

## Activités d'auto-évaluation

Il est fondamental de garder à l'esprit que ce n'est pas en écoutant un cours qu'un élève apprend, mais que c'est en faisant, en étant actif. Lors d'un apprentissage en ligne, les activités d'auto-évaluation permettent précisément de le faire, même si ce ne sont pas les seules.

Une des principales caractéristiques des cours en ligne est que les apprenants n'ont pas de contact direct avec l'enseignant et ne peuvent interagir avec lui que de manière beaucoup plus limitée qu'en présentiel. Ils ne peuvent donc pas l'interroger lorsqu'ils ont des doutes ou des questions.

Il est donc fondamental, lors de la conception des cours en ligne, de prévoir de multiples activités d'auto-évaluation afin que chaque apprenant puisse vérifier sa compréhension des notions qui lui sont présentées.

Ces activités doivent être proposées après chaque nouvelle vidéo ou page de cours introduisant de nouveaux concepts. Il n'est pas nécessaire que les notes obtenues soient mémorisées. Sans doute même est-il préférable que seul l'apprenant en ait connaissance afin qu'il ne craigne pas qu'elles soient utilisées pour son évaluation finale.

Voici quatre exemples d'activités d'auto-évaluation faciles à construire pour un enseignant s'il dispose des outils appropriés :

- Les exercices de glisser-déposer sur image permettent aux apprenants de vérifier s'ils peuvent s'orienter dans un croquis ou un graphique. Ils fonctionnent par simple glisser-déposer ;
- Les textes à trous avec ou sans image contextuelle demandent un peu plus de concentration de la part de l'apprenant, mais sont très fructueux pour s'assurer que les concepts difficiles sont bien compris ;
- Les exercices de catégorisation complètent bien les deux activités précédentes : les apprenants organisent les éléments en catégories et apprennent ainsi à distinguer leurs caractéristiques ;
- Enfin, les questions à choix unique ou multiples leur permettent de tester leurs connaissances de manière assez large, mais ce ne sont pas des activités très conviviales. Elles sont par ailleurs très utilisées pour les évaluations partielles ou finales des apprenants.

## Outils de création d'activités d'auto-évaluation

Il existe de nombreux environnements de développement de telles activités, mais la plupart sont payantes et leur déploiement peut poser divers problèmes.

Il nous a donc semblé justifié de développer des solutions faciles à mettre en œuvre et libres de droit, diffusées comme ressources éducatives libres.

Nous avons partagé début novembre 2023 un ensemble correspondant aux trois premiers types d'activité indiquées ci-dessus, utilisant la technologie H5P, mais cette solution pose quelques difficultés de déploiement.

C'est pourquoi nous en avons préparé un second ensemble beaucoup plus léger, développé lui aussi en html 5.

## Exercices de glisser-déposer sur image

## Interface graphique

L'écran de l'exercice de glisser-déposer sur image se présente comme indiqué figure 1.

