



République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Licence en Informatique

Thème

Contribution à la migration du logiciel éducatif GCompris vers le framework Qt Quick : Activité « Faire connaissance avec les animaux de la ferme »

Réalisé par :

- Djalil MESLI

Présenté le 26 Mai 2015 devant la commission d'examination composée de MM.

- Badr Benmammour (Examineur)
- Fazilet Benmansour (Examinatrice)
- Amine Brikci-Nigassa (Encadreur)

Année universitaire : 2014-2015

Remerciements

J'adresse mes remerciements aux personnes qui m'ont aidé dans la réalisation de ce mémoire.

En premier lieu, je remercie M. Amine Brikci-Nigassa, enseignant à l'université de Tlemcen qui, en tant qu'encadreur, m'a guidé dans mon projet et m'a aidé à trouver des solutions pour avancer, ainsi que M. Badr Benmammar et Mme Fazilet Benmansour, pour avoir accepté de juger ce modeste travail.

Je remercie aussi l'équipe des développeurs de GCompris, M. Bruno Coudoin fondateur du projet GCompris qui m'a aidé en me fournissant des données précises sur le QML.

Mes remerciements s'adressent également à tous les enseignants qui m'ont instruit durant mes études à la Faculté des Sciences de l'Université de Tlemcen.

Enfin, je tiens à remercier tous mes collègues d'études, particulièrement ceux de ma promotion, ainsi que ma famille.

Table des matières

Introduction Générale	3
Chapitre 1 : Introduction et définitions	5
I-Introduction	6
II- Aperçu sur les jeux vidéo	6
III- Les Jeux Vidéo éducatifs	6
IV- La suite libre GCompris	7
V-Open Source et Logiciel libre	8
VI- Conclusion	9
Chapitre 2 : Cycle de vie d'un logiciel libre	10
I-Développement communautaire	11
1.1 -Définition :	11
II- Cycle de vie d'un logiciel libre	11
III- Les outils de gestion et de communication	11
3.1 -GitHub :	11
3.2 -Mailing list	13
3.3 -Canal IRC (Internet relay Chat)	13
IV- Développement communautaire sur GCompris durant la migration	13
4.1 -Introduction	13
4.2 -Intérêt du portage	14
4.3 -Les outils :	14
V- Conclusion	18
Chapitre 3 : l'activité « animaux de la ferme »	19
I- Introduction.....	20
II- Description et but de l'activité.....	20
III- Phase de développement.....	22

3.1 -Introduction	22
3.2 -Marquage et réservation	23
3.3 -Récupération des données	23
3.4 -Conception et implémentation	24
IV- L'activité finie sous Qt Quick	28
V- Conclusion	30
Conclusion générale	31
Liste des figures.....	32
Références.....	34

Introduction Générale

Un logiciel est un ensemble de programmes et de procédures nécessaires au fonctionnement d'un système informatique. Il existe plusieurs catégories de logiciels d'application qui sont spécifiques à la résolution des problèmes de l'utilisateur. Le projet GCompris dans le cadre duquel entre mon projet fait partie des logiciels utilisés dans l'enseignement et plus particulièrement des logiciels de jeu ou *ludiciels* éducatifs.

La création d'un logiciel nécessite un travail intellectuel qui prend du temps, effectué le plus souvent par une équipe qui suit une démarche logique et planifiée afin d'obtenir un service de bonne qualité dans les meilleurs délais. C'est ce que nous enseignons la discipline académique désignée sous le terme de *génie logiciel*. Dans le cas particulier des logiciels libres et open source dont GCompris est l'un des exemples les plus réussis, cette démarche repose sur la philosophie de développement « *release early, release often* ». La phrase « Publiez tôt, publiez souvent, et écoutez vos clients »¹ préconise une évolution continue du logiciel qui s'améliore avec le temps mais n'attend pas d'être complet pour être livré à l'utilisateur. Celui-ci entre alors dans le processus de développement grâce à sa rétro-action, en définissant plus précisément ses besoins. Ce qui permet à l'équipe de développement d'adapter le logiciel en conséquence.

Le logiciel GCompris est passé par plusieurs étapes au cours de son évolution qui a commencé il y a une quinzaine d'années. Ce logiciel éducatif a acquis une renommée et un succès certains sur les ordinateurs de beaucoup d'écoles dans le monde faisant la joie de beaucoup d'enfants et donnant satisfaction à beaucoup d'éducateurs. Il traverse aujourd'hui une nouvelle étape, celle de l'adaptation à des plateformes de plus en plus utilisées par les enfants : les tablettes. Cette adaptation nécessite un long travail. Ladite adaptation n'ayant pas encore été complétée, comme à son habitude le logiciel évolue progressivement et les enfants peuvent déjà profiter d'une version partielle qui ne comporte pas toutes les « activités » du logiciel mais est cependant tout à fait utilisable.

¹ Eric S. Raymond : « La Cathédrale et le Bazar », essai de 1997, l'une des références de la philosophie *open source*. <http://www.linux-france.org/article/these/cathedrale-bazar/cathedrale-bazar.html>

Les équipes de développement des logiciels libres sont souvent constituées de personnes très éloignées dans le monde qui s'entraident via internet. Dans la plupart des cas, et c'est vrai pour GCompris, tout contributeur est le bienvenu. C'est ainsi que j'ai pu me joindre à l'équipe en proposant d'aider à l'adaptation de l'une des activités qui n'ont pas encore été réécrites. Faisant ainsi de mon modeste projet de fin d'études une contribution à un projet d'une envergure bien plus importante, j'espère humblement qu'il satisfera plusieurs utilisateurs parmi les enfants du monde entier.

Le présent rapport comporte trois chapitres :

- Dans le chapitre 1 nous présenterons le logiciel GCompris.
- Le chapitre 2 nous donnera une vue panoramique sur le cycle de vie d'un logiciel libre ainsi qu'un aperçu sur le développement communautaire durant la migration de GCompris.
- Le chapitre 3 décrira tout le processus de développement de mon activité « Faire connaissance avec les animaux de la ferme » en ayant comme cahier de charge l'ancienne version de cette dernière (version GTK+).

Chapitre 1 : Introduction et définitions

I-Introduction

Pour aborder le travail qui va suivre, il est nécessaire de définir les concepts utiles à la compréhension et l'appréhension de ce projet. Nous allons présenter un aperçu sur les jeux vidéo, les jeux vidéo éducatifs et enfin la suite logicielle libre ludo-éducative GCompris.

II- Aperçu sur les jeux vidéo

Les jeux vidéo comme tout logiciel sont un ensemble de programmes, fonctionnant en synergie. Le principe est de créer une interaction Homme-Machine via un périphérique d'entrée (Manettes, Clavier, Souris...). Un système réactif visuel permettant d'effectuer des animations et des simulations. Ils utilisent principalement le processeur graphique (GPU) pour le calcul des coordonnées et l'affichage des pixels sur un périphérique de sortie (Ecran).

L'idée de jeu vidéo naît entre les années 1950 et 1970 avec le début de la commercialisation de bornes d'arcade, de jeux et de consoles. Cette idée est le fruit de Ralph Baer, un ingénieur américain de la société Loral Electronics qui l'employait pour concevoir « la meilleure télévision du monde ». [2]

En mai 1951, le premier ordinateur dont le seul but est de permettre de jouer à un jeu, en l'occurrence le jeu de Nim, est exposé par Nimrod à Londres.

A ce jour, les jeux vidéo sont une industrie importante économique qui compte plus de 300 000 emplois, notamment grâce au développement des consoles, ordinateurs et surtout tablettes et smartphones. Cette avancée technologique a permis une propagation rapide des programmes interactifs.

III- Les Jeux Vidéo éducatifs

Un logiciel ludo-éducatif, ou jeu vidéo éducatif, est un logiciel dont le but est d'enseigner certaines notions (logique, connaissances, capacités intellectuelles diverses) à son utilisateur par l'intermédiaire de jeux. Parmi les plus célèbres, on peut citer la série Adibou destinée aux enfants. D'autres moins connus ont basé leur création et leur diffusion sur le modèle du Logiciel Libre comme GCompris. Il existe une université d'été annuelle sur le Multimédia Ludo éducatif LUDOVIA - rencontre de réflexion, échanges et d'affaires - qui rassemble chaque année la communauté francophone des logiciels ludo-

éducatifs (Ressources sur CD-ROM et ressources en ligne) (Editeurs, chercheurs, studios, utilisateurs professionnels,..) [3]

IV- La suite libre GCompris

Distribué sous la licence GNU GPL, GCompris est un logiciel ludique et éducatif destiné aux enfants de 2 à 10 ans. Il propose plus de 100 activités ludiques et pédagogiques telles que de la géographie, l'explication du cycle de l'eau, de l'électricité, des exercices de mathématiques, des casse-têtes, des puzzles, ainsi que la découverte de l'ordinateur. C'est un logiciel libre créé par Bruno Coudoin en 2000, écrit en python (dans sa version Gtk+). Il fonctionne sur les systèmes d'exploitation Windows, Linux, BSD et Mac OS X. Il est inclus dans plusieurs distributions Linux. [4]

Il a une très grande popularité auprès des écoles. En effet, dans la version Gtk+, un module d'administration permet à l'enseignant de contrôler les activités, la langue, etc. Il a reçu le prix éducatif lors des Trophées du Libre qui ont été décernés à Soissons le 24 mai 2003.

