

Document d'initialisation

Implémentation d'un système d'aide à la vision sur des lunettes de réalité virtuelle

- **Client** : Olivier BODINI.
 - **Équipe de suivi** : Thierry HAMON et Sophie TOULOUSE.
 - **Groupe** : Mohamed Ali YACOUBI, Flavien HAMELIN, Safa KASSOUS, Farah CHERIF et Saad AMMARI.
-



Sommaire

1. Etude de l'existant	2
2. Critiques	3
3. Solution proposée	4
4. Vue conceptuelle	6
4.1. Diagrammes de cas d'utilisation	6
4.2. Diagramme de séquences	11
5. En direction de l'élaboration 1	13

1. Etude de l'existant

Dans cette section nous allons recueillir les informations des différentes parties existantes telles que l'environnement technique et logiciel. Le système d'exploitation open source que nous allons traiter est Moverio OS des lunettes de réalité virtuelle Moverio BT 300.

Le système assure déjà les fonctionnalités suivantes :

Générer une cartographie

- Affichage des points lumineux : l'unité de traitement fait appel à une primitive permettant de trouver une zone non traitée, une fois cette zone récupérée, l'application affiche un point sur l'écran.
- Réponse du patient : l'unité de traitement reste en attente d'une réponse de l'utilisateur pendant un court laps de temps, dès que l'utilisateur clique sur la télécommande connectée aux lunettes, le système comprend que le point a été vu, l'unité de traitement marque la zone comme visible jusqu'au prochain test. Si l'utilisateur ne clique pas, la zone est considérée comme non visible et permet la cartographie de la zone lésée de l'œil.
- Mémorisation des réponses de l'utilisateur : l'unité de traitement enregistre, dans une structure de données, les coordonnées du point et si celui-ci est visible ou non par l'utilisateur.
- Délimitation des zones lésées : l'unité de traitement utilise la structure de données créée à travers les réponses de l'utilisateur et génère la cartographie en fonction des informations stockées (points vus et non vus).
- Sauvegarde de la cartographie : sauvegarde de la cartographie de l'utilisateur dans l'espace de stockage des lunettes.



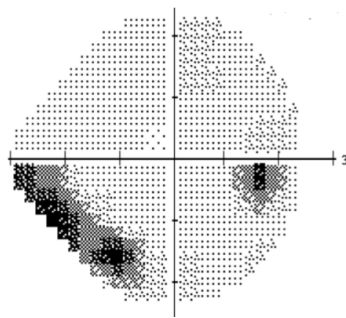
2. Critiques

En se basant sur l'existant, nous avons remarqué les points suivants :

- Le test prend énormément de temps pour générer une cartographie.
- La solution produit des résultats binaires (deux choix : voir ou ne pas voir le point), ces résultats ne mettent pas en valeur le fait qu'il y a des zones où le patient ne voit pas très bien mais voit encore (vision brouillée).
- La forme de la cartographie générée n'est pas très appréciée par le client, une cartographie utilisant un modèle de champ de vision est préférée.
- Les points lumineux durant le test ne changent pas de taille ni de luminosité.
- La phase de test est basée sur un algorithme naïf qui ne recherche pas concrètement la zone lésée et qui n'utilise pas d'apprentissage.


3. Solution proposée

Notre solution consiste à améliorer la partie : “générer une cartographie”, qui doit être différente de la version précédente. Une préférence a été soulignée par le client, celle de générer une cartographie utilisant le modèle de champ de vision. Ce modèle correspond aux résultats que nous pouvons avoir lorsque nous effectuons des tests chez l'ophtalmologue (Figure ci-dessous).



Nous allons utiliser ce qui a déjà été fait. A chaque affichage de points lumineux, si l'utilisateur le voit, il doit cliquer sur la télécommande pour ainsi déterminer les zones lésées et les zones non lésées. Nous devons ajouter une certaine nuance au niveau des points lumineux, ils ne doivent pas avoir tout le temps la même luminosité ni la même taille et le temps d'affichage de chacun des points ne doit pas être périodique. Nous devons jouer sur la taille des points, la luminosité et la position d'affichage pour déterminer les différentes nuances de gris correspondantes à la détérioration de l'œil : si une zone est noire il y a une incapacité totale de vision, si une zone est grise il y a un brouillage de la vision, si une zone est blanche la vision n'est pas entachée par la dégénérescence maculaire liée à l'âge. Cette phase de tests et la génération de la cartographie doit se faire en réduisant le temps global qu'il faut avec l'application déjà existante.

