

LogBook

Auteur	Date de début	Projet	Description	Communication
Joey Martig	19.04.2021	Travail de diplôme Covid propagation	Applications permettant de voir l'évolution d'un virus dans un environnement peuplé d'individus.	Joey.mrtg@eduge.ch Joey.mrtg@gmail.com

19.04.2021 - 08h05/17h00

- Rendu appréciation du travail de stage.
- Démarrage du travail de diplôme et présentation moodle.
- Documentation
 - Intégration CDC
 - Introduction
 - Résumé
 - Planning prévisionnel
- Comparaison WPF et winforms
 - [Référence 1](#)
 - Peu de différence autre que la structure
 - WPF semble plus récent et puissant
 - [Référence 2](#)
 - Interface GUI plus performante (très intéressant)
 - Plus récent (Plus de librairies mise à jour)
 - [Référence 3](#)
 - WPF est plus rapide
 - Aussi plus complexe
 - [Référence 4](#)
 - Les deux se valent
 - WPF est meilleur pour les UI
 - WPF n'est pas disponible pour linux et mac
 - Le data binding est meilleur en WPF ainsi que le design
 - Conclusion
 - L'apprentissage de la structure WPF va prendre du temps mais semble en valoir la peine.
 - L'UI sera plus belle et simple à réaliser en WPF
 - Le plus gros avantage semble être BEAUCOUP plus efficace pour l'affichage de l'interface graphique me permettant d'utiliser le GPU à pleine puissance contrairement au winform qui ne l'utilise pas. S'achant que les pc sont équipé de gtx 1060, le changement en wpf semble être le meilleur choix.
 - WPF semble donc être le meilleur choix.
 - Une autre alternative à l'utilisation de l'interface graphique de WPF est unity qui peut être intégré et communiquer avec le projet. [Intégration d'unity en WPF](#). [Communication](#). [Génération dynamique](#)

- Unity semble être une bonne idée cependant, dans mon cas, en prenant en compte le nombre d'appels, utiliser unity et faire communiquer les deux projets me semblent trop compliqué. Peut-être que rester sur WPF est plus sûr.
- Documentation
 - Analyse interface graphique
 - Comparaison WPF - WinForms - Unity
 - Choix de technologie
- Tentative de communication réussie à l'aide de ce [tutoriel](#).
 - inutilisable pour ce projet car la communication est trop restreinte.
 - La communication se fait uniquement avec des string ou des images mais pas d'objets c#.
- Tentative d'intégration d'unity dans un projet WPF
 - Réussi
 - Problème de resize
 - Au lancement prend toute la fenêtre
 - Le resize et positionnement fonctionnent
 - À la fermeture du programme, unity ne s'arrêtait pas et utilisait 15% du processeur à chaque ouverture.

20.04.2021 - 08h05/17h00

- Retrospective 19.04.2021
- Recherche des méthodes dans la dll user32.dll
- Recherche de différents moyens de communications entre WPF et Unity
 - Essai d'envois de données de WPF à unity avec un pipeline anonyme
 - Essai d'envois de données de WPF à unity avec un pipeline nommé
 - Réussi entre unity et WPF mais inutilisable actuellement
 - Il faut stopper le programme pour lire le message
 - Peut-être un problème lié au fait que les deux programmes essaient de lire au même moment.
- Problèmes d'intelliSense sur unity
 - la connexion entre le script vs 2019 et unity n'est pas établie.
 - Mise à jour de vs 2019 et installation des paquets unity
 - Redémarrage du pc
 - Modifications de paramètre Unity et vs sans succès
 - Recherche de solutions concernant vs 2019 et non 2015
 - Pas suffisant. Ne fonctionne toujours pas
 - [Le point e. à régler le problème](#)
- Problème avec unity qui supprime mes objets à chaque fermeture du projet
 - La scène avait été supprimée et ne s'était pas rajoutée automatiquement
- Communication fonctionnelle mais le projet unity freeze après chaque message.

- Tentative d'utilisation de l'asynchrone pour palier au problème
 - Ce problème est réglé cependant un problème de décalage fait que WPF parle chinois.



- Essai de lecture de byte pour obtenir le résultat. sans succès
- Refactoring du code pour fonctionner en async.
- Le code est toujours imparfait et décalé
 - "Test communication" => "Test communication"
 - "Test communication" => "est communication"
 - "Test communication" => Signes chinois
- Communication en string fonctionnelle à 100% en utilisant une méthode async réursive appelée au démarrage du projet unity.

