



# RÉMI RUJOJEVIC

Développeur web



# SOMMAIRE

## MES PROJETS FINIS :

- ◆ **PenaltyFoof** Jeu javascript ..... 4
- ◆ **Météo Rite** Application météo ..... 8
- ◆ **MyParentsMind** Site vitrine ..... 10
- ◆ **Deal or Laugh** Jeu en WebGL ..... 12
- ◆ **ToWatchList** Appli web ..... 16
- ◆ **Azbooka** Appli web ..... 20

## PORTFOLIO EN LIGNE :

**[HTTPS://REMI-RUCOJEVIC.COM](https://remi-rucojevic.com)**

# PENALTYFOOF

## JEU JAVASCRIPT



## Fiche technique

Décembre 2017

**PenaltyFoof** est un jeu de football réalisé en **Javascript** reprenant le principe des séances de tirs au but en fin de match. Le joueur sélectionne son équipe puis l'équipe adverse. Débute ensuite un match en 10 tours où l'utilisateur et l'IA alterneront à chaque tour entre le poste de défenseur et d'attaquant. Le vainqueur est celui qui aura marqué le plus de points à la fin des 10 tours.

**Les objectifs** principaux de ce projet personnel étaient de s'initier au **canvas**, s'entraîner à la récupération d'**APIs** via la technologie **AJAX** et créer un premier jeu jouable par tout le monde. Les objectifs secondaires étaient de s'entraîner au **JQuery** et modifier dynamiquement une page web via des options grâce au **Javascript**.

## Outils utilisés



HTML 5



CSS 3



Javascript



JQuery



API REST



AJAX

# Une interface gérée en JQuery

JQuery est un framework de Javascript. Le jeu disposant de plusieurs menus différents, il fallait donc trouver un moyen de modifier le contenu HTML de la page sans avoir à recharger la page. Javascript est donc l'outil qui permet cela et son framework, JQuery, facilite grandement sa mise en place. Plusieurs div sont donc créés dans le HTML mais ne s'affichent et disparaissent qu'en fonction des actions de l'utilisateur.

Les balises <header> et <Footer> ne bougent pas. La balise #contenu contient plusieurs div mais une seule s'affiche à la fois

```
<div id="background"> ***
</div>
<header> ***
</header>
<div id="contenu">
    <div id="demarrage"> ***
    </div>

    <div id="readMeContenu"> ***
    </div>

    <div id="continent">
        <ul id="listeContinent"></ul>
    </div>

    <div id="pays"> ***
    </div>

    <div id="info"> ***
    </div>

    <div id="match"> ***
    </div>
</div>
<footer> ***
</footer>
```

```
$.ajax({
    url: "https://restcountries.eu/rest/v2/all",
    success : function(result){},
    error: function(){}
}).done(function(result){
    donnees=result;
    $("#demarrage").fadeOut(500);
    $("#continent").fadeIn(500);
    $("#retourDemarrer").fadeIn(500);
    $("#menuInfo").html("Sélectionnez votre équipe");
    //On récupère les différents continents
    var verification=false;
    for (var i = 0; i < donnees.length; i++) { ***}
    //On les affiche dans la div continent
    for (var i = 0; i < region.length; i++) { ***}
    //Choix du continent
    $(".bouttonContinent").click(function(){ ***}
});
```

La fonction ajax() est chargée de récupérer les données de l'API. La fonction done() est chargée de traiter les résultats et de créer une interface pour l'utilisateur en affichant les différents pays sous forme de liste en fonction de leur continent.

## Utilisation d'API

L'utilisateur doit choisir son équipe et son adversaire parmi tous les pays du monde. Ceux-ci ont été récupérés via les données fournies par l'API gratuite REST Countries et la technologie AJAX. AJAX permet de récupérer des données externes et de modifier le contenu HTML sans avoir à recharger la page. Grâce aux données récupérées, chaque pays a pu être classé par continent via l'interface en JQuery.





Asia



Europe



Africa



Oceania



Americas



Polar

Changez le fond ! : Blanc ▾

Changez le titre du jeu ! : PenaltyFooF

Jouer l'animation : ▾

La div #continent est affichée. Lorsque l'on clique sur un des continents, cette div disparait et la div #pays apparait.

< Surprise !!

PenaltyFooF  
Sélectionnez votre équipe

Retour aux choix des continents

Åland Islands	Albania	Andorra	Austria	Belarus	Belgium
Bosnia and Herzegovina	Bulgaria	Croatia	Cyprus	Czech Republic	Denmark
Estonia	Faroe Islands	Finland	France	Germany	Gibraltar

Changez le fond ! : Blanc ▾

Changez le titre du jeu ! : PenaltyFooF

Jouer l'animation : ▾

Dans la div #pays, seuls les pays du continent sélectionné sont affichés.