Après avoir pu générer une cartographie, nous allons tout d'abord réduire les dimensions de l'image filmée par la caméra des lunettes et décaler cette image dans une zone où la vision est encore possible. Ensuite nous pourrons améliorer cette approche en utilisant des algorithmes de segmentation d'image comme l'algorithme de ligne de partage des eaux pour permettre à l'utilisateur d'avoir une idée de ce qui est présent



autour de lui grâce aux contours surlignés des objets sans avoir à regarder constamment le retour de la caméra.

Après cette étape, nous aimerions aborder un aspect nouveau, nous voudrions permettre à l'utilisateur de ne voir qu'à travers la caméra, ce qui se résume à utiliser un casque de réalité virtuelle et plus des lunettes, la caméra filmerait l'environnement, une tâche noire fait partie de la vision de l'utilisateur et il s'agirait de déformer l'image autour de la tâche en restituant toutes les informations présentes dans l'image filmée. L'image perçue à travers ces lunettes serait différente de celle perçue normalement mais nous comptons sur la capacité d'adaptation et la plasticité du cerveau humain pour permettre à l'utilisateur de s'adapter à cette vision et ainsi utiliser les zones non-lésées de l'oeil pour obtenir toutes les informations de l'environnement.



4. Vue conceptuelle

Dans cette partie, nous présentons la vue conceptuelle de notre projet.

4.1. Diagrammes de cas d'utilisation

Les cas d'utilisation concernent ce que le système doit faire et décrivent un comportement souhaité indépendamment des détails de l'implémentation. L'objectif des cas d'utilisation est de capturer toutes les fonctions au niveau du système que les acteurs envisagent.

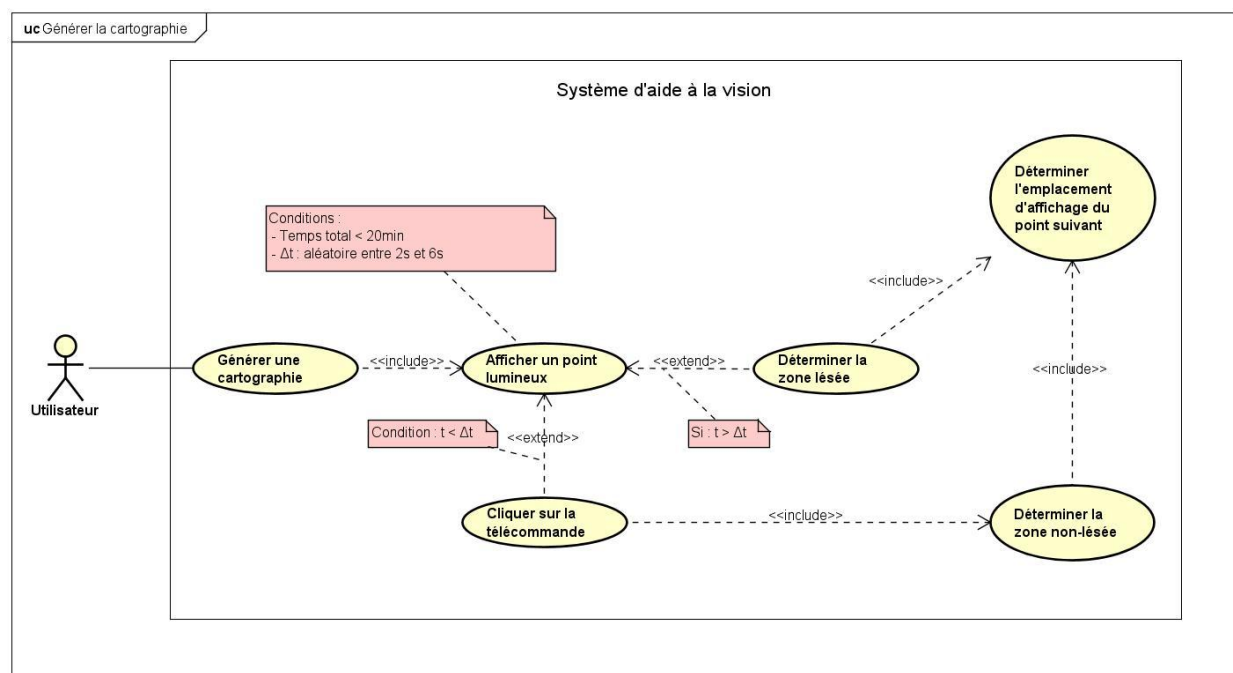
L'acteur : "utilisateur" cité dans les cas d'utilisation représente le patient souffrant de DMLA. Ce dernier peut voir à travers les verres des lunettes, allumer la caméra, charger la cartographie, et visionner l'image de la caméra sur les verres des lunettes. Nous avons hiérarchisé les cas d'utilisation selon les priorités du client.

