



Interface tactiles et interfaces réparties sur plusieurs supports

Course de haies interactive

Contexte	2
Solution proposée	2
Idée de la solution	2
5W1H	3
Fonctionnement	5
Equipement nécessaire	5
Réalisation de la solution	7
Aspects techniques	7
Répartition du travail	9
Planning	10
Scénario final	10

Julien MAUREILLE
Gregory MERLET
Pierre RAINERO
Gauthier TOUSSAINT

1. Contexte

Le musée national du Sport (<http://www.museedusport.fr/>) situé à Nice s'articule autour de plusieurs espaces consacrés au sport :

- Un espace : Défi sur soi consacré à des sports individuels ;
- Un espace : Défi d'homme à Homme consacré aux sports qui se pratiquent à deux ;
- Un espace : Défi collectif avec une grande partie consacrée au football ;
- Un espace : Défi au-delà des limites consacré à des excursions ou exploits qui ont marqué le monde du sport ;
- Un espace réservé aux expositions temporaires.

Le musée souhaite être un exemple en termes de solution "HighTech". Il souhaite également attirer des groupes à des fins pédagogiques (scolaires), des experts (à travers des activités adaptées pour eux), des familles et des sportifs en leur faisant vivre des expériences inoubliables (présences d'un escape game mais aussi utilisation de nouvelles technologies) .

L'objectif est de proposer une solution répondant à ces attentes, à savoir, faciliter et rendre les visites ludiques et pouvoir mieux comprendre les points forts et faibles de leur musée en analysant les visites.

2. Solution proposée

a. Idée de la solution

Afin de rendre le musée toujours plus interactif, nous souhaitons proposer une solution qui fait participer activement les visiteurs, de manière collective. Nous sommes particulièrement intéressés par les thématiques "Défi sur soi consacré aux sports individuels" et "Défi au-delà des limites consacré à des excursions ou exploits qui ont marqué le monde du sport".

La première car c'est un espace où il semble difficile de faire participer les visiteurs collectivement (par la nature même du sport individuel), c'est donc un vrai défi à relever.

La seconde car souvent c'est une thématique souvent sous-exploitée où l'information est présentée de manière textuelle sans vraiment que les visiteurs y portent un grand intérêt. Là aussi c'est un vrai défi de rendre cet espace interactif.

Dans le cadre de ces deux thématiques, nous proposons au visiteur une expérience inédite qui va lui permettre de faire un 110 mètres haies avec le recordman mondial et se comparer à lui sur le temps mais aussi sur le rythme cardiaque tout au long de la course. Afin d'ajouter encore plus d'interaction, la course pourra accueillir plusieurs visiteurs simultanément et ceux-ci pourront se comparer entre eux en temps réel. C'est une façon de rendre le sport individuel un peu plus collectif : on imagine des amis venus visiter le musée se lancer un défi entre eux.

b. 5W1H

On peut distinguer deux tâches différentes dans notre application, la première étant la course en tant que telle et la deuxième étant la phase de résultat.

1. Phase de course

WHO (utilisateurs) :

Nous visons avec cette application principalement les visiteurs en petits groupes, de deux à cinq personnes (ou seuls). Ceux qui sont en tenus décontractés et qui voudraient s'essayer à une petite épreuve sportives pour tester leur performance sur une simulation de 110m haies.

WHICH (type de tâche) :

C'est une épreuve qui peut aussi bien se jouer tout seul qu'à deux pour se confronter à une autre personne. L'enjeu principal est évidemment de finir la course en premier mais il peut y avoir également d'autres objectifs comme faire tomber le moins de haies possible, faire le meilleur temps possible ou encore voir comment son cœur gère une activité physique.

WHERE (territorialité) :

(voir "d. Équipement nécessaires")

WHEN (temporalité) :

Cette tâche est la tâche initiale de l'application. La tâche suivant est la phase de résultat, il n'y a pas de tâche en simultanée.

WHAT (données) :

Les données manipulées par cette tâche sont les données récoltées par la Kinect à savoir les mouvements des participants, et les données récoltées par les montres connectées (rythme cardiaque).

HOW (Interaction et visualisation) :

Les interactions sont réalisées au travers des mouvements des participants, par exemple pour lancer la course ils doivent tous les deux lever le bras droit pendant quelques secondes. Nous avons pris la décision de n'avoir aucune interaction tactile pour permettre aux participants de jouer le plus simplement possible et rendre l'application la moins contraignante pour l'utilisateur. Les interactions sont donc uniquement gestuelles et sonores.

