M3203

Programmation objet et évènementielle

25 octobre 2018

Séance n° 5	**Évaluation finale – Groupe TP3**

Objectif :
Déposez sur UniversiTICE dans le dépôt correspondant à votre groupe trois fichiers contenant les codes sources résolvant les 3 exercices ci-dessous. Les fichiers devront être nommés par l'intitulé de votre groupe, votre nom et le numéro de l'exercice, cad TP3-Nom-1.html, TP3-Nom-2.html et TP3-Nom-3.html.
Les supports de cours accessibles directement sur le site du cours et la documentation sont autorisés.
Vous accorderez un soin particulier à la présentation de votre code : respectez les conventions de nommage des variables, l'indentation, commentez votre code quand nécessaire.
Assurez-vous de travailler avec un navigateur à jour.

1 Champs vides ? (5 points)

Vous allez réaliser un script qui va vérifier si les champs de saisie d'une page sont bien remplis.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Champs vides</title>
  </head>

  <body>
    <label>Nom : </label>
    <input type="text" value="Le chat"> <br>
    <label>Prénom : </label>
    <input type="text" value=""> <br>
    <label>Age : </label>
    <input type="text" value="8"> <br>
    <label>Occupations : </label>
    <input type="text" value="Manger et dormir"> <br>
    <label>Robe : </label>
    <input type="text" value="Tigré"> <br>
    <label>Vacciné ? : </label>
    <input type="text" value="Pitié pas le véto"> <br>
    <label>Jouets : </label>
    <input type="text" value=""> <br>
    <label>Préfère : </label>
    <input type="text" value="Sardines"> <br>

    <button id="go">Go !</button>

    <p id="message">

    </p>

    <script type="text/javascript">
    </script>
  </body>
</html>
```

Le document HTML est téléchargeable ici (Clic droit > télécharger la cible du lien sous).

1. Créez le fichier ci-dessus et enregistrez-le
2. Créez la fonction `valid` qui va :
   — récupérer les champs de saisie de la page **0.5 pt**
   — parcourir ces champs **1 pt**
   — et afficher le message "Le champ X est vide" dans l'élement d'identifiant `message` si le champ est vide **1 pt**
   — ou colorer le champ en vert s'il est rempli **0.5 pt**
3. Ajoutez un écouteur d'évènements sur le bouton d'identifiant `go` qui appelera la fonction `valid` lors d'un clic **1 pt**

Voici ce que vous pouvez obtenir :

Nom : Le chat
Prénom :
Age : 8
Occupations : Manger et dormir
Robe : Tigré
Vacciné ? : Pitié pas le véto
Jouets :
Préfère : Sardines
Go !

Le champ 1 est vide
Le champ 6 est vide

Présentation du code : **1 pt**

2 Animaux (5 points)

Vous allez écrire des classes permettant de gérer des animaux pour un jeu vidéo. Dans ce jeu, un mammifère est caractérisé par un nom et un nombre de points de vie. Un chat est un mammifère qui possède en outre un stock de ronrons (oui).

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Animal Game</title>
  </head>

  <body>

    <p>Caractéristiques des animaux :</p>

    <script type="text/javascript">
    </script>
  </body>
</html>
```

Le document HTML est téléchargeable ici (Clic droit > télécharger la cible du lien sous).

1. Créez le fichier ci-dessus et enregistrez-le.
2. Écrivez la classe `Mammifere` qui possède un constructeur prenant en paramètre le nom de l'animal et son nombre de points de vie et initialise les propriétés `nom` et `ptVie`. **1 pt**

3. Ajoutez une méthode `affiche` à votre classe `Mammifere` qui affiche une description dans le document HTML (Exemple : Mammifère X a Y points de vie). **1 pt**
4. Un chat est un mammifère qui possède en outre un stock de ronrons, une valeur numérique. Écrivez la classe `Chat` et son constructeur. **0.5 pt**
5. Ajoutez une méthode `affiche` à votre classe `Chat` qui affiche une description dans le document HTML. **0.5 pt**
6. Créez un objet de type `Mammifere` et un objet de type `Chat`. Affichez leur description. **1 pt**

Présentation du code : **1 pt**

3 Tami (10 points)

Tama n'a pas bien supporté ses dernières aventures en TP... Vous allez plutôt écrire un nouveau jeu avec sa copine Tami. Tami a besoin de manger et de jouer. Pour ça on peut lui donner un bol de croquettes pour la nourrir, des jouets pour la distraire ou un os pour faire les deux à la fois.