- Méthode appelée chaque seconde ayant des résultats très aléatoire.

```
void Start()
{
    // Code
    InvokeRepeating("ReadPipeData", 0, 1f);
}

// Références
private async void ReadPipeData()
{
    string result = await ss.ReadStringAsync();
    ChangingText.GetComponent<Text>().text = result;
}
```

- Méthode appelée une fois au démarrage puis à chaque fois qu'elle termine l'affichage de données reçues.

```
void Start()
{
    //Code
    ReadPipeData();
}

private async void ReadPipeData()
{
    string result = await ss.ReadStringAsync();
    ChangingText.GetComponent<Text>().text = result;
    ReadPipeData();
}
```

- Idée pour l'implémentation en Unity : Envoyer des objets en JSON via le pipeline.
- Test d'envoi d'un objet de WPF à Unity en JSON

```

public class WeatherForecastWithPOCOs
{
    0 références
    public DateTimeOffset Date { get; set; }
    0 références
    public int TemperatureCelsius { get; set; }
    0 références
    public string Summary { get; set; }
    public string SummaryField;
    0 références
    public IList<DateTimeOffset> DatesAvailable { get; set; }
    0 références
    public Dictionary<string, HighLowTemps> TemperatureRanges { get; set; }
    0 références
    public string[] SummaryWords { get; set; }
}

1 référence
public WeatherForecastWithPOCOs()
{
    SummaryWords = new string[] { "1", "3", "3" };
    TemperatureRanges = new Dictionary<string, HighLowTemps>();
    TemperatureRanges.Add("range1", new HighLowTemps());
}

2 références
public class HighLowTemps
{
    0 références
    public int High { get; set; }
    0 références
    public int Low { get; set; }
}

```

```

[16:48:48] {"Date":"0001-01-01T00:00:00+00:00","TemperatureCelsius":0,"Summary":null,"
DatesAvailable":null,"TemperatureRanges":{"range1":{"High":0,"Low":0}},"SummaryWords":["1","3","3"]}