Objectif 1 (Plus prioritaire) : Générer la cartographie

L'utilisateur peut choisir de générer une cartographie, ainsi, la phase de test commence. L'unité de traitement va afficher un point lumineux, l'utilisateur doit répondre en cliquant sur la télécommande s'il voit le point. En fonction de la réponse, un algorithme d'apprentissage dédié va déterminer les zones lésées, les zones qui ne sont pas lésées et les frontières de brouillage de vision entre ces zones. Le placement du point suivant dépend de la réponse de l'utilisateur.

$\Delta t \in [2s, 6s]$ correspond à la différence de temps entre le début de l'affichage du point suivant et le début de l'affichage du point actuel.

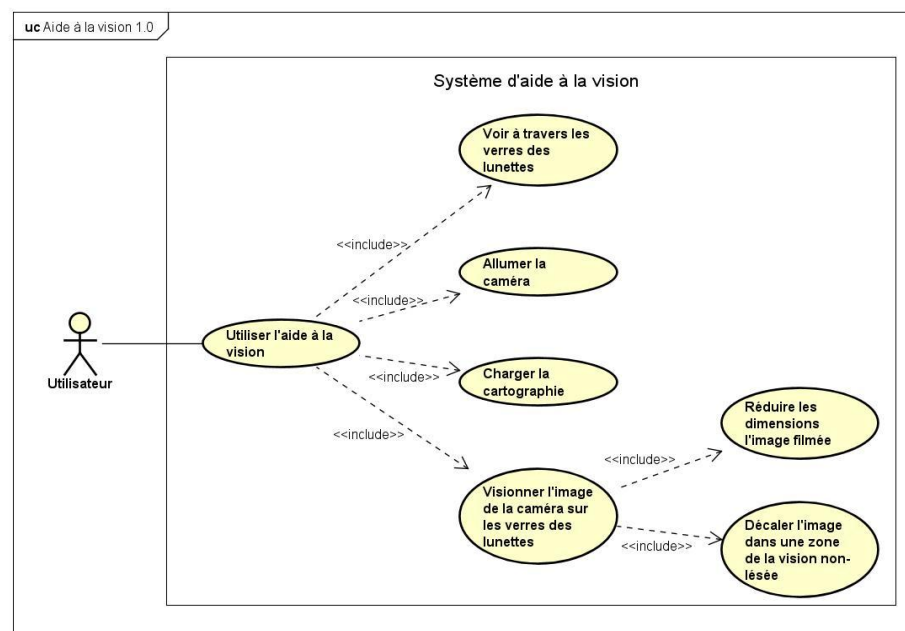
Le temps total du test permettant de générer une cartographie ne doit pas dépasser 20 minutes.



Objectif 2 : Aider à la vision en décalant l'image dans une zone de vision non lésée.

