

Rapport de laboratoire : Travail de session

Implémentation d'un environnement virtuel permettant d'étudier l'impact de la musique sur les réflexes et la coordination motrice

Cours	MTI860	
Session	A-2022	
Numéro de votre équipe	7	
Étudiant(s)	<i>Sordoillet Olivier</i> <i>Viera Baptiste</i>	<i>SORO07059803</i> <i>VIEB92370001</i>
Date	10 décembre 2022	

GRILLE DE CORRECTION DU RAPPORT

Introduction		/0.5
Description de l'environnement		/1
Hypothèses		/2
Implémentation		/3
Protocole expérimental		/5
Résultats		/5
Discussion		/3
Conclusion		/0.5
Total partiel		/0.5
<i>Orthographe et grammaire (-10% max)</i>		
<i>Présentation (-10% max)</i>		
<i>Retard (-10% par jour)</i>		
Note du rapport / 20		/20

Table des matières

1.	Introduction.....	6
2.	Description de l'environnement	7
3.	Hypothèses	8
4.	Implémentation	9
	Environnement.....	9
	Interface	10
	Boutons du menu.....	10
	Boutons du Batak.....	11
	Scripts	12
	Traitement des données.....	14
5.	Protocole expérimental	15
	Protocole.....	15
	Participants	15
	Matériel	16
6.	Résultats.....	17
	Questionnaire.....	21
	Sentiment d'immersion.....	21
	Perception de leur performance	22
7.	Discussion.....	24
	Analyse des résultats	24
	Expérience de VR	24
	Formulaire	24
	Limites.....	24
8.	Conclusion	26

9. Références	27
Hypothèses	27
Assets Unity Version Heavy	27
Assets Unity Version Light & Heavy	27
Images	27
Documentation Oculus	27
Ecrire / Lire un fichier sur l'Oculus	28
Annexes	29
Annexe A : Script Python	29

Tables des Figures

Figure 1:Exemple d'interface scrollable, ici le tutoriel	9
Figure 2: Capture d'écran de l'interface de sélection de la musique	10
Figure 3: Capture d'écran du menu de sélection des options	10
Figure 4: Vue éclatée du bouton du Batak.....	11
Figure 5: Fonction RestartPlay dans SceneManage.....	12
Figure 6: Capture d'écran des fonctions d'écriture utilisées pour exporter les données du casque.....	13
Figure 7: Diagramme résumant l'expérience des participants sur la VR	15
Figure 8: Temps moyen entre chaque activation de boutons selon le genre de musique	17
Figure 9: Score moyen en fonction de la musique	18
Figure 10: Ecart type des scores en fonction du genre de musique	18
Figure 11: Délai moyen entre chaque activation de bouton sur la musique calme	19
Figure 12: Délai moyen entre chaque activation de bouton sur la musique rap	19
Figure 13: Délai moyen entre chaque activation de bouton quand il n'y a pas de musique	20
Figure 14: Sentiment d'immersion dans le jeu	21
Figure 15: Les participants avaient unanimement fortement l'impression d'être dans un endroit réel et non virtuel	21
Figure 16: Sentiment de bien-être des participants lors de l'expérience	22
Figure 17: Perception de leur propre performance en fonction de la musique	22
Figure 18: Impact de la musique sur leur performance telle que perçue par les participants	23

1. Introduction

Il a été demandé de réaliser au cours de cette session un environnement virtuel, dans lequel l'utilisateur peut interagir avec ses mouvements à l'aide d'Unity. Cet environnement virtuel doit pouvoir servir de base afin de réaliser une acquisition de données avec des participants en suivant un protocole expérimental qui est détaillé ci-après dans ce rapport. Une fois l'acquisition des données complétée, une analyse desdites données est réalisée afin de répondre à l'hypothèse suivante : la musique influence-t-elle les réflexes et si oui, dans quelle mesure ?

Ce rapport est composé de la présente introduction, suivie d'une description de l'environnement, puis des hypothèses. Ensuite, nous présenterons l'implémentation de l'EV, qui décrira le fonctionnement détaillé de l'application, avec l'appui de captures d'écran. Nous trouverons ensuite le protocole expérimental, les résultats, une discussion et enfin une conclusion.

2. Description de l'environnement

L'environnement qui a été recréé a pour but de simuler l'expérience du Batak. Le Batak consiste en une structure fixe avec des boutons situés à différents endroits. Le but est d'appuyer sur le bouton allumé le plus rapidement possible, ce qui déclenchera l'allumage d'un autre bouton aléatoire, etc... Une fois un certain temps écoulé, la session s'arrête et un score est affiché dépendant du nombre de boutons appuyés lors du laps de temps alloué.

Lors de l'étude, nous nous intéressons à l'influence de la musique sur les performances des participants. Nous émettons l'hypothèse que la musique influe bien leur temps de réaction, cependant de manière différente en fonction du type de musique (rock vs classique par exemple). En créant un Batak dans un environnement virtuel, on peut ainsi tester directement l'impact de la musique sur les réflexes des participants à l'aide de leur score ainsi que du délai entre chaque activation de bouton.

Afin de ne pas déconcentrer le participant, nous avons implémenté un environnement relativement neutre, sans animation, à l'apparence futuriste blanc.

3. Hypothèses

Dans cette étude, nous nous demandons quel impact pourrait avoir la musique sur les performances des participants à la simulation du Batak. Nos premières réflexions nous menaient au fait que l'apport de musique rythmée améliorerait le score des participants, et qu'à l'inverse une musique calme au tempo lent aurait des effets négatifs sur leur score.

L'hypothèse se base sur plusieurs recherches effectuées sur la relation entre musique et temps de réaction. De très nombreuses études se sont déjà penchées sur la question, avec des angles d'approches différents, notamment celui de la sécurité routière. [Welz et son équipe \(2020\)](#) ont trouvé que les différences entre un groupe qui écoutait une radio et un groupe qui écoutait de la musique n'étaient pas significatives sur le délai de freinage lors de la conduite. En revanche, ces deux groupes avaient en moyenne un temps de réaction plus élevé que le groupe de contrôle (sans musique ni radio). De plus, [Paridon & al. \(2012\)](#) ont pu observer que le délai entre un son qui représente un danger durant la conduite et le freinage était significativement augmenté lorsque le sujet écoutait de la musique, particulièrement avec un casque ou lorsque le volume était élevé. Le temps de réaction des participants se retrouvait bien impacté par la présence ou non de musique.

D'autres études, comme celle de [Edworthy & al. \(2006\)](#) ont montré que la musique et plus particulièrement son tempo et son volume avait un réel impact sur les performances des participants sur un tapis de course, aussi bien du point de vue subjectif (le participant se sentait plus performant et moins fatigué) que du point de vue physiologique (impact sur la fréquence cardiaque notamment). Cette observation est surtout vraie pour le tempo de la musique, ainsi que, dans une moindre mesure, le volume de cette dernière.

Nous avons donc les deux études de Paridon & al. (2012) ainsi que Welz & al. (2020) qui montrent un impact significatif de la musique sur le temps de réaction. Bien que ces études soient dans le domaine de la conduite, on est en droit de se demander si cet effet peut se transposer au Batak et si la musique peut aussi affecter les réflexes dans le cadre d'une simulation de Batak.

Bien que ce dernier repose sur les réflexes, il n'en reste pas moins une activité physique (la rapidité et l'endurance sont également des facteurs clés) et c'est pourquoi l'expérience d'Edworthy & al. (2006) est intéressante. Si la musique réduit la sensation de fatigue et augmente l'effort effectué, il peut être intéressant de voir l'impact (positif ?) qu'il peut avoir sur la simulation du Batak.

4. Implémentation

Environnement

L'environnement virtuel est constitué de deux pièces à l'apparence futuriste blanches. L'une, la plus petite est utilisée pour le menu d'accueil. Celui-ci est composé de 4 boutons, l'un pour lancer une partie, l'autre pour afficher un tutoriel, le troisième pour gérer quelques paramètres et enfin le dernier bouton permet de quitter l'application. Si l'utilisateur choisi d'afficher le tutoriel, il lui sera affiché une fenêtre interactive où il peut scroll afin de faire défiler les instructions, comme montré sur la figure 1.



Figure 1:Exemple d'interface scrollable, ici le tutoriel

Lorsque l'utilisateur décide de lancer une partie à la place, il est mis en face d'une interface simple où il peut choisir la musique, qui lui sera indiqué par la personne en charge de l'expérience. Une autre version de ce menu dans la partie *heavy* propose également de changer l'apparence de l'environnement et des mains (dans l'apk remis, il s'agit encore d'un travail en cours).

