Problème avec les Falcons, le serveur ne se ferme pas correctement (Du coup, au rechargement de la même scène, les falcons ne marchent pas). J'arrête après 3h passé dessus sans avancer, le problème vient probablement du serveur (qui ne se ferme par correctement).
- Commencement des documents Hi5
- TODO : Document Hi5 et Falcon

11.07.2019

- J'ai modifié les documents sur les SenseGlove, Senso et Hi5
- J'ai commencé le document sur les Falcon, il reste la partie intégration
- TODO : Document sur les Falcons
- TODO : Etude comparative

12.07.2019 – 26.07.2019

- Rapport
- Documents SENSEGLOVE – DETAILS
- Document SENSOGLOVE – DETAILS
- Document HI5GLOVE – DETAILS
- Document FALCON – DETAILS
- Etude comparative des dispositifs haptiques
- Conversion en LateX