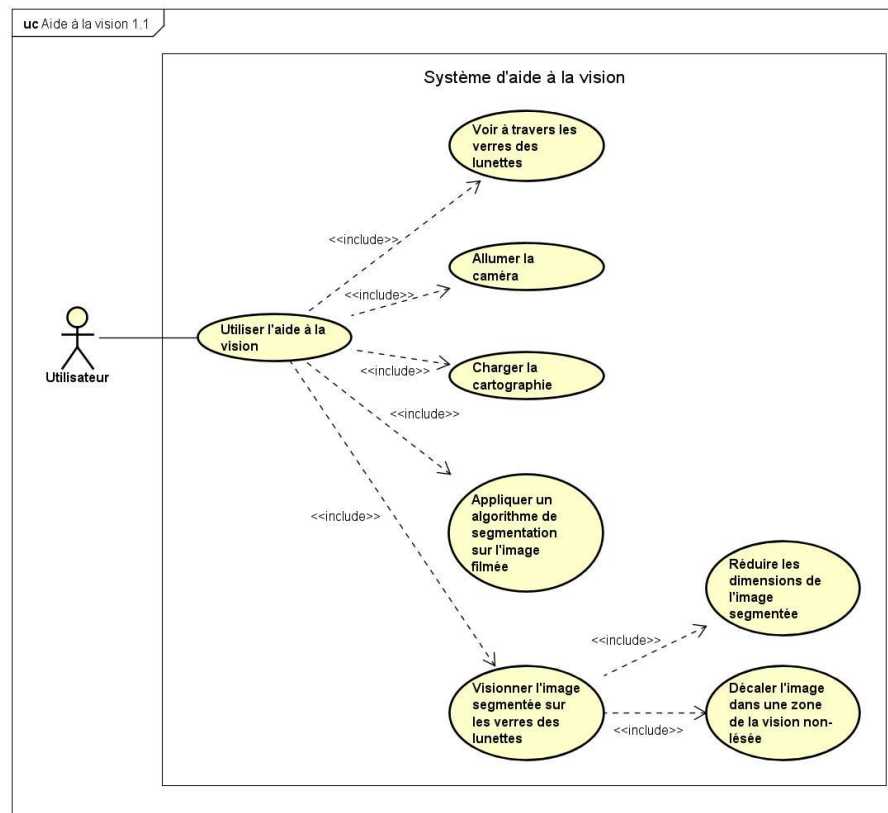
L'utilisateur utilise le système d'aide à la vision en portant des lunettes de réalité virtuelle pour voir les points lumineux affichés sur les verres de ces dernières. A chaque fois, l'utilisateur doit signaler quand il voit un point lumineux pour générer à la fin une cartographie bien précise qui indique les zones plus ou moins lésées.

L'utilisateur peut allumer la caméra pour visionner l'image de la caméra sur les verres des lunettes. Pour cela, le système doit assurer la réduction des dimensions de l'image filmée et décaler l'image dans une zone non lésée afin d'assurer la vision à l'utilisateur. Par exemple, si l'utilisateur a un problème de vision au centre, le système décale l'image affichée sur les verres dans un coin à gauche ou à droite.



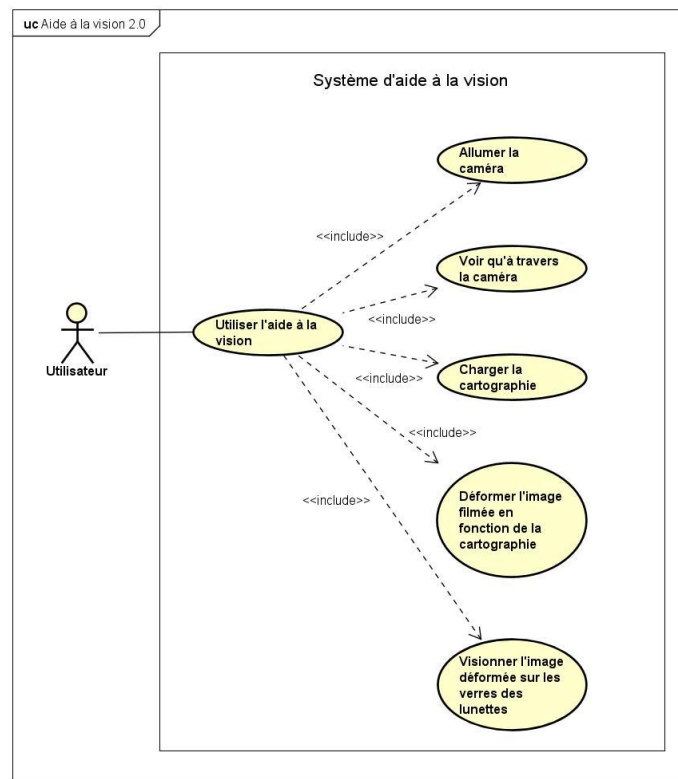
Objectif 3 : Aider à la vision en décalant l'image dans une zone de vision non lésée tout en segmentant l'image filmée par la caméra