## Un jeu en Canvas

Une fois les équipes choisies, on peut maintenant passer au jeu. Canvas est un composant HTML permettant de créer des rendus d'images dynamiques. Pour m'initier aux Canvas et parce qu'ils sont une option de premier choix pour un jeu en 2D, j'ai donc choisi de les utiliser pour mon jeu.

```
function redessiner(couleur){
    ctx.clearRect(0, 0, 1000, 500);
    //Zone de tir
    ctx.lineWidth=3;
    ctx.strokeStyle=couleur;
    ctx.font="bold 20pt Arial";
    ctx.fillText("1",200,125);
    ctx.strokeRect(50,50,300,150);
    ctx.fillText("2",500,125);
    ctx.strokeRect(350,50,300,150);
    ctx.fillText("3",800,125);
    ctx.strokeRect(650,50,300,150);
    //...
}
```

< Surprise !!

PenaltyFooF

Retour au menu démarre

France 3 - 3 Germany

Votre adversaire a marqué

À vous de tirer

Changez le fond ! : Blanc ▾

Changez le titre du jeu ! : PenaltyFooF

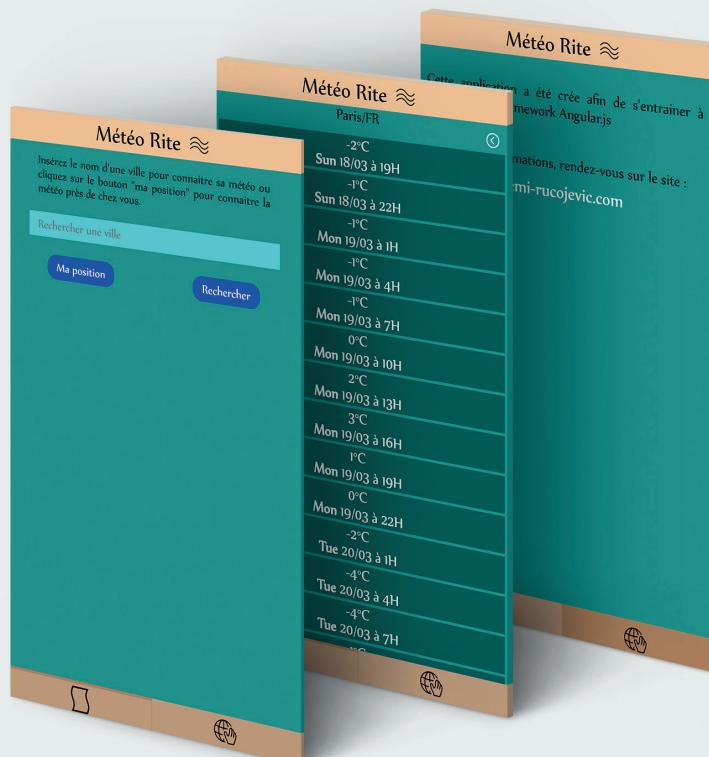
Jouer l'animation : ▾

La fonction redessiner(), appelée à chaque action de l'utilisateur sur le canvas. La variable couleur est soit «noir» soit «rouge» en fonction de si l'utilisateur est en attaque ou en défense.



# MÉTÉO RITE

## APPLICATION MÉTÉO



## Fiche technique

Janvier 2018

**Météo Rite** est une application météo réalisée à l'aide d'Angular.JS et de Node.JS. En entrant le nom d'une ville, on peut connaitre le temps qu'il fait là-bas. On peut également, si la géo localisation est activée, connaitre le temps qu'il fait sur place en sélectionnant l'option «ma position».

**Les objectifs** principaux de ce projet personnel était de s'initier à la création d'applications, et plus précisément d'applications hybrides fonctionnant sur Androïd et iOS avec le même code. Le but était donc de s'initier à Angular.JS et Node.JS, deux frameworks connus pour créer des applications mobiles hybrides puissantes.

## Outils utilisés



HTML 5



CSS 3



Javascript



Angular.js



Node.js



AJAX

# Une application hybride

Selon l'OS, une application ne peut être codée avec le même langage. J'ai donc voulu essayer une alternative pour pouvoir réaliser une application aussi bien sur Android que sur iOS et donc faire une application en HTML et Javascript. J'ai donc utilisé les frameworks Angular.js et Node.js, connus pour aider à réaliser des applications mobile puissantes et hybrides.

```
document.addEventListener('deviceready',function() {
    // body...
}, false);

var app=angular.module('app',[ 'ngRoute']);
app.config(function($routeProvider){
    $routeProvider
        .when('/home',{templateUrl: 'partials/home.html'})
        .when('/about',{templateUrl: 'partials/about.html'})
        .otherwise({redirectTo : '/home'});
});

app.controller('NavController',function ($scope) {
    $scope.menu='home';
});
```

La gestion des menus grâce à Angular.js : \$routeProvider permet de modifier le HTML en fonction du menu sélectionné.

