

## Exercice guidé – takeYourTime-Ex 20220501

**Description du site : Un jeu d'arcade avec un personnage principal qui essaie de gagner le plus de points possible tout en évitant le contact avec les ennemis**

**Objectifs de l'exercice:**

- Créer une minuterie pour le jeu
- Affichage d'un message image pour la fin du jeu

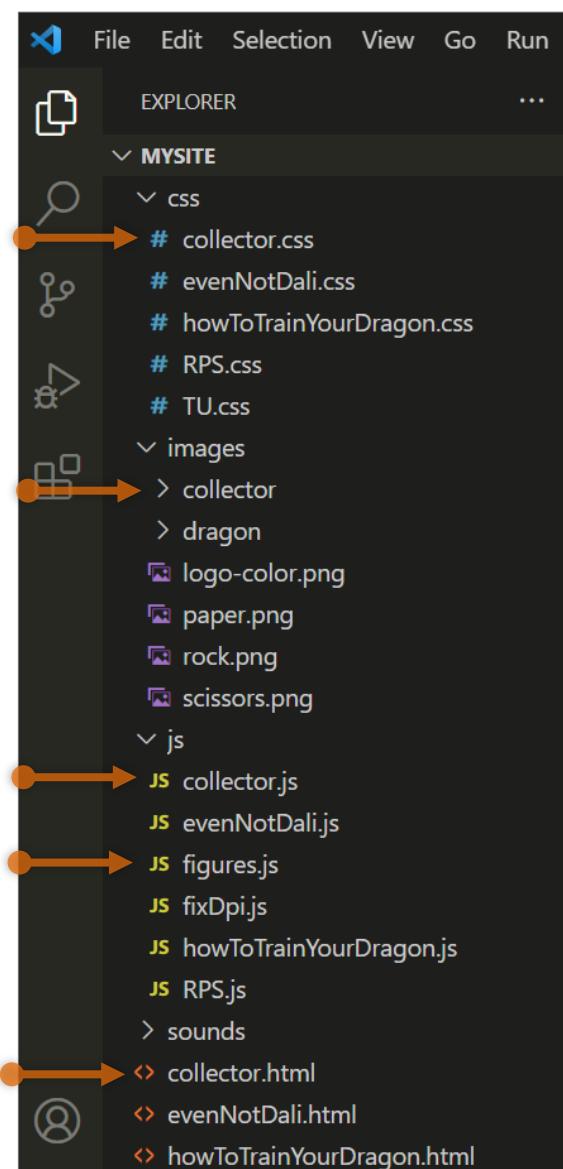
### Créer un autre nouveau projet (sur le site principal)

**Préparer l'infrastructure à partir de ce qui existe déjà**

**Ajouter les fichiers html et css sur notre site Web principal**

1. Clique [ici](#) pour télécharger les fichiers suivants :
  - 1.1. [collector.css](#)
  - 1.2. [collector.html](#)
  - 1.3. [collector.js](#)
  - 1.4. [figures.js](#)
  - 1.5. Le dossier images [collector](#)
2. Transfère les fichiers dans le projet comme suit :
  - 2.1. Place le fichier [css](#) dans le dossier [css](#)
  - 2.2. Place les [2](#) fichiers [js](#) dans le dossier [js](#)
  - 2.3. Place le fichier [html](#) dans le dossier du projet
  - 2.4. Place le dossier [images](#) dans le dossier [images](#)

Note, le fichier [html](#) est basé sur le modèle que tu as vu lors de notre précédente réunion et il contient déjà les éléments requis pour ce projet



