Dario, Jean-Daniel, Romain, Ronaldo

CFPT-I | M306 | 23.02.2018

MBE

Multiple Browsers Emulator

**Suivi des modifications**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Modification** | **Auteur** |
| 23.02.18 | 1.0 | Création | Ronaldo & Jean-Daniel |
| 02.03.18 | 1.1 | Analyse fonctionnelle | Ronaldo |
| 09.03.2018 | 1.2 | Analyse fonctionnelle | Ronaldo & Jean-Daniel |
| 09.03.18 | 1.2 | Analyse organique | Dario & Romain |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table des matières

[1 Introduction 3](#_Toc508357161)

[2 Rappel du cahier des charges 3](#_Toc508357162)

[2.1 But 3](#_Toc508357163)

[2.2 Spécifications 3](#_Toc508357164)

[2.3 Restrictions 3](#_Toc508357165)

[2.4 Environnement 3](#_Toc508357166)

[2.5 Livrables 3](#_Toc508357167)

[3 Analyse fonctionnelle 4](#_Toc508357168)

[3.1 Fonctionnalités 4](#_Toc508357169)

[3.1.1 Afficher une vue d’ensemble des appareils sous forme de pages web 4](#_Toc508357170)

[3.1.2 Les URLS des pages seront paramétrables 4](#_Toc508357171)

[3.1.3 Modifier le nombre de pages affichées 4](#_Toc508357172)

[3.1.4 Afficher le code source de chacune des pages web (FACULTATIF) 4](#_Toc508357173)

[3.1.5 Concevoir un zoom de plusieurs pages web en une (FACULTATIF) 4](#_Toc508357174)

[3.1.6 Créer des configurations personnalisées (FACULTATIF) 4](#_Toc508357175)

[3.2 Interfaces 5](#_Toc508357176)

[3.2.1 Fenêtre de configuration 5](#_Toc508357177)

[3.2.2 Fenêtre principale 7](#_Toc508357178)

[3.2.3 Fenêtre d’aide 8](#_Toc508357179)

[3.2.4 Fenêtres à propos 8](#_Toc508357180)

[4 Analyse organique 9](#_Toc508357181)

[4.1 Diagramme de classes 9](#_Toc508357182)

[4.1.1 Classe – FrmMain 9](#_Toc508357183)

[4.1.2 Classe – Emulator 9](#_Toc508357184)

[4.1.3 Classe – Config 10](#_Toc508357185)

[5 Conclusion 10](#_Toc508357186)

[6 Annexes 10](#_Toc508357187)

# Introduction

Il s’agit d’une application C# qui aura pour but de simuler la vue d’ensemble des appareils sous forme des pages WEB.

Cette application est développée pour faciliter les différents tests du projet IMDD.

# Rappel du cahier des charges

## But

Émuler les appareils sous forme de pages WEB afin de pouvoir faciliter des tests d’applications.

## Spécifications

L’application sera capable de :

* Afficher une vue d’ensemble des appareils sous forme de pages web
* Les URLS des pages seront paramétrables
* Afficher le code source de chacune des pages web
* Modifier le nombre de pages affichées
* Concevoir un zoom de plusieurs pages web en une
* Créer des configurations personnalisées

## Restrictions

* L’application ne fonctionnera pas sur les téléphones mobiles et sur des ordinateurs ayant un système d’exploitation autre que Microsoft Windows®
* Les zooms ne sont que possibles aux extrémités du rectangle (page émulées)
* Les pages web forme uniquement un rectangle

## Environnement

Matériels nécessaires au développement de l’application :

* Ordinateur de type PC
* Système d’exploitation : Windows 10 Entreprise
* Outil de développement : Visual Studio 2017
* Outil de sauvegarde : Git, Drive, Local

## Livrables

* Documentation technique
* Manuel utilisateurs
* Application

# Analyse fonctionnelle

## Fonctionnalités

### Afficher une vue d’ensemble des appareils sous forme de pages web

Cette fonctionnalité permettra à l’utilisateur d’émuler les appareils sous forme de pages web. Il pourra choisir le nombre d’appareil **(A DEFINIR)**. L’application affichera des pages web sous forme d’une grille.

