

History Pub

Découvrez Soignies en boisson

PROJET DE DÉVELOPPEMENT D'APPLICATION MOBILE

Nathan Raspe & Matteo Taroli

27 Novembre 2015

Table des matières

Table des matières	i
Table des illustrations	ii
1 Introduction	1
2 Mode d'emploi	2
2.1 Caractéristiques communes aux épreuves	2
2.2 Epreuve Question à Choix Multiple	3
2.3 Epreuve de texte à trous	4
2.4 Epreuve Question Ouverte	5
2.5 Epreuve Photographique	6
2.6 Fin du jeu	7
2.7 Informations générales	8
3 Architecture	9
4 Bugs connus	10
5 Utilisation de code open-source	11
6 License	12

Table des illustrations

1	Ecran indiquant à l'utilisateur où se déplacer pour commencer une étape.	2
2	Question à choix multiple, avant et après avoir triché	3
3	Question à trous, avant et après avoir triché	4
4	Question ouverte, avant et après avoir triché	5
5	Question photographique, avant et après avoir triché	6
6	Ecran final	7
7	Menu de la barre d'action et popup d'avancement	8

1 Introduction

Dans le cadre du cours de Développement d'Application Mobile, nous avons dû développer une application reprenant le concept du jeu de piste. Cette application, nommée HISTORY PUB vous propose de découvrir la ville de Soignies en passant par les différents bars qu'offre celle-ci.

Sur le chemin entre ces différents établissements, vous aurez l'occasion d'en apprendre plus sur l'histoire et le folklore de la ville. Durant les différents arrêts, vous aurez bien entendu l'occasion de profiter des bars et des diverses boissons proposées avant de reprendre le chemin vers d'autres découvertes !

Attention tout de même, l'alcool est à consommer avec modération, sous peine de ne plus pouvoir répondre aux questions données !

2 Mode d'emploi

L'utilisation de HISTORY PUB est très facile. Une fois l'application lancée, vous recevez les indications vous permettant de trouver la zone de la première étape du jeu.



FIGURE 1 – Écran indiquant à l'utilisateur où se déplacer pour commencer une étape.

Une fois dans la zone, et après avoir lu une courte description de l'endroit et où de son histoire, vous vous retrouverez devant une épreuve qui peut prendre différentes formes. Ces épreuves sont décrites ci-dessous.

2.1 Caractéristiques communes aux épreuves

Lors des épreuves de question à choix multiples ou de question ouvertes, votre réponse doit être vérifiée. Cette vérification n'est faite qu'une fois la réponse validée par l'utilisateur, c'est à dire une fois que le bouton OK est pressé. Cette vérification peut mener à :

Pas de réponse donnée

Un toast invite l'utilisateur à répondre à la question.

Réponse correcte

Une fenêtre popup apparait pour féliciter l'utilisateur et afficher le nombre de points gagnés.

Réponse incorrecte

La fenêtre popup se désole de la mauvaise réponse et donne la réponse correcte, permettant à l'utilisateur d'apprendre de ses erreurs.

Dans les deux cas où une réponse est donnée, la fenêtre popup donne de plus ample informations par rapport à la réponse à la question, comme vous pouvez le voir sur la capture d'écran suivante. Cette fenêtre permet aussi de partager l'avancement dans le jeu.

De plus, 2 autres boutons sont également présents. Ceux-ci sont communs à toutes les épreuves, réalisent les mêmes actions mais de manières différentes en fonction du type d'épreuve. (Voir sections 2.2 à 2.5)

Bouton de triche

Ce bouton sert à obtenir la réponse de l'épreuve. Aucun point ne sera accordé à l'utilisateur si ce bouton est pressé. L'aide et la triche ne seront plus disponibles. *Remarque* : le message affiché reste « Bonne réponse », bien que ce message soit inadapté.

Bouton d'aide

Ce bouton sert à obtenir un indice. La moitié des points sera accordé à l'utilisateur si ce bouton est pressé. L'aide ne sera plus disponible.