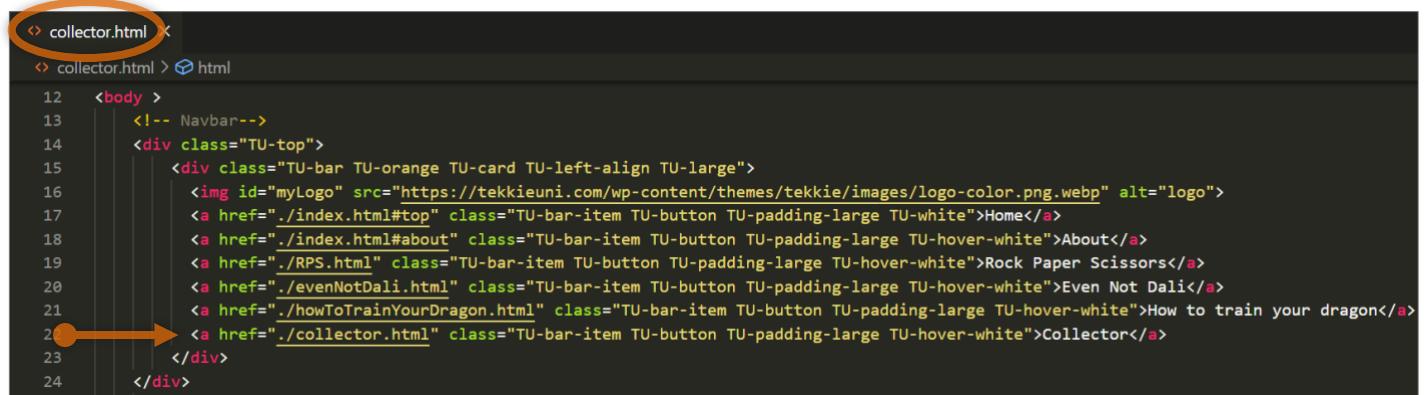
**continue sur la prochaine page**

## Mise à jour du menu de navigation dans les fichiers html

### Ajouter un onglet dans le menu de navigation pour la nouvelle page

3. Dans le fichier `collector.html`, trouve le `div` du menu de navigation au début du tag `body`
4. Et ajoute un tag `a` qui se connecte et envoie l'utilisateur à la nouvelle page Web

Remarque que tu peux copier un tag existant et le modifier en conséquence



```

12 <body>
13   <!-- Navbar-->
14   <div class="TU-top">
15     <div class="TU-bar TU-orange TU-card TU-left-align TU-large">
16       
17       <a href=".index.html#top" class="TU-bar-item TU-button TU-padding-large TU-white">Home</a>
18       <a href=".index.html#about" class="TU-bar-item TU-button TU-padding-large TU-hover-white">About</a>
19       <a href=".RPS.html" class="TU-bar-item TU-button TU-padding-large TU-hover-white">Rock Paper Scissors</a>
20       <a href=".evenNotDali.html" class="TU-bar-item TU-button TU-padding-large TU-hover-white">Even Not Dali</a>
21       <a href=".howToTrainYourDragon.html" class="TU-bar-item TU-button TU-padding-large TU-hover-white">How to train your dragon</a>
22       <a href=".collector.html" class="TU-bar-item TU-button TU-padding-large TU-hover-white">Collector</a>
23     </div>
24   </div>