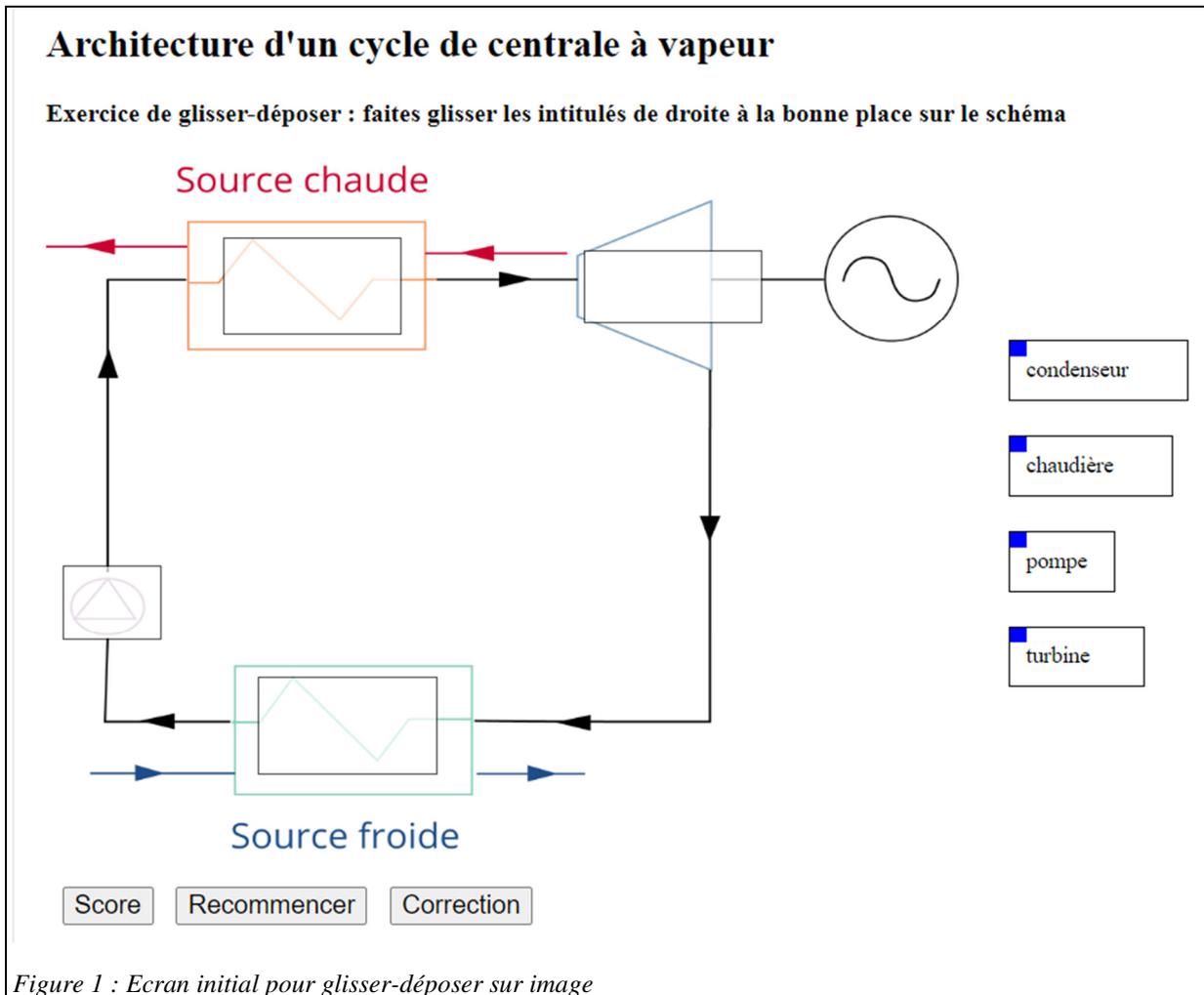


Figure 1 : Ecran initial pour glisser-déposer sur image

Le schéma de l'installation apparaît sur la gauche de l'écran, certains composants étant surmontés d'un rectangle semi-transparent définissant une zone réactive. Sur la droite de l'écran sont situés des éléments mobiles comportant les noms des différents composants, munis d'un petit rectangle bleu dans leur partie supérieure gauche.

Ces éléments peuvent être placés par glisser-déposer sur les zones sensibles de l'image de fond, le but de l'exercice étant d'apparier correctement ces zones et ces éléments mobiles.

Trois boutons permettent d'obtenir le score à tout moment, de recommencer l'exercice ou de connaître la correction.

La figure 2 montre le résultat obtenu lorsqu'une erreur a été commise, les positions de la pompe et de la turbine ayant été inversées.

## Architecture d'un cycle de centrale à vapeur

Exercice de glisser-déposer : faites glisser les intitulés de droite à la bonne place sur le schéma