2.2 Epreuve Question à Choix Multiple

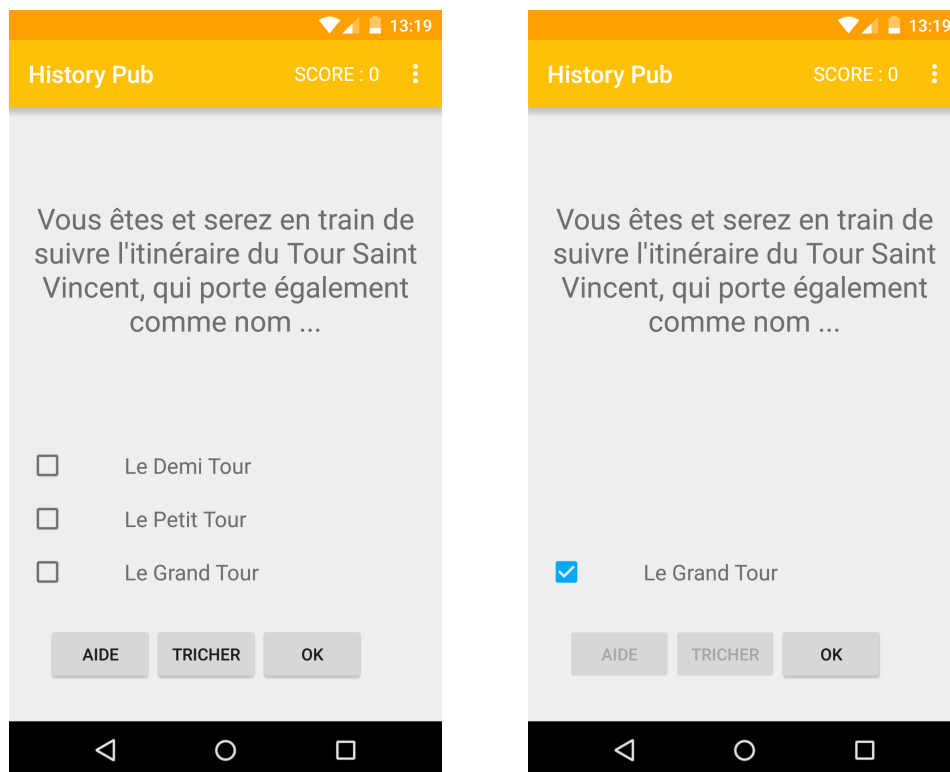


FIGURE 2 – Question à choix multiple, avant et après avoir triché

Pour répondre à une question à choix multiples, il vous suffit de cliquer sur la réponse que vous pensez correcte. Vous pouvez cliquer n'importe où sur la ligne de ce choix, que ce soit sur la

checkbox ou sur le texte de la réponse proposée.

Un symbole ✓ s’affiche dans la checkbox de la réponse sélectionnée. Si celle ci vous convient, vous pouvez confirmer votre choix grâce au bouton OK. Sinon, vous êtes libre de changer ce choix autant de fois que vous le désirez.

Pour ce type d’épreuve :

Bouton de triche

Il retire toutes les réponses incorrectes et coche automatiquement la bonne réponse. Il faut ensuite appuyer sur le bouton OK.

Bouton d’aide

Il enlève aléatoirement une mauvaise réponse.

