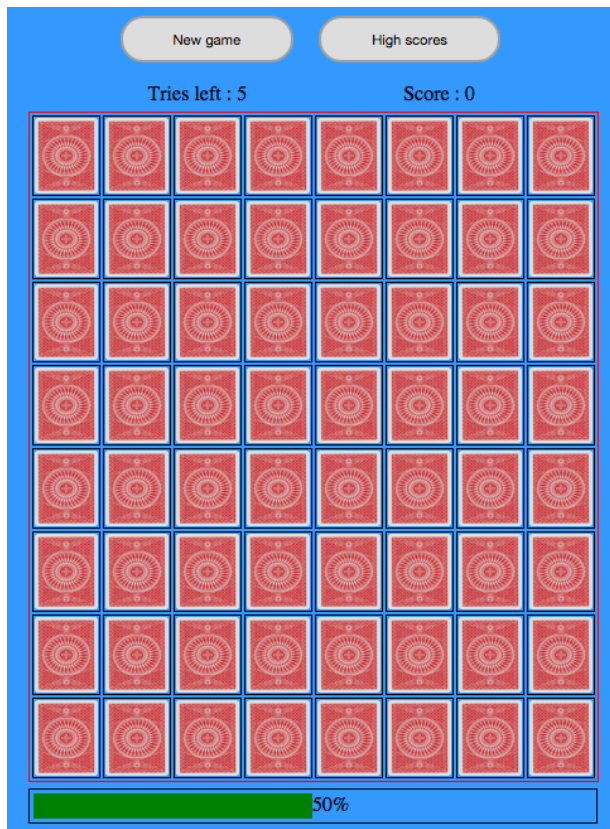


# Projet tuteuré

## Enoncé :

### Description générale :

Vous devez créer le jeu [Bejeweled](#) sous forme d'une application mobile (vous trouverez l'idée ci-dessous mais visuellement parlant, vous êtes totalement libre) :



Au début d'une partie, l'utilisateur va remplir son nom, prénom et son email. L'application génère une grille d'images de taille 8X8. Pour chaque cellule de la grille une image de diamant est choisie aléatoirement parmi 8 et affichée dans la cellule. Lors de la génération de la grille, il n'est pas acceptable d'obtenir trois cellules successives (verticalement ou horizontalement) qui ont la même image.

Le but du jeu est d'aligner verticalement ou horizontalement au moins trois cellules ayant la même image. L'alignement d'au moins trois cellules avec la même image augmente le score du joueur et augmente le temps restant dans la barre de progression qui est décrétementée de « niveau » toutes les 3 secondes. « niveau » est la variable stockant le niveau qui est initialisé à 1. Si la barre de progression atteint 0, le joueur perd la partie. Si la barre de progression atteint 100, il passe au niveau suivant où la barre de progression décrémente plus rapidement mais les gains en score sont plus importants. Au début de la partie la barre de progression est à moitié remplie.

Le joueur ne peut modifier l'ordre des images qu'en permutant (horizontalement ou verticalement) deux images dans deux cellules adjacentes. Si la permutation de deux images de deux cellules adjacentes ne donne pas lieu à au moins trois cellules successives ayant la même image, l'opération de permutation est annulée et le joueur perd un essai. Si au moins trois images identiques sont alignées après une permutation, ces images seront supprimées et remplacées par les images situées au-dessus de celles-là. Donc pour chaque image à remplacer, les images de la même colonne sont décalées vers le bas et de nouvelles images sont sélectionnées aléatoirement pour combler le vide.

Quand le joueur aligne 3 images il gagne  $50 \times \text{« niveau »}$  points, pour 4 images  $150 \times \text{« niveau »}$  et pour 5 images  $500 \times \text{« niveau »}$ . « niveau » est égale à 1 au début et incrémenté de 1 chaque fois qu'un niveau est terminé.

Exemple : nous sommes au niveau 1, j'ai donc 50/100 dans ma barre de progression. J'aligne 4 images, donc je marque 150 points. La somme est donc de 200 points. Je passe donc du niveau 1 au niveau 3 (car quand j'atteins 200 points, j'avais validé le niveau 2). Donc le score de ma barre de progression est de 200/300. Elle est donc remplie au 2 tiers.

