

# Smart Mirror Interface – SMI

---

## Description du Projet

### Présentation



Aujourd'hui, avec l'explosion de l'informatique ambiantes, on voit nos habitations se peupler de nombreuses solutions domotiques et d'objets connectés/communisants (OC). Cependant, actuellement, seul des interfaces web propriétaires ou pour smartphones permettent de configurer et piloter les OC.

Le but de ce projet est développer une « Smart Mirror Interface ». « Smart Mirror Interface » sera une interface visuelle intégrée dans un miroir et par conséquent elle ne pourra ni utiliser un clavier ni utiliser une souris. De plus, de manière à avoir un miroir propre (sans trace de doigts), l'interface ne pourra pas être tactile non plus.

### Objectifs

- **Objectif 1 : Créer une interface composée de tuiles dynamiques.** Il s'agit de l'objectif principal du projet «Smart Mirror Interface». En effet, chaque service attaché au « Smart Mirror » se verra associé à une tuile qui affichera dynamiquement l'information le concernant. Par exemple, une station météo affichera la température, la précision atmosphérique, le taux d'humidité ... Bien entendu on pourra ajouter ou supprimer des services sans avoir besoin de redémarrer l'application.
- **Objectif 2 : Contrôle vocal de l'application.** Puisque le « Smart Mirror » ne possède ni souris, ni clavier, ni même une interface tactile, il faut fournir un moyen à l'utilisateur d'interagir avec celui-ci. Le choix de disposer d'un contrôle vocal devient donc nécessaire. Les commandes vocales commenceront par un mot clé (configurable) comme cela est le cas sur Xbox 360. Cela permettra d'éviter d'interpréter des conversations comme des commandes pour le « Smart Mirror ».
- **Objectif 3 : Reconnaissance de visages.** Le « Smart Mirror » est avant tout un miroir. La « Smart Mirror Interface » sera donc non visible sauf si l'utilisateur en fait la demande et qu'il est reconnu par le système. Pour cela le système dispose d'une caméra vidéo enregistrant qui se trouve devant le miroir. Il faudra donc, lors de la commande vocale d'affichage, être capable de vérifier si la personne devant le miroir est une personne connue. Si oui on affichera la « Smart Mirror Interface », sinon on ne fera rien.
- **Objectif 4 : Intégration des services.** De manière à illustrer le fonctionnement des tuiles et du contrôle vocal, on agrémentera la « Smart Mirror Interface » de quelques services (1 à 3) supplémentaires. Les services liés au contrôle vocal et à la reconnaissance de visages ne compte pas comme des services supplémentaires.

### Scénario(s)

- N'hésitez pas à consulter la vidéo disponible dans les liens de la bibliographie. Celle-ci n'a pas pour vocation à présenter les services attendus ni même l'esthétique de l'interface graphique. Le but de cette vidéo est d'illustrer les usages.

## Critères de succès

- **Critère 1 : Les objectifs 1 à 3 sont atteints.** C'est-à-dire possibilité d'utiliser l'interface du miroir pour configurer les modules vocaux et visuel, l'activation de l'interface à la voix en fonction de la personne présente devant le miroir ainsi que le fait de pouvoir ajouter des nouveaux services.
- **Critère 2 : La qualité, la richesse et la souplesse des commandes vocales.** C'est-à-dire que l'utilisateur puisse exprimer de nombreuses commandes de manière la plus libre possible et surtout que le système n'interprète pas une commande pour une autre.
- **Critère 3 : La qualité et la rapidité de l'identification de visages.** Il est évident que si l'identification du visage prend plus de quelques secondes, l'utilisateur va trouver cela très long devant son miroir. De plus, il faudra bien garantir qu'une personne ne soit pas prise pour une autre et que l'identification d'une personne soit robuste dans le temps.
- **Critère 4 : la richesse, l'intérêt et l'originalité des services supplémentaires.** On ne prendra ici en compte que les services complets, c'est-à-dire que l'on puisse ajouter et supprimer dynamiquement, qu'on puisse configurer et qui rendent leur service. Ce dernier critère ne sera pris en compte que si les 3 critères précédents ont un niveau de satisfaction suffisant.

## Contraintes

### Les contraintes matérielles

- L'application doit fonctionner sur un ordinateur portable standard ou un mini PC.
- La caméra sera soit la caméra intégrée à l'ordinateur portable soit une webcam USB classique.
- L'application sera développée sur un écran LCD. On fera en sorte qu'elle s'adapte à la résolution de celui-ci.

### Les contraintes logicielles

- Le développement sur Windows 8 (fonction système) impose l'environnement .NET. Le choix de C# est largement préférable. Cependant si les étudiants sont capables de justifier le choix d'un autre langage, la migration vers ce langage n'est pas à exclure.
- D'autres bibliothèques pourront être intégrées au projet à deux conditions. La première est qu'elles soient bien identifiées et nécessaires au projet ; et la seconde qu'elles aient des licences d'exploitations compatibles entre elles et avec les autres outils utilisés dans le projet.

### Les contraintes externes

- Toutes les ressources (codes, images, sons, vidéos, ...) utilisées dans la solution finale devront être clairement identifiées (faire un document récapitulatif). On ne peut pas prendre comme on le souhaite une photo ou une icône directement sur internet. On prendra de préférence des ressources libres de droit (mais on gardera quand même la trace de leur origine) ou n'ayant pas d'incidence sur une exploitation de la solution. Attention, de nombreuses ressources demandent à être référencées lors de leur utilisation. Il faudra alors que la solution logicielle mise en œuvre respecte ces conditions.
- Le projet étant fait dans le cadre de la formation de DUT, l'universitaire Nice Sophia Antipolis reste propriétaire du dit projet. Une valorisation de ce projet pourra cependant avoir lieu. Elle

devra se faire en accord avec l'UNS et devra impliquer l'intégralité des membres du projet (Responsable, encadrant ainsi que les étudiants ayant participé au projet).

## Bibliographie & Références

### Vidéo d'illustration

- [Techdays 2014 : Geek in a House](#)

### Documentation et aide en C#

- [Les cours C#](#)
- [Didacticiels C#](#)
- [C# Reference](#)

### Documentation et aide pour l'extensibilité et les plugins

- [Managed Extensibility Framework \(MEF\)](#)
- [Compléments et extensibilité](#)
- [managed addin framework \(system.addin\) with wpf](#)

### Documentation et aide pour le développement Windows 8

- [Tout savoir sur les tuiles de Windows 8 \(C#/XAML\)](#)
- [Blog des développeurs d'applications Windows 8](#)