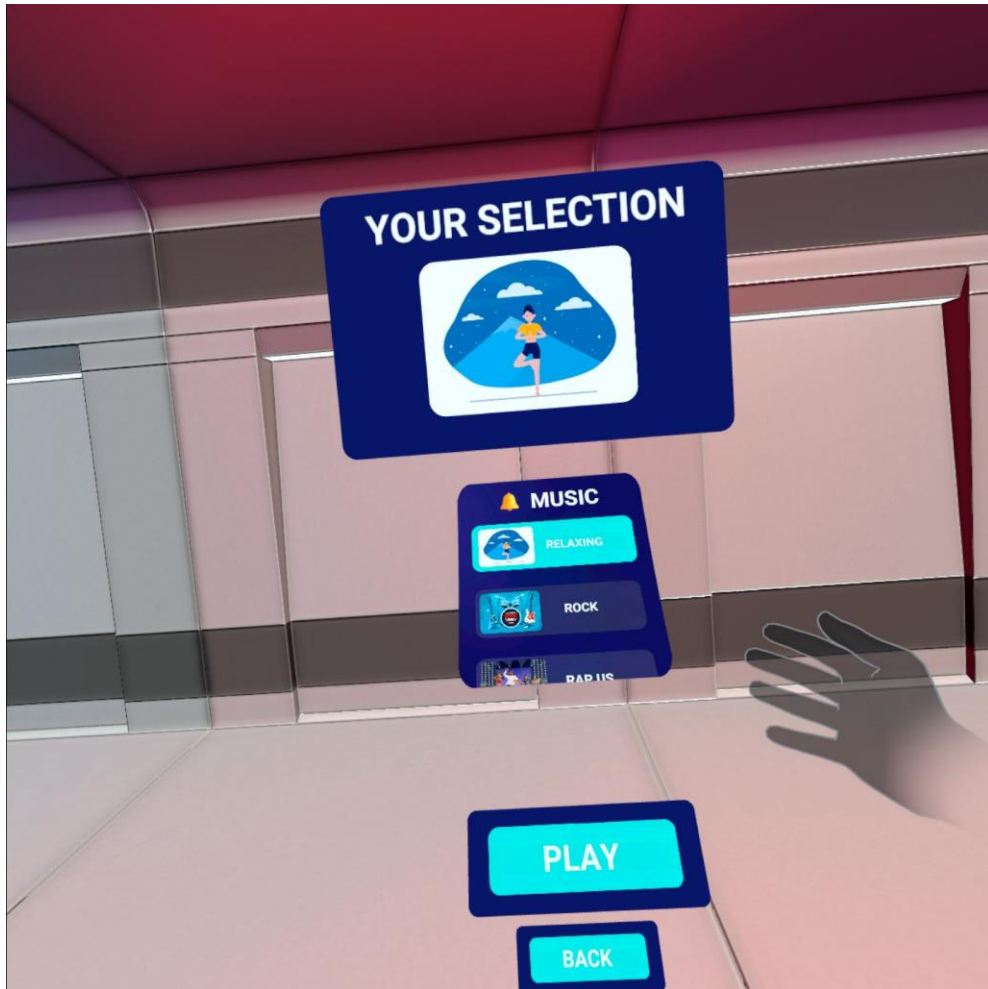


Figure 2: Capture d'écran de l'interface de sélection de la musique

Figure 3: Capture d'écran du menu de sélection des options

Interface

Boutons du menu

Les boutons présents dans les différents menus sont des boutons constitués d'un Canvas Mesh, qui sert uniquement de collider, pour détecter quand l'utilisateur interagit avec cette partie du menu. Au même niveau se trouve un Canvas dans lequel on affiche les éléments. Ainsi, on trouve dans le Canvas un élément Scroll View, qui comme son nom l'indique permet de scroll et est utilisé pour l'interface du tutoriel, ou le menu de sélection de la musique, par exemple. A l'intérieur de celui-ci se trouve un Viewport, qui lui-même contient les éléments

à afficher (les différentes musiques, par exemple) ainsi qu'un élément, Content, qui sert à stocker la valeur qui sera sélectionnée par l'utilisateur. Ainsi, si l'utilisateur choisit la musique calme, cliquer sur le bouton de la musique calme déclenchera une fonction qui se charge de changer la valeur de Content avec l'index du bouton (la musique relaxante étant la première, Content sera changé à 0). Concernant le bouton Play, il se charge d'afficher un écran de chargement jusqu'à ce que le niveau finisse entièrement de charger, et d'initialiser le score à 0.

Il est important de noter que nos boutons fonctionnent directement en appuyant dessus avec la main, comme pour un bouton physique. Cela est dû à l'utilisation d'un script Poke Interactable fourni par le plugin d'Oculus. Ce dernier nécessite un proximity field (qui va déterminer la distance à laquelle la main a besoin de se trouver pour activer le bouton) ainsi qu'une surface sur laquelle appliquer ce proximity field. Ensemble, cela permet de trigger un bouton dès que la main se rapproche assez de l'élément en question. En réglant un proximity field très petit, on force l'utilisateur à appuyer physiquement sur le bouton, comme dans la vraie vie. C'est cette mécanique qui a été appliquée sur tous les boutons de l'environnement virtuel afin d'avoir une expérience consistante et un peu plus immersive pour les participants.

Boutons du Batak

En ce qui concerne les boutons du Batak, ils sont constitués de différents asset qui ensemble constituent le bouton comme montré sur la figure 3.

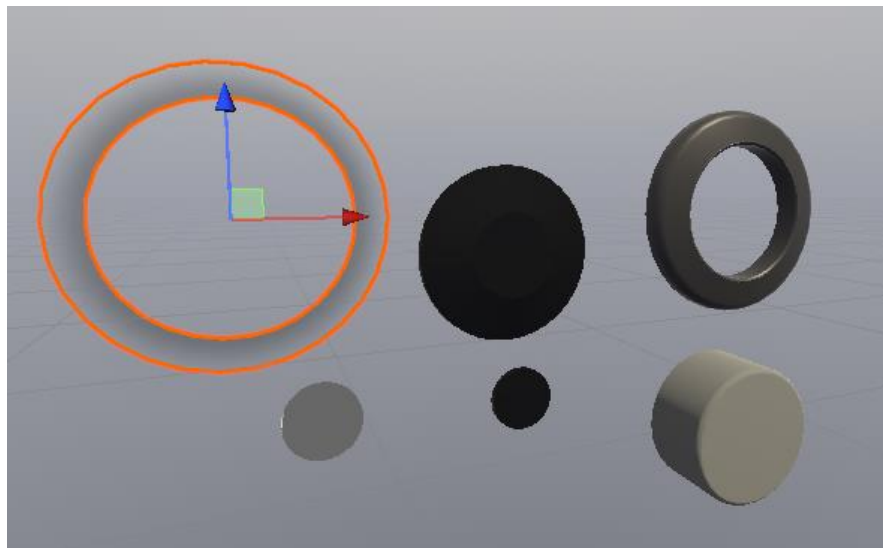


Figure 4: Vue éclatée du bouton du Batak

L'élément interactif du bouton est celui présenté tout en bas à droite. Lorsque cet élément est touché, une animation est déclenchée (le bouton se rétracte, puis revient à sa position initiale). Si le bouton était allumé, un appel à la fonction ActionAccordingStatus. Sur chaque bouton se trouve un tag nommé soit « ButtonActivated », soit « ButtonUnactivated ». Si le bouton était activé, alors on calcule le temps passé depuis le dernier bouton pressé que l'on enregistre dans un fichier texte (et l'on sauvegarde le temps actuel pour pouvoir le recalculer au prochain bouton appuyé), on rajoute 1 sur le ScoreValue, puis on désactive le bouton en

changeant son tag. Enfin, on termine en changeant la couleur du bouton qu'il redevienne gris, et non rouge. La désactivation du bouton déclenche un autre script, qui si aucun bouton n'est actif, en tire un au hasard (avec la possibilité de réactiver le dernier) et l'active en changeant son tag (pour que cela soit pris en compte au niveau du code) et sa couleur (pour que le joueur puisse le voir).

Scripts

Durant la préparation du projet nous avons conçus divers scripts pour le bon fonctionnement de l'environnement virtuel. Certains ont déjà été cités, notamment celui des boutons. Néanmoins, nous avons également d'autres scripts qui seront expliqués plus en détails ici.