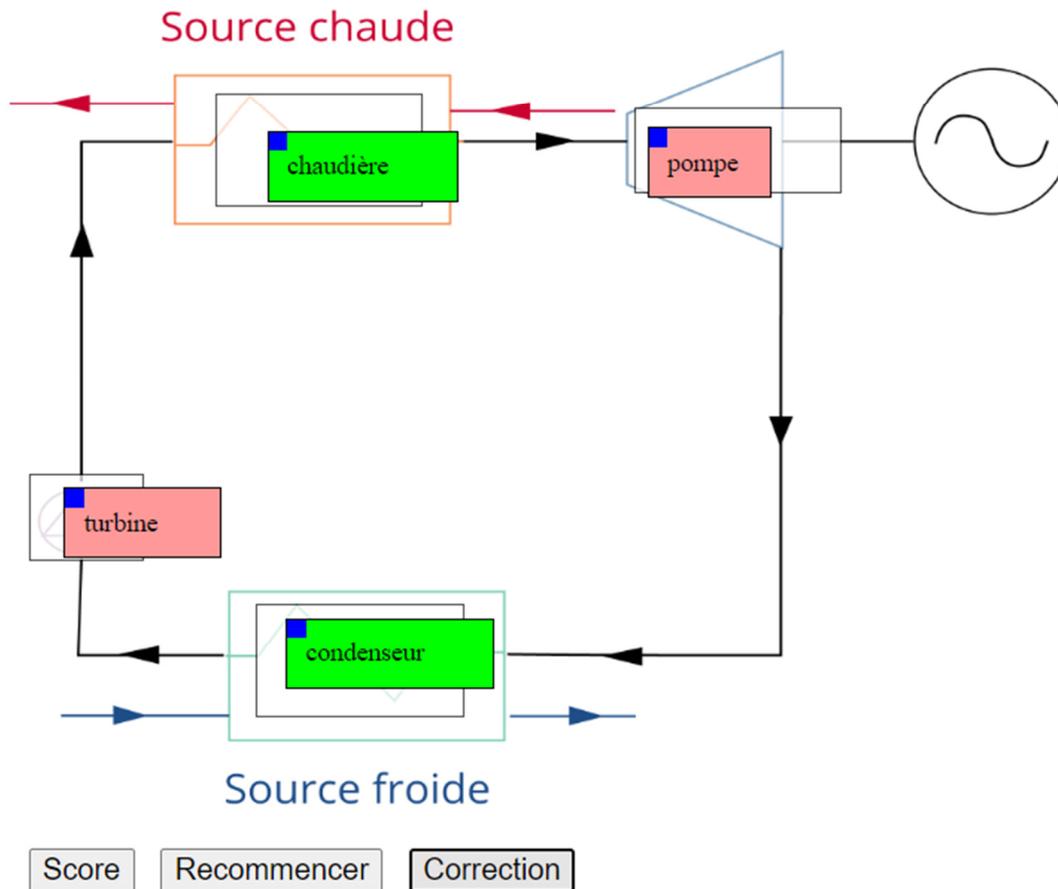


Figure 2 : Ecran avec réponse erronée

Pour que l'évaluation de l'appariement d'un composant et de son intitulé (label dans le code) soit effectué correctement, il faut que plus de la moitié de ce dernier soit situé dans la zone sensible (la cible, target dans le code), et que le petit carré bleu ne déborde pas vers le haut du cadre de la zone.

### Analyse du code html

La structure globale du fichier html est donnée figure 3. Pour pouvoir gérer le glisser-déposer, il a besoin des bibliothèques jQuery définies dans les scripts de la balise <head>

La balise <style>, ici repliée, contient le code CSS de la page, qui permet de définir son apparence, et notamment de positionner les targets.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>
7    <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
8    <link rel="stylesheet" href="https://code.jquery.com/ui/1.12.1/themes/smoothness/jquery-ui.css">
9
10   <style>
109  </head>
110  <body>
111
112  <div class="container">
37
38  <script>
35
36  </body>
37  </html>

```

Figure 3 : Structure du fichier html

La division container contient les différents éléments qui apparaissent à l'écran, et la balise <script> le code JavaScript qui gère le fonctionnement de la page.

Nous allons maintenant expliquer comment chacun de ces blocs est construit, sachant que seuls les premier et deuxième ont besoin d'être modifiés pour obtenir une nouvelle activité de glisser-déposer.

#### Balise <style>

C'est elle qui contient le code CSS du fichier. Les CSS (Cascading Style Sheets en anglais), ou « feuilles de style en cascade » contiennent le code utilisé pour mettre en forme une page web. Si vous souhaitez le modifier, excepté pour le positionnement des targets, il est préférable de commencer par vous initier à ce langage.

Les cibles sont définies, comme on le verra plus loin, par des identifiants obtenus en ajoutant leur numéro au mot target. Leurs positions sont définies dans ce bloc, comme indiqué figure 4.

Par exemple, le target 3 situé au-dessus de la pompe dans le schéma de la figure 1, est situé à 342 pixels du haut de la page, à 10 pixels de la gauche, sa largeur est de 60 pixels, et sa hauteur de 45 pixels. Pour les targets 1 et 4, il suffit de fournir leurs positions, les hauteurs et largeurs étant celles définies pour l'ensemble de targets un peu plus haut dans le code.

```

85
86  #target1 {
87    top: 135px;
88    left: 110px;
89  }
90
91  #target2 {
92    top: 143px;
93    left: 335px;
94    height: 45px;
95  }
96
97  #target3 {
98    top: 342px;
99    left: 10px;
100   width: 60px;
101   height: 45px;
102  }
103
104  #target4 {
105    top: 412px;
106    left: 132px;
107  }

```

Figure 4 : Positionnement des target

En pratique, pour trouver les bonnes valeurs de ces paramètres, la solution la plus simple est d'ouvrir le fichier html de deux manières :

- D'une part dans un éditeur du type NotePad++ qui est bien adapté pour cela
- D'autre part dans un navigateur Internet

Dans le navigateur, vous visualisez le résultat obtenu après avoir modifié dans NotePad++ les valeurs des paramètres et sauvegardé le fichier. En quelques minutes, vous obtenez le résultat désiré.

### Division container

C'est elle qui fournit les liens vers le fichier jpg ou png précise les noms des différents labels, comme le montre la figure 5.

```

111 <div class="container">
112   <div class="title">
113     <h2>Architecture d'un cycle de centrale à vapeur</h2>
114     <h4>Exercice de glisser-déposer : faites glisser les intitulés de droite à la bonne place sur le schéma</h4>
115   </div>
116
117   
118
119   <div class="label target4" id="label4" data-correct-target="target4">condenseur</div>
120   <div class="label target1" id="label1" data-correct-target="target1">chaudière</div>
121   <div class="label target3" id="label3" data-correct-target="target3">pompe</div>
122   <div class="label target2" id="label2" data-correct-target="target2">turbine</div>
123
124
125   <div class="target" id="target1"></div>
126   <div class="target" id="target2"></div>
127   <div class="target" id="target3"></div>
128   <div class="target" id="target4"></div>
129
130
131   <div id="buttons">
132     <button onclick="checkScore()">Score</button>
133     <button onclick="recommencer()">Recommencer</button>
134     <button onclick="corriger()">Correction</button>
135   </div>
136 </div>

```

Figure 5 : Division container

Les lignes 114 et 115 permettent d'indiquer le titre de la page et la consigne donnée aux apprenants.

La ligne 118 définit l'image de fond, ici placée dans un sous-dossier « img ».

Les lignes 120 à 123 permettent de paramétrer les labels, et les lignes 126 à 129 les targets.

Enfin, les lignes 131 à 136 définissent les boutons et les fonctions qui sont appelées lorsque l'on clique dessus.

Le lien entre les labels et les targets est défini dans les lignes 120 à 123. Considérons la première :