2. Phase de résultat*WHO* (utilisateurs) :

Pour cette tâche nous visons évidemment les participants de la course, mais également les visiteurs qui pourront se trouver autour des joueurs. En effet comme les résultats et la course seront affichés sur un grand écran grâce au projecteur, les visiteurs du musée pourront observer la course et regarder les résultats des participants en les comparant aux résultats du record mondial.

WHICH (type de tâche) :

C'est une tâche qui se veut être un moment où les participants peuvent partager leurs impressions et analyser les résultats entre eux, et le public peut aussi y être convié.

WHERE (territorialité) :

(voir "d. Équipement nécessaires")

WHEN (temporalité) :

Cette tâche est la tâche finale de l'application. Une fois que cette tâche est terminée, l'application va recommencer et la phase de course va débuter.

WHAT (données) :

Les données de cette tâche seront les résultats de la phase 1.

HOW (Interaction et visualisation) :

Cette tâche se terminera au bout d'un certain temps automatiquement, les résultats seront affichés pendant x secondes puis une nouvelle course pourra démarrer.

c. Fonctionnement

L'activité consiste à courir sur place devant un écran qui affiche en temps réel les informations de la course ainsi qu'une visualisation des coureurs sur une piste de course sur laquelle sont alignés 10 haies (comme pour l'épreuve olympique). La vitesse de course sur place du joueur va influencer sur la vitesse du coureur virtuel le représentant, et les sauts effectués par le joueur sont également effectués par son double virtuel. Ainsi le joueur doit atteindre la ligne d'arrivée le plus rapidement possible en essayant de ne pas renverser de haies.

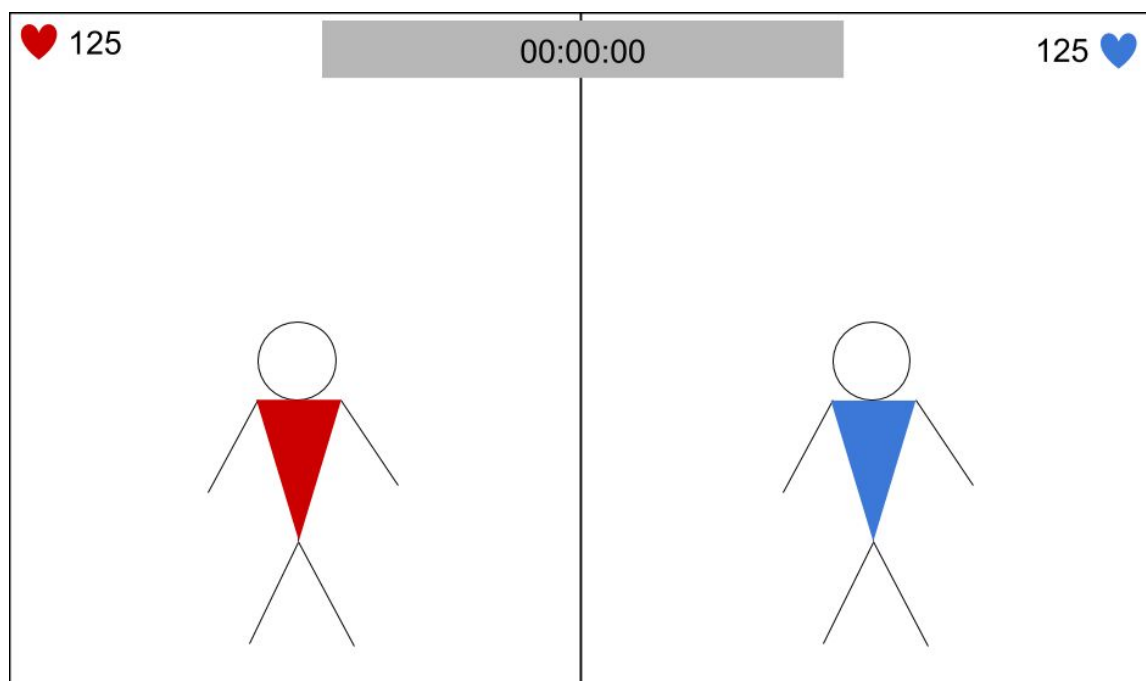


Figure 1 - Affichage proposée aux joueurs pour suivre leur course

d. Equipement nécessaire

Deux couloirs d'une piste d'athlétisme seront tracés au sol, chaque participant se placera dans un couloir et pourra mettre à son poignet une montre connectée. Les montres seront placées de part et d'autre des couloirs sur des supports et leurs bracelets seront de la couleur de leur athlète (rouge ou bleu).

