

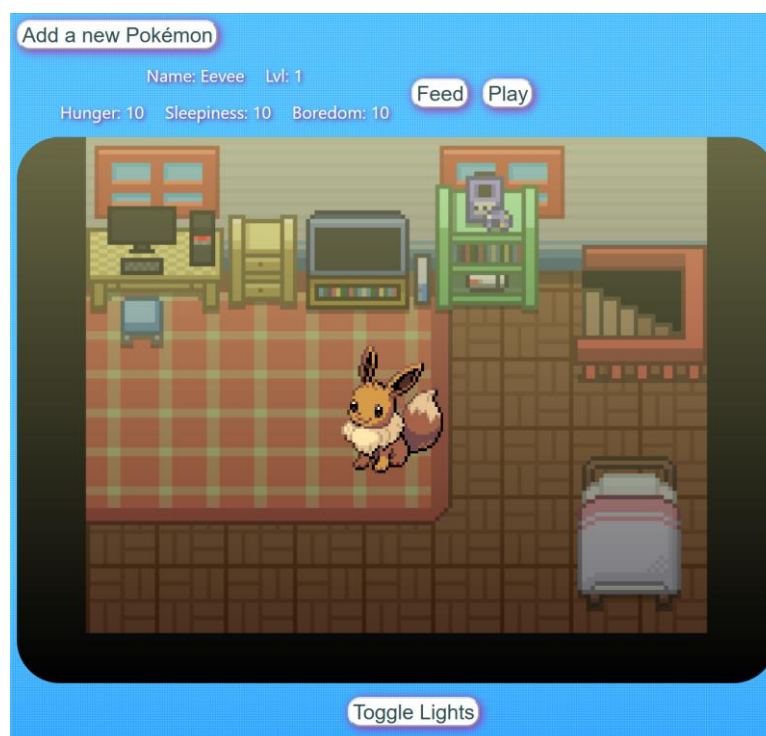
## TD2 – OBJETS

Le but de ce dernier TD est de faire usage des objets/classes en JS.

### EXERCICE TAMAGOTCHI

Définition Wikipedia : *Un Tamagotchi (たまごっち?) est un animal de compagnie virtuel japonais, créé en 1996 par la société japonaise Bandai1. Son nom est un mot-valise créé à partir des mots « œuf » (卵, tamago?) et « montre » (en fait la retranscription en katakana du mot anglais watch) (ウォッチ, watchi?). Le jeu consiste à simuler l'éducation d'un animal à l'aide d'une console miniature, de la taille d'une montre, dotée d'un programme informatique.*

Nous allons reprendre ce concept ici mais sous forme d'application en JS dans un navigateur en utilisant le prétexte des Pokémons.



La particularité ici étant le fait qu'on va pouvoir gérer plusieurs Pokémons. Vous devrez faire usage de la

Voici le cahier des charges que vous devrez réaliser :

- Vous devez représenter vos Pokémons sous forme de classe en JavaScript, avec tous les attributs et méthodes relatives à sa gestion. Un Pokémon dans notre modèle possède au minima les caractéristiques suivantes :
  - Son numéro dans le pokédex (identifiant allant de 1 à 151)
  - Un prénom
  - Un niveau (valeur par défaut 1)
  - Sa stat de faim « hunger » (valeur par défaut 5)
  - Sa stat de fatigue « sleepiness » (valeur par défaut 0)
  - Sa stat d'ennui « boredom » (valeur par défaut 3)
  - Ses coordonnées x/y dans la maison

- Vous pouvez rajouter autant d'attributs que nécessaire (par exemple, pour stocker des références DOM)
- A la construction d'un Pokémon, l'utilisateur devra saisir un numéro entre 1 et 151 pour choisir un Pokémon, et saisir son prénom. Tous les autres attributs prendront les valeurs par défaut.
- Dès qu'un Pokémon est construit, il devra être placé dans la maison (donc dans le DOM), et une nouvelle ligne avec ses stats et actions devra être rajoutée.
  - Pour afficher son sprite (image), vous devez simplement prendre la balise img déjà présente dans le fichier et intégrer le numéro du Pokémon dans l'URL :  
src=<https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/versions/generation-v/black-white/animated/133.gif>
  - Dans un premier temps, placez-le à des coordonnées aléatoires.
  - Dans un second temps, faites le bouger de façon aléatoire dans la pièce.
- Les stats et attributs du pokémon vont évoluer de la façon suivante en fonction du temps :
  - Un Pokémon monte de niveau toutes les minutes
  - Un Pokémon gagne 1 en « hunger » toutes les 10 secondes
  - Un Pokémon gagne 1 en « sleepiness » toutes les 30 secondes
  - Un Pokémon gagne 1 « boredom » toutes les 15 secondes
  - Lorsque vous nourrissez un Pokémon, il perd 2 en « hunger » et gagne 1 en « sleepiness »
  - Lorsque vous jouez avec un Pokémon, il perd 2 en « boredom » et gagne 1 en « sleepiness »
- Il est possible d'allumer et éteindre la pièce avec le bouton « Toggle Lights ». Lorsque vous éteignez la lumière, vous pouvez changer la classe de `<div class="on" id="pet-image">` en « off ». Lorsque la lumière est éteinte, les Pokémon doivent s'arrêter de bouger dans la pièce, et ils perdent 1 de « sleepiness » toutes les 5 secondes. Le statut de la lumière peut être stocké comme attribut static dans la classe.
- Lorsque n'importe laquelle des stats du pokémon dépasse 10, le Pokémon s'enfuit de votre maison ☹