```

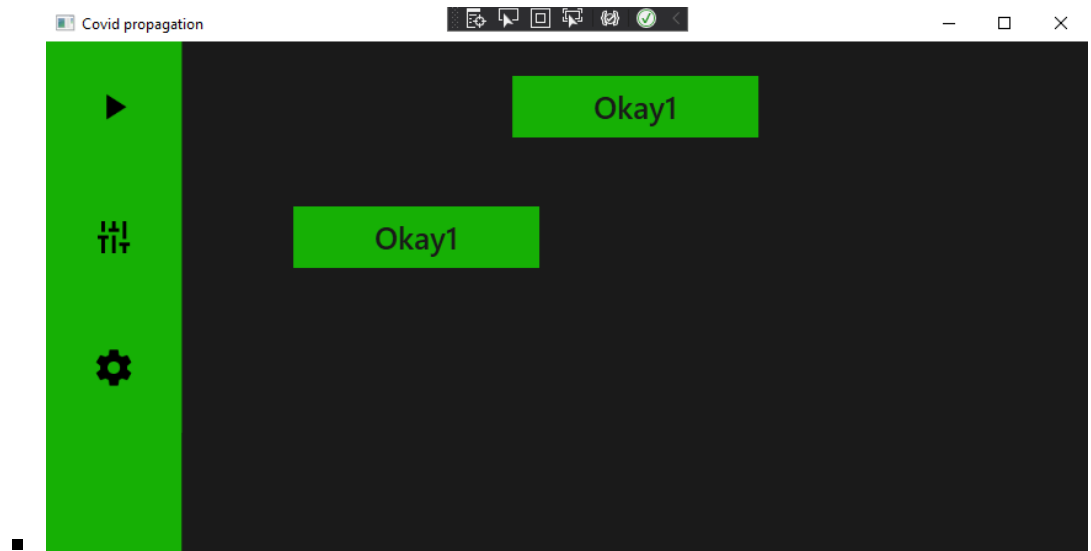
21.04.2021 - 08h05/17h00

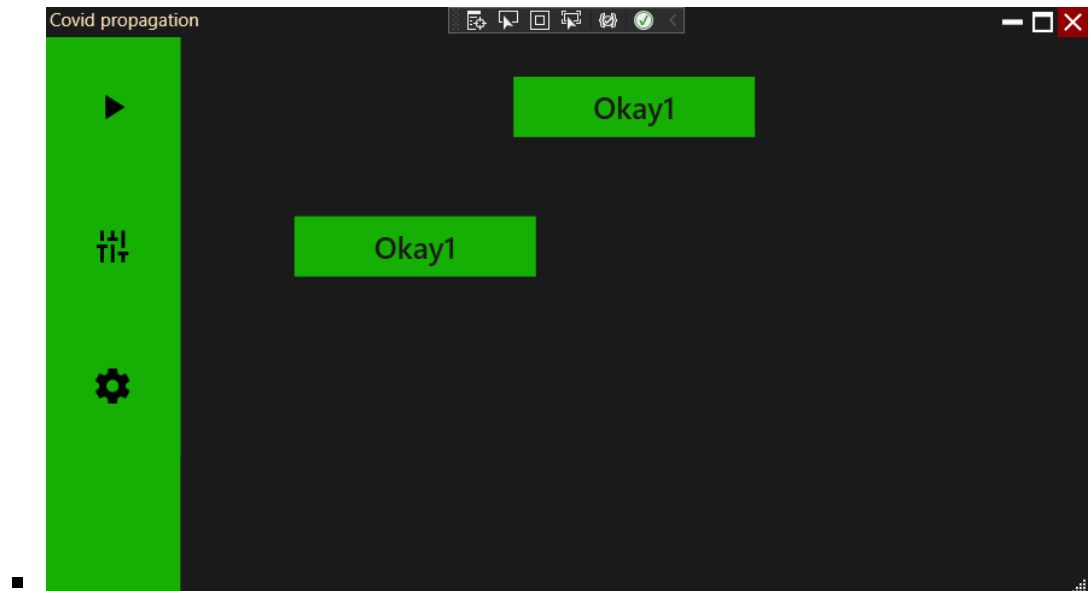
- Documentation
 - Complétion de la comparaison de technologie d'interface graphique des pipelines.
 - Problèmes rencontrés - Pipelines
 - Intégration d'unity à WPF
- Modification de l'envoi de données pour utiliser un BinaryWriter à la place d'une Stream
 - Augmente la limite de données transmissible d'un Uin16 à int32.
- Modification de la réception des données pour utiliser un BinaryReader à la place d'un stream.
- Test d'intégration + communication réussi
- Début de la création de l'UI
 - Recherche d'un thème pour l'application
 - [Thème prometteur](#)
 - Test du thème
 - Le thème fonctionne et possède énormément de possibilités
 - Test de différents composants du thème
 - Boutons
 - TabControl
 - Changement d'icônes

- Changements de la couleur
- Tentative de changer le background pour un background non pré-défini.
- Intégration d'une autre page xaml depuis le TabControl
- Création d'une page, pour la simulation, les paramètres graphiques et les paramètres de la simulation
- Abandon du thpme trouvé du à son manque de responsivité malgré sa grande variété de contenu.

22.04.2021 - 08h05/17h00

- UI
 - Ajouts d'icône de boutons
 - Création de l'interface principal
 - Menu
 - Sections
 - Responsivité
 - Utilisation des columns et rows
 - Impossible de changer la couleur de la barre de titre du projet
 - Désaffichage de la barre
 - Création de ma propre barre
 - Différence entre les deux versions

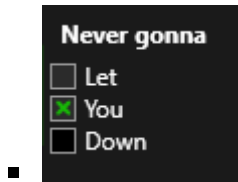




- Modification des méthodes de la barre de titre pour être utilisable sur toutes les pages
- Modification de la structure des boutons pour intégrer directement l'image dans leur balise et non appeler une image. La version précédente empêchait plusieurs boutons d'avoir la même image. Résultat en l'affichage de premier bouton sans image.
- Création de différentes pages de contenus
- Navigation entre les pages
- Modification de l'affichage d'un slider
 - Difficulté à trouver les bon paramètres
 - Binding de la couleur à la valeur du slider
 - Création d'un template pour le slider en récupérant la version de base de [WPF](#)
 - Suppression de la track bar
 - Modification du RepeatButton
 - Création de deux version se situant à gauche et droit du thumb et ayant des couleurs différentes
 - Comparaison du slider de windows et du slider personnalisé à l'aide d'un template