```

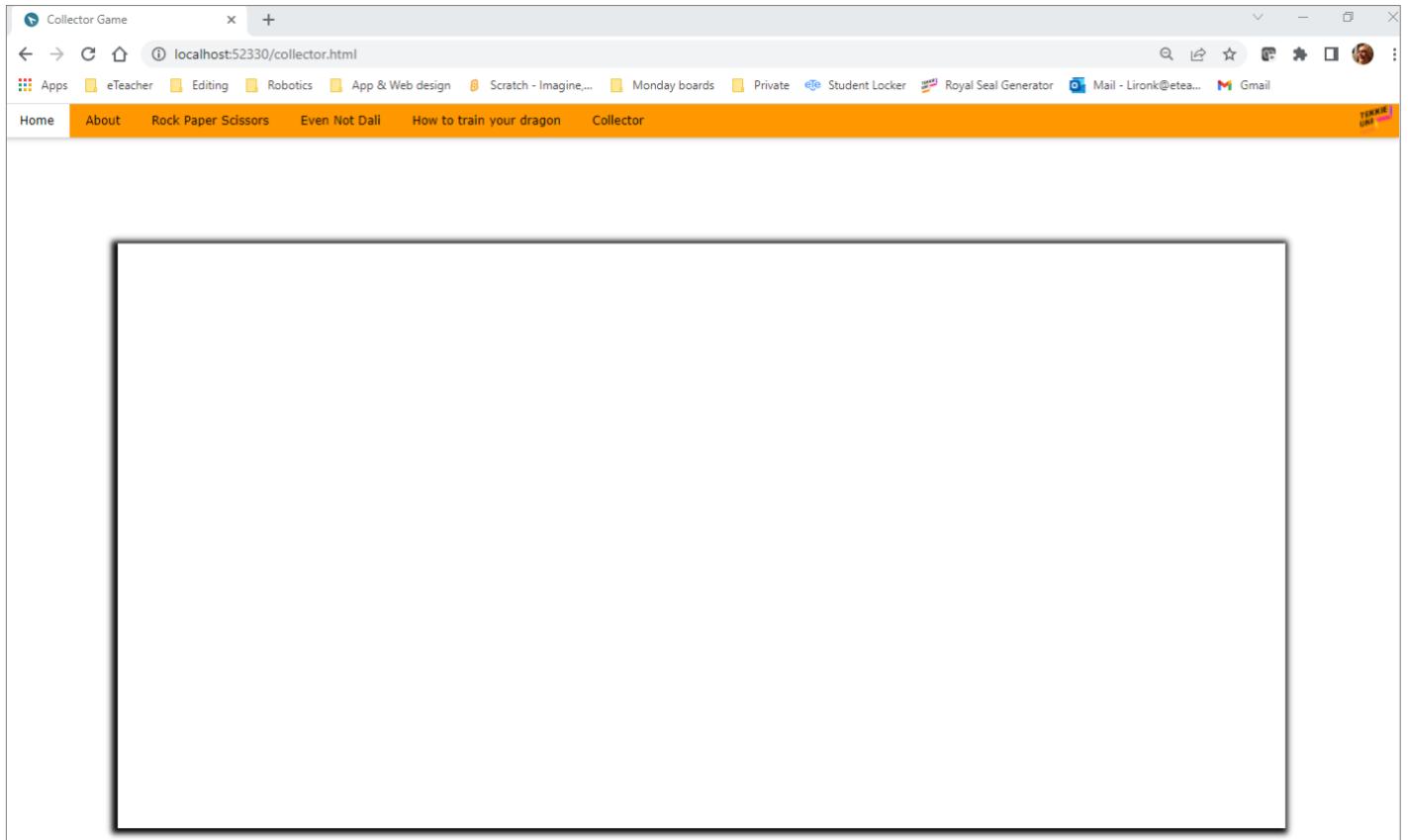
### Copier le lien et l'ajouter au reste des pages du site Web

5. Toujours dans le fichier `collector.html`, copie le tag qui contient le lien vers la nouvelle page (le lien que tu as ajouté dans le paragraphe précédent)
6. Colle le lien que tu as copié dans tous les fichiers `html` des projets

**continue sur la prochaine page**

Exécute le site Web et vérifie les transitions entre toutes les pages  
Vérifie la nouvelle page Web que nous avons ajoutée

Le résultat devrait ressembler à l'image ci-dessous



continue sur la prochaine page

## Créer, afficher et activer la minuterie

### Annoncer des variables

7. En haut de l'éditeur de code pour `collector.js`, après l'appel à la fonction `updateCanvasSize`, annonce une nouvelle variable de programme appelée `time` et définis sa valeur comme étant 10

Remarque, plus tard, tu pourras modifier le nombre initial de secondes de la minuterie

Le résultat devrait ressembler à l'image ci-dessous

```
JS collector.js ×
js > JS collector.js > ...
1 const CANVAS = document.getElementById('canvas');
2 const CTX = CANVAS.getContext('2d');
3
4 updateCanvasSize();
5 window.onresize = updateCanvasSize();
6
7 ➔ Let time = 10;
```

### Créer une fonction pour définir la minuterie

8. Toujours dans le fichier `collector.js`, crée une nouvelle fonction appelée `timer` et ajoutes-y les commandes suivantes :

- 8.1. Une règle qui vérifie si la variable `time` est supérieure à zéro

- **si la réponse est oui**

- 8.1.1. Une commande pour soustraire 1 de la variable `time`

- **Si la réponse est non**

- 8.1.2. Un appel pour la fonction `gameOver`

Le résultat devrait ressembler à l'exemple de droite

```
JS collector.js ×
js > JS collector.js > ...
20 function timer() {
21   if(time > 0){
22     time--;
23   }
24   else{
25     gameOver();
26   }
27 }
```

**continue sur la prochaine page**

## Créer un élément image et définir son image source

9. En haut de l'éditeur de code pour `collector.js`, dans la zone de la variable constante (avant l'appel de la fonction `updateCanvasSize`) ajoute les commandes suivantes :

9.1. Annonce ces **3** variables :

9.1.1. Une variable constante pour stocker la largeur de l'image game over

9.1.2. Une variable constante pour stocker la hauteur de l'image game over

9.1.3. Une variable pour stocker un nouveau type objet `Image`

9.2. Une commande pour définir la source d'image du nouvel élément `img` comme étant l'une des images game over du dossier `collector`

10. Dans la zone des variables du programme (après l'appel de la fonction `updateCanvasSize`) annonce ces **2** variables :

10.1. Une variable pour stocker l'emplacement de l'image sur l'axe **X**, au milieu de l'axe **X**

10.2. Une variable pour stocker l'emplacement de l'image sur l'axe **Y**, au milieu de l'axe **Y**

Remarque que l'ordre et l'emplacement de la commande dans le fichier ont un sens !