## Le temps selon sa position

Pour voir la météo, il suffit de rentrer le nom d'une ville dans la barre de recherche et l'application ira récupérer les données via l'API OpenWeatherMap puis les affichera. Si elle ne trouve pas de ville correspondante, l'utilisateur sera bien évidemment averti de l'échec. Enfin, l'utilisateur peut également demander la météo via sa position.

```
$scope.search=function(){
    var urlVille="https://api.openweathermap.org/data/2.5/forecast?q="+$scope.city+",fr&app
    $scope.loader=true;
    $http({
        methode: 'GET',
        url:urlVille
    })
```

Search() demande à l'API OpenWeatherMap la météo en fonction de la ville entrée par l'utilisateur.

```
navigator.geolocation.getCurrentPosition(function(position){
    var lat=Math.round(position.coords.latitude);
    var lon=Math.round(position.coords.longitude);
    var urlPos="https://api.openweathermap.org/data/2.5/forecast?lat="+lat+"&lon="+lon+"&app
    $scope.loader=true;
    $http({
        methode: 'GET',
        url:urlPos
    })
```

Le même code mais cette fois en fonction de la géolocalisation de l'utilisateur.





## Fiche technique

Octobre 2017

**MyParentsMind.com** est un site vitrine présentant mon projet tuteuré réalisé au cours de mes deux années de MMI. Le projet est un court métrage d'animation découpé en quatre épisodes. Le site est là pour permettre de présenter le projet et les différents épisodes.

**L'objectif** principal de ce projet est de promouvoir le court métrage d'animation «Ce que disent mes parents - My Parent's Mind», en jouant sur l'originalité.

## Outils utilisés



HTML 5



CSS 3



Javascript



JQuery

# Un site pensé pour le film

Afin de promouvoir le court métrage, nous avons décidé de reprendre totalement l'univers de celui-ci et de se détacher des standards du web. Nous avons donc imaginé un site où nous nous déplacerions dans une ville, celle où évolue le personnage. Les décors et les éléments du site sont donc des références directes à l'univers du court métrage.

## Le scrolling paralaxe

Dans la mesure où nous nous déplaçons dans une ville, nous avons donc imaginé un site où la navigation se fait à l'horizontal. Pour donner de la profondeur à la ville et au site, nous avons réalisé un «scrolling paralaxe» à l'aide de JQuery, un effet où les décors ne se déplacent pas à la même vitesse et donne donc un effet de perspective.



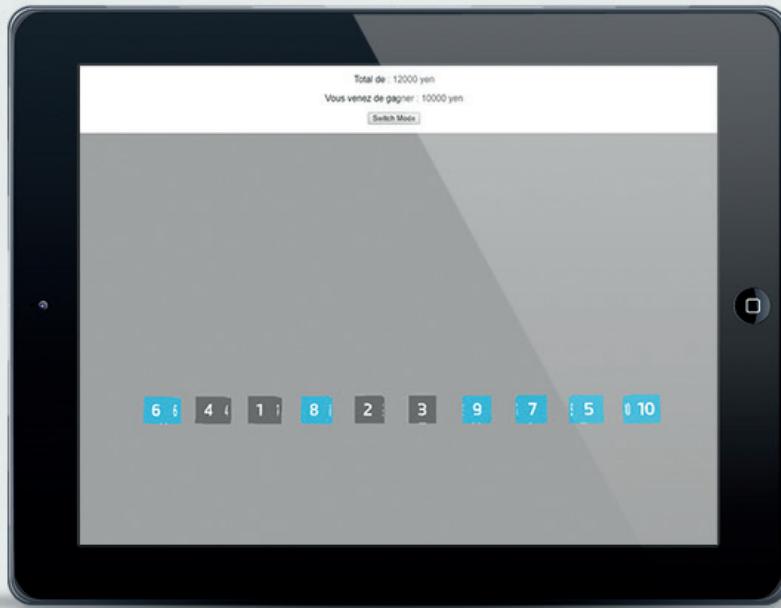
```
$(document).scroll(function(){
    var scroll=$(document).scrollLeft();
    redefineScrolling(scroll);
});

function redefineScrolling(scroll){
    //1er niveau BackGround
    var devant=(scroll*10)/maxScroll;
    $("#devant").css("transform","translate3d("+devant+"%,0,0)");
    //2ème niveau BackGround
    var bga=(scroll*20)/maxScroll;
    $("#bgA").css("transform","translate3d("+bga+"%,0,0)");
    //3ème niveau BackGround
    var bgb=(scroll*30)/maxScroll;
    $("#bgB").css("transform","translate3d("+bgb+"%,0,0)");
}
```

Au moment du scroll de l'utilisateur, la fonction `redefineScrolling()` est appelée et recalcule la position des différents niveaux d'arrière-plan à l'aide d'un produit en croix.

# DEAL OR LAUGH

## JEU EN WEB GL



## Fiche technique

Février 2018

**Deal or Laugh** est un jeu réalisé en Web GL à l'aide de la library Three.JS. Le but du jeu est repris de celui du jeu télévisé «A prendre ou à laisser». Dix boîtes sont numérotées aux hasard. Cinq d'entre elles contiennent une somme d'argent, quatre une blague et une contient une bombe qui stoppe la partie. On ne peut ouvrir les boîtes que dans l'ordre dans lesquelles elles sont numérotées. On peut cependant switcher l'ordre des boîtes autant qu'on le souhaite. L'objectif est de terminer la partie avec la somme la plus élevée possible.

