

**PLAN DE TESTS**

**Projet :**

**Modelling and simulating**

**human behavioural factors**

**influencing the spread of Covid-19**

Rémi HUYGHE

Moustafa MOHAMED

Maminiaina RAZANAMARIMANDIMBY

**UNIVERSITÉ GRENOBLE ALPES**

**PLAN DE TESTS**

**Modelling and simulating human behavioural factors**

**influencing the spread of Covid-19**

| Référence du document | D7 |
| --- | --- |
| Version du document : | 1 |
| Date du document : | 26 Mai 2021 |
| Auteurs : | Rémi HUYGHE  Moustafa MOHAMED  Maminiaina RAZANAMARIMANDIMBY |

| Validé par : |  |
| --- | --- |
| Validé le : |  |
| Soumis le : |  |
| Type de diffusion : | Document électronique partagé  (Google Drive) |
| Confidentialité : | Chef de projet  Membres de l’équipe |

| **Mots clés : Plan de test** |
| --- |

**Sommaire**

[**I. Préambule**](#_yeiri3o6ilr9) **3**

[I.1. Objectifs et méthodes](#_pgcqc0sxz9zt) 3

[I.2. Documents de référence](#_pmqhvspk4g41) 4

[**II. Guide de lecture**](#_99geq51b0efd) **4**

[**III. Concepts de base**](#_pf4twoduxdf) **4**

[III.1. Les agents](#_axbpmw34l705) 4

[III.2. Les fonctions](#_cal4l7xgamno) 4

[**IV. Tests fonctionnels**](#_6f3hqi7b2pjt) **5**

[IV.1. Scénario : Réplique des résultats du questionnaire](#_x5hyt5rexmx7) 5

[IV.1.1. Description](#_gp3dqamw0crm) 5

[IV.1.2. Contraintes](#_ugwnuexrjg2b) 5

[IV.1.3. Dépendances](#_uyxb2v86u35k) 5

[IV.1.5. Procédures de test](#_x9i6566nw4rs) 5

[**V. Tests unitaires**](#_y6l7w8e3wtql) **5**

[V.1. Procédure 1 : Initialisation de l’interface](#_263g4e6o16de) 6

[V.1.1. Identification](#_37a5jbgrqzdf) 6

[V.1.2. Description](#_wlzacrhjvrj5) 6

[V.1.3. Contraintes](#_3vauqpc4izgg) 6

[V.1.4. Dépendances](#_s9a8tkw17u0o) 6

[V.2. Procédure 2 : Initialisation des variables](#_me1wim3yokrd) 6

[V.1.1. Identification](#_4n06agogk3v) 6

[V.1.2. Description](#_59a88yh9s58u) 6

[V.1.3. Dépendances](#_p6gagjqvp50n) 6

[**VI. Annexes**](#_44grhga3asef) **7**

[**VII. Références**](#_4ej7up8lwap7) **7**

# 

# I. Préambule

Le Plan de tests permet de s’assurer que chaque élément du modèle répond aux besoins, est fonctionnel et est robuste dans ses interactions.

## I.1. Objectifs et méthodes

L’objectif est de pouvoir vérifier si l’implémentation est correctement réalisée, et comment est-ce que cela a été fait.

## I.2. Documents de référence

Les demandes explicites de fonctionnalités que doit répondre notre modèle peuvent être retrouvées au niveau du Cahier des Charges. Les documents relatifs aux questionnaires, justifiant des choix pris dans ce modèle sont disponibles en Annexe, à savoir les Résultats du questionnaire[[1]](#footnote-0), les Analyses qui ont permis l’élaboration des hypothèses et des scénarios[[2]](#footnote-1), ainsi que le Rapport[[3]](#footnote-2) qui synthétise les analyses menant aux scénarios du modèle.

# II. Guide de lecture

Avant toute lecture, et pour une meilleure compréhension des termes spécifiques au logiciel Netlogo, il est fortement recommandé de lire la partie **III. Concepts de base**. Elle reprend les principaux termes utilisés au sein du logiciel qui permettent une meilleure compréhension de ce qui suit.

Pour la suite, si vous vous intéressez aux réponses du modèle aux demandes du cahier des charges, merci de vous référer à la partie **IV. Tests fonctionnels**.

Pour une lecture globale des interactions entre les différentes fonctions du modèle, merci de vous référer à la partie **VI. Tests d’intégration**.

Pour une lecture plus approfondie des différentes fonctions du modèle, la partie consacrée aux **V. Tests unitaires** permet de voir en détail ce que fait chaque fonction du modèle.

# III. Concepts de base

Pour une meilleure compréhension, nous reprenons ici quelques termes génériques du logiciel Netlogo.

## III.1. Les agents

Pour rappel, Netlogo est un simulateur de comportements que peuvent avoir des agents. Ici, la notion d’agents renvoie à “des *êtres* capables d’exécuter des instructions. Chaque agent peut avoir sa propre activité, mais tous agissent simultanément.” (Romain Jean, *Guide de la programmation,* <https://romainmejean.fr/manuel_netlogo/> - Juin 2009)

Netlogo connaît 4 types d’agents, mais nous nous sommes intéressés aux turtles, ayant comme spécificité la capacité de se déplacer dans un espace défini, le world.

