

Défi du mois ! Battleship - Simulateur de bataille navale

Embarquez un défi épique : **concevez un simulateur de bataille navale en Java ou python**. Ce classique mettra à l'épreuve vos capacités en programmation et de résolution de problème ! Seriez-vous en mesure de recréer la tension des mers et coulera les navires ennemis avant qu'il ne soit trop tard ?

Quel est l'objectif du défi ?

Coder le célèbre jeu de bataille navale tout en respectant les règles traditionnelles, où un joueur est contre l'ordinateur et place ses navires afin de s'affronter à tour de rôle. Le gagnant est le premier à couler tous les navires adverses.

Règles

- Chaque joueur a une grille de 10x10 cases. Une case peut contenir de l'eau [~], un navire [O], un coup touché [X], ou un coup manqué [-].
- Voici la composition générale des navires, vous pouvez augmenter le nombre de navire dans votre jeu :
 - o 1 croiseur : 3 cases
 - o 1 porte-avions : 5 cases
 - o 1 destroyer : 2 cases
 - o 1 sous-marin : 3 cases
 - o 1 cuirassé : 4 cases
- Les navires doivent être placés horizontalement ou verticalement sans se chevaucher et sans dépasser les limites de la grille.
- Le joueur et l'ordinateur s'attaquent tour par tour en désignant une case à attaquer (par ex. A5).
- Si la case attaquée contient une partie d'un navire, ``Touché`` est déclaré. Sinon, c'est ``À l'eau``. Lorsqu'un navire est entièrement touché, c'est ``Coulé``.
- Le premier joueur à couler tous les navires de son adversaire est gagnant, cet événement marque la fin de la partie.

Exemple du gameplay

Phase initiale : L'ordinateur place ses navires aléatoirement. Le joueur place ses navires ``A1 V (Vertical)`` , ``B4 H (Horizontal)`` , etc.

Phase d'attaque : Le joueur attaque ``B5``. Le résultat est ``Touché`` ou ``À l'eau``. L'ordinateur attaque ``C3``, le résultat est ``Touché`` ou ``À l'eau``.

La fin de jeu est déclaré quand le joueur ou l'ordinateur a coulé tous les navires de son adversaire.

Conseils

- Implémentez les fonctionnalités nécessaires tels que : le placement des navires en début de partie, le tour par tour, l'indication des coups touchés, manqués et navires coulés.
- Affichez vos navires, les coups manqués et les parties touchés de vos navires. Affichez les coups touchés ou manqués de la grille de l'adversaire (sans révéler leurs navires).
- Commentez et organisez votre code.
- Les coups de l'ordinateur (votre adversaire) sont totalement aléatoires.

Récompense

Le gagnant ou la gagnante recevra une carte-cadeau d'une valeur de 50 \$. Pour départager les participants, nous utiliserons les critères d'évaluation suivants :

- Précision et respect des contraintes
- Originalité et lisibilité du code
- Performance

Assurez que votre jeu soit :

- **Fonctionnel** : *Mon jeu respecte-t-il les règles ? Peut-on faire une partie complète sans bug ?*
- **Ergonomique** : *Mon jeu est-il facile à comprendre et à jouer ?*

Des points supplémentaires seront attribués pour l'originalité, voici quelques idées :

- *Jouer sur une interface graphique au lieu de la console (par ex. Java Swing ou Tkinter pour python)*
- *Intégrer un algorithme qui permet à l'ordinateur de faire des coups stratégiques en se basant sur les coups précédents.*
- *Mode multijoueur, personnalisation des grilles et navires, Etc.*

Date limite de soumission

Les solutions doivent être soumises avant le mois de mars. Veuillez soumettre votre fichier source et un court document Word expliquant votre réponse avec des captures d'écran.

UQODE, club de programmation de l'UQO

Sosane Mahamoud Houssein (hous44@uqo.ca)