Le document HTML est téléchargeable ici (Clic droit > télécharger la cible du lien sous).

3.1 Évolution de Tami

Dans la première partie, vous allez créer la classe `Tami` qui gèrera la vie de Tami et son évolution : son besoin en nourriture et son niveau de distraction.

1. Créez le fichier ci-dessus et enregistrez-le.
2. Déclarez la classe `Tami` et son constructeur qui ne prend aucun paramètre. **0.5 pt**
3. Initialisation des propriétés dans le constructeur : **1,5 pt**
   (a) La propriété `hunger` servira à stocker l'élément HTML indiquant l'appétit de Tami. La propriété `entertainment` servira à stocker l'élément HTML indiquant le niveau de distraction de Tami. La propriété `tamiImg` servira à stocker l'élément HTML image représentant Tami. Stockez dans les propriétés `hunger, entertainment, tamiImg` les éléments HTML d'identifiant `hunger, entertainment, tami`.
   (b) Donnez les conditions de départ de Tami : le contenu textuel de `hunger` est de 0, celui de `entertainment` de 100% et la source de l'image est `http://chloecabot.com/mmi/M3203/img/tami_alive.png`.
   (c) Initialisez la propriété `alive` à `True`. Elle indiquera si Tami est vivante.
   (d) Initialisez la propriété `intervalID` à `0`.
4. Écrivez la méthode `life` qui gérera l'évolution de Tami lors d'une étape de sa vie : **1 pt**
   (a) Son appétit gagne 5 points
   (b) Son niveau de distraction perd 5 points
   (c) Si l'appétit de Tami est supérieur ou égal à 70 ou si son niveau de distraction est inférieur ou égal à 50, elle change d'apparence (url : `http://chloecabot.com/mmi/M3203/img/tami_angry.png`)

(d) Sinon elle a une apparence normale (url : http://chloecabot.com/mmi/M3203/img/tami_alive.png)

(e) Si son appétit est de 100 ou son niveau de distraction nul, Tami meurt (url : http://chloecabot.com/mmi/M3203/img/tami_dead.png)

5. Écrivez la méthode `start` qui lance la méthode `life` toutes les 500ms. **1 pt**
6. Ajoutez un appel à la méthode `start` dans le contructeur de `Tami`. **0.5 pt**
7. Écrivez la méthode `stop` qui va : **1 pt**
   (a) Stopper l'intervalle.
   (b) Donner la valeur `False` à la propriété `alive`.
   (c) Créer un nouvel élément HTML paragraphe qui sera le message de fin de partie. Le paragraphe possède la classe CSS `message` et un contenu textuel (choisissez ce que vous voulez).
   (d) Ajouter ce paragraphe à son parent dans le document HTML : l'élément HTML d'identifiant `end`.
   (e) Ajoutez un appel à la méthode `stop` dans la méthode `life`.

3.2 Mise en place des écouteurs d'évènements

1. Dans le script principal, ajoutez un écouteur d'évènements au bouton d'identifiant `start` lors d'un clic. La fonction appelée devra effacer le contenu textuel de l'élément d'identifiant `end` et créer un nouvel objet de type `Tami`. **1 pt**
2. Ajoutez à votre classe `Tami` la méthode `feed` qui permettra de modifier l'appétit de Tami. Cette méthode prend en paramètre un nombre entier et l'additionne à l'appétit de Tami seulement si elle est vivante. **0.5 pt**
3. Ajoutez à votre classe `Tami` la méthode `play` qui permettra de modifier le niveau de distraction de Tami. Cette méthode prend en paramètre un nombre entier et l'additionne au niveau de distraction de Tami seulement si elle est vivante. **0.5 pt**
4. Dans le constructeur de la classe `Tami`, ajoutez des écouteurs d'évènements : **1.5 pt**
   (a) Au bouton d'identifiant `feedbone_button` lors d'un clic. La fonction exécutée devra diminuer l'appétit de 5 et augmenter le niveau de distraction de 5 (utilisez vos méthodes `feed` et `play`).
   (b) Au bouton d'identifiant `feedbowl_button` lors d'un clic. La fonction exécutée devra diminuer l'appétit de 15.
   (c) Au bouton d'identifiant `entertainment_button` lors d'un clic. La fonction exécutée devra augmenter le niveau de distraction de 15.

Bonus Comment pourrait-on améliorer notre méthode `life` ? Indiquez moi votre réponse en commentaire dans votre code. **1 pt**

Présentation du code : **1 pt**