

MINI PROJET I

Jeu sur navigateur

OBJECTIF

L'objectif de ce premier *mini projet* est le développement d'un petit jeu sur navigateur. Il y a deux projets distincts et l'attribution du projet à chaque équipe se fera de façon aléatoire :

- **Twin Images Game** *c'est un jeu de mémoire dont le but est d'identifier deux images identiques de suite.*
- **Puzzle Image Game** *c'est un jeu de réflexion dont le but est de reconstituer une image à partir de ses sous-parties.*

IMPORTANT

Lors de la réalisation de ce premier *mini projet*, il est nécessaire de respecter les directives suivantes :

- C'est une application qui tient sur une même et unique page Web (*un fichier .HTML*),
- Il faut impérativement utiliser un Framework CSS pour le rendu graphique : **Bootstrap 4**,
- **Vous ne devez utiliser que ce qu'on a vu en cours**,
- Avoir une bonne organisation au niveau des fichiers (*HTML, CSS, JS*),
- Utiliser les *notions d'orientée objet* lors de l'implémentation,
- Niveau **JavaScript**, vous ne pouvez utiliser que le Framework **jQuery**.

DESIGN GRAPHIQUE

Afin d'avoir une interface conviviale, vous devez respecter le design qui est défini ci-dessous dans les **Figure 1 & Figure 2** suivant le projet attribué :