GameManage sert à gérer tous les aspects de la simulation de Batak. Il importe les choix faits par l'utilisateur sur le menu précédent et les applique. Ainsi, au démarrage, il applique les choix de l'utilisateur, lance le décompte et l'affiche sur l'UI pour que l'utilisateur puisse le voir pendant la partie. Ce décompte est ensuite actualisé à chaque frame, arrondi à la seconde. Lorsque que la partie est terminée, la fonction `GameSessionFinished` est appelée et va se charger d'afficher son score au participant, et mettre à jour le leaderboard. GameManager contient aussi les fonctions qui servent à mettre en pause une simulation ainsi qu'à changer le statut d'un bouton (comme expliqué plus haut).

SceneManage contient les fonctions qui permettent de changer de Scene (comme son nom l'indique) et ses fonctions sont donc généralement reliées au bouton qui servent à lancer une partie, changer de menu ou quitter l'application. On y retrouve entre autres `ScenePlay` qui initie le score à 0 et qui affiche un écran d'attente avant d'afficher le Batak, ou `SceneSelection` `SceneMenu` ou encore `RestartPlay` qui sont qui autologiques.

```
public void RestartPlay(){
    GameManager.timeRemaining = 0;
    GameManager.timerIsRunning = false;
    Time.timeScale = 1;
    GameManager.buttonOn = false;
    Score.ScoreValue = 0;
    SceneManager.LoadScene(SceneManager.GetActiveScene().name);
}
```

Figure 5: Fonction `RestartPlay` dans `SceneManage`

LoadingManage sert pour l'écran de chargement. Il est utilisé pour lancer de manière asynchrone le chargement de la scene du Batak avec les paramètres demandés par l'utilisateur tout en affichant une barre de chargement qui est mise à jour à chaque update de frame.

SelectionGame est utilisé dans le menu de sélection des paramètres pour colorer les boutons et fournir un feedback visuel quand une option est sélectionnée à travers la fonction `ChangeselectionImageSelection`.

SaveReactiontime est un script important de notre projet car comme son nom l'indique, il est chargé, à chaque partie, de créer un fichier texte nommé d'une manière spécifique <NumeroUser>Reaction_Time_Of_Game<NumeroPartie>. Cela permet par la suite de traiter les fichiers automatiquement (moyennant un prétraitement à la main pour éliminer les parties nulles qui ont rencontré un problème lors de leur déroulement). Deux fichiers sont créés, un pour stocker le délai entre chaque touche, et un autre pour y stocker le score. Cela n'est pas optimal certes, mais notre approche du traitement via un script en Python nous élimine le problème des démarches supplémentaires éventuelles dues à la répartition des données de l'expérience dans plusieurs fichiers.

```
public static void WriteString(string ReactionTime)
{
    endGame = false;
    // "/Data_Of_Game_" + numGame.ToString() +
    nameFile = "/" + WriteSubjectNumber.staticSubjectNumber + "_Reaction_Times_Of_Game_" + numGame.ToString() + ".txt";
    path = Application.persistentDataPath + nameFile;
    //Write some text to the test.txt file
    StreamWriter writer = new StreamWriter(path, true);
    writer.WriteLine(ReactionTime);
    writer.Close();
    StreamReader reader = new StreamReader(path);
    //Print the text from the file
    Debug.Log(reader.ReadToEnd());
    reader.Close();
}

public static void ReadString()
{
    string path = "/" + WriteSubjectNumber.staticSubjectNumber + "_Reaction_Times_Of_Game_" + numGame.ToString() + ".txt";
    //Read the text from directly from the test.txt file
    StreamReader reader = new StreamReader(path);
    Debug.Log(reader.ReadToEnd());
    reader.Close();
}
```

Figure 6: Capture d'écran des fonctions d'écriture utilisées pour exporter les données du casque

Enfin, WriteSubjectNumber est le premier script à être appelé puisqu'il se charge d'afficher l'interface qui va servir au participant pour rentrer son numéro de participant, qui sera plus tard utilisé pour la création des fichiers exportés pour traitement ainsi que pour le formulaire.

Traitement des données

Les données issues du casque sont d'abord prétraitées à la main pour créer un fichier texte avec au début le score de la partie, suivi par tous les timers séparés par des retours à la ligne. Ces fichiers textes sont ensuite renommés <numeroParticipant>-<NumeroPartie>-Musique, par exemple 30-2-rap.

Le script python est divisé en 3 parties, une pour extraire les données des fichiers textes et les importer dans des dataframes (« data extraction », plus d'information dessus juste après), une deuxième partie pour le traitement des données (« data processing »), et enfin la visualisation des données (« data visualization »). Comme il n'est pas remis avec les fichiers Unity (le code ne fait pas partie de l'apk), **il est disponible en annexe du présent rapport**. Le code est fortement commenté, il est donc tout à fait possible de suivre la partie qui suit avec le code sous les yeux.

Le script fait lourdement usage des librairies pandas (pour le traitement des données) matplotlib pour l'affichage et la création des graphiques. Il utilise également numpy et glob pour l'import des fichiers textes.

Une fois les fichiers textes localisés en fonction de leurs noms, on les ouvre pour récupérer la première ligne, qui est notre score. Toutes les autres lignes sont stockées dans une liste. Une fois que l'on a récupéré toutes les informations du fichier texte, on ajoute toutes les données à la dataframe (l'ID du participant, ainsi que le genre de musique est récupéré depuis le titre du fichier du texte), puis l'on passe au fichier texte suivant.

Afin de pouvoir présenter les graphiques représentés dans les figures 10 à 12, il nous faut calculer le temps cumulé pour chacun des temps enregistrés dans chaque partie. On entre alors dans la partie « traitement des données ». On ajoute une nouvelle colonne au dataframe nommé « cumulative_time », peuplé pour l'instant uniquement de listes vides à l'aide de numpy. Pour les peupler, il suffit d'itérer sur toutes les lignes de la dataframe pour accéder à la liste des temps, puis pour chaque temps présent dans la liste, ajouter dans la colonne des temps cumulés le nouveau temps cumulé calculé. Une fois que cela est fait, on peut diviser notre dataframe en 3 parties différentes en fonction de la musique et l'on passe à la partie « Visualisation des données ».

Ici, il ne nous reste qu'à afficher les données. Pour récupérer les données à afficher dans le graphique, il suffit d'itérer dans les 3 dataframes en ajoutant les lignes au fur et à mesure et de les afficher à la fin. On peut également profiter de l'itération pour grappiller quelques données et calculer l'écart type et le score moyen de chacun des dataframe. Finalement, on affiche tous les graphiques que l'on a construit lors de l'exécution du code avec plot.show(), en prenant bien soin auparavant d'ajouter les labels et un titre ainsi que définir les fenêtres d'affichage.

5. Protocole expérimental

Protocole

Concernant le protocole, nous avons choisi de faire réaliser aux participants 3 sessions de 1 minute de Batak avec soit une musique calme, soit sans son, soit une musique rythmée de Rap US avec un tempo plus rapide. La session commence toujours par 3 parties de 60 secondes où le participant peut choisir la musique de son choix pour se familiariser avec l'environnement, le fonctionnement du Batak, etc...

Au bout des 3 sessions, la musique est imposée. Le choix de la première musique jouée est décidé aléatoirement entre calme, sans son, ou rap US, et restera la même pour les 3 prochaines parties, après quoi les autres options sont jouées pour les 3 parties suivantes. Ainsi, l'utilisateur participe à 12 parties maximum, soit 3 d'entraînement (ou moins si le participant a déjà de l'expérience avec la VR et souhaite passer à la suite), puis 3 avec chacune des 3 options imposées. En tout, le temps passé dans la VR doit être d'environ 15 minutes. Les participants peuvent à tout moment demander une pause (que certains ont demandé car l'exercice utilise beaucoup les bras).

Participants

Les participants de l'expérience sont des jeunes de 22 à 32 ans. Il est constitué de 5 femmes et 3 hommes pour un total de 8. Quatre d'entre eux portaient un système de correction de la vue au moment de l'expérience (1 participant avait des lunettes, 3 des lentilles).