### Les URLS des pages seront paramétrables

Cette fonctionnalité permettra à l’utilisateur de personnalisé l’URL des pages WEB. Il peut définir les paramètres qu’il souhaite et les variables.

### Modifier le nombre de pages affichées

Cette fonctionnalité permettra à l’utilisateur de modifier le nombre de pages web affichées **(A DEFINIR)**. Il pourra choisir le nombre d’appareils qu’il souhaite par ligne et le nombre d’appareils qu’il souhaite par colonnes.

### Afficher le code source de chacune des pages web (FACULTATIF)

Cette fonctionnalité permettra à l’utilisateur d’afficher le code sources de chacune des pages web (F12 de Chrome) afin de pouvoir les débugger.

### Concevoir un zoom de plusieurs pages web en une (FACULTATIF)

Cette fonctionnalité permettra à l’utilisateur de choisir les côtés qu’il veut réunir en une page (zoom). Il aura le choix d’afficher la ligne de gauche, de droite, du haut et/ou du bas.

### Créer des configurations personnalisées (FACULTATIF)

Cette fonctionnalité permettra à l’utilisateur de créer des configurations et de les enregistrer. Il pourra choisir le nombre de page web qu’il souhaite par ligne et par colonne, les zooms qu’il souhaite et les URLS des pages web. Il pourra enregistrer une configuration en indiquant un nom.

## Interfaces

### Fenêtre de configuration

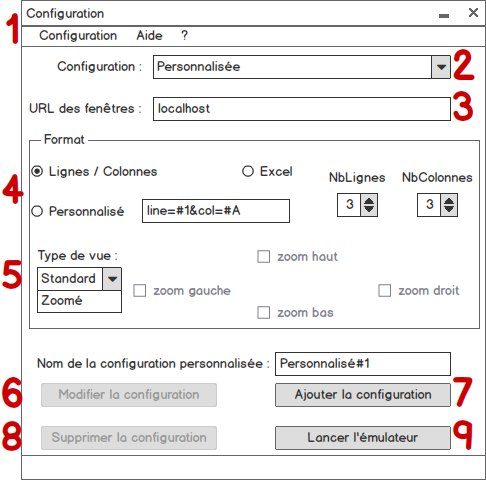


Fig. - Fenêtre de configuration

1 – Menu de navigation, il permet de naviguer entre les fenêtres de configuration, d’aide et à propos.

2 – Menu déroulant, il permet à l’utilisateur de sélectionner une configuration existante ou de sélectionner « Personnalisée » s’il souhaite en créer une nouvelle.

3 – Url des fenêtres, le champ texte permet d’insérer l’URL du serveur dans lequel on veut que les fenêtres se connectent.

4 – Les radios boutons permettent de sélectionner les paramètres que l’on veut dans l’URL (Lignes/Colonnes : line=1&col=1 ; Excel : line=1&col=A ; Personnalisé permet de mettre les paramètres que l’on veut). Ensuite l’utilisateur peut sélectionner le nombre de lignes et de colonnes qu’il souhaite.

5 – L’utilisateur a aussi le choix au mode de vue. La vue standard est une vue avec les par exemple les 9 tablettes affichées en rectangle. La vue zoomée permet à l’utilisateur de choisir les bords qu’il souhaite avoir sur un écran supplémentaire.

6 – Si l’utilisateur sélectionne une configuration qu’il a fait lui-même et qu’il la modifie, il a la possibilité d’enregistrer les modifications avec le bouton « Modifier la configuration ».

7 – Si l’utilisateur sélectionne dans le menu déroulant (n°2), l’option « Personnalisée », avec le bouton « Ajouter la configuration », il a la possibilité en indiquant un nom, d’enregistrer la configuration.

