Faculté des Sciences
Aix*Marseille Université

Projet Génie Logiciel

LibreDragon

Réalisé par :

Pierre-Antoine Martrice
Charbel Fourel
Jules Cazes
Eliott Bourlon
Scholivet Manon
Sakakini Mouhssine Eddine

Encadré par :

Pablo Arrighi
Kevin Perrot

Cahier des charges initial

Sommaire

Introduction....3
Étude de l’existant....3
Étude des besoins....3
Étude de faisabilité....3
Description détaillée de l’ensemble des fonctions....4
1.MENU Principal....4
2.Interfaces du jeu....4
3.Déroulement du Jeu....4
Critères de réussite....6
Risques....6

Introduction

Dans le cadre du projet court de Génie Logiciel, l'équipe Tagada se penchera sur l'amélioration du projet MagiCarpe, dérivé libre du jeu DragonBox. L'équipe sera notamment amenée à ajouter des fonctionnalités et à revoir l'interface de l'application.

Étude de l’existant

Le projet se base sur le jeu DragonBox, un jeu éducatif qui a la vocation de faciliter l'apprentissage de l'algèbre de manière intuitive et interactive aux enfants. Pour se faire, l'écran est séparé en deux parties représentant les deux parties de l'équation, et le joueur manipule des constantes et des variables sous la forme d'icônes dans le but de la résoudre.

Le projet MagiCarpe est une adaptation libre de ce jeu. A ce stade du projet, il permet de résoudre des équations en utilisant addition, multiplication et division. Il manque cependant plusieurs éléments pour permettre à un enfant de comprendre le fonctionnement et le but du jeu, comme par exemple un tutoriel ou un menu principal.

Étude des besoins

Nos objectifs se divisent en plusieurs catégories :

- compléter l'adaptation libre du système inventé par DragonBox, en intégrant l'ensemble des règles d'arithmétique de base : addition, division, multiplication et soustraction ;

- ajouter un formalisme permettant d'effectuer des calculs logiques ;

- améliorer l’ergonomie en ajoutant des menus, des explications et la possibilité de choisir un niveau ;

- permettre à l'utilisateur de composer lui-même ses propres règles à partir des règles de base ;

- l'application doit également être utilisable indifféremment sur un ordinateur (Windows, Linux) ou sur une tablette (Android).

Étude de faisabilité

A hauteur de ce que nous avons put percevoir de l'existant et en prennant compte les besoins du client :

- L'ensemble du programme sera codé en Java (Langage naturel de l'equipe et langage utilisé dans les prototypes précédents)

- Il est envisagé d'aborder l'architecture du programme en MVC (flexibilité d'entretient, robustesse)

- Les Patrons de conception utilisés seront vraisemblablement les patrons observer, observable, strategy. (faciles à intégrer et quasiment indispensables en MVC)

- Le programme supportera l'import de données via des fichiers externes (XML ou JSON)

- Le programme supportera la sauvegarde de données utilisateur dans des fichiers externes (XML ou JSON)

- L'interface graphique sera vraisemblablement développée à l'aide de LibGDX, dans un soucis de rapidité et de portabillité. L'architecture du programme dépendra énormement de la souplesse de LibGDX, il se peut que le MVC soit abandonné si LibGDX ne permet pas sa mise en place

Description détaillée de l’ensemble des fonctions

1. MENU Principal

Le menu principal doit permettre d'accéder aux différents modes de jeu : Le mode algèbre, le mode logique et l'éditeur de règles.

Il donnera également accès à un menu d'options et au bouton pour mettre fin au programme.

2. Interfaces du jeu

-Interface de transition : Après le choix du mode jeu, une interface de transition montrera les différents exercices accomplis ou à accomplir, l'utilisateur pourra choisir le niveau de jeu à partir de là ou revenir au menu principal.

-Interface de jeu : En jeu nous aurons trois espaces principaux : deux espaces représentant deux ensembles pouvant contenir des expressions mathematiques, des variables ou des constantes ou le tout en meme temps (ces deux ensembles représentent les deux coté d'une équation et l'ensemble des éléments qui les composent peuvent être traduits à l'écran sous la forme de symboles naïfs ou sous une forme représentant directement l'objet en question. Par exemple x aura pour symbole la dragon box et la variable g pourra avoir pour forme un poisson). A ces deux espaces on ajoutera un espace pour les éléments que l'on pourra apporter a ces ensembles.

l'interface de jeu comprendra également un bouton pour revenir au choix des exercices , un bouton pour faire un retour d'un mouvement en arrière , un bouton d'indice et un bouton pour redémarrer le niveau.