En ce qui concerne leur expérience à la VR, le formulaire nous a permis de créer le diagramme suivant :

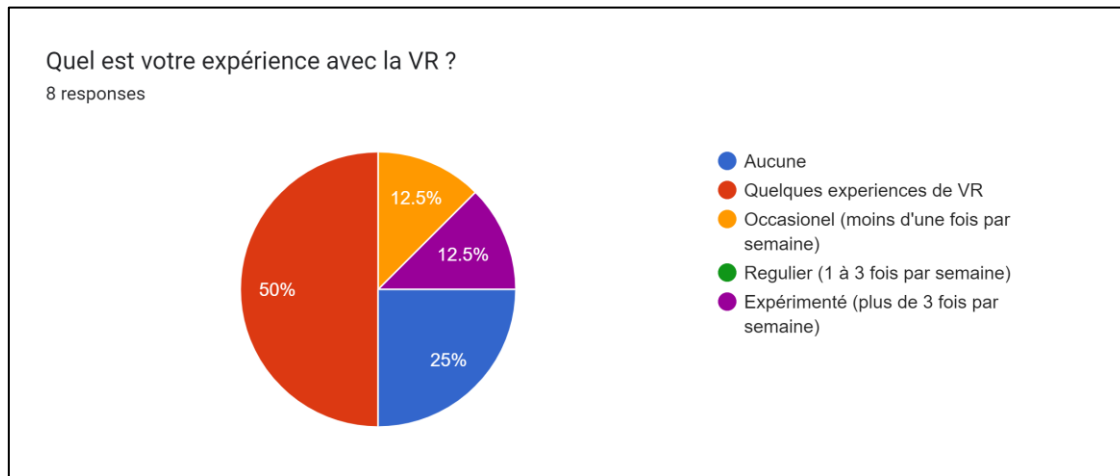


Figure 7: Diagramme résumant l'expérience des participants sur la VR

On a donc majoritairement des gens qui ont déjà pu expérimenter de la VR sans pour autant réellement avoir de l'expérience, et un quart des participants qui n'avait jamais mis un casque de VR avant cette étude.

Matériel

Le matériel utilisé est un casque de VR Meta Quest 2 pour l'environnement virtuel, ainsi qu'un laptop pour concevoir l'EV dans Unity et faire tourner un script python pour l'analyse des données.

L'APK est codée de sorte que le casque récupère automatiquement le score ainsi que le délai entre chaque bouton appuyé.

6. Résultats

En ce qui concerne les résultats, nous commencerons par les résultats de l'expérience de VR issus du casque, puis nous mettrons cela en relation avec les réponses que les candidats ont donné au formulaire qui était à remplir à la fin de l'expérience.

Tout d'abord, l'analyse des données qui ressort de l'étude : lorsque l'on regroupe les données de chaque partie en fonction de la musique écoutée, on observe les éléments affichés dans la figure 7.

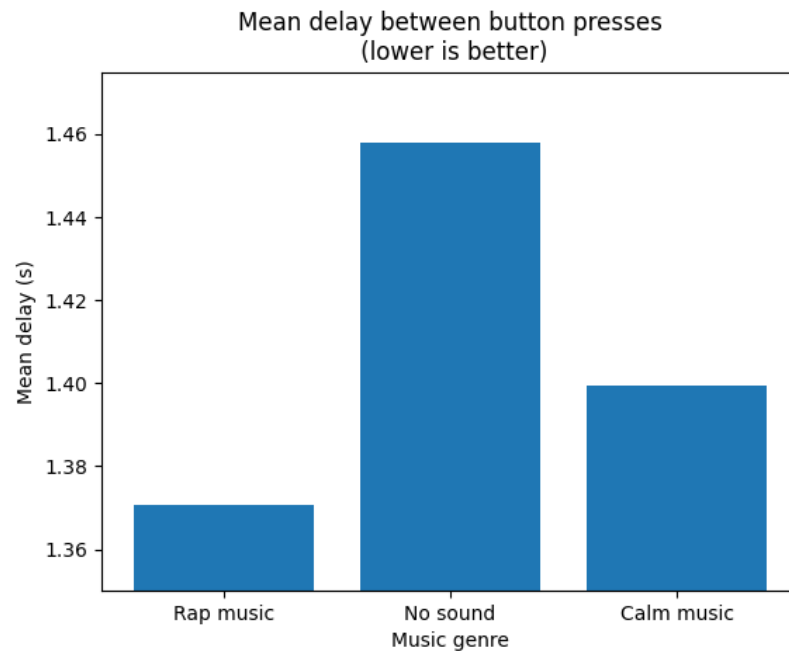


Figure 8: Temps moyen entre chaque activation de boutons selon le genre de musique

Comme nous pouvons le constater, les participants appuient en moyenne plus rapidement sur les boutons lorsqu'une musique avec un tempo élevé est diffusée, avec un délai moyen de 1.370764 seconde. Ensuite vient la musique relaxante, avec un délai moyen de 1.399324 seconde, et enfin l'absence complète de musique avec 1.458037 seconde en moyenne.

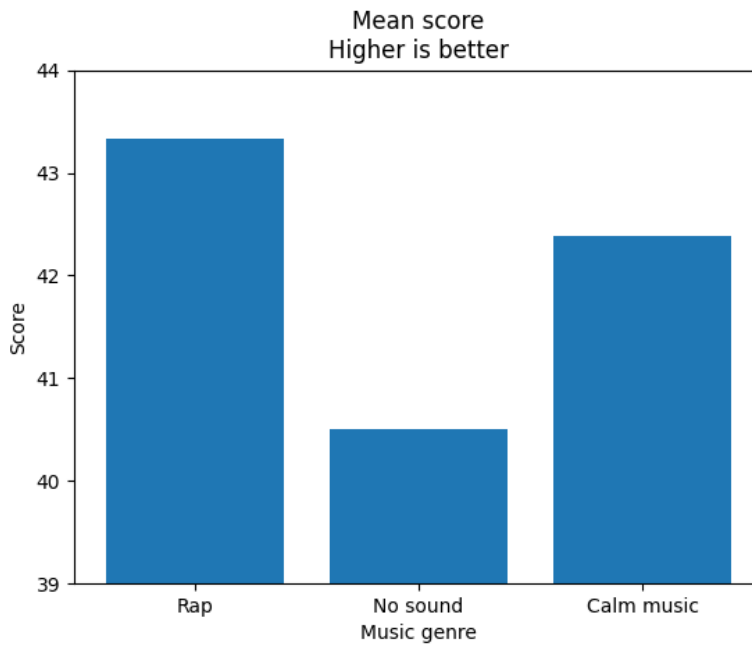


Figure 9: Score moyen en fonction de la musique

Ces résultats se traduisent également sur le score (plus le délai entre chaque appui est court, plus le participant peut réaliser d'appuis dans un laps de temps donné) comme le montre la figure 8.

Ainsi, le score moyen avec le rap est de 43.333333, avec la musique calme 42.384615, et sans son 40.5.

Nous avons aussi calculé l'écart type des scores en fonction des musiques pour voir si les données étaient très dispersées ou non. Ces données sont montrées dans la figure 9. Nous pouvons ainsi constater que l'écart type pour le rap est de 5.472877, la musique relaxante de 3.428762 et le silence de 5.244044. Ces écarts types relativement élevés (plus de 10% du score moyen, sauf pour la musique relaxante) indiquent une grande variabilité dans les données recueillies. Un plus grand nombre de participants pourrait servir à réduire les écarts types et donner des résultats solides.

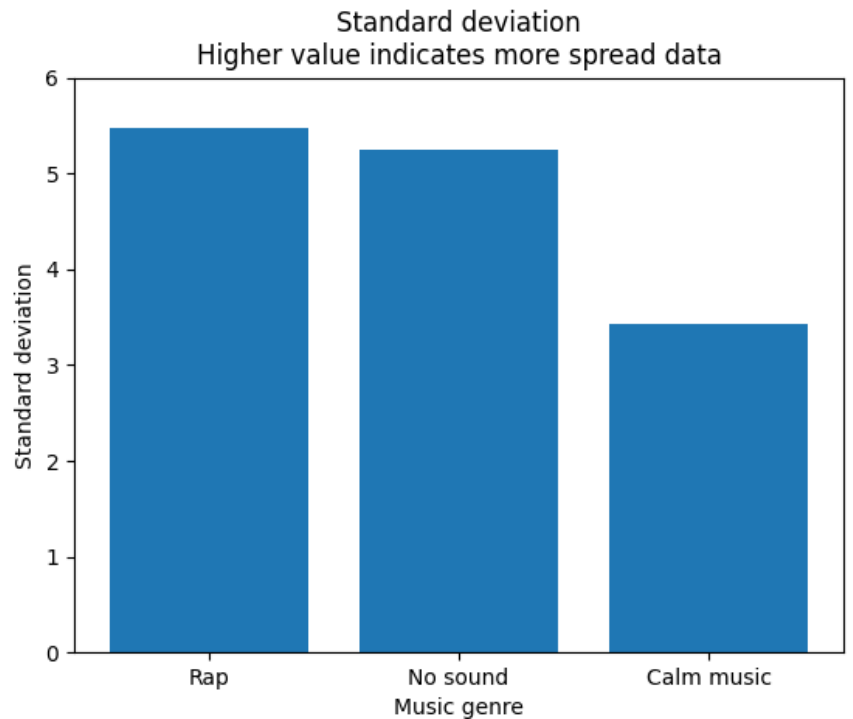


Figure 10: Ecart type des scores en fonction du genre de musique

Enfin, nous avons également construit des graphes qui montrent les délais moyens entre les appuis sur les boutons au cours de la musique sur une session de 60 secondes. Les 3 prochaines figures en sont leur illustration.