2.3 Epreuve de texte à trous

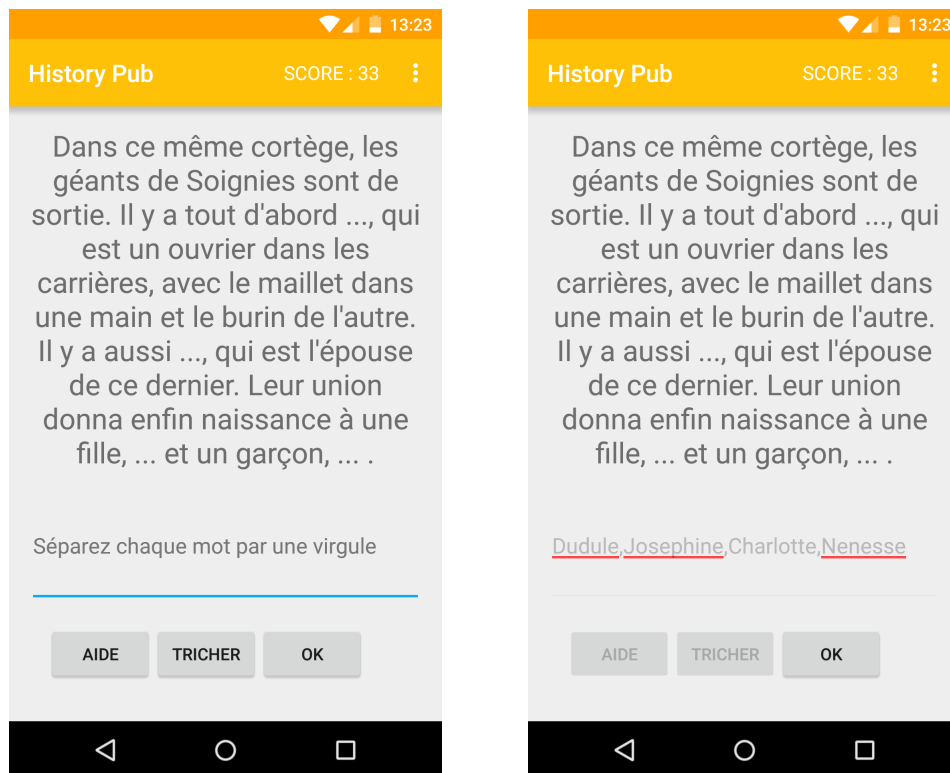


FIGURE 3 – Question à trous, avant et après avoir triché

Pour réponse à une question de texte à trous, il vous suffit d’entrer les différents mots manquants dans l’unique champs de texte. Ces différents mots doivent être séparés par une virgule. Seules les réponses comprenant tous les mots corrects sont acceptées.

Pour ce type d’épreuve :

Bouton de triche

Il remplit automatiquement le champs de réponse avec la ou les bonne(s) réponse(s). Aucune

modification ne sera alors possible dans cet EditText. Il faut ensuite appuyer sur le bouton OK.

Bouton d'aide

Il remplit automatiquement le champs de réponse avec la première lettre de chaque mot, et une virgule entre les différentes lettres ainsi révélées. Il faudra cependant compléter la réponse.

2.4 Epreuve Question Ouverte

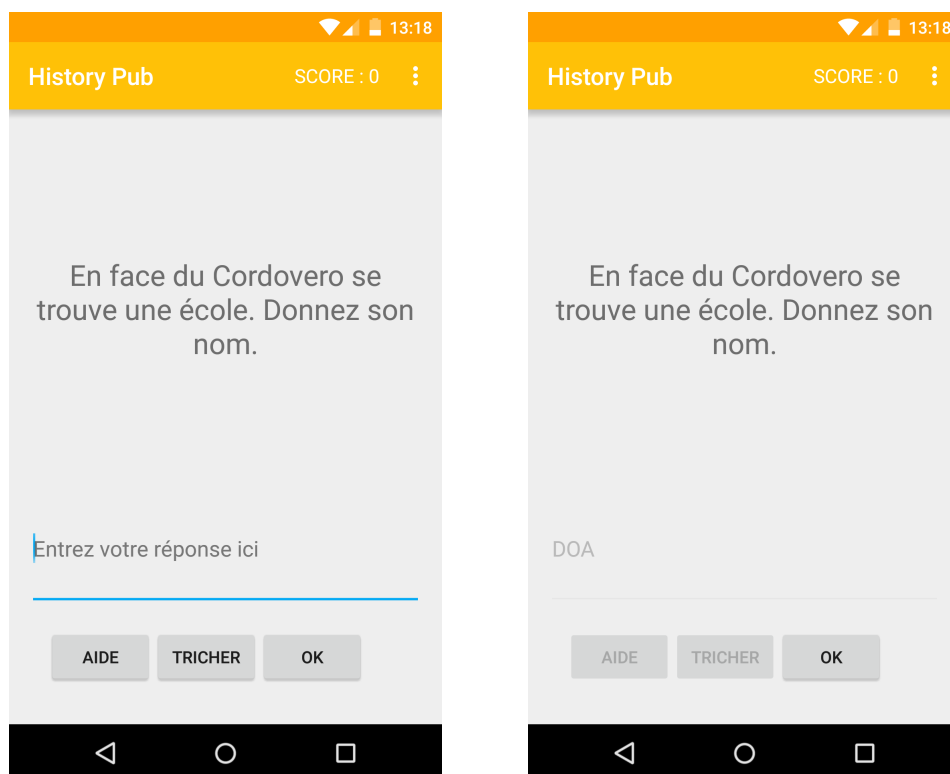


FIGURE 4 – Question ouverte, avant et après avoir triché

Pour répondre à une question ouverte, il vous suffit d'écrire la réponse demandée dans la zone prévue à cet effet. Cette réponse est unique mais peut être entrée de différente manière. Par exemple pour une question dont la réponse attendu serait « Institut Paul Lambin », la réponse « IPL » serait aussi acceptée.