La pièce accueillant l'activité devra donc contenir :

- Un dispositif d'affichage (écran ou vidéoprojecteur avec écran) ;
- Un dispositif permettant de suivre les mouvements des 2 participants (Kinect ou caméra par exemple) ;
- Deux couloirs d'athlétisme tracés au sol sur lesquels les participants devront se placer ;
- Deux montres connectés avec bracelet rouge et bleu ;
- Des supports pour les montres de chaque côté des couloir.

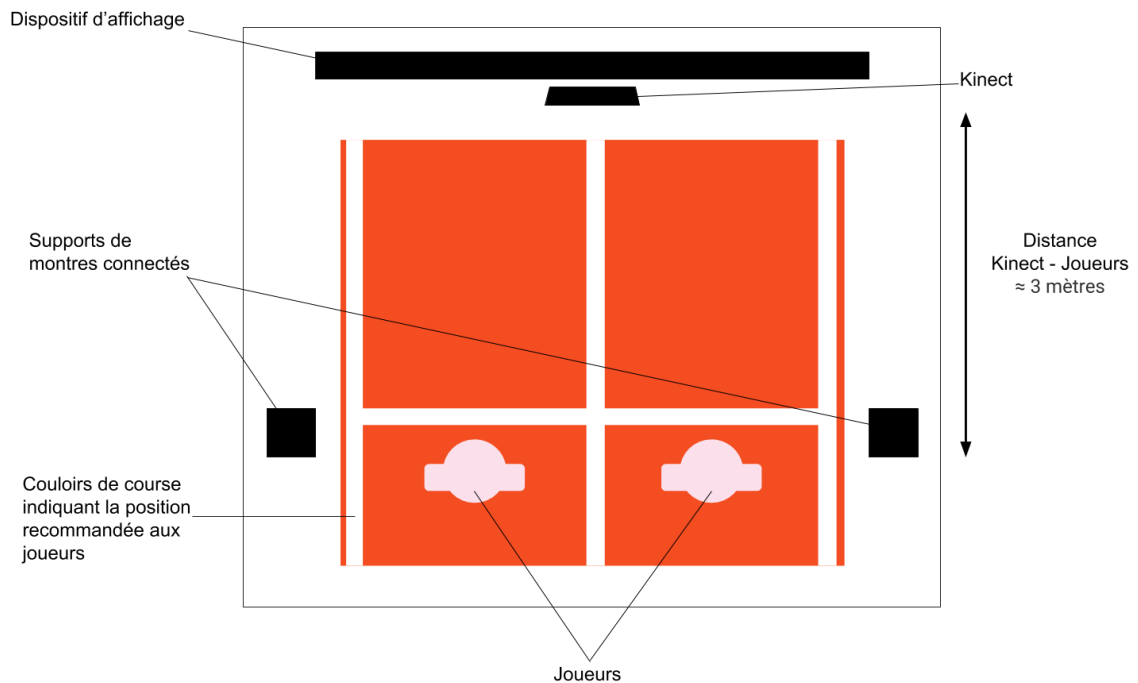


Figure 2 - Représentation spatiale de l'installation du système

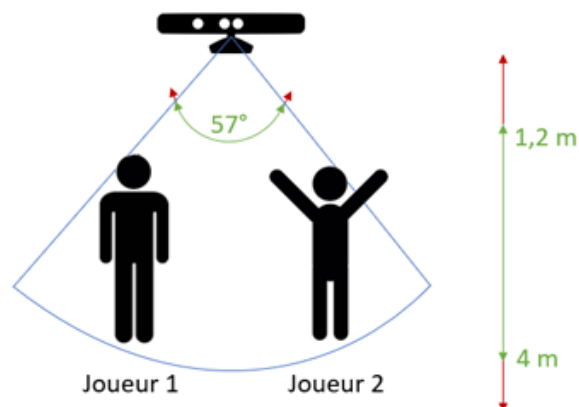


Figure 3 - Distance de détection de la Kinect (distance préférée 3 à 3,5 mètres)

3. Réalisation de la solution

a. Aspects techniques

Au vu de la [Solution proposée](#), on est face au système suivant :

