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 |
| 09.03.18 | 1.2 | Analyse fonctionnelle | Ronaldo & Jean-Daniel |
| 09.03.18 | 1.3 | Analyse organique | Dario & Romain |
| 13.04.18 | 1.4 | Correction globale | Romain |
|  |  |  |  |
|  |  |  |  |

Table des matières

[1 Introduction 3](#_Toc511374485)

[2 Rappel du cahier des charges 3](#_Toc511374486)

[2.1 But 3](#_Toc511374487)

[2.2 Spécifications 3](#_Toc511374488)

[2.3 Restrictions 3](#_Toc511374489)

[2.4 Environnement 3](#_Toc511374490)

[2.5 Livrables 3](#_Toc511374491)

[3 Analyse fonctionnelle 4](#_Toc511374492)

[3.1 Fonctionnalités 4](#_Toc511374493)

[3.1.1 Afficher une vue d’ensemble des appareils sous forme de pages WEB 4](#_Toc511374494)

[3.1.2 Les URLS des pages sont paramétrables 4](#_Toc511374495)

[3.1.3 Modifier le nombre de pages affichées 4](#_Toc511374496)

[3.1.4 Afficher le code source de chacune des pages WEB (FACULTATIF) 4](#_Toc511374497)

[3.1.5 Concevoir un zoom de plusieurs pages WEB en une (FACULTATIF) 4](#_Toc511374498)

[3.1.6 Créer des configurations personnalisées (FACULTATIF) 4](#_Toc511374499)

[3.2 Interfaces 5](#_Toc511374500)

[3.2.1 Fenêtre de configuration 5](#_Toc511374501)

[3.2.2 Fenêtre principale 7](#_Toc511374502)

[3.2.3 Fenêtre d’aide 8](#_Toc511374503)

[3.2.4 Fenêtres à propos 8](#_Toc511374504)

[4 Analyse organique 9](#_Toc511374505)

[4.1 Diagramme de classes 9](#_Toc511374506)

[4.1.1 Classe – FrmView 9](#_Toc511374507)

[4.1.2 Classe – FrmConfig 9](#_Toc511374508)

[4.1.3 Classe – Emulator 9](#_Toc511374509)

[4.1.4 Classe – Config 10](#_Toc511374510)

[5 Tests 11](#_Toc511374511)

[6 Conclusion 12](#_Toc511374512)

[6.1 Améliorations envisageables : 12](#_Toc511374513)

# Introduction

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

## Spécifications

L’application est capable de :

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

## Restrictions

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

## Environnement

Matériel nécessaire 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
* Application

# Analyse fonctionnelle

## Fonctionnalités

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

Cette fonctionnalité permet à l’utilisateur d’émuler les appareils sous forme de pages WEB. Il peut choisir le nombre de navigateurs. L’application affiche des pages WEB sous forme d’une grille.

### Les URLS des pages sont paramétrables

Cette fonctionnalité permet à l’utilisateur de personnaliser l’URL des pages WEB. Il peut définir les noms des paramètres ainsi que leurs valeurs.

### Modifier le nombre de pages affichées

Cette fonctionnalité permet à l’utilisateur de modifier le nombre de pages WEB affichées. Il peut choisir le nombre d’appareils qu’il souhaite afficher par ligne et par colonnes.

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

Cette fonctionnalité permet à l’utilisateur d’afficher le code source de chacune des pages WEB afin de pouvoir les débugger.

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

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

### Créer des configurations personnalisées (FACULTATIF)

Cette fonctionnalité permet à l’utilisateur de créer des configurations et de les enregistrer. Il peut choisir le nombre de page WEB qu’il souhaite afficher par ligne et par colonne, les zooms qu’il souhaite appliquer ainsi que les URLS des pages WEB. Il peut enregistrer une configuration en lui donnant un nom.

## Interfaces

### Fenêtre de configuration

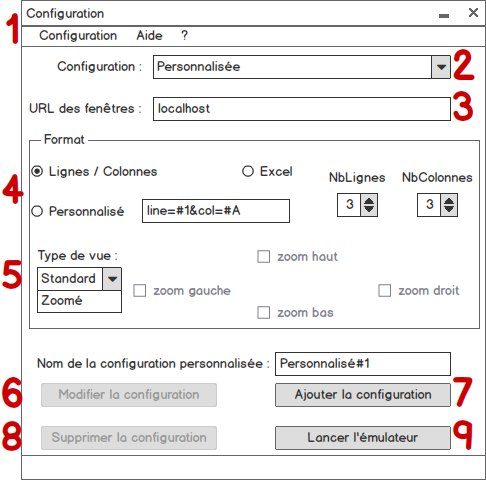


Fig. 1 - Fenêtre de configuration

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 sur lequel on souhaite que les fenêtres se connectent.