Les objectifs principaux de ce projet étaient de s'initier au Web GL et de le rendre interactif.

## Outils utilisés



HTML 5



CSS 3



Javascript



WebGL



ThreeJS



# Des boites en WebGL ...

L'objet principal de ce jeu, ce sont les boites. Elles doivent être gérées du côté Front (affichage) à l'aide de WebGL et au niveau Back (Fonctionnalités) à l'aide de Javascript. 10 Boites sont d'abord créées en WebGL et placées dans la scène...

```
//Boites
var geometry = new THREE.CubeGeometry(50,50,50);
var material = new THREE.MeshBasicMaterial( {color:0xffffffff,
    map: THREE.ImageUtils.loadTexture( 'img/ferme/1.jpg' ) });
var cube = new THREE.Mesh(geometry, material);
cube.name=1;
scene.add(cube);
for (var i = 1; i < 10; i++) {
    var cubeClone=cube.clone();
    cubeClone.position.set(i*100,0,0);
    cubeClone.name=i+1;
    scene.add(cubeClone);
}
```

10 boites sont créées et ajoutées à la scène en WebGL. Elles ont toutes un nom allant de 1 à 10 permettant de les identifier.

## ... et en Javascript

Placées dans la scène, chaque boite doit maintenant obtenir un contenu. On crée donc en Javascript une classe Boite. Cette classe permettra de générer 10 objets qui renverront chacun vers une des boites créées en WebGL grâce à la correspondance entre l'id de la classe et le nom de la boite en WebGL.

```
//Tableaux
var ordre=[1,2,3,4,5,6,7,8,9,10];
var type=[1,1,1,1,1,5,5,5,5,9];
//1=argent 5=blagues 9=bombe
var tabBlague=["blague 1","blague 2","blague 3","blague 4",
var tabArgent=[1000,2000,5000,7000,10000];

var Boite={
    init: function(id,ordre,type,contenu){
        this.id=id;
        this.ordre=ordre;
        this.ouvert=false;
        this.type=type;
        this.contenu=contenu;
    }
}
```

4 tableaux sont créés. Un pour l'ordre dans lequel les boites doivent s'ouvrir, un autre pour le type de boite et deux autres avec le contenu que peut obtenir une boite selon son type.



```

for (var i = 0; i < 10; i++) {
    //Definie l'ordre
    var ordreFor= Math.floor(Math.random()*(ordre.length));
    var boiteOrdre=ordre[ordreFor];
    ordre.splice(ordreFor,1);
    //Definie le type
    var typeFor= Math.floor(Math.random()*(type.length));
    var boiteType=type[typeFor];
    type.splice(typeFor,1);
    // Definie le contenu en fonction du type
    switch (boiteType) {
        case 1: ...
        case 5: ...
        case 9: ...
    }
    //Creer l'objet
    var boiteFor=Object.create(Boite);
    boiteFor.init(i+1,boiteOrdre,boiteType,boiteContenu);
    boites.push(boiteFor);
}

```

La boucle for va générer 10 objets de la classe Boite. A l'aide de fonctions random, le programme va piocher dans les tableaux précédents des valeurs et créer des objets Boite avec.

```

▼ (10) [{...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}] ⓘ
▶ 0: {id: 1, ordre: 2, ouvert: false, type: 1, contenu: 7000}
▶ 1: {id: 2, ordre: 6, ouvert: false, type: 5, contenu: "blague 5"}
▶ 2: {id: 3, ordre: 3, ouvert: false, type: 1, contenu: 2000}
▶ 3: {id: 4, ordre: 10, ouvert: false, type: 5, contenu: "blague 3"}
▶ 4: {id: 5, ordre: 9, ouvert: false, type: 1, contenu: 5000}
▶ 5: {id: 6, ordre: 8, ouvert: false, type: 9, contenu: "Boom"}
▶ 6: {id: 7, ordre: 7, ouvert: false, type: 1, contenu: 10000}
▶ 7: {id: 8, ordre: 1, ouvert: false, type: 5, contenu: "blague 2"}
▶ 8: {id: 9, ordre: 5, ouvert: false, type: 5, contenu: "blague 4"}
▶ 9: {id: 10, ordre: 4, ouvert: false, type: 1, contenu: 1000}
length: 10

```

Exemple du résultat d'une boucle for. 10 objets Boites sont créés. Leur ID correspondent à des noms donnés aux boites créés en WebGL et leur contenu correspond au type de boite.