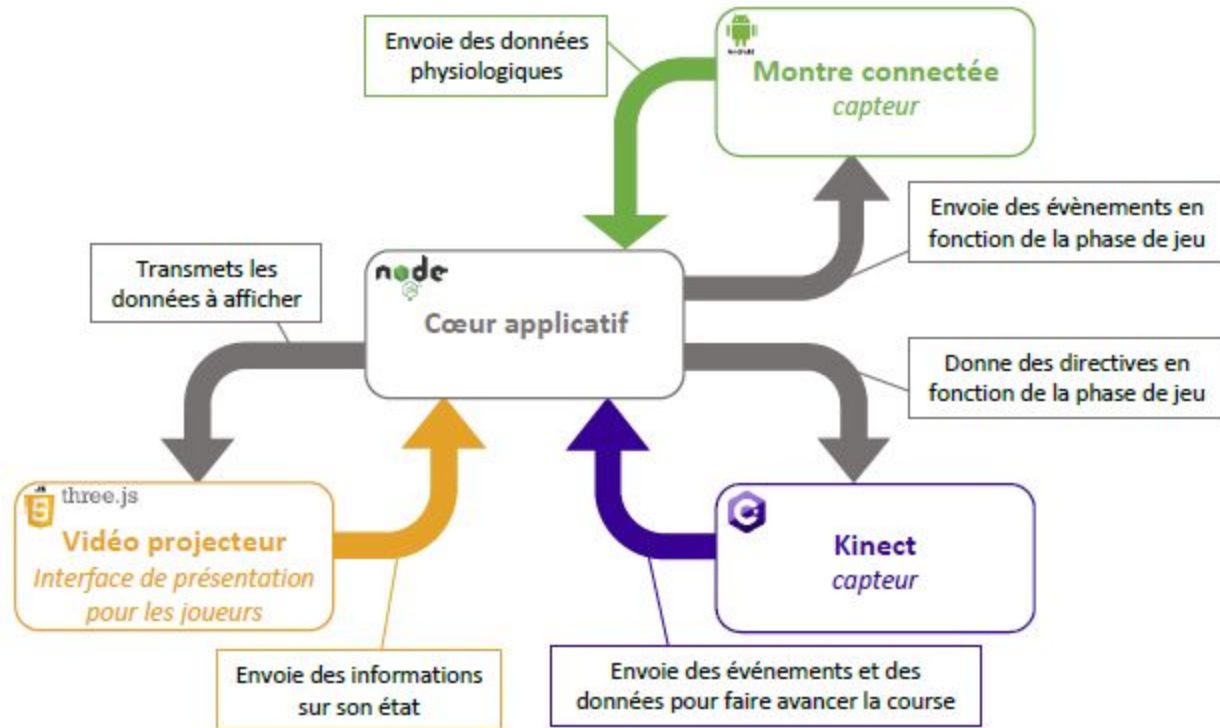


Figure 4 - Représentation globale de l'architecture du système et des interactions entre les différentes parties

L'application va donc partager un cœur applicatif unique qui sera soumis à des contraintes de performances fortes. L'épreuve étant assez courte et les échanges permanents, il est nécessaire d'avoir un temps de réponse rapide entre les différentes parties du système, surtout que l'épreuve est en temps réel. Nous avons donc porté notre choix sur un langage de programmation offrant un temps de démarrage et de réponse élevé mais d'un assez haut niveau : "[Node.js](#)". Ce langage se base sur du [Javascript](#) et le paradigme de programmation orienté objet que nous avons déjà rencontré de nombreuses fois, ce qui nous permettra de ne pas perdre trop de temps d'apprentissage (le cycle de développement du projet étant court). Il permet également de communiquer aisément avec des "socket" ("[socketIO](#)"), ce qui est très utile dans le cas d'une communication en temps réel.

La Kinect est un capteur tout à fait adapté à la tâche demandée puisqu'elle est capable d'extraire un squelette de points pour une personne filmée et possède un bon taux de rafraîchissement. De base prévue pour une programmation en C++, C# ou VB.NET il existe pour la version 2 une librairie Javascript permettant de l'exploiter ([kinect2](#)). Un langage commun entre deux parties du système nous aurait une fois encore permis de gagner du temps de développement. Malheureusement les contraintes matérielles nous ont poussé à nous tourner vers la première version de la Kinect, limitant ainsi le nombre de joueurs supportés à deux et nous forçant à écrire une partie du système en C#. L'absence de paquet "[NuGet](#)" pour permettre une communication via des "socketIO" entre le coeur applicatif et cette partie du système a également coûté cher en temps de développement. Il a été nécessaire de passer par une "websocket" générique spécialement configurée pour supporter cette communication.