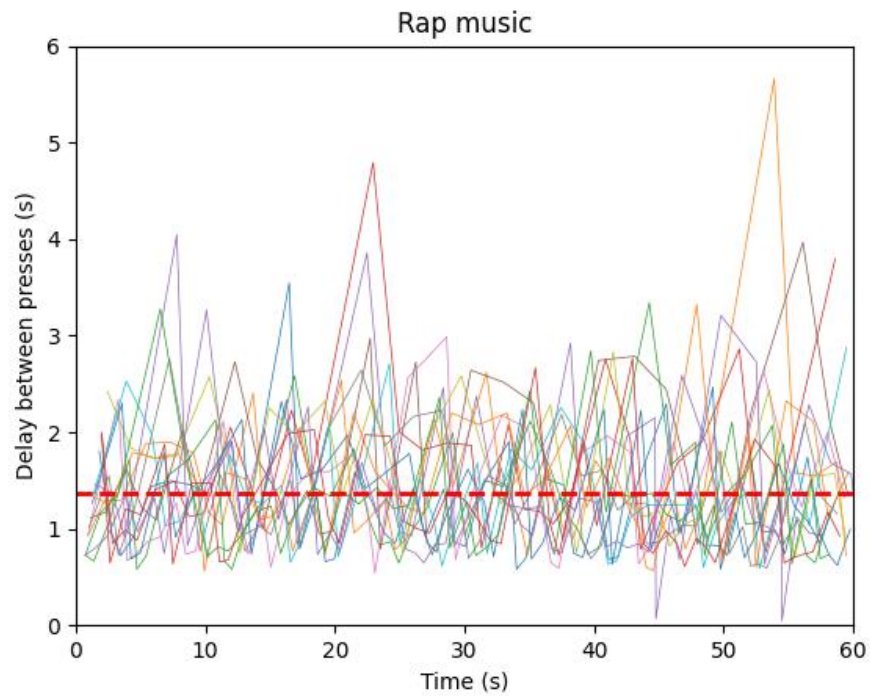


Figure 12: Délai moyen entre chaque activation de bouton sur la musique rap

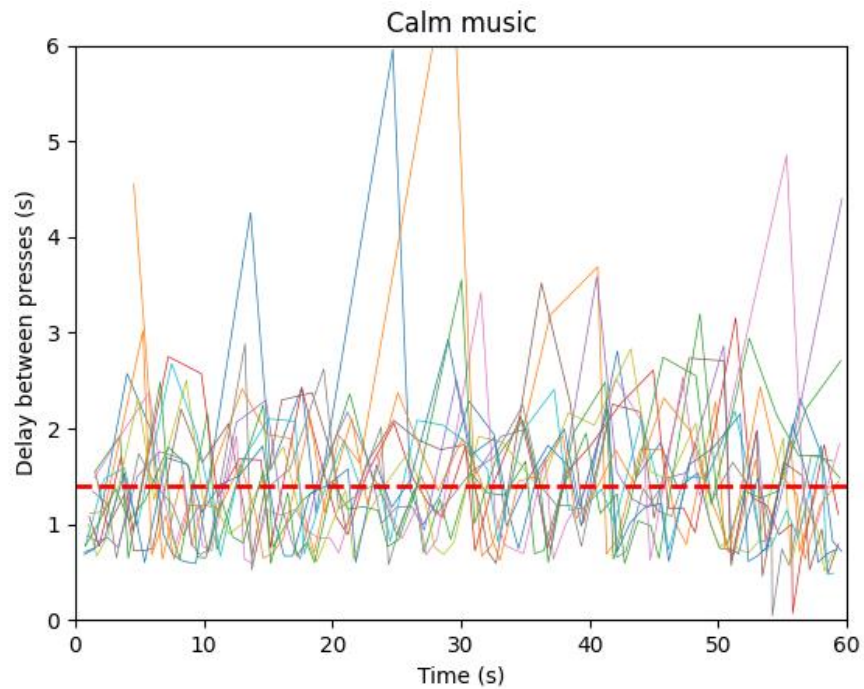


Figure 11: Délai moyen entre chaque activation de bouton sur la musique calme

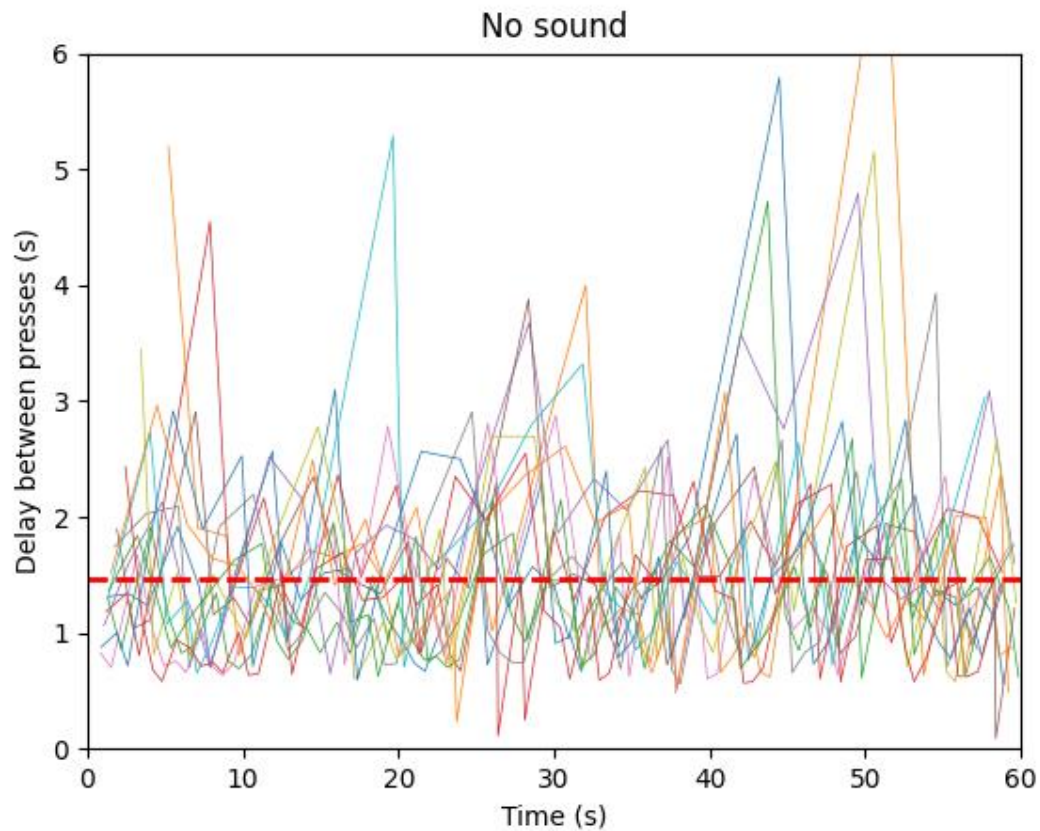


Figure 13: Délai moyen entre chaque activation de bouton quand il n'y a pas de musique

On peut grâce à ces graphes, visualiser l'écart type : on peut en effet se rendre compte que dans le cas de la musique calme, même si des pics sont présents, ils sont beaucoup moins nombreux, ce qui explique l'écart type plus bas. Nous voulions également tracer une courbe moyenne pour chacun des genres de musique mais cela posait plusieurs problèmes, qui seront abordés plus en détails dans la partie Discussion.

Questionnaire

Sentiment d'immersion

Quelques données concernant le formulaire, tout d'abord sur l'immersion des participants dans l'environnement virtuel.