8 – Si l’utilisateur sélectionne une configuration qu’il fait a lui-même et qu’il souhaite la supprimer, il a la possibilité de le faire en cliquant sur le bouton « Supprimer la configuration ».

9 – Quand l’utilisateur a fini de configurer l’émulateur, il peut le lancer en cliquant sur le bouton « Lancer l’émulateur », ce qui ouvrira la fenêtre principale (Fig. 2).

### Fenêtre principale

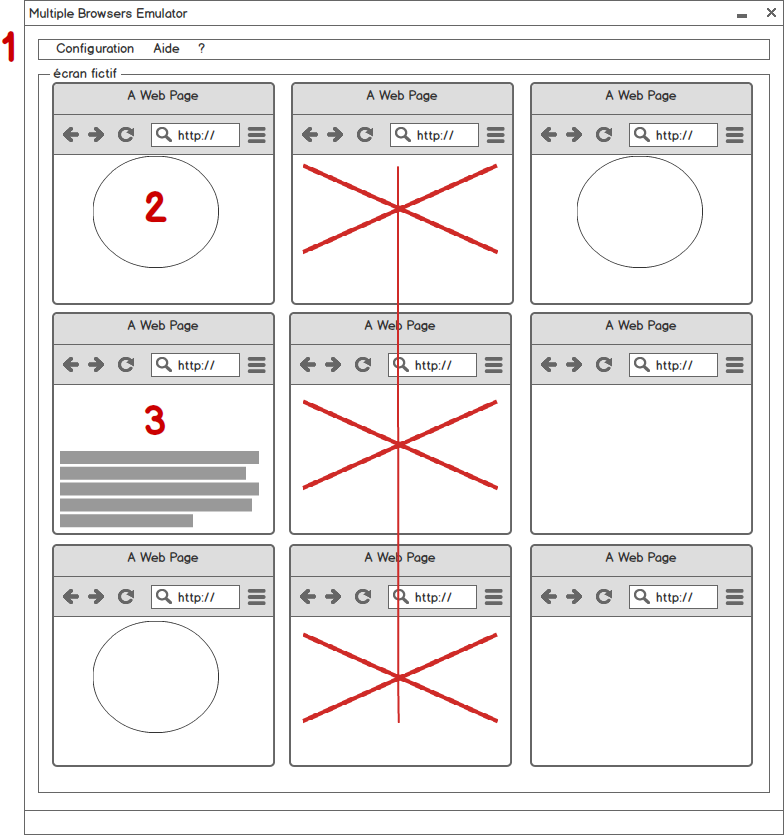


Fig. - Fenêtre d’émulation

1 – Menu de navigation, il permet de naviguer entre les fenêtres de configuration, d’aide et à propos.

