



PROJET VUEJS – DNR2I

La quête de l'Abacus



25 FEVRIER 2018

UNIVERSITE DE CAEN

Julien Pezant & Yanis Abounacer

Contexte du projet

Comment réaliser une application ludique destinée à aider les plus jeunes dans leur apprentissage des tables de multiplication ? Tel était l'enjeu du projet de VueJS cette année. La problématique est à la fois technique mais aussi pédagogique : le parent doit pouvoir suivre l'apprentissage de son enfant tandis que ce dernier doit avoir le sentiment de progresser.

De même, plusieurs consignes devaient impérativement avoir été respectées :

- Générer des questions aléatoirement
- Stocker les réponses de l'enfant
- Afficher une synthèse de la progression de l'enfant

La quête de l'Abacus

Il est apparu très tôt que la réalisation de l'application seule n'offrirait pas suffisamment de consistance pour inciter l'enfant à apprendre ses tables. C'est la raison pour laquelle nous avons décidé d'imaginer une quête dont l'objectif serait de maîtriser les tables de multiplication. A partir d'ici, nous ne parlerons donc plus d'application mais bien de jeu.

Au premier lancement du jeu, l'enfant découvre le contexte de sa quête. Un vieux roi envoie son fils récupérer un artefact protégé par des gardiens : l'Abacus. Maîtriser une table revient à vaincre le gardien et à récupérer un morceau de l'artefact. Lorsque l'enfant a maîtrisé toutes les tables, il dispose de suffisamment d'éléments pour passer le test final et récupérer l'Abacus. Ce test final est plus long que n'importe quel autre et tire des questions au hasard dans toutes les tables.

Une fois l'Abacus récupéré, le texte de fin du jeu s'affiche puis redirige l'enfant au menu principal.

Construction du socle technique

C'est avec tout cela en tête que nous avons entamé notre travail. L'application en VueJS a rapidement pris forme et les premiers écrans ont pu être conçus. Les différents éléments vus en cours comme Vue-router et Vue-ressource sont intervenus dans ce socle technique. Afin de nous concentrer sur l'application, nous avons délégué une partie du style à Bootstrap puis avons affiné les détails pour obtenir ce que nous voulions.

Evaluation, entraînement, résultats

Tels sont les 3 écrans principaux du jeu. Les évaluations tirent aléatoirement des questions sur toutes les tables tandis que les entraînements permettent de se focaliser sur une table spécifique. Les réponses des évaluations et des entraînements sont consignées en tant que résultats. L'enfant avancera donc dans sa quête dès lors qu'il s'entraînera ou s'évaluera.

Les résultats fonctionnent sur la base d'une code couleur. Une opération maîtrisée tendra vers le vert tandis que le rouge et l'orange indiqueront des opérations à retravailler. Un message au-dessus de chaque table indique si elle est maîtrisée ou non. Dans ce cas, l'enfant récupère un indice vers l'Abacus.

Sauvegarde des résultats

Au premier démarrage du jeu, nous nous servons d'un fichier JSON pour initialiser le contenu des tables. Tout est ensuite géré au sein du Local Storage tant que l'utilisateur ne l'efface pas. Structuellement, le JSON s'architecture comme suit :

```

tables: [{
  nom de la table : valeur (compris entre 0 et 10 inclus)
  operations de la table : [{
    autre facteur : valeur (compris entre 0 et 10 inclus)
    produit : résultat du produit
    reussite : nombre de fois où l'enfant a donné le bon résultat
    echec : nombre de fois où l'enfant a donné le mauvais résultat
    tempsMoyen : le temps moyen passé sur cette question
  }, questions suivantes
}, tables suivantes
}]

```

Cette structure possède plusieurs avantages. Tout d'abord, elle est indépendante du type d'application. En effet, un fichier en JSON peut très facilement être lu dans d'autres langages. L'utilisation d'une base de données aurait rendu le tout plus dépendant du web bien qu'il soit possible d'utiliser des connecteurs pour lire des valeurs en base de données. La solution privilégiée ici a l'avantage d'être simple à mettre en œuvre et efficace en JavaScript.

Génération des réponses

Les évaluations et entraînements comptent chacun dix questions. Seul le test final en compte 20. Avant de commencer le test, nous sélectionnons aléatoirement 10 opérations que nous plaçons dans une liste. Nous vérifions ensuite qu'il n'existe pas de doublons de façon à ne pas poser plusieurs fois la même question durant un test. Pour chaque opération, nous générons ensuite deux mauvaises réponses sur la base de celle correcte. Une fausse réponse est un entier aléatoire sélectionné dans $[p - x, p + x]$ où p est le produit de l'opération et x le nom de la table.