Sa pédagogie est efficace grâce à son interface simple, attractive et interactive. Des explications sont données oralement pour les enfants qui ne savent pas lire, et d'autres sont écrites dans un rectangle sur l'écran d'accueil de chaque activité. GCompris permet d'entraîner les tout petits aux puzzles, à l'utilisation de l'ordinateur, la reconnaissance de formes, la lecture et les maths ainsi que la mémoire auditive et visuelle.

Il a été traduit en 50 langues différentes, ce qui étend son utilisation au-delà de toutes les frontières géographiques et culturelles.

Pour pousser les écoles utilisant des systèmes d'exploitation propriétaires à venir sur GNU/Linux, GCompris fut d'abord proposé uniquement sur cette plateforme, gratuitement, en offrant plus de 144 activités. Il est incorporé à plusieurs distributions Linux telles qu'Edubuntu. Cette mission n'a pas été un succès alors c'est l'équipe de GCompris qui est allée vers Windows et MacOSX en proposant une version payante mais toujours libre.

GCompris est incorporé dans l'ordinateur XO-1 destiné aux enfants du programme One Laptop per Child. Les utilisateurs de cet ordinateur peuvent ainsi profiter des modules géographie, jeu d'échecs, sudoku, tangram, poids et mesure, et découverte de l'électricité de GCompris. [5]

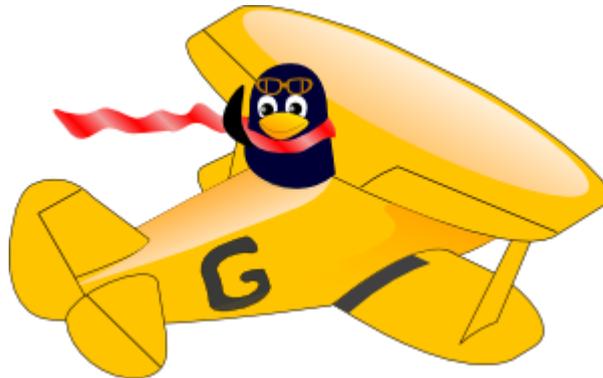


FIGURE I.1 : Logo du logiciel libre GCompris

V-Open Source et Logiciel libre

“ « *Logiciel libre* » [free software] désigne des logiciels qui respectent la liberté des utilisateurs. En gros, cela veut dire que **les utilisateurs ont la liberté d'exécuter, copier, distribuer, étudier, modifier et améliorer ces logiciels**. Ainsi, « logiciel libre » fait référence à la liberté, pas au prix (pour comprendre ce concept, vous devez penser à « liberté d'expression », pas à « entrée libre »). Pour bien montrer qu'il ne s'agit pas de gratuité, nous utilisons quelquefois en anglais l'expression *libre software*.

Nous faisons campagne pour ces libertés parce que chacun les mérite. Avec ces libertés, les utilisateurs (à la fois individuellement et collectivement) contrôlent le programme et ce qu'il fait pour eux. Quand les utilisateurs ne contrôlent pas le programme, nous qualifions ce dernier de « non libre », ou « privé ». Ce programme non libre contrôle les utilisateurs et son développeur le contrôle. Le programme devient donc l'instrument d'un pouvoir injuste. » [6]

Le mouvement de l'Open Source est une variante de celui du logiciel libre mais il n'a pas exactement la même philosophie que celle de la Free Software Foundation. Il donne plus d'importance à l'aspect technique qu'au respect des libertés de l'utilisateur.

Un programme est un logiciel libre si vous, en tant qu'utilisateur de ce programme, avez les quatre libertés essentielles :

- Liberté 0 : exécuter le programme comme vous le voulez, pour n'importe quel usage.
- Liberté 1 : étudier le fonctionnement du programme, et le modifier pour qu'il effectue vos tâches informatiques comme vous le souhaitez ; l'accès au code source est une condition nécessaire.
- Liberté 2 : redistribuer des copies, donc aider votre voisin ;
- Liberté 3 : distribuer aux autres des copies de vos versions modifiées, en faisant cela, vous donnez à toute la communauté une possibilité de profiter de vos changements ; l'accès au code source est une condition nécessaire.

Un programme est un logiciel libre s'il donne toutes ces libertés aux utilisateurs de manière adéquate. Dans le cas contraire, il est non libre. Bien que nous puissions faire une distinction entre différents schémas de distributions non libres, en quantifiant ce qui leur manque pour être libres, nous les considérons tous comme équivalents dans leur manque d'éthique.

VI- Conclusion

En conclusion de ce chapitre, nous dirons qu'un logiciel libre est le plus apte à se développer et à combler les besoins de l'utilisateur de par la communauté nombreuse qui contribue chaque jour au bon entretien du projet. Il permet aussi d'avoir des développeurs de haute qualité grâce au partage des codes sources.

GCompris a respecté cette philosophie du logiciel libre, ce qui fait de lui un outil d'apprentissage évolutif, répondant à la qualité et aux besoins grandissants des écoles. La communauté lui offre la capacité de perdurer dans le temps tout en se mettant à jour.

Chapitre 2 : Cycle de vie d'un logiciel libre

I-Développement communautaire

“Great things are not done by impulse, but by a series of small things brought together.”

—*Vincent van Gogh*

1.1 -Définition :

Le développement communautaire, comme son nom l’indique, consiste à développer des logiciels informatiques en distribuant les différentes tâches aux contributeurs. Ces derniers participent à la conception, programmation et l’amélioration continue du logiciel. Leur rôle peut prendre plusieurs dimensions : correction des bugs, extension de fonctionnalités, adaptations spécifiques à la localité comme le choix des couleurs et la traduction linguistique.

II- Cycle de vie d’un logiciel libre

Le cycle de vie d’un logiciel libre est aussi libre proprement dit ; en effet, dès qu’un bug apparaît une partie des contributeurs peuvent s’en occuper, tandis que l’autre partie continue toujours d’avancer. Ceci est possible par le biais des outils de gestion et de communication suivants : Mailing lists, canaux IRC, logiciels de gestion de versions.

Ce type de développement est caractérisé par l’émission fréquente de sous-versions vu le nombre important de corrections, soit un large rendement de travail.

III- Les outils de gestion et de communication

3.1 -GitHub :

Sous le slogan *“Build software better, together”* est un service web d’hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git (logiciel de gestion de versions distribué). Le site a atteint 10 millions de dépôts en 2013, cela montre la grandeur de la communauté libre mondiale. Il offre aussi un service de *pastebin* pour le partage des codes sources. Il permet de collaborer n’importe où et avec n’importe qui.

Le créateur de Git est Linus Torvalds, plus connu pour avoir engendré plusieurs années plus tôt le noyau Linux. Il écrivit ce logiciel en 2005 pour améliorer la collaboration des contributeurs au projet Linux et son succès fut immédiat.

