Retour

Le travail est à réaliser individuellement.
Le respect des consignes sera pris en compte pour l'évaluation de ce travail, en plus de la qualité du travail rendu.

Mini-projet JavaScript

Exercice 1 - Sujet

L'objectif est de réaliser un jeu vidéo simple dans lequel le joueur contrôle un vaisseau à l'aide du clavier. Des soucoupes volantes arrivent de la droite de l'écran et le joueur doit les détruire en leur tirant dessus. A chaque tir réussit le joueur marque des points, à l'inverse si un vaisseau parvient à passer sans être détruit, le joueur perd des points.

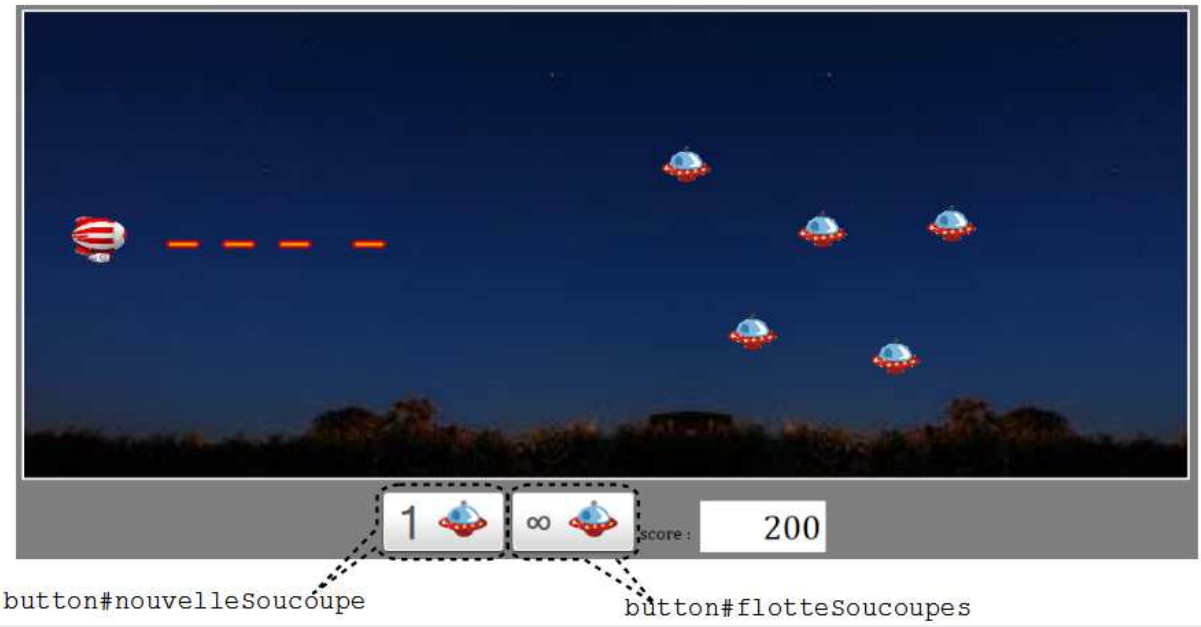

Fig. 1. le jeu à créer en action

Vous devez utiliser les fichiers contenus dans l'archive **fichiers-mini-projet-js2015.zip** fournie. Cette archive contient :

- le fichier **mini-projet-js-2015.html** que vous renommerez en **votreNom-projet.html** mais dont vous ne modifierez pas le contenu,
- le fichier **style/style-projet2015.css**, vous pouvez améliorez si vous le souhaitez,
- le fichier **scripts/projet2015.js** que vous compléterez avec le code javascript que vous écrirez. Toutes les fonctions que vous écrirez dans ce fichier devront être commentées.
- le dossier **images** contenant les images proposées, vous pouvez en utiliser d'autres si vous le souhaitez.

Il est conseillé de traiter les questions dans l'ordre.

Q 1. Le vaisseau du joueur est représenté par l'image **images/vaisseau-ballon-petit.png**. Ce vaisseau se trouve sur la gauche du canvas. Initialement il est positionné horizontalement à *40px* du bord gauche et verticalement à la moitié de la hauteur du canvas.

Réalisez les fonctions nécessaires pour créer l'affichage de la situation initiale du jeu. Vous pouvez utiliser des variables globales pour mémoriser les coordonnées du vaisseau.

Q 2. Le joueur peut déplacer son vaisseau verticalement vers le haut ou le bas à l'aide des touches `flèche haut` et `flèche bas`. Le pas de déplacement est fixe, par exemple *8px*. Evidemment le vaisseau ne peut sortir du canvas.

Faites les définitions nécessaires pour gérer les déplacements du vaisseau du joueur.

Q 3. A chaque fois qu'il appuie sur la `barre espace` le joueur déclenche un nouveau tir. Les tirs se déplacent horizontalement vers la droite, à partir de la position du vaisseau au moment où celui-ci déclenche son tir. Il n'y a pas de limite au nombre de tirs qui existent à un moment donné, mais lorsqu'à la suite de déplacement un tir «sort» des limites du canvas (à droite) il est supprimé. Les tirs sont représentés par l'image **tir.png**.

1. Créez la structure de données qui permet de représenter les objets de type *Tir*.
2. Faites en sorte que l'appui par le joueur de la `barre espace` crée un nouveau tir.
3. Réalisez une fonction qui calcule le déplacement d'un tir, ce déplacement pourra par exemple correspondre à un décalage de *8px* vers la droite.
4. Ajoutez le code nécessaire pour gérer l'animation des tirs.