Il s'agit d'une amélioration de l'objectif 2. Le système doit prendre en compte la segmentation des objets dans l'affichage de l'image renvoyée par la caméra.



Objectif 4 (Moins prioritaire) : Aider à la vision en déformant l'image en fonction de la cartographie.

Le système doit déformer l'image filmée en fonction de la cartographie générée par l'utilisateur. L'utilisateur se sert des zones non lésées de son œil pour visionner une image déformée autour de la zone lésée, cette image déformée doit restituer toutes les informations d'une image obtenue sans déformation. La capacité d'adaptation du cerveau à une représentation nouvelle du monde est sollicitée.



4.2. Diagramme de séquences

Les diagrammes de séquences représentent graphiquement les interactions entre les acteurs et le système suivant un ordre chronologique.

Le diagramme suivant décrit le fonctionnement de la génération d'une cartographie. L'utilisateur demande de générer une cartographie, par conséquent notre système affiche des points lumineux positionnés aléatoirement dans l'espace.

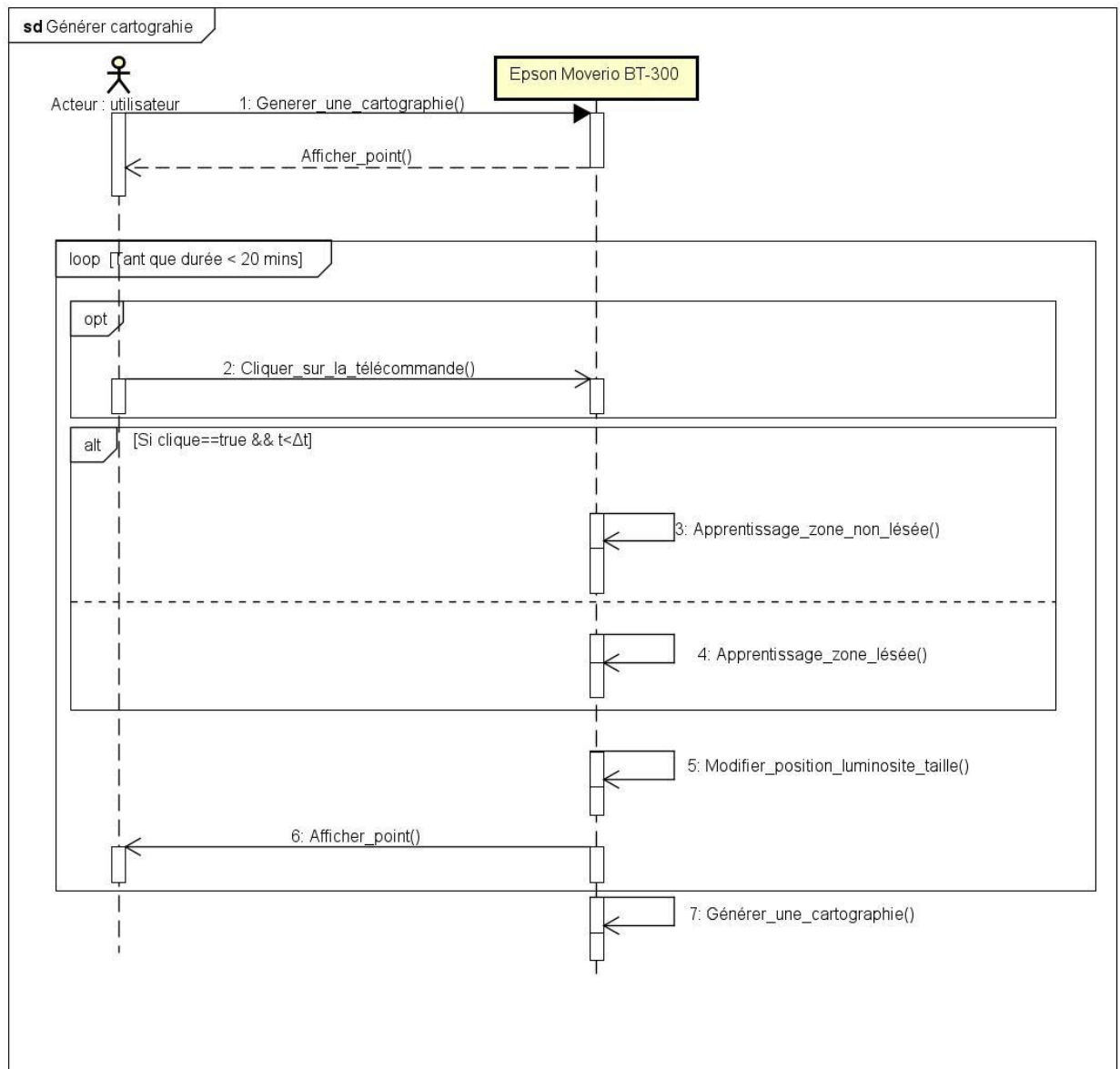
Tant que la durée de la génération d'une cartographie n'a pas dépassé les 20 minutes, l'utilisateur peut cliquer sur la télécommande dès qu'il voit un point lumineux affiché.