Pour notre interface utilisateur, à savoir le projecteur, nous nous sommes tournés vers "[Three.js](#)". Il s'agit d'une librairie Javascript permettant d'afficher une scène 3D et d'y effectuer des animations. Ce choix a été en grande partie motivé par le fait qu'un membre du groupe avait déjà une connaissance de cette librairie et était donc capable d'apporter son aide aux autres. Le choix d'une visualisation 3D permettant une plus grande immersion dans l'épreuve (en particulier avec une caméra à la troisième personne suivant notre représentant virtuel voir *Figure 1*). Le reste de l'interface est réalisé en HTML et CSS.

La montre connectée sert ici de capteur supplémentaire pour enrichir l'expérience utilisateur. Elle permet en effet de relever le rythme cardiaque du participant et à envoyer cette valeur au backend pour ensuite l'afficher sur le projecteur. À la fin de la course les fréquence minimum, moyenne et maximum seront affichés à la fois sur la montre mais également sur le projecteur. De manière à accéder le plus aisément possible aux capteurs de la montre nous avons donc opté pour le langage natif du dispositif, à savoir [Android](#) (Java). Les deux montres connectées seront associés à un même smartphone android qui permettra de faire le lien entre les montres et la backend sachant qu'une montre ne peut communiquer qu'avec le smartphone auquel elle est associée.

Nous devons également rendre les données échangées interprétables par le coeur applicatif. Chaque partie aura alors la responsabilité de les transcrire dans un format compréhensible par ce dernier en retirant toutes les informations superflues de manière à alléger les échanges (toujours dans une optique de performance).

b. Répartition du travail

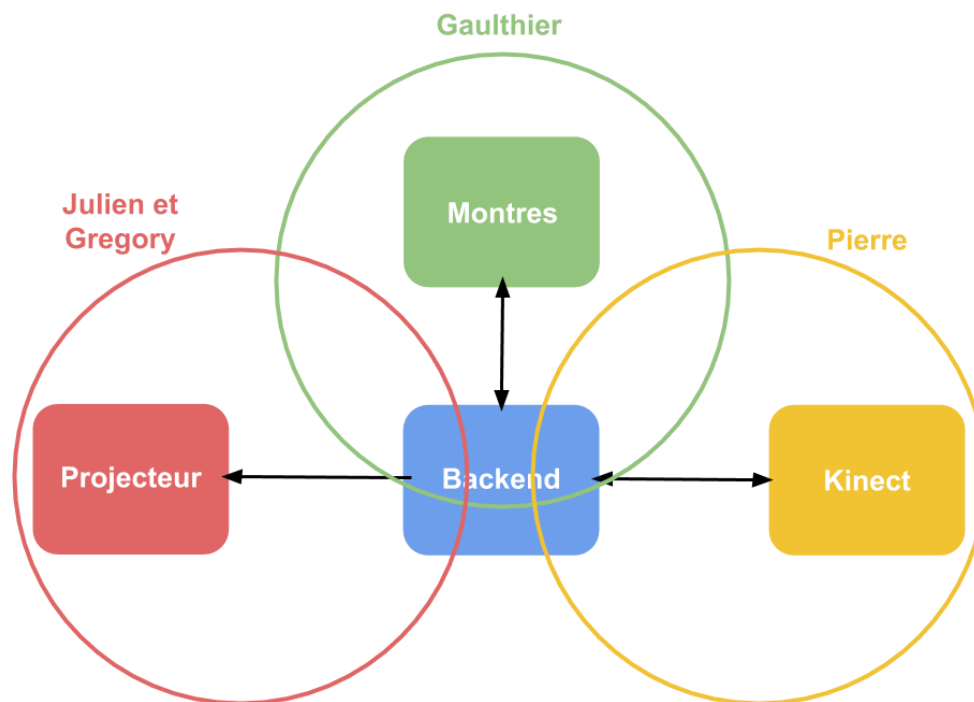


Figure 5 - Découpage du système pour chaque membre

Chaque membre aura en charge un dispositif du système, la partie du coeur applicatif qui gère ce dispositif ainsi que la communication entre les deux. Ainsi, aucun membre ne s'occupe intégralement du coeur applicatif. L'objectif est que chaque membre ait connaissance de la globalité du système (en particulier par le biais du coeur applicatif), tout en spécialisé par une sous-partie de ce dernier. De cette manière le développement est très souple et permet aux membres d'avancés en parallèle sur une même tâche à terminer pour une démonstration donnée (voir [Planning](#)).

