WEB@CADEMIE
We Are Coders

W1 – Piscine PHP

Colle 2

Responsables Pédagogiques

pedagowac@epitech.eu

Sommaire

Détails administratifs

- Le projet est à réaliser seul.
- Les sources doivent être rendues avec BLIH.
- Répertoire de rendu : **Piscine_PHP_Colle_02**

Introduction

Lors des colles, vous devez travailler seul sur un projet plus long et plus complexe que ce que vous voyez au cours des « jours » de piscine.

Restrictions

- Aucune communication entre étudiants n'est tolérée.
- Vous ne pouvez sortir de la salle qu'au bout de la moitié du temps maximum imparti à la colle.
- Toute sortie est définitive.

Conseil : Bien que l'usage d'internet soit autorisé, il est fortement déconseillé de l'interroger afin de trouver une solution au problème posé.
En effet, ce problème est unique et vous ne trouverez pas de solutions toute faite sur internet.
Vous avez seulement besoin de votre cerveau, de courage, et des fonctions PHP pour résoudre cet exercice !

Etape 1

Faire une fonction « laby » dont le prototype est le suivant :

Prototype: bool laby(string $str);

Cette fonction devra prendre en paramètre une chaine de caractère composée de « 0 », de « 1 » ou de « 2 » et devra générer visuellement une map en se basant sur cette chaine.

Ainsi si on a :

```
$str = "
20002111111111111111111
20002000200000020000002
20002111200020002000 20002
20000000000020002000 20002
21112000211120002111 20002
20000000200000000000002
21112000211111112000 21112
20000000000020000000002
21111111200021112000 20002
20000000000020002000 20002
20002111111120002111 20002
20000000200000000000002
21111111200021111111112
";
```

Alors :

```
laby($str)
```

doit afficher :

```
|   |--------------------
|   |   |       |       |
|   |---|   |   |   |   |
|           |   |   |   |
|---|   |---|   |---|   |
|       |               |
|---|   |-------|   |---|
|           |           |
|-------|   |---|   |   |
|           |   |   |   |
|   |-------|   |---|   |
|       |               |
|-------|   |-----------|
```

En cas d'erreur, si la fonction est appelée sans paramètres ou avec des paramètres invalides, la fonction doit retourner « FALSE ». Dans le cas contraire la fonction doit retourner « TRUE ».

Checkpoint : Attention, il est inutile de poursuivre si vous n'êtes pas arrivé correctement à ce stade. En cas de doute, vérifiez à nouveau votre travail.

Vous pouvez à présent choisir entre l'étape 2 et l'étape 3 pour continuer cette colle.

Etape 2

Créez une fonction « generateMap » dont le prototype est le suivant :

Prototype: mixed generateMap(int $x, int $y [, int $exit_count]);

Cette fonction devra retourner une chaine de caractères que vous pourrez ensuite passer en paramètre à votre précédente fonction « laby » pour générer un labyrinthe. Ce labyrinthe devra être de composé $x caractères en largeur et de $y caractères en hauteur et doit être tout autant conforme visuellement que celui fourni en exemple à l'étape 1.

La chaine générée doit être différente à chaque appel. Par conséquent vous devrez judicieusement faire appel à l'aléatoire pour remplir au mieux cette consigne.

Si le paramètre « exit_count » est renseigné, votre labyrinthe devra comporter $exit_count sorties (ou entrées), et aucune sortie si le paramètre n'est pas renseigné ou s'il vaut 0.

En cas d'erreur, si les paramètres obligatoires ne sont pas renseignés ou si les paramètres sont invalides la fonction devra retourner NULL.

Ainsi `laby(generateMap(24, 42, 2));` doit afficher un labyrinthe de 42 lignes de 24 caractères et comporter 2 sorties.

Vous serez d'autant mieux noté que le labyrinthe que vous aurez généré grâce à l'aléatoire est réalisable
(cases de la bonne largeur, pas de mur côte à côte, une sortie possible etc…)

Etape 3

Indices : stdin

Maintenant que vous êtes parvenu à afficher un labyrinthe il faudrait maintenant pouvoir se déplacer dedans !

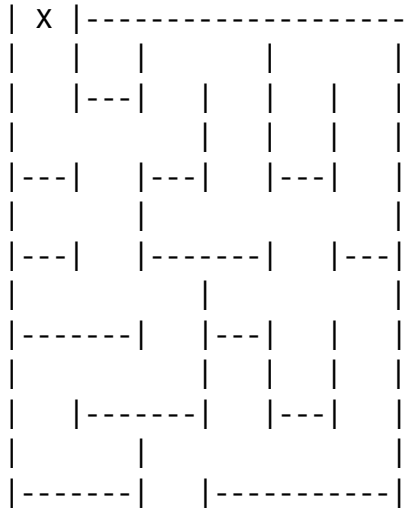
Faites en sorte qu'après que votre labyrinthe soit affiché, vous puissiez déplacer une croix « X » dans votre labyrinthe à l'aide des touches Z, Q, S et D. Evidemment Z vous permet d'aller en haut, Q à gauche, D à droite et S en bas.

La croix pourra avoir la position que vous souhaitez initialement, cependant, la croix ne peut que prendre la place du caractère « espace ». Par conséquent, il ne doit pas être possible de sortir du labyrinthe ou de placer la croix sur les murs !

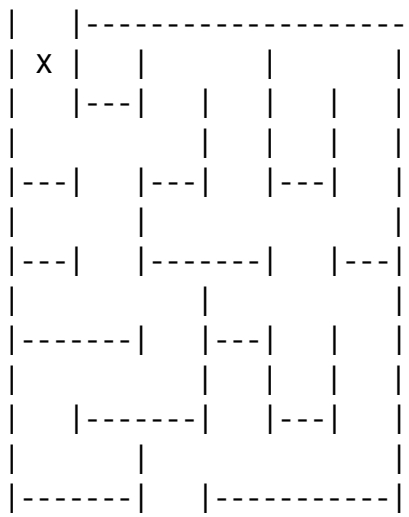
Idéalement, il faut qu'à chaque déplacement, vous « effaciez » l'ancien labyrinthe pour en afficher un nouveau avec la position de la croix à jour. Afficher le contenu du fichier « clear » déjà présent dans votre dossier « rendu » vous permettra de « clear » votre écran.

Exemple :

Position de départ



Après avoir appuyé sur S



etc…

Bonne chance !