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

5 – L’utilisateur a aussi le choix du mode de vue. La vue standard est une vue avec les 9 tablettes affichées en rectangle par exemple. La vue zoomée permet à l’utilisateur de choisir les bords qu’il souhaite afficher 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 d’enregistrer la configuration en indiquant un nom.

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

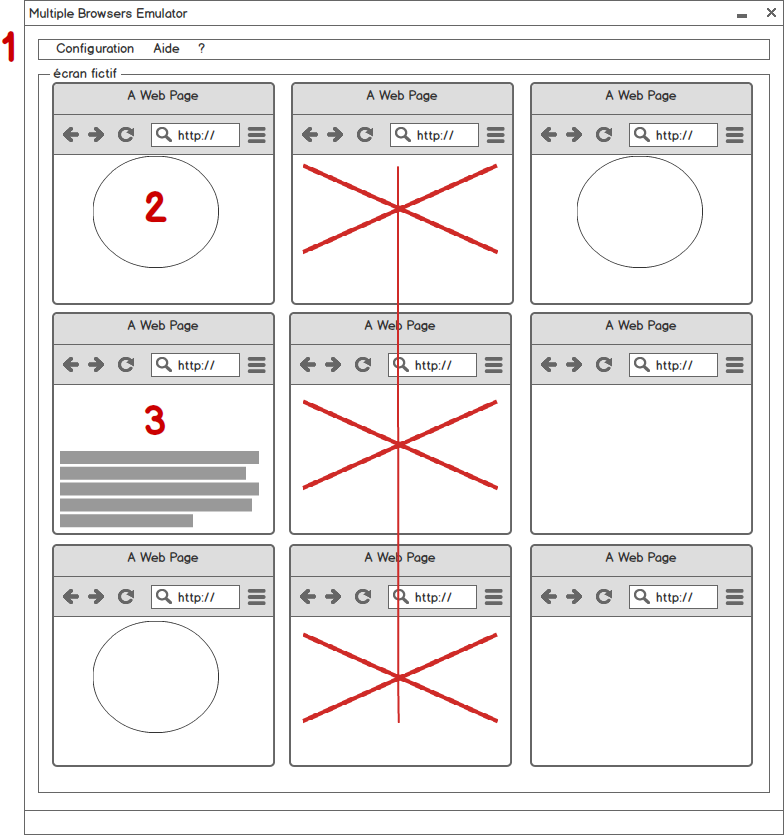


Fig. 2 - Fenêtre d’émulation

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

2 – Émule l’appareil en se connectant à une page WEB

3 – Code source de la page WEB en question

### Fenêtre d’aide

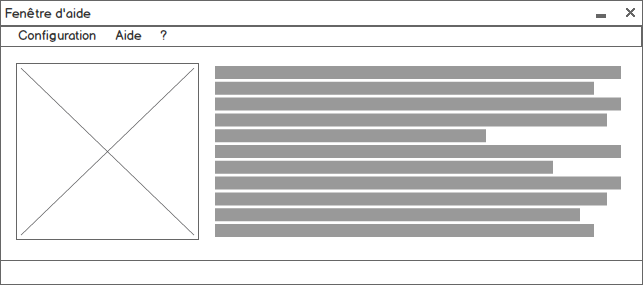


Fig. 3 - Fenêtre d'aide

Fenêtre expliquant comment utiliser l’application

### Fenêtres à propos

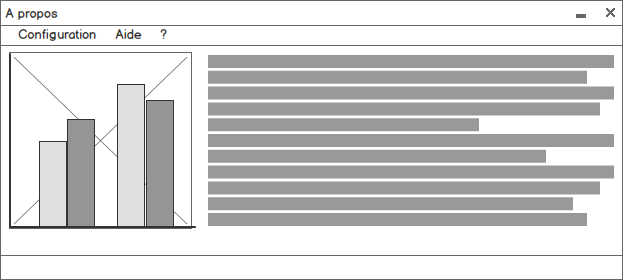


Fig. 4 - Fenêtre à propos

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

# Analyse organique

## Diagramme de classes

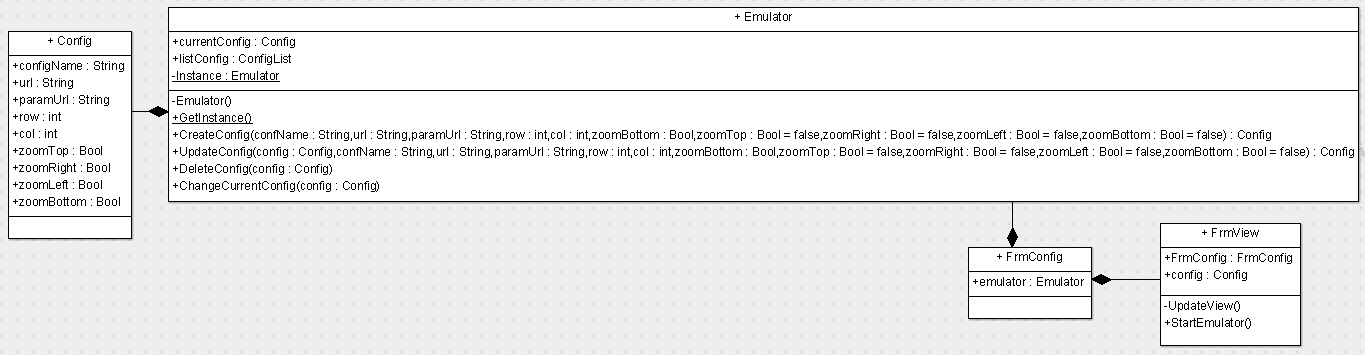


Fig. 5 - Diagramme de classes

### 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 stockées 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 utilisée par l’application.
* **listConfig : ConfigList** -> Liste de configuration, permet de stocker les configurations enregistrées.

#### 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 fenêtres du haut.
* **Zoom Bottom : Boolean**-> Valeur booléenne qui spécifie si l’appareil affiche un ensemble de fenêtres constitué de la ligne de fenêtres du bas.
* **Zoom Left : Boolean**-> Valeur booléenne qui spécifie si l’appareil affiche un ensemble de fenêtres constitué de la ligne de fenêtres de gauche.
* **Zoom Right : Boolean**-> Valeur booléenne qui spécifie si l’appareil affiche un ensemble de fenêtres constitué de la ligne de fenêtres de droite.

# Tests

|  |  |  |  |
| --- | --- | --- | --- |
| **A tester** | **Comment ?** | **Résultat** | **Bug résolu ?** |
| L’application démarre | On double-clic sur l’exécutable | L’application se lance | Ø |
| Une seule configuration est disponible (Morpion) | On clique sur la liste déroulante et on regarde si « Morpion » est le seul élément de la liste | « Morpion » est le seul élément de la liste | Ø |
