

BDW1 - TD 6

Programmation Web - PHP

UCBL - Département Informatique de Lyon 1 – 2018

L'objectif de ce TD est de vous familiariser avec la syntaxe HTML et PHP. Pour cela, nous allons nous mettre en place des fonctions qui vous permettront de construire l'espace de jeu utile au projet.

Nous considérons le fichier *mesFonctions.php* dont les premières lignes sont les suivantes :

```
<?php
\* Declaration des constantes *\
define( 'RIEN' , 'vide' );
define( 'DALLE', 'dalle' );
define( 'FIN' , 'carotte' );
define( 'TROU', 'trou' );

\* Declaration des variables globales *\
$dim = 5;           // dimension du plateau
$numCase = 0;      // compteur d'affectation d'un numero de dalle
$plateau = array(); // espace de jeu
$schemin = array(); /* coordonnees de dalles dans le plateau
                    correspondant au chemin */

...
?>
```

Une fois toutes les fonctionnalités implémentées, vous serez en mesure de générer un espace de jeu composé d'un chemin lui-même composé de dalles numérotées aboutissant à une dalle finale représentant une carotte. Trois chemins en spirale sont représentés sur la figure 1.

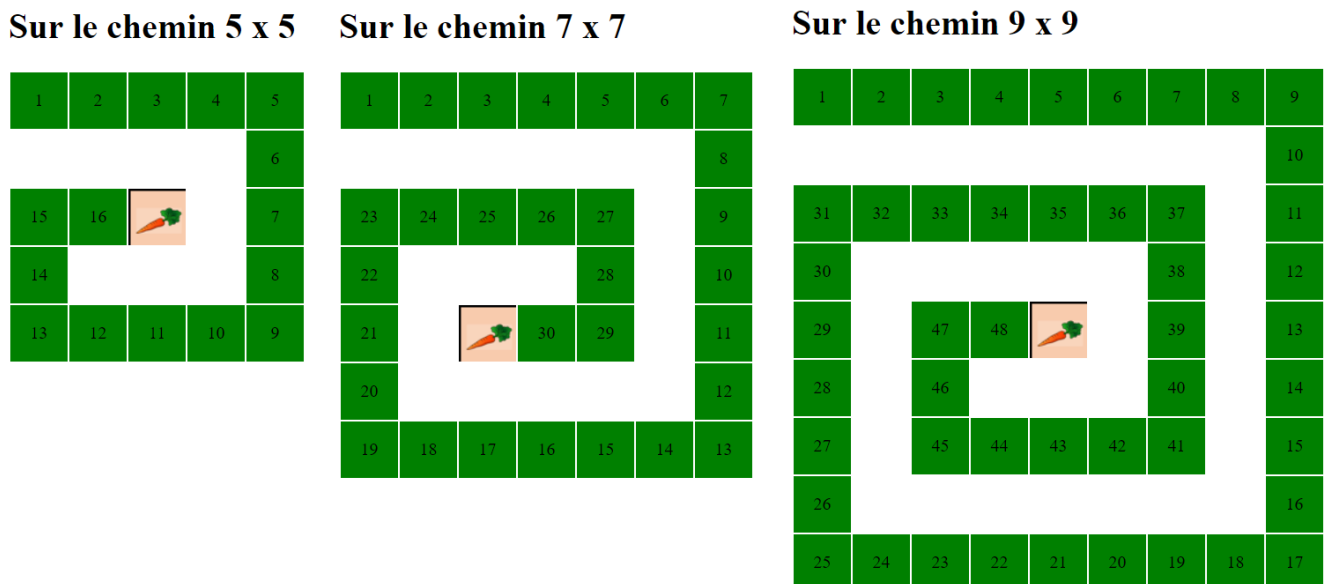


FIGURE 1 – Trois chemins pour des valeurs de \$dim valant 5, 7 et 9.

Exercice 1 : Construction d'un chemin en spirale

Écrire les fonctions PHP répondant aux besoins spécifiés dans les questions.

