2020 English Project Report

Malaury KESLICK and Maxime MAINGUET

1 Introduction

This paper presents the work we have done in the context of the project requested for the English unit at TELECOM Nancy. Its purpose was to develop a computer-based tool designed to help learning English as a language, or about an anglophone country's culture. We chose to combine both by working on American National Days. As we were amused to realise that there exists a least one reason to celebrate each day in the United States of America, we thought that presenting these things in an application would help to learn more about American national customs and oddities while discovering some vocabulary all at once.

Thus, as a first step, we will present our tool and its operation. Then we will switch for French to develop our work in a technical way. Finally you will find the conclusion and annexes. We wish you an interesting and enjoyable reading.

To those who help us face these tough times.
To those who fight for the silent majority.

Contents

2 Tool presentation

As above-mentioned, our tool is an application. Anyone having Java8 as current version of Java on his computer can launch it by a double click on the "PigsinaBlanket.jar" icon. A square window will appear, and show you the national day of the current date, illustrated by a picture, along with a description of this day and five English words related to the theme. By simply passing your mouse over one of these words, the French translation will replace it. If your mouse goes out of the word field, the English version will appear again. Furthermore, if the description is too long to fit in the window, a scroll bar appears.

Thus, our application is thought for a daily use, to help learning a few things day by day. But to fulfill everyone's curiosity and let you access your date of birth or relationships anniversary, there are two ways to switch the presented day. Indeed you can use the back and left arrows, which respectively lead to the previous and the next day. You will still be able to know what day you are looking at thanks to the written date at the top of the window.

The user may also want to access a precise date, far from the current one. That is why there is a little calendar icon at the right of the current date. By clicking on it, a calendar of the current month will appear and you will be able to access any day, of any month, of any year.

We chose to offer a wide variety of themes in this application, from the most serious to the silliest. We thought it is as interesting to learn more about America's history as about American's fantasies. That is why a user can enjoy *Mental Health Day* as well as *Pigs-In-a-Blanket Day*!

The following picture exposes the very simple use of our interface. The top left picture is what the user would have seen if he had launched the application on the 11th of February : *Cheese Fondue Day*! Whenever he passes its mouse on one of the five written words/expressions at the bottom, the french translation appears as the top right screenshot shows, for the expression "Communal pot". The bottom left picture points the calendar out, obtained by clicking on the button at the right of the current date. Finally, on the bottom right image, we can see the state of our application after a click on the "back" arrow, or on the previous date, with the *Siblings Day*.

Figure 1: Top left : 11th April main view — Top right : Passing mouse over "Communal Pot" box — Bottom left : Click on the calendar icon — Bottom right : Click on "previous" arrow

3 Conception de l'application

Afin de mener à bien ce projet, nous avons choisi d'utiliser la framework JavaFX, qui permet de créer des interfaces graphiques en langage Java. Nous avons adopté une architecture Model-Controller-View, qui sera explicitée dans la première sous-partie.

3.1 Mise en oeuvre informatique

Tout d'abord, l'une de nos tâches les plus chronophages lors de ce projet fût la récupération des informations nécessaires à la complétude de notre application. En effet il nous a fallu, pour chaque jour de l'année, sélectionner l'une des journées nationales se tenant ce jour, en récupérer une description, une illustration et trouver cinq mots de vocabulaires intéressants. Nous avons regroupé ces informations dans un tableur excel.

A partir de ce tableau, nous avons construit un parseur permettant de récupérer ces informations de manière automatique, et de créer des objets Day, comportant autant d'attributs que d'éléments cités dans le paragraphe ci-dessus. Ces deux classes (Parser et Day) constituent la partie *Model* de l'architecture.

La partie *Controller* est assurée par huit classes, plus ou moins imbriquées les unes dans les autres. Nous vous renvoyons au diagramme de classes clôturant cette sous-partie, celui-ci appuyant de manière plus visuelle ce qui va suivre. La classe GlobalView est celle permettant de regrouper chaque morceaux de l'application en un seul bloc. Ces morceaux sont au nombre de quatre : la partie principale, la bande gérant les boutons *Précédent* et *Suivant*, la bande gérant l'affichage du nom de la journée et la section présentant les mots de vocabulaire. La partie principale est elle-même constituée de deux sous-parties : la partie gauche, hôte de l'image illustrant la journée, et la partie droite affichant la description de la journée. Enfin, la partie vocabulaire contient cinq objets "Label", qui sont les objets responsables de l'affichage des mots de vocabulaire de la journée. Les différents layout utilisés pour cette structure sont explicités dans le diagramme suivant.

La partie *View* de l'architecture est alors gérée par la classe Launcher, regroupant les classes précédemment citées pour les imbriquer dans la fenêtre finale de l'application. C'est également cette classe qui gère les changements de vue, et les dimensions finales de l'outil.