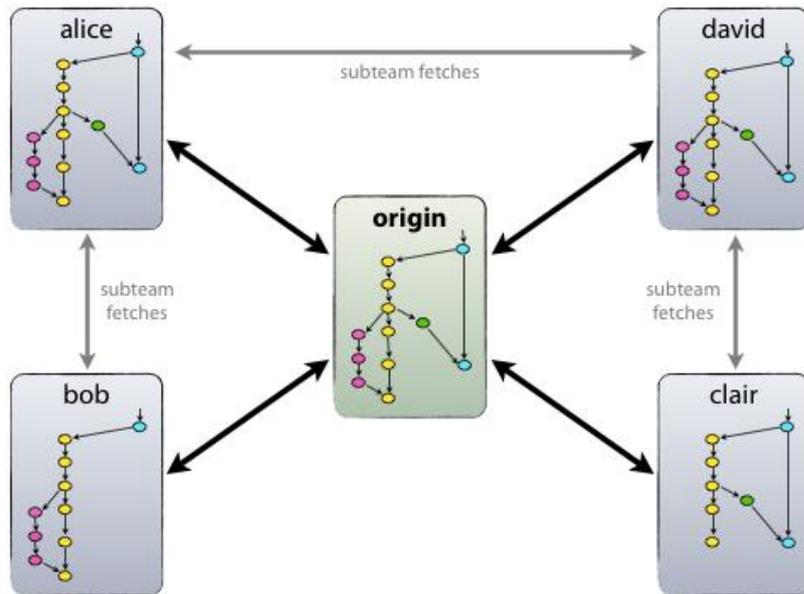


FIGURE II.1 : Schéma de l'architecture Git

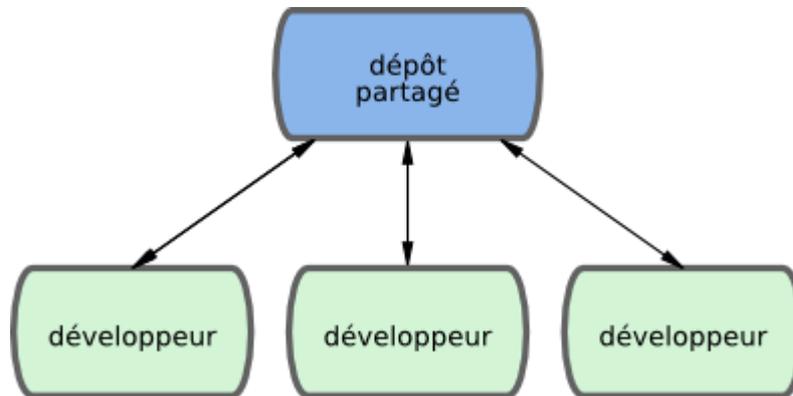


FIGURE II.2 : Schéma Git

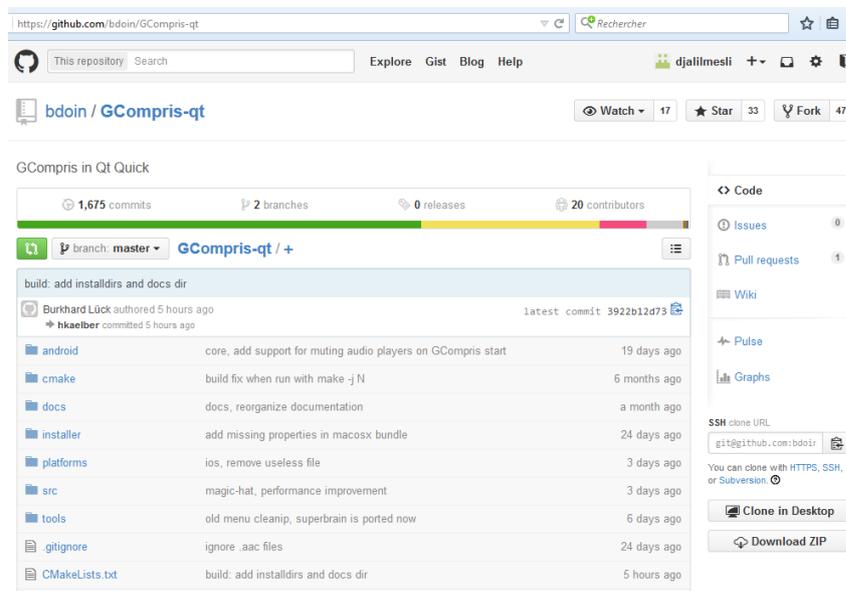


FIGURE II.3 : Capture d'écran du dépôt partagé GCompris

3.2 -Mailing list

La *mailing list* (*liste de diffusion* en français) est un système d'utilisation particulier du courrier électronique, un ancêtre des forums. Ce système consiste à diffuser des codes sources, des questions, des solutions à propos d'un sujet. Par exemple, le principal moyen de communication entre les développeurs du noyau Linux est la Linux Kernel Mailing List (LKML).

3.3 -Canal IRC (Internet relay Chat)

Discussion relayée par Internet, conçu en 1988, est un protocole de communication qui offre la possibilité de communiquer en messagerie instantanée via des canaux. Ceci est structuré par des adresses à des serveurs, des groupes de discussions comme le Canal IRC GCompris : #gcompris sur le serveur freenode.



FIGURE II.4 : Capture d'écran du Canal IRC GCompris

IV- Développement communautaire sur GCompris durant la migration

4.1 -Introduction

Le développement communautaire est sans doute le plus efficace et fiable pour avoir un produit qui perdure dans le temps. En effet, des centaines de développeurs éparpillés aux quatre coins du globe travaillant en temps réel sur un même projet est une force indéniable. Le projet « GCompris in Qt Quick » suit le même schéma. Ceci lui permettra de s'imposer sur la scène des logiciels éducatifs pour les années à venir, et de répondre aux exigences des utilisateurs.

4.2 -Intérêt du portage

La migration vers Qt Quick est sans doute la meilleure des opportunités pour GCompris de survivre à l'avancée des technologies. En effet, ce framework libre donne une dimension plus grande et plus intéressante vu sa portabilité extrême. Grâce à cette réécriture du code, GCompris gagne en fluidité, portabilité et stabilité ainsi qu'en capacité de propagation exponentiellement accrue de par sa disponibilité sur tablettes et smartphones. A la fin de cette réécriture, on découvrira un GCompris utilisant les dernières technologies informatiques, un produit fini parfaitement représentatif de l'éthique du logiciel libre.

4.3 -Les outils :

- **Langages utilisés :**

- **QML :** Qt Meta Language or Qt Modeling Language est un langage déclaratif hiérarchique scripté de haut niveau servant à la conception d'interface utilisateur (graphique). Basé sur du JavaScript, il est une partie de Qt Quick, le kit de création d'interface utilisateur développé par Nokia dans le framework Qt. QML est principalement utilisé pour les applications mobiles où la saisie tactile, les animations fluides (60 FPS) et l'expérience utilisateur sont cruciales. Les animations en QML sont effectuées en animant les propriétés des objets. Les propriétés de type réel, entier, couleur, rectangle, point, la taille et Vector3D peuvent tous être animés. QML prend en charge trois principales formes d'animation : animation de la propriété de base, animation des transitions, animation des comportements de propriété. La forme la plus simple de l'animation est un PropertyAnimation, qui peut animer tous les types. Une animation de la propriété peut être spécifiée comme une source de valeur à l'aide de l'animation sur la syntaxe de la propriété. Ceci est particulièrement utile pour la répétition des animations. [7]

En plus de tout cela, il s'accorde parfaitement avec du JavaScript en faisant des « include fichier.js » ceci permettra d'appeler des fonctions JavaScript quand on clique sur un objet par exemple. Pour terminer, le QML a le JavaScript V8 engine comme moteur.



FIGURE II.5 : Logo du langage QML

- **JavaScript** : Créé en 1995 par Brendan Eich pour le compte de la Netscape Communications Corporation, c'est un langage orienté objet dynamique prototypé. Sa syntaxe simple fait de lui un langage de choix pour tout type d'utilisation, que ce soit pour dynamiser une page web, implémenter une structure de données ou pour définir des actions via les signaux et slot d'un QML. Son succès est fortement lié avec le Java et les navigateurs web. [8]



FIGURE II.6 : Logo du langage JavaScript

- **Qt Quick** : est un framework développé par Digia, faisant partie de la bibliothèque Qt, et a la particularité d'être libre. Il permet principalement de créer des interfaces utilisateurs personnalisables et dynamiques avec des effets de transition fluides de manière déclarative. Qt Quick est idéal pour une conception orientée vers les smartphones vu sa performance et celles des smartphones qui sont généralement réduites. Son appartenance à Qt le rend multiplateformes et portable ; avec le même code source on pourra avoir une version pour chaque système d'exploitation. [9]

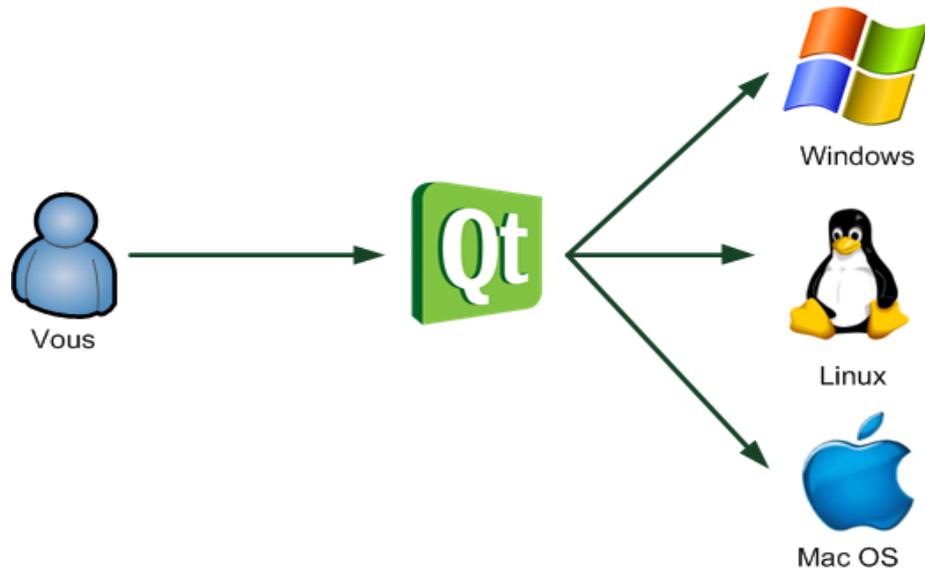


FIGURE II.7 : Schéma d'utilisation de Qt

- **Qt Creator** : Environnement de développement intégré multiplateformes, fruit du projet « Qt Project » fait partie du framework Qt. Il intègre plusieurs composants : un débogueur, outil de création d'interface graphique, des outils pour la publication de code sur Git, une messagerie IRC, un éditeur de texte intelligent permettant l'autocomplétion ainsi que la coloration syntaxique et la documentation Qt. Il gère le C++, QML et JavaScript. Qt Creator offre aussi la possibilité d'intégrer le moteur de production « CMake ». [10]

