Correction du TP2

Quentin Bouillaguet

16 mars 2020

Sujet, Cours et Rappels

tp2.html: Entête <head>

Information générales (metadonnées) sur la page: titre, encodage, ressources, ...

Éléments de métadonnées

- <title>: titre de la page
- <script>: intégration de Javascript
- <link>: liaison d'une ressource externe
  - feuille de style:

```
<link rel="stylesheet" href="tp2.css" />
```

- <meta>: autres métadonnées
  - encodage:

```
<meta charset="utf-8 />
```

Liste exhaustive des éléments

- Voir developer.mozilla.org/fr/docs/Web/HTML/Element

tp2.html: Corps

Contenu principal du document HTML

Division du contenu

- <header>, <article>, ...: éléments sémantiques
- : conteneur générique

Contenu textuel

- <p>: paragraphe
- lien: hyperlien

Attribut id

- Identifie de manière **unique** un élément du document

```
<div id="ciel"></div>
```

Liste exhaustive des éléments

- Sectionnement du contenu, Contenu textuel et Sémantique du texte en ligne
- Voir developer.mozilla.org/fr/docs/Web/HTML/Element

`tp2.html`: HTML5

Liste exhaustive des élements

- Voir developer.mozilla.org/fr/docs/Web/HTML/Element

Pour aller plus loin...

- developer.mozilla.org/fr/docs/Web/HTML
- www.w3schools.com/html/

tp2.css: Feuille de style – CSS3

- Utilisé pour décrire la présentation d'un document HTML

```
s1, s2, ... {
  prop1: valeur1;
  prop2: valeur2;
  ...
}
```

- **Selecteurs**: `s1`, `s2`, `...`
- **Règles**: `prop1: valeur1`, `prop2: valeur2`, `...`
- Les **sélecteurs** définissent les éléments sur lesquels s'applique un ensemble de **règles** CSS.
- **Héritage**: Toute propriété non spécifiée pour un élément est héritée de son élément parent.

tp2.css: Feuille de style – Sélecteur

Sélecteur de type

- Cible les éléments correspondant au nom indiqué

```
p {
  color: white;
}
```

Sélecteur d'identifiant (#id)

- Cible les éléments en fonction de la valeur de leur attribut `id`

```
#ciel {
  /* ... */
  background-color: #A0F0FF;
}
```

Autres sélecteurs

- Voir developer.mozilla.org/docs/Web/fr/CSS

`tp2.css`: Feuille de style – Règles

Propriétés de couleurs

- `color`: couleur du texte
- `background-color`: couleur de l'arrière plan

Propriétés de taille

- `width`, `height`: hauteur et largeur
- `max-width`: largeur maximale
- Exprimé en `px`, `%` (par rapport à la taille du parent), ...

Propriété `position`

- `absolute`: par rapport à la position `(0, 0)` du parent
- `relative`: par rapport aux éléments frères
- Placement vertical avec `top` et `bottom`, horizontal avec `left` et `right`
- Voir developer.mozilla.org/fr/docs/Web/CSS/position

`tp2.css`: Feuille de style – CS3

Attention: les noms de l'API DOM JavaScript peuvent différer des noms standards CSS.

- Exemple: `zIndex` en Javascript, `z-index` en CSS

Liste des sélecteurs

- Voir developer.mozilla.org/docs/Web/fr/CSS

Liste exhaustive des propriétés

- Voir developer.mozilla.org/fr/docs/Web/CSS/Reference

Pour aller plus loin...

- Voir developer.mozilla.org/fr/docs/Web/CSS
- Voir www.w3schools.com/css/

tp2.js: Script d'animation

```
function move() {
  num += 1;
  img.src = 'img/tux-walk-right-' + (num % 6) + '.png';
  img.style.left = position + 'px';
}
```

- Animation "image par image":
  - Compteur d'animation `num`
  - Modification de la ressource liée à l'image (avec `img.src`)
    - `'img/tux-walk-left-<num>.png'`: 
    - `'img/tux-walk-right-<num>.png'`: 
- Positionnement **absolu** par rapport au parent
  - `#tux { position: absolute; left: 50px; top: 150px }`
  - **origine**: coin supérieur gauche
  - modification de la position horizontale via le DOM (avec `img.style.left`)

Correction

tp2.js: Affichage gauche et droite

- Direction courante du pingouin: `left` ou `right`

```
var direction = 'right'; // ou 'left'
```

- Dessin en fonction de l'état courant du pingouin (`direction`, `num` et `position`)

```
function draw() {
  img.src = 'img/tux-walk-' + direction + '-' +
    (num % 6) + '.png';
  img.style.left = position + 'px';
}
```

`tp2.js`: Animation par timer

- Fonction d'animation
  - `update()`: mise à jour de l'état du pingouin selon les entrées utilisateur
  - `draw()`: dessin du pingouin (voir diapositive précédente)
- Programmée 20 fois par seconde
  - `setInterval(animate, 50)` appelle `animate` toutes les 50ms au minimum

```
function animate() {
  update();
  draw();
}
setInterval(animate, 50);
```

tp2.js: Mise à jour de l'état du pingouin

- Conservation de l'état des entrées utilisateur

```
var isLeft = false;
var isRight = false;
```

- Conservation de la vitesse de déplacement

```
var speed = 2;
```

- Mis à jour par les entrées utilisateurs (clavier, souris, boutons, ...)

tp2.js: Mise à jour de l'état du pingouin

```
function update() {
  if (isLeft) {
    direction = 'left';
    position -= speed;
    num += 1;
  } else if (isRight) {
    direction = 'right';
    position += speed;
    num += 1;
  }
}
```

tp2.js: Limitation des déplacements

- Dimension des éléments:
  - `offsetWidth`/`offsetHeight`: dimensions totales
  - `clientWidth`/`clientHeight`: dimensions sans les bordures, marges, ...
  - Voir "Déterminer les dimensions des éléments"

```
var cadre = document.getElementById("cadre");
function update() {
  if (isLeft && position >= 0) {
    // ...
  } else if (isRight && position <
              cadre.offsetWidth - img.offsetWidth) {
    // ...
  }
}
```

tp2.js: Contrôle du pingouin

```
function stop() {
  speed = 0;
}
document.getElementById("stopButton")
        .addEventListener('click', stop, false);

function fast() {
  speed += 2;
}
document.getElementById("speedButton")
        .addEventListener('click', fast, false);
```

tp2.js: Gestion des entrées clavier

- Mise à jour des états selon les entrées utilisateur:

```
function press(e) {
  if (e.keycode == 37) { isLeft = true; }
  if (e.keycode == 39) { isRight = true; }
}
window.addEventListener('keydown', press, false);

function release(e) {
  if (e.keycode == 37) { isLeft = false; }
  if (e.keycode == 39) { isRight = false; }
}
window.addEventListener('keyup', press, false);
```