2 – Émulation d’une page web, il faut que la page web émulée existe réellement

3 – Code source de la page web en question

### Fenêtre d’aide

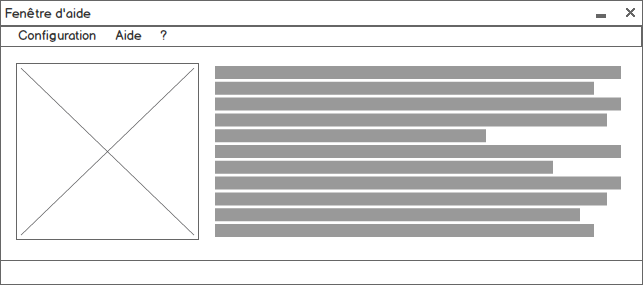


Fig. - Fenêtre d'aide

Fenêtre expliquant comment utiliser l’application

### Fenêtres à propos

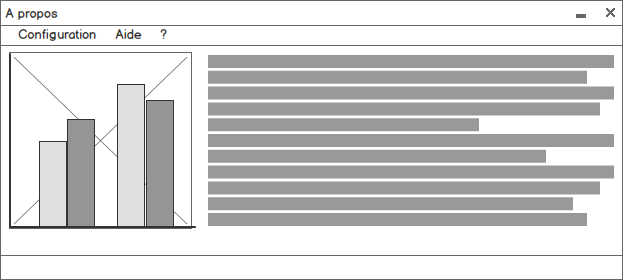


Fig. - Fenêtre à propos

Fenêtre d’information générale sur l’application et ses développeurs

# Analyse organique

## Diagramme de classes

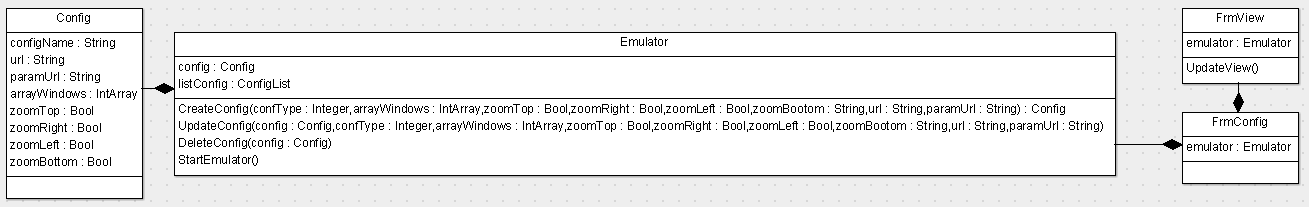


Fig. - Diagramme de classes

### Classe – FrmView

Cette classe est la vue de l’application. Elle permet d’afficher les fenêtres de l’application et de se rafraîchir avec les informations stockée dans la fenêtre de configuration.

#### Attribut

* **emulator : Emulator** -> Instance du contrôleur pour accéder à ses méthodes.

#### Méthode

**UpdateView()**

* Rafraîchit la vue en fonction des informations reçues de FrmConfig.

### Classe – FrmConfig

Cette classe est la vue de démarrage de l’application. Elle permet de créer et gérer les configurations de l’émulateur.

#### Attribut

* **emulator : Emulator** -> Instance du contrôleur pour accéder à ses méthodes.

### Classe – Emulator

Cette classe s’occupe de gérer les différentes configurations. Il s’agit du contrôleur de l’application.

#### Attributs

* **config : Config** -> Stocke la configuration actuellement utilisé par l’application.
* **listConfig : ConfigList** -> Liste de configuration, permet de stocker les configurations enregistrer.

#### Méthodes

**CreateConfig(…) : Config**

* Créer une nouvelle configuration.
* Prend en paramètre les différents attributs nécessaires à une configuration.

**UpdateConfig(…)**

* Met à jour une configuration enregistrée dans listConfig.
* Prend en paramètre la configuration que l’on souhaite modifier ainsi que les différents attributs nécessaires à une configuration.

**DeleteConfig(config : Config)**

* Supprime une configuration enregistrée dans listConfig.
* Prend en paramètre la configuration que l’on veut supprimer.

### Classe – Config

Cette classe s’occupe de gérer les informations liées aux configurations. Il s’agit du modèle de l’application

#### Attributs :

* **ConfigName : String** -> Le nom de la configuration
* **URL : String** -> L’URL de la page WEB
* **ParamURL : String** -> Les paramètres de l’URL pour les lignes et les colonnes
* **ArrayWindows : IntArray** -> Tableau de chiffres représentant les lignes et les colonnes de l’application
* **Zoom Top : Boolean** -> Valeur booléenne qui spécifie si l’appareil affiche un ensemble de fenêtres constitué de la ligne de fenêtre du haut.
* **Zoom Bottom : Boolean**-> Valeur booléenne qui spécifie si l’appareil affiche un ensemble de fenêtres constitué de la ligne de fenêtre du bas.
* **Zoom Left : Boolean**-> Valeur booléenne qui spécifie si l’appareil affiche un ensemble de fenêtres constitué de la ligne de fenêtre de gauche.
* **Zoom Right : Boolean**-> Valeur booléenne qui spécifie si l’appareil affiche un ensemble de fenêtres constitué de la ligne de fenêtre de droite.

# Conclusion

# Annexes