| La configuration « Morpion » n’est pas modifiable | On essaye de modifier la configuration en changeant les valeurs dans les champs | Les champs sont désactivé et donc impossibles à modifier | Ø |
| Le bouton « Lancer l’émulateur » lance l’émulateur | On clique sur le bouton « Lancer l’émulateur » | L’émulateur se lance | Ø |
| La configuration « Morpion » est créer dans l’émulateur | On lance l’émulateur | La configuration « Morpion » est créé dans l’émulateur (9 WEB Browsers) | Ø |
| Les WEB Browsers affichent une page | On regarde si les WEB Browsers affichent une page | Les WEB Browsers affichent une page | Ø |
| Fermer la fenêtre de l’émulateur fait réapparaître la fenêtre de configuration | On ferme la fenêtre de l’émulateur | La fenêtre de l’émulateur se ferme et la fenêtre de configuration réapparaît | Ø |

# Conclusion

L’application est capable de générer 9 navigateurs WEB avec une configuration donnée. Il n’est pas encore possible de gérer les configurations ainsi que le nombre de fenêtres affichées. Il y a également un problème de focus lorsque l’on utilise le clavier dans les navigateurs émulés. Actuellement, afin de garder le focus il faut cliquer sur le navigateur. Il est actuellement impossible de débugger l’application via les navigateurs.

## Améliorations envisageables :

* Gérer les configuration (CRUD) en les sérialisant dans un fichier appart
* Rendre les réglages dynamiques en chargeant la configuration
* Appliquer les changements de l’utilisateur à la configuration courante (nombre de lignes et colonnes, format, type de vue, etc.)
* Rendre l’interface des navigateurs redimensionnable
* Gérer le focus des navigateurs
* Utiliser une API afin de pouvoir afficher les outils de développement dans les navigateurs et pouvoir le debugger.

Au final, on peut s’apercevoir que le projet était trop ambitieux par rapport au temps que nous avions à disposition pour concevoir l’application dans son intégralité (documentation, implémentation et tests). Cependant, les fonctionnalités minimales sont développées et fonctionnelles.