```
<div class="label target4" id="label4" data-correct-target="target4">condenseur</div>
```

L'attribut id indique que son identifiant est « label4 », et l'attribut data-correct-target que la cible correspondante est « target4 ». Son libellé est « condenseur ».

L'ordre dans lequel ces lignes sont placées dans le code est celui de leur apparition à l'écran, comme vous pouvez le constater en vous reportant à la figure 1.

Vous noterez aussi qu'aucun libellé n'est donné aux target. Si vous désirez en indiquer un, insérez-le juste avant </div> de la ligne du target. Cela peut notamment être utile pendant la phase de positionnement des target au-dessus de l'image, quitte à les supprimer une fois qu'ils sont bien placés.

Pour modifier le fichier html pour un nouvel exercice de glisser-déposer, il vous suffit de changer le lien vers l'image ligne 118, puis de renommer les intitulés des labels, et enfin de positionner les target.

Vous pouvez sans grande difficulté changer le nombre de targets et de labels en ajoutant ou supprimant des lignes et en ajoutant dans la balise <style> les paramètres de positionnement des nouvelles targets s'il y en a.

### *Balise <script>*

C'est elle qui contient le code JavaScript qui gère le fonctionnement de la page. Si vous souhaitez le modifier, sachez qu'il est préférable de bien maîtriser ce langage et faites ensuite très attention, faute de quoi des erreurs peuvent survenir.

### *Modification du fichier pour créer un nouvel exercice de glisser-déposer*

Récapitulons les éléments à modifier pour créer un nouvel exercice de glisser-déposer :

- 1) Commencez par préparer votre image de fond et mettez à jour le lien vers elle dans le fichier
- 2) Déterminez le nombre de couples (label, target) dont vous avez besoin et modifiez la division container en conséquence
- 3) Saisissez les libellés des différents labels
- 4) Changez le titre de l'activité et éventuellement la consigne si vous le désirez
- 5) Effectuez le positionnement des target en jouant sur le paramétrage de la balise <style>, en sauvegardant le fichier après chaque modification et en actualisant la page du navigateur où l'activité s'affiche
- 6) Si vous le souhaitez, enrichissez la présentation en jouant sur les CSS et en ajoutant des éléments dans la page

Une fois ces modifications effectuées, votre nouvelle activité devrait être opérationnelle.

## Exercices de catégorisation

Nous présentons maintenant les exercices de catégorisation parce que leur fichier html n'est qu'une variante de celui qui permet d'effectuer des glisser-déposer.

En effet, ce dernier est paramétré en indiquant qu'une relation bijective existe entre les targets et les labels, alors que dans les exercices de catégorisation les targets peuvent chacun recevoir plusieurs labels. De plus, il n'y a pas besoin d'image de fond d'écran.

### Interface graphique

Les catégories apparaissent sur la droite de l'écran de la figure 6. Sur la gauche sont situés des éléments mobiles comportant les noms des différents éléments à catégoriser, munis d'un petit rectangle bleu dans leur partie supérieure gauche.

Ces éléments peuvent être placés par glisser-déposer dans les grands rectangles situés sur la droite, le but de l'exercice étant de les ranger correctement.

Trois boutons permettent d'obtenir le score à tout moment, de recommencer l'exercice ou de connaître la correction.

**Fonctions d'état et fonctions de parcours**

Exercice de catégorisation : Faites glisser les intitulés de gauche dans les bonnes catégories

énergie interne  $u$

travail  $W$

pression  $P$

travail utile  $\tau$

enthalpie  $h$

température  $T$

chaleur  $Q$

Score Recommencer Correction

Fonctions d'état

Fonctions de parcours

*Figure 6 : Exercice de catégorisation*

La figure 7 montre le résultat obtenu lorsqu'une erreur a été commise, les positions du travail  $W$  et de la pression  $P$  ayant été inversées.

Pour que l'évaluation de l'appariement d'un composant et de son intitulé (label dans le code) soit effectué correctement, il faut que plus de la moitié de ce dernier soit situé dans la zone sensible (la cible, target dans le code), et que le petit carré bleu ne déborde pas vers le haut du cadre de la zone.

### Fonctions d'état et fonctions de parcours

Exercice de catégorisation : Faites glisser les intitulés de gauche dans les bonnes catégories

Score

Recommencer

Correction

Fonctions d'état

énergie interne u

travail W

température T

enthalpie h

Fonctions de parcours

travail utile  $\tau$

pression P

chaleur Q

*Figure 7 : Exercice de catégorisation avec erreurs*

#### Analyse du code html

La structure globale du fichier html est donnée figure 8. Pour pouvoir gérer le glisser-déposer, il a besoin des bibliothèques jQuery définies dans les scripts de la balise <head>

La balise <style>, ici repliée, contient le code CSS de la page, qui permet de définir son apparence, et notamment de positionner les target.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>
7    <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"></script>
8    <link rel="stylesheet" href="https://code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
9
10   <style>
135  </head>
136  <body>
137
138  <div class="container">
163
164  <script>
265
266  </body>
267  </html>

```

*Figure 8 : Structure du fichier html*

La division container contient les différents éléments qui apparaissent à l'écran, et la balise <script> le code JavaScript qui gère le fonctionnement de la page.