# Des informations affichées en HTML

Une troisième partie permet d'afficher et d'informer le joueur sur ce qu'il doit faire et sur ce qu'il vient de se passer. Une fonction s'active donc à chaque fois que l'utilisateur clique sur une boîte. Les informations de la boîte en question sont donc actualisées dans l'objet Boîte correspondante et ces informations sont affichées à l'utilisateur en HTML.

```
if (boites[i].id==target.name) {//Target.name = nom de l'objet 3D sur lequel l'utilisateur a cliqué.  
if (boites[i].ordre==ordreActuel) {//OrdreActuel = Variable vérifiant quel boîte doit être ouverte.  
switch (boites[i].type){  
    case 1:  
        cagnotte+=boites[i].contenu;  
        document.getElementById("score").innerHTML=cagnotte+" yen";  
        document.getElementById("jeuInfo").innerHTML="Vous venez de gagner : "+boites[i].contenu+" yen";  
        break;  
    case 5:  
        document.getElementById("jeuInfo").innerHTML="Voici une blague : "+boites[i].contenu;  
        break;  
    case 9:  
        finDuJeu();  
        break;  
    }  
    boites[i].ouvert=true;  
    actualiseBoites(); //Actualise la texture des boîtes en WebGL  
    ordreActuel++;  
} else {  
    document.getElementById("jeuInfo").innerHTML="Veuillez choisir la boîte n°"+ordreActuel;  
}  
}
```

Un bout du code permettant d'actualiser l'objet Boîte et l'affichage HTML lorsqu'une boîte est ouverte.

## Des améliorations à venir

Le jeu n'est pas encore totalement terminé. Si les boîtes fonctionnent du côté front et back, il serait désormais intéressant de placer les boîtes dans un décor (Une maison, un plateau de télévision...).



# TOWATCHLIST

## APPLICATION WEB



## Fiche technique

Mars 2018

**ToWatchList** est une application disponible sur navigateur. Reprenant le même principe d'une to do list adaptée aux séries télévisées, l'utilisateur pourra créer plusieurs séries en y renseignant le nom et le nombre d'épisodes. Il pourra ensuite cocher les épisodes qu'il aura vu ou non. A l'aide d'un compte utilisateur et d'une base de données, il aura accès à ses séries sur son navigateur.

**Les objectifs** principaux de ce projet sont de créer une application web et d'apprendre à gérer des bases de données et des comptes utilisateurs.

## Outils utilisés



HTML 5



CSS 3



PHP



MySQL



# Des comptes utilisateurs

Afin que plusieurs personnes puissent utiliser l'application, des comptes utilisateurs peuvent être créés.

#	Nom	Type
1	pseudo	varchar(255)
2	mdp	varchar(255)
3	mail	text
4	<u>idUser</u>	int(11)

```
if ((isset($pseudo))&&(isset($mdp1))&&(isset($mdp2))&&(isset($mail))) { //On vérifie que le formulaire d'inscription a été remplis
    if ($strcmp($mdp1,$mdp2)==0) {
        $reponse = $dbh->query("SELECT * FROM user");
        while($donnees = $reponse->fetch()){ //On vérifie que le pseudo ou le mot de passe n'est pas déjà utilisé
            if ((strcmp($mail,$donnees['mail'])==0)||strcmp($pseudo,$donnees['pseudo'])==0)) {
                $verification=1;
            }
        }
        if ($verification==1) { //Si la vérification ne détecte rien d'anormal...
            echo("<p>Le pseudo ou le mail est déjà utilisé (réactivez le javascript pour plus d'informations)</p>");
        } else {
            $req = $dbh->prepare('INSERT INTO user(pseudo, mdp, mail) VALUES(:pseudo, :mdp, :mail)');
            $req->execute(array(
                'pseudo' => $pseudo,
                'mdp' => $mdp2,
                'mail' => $mail
            ));
            header('Location: index.php');
        }
    } else {
        echo "<p>Les 2 mots de passe ne correspondent pas</p>";
    }
}
```

La table MySQL des utilisateurs et le code permettent d'inscrire un utilisateur une fois un formulaire correctement remplis.

```
$pseudo=$_POST["pseudoConnexion"];
$mdp=$_POST["mdpConnexion"];
if ((isset($pseudo))&&(isset($mdp))) {
    $reponse = $dbh->query("SELECT * FROM user");
    while($donnees = $reponse->fetch()){
        if ((strcmp($donnees["pseudo"],$pseudo)==0)&&(strcmp($donnees["mdp"],$mdp)==0)) {
            $verification=1;
            $id=$donnees["idUser"];
        }
    }
}
```

Le code qui vérifie que le pseudo et le mot de passe entrés par l'utilisateur dans le formulaire de connexion correspondent bien à un utilisateur inscrit.



```

if ($verification==1) {
    $_SESSION['pseudo'] = $pseudo;
    $_SESSION['id']=$id;
    header('Location: user.php');
} else {
    header('Location: index.php');
}

```

Si la vérification réussit, alors on crée une session et on envoie l'utilisateur vers la page d'accueil des utilisateurs.

## Des séries et des épisodes

Une fois un compte créé, il est désormais possible pour l'utilisateur de créer des séries. Via un formulaire, l'utilisateur pourra entrer le nom de la série, l'image de la série et le nombre de saisons ainsi que le nombre d'épisodes dans chaque saison.



Exemple de formulaire rempli pour créer une série. Le formulaire réagit dynamiquement en fonction du nombre de saison.