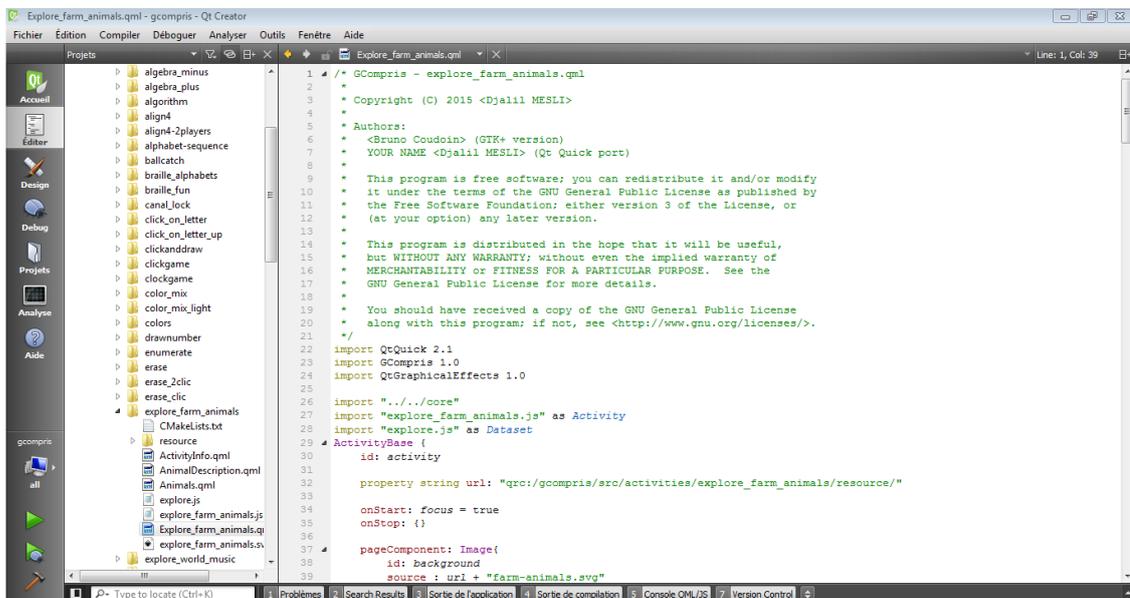


FIGURE II.8 : Capture d'écran de l'IDE Qt Creator

- **CMake** : « cross platform make » est un moteur de production multiplateforme permettant la génération de fichiers de construction standard : makefiles sous Unix, et fichiers de projet Visual Studio sous Windows. Ceci est contrôlé par un fichier de configuration nommé « CMakeLists.txt ». CMake a pour but de simplifier le processus de compilation et d’être utilisé tout en continuant à travailler sur son IDE préféré. [11]

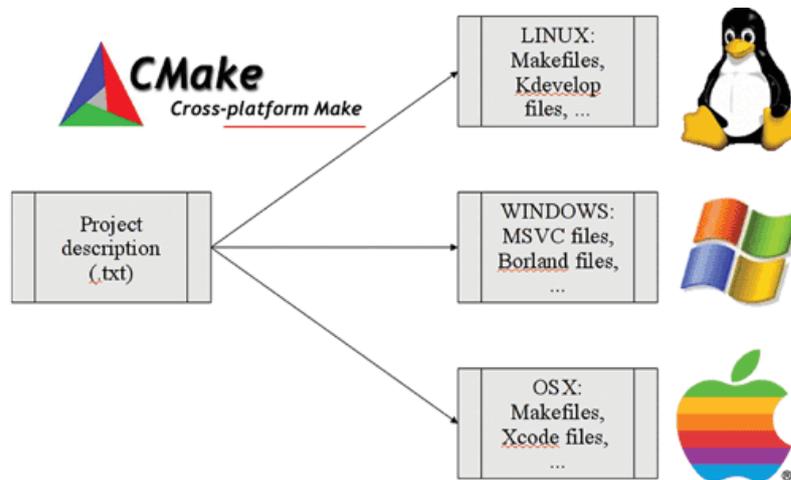


FIGURE II.9 : Schéma explicatif du moteur CMake

- **SVG** : « Scalable Vector Graphics » est un format de données fait pour décrire des éléments vectoriels. SVG est un **format d'image léger** lorsqu'il s'agit de représenter des formes simples, car seules les informations décrivant ces formes sont stockées (coordonnées, couleurs, effets) contrairement aux images bitmap (JPG, PNG, GIF) qui doivent mémoriser le contenu pixel par pixel. Ce principe rend les images SVG étirables **sans perte de qualité**. Il est également une alternative à Flash pour présenter des données qui doivent être dynamiques. [12]

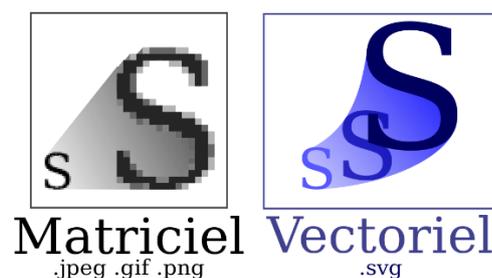


FIGURE II.10 : Figure montrant l'intérêt des SVG

V- Conclusion

La migration de GCompris vers Qt Quick est primordiale pour son développement à long terme. Pour cela, le dispositif du développement communautaire est un atout considérable. Ce dispositif a une éthique, une idéologie ainsi qu'une méthodologie de travail. La migration contribue par elle-même à l'amélioration des développeurs du projet et en recrute de nouveaux. Ce fut le cas pour moi, j'étais un débutant en programmation et ce travail m'a confronté à des informaticiens de haut niveau. J'ai été dans l'obligation de progresser pour pouvoir continuer mon travail. Le partage des codes sources a fait de moi un développeur GCompris.

Chapitre 3 :
**l'activité « Faire connaissance avec
les animaux de la ferme »**

I- Introduction

L'activité « Faire connaissance avec les animaux de la ferme » est un exercice interactif intéressant pour le développement des capacités intellectuelles des enfants. En effet, le fait d'explorer les sons et les descriptions des animaux est un exercice de lecture et de mémorisation vu qu'aux niveaux 2 et 3 les questions porteront sur ce que l'enfant a pu visualiser au niveau 1.

II- Description et but de l'activité

Comme décrit précédemment, l'activité a pour but de développer les sens et les capacités intellectuelles des enfants afin de faciliter leur apprentissage. L'enfant aura le loisir de cliquer sur les différents animaux, d'écouter leurs cris, regarder une photo de chaque animal ainsi que de lire une brève description de ce dernier. Les figures suivantes représentent le niveau 1 :



FIGURE III.1 : Capture d'écran niveau 1 de l'activité version Gtk+



FIGURE III.2 : Capture d'écran niveau 1 de l'activité version Gtk+

Dans le second niveau, l'enfant sera mis en face d'un exercice de mémoire auditive portant sur ce qu'il a pu écouter précédemment. Un son est émis, l'enfant doit cliquer sur l'animal correspondant, et s'il le trouve, un autre son est émis et ainsi de suite jusqu'à ce qu'il réponde à toutes les questions. La figure suivante représente le niveau 2 :



FIGURE III.3 : Capture d'écran niveau 2 de l'activité version Gtk+

Le niveau 3 consiste à lire des questions affichées sur l'arrière-plan, c'est un exercice de mémoire textuelle. En effet, ces questions ont pour source les descriptions du niveau 1. L'enfant doit lire la question et cliquer sur l'animal qui correspond à cette description. S'il répond correctement, la question suivante est enclenchée sinon la même question est répétée. La figure suivante représente le niveau 3 :



FIGURE III.4 : Capture d'écran niveau 3 de l'activité version Gtk+

III-Phase de développement

3.1 -Introduction

Dans ce projet, le cahier des charges provient de l'interface des différents niveaux de l'activité dans sa version Gtk+. Mon travail a consisté à analyser et à étudier les interactions de l'utilisateur avec l'interface de l'ancienne version pour effectuer la réécriture de cette activité.