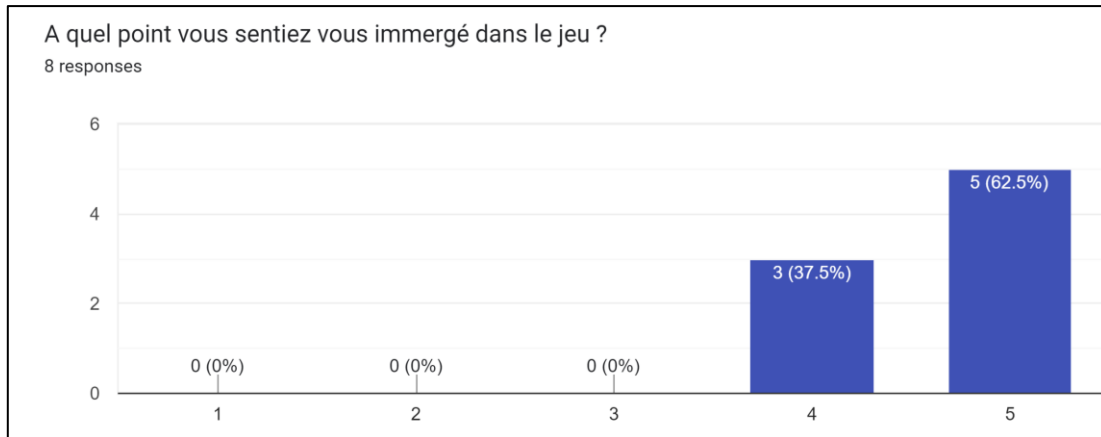


Figure 14: Sentiment d'immersion dans le jeu

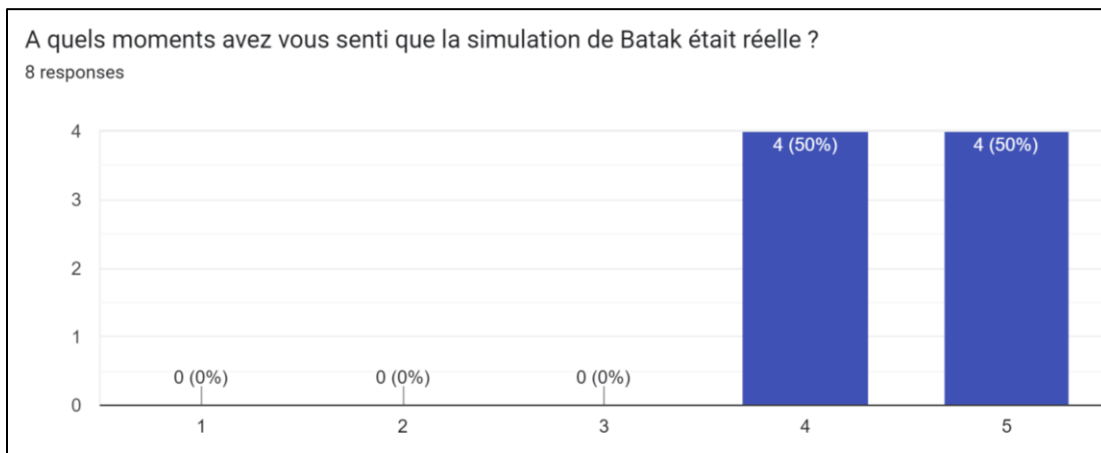


Figure 15: Les participants avaient unanimement fortement l'impression d'être dans un endroit réel et non virtuel

Avec les deux figures 13 et 14, on peut attester que le sentiment d'immersion était bien là pour les participants. Le sentiment de cyber malaise était également très peu présent, comme le relève la figure 15. Aucun participant n'a eu de nausées, ou de maux de têtes à la suite de l'expérience de VR.

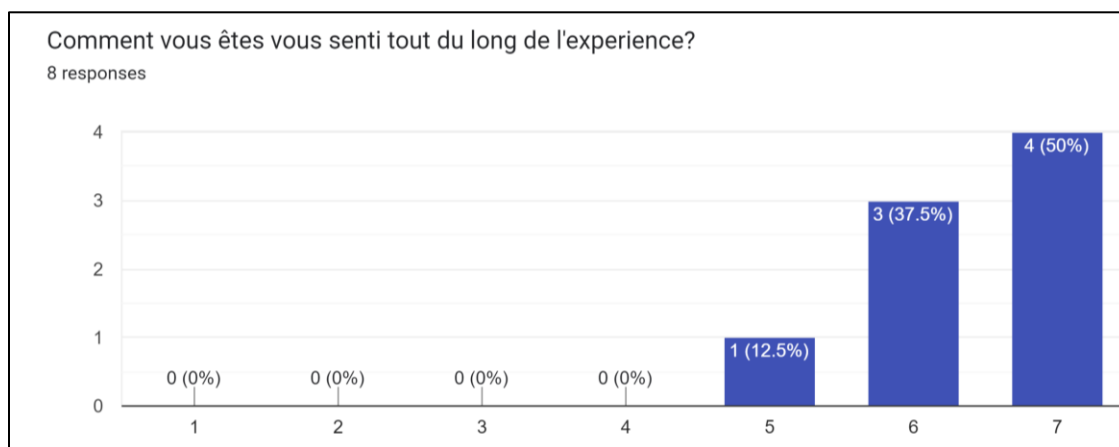


Figure 16: Sentiment de bien-être des participants lors de l'expérience

Perception de leur performance

Pour évaluer la perception qu'ont les participants sur leurs propres performances, les deux questions présentées dans la figure 16 leur sont posées lors du questionnaire.

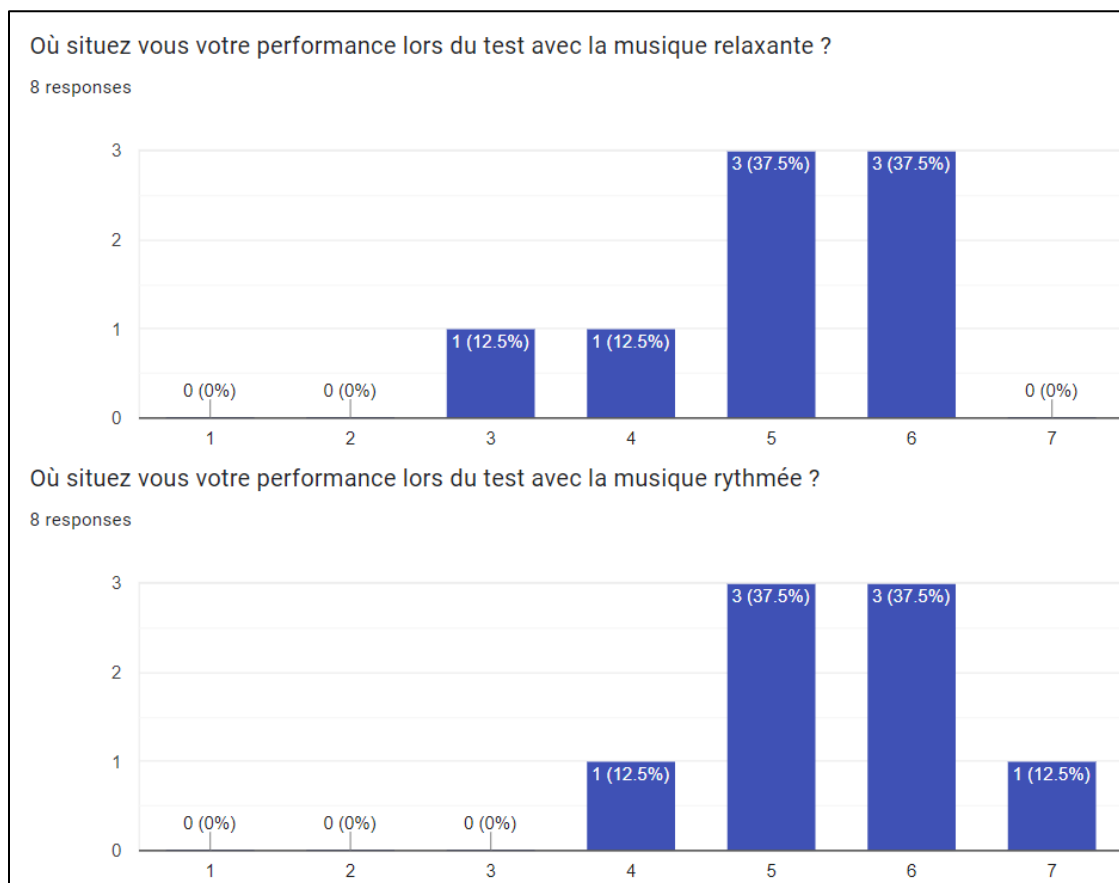


Figure 17: Perception de leur propre performance en fonction de la musique

Les candidats avaient tout de même l'impression de mieux performer lorsque de la musique rythmée était diffusée par le casque de VR.

