

AJAX : Projet 2025-2026

D.Moreaux

22 septembre 2025

1 Abstract

L'exercice ci-présent vise à réaliser une application AJAX basée sur le Javascript côté navigateur et sur PHP/Postgresql côté serveur. Cette application utilisera des méthodes de programmation avancée telles que la machine d'état, l'envoi de données structurées par le serveur dont le rendu sera géré par le navigateur.

La mise en page sera laissée partiellement à l'appréciation de chacun, le but étant de concevoir une interface dont la présentation est correcte, entre autres au moyen de CSS appropriées.

2 Introduction

Le projet constitue en l'implémentation d'une visual novel en AJAX. La visual novel qui sera implémentée sera une adaptation de l'acte 1¹ de la Visual Novel "Katawa Shoujo" (<https://www.katawa-shoujo.com/>). Cette Visual Novel est en Creative Common et peut être téléchargée sur le site (ou sur Steam).

La version à implémenter ne contiendra ni les effets visuels (transitions, animations,...), ni les sons/musiques. Le résultat ne sera donc pas semblable à 100% à la VN d'origine mais restera quand même similaire.

La VN d'origine est programmée en Ren'Py. Elle a été convertie pour que d'une part les données se retrouvent dans une base de données et d'autre part la mécanique de jeu soit réduite à un langage "bytecode" très simple et limité.

Les ressources sont prévues pour un affichage de 800x600.

1. Si la visual novel contient du contenu *18+*, ce contenu n'est présent que dans les actes suivants et est totalement absent de l'acte 1

3 L'interface web

L'interface devra permettre de se connecter avec un nom d'utilisateur et un mot de passe. Si le compte n'existe pas, un nouveau compte sera créé. L'état de la VN, contenant les séquences lues, les variables, ... seront sauveés dans le profil d'utilisateur sous la forme d'une chaîne obtenue avec la fonction `serialize()` du PHP.

L'écran de jeu proposera une zone de 800x600 pour la VN mais également des contrôles (un bouton de déconnexion, un bouton pour masquer/afficher la zone de texte).

On utilisera des `<div>` superposées (z-index) pour les différentes zones du jeu : image de fond, zone de texte, autres images.

4 Le Javascript

La partie Javascript de l'interface de jeu réagira à un clic en demandant le texte suivant. Le serveur peut envoyer plusieurs messages successifs d'une séquence indiquant l'image de fond, les portraits, et le texte à afficher.

Les types de messages sont les suivants :

1	Texte normal (TXT)
2	image de fond (BG)
3	affichage portrait (SHOW)
4	masque portrait (HIDE)
5	note (NOTE)
6	Doublespeak (DBL)

La note apparaît au milieu de l'écran, dans une `<div>` supplémentaire. Le doublespeak apparaîtra dans la zone texte mais contient du HTML qu'il faudra interpréter (et donc, utiliser `.innerHTML`). Ces deux messages fonctionneront pour le reste comme les textes normaux (attendre un clic pour continuer).

Le serveur peut aussi proposer une question avec plusieurs options. Il faudra proposer les différents choix et attendre que l'utilisateur clique sur l'un d'entre eux.

5 Le PHP

Le PHP travaillera dans trois modes : bytecode, QCM ou séquence.

En mode séquence, il se contentera de retourner le message suivant de la séquence en cours. Si la séquence se termine, on repassera en mode bytecode.

En mode QCM, on enverra les différentes options avant d'attendre la réponse du client. Une fois la réponse obtenue, on repasse en mode bytecode.

Le mode bytecode exécutera un programme réalisé dans un bytecode de 10 instructions :

- 0 : NOP** No Operation. ce bytecode est utilisé par le programme qui a créé le bytecode à partir du programme Ren'Py
- 1 : JMP** Jump, saut inconditionnel à l'instruction bytecode dont le numéro est transmis
- 2 : SCENE** Lance une séquence de messages
- 3 : MENU** Affiche un menu. La valeur de retour sera conservée pour la suite
- 4 : IFRET** Saut conditionnel : si la valeur de retour du dernier menu est la valeur fournie, saute à l'instruction bytecode correspondant
- 5 : IFSEEN** Saut conditionnel : si la scène indiquée a été vue, saute à la position indiquée
- 6 : IFGT** Saut conditionnel : si la variable indiquée a une valeur supérieure à la valeur donnée, saute à l'instruction bytecode indiquée
- 7 : VARADD** Ajoute la valeur indiquée à la variable donnée. Les variables non définies valent 0
- 8 : ACT** Indique le numéro d'acte. Peut être ignoré ici (et considéré comme NOP)
- 9 : END** Indique la fin de la VN. Ce bytecode est utilisé quand on arrive en fin d'acte 1 ou d'une Bad-End.

6 La DB

Une série de tables sont fournies (katawa.sql). Ces tables contiennent les données nécessaires à la VN. Il sera nécessaire d'ajouter une table contenant les utilisateurs et leur avancement.

6.1 story

Cette table contient le bytecode de l'histoire. Le champ `id` contient le numéro d'instruction (JMP,...), le champ `type` contient le type d'instruction (de 0 à 9), et les champs `name`, `param` et `next` les paramètres des bytecode,

respectivement texte (nom de la variable ou de la séquence), numérique (valeur de retour, valeur de variable,...) et adresse bytecode (destination d'un saut).

Sauf en cas d'instruction de saut (JMP, IFRET, IFSEEN, IFGT), les instructions sont exécutées dans l'ordre de leur id. En cas de saut non exécuté, on continue également sur l'id suivant.

6.2 seqtag

Contient les associations entre les noms des scènes (utilisés dans le bytecode) et QCM et leur numéro (utilisés dans la DB). Le champ **seqid** contient le numéro de séquence ou question (un numéro ne peut être utilisé que pour un seul des deux), le champ **tag** contient le nom de la scène et le champ **tt** type vaudra 0 pour une question et 1 pour une scène.

6.3 seqtext

Cette table contient les différentes scènes de dialogue.

Les champs **seqid**, **seqserial** et **type** corresponde au numéro de scène, au numéro de pas dans la scène (à effectuer dans l'ordre) et au type de donnée (TXT, BG,...).

La signification des champs qui suivent dépend du type de ligne.

TXT, NOTE, DBL data est le texte à afficher, **elid** le numéro du personnage

BG elid est le numéro de l'image de fond à afficher.

SHOW elid est le numéro de l'image à afficher, **data** est le tag de l'image à afficher (utilisé pour HIDE et en cas de SHOW multiples de la même image), **xpos** est la position horizontale du centre de l'image et **z** est le Z-Index à utiliser (à multiplier par 10 et augmenter du nombre d'images déjà affichées).

HIDE data contient le tag de l'image à masquer

6.4 character

Cette table contient les définitions des personnages.

A l'id **cid** du personnage sera associé un nom **name** et une couleur **color**.

6.5 image

Cette table contient le nom de l'image et ses dimensions pour associer un ID numérique (elid) à une image.

6.6 menu

Cette table contient les questions posées au joueur.

seqid est le numéro de la question (similaire au numéro de scène). **qid** est le numéro du personnage qui pose la question (NULL si il n'y en a pas). **qtext** est le texte de la question et **q1,q2,...** les textes des options.