

Exercice 3 : Wumpus

Introduction

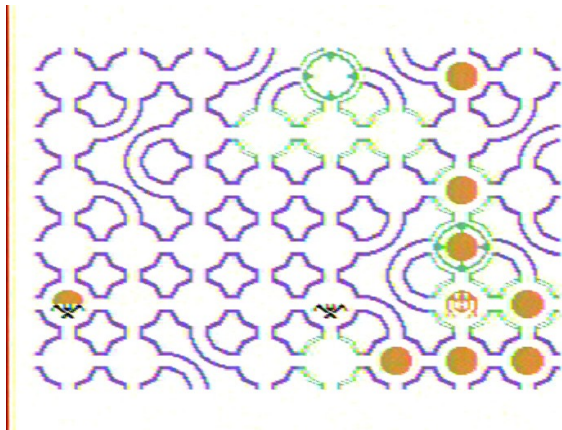


Le jeu Hunt the Wumpus est un jeu qui a eu deux formes : une forme en mode texte et une forme graphique qui était sur TI99/4A.

Le joueur s'aventure dans un réseau de cavernes, armé d'un arc et d'une seule flèche,

pour tuer le terrible Wumpus.

Le plateau de jeu est sur une grille de 8 sur 6 bouclant sur elle-même (si on sort par la gauche, on rentre par la droite du côté opposé ou si on sort par le bas, on rentre par le côté haut opposé). Dans cette caverne, on trouve non seulement le Wumpus, mais aussi des chauve souris géantes et des puits de slime mortels.



Les quatre cases autour des puits de slime sont couvertes de mousse (verts). Le Wumpus a laissé des traces autour de son repère, sur une distance de deux cases.

Si on dérange une chauve souris, elle nous transporte à un endroit au hasard dans le plateau de jeu. Si on arrive sur le Wumpus ou sur un puit, on est mort.

Au début, seule la case où on se trouve est révélée, les autres cases ne sont pas encore visibles. Au fur et

à mesure que l'on se déplace, les cases deviennent visibles les unes après les autres, permettant de repérer les puits de slime et le Wumpus.

Quand on pense avoir trouvé le Wumpus, on doit se rendre sur une case à côté de lui, demander pour tirer et indiquer la direction. Si on a bien touché le Wumpus, on a gagné la partie. Si on l'a raté, on a perdu.

Le niveau est généré au hasard selon un niveau de difficulté. Dans le jeu d'origine, le niveau de difficulté changeait uniquement le nombre de chemins tournants. Ici, on changera également le nombre de chauve-souris et de puits de mousse.

Niveau	Puits	Tournants ¹	Chauve-Souris
Très Facile	1	0	1
Facile	2	4	2
Moyen	2	8	2
Difficile	2	12	2

¹ Dans le jeu d'origine, le nombre de tournants est plus élevé mais cela demande de s'assurer qu'aucune case n'est inaccessible.


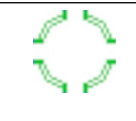

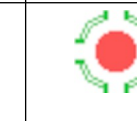
Developpement : Etape 1

La première version se fera sans les chemins tournants qui amènent des difficultés supplémentaires au niveau programmation.

Génération du Labyrinthe

La page Start demandera le niveau de difficulté. Elle amènera sur une page d'initialisation qui se chargera des opérations suivantes :

- créer le tableau de 6 lignes et 8 colonnes contenant des cases normales
- tirer au hasard la position du ou des puits. Pour chaque puit, changer la valeur de la case du tableau de jeu pour indiquer le puit de slime.
- pour chaque puit, changer la valeur des 4 cases voisines pour indiquer qu'elles doivent être affichées en vert
- tirer au hasard la position du Wumpus, il ne peut pas se trouver sur un puit de slime. Changer la valeur de la case.
- pour chaque case autour du Wumpus, changer la valeur de la case et des 4 cases adjacentes pour indiquer que le Wumpus est proche. Si la case est Wumpus, on ne la change pas, si elle est normale, on met la case avec le point rouge et si elle est verte, on met la case verte avec le point rouge.

Case.png	CaseG.png	CaseR.png	CaseGR.png
			

- tirer au hasard la position des Chauve-Souris. Elles ne peuvent pas être sur un puit de slime ou sur le Wumpus. Les positions seront sauvées dans un tableau
- tirer au hasard la position du joueur. Il doit se trouver sur une case normale et ne peut pas être sur la case d'une chauve souris.

Une fois le labyrinthe généré, on passera à l'affichage du jeu.

Affichage du Labyrinthe (1)

La première version de l'affichage affichera le labyrinthe complet. Les cases mousse et Wumpus seront laissées vides (pas d'images prévues pour ces cases vu que le joueur n'est pas sensé pouvoir les visiter (game over)).