Pensez vous que la musique à eu un impact sur votre performance ?

8 responses

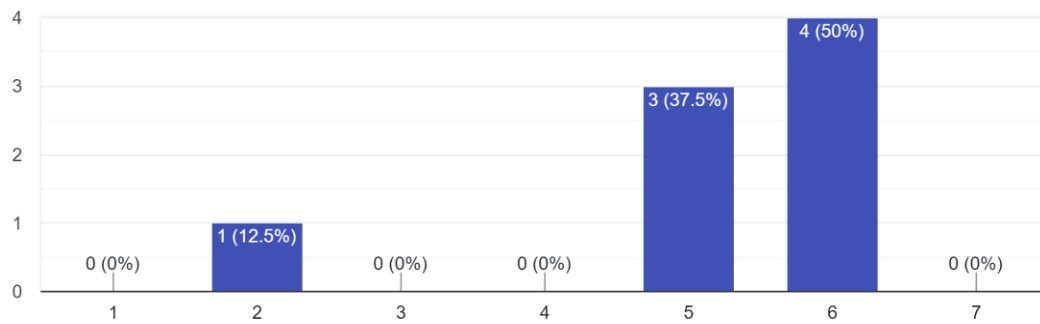


Figure 18: Impact de la musique sur leur performance telle que perçue par les participants

Enfin, la figure 17 illustre l'impact que la musique a eu sur les performances des participants, d'après les participants eux-mêmes.

7. Discussion

Analyse des résultats

Expérience de VR

Les résultats obtenus indiquent tout de même un avantage à écouter de la musique lors que l'on s'entraîne au Batak. En effet, nous avons vu dans les résultats que les gens écoutant de la musique avaient tendances à mieux performer que lorsqu'ils écoutaient une musique relaxante avec un tempo très bas ou pas de musique du tout. Cela peut s'expliquer par le fait que les participants ont en fait tendance à se caler sur le tempo de la musique pour appuyer sur les boutons en rythme avec la musique. Dans le cas de la musique relaxante, quand le tempo est bas, on peut penser que la présence de son ambient aiderait les participants à se concentrer sur la tâche, tandis que sans son, rien ne motive les participants à particulièrement appuyer plus vite sur les boutons.

Le score moyen avec Rap est tout de même 6.996% plus élevé que le score moyen sans son, contre 4.653% pour la musique calme par rapport au manque de musique également.

Néanmoins, la force de ces résultats sont faibles. En effet, l'étude n'a porté que sur 8 personnes, ce qui reste très peu par rapport à la variabilité montrée par l'écart types de nos résultats. Il n'en reste pas moins que ces résultats vont dans le sens de nos hypothèses, qui suggéraient que les participants seraient plus performants lorsqu'exposé à une musique possédant un tempo élevé.

Formulaire

Comme le montre les deux graphes sur la figure 16, aucun participant n'a noté sa performance comme très bonne lors de la musique relaxante. De plus, une personne a noté sa performance à 3 avec la musique relaxante contre personne pour la musique de rap. On peut donc en conclure que les participants trouvent qu'ils performant tout de même légèrement mieux lorsqu'ils entendent de la musique rythmée. Là où cette tendance est faible, il n'en est pas moins de ce que les participants pensent de l'impact qu'a eu la musique sur leur performance, où l'écrasante majorité a pensé que la musique avait un impact fort sur leur performance physique au Batak. Ce sentiment est quelque peu appuyé par les données recueillies par le casque avec une amélioration de 2.23% du score moyen des participants avec la musique rythmée par rapport à la musique calme. Comme précisé pour les données de l'expérience au-dessus, la manque de participant n'accorde un poids que très léger aux conclusions menées ici, sur des chiffres d'amélioration qui sont ici de l'ordre de l'écart type mesuré sur les données, donc non significatifs.

Limites

Comme mentionné plus haut, notre étude est limitée de plusieurs manières. Dans le choix des participants premièrement, nous pensons qu'un plus grand nombre de participants serait nécessaire pour avoir des résultats véritablement significatifs. De plus, la manière donc le casque suit les mains des participants limite également l'acquisition des résultats, car il faut

que les mains soient en permanence dans le champ de vision du casque pour pouvoir appuyer sur un bouton, ce qui empêche de commencer à regarder les autres boutons pendant que notre main va automatiquement appuyer sur le bouton allumé. Cela s'est particulièrement observé pour les boutons dans les coins du Batak, tout en haut ou tout en bas de la structure. Il faut également éviter les mouvements trop brusques qui peuvent causer au casque de perdre la main quelques fractions de secondes, qui cumulées font la différence au niveau du score. Comme toutes les parties se sont déroulées dans les mêmes conditions, cela n'est pas réellement un problème ici, mais dans le cadre d'une étude plus robuste, on essaiera de trouver une manière plus fiable de traquer les mains des participants.

Une autre limite dont nous nous sommes rendu compte durant le déroulement des expériences est l'affichage du score à la fin de chaque partie, qui peut potentiellement biaiser le candidat lorsqu'il remplira ensuite le questionnaire, particulièrement la partie sur ses performances. Il peut être ainsi tenté de remplir le formulaire en fonction des scores dont il se souvient, plutôt qu'en fonction de son ressenti, ce qui était la manière de répondre attendue ici.

De plus, nous est venu à l'esprit l'idée de tracer quelques courbes de manière à comparer la première, la deuxième ou la troisième partie de tous les joueurs sur une musique donnée, afin de voir s'il y avait des similitudes dans les mesures. Il ne nous restait malheureusement pas assez de temps pour l'implémenter en python, mais c'est une donnée qu'il pourrait être intéressant d'extraire des données récupérées par le casque.

Enfin, nous voulions implémenter le tracé d'une courbe moyenne du délai de l'activation des boutons par les participants en fonction de l'avancement de la musique, et ce pour chaque genre. Cela était un obstacle majeur car les moments auxquels les joueurs pressent les boutons ne sont évidemment pas synchronisés, et de nombres différents (étant donné que les scores varient souvent). Or, si on n'a pas le même nombre de mesures et qu'elles ne sont pas synchronisées, alors il faudra faire de l'interpolation pour avoir le même nombre de valeurs prises au même moment pour toutes les parties. Néanmoins, cette approche apporte plusieurs questions : comment interpoler correctement ? Où ajouter les valeurs dans les parties qui ont un plus petit score que les autres, ou où retirer les valeurs si on nivelle par le bas ? Au-delà de ces questions, se trouve aussi le problème de la manière dont on interpole ? Interpolation linéaire ? Polynomiale ? Peut-être en calculant une moyenne pour des fenêtres d'une seconde, au risque de créer des discontinuités ? Pour ces raisons, nous avons décidé de ne pas perdre de temps sur cet élément, qui ne nous aurait pas beaucoup apporté en dehors de mettre en évidence un moment fort de la musique où les participants auraient brusquement performé au-dessus ou en deçà de la moyenne de la partie.

8. Conclusion

Notre environnement possède différents points forts, l'un d'entre eux étant la qualité de l'environnement. Comme le montre certaines des figures, l'environnement est assez bien réalisé de telle manière que les participants se sentent réellement immergés dans la simulation. Les boutons de l'interface qui sont tactiles sont également un élément fort qui participe à l'immersion du participant, et le fait de ne pas utiliser les manettes permet également de pouvoir interagir avec chacun de ses doigts et de pouvoir les visualiser directement dans le casque.

La prochaine étape pour cette étude serait de trouver plus de participants, afin de pouvoir avoir des résultats réellement plus significatifs. Avec plus de temps, nous pouvons plus développer l'application de manière qu'il ressemble plus à un jeu. En effet, nous avons commencé à implanter des changements de décors ainsi qu'une manière de changer l'apparence des mains (ces fonctions sont disponibles dans la version *heavy* du jeu lorsque l'on lance une partie. Le changement de décor fonctionne, celui des mains était en cours de développement). Nous aurions aussi pu varier le placement des boutons pour se différencier du Batak et créer notre propre jeu. On aurait pu aussi proposer des niveaux avec des difficultés à l'aide de séquences d'allumage prédéfinis, ou en proposant des variations de gameplay, par exemple en permettant à plusieurs lumières de s'allumer en même temps... les possibilités sont infinies.