Critères de réussite et d'échec

Reste maintenant à déterminer les opérations où l'enfant est en situation de réussite ou d'échec. Nous considérons que l'enfant rencontre un problème avec une opération lorsque :

- Il comptabilise plus de mauvaises que de bonnes réponses sur cette dernière
- Son temps de réponse moyen est supérieur à 7 secondes

Dans les deux cas, ces opérations reviendront plus fréquemment dans les tests de façon à confronter l'enfant à ses difficultés.

Choix de design

Si l'interface du jeu est plutôt simple et accessible, nous avons souhaité travailler sur d'autres détails. Le nom de la quête en premier lieu : l'Abacus. Il s'agit du mot anglais pour désigner un boulier, instrument de calcul utilisé par les grecs et les chinois bien avant les mathématiques modernes. Nous ne souhaitons pas non plus céder à la facilité de nommer les personnages ou les lieux selon des noms connus du monde des mathématiques. Finalement, le seul terme que nous avons retenu est l'Abacus, le reste est anonyme et permet à l'enfant de faire travailler son imagination.

Thématique médiévale

Le choix du roi, du prince et des gardiens suggère une thématique médiévale capable de faire écho à de nombreux mythes et légendes exploitable pour rendre l'aventure plus ludique. Ainsi, le menu principal affiche un arrière-plan de forêt tandis que les entraînements se passent au château. Les évaluations, plus difficiles, sont associées à de dangereuses aventures avec un arrière-plan plus inquiétant.



La forêt



Le château



Le combat

Police d'affichage

Pour s'accorder avec la thématique médiévale en conservant une esthétique qui s'adresse aux jeunes enfants, nous avons choisi une police très ronde et manuscrite pour l'ensemble de l'application. Cette dernière, fournie par Google, s'intègre facilement au code final et convient pour l'affichage des éléments interactifs et d'interface.

Couleurs d'affichage

La première version de l'application était plutôt sombre. Finalement, nous avons opté pour quelque chose de globalement clair à l'exception des évaluations. Les teintes oscillent entre le blanc, le bleu et le vert la plupart du temps, rappelant les grandes forêts, les châteaux et l'imaginaire médiéval.

Le logo

Fruit de plusieurs tentatives, le logo rappelle les gravures médiévales grâce à la police avec serif utilisé pour le A et les différents éléments autour de ce dernier. Il pourrait être utilisé comme seau.



Logo du jeu

Pistes d'amélioration

Avec plus de temps, nous aurions pu encore enrichir l'application et fournir un pan scénaristique plus solide avec des images et des animations. Plusieurs améliorations étaient à l'étude :

- Demander à l'enfant son nom et son sexe
 - Cela aurait permis d'adapter le scénario avec une princesse plutôt qu'un prince, une reine plutôt qu'un roi
- Notifier les parents de la progression de l'enfant
 - Grâce à son mail, nous aurions pu lui transmettre les chiffres détaillés du temps passé à s'entraîner ou à s'évaluer
- Laisse le parent choisir la plage de tables
 - Dans les faits, rien n'empêche l'application de fonctionner avec la table de 16, 67 ou 255, il aurait alors été possible de proposer un outil très personnalisable
- Changer le mode de réponse
 - Actuellement, l'enfant choisi parmi trois réponses
 - Il aurait été possible de pousser la dimension jeu plus loin avec des animations, des bruitages et un système de réponse plus ludique que de simplement cliquer sur un bouton
- Un jeu en ligne
 - Comparer ses scores à ceux de ses amis ou des élèves de sa classe, proposer des épreuves quotidiennes depuis un service en ligne et même jouer à plusieurs pour créer de la compétition

Certaines de ses options sont simples à réaliser, d'autres prendraient beaucoup de temps.

Conclusion

La mise à jour en temps réel des données de VueJS facilite grandement la réalisation d'applications. On ne peut que regretter de ne pas avoir eu plus de temps alloué à la découverte et la prise en main de ce puissant Framework. Le projet en lui-même était très sympathique à réaliser. Travailler en temps limité oblige à faire des choix d'implémentation et de conception pertinent pour obtenir un produit fonctionnel le plus vite possible.