Une fois la réponse entrée, vous pouvez confirmer votre solution avec le bouton entrée de votre clavier virtuel ou après avoir quitté ce clavier, avec la bouton OK.

Pour ce type d'épreuve :

Bouton de triche

Il remplit automatiquement le champs de réponse avec la bonne réponse. Aucune modification ne sera alors possible dans cet EditText. Il faut ensuite appuyer sur le bouton OK.

Bouton d'aide

Il remplit automatiquement le champs de réponse avec la première lettre de la réponse. Il faudra cependant compléter la réponse.

2.5 Epreuve Photographique

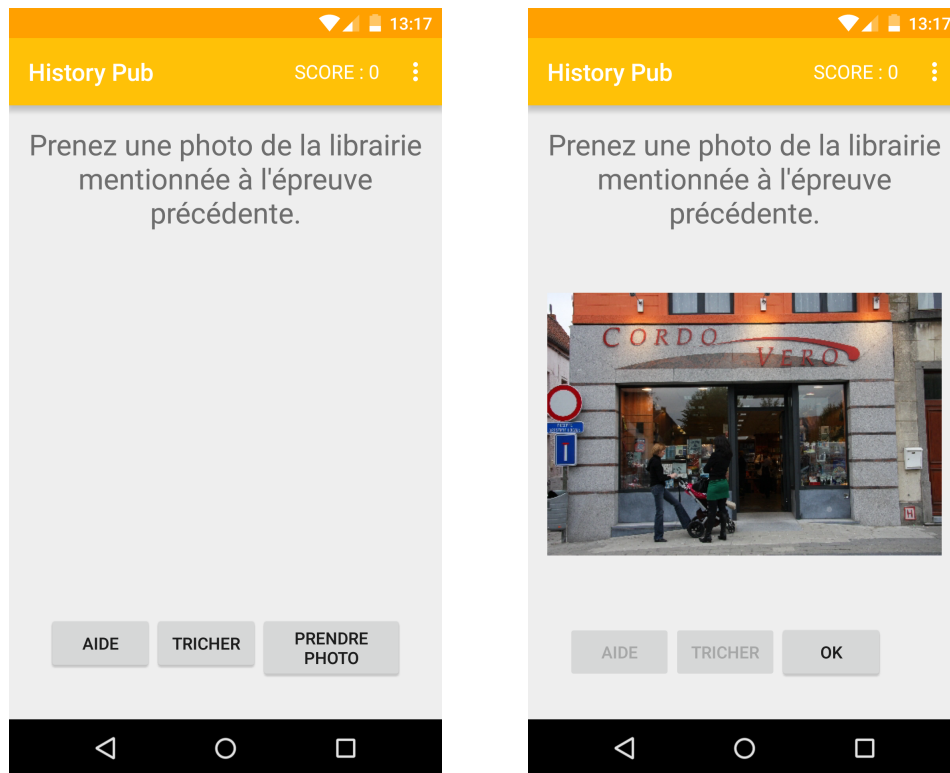


FIGURE 5 – Question photographique, avant et après avoir triché

Pour répondre à une question photographique, il vous suffit de vous trouver dans la zone requise et de prendre une photo du bâtiment ou autre objet demandé. Une fois la photo prise, vous avez un aperçu de l'image et il ne vous reste plus qu'à accepter cette photo.

Aucune vérification n'est faite sur ce type d'épreuve. Faites donc de votre mieux, on compte sur vous !

Pour ce type d'épreuve :

Bouton de triche

Il affiche automatiquement la bonne image dans la zone de prévisualisation. Il faut ensuite appuyer sur le bouton OK.

Bouton d'aide

Il affiche automatiquement la bonne image dans la zone de prévisualisation pendant 5 secondes. Il faudra cependant prendre une photo.

Remarque : Les épreuves photographiques ne sont pas géolocalisées, bien qu'elles devraient l'être.