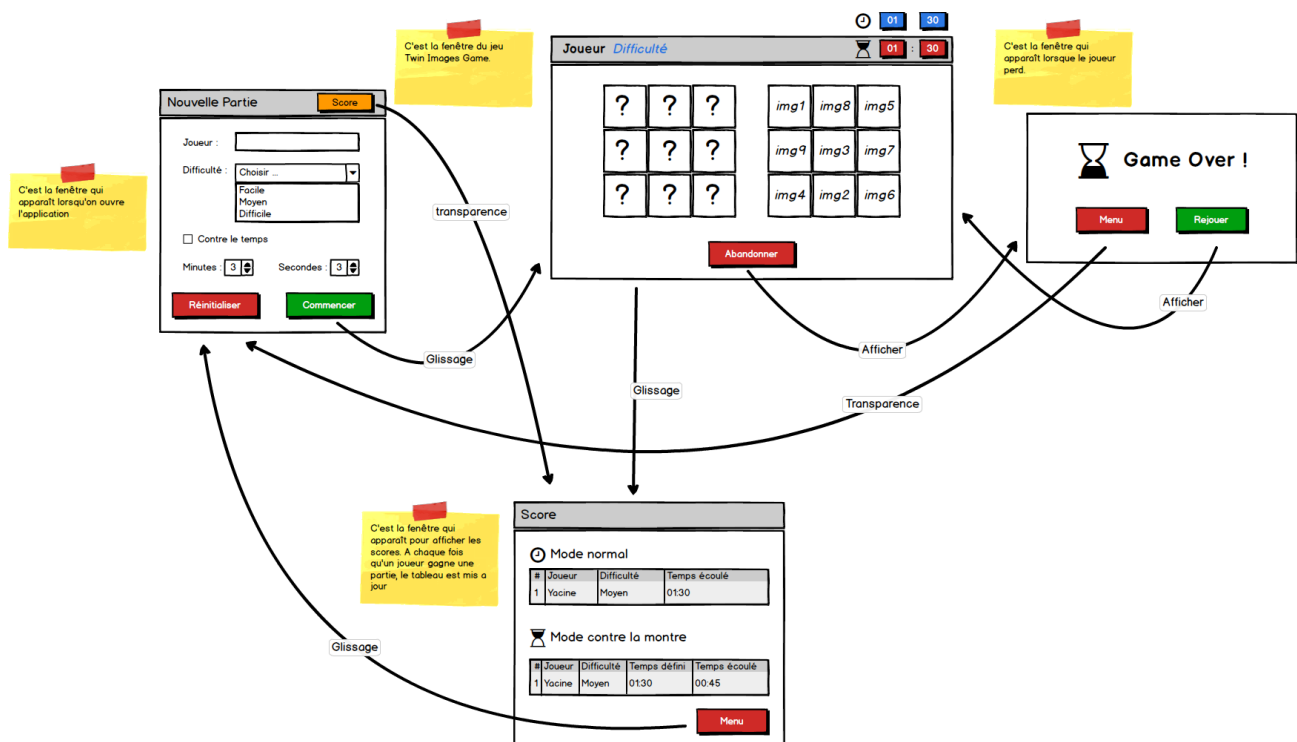


Figure 1 : Puzzle Images Game

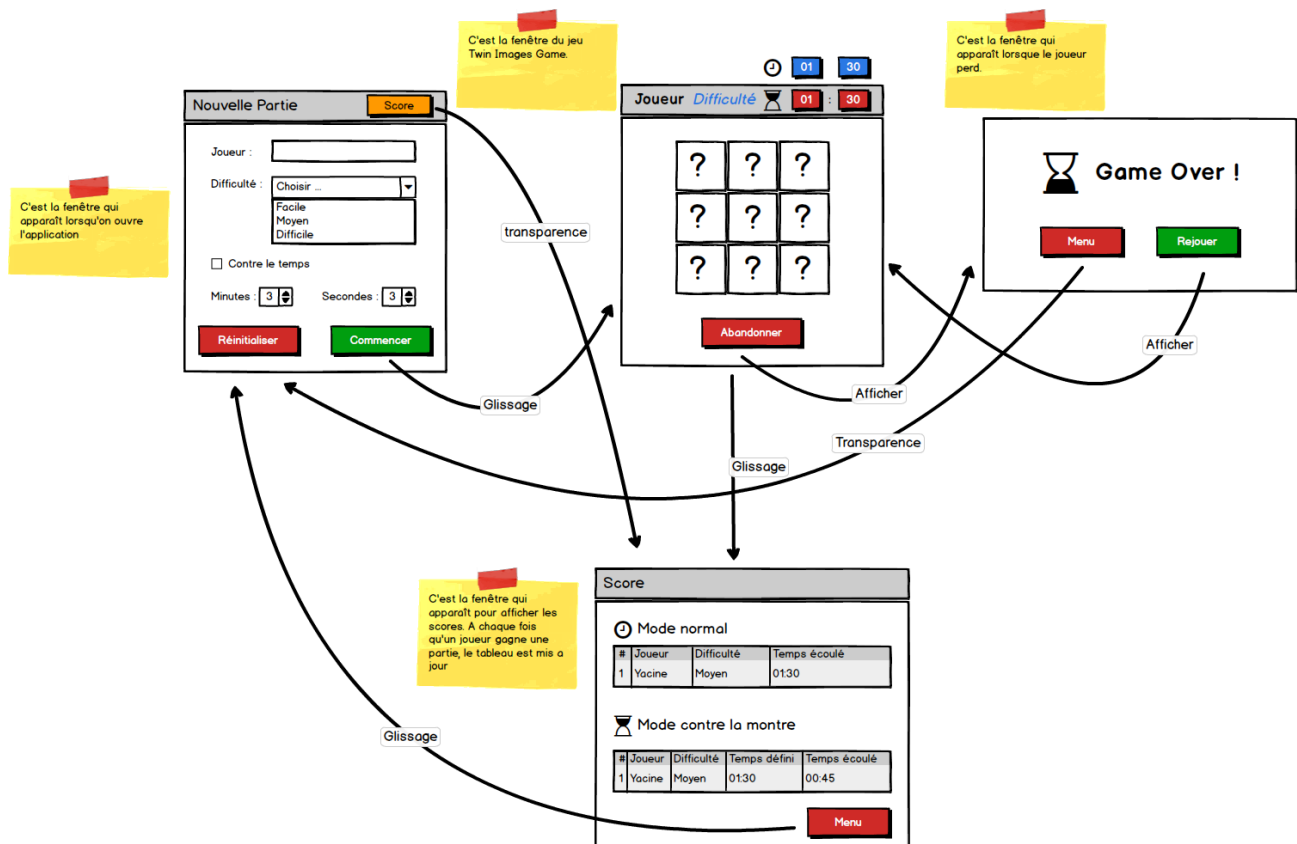


Figure 2 : Twin Images Game

LES COMPOSANTS

Comme défini ci-dessus dans les deux figures, on distingue quatre composants principaux :

1. Démarrage

C'est le composant initial qui est affiché lors de l'ouverture de l'application. Il y a les champs suivants :

- **Joueur** : *Le joueur doit y entrer son nom ou bien un pseudo.*
- **Difficulté** : *Le joueur choisit le niveau de difficulté parmi :*
 - Facile : *Twin Images 4 x 4 & Puzzle Image 3 x 3*
 - Moyen : *Twin Images 6 x 6 & Puzzle Image 4 x 4*
 - Difficile : *Twin Images 8 x 8 & Puzzle Image 5 x 5*
- **Mode contre la montre** : *C'est une case à cocher pour activer le mode contre la montre.*
- **Minutes & Secondes** : *Ces deux champs sont activés que si la case précédente est cochée. Une période est définie et le joueur essaye de finir le jeu avant le temps entré.*