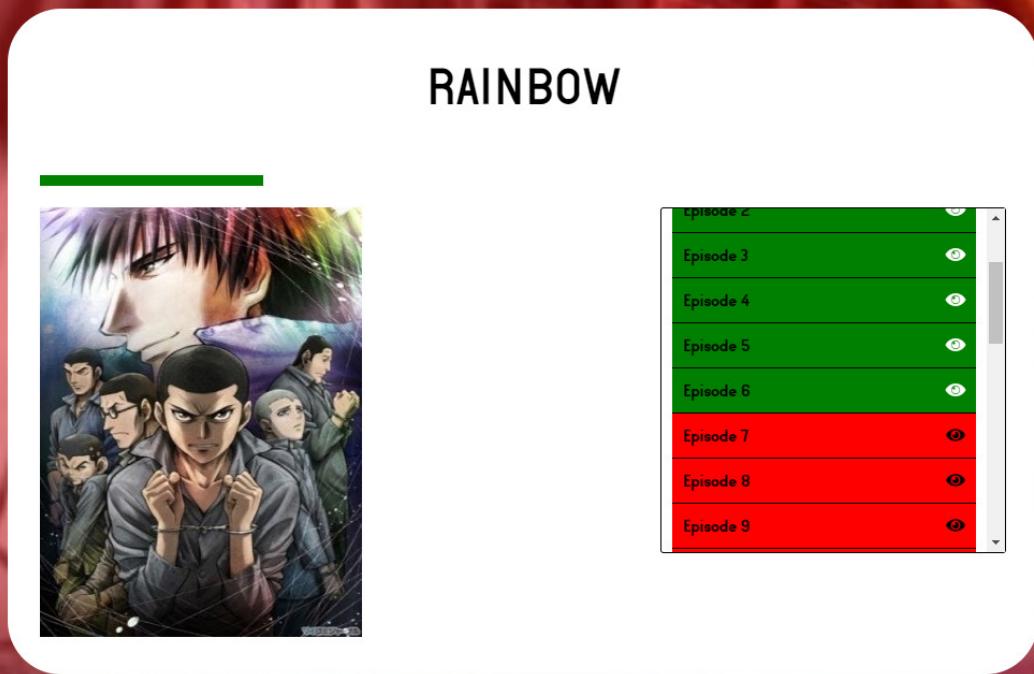
Afficher							Structure		SQL		Rechercher		Insérer		E		
#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra										
1	nomSerie	text	utf8_bin		Non	Aucune											
2	<u>idSerie</u>	int(11)			Non	Aucune	AUTO_INCREMENT										
3	<u>idUser</u>	int(11)			Non	Aucune											

#	Nom	Type	Int
1	<u>idEpisode</u>	int(11)	
2	<u>idSerie</u>	int(11)	
3	<u>idUser</u>	int(11)	
4	<u>saison</u>	int(11)	
5	<u>numEpisode</u>	int(11)	
6	<u>vu</u>	tinyint(1)	

A gauche la table des séries et à droite la table des épisodes. Une série appartient à un utilisateur (grâce à idUser) et seul lui peut la modifier ou voir ses informations. Un épisode appartient à un utilisateur et une série (grâce à idSerie) et lorsque l'on regarde les informations d'une série, seuls les épisodes appartenant à cette série s'affichent. Le champ saison permet de savoir à quelle saison appartient l'épisode et le champ vu permet de savoir si l'épisode a été vu (=1) ou non (=0).





Exemple d'affichage d'une série.

```
$requete = "SELECT * FROM episodes WHERE idSerie='".$_GET["id"]';
$requeteTitre = "SELECT nomSerie AS titre FROM series WHERE idSerie='".$_.GET["id"]."'";
$response = $dbh->query($requete);
$responseTitre = $dbh->query($requeteTitre);
```

Code permettant de récupérer les données d'une série en fonction de son ID.