Nous allons maintenant expliquer comment chacun de ces blocs est construit, sachant que seul le deuxième a besoin d'être modifiés pour obtenir une nouvelle activité de catégorisation.

### Balise <style>

C'est elle qui contient le code CSS du fichier. Les CSS (Cascading Style Sheets en anglais), ou « feuilles de style en cascade ») contiennent le code utilisé pour mettre en forme une page web. Si vous souhaitez le modifier, il est préférable de commencer par vous initier à ce langage.

### Division container

C'est elle qui fournit les liens vers le fichier jpg ou png précise les noms des différents labels et targets, comme le montre la figure 9.

```

138 <div class="container">
139   <div class="title">
140     <h2>Fonctions d'état et fonctions de parcours</h2>
141     <h4>Exercice de catégorisation : Faites glisser les intitulés de gauche dans les bonnes catégories</h4>
142   </div>
143
144   <div class="label target1" id="label1" data-correct-target="target1">énergie interne u</div>
145   <div class="label target2" id="label2" data-correct-target="target2">travail W</div>
146   <div class="label target1" id="label3" data-correct-target="target1">pression P</div>
147   <div class="label target2" id="label4" data-correct-target="target2">travail utile r</div>
148   <div class="label target1" id="label5" data-correct-target="target1">enthalpie h</div>
149   <div class="label target1" id="label6" data-correct-target="target1">température T</div>
150   <div class="label target2" id="label7" data-correct-target="target2">chaleur q</div>
151
152   <div class="target" id="target1"><h3>Fonctions d'état</h3></div>
153   <div class="target" id="target2"><h3>Fonctions de parcours</h3></div>
154
155   <div id="score">Score: 0</div>
156
157   <div id="buttons">
158     <button onclick="checkScore()">Score</button>
159     <button onclick="recommencer()">Recommencer</button>
160     <button onclick="corriger()">Correction</button>
161   </div>
162 </div>

```

Figure 9 : Division container

Les lignes 140 et 141 permettent d'indiquer le titre de la page et la consigne donnée aux apprenants.

Les lignes 144 à 150 permettent de paramétrer les labels, et les lignes 152 et 153 les targets.

Enfin, les lignes 158 à 160 définissent les boutons et les fonctions qui sont appelées lorsque l'on clique dessus.

Le lien entre les labels et les targets est défini dans les lignes 144 à 150. Considérons la première :

```
<div class="label target1" id="label1" data-correct-target="target1">énergie interne u</div>
```

L'attribut id indique que son identifiant est « label1 », et l'attribut data-correct-target que la cible correspondante est « target1 ». Son libellé est « énergie interne u ».

L'ordre dans lequel ces lignes sont placées dans le code est celui de leur apparition à l'écran, comme vous pouvez le constater en vous reportant à la figure 6.

Vous noterez que les deux targets peuvent recevoir plusieurs labels, ce qui ne se faisait pas dans les exercices de glisser-déposer.

Pour modifier le fichier html pour un nouvel exercice de catégorisation, il vous suffit de renommer les intitulés des labels et des targets.

Vous pouvez sans grande difficulté changer le nombre de targets et de labels en ajoutant ou supprimant des lignes et en ajoutant dans la balise <style> les paramètres de positionnement des nouvelles targets s'il y en a.

#### *Balise <script>*

C'est elle qui contient le code JavaScript qui gère le fonctionnement de la page. Si vous souhaitez le modifier, sachez qu'il est préférable de bien maîtriser ce langage et faites ensuite très attention, faute de quoi des erreurs peuvent survenir.

#### *Modification du fichier pour créer un nouvel exercice de catégorisation*

Récapitulons les éléments à modifier pour créer un nouvel exercice de catégorisation :

- 1) Déterminez le nombre de labels et de targets dont vous avez besoin et modifiez la division container en conséquence
- 2) Saisissez les libellés des différents labels et targets
- 3) Changez le titre de l'activité et éventuellement la consigne si vous le désirez
- 4) Si vous le souhaitez, enrichissez la présentation en jouant sur les CSS et en ajoutant des éléments dans la page

Une fois ces modifications effectuées, votre nouvelle activité devrait être opérationnelle.

## Textes à trous

Pour spécifier les endroits où se situent les trous dans un texte, nous avons repris la solution implémentée dans le module H5P correspondant : les trous sont repérés en étant insérés entre deux « \* » , comme dans cet exemple : « Les systèmes fermés \*n'échangent pas\* de matière avec \*l'environnement\* , alors que les systèmes ouverts \*le font\* . »

Dans cette phrase, les trois trous sont : \*n'échangent pas\* , \*l'environnement\* , et \*le font\* .