La répartition des "modules" en fonction des membres s'est faite par rapport aux connaissances de chacun, l'idée était que chaque membre travaille sur une technologie qu'il ne connaissait pas (dans l'optique de se former) tout en pouvant demander de l'aide aux membres qui avaient déjà eut à faire cette technologie.

c. Planning

Semaine du 21/12	Découverte du sujet, proposition de projet, écriture des spécifications.
Semaines du 11/01 et 18/01	Prise en main et apprentissage des nouveaux outils (Kinect, développement sur montre connectée et développement 3D). Mise en place de l'architecture du projet et première version du coeur applicatif et de la communication avec tous les dispositifs.
Semaines du 25/01 et 01/02	Implémentation successive des fonctionnalités (sur les dispositifs et portions de coeur métier) à savoir détection des sauts, détection de la vitesse, détection des pulsations cardiaques et affichage de modèles 3D avec animations.
Semaine du 08/02	Améliorations après retours utilisateurs.
Semaine du 15/02 et 24/02	Préparation démonstration finale et montage de la vidéo.

d. Scénario final

- L'écran de projection affiche un exemple de partie montrant l'athlète courant seul (afin de donner envie aux visiteurs de venir courir avec lui) accompagné d'un message indiquant que le système est en attente de participants.
- L'écran de projection affiche un message indiquant que le système est en attente de participants, les invitant à se placer face au capteur.
- Les joueurs se placent sur les couloirs de couleur situés face au mur de projection : un joueur par couloir.
- Les joueurs mettent à leur poignet la montre associée à leur couloir.
- L'écran de projection affiche un message indiquant que le système a détecté les joueurs. Il affiche également un message invitant les joueurs à lever la main droite dès qu'ils sont prêt.
- Chaque joueur lève la main droite.
- Le personnage virtuel représentant le joueur lève la main.
- Le système détecte le mouvement (Kinect) et lance la partie. L'écran de projection affiche un message "3... 2... 1 C'est parti" et lance le chronomètre pour la phase de

course. Les montres connectées émettent une vibration pour indiquer aux joueurs que la course commence. L'écran affiche désormais une vue composée de :

- Le chronomètre en haut ;
 - Une vue par joueur à la troisième personne représentant le coureur sur une piste de course avec des haies (écran divisé) ;
 - Une mini-carte d'avancement en bas à droite ou au centre de l'écran en fonction du nombre de joueurs.
- Les joueurs doivent alors courir sur place afin de faire avancer leur personnage. Ils doivent également sauter pour éviter les haies qui se présentent à eux. Le vidéoprojecteur indique le meilleur moment où le joueur doit sauter et les montres connectées émettent une vibration si une haie est percutée.
 - Un joueur virtuel, imitant les performances d'un athlète de haut niveau, court en même temps que les participants. Sa progression est renseignée en temps réel sur la mini-carte. Ce joueur virtuel suit les performances du recordman en titre à savoir Aries Merritt.
 - Les participants portant une montre ont accès tout au long de la course à des données biométriques comme leur rythme cardiaque.
 - Dès qu'un joueur passe la ligne d'arrivée virtuelle, sa vue à la troisième personne affiche un message "Terminé !" accompagné de son classement.
 - Lorsque tous les joueurs ont passé la ligne d'arrivée, le dispositif de projection bascule sur l'écran de résultats. Ce dernier affiche pour chaque joueur son classement, sa vitesse moyenne, le nombre de haies qu'il a passé avec succès et son rythme cardiaque moyen. Il affiche également à côté les résultats (classement, vitesse moyenne, rythme cardiaque moyen) du joueur virtuel imitant un athlète.
 - Les participants ont également un récapitulatif de leur rythme cardiaque sur leur montre, ils peuvent voir leurs pulsations cardiaque min, moyenne et max.
 - Au bout d'une minute, le système est réinitialisé et l'écran de projection affiche un message indiquant que le système est en attente de participants pour une nouvelle partie.