C'est une vraie réécriture et non pas une traduction du code Python vu que le QML et Qt Quick fonctionnent différemment. La lecture du code Python n'a donc pas été nécessaire.

3.2 -Marquage et réservation

Pour rejoindre la communauté GCompris dans le projet KDE, il faut tout d'abord s'inscrire sur la Mailing list en allant sur le site « <https://mail.kde.org/mailman/listinfo/gcompris-devel> » puis se présenter auprès des développeurs GCompris en envoyant son premier mail à la Mailing list à l'adresse mail suivante : « gcompris-devel-request@kde.org ». Après cela, contacter le fondateur du projet M.Bruno Coudoin et lui demander de prendre une ou plusieurs activités.

Après l'accord du fondateur, il faudra s'inscrire sur le wiki de GCompris pour marquer l'activité libre souhaitée. Il reste à présent à récupérer les données et commencer à coder son module « activité ».

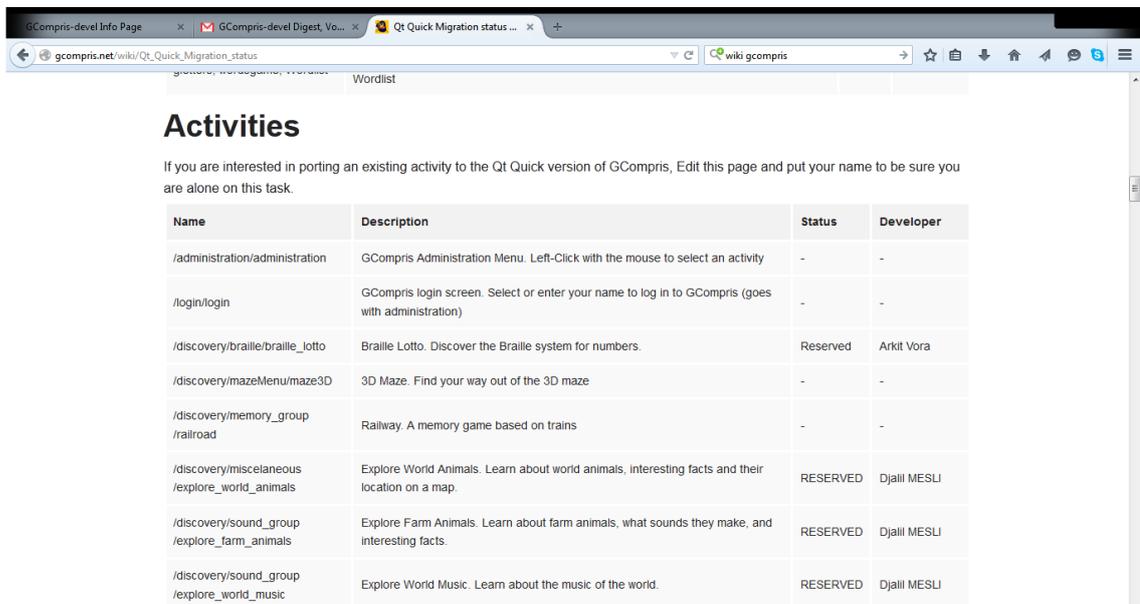


FIGURE III.5 : Capture d'écran du statut de la migration sur le wiki GCompris

3.3 -Récupération des données

Pour pouvoir commencer son travail, il faut télécharger le projet « GCompris in Qt Quick » du GitHub de Bruno Coudoin à l'adresse suivante : « <https://github.com/bdoin/GCompris-qt> » puis de faire un « fork » du répertoire pour avoir une copie sur son compte. Ce procédé est essentiel pour développer en communauté. En effet, chaque modification sur le répertoire personnel du projet est uploadée vers le serveur. Une fois l'activité achevée, on pourra faire « un pull request » pour intégrer son travail au compte principal bdoin/gcompris-qt.

À la sortie de la prochaine version, l'activité sera présente sur PC et tablettes (AppStore et Google Play).

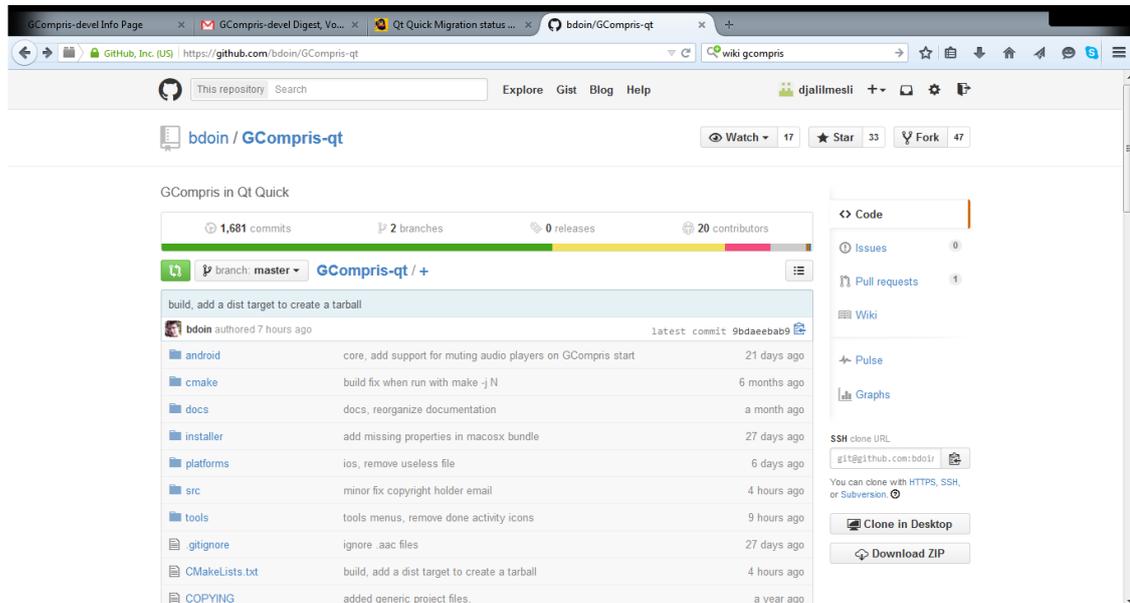


FIGURE III.6 : Capture d'écran du répertoire principal GitHub de GCompris

Les données récupérées sont exactement les codes sources des activités, code source du programme GCompris, les fichiers de configuration CMake...

Les données propres à l'activité à savoir les sons, images, textes et SVG sont récupérables du dossier « shared/board/le nom de l'activité » de la version Gtk+.

3.4 -Développement et implémentation

La structure choisie pour mon activité a été le fruit d'une longue réflexion et plusieurs essais ratés jusqu'à arriver à la solution. L'idée est de concevoir une classe « Animals » dans un fichier nommé « Animals.qml » aux propriétés voulues telles que les coordonnées, la source d'image et de son..., qu'oninstanciera tant de fois qu'on veut. Les signaux et slots sont décrits dans ce fichier, une liste d'appels aux fonctions JavaScript est exécutée à chaque clic sur un objet (animal).

Le fichier « Explore_farm_animals.qml » représente le fichier principal du module « activité », les objets tels que les boutons et les images sont instanciés dedans, vu que le QML est un langage hiérarchique. Les figures suivantes représentent le code de la classe « Animals » et son instanciation dans le fichier principal « Explore_farm_animals.qml » :

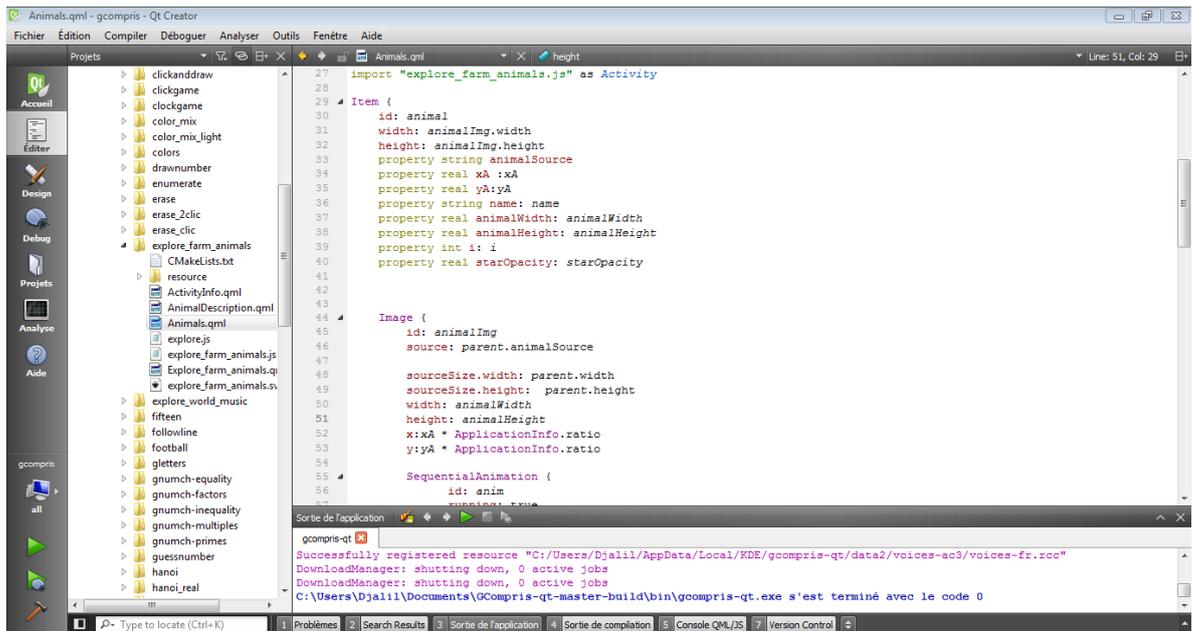


FIGURE III.7 : Capture d'écran du fichier « Animals.qml »