Assure-toi que l'appel de la fonction qui met à jour la taille du canvas apparaît avant les variables du programme qui changent pendant l'exécution du programme

Le résultat devrait ressembler à l'image ci-dessous

```
JS collector.js ×
js > JS collector.js > ...
1  const CANVAS = document.getElementById('canvas');
2  const CTX = CANVAS.getContext('2d');
3  const GAME_OVER_IMG_WIDTH = 960;
4  const GAME_OVER_IMG_HEIGHT = 574;
5  const GAME_OVER_IMG = new Image();
6  GAME_OVER_IMG.src = "./images/collector/GameOverMessage.png";
7
8  updateCanvasSize();
9  window.onresize = updateCanvasSize();
10
11 let gameOverImgX = CANVAS.width / 2 - GAME_OVER_IMG_WIDTH / 2;
12 let gameOverImgY = CANVAS.height / 2 - GAME_OVER_IMG_HEIGHT / 2;
13 let time = 10;
```

continue sur la prochaine page

## Créer une fonction pour afficher le message de fin du jeu

11. Crée la nouvelle fonction appelée `gameOver` et ajoutes-y les commandes suivantes :

11.1. Une commande pour effacer tous les pixels du canvas

11.2. Une commande pour dessiner l'image "game over" au centre du canvas

Le résultat devrait ressembler à l'image ci-dessous

```
JS collector.js ×
js > JS collector.js > ...
35  function gameOver() {
36      CTX.clearRect(0, 0, CANVAS.width, CANVAS.height);
37      CTX.drawImage(GAME_OVER_IMG, gameOverImgX, gameOverImgY, GAME_OVER_IMG_WIDTH, GAME_OVER_IMG_HEIGHT);
38 }
```

## Créer et définir la fonction `gameLoop`

12. Sous la fonction `writeText`, crée la nouvelle fonction appelée `gameLoop` et ajoutes-y les commandes suivantes :

12.1. Une commande pour effacer tous les pixels du canvas

12.2. Une commande pour activer la fonction `writeText` qui écrit et affiche la minuterie au joueur

12.3. Un appel pour la fonction `timer`

12.4. Un appel à la fonction `setTimeout` avec les deux arguments suivants :

12.4.1. La fonction `requestAnimationFrame`

12.4.2. Chronométrier l'appel toutes les `1000` millisecondes (une seconde)

Le résultat devrait ressembler à l'image ci-dessous

```
JS collector.js ×
js > JS collector.js > ...
26  function gameLoop() {
27      CTX.clearRect(0, 0, CANVAS.width, CANVAS.height);
28      writeText("Time: " + time, "black", 50, undefined, "bold", 1000, 70);
29      timer();
30      setTimeout(function() {requestAnimationFrame(gameLoop);}, 1000);
31 }
```

continue sur la prochaine page

## Activation de la minuterie dès l'ouverture de la page Web

13. Après avoir défini la fonction `gameLoop`, ajoute un appel pour la fonction `gameLoop`

```
JS collector.js X
js > JS collector.js > ...
26   function gameLoop() {
27     CTX.clearRect(0, 0, CANVAS.width, CANVAS.height);
28     writeText("Time: " + time, "black", 50, undefined, "bold", 1000, 70);
29     timer();
30     setTimeout(function() {requestAnimationFrame(gameLoop);}, 1000);
31   }
32 → gameLoop();
```

C'est le bon moment pour vérifier le site !

Le résultat devrait ressembler à l'image ci-dessous



### Avancé

- Réfléchis et vérifie ce que nous devons changer dans le code pour que l'image du jeu soit plus petite
- Recherche ou crée une nouvelle image de game over et remplace l'image dans le code par ta nouvelle image

Fais attention et assure-toi d'adapter la taille de l'image que tu as choisie à l'écran

- Essaie de concevoir la minuterie comme tu le souhaites

**Indice:** Change les arguments dans l'appel de la fonction