## Interface graphique

L'écran du texte à trous est donné figure 10. Le texte s'affiche normalement, à la réserve près que des menus déroulants comportant la liste de l'ensemble des bouts de phrase manquants sont affichés à la place des trous. L'ordre du contenu de ces menus déroulants varie de manière aléatoire d'un trou à l'autre.

**Systèmes ouverts et fermés**

Texte à trous : Complétez les blancs à partir des choix proposés

Les systèmes fermés [l'environnement] de matière avec [en système ouvert] , alors que les systèmes ouverts [l'énergie interne u] .

Attention : la nature [l'enthalpie h] système [dépend des frontières] que vous choisissez pour le définir, ce qui peut induire de petites difficultés.

La plupart des composants impliqués dans les machines thermiques fonctionnent [l'enthalpie h] , tous étant traversés par [le font] .

Un système fermé échange [n'échangent pas le font] , du travail noté  $W$ , et la chaleur, notée  $Q$ . La fonction d'état représentant son énergie propre est [avec l'extérieur] .

Un système ouvert échange un travail utile avec l'extérieur, noté  $\tau$ , et la chaleur, notée  $Q$ . La fonction d'état représentant sa propre énergie est [n'échangent pas] .

Vérifier

Score: 0

Figure 10 : Texte à trous

Un bouton « Vérifier » permet d'obtenir le score à tout moment et de connaître la correction.

La figure 11 montre le résultat obtenu lorsque des erreurs ont été commises, les bonnes réponses apparaissant en vert et les mauvaises en rouge.

**Systèmes ouverts et fermés**

Texte à trous : Complétez les blancs à partir des choix proposés

Les systèmes fermés [n'échangent pas] de matière avec [l'environnement] , alors que les systèmes ouverts [le font] .

Attention : la nature ouverte ou fermée d'un système [dépend des frontières] que vous choisissez pour le définir, ce qui peut induire de petites difficultés.

La plupart des composants impliqués dans les machines thermiques fonctionnent [en système ouvert] , tous étant traversés par [des fluides] .

Un système fermé échange [avec l'extérieur] , du travail noté  $W$ , et la chaleur, notée  $Q$ . La fonction d'état représentant son énergie propre est [l'enthalpie h] .

Un système ouvert échange un travail utile avec l'extérieur, noté  $\tau$ , et la chaleur, notée  $Q$ . La fonction d'état représentant sa propre énergie est [l'énergie interne u] .

Vérifier

Veuillez vérifier vos réponses. Certains champs sont incorrects. Score: 7/9

Score: 7/9

Figure 11 : Texte à trous avec erreurs

## Analyse du code html

La structure globale du fichier html est donnée figure 12.

La balise `<style>`, ici repliée, contient le code CSS de la page, qui permet de définir son apparence.

```

1  <!doctype html>
2  <html lang="en">
3  <head>
4  <meta charset="UTF-8">
5  <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  <title>Texte à trous</title>
7  <style>
28 </head>
29 <body>
30 <div class="title">
31 <h2>Systèmes ouverts et fermés</h2>
32 <h4>Texte à trous : Complétez les blancs à partir des choix proposés</h4>
33 </div>
34 <div class="main">
35 <form id="gapFillForm">
36 <p>Les systèmes fermés <select class="gapSelect"></select> de matière avec <select class="gapSelect"></select>, alors que
37 <p>Attention : la nature ouverte ou fermée d'un système <select class="gapSelect"></select> que vous choisissez pour le d
38 <p>La plupart des composants impliqués dans les machines thermiques fonctionnent <select class="gapSelect"></select>, tou
39 <p>Un système fermé échange <select class="gapSelect"></select>, du travail noté W, et la chaleur, notée Q. La fonction d
40 <p>Un système ouvert échange un travail utile avec l'extérieur, noté  $\tau$ , et la chaleur, notée Q. La fonction d'état repré
41 <p></p><button type="button" onclick="verifier()">Vérifier</button>
42 </form>
43 <p id="result"></p>
44 <p id="score">Score: 0</p>
45 </div>
46 <script>
119 </body>
120 </html>

```

Figure 12 : Structure du fichier html

Les divisions « title » et « main » contiennent les différents éléments qui apparaissent à l'écran, et la balise `<script>` le code JavaScript qui gère le fonctionnement de la page.

### Balise `<style>`

C'est elle qui contient le code CSS du fichier. Les CSS (Cascading Style Sheets en anglais), ou « feuilles de style en cascade ») contiennent le code utilisé pour mettre en forme une page web. Si vous souhaitez le modifier, il est préférable de commencer par vous initier à ce langage.

### Division title

Les lignes 31 et 32 permettent d'indiquer le titre de la page et la consigne donnée aux apprenants.

### Division main

Les lignes 36 à 40 contiennent le texte et ses trous, ceux-ci étant signalés par :

```
<select class="gapSelect"></select>
```

Enfin, la ligne 41 définit le bouton et la fonction qui est appelée lorsque l'on clique dessus.