1. Définir la fonction *initPlateau()* qui initialise la variable globale *\$plateau* en un tableau à 2 dimensions de taille $\$dim \times \dim , rempli par défaut avec la valeur de la constante RIEN.
2. Définir le prédicat *estLibre(\$x,\$y)* qui retourne faux si les coordonnées $(\$x,\$y)$ ne correspondent pas à un élément du plateau ou si les coordonnées correspondent à une valeur dans le plateau qui n'est pas la valeur de la constante RIEN. Sinon la fonction retourne vrai.
3. Définir le prédicat *estAuBord(\$x,\$y,\$sens)* qui, étant donnée la cellule courante $(\$x,\$y)$ dans le plateau et le sens d'avancement *\$sens*, retourne vrai, :
 - si la valeur de *\$y* correspond à la dernière colonne du plateau, pour *\$sens* qui vaut 'est' ;
 - si la valeur de *\$x* correspond à la dernière ligne du plateau, pour *\$sens* qui vaut 'sud' ;
 - si la valeur de *\$y* correspond à la première colonne du plateau, pour *\$sens* qui vaut 'ouest' ;
 - si la valeur de *\$x* correspond à la première ligne du plateau, pour *\$sens* qui vaut 'nord' ;Sinon la fonction retourne faux.
4. Définir le prédicat *avancerEstPossible(\$x,\$y,\$sens)* qui, à partir de la cellule courante $(\$x,\$y)$, retourne vrai si les deux cases suivantes selon la direction *\$sens* sont libres ou si la case suivante (toujours selon la direction *\$sens*) est au bord du plateau et libre.
5. Définir la fonction *avance(\$x,\$y,\$sens)* qui, étant donnée la cellule courante $(\$x,\$y)$ dans le plateau, retourne un tableau associatif de clés 'x' et 'y' de valeurs respectives correspondantes aux coordonnées de la cellule :
 - à droite de $(\$x,\$y)$, pour *\$sens* qui vaut 'est' ;
 - au dessous de $(\$x,\$y)$, pour *\$sens* qui vaut 'sud' ;
 - à gauche de $(\$x,\$y)$, pour *\$sens* qui vaut 'ouest' ;
 - au dessus de $(\$x,\$y)$, pour *\$sens* qui vaut 'nord' ;
6. Définir la fonction *sensSuivant(\$sens)* qui retourne le point cardinal qui se trouve à tribord (droite) du point cardinal *\$sens*. Donc pour 'est' la fonction retournera 'sud', pour 'sud' la fonction retournera 'ouest'...
7. Définir le prédicat *peutTournerTribord(\$x,\$y,\$sens)* qui, étant donnée la cellule courante $(\$x,\$y)$ dans le plateau, retourne vrai s'il existe au moins deux cases libres adjacentes dans la direction se trouvant à tribord (par la droite) du sens de direction *\$sens*.
8. Définir la fonction *affecte(\$x,\$y,\$val)* qui :
 - si *\$val* correspond à la valeur de la constante 'DALLE', affecte :
 - les coordonnées $(\$x,\$y)$ dans le chemin *\$chemin* (La place de la dalle dans le chemin sera géré *via* *\$numCase*).
 - le numéro de case dans la case $(\$x,\$y)$ du plateau.
 - sinon, affecte la valeur *\$val* aux coordonnées $(\$x,\$y)$ dans le plateau
9. Définir la fonction *chemine(\$x,\$y,\$sens)* qui permet de construire un chemin à travers le plateau selon le principe : "A partir de la cellule initiale $(\$x,\$y)$, on avance dans la direction *\$sens* tant que possible. Si ce n'est plus possible, on cherche toujours à tourner à tribord (droite). Quand il n'est plus possible d'avancer et de tourner à tribord, la dernière dalle du chemin est marquée dans le plateau par la valeur de la constante 'FIN'".

Exercice 2 : Affichage d'un chemin en spirale

1. Définir la fonction *printChemin()* qui à partir du plateau construit un tableau HTML où chaque cellule du plateau correspond à une case du tableau. En fonction de la valeur de la cellule du plateau, vous affecterez à la case la classe correspondant à la valeur de la cellule. Attention, dans le cas où la valeur de la cellule est numérique, il faudra affecter la classe 'dalle'.
2. Définir un fichier *monStyle.css*. Pour chaque valeur de constante définie dans le fichier *mesFonctions.php*, déclarer un sélecteur de classe. Au sélecteur de cellule de tableau, associer une hauteur et une largeur de 50px. A chaque sélecteur de classe, associer un fond comme suit :
 - fond blanc pour le sélecteur *.vide*
 - fond vert pour le sélecteur *.dalle*
 - fond avec image "carotte.png" pour le sélecteur *.carotte*
 - fond noir pour le sélecteur *.trou*
3. Définir le fichier *monJeu.php* qui permet l'affichage d'un chemin en spirale. Ce fichier référencera le fichier *monStyle.css* et inclura le fichier *mesFonctions.php*.

Exercice 3 : Affichage de trou dans le chemin

1. Dans le fichier *mesFonctions.php*, définir la fonction *placeTrou(\$case)* qui permet d'affecter la valeur de la constante 'TROU' dans le plateau aux coordonnées correspondant au numéro de case \$case.

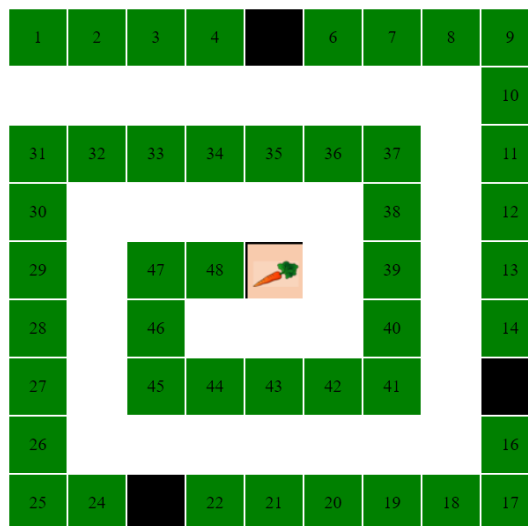


FIGURE 2 – Chemin 9x9 avec des trous aux cases 5, 15, 23