On affichera un tableau de 6 lignes et 8 colonnes. Dans chaque case, on affichera l'image correspondant en image de fond d'une division de 128x128px.

On ajoutera ensuite les images du personnage et des Chauve-Souris. Elles seront affichées sous la forme d'images dans les divisions présentes dans les cases de la table. Pour les positionner correctement, on utilisera une propriété text-align : center ; sur les divisions et une propriété margin-top sur les images.

Pour les cases vides, on disposera d'un style sans image de fond mais toujours avec les dimensions de 128x128px.

Déplacement (1)

A côté du labyrinthe, on prévoira 4 boutons ou liens pour les 4 directions. Les scripts PHP correspondant changeront la position du personnage et reviendront à l'affichage.

Si on arrive sur une Chauve-Souris, il y a 1 chance sur 4 que cette dernière se réveille et nous envoie sur une autre case. Si cela se passe, on déplace la Chauve-Souris en-dehors de l'écran de jeu et on choisit une nouvelle position pour le joueur au hasard (pas sur une chauve-souris, un puit de Slime ou le Wumpus).

Si on arrive sur un puit ou le Wumpus, on affiche l'écran de Game Over

Affichage du Labyrinthe (2)

Dans l'initialisation, on crée un 2eme tableau qui indiquera les cases visitées. La case du joueur sera marquée comme visitée.

Au niveau de l'affichage, on n'affichera le contenu de la case que si elle a été visitée (image de fond ET chauve-souris).

Au niveau du déplacement, on marquera la case où on arrive comme visitée. Dans le cas où on réveille une chauve-souris, la nouvelle case sera aussi marquée comme visitée.

Tir et victoire

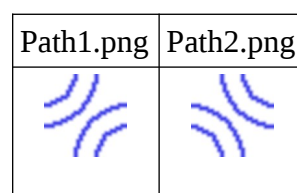
On ajoutera une case à cocher (check-box) pour indiquer que l'on désire tirer.

La fonction de déplacement détectera si on a demandé pour tirer. Si c'est le cas, on calcule la position de la destination et si c'est le Wumpus, le joueur a gagné sinon, il a perdu.

Développement : Etape 2

Une fois la partie précédente fonctionnel, on ajoutera les chemins coudés. Quand un joueur est sur une telle case, il faudra savoir selon quelle direction il se déplace. On aura donc besoin d'une valeur qui indiquera si on est sur le chemin Haut-Droit/Haut-Gauche ou sur le chemin Bas-Droit/Bas-Gauche (selon la case).

Dans l'initialisation, après le remplissage du tableau de cases vides et avant de placer les puits de slime et le Wumpus, on placera les chemins coudés au hasard.



Pour placer le Wumpus ou les puits de Slime, on s'assurera que la case tirée au hasard n'est pas un chemin coudé.

Pour trouver les cases voisines, quand on examine une direction, si on est sur un chemin coudé, on suit le chemin jusqu'à la case suivante et ainsi de suite jusqu'à arriver jusqu'à une case normale.

Les chauve-souris et le joueur ne commenceront pas non plus sur des cases coudées.

Au niveau affichage du personnage, on peut (pas obligatoire mais préférable) décaler l'image vers le haut ou vers le bas quand on est sur un chemin coudé selon le côté où on se trouve²

Au niveau du déplacement, quand on arrive sur un chemin coudé, une variable devra être mise à jour pour indiquer dans quelle moitié on se trouve.

Au niveau tir, la flèche devra suivre le chemin jusqu'à atteindre une case normale avant de tester si on a trouvé le Wumpus ou non.

Conseils

1) Développer les deux premiers points (initialisation et affichage du labyrinthe) en parallèle : commencer par juste générer le labyrinthe sans rien d'autre (pas de puits, de Wumpus ou de chauves-souris) et l'afficher. Ensuite, ajouter les Chauves-Souris et le Joueur. Puis ajouter les puits de slime et les cases vertes autour. Et terminer en ajoutant le Wumpus et les cases avec la marque rouge. A chaque étape, vérifier que l'affichage est correct avant de passer à la suite.

2) Un tableau à deux dimensions en PHP est un tableau de tableaux : créer le tableau principal, faire une boucle pour créer les lignes, pour chaque ligne, mettre un tableau vide et faire une boucle pour mettre les éléments de la ligne. On peut accéder directement à une case en utilisant `$a[$ligne][$col]`.

3) Le programme est relativement long, découper la programmation en plusieurs étapes et vérifier ces étapes au fur et à mesure aide à éviter les erreurs

2 On peut aussi décaler vers la gauche ou la droite mais cela est plus difficile à gérer