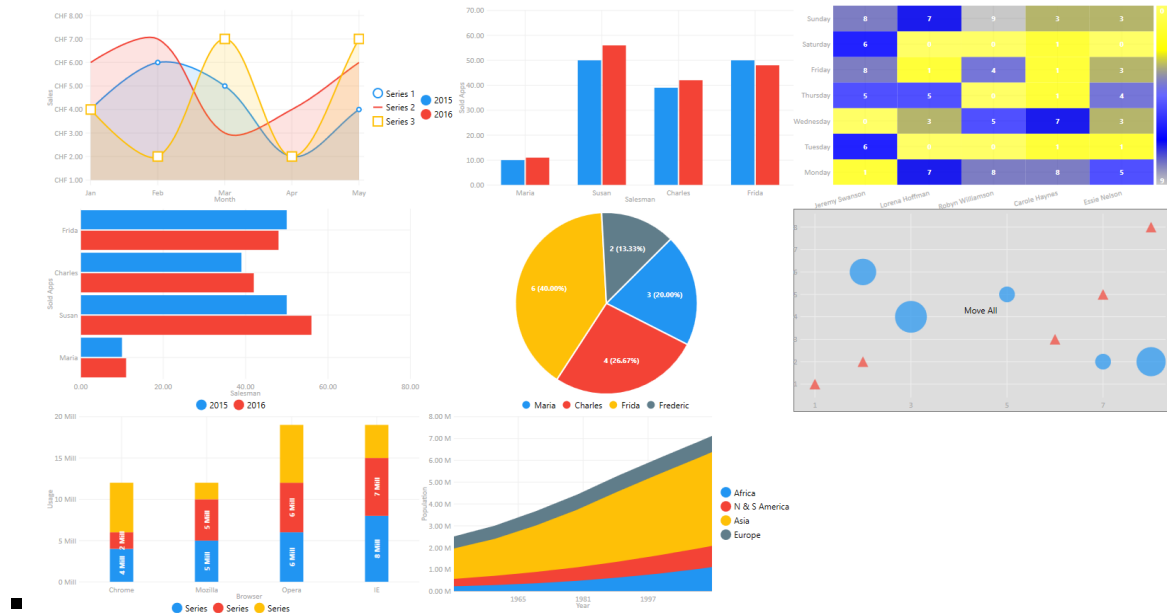
- Modification de l'affichage des radioBoutons en récupérant la version de base de [WPF](#)
 - Nouveau design
- Modification des couleurs pour qu'elle soient contenuent dans des balisent facilement modifiable.
- Modification de l'affichage des checkbox en récupérant la version de base de [WPF](#)
 - Nouveau design



- Modification de la fermeture des fenêtres pour cacher les fenêtres secondaires et fermer le programme en cas de fermeture de la fenêtre principale.

23.04.2021 - 08h05/16h10

- Documentation
 - Organisation
 - Maquettes
 - Introduction
 - Technologies utilisées
 - Schéma de fonctionnement
 - Description des technologies utilisées
- Réflexion sur le fonctionnement des threads pour l'application.
 - Parallel
- Installation LiveCharts
 - Installation packet LiveCharts
 - Installation packet LiveCharts.Wpf
 - Installation packet LiveCharts.Geared
- Graphiques
 - Bug à chaque changement de taille, régénérer la solution est nécessaire.
 - Uniquement dans le concepteur et non au lancement du programme
 - Nettoyer la solution permet de tout de même voir l'affichage de la fenêtre
 - Intégration de graphiques
 - Courbe
 - HeatMap
 - Colonne
 - Ligne
 - Bule
 - Circulaire (Camembert)
 - Area Stack

