

Sokoban

Lien du repo (Front) : <https://github.com/BastienBYRA/Sokoban-Front>

Lien du repo (Back) : <https://github.com/OlivierButtner/sokoban-back>

Lien de l'APK Android : [Cliquez ici pour installer !](#)

Groupe: Les 3 mousquetaires.

Bastien Byra 

Etienne Legrand 

Olivier Buttner 

Sommaire

Introduction.....	2
Technologie utilisé.....	2
Technologie Front-End.....	2
Technologie Back-End.....	3
Base de données.....	3
Deploiement.....	4
Recul.....	5
Screencast.....	6

Introduction

Le jeu Sokoban est un jeu avec un terrain rectangulaire. Certaines cases sont occupées par des murs, qui bloquent le passage. Sur le terrain, il y a des caisses, des espaces vides et un personnage qui peut se déplacer dans 4 directions par le joueur. Le personnage doit déplacer les caisses sur les emplacements de destination. Il ne peut que pousser les caisses seulement s'il y a un espace vide après la caisse.

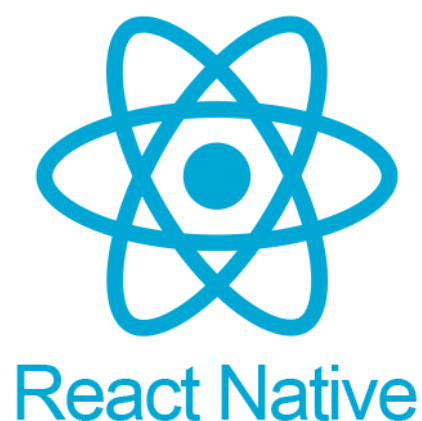
Techno utilisée:

Nous nous sommes regroupés sur le principe d'utiliser React Native en Fonctionnel. Cette manière de coder, conseillée par React, nous a semblé être le meilleur choix de par sa plus grande lisibilité, facilité et cohérence.

Pour le front-end

Le front est, bien entendu, du React Native. Choix ô combien pertinent. Je n'expliquerai pas puisque nous ne sommes pas décisionnaires, mais nous en sommes convaincus.

React Native est une techno créée par Facebook en 2015 suite à la déclaration de Mark Zuckerberg admettant avoir négligé le concept de natif dans le mobile.



Pour le back-end

La deuxième raison de notre association est le souhait de réaliser le back en Symfony. Framework le plus utilisé en France pour PHP, celui-ci nous offre la possibilité de gérer facilement notre API de gestion de l'application.

PHP étant un langage que nous maîtrisons tous, il était plus simple et plus rapide afin de coder chacun sur nos ordinateurs. Sans oublier de le faire en groupe.

D'ailleurs, l'échange est le maître mot de notre groupe. Nous avons énormément échangé sur le sujet. Le but est de se mettre d'accord sur l'idée générale, la ligne directrice de ce projet.

Partage des tâches, leurs attributions, tout a été fait en amont pour que chacun soit clair sur le travail à effectuer.

Ainsi, le jeu a été très rapidement fonctionnel.

Rendu à César ce qui appartient à César, Bastien a sur performé l'application. Avec Etienne, nous avons eu l'impression de n'avoir que l'emballage à réaliser.

Quelques touches esthétiques, un peu de CSS, 2 ou 3 images.



Pour la BDD

Concernant la BDD, je ne sais pas si on peut, à proprement parler, choisir. En réalisant un back-end en Symfony, le choix d'une base relationnelle s'est imposé de lui-même. Ne restait que le choix de MySQL ou SQLite. L'expérience a pris le dessus, nous n'étions pas là pour étudier les BDD.



Déploiement

Ça marche sur nos ordis, c'est bien. Ça marche sur les autres, c'est mieux !

La question de la persistance des données s'est posée au milieu du projet. Maîtres ès méthodes AGILE, nous avons pris parti pour réaliser un back-end déployé et une BDD en ligne.

Nous avons pu créer des niveaux que vous pourrez jouer !

Le back-end est déployé via HEROKU, plateforme de déploiement, gratuite, reliée à Github. Tout en ayant la mise en ligne, nous avons ainsi un pipeline qui met à jour l'appli à chaque push sur la branche master.

Simplicité et efficacité.



La BDD est sur "alwaysdata". Plateforme Open Source et gratuite mettant à disposition des serveurs de sauvegardes.



Recul

Avec le recul, nous aurions pu choisir une ou plusieurs technologies différentes pour le Back et BDD;

Nous aurions pu, au lieu d'une API Symfony et d'une base MySQL utiliser NodeJS avec sa librairie Express et utiliser une base MongoDB

- MongoDB offre une base de données bien plus flexible dans ses formes de données, nous rendant plus évolutifs si le besoin était.
- NodeJS et Express nous permettent de rester dans l'environnement JavaScript.

Nous aurions aussi pu le faire avec d'autres technologies comme .NET ou Java pour l'API et PostgreSQL ou SQL Server pour la base de données par exemple.

Cependant, nous avons fait le choix d'une API Symfony car Symfony génère par défaut un bon nombre de configuration/éléments, permettant de nous concentrer alors uniquement sur la partie qui, en tant que développeur, nous intéresse ; Le développement de fonctionnalités utile à l'application.

Quant à la base de données, notre choix s'est porté sur MySQL car c'est une base de données très facile à mettre en place.

Enfin, nous avons choisi spécifiquement Symfony et MySQL car se sont deux technologies que nous savons tous maîtriser dans notre groupe, permettant alors de travailler rapidement et efficacement, au lieu de devoir déléguer le travail à uniquement celui qui maîtrise la dite technologie.

Screencast

Un enregistrement de l'application, montrant l'utilisation du Back pour récupérer la liste des niveaux ainsi que la "forme" des niveaux.

L'utilisation du Front pour l'affichage des données du back ainsi que la gestion des mouvements du personnages, le calcul des tours et la validation du jeu quand les boîtes sont positionnés sur les cases "triangle".