9. Références

Hypothèses

Welz W, Voelter-Mahlknecht S, Große-Siestrup C, Preuß G. The Influence of Different Auditory Stimuli on Attentiveness and Responsiveness in Road Traffic in Simulated Traffic Situations. Int J Environ Res Public Health. 2020 Dec 10;17(24):9226. doi: 10.3390/ijerph17249226. PMID: 33321821; PMCID: PMC7764073.

Paridon H., Springer J. Effekte von Musik per Kopfhörer auf das Reaktionsverhalten bei unterschiedlichen Verkehrsgeräuschen. Z. Verk. 2012;4:192–195.

Edworthy, J., & Waring, H. (2006). The effects of music tempo and loudness level on treadmill exercise. Ergonomics, 49(15), 1597–1610. doi:10.1080/00140130600899104

Scroll Rect | Documentation Unity UI | 1.0.0. (Visité le 7/12/2022). Repéré à <https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/script-ScrollRect.html>

Assets Unity Version Heavy

Essential VR Hands Collection | 3D Characters | Unity Asset Store (visité le 10/12/2022). Repéré à <https://assetstore.unity.com/packages/3d/characters/essential-vr-hands-collection-227788>

Low Poly Ultimate Home Pack | 3D Environments | Unity Asset Store. (indisponible au 10/12/2022). Repéré à <https://assetstore.unity.com/packages/3d/environments/low-poly-ultimate-home-pack-164671>

Medieval Tombs - Egypt | 3D Dungeons | Unity Asset Store. (visité le 10/12/2022). Repéré à <https://assetstore.unity.com/packages/3d/environments/dungeons/medieval-tombs-egypt-96865>

Assets Unity Version Light & Heavy

3D Free Modular Kit | 3D Environments | Unity Asset Store. (visité le 10/12/2022). Repéré à <https://assetstore.unity.com/packages/3d/environments/3d-free-modular-kit-85732>

Images

Free Vectors, Stock Photos & PSD Downloads. (visité le 10/12/2022). Freepik. Repéré à <https://www.freepik.com>

Quantum UI | GUI Tools | Unity Asset Store. (visité le 10/12/2022). Repéré à <https://assetstore.unity.com/packages/tools/gui/quantum-ui-162077>

Documentation Oculus

Poke | Oculus Developers. (visité le 10/12/2022). Repéré à <https://developer.oculus.com/documentation/unity/unity-isdk-poke-interaction/>

Ecrire / Lire un fichier sur l'Oculus

How do I read and write data from a text file? – Unity. (visité le 10/12/2022). Repéré à <https://support.unity.com/hc/en-us/articles/115000341143-How-do-I-read-and-write-data-from-a-text-file->

Nunes, M. (publié le 2019a, 5 décembre). Answer to « How to write file to Oculus Quest internal storage ». Stack Overflow. Repéré à <https://stackoverflow.com/a/59203146>

Nunes, M. (publié le 2019b, 6 décembre). How to write file to Oculus Quest internal storage [Forum post]. Stack Overflow. Repéré à <https://stackoverflow.com/q/59203145>

Question - Writing files to Oculus Quest. (visité le 10/12/2022). Unity Forum. Repéré à <https://forum.unity.com/threads/writing-files-to-oculus-quest.1076615/>

Technologies, U. (visité le 10/12/2022). Unity - Scripting API: Application.persistentDataPath. Repéré à <https://docs.unity3d.com/ScriptReference/Application-persistentDataPath.html>

Annexes

Annexe A : Script Python

```
import pandas as pd
import glob
import matplotlib.pyplot as plt
import numpy as np

# DATA EXTRACTION #
=====

# create an empty DataFrame
df = pd.DataFrame()
filenames = glob.glob("*.txt")

# iterate over the files
for filename in filenames:
    # extract the id, run number, and genre from the filename
    id, run_number, genre = filename.split("-")

    # read the score from the file
    with open(filename) as f:
        score_str = f.readline().strip()
        score = int(score_str)

    # read the times from the file
    times = []
    for i in range(score):
        to_append = float(f.readline())
        times.append(to_append)

    # add the data to the DataFrame
    df = df.append({
        "id": id,
        "run_number": run_number,
        "genre": genre,
        "score": score,
        "times": times
    }, ignore_index=True)

# DATA PROCESSING #
=====

# We want a column with all the cumulative times that will be our x-
axis for the plots
# Creates a column filled with empty lists to populate with cumulative
time
df['cumulative_times'] = np.empty((len(df), 0)).tolist()

# Now, we want to calculate the cumulative times for all the runs:
# Iterate over the rows of the dataframe
for i, row in df.iterrows():
    # Append the times in the current row to the list of cumulative
times
    cumulative_times = []
```

```

        for time in row["times"]:
            # If the list is empty, we simply add the first value
            if len(df.at[i, "cumulative_times"]) == 0:
                df.at[i, "cumulative_times"].append(time)
            # Else, we add the last value to our latest time and add it to
the list
            else:
                df.at[i, "cumulative_times"].append(time + df.at[i,
"cumulative_times"][-1])

# Now we regroup the data of each genre to plot them. We also set the
name of the df we can use it to title the graph
df_rap = df.loc[df['genre'] == 'rap.txt']
df_rap.name = "Rap music"

df_nosound = df.loc[df['genre'] == 'nosound.txt']
df_nosound.name = "No sound"

df_calm = df.loc[df['genre'] == 'calm.txt']
df_calm.name = "Calm music"

# DATA VISUALIZATION #
=====

# This list will be used to iterate on all the new df
list_genre = [df_rap, df_nosound, df_calm]

# Those values will be used for a bar plot
x2 = []
y2 = []
fig2, ax2 = plt.subplots()

# Let's plot each genre now!
print("Mean delay between presses when using :") # the other part of
the sentence is inside the loop
# for each genre
for dataframe in list_genre:
    # we prepare the plot, and initialize a variable to store the times
and calculate the mean
    fig, ax = plt.subplots()
    all_times = []
    ax.set_title(dataframe.name)
    # We query all the times in each row and plot them on the graph
    for i, row in dataframe.iterrows():
        y = dataframe.at[i, "times"]
        x = dataframe.at[i, "cumulative_times"]
        ax.plot(x, y, lw=0.5)
        # we also add all times to our list. We loop in the row because
directly adding dataframe.at[i, "times"] will
        # store a list inside the list
        for time in row["times"]:
            all_times.append(time)
    # We can now calculate the mean time of this dataframe and plot the
mean line before going to the next df
    mean_time = sum(all_times) / len(all_times)
    plt.axhline(sum(all_times) / len(all_times), color='r',
linestyle='dashed', linewidth=2)

```

```

# This is for consistency in scale between all the graphs
# This must be in the loop because we're doing this for all 3 of
them
ax.set_xlim(0, 60)
ax.set_ylim(0, 6)
ax.set_xlabel("Time (s)")
ax.set_ylabel("Delay between presses (s)")
print("- %s is %f seconds" % (dataframe.name, mean_time))
# We also add some values for the bar plot
x2.append(mean_time)
y2.append(dataframe.name)

# Now we prepare the bar graph
ax2.set_ylim(1.35, 1.475)
ax2.bar(y2, x2)
ax2.set_title("Mean delay between button presses\n (lower is better)")
ax2.set_xlabel("Music genre")
ax2.set_ylabel("Mean delay (s)")

# Now we show all the plot we created
plt.show()

# group the data by genre and calculate the means and std
genre_groups = df.groupby("genre")
mean_times = genre_groups.mean(numeric_only=True)
mean_std = genre_groups.std(numeric_only=True)
print("\nMean time for each genre :\n", mean_times, "\n")
print("Standard deviation for each genre :\n", mean_std, "\n")

# We create some chart to put into the final report based on the mean
time and std obtained above
fig3, ax3 = plt.subplots()
ax3.set_ylim(39, 44)
ax3.set_title("Mean score\nHigher is better")
ax3.set_xlabel("Music genre")
ax3.set_ylabel("Score")
plt.bar(["Rap", "No sound", "Calm music"], [43.333333, 40.500000,
42.384615])

fig4, ax4 = plt.subplots()
ax4.set_ylim(0, 6)
ax4.set_title("Standard deviation\nHigher value indicates more spread
data")
ax4.set_xlabel("Music genre")
ax4.set_ylabel("Standard deviation")
plt.bar(["Rap", "No sound", "Calm music"], [5.472877, 5.244044,
3.428762])

plt.show()

```