Supposons que nous affichons un point P de luminosité I , de position (x,y) et de rayon r . Si l'utilisateur clique sur la télécommande pendant un certain temps aléatoire Δt appartenant à l'intervalle $[2s,6s]$ correspondant à la différence de temps entre le début de l'affichage du point suivant et le début de l'affichage du point actuel alors nous considérons que le point $P[I,(x,y),r]$ a été vu et nous stockons cette information. L'ensemble des informations recueillies (points vus et non vus) permet d'actualiser l'apprentissage des zones lésées et non lésées à travers les fonctions éponymes, ces fonctions permettent également de dessiner une frontière entre les zones et de déterminer les nuances de gris qui correspondent à la détérioration partielle de la vision, ces nuances de gris sont déterminées grâce aux variations de luminosités et de tailles des points vus et non vus.

Les réponses enregistrées au cours du test doivent permettre d'utiliser une méthode d'apprentissage par renforcement afin de décider du triplet (luminosité, position, taille) le plus intéressant à affecter au prochain point à afficher, ce choix doit se faire à travers la fonction `5.Modifier_position_luminosite_taille()`.

Le test doit durer 20 minutes au total, un certain nombre d'informations manqueraient pour générer une cartographie convenable. Dans l'hypothèse où l'affichage d'un point dure en moyenne quatre secondes, on pourrait afficher 300 points sur l'ensemble du test. Nous avons l'idée de calculer les nuances de gris manquantes en fonction des points voisins. La moyenne des informations recueillies sur les points voisins pourrait être une solution au manque d'information sur un point donné. Ce calcul final

ainsi que l'édition de la cartographie se feraient à travers la fonction 7.Générer_une_cartographie().





5. En direction de l'élaboration 1

Pour aboutir à nos objectifs, nous avons commencé à analyser le code déjà existant. Nous nous consacrons principalement à améliorer la phase de génération de cartographie en utilisant Android Studio comme environnement de développement. L'exécution du code source peut se faire sur nos appareils mobiles et sur l'émulateur d'Android Studio, de ce fait chacun des membres du groupe peut tester l'application sans devoir l'installer sur les lunettes à chaque mise à jour. Nous gérons les différentes versions du système avec GitHub.