Finalement, le design de l'application a été choisi sobre et épuré, dans des tons neutres qui ne jugeront avec aucune des illustrations aux tons très différents. Les polices d'écriture ont été choisies arrondies et chaleureuses afin de rendre la lecture agréable.

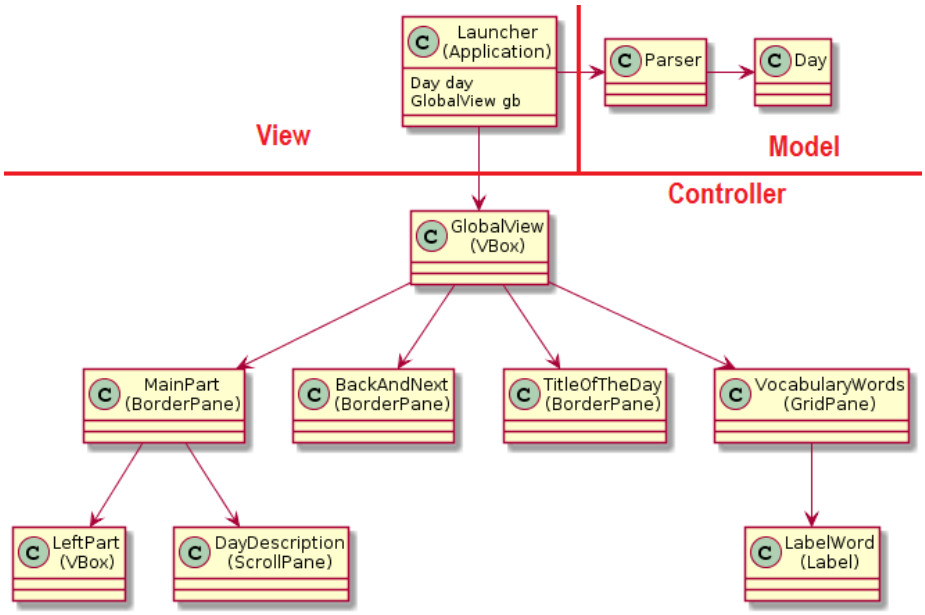

Figure 2: Diagramme de classes

3.2 Problèmes rencontrés et solutions apportées

Le premier problème rencontré fût la lecture du fichier Excel contenant les données relatives aux différents jours de l'année. En effet, on ne peut lire le fichier directement comme un CSV (Comma-separated values) car les données sont cachées. Par conséquent nous avons utilisé l'interface **xml-beans** permettant de lire un fichier de type xlsx. Cependant le .jar (archive exécutable) correspondant nécessitait plusieurs autres .jar afin de fonctionner. Il a donc était nécessaire de trouver la bonne version de chaque .jar, ce qui nous a pris beaucoup de temps puisqu'il fallait déterminer les versions étant compatibles entre elles.

Le second problème rencontré fût le temps nécessaire à la construction de la base de données. En effet, la base de données comprend 366 jours, 366 photos, 366 descriptions et surtout $5 * 366 = 1830$ mots de vocabulaire ainsi déterminés. La solution de ce problème fût de démarrer à chercher les descriptions et le vocabulaire nécessaires en avance. Une bonne répartition des tâches au sein du groupe de projet a aussi fait partie de la solution à ce problème de taille.

4 Conclusion

During this project, we were amused by the enormous amount of celebration happening every single day in the USA. We learned about events and things we didn't know much about, and sometimes didn't even know it existed at all. We also discovered some vocabulary about a wide range of topics. Furthermore we praticed our application development skills, especially using the JavaFX framework. In a nutshell, even though this project was time consuming - more than 50 hours per person - it was an interesting learning experience.

(Malaury) I would personally add that it is really damageable to not really evaluate our technical contribution to this project. This is what had taken us so long to do, thus it is really frustrating to be graded on the final overview only. Moreover, the scheduling of this project is, in my opinion, the worst choice possibly made. Even if I understand that we don't have the required qualifications until then, and that this is not a priority in third year, the second semester of the second year is well-known to be the toughest of this curriculum. As a consequence, it is really complicated to deliver something matching our own expectations without impacting on the rest of our hectic agenda, especially when we know we are not graded on the technical side of this work. I really appreciated developing this application, but I always had the thought of 'They won't grade that anyway' and it gave me the feeling of and under-recognized work, which was really demotivating.

5 Annexes

5.1 Computer-based sources

The whole operation of our application is based on our **P**rojet de **C**onception et de **D**éveloppement unit from the first semester of the second year at TELECOM Nancy, during which we learned to develop applications using JavaFx.

5.2 Content sources

The website **National Day Calendar** was really helpful while collecting data for our application. That is where we got descriptions from. Each day's illustration comes from **Google Image**, and the vocabulary is picked from descriptions or our mind. We used **Linguee** a lot to help us translate these words and expressions.