Q 4. Le joueur doit se défendre contre des soucoupes volantes qui arrivent de la gauche du canvas et se déplacent horizontalement vers la gauche. La position verticale des soucoupes est aléatoire. Il n'y a pas de limite au nombre de soucoupes volantes qui peuvent exister à un moment donné, mais lorsqu'à la suite de déplacement une soucoupe «sort» des limites du canvas (à gauche) elle est supprimée. Les soucoupes sont représentés par l'image **flyingSaucer-petit.png**.

1. Créez la structure de données qui permet de représenter les objets de type *SoucoupeVolante* et ajoutez le code nécessaire pour gérer l'animation des soucoupes. Les soucoupes volantes peuvent par exemple bouger de *3px* vers la gauche à chaque déplacement.
2. Faites en sorte qu'un clic sur le bouton d'id *nouvelleSoucoupe* crée une nouvelle soucoupe volante.

Remarque *L'appui sur la barre espace du clavier a le même effet qu'un clic lorsqu'un bouton est actif. Pour éviter des effets de bord liés au tir et non désiré après avoir cliqué que le bouton d'id nouvelleSoucoupe il faut fait «perdre le focus» à ce bouton. Pour y parvenir, ajouter à la fin de la fonction d'écoute lié à votre bouton la ligne :*

```
1  this.blur();      // fait perdre le focus à l'élément "this"
```

Q 5. Réalisez une fonction *collision* qui prend deux paramètres, le premier est un objet de type *Tir*, le second un objet de type *SoucoupeVolante*. Cette fonction renvoie *true* si et seulement si il y a collision entre les rectangles délimitant les images de chacun de ces objets, le résultat est *false* sinon.

Q 6. Utilisez la fonction *collision* pour réalisez une fonction *tirReussi* qui prend pour paramètre une valeur *t* de type *Tir*. Si il y a collision entre *t* et l'une au moins des soucoupes volantes alors cette fonction a pour résultat une soucoupe volante en collision avec ce tir, sinon le résultat est `undefined` ou `null`.

Q 7. Complétez votre code pour qu'après chaque déplacement d'un objet de type *Tir*, si ce tir entre en collision avec une soucoupe volante, alors le tir et la soucoupe volante sont supprimés.

Q 8. Un tir réussi rapporte *200 points*, une soucoupe volante qui parvient à s'échapper et sort sur la gauche fait perdre *1000 points*. Faites les ajouts nécessaires pour gérer ces points. Le score actualisé doit être affiché comme contenu de l'élément d'id *score*.

Q 9. Utilisez un timer pour qu'un clic sur le bouton d'id *flotteSoucoupes* crée une nouvelle soucoupe volante à intervalle régulier, par exemple toutes les *750ms*. Il y aura cependant qu'une chance (ou risque) sur 2 qu'une nouvelle soucoupe soit effectivement créée.

Un nouveau clic sur ce même bouton devra interrompre l'arrivée de nouvelles soucoupes, jusqu'au prochain clic qui redémarre le processus et ainsi de suite...

Q 10.

Bonus

Vous pouvez complétez votre projet de nombreuses manières, ceci est falcutatif, voici quelques pistes qui peuvent se cumuler... Il est évident que les extensions n'ont de sens que si le reste a été parfaitement réalisé.

- *Lorsqu'une soucoupe volante est touchée par un tir, elle n'est pas supprimée immédiatement, simplement ses déplacements suivants se font «vers le bas» à partir de la position qu'elle occupée au moment du tir jusqu'à atteindre le «bas» du canvas où soit elle est supprimée, soit elle reste immobile jusqu'à la fin de la partie.*

- *La probabilité qu'une nouvelle soucoupe qui est de `0.5` à la question 9 augmente progressivement avec le score du joueur, par exemple `0,05` tous les 1000 points (ou toute autre progression).*
- *Le joueur n'a pas le droit de laisser s'échapper plus de trois soucoupes volantes. A la troisième le jeu s'arrête : plus de nouvelle soucoupe et les touches du clavier reste sans effet. Le nombre de soucoupes volantes restantes peut être annoncé au joueur sous la forme d'images affichées quelque part sur la page, elles disparaissent au fur et à mesure où l'une s'échappe.*
- *La gestion d'un «meilleur score» sur un navigateur donné peut être réalisée à l'aide des fonctionnalités appelées «Web Storage». Après une recherche sur internet, utilisez l'objet `localStorage` et ses fonctions `setItem` et `getItem` pour stocker ce meilleur score, avec éventuellement en plus le nom du joueur. Le déclenchement de cette mémorisation peut être lié à un clic sur un bouton qui marquera la fin de la partie, ou si l'on a fait l'extension précédente après la perte des trois soucoupes.*
- *Ou à vous d'imaginer et de proposer une autre variante/extension...*

*Vous préciserez dans votre fichier **lisezmoi.txt** si vous avez mis en place des fonctionnalités supplémentaires, en précisant lesquelles.*

Q 11. Rendez votre travail sous la forme d'une archive. Le nom de cette archive sera ***votreNom*-projetJS.zip**. Cette archive contiendra un répertoire dont le nom sera ***votre-nom*-projet**. Le contenu de ce répertoire sera organisé ainsi :

- un fichier **lisezmoi.txt** avec votre nom et prénom, votre numéro de groupe, ce fichier mentionnera pour chacune des questions précédentes si elle a été traitée ou non et les éventuels problèmes de votre solution par rapport au cahier des charges demandé,
- les fichiers mentionnés en introduction.

Tous vos fichiers seront codés en UTF-8. Les noms des fichiers (y compris leurs extensions) seront en minuscules.