## III.2. Les fonctions

Dans le logiciel Netlogo, les fonctions du modèle sont appelées des **commandes** et de **reporters**. Ces deux types de fonctions disent aux agents - les turtles - ce qu’ils doivent faire (les *commandes*) et communiquent des valeurs calculées (les *reporters*).

Comme dans tout langage de programmation, Netlogo dispose de commandes et de reporters prédéfinies - ce sont les **primitives[[4]](#footnote-3)**.

# IV. Tests fonctionnels

Dans le cahier des charges, on nous a soumis le projet dans le but de modéliser trois différentes périodes de la pandémie à travers le modèle, notamment l’avant COVID-19, le premier confinement et l’automne 2020.

Au cours du projet, il a été jugé peu pertinent de faire cet enchaînement dans notre analyse, du fait que les réponses risquent d’être très vagues, donc peu adaptées aux besoins de notre modèle. En effet, notre démarche initiale était de reproduire, à partir de données empiriques d’un questionnaire, les comportements des étudiants de l’Université Grenoble Alpes. C’est en ce sens qu'avec l’accord de notre client, nous avons changé de ligne de conduite et nous avons établi de réaliser plusieurs scénarios. Nous avons réalisé un scénario portant sur la diffusion du virus dans une population basé sur notre questionnaire.

## IV.1. Scénario : Réplique des résultats du questionnaire

### IV.1.1. Description

Ici, nous reprenons les résultats empiriques, tels qu’ils ont été fournis par nos répondants dans le cadre de notre questionnaire. En ce sens, nous implémentons toutes les données descriptives de la population des 153 répondants.

### IV.1.2. Contraintes

Les contraintes sont citées dans la *Documentation interne*.

### IV.1.3. Dépendances

Il n’y a pas de tests préalable à réaliser, le modèle est fonctionnel seul.

### IV.1.5. Procédures de test

Avec les paramètres de base, il est attendu qu’en une trentaine de jours toute la population soit infectée.

# V. Tests unitaires

Les tests unitaires concernent la phase de test de chaque procédure de notre modèle. Ces procédures nous serviront, par la suite, à faire tourner les différents scénarios.

## V.1. Procédure 1 : Initialisation de l’interface

### V.1.1. Identification

L’initialisation du modèle se fait avec la commande setup

### V.1.2. Description

La commande setup permet d’initialiser et/ou réinitialiser l’interface aux valeurs initiales de départ, pour remettre les compteurs à zéro. Pour ce faire, cette commande est associée au bouton setup de l’interface.

### V.1.3. Contraintes

Il n’y a pas de contrainte particulière à cette procédure.

### V.1.4. Dépendances

Le fonctionnement de cette procédure dépend du fonctionnement des procédures setup-globals, setup-population et set-explanation - procédures définies plus bas.

Les procédures clear-all et reset-ticks sont des procédures déjà prédéfinies dans le logiciel Netlogo.

## V.2. Procédure 2 : Initialisation des variables

### V.1.1. Identification

L’initialisation des variables se fait avec le reporter setup-globals

### V.1.2. Description

Avec ce reporter, nous définissons les variables initiales de notre modèle. Elle servira de base à tous les scénarios, hormis quelques détails qui changent, le cas échéant.

La fonction setup-globals est lancée lorsque la fonction setup est activée.

Ici, on définit les variables globales de notre modèle, qui reprennent les données descriptives des répondants de notre questionnaire :

* population-size 153 la taille de la population dans le modèle est fixée aux 153 répondants du questionnaire.

### V.1.3. Dépendances

Cette fonction n’utilise pas d’autres procédures définies au sein de notre modèle. Pour définir les variables, elle n’utilise que la procédure prédéfinie de Netlogo set.

# VI. Annexes

Cahier des Charges

Manuel d’utilisation

Annexe - Résultats questionnaire

Annexe - Questionnaire

AnalyseFinale

# VII. Références

Wilensky, U. (1999). NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Romain Jean (2009) Dictionnaire NetLogo, <https://romainmejean.fr/manuel_netlogo/primindex.html>

Romain Jean (Juin 2009) *Guide de la programmation,* <https://romainmejean.fr/manuel_netlogo/>

1. Annexe - Résultats questionnaire.xlsx [↑](#footnote-ref-0)
2. AnalyseFinale.R [↑](#footnote-ref-1)
3. Rapport.pdf [↑](#footnote-ref-2)
4. L’ensemble des primitives peuvent être retrouvées dans le **Dictionnaire NetLogo**, <https://romainmejean.fr/manuel_netlogo/primindex.html>. La liste des principales primitives utilisées dans notre modèle peut être retrouvée en partie **IX. Glossaire** [↑](#footnote-ref-3)