

# TP

## DryField

### Jeux de simulation web

*Principes illustés :*

*REST*

*MVC*

*Technologies :*

*JavaScript*

*AJAX*

*JQUERY*

# Objectif

Réaliser une application autonome javascript qui communique avec une api web REST.

# Besoin

Le but du jeu est de faire le plus grand nombre de récolte possible.

Le principe est de devoir remplir les citernes d'eau de 3 champs qui en ont besoin d'être irrigué tout les jours (seconde dans le jeux) pour pousser.

Le joueur à une réserve d'eau globale qu'il peut remplir en achetant de l'eau. Il irrigue ensuite ses champs en puisant dans cette réserve globale. Le joueur vend les champs arrivé à maturité pour récupérer de l'argent qu'il réinvestira dans de l'eau.

Si un champ est en rupture d'eau (plus d'eau dans la citerne du champ) sa récolte est perdue. quand la citerne sera réalimentée, le décompte de maturité du champ repart à 0.

La consommation des champs va augmenter pour simuler une aridité. Le rythme de jeux va donc s'accélérer jusqu'à avoir les trois champs sans eau dans leurs citerne. La partie est alors terminée. Le nombre de récoltes effectuées est le score de la partie.

Le score est envoyé sur un serveur qui met à disposition une API web Service REST pour recevoir les nouveaux score et mettre à disposition les scores déjà enregistrés. Le joueur peut consulter les scores de tout le monde

L'objectif de temps moyen de jeux est de 10 minutes et le temps maximum 15 minutes.

# Règles de gestion

- argent initiale : 50\$
- réserve initiale : 3 L / citerne + 3 litres globale
- consommation initiale : 1l / seconde
- consommation maximum : 2L / seconde
- Vous devez trouver une règle de l'augmentation de la consommation pour atteindre les objectifs de temps de jeux
- nombre de seconde pour qu'un champ soit mur : 20 secondes
- prix de l'eau : 1\$ / L
- prix d'une récolte : 40\$
- un champ prêt à récolté ne consomme pas d'eau

# Ecran principale

## Wireframe

[https://moqups.com/simon\\_louvet/7gfaQ7OA/p:ab7fe3653](https://moqups.com/simon_louvet/7gfaQ7OA/p:ab7fe3653)

## Interaction

L'achat d'eau ouvre une popin met en pause le jeux.

La couleur du bouton récolter change de couleur quand le champ est mur.

Une fenêtre s'ouvre à la fin du jeux pour saisir le nom du joueur.

# Écran des scores

## Wireframe

[https://moqups.com/simon\\_louvet/7gfaQ7OA/p:aeb7fb4d5](https://moqups.com/simon_louvet/7gfaQ7OA/p:aeb7fb4d5)

## Contraintes technique

- serveur REST conseillé : MongoLab
- Utilisation de JQuery pour la manipulation du DOM et les appels AJAX
- Application du MVC sur toute la conception de l'application javascript. Vous devez avoir des objet Vues, des objets Contrôleur et un objet Modèle qui communiquent dans le principe du MVC. Le passage de la vue Jeux à la vue Score n'arrête pas le modèle qui applique les règles de gestion toute les secondes.

## coup de pouce

Utiliser setInterval pour la gestion du temps.

Les scores des joueurs récupérés en AJAX modifie le model qui notifie la vue.