2.6 Fin du jeu



FIGURE 6 – Ecran final

Une fois tout le parcours fini, vous serez face à un écran récapitulatif vous permettant de savoir le nombre de points que vous avez marqués ainsi que le temps que vous avez mis pour finir ce parcours.

Cette page vous permet aussi de partager votre score avec vos connaissances ainsi que de les mettre au défi de vous battre !

2.7 Informations générales

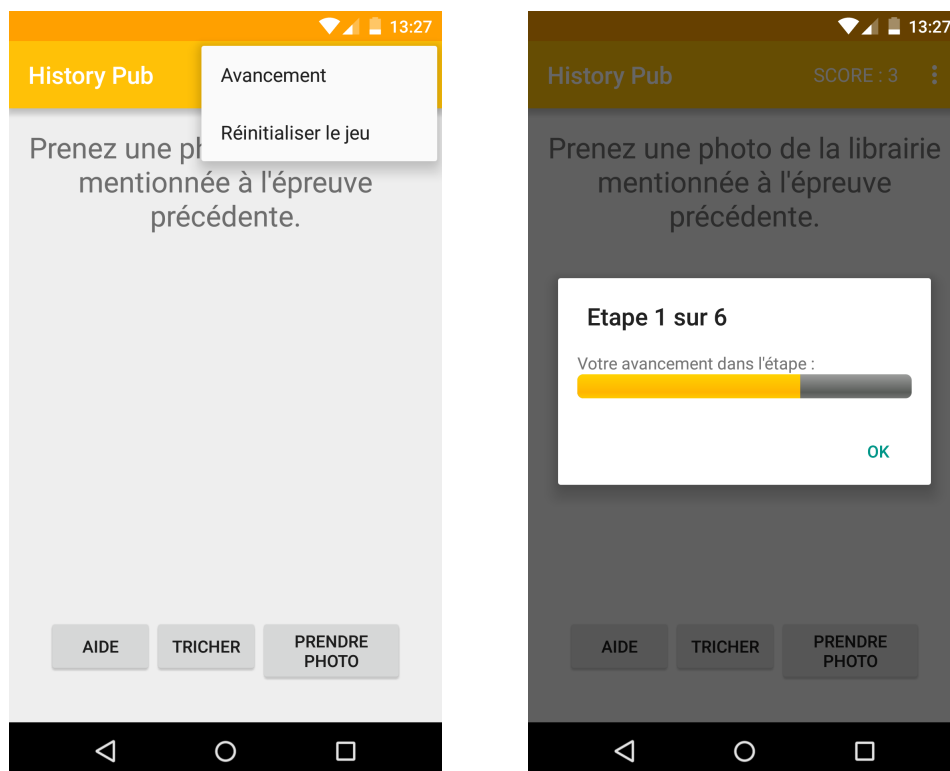


FIGURE 7 – Menu de la barre d'action et popup d'avancement

A tout instant durant le jeu, vous avez la possibilité de voir votre score dans la barre d'action de votre application.

Vous pouvez aussi, via le menu de cette barre d'action, voir votre avancement ou remettre le jeu à zéro. Ces deux actions sont montrées dans les captures d'écran ci-dessus.

3 Architecture

L'architecture de l'application suit, autant que faire se peut, les règles énoncées par GOOGLE.

Les messages et autres chaînes de caractères affichés à l'utilisateur sont repris dans le fichier *string.xml*, permettant une traduction et donc une localisation aisée. Certains éléments visuels ont un style et des caractéristiques communs repris dans le fichier *styles.xml*, permettant de mettre à jour le design de l'application de façon simple et sans devoir rechercher dans les différents fichiers de layout.

Concernant le code java, nous avons suivi les conventions de nommage des attributs énoncés par l' ANDROID OPEN SOURCE.PROJECT. C'est à dire que :

- les attributs **non public**, **non static** commencent par un m
- les attributs **static** commencent par un s
- les autres attributs commencent par une lettre minuscule
- les attributs **public static final** s'écrivent en MAJUSCULE_AVEC_UNDERSCORES

Le code est de plus, bien entendu, commenté et contient de la javadoc la plus compréhensible possible.

Concernant les classes java, nous avons séparé les classes dite « POJO », qui sont de simples objets Java de classes relative à ANDROID, c'est à dire les activités ainsi que des « User Case Controller » (Correspondant ici à la classe gérant le fichier XML et également le déroulement du jeu).