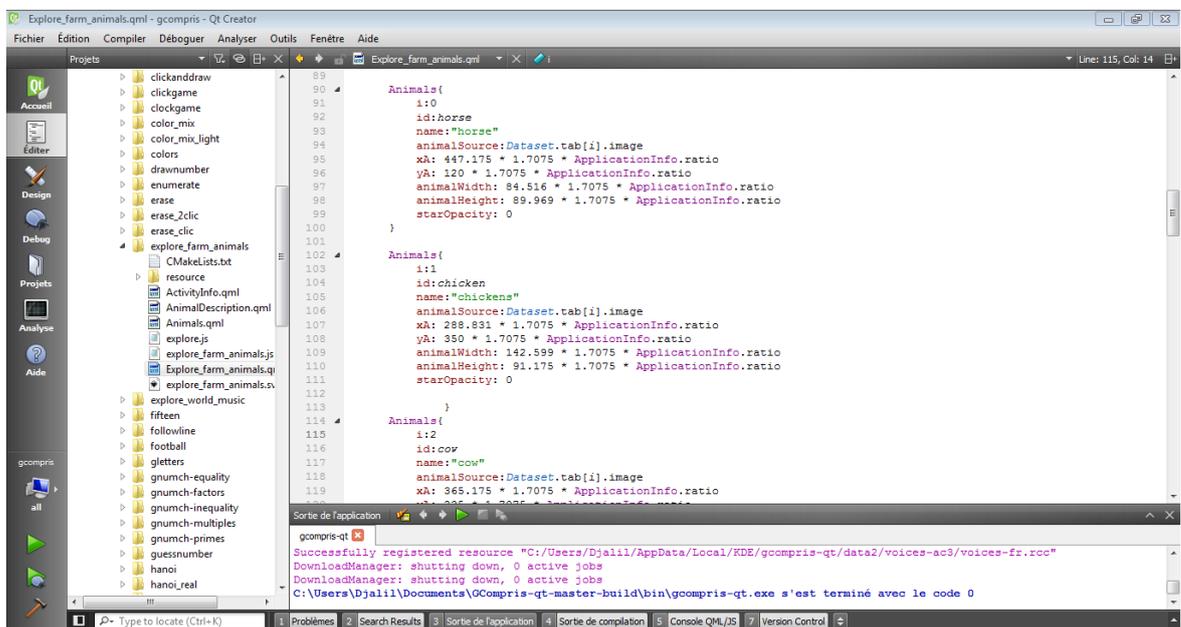


FIGURE III.8 : Capture d'écran montrant l'instanciation d'objet Animals dans « Explore_farm_animals.qml »

La deuxième problématique était de faire en sorte que l'activité soit modifiable ou réutilisable juste en modifiant un fichier de configuration de l'activité. Cela a été possible en optant pour un tableau de tableaux implémenté en JavaScript comme structure de données. Chaque case de ce tableau pointe vers un tableau où chacune de ces cases est un chemin vers un son, une image (ou SVG) ou un texte. En conclusion, pour créer une activité du même genre, il faudra tout simplement implémenter une autre

structure de données. La figure suivante représente le fichier « explore.js », la structure de données citée précédemment :

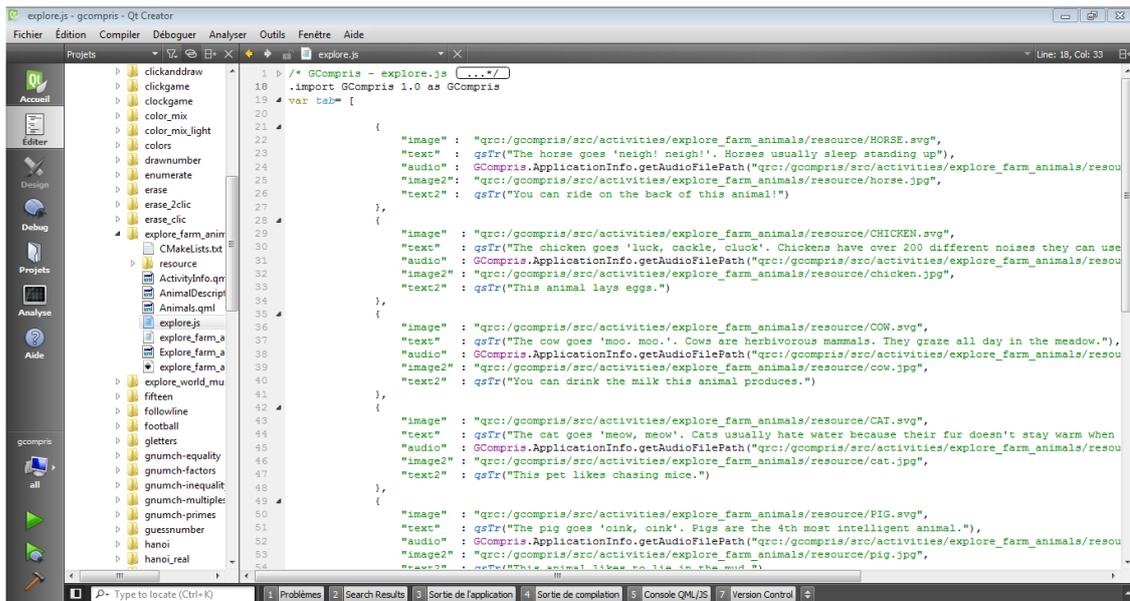


FIGURE III.9 : Capture d'écran du fichier « explore.js »

Le fichier « explore_farm_animals.js » est le moteur du module. En effet, il contient toutes les fonctions et variables utiles à l'activité telles que les actions, la logique et les procédures. Il décrit les fonctions exécutées grâce aux signaux et slots du QML ce qui fait de lui le moteur d'action de ce dernier. La figure suivante donne un aperçu du fichier « explore_farm_animals.js » :

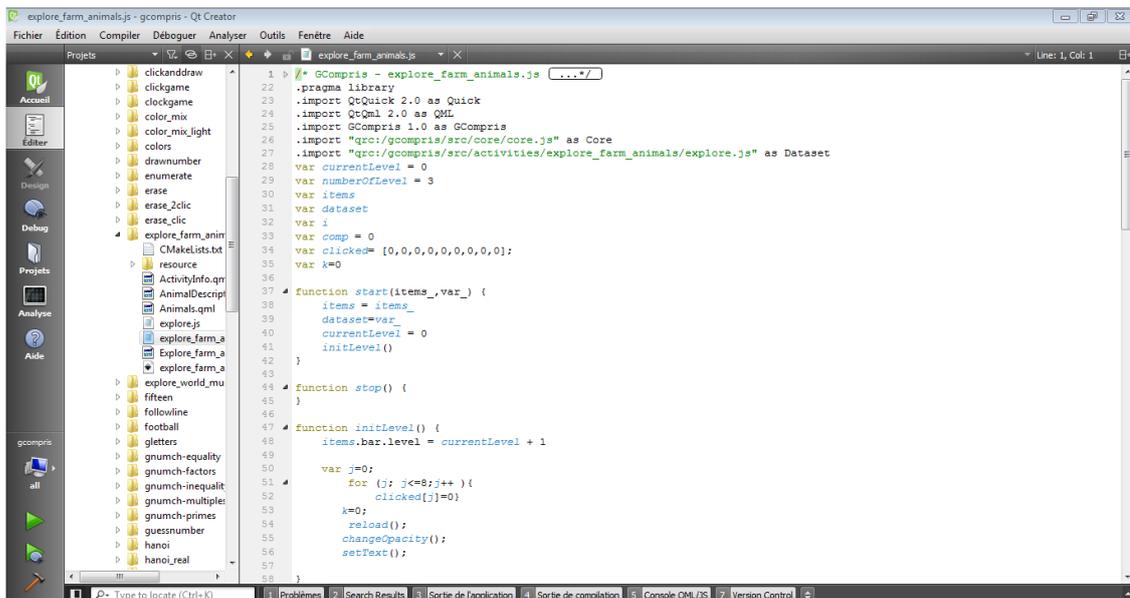


FIGURE III.10 : Capture d'écran du fichier « explore_farm_animals.js »

Le fichier décrivant la scène de la description du niveau 1 est le fichier « AnimalDescription.qml ». Ce fichier décrit les coordonnées de l'image, le texte à afficher et diverses actions comme la destruction de cet objet après clic sur le bouton « Back to Homepage » ou les animations à effectuer. Il affiche l'image et le texte depuis le lien de la structure de données ; exemple « animalSource : Dataset.tab[i].image ». Cette façon de faire rend le fichier réutilisable pour chaque animal.