### Balise `<script>`

C'est elle qui contient le code JavaScript qui gère le fonctionnement de la page. Si vous souhaitez le modifier, sachez qu'il est préférable de bien maîtriser ce langage et faites ensuite très attention, faute de quoi des erreurs peuvent survenir.

```

55
56 // Ensemble des réponses exactes pour chaque espace vide
57 const reponsesExactes = [{"n'échangent pas"},
58 ["l'environnement"],
59 ["le font"],
60 ["dépend des frontières"],
61 ["en système ouvert"],
62 ["des fluides"],
63 ["avec l'extérieur"],
64 ["l'énergie interne u"],
65 ["l'enthalpie h"]
66 ];
67 total=reponsesExactes.length;
68
69 // Vérifier les réponses
70 reponsesExactes.forEach((exactes, index) => {
83
84 // Afficher le résultat
85 const toutesCorrectes = reponsesUtilisateur.every((reponse, index) => {
86   return reponsesExactes[index].includes(reponse);
87 });
88
89 const resultatMessage = toutesCorrectes
90   ? `Bravo! Toutes les réponses sont correctes. Score: ${score}/${total}`
91   : `Veuillez vérifier vos réponses. Certains champs sont incorrects. Score: ${score}/${total}`;
92
93 document.getElementById("result").textContent = resultatMessage;
94 document.getElementById("score").textContent = `Score: ${score}/${total}`;
95 }
96
97 // Préremplir les menus déroulants avec toutes les réponses possibles
98 const reponsesPossibles = [{"n'échangent pas","l'environnement","le font","dépend des frontières","en système ouvert","des flu
99 ];
100
101 const selectElements = document.getElementsByClassName("gapSelect");

```

Figure 13 : vue partielle du code JavaScript

La figure 13 montre une vue partielle du code JavaScript.

Les lignes 57 à 66 définissent le tableau `reponsesExactes` qui comme son nom l'indique contient l'ensemble des bonnes réponses, dans l'ordre d'apparition des trous.

Les lignes 98 et 99 montrent les mêmes éléments, mais sous une forme différente, rassemblés dans un tableau appelé `reponsesPossibles`, qui sert à remplir les menus déroulants de manière aléatoire.

Notez que tant la division « main » que ces parties du code JavaScript peuvent se déduire directement de l'énoncé de l'exercice sous forme de texte avec des « \* », mais que ce travail peut être assez laborieux s'il est fait à la main, avec des risques d'erreurs non négligeables.

C'est pourquoi nous avons mis au point un petit utilitaire Java permettant de facilement construire ces parties du fichier html tout en respectant la syntaxe requise.

## Constructeur de fichiers html pour textes à trous

### Utilisation

Ce constructeur est une application qui se présente sous forme d'une archive Java exécutable (fichier `GapFillHtmlBuilder_Fr.jar`) contenant toutes les bibliothèques dont elle a besoin.

Sous Windows et Mac, si elle ne se lance pas lorsque vous double-cliquez sur elle, c'est que Java n'est pas installé sur votre machine. Vous trouverez les explications pour l'installer à l'adresse suivante : <http://java.com/fr/download/index.jsp>

Sous Linux, entrez `java -jar GapFillHtmlBuilder_Fr.jar` en ligne de commande, ou créez un shell de lancement adapté à votre environnement.

Ce constructeur crée un nouveau fichier html en se contentant de modifier uniquement le texte à trous. Il conserve intégralement la balise `<style>` et les parties des autres balises qui ne dépendent pas du texte soumis. Il part donc d'un fichier existant qu'il recherche par défaut dans le dossier "distrib".

## Interface graphique

La figure 14 montre l'écran du constructeur après qu'on ait ouvert un fichier existant.

De haut en bas, il contient deux champs pour la définition du titre et de la consigne, une zone de texte supérieure permettant de saisir l'énoncé de l'exercice sous forme de texte avec des « \* » , et une zone de texte inférieure où apparaît le code html de l'exercice.

Enfin sont placés en bas de l'écran un bouton de génération du code html et un hyperlien permettant de vérifier la syntaxe du code si on le désire.

Le menu vous permet d'ouvrir un fichier, de sauvegarder celui avec lequel vous travaillez, d'ouvrir une fenêtre « A propos » et de quitter l'application.

Lorsque vous ouvrez un fichier html de texte à trous, son titre et sa consigne sont affichés en haut de l'écran, et l'ensemble du code html est placé dans la zone de texte inférieure.