Il y a également trois boutons sur le composant initial :

- **Réinitialiser** : *Il va permettre de remettre à zéro les différents champs.*
- **Commencer** : *Il va permettre de passer à la fenêtre du jeu avec un effet de **glissement**.*
- **Score** : *Il va permettre d'afficher le composant des scores avec un effet de **transparence**.*

2. Jeu

Il y a deux cas possibles selon le jeu à développer :

a. Twin Images Game

Il y a un tableau avec des cases en « ? ». Le nombre de cases est défini selon la difficulté sélectionnée par le joueur. Il faut sélectionner plusieurs paires d'images à partir du dossier d'images. Le choix des images et leur disposition sont définis de façon aléatoire.

A chaque fois que le joueur clique sur une case spécifique du tableau, l'image qui y est contenu est affichée avec un effet de **transparence**. L'image reste affichée jusqu'à ce que le joueur sélectionne une autre case, il y a deux cas possibles :

1. **Les deux images sont identiques** : *dans ce cas, les deux images restent affichées,*
2. **Les deux images sont différentes** : *dans ce cas, les deux images sont remises comme elles étaient initialement avec le « ? ».*

Le but du jeu est d'identifier tous les couples d'images. Le score de chaque joueur est défini par le temps qu'il a mis pour trouver tous les couples d'images.

Base d'images : elle se trouve dans le dossier « *Twin_Image_Game_Dataset* » qui contient les images des différents Pokémons de la première génération (**151 images**).

b. Puzzle Images Game

Il y a deux tableaux côte à côte, l'un va contenir des cases vides où viendront se loger les parties de l'image à reconstituer en utilisant le *drag & drop*. Le second tableau va contenir les sous parties d'une image sélectionnée de façon aléatoire et disposées de façon aléatoire également.

Le but est de réarranger les sous parties afin de reconstituer l'image d'origine. Un fois que c'est fait, il va y avoir un effet de **transparence** qui cachera le tableau et affichera l'image complète.

Base d'images : elle se trouve dans le dossier « *Puzzle_Image_Game_Dataset* » qui contient quatre dossiers représentant une image/puzzle en particulier. Pour chacune des images, il y a trois sous dossiers correspondant au découpage 3x3, 4x4 et 5x5 selon la difficulté choisie.

Dans les deux cas, il y un seul bouton :

- **Abandonner** : *Il permet d'interrompre la partie et la fenêtre de **Fin de jeu** s'affiche.*

Pour l'aspect aléatoire dans le jeu, il faut utiliser le code suivant :

```
Math.floor(Math.random() * 5)
```

Il permet de générer de façon aléatoire une valeur comprise dans l'intervalle **0** et **4**. Afin d'étendre l'intervalle, il faut juste changer la valeur **5** vers la valeur voulue.

3. Fin de jeu

Ce composant s'affiche dans le *mode contre la montre*. Plus précisément, Quand le temps défini par le joueur au début est écoulé avant que le joueur ne finisse le jeu, ce composant apparaît. Il s'affiche également si l'utilisateur appuie sur le bouton **Abandonner**.

Il y a deux différents boutons :

- **Menu** : Il permet de retourner au composant initial avec l'effet de **transparence**.
- **Rejouer** : Il permet de rejouer une nouvelle partie avec les mêmes paramètres que dans la partie précédente. L'effet utilisé est un simple **affichage**.

4. Score

Dans ce composant, il y a deux tableaux qui sont affichés :

- Le premier tableau représente les résultats pour le *mode normal*,
- Le deuxième tableau représente les résultats pour le *mode contre la montre*.

En ce qui concerne les différents champs dans les tableaux des scores, ils sont définis dans les **Figures 1 & 2**. Il y a également un seul bouton qui est le :

- **Menu** : Il permet de retourner au composant initial avec l'effet de **transparence**.

Dans les tableaux des scores, il faut qu'ils soient triés de façon croissante selon le temps qui a été mis par le joueur pour finir la partie.

FONCTIONNEMENT

En ce qui concerne le fonctionnement des deux jeux sur navigateur, ça va se passer en trois étapes distinctes :

1. *Entrer les informations du joueur et de la partie,*
2. *Jouer la partie avec le mode et la difficulté sélectionnée,*
3. *Voir le score global des autres parties par mode.*

EVALUATION

- Le travail doit être réalisé en équipe de deux étudiants,
- Un seul fichier compressé contenant le *code source* sera remis sur **Léa**,
- Ce travail sera évalué et représentera **15%** de la note finale.
- La date de remise est pour la semaine prochaine **10/10/2019**.