La partie est terminée si l'utilisateur n'a plus d'essais ou bien si la barre de progression atteint 0. Quand elle est terminée, le score est comparé avec les 6 meilleurs scores dans la base de données, s'il est meilleur que l'un d'eux, il apparaît dans le classement final.

Vous devrez ensuite ajouter un bouton « Hint » qui aide le joueur à trouver une opération de permutation et s'il y en a pas, termine la partie.

Vous allez ajouter un bouton « Pause » qui arrête la partie en attendant que l'utilisateur appuie de nouveau sur le bouton « Pause » qui s'est transformé en « Resume ». Pendant l'arrêt, la barre de progression n'est pas décrétementée, les images des boutons sont cachées et les boutons sont désactivés.

Lorsque l'utilisateur joue (une ou plusieurs parties) et qu'ensuite il s'arrête, un e-mail lui est envoyé (1h après la fin de sa dernière partie) comprenant la liste de ses différentes parties et les scores qu'il a fait. De même si un utilisateur perd une place au classement, on l'informe de cela par email en lui indiquant un récapitulatif du classement des 6 meilleurs ainsi que de leurs points.

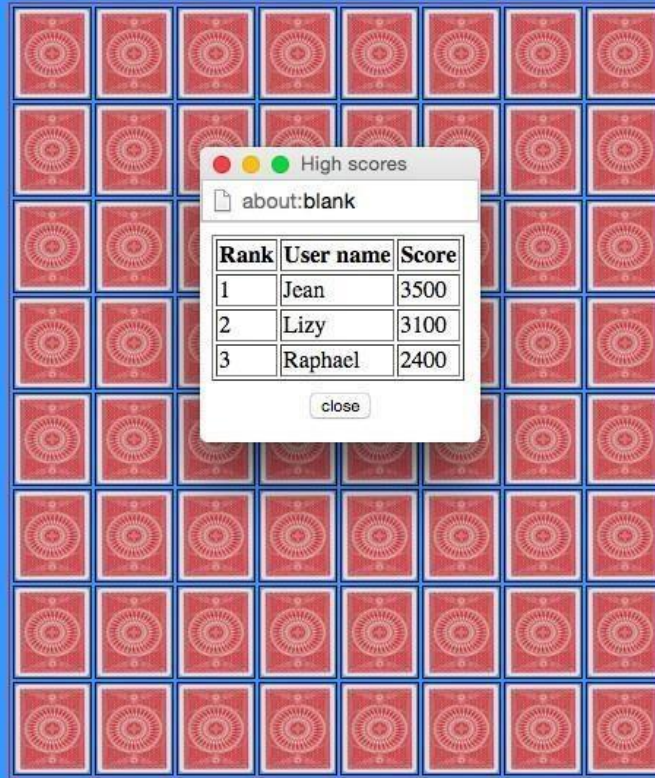
La fenêtre principale contient aussi deux boutons : le premier pour recommencer le jeu et le second pour afficher dans une nouvelle fenêtre du navigateur les meilleurs scores selon la figure suivante :

New game

High scores

Tries left : 5

Score : 0



High scores

about:blank

Rank	User name	Score
1	Jean	3500
2	Lizy	3100
3	Raphaël	2400

close

50%

### Implémentation :

Tout le jeu doit être implémenté avec langage mobile. **Les langages back-end tel que PHP** ne peut être utilisé que pour répondre aux requêtes **front-end**. La récupération des scores de la base de données doit être faite avec une requête. La sauvegarde du nouveau record doit être effectuée à travers une requête.

### Fonctionnalité à implémenter pour gagner des points bonus :

1. Dans le mail final envoyé, indiquer également de temps qu'a passé l'utilisateur sur le jeu
2. L'utilisation d'un framework back-end tel que Symfony, Laravel ou encore NodeJS (vous ne les verrez pas durant la formation donc c'est en auto-formation)
3. Donner la possibilité au joueur de personnaliser les images qui s'affichent dans sa grille