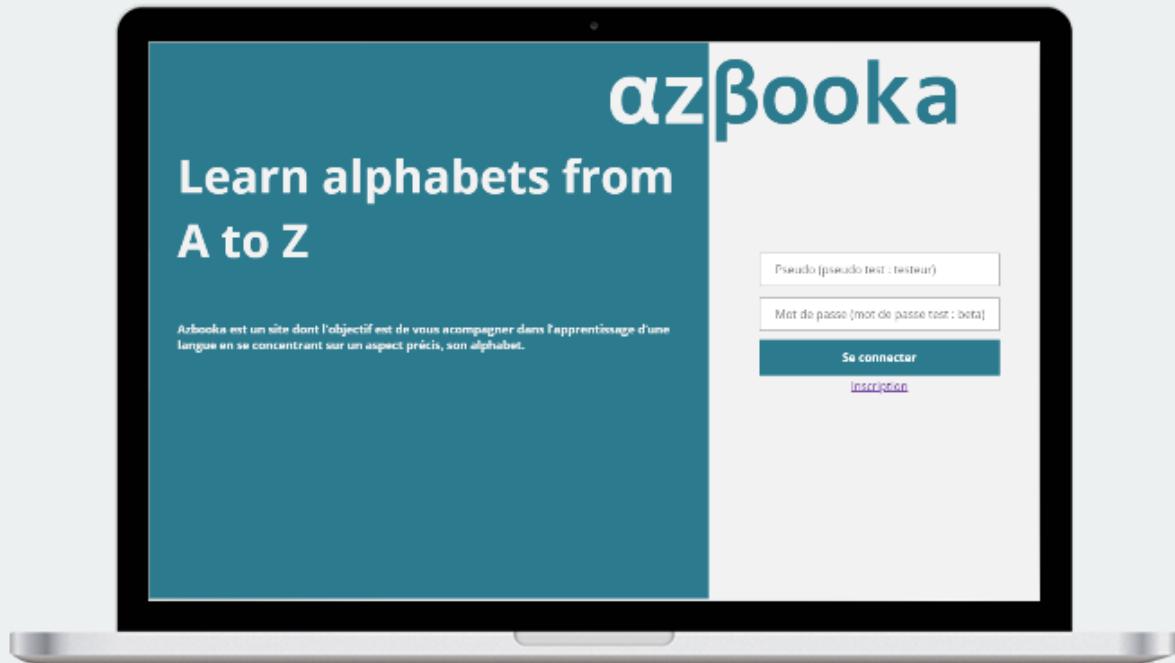
## Une application à terminer

L'application est bien avancée. Cependant elle n'est toujours pas terminée. Le design n'est toujours pas codé, me concentrant d'abord sur l'aspect fonctionnel, et quelques fonctionnalités sont encore à ajouter comme la possibilité de fermer son compte utilisateur, supprimer ou modifier une série. Pour finir, il faudra également exporter l'application sur Android à l'aide de Java et d'Android Studio.



# AZBOOKA

## APPLICATION WEB



## Fiche technique

Mai 2018

**Azbooka** est un petit projet personnel dont le but est d'aider les personnes à apprendre des alphabets. Le site a pour but de mettre à disposition des cours sur les différents alphabets qui existent dans le monde puis permet de tester ses connaissances grâce à des quiz. La version 1 beta, est disponible en ligne mais une 2ème version est en cours de réalisation et prévoit une refonte complète.

Les objectifs principaux de ce projet étaient de mettre en pratique mes compétences en back et en front et de créer un site à but pédagogique.

## Outils utilisés



HTML 5



CSS 3



PHP



MySQL



# Plusieurs cours disponibles

Gérés à l'aide d'une base de données, il existe plusieurs alphabets réparties dans plusieurs catégories et possédants plusieurs cours.

The screenshot shows the azBooka application interface. On the left, a sidebar lists database tables: Table, alphabets (highlighted in green), cours (highlighted in pink), dico (highlighted in blue), groups\_alphabets (highlighted in yellow), and users. The main content area has two sections: 'Cyrillique' (with a Cyrillic character icon) and 'Japonais' (with a Japanese character icon). Each section contains a brief description and links to specific sub-sections. Below these are three cards for 'Les lettres communes au français 1'. The first card shows a Cyrillic character with its name in Serbian and its name in French. The second card shows a Cyrillic character with its name in French and its pronunciation. The third card shows a French character with its name in French and its pronunciation. A 'Clavier' section is also visible on the left.

**azBooka**

**Cyrillique**

**Japonais**

Le cyrillique est un alphabet principalement utilisé dans les pays d'Europe de l'est. Il dispose de plusieurs variantes selon les pays :

Cyrillique Serbe

Le japonais dispose de 3 alphabets différents pour une seul et même langue ! Les hiraganas, les katakanas et les kanjis :

Hiragana

Katakana

**Table**

**alphabets**

**cours**

**dico**

**groups\_alphabets**

**users**

**Cyrillique Serbe**

Les lettres identiques

Les faux amis

les lettres communes au français 1

les lettres communes au français 2

Les nouvelles lettres 1

Les nouvelles lettres 2

Clavier

**Les lettres communes au français 1**

En plus des précédentes, il y a 9 lettres issues de l'alphabet français. Nous allons ici en voir 5 :

**Д**  
Minuscule : д  
Prononciation : comme dans "deux"

**И**  
Minuscule : и  
Prononciation : comme dans "idée"

**L**  
Minuscule : l  
Prononciation : comme dans "lac"

Architecture de la base de données et présence des différents éléments sur le site.

## Des tests qui se renouvellent

Le système de test, codé en javascript et en php, va récupérer au hasard 10 mots dans la base de données (la table «dico») qui correspondent au niveau du test. (Exemple : «Les lettres communes au français 1» est le troisième niveau du cours sur le cyriliqe serbe. Le test pour ce cours ne peut contenir que des mots contenant des lettres apprises lors de ce cours ou des précédents, afin que l'utilisateur puisse répondre aux questions.



```

$.ajax({
    url : '/templates/getTest.php',
    type : 'GET',
    data : 'level=' + level + '&id=' + id,
    dataType : 'json',
    success : function(data, statut){
        $("#question-info").html("Question n°"+(actual+1));
        $("#question").html(data[0]["mot_et"]);
        $("#trad").html(data[0]["pronunciation"]);

        $("#test-form").submit(function(e){
            var answer=$('#test-form-answer').val().toLowerCase();
            var goodAnswer=data[actual]["mot_fr"];
            if (answer === goodAnswer) {
                nextQuestion(data);
            } else {
                numberTry++;
                if (numberTry<3){
                    $("#try").html("Faux, il vous reste "+(3-numberTry)+" essais");
                } else {
                    nextQuestion(data);
                }
            }
            e.preventDefault();
        });
    }
});

```

Code Ajax permettant de charger et de gérer le test.