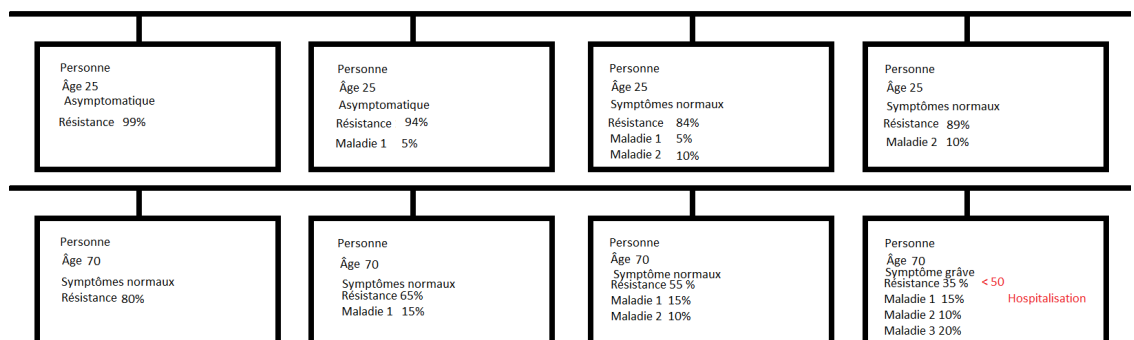


26.04.2021 08h05 / 16h10

- Début de la création de la simulation
- Récupération du code du stage et suppression des éléments graphiques
- Commentaires
- Création d'une classe "Site" regroupant tous les lieux et véhicules
- Relecture du cahier des charges
- Relecture du fichier excel contenant la transmission par aérosol [2021_COVID-19_Aerosol_Transmissionn_Estimator](#)
- Création du code des lieux en fonction du fichier excel
 - Taille du bâtiment, longueur - largeur - hauteur - air - volume
 - Paramètres de l'air et de déposition
 - Utilisation du type double pour le calcul du taux d'infection
 - Suppression de certains paramètres non-utilisés
- Réflexion sur la structure du code et l'interaction entre les individus et les lieux
 - Résistance au virus
 - Maladies
 - Age

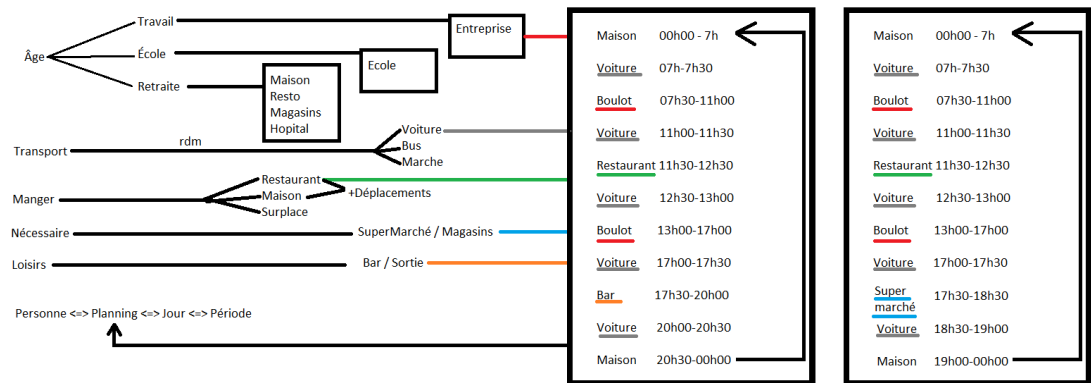
(Valeurs non représentative de la réalité)

Temps -->



- Création des planning
 - Pattern

- (Gérer les cercles sociaux)
- Création en fonction de l'âge



○ Mesures & Symptômes

- résistance
- comportement
- Augmentent certains paramètres

(Valeurs non représentative de la réalité)




- Les activités ne contiennent que des types d'objets permettant de définir ce que l'individu fait. Le planning contient les lieux et les indique à l'individu.
- Création des plannings en fonction de la figure concernant leur création.
- Nombre d'école et autres bâtiments en fonction du nombre de personnes. Genève servant d'exemple.
 - ~0.03% --> 165 écoles pour 500'000 personnes
 - ~6,74% --> 33'700 entreprises pour 231'000 emplois
 - ~0.01% --> 8 hôpitaux, 2 cliniques, 30 lieux de soins
 - ~0,03% --> ~136 supermarchés
 - ~50% --> ~250'000 appartements
- pareil mais pour l'âge de la population
 - ~22% 0-19 ans
 - ~63% 20-64 ans
 - ~10% 65-79 ans
 - ~5% 80+ ans
 - [Source](#)
- Pareil pour les transports
 - ~37% --> à pied
 - ~36% --> voiture (Chauffeur ou passagers)
 - ~15% --> tpg
 - ~10% --> vélos

- [Source](#)
- Réflexion sur la création des transports et batiments et sur la source de certaines données.
 - On a une population de 100 personnes
 - 22% 0-19 ans
 - 63% 20-64 ans
 - 15% 65+ ans
 - Il faut donc au minimum
 - Une école
 - Un supermarché
 - Une maison par personne (Pour le moment les < 19 ans vivent seuls)
 - On crée les batiments en fonction des besoins cité
 - Pour les véhicules
 - On crée 36 voitures
 - et des bus (a voir plus tard pour la quantité)
 - Ensuite on crée les 100 individus
 - En fonction du pourcentage d'âge
 - On leur attribut
 - Une maison vide
 - Une école
 - Un moyen de transport (vide si voiture)
 - On crée leur résistance