-Interface de l'editeur de regles : L'editeur de regles se composera d'un champ pour nomer la fonction ou le nouveau corps de regles, et d'un ensemble de champs et combobox qui seront renseignés par des éléments constants de mathématique fondamentale. Sur la base de ces éléments on pourra reconstituer les fonctions mathématiques manquantes, et aussi créer des excercices apropriés. Cet élément est encore en cours d'étude au moment ou nous vous distribuons ce cahier des charges.

3. Déroulement du Jeu

Pour gagner une partie il faut que la dragon box soit seule de son coté (le dragon a l'intérieur est timide) et qu'il reste des éléments de l'autre coté pour qu'il puisse les manger une fois sorti de sa boite. Ces éléments peuvent prendre des formes variées :

- x sera toujours la dragon box mais pourra prendre sa forme littérale selon le niveau de difficulté de l'exercice.

- n'importe quelle variable poura étre traduite à l'écran soit sous une forme littérale (g, h, l, ...) soit sous une forme symbolique (un poisson, un légume, ...). Les valeurs négatives seront soit exprimées littéralement quand la variable est exprimée littéralement, soit de manière symbolique (pour un poisson sur un fond clair traduisant la variable g, on aura le même poisson mais sur un fond noir pour -g)

- Les opérateurs, quant ils sont nécessaires à l'exercice, sont traduits littéralement.

Le joueur aura à sa disposition tout un ensemble d'actions (clic, drag and drop, ...) pour interagir avec la majorité de ces éléments et chaque action aura un résultat contextuel. A titre d'exemple (non exhaustif) :

- Pour les éléments dans une des zones d'ensembles :

    - Drag and drop d'un élément positif vers l'ensemble antagoniste passera l'élément en négatif et inverssement.

    - Drag and drop dun denominateur vers un numérateur identique de la même fraction supprimera le denominateur et le numérateur concerné.

    - Drag and drop d'un élément d'un ensemble vers un élément identique du même ensemble mais avec un signe opposé suppirmera les deux éléments (+1 ,-1 ou encore « poisson sur fond clair », « poisson sur fond sombre »)

    - Drag and drop d'une constante d'un ensemble vers une constante du même ensemble additionnera les deux élements et créera la constante avec pour valeur le résultat de l'addition. Les deux éléments qui ont servit à l'addition sont supprimés.

    - ...

- Pour les éléments dans la zone réservée aux éléments extérieurs aux deux ensembles et pouvant servir a compléter l'équation :

    - clic sur l'élément changera son signe de positif à négatif et inversement.

    - Drag and drop d'un élément vers un des ensembles aura plusieurs répercussions selon le contexte : Par exemple si je prends une variable g et que je la drag and drop sous la dragon box, on obtiiendra dragonBox/g. Évidement la régle mathématiques imposera que l'on place g en dénominateur des autres éléments des deux ensembles.

Tous les éléments placé dans la zone des éléments extérieurs ne seront pas forcément nécessaires à la résolution de l'exercice. Ce sera au joueur de juger si oui ou non un élément est utile ou pas.

Au final la mécanique est toujours la même. Le joueur résoud l'exercice, la résolution de l'exercice permet au dragon de sortir de sa boite et de manger l'autre ensemble, en d'autres mots on résout l'équation pour réussir l'exercice. Une fois le dragon assez nourrit il sort définitivement de sa boite, la série

d'exercice est finie, le jouer peut passer à une nouvelle série et à un nouveau dragon à nourrir.

Critères de réussite

A hauteur de ce qui a été énnoncé précédement les critères de réussite seront :

- Reprise de l'existant avec amélioration de l'accéssibilité

- Mise en place d'exercices simples sur la base des mathématiques fondamentales (algèbre, ...)

- Application multiplateforme

- Création d'un modèle dynamique acceptant les modifications par l'utilisateur. Ces modifications seront sauvegardées localement pour ne pas être perdues et pour permettre leur accessibilité même après le redémarage du programme.

Risques

- LibGDX ne permettra peut être pas le déploiement d'une architecture MVC

- La complexité de l'intégratin d'un modèle dynamique sera une difficulté majeure. Plusieurs questions seront à résoudre, la principale étant : Quels sont les éléments fondamentaux des mathémtiques à intégrer au modèle pour que tout les autres éléments antérieur puissent être ajoutés ?

- Il sera nécessaire de développer de nombreux écouteurs d'événement avec des actions contextuelles extrèmements variées.

- Le temps de développement est assez court, une bonne organisation sera vitale pour la réussite. A ce titre un dossier d'organisatrion de projet sera mis à disposition du client dès la validation du cahier des charges.