Dans les POJO, nous avons différentes classes, correspondant aux différentes épreuves, aux étapes ainsi qu'à une zone autour d'un point géographique.

Les activités elles correspondent chacune à une étape, épreuve ou autre écran affiché à l'utilisateur. Elles gèrent donc l'interaction entre l'utilisateur et l'interface visuelle. L'interface en elle-même étant gérées par les fichier XML du dossier layout.

L'UCC gère le chargement du fichier XML et est implémenté en tant que **singleton**. Cela permet de charger le jeu complet en une seule fois (comprenez les différentes étapes, épreuves et tout ce qui les compose) et de permettre par la suite à toutes les activités de piocher l'étape et/ou l'épreuve qui lui est nécessaire.

Nous avons bien conscience que le pattern Singleton n'est plus vu comme une bonne pratique voire même comme un anti-pattern, mais n'ayant aucune connaissance en injection de dépendance sur la plateforme d'Android, nous avons finalement opté pour ce pattern.

4 Bugs connus

Le seul bug connu à ce jour réside dans l’affichage des questions et propositions de réponses sur les petits écrans. En effet, ceux-ci sont hachés (entendez par là que le texte dépasse de l’emplacement prévu) et il est donc impossible de les lire dans leur intégralité. Nous avons cependant plusieurs pistes pour le résoudre :

- Ajout d’un ScrollView pour les questions. Dans ce cas, les questions nécessitant trop de place seront lisibles en scrollant.
- Créer un dossier values pour petits écrans. Dans ce cas, les styles (donc la taille de la police, des layouts, ...) seraient adaptés en fonction de la taille de l’écran.
- Combinaison des 2 solutions ci-dessus.

Dans notre cas, nous avons opté pour la deuxième solution (la première ne nous convenant pas et la troisième étant trop ambitieuse dans le temps imparti). Cependant, par manque de temps, nous n’avons pas réussi à mettre cette solution en place.

5 Utilisation de code open-source

Le code de HISTORY PUB reprend, à certains endroits précis, du code open-source. Ces parties de code open-source se retrouvent ci-dessous :

EtapeActivity.java et PhotoActivity.java

La partie géolocalisation reprend en grande partie le code fourni par GOOGLE sur leur plateforme `developer.android.com` ainsi que sur GITHUB à cette adresse : <https://github.com/googlesamples/android-play-location>. Ce code est fourni sous la license **Apache**, permettant son inclusion dans un projet sous license **GPLv3**.

PhotoActivity.java

La partie prise de photo reprend le code fourni par GOOGLE sur la plateforme précédemment citée, toujours sous la license **Apache**.

PhotoActivity.java

La méthode permettant de charger et d'afficher une image depuis les assets (lors d'un cheat ou d'une aide), provient de STACKOVERFLOW à cette adresse : <http://stackoverflow.com/a/11734899>. Le code se trouvant sur STACKOVERFLOW est sous licence **Apache**, qui est donc compatible avec la **GPLv3**.

Style.css

Le design des boutons, façon Material Design, est repris du CODEPEN créé par RayCh et trouvable à cette adresse : <http://codepen.io/iraycd/pen/dHrxv>. Le code posté sur CODEPEN est sous license **MIT**, elle aussi compatible avec la **GPLv3**.

6 License

HISTORY PUB est placé sous license GNU GENERAL PUBLIC LICENSE version 3.

Cela signifie que n'importe qui **peut** :

- utiliser notre code de façon privée
- utiliser notre code dans des projets à but commercial
- redistribuer notre code
- modifier notre code

Mais cela signifie aussi qu'une personne utilisant tout ou partie du code de HISTORY PUB dans un projet **doit** :

- publier les sources de ce projet
- inclure une copy de la license et copyright dans ce projet
- préciser les changements importants fait à notre code

Cela signifie aussi que notre code est fourni sans garantie et que nous ne pouvons donc pas être tenus responsable de problèmes quelconques liés à son utilisation en dehors de HISTORY PUB.

Pour plus d'information, la license complète peut être trouvée à l'adresse suivante : <https://www.gnu.org/licenses/gpl.html>

Le code source de l'application peut quant à lui être trouvé sur GITHUB à l'adresse suivant : <https://github.com/Crapoo/HistoryPub>. Le code remis avec ce rapport correspond à la release 1.0.0.