**Fonctionnalités du prototype**

**Développement collaboratif**

**Acteurs :**

Nicolas Buffon, Marc Plano-Lesay, Jean-Christophe Ricard, Adrien Vetillart.

**Versions :**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Auteur | Date | Description |
| 1.0 | VETILLART Adrien | 13/01/2012 | Définir les fonctionnalités à développer pour le prototype (période 2) |

**Introduction**

Ce document vise à définir les fonctionnalités que nous allons développer pour produire le prototype de notre logiciel. Cette version minimale s’appuiera sur certains cas précis, non implémentés dans leur totalité ; l’objectif étant de pouvoir réaliser les actions essentielles à un premier lancement de l’outil.

**SOMMAIRE**

I. Fonctionnalités principales du prototype et déroulement du développement 5

A. Réaliser la simulation 5

B. Simuler l’affichage de la map 5

C. Lancer la simulation 5

D. Arrêter la simulation 5

E. Lancer la 2ème simulation 5

F. Arrêter la simulation 6

G. Lancer la 3ème simulation 6

H. Arrêter la simulation 6

II. Bilan sur l’état final du prototype 6

# Fonctionnalités principales du prototype et déroulement du développement

Les fonctionnalités qui suivent sont classées dans l’ordre d’implémentation au sein du prototype afin que les développeurs puissent s’appuyer sur ce document pour réaliser leur code.

## Réaliser la simulation

Cette fonctionnalité est la plus importante de l’application. Elle est implicite et repose sur le bon fonctionnement des fonctionnalités que nous allons développer et intégrer. Nous devons nous assurer que l’application tourne correctement à chaque ajout d’un nouveau composant, comportement etc.

## Simuler l’affichage de la map

Pour simuler l’affichage de la map, nous allons utiliser des données écrites en dur dans nos classes java. Il y aura un seul type de terrain et une seule altitude pour le prototype. On affichera cette map dans une fenêtre.

## Lancer la simulation

Cette fonctionnalité se résume à l’action suivante : on appuie sur un bouton de l’IHM (le bouton Start) et un timer se lance.

## Arrêter la simulation

On appuie sur le bouton stop de l’IHM : le timer s’arrête.

***Pour la suite, on ajoute deux robots et un incendie à notre map. Il est important d’avoir un robot plus loin de l’incendie que son homologue. Ces données sont également chargées en dur avec la map. Il y a la présence du manager.***

## Lancer la 2ème simulation

On relance la simulation avec les nouvelles données chargées. Comme le paragraphe C, le timer se lance. De plus, le robot le plus près de l’incendie doit partir en ligne droite pour aller éteindre le feu sur ordre du manager (algorithme de pathfinding simple : déplacement linéaire). Quand il arrive au feu, l’incendie s’arrête immédiatement.

## Arrêter la simulation

On stoppe la simulation en appuyant sur le bouton Stop : le timer s’arrête et les robots ne bougent plus.

***Pour la dernière partie de la simulation, on va simuler un deuxième incendie et ajouter une difficulté pour éteindre le premier. Ainsi, on pourra envoyer le deuxième robot sur l’incendie qui se déclenchera pendant la simulation.***

## Lancer la 3ème simulation

On lance la simulation avec le bouton Start, le robot le plus près de l’incendie 1 part l’éteindre. Il met plus de temps que pour la simulation 2 pour effectuer son travail car il y a une difficulté associée au feu. Pendant ce temps, un deuxième incident surgit (à noter que l’apparition est programmée pour se déclencher après un laps de temps T fixé, après lancement de la simulation). Le deuxième robot (inoccupé) va partir éteindre ce feu sur ordre du manager.

## Arrêter la simulation

On arrête la simulation : on stoppe le timer et les robots ne bougent plus.

# Bilan sur l’état final du prototype

Le prototype final comportera une fenêtre principale composée d’une map chargée en dur, sur laquelle sera visible deux robots et un feu (avec un robot plus près de l’incendie que son homologue). De plus il y aura deux boutons : Stop et Start. Quand l’utilisateur va cliquer sur le bouton Start, un timer va se lancer et le robot le plus près de l’incendie va partir l’éteindre sur ordre du manager (utilisation d’un algorithme de pathfinding simple : en ligne droite). Pendant ce temps, un autre incendie va apparaître sur la map et le robot 2 inoccupé va partir l’éteindre, toujours sur ordre du manager. Pour finir, nous pourrons stopper la simulation : les robots restent à leur place.