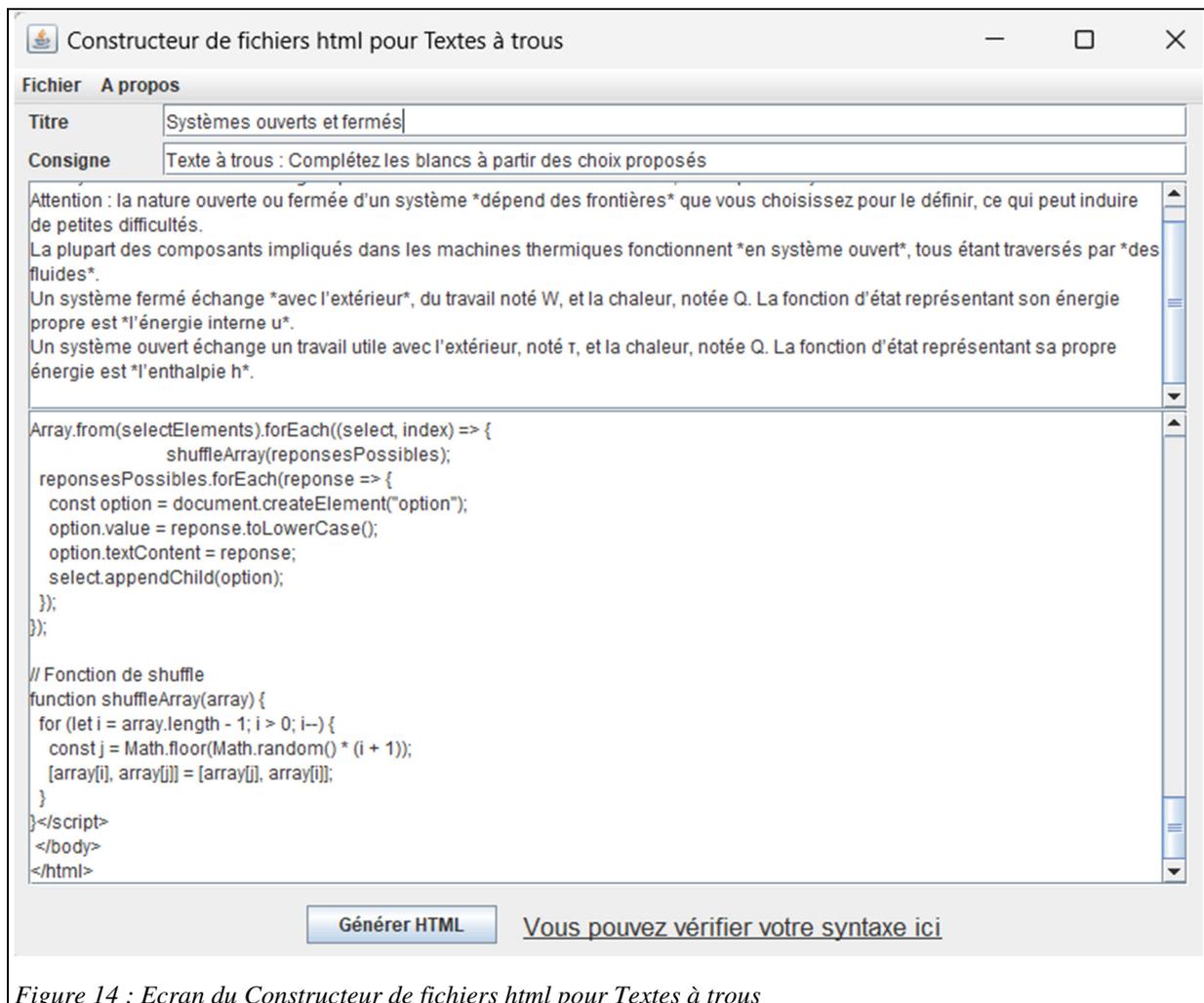


Figure 14 : Ecran du Constructeur de fichiers html pour Textes à trous

Il vous suffit alors de saisir dans la zone de texte supérieure l'énoncé de l'exercice sous forme de texte avec des « \* », de modifier le titre et la consigne, et vous pouvez générer le code html correspondant, que vous pouvez sauvegarder sous le nom que vous désirez. Vous pouvez aussi modifier vous-même à la main le code html pour ajouter ce que vous désirez à la page.

### Validation du code html

Si vous avez modifié à la main le code html, il est préférable d'effectuer une vérification de sa syntaxe avant de sauvegarder le fichier. Cela peut être fait en utilisant la page de Nu Html Checker vers laquelle pointe l'hyperlien situé en bas de la fenêtre d'édition.

Attention : supprimez la balise <script> du code soumis, sinon la procédure n'aboutira pas.

**Nu Html Checker**

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

**Showing results for contents of text-input area**

Checker Input

Show  source  outline  image report Options...

Check by text input  css

```

<p>Attention : la nature ouverte ou fermée d'un système <select class="gapSelect"></select> que vous choisissez pour le définir, ce
qui peut induire de petites difficultés.</p>
<p>La plupart des composants impliqués dans les machines thermiques fonctionnent <select class="gapSelect"></select>, tous étant
traversés par <select class="gapSelect"></select>.</p>
<p>Un système fermé échange <select class="gapSelect"></select>, du travail noté W, et la chaleur, notée Q. La fonction d'état
représentant son énergie propre est <select class="gapSelect"></select>.</p>
<p>Un système ouvert échange un travail utile avec l'extérieur, noté  $\tau$ , et la chaleur, notée Q. La fonction d'état représentant sa
propre énergie est <select class="gapSelect"></select>.</p>
<p></p><button type="button" onclick="verifier()">Vérifier</button>
</form>
<p id="result"></p>
<p id="score">Score: 0</p>
</div>
</body>
</html>

```

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

Message Filtering

Document checking completed. No errors or warnings to show.

*Figure 12 : Vérification du code html*