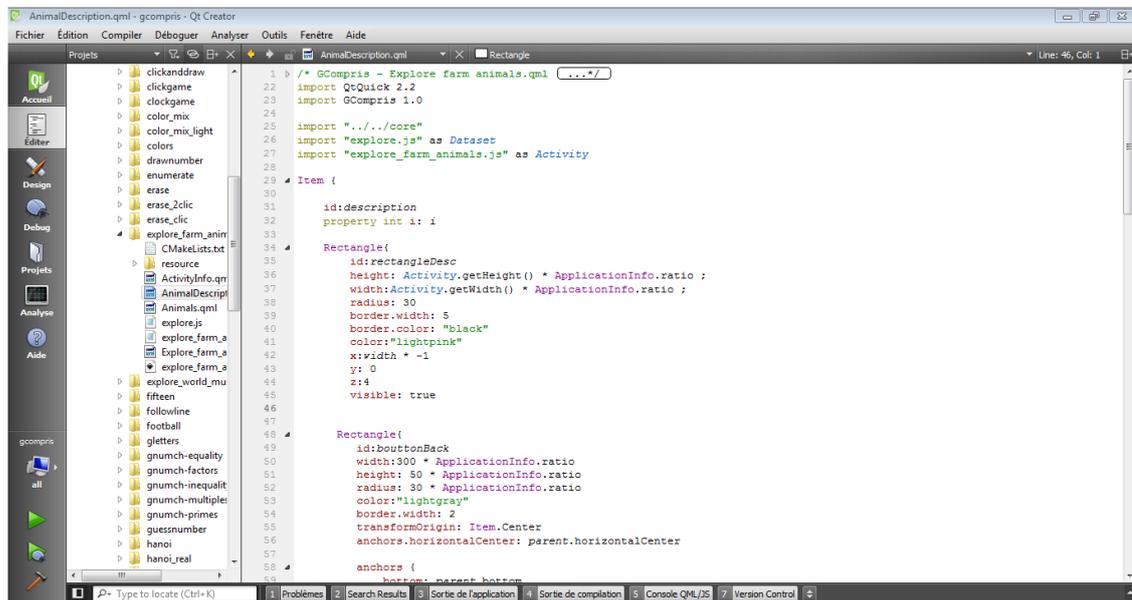


FIGURE III.11 : Capture d'écran du fichier « AnimalDescription.qml »

Enfin, pour terminer, les informations comme l'icône, auteur, manuel, but de l'activité et catégories: sont décrites dans le fichier « ActivityInfo.qml ».

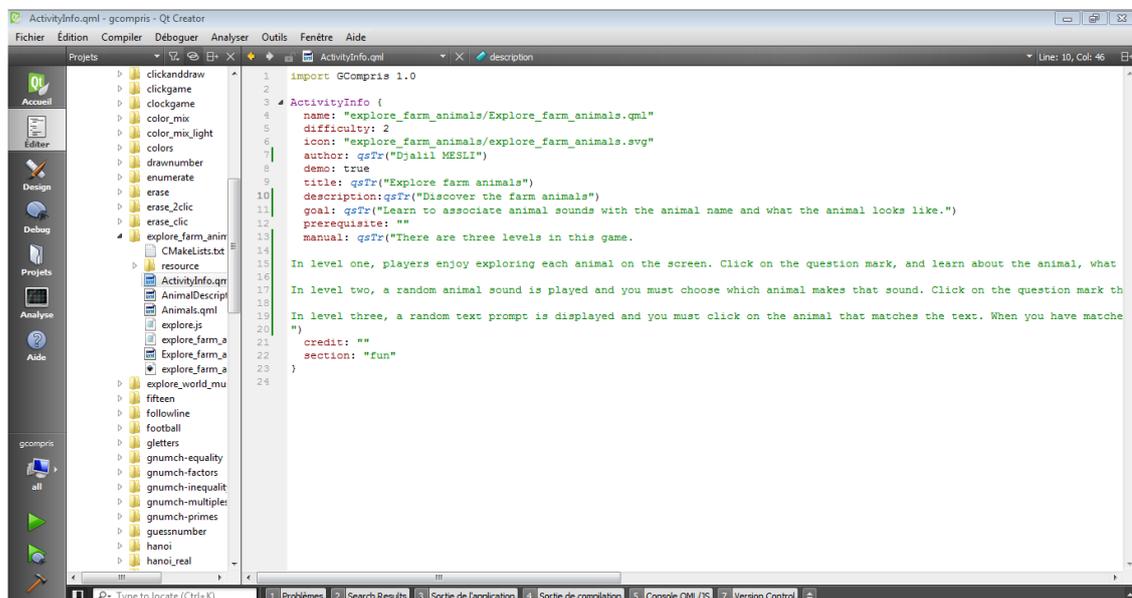


FIGURE III.12 : Capture d'écran du fichier « ActivityInfo.qml »

IV- L'activité finie sous Qt Quick

Grace aux SVG l'activité ne perd jamais en qualité d'image quelle que soit la résolution et la grandeur de l'écran. Cette avancée technologique propulse GCompris dans le rang des logiciels de qualité. Voici l'aperçu des trois niveaux de l'activité « les animaux de la ferme »