The screenshot shows the azBooka application interface. On the left, there is a sidebar with the following menu items:

- Cyrillique Serbe (with a flag icon)
- Les lettres identiques
- Les faux amis
- Les lettres communes au français 1 (highlighted in dark blue)
- Les lettres communes au français 2
- Les nouvelles lettres 1
- Les nouvelles lettres 2
- Clavier

The main content area displays a test question:

Question n°9

francaуска

la France

A red input field is present, and a teal "Valider" button is at the bottom right.

Screen d'un test.

	<input type="checkbox"/> Modifier	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	id	mot_fr	mot_et	pronunciation	level	id_alphalet	lettre	visible
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	1	a	а	"a" comme dans "âge" ou "arc".	1	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	2	e	е	"é" comme dans "école".	1	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	3	j	ј	"y" comme dans "yaourt"	1	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	4	k	к	"k" comme dans "kilo"	1	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	5	m	м	"m" comme dans "mai"	1	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	6	o	о	"o" comme dans "ôte" ou "ordre"	1	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	7	t	т	"t" comme dans "tour"	1	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	8	ja	ја	je	1	1	0	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	9	moja	моја	ma (adjectif possessif)	1	1	0	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	10	mama	мама	maman	1	1	0	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	11	h	х	H inspiré comme dans "Hahaha !"	2	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	12	n	н	comme dans "nez"	2	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	13	r	р	roulée comme en espagnol	2	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	14	s	с	comme dans sac	2	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	15	u	у	comme un "ou" dans "cour"	2	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	16	v	в	comme dans "voix"	2	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	17	ja sam	ја сам	je suis	2	1	0	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	18	tri evra	три евра	trois euros	2	1	0	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	19	hitno je	хитно је	C'est urgent !	2	1	0	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	20	ne	не	Non	2	1	0	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	21	d	д	comme dans "deux"	3	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	22	i	и	comme dans "idée"	3	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	23	l	л	comme dans "lac"	3	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	24	p	п	comme dans "pas"	3	1	1	1
	<input type="checkbox"/>	<input type="button" value="Copier"/>	<input type="button" value="Effacer"/>	25	b	б	comme dans "bon"	3	1	1	1

Table «dico» : permet de créer les tests et les cours.

**EN VOUS REMERCIAINT D'AVOIR PRIS LE  
TEMPS DE REGARDER CE BOOK.**

---

**[HTTPS://REMI-RUCOJEVIC.COM](https://REMI-RUCOJEVIC.COM)**





# RÉMI RUCOJEVIC

Recherche alternance :  
Développeur full-stack

@ remi.rucojevic@gmail.com  
remi-rucojevic.com  
+33 6 3159 4162

## Compétences

### FRONT-END

- Concevoir un site web
- Développer son aspect visuel



### BACK-END

- Heberger un site web
- Développer ses fonctionnalités
- Gérer une base de données



### GRAPHISME

- Créer une maquette graphique
- Concevoir une charte graphique



### GESTION DE PROJET

- Outils UML
- Méthode agile



### LANGUES

Anglais	A1	A2	B1	B2	C1	C2
Serbo-croate	A1	A2	B1	B2	C1	C2
Japonais	A1	A2	B1	B2	C1	C2

## Portfolio

[remi-rucojevic.com](http://remi-rucojevic.com)

## Expériences

Avril 2018 -  
Juin 2018

### TOOSLA (PARIS - FRANCE)

Stagiaire développeur full-stack

- Reprise du Back office et de l'API
- Correction de bugs
- Créations de nouvelles fonctionnalités
- Utilisation des frameworks Laravel et Angular.JS

Janvier 2018

### CAQ.FR (TÉLÉTRAVAIL)

Rédacteur web et assistant webmaster

Juillet-Août  
2017

### HÔTEL LAFORET (HAKUBA - JAPON)

Employé polyvalent en hôtellerie

Mars 2017 -  
Mars 2018

### DYSNOMIA (PROJET ÉTUDIANT)

- Développeur web, ingénieur son et manager de l'équipe artistique

## Formations

À partir de  
septembre  
2018

### GOBELINS (PARIS)

Bachelor Designer & Développeur Interactif

- Option : Développement

2016 - 2018

### UPEM - IUT DE MEAUX (MEAUX)

DUT MMI (Métiers du multimédia et de l'internet)

- Spécialité : Développement web et androïd

2014 - 2016

### LYCÉE RENÉ CASSIN (NOISIEL)

Baccalauréat S (Scientifique)

## Intérêts



Tennis



Technologie /  
ingénierie



Culture  
japonaise



Musique  
étrangère



Géographie



