

Choix de modélisation

La génération du plateau

Nous avons décidé de générer un plateau vierge dans un premier temps (c'est-à-dire juste de l'eau et du sable) pour ensuite rajouter un bateau dans la première case (de sable) et des rochers aléatoirement (selon le pourcentage de rocher choisi) sur le reste du plateau. Pour vérifier que tout fonctionne et que toutes les cases sont accessibles nous avons une méthode « `verifierIle()` » qui va comme son nom l'indique vérifier que toutes les rochers sont accessibles. Dans cette méthode, nous initialisons un tableau de `int` à 0 (qui représente le fait que la case n'a pas été visitée) sauf le bateau qui lui est initialisé à 1 (point de départ). Ensuite vient le parcours du plateau. Pour chaque case nous regardons si les voisins de celle-ci sont traversable (en gros si c'est un rocher ou non). Si c'est le cas, la case voisine traversable prend la valeur 1 sinon 2. Ainsi, après parcours complet du plateau, s'il reste des cases contenant la valeur 0 (qui n'a donc pas été visitée) nous régénérons un nouveau plateau jusqu'à ce que le plateau soit validé par la méthode.

Nous avons choisi cette méthode de génération car c'est la seule que nous avons réussis à mettre en place.

La façon de penser de l'intelligence artificielle

La façon dont nous avons codé notre intelligence artificielle est la suivante. Elle se base sur la chance. Tout d'abord, on regarde si le navire est plein. Si c'est le cas on choisit un personnage au hasard dans le bateau et on le débarque. Si le bateau n'est pas plein, l'intelligence artificielle a 3 chances sur 5 de déplacer (aléatoirement selon les positions accessibles) un personnage du plateau. Elle fera aussi 2 chances sur 5 de faire une action sur un rocher ou un autre personnage selon le type de personnage sélectionné.

L'interface graphique

Nous avons choisi de mettre des `JTextArea` pour afficher les informations sur la partie en cours. Chaque équipe a sa propre zone de texte. Cette zone de texte affiche la vie des différents personnages présents sur le plateau ainsi que l'équipe qui doit jouer. Pour le déplacement, nous avons mis des `JButton` qui représentent les différents points cardinaux pour effectuer des déplacements plus facilement que sur notre ancienne version dans laquelle le déplacement se faisait avec des `JOptionPane`. Les différents menus se parcourent à l'aide de `JButton`. Les paramètres de début de partie (nombre de personnage, taille du plateau, pourcentage de rocher) se font à l'aide de `JSlider` pour une meilleure ergonomie.