27.04.2021 08h05 / 16h30

- Les véhicules dans la simulation agissent exactement comme les batiments.
 - Pas de déplacements
 - La voiture étant la seule d'"unique" à une personne.
 - La différence réside dans l'assignation et non dans le fonctionnement.
- Suppression du parent "Vehicle" et "Building"
- Création d'un objet parent "Site" représentant tous les lieux dont les véhicules et l'extérieur.
- Création de variables constantes global pour la création des sites
- Réflexion sur le fonctionnement des individus
 - Marche à suivre.
 - Utiliser du async pour récupérer le taux de contamination ? (Attendre que tout le monde soit bien entré)
- Création des individus
 - Maladies
 - Résistance au virus
 - Durée de l'immunité
 - Durée de l'infection par le virus
- Création des maladies
 - Puissance
 - En fonction de l'âge
 - Durée
- Recherche de source pour gérer mathématiquement les probabilité qu'une personne attrape des maladies en fonction de son âge.

- Meilleure compréhension du sujet mais rien de concluant.
- Modification des namespaces qui avaient automatiquement créé sous-namespaces en créant des dossiers dans le programme WPF
- Modification du site pour prendre en compte les personnes qui se trouvent dedans.
 - Cette modification impacte aussi à quel moment le calcul des probabilités est effectué.
 - Si une personne entre ou sort. Les probabilités sont recalculées.
 - Sinon on récupère la dernière valeur sans recalculer
 - Doit recalculer si une personne a été infectée. (en prenant en compte le temps d'incubation du virus, ce n'est pas probablement pas nécessaire.)
- Données récupérées concernant le temps d'incubation et d'infection du covid
 - [Durée du covid](#)
 - [Incubation et durée](#)
- Suppression des cercles familiaux et d'amis du planning
 - Raison :
 - Pas par manque de temps mais par logique du fonctionnement de la simulation
 - Les plannings gèrent entièrement la simulation, les lieux, les horaires, les contacts.
 - Au final, la famille sont les personnes habitants au même endroit. Les amis, les personnes qui vont aux mêmes bars.
 - La création du planning s'occupera donc de gérer les contacts proches
 - Cette modification pourrait être revue plus tard en fonction de l'évolution du fonctionnement de la simulation et des interactions entre les objets.
- Commentaires et création d'en-tête dans le code
- Recherche de conversion de probabilité d'années en mois
 - Rien trouvé
 - Uniquement dans l'autre sens
- Est-il justifié d'avoir des symptômes de virus ?
 - La résistance au virus étant simplifiée et l'accent étant majoritairement mis sur la propagation, est-il nécessaire d'intégrer des symptômes qui dans 90% des cas ne changent en rien le taux de contamination (excepté la toux donc).
 - Simplifier en utilisant simplement le taux de transmission ?
 - Pas viable car asymptomatiques
 - Les symptômes pourraient être utilisés pour détecter le virus dans la simulation pour les quarantaines et autres.
 - Garder les symptômes mais avec pour la majorité, uniquement un indice de détection
- Création de la classe Virus, Transmission et Symptom
- Création du fichier XML comportant les données du Covid
- Lecture du fichier depuis le code et récupération des données.
- Fonctionnement imaginé du virus et de ses composants
 -  Virus transmissions and symptoms
 - Informations importantes concernant la respiration et affectant donc la toux [Source](#)
 - Resting – Oral breathing = 2.0 quanta/h
 - Resting – Speaking = 9.4 quanta/h
 - Resting – Loudly speaking = 60.5 quanta/h
 - Standing – Oral breathing = 2.3 quanta/h
 - Standing – Speaking = 11.4 quanta/h
 - Standing – Loudly speaking = 65.1 quanta/h

- Light exercise – Oral breathing = 5.6 quanta/h
- Light exercise – Speaking = 26.3 quanta/h
- Light exercise – Loudly speaking = 170 quanta/h
- Heavy exercise – Oral breathing = 13.5 quanta/h
- Heavy exercise – Speaking = 63.1 quanta/h
- Heavy exercise – Loudly speaking = 408 quanta/h
- Modification de la transmission par aérosol sur les sites pour prendre en compte les paramètres individuels de chaque individu présent sur les lieux.