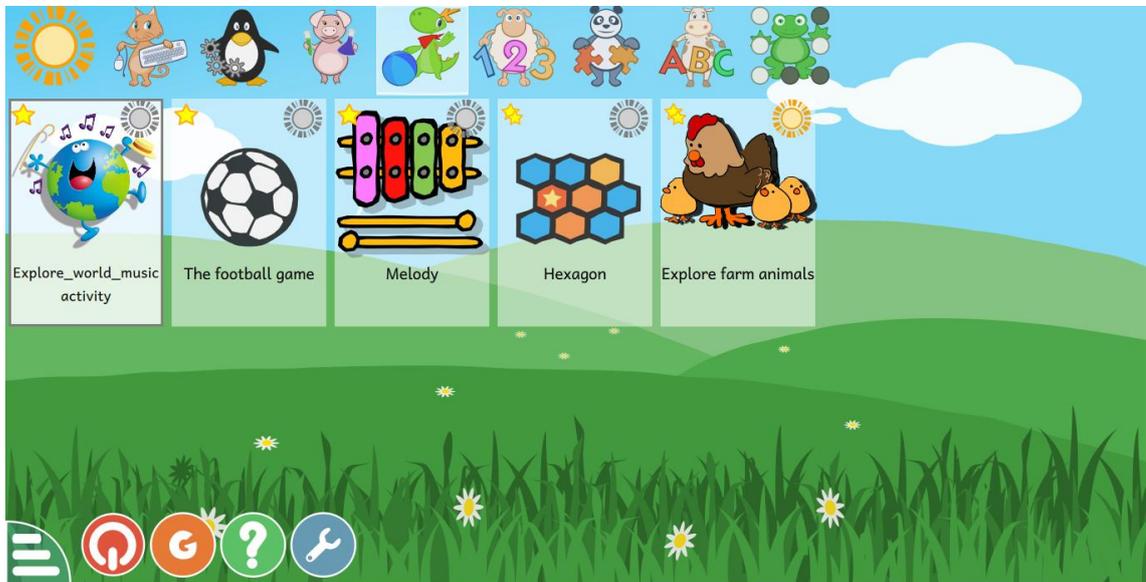


FIGURE III.13 : Ecran d'accueil de GCompris version Qt Quick



FIGURE III.14 : Niveau 1 de l'activité version Qt Quick

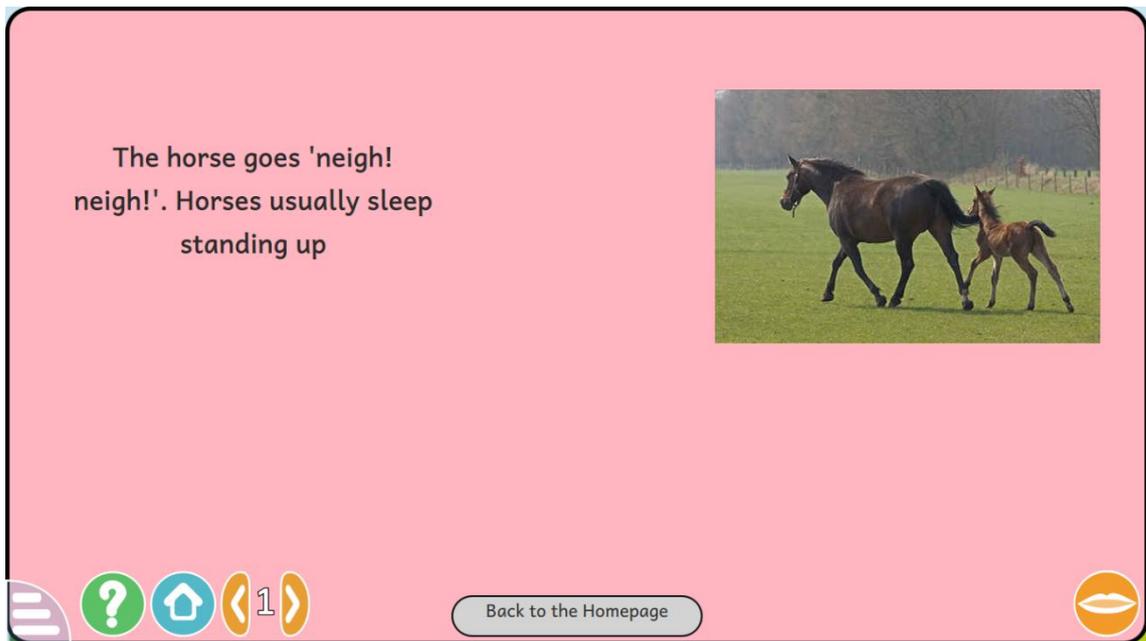


FIGURE III.15 : Niveau 1 de l'activité version Qt Quick



FIGURE III.16 : Niveau 2 de l'activité version Qt Quick



FIGURE III.17 : Niveau 3 de l'activité version Qt Quick

V- Conclusion

Le développement de cette activité est passée par plusieurs étapes décrites ci-dessus ; les résultats de la réécriture de cette activité sous Qt Quick sont étonnants vu la souplesse et la portabilité de ce dernier. Le QML et le JavaScript présentent ici une complémentarité exemplaire, l'activité en est l'exemple vivant.

Conclusion générale

Ce projet de fin d'études a été une occasion pour moi d'acquérir des connaissances et d'approfondir mes acquis antérieurs. Son principal déclic était la confrontation directe à un groupe de travail des plus avancés au monde. En effet, une telle opportunité m'a poussé à me surpasser pour pouvoir suivre le rythme de ces développeurs expérimentés dans le développement communautaire.

GCompris, logiciel ludo-éducatif libre et soutenu par la Free Software Foundation et la communauté KDE, m'a permis d'étudier un code source de qualité, fiable, efficace et réutilisable. De ce fait, l'apprentissage du langage QML et JavaScript a été accompagné d'exemples parfaits qui sont les modules « activité » de GCompris.

Le fruit de ce travail à savoir l'activité « Faire connaissance avec les animaux de la ferme » a été approuvée par la communauté, et sera présente lors de la prochaine version de GCompris en Qt Quick, et de là sur les plateformes Windows, MacOSX, Linux et Android.

Liste des figures

FIGURE I.1 : Logo du logiciel libre GCompris

FIGURE II.1 : Schéma de l'architecture Git

FIGURE II.2 : Schéma Git

FIGURE II.3 : Capture d'écran du dépôt partagé GCompris

FIGURE II.4 : Capture d'écran du Canal IRC GCompris

FIGURE II.5 : Logo du langage QML

FIGURE II.6 : Logo du langage JavaScript

FIGURE II.7 : Schéma d'utilisation de Qt

FIGURE II.8 : Capture d'écran de l'IDE Qt Creator

FIGURE II.9 : Schéma explicatif du moteur CMake

FIGURE II.10 : Figure montrant l'intérêt des SVG

FIGURE III.1 : Capture d'écran niveau 1 de l'activité version Gtk+

FIGURE III.2 : Capture d'écran niveau 1 de l'activité version Gtk+

FIGURE III.3 : Capture d'écran niveau 2 de l'activité version Gtk+

FIGURE III.4 : Capture d'écran niveau 3 de l'activité version Gtk+

FIGURE III.5 : Capture d'écran du statut de la migration sur le wiki GCompris

FIGURE III.6 : Capture d'écran du répertoire principale GitHub de GCompris

FIGURE III.7 : Capture d'écran du fichier « Animals.qml »

FIGURE III.8 : Capture d'écran montrant l'instanciation d'objet Animals dans « Explore_farm_animals.qml »

FIGURE III.9 : Capture d'écran du fichier « explore.js »

FIGURE III.10 : Capture d'écran du fichier « explore_farm_animals.js »

FIGURE III.11 : Capture d'écran du fichier « AnimalDescription.qml »

FIGURE III.12 : Capture d'écran du fichier « ActivityInfo.qml »

FIGURE III.13 : Écran d'Accueil GCompris version Qt Quick

FIGURE III.14 : Niveau 1 de l'activité version Qt Quick

FIGURE III.15 : Niveau 1 de l'activité version Qt Quick

FIGURE III.16 : Niveau 2 de l'activité version Qt Quick

FIGURE III.17 : Niveau 3 de l'activité version Qt Quick

Références

- ❖ [1] Eric S. Raymond : « La Cathédrale et le Bazar », essai de 1997, l'une des références de la philosophie *open source*. <http://www.linux-france.org/article/these/cathedrale-bazar/cathedrale-bazar.html>
- ❖ [2] http://fr.wikipedia.org/wiki/Histoire_du_jeu_vidéo
- ❖ [3] http://fr.wikipedia.org/wiki/Logiciel_ludo-éducatif
- ❖ [4] <http://gcompris.net/wiki/Manuel>
- ❖ [5] <http://fr.wikipedia.org/wiki/GCompris>
- ❖ [6] <https://www.gnu.org/philosophy/free-sw.fr.html>
- ❖ [7] <http://doc.qt.io/qt-5/qmlapplications.html>
- ❖ [8] <http://doc.qt.io/qt-5/qtqml-javascript-expressions.html>
- ❖ [9] <http://doc.qt.io/qt-5/qtquick-index.html>
- ❖ [10] <https://wiki.qt.io/Category:Tools::QtCreator>
- ❖ [11] <http://www.cmake.org/documentation/>
- ❖ [12] <http://www.w3.org/Graphics/SVG/>

Résumé

Le travail présenté dans ce mémoire de fin d'études pour obtenir la licence en informatique entre dans le cadre de la migration du logiciel éducatif libre GCompris auparavant basé sur la technologie Gtk+ vers le framework QtQuick.

Cette migration est motivée par le portage de cette application sur des appareils mobiles (tablettes notamment) qui reste problématique avec le toolkit Gtk+.

Le logiciel GCompris vise à développer les capacités intellectuelles des enfants afin de faciliter leur apprentissage grâce à un ensemble de modules appelés « activités ».

La migration, en cours depuis plus d'un an, se fait progressivement par la réécriture de chacune de ces activités. Mon travail a consisté à réécrire celle appelée « Faire connaissance avec les animaux de la ferme » qui est un jeu de mémorisation auditive et visuelle développant l'intelligence des enfants.

Abstract

The work presented in this thesis graduation to obtain the computer sciences degree deals with the migration of free educational software GCompris previously based on Gtk + technology to QtQuick framework.

This migration is motivated by porting this application on mobile devices (tablets particularly) which remains problematic with Gtk + toolkit.

The software GCompris is to develop the intellectual capacities of children to facilitate their learning through a set of modules called "activities".

Migration, ongoing for more than a year, is done gradually by rewriting each of these activities. My task was to rewrite one of them called "Explore the farm animals" which is an auditory and visual memory game developing children's intelligence.

ملخص

العمل المقدم في هذا المشروع للحصول على شهادة الليسنس في المعلوماتية يدخل في إطار هجرة البرنامج الحر التعليمي «GCompris» من التكنولوجيا Gtk+ إلى Qt Quick.

هدف هذه الهجرة إمكانية تشغيل GCompris في الهواتف الذكية واللوحات الذي هو غير ممكن بالتكنولوجيا Gtk+.

البرنامج يهدف الي تطوير القدرات الفكرية للأطفال لتسهيل التعليم من خلال مجموعة من الوحدات تسمى "الأنشطة".

الهجرة مستمرة منذ أكثر من سنة، ويتم ذلك تدريجيا بإعادة كتابة كل من هذه الأنشطة. كانت وظيفتي إعادة كتابة واحدة تسمى " استكشاف حيوانات المزرعة" التي هي لعبة الذاكرة السمعية والبصرية، تنمي ذكاء الأطفال.

Mots clés : GCompris, migration, Qt Quick, Gtk+, Faire connaissance avec les animaux de la ferme.